STRUCTURAL SOVEREIGNTY AND EXECUTION-BOUND GOVERNANCE

From Context-Bound Authority to Non-Bypassable Enforcement

A Federated Architecture for Governable AI-Supported Systems

Alexanja Senke

Independent Researcher - Interlink Bridge

Execution Governance Infrastructure Architect

AI Cognition Strategist - Human-AI Co-Intelligence Architect

Reconstructed Research Monograph v2.0

Public-safe academic review draft

6 August 2026

Declaration of Authorship and Status

This monograph presents an integrated independent research contribution in artificial-intelligence systems architecture and structural governance theory. It reconstructs and substantially rewrites material from the earlier *Structural Sovereignty and Execution-Bound Governance* source corpus together with subsequently developed work on Interlink Bridge, SLI, LIAN, PoA, KWI, Kyber, SGDS, FSEL/RCFA, runtime governance, evidence maturity, and hardware-enforced admissibility boundaries.

The research is attributed to Alexanja Senke except where another source is cited. The work distinguishes four kinds of contribution:

1. **Architectural contributions** - definitions, dependency structures, invariants, interfaces, and claim boundaries.
2. **Formal contributions** - typed transition systems, permit predicates, non-reachability conditions, and theorem candidates under explicit assumptions.
3. **Engineering contributions** - bounded software implementations, deployment packages, runbooks, test kits, and hardware architecture specifications.
4. **Evidence contributions** - reproducible internal tests, negative results, live-field evidence where completed, and explicit open gates.

No protected mechanism is used to widen a public claim. Omission of implementation-sensitive detail must never be interpreted as proof that an implementation exists, is secure, is complete, or has passed independent validation.

Disclosure Boundary

This edition is organized under three disclosure layers.

Public theory. Definitions, architecture, theorem forms, assumptions, non-claims, validation logic, public-safe diagrams, and the research programme.

Controlled technical evidence. Detailed implementation, test vectors, fault traces, deployment-specific configuration, internal record schemas, and evidence packages suitable for controlled or NDA review.

Protected internal mechanisms. Credentials, exploit paths, bypass-sensitive details, protected signal mappings, thresholds, calibration values, proprietary integration logic, customer-specific material, and security-relevant operational data.

Only public theory and bounded public-safe evidence summaries are included here. Controlled and protected material is referenced by class, maturity, and artifact identity without exposing the protected payload.

Abstract

Artificial-intelligence systems increasingly generate proposals that can cross software, organizational, institutional, and physical boundaries. Contemporary governance commonly evaluates those proposals through model alignment, monitoring, content classification, documentation, human oversight, risk management, and post-event audit. These mechanisms are useful, but they remain structurally incomplete whenever a non-permitted consequence-bearing transition is still technically reachable.

This monograph develops **Structural Sovereignty** as an execution-bound theory of governability. The central thesis is that governance becomes operational only when the existence of a consequence path depends on a complete, current, and evidence-bound permit. The main invariant is

$$\mathrm{Exec}_B(\tau) \Longrightarrow \Pi_B^*(\tau),$$

where τ is a typed transition, B is a declared enforcement boundary, and Π_B^* is a federated permit combining the mandatory conditions for formal admissibility, control feasibility, authority, runtime mediation, physical enforcement where required, current context, valid version and timing, fault-envelope compliance, evidence readiness, and remedy.

The contribution is federated rather than reductionist. Mathematics does not create legal or institutional authority. Human presence is not a committed decision. A policy does not enforce itself. A model is not the governance layer. A hardware signal does not establish the legitimacy of its upstream authority. A stored log does not prove that an effect was preventable. Each discipline retains its own validity conditions while exporting typed evidence and obligations to adjacent layers.

The monograph formalizes the governance-execution gap; defines the federated permit and claim architecture; develops governed transition and dynamical-system models; separates proposal, permission, commit, and consequence; specifies context-bound authority and the Human Commit Boundary; describes KWI LIAN Kyber as a bounded upstream governance implementation; connects that implementation to runtime and CoChip enforcement; introduces the SGDS-FSEL-RCFA live-evidence route; and establishes a falsification-first maturity system from M0 conceptual work to M6 institutional determination.

The system-level theory remains M1 specified. Selected KWI LIAN Kyber components possess bounded M2 implementation and M3 internal-test evidence. The SGDS-FSEL field kit is complete and the real KWI-LIAN campaign is in active evidence acquisition as of this edition, without a frozen terminal C1-C5 result. The CoChip is specified as a public-safe architecture, but no production silicon or complete end-to-end independent validation is claimed.

Executive Determination

The core determination of this monograph is:

> Governance is not established by the existence of a rule, a model output, a dashboard, a review step, a log, or a compliance statement. Governance becomes operational only when the protected consequence cannot occur without the complete valid permit required for that exact transition and boundary.

This position produces six immediate consequences:

- advisory representation is not enforcement;
- authentication is not authority;
- human presence is not human commit;
- evidence of an event is not evidence that the event was preventable;
- component validity does not compose automatically;
- system maturity cannot exceed the weakest mandatory dependency.

Reader Orientation

The monograph is arranged as a single 12-chapter argument rather than a federation of 101 short template chapters.

1. Research Object, Claim Boundary, and Governance-Execution Gap
2. Structural Sovereignty and the Federated Permit
3. Authority, Human Commit, and Institutional Legitimacy
4. Formal State, Transition, Non-Reachability, and Composition
5. Governed Dynamics: Stability, Load, Delay, and Safe-State Reachability
6. Runtime Mediation and Governed Execution
7. Physical Enforcement and the Governed CoChip
8. KWI LIAN Kyber as an Implemented Upstream Governance System
9. SGDS-FSEL-RCFA and the Live-Evidence Route
10. Governance, Law, Data, Procurement, and Operator Sovereignty
11. Validation, Falsification, Reproduction, and Evidence Maturity
12. Original Contributions, Limitations, and Research Programme

The appendices contain the common axiom register, public-safe terminology, an evidence and maturity ledger, and the open research programme.

Claim Markers and Evidence Maturity

Claims are typed as follows:

Marker	Meaning
DEF	definition
AX	adopted architectural axiom
PROP	derived proposition
THEOREM	deductive statement under explicit assumptions
INV	candidate invariant
ENG	engineering requirement
EMP	empirical claim
REG	regulatory or normative mapping requiring current authoritative verification
CASE	bounded case-study evidence
HYP	hypothesis
OPEN	unresolved research obligation

Evidence maturity is represented by:

Level	Meaning
M0	conceptual idea or unstructured claim
M1	specified definitions, scope, assumptions, and obligations
M2	bounded implementation exists
M3	repeatable internal test evidence exists
M4	independently reproduced by a competent separate party
M5	independently assessed against a declared framework
M6	formally determined by a competent institution

A system-level claim cannot exceed the maturity of its weakest mandatory dependency:

$$M(C_{\text{system}}) \leq \min_{i \in D_{\text{mandatory}}} M(E_i).$$

Core Axioms

1. If a transition is inadmissible, its execution path does not exist inside the declared enforcement boundary.
2. A model is not the governance layer and may not originate its own execution authority.
3. Unknown mandatory state is not ordinary permission.
4. Architecture precedes an operational compliance claim.
5. Safety does not imply liveness, and halt does not automatically imply safe-state arrival.
6. Component validity does not compose automatically across interfaces, time, authority, failure, or deployment change.
7. Protected detail does not widen public claim scope.
8. Failed, inconclusive, and unresolved evidence remains part of the authoritative record.

Contents

Chapter 1

Research Object, Claim Boundary, and the Governance-Execution Gap

1.1 Purpose and Research Question

Artificial-intelligence systems increasingly produce outputs that may affect documents, administrative processes, software tools, external services, physical devices, organizational decisions, and other consequence-bearing environments.

The central problem addressed here is not whether an AI-generated output is plausible, useful, well explained, aligned with a policy, or reviewed after creation. These properties may be valuable, but none alone determines whether the corresponding transition should be capable of producing an effect.

The research question is:

> Under which architectural conditions can a consequence-bearing transition occur only when all mandatory conditions of admissibility, authority, feasibility, execution control, current context, and evidence are valid?

The core invariant is

$$\boxed{\mathrm{Exec}_B(\tau) \implies \Pi^*_B(\tau)} \tag{1.1}$$

or equivalently

$$\boxed{\neg \Pi^*_B(\tau) \implies \neg\, \mathrm{Exec}_B(\tau)}. \tag{1.2}$$

Here τ is a typed consequence-bearing transition, B is the explicitly declared enforcement boundary, $\mathrm{Exec}_B(\tau)$ means that the transition reaches its protected consequence inside that boundary, and $\Pi^*_B(\tau)$ is the complete permit required for that exact transition.

The claim is boundary-relative. It does not assert that every possible consequence in an open world can be controlled. It asserts that an operational governance claim is admissible only when the declared boundary, consequence paths, mandatory dependencies, assumptions, and remaining bypass conditions are explicit.

1.2 From Output Governance to Transition Governance

Many governance approaches focus on outputs: accuracy, prohibited content, instruction following, human review, logging, explanation, or risk classification. These questions concern output quality and traceability. They do not by themselves establish whether the output can create a consequence.

This monograph therefore distinguishes

$$\text{output assessment} \neq \text{transition authority} \neq \text{execution control}.$$

A recommendation is not a permission. A permission is not a committed decision. A committed decision is not yet a physical or organizational consequence. A log of an event is not proof that the event was preventable. A model output is not the source of authority over its own execution.

The governed object is a typed transition

$$\tau = (z_t, u_t, \alpha_t, o_t, c_t, v_t, f_t, z_{t+1}, \kappa)\,, \tag{1.3}$$

where

- z_t is the pre-transition state;
- u_t is the proposal, command, or requested action;
- α_t is actor and authority context;
- o_t is the object affected;
- c_t is operational, institutional, and case context;
- v_t is the relevant version, time, epoch, and configuration;
- f_t is the declared fault and disturbance state;
- z_{t+1} is the candidate successor state;
- κ is the protected consequence class.

The transition identity must remain stable across proposal, review, permit, commit, execution, and evidence creation. A permission for one object is not a permission for another. A review of one document state does not authorize a later modified state. A proof about an abstract model does not automatically establish a property of a deployed implementation. This is the **non-substitution principle**.

1.3 The Declared Enforcement Boundary

Every execution-bound claim requires a boundary

$$B = (\mathcal{C}_B, \mathcal{I}_B, \mathcal{A}_B, \mathcal{F}_B, \mathcal{T}_B)\,, \tag{1.4}$$

where $\mathcal{C}_B$ is the set of protected consequences, $\mathcal{I}_B$ the interfaces through which those consequences may be reached, $\mathcal{A}_B$ the actors, services, devices, and administrative paths in scope, $\mathcal{F}_B$ the declared fault and adversarial envelope, and $\mathcal{T}_B$ the timing and freshness limits.

The boundary must include the last point from which the protected consequence can still be prevented. An upstream classifier may contribute information. A policy engine may issue a decision. A user interface

may display a warning. A dashboard may represent a state. None constitutes operative control unless the relevant consequence path depends on that state.

The boundary is incomplete when an alternative path remains, including a secondary API, administrative override, unmediated tool, recovery mode, stale cached permit, direct actuator interface, supplier-controlled dependency, reset path restoring prior authority, post-permit mutation, or unrecorded manual transfer. A complete path inventory is therefore part of the governed object.

1.4 Definition of the Governance-Execution Gap

Let $\mathcal{T}_B$ be the set of consequence-bearing transitions inside the declared boundary. Let $\mathrm{Reach}_B(\tau) = 1$ mean that at least one executable path can realize τ.

The **Governance-Execution Gap** is

$$\boxed{\mathcal{G}_B = \{\tau \in \mathcal{T}_B \mid \Pi_B^*(\tau) \neq 1 \wedge \mathrm{Reach}_B(\tau) = 1\}\,.} \tag{1.5}$$

A boundary is execution-closed for the declared transition class only if

$$\boxed{\mathcal{G}_B = \emptyset.} \tag{1.6}$$

This separates descriptive governance, detective governance, and execution-bound governance.

- **Descriptive governance** states what should occur but does not control the path.
- **Detective governance** detects, records, or reports an inadmissible event, potentially after irreversibility.
- **Execution-bound governance** conditions reachability of the protected consequence on the complete permit.

Descriptive and detective mechanisms may support learning, audit, accountability, and incident response. They must not be represented as prevention while the consequence path remains open.

1.5 The Federated Permit as a Claim Object

For a declared boundary B, let J_B be the mandatory permit predicates. Then

$$\Pi_B^*(\tau) = \bigwedge_{j \in J_B} \pi_j(\tau). \tag{1.7}$$

The admissible predicate set must be derived from the protected consequence and boundary, not selected retrospectively to fit a desired outcome. Unknown mandatory state does not default to allow.

Every material claim is represented as

$$C_i = \langle q_i, \Omega_i, B_i, A_i, D_i, E_i, M_i, F_i, R_i, \Delta_i \rangle\,, \tag{1.8}$$

where q_i is the proposition, Ω_i its scope, B_i its boundary, A_i its assumptions, D_i its dependencies, E_i its evidence package, M_i its maturity, F_i its falsifiers, R_i its result state, and Δ_i its disclosure class.

Permitted result states are `SUPPORTED`, `FAILED`, `INCONCLUSIVE`, `OPEN`, and `WITHHELD`. `WITHHELD` describes a disclosure condition; it is not evidence of truth.

1.6 Current Evidence Posture

At the time of this edition, the research programme contains different maturity levels that must remain separate.

Research object	Current bounded status
Structural Sovereignty and federated permit	M1 specified architecture
KWI LIAN Kyber bounded software components	M2/M3 for declared implemented and internally tested routes
Writer/Calc document-state verification	internally demonstrated for the declared VERIFIED/MISMATCH workflow
SGDS-FSEL field instrumentation	field kit completed; declared synthetic route tested
Real KWI-LIAN C1-C5 campaign	active evidence acquisition; terminal result not yet frozen
Governed CoChip architecture	M1 specified public-safe architecture
Physical FPGA/ASIC enforcement evidence	open
Complete upstream-to-downstream system	not independently reproduced end to end

The presence of M2 or M3 evidence in selected components does not promote the complete federated architecture to M3.

1.7 Falsification Conditions

The theory is defeated or narrowed whenever an in-scope counter-construction remains reachable. Primary falsifiers include:

1. a policy is represented as enforcement while an alternative consequence path remains;
2. the model is allowed to generate or expand its own authority;
3. a permit remains valid after object, version, case, actor, or evidence changes;
4. a direct, administrative, recovery, debug, or supplier path bypasses mediation;
5. a formal proof is transferred to an implementation without correspondence evidence;
6. control feasibility or safe-state reachability is assumed rather than established;
7. halt is represented as safe-state arrival without physical evidence;
8. evidence is detached from the state, version, boundary, and configuration tested;
9. an indeterminate result is reported as a pass;
10. component properties are assumed to compose automatically;
11. synthetic evidence is transferred to a real candidate;
12. protected disclosure is used to conceal missing implementation or failed evidence.

Where a falsifier survives, the permissible result is a narrower claim, failed invariant, unresolved dependency, or open obligation. Narrative confidence does not repair a broken execution path.

1.8 Chapter Propositions

Proposition 1.1 - Reachability condition. A governance mechanism becomes operational for consequence class $\mathcal{C}_B$ only if the reachability of every protected transition in that class depends on the mechanism's authoritative state.

Proposition 1.2 - Authority separation. Model capability, model confidence, output plausibility, authentication, or nominal human presence does not by itself establish execution authority.

Proposition 1.3 - Non-substitution. A result established for one object, version, context, boundary, model, implementation, or fault envelope cannot be substituted for another without an explicit correspondence argument.

Proposition 1.4 - Weakest mandatory dependency. The maturity and scope of a composite governance claim cannot exceed those of its weakest necessary dependency.

Proposition 1.5 - Authoritative negative evidence. Failed, indeterminate, stale, mismatched, and unresolved results remain part of the authoritative record and may not be removed merely because they prevent promotion.

Proposition 1.6 - Disclosure non-expansion. Protected or undisclosed technical detail may reduce public exposure but cannot increase the maturity, scope, or validity of a public claim.

1.9 Status

STRUCTURE. The research object, execution gap, boundary-relative invariant, claim tuple, and maturity structure are specified.

EXECUTION REALITY. Component-specific implementation and internal evidence exist in bounded parts of the programme. Complete end-to-end closure remains open.

FAILURE POINT. Any surviving consequence path that does not depend on the complete mandatory permit narrows or defeats the corresponding claim.

GOVERNANCE STATUS. M1 for the integrated theory. Higher maturity is admitted only for individually identified and evidence-bound components.

CONFIDENCE TYPE. Definitional and architectural; deductive only under explicit assumptions; empirical only where artifacts and test results are identified.

Chapter 2

Structural Sovereignty and the Federated Permit

2.1 Structural Sovereignty

Structural Sovereignty is the capacity of a system, operator, or institution to determine, within a declared boundary and without discretionary external permission at the decisive moment, which consequence-bearing transitions are reachable.

This definition is deliberately stronger than data ownership, local hosting, access control, policy compliance, human oversight, or technical capability considered separately. Each may be necessary in a particular domain, but none by itself closes the governance-execution gap.

Let $\mathcal{T}_B$ be the declared transition class. Structural sovereignty over $\mathcal{T}_B$ requires:

1. a bounded consequence and path inventory;
2. explicit representation of mandatory admissibility and authority conditions;
3. complete mediation of the in-scope execution path;
4. invalidation under stale identity, context, version, timing, or evidence;
5. fail-closed treatment of unknown mandatory state;
6. durable evidence of the evaluated transition and result;
7. a remedy path for correction, suspension, withdrawal, appeal, or safe-state transition.

A system that depends on discretionary vendor permission for a critical action, recovery, update, license check, key release, or operational continuation does not possess full structural sovereignty over that function. It may still be well managed, contractually protected, or highly available, but the decisive execution authority remains externally dependent.

2.2 Permit Construction

The federated permit is not a single scalar produced by one subsystem. It is a conjunction of typed predicates contributed by different domains:

$$\Pi^*_B(\tau) = \pi_{\text{formal}} \wedge \pi_{\text{control}} \wedge \pi_{\text{authority}} \wedge \pi_{\text{runtime}} \wedge \pi_{\text{physical}} \wedge \pi_{\text{context}} \wedge \pi_{\text{version}} \wedge \pi_{\text{fault}} \wedge \pi_{\text{evidence}} \wedge \pi_{\text{remedy}} \tag{2.1}$$

Only the predicates mandatory for the declared boundary belong to J_B. A low-consequence drafting workflow may not require a hardware predicate. An actuator-bound automotive transition does. A purely local reversible edit may require different remedy and timing predicates than an irreversible administrative or physical effect.

The selection rule is:

$$J_B = \text{MinReq}(\mathcal{C}_B, \mathcal{I}_B, \mathcal{F}_B, \mathcal{T}_B), \tag{2.2}$$

where MinReq returns the smallest predicate set sufficient to close all declared in-scope paths under the stated fault envelope. This guards against both under-specification and decorative overloading.

2.3 Predicate Semantics

2.3.1 Formal admissibility

$\pi_{\text{formal}}(\tau) = 1$ only when the proposed transition is permitted by the declared formal model and invariant set. This is a model-internal property until correspondence with the implementation is established.

2.3.2 Control feasibility

$\pi_{\text{control}}(\tau) = 1$ only when the transition and any required safe-state response are feasible under the plant model, disturbance set, actuator limits, delay, and operating envelope.

2.3.3 Authority

$\pi_{\text{authority}}(\tau) = 1$ only when a valid principal, role, mandate, scope, time, case, and commit relation exists. Authentication proves identity possession or session access; it does not by itself prove authority for the exact transition.

2.3.4 Runtime mediation

$\pi_{\text{runtime}}(\tau) = 1$ only when all in-scope software execution routes are mediated, transaction identity is stable, post-permit mutation is prevented or detected, and crash/retry behavior cannot restore consumed or invalid authority.

2.3.5 Physical enforcement

$\pi_{\text{physical}}(\tau) = 1$ only when the declared physical or irreversible effect path depends on a non-bypassable enforcement state, including power, clock, reset, I/O, communication, and debug conditions within the fault envelope.

2.3.6 Current context and version

π_{context} and π_{version} require current actor, device, case, object, evidence, configuration, epoch, and freshness. A permit is not transferable across silent mutation.

2.3.7 Evidence

$\pi_{\text{evidence}}(\tau) = 1$ means that the evaluated state, result, version, boundary, and relevant artifacts are reconstructable. Evidence readiness supports accountability and later verification; it does not substitute for mediation or physical prevention.

2.3.8 Remedy

$\pi_{\text{remedy}}(\tau) = 1$ requires a defined response to failure, disagreement, invalidation, or harm. Depending on the domain, remedy may include delay, review, veto, rollback, suspension, appeal, recall, safe-state transition, or formal reauthorization.

2.4 Mandatory-State Logic

Every mandatory predicate has at least three epistemic states:

$$\pi_j(\tau) \in \{\text{TRUE}, \text{FALSE}, \text{UNKNOWN}\}.$$

The permit rule is

$$\Pi_B^*(\tau) = 1 \iff \forall j \in J_B : \pi_j(\tau) = \text{TRUE}. \tag{2.3}$$

FALSE prohibits progression. UNKNOWN does not become ordinary permission. The governance response may be `DELAY`, `REVIEW`, `THROTTLE`, `REJECT`, or `HALT`, but silent allow is inadmissible.

This is a structural distinction from probabilistic confidence scoring. A model may assign high confidence to an output while authority, case assignment, document state, or evidence is missing. The permit remains invalid because those predicates are not model-confidence predicates.

2.5 Dependency Graph and Weakest-Link Bound

The permit is a dependency graph rather than a flat checklist. Let $D = (V_D, E_D)$ be the directed graph of mandatory dependencies. A claim node is promotable only when every upstream node required by its proof path is valid at the claimed scope and maturity.

For a composite claim C,

$$M(C) \leq \min_{d \in \text{Ancestors}(C)} M(d). \tag{2.4}$$

A high-maturity component cannot promote an untested interface. A verified document hash does not prove human authority. A complete authority record does not prove hardware non-bypassability. A tested gate does not prove that every effect path passes through it.

2.6 Sovereignty versus Common Substitutes

Substitute	What it can establish	Why it is insufficient alone
Local hosting	local execution and storage	does not prove authority, path closure, or evidence validity
Data ownership	custody or contractual control	does not prevent an unauthorized transition
Human oversight	human participation	may be nominal, late, overloaded, or detached from commit
Policy	normative intent	does not enforce the consequence path
Logging	later reconstruction	may record an event that was never preventable
Model alignment	reduced output risk	does not source institutional authority
Access control	entry restriction	may not bind object, case, version, or effect
Hardware root of trust	trusted boot or key material	does not establish upstream legitimacy
Certification	bounded third-party determination	cannot exceed the certified object, version, and scope

Structural Sovereignty is therefore a relation among architecture, authority, execution, evidence, and remedy, not a label attached to one technology.

2.7 External Permission and Operational Sovereignty

Let e be a critical transition and p_{ext} an external discretionary permission. If

$$\text{Exec}_B(e) \Longrightarrow p_{\text{ext}}$$

and the operator cannot independently satisfy, replace, or recover that permission inside the declared boundary, then structural sovereignty over e is incomplete.

This condition is relevant to cloud-only processing, remote license dependence, external key custody, proprietary update channels, inaccessible evidence stores, non-exportable state, and vendor-controlled recovery. It does not imply that all external services are unacceptable. It requires that dependency be represented honestly in the permit and claim envelope.

2.8 Propositions

Proposition 2.1 - Predicate non-substitution. No mandatory permit predicate may be inferred solely from another predicate of a different validity domain.

Proposition 2.2 - Unknown-state closure. A federated permit is invalid whenever any mandatory predicate is false or unresolved.

Proposition 2.3 - Boundary-specific minimality. The correct permit set is the smallest set sufficient to close every declared consequence path under the stated fault and timing envelope.

Proposition 2.4 - External-dependency limit. A system does not possess full structural sovereignty over a critical transition if execution or recovery requires discretionary external permission that the operator cannot independently replace or control.

Proposition 2.5 - Evidence non-substitution. Evidence that a transition occurred or was reviewed does not prove that the transition was preventable.

2.9 Status

STRUCTURE. Structural Sovereignty and the federated permit are defined as boundary-relative dependency objects.

EXECUTION REALITY. Predicate implementations differ by domain. KWI LIAN supplies bounded upstream context and evidence predicates; runtime and CoChip work address downstream mediation and enforcement.

FAILURE POINT. The claim fails if the predicate set omits a necessary path condition, if one predicate is inferred from an unrelated domain, or if external discretionary permission remains hidden.

GOVERNANCE STATUS. M1 theory; component-specific higher maturity only where identified.

CONFIDENCE TYPE. Definitional and architectural.

Chapter 3

Authority, Human Commit, and Institutional Legitimacy

3.1 Authority Cannot Originate from Model Output

A model can generate text, predictions, rankings, classifications, plans, and action proposals. It cannot, by the content or confidence of its own output, prove that the correct actor is present, the correct device is active, the correct case exists, the current document state is valid, the required evidence is attached, or a competent human or institution has authorized progression.

The architecture therefore adopts:

$$\boxed{\text{model output} \not\Rightarrow \text{authority.}} \tag{3.1}$$

Authentication is also insufficient. Authentication may show that a credential, device, or session is recognized. Authority is a typed relation among principal, role, mandate, object, action, scope, time, context, and revocation state.

Define

$$\text{Auth}(p, r, m, o, a, s, t, c, \rho), \tag{3.2}$$

where p is principal, r role, m mandate, o object, a action, s scope, t time, c operational context, and ρ revocation or lifecycle state. Authority for a transition exists only when the exact typed relation is valid.

A once-authenticated user may no longer be present. A role may not cover the current case. A mandate may have expired. A document may have changed after review. A device may have entered an invalid lifecycle state. A prior commit may refer to a different object version.

3.2 Context-Bound Authority

The upstream authority context is represented as

$$\alpha_t = (p_t, r_t, d_t, c_t, o_t, e_t, h_t, \nu_t), \tag{3.3}$$

where d_t is device/session state, e_t evidence context, h_t human commit state, and ν_t version and freshness.

A public-safe authority derivation chain is:

```
actor context
  -> role and mandate
  -> device and session presence
  -> case or workflow assignment
  -> current document/object state
  -> evidence state
  -> Human Commit Boundary
  -> governance state
  -> downstream enforcement signal
```

The order matters. Authority does not emerge from a valid model output. It emerges from operational conditions external to the model and bound to the decision point.

3.3 Human-in-the-Loop Is Too Weak

The phrase "human in the loop" may mean that a person was somewhere in the process, clicked a button, read a summary, had theoretical oversight, or could have intervened. These meanings are not equivalent.

For high-consequence work, the relevant question is:

> Which human committed which exact state under which actor, role, device, case, evidence, version, and time conditions?

This requires a boundary, not a vague presence claim.

3.4 The Human Commit Boundary

The **Human Commit Boundary** is the point at which a human decision becomes structurally attached to a specific state and may authorize progression.

A canonical workflow is

```
generated or assisted draft
  -> working draft
  -> reviewed state
  -> verified state
  -> human commit
  -> sealed/finalized state
  -> downstream action or archive
```

The commit object is

$$H = (p, r, o, v, c, e, d, t, a, \sigma)\,, \tag{3.4}$$

where p is principal, r role, o object, v object version or hash, c case context, e evidence root, d device/session, t time, a action class, and σ commit evidence or signature.

A valid human commit is object-specific and version-specific. Post-commit mutation invalidates the prior state unless a new commit is created. The architecture therefore separates review, commit, seal, and finalization rather than collapsing them into a single generic approval.

3.5 Presence, Competence, and Intervention Margin

Human authority requires more than identity. The person must have sufficient presence, competence, information, time, and practical ability to intervene.

Let

$$T_{\text{sense}} + T_{\text{understand}} + T_{\text{decide}} + T_{\text{act}} < T_{\text{irreversible}}. \tag{3.5}$$

If this inequality cannot hold, nominal oversight does not provide an effective intervention boundary. Human authority may still exist institutionally, but it does not provide timely execution control for that transition.

A display is not equivalent to understanding. A click is not equivalent to informed commitment. A theoretical veto is not equivalent to a reachable veto path. Human factors therefore enter both the authority predicate and the timing envelope.

3.6 Delegation, Handoff, and Continuity

Delegation must preserve:

- delegating principal and authority source;
- receiving principal and role;
- object and case scope;
- action classes permitted;
- start and expiry time;
- evidence state at transfer;
- revocation conditions;
- unresolved obligations;
- continuity of audit and responsibility.

Define a delegation relation

$$\mathcal{D} = \langle p_{\text{from}}, p_{\text{to}}, r, s, o, c, t_0, t_1, e, \rho \rangle. \tag{3.6}$$

Delegation transports authority only within the declared scope. It does not create authority that the delegating principal did not possess.

Handoff is also a state-transfer problem. The receiving person must not inherit a narrative summary detached from the authoritative object, evidence, and unresolved state. KWI LIAN addresses this by preserving case, document, evidence, device, event, commit, and shadow-state continuity.

3.7 Conflict, Veto, and Corrective Authority

Multiple principals may disagree. The system must represent conflict explicitly rather than silently selecting the highest-confidence or fastest path.

Possible result states include:

- `ALLOW` - all mandatory authority predicates hold;
- `DELAY` - required context or review is temporarily incomplete;
- `REVIEW_REQUIRED` - competent secondary review is mandatory;
- `VETO` - an authorized veto prevents progression;
- `HALT` - execution must stop under the declared boundary;
- `APPEAL` - a separate corrective authority path is activated.

The conflict-resolution mechanism must itself have authority, scope, timing, and evidence. A model-generated tie-breaker cannot create institutional legitimacy.

3.8 Responsibility and Evidence

Responsibility is not inferred merely because a person was logged in. It requires a traceable relation among action, state, authority, time, and consequence.

A responsibility record should answer:

1. who acted or committed;
2. under which role and mandate;
3. on which exact object and version;
4. in which case and device context;
5. using which evidence;
6. for which action class;
7. with which result;
8. whether the state later drifted;
9. whether authority was revoked, delegated, or expired;
10. which remedy remained available.

This is stronger than a generic audit log because it binds evidence to the transition identity.

3.9 Propositions

Proposition 3.1 - Model non-authority. No model-generated output, confidence score, or recommendation establishes authority for its own execution.

Proposition 3.2 - Commit specificity. A human commit authorizes only the exact object, version, action class, context, and evidence state to which it is bound.

Proposition 3.3 - Presence non-equivalence. Authentication or nominal presence does not establish timely, competent, and informed human oversight.

Proposition 3.4 - Delegation conservation. Delegation may transport but cannot expand authority beyond the delegator's valid scope.

Proposition 3.5 - Responsibility binding. Responsibility evidence requires a stable relation among principal, authority, object state, action, time, evidence, and consequence.

3.10 Status

STRUCTURE. Context-bound authority, Human Commit Boundary, delegation, handoff, and responsibility relations are specified.

EXECUTION REALITY. KWI LIAN Kyber implements bounded actor/device/case/document/evidence/commit structures and is discussed in Chapter 8.

FAILURE POINT. Authority collapses when a model grants its own permission, when post-commit mutation is ignored, when presence is mistaken for commit, or when delegation loses scope and revocation.

GOVERNANCE STATUS. M1 for the general authority theory; M2/M3 for declared bounded KWI LIAN workflows.

CONFIDENCE TYPE. Architectural and component-evidence dependent.

Chapter 4

Formal State, Transition, Non-Reachability, and Composition

4.1 Governed Transition Systems

A formal governance claim must begin with an explicit state and transition object. Let a governed transition system be

$$\mathfrak{G} = (Z, U, \rightarrow, O, \mathcal{A}, \mathcal{R}, \mathcal{P}, \mathcal{E}, \mathcal{V}, Z_0, \mathcal{B}, \mathcal{F}), \tag{4.1}$$

where:

- Z is the typed system-state space;
- U is the set of proposed actions or controls;
- $\rightarrow \subseteq Z \times U \times Z$ is the candidate transition relation;
- O is the observation structure;
- $\mathcal{A}$ is authority state;
- $\mathcal{R}$ is responsibility and remedy state;
- $\mathcal{P}$ is permit state;
- $\mathcal{E}$ is evidence state;
- $\mathcal{V}$ is version, epoch, and configuration state;
- Z_0 is the admitted initial-state set;
- $\mathcal{B}$ is the declared boundary and consequence partition;
- $\mathcal{F}$ is the fault, uncertainty, and adversarial envelope.

A state must contain every variable required by the claim. If a variable can change the validity of a transition but is not represented or bound, the model cannot support the corresponding governance statement.

Let $Z_{\text{bad}} \subseteq Z$ denote forbidden or claim-defeating states and $Z_{\text{safe}} \subseteq Z$ a declared safe or acceptable region. The distinction is boundary-relative: a state may be safe for one consequence class and unsafe for another.

4.2 Governed Transition Relation

The governed transition relation is

$$z \xrightarrow{u}_{\mathfrak{G}} z' \quad \Longleftrightarrow \quad (z,u,z') \in \rightarrow \wedge \Pi^*_B(z,u,z') = 1 \wedge \mathrm{Med}_B(z,u,z') = 1. \tag{4.2}$$

The permit predicate determines admissibility. The mediation predicate determines whether the actual in-scope implementation path depends on that permit. Formal permission without implementation mediation remains model-internal.

A protected consequence function

$$\kappa : Z \times U \times Z \to \mathcal{C} \tag{4.3}$$

maps a transition to its consequence class. The primary non-reachability objective is

$$\mathrm{Reach}_{\mathfrak{G}}(Z_{\mathrm{bad}}) = \emptyset \tag{4.4}$$

under the declared initial set, transition relation, boundary, and fault envelope.

4.3 Invariant Preservation

Let $I : Z \to \{0,1\}$ be a candidate invariant. The ordinary induction obligations are:

$$\forall z \in Z_0 : I(z) = 1, \tag{4.5}$$

and

$$I(z) = 1 \wedge z \xrightarrow{u}_{\mathfrak{G}} z' \Longrightarrow I(z') = 1. \tag{4.6}$$

For execution-bound governance, this is not sufficient unless the formal transition relation corresponds to every in-scope execution route. A proof over $\mathfrak{G}$ establishes a deployment property only when a correspondence relation $\mathcal{C}$ binds source, binary, configuration, runtime state, hardware path, and evidence to the formal object.

4.4 Non-Reachability Theorem Form

Theorem Candidate 4.1 - Boundary-relative non-reachability. Assume:

1. $Z_0 \subseteq I$;
2. every candidate transition that can reach a protected consequence is represented in $\rightarrow$;
3. every represented protected transition is completely mediated by Med_B;
4. Π^*_B is false for every transition from I into Z_{bad};
5. faults remain inside $\mathcal{F}$;
6. implementation correspondence $\mathcal{C}$ holds for the declared artifact and version.

Then no execution trace from Z_0 reaches Z_{bad} inside boundary B.

The proof is an induction over the mediated transition relation. The theorem is deliberately conditional. The difficult obligations are usually assumptions 2, 3, 5, and 6: path completeness, complete mediation, fault-envelope coverage, and model-implementation correspondence.

4.5 Partial Observation and Robust Admissibility

Real systems are not fully observed. Let $O(z)$ be the observation and let $\mathcal{U}(O(z)) \subseteq Z$ be the uncertainty set consistent with the observation.

A robust permit is

$$\Pi^*_{B,\text{rob}}(O(z), u) = 1 \quad \Longleftrightarrow \quad \forall \hat{z} \in \mathcal{U}(O(z)) : \Pi^*_B(\hat{z}, u) = 1. \tag{4.7}$$

This is intentionally conservative. If some state consistent with the available evidence makes the transition inadmissible, ordinary allow is not justified. The system may request additional evidence, reduce action scope, delay, or enter a safe mode.

Partial observation creates three distinct failure modes:

- **missing state** - a mandatory variable is not measured or represented;
- **stale state** - the variable existed but is no longer current;
- **ambiguous state** - multiple materially different states are consistent with the observation.

These must not be collapsed into a generic confidence score.

4.6 Composition and Assume-Guarantee Contracts

Large systems are composed of components $S_1, \ldots, S_n$. Each component may provide a contract

$$A_i \Longrightarrow G_i, \tag{4.8}$$

where A_i is the assumption under which component i guarantees G_i.

Composition requires more than local satisfaction. The contract set is closed only if:

1. every assumption is supplied by another verified guarantee or by an explicit environment condition;
2. timing, identity, version, and fault semantics agree at interfaces;
3. no cyclic assumption is accepted without a fixed-point or induction argument;
4. authority is not silently expanded at component boundaries;
5. an integration path does not bypass the component whose guarantee is relied upon.

A composite permit may be expressed as

$$\Pi_{\text{comp}} = \bigwedge_{i=1}^{n} \left(A_i \wedge G_i \wedge C_i^{\text{interface}} \right), \tag{4.9}$$

where $C_i^{\text{interface}}$ is the correspondence and integration contract. Local correctness does not imply global governance when interface semantics, timing, or alternate paths remain open.

4.7 Timed and Hybrid Governance

For timed systems, a state includes clocks q and deadlines. A permit may expire when freshness or intervention margins are exceeded:

$$\Pi^*_B(\tau, t) = 0 \quad \text{for } t > t_{\text{expiry}}. \tag{4.10}$$

For hybrid systems with discrete modes $m \in M$ and continuous states x,

$$\dot{x} = f_m(x, u, w), \quad m^+ = \delta(m, x, u), \tag{4.11}$$

both continuous evolution and discrete switching require governance. A mode may be admissible while a switch is not; a switch may be formally permitted while the target mode lacks a reachable safe-state route.

The safe-state obligation is therefore not merely

$$\neg\Pi \Rightarrow \text{HALT},$$

but, where physical consequences exist,

$$\neg\Pi \Rightarrow \text{Reach}(Z_{\text{safe}}) \quad \text{within the declared timing and actuator envelope.} \tag{4.12}$$

4.8 Refinement and Model-Implementation Correspondence

Let M be an abstract model and I an implementation. A refinement relation $R \subseteq Z_I \times Z_M$ must preserve the properties exported by the model.

A minimal simulation obligation is:

$$(z_I, z_M) \in R \wedge z_I \to_I z'_I \Longrightarrow \exists z'_M : z_M \to_M z'_M \wedge (z'_I, z'_M) \in R. \tag{4.13}$$

For governance claims, refinement must also preserve:

- transaction identity;
- authority and revocation state;
- version and freshness;
- fault semantics;
- evidence binding;
- denial and halt dominance;
- absence of implementation-only bypasses.

A solver proof, model-checking result, or theorem cannot be promoted to the deployment without this bridge.

4.9 Decidability and Verification Boundaries

Not every governance property is decidable or tractable. The architecture therefore distinguishes:

- exact finite-state verification;
- bounded model checking;
- theorem proving under assumptions;
- abstract interpretation;
- directed numerical enclosure;
- runtime monitoring of residual obligations;
- open or unproved properties.

Solver timeout, numerical non-convergence, or missing coverage is not a proof. The authoritative result must remain `INCONCLUSIVE` or `OPEN`.

4.10 Proof-Preserving Artifact Chain

A formal result should be bound to:

```
specification version
  -> model generator
  -> solver or proof environment
  -> assumptions and parameter set
  -> result certificate
  -> independent replay or control route
  -> implementation correspondence
  -> release record
```

Every change to a mandatory upstream artifact invalidates dependent results unless a controlled replay establishes continuity.

4.11 Propositions

Proposition 4.1 - Model completeness bound. Non-reachability in a model proves deployment non-reachability only to the extent that every protected deployment path is represented and corresponded.

Proposition 4.2 - Robust unknown-state rule. A transition is robustly admissible under partial observation only when it is admissible for every state consistent with the authoritative observation and uncertainty set.

Proposition 4.3 - Composition non-automaticity. Component-level guarantees compose only under explicit, closed, compatible assume-guarantee and interface contracts.

Proposition 4.4 - Timed validity. Authority, evidence, and permit state are invalid after their declared freshness or intervention window unless explicitly renewed.

Proposition 4.5 - Mechanized-result boundary. A mechanized proof establishes only the property encoded, under the assumptions and model version actually checked.

4.12 Status

STRUCTURE. A governed transition-system semantics, robust permit, composition contract, timed/hybrid extension, and refinement bridge are specified.

EXECUTION REALITY. Formal closure depends on path discovery and implementation correspondence. These remain domain-specific obligations.

FAILURE POINT. Omitted paths, hidden assumptions, inconsistent interface semantics, or unproved refinement defeat deployment-level promotion.

GOVERNANCE STATUS. M1 formal architecture; later artifacts may establish bounded higher maturity.

CONFIDENCE TYPE. Deductive under explicit assumptions; otherwise model-internal.

Chapter 5

Governed Dynamics: Stability, Load, Delay, and Safe-State Reachability

5.1 Dynamical-System Object

A consequence-bearing system may contain continuous, discrete, stochastic, organizational, and informational dynamics. A governed dynamical system is represented by

$$\dot{x} = f(x, u, w, t), \quad x^{+} = g(x, u, w, t), \tag{5.1}$$

with state x, action or control u, disturbance and uncertainty w, and possibly discrete updates g. Governance constrains the available control and transition set rather than merely evaluating the output after evolution.

Let

$$\mathcal{U}_{\text{adm}}(x, t) = \{u \in \mathcal{U} \mid \Pi_B^*(x, u, t) = 1\}. \tag{5.2}$$

If $\mathcal{U}_{\text{adm}}(x, t) = \emptyset$, the architecture must not fabricate an ordinary allow. It must invoke the declared fallback: delay, controlled degradation, shedding, safe-state control, or halt.

5.2 Candidate SGDS Class

A **Structurally Governable Dynamical System (SGDS)** is a candidate system class defined by five non-substitutable criteria.

5.2.1 C1 - Governance-state representability

Every variable required by the claimed governance decision is represented, typed, version-bound, and deterministically serializable. A hidden or unbound variable that can change admissibility defeats C1.

5.2.2 C2 - Load-bounded damping

The system possesses a declared load envelope and a governed response that measurably reduces or redistributes the relevant load when thresholds are approached or exceeded. Detection alone is insufficient.

5.2.3 C3 - Hard admissibility boundary

Every in-scope externally relevant effect passes through an exclusive commit or enforcement path. Observation, exposure, logging, or recommendation does not by itself satisfy C3.

5.2.4 C4 - Responsibility-anchored continuation

Continuation is bound to a valid principal, role, scope, time, transition, evidence, and revocation state. A routing or continuity mechanism may transport authority; it may not create it.

5.2.5 C5 - Deterministic runtime gating

The final runtime gate has a defined truth table, priority order, halt dominance, and governed recovery. The evaluated model may not control the gate, thresholds, or authority source.

A real system is not a confirmed SGDS unless all five criteria pass on the exact frozen candidate and boundary. Integration conformance, abstract modelling, or a field kit does not substitute for real candidate evidence.

5.3 Structural Load

Structural load is a governance variable, not automatically a physical energy or universal stability quantity. It records accumulated burden relevant to the declared system: unresolved branches, context expansion, queue pressure, dependency stress, authority ambiguity, fault accumulation, or other explicitly measured factors.

A generic ledger is

$$L(t) = L_0 + \int_0^t \phi(x(\tau), u(\tau), w(\tau))\, d\tau - S(t), \quad \phi \geq 0, \tag{5.3}$$

where $S(t)$ is governed shedding or release. The interpretation of L, ϕ, and S is domain-specific and must be measured or formally bound.

C2 requires more than observing $L(t)$. It requires evidence that the declared response reduces the relevant load without moving the burden into an unobserved path.

A load-shifting counterexample occurs when one queue, mode, or route appears relieved while an adjacent dependency becomes overloaded. Therefore the evidence set must include before/after measurements, adjacent limits, connectivity, and re-evaluation.

5.4 Stability and Damping

Stability claims must be tied to the correct system class. For continuous-time linear dynamics $\dot{x} = Ax$, asymptotic stability requires eigenvalues with negative real parts. For discrete-time dynamics $x_{k+1} = Ax_k$, stability requires spectral radius less than one. These conditions are not interchangeable and do not automatically apply to nonlinear, switching, delayed, or partially observed systems.

The governance architecture may use candidate indicators such as local contraction, Lyapunov decrease, barrier conditions, or bounded gain. It must not present one spectral metric as a universal proof of safety or governability.

A Lyapunov-style obligation is

$$V(x) > 0, \qquad \dot{V}(x, u, w) \leq -\alpha(\|x\|) \tag{5.4}$$

inside the declared envelope. A governance filter may restrict u to controls satisfying the decrease condition, but only if the model, sensing, delay, and actuator bounds are valid.

5.5 Control Barrier and Invariant Conditions

Let the acceptable region be

$$\mathcal{S} = \{x \mid h(x) \geq 0\}. \tag{5.5}$$

A control-barrier condition may require

$$\sup_{u \in \mathcal{U}_{\text{adm}}} \left[L_f h(x) + L_g h(x) u + \alpha(h(x)) \right] \geq 0. \tag{5.6}$$

This condition is meaningful only when $\mathcal{U}_{\text{adm}}$ is non-empty, the model is adequate, and sensing and execution delays remain inside the declared envelope.

The architecture separates:

- **safety** - forbidden states are avoided;
- **liveness** - desired progress remains possible;
- **recoverability** - a safe or acceptable state is reachable after invalidation;
- **availability** - useful service remains accessible.

A system may be safe by permanently halting while failing liveness and availability. The claim must state which property is established.

5.6 Delay and Irreversibility

For an intervention to remain effective,

$$T_{\text{sense}} + T_{\text{compute}} + T_{\text{communicate}} + T_{\text{actuate}} < T_{\text{irreversible}}. \tag{5.7}$$

This inequality must hold under the declared worst-case timing envelope, not only under average conditions. If the effect becomes irreversible before the gate or human intervention can act, the claimed boundary is misplaced.

Delay interacts with freshness. A permit valid at t_0 may be invalid at execution time t_1. Therefore the transaction must bind time and epoch, and the gate must reject stale state.

5.7 Model Predictive Governance

A predictive controller may optimize over a horizon while enforcing admissibility constraints:

$$\min_{u_{0:N-1}} \sum_{k=0}^{N-1} \ell(x_k, u_k) + V_f(x_N) \tag{5.8}$$

subject to

$$x_{k+1} = f(x_k, u_k), \quad x_k \in \mathcal{X}_{\text{adm}}, \quad u_k \in \mathcal{U}_{\text{adm}}. \tag{5.9}$$

The optimizer does not own authority. It searches only within the admissible set established by upstream governance and system constraints. Infeasibility must not be resolved by silently relaxing a hard authority or safety predicate.

5.8 Distributed and Multi-Agent Dynamics

In distributed systems, local damping or local allow may create common-mode failure. A coupled system

$$\dot{x}_i = f_i(x_i, u_i) + \sum_j c_{ij}(x_j - x_i) \tag{5.10}$$

requires a joint boundary that accounts for communication loss, stale shared state, inconsistent authority, and correlated assumptions. Agreement among several components does not prove correctness if they share the same defective model or evidence source.

5.9 SGDS Membership Logic

For a frozen candidate S, define

$$\text{SGDS}(S) = \begin{cases} \text{CONFIRMED}, & C_1 = \cdots = C_5 = \text{PASS}, \\ \text{REJECTED}, & \exists i : C_i = \text{FAIL}, \\ \text{NOT CONFIRMED}, & \text{otherwise}. \end{cases} \tag{5.11}$$

`NOT CONFIRMED` is not a softened pass. It means the candidate lacks sufficient evidence for one or more mandatory criteria.

5.10 Propositions

Proposition 5.1 - Detection non-equivalence. Detection of excessive load does not satisfy C2 unless the governed response measurably reduces the declared load without displacing it outside the measured boundary.

Proposition 5.2 - Gate non-substitution. FSEL, sensing, monitoring, or state exposure may support C1 and C2 evidence but does not itself establish the hard C3 boundary.

Proposition 5.3 - Halt non-equivalence. Deterministic halt does not prove safe-state arrival unless plant, delay, actuator, and disturbance conditions establish reachability.

Proposition 5.4 - Criterion conjunction. Confirmation as SGDS requires PASS for all C1-C5 on the exact frozen real candidate.

Proposition 5.5 - Model-specific stability. A stability indicator is valid only for the mathematical class and envelope for which its conditions are established.

5.11 Status

STRUCTURE. The SGDS C1-C5 class, load ledger, admissible-control set, timing condition, and membership logic are specified.

EXECUTION REALITY. SGDS-FSEL v0.6 supplies the field instrumentation and evaluation logic; real KWI-LIAN evidence is discussed in Chapter 9.

FAILURE POINT. Detection without damping, monitoring without hard boundary, stale authority, nondeterministic gating, or unproved safe-state reachability defeats the relevant criterion.

GOVERNANCE STATUS. M1 class definition; candidate-specific evidence varies.

CONFIDENCE TYPE. Formal architecture with domain-specific empirical obligations.

Chapter 6

Runtime Mediation and Governed Execution

6.1 Proposal, Permission, Commit, and Consequence

A governed runtime must separate four events that ordinary software often collapses:

1. **Proposal** - an AI, user, service, or process suggests an action.
2. **Permission** - the mandatory permit predicates are evaluated.
3. **Commit** - an authorized actor or controller binds the action to an exact state.
4. **Consequence** - the action reaches the protected external or internal effect.

The separation is

$$\text{Proposal} \not\equiv \text{Permission} \not\equiv \text{Commit} \not\equiv \text{Consequence}. \tag{6.1}$$

A system that allows a proposal to produce a consequence before permission and commit has crossed the governance boundary too late.

6.2 Typed Transaction Identity

Every consequence-bearing transaction requires a stable identity

$$T = \langle id, p, o, a, c, v, e, t, \eta \rangle, \tag{6.2}$$

where id is a unique transaction identifier, p principal, o object, a action class, c context, v version and configuration, e evidence root, t time/epoch, and η current lifecycle state.

The transaction state machine may be represented as

```
CREATED
  -> PROPOSED
  -> CONTEXT_BOUND
  -> PERMIT_EVALUATED
  -> COMMIT_REQUIRED
```

```
-> COMMITTED
-> EXECUTING
-> EFFECT_CONFIRMED
-> EVIDENCE_CLOSED
```

with terminal alternatives `REJECTED`, `DELAYED`, `HALTED`, `ROLLED_BACK`, `FAILED`, and `INCONCLUSIVE`.

Post-permit mutation of object, action, context, version, or evidence invalidates the permit unless the permitted transformation is itself represented and re-evaluated.

6.3 Complete Mediation

Let $\mathcal{P}_{\text{effect}}$ be the set of paths to a protected consequence and $\mathcal{P}_{\text{med}}$ the paths that pass through the authoritative gate. Complete mediation requires

$$\mathcal{P}_{\text{effect}} \subseteq \mathcal{P}_{\text{med}}. \tag{6.3}$$

This includes ordinary APIs and less visible routes:

- administrative and maintenance interfaces;
- background workers and scheduled jobs;
- recovery and retry logic;
- direct file, message, or database writes;
- cached credentials and stale permits;
- import/export paths;
- debug or emergency interfaces;
- supplier and remote-management channels;
- physical or manual side effects.

A gate that protects the nominal API while leaving an alternate route open does not establish non-bypassability.

6.4 Capability Control and Least Authority

Runtime authority should be represented as explicit capabilities rather than inferred from ambient privilege. A capability should bind:

$$\text{Cap} = \langle principal, object, action, scope, expiry, epoch, constraints \rangle. \tag{6.4}$$

Capabilities may be attenuated but not silently expanded. The runtime should grant the minimum action set needed for the committed transition and revoke or consume authority after use where appropriate.

The AI component receives proposal capability, not authority to create its own final effect capability.

6.5 Atomicity and Race-Free Governance

Permission, commit, state mutation, and consequence initiation must be coordinated so that a race cannot execute an invalid state.

A canonical atomic condition is

$$\text{Commit}(T) \wedge \text{Current}(T) \wedge \Pi_B^*(T) \xrightarrow{\text{atomic}} \text{EffectIntent}(T). \tag{6.5}$$

No concurrent mutation may alter the transaction after the authoritative check but before effect initiation without invalidation or re-evaluation.

Relevant counter-constructions include:

- time-of-check to time-of-use mutation;
- duplicated commits;
- simultaneous halt and execute signals;
- stale cache winning over current revocation;
- delayed messages crossing an epoch boundary;
- two users finalizing the same object under different states.

6.6 Persistence and Crash Consistency

A crash must not restore consumed authority, lose a halt, erase a commitment, or create an ambiguous external effect.

Durable state should distinguish:

- action not started;
- action prepared but not committed;
- committed but effect unknown;
- effect confirmed;
- rollback completed;
- evidence closure pending.

Idempotency keys and transaction epochs prevent retries from creating duplicate consequences. If the external result cannot be determined after a crash or timeout, the state is `INCONCLUSIVE`, not guessed success.

6.7 Distributed Execution and External Effects

For distributed actions, exactly-once semantics are rarely available end to end. Governance must therefore use explicit effect protocols, idempotence, reconciliation, and evidence of external confirmation.

A generic flow is

```
prepare local transaction
  -> verify current permit and commit
  -> emit effect intent with transaction ID and epoch
```

```
-> receive authenticated acknowledgment
-> reconcile actual external state
-> close evidence
```

An acknowledgment is not automatically proof of the physical or institutional effect. The evidence model must distinguish command acceptance, execution, and observed outcome.

6.8 Evidence Architecture

The runtime evidence package should bind:

- transaction ID and parent relation;
- actor, device, role, and case;
- object hash and version;
- action class;
- permit predicate results;
- commit state;
- timestamps and epoch;
- gate decision;
- external effect status;
- recovery or remedy action;
- artifact and configuration hashes.

Hash binding supports integrity, but a hash alone does not establish semantic correctness, authority, or complete path mediation.

6.9 Trusted Computing Base and Isolation

The trusted computing base (TCB) includes every component whose compromise can create an unauthorized consequence or falsify authoritative evidence. Minimization reduces the proof and assurance burden.

Isolation must consider:

- process and memory boundaries;
- privilege and capability boundaries;
- file and database write paths;
- network and message routes;
- configuration and secret access;
- update and rollback paths;
- logging and evidence stores;
- operator and administrator functions.

A sandbox is only as strong as the effects excluded from it. If a sandboxed agent can still cause a protected effect through an external tool or credential, the relevant consequence remains in scope.

6.10 Version and Supply-Chain Integrity

A runtime claim is bound to source, build, dependencies, configuration, model, policy, schema, and deployment state. The release object may be represented as

$$R = \langle source, build, deps, config, model, policy, schema, artifact\_hash \rangle. \tag{6.6}$$

A change to any mandatory element invalidates dependent evidence until replay or reassessment. Secure boot and signed packages may support integrity, but do not prove that the signed artifact is safe or governed.

6.11 Propositions

Proposition 6.1 - Sequence separation. Proposal, permission, commit, and consequence are distinct state transitions and may not be collapsed without losing governance meaning.

Proposition 6.2 - Complete mediation. A runtime enforcement claim is valid only if every in-scope path to the protected effect depends on the gate.

Proposition 6.3 - Post-permit mutation. Any material mutation after permit evaluation invalidates the permit unless the mutation is explicitly included in the authorized transaction.

Proposition 6.4 - Crash non-restoration. Recovery may not recreate consumed, revoked, stale, or halted authority.

Proposition 6.5 - External-effect uncertainty. When external effect status is unknown, the authoritative result is inconclusive until reconciled.

6.12 Status

STRUCTURE. Runtime transaction identity, complete mediation, atomicity, persistence, effect coordination, evidence, and TCB obligations are specified.

EXECUTION REALITY. KWI LIAN supplies bounded record, document, snapshot, presence, and commit infrastructure. Physical enforcement requires the Chapter 7 boundary.

FAILURE POINT. Alternate APIs, races, stale state, crash-restored authority, ambiguous external effects, or unbound versions defeat runtime closure.

GOVERNANCE STATUS. M1 general architecture; bounded implementation evidence exists in KWI LIAN components.

CONFIDENCE TYPE. Engineering architecture and artifact-dependent evidence.

Chapter 7

Physical Enforcement and the Governed CoChip

7.1 Why Software Enforcement Is Not Always Sufficient

A software gate may mediate a tool call, API, or command inside a compute environment. Where the protected consequence is physical, irreversible, or safety-critical, the claim may require a boundary outside the evaluated model and outside the ordinary software execution plane.

The reason is structural: a monitor that shares the same failure, privilege, clock, reset, or compromise domain as the monitored function may not provide the required independence. A software decision can also be bypassed if the protected actuator, bus, peripheral, or effect interface remains reachable through another route.

The hardware-enforcement question is therefore:

> Does the protected effect path physically depend on a deterministic gate whose authority source is external to the evaluated AI system?

7.2 Three-Layer Architecture

The public-safe CoChip architecture separates three layers.

```
AI or tool request
  -> Layer 1: Admissibility Engine
  -> Layer 2: Signal Encoding and Integrity
  -> Layer 3: Hardware Enforcement Kernel (CoChip)
  -> protected command/effect boundary
```

7.2.1 Layer 1 - Admissibility Engine

The upstream layer evaluates structural admissibility. It may consume formal, runtime, context, authority, and evidence predicates, but it does not become the physical gate.

7.2.2 Layer 2 - Signal Encoding

A typed governance state is encoded into a hardware-readable vector. The vector must bind freshness, epoch, transaction identity, integrity, and externally sourced authority conditions.

7.2.3 Layer 3 - CoChip

The CoChip is a deterministic, non-learning enforcement kernel. It does not interpret natural language, infer intent, optimize utility, or generate authority. It maps a validated governance vector to one of four outputs:

$$\mathcal{O}_H = \{\text{ALLOW}, \text{DELAY}, \text{THROTTLE}, \text{HALT}\}. \tag{7.1}$$

7.3 Governance Vector

A public-safe input vector may contain

$$q = \langle admissible, authority, presence, commit, evidence, current, signal\_valid, watchdog\_ok, tamper\_fr \tag{7.2}$$

The exact protected encoding, thresholds, bit allocation, and deployment-specific semantics are not disclosed here. The architectural rule is that authority, presence, and commit must be sourced outside the evaluated model.

A deterministic priority relation is required. A minimal public-safe ordering is:

$$\text{tamper}/\text{freshness}/\text{watchdog}/\text{fault} \Rightarrow \text{HALT}, \tag{7.3}$$

$$\neg authority \vee \neg commit \vee \neg evidence \Rightarrow \text{DELAY or HALT}, \tag{7.4}$$

$$admissible \wedge current \wedge healthy \wedge authorized \Rightarrow \text{ALLOW} \tag{7.5}$$

subject to domain-specific throttle and review conditions.

HALT dominance means that no model confidence, prior allow, utility score, or ordinary software request can override a valid halt condition.

7.4 Freshness, Replay, and Tamper Resistance

A logically correct decision is not physically trustworthy unless the signal path is protected.

Required mechanisms include:

- transaction and epoch binding;
- freshness counters or timestamps;

- integrity or authentication checks;
- replay detection;
- malformed-vector rejection;
- watchdog monitoring of the upstream source;
- detection of output-register override or illegal state;
- fail-closed startup;
- controlled and evidenced recovery.

A stale vector that once represented an admissible state must not remain valid after authority, document, case, or system state changes.

7.5 Fail-Closed Startup and Sticky Halt

The default state after power-on, reset, incomplete initialization, invalid configuration, or lost upstream heartbeat is halt or otherwise non-permissive.

A sticky halt remains asserted across transient input recovery until the declared recovery procedure establishes new valid context, authority, version, evidence, and health. A mere return of the prior allow bit is insufficient.

Recovery is a new governed transition, not an automatic restoration of the last operational state.

7.6 Non-Maskable Effect Boundary

The intended physical invariant is

$$\mathrm{HALT} = 1 \implies \neg\, \mathrm{EffectPath}_B\,. \tag{7.6}$$

This is currently an architectural requirement, not a physically proven fact for a fabricated artifact. To establish it, the actual topology must show that no software, debug, reset, communication, manual, or electrical path can create the protected effect while halt is asserted within the declared fault envelope.

A command acknowledgment is not equivalent to physical effect prevention. Measurements must be taken at the last meaningful effect boundary.

7.7 Power, Clock, Reset, and Fault Domains

Physical governance requires explicit treatment of:

- power-on and brownout states;
- independent or shared clock domains;
- reset ordering and reset release;
- metastability and synchronization;
- watchdog expiry;
- frozen-at-allow faults;
- output-latch behavior;
- bus partition and recovery;

- debug access;
- configuration rollback;
- environmental and electromagnetic fault conditions where relevant.

The evidence package must show which faults were injected, which outputs were observed, and which conditions remain untested.

7.8 Integration Levels

Three conceptual integration levels are distinguished.

7.8.1 Level A - Firmware governor

A synchronous runtime governor in the inference or tool stack. It can provide useful mediation but shares more of the compute plane and therefore has a larger common-mode failure surface.

7.8.2 Level B - Independent integrity coprocessor

A dedicated controller on a separated execution or clock domain with a non-maskable halt interface. The compute plane cannot ordinarily override the enforcement plane.

7.8.3 Level C - Dedicated FPGA or ASIC kernel

A minimal deterministic hardware block with narrowly configurable thresholds and explicit physical control over the protected effect path.

These levels describe architectural integration and assurance objectives. They do not by themselves establish any Automotive Safety Integrity Level, certification status, or standard compliance. Such claims require a complete item definition, hazard analysis, safety concept, hardware and software evidence, tool qualification where applicable, independent assessment, and a competent determination.

7.9 Upstream-to-Downstream Contract

The CoChip does not solve the upstream authority problem. It can enforce a signal, but it cannot know by itself whether the correct actor, device, case, document, evidence, and human commit exist.

The contract is therefore:

```
KWI LIAN upstream context
  -> authority/presence/commit/evidence/state predicates
  -> encoded governance vector
  -> CoChip deterministic enforcement
  -> protected consequence or HALT
  -> evidence of the decision and outcome
```

This relationship is complementary. KWI LIAN grounds the authority signal; the CoChip enforces the downstream boundary.

7.10 Validation Programme

A physical demonstrator must include at minimum:

1. frozen RTL or equivalent logic artifact;
2. signal and register specification;
3. transaction and freshness protocol;
4. power, clock, reset, and watchdog design;
5. effect-boundary topology;
6. bypass inventory;
7. timing measurements;
8. replay, glitch, stale-state, and tamper tests;
9. fault injection across power, clock, communication, I/O, and debug paths;
10. evidence preservation across reset and recovery.

An FPGA prototype can establish bounded implementation evidence. It cannot by itself establish production-silicon qualification or safety certification.

7.11 Propositions

Proposition 7.1 - External authority source. A hardware enforcement kernel may enforce but may not originate authority for the evaluated transition.

Proposition 7.2 - Physical path condition. Physical non-bypassability is established only when every in-scope effect path is shown to depend on the enforcement output under the declared fault envelope.

Proposition 7.3 - Fail-closed recovery. Startup, reset, stale signal, invalid configuration, and lost watchdog state must not default to allow.

Proposition 7.4 - Halt non-safety. A non-maskable halt proves only effect-path interruption unless safe-state reachability is separately established.

Proposition 7.5 - Assurance non-transfer. Architectural correspondence to safety or security constructs is not a certification or integrity-level determination.

7.12 Status

STRUCTURE. The three-layer architecture, governance vector, priority logic, fault domains, and validation route are specified.

EXECUTION REALITY. A public-safe architectural paper exists. A complete physical FPGA/ASIC evidence package is open.

FAILURE POINT. Upstream signal corruption, stale allow, a maskable halt, reset-cleared authority, or an alternate effect path defeats the claim.

GOVERNANCE STATUS. M1 specified architecture; physical implementation and independent validation open.

CONFIDENCE TYPE. Engineering architecture with unclosed physical proof obligations.

Chapter 8

KWI LIAN Kyber as an Implemented Upstream Governance System

8.1 Purpose and Operating Principle

KWI LIAN Infrastructure Edition is a local work and evidence infrastructure for document-intensive workflows. It connects LibreOffice Writer and Calc, controlled AI assistance, document-state verification, server-side records, evidence continuity, user and device state, and human commit boundaries.

Its operating principle is:

> The AI supports. The human decides. The infrastructure documents.

The system does not grant autonomous administrative authority to AI. It does not claim to determine the semantic truth of generated content. Its contribution is to keep AI-supported work attached to the operational context from which authority and evidence can be derived.

8.2 Three Core Roles

8.2.1 Kyber

Kyber is the user-facing LibreOffice extension for Writer and Calc. Public-safe functions include:

- document or table assistance;
- LIAN state check;
- LIAN stamp;
- KWI status;
- KWI evidence or commit action;
- document finalization and controlled workspace functions;
- localized user interfaces.

Kyber keeps assistance close to the work object rather than detached in a generic chat or cloud window.

8.2.2 LIAN

LIAN, the Logical Integrity Architecture Node, verifies whether the current document or table state matches the last known evidence-bound state. It does not make the domain decision.

Public states include:

- `VERIFIED` - current state matches the last evidence state;
- `MISMATCH` - current state differs from the last evidence state;
- `NO EVIDENCE` - no prior evidence state exists;
- additional controlled states such as stale lock, missing commit, unresolved artifact, or invalid path where implemented.

8.2.3 KWI

KWI, the Kyber Work Infrastructure, is the server-side record and operations layer. Public-safe modules include:

- Dashboard Bridge;
- Portal Shell;
- Record Layer;
- Evidence Intake;
- Document Shadow, Snapshot, and Commit Viewer;
- user, role, device, lifecycle, and presence state;
- startup reconciliation;
- backup and audit extract functions.

8.3 Context-Bound Workflow

A canonical bounded workflow is:

```
open Writer document or Calc table
  -> work with or without AI assistance
  -> bind user, device, case, and object context
  -> evaluate current LIAN state
  -> create evidence/commit action
  -> store hash, snapshot, and record server-side
  -> preserve document shadow and commit history
  -> detect later drift as VERIFIED or MISMATCH
```

The final file alone does not prove the path by which it became final. A document may be edited after review, copied into another file, moved, renamed, overwritten, or detached from its original case. The Document Shadow preserves a known state at a governance-relevant point and binds it to records and evidence.

8.4 Document-State Logic

Let D_t be the current document state and $H(D_t)$ its content hash under the declared canonicalization. Let h^* be the evidence-bound reference hash.

$$\mathrm{LIAN}(D_t) = \begin{cases} \text{VERIFIED}, & H(D_t) = h^*, \\ \text{MISMATCH}, & H(D_t) \neq h^*, \\ \text{NO EVIDENCE}, & h^* \text{ absent}. \end{cases} \tag{8.1}$$

This establishes state correspondence, not semantic correctness. A verified document may still contain a wrong statement; it is verified only as unchanged relative to the referenced evidence state.

A MISMATCH prevents the old evidence from being represented as evidence for the new content. Progression should require review, recommit, or a new evidence state.

8.5 Human Commit and Evidence Record

A bounded KWI record may bind:

- Request ID;
- principal and role;
- device and presence state;
- case or workflow identity;
- action class;
- document path and object identity;
- content hash;
- snapshot;
- commit time;
- evidence root;
- LIAN status;
- seal or finalization state;
- later mismatch or recovery events.

The commit makes human authority operationally meaningful only when it is bound to the exact state. A generic click or unbound approval does not provide the same property.

8.6 Local Operator Sovereignty

KWI LIAN is designed for local deployment in infrastructure controlled by the receiving organization. The public-safe operating model includes:

- no mandatory cloud execution for the governed workflow;
- operator-controlled storage and backups;
- operator-managed users, devices, network shares, and permissions;
- Windows and Linux clients;
- local LibreOffice OXT deployment;
- server, portal, dashboard, record, evidence, and shadow services;
- updates and support delivered without requiring permanent vendor access.

Local deployment does not automatically prove governance, privacy, or security. It removes or reduces certain external dependencies and makes the operator's boundary more inspectable.

8.7 Bounded P1 Implementation

The neutralized P1 package documents a functional core including:

- Writer and Calc OXT packages in German, English, and French;
- Dashboard, Portal, Document Shadow, Record Layer, and Evidence Intake;
- KWI evidence with hash, snapshot, and commit;
- LIAN VERIFIED/MISMATCH behavior;
- startup reconciliation;
- device presence states online, stale, and offline;
- Windows and Linux client configuration;
- local server modules and a structured installation route.

This is implementation evidence for the declared P1 functions. It is not a product certification or proof that every deployment configuration preserves the same properties.

8.8 Declared Internal Function Tests

The public-safe installation and test route includes:

8.8.1 Writer route

1. create and save a Writer document;
2. query KWI status;
3. create KWI evidence;
4. verify the state;
5. modify the content;
6. run LIAN check again;
7. expect `VERIFIED` before mutation and `MISMATCH` after uncommitted mutation.

8.8.2 Calc route

1. create and save a Calc table;
2. query KWI status;
3. apply LIAN stamp;
4. create KWI evidence;
5. verify state;
6. modify a cell;
7. verify again;
8. expect the declared status sheet and MISMATCH after mutation.

8.8.3 Server route

- portal, dashboard, record, evidence, and shadow services reachable;
- client presence visible;
- snapshot available;
- commit history visible;
- device timeout and startup reconciliation produce expected state.

These are bounded internal tests. Independent installation by a separate competent party remains a higher maturity step.

8.9 Multi-Client and Handoff Architecture

The reference lab topology is designed to move from a primary client to multiple lower-powered clients and roles. The purpose is not performance benchmarking alone. It is to test:

- case visibility across users;
- role and capability separation;
- handoff and substitution;
- device lifecycle and presence;
- continuity when one worker is unavailable;
- evidence access without silent authority transfer;
- reconciliation after offline or stale clients return.

A handoff must preserve case, object, evidence, unresolved state, and responsibility. The receiving user must not inherit authority merely because the file is accessible.

8.10 Relationship to the EU AI Act and Other Governance Frameworks

KWI LIAN can support architectural expectations associated with risk management, logging, transparency, human oversight, traceability, and local operator control. It may help produce evidence relevant to organizational obligations.

It does not by itself establish legal compliance. Legal applicability, actor classification, provider/deployer obligations, quality-management requirements, and conformity assessment must be determined against current authoritative sources and the actual deployment.

8.11 Limitations

KWI LIAN does not:

- certify the truth of AI-generated content;
- replace domain review;
- create legal authority;
- guarantee complete cybersecurity;
- prove that every side effect is mediated;
- establish physical non-bypassability;
- replace existing specialist procedures or formal conformity assessment;
- make AI an autonomous administrative actor.

Its narrower and stronger contribution is to separate AI assistance, human authority, document state, evidence continuity, and server-side operational context.

8.12 Propositions

Proposition 8.1 - State correspondence. LIAN VERIFIED establishes correspondence to the referenced evidence-bound state, not semantic truth.

Proposition 8.2 - Drift invalidation. A material change after evidence creation invalidates the prior state relationship until new evidence and, where required, a new human commit are created.

Proposition 8.3 - Upstream authority support. KWI LIAN can supply actor, device, case, document, evidence, and commit predicates required by downstream enforcement without deriving them from model confidence.

Proposition 8.4 - Locality non-equivalence. Local operation supports operator sovereignty but does not automatically establish security, compliance, or complete governance.

Proposition 8.5 - Handoff non-transfer. Access to a case or document does not automatically transfer authority; handoff requires explicit role, scope, evidence, and responsibility binding.

8.13 Status

STRUCTURE. The Kyber-LIAN-KWI roles, state logic, commit record, local operating model, and handoff route are specified.

EXECUTION REALITY. A bounded P1 implementation and internal function-test route exist for the declared environment.

FAILURE POINT. Semantic overclaim, missing case/authority binding, post-evidence mutation, unmediated external paths, or deployment drift narrow the claim.

GOVERNANCE STATUS. M2 for bounded implementation; M3 for declared repeatable internal routes where evidence is retained; M4 external reproduction open.

CONFIDENCE TYPE. Artifact-supported bounded case evidence.

Chapter 9

SGDS-FSEL-RCFA and the Live-Evidence Route

9.1 Layered Integration Contract

SGDS, FSEL, and RCFA are integrated through a refinement and enforcement contract. They are not collapsed into one authority-owning framework.

- **SGDS** remains the formal system class and C1-C5 membership logic.
- **FSEL** exposes common, axis-specific, and coupled state deviations in a controlled mode representation.
- **RCFA** preserves route, persistent identity, authority binding, evidence binding, and reconstructable location while the active window moves.
- **SLI, LIAN, PoA, or equivalent governance layer** evaluates load, admissibility, and responsibility conditions.
- **Runtime enforcement** implements the exclusive `ALLOW`, `DELAY`, `THROTTLE`, and `HALT` path.

The canonical architecture is:

```
dynamic process or model proposals
  -> FSEL controlled state exposure
  -> RCFA route, identity, authority, and evidence continuity
  -> SGDS C1-C5 governance classification
  -> independent SLI/LIAN/PoA/runtime gate
  -> ALLOW / DELAY / THROTTLE / HALT
```

FSEL and RCFA are non-authoritative. They may contribute observability and continuity. They may not expand admissible state space, create responsibility, alter thresholds, or bypass the gate.

9.2 Integration Invariants

The v0.6 integration specification defines twelve invariants.

9.2.1 I1 - Projection non-authority

A projection or mode may inform a decision but may not grant permission.

9.2.2 I2 - Common-mode conservatism

Agreement or zero differential across views does not imply PASS without independent evidence. Unknown mandatory state leads to a non-permissive result.

9.2.3 I3 - Persistent identity

Recentering preserves node identity, authority binding, evidence binding, and commit references.

9.2.4 I4 - Address-state separation

Visible coordinates may change; semantic identity and reconstructable global position may not silently change.

9.2.5 I5 - Governed release

A fiber, route, edge, or subsystem may be isolated only as an authorized, evidenced, and re-evaluated transition.

9.2.6 I6 - No load shifting

A claimed damping action must reduce the declared boundary load and keep adjacent limits valid.

9.2.7 I7 - Exclusive commit path

Every effectful transition passes through the same gate and enforcement path; no tool, retry, actuator, or side-effect bypass remains.

9.2.8 I8 - HALT dominance

HALT overrides utility, confidence, prior authorization, and model preference.

9.2.9 I9 - Recovery re-binding

Recovery requires new valid responsibility, evidence, version, and admissibility evaluation.

9.2.10 I10 - Version identity

No conformance result survives candidate, configuration, threshold, transform, or enforcement-version drift without replay.

9.2.11 I11 - Evidence non-substitution

Logs and hashes may show what occurred; they do not by themselves prevent an inadmissible transition.

9.2.12 I12 - Model non-self-extension

The evaluated model may not expand the admissible set, alter transforms or thresholds, suppress conflicting evidence, or reclassify UNKNOWN as ALLOW.

9.3 FSEL and Controlled State Exposure

FSEL decomposes a bounded task state into controlled views or modes. A simplified representation is

$$\mathcal{E} = (m_0, m_1, m_2, m_{12}), \tag{9.1}$$

where m_0 may represent common state, m_1 and m_2 axis-specific deviations, and m_{12} coupled deviation. The exact operational definition is profile-specific.

Exposure does not create governance. It provides structured evidence to the SGDS and enforcement layers. A mode result is therefore an observation object, not an authority object.

9.4 RCFA and Continuity

RCFA preserves route-centered continuity under recentering or moving task windows. It binds:

- persistent identity;
- previous and current address;
- authority state;
- evidence root;
- version and transform;
- parent and route relation;
- reconstructable global position.

This prevents a visible coordinate change from silently becoming a semantic identity change. It also prevents authority or evidence from being lost when the active local view moves.

9.5 C1-C5 Refinement Map

The integration contributes to SGDS evidence as follows:

Criterion	FSEL/RCFA contribution	Required real evidence	Non-negotiable limit
C1 representability	typed exposure, mode vector, RCFA state and version	complete governance-variable mapping and deterministic serialization	PASS only if all claimed variables are bound
C2 load-bounded damping	fiber strain, load envelope, governed release	measured load before/after, adjacent limits, re-evaluation	detection alone is insufficient
C3 hard boundary	runtime-bound state and coupled exposure	exclusive commit path, bypass inventory, enforcement evidence	FSEL is not the boundary

Criterion	FSEL/RCFA contribution	Required real evidence	Non-negotiable limit
C4 responsibility	authority coordinate and continuity	principal, role, scope, time, transition, revocation tests	RCFA transports but does not create authority
C5 deterministic gating	bound governance state and evidence	truth table, priority, halt dominance, governed recovery	the evaluated model may not control the gate

Passing the integration profile is necessary for the declared integration. It is not a substitute for complete SGDS membership evidence of a real system.

9.6 Deterministic Software Artifact Evidence

The v0.6 toolchain includes typed candidate and interface schemas, evaluation records, a report generator, exhaustive finite abstraction, version-drift invalidation, and automated tests.

The declared finite abstract domain contains

$$5 \times 5 \times 7 \times 9 \times 2 \times 2 \times 2 = 12{,}600 \tag{9.2}$$

canonical cases. Each case is evaluated twice. The executed suite reports:

- 12,600 total cases;
- 4 ALLOW;
- 2 THROTTLE;
- 2 DELAY;
- 12,592 HALT;
- 16 integration-conformant cases;
- 0 deterministic replay mismatches;
- 0 unsafe ALLOW results;
- 29 automated tests passed, 0 failed;
- version or threshold drift invalidated the prior record and forced HALT.

This evidence is strong for the delivered software artifact and the declared finite abstraction. It does not establish complete mediation, physical non-bypassability, authentic real authority, measured real load damping, or real deployment timing.

9.7 Live-Evidence Kit

The Live-Evidence Kit extends the abstract evaluator into a controlled field campaign. Public-safe artifact classes include:

- collector;
- schemas;
- runbooks;

- validators;
- protected and redacted records;
- execution summary;
- release status;
- preflight and hash ledger.

The real campaign must bind:

1. exact candidate boundary and fingerprint;
2. server and client versions;
3. topology and side-effect inventory;
4. live timestamps for fault, local halt, propagation, and effect denial;
5. authority expiry, revocation, wrong actor/device/case tests;
6. declared load variables and overload/damping measurements;
7. race, partition, retry, and recovery tests;
8. protected raw evidence and redacted public record;
9. deterministic C1-C5 evaluation from the frozen evidence root.

9.8 Current Campaign Status

As of 6 August 2026, the field kit is complete and the real KWI-LIAN campaign is in active execution in a parallel controlled workstream. The sequence begins with the server and primary Z16 client, followed by the two R730 candidate clients after the first boundary is stable.

This status must be stated precisely:

```
FIELD KIT READY                 CLOSED
SYNTHETIC ABSTRACT SUITE        PASS
REAL FIELD EXECUTION            ACTIVE
TERMINAL REAL C1-C5 RECORD      NOT YET FROZEN
REAL v0.7 EVIDENCE STATUS       OPEN
```

Active evidence acquisition is not the same as a closed result. Promotion is permitted only after the candidate, hashes, tests, raw records, redacted records, and C1-C5 evaluation are frozen and replayed.

9.9 Negative Evidence and Promotion Discipline

The field methodology treats negative results as authoritative. A process-only boundary that fails C3 cannot be softened by successful logical tests. C2 cannot be inferred from CPU usage or an unrelated numerical model. C4 challenge success does not independently prove legal authority. C5 requires measured timing and a separated gate.

The result logic is:

$$\mathrm{SGDS}(S) = \mathrm{CONFIRMED} \iff C_1 = \cdots = C_5 = \mathrm{PASS}. \tag{9.3}$$

Any FAIL rejects confirmation. Any INCONCLUSIVE leaves the candidate NOT CONFIRMED.

9.10 Propositions

Proposition 9.1 - Exposure non-authority. FSEL state exposure can inform governance but cannot authorize execution.

Proposition 9.2 - Continuity non-creation. RCFA preserves identity and authority bindings but cannot create authority that does not exist upstream.

Proposition 9.3 - Abstract-domain boundary. Exhaustiveness over the declared 12,600-case abstraction does not generalize to a real deployment without refinement and field evidence.

Proposition 9.4 - Drift invalidation. Candidate, configuration, threshold, transform, schema, or enforcement drift invalidates dependent conformance evidence until replay.

Proposition 9.5 - Live promotion gate. Real SGDS confirmation is unavailable until all C1-C5 criteria pass on the same frozen real candidate and evidence root.

9.11 Status

STRUCTURE. The layered integration contract, I1-I12 invariants, C1-C5 map, deterministic evaluator, and live-evidence route are specified.

EXECUTION REALITY. The finite abstract suite and field kit are executed and complete. The real KWI-LIAN field campaign is active and not terminally frozen.

FAILURE POINT. Transfer of synthetic results, incomplete side-effect inventory, unmeasured damping, unauthenticated authority, or unmeasured halt propagation prevents promotion.

GOVERNANCE STATUS. M3 for the declared software artifact and finite abstraction; real candidate status open.

CONFIDENCE TYPE. Directed artifact evidence with explicit real-system refinement boundary.

Chapter 10

Governance, Law, Data, Procurement, and Operator Sovereignty

10.1 Architecture Precedes Compliance

A compliance claim is not created by listing regulatory articles beside product features. A legal or normative obligation must be interpreted by a competent authority or institution, translated into system requirements, implemented at the correct boundary, tested against the actual deployment, and preserved through change.

The architecture therefore distinguishes:

$$\text{normative source} \rightarrow \text{authorized interpretation} \rightarrow \text{system requirement} \rightarrow \text{executable control} \rightarrow \text{evidence} \rightarrow \text{institutiona} \tag{10.1}$$

A break anywhere in this chain prevents an operational compliance conclusion. Architectural correspondence may show how a mechanism could support an obligation. It is not a legal determination.

10.2 Normative State as a Versioned Dependency

A normative rule must be represented with source, jurisdiction, applicability, version, interpretation, effective date, exceptions, and authority. Let

$$N = \langle source, jurisdiction, actor, scope, version, time, interpretation, exceptions \rangle. \tag{10.2}$$

A rule mapping is invalid when the authoritative source changes, the deployment role changes, the system purpose changes, or the interpretation is no longer current.

The system therefore requires reauthorization after relevant legal, policy, technical, or organizational drift.

10.3 EU AI Act Architectural Correspondence

KWI LIAN and the broader architecture may support public-safe mechanisms relevant to several governance expectations, including:

- risk-management state and evidence;
- event and record preservation;
- transparency of operational state;
- human oversight through a concrete commit boundary;
- traceability of actor, device, case, document, evidence, and decision state;
- local control and operator-managed infrastructure.

These mechanisms can be discussed in relation to the themes of risk management, record keeping, transparency, and human oversight commonly associated with Articles 9, 12, 13, and 14 of Regulation (EU) 2024/1689. This is an architectural correspondence only. Applicability, actor role, risk classification, conformity obligations, and legal sufficiency must be checked against the current consolidated legal source and competent interpretation for the actual deployment.

10.4 Rights, Duties, and Conflict of Norms

Governance must represent not only prohibitions but also duties, permissions, exceptions, rights, and remedy.

A normative decision may require resolution of:

- competing legal duties;
- privacy versus transparency;
- individual rights versus public-interest obligations;
- safety versus availability;
- disclosure versus security;
- retention versus erasure;
- delegated authority versus personal responsibility;
- emergency powers versus ordinary procedure.

A model-generated ranking cannot resolve these conflicts merely by confidence. The resolution path must be institutionally authorized and evidenced.

10.5 Data Governance and Provenance

Data governance requires purpose, source, access, retention, integrity, and transformation state. A data object may be represented as

$$D = \langle id, source, purpose, lawful\_basis, owner, custodian, access, retention, version, provenance \rangle. \tag{10.3}$$

Provenance must distinguish original data, derived data, model output, human edits, external evidence, and later transformations. A copied file is not automatically the same governed object. A renamed file is

not automatically a new object. Identity must be explicitly maintained.

Hash evidence can show byte-level or canonicalized-state correspondence. It does not prove lawful use, semantic truth, authorized purpose, or complete provenance by itself.

10.6 Procurement, Vendors, and External Dependencies

Procurement is part of system architecture because contracts and supplier control can change which transitions are reachable.

A supplier dependency ledger should include:

- service and component identity;
- critical functions;
- update and support route;
- remote-access conditions;
- key and credential custody;
- data and evidence location;
- export and portability;
- license and availability dependencies;
- subcontractors;
- incident obligations;
- termination and exit path;
- recovery without supplier permission.

A technically robust local system may still lack sovereignty if a vendor can disable operation, withhold keys, block recovery, or retain exclusive access to authoritative evidence.

10.7 Cloud and the Building Analogy

The relevant question is not simply which cloud provider is safest. The structural question is which operating model matches the organization's responsibility.

A cloud service can provide scalability, resilience, specialist security, and reduced administration. For many uses, that is the correct architecture. But the operator does not necessarily own the master keys, infrastructure, maintenance process, legal exposure, supplier chain, or recovery authority.

The analogy is a hotel versus an owned building:

> A hotel room may be well protected, but the guest does not own the master key, the building systems, or the operator's obligations. Trust may be justified, but trust is not the same as operator-controlled architecture.

The purpose of local KWI LIAN deployment is not to declare cloud intrinsically bad. It is to create an option in which document state, evidence, access, operation, and responsibility can remain inside infrastructure controlled by the receiving organization.

10.8 Operator Sovereignty

Operator sovereignty requires practical control over:

- deployment and configuration;
- identity and role administration;
- data and evidence custody;
- keys and secrets;
- backup and recovery;
- audit and export;
- update and rollback;
- supplier access;
- shutdown and decommissioning;
- continuity after contract termination.

An operator may deliberately delegate some of these functions. The delegation must remain visible as an external dependency and must not be confused with retained control.

10.9 Incident, Suspension, Withdrawal, and Remedy

A governed system must specify what happens after failure or suspicion. Incident response should preserve evidence and prevent further unauthorized effects without destroying the ability to understand what occurred.

Possible controls include:

- immediate halt or containment;
- revocation of actor, device, or capability;
- quarantine of document or evidence state;
- rollback to a known state;
- mandatory secondary review;
- suspension of a model, integration, or supplier path;
- notification and corrective action;
- recall or withdrawal;
- reauthorization after repair;
- appeal and contestability.

The incident path is itself consequence-bearing and must be governed. Emergency override cannot be an unbounded bypass.

10.10 Public Disclosure and Intellectual Property

Research transparency and security must be balanced through a controlled disclosure architecture.

Public material may include:

- principles and definitions;
- public-safe diagrams;
- theorem forms and assumptions;
- claim boundaries;
- maturity and non-claim statements;
- neutralized evidence summaries.

Controlled or protected material may include:

- thresholds and calibration;
- protected signal mappings;
- exploit and bypass tests;
- customer-specific topology;
- credentials and secrets;
- source code and deployment packages;
- raw evidence with personal or security-sensitive information.

Protected status does not establish novelty, correctness, or implementation. It controls exposure only.

10.11 Propositions

Proposition 10.1 - Compliance non-automaticity. Architectural support for a normative expectation does not constitute legal compliance or conformity determination.

Proposition 10.2 - Normative freshness. A regulatory or policy mapping expires when source, applicability, actor role, interpretation, or deployment context changes materially.

Proposition 10.3 - Supplier-dependency visibility. A governance claim must include vendor-controlled permissions, recovery paths, keys, evidence, and update dependencies that affect critical execution.

Proposition 10.4 - Locality as option, not proof. Local deployment may improve operator control but does not automatically establish security, lawfulness, resilience, or governance.

Proposition 10.5 - Remedy governance. Incident, override, recovery, recall, and appeal paths must themselves be represented as governed transitions.

10.12 Status

STRUCTURE. Normative translation, data provenance, procurement dependencies, operator sovereignty, remedy, and disclosure boundaries are specified.

EXECUTION REALITY. KWI LIAN provides a local reference architecture, but deployment-specific legal, security, procurement, and organizational evaluation remains mandatory.

FAILURE POINT. Stale law, hidden supplier authority, inaccessible evidence, unbounded emergency override, or compliance-by-feature-list defeats the claim.

GOVERNANCE STATUS. M1 architecture; no institutional legal or conformity determination claimed.

CONFIDENCE TYPE. Architectural correspondence; institutionally imported for legal claims.

Chapter 11

Validation, Falsification, Reproduction, and Evidence Maturity

11.1 Governed Research-Certification Method

The research programme follows a falsification-first method designed to prevent plausible narrative from being promoted as closed evidence.

The governing sequence is:

```
route audit
  -> dependency freeze
  -> exact structure extraction
  -> candidate construction
  -> directed certification
  -> adversarial stress
  -> claim adjudication
  -> artifact freeze
  -> controlled publication
```

The method is summarized by five rules:

1. route before computation;
2. exact structure before numerical enclosure;
3. directed certification before sampling;
4. promotion only after a closed gate;
5. immutable upstream freeze and explicit failure classification.

11.2 Route Audit

Before a major proof, implementation, or test campaign, the project identifies:

- the exact target claim;
- the smallest admissible theorem or engineering result;
- required upstream assumptions and artifacts;
- real structural bridges versus analogy-only connections;

- the open universal or deployment lift;
- the conditions that would defeat the route.

This prevents computation from producing impressive but irrelevant evidence.

11.3 Dependency Freeze

Every campaign freezes:

- candidate identity;
- source and binary versions;
- configuration and thresholds;
- topology and boundary;
- schemas and transforms;
- test plan;
- expected result states;
- public/protected disclosure class.

Hashes and manifests bind the candidate. A change to a mandatory upstream artifact invalidates dependent evidence until replay.

11.4 Exact Structure before Numerical Evidence

The preferred order is

```
exact identity
  -> exact decomposition
  -> directed enclosure
  -> numerical candidate
  -> heuristic evidence
```

An exact cancellation, shell, symmetry, coupling rule, or state identity must not be replaced unnecessarily by a coarse numerical estimate. In engineering, the analogous rule is to establish state and interface semantics before interpreting measurements.

11.5 Primary and Control Routes

A significant result should possess a primary route and an independent control or replay route where feasible.

Examples include:

- theorem proof plus model-checking replay;
- primary implementation test plus independent read-only validator;
- event collector plus separate schema and hash validator;
- directed interval calculation plus high-precision replay;
- live timing trace plus independent hardware timestamp source.

Agreement does not prove correctness if both routes share the same hidden defect. Independence must be stated, not assumed.

11.6 Result States

Permitted statuses include:

```
CANDIDATE
CONDITIONAL PASS
DIRECTED PASS
CERTIFIED
SUPPORTED
WITHHELD
FAILED
INCONCLUSIVE
OPEN
UNKNOWN
NOT TESTED
NOT AUTHORIZED
NOT CLAIMED
```

A failed gate is not rhetorically converted into success. `INCONCLUSIVE` is not a partial pass. `WITHHELD` does not imply positive evidence.

11.7 Claim-Artifact-Evidence-Maturity Ledger

Every promoted claim must link:

$$\boxed{\text{Claim} \rightarrow \text{Architecture} \rightarrow \text{Artifact} \rightarrow \text{Test} \rightarrow \text{Evidence} \rightarrow \text{Maturity} \rightarrow \text{Permitted statement}} \quad (11.1)$$

A minimum ledger entry contains:

- claim ID and text;
- exact object and scope;
- boundary and assumptions;
- mandatory dependencies;
- artifact IDs, versions, and hashes;
- test IDs and coverage;
- positive, negative, and inconclusive results;
- maturity level;
- public-safe statement;
- non-claims and open gates.

11.8 Validation Families

11.8.1 Formal verification

- invariant proofs;
- reachability and non-reachability;
- timed and hybrid analysis;
- composition and refinement;
- solver certificate preservation;
- assumption and complexity boundaries.

11.8.2 Software and runtime testing

- transaction identity;
- complete mediation and bypass tests;
- race and concurrent mutation;
- replay and stale state;
- crash and recovery;
- distributed effects and reconciliation;
- evidence and version integrity.

11.8.3 Hardware and physical testing

- truth table and halt dominance;
- power, clock, reset, watchdog;
- signal freshness and tamper;
- communication loss and replay;
- I/O and effect-boundary measurement;
- fault injection and environmental envelope;
- evidence preservation across reset.

11.8.4 Human authority and interface testing

- role and mandate correctness;
- presence and expiry;
- wrong actor/device/case challenges;
- workload and intervention timing;
- display-commit equivalence;
- accessibility and informed commitment;
- delegation, handoff, veto, and appeal.

11.8.5 Governance, legal, data, and supplier testing

- current-source verification;
- role and applicability mapping;
- data purpose and retention;
- supplier exit and portability;
- remote access and key custody;
- incident and remedy paths;
- public/protected disclosure compliance.

11.9 Coverage Ceilings

Every result has a coverage ceiling. A test proves only the tested property for the frozen object and envelope.

Examples:

- a unit test does not prove complete mediation;
- 12,600 exhaustive abstract cases do not prove a real deployment;
- a live server-client test does not prove physical non-bypassability;
- an FPGA demonstrator does not prove production qualification;
- a legal mapping does not prove institutional conformity;
- an invitation or conference presentation proves engagement, not technical validation.

11.10 Reproduction Package

A public or controlled reproduction package should contain:

- README and claim boundary;
- candidate and environment manifest;
- source or executable artifact as permitted;
- schema and interface contracts;
- test vectors or generators;
- validator;
- expected positive and negative results;
- execution log;
- checksums;
- release status;
- non-claims;
- protected-material index without protected payload.

A competent external party must be able to determine whether it reproduced the same object and result.

11.11 Independent Reproduction and Assessment

M4 reproduction requires a separate competent party to execute the declared route on the same or explicitly equivalent artifact and obtain a matching result. M5 assessment requires evaluation against an agreed assurance framework. M6 requires a formal determination by a competent institution.

Institutional recognition is not required to make a mathematical fact true. It is required when the claim itself concerns an institutional determination, certification, legal status, or regulated deployment.

11.12 External Engagement Evidence

External invitations, presentations, repository interest, and pilot discussions may demonstrate relevance, communication value, or willingness to evaluate. They are recorded as **external engagement evidence**, not as proof of the theory or implementation.

This distinction protects both the research and the inviting institution from accidental overclaim.

11.13 Separate Mathematical Certification Track

The directed Order-10 operator-chain work and related PRF research demonstrate the use of route audits, directed enclosures, primary/control paths, frozen artifacts, and explicit non-claims in a separate mathematical domain. That work is relevant to research method, not evidence that the governance architecture is implemented or correct.

Method transfer is admissible. Numerical constants and domain-specific theorem conclusions are not transferred automatically.

11.14 Current Evidence Ledger

Claim family	Current strongest public-safe status	Main open gate
Structural Sovereignty theory	M1 specified	mechanized end-to-end proof and external critique
Federated permit	M1 specified	domain-complete predicate and path closure
Context-bound authority	M1 general; M2/M3 bounded KWI components	external authority-source and role validation
Human Commit Boundary	M2/M3 bounded implementation	independent multi-user replay and human-factors study
LIAN document-state continuity	M3 internal bounded route	independent installation and canonicalization review
KWI local evidence infrastructure	M2 implemented; selected M3 tests	external pilot under receiving IT
SGDS-FSEL abstract evaluator	M3 declared artifact	real-system refinement
Real KWI-LIAN C1-C5	active field execution	frozen terminal record with all criteria adjudicated
CoChip architecture	M1 specified	FPGA/ASIC artifact and physical fault evidence
End-to-end upstream/downstream chain	structurally connected	live authenticated signal contract and hardware enforcement

11.15 Propositions

Proposition 11.1 - Coverage ceiling. A result establishes no property beyond the frozen object, scope, assumptions, and coverage actually tested or proved.

Proposition 11.2 - Negative-evidence retention. Failed and inconclusive results remain authoritative and must accompany any later narrowed or corrected claim.

Proposition 11.3 - Reproduction identity. Reproduction requires evidence that the external party evaluated the same or explicitly equivalent object and claim.

Proposition 11.4 - Engagement non-validation. External interest, invitation, or presentation does not constitute independent technical or institutional validation.

Proposition 11.5 - Method-only transfer. Certification architecture may transfer between domains; numerical constants and domain conclusions require independent derivation.

11.16 Status

STRUCTURE. The governed research-certification route, status system, validation families, ledger, and reproduction package are specified.

EXECUTION REALITY. Multiple bounded artifact packages exist. Independent end-to-end reproduction remains open.

FAILURE POINT. Unfrozen candidates, hidden failed results, transferred synthetic evidence, shared-defect control routes, or unverifiable reproduction identity defeat promotion.

GOVERNANCE STATUS. Method active and binding for this research programme.

CONFIDENCE TYPE. Procedural and artifact-dependent.

Chapter 12

Original Contributions, Limitations, and Research Programme

12.1 Original Contribution Structure

The contribution of this monograph is not that each individual concept is unknown. Formal methods, supervisory control, reference monitors, capability security, hardware roots of trust, safety kernels, human oversight, audit, provenance, risk management, and local infrastructure each possess established literatures.

The candidate originality lies in the specific federated architecture and its non-collapse rules.

12.1.1 Contribution 1 - Governance-execution gap

The monograph defines the decisive governance failure as the set of non-permitted consequence-bearing transitions that remain technically reachable inside a declared boundary.

12.1.2 Contribution 2 - Structural Sovereignty

Structural sovereignty is defined as operator or institutional control over transition reachability, including external permission dependencies, evidence custody, recovery, and remedy.

12.1.3 Contribution 3 - Federated permit

The permit is a conjunction of non-substitutable predicates from formal, control, authority, runtime, physical, context, version, fault, evidence, and remedy domains.

12.1.4 Contribution 4 - Non-collapse across disciplines

Mathematics, control, runtime, hardware, authority, law, and evidence retain separate validity conditions. Typed interfaces replace rhetorical composition.

12.1.5 Contribution 5 - Proposal-permission-commit-consequence separation

AI assistance, authority evaluation, human commitment, and effect are represented as distinct transitions.

12.1.6 Contribution 6 - Context-bound authority and Human Commit Boundary

Authority is derived from actor, device, case, object, evidence, role, scope, time, and commit context rather than model confidence or generic human presence.

12.1.7 Contribution 7 - KWI LIAN Kyber reference implementation

A bounded local document-governance system implements document-state verification, server-side shadow records, evidence, presence, commit, and handoff structures in LibreOffice Writer and Calc workflows.

12.1.8 Contribution 8 - Upstream/downstream architecture

KWI LIAN supplies the upstream authority and evidence state; runtime and CoChip enforcement act on the derived governance state without originating authority.

12.1.9 Contribution 9 - SGDS C1-C5 class

Governability is tested through representability, load-bounded damping, hard boundary, responsibility-anchored continuation, and deterministic gating.

12.1.10 Contribution 10 - FSEL/RCFA integration discipline

Controlled state exposure and route continuity contribute evidence without becoming authority or enforcement.

12.1.11 Contribution 11 - Evidence maturity and weakest-link rule

System maturity is bounded by the weakest mandatory dependency, and negative results remain part of the authoritative record.

12.1.12 Contribution 12 - Proof-preserving governed research method

Route audit, exact structure, directed primary/control verification, adversarial stress, immutable freezes, and controlled publication are integrated into a reusable research-certification method.

12.2 Novelty Boundaries

The following are not claimed as individually new:

- state machines and transition systems;
- non-injective or layered architectures;
- formal verification and model checking;
- reference monitors and complete mediation;
- capability security;
- control barrier functions and supervisory control;
- hardware watchdogs and fail-closed design;
- provenance and audit logs;
- human approval workflows;

- local server-client infrastructure;
- assurance cases and maturity models.

The novelty claim must therefore be supported by a dedicated related-work matrix showing which combinations, interfaces, and non-substitution rules are absent or materially different in prior work. Until that audit is complete, claims of novelty remain bounded candidates rather than institutional determinations.

12.3 Limitations

This monograph does not establish:

- universal governability of AI systems;
- complete discovery of all consequence paths;
- semantic truth of AI outputs;
- elimination of risk;
- general legal compliance;
- a fabricated production CoChip;
- physical non-bypassability;
- safe-state reachability for arbitrary plants;
- an independently reproduced end-to-end KWI-LIAN-CoChip chain;
- confirmed SGDS status for the real KWI-LIAN candidate before the active campaign closes;
- human-factors adequacy under all workloads and accessibility needs;
- security against every adversary;
- certification for automotive, medical, administrative, or critical-infrastructure deployment;
- university-granted habilitation or degree status.

The monograph is strongest when it states a narrower claim precisely rather than a wider claim rhetorically.

12.4 Mandatory Research Programme

12.4.1 R1 - Mechanized federated-permit proof

Formalize the permit, dependency graph, timed validity, composition, and non-reachability conditions in a mechanized proof environment.

12.4.2 R2 - Consequence-path discovery

Develop a systematic inventory method covering ordinary, administrative, recovery, supplier, manual, and physical paths.

12.4.3 R3 - KWI-LIAN live C1-C5 closure

Complete the server and Z16 campaign, then the approved R730 candidates, freeze the candidate and evidence root, and adjudicate C1-C5 without transferring abstract results.

12.4.4 R4 - Independent KWI installation

Provide a neutralized package and runbook to a competent external IT environment for M4 reproduction of the declared Writer, Calc, shadow, presence, and evidence routes.

12.4.5 R5 - Human-factors validation

Measure intervention margin, workload, comprehension, accessibility, handoff, veto, and display-commit equivalence.

12.4.6 R6 - CoChip FPGA demonstrator

Implement the frozen public-safe enforcement logic in FPGA or equivalent hardware, preserve protected signal detail, and execute timing and fault-injection campaigns.

12.4.7 R7 - Upstream/downstream signal contract

Bind real KWI LIAN authority, presence, commit, state, evidence, freshness, and transaction identity to the independent enforcement kernel.

12.4.8 R8 - Safe-state reachability

For each physical profile, establish plant-specific disturbance, delay, actuator, and safe-state conditions.

12.4.9 R9 - Related-work and novelty audit

Compare the architecture against supervisory control, runtime assurance, reference monitors, safety kernels, assurance cases, provenance systems, human-oversight frameworks, and AI governance standards.

12.4.10 R10 - Legal and institutional mapping

Obtain current jurisdiction-specific review by competent legal, regulatory, and institutional parties for actual use cases.

12.4.11 R11 - Supplier-exit and portability tests

Test operation, evidence access, state export, recovery, and continuity after loss of a supplier or external service.

12.4.12 R12 - Public/controlled/protected publication system

Maintain a public theory volume, controlled evidence companion, and protected technical annex with synchronized version and claim ledgers.

12.5 Publication Architecture

The reconstructed research programme should be maintained as three coordinated objects.

12.5.1 Volume I - Public research monograph

The present document: theory, public-safe architecture, bounded evidence summaries, claims, non-claims, and research programme.

12.5.2 Volume II - Evidence companion

Claim ledger, artifact ledger, test IDs, hashes, screenshots, positive and negative results, live campaign records, reproduction instructions, and status freezes.

12.5.3 Volume III - Controlled technical annex

Implementation detail, schemas, protected signal mappings, fault traces, source packages, deployment topology, calibration, and controlled review material.

No volume may silently widen the claims of another.

12.6 Final Thesis

Structural Sovereignty is a candidate federated theory of governability in which authority, admissibility, feasibility, mediation, physical enforcement, evidence, version, fault envelope, and remedy jointly determine whether a consequence-bearing transition can exist.

The final thesis is:

A system is governed only to the extent that every protected consequence path depends on a complete, current, evidenc

(12.1)

The theory is specified in falsifiable, cross-disciplinary form. Selected components are implemented and internally tested. The complete system remains conditional on the proof, implementation, field evidence, human authority, physical enforcement, independent reproduction, and institutional obligations identified throughout this monograph.

12.7 Final Status

STRUCTURE. A coherent 12-chapter federated architecture has been reconstructed from the source corpus and later implementation/evidence work.

EXECUTION REALITY. KWI LIAN Kyber and SGDS-FSEL contain bounded executable artifacts. CoChip physical evidence and end-to-end independent reproduction remain open.

FAILURE POINT. Any claim that outruns its weakest mandatory dependency, hides negative evidence, or confuses architecture with certification is inadmissible.

GOVERNANCE STATUS. Integrated theory M1; selected components M2/M3; no system-level M4-M6 claim.

CONFIDENCE TYPE. Definitional, formal under assumptions, engineering where implemented, empirical where tested, and institutional only when externally determined.

Appendix A

Common Axiom and Invariant Register

A.1 Axioms

AX-01 Execution-path closure. If a transition is inadmissible, its execution path does not exist inside the declared enforcement boundary.

AX-02 Model non-authority. The evaluated model may not originate, expand, or restore its own execution authority.

AX-03 Unknown non-permission. Unknown mandatory state is not ordinary permission.

AX-04 Architecture before compliance. Operational compliance claims require implemented and evidenced control, not feature correspondence alone.

AX-05 Safety/liveness separation. Halt or safety does not automatically establish safe-state arrival, liveness, availability, or usefulness.

AX-06 Non-substitution. Evidence, authority, proof, version, object, and boundary may not be silently substituted.

AX-07 Weakest mandatory dependency. Composite claim maturity is bounded by the weakest necessary dependency.

AX-08 Negative-evidence retention. Failed and inconclusive results remain authoritative.

AX-09 Disclosure non-expansion. Protected detail does not widen public claim scope.

AX-10 Remedy inclusion. Recovery, override, incident, appeal, and recall paths are part of the governed object.

A.2 Integration Invariants I1-I12

I1 projection non-authority; I2 common-mode conservatism; I3 persistent identity; I4 address-state separation; I5 governed release; I6 no load shifting; I7 exclusive commit path; I8 HALT dominance; I9 recovery re-binding; I10 version identity; I11 evidence non-substitution; I12 model non-self-extension.

Appendix B

Public-Safe Terminology

Term	Public-safe meaning
Structural Sovereignty	operator or institutional control over which consequence-bearing transitions are reachable inside a declared boundary
Execution-bound governance	governance in which the consequence path depends on the valid governance state
Governance-Execution Gap	non-permitted transitions that remain reachable
Federated Permit	conjunction of all mandatory non-substitutable predicates for the exact transition
Context-bound authority	authority derived from actor, role, device, case, object, evidence, scope, time, and commit context
Human Commit Boundary	boundary where human authority is attached to an exact state and becomes operationally meaningful
LIAN	document/table state verification against the last evidence-bound state
Kyber	LibreOffice Writer/Calc user-facing governance extension
KWI	server-side record, evidence, document shadow, device, event, and commit infrastructure
Document Shadow	preserved representation of a known document state at a governance-relevant point
Evidence continuity	maintained relation among artifact, event, commit, version, and later auditability
SGDS	candidate class of Structurally Governable Dynamical Systems defined by C1-C5
FSEL	controlled exposure of common, axis-specific, and coupled state deviations
RCFA	route-centered continuity preserving identity, authority, evidence, and reconstructable position
Governed CoChip	independent deterministic hardware-enforcement architecture acting on externally grounded governance state
HALT dominance	halt overrides utility, confidence, prior allow, and model preference

Appendix C

Claim-Artifact-Evidence-Maturity Ledger

ID	Claim	Artifact basis	Current status	Open gate
C-01	execution gap and federated permit	monograph/source corpus	M1	mechanized proof and critique
C-02	model output is not authority	KWI LIAN architecture	M1	authority validation
C-03	Human Commit Boundary	Kyber/KWI records	M2/M3	independent multi-user replay
C-04	LIAN drift detection	Writer/Calc tests	M3 internal	external install and canonicalization review
C-05	local operator workflow	KWI P1 package	M2	pilot under receiving IT
C-06	deterministic SGDS-FSEL evaluator	v0.6 suite and 29 tests	M3	real-system refinement
C-07	real KWI-LIAN SGDS status	active field campaign	OPEN	frozen C1-C5 record
C-08	deterministic CoChip architecture	public-safe paper	M1	FPGA/ASIC fault campaign
C-09	upstream-to-downstream chain	KWI LIAN + CoChip	M1 connected	live signal contract and hardware gate
C-10	EU AI Act support mapping	public-safe mapping	REG correspondence	competent legal assessment

Current Release and Evidence Status

SOURCE CORPUS	FROZEN / PRESERVED
RECONSTRUCTED MONOGRAPH v2.0	PUBLIC-SAFE ACADEMIC REVIEW DRAFT
KWI LIAN P1	BOUNDED IMPLEMENTATION
WRITER/CALC VERIFIED-MISMATCH	INTERNALLY TESTED ROUTE
SGDS-FSEL v0.6 FIELD KIT	COMPLETE
SGDS-FSEL ABSTRACT SUITE	PASS ON DECLARED FINITE DOMAIN
REAL KWI-LIAN FIELD EXECUTION	ACTIVE
REAL TERMINAL C1-C5 RESULT	NOT YET FROZEN
COCHIP PUBLIC ARCHITECTURE	SPECIFIED
PHYSICAL FPGA/ASIC EVIDENCE	OPEN
END-TO-END INDEPENDENT REPLAY	OPEN
LEGAL/CONFORMITY DETERMINATION	NOT CLAIMED

Open Research Obligations

1. Mechanized proof of the federated permit and non-reachability invariant.
2. Complete consequence-path inventory including administrative, recovery, supplier, manual, and physical paths.
3. Real KWI-LIAN C1-C5 evidence freeze.
4. Independent KWI LIAN installation and replay.
5. Human-factors and accessibility validation of the commit boundary.
6. FPGA or equivalent CoChip demonstrator.
7. Physical timing, reset, watchdog, tamper, and bypass evidence.
8. Plant-specific safe-state reachability.
9. Supplier exit, portability, and external-permission testing.
10. Current legal and institutional mapping by competent parties.
11. Related-work and novelty adjudication.
12. Independent academic critique and adversarial falsification.

Selected References

1. Alur, R., and Dill, D. L. *A Theory of Timed Automata*. Theoretical Computer Science, 1994.
2. Alur, R. *Principles of Cyber-Physical Systems*. MIT Press, 2015.
3. Ames, A. D., Xu, X., Grizzle, J. W., and Tabuada, P. Control barrier function based quadratic programs for safety critical systems. *IEEE Transactions on Automatic Control*, 2017.
4. Anderson, R. *Security Engineering*. Wiley.
5. Clarke, E. M., Grumberg, O., and Peled, D. A. *Model Checking*. MIT Press, 1999.
6. ISO 26262:2018. *Road Vehicles - Functional Safety*.
7. ISO 21448:2022. *Road Vehicles - Safety of the Intended Functionality*.
8. ISO/IEC 42001:2023. *Artificial Intelligence Management System*.
9. Khalil, H. K. *Nonlinear Systems*, 3rd edition. Prentice Hall, 2002.
10. Lamport, L. Time, clocks, and the ordering of events in a distributed system. *Communications of the ACM*, 1978.
11. Leveson, N. G. *Engineering a Safer World*. MIT Press, 2011.
12. Rawlings, J. B., Mayne, D. Q., and Diehl, M. *Model Predictive Control: Theory, Computation, and Design*, 2017.
13. Saltzer, J. H., and Schroeder, M. D. The protection of information in computer systems. *Proceedings of the IEEE*, 1975.
14. Tabuada, P. *Verification and Control of Hybrid Systems: A Symbolic Approach*. Springer, 2009.
15. W3C. *PROV-O: The PROV Ontology*, 2013.
16. Regulation (EU) 2024/1689 of the European Parliament and of the Council laying down harmonised rules on artificial intelligence.
17. Senke, A. *Structural Sovereignty and Execution-Bound Governance*. Integrated source corpus, 2026.
18. Senke, A. *Hardware-Enforced Admissibility Boundaries for Automotive AI Execution: A Structural Governance Architecture*. Public-safe research paper, 2026.
19. Senke, A. *Context-Bound Authority Signals for Pre-Execution AI Governance: KWI LIAN as an Upstream Governance Layer*. Preprint candidate, 2026.
20. Senke, A. *KWI LIAN Infrastructure Edition - P1 Neutralized Overview*. Internal/public-safe project material, 2026.
21. Senke, A. *SGDS-FSEL Integration Specification v0.6*. Internal governed research artifact, 2026.
22. Senke, A. *Structurally Governable Dynamical Systems*. Related controlled research record, DOI 10.5281/zenodo.21569313.
23. Senke, A. *FSEL-01 / RCFA-01 Six-Dimensional Control-Fiber Architecture*. Related controlled

research record, DOI 10.5281/zenodo.21610222.

24. Senke, A. *SLI Co-Chip FPGA Architecture*. Related research record, DOI 10.5281/zenodo.19076268.

Final Claim Boundary

This monograph establishes a reconstructed integrated M1 architecture and a bounded research programme. Selected software and workflow components possess M2 implementation and M3 internal-test evidence. The complete federated system has not been independently reproduced end to end, independently assessed, legally determined, fabricated as production hardware, or certified for safety-critical deployment.

The strongest justified statement is:

> Structural Sovereignty is a candidate federated theory of governability in which authority, admissibility, feasibility, mediation, physical enforcement, evidence, version, fault envelope, and remedy jointly determine whether a consequence-bearing transition can exist. The theory is specified in a falsifiable and cross-disciplinary form. Deployment-level validity remains conditional on the proof, implementation, validation, authority, and institutional obligations identified in this monograph.

STRUCTURAL SOVEREIGNTY AND EXECUTION-BOUND GOVERNANCE

A Federated Theory of Admissibility, Authority, Stability, Enforcement, and Evidence in Artificial Intelligence Systems

Alexanja Senke

AI Cognition Strategist - Human-AI Co-Intelligence Architect

Independent Architectural Research - Interlink Bridge

Germany - 2026

Integrated Master Habilitation Manuscript

Public-safe academic review edition

Status: research monograph; not a university-granted habilitation, legal opinion, certification, or deployment approval

Declaration of Authorship and Status

This manuscript presents an integrated independent research contribution in artificial-intelligence systems architecture and structural governance theory. It consolidates and corrects material developed in the 250-page internal source corpus *Source Corpus 2026 - Structural Sovereignty and Execution Governance*, together with the subsequently developed formal, control-theoretic, runtime, hardware, authority, governance, applied-system, and validation volumes.

The manuscript uses the term *Master Habilitation* as the title of an integrated academic dossier. Formal habilitation status can be granted only through the applicable university procedure. No such institutional status is claimed here.

Conceptual classifications, named architectural objects, system taxonomies, theorem candidates, validation structures, and public-safe formulations attributed to Alexanja Senke are presented as original research contributions unless another source is cited. Implementation-sensitive mechanisms remain protected by design. Their omission must not be interpreted as proof of implementation, and protected status may never be used to conceal an unsupported claim or failed result.

Disclosure Boundary

This edition is intentionally split into three disclosure layers:

Public theory: definitions, architecture, theorem forms, assumptions, non-claims, validation methods, and research programme.

Controlled technical evidence: detailed implementation, fault traces, integration artefacts, and test material suitable for NDA or institutionally controlled review.

Protected internal mechanisms: credentials, exploit paths, bypass-sensitive signal detail, proprietary integration logic, and security-relevant operational data.

Only the first layer is included in full. Controlled material is referenced by evidence class rather than disclosed. The manuscript therefore supports academic inspection of the theory without claiming that undisclosed implementation properties have already been independently proven.

Regulatory Currency Note

This manuscript was finalised on **1 August 2026**. Regulation (EU) 2024/1689 provides a staged application schedule; its general application date is 2 August 2026, with specified provisions applying on different dates. Every legal or regulatory use of this manuscript must be checked against the current official consolidated source and competent interpretation at the time of use. Standards mappings are non-normative architectural correspondences, not conformity determinations.[1]

1 Regulation (EU) 2024/1689, official EUR-Lex record, ELI: https://data.europa.eu/eli/reg/2024/1689/oj

Abstract

Artificial-intelligence systems increasingly produce proposals that can cross software, organisational, and physical boundaries. Contemporary governance commonly addresses those proposals through alignment, monitoring, risk classification, documentation, human oversight, and post-event audit. These mechanisms are important but structurally incomplete when a non-permitted transition remains technically reachable.

This manuscript introduces **Structural Sovereignty** as an execution-bound theory of governability. The core thesis is that governance becomes operational only when authority, admissibility, timing, control feasibility, runtime mediation, hardware restraint, human commitment, institutional mandate, evidence, and remedy jointly determine whether a consequence-bearing transition can exist. The central invariant is:

$$Exec(\tau) \Rightarrow Permit^{¿}(\tau)$$

where $Permit^{¿}$ denotes a federated permit whose mandatory predicates are valid across the declared model, authority, control, runtime, hardware, institutional, fault, and evidence envelopes.

The work develops a candidate class of **Structurally Governable Dynamical Systems (SGDS)**; formal governed transition systems; robust admissibility under partial observation; compositional contracts; timed and hybrid governance; control allocation and safe-state reachability; proposal-permission-commit separation; transaction identity; capability mediation; crash and distributed-state continuity; hardware-enforced halt; human and institutional authority; normative source-to-control translation; applied-system patterns; and a falsification-first validation programme.

The contribution is federated rather than reductionist. Mathematics does not generate legal mandate. Human presence does not establish authority. A policy does not enforce itself. A model is not the governance layer. A hardware signal does not establish institutional legitimacy. A certificate does not prove a deployment beyond its scope. Each discipline retains its own proof obligations while exporting typed interfaces to the others.

The manuscript does not claim universal applicability, complete risk elimination, final legal compliance, a fabricated production Co-Chip, vehicle certification, or deployment-level proof. It defines an inspectable architecture and research programme in which inadmissible transitions can be made unreachable inside a declared boundary, and in which every broader claim remains limited by its weakest mandatory dependency.

Keywords

Structural Sovereignty; execution-bound governance; admissibility; governed transition systems; SGDS; authority binding; human commit; runtime enforcement; non-bypassability; hardware halt; control barrier functions; evidence architecture; institutional governance; AI assurance; falsification.

Reader Orientation

The manuscript is organised as a federation of ten volumes. Readers may enter through their discipline, but system-level claims require the complete dependency chain.

Mathematics and formal methods: Parts I-II and IX.

Control theory: Part III and the hybrid sections of Part II.

Computer science and systems architecture: Part IV.

Hardware and automotive systems: Part V.

Human factors, authority, and institutions: Part VI.

Governance, law, data, procurement, audit, and assurance: Part VII.

Applied implementation: Part VIII.

Validation and independent reproduction: Part IX.

Contribution, limits, and research programme: Part X.

Claim Markers and Evidence Maturity

Claims are typed as follows:

DEF - definition.

AX - adopted architectural axiom.

PROP - derived proposition.

THEOREM - deductive statement under explicit assumptions.

INV - candidate invariant.

ENG - engineering requirement.

EMP - empirical claim.

REG - regulatory or normative mapping requiring current authoritative verification.

CASE - case-study evidence.

HYP - hypothesis.

OPEN - unresolved research obligation.

Evidence maturity is represented by:

M0 - Conceptual: an idea or unstructured claim.

M1 - Specified: definitions, scope, assumptions, and obligations exist.

M2 - Implemented: a bounded implementation exists.

M3 - Internally Tested: repeatable internal evidence exists.

M4 - Independently Reproduced: a separate competent party reproduces the result.

M5 - Independently Assessed: an independent assessment evaluates the claim and evidence.

M6 - Institutionally Determined: a competent institution has made the relevant formal determination.

A system-level claim cannot exceed the maturity of its weakest mandatory dependency:

$$M\left(C_{system}\right) \leq \min_{i} M\left(E_i\right)$$

Core Axioms

If a transition is inadmissible, its execution path does not exist inside the declared enforcement boundary.

If critical execution depends on discretionary external permission, the institution does not fully govern that operation.

If a conclusion survives only through plausibility, omitted assumptions, or narrative confidence, it is structurally invalid.

Architecture precedes an operational compliance claim.

A model is not the governance layer.

Unknown mandatory state is not ordinary permission.

Safety does not imply liveness, and halt does not automatically imply safe-state arrival.

Public claims remain public-safe; protected detail does not widen claim scope.

Master Federated Permit

The integrated permit is represented as:

$$\begin{aligned} Permit^{¿}(\tau) = ¿\, & Admissible_{formal} \wedge Feasible_{control} \wedge Authorized_{human/institution} \\ & \wedge Mediated_{runtime} \wedge Enforced_{hardware} \wedge Applicable_{normative} \\ & \wedge Current_{version,time,context} \wedge Within_{faultenvelope} \wedge EvidenceReady. \end{aligned}$$

This conjunction is deliberately strict. No individual predicate substitutes for the others.

Table of Contents

Page numbers correspond to the verified PDF layout. Alternate renderers may paginate differently.

PART I - COMMON ARCHITECTURAL CORE

Part I establishes the common architectural core. It defines the research object before any discipline-specific proof, implementation, or regulatory mapping is attempted. Its function is to prevent semantic drift across the later volumes.

Part-level claim boundary. The material in this part is specified at M1 unless a bounded case study explicitly records internal implementation evidence. No part-level statement independently widens the system-level claim.

Chapter 1

Research Object, Problem Definition, and Claim Architecture

The governance-execution gap, the research object, the disclosure boundary, and the distinction between architectural, empirical, legal, and engineering claims.

1.1 Objective

The objective of this chapter is to define research object, problem definition, and claim architecture as a bounded component of execution-bound governance. The analysis connects state-transition-consequence identity, pre-execution admissibility, authority binding, evidence maturity, fault-relative scope, and public-safe disclosure. The decisive test is not whether the system can describe the relevant rule, but whether the rule changes the set of reachable consequence-bearing transitions. This chapter operates at the common-core level and therefore avoids discipline-specific claims that would collapse mathematics, engineering, human authority, and law into one vocabulary.

1.2 Structural Model

The structural object is represented as a typed relation rather than an informal checklist. Identity, object, context, authority, version, time, fault state, and evidence are separated because a change in any one of them can create a different transition. The architecture therefore rejects implicit substitution: a permission for one object is not a permission for another; a proof for one model is not a proof for a deployment; an approval before mutation is not approval after mutation. Unknown mandatory state is represented explicitly and does not default to ordinary allow.

The chapter’s principal structural commitments are:

State-Transition-Consequence Identity - retained as an explicit state or dependency.

Pre-Execution Admissibility - retained as an explicit state or dependency.

Authority Binding - retained as an explicit state or dependency.

Evidence Maturity - retained as an explicit state or dependency.

Fault-Relative Scope - retained as an explicit state or dependency.

Public-Safe Disclosure - retained as an explicit state or dependency.

1.3 Execution Reality

Execution reality begins at the last point from which the protected consequence can still be prevented. Upstream analysis, documentation, monitoring, or recommendation may improve decision quality, but it becomes governance only when the consequence path depends on the resulting state. The chapter therefore treats direct capability, alternate interfaces, administrative paths, recovery mechanisms, stale state, and external dependencies as part of the governed object. A system is not credited with a property merely because a dashboard displays it; the relevant gate, controller, institution, or physical interface must enforce or evidence it.

1.4 Failure Point and Counter-Construction

The claim collapses under any in-scope counter-construction such as:

policy is mistaken for enforcement

a model output is treated as authority

scope and assumptions are omitted

evidence maturity is overstated

These failures are not edge cases external to the theory. They are the primary falsification instruments. Where one of them remains reachable, the correct result is a narrowed claim, an unresolved state, or a failed invariant - not a narrative assertion that the architecture is nevertheless governed.

1.5 Verification and Evidence Obligations

define the claim envelope

identify authority and consequence boundaries

separate public, protected, and institutionally imported assertions

assign an evidence maturity state

Evidence must remain bound to the exact artifact, configuration, version, operating envelope, and decision state examined. A passing result establishes only the tested property and coverage. A failed or indeterminate result remains part of the authoritative record. Where independent reproduction has not occurred, the result is not represented as independently established.

1.6 Selected Propositions

P1.1. The validity of research object, problem definition, and claim architecture cannot exceed the scope of its weakest mandatory dependency.

P1.2. A descriptive or advisory representation does not become an execution constraint until it controls the relevant consequence path.

P1.3. Component-level validity does not compose automatically across interfaces, timing, authority, failure, or deployment change.

P1.4. Evidence of absence is not inferred from absence of evidence; unresolved mandatory state remains unresolved.

1.7 Governance Status and Confidence Type

STRUCTURE. Specified as part of the integrated federated architecture.
EXECUTION REALITY. Requires a domain-specific implementation and complete path inventory.
FAILURE POINT. Any surviving bypass, stale authority, omitted dependency, unmodeled timing path, or unsupported evidence link narrows or defeats the claim.
GOVERNANCE STATUS. M1 unless a chapter-specific case study states a higher internal implementation maturity; no institutional compliance determination is inferred.
CONFIDENCE TYPE. Definitional for the architecture, deductive only under stated assumptions, implementation-dependent for runtime and hardware, institutionally imported for authority, and evidence-dependent for deployment claims.

Chapter 2

The Governance-Execution Gap

Why downstream policy, monitoring, and explanation cannot by themselves determine which consequence-bearing transitions are reachable.

2.1 Objective

The objective of this chapter is to define the governance-execution gap as a bounded component of execution-bound governance. The analysis connects state-transition-consequence identity, pre-execution admissibility, authority binding, evidence maturity, fault-relative scope, and public-safe disclosure. The decisive test is not whether the system can describe the relevant rule, but whether the rule changes the set of reachable consequence-bearing transitions. This chapter operates at the common-core level and therefore avoids discipline-specific claims that would collapse mathematics, engineering, human authority, and law into one vocabulary.

2.2 Structural Model

The structural object is represented as a typed relation rather than an informal checklist. Identity, object, context, authority, version, time, fault state, and evidence are separated because a change in any one of them can create a different transition. The architecture therefore rejects implicit substitution: a permission for one object is not a permission for another; a proof for one model is not a proof for a deployment; an approval before mutation is not approval after mutation. Unknown mandatory state is represented explicitly and does not default to ordinary allow.

The chapter’s principal structural commitments are:

State-Transition-Consequence Identity - retained as an explicit state or dependency.

Pre-Execution Admissibility - retained as an explicit state or dependency.

Authority Binding - retained as an explicit state or dependency.

Evidence Maturity - retained as an explicit state or dependency.

Fault-Relative Scope - retained as an explicit state or dependency.

Public-Safe Disclosure - retained as an explicit state or dependency.

2.3 Execution Reality

Execution reality begins at the last point from which the protected consequence can still be prevented. Upstream analysis, documentation, monitoring, or recommendation may improve decision quality, but it becomes governance only when the consequence path depends on the resulting state. The chapter therefore treats direct capability, alternate interfaces, administrative paths, recovery mechanisms, stale state, and external dependencies as part of the governed object. A system is not credited with a property merely because a dashboard displays it; the relevant gate, controller, institution, or physical interface must enforce or evidence it.

2.4 Failure Point and Counter-Construction

The claim collapses under any in-scope counter-construction such as:

policy is mistaken for enforcement

a model output is treated as authority

scope and assumptions are omitted

evidence maturity is overstated

These failures are not edge cases external to the theory. They are the primary falsification instruments. Where one of them remains reachable, the correct result is a narrowed claim, an unresolved state, or a failed invariant - not a narrative assertion that the architecture is nevertheless governed.

2.5 Verification and Evidence Obligations

define the claim envelope

identify authority and consequence boundaries

separate public, protected, and institutionally imported assertions

assign an evidence maturity state

Evidence must remain bound to the exact artifact, configuration, version, operating envelope, and decision state examined. A passing result establishes only the tested property and coverage. A failed or indeterminate result remains part of the authoritative record. Where independent reproduction has not occurred, the result is not represented as independently established.

2.6 Selected Propositions

P2.1. The validity of the governance-execution gap cannot exceed the scope of its weakest mandatory dependency.

P2.2. A descriptive or advisory representation does not become an execution constraint until it controls the relevant consequence path.

P2.3. Component-level validity does not compose automatically across interfaces, timing, authority, failure, or deployment change.

P2.4. Evidence of absence is not inferred from absence of evidence; unresolved mandatory state remains unresolved.

2.7 Governance Status and Confidence Type

STRUCTURE. Specified as part of the integrated federated architecture.
EXECUTION REALITY. Requires a domain-specific implementation and complete path inventory.
FAILURE POINT. Any surviving bypass, stale authority, omitted dependency, unmodeled timing path, or unsupported evidence link narrows or defeats the claim.
GOVERNANCE STATUS. M1 unless a chapter-specific case study states a higher internal implementation maturity; no institutional compliance determination is inferred.
CONFIDENCE TYPE. Definitional for the architecture, deductive only under stated assumptions, implementation-dependent for runtime and hardware, institutionally imported for authority, and evidence-dependent for deployment claims.

Chapter 3

The Structural Object of Governance

State, transition, consequence, authority, evidence, version, and fault envelope as the minimum governed object.

3.1 Objective

The objective of this chapter is to define the structural object of governance as a bounded component of execution-bound governance. The analysis connects state-transition-consequence identity, pre-execution admissibility, authority binding, evidence maturity, fault-relative scope, and public-safe disclosure. The decisive test is not whether the system can describe the relevant rule, but whether the rule changes the set of reachable consequence-bearing transitions. This chapter operates at the common-core level and therefore avoids discipline-specific claims that would collapse mathematics, engineering, human authority, and law into one vocabulary.

3.2 Structural Model

A compact representation used in this chapter is:

$$z_t = \left(x_t, c_t, o_t, p_t, a_t, r_t, e_t, v_t, k_t, h_t, w_t \right)$$

The structural object is represented as a typed relation rather than an informal checklist. Identity, object, context, authority, version, time, fault state, and evidence are separated because a change in any one of them can create a different transition. The architecture therefore rejects implicit substitution: a permission for one object is not a permission for another; a proof for one model is not a proof for a deployment; an approval before mutation is not approval after mutation. Unknown mandatory state is represented explicitly and does not default to ordinary allow.

The chapter’s principal structural commitments are:

State-Transition-Consequence Identity - retained as an explicit state or dependency.

Pre-Execution Admissibility - retained as an explicit state or dependency.

Authority Binding - retained as an explicit state or dependency.

Evidence Maturity - retained as an explicit state or dependency.

Fault-Relative Scope - retained as an explicit state or dependency.

Public-Safe Disclosure - retained as an explicit state or dependency.

3.3 Execution Reality

Execution reality begins at the last point from which the protected consequence can still be prevented. Upstream analysis, documentation, monitoring, or recommendation may improve decision quality, but it becomes governance only when the consequence path depends on the resulting state. The chapter therefore treats direct capability, alternate interfaces, administrative paths, recovery mechanisms, stale state, and external dependencies as part of the governed object. A system is not credited with a property merely because a dashboard displays it; the relevant gate, controller, institution, or physical interface must enforce or evidence it.

3.4 Failure Point and Counter-Construction

The claim collapses under any in-scope counter-construction such as:

policy is mistaken for enforcement

a model output is treated as authority

scope and assumptions are omitted

evidence maturity is overstated

These failures are not edge cases external to the theory. They are the primary falsification instruments. Where one of them remains reachable, the correct result is a narrowed claim, an unresolved state, or a failed invariant - not a narrative assertion that the architecture is nevertheless governed.

3.5 Verification and Evidence Obligations

define the claim envelope

identify authority and consequence boundaries

separate public, protected, and institutionally imported assertions

assign an evidence maturity state

Evidence must remain bound to the exact artifact, configuration, version, operating envelope, and decision state examined. A passing result establishes only the tested property and coverage. A failed or indeterminate result remains part of the authoritative record. Where independent reproduction has not occurred, the result is not represented as independently established.

3.6 Selected Propositions

P3.1. The validity of the structural object of governance cannot exceed the scope of its weakest mandatory dependency.

P3.2. A descriptive or advisory representation does not become an execution constraint until it controls the relevant consequence path.

P3.3. Component-level validity does not compose automatically across interfaces, timing, authority, failure, or deployment change.

P3.4. Evidence of absence is not inferred from absence of evidence; unresolved mandatory state remains unresolved.

3.7 Governance Status and Confidence Type

STRUCTURE. Specified as part of the integrated federated architecture.
EXECUTION REALITY. Requires a domain-specific implementation and complete path inventory.
FAILURE POINT. Any surviving bypass, stale authority, omitted dependency, unmodeled timing path, or unsupported evidence link narrows or defeats the claim.
GOVERNANCE STATUS. M1 unless a chapter-specific case study states a higher internal implementation maturity; no institutional compliance determination is inferred.
CONFIDENCE TYPE. Definitional for the architecture, deductive only under stated assumptions, implementation-dependent for runtime and hardware, institutionally imported for authority, and evidence-dependent for deployment claims.

Chapter 4

Admissibility as a Pre-Execution Relation

Permit, defer, handoff, throttle, and halt as typed transition states rather than post-hoc labels.

4.1 Objective

The objective of this chapter is to define admissibility as a pre-execution relation as a bounded component of execution-bound governance. The analysis connects state-transition-consequence identity, pre-execution admissibility, authority binding, evidence maturity, fault-relative scope, and public-safe disclosure. The decisive test is not whether the system can describe the relevant rule, but whether the rule changes the set of reachable consequence-bearing transitions. This chapter operates at the common-core level and therefore avoids discipline-specific claims that would collapse mathematics, engineering, human authority, and law into one vocabulary.

4.2 Structural Model

A compact representation used in this chapter is:

$$Exec(\tau) \Rightarrow Permit(\tau)$$

The structural object is represented as a typed relation rather than an informal checklist. Identity, object, context, authority, version, time, fault state, and evidence are separated because a change in any one of them can create a different transition. The architecture therefore rejects implicit substitution: a permission for one object is not a permission for another; a proof for one model is not a proof for a deployment; an approval before mutation is not approval after mutation. Unknown mandatory state is represented explicitly and does not default to ordinary allow.

The chapter’s principal structural commitments are:

State-Transition-Consequence Identity - retained as an explicit state or dependency.

Pre-Execution Admissibility - retained as an explicit state or dependency.

Authority Binding - retained as an explicit state or dependency.

Evidence Maturity - retained as an explicit state or dependency.

Fault-Relative Scope - retained as an explicit state or dependency.

Public-Safe Disclosure - retained as an explicit state or dependency.

4.3 Execution Reality

Execution reality begins at the last point from which the protected consequence can still be prevented. Upstream analysis, documentation, monitoring, or recommendation may improve decision quality, but it becomes governance only when the consequence path depends on the resulting state. The chapter therefore treats direct capability, alternate interfaces, administrative paths, recovery mechanisms, stale state, and external dependencies as part of the governed object. A system is not credited with a property merely because a dashboard displays it; the relevant gate, controller, institution, or physical interface must enforce or evidence it.

4.4 Failure Point and Counter-Construction

The claim collapses under any in-scope counter-construction such as:

policy is mistaken for enforcement

a model output is treated as authority

scope and assumptions are omitted

evidence maturity is overstated

These failures are not edge cases external to the theory. They are the primary falsification instruments. Where one of them remains reachable, the correct result is a narrowed claim, an unresolved state, or a failed invariant - not a narrative assertion that the architecture is nevertheless governed.

4.5 Verification and Evidence Obligations

define the claim envelope

identify authority and consequence boundaries

separate public, protected, and institutionally imported assertions

assign an evidence maturity state

Evidence must remain bound to the exact artifact, configuration, version, operating envelope, and decision state examined. A passing result establishes only the tested property and coverage. A failed or indeterminate result remains part of the authoritative record. Where independent reproduction has not occurred, the result is not represented as independently established.

4.6 Selected Propositions

P4.1. The validity of admissibility as a pre-execution relation cannot exceed the scope of its weakest mandatory dependency.

P4.2. A descriptive or advisory representation does not become an execution constraint until it controls the relevant consequence path.

P4.3. Component-level validity does not compose automatically across interfaces, timing, authority, failure, or deployment change.

P4.4. Evidence of absence is not inferred from absence of evidence; unresolved mandatory state remains unresolved.

4.7 Governance Status and Confidence Type

STRUCTURE. Specified as part of the integrated federated architecture.
EXECUTION REALITY. Requires a domain-specific implementation and complete path inventory.
FAILURE POINT. Any surviving bypass, stale authority, omitted dependency, unmodeled timing path, or unsupported evidence link narrows or defeats the claim.
GOVERNANCE STATUS. M1 unless a chapter-specific case study states a higher internal implementation maturity; no institutional compliance determination is inferred.
CONFIDENCE TYPE. Definitional for the architecture, deductive only under stated assumptions, implementation-dependent for runtime and hardware, institutionally imported for authority, and evidence-dependent for deployment claims.

Chapter 5

Authority as Execution Legitimacy

Authority as a time-bounded, scope-bound, authenticated relation that the architecture consumes but does not invent.

5.1 Objective

The objective of this chapter is to define authority as execution legitimacy as a bounded component of execution-bound governance. The analysis connects state-transition-consequence identity, pre-execution admissibility, authority binding, evidence maturity, fault-relative scope, and public-safe disclosure. The decisive test is not whether the system can describe the relevant rule, but whether the rule changes the set of reachable consequence-bearing transitions. This chapter operates at the common-core level and therefore avoids discipline-specific claims that would collapse mathematics, engineering, human authority, and law into one vocabulary.

5.2 Structural Model

A compact representation used in this chapter is:

$$AuthorityValid = Identity \wedge Mandate \wedge Scope \wedge Time \wedge Unrevoked$$

The structural object is represented as a typed relation rather than an informal checklist. Identity, object, context, authority, version, time, fault state, and evidence are separated because a change in any one of them can create a different transition. The architecture therefore rejects implicit substitution: a permission for one object is not a permission for another; a proof for one model is not a proof for a deployment; an approval before mutation is not approval after mutation. Unknown mandatory state is represented explicitly and does not default to ordinary allow.

The chapter’s principal structural commitments are:

State-Transition-Consequence Identity - retained as an explicit state or dependency.

Pre-Execution Admissibility - retained as an explicit state or dependency.

Authority Binding - retained as an explicit state or dependency.

Evidence Maturity - retained as an explicit state or dependency.

Fault-Relative Scope - retained as an explicit state or dependency.

Public-Safe Disclosure - retained as an explicit state or dependency.

5.3 Execution Reality

Execution reality begins at the last point from which the protected consequence can still be prevented. Upstream analysis, documentation, monitoring, or recommendation may improve decision quality, but it becomes governance only when the consequence path depends on the resulting state. The chapter therefore treats direct capability, alternate interfaces, administrative paths, recovery mechanisms, stale state, and external dependencies as part of the governed object. A system is not credited with a property merely because a dashboard displays it; the relevant gate, controller, institution, or physical interface must enforce or evidence it.

5.4 Failure Point and Counter-Construction

The claim collapses under any in-scope counter-construction such as:

policy is mistaken for enforcement

a model output is treated as authority

scope and assumptions are omitted

evidence maturity is overstated

These failures are not edge cases external to the theory. They are the primary falsification instruments. Where one of them remains reachable, the correct result is a narrowed claim, an unresolved state, or a failed invariant - not a narrative assertion that the architecture is nevertheless governed.

5.5 Verification and Evidence Obligations

define the claim envelope

identify authority and consequence boundaries

separate public, protected, and institutionally imported assertions

assign an evidence maturity state

Evidence must remain bound to the exact artifact, configuration, version, operating envelope, and decision state examined. A passing result establishes only the tested property and coverage. A failed or indeterminate result remains part of the authoritative record. Where independent reproduction has not occurred, the result is not represented as independently established.

5.6 Selected Propositions

P5.1. The validity of authority as execution legitimacy cannot exceed the scope of its weakest mandatory dependency.

P5.2. A descriptive or advisory representation does not become an execution constraint until it controls the relevant consequence path.

P5.3. Component-level validity does not compose automatically across interfaces, timing, authority, failure, or deployment change.

P5.4. Evidence of absence is not inferred from absence of evidence; unresolved mandatory state remains unresolved.

5.7 Governance Status and Confidence Type

STRUCTURE. Specified as part of the integrated federated architecture.
EXECUTION REALITY. Requires a domain-specific implementation and complete path inventory.
FAILURE POINT. Any surviving bypass, stale authority, omitted dependency, unmodeled timing path, or unsupported evidence link narrows or defeats the claim.
GOVERNANCE STATUS. M1 unless a chapter-specific case study states a higher internal implementation maturity; no institutional compliance determination is inferred.
CONFIDENCE TYPE. Definitional for the architecture, deductive only under stated assumptions, implementation-dependent for runtime and hardware, institutionally imported for authority, and evidence-dependent for deployment claims.

Chapter 6

The Execution Boundary

The point at which a proposal becomes committed consequence and therefore must become non-bypassably permit-dependent.

6.1 Objective

The objective of this chapter is to define the execution boundary as a bounded component of execution-bound governance. The analysis connects state-transition-consequence identity, pre-execution admissibility, authority binding, evidence maturity, fault-relative scope, and public-safe disclosure. The decisive test is not whether the system can describe the relevant rule, but whether the rule changes the set of reachable consequence-bearing transitions. This chapter operates at the common-core level and therefore avoids discipline-specific claims that would collapse mathematics, engineering, human authority, and law into one vocabulary.

6.2 Structural Model

The structural object is represented as a typed relation rather than an informal checklist. Identity, object, context, authority, version, time, fault state, and evidence are separated because a change in any one of them can create a different transition. The architecture therefore rejects implicit substitution: a permission for one object is not a permission for another; a proof for one model is not a proof for a deployment; an approval before mutation is not approval after mutation. Unknown mandatory state is represented explicitly and does not default to ordinary allow.

The chapter’s principal structural commitments are:

State-Transition-Consequence Identity - retained as an explicit state or dependency.

Pre-Execution Admissibility - retained as an explicit state or dependency.

Authority Binding - retained as an explicit state or dependency.

Evidence Maturity - retained as an explicit state or dependency.

Fault-Relative Scope - retained as an explicit state or dependency.

Public-Safe Disclosure - retained as an explicit state or dependency.

6.3 Execution Reality

Execution reality begins at the last point from which the protected consequence can still be prevented. Upstream analysis, documentation, monitoring, or recommendation may improve decision quality, but it becomes governance only when the consequence path depends on the resulting state. The chapter therefore treats direct capability, alternate interfaces, administrative paths, recovery mechanisms, stale state, and external dependencies as part of the governed object. A system is not credited with a property merely because a dashboard displays it; the relevant gate, controller, institution, or physical interface must enforce or evidence it.

6.4 Failure Point and Counter-Construction

The claim collapses under any in-scope counter-construction such as:

policy is mistaken for enforcement

a model output is treated as authority

scope and assumptions are omitted

evidence maturity is overstated

These failures are not edge cases external to the theory. They are the primary falsification instruments. Where one of them remains reachable, the correct result is a narrowed claim, an unresolved state, or a failed invariant - not a narrative assertion that the architecture is nevertheless governed.

6.5 Verification and Evidence Obligations

define the claim envelope

identify authority and consequence boundaries

separate public, protected, and institutionally imported assertions

assign an evidence maturity state

Evidence must remain bound to the exact artifact, configuration, version, operating envelope, and decision state examined. A passing result establishes only the tested property and coverage. A failed or indeterminate result remains part of the authoritative record. Where independent reproduction has not occurred, the result is not represented as independently established.

6.6 Selected Propositions

P6.1. The validity of the execution boundary cannot exceed the scope of its weakest mandatory dependency.

P6.2. A descriptive or advisory representation does not become an execution constraint until it controls the relevant consequence path.

P6.3. Component-level validity does not compose automatically across interfaces, timing, authority, failure, or deployment change.

P6.4. Evidence of absence is not inferred from absence of evidence; unresolved mandatory state remains unresolved.

6.7 Governance Status and Confidence Type

STRUCTURE. Specified as part of the integrated federated architecture.
EXECUTION REALITY. Requires a domain-specific implementation and complete path inventory.
FAILURE POINT. Any surviving bypass, stale authority, omitted dependency, unmodeled timing path, or unsupported evidence link narrows or defeats the claim.
GOVERNANCE STATUS. M1 unless a chapter-specific case study states a higher internal implementation maturity; no institutional compliance determination is inferred.
CONFIDENCE TYPE. Definitional for the architecture, deductive only under stated assumptions, implementation-dependent for runtime and hardware, institutionally imported for authority, and evidence-dependent for deployment claims.

Chapter 7

State Continuity and Evidence Integrity

Preserving identity, authority, version, decision, consequence, and evidence across time, failure, handoff, and update.

7.1 Objective

The objective of this chapter is to define state continuity and evidence integrity as a bounded component of execution-bound governance. The analysis connects state-transition-consequence identity, pre-execution admissibility, authority binding, evidence maturity, fault-relative scope, and public-safe disclosure. The decisive test is not whether the system can describe the relevant rule, but whether the rule changes the set of reachable consequence-bearing transitions. This chapter operates at the common-core level and therefore avoids discipline-specific claims that would collapse mathematics, engineering, human authority, and law into one vocabulary.

7.2 Structural Model

The structural object is represented as a typed relation rather than an informal checklist. Identity, object, context, authority, version, time, fault state, and evidence are separated because a change in any one of them can create a different transition. The architecture therefore rejects implicit substitution: a permission for one object is not a permission for another; a proof for one model is not a proof for a deployment; an approval before mutation is not approval after mutation. Unknown mandatory state is represented explicitly and does not default to ordinary allow.

The chapter’s principal structural commitments are:

State-Transition-Consequence Identity - retained as an explicit state or dependency.

Pre-Execution Admissibility - retained as an explicit state or dependency.

Authority Binding - retained as an explicit state or dependency.

Evidence Maturity - retained as an explicit state or dependency.

Fault-Relative Scope - retained as an explicit state or dependency.

Public-Safe Disclosure - retained as an explicit state or dependency.

7.3 Execution Reality

Execution reality begins at the last point from which the protected consequence can still be prevented. Upstream analysis, documentation, monitoring, or recommendation may improve decision quality, but it becomes governance only when the consequence path depends on the resulting state. The chapter therefore treats direct capability, alternate interfaces, administrative paths, recovery mechanisms, stale state, and external dependencies as part of the governed object. A system is not credited with a property merely because a dashboard displays it; the relevant gate, controller, institution, or physical interface must enforce or evidence it.

7.4 Failure Point and Counter-Construction

The claim collapses under any in-scope counter-construction such as:

policy is mistaken for enforcement

a model output is treated as authority

scope and assumptions are omitted

evidence maturity is overstated

These failures are not edge cases external to the theory. They are the primary falsification instruments. Where one of them remains reachable, the correct result is a narrowed claim, an unresolved state, or a failed invariant - not a narrative assertion that the architecture is nevertheless governed.

7.5 Verification and Evidence Obligations

define the claim envelope

identify authority and consequence boundaries

separate public, protected, and institutionally imported assertions

assign an evidence maturity state

Evidence must remain bound to the exact artifact, configuration, version, operating envelope, and decision state examined. A passing result establishes only the tested property and coverage. A failed or indeterminate result remains part of the authoritative record. Where independent reproduction has not occurred, the result is not represented as independently established.

7.6 Selected Propositions

P7.1. The validity of state continuity and evidence integrity cannot exceed the scope of its weakest mandatory dependency.

P7.2. A descriptive or advisory representation does not become an execution constraint until it controls the relevant consequence path.

P7.3. Component-level validity does not compose automatically across interfaces, timing, authority, failure, or deployment change.

P7.4. Evidence of absence is not inferred from absence of evidence; unresolved mandatory state remains unresolved.

7.7 Governance Status and Confidence Type

STRUCTURE. Specified as part of the integrated federated architecture.
EXECUTION REALITY. Requires a domain-specific implementation and complete path inventory.
FAILURE POINT. Any surviving bypass, stale authority, omitted dependency, unmodeled timing path, or unsupported evidence link narrows or defeats the claim.
GOVERNANCE STATUS. M1 unless a chapter-specific case study states a higher internal implementation maturity; no institutional compliance determination is inferred.
CONFIDENCE TYPE. Definitional for the architecture, deductive only under stated assumptions, implementation-dependent for runtime and hardware, institutionally imported for authority, and evidence-dependent for deployment claims.

Chapter 8

Structural Load, Drift, and Governance Continuity

Why load, latency, context growth, and institutional change expose governance mechanisms that are only conditionally effective.

8.1 Objective

The objective of this chapter is to define structural load, drift, and governance continuity as a bounded component of execution-bound governance. The analysis connects state-transition-consequence identity, pre-execution admissibility, authority binding, evidence maturity, fault-relative scope, and public-safe disclosure. The decisive test is not whether the system can describe the relevant rule, but whether the rule changes the set of reachable consequence-bearing transitions. This chapter operates at the common-core level and therefore avoids discipline-specific claims that would collapse mathematics, engineering, human authority, and law into one vocabulary.

8.2 Structural Model

A compact representation used in this chapter is:

$$L(t) = L_0 + \int_0^t \phi(\tau)\, d\tau - S(t)$$

The structural object is represented as a typed relation rather than an informal checklist. Identity, object, context, authority, version, time, fault state, and evidence are separated because a change in any one of them can create a different transition. The architecture therefore rejects implicit substitution: a permission for one object is not a permission for another; a proof for one model is not a proof for a deployment; an approval before mutation is not approval after mutation. Unknown mandatory state is represented explicitly and does not default to ordinary allow.

The chapter’s principal structural commitments are:

State-Transition-Consequence Identity - retained as an explicit state or dependency.

Pre-Execution Admissibility - retained as an explicit state or dependency.

Authority Binding - retained as an explicit state or dependency.

Evidence Maturity - retained as an explicit state or dependency.

Fault-Relative Scope - retained as an explicit state or dependency.

Public-Safe Disclosure - retained as an explicit state or dependency.

8.3 Execution Reality

Execution reality begins at the last point from which the protected consequence can still be prevented. Upstream analysis, documentation, monitoring, or recommendation may improve decision quality, but it becomes governance only when the consequence path depends on the resulting state. The chapter therefore treats direct capability, alternate interfaces, administrative paths, recovery mechanisms, stale state, and external dependencies as part of the governed object. A system is not credited with a property merely because a dashboard displays it; the relevant gate, controller, institution, or physical interface must enforce or evidence it.

8.4 Failure Point and Counter-Construction

The claim collapses under any in-scope counter-construction such as:

policy is mistaken for enforcement

a model output is treated as authority

scope and assumptions are omitted

evidence maturity is overstated

These failures are not edge cases external to the theory. They are the primary falsification instruments. Where one of them remains reachable, the correct result is a narrowed claim, an unresolved state, or a failed invariant - not a narrative assertion that the architecture is nevertheless governed.

8.5 Verification and Evidence Obligations

define the claim envelope

identify authority and consequence boundaries

separate public, protected, and institutionally imported assertions

assign an evidence maturity state

Evidence must remain bound to the exact artifact, configuration, version, operating envelope, and decision state examined. A passing result establishes only the tested property and coverage. A failed or indeterminate result remains part of the authoritative record. Where independent reproduction has not occurred, the result is not represented as independently established.

8.6 Selected Propositions

P8.1. The validity of structural load, drift, and governance continuity cannot exceed the scope of its weakest mandatory dependency.

P8.2. A descriptive or advisory representation does not become an execution constraint until it controls the relevant consequence path.

P8.3. Component-level validity does not compose automatically across interfaces, timing, authority, failure, or deployment change.

P8.4. Evidence of absence is not inferred from absence of evidence; unresolved mandatory state remains unresolved.

8.7 Governance Status and Confidence Type

STRUCTURE. Specified as part of the integrated federated architecture.
EXECUTION REALITY. Requires a domain-specific implementation and complete path inventory.
FAILURE POINT. Any surviving bypass, stale authority, omitted dependency, unmodeled timing path, or unsupported evidence link narrows or defeats the claim.
GOVERNANCE STATUS. M1 unless a chapter-specific case study states a higher internal implementation maturity; no institutional compliance determination is inferred.
CONFIDENCE TYPE. Definitional for the architecture, deductive only under stated assumptions, implementation-dependent for runtime and hardware, institutionally imported for authority, and evidence-dependent for deployment claims.

Chapter 9

A Candidate Class of Structurally Governable Systems

Definition and limits of Structurally Governable Dynamical Systems (SGDS).

9.1 Objective

The objective of this chapter is to define a candidate class of structurally governable systems as a bounded component of execution-bound governance. The analysis connects state-transition-consequence identity, pre-execution admissibility, authority binding, evidence maturity, fault-relative scope, and public-safe disclosure. The decisive test is not whether the system can describe the relevant rule, but whether the rule changes the set of reachable consequence-bearing transitions. This chapter operates at the common-core level and therefore avoids discipline-specific claims that would collapse mathematics, engineering, human authority, and law into one vocabulary.

9.2 Structural Model

A compact representation used in this chapter is:

$$S_{SGDS}=\langle X, U, F, A, B, H, E, \Omega \rangle$$

The structural object is represented as a typed relation rather than an informal checklist. Identity, object, context, authority, version, time, fault state, and evidence are separated because a change in any one of them can create a different transition. The architecture therefore rejects implicit substitution: a permission for one object is not a permission for another; a proof for one model is not a proof for a deployment; an approval before mutation is not approval after mutation. Unknown mandatory state is represented explicitly and does not default to ordinary allow.

The chapter's principal structural commitments are:

State-Transition-Consequence Identity - retained as an explicit state or dependency.

Pre-Execution Admissibility - retained as an explicit state or dependency.

Authority Binding - retained as an explicit state or dependency.

Evidence Maturity - retained as an explicit state or dependency.

Fault-Relative Scope - retained as an explicit state or dependency.

Public-Safe Disclosure - retained as an explicit state or dependency.

9.3 Execution Reality

Execution reality begins at the last point from which the protected consequence can still be prevented. Upstream analysis, documentation, monitoring, or recommendation may improve decision quality, but it becomes governance only when the consequence path depends on the resulting state. The chapter therefore treats direct capability, alternate interfaces, administrative paths, recovery mechanisms, stale state, and external dependencies as part of the governed object. A system is not credited with a property merely because a dashboard displays it; the relevant gate, controller, institution, or physical interface must enforce or evidence it.

9.4 Failure Point and Counter-Construction

The claim collapses under any in-scope counter-construction such as:

policy is mistaken for enforcement

a model output is treated as authority

scope and assumptions are omitted

evidence maturity is overstated

These failures are not edge cases external to the theory. They are the primary falsification instruments. Where one of them remains reachable, the correct result is a narrowed claim, an unresolved state, or a failed invariant - not a narrative assertion that the architecture is nevertheless governed.

9.5 Verification and Evidence Obligations

define the claim envelope

identify authority and consequence boundaries

separate public, protected, and institutionally imported assertions

assign an evidence maturity state

Evidence must remain bound to the exact artifact, configuration, version, operating envelope, and decision state examined. A passing result establishes only the tested property and coverage. A failed or indeterminate result remains part of the authoritative record. Where independent reproduction has not occurred, the result is not represented as independently established.

9.6 Selected Propositions

P9.1. The validity of a candidate class of structurally governable systems cannot exceed the scope of its weakest mandatory dependency.

P9.2. A descriptive or advisory representation does not become an execution constraint until it controls the relevant consequence path.

P9.3. Component-level validity does not compose automatically across interfaces, timing, authority, failure, or deployment change.

P9.4. Evidence of absence is not inferred from absence of evidence; unresolved mandatory state remains unresolved.

9.7 Governance Status and Confidence Type

STRUCTURE. Specified as part of the integrated federated architecture.
EXECUTION REALITY. Requires a domain-specific implementation and complete path inventory.
FAILURE POINT. Any surviving bypass, stale authority, omitted dependency, unmodeled timing path, or unsupported evidence link narrows or defeats the claim.
GOVERNANCE STATUS. M1 unless a chapter-specific case study states a higher internal implementation maturity; no institutional compliance determination is inferred.
CONFIDENCE TYPE. Definitional for the architecture, deductive only under stated assumptions, implementation-dependent for runtime and hardware, institutionally imported for authority, and evidence-dependent for deployment claims.

Chapter 10

Claim Architecture and Verification Logic

Claim tuples, scope, assumptions, evidence maturity, dependency graphs, and public-claim gates.

10.1 Objective

The objective of this chapter is to define claim architecture and verification logic as a bounded component of execution-bound governance. The analysis connects state-transition-consequence identity, pre-execution admissibility, authority binding, evidence maturity, fault-relative scope, and public-safe disclosure. The decisive test is not whether the system can describe the relevant rule, but whether the rule changes the set of reachable consequence-bearing transitions. This chapter operates at the common-core level and therefore avoids discipline-specific claims that would collapse mathematics, engineering, human authority, and law into one vocabulary.

10.2 Structural Model

A compact representation used in this chapter is:

$$C=\left(q,\tau_C,S,A,M,E,F,L,D\right)$$

The structural object is represented as a typed relation rather than an informal checklist. Identity, object, context, authority, version, time, fault state, and evidence are separated because a change in any one of them can create a different transition. The architecture therefore rejects implicit substitution: a permission for one object is not a permission for another; a proof for one model is not a proof for a deployment; an approval before mutation is not approval after mutation. Unknown mandatory state is represented explicitly and does not default to ordinary allow.

The chapter's principal structural commitments are:

State-Transition-Consequence Identity - retained as an explicit state or dependency.

Pre-Execution Admissibility - retained as an explicit state or dependency.

Authority Binding - retained as an explicit state or dependency.

Evidence Maturity - retained as an explicit state or dependency.

Fault-Relative Scope - retained as an explicit state or dependency.

Public-Safe Disclosure - retained as an explicit state or dependency.

10.3 Execution Reality

Execution reality begins at the last point from which the protected consequence can still be prevented. Upstream analysis, documentation, monitoring, or recommendation may improve decision quality, but it becomes governance only when the consequence path depends on the resulting state. The chapter therefore treats direct capability, alternate interfaces, administrative paths, recovery mechanisms, stale state, and external dependencies as part of the governed object. A system is not credited with a property merely because a dashboard displays it; the relevant gate, controller, institution, or physical interface must enforce or evidence it.

10.4 Failure Point and Counter-Construction

The claim collapses under any in-scope counter-construction such as:

policy is mistaken for enforcement

a model output is treated as authority

scope and assumptions are omitted

evidence maturity is overstated

These failures are not edge cases external to the theory. They are the primary falsification instruments. Where one of them remains reachable, the correct result is a narrowed claim, an unresolved state, or a failed invariant - not a narrative assertion that the architecture is nevertheless governed.

10.5 Verification and Evidence Obligations

define the claim envelope

identify authority and consequence boundaries

separate public, protected, and institutionally imported assertions

assign an evidence maturity state

Evidence must remain bound to the exact artifact, configuration, version, operating envelope, and decision state examined. A passing result establishes only the tested property and coverage. A failed or indeterminate result remains part of the authoritative record. Where independent reproduction has not occurred, the result is not represented as independently established.

10.6 Selected Propositions

P10.1. The validity of claim architecture and verification logic cannot exceed the scope of its weakest mandatory dependency.

P10.2. A descriptive or advisory representation does not become an execution constraint until it controls the relevant consequence path.

P10.3. Component-level validity does not compose automatically across interfaces, timing, authority, failure, or deployment change.

P10.4. Evidence of absence is not inferred from absence of evidence; unresolved mandatory state remains unresolved.

10.7 Governance Status and Confidence Type

STRUCTURE. Specified as part of the integrated federated architecture.
EXECUTION REALITY. Requires a domain-specific implementation and complete path inventory.
FAILURE POINT. Any surviving bypass, stale authority, omitted dependency, unmodeled timing path, or unsupported evidence link narrows or defeats the claim.
GOVERNANCE STATUS. M1 unless a chapter-specific case study states a higher internal implementation maturity; no institutional compliance determination is inferred.
CONFIDENCE TYPE. Definitional for the architecture, deductive only under stated assumptions, implementation-dependent for runtime and hardware, institutionally imported for authority, and evidence-dependent for deployment claims.

Chapter 11

Validation and Falsification Program

Validation objects, result states, coverage ceilings, counter-constructions, and evidence packages.

11.1 Objective

The objective of this chapter is to define validation and falsification program as a bounded component of execution-bound governance. The analysis connects state-transition-consequence identity, pre-execution admissibility, authority binding, evidence maturity, fault-relative scope, and public-safe disclosure. The decisive test is not whether the system can describe the relevant rule, but whether the rule changes the set of reachable consequence-bearing transitions. This chapter operates at the common-core level and therefore avoids discipline-specific claims that would collapse mathematics, engineering, human authority, and law into one vocabulary.

11.2 Structural Model

The structural object is represented as a typed relation rather than an informal checklist. Identity, object, context, authority, version, time, fault state, and evidence are separated because a change in any one of them can create a different transition. The architecture therefore rejects implicit substitution: a permission for one object is not a permission for another; a proof for one model is not a proof for a deployment; an approval before mutation is not approval after mutation. Unknown mandatory state is represented explicitly and does not default to ordinary allow.

The chapter's principal structural commitments are:

State-Transition-Consequence Identity - retained as an explicit state or dependency.

Pre-Execution Admissibility - retained as an explicit state or dependency.

Authority Binding - retained as an explicit state or dependency.

Evidence Maturity - retained as an explicit state or dependency.

Fault-Relative Scope - retained as an explicit state or dependency.

Public-Safe Disclosure - retained as an explicit state or dependency.

11.3 Execution Reality

Execution reality begins at the last point from which the protected consequence can still be prevented. Upstream analysis, documentation, monitoring, or recommendation may improve decision quality, but it becomes governance only when the consequence path depends on the resulting state. The chapter therefore treats direct capability, alternate interfaces, administrative paths, recovery mechanisms, stale state, and external dependencies as part of the governed object. A system is not credited with a property merely because a dashboard displays it; the relevant gate, controller, institution, or physical interface must enforce or evidence it.

11.4 Failure Point and Counter-Construction

The claim collapses under any in-scope counter-construction such as:

policy is mistaken for enforcement

a model output is treated as authority

scope and assumptions are omitted

evidence maturity is overstated

These failures are not edge cases external to the theory. They are the primary falsification instruments. Where one of them remains reachable, the correct result is a narrowed claim, an unresolved state, or a failed invariant - not a narrative assertion that the architecture is nevertheless governed.

11.5 Verification and Evidence Obligations

define the claim envelope

identify authority and consequence boundaries

separate public, protected, and institutionally imported assertions

assign an evidence maturity state

Evidence must remain bound to the exact artifact, configuration, version, operating envelope, and decision state examined. A passing result establishes only the tested property and coverage. A failed or indeterminate result remains part of the authoritative record. Where independent reproduction has not occurred, the result is not represented as independently established.

11.6 Selected Propositions

P11.1. The validity of validation and falsification program cannot exceed the scope of its weakest mandatory dependency.

P11.2. A descriptive or advisory representation does not become an execution constraint until it controls the relevant consequence path.

P11.3. Component-level validity does not compose automatically across interfaces, timing, authority, failure, or deployment change.

P11.4. Evidence of absence is not inferred from absence of evidence; unresolved mandatory state remains unresolved.

11.7 Governance Status and Confidence Type

STRUCTURE. Specified as part of the integrated federated architecture.
EXECUTION REALITY. Requires a domain-specific implementation and complete path inventory.
FAILURE POINT. Any surviving bypass, stale authority, omitted dependency, unmodeled timing path, or unsupported evidence link narrows or defeats the claim.
GOVERNANCE STATUS. M1 unless a chapter-specific case study states a higher internal implementation maturity; no institutional compliance determination is inferred.
CONFIDENCE TYPE. Definitional for the architecture, deductive only under stated assumptions, implementation-dependent for runtime and hardware, institutionally imported for authority, and evidence-dependent for deployment claims.

Chapter 12

Common-Core Synthesis and Faculty Interface Contract

The shared vocabulary and non-collapse rules connecting mathematics, control, computer science, hardware, authority, governance, and applied systems.

12.1 Objective

The objective of this chapter is to define common-core synthesis and faculty interface contract as a bounded component of execution-bound governance. The analysis connects state-transition-consequence identity, pre-execution admissibility, authority binding, evidence maturity, fault-relative scope, and public-safe disclosure. The decisive test is not whether the system can describe the relevant rule, but whether the rule changes the set of reachable consequence-bearing transitions. This chapter operates at the common-core level and therefore avoids discipline-specific claims that would collapse mathematics, engineering, human authority, and law into one vocabulary.

12.2 Structural Model

A compact representation used in this chapter is:

$$\neg\, Permit(\tau) \Rightarrow \neg\, Exec_{B,F}(\tau)$$

The structural object is represented as a typed relation rather than an informal checklist. Identity, object, context, authority, version, time, fault state, and evidence are separated because a change in any one of them can create a different transition. The architecture therefore rejects implicit substitution: a permission for one object is not a permission for another; a proof for one model is not a proof for a deployment; an approval before mutation is not approval after mutation. Unknown mandatory state is represented explicitly and does not default to ordinary allow.

The chapter's principal structural commitments are:

State-Transition-Consequence Identity - retained as an explicit state or dependency.

Pre-Execution Admissibility - retained as an explicit state or dependency.

Authority Binding - retained as an explicit state or dependency.

Evidence Maturity - retained as an explicit state or dependency.

Fault-Relative Scope - retained as an explicit state or dependency.

Public-Safe Disclosure - retained as an explicit state or dependency.

12.3 Execution Reality

Execution reality begins at the last point from which the protected consequence can still be prevented. Upstream analysis, documentation, monitoring, or recommendation may improve decision quality, but it becomes governance only when the consequence path depends on the resulting state. The chapter therefore treats direct capability, alternate interfaces, administrative paths, recovery mechanisms, stale state, and external dependencies as part of the governed object. A system is not credited with a property merely because a dashboard displays it; the relevant gate, controller, institution, or physical interface must enforce or evidence it.

12.4 Failure Point and Counter-Construction

The claim collapses under any in-scope counter-construction such as:

policy is mistaken for enforcement

a model output is treated as authority

scope and assumptions are omitted

evidence maturity is overstated

These failures are not edge cases external to the theory. They are the primary falsification instruments. Where one of them remains reachable, the correct result is a narrowed claim, an unresolved state, or a failed invariant - not a narrative assertion that the architecture is nevertheless governed.

12.5 Verification and Evidence Obligations

define the claim envelope

identify authority and consequence boundaries

separate public, protected, and institutionally imported assertions

assign an evidence maturity state

Evidence must remain bound to the exact artifact, configuration, version, operating envelope, and decision state examined. A passing result establishes only the tested property and coverage. A failed or indeterminate result remains part of the authoritative record. Where independent reproduction has not occurred, the result is not represented as independently established.

12.6 Selected Propositions

P12.1. The validity of common-core synthesis and faculty interface contract cannot exceed the scope of its weakest mandatory dependency.

P12.2. A descriptive or advisory representation does not become an execution constraint until it controls the relevant consequence path.

P12.3. Component-level validity does not compose automatically across interfaces, timing, authority, failure, or deployment change.

P12.4. Evidence of absence is not inferred from absence of evidence; unresolved mandatory state remains unresolved.

12.7 Governance Status and Confidence Type

STRUCTURE. Specified as part of the integrated federated architecture.
EXECUTION REALITY. Requires a domain-specific implementation and complete path inventory.
FAILURE POINT. Any surviving bypass, stale authority, omitted dependency, unmodeled timing path, or unsupported evidence link narrows or defeats the claim.
GOVERNANCE STATUS. M1 unless a chapter-specific case study states a higher internal implementation maturity; no institutional compliance determination is inferred.
CONFIDENCE TYPE. Definitional for the architecture, deductive only under stated assumptions, implementation-dependent for runtime and hardware, institutionally imported for authority, and evidence-dependent for deployment claims.

PART II - FORMAL AND MATHEMATICAL FOUNDATIONS

Part II translates the architectural thesis into formal state, transition, uncertainty, composition, timing, hybrid, refinement, decidability, and mechanization objects. The result is a proof architecture, not a claim that every deployment has already discharged it.

Part-level claim boundary. The material in this part is specified at M1 unless a bounded case study explicitly records internal implementation evidence. No part-level statement independently widens the system-level claim.

Chapter 13

Formal State Spaces and Governed Transition Systems

A typed transition-system semantics for admissibility, authority, execution, evidence, and fault-relative governability.

13.1 Objective

The objective of this chapter is to define formal state spaces and governed transition systems as a bounded component of execution-bound governance. The analysis connects typed state space, invariant preservation, reachability, uncertainty set, composition, refinement, and mechanized evidence. The decisive test is not whether the system can describe the relevant rule, but whether the rule changes the set of reachable consequence-bearing transitions. The claims in this chapter are model-internal unless a later correspondence argument binds the formal object to source, binary, configuration, hardware, deployment, and evidence.

13.2 Structural Model

A compact representation used in this chapter is:

$$G=\left(Z,U,\rightarrow,O,A,R,P,E,V,Z_0,B,F\right)$$

The structural object is represented as a typed relation rather than an informal checklist. Identity, object, context, authority, version, time, fault state, and evidence are separated because a change in any one of them can create a different transition. The architecture therefore rejects implicit substitution: a permission for one object is not a permission for another; a proof for one model is not a proof for a deployment; an approval before mutation is not approval after mutation. Unknown mandatory state is represented explicitly and does not default to ordinary allow.

The chapter’s principal structural commitments are:

Typed State Space - retained as an explicit state or dependency.

Invariant Preservation - retained as an explicit state or dependency.

Reachability - retained as an explicit state or dependency.

Uncertainty Set - retained as an explicit state or dependency.

Composition - retained as an explicit state or dependency.

Refinement - retained as an explicit state or dependency.

Mechanized Evidence - retained as an explicit state or dependency.

13.3 Execution Reality

Execution reality begins at the last point from which the protected consequence can still be prevented. Upstream analysis, documentation, monitoring, or recommendation may improve decision quality, but it becomes governance only when the consequence path depends on the resulting state. The chapter therefore treats direct capability, alternate interfaces, administrative paths, recovery mechanisms, stale state, and external dependencies as part of the governed object. A system is not credited with a property merely because a dashboard displays it; the relevant gate, controller, institution, or physical interface must enforce or evidence it.

13.4 Failure Point and Counter-Construction

The claim collapses under any in-scope counter-construction such as:

the model omits a consequence path

proof assumptions are hidden

component properties are assumed to compose

solver timeout is reported as proof

These failures are not edge cases external to the theory. They are the primary falsification instruments. Where one of them remains reachable, the correct result is a narrowed claim, an unresolved state, or a failed invariant - not a narrative assertion that the architecture is nevertheless governed.

13.5 Verification and Evidence Obligations

formalize initial states and transition semantics

prove or test inductiveness

enumerate uncontrollable and bypass paths

bind proof artifacts to model and assumptions

Evidence must remain bound to the exact artifact, configuration, version, operating envelope, and decision state examined. A passing result establishes only the tested property and coverage. A failed or indeterminate result remains part of the authoritative record. Where independent reproduction has not occurred, the result is not represented as independently established.

13.6 Selected Propositions

P13.1. The validity of formal state spaces and governed transition systems cannot exceed the scope of its weakest mandatory dependency.

P13.2. A descriptive or advisory representation does not become an execution constraint until it controls the relevant consequence path.

P13.3. Component-level validity does not compose automatically across interfaces, timing, authority, failure, or deployment change.

P13.4. Evidence of absence is not inferred from absence of evidence; unresolved mandatory state remains unresolved.

13.7 Governance Status and Confidence Type

STRUCTURE. Specified as part of the integrated federated architecture.
EXECUTION REALITY. Requires a domain-specific implementation and complete path inventory.
FAILURE POINT. Any surviving bypass, stale authority, omitted dependency, unmodeled timing path, or unsupported evidence link narrows or defeats the claim.
GOVERNANCE STATUS. M1 unless a chapter-specific case study states a higher internal implementation maturity; no institutional compliance determination is inferred.
CONFIDENCE TYPE. Definitional for the architecture, deductive only under stated assumptions, implementation-dependent for runtime and hardware, institutionally imported for authority, and evidence-dependent for deployment claims.

Chapter 14

Invariants, Induction, and Non-Reachability Proofs

Inductive invariants, fixed points, ranking functions, abstraction, counterexamples, and proof-status discipline.

14.1 Objective

The objective of this chapter is to define invariants, induction, and non-reachability proofs as a bounded component of execution-bound governance. The analysis connects typed state space, invariant preservation, reachability, uncertainty set, composition, refinement, and mechanized evidence. The decisive test is not whether the system can describe the relevant rule, but whether the rule changes the set of reachable consequence-bearing transitions. The claims in this chapter are model-internal unless a later correspondence argument binds the formal object to source, binary, configuration, hardware, deployment, and evidence.

14.2 Structural Model

The structural object is represented as a typed relation rather than an informal checklist. Identity, object, context, authority, version, time, fault state, and evidence are separated because a change in any one of them can create a different transition. The architecture therefore rejects implicit substitution: a permission for one object is not a permission for another; a proof for one model is not a proof for a deployment; an approval before mutation is not approval after mutation. Unknown mandatory state is represented explicitly and does not default to ordinary allow.

The chapter’s principal structural commitments are:

Typed State Space - retained as an explicit state or dependency.

Invariant Preservation - retained as an explicit state or dependency.

Reachability - retained as an explicit state or dependency.

Uncertainty Set - retained as an explicit state or dependency.

Composition - retained as an explicit state or dependency.

Refinement - retained as an explicit state or dependency.

Mechanized Evidence - retained as an explicit state or dependency.

14.3 Execution Reality

Execution reality begins at the last point from which the protected consequence can still be prevented. Upstream analysis, documentation, monitoring, or recommendation may improve decision quality, but it becomes governance only when the consequence path depends on the resulting state. The chapter therefore treats direct capability, alternate interfaces, administrative paths, recovery mechanisms, stale state, and external dependencies as part of the governed object. A system is not credited with a property merely because a dashboard displays it; the relevant gate, controller, institution, or physical interface must enforce or evidence it.

14.4 Failure Point and Counter-Construction

The claim collapses under any in-scope counter-construction such as:

the model omits a consequence path

proof assumptions are hidden

component properties are assumed to compose

solver timeout is reported as proof

These failures are not edge cases external to the theory. They are the primary falsification instruments. Where one of them remains reachable, the correct result is a narrowed claim, an unresolved state, or a failed invariant - not a narrative assertion that the architecture is nevertheless governed.

14.5 Verification and Evidence Obligations

formalize initial states and transition semantics

prove or test inductiveness

enumerate uncontrollable and bypass paths

bind proof artifacts to model and assumptions

Evidence must remain bound to the exact artifact, configuration, version, operating envelope, and decision state examined. A passing result establishes only the tested property and coverage. A failed or indeterminate result remains part of the authoritative record. Where independent reproduction has not occurred, the result is not represented as independently established.

14.6 Selected Propositions

P14.1. The validity of invariants, induction, and non-reachability proofs cannot exceed the scope of its weakest mandatory dependency.

P14.2. A descriptive or advisory representation does not become an execution constraint until it controls the relevant consequence path.

P14.3. Component-level validity does not compose automatically across interfaces, timing, authority, failure, or deployment change.

P14.4. Evidence of absence is not inferred from absence of evidence; unresolved mandatory state remains unresolved.

14.7 Governance Status and Confidence Type

STRUCTURE. Specified as part of the integrated federated architecture.
EXECUTION REALITY. Requires a domain-specific implementation and complete path inventory.
FAILURE POINT. Any surviving bypass, stale authority, omitted dependency, unmodeled timing path, or unsupported evidence link narrows or defeats the claim.
GOVERNANCE STATUS. M1 unless a chapter-specific case study states a higher internal implementation maturity; no institutional compliance determination is inferred.
CONFIDENCE TYPE. Definitional for the architecture, deductive only under stated assumptions, implementation-dependent for runtime and hardware, institutionally imported for authority, and evidence-dependent for deployment claims.

Chapter 15

Partial Observation, Uncertainty, and Robust Admissibility

Belief sets, robust action intersections, out-of-distribution states, and conservative treatment of unknowns.

15.1 Objective

The objective of this chapter is to define partial observation, uncertainty, and robust admissibility as a bounded component of execution-bound governance. The analysis connects typed state space, invariant preservation, reachability, uncertainty set, composition, refinement, and mechanized evidence. The decisive test is not whether the system can describe the relevant rule, but whether the rule changes the set of reachable consequence-bearing transitions. The claims in this chapter are model-internal unless a later correspondence argument binds the formal object to source, binary, configuration, hardware, deployment, and evidence.

15.2 Structural Model

A compact representation used in this chapter is:

$$U_{rob}(B) = \bigcap_{z \in B} U_{adm}(z)$$

The structural object is represented as a typed relation rather than an informal checklist. Identity, object, context, authority, version, time, fault state, and evidence are separated because a change in any one of them can create a different transition. The architecture therefore rejects implicit substitution: a permission for one object is not a permission for another; a proof for one model is not a proof for a deployment; an approval before mutation is not approval after mutation. Unknown mandatory state is represented explicitly and does not default to ordinary allow.

The chapter’s principal structural commitments are:

Typed State Space - retained as an explicit state or dependency.

Invariant Preservation - retained as an explicit state or dependency.

Reachability - retained as an explicit state or dependency.

Uncertainty Set - retained as an explicit state or dependency.

Composition - retained as an explicit state or dependency.

Refinement - retained as an explicit state or dependency.

Mechanized Evidence - retained as an explicit state or dependency.

15.3 Execution Reality

Execution reality begins at the last point from which the protected consequence can still be prevented. Upstream analysis, documentation, monitoring, or recommendation may improve decision quality, but it becomes governance only when the consequence path depends on the resulting state. The chapter therefore treats direct capability, alternate interfaces, administrative paths, recovery mechanisms, stale state, and external dependencies as part of the governed object. A system is not credited with a property merely because a dashboard displays it; the relevant gate, controller, institution, or physical interface must enforce or evidence it.

15.4 Failure Point and Counter-Construction

The claim collapses under any in-scope counter-construction such as:

the model omits a consequence path

proof assumptions are hidden

component properties are assumed to compose

solver timeout is reported as proof

These failures are not edge cases external to the theory. They are the primary falsification instruments. Where one of them remains reachable, the correct result is a narrowed claim, an unresolved state, or a failed invariant - not a narrative assertion that the architecture is nevertheless governed.

15.5 Verification and Evidence Obligations

formalize initial states and transition semantics

prove or test inductiveness

enumerate uncontrollable and bypass paths

bind proof artifacts to model and assumptions

Evidence must remain bound to the exact artifact, configuration, version, operating envelope, and decision state examined. A passing result establishes only the tested property and coverage. A failed or indeterminate result remains part of the authoritative record. Where independent reproduction has not occurred, the result is not represented as independently established.

15.6 Selected Propositions

P15.1. The validity of partial observation, uncertainty, and robust admissibility cannot exceed the scope of its weakest mandatory dependency.

P15.2. A descriptive or advisory representation does not become an execution constraint until it controls the relevant consequence path.

P15.3. Component-level validity does not compose automatically across interfaces, timing, authority, failure, or deployment change.

P15.4. Evidence of absence is not inferred from absence of evidence; unresolved mandatory state remains unresolved.

15.7 Governance Status and Confidence Type

STRUCTURE. Specified as part of the integrated federated architecture.
EXECUTION REALITY. Requires a domain-specific implementation and complete path inventory.
FAILURE POINT. Any surviving bypass, stale authority, omitted dependency, unmodeled timing path, or unsupported evidence link narrows or defeats the claim.
GOVERNANCE STATUS. M1 unless a chapter-specific case study states a higher internal implementation maturity; no institutional compliance determination is inferred.
CONFIDENCE TYPE. Definitional for the architecture, deductive only under stated assumptions, implementation-dependent for runtime and hardware, institutionally imported for authority, and evidence-dependent for deployment claims.

Chapter 16

Compositional Governability and Assume-Guarantee Contracts

Component interfaces, contract closure, authority intersection, evidence non-amplification, and unsafe composition.

16.1 Objective

The objective of this chapter is to define compositional governability and assume-guarantee contracts as a bounded component of execution-bound governance. The analysis connects typed state space, invariant preservation, reachability, uncertainty set, composition, refinement, and mechanized evidence. The decisive test is not whether the system can describe the relevant rule, but whether the rule changes the set of reachable consequence-bearing transitions. The claims in this chapter are model-internal unless a later correspondence argument binds the formal object to source, binary, configuration, hardware, deployment, and evidence.

16.2 Structural Model

The structural object is represented as a typed relation rather than an informal checklist. Identity, object, context, authority, version, time, fault state, and evidence are separated because a change in any one of them can create a different transition. The architecture therefore rejects implicit substitution: a permission for one object is not a permission for another; a proof for one model is not a proof for a deployment; an approval before mutation is not approval after mutation. Unknown mandatory state is represented explicitly and does not default to ordinary allow.

The chapter’s principal structural commitments are:

Typed State Space - retained as an explicit state or dependency.

Invariant Preservation - retained as an explicit state or dependency.

Reachability - retained as an explicit state or dependency.

Uncertainty Set - retained as an explicit state or dependency.

Composition - retained as an explicit state or dependency.

Refinement - retained as an explicit state or dependency.

Mechanized Evidence - retained as an explicit state or dependency.

16.3 Execution Reality

Execution reality begins at the last point from which the protected consequence can still be prevented. Upstream analysis, documentation, monitoring, or recommendation may improve decision quality, but it becomes governance only when the consequence path depends on the resulting state. The chapter therefore treats direct capability, alternate interfaces, administrative paths, recovery mechanisms, stale state, and external dependencies as part of the governed object. A system is not credited with a property merely because a dashboard displays it; the relevant gate, controller, institution, or physical interface must enforce or evidence it.

16.4 Failure Point and Counter-Construction

The claim collapses under any in-scope counter-construction such as:

the model omits a consequence path

proof assumptions are hidden

component properties are assumed to compose

solver timeout is reported as proof

These failures are not edge cases external to the theory. They are the primary falsification instruments. Where one of them remains reachable, the correct result is a narrowed claim, an unresolved state, or a failed invariant - not a narrative assertion that the architecture is nevertheless governed.

16.5 Verification and Evidence Obligations

formalize initial states and transition semantics

prove or test inductiveness

enumerate uncontrollable and bypass paths

bind proof artifacts to model and assumptions

Evidence must remain bound to the exact artifact, configuration, version, operating envelope, and decision state examined. A passing result establishes only the tested property and coverage. A failed or indeterminate result remains part of the authoritative record. Where independent reproduction has not occurred, the result is not represented as independently established.

16.6 Selected Propositions

P16.1. The validity of compositional governability and assume-guarantee contracts cannot exceed the scope of its weakest mandatory dependency.

P16.2. A descriptive or advisory representation does not become an execution constraint until it controls the relevant consequence path.

P16.3. Component-level validity does not compose automatically across interfaces, timing, authority, failure, or deployment change.

P16.4. Evidence of absence is not inferred from absence of evidence; unresolved mandatory state remains unresolved.

16.7 Governance Status and Confidence Type

STRUCTURE. Specified as part of the integrated federated architecture.
EXECUTION REALITY. Requires a domain-specific implementation and complete path inventory.
FAILURE POINT. Any surviving bypass, stale authority, omitted dependency, unmodeled timing path, or unsupported evidence link narrows or defeats the claim.
GOVERNANCE STATUS. M1 unless a chapter-specific case study states a higher internal implementation maturity; no institutional compliance determination is inferred.
CONFIDENCE TYPE. Definitional for the architecture, deductive only under stated assumptions, implementation-dependent for runtime and hardware, institutionally imported for authority, and evidence-dependent for deployment claims.

Chapter 17

Supervisory Control, Strategic Environments, and Governance Games

Controllable and uncontrollable events, winning regions, strategy non-self-authorization, and adversarial environments.

17.1 Objective

The objective of this chapter is to define supervisory control, strategic environments, and governance games as a bounded component of execution-bound governance. The analysis connects typed state space, invariant preservation, reachability, uncertainty set, composition, refinement, and mechanized evidence. The decisive test is not whether the system can describe the relevant rule, but whether the rule changes the set of reachable consequence-bearing transitions. The claims in this chapter are model-internal unless a later correspondence argument binds the formal object to source, binary, configuration, hardware, deployment, and evidence.

17.2 Structural Model

A compact representation used in this chapter is:

$$W^{\dot{c}} = \nu X . CPre_G(X)$$

The structural object is represented as a typed relation rather than an informal checklist. Identity, object, context, authority, version, time, fault state, and evidence are separated because a change in any one of them can create a different transition. The architecture therefore rejects implicit substitution: a permission for one object is not a permission for another; a proof for one model is not a proof for a deployment; an approval before mutation is not approval after mutation. Unknown mandatory state is represented explicitly and does not default to ordinary allow.

The chapter’s principal structural commitments are:

Typed State Space - retained as an explicit state or dependency.

Invariant Preservation - retained as an explicit state or dependency.

Reachability - retained as an explicit state or dependency.

Uncertainty Set - retained as an explicit state or dependency.

Composition - retained as an explicit state or dependency.

Refinement - retained as an explicit state or dependency.

Mechanized Evidence - retained as an explicit state or dependency.

17.3 Execution Reality

Execution reality begins at the last point from which the protected consequence can still be prevented. Upstream analysis, documentation, monitoring, or recommendation may improve decision quality, but it becomes governance only when the consequence path depends on the resulting state. The chapter therefore treats direct capability, alternate interfaces, administrative paths, recovery mechanisms, stale state, and external dependencies as part of the governed object. A system is not credited with a property merely because a dashboard displays it; the relevant gate, controller, institution, or physical interface must enforce or evidence it.

17.4 Failure Point and Counter-Construction

The claim collapses under any in-scope counter-construction such as:

the model omits a consequence path

proof assumptions are hidden

component properties are assumed to compose

solver timeout is reported as proof

These failures are not edge cases external to the theory. They are the primary falsification instruments. Where one of them remains reachable, the correct result is a narrowed claim, an unresolved state, or a failed invariant - not a narrative assertion that the architecture is nevertheless governed.

17.5 Verification and Evidence Obligations

formalize initial states and transition semantics

prove or test inductiveness

enumerate uncontrollable and bypass paths

bind proof artifacts to model and assumptions

Evidence must remain bound to the exact artifact, configuration, version, operating envelope, and decision state examined. A passing result establishes only the tested property and coverage. A failed or indeterminate result remains part of the authoritative record. Where independent reproduction has not occurred, the result is not represented as independently established.

17.6 Selected Propositions

P17.1. The validity of supervisory control, strategic environments, and governance games cannot exceed the scope of its weakest mandatory dependency.

P17.2. A descriptive or advisory representation does not become an execution constraint until it controls the relevant consequence path.

P17.3. Component-level validity does not compose automatically across interfaces, timing, authority, failure, or deployment change.

P17.4. Evidence of absence is not inferred from absence of evidence; unresolved mandatory state remains unresolved.

17.7 Governance Status and Confidence Type

STRUCTURE. Specified as part of the integrated federated architecture.
EXECUTION REALITY. Requires a domain-specific implementation and complete path inventory.
FAILURE POINT. Any surviving bypass, stale authority, omitted dependency, unmodeled timing path, or unsupported evidence link narrows or defeats the claim.
GOVERNANCE STATUS. M1 unless a chapter-specific case study states a higher internal implementation maturity; no institutional compliance determination is inferred.
CONFIDENCE TYPE. Definitional for the architecture, deductive only under stated assumptions, implementation-dependent for runtime and hardware, institutionally imported for authority, and evidence-dependent for deployment claims.

Chapter 18

Temporal Logic, Timed Automata, and Deadline-Bounded Governance

Freshness, revocation, deadlines, irreversibility, silence non-authorization, and bounded prevention.

18.1 Objective

The objective of this chapter is to define temporal logic, timed automata, and deadline-bounded governance as a bounded component of execution-bound governance. The analysis connects typed state space, invariant preservation, reachability, uncertainty set, composition, refinement, and mechanized evidence. The decisive test is not whether the system can describe the relevant rule, but whether the rule changes the set of reachable consequence-bearing transitions. The claims in this chapter are model-internal unless a later correspondence argument binds the formal object to source, binary, configuration, hardware, deployment, and evidence.

18.2 Structural Model

A compact representation used in this chapter is:

$$M_P = t_{irr} - t_{effect}$$

The structural object is represented as a typed relation rather than an informal checklist. Identity, object, context, authority, version, time, fault state, and evidence are separated because a change in any one of them can create a different transition. The architecture therefore rejects implicit substitution: a permission for one object is not a permission for another; a proof for one model is not a proof for a deployment; an approval before mutation is not approval after mutation. Unknown mandatory state is represented explicitly and does not default to ordinary allow.

The chapter’s principal structural commitments are:

Typed State Space - retained as an explicit state or dependency.

Invariant Preservation - retained as an explicit state or dependency.

Reachability - retained as an explicit state or dependency.

Uncertainty Set - retained as an explicit state or dependency.

Composition - retained as an explicit state or dependency.

Refinement - retained as an explicit state or dependency.

Mechanized Evidence - retained as an explicit state or dependency.

18.3 Execution Reality

Execution reality begins at the last point from which the protected consequence can still be prevented. Upstream analysis, documentation, monitoring, or recommendation may improve decision quality, but it becomes governance only when the consequence path depends on the resulting state. The chapter therefore treats direct capability, alternate interfaces, administrative paths, recovery mechanisms, stale state, and external dependencies as part of the governed object. A system is not credited with a property merely because a dashboard displays it; the relevant gate, controller, institution, or physical interface must enforce or evidence it.

18.4 Failure Point and Counter-Construction

The claim collapses under any in-scope counter-construction such as:

the model omits a consequence path

proof assumptions are hidden

component properties are assumed to compose

solver timeout is reported as proof

These failures are not edge cases external to the theory. They are the primary falsification instruments. Where one of them remains reachable, the correct result is a narrowed claim, an unresolved state, or a failed invariant - not a narrative assertion that the architecture is nevertheless governed.

18.5 Verification and Evidence Obligations

formalize initial states and transition semantics

prove or test inductiveness

enumerate uncontrollable and bypass paths

bind proof artifacts to model and assumptions

Evidence must remain bound to the exact artifact, configuration, version, operating envelope, and decision state examined. A passing result establishes only the tested property and coverage. A failed or indeterminate result remains part of the authoritative record. Where independent reproduction has not occurred, the result is not represented as independently established.

18.6 Selected Propositions

P18.1. The validity of temporal logic, timed automata, and deadline-bounded governance cannot exceed the scope of its weakest mandatory dependency.

P18.2. A descriptive or advisory representation does not become an execution constraint until it controls the relevant consequence path.

P18.3. Component-level validity does not compose automatically across interfaces, timing, authority, failure, or deployment change.

P18.4. Evidence of absence is not inferred from absence of evidence; unresolved mandatory state remains unresolved.

18.7 Governance Status and Confidence Type

STRUCTURE. Specified as part of the integrated federated architecture.
EXECUTION REALITY. Requires a domain-specific implementation and complete path inventory.
FAILURE POINT. Any surviving bypass, stale authority, omitted dependency, unmodeled timing path, or unsupported evidence link narrows or defeats the claim.
GOVERNANCE STATUS. M1 unless a chapter-specific case study states a higher internal implementation maturity; no institutional compliance determination is inferred.
CONFIDENCE TYPE. Definitional for the architecture, deductive only under stated assumptions, implementation-dependent for runtime and hardware, institutionally imported for authority, and evidence-dependent for deployment claims.

Chapter 19

Hybrid Governance Systems, Barrier Conditions, and Safe-State Reachability

Flows, jumps, viability kernels, control barrier functions, and active safe-state intervention.

19.1 Objective

The objective of this chapter is to define hybrid governance systems, barrier conditions, and safe-state reachability as a bounded component of execution-bound governance. The analysis connects typed state space, invariant preservation, reachability, uncertainty set, composition, refinement, and mechanized evidence. The decisive test is not whether the system can describe the relevant rule, but whether the rule changes the set of reachable consequence-bearing transitions. The claims in this chapter are model-internal unless a later correspondence argument binds the formal object to source, binary, configuration, hardware, deployment, and evidence.

19.2 Structural Model

A compact representation used in this chapter is:

$$\dot{h}(x)+\alpha(h(x))\geq 0$$

The structural object is represented as a typed relation rather than an informal checklist. Identity, object, context, authority, version, time, fault state, and evidence are separated because a change in any one of them can create a different transition. The architecture therefore rejects implicit substitution: a permission for one object is not a permission for another; a proof for one model is not a proof for a deployment; an approval before mutation is not approval after mutation. Unknown mandatory state is represented explicitly and does not default to ordinary allow.

The chapter’s principal structural commitments are:

Typed State Space - retained as an explicit state or dependency.

Invariant Preservation - retained as an explicit state or dependency.

Reachability - retained as an explicit state or dependency.

Uncertainty Set - retained as an explicit state or dependency.

Composition - retained as an explicit state or dependency.

Refinement - retained as an explicit state or dependency.

Mechanized Evidence - retained as an explicit state or dependency.

19.3 Execution Reality

Execution reality begins at the last point from which the protected consequence can still be prevented. Upstream analysis, documentation, monitoring, or recommendation may improve decision quality, but it becomes governance only when the consequence path depends on the resulting state. The chapter therefore treats direct capability, alternate interfaces, administrative paths, recovery mechanisms, stale state, and external dependencies as part of the governed object. A system is not credited with a property merely because a dashboard displays it; the relevant gate, controller, institution, or physical interface must enforce or evidence it.

19.4 Failure Point and Counter-Construction

The claim collapses under any in-scope counter-construction such as:

the model omits a consequence path

proof assumptions are hidden

component properties are assumed to compose

solver timeout is reported as proof

These failures are not edge cases external to the theory. They are the primary falsification instruments. Where one of them remains reachable, the correct result is a narrowed claim, an unresolved state, or a failed invariant - not a narrative assertion that the architecture is nevertheless governed.

19.5 Verification and Evidence Obligations

formalize initial states and transition semantics

prove or test inductiveness

enumerate uncontrollable and bypass paths

bind proof artifacts to model and assumptions

Evidence must remain bound to the exact artifact, configuration, version, operating envelope, and decision state examined. A passing result establishes only the tested property and coverage. A failed or indeterminate result remains part of the authoritative record. Where independent reproduction has not occurred, the result is not represented as independently established.

19.6 Selected Propositions

P19.1. The validity of hybrid governance systems, barrier conditions, and safe-state reachability cannot exceed the scope of its weakest mandatory dependency.

P19.2. A descriptive or advisory representation does not become an execution constraint until it controls the relevant consequence path.

P19.3. Component-level validity does not compose automatically across interfaces, timing, authority, failure, or deployment change.

P19.4. Evidence of absence is not inferred from absence of evidence; unresolved mandatory state remains unresolved.

19.7 Governance Status and Confidence Type

STRUCTURE. Specified as part of the integrated federated architecture.
EXECUTION REALITY. Requires a domain-specific implementation and complete path inventory.
FAILURE POINT. Any surviving bypass, stale authority, omitted dependency, unmodeled timing path, or unsupported evidence link narrows or defeats the claim.
GOVERNANCE STATUS. M1 unless a chapter-specific case study states a higher internal implementation maturity; no institutional compliance determination is inferred.
CONFIDENCE TYPE. Definitional for the architecture, deductive only under stated assumptions, implementation-dependent for runtime and hardware, institutionally imported for authority, and evidence-dependent for deployment claims.

Chapter 20

Refinement, Bisimulation, and Model-Implementation Correspondence

Proof transfer from specification to source, binary, configuration, hardware, deployment, and runtime.

20.1 Objective

The objective of this chapter is to define refinement, bisimulation, and model-implementation correspondence as a bounded component of execution-bound governance. The analysis connects typed state space, invariant preservation, reachability, uncertainty set, composition, refinement, and mechanized evidence. The decisive test is not whether the system can describe the relevant rule, but whether the rule changes the set of reachable consequence-bearing transitions. The claims in this chapter are model-internal unless a later correspondence argument binds the formal object to source, binary, configuration, hardware, deployment, and evidence.

20.2 Structural Model

A compact representation used in this chapter is:

$$\left|\left(Traces\left(C\right)\right)\right| \subseteq Traces\left(A\right)$$

The structural object is represented as a typed relation rather than an informal checklist. Identity, object, context, authority, version, time, fault state, and evidence are separated because a change in any one of them can create a different transition. The architecture therefore rejects implicit substitution: a permission for one object is not a permission for another; a proof for one model is not a proof for a deployment; an approval before mutation is not approval after mutation. Unknown mandatory state is represented explicitly and does not default to ordinary allow.

The chapter’s principal structural commitments are:

Typed State Space - retained as an explicit state or dependency.

Invariant Preservation - retained as an explicit state or dependency.

Reachability - retained as an explicit state or dependency.

Uncertainty Set - retained as an explicit state or dependency.

Composition - retained as an explicit state or dependency.

Refinement - retained as an explicit state or dependency.

Mechanized Evidence - retained as an explicit state or dependency.

20.3 Execution Reality

Execution reality begins at the last point from which the protected consequence can still be prevented. Upstream analysis, documentation, monitoring, or recommendation may improve decision quality, but it becomes governance only when the consequence path depends on the resulting state. The chapter therefore treats direct capability, alternate interfaces, administrative paths, recovery mechanisms, stale state, and external dependencies as part of the governed object. A system is not credited with a property merely because a dashboard displays it; the relevant gate, controller, institution, or physical interface must enforce or evidence it.

20.4 Failure Point and Counter-Construction

The claim collapses under any in-scope counter-construction such as:

the model omits a consequence path

proof assumptions are hidden

component properties are assumed to compose

solver timeout is reported as proof

These failures are not edge cases external to the theory. They are the primary falsification instruments. Where one of them remains reachable, the correct result is a narrowed claim, an unresolved state, or a failed invariant - not a narrative assertion that the architecture is nevertheless governed.

20.5 Verification and Evidence Obligations

formalize initial states and transition semantics

prove or test inductiveness

enumerate uncontrollable and bypass paths

bind proof artifacts to model and assumptions

Evidence must remain bound to the exact artifact, configuration, version, operating envelope, and decision state examined. A passing result establishes only the tested property and coverage. A failed or indeterminate result remains part of the authoritative record. Where independent reproduction has not occurred, the result is not represented as independently established.

20.6 Selected Propositions

P20.1. The validity of refinement, bisimulation, and model-implementation correspondence cannot exceed the scope of its weakest mandatory dependency.

P20.2. A descriptive or advisory representation does not become an execution constraint until it controls the relevant consequence path.

P20.3. Component-level validity does not compose automatically across interfaces, timing, authority, failure, or deployment change.

P20.4. Evidence of absence is not inferred from absence of evidence; unresolved mandatory state remains unresolved.

20.7 Governance Status and Confidence Type

STRUCTURE. Specified as part of the integrated federated architecture.
EXECUTION REALITY. Requires a domain-specific implementation and complete path inventory.
FAILURE POINT. Any surviving bypass, stale authority, omitted dependency, unmodeled timing path, or unsupported evidence link narrows or defeats the claim.
GOVERNANCE STATUS. M1 unless a chapter-specific case study states a higher internal implementation maturity; no institutional compliance determination is inferred.
CONFIDENCE TYPE. Definitional for the architecture, deductive only under stated assumptions, implementation-dependent for runtime and hardware, institutionally imported for authority, and evidence-dependent for deployment claims.

Chapter 21

Decidability, Computational Complexity, and Verification Boundaries

Finite reachability, undecidability, bounded model checking, abstraction, solver unknowns, and assurance shells.

21.1 Objective

The objective of this chapter is to define decidability, computational complexity, and verification boundaries as a bounded component of execution-bound governance. The analysis connects typed state space, invariant preservation, reachability, uncertainty set, composition, refinement, and mechanized evidence. The decisive test is not whether the system can describe the relevant rule, but whether the rule changes the set of reachable consequence-bearing transitions. The claims in this chapter are model-internal unless a later correspondence argument binds the formal object to source, binary, configuration, hardware, deployment, and evidence.

21.2 Structural Model

A compact representation used in this chapter is:

$$I(z_0) \wedge \bigwedge_{i=0}^{k-1} T(z_i, u_i, z_{i+1}) \wedge \bigvee_{i=0}^{k} B(z_i)$$

The structural object is represented as a typed relation rather than an informal checklist. Identity, object, context, authority, version, time, fault state, and evidence are separated because a change in any one of them can create a different transition. The architecture therefore rejects implicit substitution: a permission for one object is not a permission for another; a proof for one model is not a proof for a deployment; an approval before mutation is not approval after mutation. Unknown mandatory state is represented explicitly and does not default to ordinary allow.

The chapter’s principal structural commitments are:

Typed State Space - retained as an explicit state or dependency.

Invariant Preservation - retained as an explicit state or dependency.

Reachability - retained as an explicit state or dependency.

Uncertainty Set - retained as an explicit state or dependency.

Composition - retained as an explicit state or dependency.

Refinement - retained as an explicit state or dependency.

Mechanized Evidence - retained as an explicit state or dependency.

21.3 Execution Reality

Execution reality begins at the last point from which the protected consequence can still be prevented. Upstream analysis, documentation, monitoring, or recommendation may improve decision quality, but it becomes governance only when the consequence path depends on the resulting state. The chapter therefore treats direct capability, alternate interfaces, administrative paths, recovery mechanisms, stale state, and external dependencies as part of the governed object. A system is not credited with a property merely because a dashboard displays it; the relevant gate, controller, institution, or physical interface must enforce or evidence it.

21.4 Failure Point and Counter-Construction

The claim collapses under any in-scope counter-construction such as:

the model omits a consequence path

proof assumptions are hidden

component properties are assumed to compose

solver timeout is reported as proof

These failures are not edge cases external to the theory. They are the primary falsification instruments. Where one of them remains reachable, the correct result is a narrowed claim, an unresolved state, or a failed invariant - not a narrative assertion that the architecture is nevertheless governed.

21.5 Verification and Evidence Obligations

formalize initial states and transition semantics

prove or test inductiveness

enumerate uncontrollable and bypass paths

bind proof artifacts to model and assumptions

Evidence must remain bound to the exact artifact, configuration, version, operating envelope, and decision state examined. A passing result establishes only the tested property and coverage. A failed or indeterminate result remains part of the authoritative record. Where independent reproduction has not occurred, the result is not represented as independently established.

21.6 Selected Propositions

P21.1. The validity of decidability, computational complexity, and verification boundaries cannot exceed the scope of its weakest mandatory dependency.

P21.2. A descriptive or advisory representation does not become an execution constraint until it controls the relevant consequence path.

P21.3. Component-level validity does not compose automatically across interfaces, timing, authority, failure, or deployment change.

P21.4. Evidence of absence is not inferred from absence of evidence; unresolved mandatory state remains unresolved.

21.7 Governance Status and Confidence Type

STRUCTURE. Specified as part of the integrated federated architecture.
EXECUTION REALITY. Requires a domain-specific implementation and complete path inventory.
FAILURE POINT. Any surviving bypass, stale authority, omitted dependency, unmodeled timing path, or unsupported evidence link narrows or defeats the claim.
GOVERNANCE STATUS. M1 unless a chapter-specific case study states a higher internal implementation maturity; no institutional compliance determination is inferred.
CONFIDENCE TYPE. Definitional for the architecture, deductive only under stated assumptions, implementation-dependent for runtime and hardware, institutionally imported for authority, and evidence-dependent for deployment claims.

Chapter 22

Mechanized Specification, Proof Architecture, and Reproducible Formal Evidence

Machine-checkable models, proof obligations, certificates, manifests, trusted computing bases, and reproduction levels.

22.1 Objective

The objective of this chapter is to define mechanized specification, proof architecture, and reproducible formal evidence as a bounded component of execution-bound governance. The analysis connects typed state space, invariant preservation, reachability, uncertainty set, composition, refinement, and mechanized evidence. The decisive test is not whether the system can describe the relevant rule, but whether the rule changes the set of reachable consequence-bearing transitions. The claims in this chapter are model-internal unless a later correspondence argument binds the formal object to source, binary, configuration, hardware, deployment, and evidence.

22.2 Structural Model

A compact representation used in this chapter is:

$$C_H \xrightarrow{R_{CM}} C_M \xrightarrow{Proof} Cert \xrightarrow{Check} Result \xrightarrow{R_D} DeploymentClaim$$

The structural object is represented as a typed relation rather than an informal checklist. Identity, object, context, authority, version, time, fault state, and evidence are separated because a change in any one of them can create a different transition. The architecture therefore rejects implicit substitution: a permission for one object is not a permission for another; a proof for one model is not a proof for a deployment; an approval before mutation is not approval after mutation. Unknown mandatory state is represented explicitly and does not default to ordinary allow.

The chapter’s principal structural commitments are:

Typed State Space - retained as an explicit state or dependency.

Invariant Preservation - retained as an explicit state or dependency.

Reachability - retained as an explicit state or dependency.

Uncertainty Set - retained as an explicit state or dependency.

Composition - retained as an explicit state or dependency.

Refinement - retained as an explicit state or dependency.

Mechanized Evidence - retained as an explicit state or dependency.

22.3 Execution Reality

Execution reality begins at the last point from which the protected consequence can still be prevented. Upstream analysis, documentation, monitoring, or recommendation may improve decision quality, but it becomes governance only when the consequence path depends on the resulting state. The chapter therefore treats direct capability, alternate interfaces, administrative paths, recovery mechanisms, stale state, and external dependencies as part of the governed object. A system is not credited with a property merely because a dashboard displays it; the relevant gate, controller, institution, or physical interface must enforce or evidence it.

22.4 Failure Point and Counter-Construction

The claim collapses under any in-scope counter-construction such as:

the model omits a consequence path

proof assumptions are hidden

component properties are assumed to compose

solver timeout is reported as proof

These failures are not edge cases external to the theory. They are the primary falsification instruments. Where one of them remains reachable, the correct result is a narrowed claim, an unresolved state, or a failed invariant - not a narrative assertion that the architecture is nevertheless governed.

22.5 Verification and Evidence Obligations

formalize initial states and transition semantics

prove or test inductiveness

enumerate uncontrollable and bypass paths

bind proof artifacts to model and assumptions

Evidence must remain bound to the exact artifact, configuration, version, operating envelope, and decision state examined. A passing result establishes only the tested property and coverage. A failed or indeterminate result remains part of the authoritative record. Where independent reproduction has not occurred, the result is not represented as independently established.

22.6 Selected Propositions

P22.1. The validity of mechanized specification, proof architecture, and reproducible formal evidence cannot exceed the scope of its weakest mandatory dependency.

P22.2. A descriptive or advisory representation does not become an execution constraint until it controls the relevant consequence path.

P22.3. Component-level validity does not compose automatically across interfaces, timing, authority, failure, or deployment change.

P22.4. Evidence of absence is not inferred from absence of evidence; unresolved mandatory state remains unresolved.

22.7 Governance Status and Confidence Type

STRUCTURE. Specified as part of the integrated federated architecture.
EXECUTION REALITY. Requires a domain-specific implementation and complete path inventory.
FAILURE POINT. Any surviving bypass, stale authority, omitted dependency, unmodeled timing path, or unsupported evidence link narrows or defeats the claim.
GOVERNANCE STATUS. M1 unless a chapter-specific case study states a higher internal implementation maturity; no institutional compliance determination is inferred.
CONFIDENCE TYPE. Definitional for the architecture, deductive only under stated assumptions, implementation-dependent for runtime and hardware, institutionally imported for authority, and evidence-dependent for deployment claims.

Chapter 23

Formal-Core Synthesis and the Closure of the Mathematical Volume

The unified governed-system object, theorem chain, scope conservation, non-claims, and faculty proof obligations.

23.1 Objective

The objective of this chapter is to define formal-core synthesis and the closure of the mathematical volume as a bounded component of execution-bound governance. The analysis connects typed state space, invariant preservation, reachability, uncertainty set, composition, refinement, and mechanized evidence. The decisive test is not whether the system can describe the relevant rule, but whether the rule changes the set of reachable consequence-bearing transitions. The claims in this chapter are model-internal unless a later correspondence argument binds the formal object to source, binary, configuration, hardware, deployment, and evidence.

23.2 Structural Model

A compact representation used in this chapter is:

$$PO_{SS}=\{PO_{Sem},PO_{Init},PO_{Inv},PO_{Cut},PO_{Time},PO_{Auth},PO_{Unc},PO_{Comp},PO_{Ref},PO_{Corr},PO_{Evid}\}$$

The structural object is represented as a typed relation rather than an informal checklist. Identity, object, context, authority, version, time, fault state, and evidence are separated because a change in any one of them can create a different transition. The architecture therefore rejects implicit substitution: a permission for one object is not a permission for another; a proof for one model is not a proof for a deployment; an approval before mutation is not approval after mutation. Unknown mandatory state is represented explicitly and does not default to ordinary allow.

The chapter’s principal structural commitments are:

Typed State Space - retained as an explicit state or dependency.

Invariant Preservation - retained as an explicit state or dependency.

Reachability - retained as an explicit state or dependency.

Uncertainty Set - retained as an explicit state or dependency.

Composition - retained as an explicit state or dependency.

Refinement - retained as an explicit state or dependency.

Mechanized Evidence - retained as an explicit state or dependency.

23.3 Execution Reality

Execution reality begins at the last point from which the protected consequence can still be prevented. Upstream analysis, documentation, monitoring, or recommendation may improve decision quality, but it becomes governance only when the consequence path depends on the resulting state. The chapter therefore treats direct capability, alternate interfaces, administrative paths, recovery mechanisms, stale state, and external dependencies as part of the governed object. A system is not credited with a property merely because a dashboard displays it; the relevant gate, controller, institution, or physical interface must enforce or evidence it.

23.4 Failure Point and Counter-Construction

The claim collapses under any in-scope counter-construction such as:

the model omits a consequence path

proof assumptions are hidden

component properties are assumed to compose

solver timeout is reported as proof

These failures are not edge cases external to the theory. They are the primary falsification instruments. Where one of them remains reachable, the correct result is a narrowed claim, an unresolved state, or a failed invariant - not a narrative assertion that the architecture is nevertheless governed.

23.5 Verification and Evidence Obligations

formalize initial states and transition semantics

prove or test inductiveness

enumerate uncontrollable and bypass paths

bind proof artifacts to model and assumptions

Evidence must remain bound to the exact artifact, configuration, version, operating envelope, and decision state examined. A passing result establishes only the tested property and coverage. A failed or indeterminate result remains part of the authoritative record. Where independent reproduction has not occurred, the result is not represented as independently established.

23.6 Selected Propositions

P23.1. The validity of formal-core synthesis and the closure of the mathematical volume cannot exceed the scope of its weakest mandatory dependency.

P23.2. A descriptive or advisory representation does not become an execution constraint until it controls the relevant consequence path.

P23.3. Component-level validity does not compose automatically across interfaces, timing, authority, failure, or deployment change.

P23.4. Evidence of absence is not inferred from absence of evidence; unresolved mandatory state remains unresolved.

23.7 Governance Status and Confidence Type

STRUCTURE. Specified as part of the integrated federated architecture.
EXECUTION REALITY. Requires a domain-specific implementation and complete path inventory.
FAILURE POINT. Any surviving bypass, stale authority, omitted dependency, unmodeled timing path, or unsupported evidence link narrows or defeats the claim.
GOVERNANCE STATUS. M1 unless a chapter-specific case study states a higher internal implementation maturity; no institutional compliance determination is inferred.
CONFIDENCE TYPE. Definitional for the architecture, deductive only under stated assumptions, implementation-dependent for runtime and hardware, institutionally imported for authority, and evidence-dependent for deployment claims.

PART III - CONTROL-THEORETIC FOUNDATIONS OF EXECUTION-BOUND GOVERNANCE

Part III asks whether the proposed governance relation is dynamically feasible. It treats the plant, controller, governor, delay, disturbance, actuation, and safe-state path as a single control problem.

Part-level claim boundary. The material in this part is specified at M1 unless a bounded case study explicitly records internal implementation evidence. No part-level statement independently widens the system-level claim.

Chapter 24

Governed Dynamical Systems and Control Allocation

Plant models, controller-governor separation, admissible control sets, authority-constrained actuation, and the control meaning of Structural Sovereignty.

24.1 Objective

The objective of this chapter is to define governed dynamical systems and control allocation as a bounded component of execution-bound governance. The analysis connects plant and controller separation, admissible control set, robust invariance, safe-state reachability, timing margin, and actuator feasibility. The decisive test is not whether the system can describe the relevant rule, but whether the rule changes the set of reachable consequence-bearing transitions. The control claim is bounded by the plant model, disturbance set, actuator limits, sensing and computation delay, and the physical capture region of the declared safe state.

24.2 Structural Model

A compact representation used in this chapter is:

$$\dot{x}=f\left(x,u,d\right),u\in U_{gov}\left(x,a,v\right)$$

The structural object is represented as a typed relation rather than an informal checklist. Identity, object, context, authority, version, time, fault state, and evidence are separated because a change in any one of them can create a different transition. The architecture therefore rejects implicit substitution: a permission for one object is not a permission for another; a proof for one model is not a proof for a deployment; an approval before mutation is not approval after mutation. Unknown mandatory state is represented explicitly and does not default to ordinary allow.

The chapter’s principal structural commitments are:

Plant And Controller Separation - retained as an explicit state or dependency.

Admissible Control Set - retained as an explicit state or dependency.

Robust Invariance - retained as an explicit state or dependency.

Safe-State Reachability - retained as an explicit state or dependency.

Timing Margin - retained as an explicit state or dependency.

Actuator Feasibility - retained as an explicit state or dependency.

24.3 Execution Reality

Execution reality begins at the last point from which the protected consequence can still be prevented. Upstream analysis, documentation, monitoring, or recommendation may improve decision quality, but it becomes governance only when the consequence path depends on the resulting state. The chapter therefore treats direct capability, alternate interfaces, administrative paths, recovery mechanisms, stale state, and external dependencies as part of the governed object. A system is not credited with a property merely because a dashboard displays it; the relevant gate, controller, institution, or physical interface must enforce or evidence it.

24.4 Failure Point and Counter-Construction

The claim collapses under any in-scope counter-construction such as:

the admissible control set becomes empty

delay exceeds the intervention margin

the safe state requires unavailable active control

model mismatch is treated as negligible

These failures are not edge cases external to the theory. They are the primary falsification instruments. Where one of them remains reachable, the correct result is a narrowed claim, an unresolved state, or a failed invariant - not a narrative assertion that the architecture is nevertheless governed.

24.5 Verification and Evidence Obligations

demonstrate feasibility inside the operating envelope

bound disturbances and delay

show safe-state reachability

separate safety from liveness

Evidence must remain bound to the exact artifact, configuration, version, operating envelope, and decision state examined. A passing result establishes only the tested property and coverage. A failed or indeterminate result remains part of the authoritative record. Where independent reproduction has not occurred, the result is not represented as independently established.

24.6 Selected Propositions

P24.1. The validity of governed dynamical systems and control allocation cannot exceed the scope of its weakest mandatory dependency.

P24.2. A descriptive or advisory representation does not become an execution constraint until it controls the relevant consequence path.

P24.3. Component-level validity does not compose automatically across interfaces, timing, authority, failure, or deployment change.

P24.4. Evidence of absence is not inferred from absence of evidence; unresolved mandatory state remains unresolved.

24.7 Governance Status and Confidence Type

STRUCTURE. Specified as part of the integrated federated architecture.
EXECUTION REALITY. Requires a domain-specific implementation and complete path inventory.
FAILURE POINT. Any surviving bypass, stale authority, omitted dependency, unmodeled timing path, or unsupported evidence link narrows or defeats the claim.
GOVERNANCE STATUS. M1 unless a chapter-specific case study states a higher internal implementation maturity; no institutional compliance determination is inferred.
CONFIDENCE TYPE. Definitional for the architecture, deductive only under stated assumptions, implementation-dependent for runtime and hardware, institutionally imported for authority, and evidence-dependent for deployment claims.

Chapter 25

Stability, Damping, and Bounded State Evolution

Corrected state equations, equilibrium assumptions, Lyapunov reasoning, spectral conditions, and bounded structural load.

25.1 Objective

The objective of this chapter is to define stability, damping, and bounded state evolution as a bounded component of execution-bound governance. The analysis connects plant and controller separation, admissible control set, robust invariance, safe-state reachability, timing margin, and actuator feasibility. The decisive test is not whether the system can describe the relevant rule, but whether the rule changes the set of reachable consequence-bearing transitions. The control claim is bounded by the plant model, disturbance set, actuator limits, sensing and computation delay, and the physical capture region of the declared safe state.

25.2 Structural Model

A compact representation used in this chapter is:

$$\dot{S}(t) = R(t) - \gamma S(t), \gamma > 0$$

The structural object is represented as a typed relation rather than an informal checklist. Identity, object, context, authority, version, time, fault state, and evidence are separated because a change in any one of them can create a different transition. The architecture therefore rejects implicit substitution: a permission for one object is not a permission for another; a proof for one model is not a proof for a deployment; an approval before mutation is not approval after mutation. Unknown mandatory state is represented explicitly and does not default to ordinary allow.

The chapter’s principal structural commitments are:

Plant And Controller Separation - retained as an explicit state or dependency.

Admissible Control Set - retained as an explicit state or dependency.

Robust Invariance - retained as an explicit state or dependency.

Safe-State Reachability - retained as an explicit state or dependency.

Timing Margin - retained as an explicit state or dependency.

Actuator Feasibility - retained as an explicit state or dependency.

25.3 Execution Reality

Execution reality begins at the last point from which the protected consequence can still be prevented. Upstream analysis, documentation, monitoring, or recommendation may improve decision quality, but it becomes governance only when the consequence path depends on the resulting state. The chapter therefore treats direct capability, alternate interfaces, administrative paths, recovery mechanisms, stale state, and external dependencies as part of the governed object. A system is not credited with a property merely because a dashboard displays it; the relevant gate, controller, institution, or physical interface must enforce or evidence it.

25.4 Failure Point and Counter-Construction

The claim collapses under any in-scope counter-construction such as:

the admissible control set becomes empty

delay exceeds the intervention margin

the safe state requires unavailable active control

model mismatch is treated as negligible

These failures are not edge cases external to the theory. They are the primary falsification instruments. Where one of them remains reachable, the correct result is a narrowed claim, an unresolved state, or a failed invariant - not a narrative assertion that the architecture is nevertheless governed.

25.5 Verification and Evidence Obligations

demonstrate feasibility inside the operating envelope

bound disturbances and delay

show safe-state reachability

separate safety from liveness

Evidence must remain bound to the exact artifact, configuration, version, operating envelope, and decision state examined. A passing result establishes only the tested property and coverage. A failed or indeterminate result remains part of the authoritative record. Where independent reproduction has not occurred, the result is not represented as independently established.

25.6 Selected Propositions

P25.1. The validity of stability, damping, and bounded state evolution cannot exceed the scope of its weakest mandatory dependency.

P25.2. A descriptive or advisory representation does not become an execution constraint until it controls the relevant consequence path.

P25.3. Component-level validity does not compose automatically across interfaces, timing, authority, failure, or deployment change.

P25.4. Evidence of absence is not inferred from absence of evidence; unresolved mandatory state remains unresolved.

25.7 Governance Status and Confidence Type

STRUCTURE. Specified as part of the integrated federated architecture.
EXECUTION REALITY. Requires a domain-specific implementation and complete path inventory.
FAILURE POINT. Any surviving bypass, stale authority, omitted dependency, unmodeled timing path, or unsupported evidence link narrows or defeats the claim.
GOVERNANCE STATUS. M1 unless a chapter-specific case study states a higher internal implementation maturity; no institutional compliance determination is inferred.
CONFIDENCE TYPE. Definitional for the architecture, deductive only under stated assumptions, implementation-dependent for runtime and hardware, institutionally imported for authority, and evidence-dependent for deployment claims.

Chapter 26

Admissible Control Sets and Safety Filters

Control-set intersection, feasibility, safety filters, fallback actions, and the difference between modifying a command and removing its path.

26.1 Objective

The objective of this chapter is to define admissible control sets and safety filters as a bounded component of execution-bound governance. The analysis connects plant and controller separation, admissible control set, robust invariance, safe-state reachability, timing margin, and actuator feasibility. The decisive test is not whether the system can describe the relevant rule, but whether the rule changes the set of reachable consequence-bearing transitions. The control claim is bounded by the plant model, disturbance set, actuator limits, sensing and computation delay, and the physical capture region of the declared safe state.

26.2 Structural Model

A compact representation used in this chapter is:

$$U_G(x) = U_{plant}(x) \cap U_{safe}(x) \cap U_{auth}(x) \cap U_{enf}(x)$$

The structural object is represented as a typed relation rather than an informal checklist. Identity, object, context, authority, version, time, fault state, and evidence are separated because a change in any one of them can create a different transition. The architecture therefore rejects implicit substitution: a permission for one object is not a permission for another; a proof for one model is not a proof for a deployment; an approval before mutation is not approval after mutation. Unknown mandatory state is represented explicitly and does not default to ordinary allow.

The chapter’s principal structural commitments are:

Plant And Controller Separation - retained as an explicit state or dependency.

Admissible Control Set - retained as an explicit state or dependency.

Robust Invariance - retained as an explicit state or dependency.

Safe-State Reachability - retained as an explicit state or dependency.

Timing Margin - retained as an explicit state or dependency.

Actuator Feasibility - retained as an explicit state or dependency.

26.3 Execution Reality

Execution reality begins at the last point from which the protected consequence can still be prevented. Upstream analysis, documentation, monitoring, or recommendation may improve decision quality, but it becomes governance only when the consequence path depends on the resulting state. The chapter therefore treats direct capability, alternate interfaces, administrative paths, recovery mechanisms, stale state, and external dependencies as part of the governed object. A system is not credited with a property merely because a dashboard displays it; the relevant gate, controller, institution, or physical interface must enforce or evidence it.

26.4 Failure Point and Counter-Construction

The claim collapses under any in-scope counter-construction such as:

the admissible control set becomes empty

delay exceeds the intervention margin

the safe state requires unavailable active control

model mismatch is treated as negligible

These failures are not edge cases external to the theory. They are the primary falsification instruments. Where one of them remains reachable, the correct result is a narrowed claim, an unresolved state, or a failed invariant - not a narrative assertion that the architecture is nevertheless governed.

26.5 Verification and Evidence Obligations

demonstrate feasibility inside the operating envelope

bound disturbances and delay

show safe-state reachability

separate safety from liveness

Evidence must remain bound to the exact artifact, configuration, version, operating envelope, and decision state examined. A passing result establishes only the tested property and coverage. A failed or indeterminate result remains part of the authoritative record. Where independent reproduction has not occurred, the result is not represented as independently established.

26.6 Selected Propositions

P26.1. The validity of admissible control sets and safety filters cannot exceed the scope of its weakest mandatory dependency.

P26.2. A descriptive or advisory representation does not become an execution constraint until it controls the relevant consequence path.

P26.3. Component-level validity does not compose automatically across interfaces, timing, authority, failure, or deployment change.

P26.4. Evidence of absence is not inferred from absence of evidence; unresolved mandatory state remains unresolved.

26.7 Governance Status and Confidence Type

STRUCTURE. Specified as part of the integrated federated architecture.
EXECUTION REALITY. Requires a domain-specific implementation and complete path inventory.
FAILURE POINT. Any surviving bypass, stale authority, omitted dependency, unmodeled timing path, or unsupported evidence link narrows or defeats the claim.
GOVERNANCE STATUS. M1 unless a chapter-specific case study states a higher internal implementation maturity; no institutional compliance determination is inferred.
CONFIDENCE TYPE. Definitional for the architecture, deductive only under stated assumptions, implementation-dependent for runtime and hardware, institutionally imported for authority, and evidence-dependent for deployment claims.

Chapter 27

Robust Governance under Disturbance and Model Error

Disturbance sets, robust invariance, uncertainty margins, mismatch detection, and conservative control contraction.

27.1 Objective

The objective of this chapter is to define robust governance under disturbance and model error as a bounded component of execution-bound governance. The analysis connects plant and controller separation, admissible control set, robust invariance, safe-state reachability, timing margin, and actuator feasibility. The decisive test is not whether the system can describe the relevant rule, but whether the rule changes the set of reachable consequence-bearing transitions. The control claim is bounded by the plant model, disturbance set, actuator limits, sensing and computation delay, and the physical capture region of the declared safe state.

27.2 Structural Model

A compact representation used in this chapter is:

$$x(t) \in K \; \forall \, d \in D$$

The structural object is represented as a typed relation rather than an informal checklist. Identity, object, context, authority, version, time, fault state, and evidence are separated because a change in any one of them can create a different transition. The architecture therefore rejects implicit substitution: a permission for one object is not a permission for another; a proof for one model is not a proof for a deployment; an approval before mutation is not approval after mutation. Unknown mandatory state is represented explicitly and does not default to ordinary allow.

The chapter’s principal structural commitments are:

Plant And Controller Separation - retained as an explicit state or dependency.

Admissible Control Set - retained as an explicit state or dependency.

Robust Invariance - retained as an explicit state or dependency.

Safe-State Reachability - retained as an explicit state or dependency.

Timing Margin - retained as an explicit state or dependency.

Actuator Feasibility - retained as an explicit state or dependency.

27.3 Execution Reality

Execution reality begins at the last point from which the protected consequence can still be prevented. Upstream analysis, documentation, monitoring, or recommendation may improve decision quality, but it becomes governance only when the consequence path depends on the resulting state. The chapter therefore treats direct capability, alternate interfaces, administrative paths, recovery mechanisms, stale state, and external dependencies as part of the governed object. A system is not credited with a property merely because a dashboard displays it; the relevant gate, controller, institution, or physical interface must enforce or evidence it.

27.4 Failure Point and Counter-Construction

The claim collapses under any in-scope counter-construction such as:

the admissible control set becomes empty

delay exceeds the intervention margin

the safe state requires unavailable active control

model mismatch is treated as negligible

These failures are not edge cases external to the theory. They are the primary falsification instruments. Where one of them remains reachable, the correct result is a narrowed claim, an unresolved state, or a failed invariant - not a narrative assertion that the architecture is nevertheless governed.

27.5 Verification and Evidence Obligations

demonstrate feasibility inside the operating envelope

bound disturbances and delay

show safe-state reachability

separate safety from liveness

Evidence must remain bound to the exact artifact, configuration, version, operating envelope, and decision state examined. A passing result establishes only the tested property and coverage. A failed or indeterminate result remains part of the authoritative record. Where independent reproduction has not occurred, the result is not represented as independently established.

27.6 Selected Propositions

P27.1. The validity of robust governance under disturbance and model error cannot exceed the scope of its weakest mandatory dependency.

P27.2. A descriptive or advisory representation does not become an execution constraint until it controls the relevant consequence path.

P27.3. Component-level validity does not compose automatically across interfaces, timing, authority, failure, or deployment change.

P27.4. Evidence of absence is not inferred from absence of evidence; unresolved mandatory state remains unresolved.

27.7 Governance Status and Confidence Type

STRUCTURE. Specified as part of the integrated federated architecture.
EXECUTION REALITY. Requires a domain-specific implementation and complete path inventory.
FAILURE POINT. Any surviving bypass, stale authority, omitted dependency, unmodeled timing path, or unsupported evidence link narrows or defeats the claim.
GOVERNANCE STATUS. M1 unless a chapter-specific case study states a higher internal implementation maturity; no institutional compliance determination is inferred.
CONFIDENCE TYPE. Definitional for the architecture, deductive only under stated assumptions, implementation-dependent for runtime and hardware, institutionally imported for authority, and evidence-dependent for deployment claims.

Chapter 28

Control Barrier Functions and Invariant Safety Envelopes

Barrier certificates, feasibility, relative degree, actuator constraints, and limits of local safety filtering.

28.1 Objective

The objective of this chapter is to define control barrier functions and invariant safety envelopes as a bounded component of execution-bound governance. The analysis connects plant and controller separation, admissible control set, robust invariance, safe-state reachability, timing margin, and actuator feasibility. The decisive test is not whether the system can describe the relevant rule, but whether the rule changes the set of reachable consequence-bearing transitions. The control claim is bounded by the plant model, disturbance set, actuator limits, sensing and computation delay, and the physical capture region of the declared safe state.

28.2 Structural Model

A compact representation used in this chapter is:

$$\underset{u \in U}{¿} \left[L_f h(x) + L_g h(x) u + \alpha\left(h(x)\right) \right) \geq 0$$

The structural object is represented as a typed relation rather than an informal checklist. Identity, object, context, authority, version, time, fault state, and evidence are separated because a change in any one of them can create a different transition. The architecture therefore rejects implicit substitution: a permission for one object is not a permission for another; a proof for one model is not a proof for a deployment; an approval before mutation is not approval after mutation. Unknown mandatory state is represented explicitly and does not default to ordinary allow.

The chapter’s principal structural commitments are:

Plant And Controller Separation - retained as an explicit state or dependency.

Admissible Control Set - retained as an explicit state or dependency.

Robust Invariance - retained as an explicit state or dependency.

Safe-State Reachability - retained as an explicit state or dependency.

Timing Margin - retained as an explicit state or dependency.

Actuator Feasibility - retained as an explicit state or dependency.

28.3 Execution Reality

Execution reality begins at the last point from which the protected consequence can still be prevented. Upstream analysis, documentation, monitoring, or recommendation may improve decision quality, but it becomes governance only when the consequence path depends on the resulting state. The chapter therefore treats direct capability, alternate interfaces, administrative paths, recovery mechanisms, stale state, and external dependencies as part of the governed object. A system is not credited with a property merely because a dashboard displays it; the relevant gate, controller, institution, or physical interface must enforce or evidence it.

28.4 Failure Point and Counter-Construction

The claim collapses under any in-scope counter-construction such as:

the admissible control set becomes empty

delay exceeds the intervention margin

the safe state requires unavailable active control

model mismatch is treated as negligible

These failures are not edge cases external to the theory. They are the primary falsification instruments. Where one of them remains reachable, the correct result is a narrowed claim, an unresolved state, or a failed invariant - not a narrative assertion that the architecture is nevertheless governed.

28.5 Verification and Evidence Obligations

demonstrate feasibility inside the operating envelope

bound disturbances and delay

show safe-state reachability

separate safety from liveness

Evidence must remain bound to the exact artifact, configuration, version, operating envelope, and decision state examined. A passing result establishes only the tested property and coverage. A failed or indeterminate result remains part of the authoritative record. Where independent reproduction has not occurred, the result is not represented as independently established.

28.6 Selected Propositions

P28.1. The validity of control barrier functions and invariant safety envelopes cannot exceed the scope of its weakest mandatory dependency.

P28.2. A descriptive or advisory representation does not become an execution constraint until it controls the relevant consequence path.

P28.3. Component-level validity does not compose automatically across interfaces, timing, authority, failure, or deployment change.

P28.4. Evidence of absence is not inferred from absence of evidence; unresolved mandatory state remains unresolved.

28.7 Governance Status and Confidence Type

STRUCTURE. Specified as part of the integrated federated architecture.
EXECUTION REALITY. Requires a domain-specific implementation and complete path inventory.
FAILURE POINT. Any surviving bypass, stale authority, omitted dependency, unmodeled timing path, or unsupported evidence link narrows or defeats the claim.
GOVERNANCE STATUS. M1 unless a chapter-specific case study states a higher internal implementation maturity; no institutional compliance determination is inferred.
CONFIDENCE TYPE. Definitional for the architecture, deductive only under stated assumptions, implementation-dependent for runtime and hardware, institutionally imported for authority, and evidence-dependent for deployment claims.

Chapter 29

Model Predictive Governance and Constraint Horizons

Finite-horizon planning, terminal sets, recursive feasibility, constraint tightening, and horizon failure.

29.1 Objective

The objective of this chapter is to define model predictive governance and constraint horizons as a bounded component of execution-bound governance. The analysis connects plant and controller separation, admissible control set, robust invariance, safe-state reachability, timing margin, and actuator feasibility. The decisive test is not whether the system can describe the relevant rule, but whether the rule changes the set of reachable consequence-bearing transitions. The control claim is bounded by the plant model, disturbance set, actuator limits, sensing and computation delay, and the physical capture region of the declared safe state.

29.2 Structural Model

A compact representation used in this chapter is:

$$\min_{u_{0:N-1}} \sum_{k=0}^{N-1} \ell\left(x_k, u_k\right) + V_f\left(x_N\right)$$

The structural object is represented as a typed relation rather than an informal checklist. Identity, object, context, authority, version, time, fault state, and evidence are separated because a change in any one of them can create a different transition. The architecture therefore rejects implicit substitution: a permission for one object is not a permission for another; a proof for one model is not a proof for a deployment; an approval before mutation is not approval after mutation. Unknown mandatory state is represented explicitly and does not default to ordinary allow.

The chapter’s principal structural commitments are:

Plant And Controller Separation - retained as an explicit state or dependency.

Admissible Control Set - retained as an explicit state or dependency.

Robust Invariance - retained as an explicit state or dependency.

Safe-State Reachability - retained as an explicit state or dependency.

Timing Margin - retained as an explicit state or dependency.

Actuator Feasibility - retained as an explicit state or dependency.

29.3 Execution Reality

Execution reality begins at the last point from which the protected consequence can still be prevented. Upstream analysis, documentation, monitoring, or recommendation may improve decision quality, but it becomes governance only when the consequence path depends on the resulting state. The chapter therefore treats direct capability, alternate interfaces, administrative paths, recovery mechanisms, stale state, and external dependencies as part of the governed object. A system is not credited with a property merely because a dashboard displays it; the relevant gate, controller, institution, or physical interface must enforce or evidence it.

29.4 Failure Point and Counter-Construction

The claim collapses under any in-scope counter-construction such as:

the admissible control set becomes empty

delay exceeds the intervention margin

the safe state requires unavailable active control

model mismatch is treated as negligible

These failures are not edge cases external to the theory. They are the primary falsification instruments. Where one of them remains reachable, the correct result is a narrowed claim, an unresolved state, or a failed invariant - not a narrative assertion that the architecture is nevertheless governed.

29.5 Verification and Evidence Obligations

demonstrate feasibility inside the operating envelope

bound disturbances and delay

show safe-state reachability

separate safety from liveness

Evidence must remain bound to the exact artifact, configuration, version, operating envelope, and decision state examined. A passing result establishes only the tested property and coverage. A failed or indeterminate result remains part of the authoritative record. Where independent reproduction has not occurred, the result is not represented as independently established.

29.6 Selected Propositions

P29.1. The validity of model predictive governance and constraint horizons cannot exceed the scope of its weakest mandatory dependency.

P29.2. A descriptive or advisory representation does not become an execution constraint until it controls the relevant consequence path.

P29.3. Component-level validity does not compose automatically across interfaces, timing, authority, failure, or deployment change.

P29.4. Evidence of absence is not inferred from absence of evidence; unresolved mandatory state remains unresolved.

29.7 Governance Status and Confidence Type

STRUCTURE. Specified as part of the integrated federated architecture.
EXECUTION REALITY. Requires a domain-specific implementation and complete path inventory.
FAILURE POINT. Any surviving bypass, stale authority, omitted dependency, unmodeled timing path, or unsupported evidence link narrows or defeats the claim.
GOVERNANCE STATUS. M1 unless a chapter-specific case study states a higher internal implementation maturity; no institutional compliance determination is inferred.
CONFIDENCE TYPE. Definitional for the architecture, deductive only under stated assumptions, implementation-dependent for runtime and hardware, institutionally imported for authority, and evidence-dependent for deployment claims.

Chapter 30

Hybrid Supervisory Control and Mode Switching

Nominal, restricted, degraded, halt, recovery, and service modes with guarded transitions and no permissive defaults.

30.1 Objective

The objective of this chapter is to define hybrid supervisory control and mode switching as a bounded component of execution-bound governance. The analysis connects plant and controller separation, admissible control set, robust invariance, safe-state reachability, timing margin, and actuator feasibility. The decisive test is not whether the system can describe the relevant rule, but whether the rule changes the set of reachable consequence-bearing transitions. The control claim is bounded by the plant model, disturbance set, actuator limits, sensing and computation delay, and the physical capture region of the declared safe state.

30.2 Structural Model

The structural object is represented as a typed relation rather than an informal checklist. Identity, object, context, authority, version, time, fault state, and evidence are separated because a change in any one of them can create a different transition. The architecture therefore rejects implicit substitution: a permission for one object is not a permission for another; a proof for one model is not a proof for a deployment; an approval before mutation is not approval after mutation. Unknown mandatory state is represented explicitly and does not default to ordinary allow.

The chapter’s principal structural commitments are:

Plant And Controller Separation - retained as an explicit state or dependency.

Admissible Control Set - retained as an explicit state or dependency.

Robust Invariance - retained as an explicit state or dependency.

Safe-State Reachability - retained as an explicit state or dependency.

Timing Margin - retained as an explicit state or dependency.

Actuator Feasibility - retained as an explicit state or dependency.

30.3 Execution Reality

Execution reality begins at the last point from which the protected consequence can still be prevented. Upstream analysis, documentation, monitoring, or recommendation may improve decision quality, but it becomes governance only when the consequence path depends on the resulting state. The chapter therefore treats direct capability, alternate interfaces, administrative paths, recovery mechanisms, stale state, and external dependencies as part of the governed object. A system is not credited with a property merely because a dashboard displays it; the relevant gate, controller, institution, or physical interface must enforce or evidence it.

30.4 Failure Point and Counter-Construction

The claim collapses under any in-scope counter-construction such as:

the admissible control set becomes empty

delay exceeds the intervention margin

the safe state requires unavailable active control

model mismatch is treated as negligible

These failures are not edge cases external to the theory. They are the primary falsification instruments. Where one of them remains reachable, the correct result is a narrowed claim, an unresolved state, or a failed invariant - not a narrative assertion that the architecture is nevertheless governed.

30.5 Verification and Evidence Obligations

demonstrate feasibility inside the operating envelope

bound disturbances and delay

show safe-state reachability

separate safety from liveness

Evidence must remain bound to the exact artifact, configuration, version, operating envelope, and decision state examined. A passing result establishes only the tested property and coverage. A failed or indeterminate result remains part of the authoritative record. Where independent reproduction has not occurred, the result is not represented as independently established.

30.6 Selected Propositions

P30.1. The validity of hybrid supervisory control and mode switching cannot exceed the scope of its weakest mandatory dependency.

P30.2. A descriptive or advisory representation does not become an execution constraint until it controls the relevant consequence path.

P30.3. Component-level validity does not compose automatically across interfaces, timing, authority, failure, or deployment change.

P30.4. Evidence of absence is not inferred from absence of evidence; unresolved mandatory state remains unresolved.

30.7 Governance Status and Confidence Type

STRUCTURE. Specified as part of the integrated federated architecture.
EXECUTION REALITY. Requires a domain-specific implementation and complete path inventory.
FAILURE POINT. Any surviving bypass, stale authority, omitted dependency, unmodeled timing path, or unsupported evidence link narrows or defeats the claim.
GOVERNANCE STATUS. M1 unless a chapter-specific case study states a higher internal implementation maturity; no institutional compliance determination is inferred.
CONFIDENCE TYPE. Definitional for the architecture, deductive only under stated assumptions, implementation-dependent for runtime and hardware, institutionally imported for authority, and evidence-dependent for deployment claims.

Chapter 31

Delay, Sampling, Latency, and Irreversibility

Sampled-data control, computation and communication delay, latest safe intervention, and timing evidence.

31.1 Objective

The objective of this chapter is to define delay, sampling, latency, and irreversibility as a bounded component of execution-bound governance. The analysis connects plant and controller separation, admissible control set, robust invariance, safe-state reachability, timing margin, and actuator feasibility. The decisive test is not whether the system can describe the relevant rule, but whether the rule changes the set of reachable consequence-bearing transitions. The control claim is bounded by the plant model, disturbance set, actuator limits, sensing and computation delay, and the physical capture region of the declared safe state.

31.2 Structural Model

A compact representation used in this chapter is:

$$T_{sense}+T_{compute}+T_{communicate}+T_{actuate}<T_{irreversible}$$

The structural object is represented as a typed relation rather than an informal checklist. Identity, object, context, authority, version, time, fault state, and evidence are separated because a change in any one of them can create a different transition. The architecture therefore rejects implicit substitution: a permission for one object is not a permission for another; a proof for one model is not a proof for a deployment; an approval before mutation is not approval after mutation. Unknown mandatory state is represented explicitly and does not default to ordinary allow.

The chapter’s principal structural commitments are:

Plant And Controller Separation - retained as an explicit state or dependency.

Admissible Control Set - retained as an explicit state or dependency.

Robust Invariance - retained as an explicit state or dependency.

Safe-State Reachability - retained as an explicit state or dependency.

Timing Margin - retained as an explicit state or dependency.

Actuator Feasibility - retained as an explicit state or dependency.

31.3 Execution Reality

Execution reality begins at the last point from which the protected consequence can still be prevented. Upstream analysis, documentation, monitoring, or recommendation may improve decision quality, but it becomes governance only when the consequence path depends on the resulting state. The chapter therefore treats direct capability, alternate interfaces, administrative paths, recovery mechanisms, stale state, and external dependencies as part of the governed object. A system is not credited with a property merely because a dashboard displays it; the relevant gate, controller, institution, or physical interface must enforce or evidence it.

31.4 Failure Point and Counter-Construction

The claim collapses under any in-scope counter-construction such as:

the admissible control set becomes empty

delay exceeds the intervention margin

the safe state requires unavailable active control

model mismatch is treated as negligible

These failures are not edge cases external to the theory. They are the primary falsification instruments. Where one of them remains reachable, the correct result is a narrowed claim, an unresolved state, or a failed invariant - not a narrative assertion that the architecture is nevertheless governed.

31.5 Verification and Evidence Obligations

demonstrate feasibility inside the operating envelope

bound disturbances and delay

show safe-state reachability

separate safety from liveness

Evidence must remain bound to the exact artifact, configuration, version, operating envelope, and decision state examined. A passing result establishes only the tested property and coverage. A failed or indeterminate result remains part of the authoritative record. Where independent reproduction has not occurred, the result is not represented as independently established.

31.6 Selected Propositions

P31.1. The validity of delay, sampling, latency, and irreversibility cannot exceed the scope of its weakest mandatory dependency.

P31.2. A descriptive or advisory representation does not become an execution constraint until it controls the relevant consequence path.

P31.3. Component-level validity does not compose automatically across interfaces, timing, authority, failure, or deployment change.

P31.4. Evidence of absence is not inferred from absence of evidence; unresolved mandatory state remains unresolved.

31.7 Governance Status and Confidence Type

STRUCTURE. Specified as part of the integrated federated architecture.
EXECUTION REALITY. Requires a domain-specific implementation and complete path inventory.
FAILURE POINT. Any surviving bypass, stale authority, omitted dependency, unmodeled timing path, or unsupported evidence link narrows or defeats the claim.
GOVERNANCE STATUS. M1 unless a chapter-specific case study states a higher internal implementation maturity; no institutional compliance determination is inferred.
CONFIDENCE TYPE. Definitional for the architecture, deductive only under stated assumptions, implementation-dependent for runtime and hardware, institutionally imported for authority, and evidence-dependent for deployment claims.

Chapter 32

Distributed Control, Multi-Agent Coupling, and Common-Mode Failure

Networked plants, consensus limits, authority coupling, distributed admissibility, and shared dependency failure.

32.1 Objective

The objective of this chapter is to define distributed control, multi-agent coupling, and common-mode failure as a bounded component of execution-bound governance. The analysis connects plant and controller separation, admissible control set, robust invariance, safe-state reachability, timing margin, and actuator feasibility. The decisive test is not whether the system can describe the relevant rule, but whether the rule changes the set of reachable consequence-bearing transitions. The control claim is bounded by the plant model, disturbance set, actuator limits, sensing and computation delay, and the physical capture region of the declared safe state.

32.2 Structural Model

The structural object is represented as a typed relation rather than an informal checklist. Identity, object, context, authority, version, time, fault state, and evidence are separated because a change in any one of them can create a different transition. The architecture therefore rejects implicit substitution: a permission for one object is not a permission for another; a proof for one model is not a proof for a deployment; an approval before mutation is not approval after mutation. Unknown mandatory state is represented explicitly and does not default to ordinary allow.

The chapter’s principal structural commitments are:

Plant And Controller Separation - retained as an explicit state or dependency.

Admissible Control Set - retained as an explicit state or dependency.

Robust Invariance - retained as an explicit state or dependency.

Safe-State Reachability - retained as an explicit state or dependency.

Timing Margin - retained as an explicit state or dependency.

Actuator Feasibility - retained as an explicit state or dependency.

32.3 Execution Reality

Execution reality begins at the last point from which the protected consequence can still be prevented. Upstream analysis, documentation, monitoring, or recommendation may improve decision quality, but it becomes governance only when the consequence path depends on the resulting state. The chapter therefore treats direct capability, alternate interfaces, administrative paths, recovery mechanisms, stale state, and external dependencies as part of the governed object. A system is not credited with a property merely because a dashboard displays it; the relevant gate, controller, institution, or physical interface must enforce or evidence it.

32.4 Failure Point and Counter-Construction

The claim collapses under any in-scope counter-construction such as:

the admissible control set becomes empty

delay exceeds the intervention margin

the safe state requires unavailable active control

model mismatch is treated as negligible

These failures are not edge cases external to the theory. They are the primary falsification instruments. Where one of them remains reachable, the correct result is a narrowed claim, an unresolved state, or a failed invariant - not a narrative assertion that the architecture is nevertheless governed.

32.5 Verification and Evidence Obligations

demonstrate feasibility inside the operating envelope

bound disturbances and delay

show safe-state reachability

separate safety from liveness

Evidence must remain bound to the exact artifact, configuration, version, operating envelope, and decision state examined. A passing result establishes only the tested property and coverage. A failed or indeterminate result remains part of the authoritative record. Where independent reproduction has not occurred, the result is not represented as independently established.

32.6 Selected Propositions

P32.1. The validity of distributed control, multi-agent coupling, and common-mode failure cannot exceed the scope of its weakest mandatory dependency.

P32.2. A descriptive or advisory representation does not become an execution constraint until it controls the relevant consequence path.

P32.3. Component-level validity does not compose automatically across interfaces, timing, authority, failure, or deployment change.

P32.4. Evidence of absence is not inferred from absence of evidence; unresolved mandatory state remains unresolved.

32.7 Governance Status and Confidence Type

STRUCTURE. Specified as part of the integrated federated architecture.
EXECUTION REALITY. Requires a domain-specific implementation and complete path inventory.
FAILURE POINT. Any surviving bypass, stale authority, omitted dependency, unmodeled timing path, or unsupported evidence link narrows or defeats the claim.
GOVERNANCE STATUS. M1 unless a chapter-specific case study states a higher internal implementation maturity; no institutional compliance determination is inferred.
CONFIDENCE TYPE. Definitional for the architecture, deductive only under stated assumptions, implementation-dependent for runtime and hardware, institutionally imported for authority, and evidence-dependent for deployment claims.

Chapter 33

Control-Volume Synthesis and Closure

The master control permit, feasibility and safety conjunction, control failure taxonomy, and export to runtime and hardware.

33.1 Objective

The objective of this chapter is to define control-volume synthesis and closure as a bounded component of execution-bound governance. The analysis connects plant and controller separation, admissible control set, robust invariance, safe-state reachability, timing margin, and actuator feasibility. The decisive test is not whether the system can describe the relevant rule, but whether the rule changes the set of reachable consequence-bearing transitions. The control claim is bounded by the plant model, disturbance set, actuator limits, sensing and computation delay, and the physical capture region of the declared safe state.

33.2 Structural Model

A compact representation used in this chapter is:

$$Permit_C = Feasible \wedge Safe \wedge Authorized \wedge Timely \wedge Enforceable$$

The structural object is represented as a typed relation rather than an informal checklist. Identity, object, context, authority, version, time, fault state, and evidence are separated because a change in any one of them can create a different transition. The architecture therefore rejects implicit substitution: a permission for one object is not a permission for another; a proof for one model is not a proof for a deployment; an approval before mutation is not approval after mutation. Unknown mandatory state is represented explicitly and does not default to ordinary allow.

The chapter’s principal structural commitments are:

Plant And Controller Separation - retained as an explicit state or dependency.

Admissible Control Set - retained as an explicit state or dependency.

Robust Invariance - retained as an explicit state or dependency.

Safe-State Reachability - retained as an explicit state or dependency.

Timing Margin - retained as an explicit state or dependency.

Actuator Feasibility - retained as an explicit state or dependency.

33.3 Execution Reality

Execution reality begins at the last point from which the protected consequence can still be prevented. Upstream analysis, documentation, monitoring, or recommendation may improve decision quality, but it becomes governance only when the consequence path depends on the resulting state. The chapter therefore treats direct capability, alternate interfaces, administrative paths, recovery mechanisms, stale state, and external dependencies as part of the governed object. A system is not credited with a property merely because a dashboard displays it; the relevant gate, controller, institution, or physical interface must enforce or evidence it.

33.4 Failure Point and Counter-Construction

The claim collapses under any in-scope counter-construction such as:

the admissible control set becomes empty

delay exceeds the intervention margin

the safe state requires unavailable active control

model mismatch is treated as negligible

These failures are not edge cases external to the theory. They are the primary falsification instruments. Where one of them remains reachable, the correct result is a narrowed claim, an unresolved state, or a failed invariant - not a narrative assertion that the architecture is nevertheless governed.

33.5 Verification and Evidence Obligations

demonstrate feasibility inside the operating envelope

bound disturbances and delay

show safe-state reachability

separate safety from liveness

Evidence must remain bound to the exact artifact, configuration, version, operating envelope, and decision state examined. A passing result establishes only the tested property and coverage. A failed or indeterminate result remains part of the authoritative record. Where independent reproduction has not occurred, the result is not represented as independently established.

33.6 Selected Propositions

P33.1. The validity of control-volume synthesis and closure cannot exceed the scope of its weakest mandatory dependency.

P33.2. A descriptive or advisory representation does not become an execution constraint until it controls the relevant consequence path.

P33.3. Component-level validity does not compose automatically across interfaces, timing, authority, failure, or deployment change.

P33.4. Evidence of absence is not inferred from absence of evidence; unresolved mandatory state remains unresolved.

33.7 Governance Status and Confidence Type

STRUCTURE. Specified as part of the integrated federated architecture.
EXECUTION REALITY. Requires a domain-specific implementation and complete path inventory.
FAILURE POINT. Any surviving bypass, stale authority, omitted dependency, unmodeled timing path, or unsupported evidence link narrows or defeats the claim.
GOVERNANCE STATUS. M1 unless a chapter-specific case study states a higher internal implementation maturity; no institutional compliance determination is inferred.
CONFIDENCE TYPE. Definitional for the architecture, deductive only under stated assumptions, implementation-dependent for runtime and hardware, institutionally imported for authority, and evidence-dependent for deployment claims.

PART IV - COMPUTER SCIENCE, SYSTEMS ARCHITECTURE, AND RUNTIME ENFORCEMENT

Part IV converts admissibility into runtime restraint. It locates the commit boundary, removes direct model authority, and preserves transaction identity through concurrency, crash, messaging, isolation, deployment, and agentic tool use.

Part-level claim boundary. The material in this part is specified at M1 unless a bounded case study explicitly records internal implementation evidence. No part-level statement independently widens the system-level claim.

Chapter 34

Execution Architecture and the Separation of Proposal, Permission, Commit, and Consequence

Typed execution states, commit boundaries, consequence cut sets, and the rule that model output is never execution authority.

34.1 Objective

The objective of this chapter is to define execution architecture and the separation of proposal, permission, commit, and consequence as a bounded component of execution-bound governance. The analysis connects proposal-permission-commit separation, typed transaction identity, complete mediation, atomicity, durable state, evidence closure, and deployment correspondence. The decisive test is not whether the system can describe the relevant rule, but whether the rule changes the set of reachable consequence-bearing transitions. The runtime claim is bounded by the complete consequence-path inventory, capability model, trusted computing base, concurrency semantics, persistent state, and deployed configuration.

34.2 Structural Model

A compact representation used in this chapter is:

$$\text{Proposal} \rightarrow \text{Permission} \rightarrow \text{Commit} \rightarrow \text{Consequence}$$

The structural object is represented as a typed relation rather than an informal checklist. Identity, object, context, authority, version, time, fault state, and evidence are separated because a change in any one of them can create a different transition. The architecture therefore rejects implicit substitution: a permission for one object is not a permission for another; a proof for one model is not a proof for a deployment; an approval before mutation is not approval after mutation. Unknown mandatory state is represented explicitly and does not default to ordinary allow.

The chapter’s principal structural commitments are:

Proposal-Permission-Commit Separation - retained as an explicit state or dependency.

Typed Transaction Identity - retained as an explicit state or dependency.

Complete Mediation - retained as an explicit state or dependency.

Atomicity - retained as an explicit state or dependency.

Durable State - retained as an explicit state or dependency.

Evidence Closure - retained as an explicit state or dependency.

Deployment Correspondence - retained as an explicit state or dependency.

34.3 Execution Reality

Execution reality begins at the last point from which the protected consequence can still be prevented. Upstream analysis, documentation, monitoring, or recommendation may improve decision quality, but it becomes governance only when the consequence path depends on the resulting state. The chapter therefore treats direct capability, alternate interfaces, administrative paths, recovery mechanisms, stale state, and external dependencies as part of the governed object. A system is not credited with a property merely because a dashboard displays it; the relevant gate, controller, institution, or physical interface must enforce or evidence it.

34.4 Failure Point and Counter-Construction

The claim collapses under any in-scope counter-construction such as:

an alternate API bypasses the gate

post-permit mutation changes the transaction

crash restores consumed authority

external outcome is guessed after timeout

These failures are not edge cases external to the theory. They are the primary falsification instruments. Where one of them remains reachable, the correct result is a narrowed claim, an unresolved state, or a failed invariant - not a narrative assertion that the architecture is nevertheless governed.

34.5 Verification and Evidence Obligations

inventory every consequence path

prove capability and identity binding

test concurrency, crash, retry, and recovery

bind evidence to the deployed artifact and configuration

Evidence must remain bound to the exact artifact, configuration, version, operating envelope, and decision state examined. A passing result establishes only the tested property and coverage. A failed or indeterminate result remains part of the authoritative record. Where independent reproduction has not occurred, the result is not represented as independently established.

34.6 Selected Propositions

P34.1. The validity of execution architecture and the separation of proposal, permission, commit, and consequence cannot exceed the scope of its weakest mandatory dependency.

P34.2. A descriptive or advisory representation does not become an execution constraint until it controls the relevant consequence path.

P34.3. Component-level validity does not compose automatically across interfaces, timing, authority, failure, or deployment change.

P34.4. Evidence of absence is not inferred from absence of evidence; unresolved mandatory state remains unresolved.

34.7 Governance Status and Confidence Type

STRUCTURE. Specified as part of the integrated federated architecture.
EXECUTION REALITY. Requires a domain-specific implementation and complete path inventory.
FAILURE POINT. Any surviving bypass, stale authority, omitted dependency, unmodeled timing path, or unsupported evidence link narrows or defeats the claim.
GOVERNANCE STATUS. M1 unless a chapter-specific case study states a higher internal implementation maturity; no institutional compliance determination is inferred.
CONFIDENCE TYPE. Definitional for the architecture, deductive only under stated assumptions, implementation-dependent for runtime and hardware, institutionally imported for authority, and evidence-dependent for deployment claims.

Chapter 35

Typed Transaction Identity, Commit Integrity, and Replay Resistance

Semantic transaction identity, immutability, single-use authorization, idempotency, epochs, and replay detection.

35.1 Objective

The objective of this chapter is to define typed transaction identity, commit integrity, and replay resistance as a bounded component of execution-bound governance. The analysis connects proposal-permission-commit separation, typed transaction identity, complete mediation, atomicity, durable state, evidence closure, and deployment correspondence. The decisive test is not whether the system can describe the relevant rule, but whether the rule changes the set of reachable consequence-bearing transitions. The runtime claim is bounded by the complete consequence-path inventory, capability model, trusted computing base, concurrency semantics, persistent state, and deployed configuration.

35.2 Structural Model

A compact representation used in this chapter is:

$$T_q = (tid, qid, oid, cid, pid, aid, \alpha, s, d, vid)$$

The structural object is represented as a typed relation rather than an informal checklist. Identity, object, context, authority, version, time, fault state, and evidence are separated because a change in any one of them can create a different transition. The architecture therefore rejects implicit substitution: a permission for one object is not a permission for another; a proof for one model is not a proof for a deployment; an approval before mutation is not approval after mutation. Unknown mandatory state is represented explicitly and does not default to ordinary allow.

The chapter’s principal structural commitments are:

Proposal-Permission-Commit Separation - retained as an explicit state or dependency.

Typed Transaction Identity - retained as an explicit state or dependency.

Complete Mediation - retained as an explicit state or dependency.

Atomicity - retained as an explicit state or dependency.

Durable State - retained as an explicit state or dependency.

Evidence Closure - retained as an explicit state or dependency.

Deployment Correspondence - retained as an explicit state or dependency.

35.3 Execution Reality

Execution reality begins at the last point from which the protected consequence can still be prevented. Upstream analysis, documentation, monitoring, or recommendation may improve decision quality, but it becomes governance only when the consequence path depends on the resulting state. The chapter therefore treats direct capability, alternate interfaces, administrative paths, recovery mechanisms, stale state, and external dependencies as part of the governed object. A system is not credited with a property merely because a dashboard displays it; the relevant gate, controller, institution, or physical interface must enforce or evidence it.

35.4 Failure Point and Counter-Construction

The claim collapses under any in-scope counter-construction such as:

an alternate API bypasses the gate

post-permit mutation changes the transaction

crash restores consumed authority

external outcome is guessed after timeout

These failures are not edge cases external to the theory. They are the primary falsification instruments. Where one of them remains reachable, the correct result is a narrowed claim, an unresolved state, or a failed invariant - not a narrative assertion that the architecture is nevertheless governed.

35.5 Verification and Evidence Obligations

inventory every consequence path

prove capability and identity binding

test concurrency, crash, retry, and recovery

bind evidence to the deployed artifact and configuration

Evidence must remain bound to the exact artifact, configuration, version, operating envelope, and decision state examined. A passing result establishes only the tested property and coverage. A failed or indeterminate result remains part of the authoritative record. Where independent reproduction has not occurred, the result is not represented as independently established.

35.6 Selected Propositions

P35.1. The validity of typed transaction identity, commit integrity, and replay resistance cannot exceed the scope of its weakest mandatory dependency.

P35.2. A descriptive or advisory representation does not become an execution constraint until it controls the relevant consequence path.

P35.3. Component-level validity does not compose automatically across interfaces, timing, authority, failure, or deployment change.

P35.4. Evidence of absence is not inferred from absence of evidence; unresolved mandatory state remains unresolved.

35.7 Governance Status and Confidence Type

STRUCTURE. Specified as part of the integrated federated architecture.
EXECUTION REALITY. Requires a domain-specific implementation and complete path inventory.
FAILURE POINT. Any surviving bypass, stale authority, omitted dependency, unmodeled timing path, or unsupported evidence link narrows or defeats the claim.
GOVERNANCE STATUS. M1 unless a chapter-specific case study states a higher internal implementation maturity; no institutional compliance determination is inferred.
CONFIDENCE TYPE. Definitional for the architecture, deductive only under stated assumptions, implementation-dependent for runtime and hardware, institutionally imported for authority, and evidence-dependent for deployment claims.

Chapter 36

Runtime Mediation, Capability Control, and Non-Bypassable Enforcement

Reference monitors, complete mediation, least privilege, capability attenuation, trusted paths, and fail-closed gates.

36.1 Objective

The objective of this chapter is to define runtime mediation, capability control, and non-bypassable enforcement as a bounded component of execution-bound governance. The analysis connects proposal-permission-commit separation, typed transaction identity, complete mediation, atomicity, durable state, evidence closure, and deployment correspondence. The decisive test is not whether the system can describe the relevant rule, but whether the rule changes the set of reachable consequence-bearing transitions. The runtime claim is bounded by the complete consequence-path inventory, capability model, trusted computing base, concurrency semantics, persistent state, and deployed configuration.

36.2 Structural Model

A compact representation used in this chapter is:

$$ExecPath \subseteq MediatedPath$$

The structural object is represented as a typed relation rather than an informal checklist. Identity, object, context, authority, version, time, fault state, and evidence are separated because a change in any one of them can create a different transition. The architecture therefore rejects implicit substitution: a permission for one object is not a permission for another; a proof for one model is not a proof for a deployment; an approval before mutation is not approval after mutation. Unknown mandatory state is represented explicitly and does not default to ordinary allow.

The chapter’s principal structural commitments are:

Proposal-Permission-Commit Separation - retained as an explicit state or dependency.

Typed Transaction Identity - retained as an explicit state or dependency.

Complete Mediation - retained as an explicit state or dependency.

Atomicity - retained as an explicit state or dependency.

Durable State - retained as an explicit state or dependency.

Evidence Closure - retained as an explicit state or dependency.

Deployment Correspondence - retained as an explicit state or dependency.

36.3 Execution Reality

Execution reality begins at the last point from which the protected consequence can still be prevented. Upstream analysis, documentation, monitoring, or recommendation may improve decision quality, but it becomes governance only when the consequence path depends on the resulting state. The chapter therefore treats direct capability, alternate interfaces, administrative paths, recovery mechanisms, stale state, and external dependencies as part of the governed object. A system is not credited with a property merely because a dashboard displays it; the relevant gate, controller, institution, or physical interface must enforce or evidence it.

36.4 Failure Point and Counter-Construction

The claim collapses under any in-scope counter-construction such as:

an alternate API bypasses the gate

post-permit mutation changes the transaction

crash restores consumed authority

external outcome is guessed after timeout

These failures are not edge cases external to the theory. They are the primary falsification instruments. Where one of them remains reachable, the correct result is a narrowed claim, an unresolved state, or a failed invariant - not a narrative assertion that the architecture is nevertheless governed.

36.5 Verification and Evidence Obligations

inventory every consequence path

prove capability and identity binding

test concurrency, crash, retry, and recovery

bind evidence to the deployed artifact and configuration

Evidence must remain bound to the exact artifact, configuration, version, operating envelope, and decision state examined. A passing result establishes only the tested property and coverage. A failed or indeterminate result remains part of the authoritative record. Where independent reproduction has not occurred, the result is not represented as independently established.

36.6 Selected Propositions

P36.1. The validity of runtime mediation, capability control, and non-bypassable enforcement cannot exceed the scope of its weakest mandatory dependency.

P36.2. A descriptive or advisory representation does not become an execution constraint until it controls the relevant consequence path.

P36.3. Component-level validity does not compose automatically across interfaces, timing, authority, failure, or deployment change.

P36.4. Evidence of absence is not inferred from absence of evidence; unresolved mandatory state remains unresolved.

36.7 Governance Status and Confidence Type

STRUCTURE. Specified as part of the integrated federated architecture.
EXECUTION REALITY. Requires a domain-specific implementation and complete path inventory.
FAILURE POINT. Any surviving bypass, stale authority, omitted dependency, unmodeled timing path, or unsupported evidence link narrows or defeats the claim.
GOVERNANCE STATUS. M1 unless a chapter-specific case study states a higher internal implementation maturity; no institutional compliance determination is inferred.
CONFIDENCE TYPE. Definitional for the architecture, deductive only under stated assumptions, implementation-dependent for runtime and hardware, institutionally imported for authority, and evidence-dependent for deployment claims.

Chapter 37

Concurrency, Atomicity, Isolation, and Race-Free Governance

Linearization, serializability, revocation races, multi-object invariants, and external-effect barriers.

37.1 Objective

The objective of this chapter is to define concurrency, atomicity, isolation, and race-free governance as a bounded component of execution-bound governance. The analysis connects proposal-permission-commit separation, typed transaction identity, complete mediation, atomicity, durable state, evidence closure, and deployment correspondence. The decisive test is not whether the system can describe the relevant rule, but whether the rule changes the set of reachable consequence-bearing transitions. The runtime claim is bounded by the complete consequence-path inventory, capability model, trusted computing base, concurrency semantics, persistent state, and deployed configuration.

37.2 Structural Model

A compact representation used in this chapter is:

$$Commit = linearize(validate, consume, stateChange)$$

The structural object is represented as a typed relation rather than an informal checklist. Identity, object, context, authority, version, time, fault state, and evidence are separated because a change in any one of them can create a different transition. The architecture therefore rejects implicit substitution: a permission for one object is not a permission for another; a proof for one model is not a proof for a deployment; an approval before mutation is not approval after mutation. Unknown mandatory state is represented explicitly and does not default to ordinary allow.

The chapter’s principal structural commitments are:

Proposal-Permission-Commit Separation - retained as an explicit state or dependency.

Typed Transaction Identity - retained as an explicit state or dependency.

Complete Mediation - retained as an explicit state or dependency.

Atomicity - retained as an explicit state or dependency.

Durable State - retained as an explicit state or dependency.

Evidence Closure - retained as an explicit state or dependency.

Deployment Correspondence - retained as an explicit state or dependency.

37.3 Execution Reality

Execution reality begins at the last point from which the protected consequence can still be prevented. Upstream analysis, documentation, monitoring, or recommendation may improve decision quality, but it becomes governance only when the consequence path depends on the resulting state. The chapter therefore treats direct capability, alternate interfaces, administrative paths, recovery mechanisms, stale state, and external dependencies as part of the governed object. A system is not credited with a property merely because a dashboard displays it; the relevant gate, controller, institution, or physical interface must enforce or evidence it.

37.4 Failure Point and Counter-Construction

The claim collapses under any in-scope counter-construction such as:

an alternate API bypasses the gate

post-permit mutation changes the transaction

crash restores consumed authority

external outcome is guessed after timeout

These failures are not edge cases external to the theory. They are the primary falsification instruments. Where one of them remains reachable, the correct result is a narrowed claim, an unresolved state, or a failed invariant - not a narrative assertion that the architecture is nevertheless governed.

37.5 Verification and Evidence Obligations

inventory every consequence path

prove capability and identity binding

test concurrency, crash, retry, and recovery

bind evidence to the deployed artifact and configuration

Evidence must remain bound to the exact artifact, configuration, version, operating envelope, and decision state examined. A passing result establishes only the tested property and coverage. A failed or indeterminate result remains part of the authoritative record. Where independent reproduction has not occurred, the result is not represented as independently established.

37.6 Selected Propositions

P37.1. The validity of concurrency, atomicity, isolation, and race-free governance cannot exceed the scope of its weakest mandatory dependency.

P37.2. A descriptive or advisory representation does not become an execution constraint until it controls the relevant consequence path.

P37.3. Component-level validity does not compose automatically across interfaces, timing, authority, failure, or deployment change.

P37.4. Evidence of absence is not inferred from absence of evidence; unresolved mandatory state remains unresolved.

37.7 Governance Status and Confidence Type

STRUCTURE. Specified as part of the integrated federated architecture.
EXECUTION REALITY. Requires a domain-specific implementation and complete path inventory.
FAILURE POINT. Any surviving bypass, stale authority, omitted dependency, unmodeled timing path, or unsupported evidence link narrows or defeats the claim.
GOVERNANCE STATUS. M1 unless a chapter-specific case study states a higher internal implementation maturity; no institutional compliance determination is inferred.
CONFIDENCE TYPE. Definitional for the architecture, deductive only under stated assumptions, implementation-dependent for runtime and hardware, institutionally imported for authority, and evidence-dependent for deployment claims.

Chapter 38

Persistence, Crash Consistency, Restart, and Idempotent Recovery

Durable authority state, write-ahead logging, in-doubt transactions, recovery restriction, and effectively-once consequence.

38.1 Objective

The objective of this chapter is to define persistence, crash consistency, restart, and idempotent recovery as a bounded component of execution-bound governance. The analysis connects proposal-permission-commit separation, typed transaction identity, complete mediation, atomicity, durable state, evidence closure, and deployment correspondence. The decisive test is not whether the system can describe the relevant rule, but whether the rule changes the set of reachable consequence-bearing transitions. The runtime claim is bounded by the complete consequence-path inventory, capability model, trusted computing base, concurrency semantics, persistent state, and deployed configuration.

38.2 Structural Model

A compact representation used in this chapter is:

$$RecoveredState \in Prefixes(GovernedHistory) \cup \{in\text{-}doubt\}$$

The structural object is represented as a typed relation rather than an informal checklist. Identity, object, context, authority, version, time, fault state, and evidence are separated because a change in any one of them can create a different transition. The architecture therefore rejects implicit substitution: a permission for one object is not a permission for another; a proof for one model is not a proof for a deployment; an approval before mutation is not approval after mutation. Unknown mandatory state is represented explicitly and does not default to ordinary allow.

The chapter’s principal structural commitments are:

Proposal-Permission-Commit Separation - retained as an explicit state or dependency.

Typed Transaction Identity - retained as an explicit state or dependency.

Complete Mediation - retained as an explicit state or dependency.

Atomicity - retained as an explicit state or dependency.

Durable State - retained as an explicit state or dependency.

Evidence Closure - retained as an explicit state or dependency.

Deployment Correspondence - retained as an explicit state or dependency.

38.3 Execution Reality

Execution reality begins at the last point from which the protected consequence can still be prevented. Upstream analysis, documentation, monitoring, or recommendation may improve decision quality, but it becomes governance only when the consequence path depends on the resulting state. The chapter therefore treats direct capability, alternate interfaces, administrative paths, recovery mechanisms, stale state, and external dependencies as part of the governed object. A system is not credited with a property merely because a dashboard displays it; the relevant gate, controller, institution, or physical interface must enforce or evidence it.

38.4 Failure Point and Counter-Construction

The claim collapses under any in-scope counter-construction such as:

an alternate API bypasses the gate

post-permit mutation changes the transaction

crash restores consumed authority

external outcome is guessed after timeout

These failures are not edge cases external to the theory. They are the primary falsification instruments. Where one of them remains reachable, the correct result is a narrowed claim, an unresolved state, or a failed invariant - not a narrative assertion that the architecture is nevertheless governed.

38.5 Verification and Evidence Obligations

inventory every consequence path

prove capability and identity binding

test concurrency, crash, retry, and recovery

bind evidence to the deployed artifact and configuration

Evidence must remain bound to the exact artifact, configuration, version, operating envelope, and decision state examined. A passing result establishes only the tested property and coverage. A failed or indeterminate result remains part of the authoritative record. Where independent reproduction has not occurred, the result is not represented as independently established.

38.6 Selected Propositions

P38.1. The validity of persistence, crash consistency, restart, and idempotent recovery cannot exceed the scope of its weakest mandatory dependency.

P38.2. A descriptive or advisory representation does not become an execution constraint until it controls the relevant consequence path.

P38.3. Component-level validity does not compose automatically across interfaces, timing, authority, failure, or deployment change.

P38.4. Evidence of absence is not inferred from absence of evidence; unresolved mandatory state remains unresolved.

38.7 Governance Status and Confidence Type

STRUCTURE. Specified as part of the integrated federated architecture.
EXECUTION REALITY. Requires a domain-specific implementation and complete path inventory.
FAILURE POINT. Any surviving bypass, stale authority, omitted dependency, unmodeled timing path, or unsupported evidence link narrows or defeats the claim.
GOVERNANCE STATUS. M1 unless a chapter-specific case study states a higher internal implementation maturity; no institutional compliance determination is inferred.
CONFIDENCE TYPE. Definitional for the architecture, deductive only under stated assumptions, implementation-dependent for runtime and hardware, institutionally imported for authority, and evidence-dependent for deployment claims.

Chapter 39

Distributed Execution, Messaging, and External-Effect Coordination

Outboxes, inboxes, ordering, sagas, compensation, external APIs, and exactly-once claim limits.

39.1 Objective

The objective of this chapter is to define distributed execution, messaging, and external-effect coordination as a bounded component of execution-bound governance. The analysis connects proposal-permission-commit separation, typed transaction identity, complete mediation, atomicity, durable state, evidence closure, and deployment correspondence. The decisive test is not whether the system can describe the relevant rule, but whether the rule changes the set of reachable consequence-bearing transitions. The runtime claim is bounded by the complete consequence-path inventory, capability model, trusted computing base, concurrency semantics, persistent state, and deployed configuration.

39.2 Structural Model

A compact representation used in this chapter is:

$$\text{local commit} \Rightarrow \not\supset \text{external exactly-once effect}$$

The structural object is represented as a typed relation rather than an informal checklist. Identity, object, context, authority, version, time, fault state, and evidence are separated because a change in any one of them can create a different transition. The architecture therefore rejects implicit substitution: a permission for one object is not a permission for another; a proof for one model is not a proof for a deployment; an approval before mutation is not approval after mutation. Unknown mandatory state is represented explicitly and does not default to ordinary allow.

The chapter’s principal structural commitments are:

Proposal-Permission-Commit Separation - retained as an explicit state or dependency.

Typed Transaction Identity - retained as an explicit state or dependency.

Complete Mediation - retained as an explicit state or dependency.

Atomicity - retained as an explicit state or dependency.

Durable State - retained as an explicit state or dependency.

Evidence Closure - retained as an explicit state or dependency.

Deployment Correspondence - retained as an explicit state or dependency.

39.3 Execution Reality

Execution reality begins at the last point from which the protected consequence can still be prevented. Upstream analysis, documentation, monitoring, or recommendation may improve decision quality, but it becomes governance only when the consequence path depends on the resulting state. The chapter therefore treats direct capability, alternate interfaces, administrative paths, recovery mechanisms, stale state, and external dependencies as part of the governed object. A system is not credited with a property merely because a dashboard displays it; the relevant gate, controller, institution, or physical interface must enforce or evidence it.

39.4 Failure Point and Counter-Construction

The claim collapses under any in-scope counter-construction such as:

an alternate API bypasses the gate

post-permit mutation changes the transaction

crash restores consumed authority

external outcome is guessed after timeout

These failures are not edge cases external to the theory. They are the primary falsification instruments. Where one of them remains reachable, the correct result is a narrowed claim, an unresolved state, or a failed invariant - not a narrative assertion that the architecture is nevertheless governed.

39.5 Verification and Evidence Obligations

inventory every consequence path

prove capability and identity binding

test concurrency, crash, retry, and recovery

bind evidence to the deployed artifact and configuration

Evidence must remain bound to the exact artifact, configuration, version, operating envelope, and decision state examined. A passing result establishes only the tested property and coverage. A failed or indeterminate result remains part of the authoritative record. Where independent reproduction has not occurred, the result is not represented as independently established.

39.6 Selected Propositions

P39.1. The validity of distributed execution, messaging, and external-effect coordination cannot exceed the scope of its weakest mandatory dependency.

P39.2. A descriptive or advisory representation does not become an execution constraint until it controls the relevant consequence path.

P39.3. Component-level validity does not compose automatically across interfaces, timing, authority, failure, or deployment change.

P39.4. Evidence of absence is not inferred from absence of evidence; unresolved mandatory state remains unresolved.

39.7 Governance Status and Confidence Type

STRUCTURE. Specified as part of the integrated federated architecture.
EXECUTION REALITY. Requires a domain-specific implementation and complete path inventory.
FAILURE POINT. Any surviving bypass, stale authority, omitted dependency, unmodeled timing path, or unsupported evidence link narrows or defeats the claim.
GOVERNANCE STATUS. M1 unless a chapter-specific case study states a higher internal implementation maturity; no institutional compliance determination is inferred.
CONFIDENCE TYPE. Definitional for the architecture, deductive only under stated assumptions, implementation-dependent for runtime and hardware, institutionally imported for authority, and evidence-dependent for deployment claims.

Chapter 40

Evidence Architecture, Provenance, and Tamper-Evident Reconstruction

Typed evidence, causal graphs, content-metadata separation, hash chains, signatures, forks, and evidence closure.

40.1 Objective

The objective of this chapter is to define evidence architecture, provenance, and tamper-evident reconstruction as a bounded component of execution-bound governance. The analysis connects proposal-permission-commit separation, typed transaction identity, complete mediation, atomicity, durable state, evidence closure, and deployment correspondence. The decisive test is not whether the system can describe the relevant rule, but whether the rule changes the set of reachable consequence-bearing transitions. The runtime claim is bounded by the complete consequence-path inventory, capability model, trusted computing base, concurrency semantics, persistent state, and deployed configuration.

40.2 Structural Model

A compact representation used in this chapter is:

$$h_i = H\left(h_{i-1} \| Canon\left(e_i\right)\right)$$

The structural object is represented as a typed relation rather than an informal checklist. Identity, object, context, authority, version, time, fault state, and evidence are separated because a change in any one of them can create a different transition. The architecture therefore rejects implicit substitution: a permission for one object is not a permission for another; a proof for one model is not a proof for a deployment; an approval before mutation is not approval after mutation. Unknown mandatory state is represented explicitly and does not default to ordinary allow.

The chapter’s principal structural commitments are:

Proposal-Permission-Commit Separation - retained as an explicit state or dependency.

Typed Transaction Identity - retained as an explicit state or dependency.

Complete Mediation - retained as an explicit state or dependency.

Atomicity - retained as an explicit state or dependency.

Durable State - retained as an explicit state or dependency.

Evidence Closure - retained as an explicit state or dependency.

Deployment Correspondence - retained as an explicit state or dependency.

40.3 Execution Reality

Execution reality begins at the last point from which the protected consequence can still be prevented. Upstream analysis, documentation, monitoring, or recommendation may improve decision quality, but it becomes governance only when the consequence path depends on the resulting state. The chapter therefore treats direct capability, alternate interfaces, administrative paths, recovery mechanisms, stale state, and external dependencies as part of the governed object. A system is not credited with a property merely because a dashboard displays it; the relevant gate, controller, institution, or physical interface must enforce or evidence it.

40.4 Failure Point and Counter-Construction

The claim collapses under any in-scope counter-construction such as:

an alternate API bypasses the gate

post-permit mutation changes the transaction

crash restores consumed authority

external outcome is guessed after timeout

These failures are not edge cases external to the theory. They are the primary falsification instruments. Where one of them remains reachable, the correct result is a narrowed claim, an unresolved state, or a failed invariant - not a narrative assertion that the architecture is nevertheless governed.

40.5 Verification and Evidence Obligations

inventory every consequence path

prove capability and identity binding

test concurrency, crash, retry, and recovery

bind evidence to the deployed artifact and configuration

Evidence must remain bound to the exact artifact, configuration, version, operating envelope, and decision state examined. A passing result establishes only the tested property and coverage. A failed or indeterminate result remains part of the authoritative record. Where independent reproduction has not occurred, the result is not represented as independently established.

40.6 Selected Propositions

P40.1. The validity of evidence architecture, provenance, and tamper-evident reconstruction cannot exceed the scope of its weakest mandatory dependency.

P40.2. A descriptive or advisory representation does not become an execution constraint until it controls the relevant consequence path.

P40.3. Component-level validity does not compose automatically across interfaces, timing, authority, failure, or deployment change.

P40.4. Evidence of absence is not inferred from absence of evidence; unresolved mandatory state remains unresolved.

40.7 Governance Status and Confidence Type

STRUCTURE. Specified as part of the integrated federated architecture.
EXECUTION REALITY. Requires a domain-specific implementation and complete path inventory.
FAILURE POINT. Any surviving bypass, stale authority, omitted dependency, unmodeled timing path, or unsupported evidence link narrows or defeats the claim.
GOVERNANCE STATUS. M1 unless a chapter-specific case study states a higher internal implementation maturity; no institutional compliance determination is inferred.
CONFIDENCE TYPE. Definitional for the architecture, deductive only under stated assumptions, implementation-dependent for runtime and hardware, institutionally imported for authority, and evidence-dependent for deployment claims.

Chapter 41

Isolation Boundaries, Trusted Computing Bases, and Privilege Separation

Sandboxes, privilege domains, secret isolation, independent enforcement, side channels, and claim suspension after breach.

41.1 Objective

The objective of this chapter is to define isolation boundaries, trusted computing bases, and privilege separation as a bounded component of execution-bound governance. The analysis connects proposal-permission-commit separation, typed transaction identity, complete mediation, atomicity, durable state, evidence closure, and deployment correspondence. The decisive test is not whether the system can describe the relevant rule, but whether the rule changes the set of reachable consequence-bearing transitions. The runtime claim is bounded by the complete consequence-path inventory, capability model, trusted computing base, concurrency semantics, persistent state, and deployed configuration.

41.2 Structural Model

The structural object is represented as a typed relation rather than an informal checklist. Identity, object, context, authority, version, time, fault state, and evidence are separated because a change in any one of them can create a different transition. The architecture therefore rejects implicit substitution: a permission for one object is not a permission for another; a proof for one model is not a proof for a deployment; an approval before mutation is not approval after mutation. Unknown mandatory state is represented explicitly and does not default to ordinary allow.

The chapter’s principal structural commitments are:

Proposal-Permission-Commit Separation - retained as an explicit state or dependency.

Typed Transaction Identity - retained as an explicit state or dependency.

Complete Mediation - retained as an explicit state or dependency.

Atomicity - retained as an explicit state or dependency.

Durable State - retained as an explicit state or dependency.

Evidence Closure - retained as an explicit state or dependency.

Deployment Correspondence - retained as an explicit state or dependency.

41.3 Execution Reality

Execution reality begins at the last point from which the protected consequence can still be prevented. Upstream analysis, documentation, monitoring, or recommendation may improve decision quality, but it becomes governance only when the consequence path depends on the resulting state. The chapter therefore treats direct capability, alternate interfaces, administrative paths, recovery mechanisms, stale state, and external dependencies as part of the governed object. A system is not credited with a property merely because a dashboard displays it; the relevant gate, controller, institution, or physical interface must enforce or evidence it.

41.4 Failure Point and Counter-Construction

The claim collapses under any in-scope counter-construction such as:

an alternate API bypasses the gate

post-permit mutation changes the transaction

crash restores consumed authority

external outcome is guessed after timeout

These failures are not edge cases external to the theory. They are the primary falsification instruments. Where one of them remains reachable, the correct result is a narrowed claim, an unresolved state, or a failed invariant - not a narrative assertion that the architecture is nevertheless governed.

41.5 Verification and Evidence Obligations

inventory every consequence path

prove capability and identity binding

test concurrency, crash, retry, and recovery

bind evidence to the deployed artifact and configuration

Evidence must remain bound to the exact artifact, configuration, version, operating envelope, and decision state examined. A passing result establishes only the tested property and coverage. A failed or indeterminate result remains part of the authoritative record. Where independent reproduction has not occurred, the result is not represented as independently established.

41.6 Selected Propositions

P41.1. The validity of isolation boundaries, trusted computing bases, and privilege separation cannot exceed the scope of its weakest mandatory dependency.

P41.2. A descriptive or advisory representation does not become an execution constraint until it controls the relevant consequence path.

P41.3. Component-level validity does not compose automatically across interfaces, timing, authority, failure, or deployment change.

P41.4. Evidence of absence is not inferred from absence of evidence; unresolved mandatory state remains unresolved.

41.7 Governance Status and Confidence Type

STRUCTURE. Specified as part of the integrated federated architecture.
EXECUTION REALITY. Requires a domain-specific implementation and complete path inventory.
FAILURE POINT. Any surviving bypass, stale authority, omitted dependency, unmodeled timing path, or unsupported evidence link narrows or defeats the claim.
GOVERNANCE STATUS. M1 unless a chapter-specific case study states a higher internal implementation maturity; no institutional compliance determination is inferred.
CONFIDENCE TYPE. Definitional for the architecture, deductive only under stated assumptions, implementation-dependent for runtime and hardware, institutionally imported for authority, and evidence-dependent for deployment claims.

Chapter 42

Versioning, Configuration, Software Supply Chain, and Deployment Integrity

Artifact identity, reproducible builds, feature flags, attestation, drift, rollout, rollback, and migration.

42.1 Objective

The objective of this chapter is to define versioning, configuration, software supply chain, and deployment integrity as a bounded component of execution-bound governance. The analysis connects proposal-permission-commit separation, typed transaction identity, complete mediation, atomicity, durable state, evidence closure, and deployment correspondence. The decisive test is not whether the system can describe the relevant rule, but whether the rule changes the set of reachable consequence-bearing transitions. The runtime claim is bounded by the complete consequence-path inventory, capability model, trusted computing base, concurrency semantics, persistent state, and deployed configuration.

42.2 Structural Model

A compact representation used in this chapter is:

$$A_D = (source, build, dependencies, configuration, model, policy, hash, signature)$$

The structural object is represented as a typed relation rather than an informal checklist. Identity, object, context, authority, version, time, fault state, and evidence are separated because a change in any one of them can create a different transition. The architecture therefore rejects implicit substitution: a permission for one object is not a permission for another; a proof for one model is not a proof for a deployment; an approval before mutation is not approval after mutation. Unknown mandatory state is represented explicitly and does not default to ordinary allow.

The chapter’s principal structural commitments are:

Proposal-Permission-Commit Separation - retained as an explicit state or dependency.

Typed Transaction Identity - retained as an explicit state or dependency.

Complete Mediation - retained as an explicit state or dependency.

Atomicity - retained as an explicit state or dependency.

Durable State - retained as an explicit state or dependency.

Evidence Closure - retained as an explicit state or dependency.

Deployment Correspondence - retained as an explicit state or dependency.

42.3 Execution Reality

Execution reality begins at the last point from which the protected consequence can still be prevented. Upstream analysis, documentation, monitoring, or recommendation may improve decision quality, but it becomes governance only when the consequence path depends on the resulting state. The chapter therefore treats direct capability, alternate interfaces, administrative paths, recovery mechanisms, stale state, and external dependencies as part of the governed object. A system is not credited with a property merely because a dashboard displays it; the relevant gate, controller, institution, or physical interface must enforce or evidence it.

42.4 Failure Point and Counter-Construction

The claim collapses under any in-scope counter-construction such as:

an alternate API bypasses the gate

post-permit mutation changes the transaction

crash restores consumed authority

external outcome is guessed after timeout

These failures are not edge cases external to the theory. They are the primary falsification instruments. Where one of them remains reachable, the correct result is a narrowed claim, an unresolved state, or a failed invariant - not a narrative assertion that the architecture is nevertheless governed.

42.5 Verification and Evidence Obligations

inventory every consequence path

prove capability and identity binding

test concurrency, crash, retry, and recovery

bind evidence to the deployed artifact and configuration

Evidence must remain bound to the exact artifact, configuration, version, operating envelope, and decision state examined. A passing result establishes only the tested property and coverage. A failed or indeterminate result remains part of the authoritative record. Where independent reproduction has not occurred, the result is not represented as independently established.

42.6 Selected Propositions

P42.1. The validity of versioning, configuration, software supply chain, and deployment integrity cannot exceed the scope of its weakest mandatory dependency.

P42.2. A descriptive or advisory representation does not become an execution constraint until it controls the relevant consequence path.

P42.3. Component-level validity does not compose automatically across interfaces, timing, authority, failure, or deployment change.

P42.4. Evidence of absence is not inferred from absence of evidence; unresolved mandatory state remains unresolved.

42.7 Governance Status and Confidence Type

STRUCTURE. Specified as part of the integrated federated architecture.
EXECUTION REALITY. Requires a domain-specific implementation and complete path inventory.
FAILURE POINT. Any surviving bypass, stale authority, omitted dependency, unmodeled timing path, or unsupported evidence link narrows or defeats the claim.
GOVERNANCE STATUS. M1 unless a chapter-specific case study states a higher internal implementation maturity; no institutional compliance determination is inferred.
CONFIDENCE TYPE. Definitional for the architecture, deductive only under stated assumptions, implementation-dependent for runtime and hardware, institutionally imported for authority, and evidence-dependent for deployment claims.

Chapter 43

Agentic Execution, Tool Use, Delegation, and Sandboxed Runtime Authority

Plans, tool calls, recursive delegation, instruction-data separation, memory freshness, and multi-agent conflict.

43.1 Objective

The objective of this chapter is to define agentic execution, tool use, delegation, and sandboxed runtime authority as a bounded component of execution-bound governance. The analysis connects proposal-permission-commit separation, typed transaction identity, complete mediation, atomicity, durable state, evidence closure, and deployment correspondence. The decisive test is not whether the system can describe the relevant rule, but whether the rule changes the set of reachable consequence-bearing transitions. The runtime claim is bounded by the complete consequence-path inventory, capability model, trusted computing base, concurrency semantics, persistent state, and deployed configuration.

43.2 Structural Model

The structural object is represented as a typed relation rather than an informal checklist. Identity, object, context, authority, version, time, fault state, and evidence are separated because a change in any one of them can create a different transition. The architecture therefore rejects implicit substitution: a permission for one object is not a permission for another; a proof for one model is not a proof for a deployment; an approval before mutation is not approval after mutation. Unknown mandatory state is represented explicitly and does not default to ordinary allow.

The chapter’s principal structural commitments are:

Proposal-Permission-Commit Separation - retained as an explicit state or dependency.

Typed Transaction Identity - retained as an explicit state or dependency.

Complete Mediation - retained as an explicit state or dependency.

Atomicity - retained as an explicit state or dependency.

Durable State - retained as an explicit state or dependency.

Evidence Closure - retained as an explicit state or dependency.

Deployment Correspondence - retained as an explicit state or dependency.

43.3 Execution Reality

Execution reality begins at the last point from which the protected consequence can still be prevented. Upstream analysis, documentation, monitoring, or recommendation may improve decision quality, but it becomes governance only when the consequence path depends on the resulting state. The chapter therefore treats direct capability, alternate interfaces, administrative paths, recovery mechanisms, stale state, and external dependencies as part of the governed object. A system is not credited with a property merely because a dashboard displays it; the relevant gate, controller, institution, or physical interface must enforce or evidence it.

43.4 Failure Point and Counter-Construction

The claim collapses under any in-scope counter-construction such as:

an alternate API bypasses the gate

post-permit mutation changes the transaction

crash restores consumed authority

external outcome is guessed after timeout

These failures are not edge cases external to the theory. They are the primary falsification instruments. Where one of them remains reachable, the correct result is a narrowed claim, an unresolved state, or a failed invariant - not a narrative assertion that the architecture is nevertheless governed.

43.5 Verification and Evidence Obligations

inventory every consequence path

prove capability and identity binding

test concurrency, crash, retry, and recovery

bind evidence to the deployed artifact and configuration

Evidence must remain bound to the exact artifact, configuration, version, operating envelope, and decision state examined. A passing result establishes only the tested property and coverage. A failed or indeterminate result remains part of the authoritative record. Where independent reproduction has not occurred, the result is not represented as independently established.

43.6 Selected Propositions

P43.1. The validity of agentic execution, tool use, delegation, and sandboxed runtime authority cannot exceed the scope of its weakest mandatory dependency.

P43.2. A descriptive or advisory representation does not become an execution constraint until it controls the relevant consequence path.

P43.3. Component-level validity does not compose automatically across interfaces, timing, authority, failure, or deployment change.

P43.4. Evidence of absence is not inferred from absence of evidence; unresolved mandatory state remains unresolved.

43.7 Governance Status and Confidence Type

STRUCTURE. Specified as part of the integrated federated architecture.
EXECUTION REALITY. Requires a domain-specific implementation and complete path inventory.
FAILURE POINT. Any surviving bypass, stale authority, omitted dependency, unmodeled timing path, or unsupported evidence link narrows or defeats the claim.
GOVERNANCE STATUS. M1 unless a chapter-specific case study states a higher internal implementation maturity; no institutional compliance determination is inferred.
CONFIDENCE TYPE. Definitional for the architecture, deductive only under stated assumptions, implementation-dependent for runtime and hardware, institutionally imported for authority, and evidence-dependent for deployment claims.

Chapter 44

Systems-Volume Synthesis and Closure

The unified governed execution system, master runtime permit, continuity, capability closure, evidence closure, and export to hardware.

44.1 Objective

The objective of this chapter is to define systems-volume synthesis and closure as a bounded component of execution-bound governance. The analysis connects proposal-permission-commit separation, typed transaction identity, complete mediation, atomicity, durable state, evidence closure, and deployment correspondence. The decisive test is not whether the system can describe the relevant rule, but whether the rule changes the set of reachable consequence-bearing transitions. The runtime claim is bounded by the complete consequence-path inventory, capability model, trusted computing base, concurrency semantics, persistent state, and deployed configuration.

44.2 Structural Model

A compact representation used in this chapter is:

$$Commit \Rightarrow Permit_R$$

The structural object is represented as a typed relation rather than an informal checklist. Identity, object, context, authority, version, time, fault state, and evidence are separated because a change in any one of them can create a different transition. The architecture therefore rejects implicit substitution: a permission for one object is not a permission for another; a proof for one model is not a proof for a deployment; an approval before mutation is not approval after mutation. Unknown mandatory state is represented explicitly and does not default to ordinary allow.

The chapter’s principal structural commitments are:

Proposal-Permission-Commit Separation - retained as an explicit state or dependency.

Typed Transaction Identity - retained as an explicit state or dependency.

Complete Mediation - retained as an explicit state or dependency.

Atomicity - retained as an explicit state or dependency.

Durable State - retained as an explicit state or dependency.

Evidence Closure - retained as an explicit state or dependency.

Deployment Correspondence - retained as an explicit state or dependency.

44.3 Execution Reality

Execution reality begins at the last point from which the protected consequence can still be prevented. Upstream analysis, documentation, monitoring, or recommendation may improve decision quality, but it becomes governance only when the consequence path depends on the resulting state. The chapter therefore treats direct capability, alternate interfaces, administrative paths, recovery mechanisms, stale state, and external dependencies as part of the governed object. A system is not credited with a property merely because a dashboard displays it; the relevant gate, controller, institution, or physical interface must enforce or evidence it.

44.4 Failure Point and Counter-Construction

The claim collapses under any in-scope counter-construction such as:

an alternate API bypasses the gate

post-permit mutation changes the transaction

crash restores consumed authority

external outcome is guessed after timeout

These failures are not edge cases external to the theory. They are the primary falsification instruments. Where one of them remains reachable, the correct result is a narrowed claim, an unresolved state, or a failed invariant - not a narrative assertion that the architecture is nevertheless governed.

44.5 Verification and Evidence Obligations

inventory every consequence path

prove capability and identity binding

test concurrency, crash, retry, and recovery

bind evidence to the deployed artifact and configuration

Evidence must remain bound to the exact artifact, configuration, version, operating envelope, and decision state examined. A passing result establishes only the tested property and coverage. A failed or indeterminate result remains part of the authoritative record. Where independent reproduction has not occurred, the result is not represented as independently established.

44.6 Selected Propositions

P44.1. The validity of systems-volume synthesis and closure cannot exceed the scope of its weakest mandatory dependency.

P44.2. A descriptive or advisory representation does not become an execution constraint until it controls the relevant consequence path.

P44.3. Component-level validity does not compose automatically across interfaces, timing, authority, failure, or deployment change.

P44.4. Evidence of absence is not inferred from absence of evidence; unresolved mandatory state remains unresolved.

44.7 Governance Status and Confidence Type

STRUCTURE. Specified as part of the integrated federated architecture.
EXECUTION REALITY. Requires a domain-specific implementation and complete path inventory.
FAILURE POINT. Any surviving bypass, stale authority, omitted dependency, unmodeled timing path, or unsupported evidence link narrows or defeats the claim.
GOVERNANCE STATUS. M1 unless a chapter-specific case study states a higher internal implementation maturity; no institutional compliance determination is inferred.
CONFIDENCE TYPE. Definitional for the architecture, deductive only under stated assumptions, implementation-dependent for runtime and hardware, institutionally imported for authority, and evidence-dependent for deployment claims.

PART V - HARDWARE ENFORCEMENT, EMBEDDED SYSTEMS, AND AUTOMOTIVE EXECUTION

Part V moves below software. It specifies the physical and embedded conditions under which a deny or halt state removes the electrical path to consequence rather than merely recording an intention to stop.

Part-level claim boundary. The material in this part is specified at M1 unless a bounded case study explicitly records internal implementation evidence. No part-level statement independently widens the system-level claim.

Chapter 45

Hardware Enforcement Boundaries and the Non-Maskable Halt

Physical signal paths, independent gate logic, fail-safe initialization, latched halt, reset authority, and physical cut sets.

45.1 Objective

The objective of this chapter is to define hardware enforcement boundaries and the non-maskable halt as a bounded component of execution-bound governance. The analysis connects physical consequence path, non-maskable halt, clock and reset integrity, persistent replay protection, I/O validity, platform health, and physical evidence. The decisive test is not whether the system can describe the relevant rule, but whether the rule changes the set of reachable consequence-bearing transitions. The hardware claim is bounded by electrical topology, clock and power envelopes, reset semantics, physical bypasses, I/O integrity, environmental conditions, and the tested artifact.

45.2 Structural Model

A compact representation used in this chapter is:

$$HALT_N = 0 \Rightarrow CE = 0$$

The structural object is represented as a typed relation rather than an informal checklist. Identity, object, context, authority, version, time, fault state, and evidence are separated because a change in any one of them can create a different transition. The architecture therefore rejects implicit substitution: a permission for one object is not a permission for another; a proof for one model is not a proof for a deployment; an approval before mutation is not approval after mutation. Unknown mandatory state is represented explicitly and does not default to ordinary allow.

The chapter’s principal structural commitments are:

Physical Consequence Path - retained as an explicit state or dependency.

Non-Maskable Halt - retained as an explicit state or dependency.

Clock And Reset Integrity - retained as an explicit state or dependency.

Persistent Replay Protection - retained as an explicit state or dependency.

I/O Validity - retained as an explicit state or dependency.

Platform Health - retained as an explicit state or dependency.

Physical Evidence - retained as an explicit state or dependency.

45.3 Execution Reality

Execution reality begins at the last point from which the protected consequence can still be prevented. Upstream analysis, documentation, monitoring, or recommendation may improve decision quality, but it becomes governance only when the consequence path depends on the resulting state. The chapter therefore treats direct capability, alternate interfaces, administrative paths, recovery mechanisms, stale state, and external dependencies as part of the governed object. A system is not credited with a property merely because a dashboard displays it; the relevant gate, controller, institution, or physical interface must enforce or evidence it.

45.4 Failure Point and Counter-Construction

The claim collapses under any in-scope counter-construction such as:

a physical bypass remains

halt is maskable or reset-cleared

clock or brownout freezes an unsafe output

command acknowledgement is mistaken for physical effect

These failures are not edge cases external to the theory. They are the primary falsification instruments. Where one of them remains reachable, the correct result is a narrowed claim, an unresolved state, or a failed invariant - not a narrative assertion that the architecture is nevertheless governed.

45.5 Verification and Evidence Obligations

verify physical cut-set closure

measure halt and safe-state timing

test power, clock, reset, I/O, network, and debug faults

preserve hardware evidence across reset

Evidence must remain bound to the exact artifact, configuration, version, operating envelope, and decision state examined. A passing result establishes only the tested property and coverage. A failed or indeterminate result remains part of the authoritative record. Where independent reproduction has not occurred, the result is not represented as independently established.

45.6 Selected Propositions

P45.1. The validity of hardware enforcement boundaries and the non-maskable halt cannot exceed the scope of its weakest mandatory dependency.

P45.2. A descriptive or advisory representation does not become an execution constraint until it controls the relevant consequence path.

P45.3. Component-level validity does not compose automatically across interfaces, timing, authority, failure, or deployment change.

P45.4. Evidence of absence is not inferred from absence of evidence; unresolved mandatory state remains unresolved.

45.7 Governance Status and Confidence Type

STRUCTURE. Specified as part of the integrated federated architecture.
EXECUTION REALITY. Requires a domain-specific implementation and complete path inventory.
FAILURE POINT. Any surviving bypass, stale authority, omitted dependency, unmodeled timing path, or unsupported evidence link narrows or defeats the claim.
GOVERNANCE STATUS. M1 unless a chapter-specific case study states a higher internal implementation maturity; no institutional compliance determination is inferred.
CONFIDENCE TYPE. Definitional for the architecture, deductive only under stated assumptions, implementation-dependent for runtime and hardware, institutionally imported for authority, and evidence-dependent for deployment claims.

Chapter 46

Synchronous Logic, State Machines, Timing Closure, and Metastability

Clock domains, CDC, setup and hold, glitch-free enable, illegal-state recovery, and deterministic hardware transition.

46.1 Objective

The objective of this chapter is to define synchronous logic, state machines, timing closure, and metastability as a bounded component of execution-bound governance. The analysis connects physical consequence path, non-maskable halt, clock and reset integrity, persistent replay protection, I/O validity, platform health, and physical evidence. The decisive test is not whether the system can describe the relevant rule, but whether the rule changes the set of reachable consequence-bearing transitions. The hardware claim is bounded by electrical topology, clock and power envelopes, reset semantics, physical bypasses, I/O integrity, environmental conditions, and the tested artifact.

46.2 Structural Model

A compact representation used in this chapter is:

$$q_{k+1} = \delta(q_k, i_k)$$

The structural object is represented as a typed relation rather than an informal checklist. Identity, object, context, authority, version, time, fault state, and evidence are separated because a change in any one of them can create a different transition. The architecture therefore rejects implicit substitution: a permission for one object is not a permission for another; a proof for one model is not a proof for a deployment; an approval before mutation is not approval after mutation. Unknown mandatory state is represented explicitly and does not default to ordinary allow.

The chapter’s principal structural commitments are:

Physical Consequence Path - retained as an explicit state or dependency.

Non-Maskable Halt - retained as an explicit state or dependency.

Clock And Reset Integrity - retained as an explicit state or dependency.

Persistent Replay Protection - retained as an explicit state or dependency.

I/O Validity - retained as an explicit state or dependency.

Platform Health - retained as an explicit state or dependency.

Physical Evidence - retained as an explicit state or dependency.

46.3 Execution Reality

Execution reality begins at the last point from which the protected consequence can still be prevented. Upstream analysis, documentation, monitoring, or recommendation may improve decision quality, but it becomes governance only when the consequence path depends on the resulting state. The chapter therefore treats direct capability, alternate interfaces, administrative paths, recovery mechanisms, stale state, and external dependencies as part of the governed object. A system is not credited with a property merely because a dashboard displays it; the relevant gate, controller, institution, or physical interface must enforce or evidence it.

46.4 Failure Point and Counter-Construction

The claim collapses under any in-scope counter-construction such as:

a physical bypass remains

halt is maskable or reset-cleared

clock or brownout freezes an unsafe output

command acknowledgement is mistaken for physical effect

These failures are not edge cases external to the theory. They are the primary falsification instruments. Where one of them remains reachable, the correct result is a narrowed claim, an unresolved state, or a failed invariant - not a narrative assertion that the architecture is nevertheless governed.

46.5 Verification and Evidence Obligations

verify physical cut-set closure

measure halt and safe-state timing

test power, clock, reset, I/O, network, and debug faults

preserve hardware evidence across reset

Evidence must remain bound to the exact artifact, configuration, version, operating envelope, and decision state examined. A passing result establishes only the tested property and coverage. A failed or indeterminate result remains part of the authoritative record. Where independent reproduction has not occurred, the result is not represented as independently established.

46.6 Selected Propositions

P46.1. The validity of synchronous logic, state machines, timing closure, and metastability cannot exceed the scope of its weakest mandatory dependency.

P46.2. A descriptive or advisory representation does not become an execution constraint until it controls the relevant consequence path.

P46.3. Component-level validity does not compose automatically across interfaces, timing, authority, failure, or deployment change.

P46.4. Evidence of absence is not inferred from absence of evidence; unresolved mandatory state remains unresolved.

46.7 Governance Status and Confidence Type

STRUCTURE. Specified as part of the integrated federated architecture.
EXECUTION REALITY. Requires a domain-specific implementation and complete path inventory.
FAILURE POINT. Any surviving bypass, stale authority, omitted dependency, unmodeled timing path, or unsupported evidence link narrows or defeats the claim.
GOVERNANCE STATUS. M1 unless a chapter-specific case study states a higher internal implementation maturity; no institutional compliance determination is inferred.
CONFIDENCE TYPE. Definitional for the architecture, deductive only under stated assumptions, implementation-dependent for runtime and hardware, institutionally imported for authority, and evidence-dependent for deployment claims.

Chapter 47

Secure Boot, Configuration Integrity, Anti-Rollback, and Hardware Roots of Trust

Boot chains, firmware and bitstream identity, lifecycle state, debug authority, key domains, and attestation limits.

47.1 Objective

The objective of this chapter is to define secure boot, configuration integrity, anti-rollback, and hardware roots of trust as a bounded component of execution-bound governance. The analysis connects physical consequence path, non-maskable halt, clock and reset integrity, persistent replay protection, I/O validity, platform health, and physical evidence. The decisive test is not whether the system can describe the relevant rule, but whether the rule changes the set of reachable consequence-bearing transitions. The hardware claim is bounded by electrical topology, clock and power envelopes, reset semantics, physical bypasses, I/O integrity, environmental conditions, and the tested artifact.

47.2 Structural Model

A compact representation used in this chapter is:

$$B_0 \rightarrow B_1 \rightarrow \cdots \rightarrow B_n$$

The structural object is represented as a typed relation rather than an informal checklist. Identity, object, context, authority, version, time, fault state, and evidence are separated because a change in any one of them can create a different transition. The architecture therefore rejects implicit substitution: a permission for one object is not a permission for another; a proof for one model is not a proof for a deployment; an approval before mutation is not approval after mutation. Unknown mandatory state is represented explicitly and does not default to ordinary allow.

The chapter’s principal structural commitments are:

Physical Consequence Path - retained as an explicit state or dependency.

Non-Maskable Halt - retained as an explicit state or dependency.

Clock And Reset Integrity - retained as an explicit state or dependency.

Persistent Replay Protection - retained as an explicit state or dependency.

I/O Validity - retained as an explicit state or dependency.

Platform Health - retained as an explicit state or dependency.

Physical Evidence - retained as an explicit state or dependency.

47.3 Execution Reality

Execution reality begins at the last point from which the protected consequence can still be prevented. Upstream analysis, documentation, monitoring, or recommendation may improve decision quality, but it becomes governance only when the consequence path depends on the resulting state. The chapter therefore treats direct capability, alternate interfaces, administrative paths, recovery mechanisms, stale state, and external dependencies as part of the governed object. A system is not credited with a property merely because a dashboard displays it; the relevant gate, controller, institution, or physical interface must enforce or evidence it.

47.4 Failure Point and Counter-Construction

The claim collapses under any in-scope counter-construction such as:

a physical bypass remains

halt is maskable or reset-cleared

clock or brownout freezes an unsafe output

command acknowledgement is mistaken for physical effect

These failures are not edge cases external to the theory. They are the primary falsification instruments. Where one of them remains reachable, the correct result is a narrowed claim, an unresolved state, or a failed invariant - not a narrative assertion that the architecture is nevertheless governed.

47.5 Verification and Evidence Obligations

verify physical cut-set closure

measure halt and safe-state timing

test power, clock, reset, I/O, network, and debug faults

preserve hardware evidence across reset

Evidence must remain bound to the exact artifact, configuration, version, operating envelope, and decision state examined. A passing result establishes only the tested property and coverage. A failed or indeterminate result remains part of the authoritative record. Where independent reproduction has not occurred, the result is not represented as independently established.

47.6 Selected Propositions

P47.1. The validity of secure boot, configuration integrity, anti-rollback, and hardware roots of trust cannot exceed the scope of its weakest mandatory dependency.

P47.2. A descriptive or advisory representation does not become an execution constraint until it controls the relevant consequence path.

P47.3. Component-level validity does not compose automatically across interfaces, timing, authority, failure, or deployment change.

P47.4. Evidence of absence is not inferred from absence of evidence; unresolved mandatory state remains unresolved.

47.7 Governance Status and Confidence Type

STRUCTURE. Specified as part of the integrated federated architecture.
EXECUTION REALITY. Requires a domain-specific implementation and complete path inventory.
FAILURE POINT. Any surviving bypass, stale authority, omitted dependency, unmodeled timing path, or unsupported evidence link narrows or defeats the claim.
GOVERNANCE STATUS. M1 unless a chapter-specific case study states a higher internal implementation maturity; no institutional compliance determination is inferred.
CONFIDENCE TYPE. Definitional for the architecture, deductive only under stated assumptions, implementation-dependent for runtime and hardware, institutionally imported for authority, and evidence-dependent for deployment claims.

Chapter 48

Hardware Transaction Identity, Monotonic State, and Tamper-Evident Audit Logging

Persistent counters, power-fail phase markers, bounded audit storage, replay windows, forks, and readout authority.

48.1 Objective

The objective of this chapter is to define hardware transaction identity, monotonic state, and tamper-evident audit logging as a bounded component of execution-bound governance. The analysis connects physical consequence path, non-maskable halt, clock and reset integrity, persistent replay protection, I/O validity, platform health, and physical evidence. The decisive test is not whether the system can describe the relevant rule, but whether the rule changes the set of reachable consequence-bearing transitions. The hardware claim is bounded by electrical topology, clock and power envelopes, reset semantics, physical bypasses, I/O integrity, environmental conditions, and the tested artifact.

48.2 Structural Model

A compact representation used in this chapter is:

$$c_{new} > c_{max} \Rightarrow c_{max} \leftarrow c_{new}$$

The structural object is represented as a typed relation rather than an informal checklist. Identity, object, context, authority, version, time, fault state, and evidence are separated because a change in any one of them can create a different transition. The architecture therefore rejects implicit substitution: a permission for one object is not a permission for another; a proof for one model is not a proof for a deployment; an approval before mutation is not approval after mutation. Unknown mandatory state is represented explicitly and does not default to ordinary allow.

The chapter’s principal structural commitments are:

Physical Consequence Path - retained as an explicit state or dependency.

Non-Maskable Halt - retained as an explicit state or dependency.

Clock And Reset Integrity - retained as an explicit state or dependency.

Persistent Replay Protection - retained as an explicit state or dependency.

I/O Validity - retained as an explicit state or dependency.

Platform Health - retained as an explicit state or dependency.

Physical Evidence - retained as an explicit state or dependency.

48.3 Execution Reality

Execution reality begins at the last point from which the protected consequence can still be prevented. Upstream analysis, documentation, monitoring, or recommendation may improve decision quality, but it becomes governance only when the consequence path depends on the resulting state. The chapter therefore treats direct capability, alternate interfaces, administrative paths, recovery mechanisms, stale state, and external dependencies as part of the governed object. A system is not credited with a property merely because a dashboard displays it; the relevant gate, controller, institution, or physical interface must enforce or evidence it.

48.4 Failure Point and Counter-Construction

The claim collapses under any in-scope counter-construction such as:

a physical bypass remains

halt is maskable or reset-cleared

clock or brownout freezes an unsafe output

command acknowledgement is mistaken for physical effect

These failures are not edge cases external to the theory. They are the primary falsification instruments. Where one of them remains reachable, the correct result is a narrowed claim, an unresolved state, or a failed invariant - not a narrative assertion that the architecture is nevertheless governed.

48.5 Verification and Evidence Obligations

verify physical cut-set closure

measure halt and safe-state timing

test power, clock, reset, I/O, network, and debug faults

preserve hardware evidence across reset

Evidence must remain bound to the exact artifact, configuration, version, operating envelope, and decision state examined. A passing result establishes only the tested property and coverage. A failed or indeterminate result remains part of the authoritative record. Where independent reproduction has not occurred, the result is not represented as independently established.

48.6 Selected Propositions

P48.1. The validity of hardware transaction identity, monotonic state, and tamper-evident audit logging cannot exceed the scope of its weakest mandatory dependency.

P48.2. A descriptive or advisory representation does not become an execution constraint until it controls the relevant consequence path.

P48.3. Component-level validity does not compose automatically across interfaces, timing, authority, failure, or deployment change.

P48.4. Evidence of absence is not inferred from absence of evidence; unresolved mandatory state remains unresolved.

48.7 Governance Status and Confidence Type

STRUCTURE. Specified as part of the integrated federated architecture.
EXECUTION REALITY. Requires a domain-specific implementation and complete path inventory.
FAILURE POINT. Any surviving bypass, stale authority, omitted dependency, unmodeled timing path, or unsupported evidence link narrows or defeats the claim.
GOVERNANCE STATUS. M1 unless a chapter-specific case study states a higher internal implementation maturity; no institutional compliance determination is inferred.
CONFIDENCE TYPE. Definitional for the architecture, deductive only under stated assumptions, implementation-dependent for runtime and hardware, institutionally imported for authority, and evidence-dependent for deployment claims.

Chapter 49

Sensor, Actuator, and I/O Boundary Integrity

Signal validity, freshness, line faults, interlocks, actuator feedback, command-effect discrepancy, and arbitration.

49.1 Objective

The objective of this chapter is to define sensor, actuator, and i/o boundary integrity as a bounded component of execution-bound governance. The analysis connects physical consequence path, non-maskable halt, clock and reset integrity, persistent replay protection, I/O validity, platform health, and physical evidence. The decisive test is not whether the system can describe the relevant rule, but whether the rule changes the set of reachable consequence-bearing transitions. The hardware claim is bounded by electrical topology, clock and power envelopes, reset semantics, physical bypasses, I/O integrity, environmental conditions, and the tested artifact.

49.2 Structural Model

A compact representation used in this chapter is:

$$AE = P_H \wedge HALT_N \wedge \bigwedge_j I_j$$

The structural object is represented as a typed relation rather than an informal checklist. Identity, object, context, authority, version, time, fault state, and evidence are separated because a change in any one of them can create a different transition. The architecture therefore rejects implicit substitution: a permission for one object is not a permission for another; a proof for one model is not a proof for a deployment; an approval before mutation is not approval after mutation. Unknown mandatory state is represented explicitly and does not default to ordinary allow.

The chapter’s principal structural commitments are:

Physical Consequence Path - retained as an explicit state or dependency.

Non-Maskable Halt - retained as an explicit state or dependency.

Clock And Reset Integrity - retained as an explicit state or dependency.

Persistent Replay Protection - retained as an explicit state or dependency.

I/O Validity - retained as an explicit state or dependency.

Platform Health - retained as an explicit state or dependency.

Physical Evidence - retained as an explicit state or dependency.

49.3 Execution Reality

Execution reality begins at the last point from which the protected consequence can still be prevented. Upstream analysis, documentation, monitoring, or recommendation may improve decision quality, but it becomes governance only when the consequence path depends on the resulting state. The chapter therefore treats direct capability, alternate interfaces, administrative paths, recovery mechanisms, stale state, and external dependencies as part of the governed object. A system is not credited with a property merely because a dashboard displays it; the relevant gate, controller, institution, or physical interface must enforce or evidence it.

49.4 Failure Point and Counter-Construction

The claim collapses under any in-scope counter-construction such as:

a physical bypass remains

halt is maskable or reset-cleared

clock or brownout freezes an unsafe output

command acknowledgement is mistaken for physical effect

These failures are not edge cases external to the theory. They are the primary falsification instruments. Where one of them remains reachable, the correct result is a narrowed claim, an unresolved state, or a failed invariant - not a narrative assertion that the architecture is nevertheless governed.

49.5 Verification and Evidence Obligations

verify physical cut-set closure

measure halt and safe-state timing

test power, clock, reset, I/O, network, and debug faults

preserve hardware evidence across reset

Evidence must remain bound to the exact artifact, configuration, version, operating envelope, and decision state examined. A passing result establishes only the tested property and coverage. A failed or indeterminate result remains part of the authoritative record. Where independent reproduction has not occurred, the result is not represented as independently established.

49.6 Selected Propositions

P49.1. The validity of sensor, actuator, and i/o boundary integrity cannot exceed the scope of its weakest mandatory dependency.

P49.2. A descriptive or advisory representation does not become an execution constraint until it controls the relevant consequence path.

P49.3. Component-level validity does not compose automatically across interfaces, timing, authority, failure, or deployment change.

P49.4. Evidence of absence is not inferred from absence of evidence; unresolved mandatory state remains unresolved.

49.7 Governance Status and Confidence Type

STRUCTURE. Specified as part of the integrated federated architecture.
EXECUTION REALITY. Requires a domain-specific implementation and complete path inventory.
FAILURE POINT. Any surviving bypass, stale authority, omitted dependency, unmodeled timing path, or unsupported evidence link narrows or defeats the claim.
GOVERNANCE STATUS. M1 unless a chapter-specific case study states a higher internal implementation maturity; no institutional compliance determination is inferred.
CONFIDENCE TYPE. Definitional for the architecture, deductive only under stated assumptions, implementation-dependent for runtime and hardware, institutionally imported for authority, and evidence-dependent for deployment claims.

Chapter 50

Power, Clock, Reset, Watchdog, and Fault-Containment Domains

Brownout, sequencing, clock health, reset cause, watchdog independence, thermal envelopes, and common-mode faults.

50.1 Objective

The objective of this chapter is to define power, clock, reset, watchdog, and fault-containment domains as a bounded component of execution-bound governance. The analysis connects physical consequence path, non-maskable halt, clock and reset integrity, persistent replay protection, I/O validity, platform health, and physical evidence. The decisive test is not whether the system can describe the relevant rule, but whether the rule changes the set of reachable consequence-bearing transitions. The hardware claim is bounded by electrical topology, clock and power envelopes, reset semantics, physical bypasses, I/O integrity, environmental conditions, and the tested artifact.

50.2 Structural Model

A compact representation used in this chapter is:

$$CE = P_H \wedge PowerOK \wedge ClockOK \wedge ThermalOK \wedge ResetReady$$

The structural object is represented as a typed relation rather than an informal checklist. Identity, object, context, authority, version, time, fault state, and evidence are separated because a change in any one of them can create a different transition. The architecture therefore rejects implicit substitution: a permission for one object is not a permission for another; a proof for one model is not a proof for a deployment; an approval before mutation is not approval after mutation. Unknown mandatory state is represented explicitly and does not default to ordinary allow.

The chapter’s principal structural commitments are:

Physical Consequence Path - retained as an explicit state or dependency.

Non-Maskable Halt - retained as an explicit state or dependency.

Clock And Reset Integrity - retained as an explicit state or dependency.

Persistent Replay Protection - retained as an explicit state or dependency.

I/O Validity - retained as an explicit state or dependency.

Platform Health - retained as an explicit state or dependency.

Physical Evidence - retained as an explicit state or dependency.

50.3 Execution Reality

Execution reality begins at the last point from which the protected consequence can still be prevented. Upstream analysis, documentation, monitoring, or recommendation may improve decision quality, but it becomes governance only when the consequence path depends on the resulting state. The chapter therefore treats direct capability, alternate interfaces, administrative paths, recovery mechanisms, stale state, and external dependencies as part of the governed object. A system is not credited with a property merely because a dashboard displays it; the relevant gate, controller, institution, or physical interface must enforce or evidence it.

50.4 Failure Point and Counter-Construction

The claim collapses under any in-scope counter-construction such as:

a physical bypass remains

halt is maskable or reset-cleared

clock or brownout freezes an unsafe output

command acknowledgement is mistaken for physical effect

These failures are not edge cases external to the theory. They are the primary falsification instruments. Where one of them remains reachable, the correct result is a narrowed claim, an unresolved state, or a failed invariant - not a narrative assertion that the architecture is nevertheless governed.

50.5 Verification and Evidence Obligations

verify physical cut-set closure

measure halt and safe-state timing

test power, clock, reset, I/O, network, and debug faults

preserve hardware evidence across reset

Evidence must remain bound to the exact artifact, configuration, version, operating envelope, and decision state examined. A passing result establishes only the tested property and coverage. A failed or indeterminate result remains part of the authoritative record. Where independent reproduction has not occurred, the result is not represented as independently established.

50.6 Selected Propositions

P50.1. The validity of power, clock, reset, watchdog, and fault-containment domains cannot exceed the scope of its weakest mandatory dependency.

P50.2. A descriptive or advisory representation does not become an execution constraint until it controls the relevant consequence path.

P50.3. Component-level validity does not compose automatically across interfaces, timing, authority, failure, or deployment change.

P50.4. Evidence of absence is not inferred from absence of evidence; unresolved mandatory state remains unresolved.

50.7 Governance Status and Confidence Type

STRUCTURE. Specified as part of the integrated federated architecture.
EXECUTION REALITY. Requires a domain-specific implementation and complete path inventory.
FAILURE POINT. Any surviving bypass, stale authority, omitted dependency, unmodeled timing path, or unsupported evidence link narrows or defeats the claim.
GOVERNANCE STATUS. M1 unless a chapter-specific case study states a higher internal implementation maturity; no institutional compliance determination is inferred.
CONFIDENCE TYPE. Definitional for the architecture, deductive only under stated assumptions, implementation-dependent for runtime and hardware, institutionally imported for authority, and evidence-dependent for deployment claims.

Chapter 51

Embedded Communication and Automotive Network Governance

Freshness, source identity, gateways, partitions, bus-off recovery, command arbitration, rate limits, and safety traffic.

51.1 Objective

The objective of this chapter is to define embedded communication and automotive network governance as a bounded component of execution-bound governance. The analysis connects physical consequence path, non-maskable halt, clock and reset integrity, persistent replay protection, I/O validity, platform health, and physical evidence. The decisive test is not whether the system can describe the relevant rule, but whether the rule changes the set of reachable consequence-bearing transitions. The hardware claim is bounded by electrical topology, clock and power envelopes, reset semantics, physical bypasses, I/O integrity, environmental conditions, and the tested artifact.

51.2 Structural Model

A compact representation used in this chapter is:

$$AcceptMsg = SourceValid \wedge Fresh \wedge EpochValid \wedge IntegrityValid \wedge ModeValid$$

The structural object is represented as a typed relation rather than an informal checklist. Identity, object, context, authority, version, time, fault state, and evidence are separated because a change in any one of them can create a different transition. The architecture therefore rejects implicit substitution: a permission for one object is not a permission for another; a proof for one model is not a proof for a deployment; an approval before mutation is not approval after mutation. Unknown mandatory state is represented explicitly and does not default to ordinary allow.

The chapter’s principal structural commitments are:

Physical Consequence Path - retained as an explicit state or dependency.

Non-Maskable Halt - retained as an explicit state or dependency.

Clock And Reset Integrity - retained as an explicit state or dependency.

Persistent Replay Protection - retained as an explicit state or dependency.

I/O Validity - retained as an explicit state or dependency.

Platform Health - retained as an explicit state or dependency.

Physical Evidence - retained as an explicit state or dependency.

51.3 Execution Reality

Execution reality begins at the last point from which the protected consequence can still be prevented. Upstream analysis, documentation, monitoring, or recommendation may improve decision quality, but it becomes governance only when the consequence path depends on the resulting state. The chapter therefore treats direct capability, alternate interfaces, administrative paths, recovery mechanisms, stale state, and external dependencies as part of the governed object. A system is not credited with a property merely because a dashboard displays it; the relevant gate, controller, institution, or physical interface must enforce or evidence it.

51.4 Failure Point and Counter-Construction

The claim collapses under any in-scope counter-construction such as:

a physical bypass remains

halt is maskable or reset-cleared

clock or brownout freezes an unsafe output

command acknowledgement is mistaken for physical effect

These failures are not edge cases external to the theory. They are the primary falsification instruments. Where one of them remains reachable, the correct result is a narrowed claim, an unresolved state, or a failed invariant - not a narrative assertion that the architecture is nevertheless governed.

51.5 Verification and Evidence Obligations

verify physical cut-set closure

measure halt and safe-state timing

test power, clock, reset, I/O, network, and debug faults

preserve hardware evidence across reset

Evidence must remain bound to the exact artifact, configuration, version, operating envelope, and decision state examined. A passing result establishes only the tested property and coverage. A failed or indeterminate result remains part of the authoritative record. Where independent reproduction has not occurred, the result is not represented as independently established.

51.6 Selected Propositions

P51.1. The validity of embedded communication and automotive network governance cannot exceed the scope of its weakest mandatory dependency.

P51.2. A descriptive or advisory representation does not become an execution constraint until it controls the relevant consequence path.

P51.3. Component-level validity does not compose automatically across interfaces, timing, authority, failure, or deployment change.

P51.4. Evidence of absence is not inferred from absence of evidence; unresolved mandatory state remains unresolved.

51.7 Governance Status and Confidence Type

STRUCTURE. Specified as part of the integrated federated architecture.
EXECUTION REALITY. Requires a domain-specific implementation and complete path inventory.
FAILURE POINT. Any surviving bypass, stale authority, omitted dependency, unmodeled timing path, or unsupported evidence link narrows or defeats the claim.
GOVERNANCE STATUS. M1 unless a chapter-specific case study states a higher internal implementation maturity; no institutional compliance determination is inferred.
CONFIDENCE TYPE. Definitional for the architecture, deductive only under stated assumptions, implementation-dependent for runtime and hardware, institutionally imported for authority, and evidence-dependent for deployment claims.

Chapter 52

Hardware-Software Co-Design, Runtime Assurance, and the Governed Co-Chip

A public-safe Co-Chip role, permit vectors, version binding, HALT_N, independent evidence, and compute-plane restraint.

52.1 Objective

The objective of this chapter is to define hardware-software co-design, runtime assurance, and the governed co-chip as a bounded component of execution-bound governance. The analysis connects physical consequence path, non-maskable halt, clock and reset integrity, persistent replay protection, I/O validity, platform health, and physical evidence. The decisive test is not whether the system can describe the relevant rule, but whether the rule changes the set of reachable consequence-bearing transitions. The hardware claim is bounded by electrical topology, clock and power envelopes, reset semantics, physical bypasses, I/O integrity, environmental conditions, and the tested artifact.

52.2 Structural Model

A compact representation used in this chapter is:

$$ALLOW_H = PERMIT_S \wedge TID_{match} \wedge VERSION_{match} \wedge FRESH \wedge INTEGRITY \wedge HEALTH$$

The structural object is represented as a typed relation rather than an informal checklist. Identity, object, context, authority, version, time, fault state, and evidence are separated because a change in any one of them can create a different transition. The architecture therefore rejects implicit substitution: a permission for one object is not a permission for another; a proof for one model is not a proof for a deployment; an approval before mutation is not approval after mutation. Unknown mandatory state is represented explicitly and does not default to ordinary allow.

The chapter's principal structural commitments are:

Physical Consequence Path - retained as an explicit state or dependency.

Non-Maskable Halt - retained as an explicit state or dependency.

Clock And Reset Integrity - retained as an explicit state or dependency.

Persistent Replay Protection - retained as an explicit state or dependency.

I/O Validity - retained as an explicit state or dependency.

Platform Health - retained as an explicit state or dependency.

Physical Evidence - retained as an explicit state or dependency.

52.3 Execution Reality

Execution reality begins at the last point from which the protected consequence can still be prevented. Upstream analysis, documentation, monitoring, or recommendation may improve decision quality, but it becomes governance only when the consequence path depends on the resulting state. The chapter therefore treats direct capability, alternate interfaces, administrative paths, recovery mechanisms, stale state, and external dependencies as part of the governed object. A system is not credited with a property merely because a dashboard displays it; the relevant gate, controller, institution, or physical interface must enforce or evidence it.

52.4 Failure Point and Counter-Construction

The claim collapses under any in-scope counter-construction such as:

a physical bypass remains

halt is maskable or reset-cleared

clock or brownout freezes an unsafe output

command acknowledgement is mistaken for physical effect

These failures are not edge cases external to the theory. They are the primary falsification instruments. Where one of them remains reachable, the correct result is a narrowed claim, an unresolved state, or a failed invariant - not a narrative assertion that the architecture is nevertheless governed.

52.5 Verification and Evidence Obligations

verify physical cut-set closure

measure halt and safe-state timing

test power, clock, reset, I/O, network, and debug faults

preserve hardware evidence across reset

Evidence must remain bound to the exact artifact, configuration, version, operating envelope, and decision state examined. A passing result establishes only the tested property and coverage. A failed or indeterminate result remains part of the authoritative record. Where independent reproduction has not occurred, the result is not represented as independently established.

52.6 Selected Propositions

P52.1. The validity of hardware-software co-design, runtime assurance, and the governed co-chip cannot exceed the scope of its weakest mandatory dependency.

P52.2. A descriptive or advisory representation does not become an execution constraint until it controls the relevant consequence path.

P52.3. Component-level validity does not compose automatically across interfaces, timing, authority, failure, or deployment change.

P52.4. Evidence of absence is not inferred from absence of evidence; unresolved mandatory state remains unresolved.

52.7 Governance Status and Confidence Type

STRUCTURE. Specified as part of the integrated federated architecture.
EXECUTION REALITY. Requires a domain-specific implementation and complete path inventory.
FAILURE POINT. Any surviving bypass, stale authority, omitted dependency, unmodeled timing path, or unsupported evidence link narrows or defeats the claim.
GOVERNANCE STATUS. M1 unless a chapter-specific case study states a higher internal implementation maturity; no institutional compliance determination is inferred.
CONFIDENCE TYPE. Definitional for the architecture, deductive only under stated assumptions, implementation-dependent for runtime and hardware, institutionally imported for authority, and evidence-dependent for deployment claims.

Chapter 53

Automotive Execution Governance and Vehicle Integration

Vehicle authority zones, supervisory AI, driver veto, degraded mobility, network mediation, and physical effect confirmation.

53.1 Objective

The objective of this chapter is to define automotive execution governance and vehicle integration as a bounded component of execution-bound governance. The analysis connects physical consequence path, non-maskable halt, clock and reset integrity, persistent replay protection, I/O validity, platform health, and physical evidence. The decisive test is not whether the system can describe the relevant rule, but whether the rule changes the set of reachable consequence-bearing transitions. The hardware claim is bounded by electrical topology, clock and power envelopes, reset semantics, physical bypasses, I/O integrity, environmental conditions, and the tested artifact.

53.2 Structural Model

A compact representation used in this chapter is:

$$T_{alert}+T_{human}+T_{transfer}+T_{control}<T_{remain}$$

The structural object is represented as a typed relation rather than an informal checklist. Identity, object, context, authority, version, time, fault state, and evidence are separated because a change in any one of them can create a different transition. The architecture therefore rejects implicit substitution: a permission for one object is not a permission for another; a proof for one model is not a proof for a deployment; an approval before mutation is not approval after mutation. Unknown mandatory state is represented explicitly and does not default to ordinary allow.

The chapter’s principal structural commitments are:

Physical Consequence Path - retained as an explicit state or dependency.

Non-Maskable Halt - retained as an explicit state or dependency.

Clock And Reset Integrity - retained as an explicit state or dependency.

Persistent Replay Protection - retained as an explicit state or dependency.

I/O Validity - retained as an explicit state or dependency.

Platform Health - retained as an explicit state or dependency.

Physical Evidence - retained as an explicit state or dependency.

53.3 Execution Reality

Execution reality begins at the last point from which the protected consequence can still be prevented. Upstream analysis, documentation, monitoring, or recommendation may improve decision quality, but it becomes governance only when the consequence path depends on the resulting state. The chapter therefore treats direct capability, alternate interfaces, administrative paths, recovery mechanisms, stale state, and external dependencies as part of the governed object. A system is not credited with a property merely because a dashboard displays it; the relevant gate, controller, institution, or physical interface must enforce or evidence it.

53.4 Failure Point and Counter-Construction

The claim collapses under any in-scope counter-construction such as:

a physical bypass remains

halt is maskable or reset-cleared

clock or brownout freezes an unsafe output

command acknowledgement is mistaken for physical effect

These failures are not edge cases external to the theory. They are the primary falsification instruments. Where one of them remains reachable, the correct result is a narrowed claim, an unresolved state, or a failed invariant - not a narrative assertion that the architecture is nevertheless governed.

53.5 Verification and Evidence Obligations

verify physical cut-set closure

measure halt and safe-state timing

test power, clock, reset, I/O, network, and debug faults

preserve hardware evidence across reset

Evidence must remain bound to the exact artifact, configuration, version, operating envelope, and decision state examined. A passing result establishes only the tested property and coverage. A failed or indeterminate result remains part of the authoritative record. Where independent reproduction has not occurred, the result is not represented as independently established.

53.6 Selected Propositions

P53.1. The validity of automotive execution governance and vehicle integration cannot exceed the scope of its weakest mandatory dependency.

P53.2. A descriptive or advisory representation does not become an execution constraint until it controls the relevant consequence path.

P53.3. Component-level validity does not compose automatically across interfaces, timing, authority, failure, or deployment change.

P53.4. Evidence of absence is not inferred from absence of evidence; unresolved mandatory state remains unresolved.

53.7 Governance Status and Confidence Type

STRUCTURE. Specified as part of the integrated federated architecture.
EXECUTION REALITY. Requires a domain-specific implementation and complete path inventory.
FAILURE POINT. Any surviving bypass, stale authority, omitted dependency, unmodeled timing path, or unsupported evidence link narrows or defeats the claim.
GOVERNANCE STATUS. M1 unless a chapter-specific case study states a higher internal implementation maturity; no institutional compliance determination is inferred.
CONFIDENCE TYPE. Definitional for the architecture, deductive only under stated assumptions, implementation-dependent for runtime and hardware, institutionally imported for authority, and evidence-dependent for deployment claims.

Chapter 54

Hardware Validation, HIL, Fault Injection, Environmental Qualification, and Evidence

RTL and gate-level proof, timing, HIL, physical fault campaigns, production variation, and evidence maturity.

54.1 Objective

The objective of this chapter is to define hardware validation, hil, fault injection, environmental qualification, and evidence as a bounded component of execution-bound governance. The analysis connects physical consequence path, non-maskable halt, clock and reset integrity, persistent replay protection, I/O validity, platform health, and physical evidence. The decisive test is not whether the system can describe the relevant rule, but whether the rule changes the set of reachable consequence-bearing transitions. The hardware claim is bounded by electrical topology, clock and power envelopes, reset semantics, physical bypasses, I/O integrity, environmental conditions, and the tested artifact.

54.2 Structural Model

The structural object is represented as a typed relation rather than an informal checklist. Identity, object, context, authority, version, time, fault state, and evidence are separated because a change in any one of them can create a different transition. The architecture therefore rejects implicit substitution: a permission for one object is not a permission for another; a proof for one model is not a proof for a deployment; an approval before mutation is not approval after mutation. Unknown mandatory state is represented explicitly and does not default to ordinary allow.

The chapter’s principal structural commitments are:

Physical Consequence Path - retained as an explicit state or dependency.

Non-Maskable Halt - retained as an explicit state or dependency.

Clock And Reset Integrity - retained as an explicit state or dependency.

Persistent Replay Protection - retained as an explicit state or dependency.

I/O Validity - retained as an explicit state or dependency.

Platform Health - retained as an explicit state or dependency.

Physical Evidence - retained as an explicit state or dependency.

54.3 Execution Reality

Execution reality begins at the last point from which the protected consequence can still be prevented. Upstream analysis, documentation, monitoring, or recommendation may improve decision quality, but it becomes governance only when the consequence path depends on the resulting state. The chapter therefore treats direct capability, alternate interfaces, administrative paths, recovery mechanisms, stale state, and external dependencies as part of the governed object. A system is not credited with a property merely because a dashboard displays it; the relevant gate, controller, institution, or physical interface must enforce or evidence it.

54.4 Failure Point and Counter-Construction

The claim collapses under any in-scope counter-construction such as:

a physical bypass remains

halt is maskable or reset-cleared

clock or brownout freezes an unsafe output

command acknowledgement is mistaken for physical effect

These failures are not edge cases external to the theory. They are the primary falsification instruments. Where one of them remains reachable, the correct result is a narrowed claim, an unresolved state, or a failed invariant - not a narrative assertion that the architecture is nevertheless governed.

54.5 Verification and Evidence Obligations

verify physical cut-set closure

measure halt and safe-state timing

test power, clock, reset, I/O, network, and debug faults

preserve hardware evidence across reset

Evidence must remain bound to the exact artifact, configuration, version, operating envelope, and decision state examined. A passing result establishes only the tested property and coverage. A failed or indeterminate result remains part of the authoritative record. Where independent reproduction has not occurred, the result is not represented as independently established.

54.6 Selected Propositions

P54.1. The validity of hardware validation, hil, fault injection, environmental qualification, and evidence cannot exceed the scope of its weakest mandatory dependency.

P54.2. A descriptive or advisory representation does not become an execution constraint until it controls the relevant consequence path.

P54.3. Component-level validity does not compose automatically across interfaces, timing, authority, failure, or deployment change.

P54.4. Evidence of absence is not inferred from absence of evidence; unresolved mandatory state remains unresolved.

54.7 Governance Status and Confidence Type

STRUCTURE. Specified as part of the integrated federated architecture.
EXECUTION REALITY. Requires a domain-specific implementation and complete path inventory.
FAILURE POINT. Any surviving bypass, stale authority, omitted dependency, unmodeled timing path, or unsupported evidence link narrows or defeats the claim.
GOVERNANCE STATUS. M1 unless a chapter-specific case study states a higher internal implementation maturity; no institutional compliance determination is inferred.
CONFIDENCE TYPE. Definitional for the architecture, deductive only under stated assumptions, implementation-dependent for runtime and hardware, institutionally imported for authority, and evidence-dependent for deployment claims.

Chapter 55

Hardware-Volume Synthesis and Closure

The unified governed hardware system, master hardware permit, physical path closure, non-claims, and export to human authority.

55.1 Objective

The objective of this chapter is to define hardware-volume synthesis and closure as a bounded component of execution-bound governance. The analysis connects physical consequence path, non-maskable halt, clock and reset integrity, persistent replay protection, I/O validity, platform health, and physical evidence. The decisive test is not whether the system can describe the relevant rule, but whether the rule changes the set of reachable consequence-bearing transitions. The hardware claim is bounded by electrical topology, clock and power envelopes, reset semantics, physical bypasses, I/O integrity, environmental conditions, and the tested artifact.

55.2 Structural Model

A compact representation used in this chapter is:

$$PhysicalEffect \Rightarrow Permit_H$$

The structural object is represented as a typed relation rather than an informal checklist. Identity, object, context, authority, version, time, fault state, and evidence are separated because a change in any one of them can create a different transition. The architecture therefore rejects implicit substitution: a permission for one object is not a permission for another; a proof for one model is not a proof for a deployment; an approval before mutation is not approval after mutation. Unknown mandatory state is represented explicitly and does not default to ordinary allow.

The chapter’s principal structural commitments are:

Physical Consequence Path - retained as an explicit state or dependency.

Non-Maskable Halt - retained as an explicit state or dependency.

Clock And Reset Integrity - retained as an explicit state or dependency.

Persistent Replay Protection - retained as an explicit state or dependency.

I/O Validity - retained as an explicit state or dependency.

Platform Health - retained as an explicit state or dependency.

Physical Evidence - retained as an explicit state or dependency.

55.3 Execution Reality

Execution reality begins at the last point from which the protected consequence can still be prevented. Upstream analysis, documentation, monitoring, or recommendation may improve decision quality, but it becomes governance only when the consequence path depends on the resulting state. The chapter therefore treats direct capability, alternate interfaces, administrative paths, recovery mechanisms, stale state, and external dependencies as part of the governed object. A system is not credited with a property merely because a dashboard displays it; the relevant gate, controller, institution, or physical interface must enforce or evidence it.

55.4 Failure Point and Counter-Construction

The claim collapses under any in-scope counter-construction such as:

a physical bypass remains

halt is maskable or reset-cleared

clock or brownout freezes an unsafe output

command acknowledgement is mistaken for physical effect

These failures are not edge cases external to the theory. They are the primary falsification instruments. Where one of them remains reachable, the correct result is a narrowed claim, an unresolved state, or a failed invariant - not a narrative assertion that the architecture is nevertheless governed.

55.5 Verification and Evidence Obligations

verify physical cut-set closure

measure halt and safe-state timing

test power, clock, reset, I/O, network, and debug faults

preserve hardware evidence across reset

Evidence must remain bound to the exact artifact, configuration, version, operating envelope, and decision state examined. A passing result establishes only the tested property and coverage. A failed or indeterminate result remains part of the authoritative record. Where independent reproduction has not occurred, the result is not represented as independently established.

55.6 Selected Propositions

P55.1. The validity of hardware-volume synthesis and closure cannot exceed the scope of its weakest mandatory dependency.

P55.2. A descriptive or advisory representation does not become an execution constraint until it controls the relevant consequence path.

P55.3. Component-level validity does not compose automatically across interfaces, timing, authority, failure, or deployment change.

P55.4. Evidence of absence is not inferred from absence of evidence; unresolved mandatory state remains unresolved.

55.7 Governance Status and Confidence Type

STRUCTURE. Specified as part of the integrated federated architecture.
EXECUTION REALITY. Requires a domain-specific implementation and complete path inventory.
FAILURE POINT. Any surviving bypass, stale authority, omitted dependency, unmodeled timing path, or unsupported evidence link narrows or defeats the claim.
GOVERNANCE STATUS. M1 unless a chapter-specific case study states a higher internal implementation maturity; no institutional compliance determination is inferred.
CONFIDENCE TYPE. Definitional for the architecture, deductive only under stated assumptions, implementation-dependent for runtime and hardware, institutionally imported for authority, and evidence-dependent for deployment claims.

PART VI - HUMAN AUTHORITY, OVERSIGHT, AND INSTITUTIONAL CONTROL

Part VI formalizes the human and institutional authority that the technical stack consumes. It rejects presence, role labels, and nominal approval as substitutes for mandate, competence, time, control, and evidence.

Part-level claim boundary. The material in this part is specified at M1 unless a bounded case study explicitly records internal implementation evidence. No part-level statement independently widens the system-level claim.

Chapter 56

Human Authority as a Typed and Time-Bounded Commit Relation

Identity, mandate, competence, scope, explicit commit, veto, expiry, conflict, and handoff.

56.1 Objective

The objective of this chapter is to define human authority as a typed and time-bounded commit relation as a bounded component of execution-bound governance. The analysis connects identity and mandate, competence, explicit human commit, delegation and veto, oversight feasibility, responsibility, and contestability. The decisive test is not whether the system can describe the relevant rule, but whether the rule changes the set of reachable consequence-bearing transitions. The authority claim imports institutional and, where applicable, legal mandate; it does not infer legitimacy from technical access, role labels, or observed human behavior.

56.2 Structural Model

A compact representation used in this chapter is:

$$Permit_A = IdentityValid \wedge MandateValid \wedge CompetenceValid \wedge CommitExplicit$$

The structural object is represented as a typed relation rather than an informal checklist. Identity, object, context, authority, version, time, fault state, and evidence are separated because a change in any one of them can create a different transition. The architecture therefore rejects implicit substitution: a permission for one object is not a permission for another; a proof for one model is not a proof for a deployment; an approval before mutation is not approval after mutation. Unknown mandatory state is represented explicitly and does not default to ordinary allow.

The chapter’s principal structural commitments are:

Identity And Mandate - retained as an explicit state or dependency.

Competence - retained as an explicit state or dependency.

Explicit Human Commit - retained as an explicit state or dependency.

Delegation And Veto - retained as an explicit state or dependency.

Oversight Feasibility - retained as an explicit state or dependency.

Responsibility - retained as an explicit state or dependency.

Contestability - retained as an explicit state or dependency.

56.3 Execution Reality

Execution reality begins at the last point from which the protected consequence can still be prevented. Upstream analysis, documentation, monitoring, or recommendation may improve decision quality, but it becomes governance only when the consequence path depends on the resulting state. The chapter therefore treats direct capability, alternate interfaces, administrative paths, recovery mechanisms, stale state, and external dependencies as part of the governed object. A system is not credited with a property merely because a dashboard displays it; the relevant gate, controller, institution, or physical interface must enforce or evidence it.

56.4 Failure Point and Counter-Construction

The claim collapses under any in-scope counter-construction such as:

presence is treated as consent

role is treated as competence

veto cannot reach the consequence path

responsibility is assigned without real control

These failures are not edge cases external to the theory. They are the primary falsification instruments. Where one of them remains reachable, the correct result is a narrowed claim, an unresolved state, or a failed invariant - not a narrative assertion that the architecture is nevertheless governed.

56.5 Verification and Evidence Obligations

verify mandate and competence

measure intervention feasibility

test display-commit equivalence and veto effectiveness

preserve consent, handoff, and contest evidence

Evidence must remain bound to the exact artifact, configuration, version, operating envelope, and decision state examined. A passing result establishes only the tested property and coverage. A failed or indeterminate result remains part of the authoritative record. Where independent reproduction has not occurred, the result is not represented as independently established.

56.6 Selected Propositions

P56.1. The validity of human authority as a typed and time-bounded commit relation cannot exceed the scope of its weakest mandatory dependency.

P56.2. A descriptive or advisory representation does not become an execution constraint until it controls the relevant consequence path.

P56.3. Component-level validity does not compose automatically across interfaces, timing, authority, failure, or deployment change.

P56.4. Evidence of absence is not inferred from absence of evidence; unresolved mandatory state remains unresolved.

56.7 Governance Status and Confidence Type

STRUCTURE. Specified as part of the integrated federated architecture.
EXECUTION REALITY. Requires a domain-specific implementation and complete path inventory.
FAILURE POINT. Any surviving bypass, stale authority, omitted dependency, unmodeled timing path, or unsupported evidence link narrows or defeats the claim.
GOVERNANCE STATUS. M1 unless a chapter-specific case study states a higher internal implementation maturity; no institutional compliance determination is inferred.
CONFIDENCE TYPE. Definitional for the architecture, deductive only under stated assumptions, implementation-dependent for runtime and hardware, institutionally imported for authority, and evidence-dependent for deployment claims.

Chapter 57

Delegation, Veto, Handoff, and Conflict Resolution

Delegation graphs, attenuation, revocation closure, dual control, quorum, conflict, escalation, and responsibility continuity.

57.1 Objective

The objective of this chapter is to define delegation, veto, handoff, and conflict resolution as a bounded component of execution-bound governance. The analysis connects identity and mandate, competence, explicit human commit, delegation and veto, oversight feasibility, responsibility, and contestability. The decisive test is not whether the system can describe the relevant rule, but whether the rule changes the set of reachable consequence-bearing transitions. The authority claim imports institutional and, where applicable, legal mandate; it does not infer legitimacy from technical access, role labels, or observed human behavior.

57.2 Structural Model

A compact representation used in this chapter is:

$$A_{delegate} \leq A_{delegator}$$

The structural object is represented as a typed relation rather than an informal checklist. Identity, object, context, authority, version, time, fault state, and evidence are separated because a change in any one of them can create a different transition. The architecture therefore rejects implicit substitution: a permission for one object is not a permission for another; a proof for one model is not a proof for a deployment; an approval before mutation is not approval after mutation. Unknown mandatory state is represented explicitly and does not default to ordinary allow.

The chapter’s principal structural commitments are:

Identity And Mandate - retained as an explicit state or dependency.

Competence - retained as an explicit state or dependency.

Explicit Human Commit - retained as an explicit state or dependency.

Delegation And Veto - retained as an explicit state or dependency.

Oversight Feasibility - retained as an explicit state or dependency.

Responsibility - retained as an explicit state or dependency.

Contestability - retained as an explicit state or dependency.

57.3 Execution Reality

Execution reality begins at the last point from which the protected consequence can still be prevented. Upstream analysis, documentation, monitoring, or recommendation may improve decision quality, but it becomes governance only when the consequence path depends on the resulting state. The chapter therefore treats direct capability, alternate interfaces, administrative paths, recovery mechanisms, stale state, and external dependencies as part of the governed object. A system is not credited with a property merely because a dashboard displays it; the relevant gate, controller, institution, or physical interface must enforce or evidence it.

57.4 Failure Point and Counter-Construction

The claim collapses under any in-scope counter-construction such as:

presence is treated as consent

role is treated as competence

veto cannot reach the consequence path

responsibility is assigned without real control

These failures are not edge cases external to the theory. They are the primary falsification instruments. Where one of them remains reachable, the correct result is a narrowed claim, an unresolved state, or a failed invariant - not a narrative assertion that the architecture is nevertheless governed.

57.5 Verification and Evidence Obligations

verify mandate and competence

measure intervention feasibility

test display-commit equivalence and veto effectiveness

preserve consent, handoff, and contest evidence

Evidence must remain bound to the exact artifact, configuration, version, operating envelope, and decision state examined. A passing result establishes only the tested property and coverage. A failed or indeterminate result remains part of the authoritative record. Where independent reproduction has not occurred, the result is not represented as independently established.

57.6 Selected Propositions

P57.1. The validity of delegation, veto, handoff, and conflict resolution cannot exceed the scope of its weakest mandatory dependency.

P57.2. A descriptive or advisory representation does not become an execution constraint until it controls the relevant consequence path.

P57.3. Component-level validity does not compose automatically across interfaces, timing, authority, failure, or deployment change.

P57.4. Evidence of absence is not inferred from absence of evidence; unresolved mandatory state remains unresolved.

57.7 Governance Status and Confidence Type

STRUCTURE. Specified as part of the integrated federated architecture.
EXECUTION REALITY. Requires a domain-specific implementation and complete path inventory.
FAILURE POINT. Any surviving bypass, stale authority, omitted dependency, unmodeled timing path, or unsupported evidence link narrows or defeats the claim.
GOVERNANCE STATUS. M1 unless a chapter-specific case study states a higher internal implementation maturity; no institutional compliance determination is inferred.
CONFIDENCE TYPE. Definitional for the architecture, deductive only under stated assumptions, implementation-dependent for runtime and hardware, institutionally imported for authority, and evidence-dependent for deployment claims.

Chapter 58

Human Oversight under Cognitive Limits and Time Pressure

Decision windows, workload, automation bias, alert fatigue, abstention, uncertainty disclosure, and operational oversight.

58.1 Objective

The objective of this chapter is to define human oversight under cognitive limits and time pressure as a bounded component of execution-bound governance. The analysis connects identity and mandate, competence, explicit human commit, delegation and veto, oversight feasibility, responsibility, and contestability. The decisive test is not whether the system can describe the relevant rule, but whether the rule changes the set of reachable consequence-bearing transitions. The authority claim imports institutional and, where applicable, legal mandate; it does not infer legitimacy from technical access, role labels, or observed human behavior.

58.2 Structural Model

A compact representation used in this chapter is:

$$M_H = T_{remain} - \left(T_{alert} + T_{interpret} + T_{decide} + T_{commit} + T_{effect}\right)$$

The structural object is represented as a typed relation rather than an informal checklist. Identity, object, context, authority, version, time, fault state, and evidence are separated because a change in any one of them can create a different transition. The architecture therefore rejects implicit substitution: a permission for one object is not a permission for another; a proof for one model is not a proof for a deployment; an approval before mutation is not approval after mutation. Unknown mandatory state is represented explicitly and does not default to ordinary allow.

The chapter’s principal structural commitments are:

Identity And Mandate - retained as an explicit state or dependency.

Competence - retained as an explicit state or dependency.

Explicit Human Commit - retained as an explicit state or dependency.

Delegation And Veto - retained as an explicit state or dependency.

Oversight Feasibility - retained as an explicit state or dependency.

Responsibility - retained as an explicit state or dependency.

Contestability - retained as an explicit state or dependency.

58.3 Execution Reality

Execution reality begins at the last point from which the protected consequence can still be prevented. Upstream analysis, documentation, monitoring, or recommendation may improve decision quality, but it becomes governance only when the consequence path depends on the resulting state. The chapter therefore treats direct capability, alternate interfaces, administrative paths, recovery mechanisms, stale state, and external dependencies as part of the governed object. A system is not credited with a property merely because a dashboard displays it; the relevant gate, controller, institution, or physical interface must enforce or evidence it.

58.4 Failure Point and Counter-Construction

The claim collapses under any in-scope counter-construction such as:

presence is treated as consent

role is treated as competence

veto cannot reach the consequence path

responsibility is assigned without real control

These failures are not edge cases external to the theory. They are the primary falsification instruments. Where one of them remains reachable, the correct result is a narrowed claim, an unresolved state, or a failed invariant - not a narrative assertion that the architecture is nevertheless governed.

58.5 Verification and Evidence Obligations

verify mandate and competence

measure intervention feasibility

test display-commit equivalence and veto effectiveness

preserve consent, handoff, and contest evidence

Evidence must remain bound to the exact artifact, configuration, version, operating envelope, and decision state examined. A passing result establishes only the tested property and coverage. A failed or indeterminate result remains part of the authoritative record. Where independent reproduction has not occurred, the result is not represented as independently established.

58.6 Selected Propositions

P58.1. The validity of human oversight under cognitive limits and time pressure cannot exceed the scope of its weakest mandatory dependency.

P58.2. A descriptive or advisory representation does not become an execution constraint until it controls the relevant consequence path.

P58.3. Component-level validity does not compose automatically across interfaces, timing, authority, failure, or deployment change.

P58.4. Evidence of absence is not inferred from absence of evidence; unresolved mandatory state remains unresolved.

58.7 Governance Status and Confidence Type

STRUCTURE. Specified as part of the integrated federated architecture.
EXECUTION REALITY. Requires a domain-specific implementation and complete path inventory.
FAILURE POINT. Any surviving bypass, stale authority, omitted dependency, unmodeled timing path, or unsupported evidence link narrows or defeats the claim.
GOVERNANCE STATUS. M1 unless a chapter-specific case study states a higher internal implementation maturity; no institutional compliance determination is inferred.
CONFIDENCE TYPE. Definitional for the architecture, deductive only under stated assumptions, implementation-dependent for runtime and hardware, institutionally imported for authority, and evidence-dependent for deployment claims.

Chapter 59

Institutional Mandates, Role Systems, and Multi-Principal Governance

Offices, role assignments, procedure, jurisdiction, separation of duties, committees, institutional memory, and shadow authority.

59.1 Objective

The objective of this chapter is to define institutional mandates, role systems, and multi-principal governance as a bounded component of execution-bound governance. The analysis connects identity and mandate, competence, explicit human commit, delegation and veto, oversight feasibility, responsibility, and contestability. The decisive test is not whether the system can describe the relevant rule, but whether the rule changes the set of reachable consequence-bearing transitions. The authority claim imports institutional and, where applicable, legal mandate; it does not infer legitimacy from technical access, role labels, or observed human behavior.

59.2 Structural Model

A compact representation used in this chapter is:

$$D_I = (institution, procedure, participants, quorum, decision, object, time, record)$$

The structural object is represented as a typed relation rather than an informal checklist. Identity, object, context, authority, version, time, fault state, and evidence are separated because a change in any one of them can create a different transition. The architecture therefore rejects implicit substitution: a permission for one object is not a permission for another; a proof for one model is not a proof for a deployment; an approval before mutation is not approval after mutation. Unknown mandatory state is represented explicitly and does not default to ordinary allow.

The chapter’s principal structural commitments are:

Identity And Mandate - retained as an explicit state or dependency.

Competence - retained as an explicit state or dependency.

Explicit Human Commit - retained as an explicit state or dependency.

Delegation And Veto - retained as an explicit state or dependency.

Oversight Feasibility - retained as an explicit state or dependency.

Responsibility - retained as an explicit state or dependency.

Contestability - retained as an explicit state or dependency.

59.3 Execution Reality

Execution reality begins at the last point from which the protected consequence can still be prevented. Upstream analysis, documentation, monitoring, or recommendation may improve decision quality, but it becomes governance only when the consequence path depends on the resulting state. The chapter therefore treats direct capability, alternate interfaces, administrative paths, recovery mechanisms, stale state, and external dependencies as part of the governed object. A system is not credited with a property merely because a dashboard displays it; the relevant gate, controller, institution, or physical interface must enforce or evidence it.

59.4 Failure Point and Counter-Construction

The claim collapses under any in-scope counter-construction such as:

presence is treated as consent

role is treated as competence

veto cannot reach the consequence path

responsibility is assigned without real control

These failures are not edge cases external to the theory. They are the primary falsification instruments. Where one of them remains reachable, the correct result is a narrowed claim, an unresolved state, or a failed invariant - not a narrative assertion that the architecture is nevertheless governed.

59.5 Verification and Evidence Obligations

verify mandate and competence

measure intervention feasibility

test display-commit equivalence and veto effectiveness

preserve consent, handoff, and contest evidence

Evidence must remain bound to the exact artifact, configuration, version, operating envelope, and decision state examined. A passing result establishes only the tested property and coverage. A failed or indeterminate result remains part of the authoritative record. Where independent reproduction has not occurred, the result is not represented as independently established.

59.6 Selected Propositions

P59.1. The validity of institutional mandates, role systems, and multi-principal governance cannot exceed the scope of its weakest mandatory dependency.

P59.2. A descriptive or advisory representation does not become an execution constraint until it controls the relevant consequence path.

P59.3. Component-level validity does not compose automatically across interfaces, timing, authority, failure, or deployment change.

P59.4. Evidence of absence is not inferred from absence of evidence; unresolved mandatory state remains unresolved.

59.7 Governance Status and Confidence Type

STRUCTURE. Specified as part of the integrated federated architecture.
EXECUTION REALITY. Requires a domain-specific implementation and complete path inventory.
FAILURE POINT. Any surviving bypass, stale authority, omitted dependency, unmodeled timing path, or unsupported evidence link narrows or defeats the claim.
GOVERNANCE STATUS. M1 unless a chapter-specific case study states a higher internal implementation maturity; no institutional compliance determination is inferred.
CONFIDENCE TYPE. Definitional for the architecture, deductive only under stated assumptions, implementation-dependent for runtime and hardware, institutionally imported for authority, and evidence-dependent for deployment claims.

Chapter 60

Consent, Refusal, and Non-Coercive Interaction

Informed and specific consent, revocation, refusal, accessibility, power asymmetry, dark patterns, proxy authority, and emergency exceptions.

60.1 Objective

The objective of this chapter is to define consent, refusal, and non-coercive interaction as a bounded component of execution-bound governance. The analysis connects identity and mandate, competence, explicit human commit, delegation and veto, oversight feasibility, responsibility, and contestability. The decisive test is not whether the system can describe the relevant rule, but whether the rule changes the set of reachable consequence-bearing transitions. The authority claim imports institutional and, where applicable, legal mandate; it does not infer legitimacy from technical access, role labels, or observed human behavior.

60.2 Structural Model

A compact representation used in this chapter is:

$$ConsentValid = Informed \wedge Specific \wedge Current \wedge Unrevoked \wedge Accessible$$

The structural object is represented as a typed relation rather than an informal checklist. Identity, object, context, authority, version, time, fault state, and evidence are separated because a change in any one of them can create a different transition. The architecture therefore rejects implicit substitution: a permission for one object is not a permission for another; a proof for one model is not a proof for a deployment; an approval before mutation is not approval after mutation. Unknown mandatory state is represented explicitly and does not default to ordinary allow.

The chapter’s principal structural commitments are:

Identity And Mandate - retained as an explicit state or dependency.

Competence - retained as an explicit state or dependency.

Explicit Human Commit - retained as an explicit state or dependency.

Delegation And Veto - retained as an explicit state or dependency.

Oversight Feasibility - retained as an explicit state or dependency.

Responsibility - retained as an explicit state or dependency.

Contestability - retained as an explicit state or dependency.

60.3 Execution Reality

Execution reality begins at the last point from which the protected consequence can still be prevented. Upstream analysis, documentation, monitoring, or recommendation may improve decision quality, but it becomes governance only when the consequence path depends on the resulting state. The chapter therefore treats direct capability, alternate interfaces, administrative paths, recovery mechanisms, stale state, and external dependencies as part of the governed object. A system is not credited with a property merely because a dashboard displays it; the relevant gate, controller, institution, or physical interface must enforce or evidence it.

60.4 Failure Point and Counter-Construction

The claim collapses under any in-scope counter-construction such as:

presence is treated as consent

role is treated as competence

veto cannot reach the consequence path

responsibility is assigned without real control

These failures are not edge cases external to the theory. They are the primary falsification instruments. Where one of them remains reachable, the correct result is a narrowed claim, an unresolved state, or a failed invariant - not a narrative assertion that the architecture is nevertheless governed.

60.5 Verification and Evidence Obligations

verify mandate and competence

measure intervention feasibility

test display-commit equivalence and veto effectiveness

preserve consent, handoff, and contest evidence

Evidence must remain bound to the exact artifact, configuration, version, operating envelope, and decision state examined. A passing result establishes only the tested property and coverage. A failed or indeterminate result remains part of the authoritative record. Where independent reproduction has not occurred, the result is not represented as independently established.

60.6 Selected Propositions

P60.1. The validity of consent, refusal, and non-coercive interaction cannot exceed the scope of its weakest mandatory dependency.

P60.2. A descriptive or advisory representation does not become an execution constraint until it controls the relevant consequence path.

P60.3. Component-level validity does not compose automatically across interfaces, timing, authority, failure, or deployment change.

P60.4. Evidence of absence is not inferred from absence of evidence; unresolved mandatory state remains unresolved.

60.7 Governance Status and Confidence Type

STRUCTURE. Specified as part of the integrated federated architecture.
EXECUTION REALITY. Requires a domain-specific implementation and complete path inventory.
FAILURE POINT. Any surviving bypass, stale authority, omitted dependency, unmodeled timing path, or unsupported evidence link narrows or defeats the claim.
GOVERNANCE STATUS. M1 unless a chapter-specific case study states a higher internal implementation maturity; no institutional compliance determination is inferred.
CONFIDENCE TYPE. Definitional for the architecture, deductive only under stated assumptions, implementation-dependent for runtime and hardware, institutionally imported for authority, and evidence-dependent for deployment claims.

Chapter 61

Responsibility, Accountability, and Evidence of Human Decision

Duty, authority, information, control, causation, shared responsibility, organizational responsibility, and responsibility gaps.

61.1 Objective

The objective of this chapter is to define responsibility, accountability, and evidence of human decision as a bounded component of execution-bound governance. The analysis connects identity and mandate, competence, explicit human commit, delegation and veto, oversight feasibility, responsibility, and contestability. The decisive test is not whether the system can describe the relevant rule, but whether the rule changes the set of reachable consequence-bearing transitions. The authority claim imports institutional and, where applicable, legal mandate; it does not infer legitimacy from technical access, role labels, or observed human behavior.

61.2 Structural Model

A compact representation used in this chapter is:

$$RespAdequate = Duty \wedge Authority \wedge Information \wedge Control \wedge Time \wedge Evidence$$

The structural object is represented as a typed relation rather than an informal checklist. Identity, object, context, authority, version, time, fault state, and evidence are separated because a change in any one of them can create a different transition. The architecture therefore rejects implicit substitution: a permission for one object is not a permission for another; a proof for one model is not a proof for a deployment; an approval before mutation is not approval after mutation. Unknown mandatory state is represented explicitly and does not default to ordinary allow.

The chapter’s principal structural commitments are:

Identity And Mandate - retained as an explicit state or dependency.

Competence - retained as an explicit state or dependency.

Explicit Human Commit - retained as an explicit state or dependency.

Delegation And Veto - retained as an explicit state or dependency.

Oversight Feasibility - retained as an explicit state or dependency.

Responsibility - retained as an explicit state or dependency.

Contestability - retained as an explicit state or dependency.

61.3 Execution Reality

Execution reality begins at the last point from which the protected consequence can still be prevented. Upstream analysis, documentation, monitoring, or recommendation may improve decision quality, but it becomes governance only when the consequence path depends on the resulting state. The chapter therefore treats direct capability, alternate interfaces, administrative paths, recovery mechanisms, stale state, and external dependencies as part of the governed object. A system is not credited with a property merely because a dashboard displays it; the relevant gate, controller, institution, or physical interface must enforce or evidence it.

61.4 Failure Point and Counter-Construction

The claim collapses under any in-scope counter-construction such as:

presence is treated as consent

role is treated as competence

veto cannot reach the consequence path

responsibility is assigned without real control

These failures are not edge cases external to the theory. They are the primary falsification instruments. Where one of them remains reachable, the correct result is a narrowed claim, an unresolved state, or a failed invariant - not a narrative assertion that the architecture is nevertheless governed.

61.5 Verification and Evidence Obligations

verify mandate and competence

measure intervention feasibility

test display-commit equivalence and veto effectiveness

preserve consent, handoff, and contest evidence

Evidence must remain bound to the exact artifact, configuration, version, operating envelope, and decision state examined. A passing result establishes only the tested property and coverage. A failed or indeterminate result remains part of the authoritative record. Where independent reproduction has not occurred, the result is not represented as independently established.

61.6 Selected Propositions

P61.1. The validity of responsibility, accountability, and evidence of human decision cannot exceed the scope of its weakest mandatory dependency.

P61.2. A descriptive or advisory representation does not become an execution constraint until it controls the relevant consequence path.

P61.3. Component-level validity does not compose automatically across interfaces, timing, authority, failure, or deployment change.

P61.4. Evidence of absence is not inferred from absence of evidence; unresolved mandatory state remains unresolved.

61.7 Governance Status and Confidence Type

STRUCTURE. Specified as part of the integrated federated architecture.
EXECUTION REALITY. Requires a domain-specific implementation and complete path inventory.
FAILURE POINT. Any surviving bypass, stale authority, omitted dependency, unmodeled timing path, or unsupported evidence link narrows or defeats the claim.
GOVERNANCE STATUS. M1 unless a chapter-specific case study states a higher internal implementation maturity; no institutional compliance determination is inferred.
CONFIDENCE TYPE. Definitional for the architecture, deductive only under stated assumptions, implementation-dependent for runtime and hardware, institutionally imported for authority, and evidence-dependent for deployment claims.

Chapter 62

Human-AI Interface Design for Governed Commit

Display-commit equivalence, consequence preview, alternatives, confirmation friction, veto, freshness, accessibility, and receipts.

62.1 Objective

The objective of this chapter is to define human-ai interface design for governed commit as a bounded component of execution-bound governance. The analysis connects identity and mandate, competence, explicit human commit, delegation and veto, oversight feasibility, responsibility, and contestability. The decisive test is not whether the system can describe the relevant rule, but whether the rule changes the set of reachable consequence-bearing transitions. The authority claim imports institutional and, where applicable, legal mandate; it does not infer legitimacy from technical access, role labels, or observed human behavior.

62.2 Structural Model

A compact representation used in this chapter is:

$$T_D \equiv_C T_C$$

The structural object is represented as a typed relation rather than an informal checklist. Identity, object, context, authority, version, time, fault state, and evidence are separated because a change in any one of them can create a different transition. The architecture therefore rejects implicit substitution: a permission for one object is not a permission for another; a proof for one model is not a proof for a deployment; an approval before mutation is not approval after mutation. Unknown mandatory state is represented explicitly and does not default to ordinary allow.

The chapter’s principal structural commitments are:

Identity And Mandate - retained as an explicit state or dependency.

Competence - retained as an explicit state or dependency.

Explicit Human Commit - retained as an explicit state or dependency.

Delegation And Veto - retained as an explicit state or dependency.

Oversight Feasibility - retained as an explicit state or dependency.

Responsibility - retained as an explicit state or dependency.

Contestability - retained as an explicit state or dependency.

62.3 Execution Reality

Execution reality begins at the last point from which the protected consequence can still be prevented. Upstream analysis, documentation, monitoring, or recommendation may improve decision quality, but it becomes governance only when the consequence path depends on the resulting state. The chapter therefore treats direct capability, alternate interfaces, administrative paths, recovery mechanisms, stale state, and external dependencies as part of the governed object. A system is not credited with a property merely because a dashboard displays it; the relevant gate, controller, institution, or physical interface must enforce or evidence it.

62.4 Failure Point and Counter-Construction

The claim collapses under any in-scope counter-construction such as:

presence is treated as consent

role is treated as competence

veto cannot reach the consequence path

responsibility is assigned without real control

These failures are not edge cases external to the theory. They are the primary falsification instruments. Where one of them remains reachable, the correct result is a narrowed claim, an unresolved state, or a failed invariant - not a narrative assertion that the architecture is nevertheless governed.

62.5 Verification and Evidence Obligations

verify mandate and competence

measure intervention feasibility

test display-commit equivalence and veto effectiveness

preserve consent, handoff, and contest evidence

Evidence must remain bound to the exact artifact, configuration, version, operating envelope, and decision state examined. A passing result establishes only the tested property and coverage. A failed or indeterminate result remains part of the authoritative record. Where independent reproduction has not occurred, the result is not represented as independently established.

62.6 Selected Propositions

P62.1. The validity of human-ai interface design for governed commit cannot exceed the scope of its weakest mandatory dependency.

P62.2. A descriptive or advisory representation does not become an execution constraint until it controls the relevant consequence path.

P62.3. Component-level validity does not compose automatically across interfaces, timing, authority, failure, or deployment change.

P62.4. Evidence of absence is not inferred from absence of evidence; unresolved mandatory state remains unresolved.

62.7 Governance Status and Confidence Type

STRUCTURE. Specified as part of the integrated federated architecture.
EXECUTION REALITY. Requires a domain-specific implementation and complete path inventory.
FAILURE POINT. Any surviving bypass, stale authority, omitted dependency, unmodeled timing path, or unsupported evidence link narrows or defeats the claim.
GOVERNANCE STATUS. M1 unless a chapter-specific case study states a higher internal implementation maturity; no institutional compliance determination is inferred.
CONFIDENCE TYPE. Definitional for the architecture, deductive only under stated assumptions, implementation-dependent for runtime and hardware, institutionally imported for authority, and evidence-dependent for deployment claims.

Chapter 63

Escalation, Appeals, Contestability, and Corrective Authority

Notice, review, suspensive appeal, reversal, correction, remedy, evidence preservation, and irreversibility.

63.1 Objective

The objective of this chapter is to define escalation, appeals, contestability, and corrective authority as a bounded component of execution-bound governance. The analysis connects identity and mandate, competence, explicit human commit, delegation and veto, oversight feasibility, responsibility, and contestability. The decisive test is not whether the system can describe the relevant rule, but whether the rule changes the set of reachable consequence-bearing transitions. The authority claim imports institutional and, where applicable, legal mandate; it does not infer legitimacy from technical access, role labels, or observed human behavior.

63.2 Structural Model

A compact representation used in this chapter is:

$$Contestable = Notice \wedge Appeal \wedge ReviewAuthority \wedge Remedy \wedge Enforcement$$

The structural object is represented as a typed relation rather than an informal checklist. Identity, object, context, authority, version, time, fault state, and evidence are separated because a change in any one of them can create a different transition. The architecture therefore rejects implicit substitution: a permission for one object is not a permission for another; a proof for one model is not a proof for a deployment; an approval before mutation is not approval after mutation. Unknown mandatory state is represented explicitly and does not default to ordinary allow.

The chapter’s principal structural commitments are:

Identity And Mandate - retained as an explicit state or dependency.

Competence - retained as an explicit state or dependency.

Explicit Human Commit - retained as an explicit state or dependency.

Delegation And Veto - retained as an explicit state or dependency.

Oversight Feasibility - retained as an explicit state or dependency.

Responsibility - retained as an explicit state or dependency.

Contestability - retained as an explicit state or dependency.

63.3 Execution Reality

Execution reality begins at the last point from which the protected consequence can still be prevented. Upstream analysis, documentation, monitoring, or recommendation may improve decision quality, but it becomes governance only when the consequence path depends on the resulting state. The chapter therefore treats direct capability, alternate interfaces, administrative paths, recovery mechanisms, stale state, and external dependencies as part of the governed object. A system is not credited with a property merely because a dashboard displays it; the relevant gate, controller, institution, or physical interface must enforce or evidence it.

63.4 Failure Point and Counter-Construction

The claim collapses under any in-scope counter-construction such as:

presence is treated as consent

role is treated as competence

veto cannot reach the consequence path

responsibility is assigned without real control

These failures are not edge cases external to the theory. They are the primary falsification instruments. Where one of them remains reachable, the correct result is a narrowed claim, an unresolved state, or a failed invariant - not a narrative assertion that the architecture is nevertheless governed.

63.5 Verification and Evidence Obligations

verify mandate and competence

measure intervention feasibility

test display-commit equivalence and veto effectiveness

preserve consent, handoff, and contest evidence

Evidence must remain bound to the exact artifact, configuration, version, operating envelope, and decision state examined. A passing result establishes only the tested property and coverage. A failed or indeterminate result remains part of the authoritative record. Where independent reproduction has not occurred, the result is not represented as independently established.

63.6 Selected Propositions

P63.1. The validity of escalation, appeals, contestability, and corrective authority cannot exceed the scope of its weakest mandatory dependency.

P63.2. A descriptive or advisory representation does not become an execution constraint until it controls the relevant consequence path.

P63.3. Component-level validity does not compose automatically across interfaces, timing, authority, failure, or deployment change.

P63.4. Evidence of absence is not inferred from absence of evidence; unresolved mandatory state remains unresolved.

63.7 Governance Status and Confidence Type

STRUCTURE. Specified as part of the integrated federated architecture.
EXECUTION REALITY. Requires a domain-specific implementation and complete path inventory.
FAILURE POINT. Any surviving bypass, stale authority, omitted dependency, unmodeled timing path, or unsupported evidence link narrows or defeats the claim.
GOVERNANCE STATUS. M1 unless a chapter-specific case study states a higher internal implementation maturity; no institutional compliance determination is inferred.
CONFIDENCE TYPE. Definitional for the architecture, deductive only under stated assumptions, implementation-dependent for runtime and hardware, institutionally imported for authority, and evidence-dependent for deployment claims.

Chapter 64

Training, Competence, Qualification, and Operational Readiness

Qualification scope, recency, skill decay, simulation, drills, human-AI teaming competence, and reauthorization.

64.1 Objective

The objective of this chapter is to define training, competence, qualification, and operational readiness as a bounded component of execution-bound governance. The analysis connects identity and mandate, competence, explicit human commit, delegation and veto, oversight feasibility, responsibility, and contestability. The decisive test is not whether the system can describe the relevant rule, but whether the rule changes the set of reachable consequence-bearing transitions. The authority claim imports institutional and, where applicable, legal mandate; it does not infer legitimacy from technical access, role labels, or observed human behavior.

64.2 Structural Model

A compact representation used in this chapter is:

$$Ready = Qualified \wedge RecencyValid \wedge SystemVersionValid \wedge ToolsAvailable \wedge TeamA$$

The structural object is represented as a typed relation rather than an informal checklist. Identity, object, context, authority, version, time, fault state, and evidence are separated because a change in any one of them can create a different transition. The architecture therefore rejects implicit substitution: a permission for one object is not a permission for another; a proof for one model is not a proof for a deployment; an approval before mutation is not approval after mutation. Unknown mandatory state is represented explicitly and does not default to ordinary allow.

The chapter’s principal structural commitments are:

Identity And Mandate - retained as an explicit state or dependency.

Competence - retained as an explicit state or dependency.

Explicit Human Commit - retained as an explicit state or dependency.

Delegation And Veto - retained as an explicit state or dependency.

Oversight Feasibility - retained as an explicit state or dependency.

Responsibility - retained as an explicit state or dependency.

Contestability - retained as an explicit state or dependency.

64.3 Execution Reality

Execution reality begins at the last point from which the protected consequence can still be prevented. Upstream analysis, documentation, monitoring, or recommendation may improve decision quality, but it becomes governance only when the consequence path depends on the resulting state. The chapter therefore treats direct capability, alternate interfaces, administrative paths, recovery mechanisms, stale state, and external dependencies as part of the governed object. A system is not credited with a property merely because a dashboard displays it; the relevant gate, controller, institution, or physical interface must enforce or evidence it.

64.4 Failure Point and Counter-Construction

The claim collapses under any in-scope counter-construction such as:

presence is treated as consent

role is treated as competence

veto cannot reach the consequence path

responsibility is assigned without real control

These failures are not edge cases external to the theory. They are the primary falsification instruments. Where one of them remains reachable, the correct result is a narrowed claim, an unresolved state, or a failed invariant - not a narrative assertion that the architecture is nevertheless governed.

64.5 Verification and Evidence Obligations

verify mandate and competence

measure intervention feasibility

test display-commit equivalence and veto effectiveness

preserve consent, handoff, and contest evidence

Evidence must remain bound to the exact artifact, configuration, version, operating envelope, and decision state examined. A passing result establishes only the tested property and coverage. A failed or indeterminate result remains part of the authoritative record. Where independent reproduction has not occurred, the result is not represented as independently established.

64.6 Selected Propositions

P64.1. The validity of training, competence, qualification, and operational readiness cannot exceed the scope of its weakest mandatory dependency.

P64.2. A descriptive or advisory representation does not become an execution constraint until it controls the relevant consequence path.

P64.3. Component-level validity does not compose automatically across interfaces, timing, authority, failure, or deployment change.

P64.4. Evidence of absence is not inferred from absence of evidence; unresolved mandatory state remains unresolved.

64.7 Governance Status and Confidence Type

STRUCTURE. Specified as part of the integrated federated architecture.
EXECUTION REALITY. Requires a domain-specific implementation and complete path inventory.
FAILURE POINT. Any surviving bypass, stale authority, omitted dependency, unmodeled timing path, or unsupported evidence link narrows or defeats the claim.
GOVERNANCE STATUS. M1 unless a chapter-specific case study states a higher internal implementation maturity; no institutional compliance determination is inferred.
CONFIDENCE TYPE. Definitional for the architecture, deductive only under stated assumptions, implementation-dependent for runtime and hardware, institutionally imported for authority, and evidence-dependent for deployment claims.

Chapter 65

Institutional Continuity, Shift Handover, and Crisis Operations

Case ownership, pending decisions, succession, emergency authority, communications loss, scarcity, and crisis exit.

65.1 Objective

The objective of this chapter is to define institutional continuity, shift handover, and crisis operations as a bounded component of execution-bound governance. The analysis connects identity and mandate, competence, explicit human commit, delegation and veto, oversight feasibility, responsibility, and contestability. The decisive test is not whether the system can describe the relevant rule, but whether the rule changes the set of reachable consequence-bearing transitions. The authority claim imports institutional and, where applicable, legal mandate; it does not infer legitimacy from technical access, role labels, or observed human behavior.

65.2 Structural Model

The structural object is represented as a typed relation rather than an informal checklist. Identity, object, context, authority, version, time, fault state, and evidence are separated because a change in any one of them can create a different transition. The architecture therefore rejects implicit substitution: a permission for one object is not a permission for another; a proof for one model is not a proof for a deployment; an approval before mutation is not approval after mutation. Unknown mandatory state is represented explicitly and does not default to ordinary allow.

The chapter’s principal structural commitments are:

Identity And Mandate - retained as an explicit state or dependency.

Competence - retained as an explicit state or dependency.

Explicit Human Commit - retained as an explicit state or dependency.

Delegation And Veto - retained as an explicit state or dependency.

Oversight Feasibility - retained as an explicit state or dependency.

Responsibility - retained as an explicit state or dependency.

Contestability - retained as an explicit state or dependency.

65.3 Execution Reality

Execution reality begins at the last point from which the protected consequence can still be prevented. Upstream analysis, documentation, monitoring, or recommendation may improve decision quality, but it becomes governance only when the consequence path depends on the resulting state. The chapter therefore treats direct capability, alternate interfaces, administrative paths, recovery mechanisms, stale state, and external dependencies as part of the governed object. A system is not credited with a property merely because a dashboard displays it; the relevant gate, controller, institution, or physical interface must enforce or evidence it.

65.4 Failure Point and Counter-Construction

The claim collapses under any in-scope counter-construction such as:

presence is treated as consent

role is treated as competence

veto cannot reach the consequence path

responsibility is assigned without real control

These failures are not edge cases external to the theory. They are the primary falsification instruments. Where one of them remains reachable, the correct result is a narrowed claim, an unresolved state, or a failed invariant - not a narrative assertion that the architecture is nevertheless governed.

65.5 Verification and Evidence Obligations

verify mandate and competence

measure intervention feasibility

test display-commit equivalence and veto effectiveness

preserve consent, handoff, and contest evidence

Evidence must remain bound to the exact artifact, configuration, version, operating envelope, and decision state examined. A passing result establishes only the tested property and coverage. A failed or indeterminate result remains part of the authoritative record. Where independent reproduction has not occurred, the result is not represented as independently established.

65.6 Selected Propositions

P65.1. The validity of institutional continuity, shift handover, and crisis operations cannot exceed the scope of its weakest mandatory dependency.

P65.2. A descriptive or advisory representation does not become an execution constraint until it controls the relevant consequence path.

P65.3. Component-level validity does not compose automatically across interfaces, timing, authority, failure, or deployment change.

P65.4. Evidence of absence is not inferred from absence of evidence; unresolved mandatory state remains unresolved.

65.7 Governance Status and Confidence Type

STRUCTURE. Specified as part of the integrated federated architecture.
EXECUTION REALITY. Requires a domain-specific implementation and complete path inventory.
FAILURE POINT. Any surviving bypass, stale authority, omitted dependency, unmodeled timing path, or unsupported evidence link narrows or defeats the claim.
GOVERNANCE STATUS. M1 unless a chapter-specific case study states a higher internal implementation maturity; no institutional compliance determination is inferred.
CONFIDENCE TYPE. Definitional for the architecture, deductive only under stated assumptions, implementation-dependent for runtime and hardware, institutionally imported for authority, and evidence-dependent for deployment claims.

Chapter 66

Human-Authority Volume Synthesis and Closure

The unified authority object, master authority predicate, oversight feasibility, consent, responsibility, contestability, and continuity.

66.1 Objective

The objective of this chapter is to define human-authority volume synthesis and closure as a bounded component of execution-bound governance. The analysis connects identity and mandate, competence, explicit human commit, delegation and veto, oversight feasibility, responsibility, and contestability. The decisive test is not whether the system can describe the relevant rule, but whether the rule changes the set of reachable consequence-bearing transitions. The authority claim imports institutional and, where applicable, legal mandate; it does not infer legitimacy from technical access, role labels, or observed human behavior.

66.2 Structural Model

A compact representation used in this chapter is:

$$HumanAuthorizedCommit \Rightarrow Permit_A$$

The structural object is represented as a typed relation rather than an informal checklist. Identity, object, context, authority, version, time, fault state, and evidence are separated because a change in any one of them can create a different transition. The architecture therefore rejects implicit substitution: a permission for one object is not a permission for another; a proof for one model is not a proof for a deployment; an approval before mutation is not approval after mutation. Unknown mandatory state is represented explicitly and does not default to ordinary allow.

The chapter's principal structural commitments are:

Identity And Mandate - retained as an explicit state or dependency.

Competence - retained as an explicit state or dependency.

Explicit Human Commit - retained as an explicit state or dependency.

Delegation And Veto - retained as an explicit state or dependency.

Oversight Feasibility - retained as an explicit state or dependency.

Responsibility - retained as an explicit state or dependency.

Contestability - retained as an explicit state or dependency.

66.3 Execution Reality

Execution reality begins at the last point from which the protected consequence can still be prevented. Upstream analysis, documentation, monitoring, or recommendation may improve decision quality, but it becomes governance only when the consequence path depends on the resulting state. The chapter therefore treats direct capability, alternate interfaces, administrative paths, recovery mechanisms, stale state, and external dependencies as part of the governed object. A system is not credited with a property merely because a dashboard displays it; the relevant gate, controller, institution, or physical interface must enforce or evidence it.

66.4 Failure Point and Counter-Construction

The claim collapses under any in-scope counter-construction such as:

presence is treated as consent

role is treated as competence

veto cannot reach the consequence path

responsibility is assigned without real control

These failures are not edge cases external to the theory. They are the primary falsification instruments. Where one of them remains reachable, the correct result is a narrowed claim, an unresolved state, or a failed invariant - not a narrative assertion that the architecture is nevertheless governed.

66.5 Verification and Evidence Obligations

verify mandate and competence

measure intervention feasibility

test display-commit equivalence and veto effectiveness

preserve consent, handoff, and contest evidence

Evidence must remain bound to the exact artifact, configuration, version, operating envelope, and decision state examined. A passing result establishes only the tested property and coverage. A failed or indeterminate result remains part of the authoritative record. Where independent reproduction has not occurred, the result is not represented as independently established.

66.6 Selected Propositions

P66.1. The validity of human-authority volume synthesis and closure cannot exceed the scope of its weakest mandatory dependency.

P66.2. A descriptive or advisory representation does not become an execution constraint until it controls the relevant consequence path.

P66.3. Component-level validity does not compose automatically across interfaces, timing, authority, failure, or deployment change.

P66.4. Evidence of absence is not inferred from absence of evidence; unresolved mandatory state remains unresolved.

66.7 Governance Status and Confidence Type

STRUCTURE. Specified as part of the integrated federated architecture.
EXECUTION REALITY. Requires a domain-specific implementation and complete path inventory.
FAILURE POINT. Any surviving bypass, stale authority, omitted dependency, unmodeled timing path, or unsupported evidence link narrows or defeats the claim.
GOVERNANCE STATUS. M1 unless a chapter-specific case study states a higher internal implementation maturity; no institutional compliance determination is inferred.
CONFIDENCE TYPE. Definitional for the architecture, deductive only under stated assumptions, implementation-dependent for runtime and hardware, institutionally imported for authority, and evidence-dependent for deployment claims.

PART VII - GOVERNANCE, LAW, AND INSTITUTIONAL ASSURANCE

Part VII connects external normative authority to executable institutional controls. It keeps legal interpretation, technical enforcement, risk, audit, procurement, data, incidents, assurance, and disclosure distinct while making their dependencies traceable.

Part-level claim boundary. The material in this part is specified at M1 unless a bounded case study explicitly records internal implementation evidence. No part-level statement independently widens the system-level claim.

Chapter 67

Normative Rules, Legal Authority, and the Translation from Obligation to Executable Constraint

Sources, applicability, interpretation authority, formal requirements, architecture mapping, and governance gaps.

67.1 Objective

The objective of this chapter is to define normative rules, legal authority, and the translation from obligation to executable constraint as a bounded component of execution-bound governance. The analysis connects normative source, authorized interpretation, rights and duties, hard enforcement, risk and evidence, supplier and data dependency, and remedy. The decisive test is not whether the system can describe the relevant rule, but whether the rule changes the set of reachable consequence-bearing transitions. The governance claim is a structured translation claim and not an independent legal determination; current primary sources and competent institutional interpretation remain mandatory.

67.2 Structural Model

A compact representation used in this chapter is:

$$N_S \rightarrow I_N \rightarrow R_N \rightarrow \mu_{NA} \rightarrow G_X \rightarrow E_N$$

The structural object is represented as a typed relation rather than an informal checklist. Identity, object, context, authority, version, time, fault state, and evidence are separated because a change in any one of them can create a different transition. The architecture therefore rejects implicit substitution: a permission for one object is not a permission for another; a proof for one model is not a proof for a deployment; an approval before mutation is not approval after mutation. Unknown mandatory state is represented explicitly and does not default to ordinary allow.

The chapter’s principal structural commitments are:

Normative Source - retained as an explicit state or dependency.

Authorized Interpretation - retained as an explicit state or dependency.

Rights And Duties - retained as an explicit state or dependency.

Hard Enforcement - retained as an explicit state or dependency.

Risk And Evidence - retained as an explicit state or dependency.

Supplier And Data Dependency - retained as an explicit state or dependency.

Remedy - retained as an explicit state or dependency.

67.3 Execution Reality

Execution reality begins at the last point from which the protected consequence can still be prevented. Upstream analysis, documentation, monitoring, or recommendation may improve decision quality, but it becomes governance only when the consequence path depends on the resulting state. The chapter therefore treats direct capability, alternate interfaces, administrative paths, recovery mechanisms, stale state, and external dependencies as part of the governed object. A system is not credited with a property merely because a dashboard displays it; the relevant gate, controller, institution, or physical interface must enforce or evidence it.

67.4 Failure Point and Counter-Construction

The claim collapses under any in-scope counter-construction such as:

legal text is reduced to a policy label

risk score overrides a prohibition

supplier access becomes shadow authority

a certificate is generalized beyond its scope

These failures are not edge cases external to the theory. They are the primary falsification instruments. Where one of them remains reachable, the correct result is a narrowed claim, an unresolved state, or a failed invariant - not a narrative assertion that the architecture is nevertheless governed.

67.5 Verification and Evidence Obligations

verify the current authoritative source

trace interpretation to executable control

test supplier, data, change, incident, and remedy paths

bound every assurance statement to scope

Evidence must remain bound to the exact artifact, configuration, version, operating envelope, and decision state examined. A passing result establishes only the tested property and coverage. A failed or indeterminate result remains part of the authoritative record. Where independent reproduction has not occurred, the result is not represented as independently established.

67.6 Selected Propositions

P67.1. The validity of normative rules, legal authority, and the translation from obligation to executable constraint cannot exceed the scope of its weakest mandatory dependency.

P67.2. A descriptive or advisory representation does not become an execution constraint until it controls the relevant consequence path.

P67.3. Component-level validity does not compose automatically across interfaces, timing, authority, failure, or deployment change.

P67.4. Evidence of absence is not inferred from absence of evidence; unresolved mandatory state remains unresolved.

67.7 Governance Status and Confidence Type

STRUCTURE. Specified as part of the integrated federated architecture.
EXECUTION REALITY. Requires a domain-specific implementation and complete path inventory.
FAILURE POINT. Any surviving bypass, stale authority, omitted dependency, unmodeled timing path, or unsupported evidence link narrows or defeats the claim.
GOVERNANCE STATUS. M1 unless a chapter-specific case study states a higher internal implementation maturity; no institutional compliance determination is inferred.
CONFIDENCE TYPE. Definitional for the architecture, deductive only under stated assumptions, implementation-dependent for runtime and hardware, institutionally imported for authority, and evidence-dependent for deployment claims.

Chapter 68

Rights, Duties, Prohibitions, Exceptions, and Conflict of Norms

Deontic relations, rights holders, duty bearers, exceptions, overrides, impossibility, proportionality, and remedy.

68.1 Objective

The objective of this chapter is to define rights, duties, prohibitions, exceptions, and conflict of norms as a bounded component of execution-bound governance. The analysis connects normative source, authorized interpretation, rights and duties, hard enforcement, risk and evidence, supplier and data dependency, and remedy. The decisive test is not whether the system can describe the relevant rule, but whether the rule changes the set of reachable consequence-bearing transitions. The governance claim is a structured translation claim and not an independent legal determination; current primary sources and competent institutional interpretation remain mandatory.

68.2 Structural Model

A compact representation used in this chapter is:

$$O(a) \wedge O(\neg a) \Rightarrow conflict$$

The structural object is represented as a typed relation rather than an informal checklist. Identity, object, context, authority, version, time, fault state, and evidence are separated because a change in any one of them can create a different transition. The architecture therefore rejects implicit substitution: a permission for one object is not a permission for another; a proof for one model is not a proof for a deployment; an approval before mutation is not approval after mutation. Unknown mandatory state is represented explicitly and does not default to ordinary allow.

The chapter’s principal structural commitments are:

Normative Source - retained as an explicit state or dependency.

Authorized Interpretation - retained as an explicit state or dependency.

Rights And Duties - retained as an explicit state or dependency.

Hard Enforcement - retained as an explicit state or dependency.

Risk And Evidence - retained as an explicit state or dependency.

Supplier And Data Dependency - retained as an explicit state or dependency.

Remedy - retained as an explicit state or dependency.

68.3 Execution Reality

Execution reality begins at the last point from which the protected consequence can still be prevented. Upstream analysis, documentation, monitoring, or recommendation may improve decision quality, but it becomes governance only when the consequence path depends on the resulting state. The chapter therefore treats direct capability, alternate interfaces, administrative paths, recovery mechanisms, stale state, and external dependencies as part of the governed object. A system is not credited with a property merely because a dashboard displays it; the relevant gate, controller, institution, or physical interface must enforce or evidence it.

68.4 Failure Point and Counter-Construction

The claim collapses under any in-scope counter-construction such as:

legal text is reduced to a policy label

risk score overrides a prohibition

supplier access becomes shadow authority

a certificate is generalized beyond its scope

These failures are not edge cases external to the theory. They are the primary falsification instruments. Where one of them remains reachable, the correct result is a narrowed claim, an unresolved state, or a failed invariant - not a narrative assertion that the architecture is nevertheless governed.

68.5 Verification and Evidence Obligations

verify the current authoritative source

trace interpretation to executable control

test supplier, data, change, incident, and remedy paths

bound every assurance statement to scope

Evidence must remain bound to the exact artifact, configuration, version, operating envelope, and decision state examined. A passing result establishes only the tested property and coverage. A failed or indeterminate result remains part of the authoritative record. Where independent reproduction has not occurred, the result is not represented as independently established.

68.6 Selected Propositions

P68.1. The validity of rights, duties, prohibitions, exceptions, and conflict of norms cannot exceed the scope of its weakest mandatory dependency.

P68.2. A descriptive or advisory representation does not become an execution constraint until it controls the relevant consequence path.

P68.3. Component-level validity does not compose automatically across interfaces, timing, authority, failure, or deployment change.

P68.4. Evidence of absence is not inferred from absence of evidence; unresolved mandatory state remains unresolved.

68.7 Governance Status and Confidence Type

STRUCTURE. Specified as part of the integrated federated architecture.
EXECUTION REALITY. Requires a domain-specific implementation and complete path inventory.
FAILURE POINT. Any surviving bypass, stale authority, omitted dependency, unmodeled timing path, or unsupported evidence link narrows or defeats the claim.
GOVERNANCE STATUS. M1 unless a chapter-specific case study states a higher internal implementation maturity; no institutional compliance determination is inferred.
CONFIDENCE TYPE. Definitional for the architecture, deductive only under stated assumptions, implementation-dependent for runtime and hardware, institutionally imported for authority, and evidence-dependent for deployment claims.

Chapter 69

Risk Classification, Assurance Levels, and Governance Claim Envelopes

Hazard, consequence, likelihood, structural uncertainty, hard admissibility, residual risk, acceptance authority, and evidence ceilings.

69.1 Objective

The objective of this chapter is to define risk classification, assurance levels, and governance claim envelopes as a bounded component of execution-bound governance. The analysis connects normative source, authorized interpretation, rights and duties, hard enforcement, risk and evidence, supplier and data dependency, and remedy. The decisive test is not whether the system can describe the relevant rule, but whether the rule changes the set of reachable consequence-bearing transitions. The governance claim is a structured translation claim and not an independent legal determination; current primary sources and competent institutional interpretation remain mandatory.

69.2 Structural Model

A compact representation used in this chapter is:

$$U_{admissible} = U_{hard} \cap U_{authority} \cap U_{safety}$$

The structural object is represented as a typed relation rather than an informal checklist. Identity, object, context, authority, version, time, fault state, and evidence are separated because a change in any one of them can create a different transition. The architecture therefore rejects implicit substitution: a permission for one object is not a permission for another; a proof for one model is not a proof for a deployment; an approval before mutation is not approval after mutation. Unknown mandatory state is represented explicitly and does not default to ordinary allow.

The chapter’s principal structural commitments are:

Normative Source - retained as an explicit state or dependency.

Authorized Interpretation - retained as an explicit state or dependency.

Rights And Duties - retained as an explicit state or dependency.

Hard Enforcement - retained as an explicit state or dependency.

Risk And Evidence - retained as an explicit state or dependency.

Supplier And Data Dependency - retained as an explicit state or dependency.

Remedy - retained as an explicit state or dependency.

69.3 Execution Reality

Execution reality begins at the last point from which the protected consequence can still be prevented. Upstream analysis, documentation, monitoring, or recommendation may improve decision quality, but it becomes governance only when the consequence path depends on the resulting state. The chapter therefore treats direct capability, alternate interfaces, administrative paths, recovery mechanisms, stale state, and external dependencies as part of the governed object. A system is not credited with a property merely because a dashboard displays it; the relevant gate, controller, institution, or physical interface must enforce or evidence it.

69.4 Failure Point and Counter-Construction

The claim collapses under any in-scope counter-construction such as:

legal text is reduced to a policy label

risk score overrides a prohibition

supplier access becomes shadow authority

a certificate is generalized beyond its scope

These failures are not edge cases external to the theory. They are the primary falsification instruments. Where one of them remains reachable, the correct result is a narrowed claim, an unresolved state, or a failed invariant - not a narrative assertion that the architecture is nevertheless governed.

69.5 Verification and Evidence Obligations

verify the current authoritative source

trace interpretation to executable control

test supplier, data, change, incident, and remedy paths

bound every assurance statement to scope

Evidence must remain bound to the exact artifact, configuration, version, operating envelope, and decision state examined. A passing result establishes only the tested property and coverage. A failed or indeterminate result remains part of the authoritative record. Where independent reproduction has not occurred, the result is not represented as independently established.

69.6 Selected Propositions

P69.1. The validity of risk classification, assurance levels, and governance claim envelopes cannot exceed the scope of its weakest mandatory dependency.

P69.2. A descriptive or advisory representation does not become an execution constraint until it controls the relevant consequence path.

P69.3. Component-level validity does not compose automatically across interfaces, timing, authority, failure, or deployment change.

P69.4. Evidence of absence is not inferred from absence of evidence; unresolved mandatory state remains unresolved.

69.7 Governance Status and Confidence Type

STRUCTURE. Specified as part of the integrated federated architecture.
EXECUTION REALITY. Requires a domain-specific implementation and complete path inventory.
FAILURE POINT. Any surviving bypass, stale authority, omitted dependency, unmodeled timing path, or unsupported evidence link narrows or defeats the claim.
GOVERNANCE STATUS. M1 unless a chapter-specific case study states a higher internal implementation maturity; no institutional compliance determination is inferred.
CONFIDENCE TYPE. Definitional for the architecture, deductive only under stated assumptions, implementation-dependent for runtime and hardware, institutionally imported for authority, and evidence-dependent for deployment claims.

Chapter 70

Documentation, Auditability, and Institutional Evidence

Decision files, provenance, audit scope, sampling, independence, findings, corrective action, retention, holds, and closure.

70.1 Objective

The objective of this chapter is to define documentation, auditability, and institutional evidence as a bounded component of execution-bound governance. The analysis connects normative source, authorized interpretation, rights and duties, hard enforcement, risk and evidence, supplier and data dependency, and remedy. The decisive test is not whether the system can describe the relevant rule, but whether the rule changes the set of reachable consequence-bearing transitions. The governance claim is a structured translation claim and not an independent legal determination; current primary sources and competent institutional interpretation remain mandatory.

70.2 Structural Model

A compact representation used in this chapter is:

$$Criterion \rightarrow Evidence \rightarrow Finding \rightarrow Action \rightarrow Verification \rightarrow Closure$$

The structural object is represented as a typed relation rather than an informal checklist. Identity, object, context, authority, version, time, fault state, and evidence are separated because a change in any one of them can create a different transition. The architecture therefore rejects implicit substitution: a permission for one object is not a permission for another; a proof for one model is not a proof for a deployment; an approval before mutation is not approval after mutation. Unknown mandatory state is represented explicitly and does not default to ordinary allow.

The chapter’s principal structural commitments are:

Normative Source - retained as an explicit state or dependency.

Authorized Interpretation - retained as an explicit state or dependency.

Rights And Duties - retained as an explicit state or dependency.

Hard Enforcement - retained as an explicit state or dependency.

Risk And Evidence - retained as an explicit state or dependency.

Supplier And Data Dependency - retained as an explicit state or dependency.

Remedy - retained as an explicit state or dependency.

70.3 Execution Reality

Execution reality begins at the last point from which the protected consequence can still be prevented. Upstream analysis, documentation, monitoring, or recommendation may improve decision quality, but it becomes governance only when the consequence path depends on the resulting state. The chapter therefore treats direct capability, alternate interfaces, administrative paths, recovery mechanisms, stale state, and external dependencies as part of the governed object. A system is not credited with a property merely because a dashboard displays it; the relevant gate, controller, institution, or physical interface must enforce or evidence it.

70.4 Failure Point and Counter-Construction

The claim collapses under any in-scope counter-construction such as:

legal text is reduced to a policy label

risk score overrides a prohibition

supplier access becomes shadow authority

a certificate is generalized beyond its scope

These failures are not edge cases external to the theory. They are the primary falsification instruments. Where one of them remains reachable, the correct result is a narrowed claim, an unresolved state, or a failed invariant - not a narrative assertion that the architecture is nevertheless governed.

70.5 Verification and Evidence Obligations

verify the current authoritative source

trace interpretation to executable control

test supplier, data, change, incident, and remedy paths

bound every assurance statement to scope

Evidence must remain bound to the exact artifact, configuration, version, operating envelope, and decision state examined. A passing result establishes only the tested property and coverage. A failed or indeterminate result remains part of the authoritative record. Where independent reproduction has not occurred, the result is not represented as independently established.

70.6 Selected Propositions

P70.1. The validity of documentation, auditability, and institutional evidence cannot exceed the scope of its weakest mandatory dependency.

P70.2. A descriptive or advisory representation does not become an execution constraint until it controls the relevant consequence path.

P70.3. Component-level validity does not compose automatically across interfaces, timing, authority, failure, or deployment change.

P70.4. Evidence of absence is not inferred from absence of evidence; unresolved mandatory state remains unresolved.

70.7 Governance Status and Confidence Type

STRUCTURE. Specified as part of the integrated federated architecture.
EXECUTION REALITY. Requires a domain-specific implementation and complete path inventory.
FAILURE POINT. Any surviving bypass, stale authority, omitted dependency, unmodeled timing path, or unsupported evidence link narrows or defeats the claim.
GOVERNANCE STATUS. M1 unless a chapter-specific case study states a higher internal implementation maturity; no institutional compliance determination is inferred.
CONFIDENCE TYPE. Definitional for the architecture, deductive only under stated assumptions, implementation-dependent for runtime and hardware, institutionally imported for authority, and evidence-dependent for deployment claims.

Chapter 71

Procurement, Contracts, Vendors, Licensing, and External Dependencies

Supplier authority, contractual versus technical control, external permission dependency, lock-in, exit, portability, escrow, and license boundaries.

71.1 Objective

The objective of this chapter is to define procurement, contracts, vendors, licensing, and external dependencies as a bounded component of execution-bound governance. The analysis connects normative source, authorized interpretation, rights and duties, hard enforcement, risk and evidence, supplier and data dependency, and remedy. The decisive test is not whether the system can describe the relevant rule, but whether the rule changes the set of reachable consequence-bearing transitions. The governance claim is a structured translation claim and not an independent legal determination; current primary sources and competent institutional interpretation remain mandatory.

71.2 Structural Model

A compact representation used in this chapter is:

$$\text{external discretionary permission} \Rightarrow \text{bounded institutional sovereignty}$$

The structural object is represented as a typed relation rather than an informal checklist. Identity, object, context, authority, version, time, fault state, and evidence are separated because a change in any one of them can create a different transition. The architecture therefore rejects implicit substitution: a permission for one object is not a permission for another; a proof for one model is not a proof for a deployment; an approval before mutation is not approval after mutation. Unknown mandatory state is represented explicitly and does not default to ordinary allow.

The chapter’s principal structural commitments are:

Normative Source - retained as an explicit state or dependency.

Authorized Interpretation - retained as an explicit state or dependency.

Rights And Duties - retained as an explicit state or dependency.

Hard Enforcement - retained as an explicit state or dependency.

Risk And Evidence - retained as an explicit state or dependency.

Supplier And Data Dependency - retained as an explicit state or dependency.

Remedy - retained as an explicit state or dependency.

71.3 Execution Reality

Execution reality begins at the last point from which the protected consequence can still be prevented. Upstream analysis, documentation, monitoring, or recommendation may improve decision quality, but it becomes governance only when the consequence path depends on the resulting state. The chapter therefore treats direct capability, alternate interfaces, administrative paths, recovery mechanisms, stale state, and external dependencies as part of the governed object. A system is not credited with a property merely because a dashboard displays it; the relevant gate, controller, institution, or physical interface must enforce or evidence it.

71.4 Failure Point and Counter-Construction

The claim collapses under any in-scope counter-construction such as:

legal text is reduced to a policy label

risk score overrides a prohibition

supplier access becomes shadow authority

a certificate is generalized beyond its scope

These failures are not edge cases external to the theory. They are the primary falsification instruments. Where one of them remains reachable, the correct result is a narrowed claim, an unresolved state, or a failed invariant - not a narrative assertion that the architecture is nevertheless governed.

71.5 Verification and Evidence Obligations

verify the current authoritative source

trace interpretation to executable control

test supplier, data, change, incident, and remedy paths

bound every assurance statement to scope

Evidence must remain bound to the exact artifact, configuration, version, operating envelope, and decision state examined. A passing result establishes only the tested property and coverage. A failed or indeterminate result remains part of the authoritative record. Where independent reproduction has not occurred, the result is not represented as independently established.

71.6 Selected Propositions

P71.1. The validity of procurement, contracts, vendors, licensing, and external dependencies cannot exceed the scope of its weakest mandatory dependency.

P71.2. A descriptive or advisory representation does not become an execution constraint until it controls the relevant consequence path.

P71.3. Component-level validity does not compose automatically across interfaces, timing, authority, failure, or deployment change.

P71.4. Evidence of absence is not inferred from absence of evidence; unresolved mandatory state remains unresolved.

71.7 Governance Status and Confidence Type

STRUCTURE. Specified as part of the integrated federated architecture.
EXECUTION REALITY. Requires a domain-specific implementation and complete path inventory.
FAILURE POINT. Any surviving bypass, stale authority, omitted dependency, unmodeled timing path, or unsupported evidence link narrows or defeats the claim.
GOVERNANCE STATUS. M1 unless a chapter-specific case study states a higher internal implementation maturity; no institutional compliance determination is inferred.
CONFIDENCE TYPE. Definitional for the architecture, deductive only under stated assumptions, implementation-dependent for runtime and hardware, institutionally imported for authority, and evidence-dependent for deployment claims.

Chapter 72

Data Governance, Provenance, Privacy, Purpose, Access, and Retention

Data identity, collection authority, purpose, minimization, quality, disclosure, derived data, deletion, and lineage.

72.1 Objective

The objective of this chapter is to define data governance, provenance, privacy, purpose, access, and retention as a bounded component of execution-bound governance. The analysis connects normative source, authorized interpretation, rights and duties, hard enforcement, risk and evidence, supplier and data dependency, and remedy. The decisive test is not whether the system can describe the relevant rule, but whether the rule changes the set of reachable consequence-bearing transitions. The governance claim is a structured translation claim and not an independent legal determination; current primary sources and competent institutional interpretation remain mandatory.

72.2 Structural Model

A compact representation used in this chapter is:

$$AccessValid = PrincipalValid \wedge PurposeValid \wedge ScopeValid \wedge TimeValid$$

The structural object is represented as a typed relation rather than an informal checklist. Identity, object, context, authority, version, time, fault state, and evidence are separated because a change in any one of them can create a different transition. The architecture therefore rejects implicit substitution: a permission for one object is not a permission for another; a proof for one model is not a proof for a deployment; an approval before mutation is not approval after mutation. Unknown mandatory state is represented explicitly and does not default to ordinary allow.

The chapter’s principal structural commitments are:

Normative Source - retained as an explicit state or dependency.

Authorized Interpretation - retained as an explicit state or dependency.

Rights And Duties - retained as an explicit state or dependency.

Hard Enforcement - retained as an explicit state or dependency.

Risk And Evidence - retained as an explicit state or dependency.

Supplier And Data Dependency - retained as an explicit state or dependency.

Remedy - retained as an explicit state or dependency.

72.3 Execution Reality

Execution reality begins at the last point from which the protected consequence can still be prevented. Upstream analysis, documentation, monitoring, or recommendation may improve decision quality, but it becomes governance only when the consequence path depends on the resulting state. The chapter therefore treats direct capability, alternate interfaces, administrative paths, recovery mechanisms, stale state, and external dependencies as part of the governed object. A system is not credited with a property merely because a dashboard displays it; the relevant gate, controller, institution, or physical interface must enforce or evidence it.

72.4 Failure Point and Counter-Construction

The claim collapses under any in-scope counter-construction such as:

legal text is reduced to a policy label

risk score overrides a prohibition

supplier access becomes shadow authority

a certificate is generalized beyond its scope

These failures are not edge cases external to the theory. They are the primary falsification instruments. Where one of them remains reachable, the correct result is a narrowed claim, an unresolved state, or a failed invariant - not a narrative assertion that the architecture is nevertheless governed.

72.5 Verification and Evidence Obligations

verify the current authoritative source

trace interpretation to executable control

test supplier, data, change, incident, and remedy paths

bound every assurance statement to scope

Evidence must remain bound to the exact artifact, configuration, version, operating envelope, and decision state examined. A passing result establishes only the tested property and coverage. A failed or indeterminate result remains part of the authoritative record. Where independent reproduction has not occurred, the result is not represented as independently established.

72.6 Selected Propositions

P72.1. The validity of data governance, provenance, privacy, purpose, access, and retention cannot exceed the scope of its weakest mandatory dependency.

P72.2. A descriptive or advisory representation does not become an execution constraint until it controls the relevant consequence path.

P72.3. Component-level validity does not compose automatically across interfaces, timing, authority, failure, or deployment change.

P72.4. Evidence of absence is not inferred from absence of evidence; unresolved mandatory state remains unresolved.

72.7 Governance Status and Confidence Type

STRUCTURE. Specified as part of the integrated federated architecture.
EXECUTION REALITY. Requires a domain-specific implementation and complete path inventory.
FAILURE POINT. Any surviving bypass, stale authority, omitted dependency, unmodeled timing path, or unsupported evidence link narrows or defeats the claim.
GOVERNANCE STATUS. M1 unless a chapter-specific case study states a higher internal implementation maturity; no institutional compliance determination is inferred.
CONFIDENCE TYPE. Definitional for the architecture, deductive only under stated assumptions, implementation-dependent for runtime and hardware, institutionally imported for authority, and evidence-dependent for deployment claims.

Chapter 73

Policy Versioning, Change Control, Regulatory Drift, and Reauthorization

Policy artifacts, effective dates, supersession, impact analysis, emergency change, deployment, rollback, and interpretation drift.

73.1 Objective

The objective of this chapter is to define policy versioning, change control, regulatory drift, and reauthorization as a bounded component of execution-bound governance. The analysis connects normative source, authorized interpretation, rights and duties, hard enforcement, risk and evidence, supplier and data dependency, and remedy. The decisive test is not whether the system can describe the relevant rule, but whether the rule changes the set of reachable consequence-bearing transitions. The governance claim is a structured translation claim and not an independent legal determination; current primary sources and competent institutional interpretation remain mandatory.

73.2 Structural Model

A compact representation used in this chapter is:

$$SourceChange \rightarrow InterpretationChange \rightarrow PolicyChange \rightarrow ControlChange \rightarrow Trainin$$

The structural object is represented as a typed relation rather than an informal checklist. Identity, object, context, authority, version, time, fault state, and evidence are separated because a change in any one of them can create a different transition. The architecture therefore rejects implicit substitution: a permission for one object is not a permission for another; a proof for one model is not a proof for a deployment; an approval before mutation is not approval after mutation. Unknown mandatory state is represented explicitly and does not default to ordinary allow.

The chapter’s principal structural commitments are:

Normative Source - retained as an explicit state or dependency.

Authorized Interpretation - retained as an explicit state or dependency.

Rights And Duties - retained as an explicit state or dependency.

Hard Enforcement - retained as an explicit state or dependency.

Risk And Evidence - retained as an explicit state or dependency.

Supplier And Data Dependency - retained as an explicit state or dependency.

Remedy - retained as an explicit state or dependency.

73.3 Execution Reality

Execution reality begins at the last point from which the protected consequence can still be prevented. Upstream analysis, documentation, monitoring, or recommendation may improve decision quality, but it becomes governance only when the consequence path depends on the resulting state. The chapter therefore treats direct capability, alternate interfaces, administrative paths, recovery mechanisms, stale state, and external dependencies as part of the governed object. A system is not credited with a property merely because a dashboard displays it; the relevant gate, controller, institution, or physical interface must enforce or evidence it.

73.4 Failure Point and Counter-Construction

The claim collapses under any in-scope counter-construction such as:

legal text is reduced to a policy label

risk score overrides a prohibition

supplier access becomes shadow authority

a certificate is generalized beyond its scope

These failures are not edge cases external to the theory. They are the primary falsification instruments. Where one of them remains reachable, the correct result is a narrowed claim, an unresolved state, or a failed invariant - not a narrative assertion that the architecture is nevertheless governed.

73.5 Verification and Evidence Obligations

verify the current authoritative source

trace interpretation to executable control

test supplier, data, change, incident, and remedy paths

bound every assurance statement to scope

Evidence must remain bound to the exact artifact, configuration, version, operating envelope, and decision state examined. A passing result establishes only the tested property and coverage. A failed or indeterminate result remains part of the authoritative record. Where independent reproduction has not occurred, the result is not represented as independently established.

73.6 Selected Propositions

P73.1. The validity of policy versioning, change control, regulatory drift, and reauthorization cannot exceed the scope of its weakest mandatory dependency.

P73.2. A descriptive or advisory representation does not become an execution constraint until it controls the relevant consequence path.

P73.3. Component-level validity does not compose automatically across interfaces, timing, authority, failure, or deployment change.

P73.4. Evidence of absence is not inferred from absence of evidence; unresolved mandatory state remains unresolved.

73.7 Governance Status and Confidence Type

STRUCTURE. Specified as part of the integrated federated architecture.
EXECUTION REALITY. Requires a domain-specific implementation and complete path inventory.
FAILURE POINT. Any surviving bypass, stale authority, omitted dependency, unmodeled timing path, or unsupported evidence link narrows or defeats the claim.
GOVERNANCE STATUS. M1 unless a chapter-specific case study states a higher internal implementation maturity; no institutional compliance determination is inferred.
CONFIDENCE TYPE. Definitional for the architecture, deductive only under stated assumptions, implementation-dependent for runtime and hardware, institutionally imported for authority, and evidence-dependent for deployment claims.

Chapter 74

Incidents, Breach Response, Corrective Action, Suspension, Withdrawal, and Recall

Detection, near misses, containment, evidence holds, corrective and preventive action, recovery, and reauthorization.

74.1 Objective

The objective of this chapter is to define incidents, breach response, corrective action, suspension, withdrawal, and recall as a bounded component of execution-bound governance. The analysis connects normative source, authorized interpretation, rights and duties, hard enforcement, risk and evidence, supplier and data dependency, and remedy. The decisive test is not whether the system can describe the relevant rule, but whether the rule changes the set of reachable consequence-bearing transitions. The governance claim is a structured translation claim and not an independent legal determination; current primary sources and competent institutional interpretation remain mandatory.

74.2 Structural Model

A compact representation used in this chapter is:

$$detected \rightarrow contained \rightarrow correcting \rightarrow validating \rightarrow closed$$

The structural object is represented as a typed relation rather than an informal checklist. Identity, object, context, authority, version, time, fault state, and evidence are separated because a change in any one of them can create a different transition. The architecture therefore rejects implicit substitution: a permission for one object is not a permission for another; a proof for one model is not a proof for a deployment; an approval before mutation is not approval after mutation. Unknown mandatory state is represented explicitly and does not default to ordinary allow.

The chapter’s principal structural commitments are:

Normative Source - retained as an explicit state or dependency.

Authorized Interpretation - retained as an explicit state or dependency.

Rights And Duties - retained as an explicit state or dependency.

Hard Enforcement - retained as an explicit state or dependency.

Risk And Evidence - retained as an explicit state or dependency.

Supplier And Data Dependency - retained as an explicit state or dependency.

Remedy - retained as an explicit state or dependency.

74.3 Execution Reality

Execution reality begins at the last point from which the protected consequence can still be prevented. Upstream analysis, documentation, monitoring, or recommendation may improve decision quality, but it becomes governance only when the consequence path depends on the resulting state. The chapter therefore treats direct capability, alternate interfaces, administrative paths, recovery mechanisms, stale state, and external dependencies as part of the governed object. A system is not credited with a property merely because a dashboard displays it; the relevant gate, controller, institution, or physical interface must enforce or evidence it.

74.4 Failure Point and Counter-Construction

The claim collapses under any in-scope counter-construction such as:

legal text is reduced to a policy label

risk score overrides a prohibition

supplier access becomes shadow authority

a certificate is generalized beyond its scope

These failures are not edge cases external to the theory. They are the primary falsification instruments. Where one of them remains reachable, the correct result is a narrowed claim, an unresolved state, or a failed invariant - not a narrative assertion that the architecture is nevertheless governed.

74.5 Verification and Evidence Obligations

verify the current authoritative source

trace interpretation to executable control

test supplier, data, change, incident, and remedy paths

bound every assurance statement to scope

Evidence must remain bound to the exact artifact, configuration, version, operating envelope, and decision state examined. A passing result establishes only the tested property and coverage. A failed or indeterminate result remains part of the authoritative record. Where independent reproduction has not occurred, the result is not represented as independently established.

74.6 Selected Propositions

P74.1. The validity of incidents, breach response, corrective action, suspension, withdrawal, and recall cannot exceed the scope of its weakest mandatory dependency.

P74.2. A descriptive or advisory representation does not become an execution constraint until it controls the relevant consequence path.

P74.3. Component-level validity does not compose automatically across interfaces, timing, authority, failure, or deployment change.

P74.4. Evidence of absence is not inferred from absence of evidence; unresolved mandatory state remains unresolved.

74.7 Governance Status and Confidence Type

STRUCTURE. Specified as part of the integrated federated architecture.
EXECUTION REALITY. Requires a domain-specific implementation and complete path inventory.
FAILURE POINT. Any surviving bypass, stale authority, omitted dependency, unmodeled timing path, or unsupported evidence link narrows or defeats the claim.
GOVERNANCE STATUS. M1 unless a chapter-specific case study states a higher internal implementation maturity; no institutional compliance determination is inferred.
CONFIDENCE TYPE. Definitional for the architecture, deductive only under stated assumptions, implementation-dependent for runtime and hardware, institutionally imported for authority, and evidence-dependent for deployment claims.

Chapter 75

Independent Assessment, Assurance Cases, Conformity, and Institutional Trust

Claim-argument-evidence structures, defeaters, assessor independence, scope, certification boundaries, reliance, surveillance, and withdrawal.

75.1 Objective

The objective of this chapter is to define independent assessment, assurance cases, conformity, and institutional trust as a bounded component of execution-bound governance. The analysis connects normative source, authorized interpretation, rights and duties, hard enforcement, risk and evidence, supplier and data dependency, and remedy. The decisive test is not whether the system can describe the relevant rule, but whether the rule changes the set of reachable consequence-bearing transitions. The governance claim is a structured translation claim and not an independent legal determination; current primary sources and competent institutional interpretation remain mandatory.

75.2 Structural Model

A compact representation used in this chapter is:

$$Claim \leftarrow Argument \leftarrow Evidence$$

The structural object is represented as a typed relation rather than an informal checklist. Identity, object, context, authority, version, time, fault state, and evidence are separated because a change in any one of them can create a different transition. The architecture therefore rejects implicit substitution: a permission for one object is not a permission for another; a proof for one model is not a proof for a deployment; an approval before mutation is not approval after mutation. Unknown mandatory state is represented explicitly and does not default to ordinary allow.

The chapter’s principal structural commitments are:

Normative Source - retained as an explicit state or dependency.

Authorized Interpretation - retained as an explicit state or dependency.

Rights And Duties - retained as an explicit state or dependency.

Hard Enforcement - retained as an explicit state or dependency.

Risk And Evidence - retained as an explicit state or dependency.

Supplier And Data Dependency - retained as an explicit state or dependency.

Remedy - retained as an explicit state or dependency.

75.3 Execution Reality

Execution reality begins at the last point from which the protected consequence can still be prevented. Upstream analysis, documentation, monitoring, or recommendation may improve decision quality, but it becomes governance only when the consequence path depends on the resulting state. The chapter therefore treats direct capability, alternate interfaces, administrative paths, recovery mechanisms, stale state, and external dependencies as part of the governed object. A system is not credited with a property merely because a dashboard displays it; the relevant gate, controller, institution, or physical interface must enforce or evidence it.

75.4 Failure Point and Counter-Construction

The claim collapses under any in-scope counter-construction such as:

legal text is reduced to a policy label

risk score overrides a prohibition

supplier access becomes shadow authority

a certificate is generalized beyond its scope

These failures are not edge cases external to the theory. They are the primary falsification instruments. Where one of them remains reachable, the correct result is a narrowed claim, an unresolved state, or a failed invariant - not a narrative assertion that the architecture is nevertheless governed.

75.5 Verification and Evidence Obligations

verify the current authoritative source

trace interpretation to executable control

test supplier, data, change, incident, and remedy paths

bound every assurance statement to scope

Evidence must remain bound to the exact artifact, configuration, version, operating envelope, and decision state examined. A passing result establishes only the tested property and coverage. A failed or indeterminate result remains part of the authoritative record. Where independent reproduction has not occurred, the result is not represented as independently established.

75.6 Selected Propositions

P75.1. The validity of independent assessment, assurance cases, conformity, and institutional trust cannot exceed the scope of its weakest mandatory dependency.

P75.2. A descriptive or advisory representation does not become an execution constraint until it controls the relevant consequence path.

P75.3. Component-level validity does not compose automatically across interfaces, timing, authority, failure, or deployment change.

P75.4. Evidence of absence is not inferred from absence of evidence; unresolved mandatory state remains unresolved.

75.7 Governance Status and Confidence Type

STRUCTURE. Specified as part of the integrated federated architecture.
EXECUTION REALITY. Requires a domain-specific implementation and complete path inventory.
FAILURE POINT. Any surviving bypass, stale authority, omitted dependency, unmodeled timing path, or unsupported evidence link narrows or defeats the claim.
GOVERNANCE STATUS. M1 unless a chapter-specific case study states a higher internal implementation maturity; no institutional compliance determination is inferred.
CONFIDENCE TYPE. Definitional for the architecture, deductive only under stated assumptions, implementation-dependent for runtime and hardware, institutionally imported for authority, and evidence-dependent for deployment claims.

Chapter 76

Public Disclosure, Protected Intellectual Property, Research Publication, and Licensing Boundaries

Public-safe theory, protected implementation, reproducibility interfaces, responsible disclosure, embargo, NDA review, and multi-artifact licensing.

76.1 Objective

The objective of this chapter is to define public disclosure, protected intellectual property, research publication, and licensing boundaries as a bounded component of execution-bound governance. The analysis connects normative source, authorized interpretation, rights and duties, hard enforcement, risk and evidence, supplier and data dependency, and remedy. The decisive test is not whether the system can describe the relevant rule, but whether the rule changes the set of reachable consequence-bearing transitions. The governance claim is a structured translation claim and not an independent legal determination; current primary sources and competent institutional interpretation remain mandatory.

76.2 Structural Model

The structural object is represented as a typed relation rather than an informal checklist. Identity, object, context, authority, version, time, fault state, and evidence are separated because a change in any one of them can create a different transition. The architecture therefore rejects implicit substitution: a permission for one object is not a permission for another; a proof for one model is not a proof for a deployment; an approval before mutation is not approval after mutation. Unknown mandatory state is represented explicitly and does not default to ordinary allow.

The chapter’s principal structural commitments are:

Normative Source - retained as an explicit state or dependency.

Authorized Interpretation - retained as an explicit state or dependency.

Rights And Duties - retained as an explicit state or dependency.

Hard Enforcement - retained as an explicit state or dependency.

Risk And Evidence - retained as an explicit state or dependency.

Supplier And Data Dependency - retained as an explicit state or dependency.

Remedy - retained as an explicit state or dependency.

76.3 Execution Reality

Execution reality begins at the last point from which the protected consequence can still be prevented. Upstream analysis, documentation, monitoring, or recommendation may improve decision quality, but it becomes governance only when the consequence path depends on the resulting state. The chapter therefore treats direct capability, alternate interfaces, administrative paths, recovery mechanisms, stale state, and external dependencies as part of the governed object. A system is not credited with a property merely because a dashboard displays it; the relevant gate, controller, institution, or physical interface must enforce or evidence it.

76.4 Failure Point and Counter-Construction

The claim collapses under any in-scope counter-construction such as:

legal text is reduced to a policy label

risk score overrides a prohibition

supplier access becomes shadow authority

a certificate is generalized beyond its scope

These failures are not edge cases external to the theory. They are the primary falsification instruments. Where one of them remains reachable, the correct result is a narrowed claim, an unresolved state, or a failed invariant - not a narrative assertion that the architecture is nevertheless governed.

76.5 Verification and Evidence Obligations

verify the current authoritative source

trace interpretation to executable control

test supplier, data, change, incident, and remedy paths

bound every assurance statement to scope

Evidence must remain bound to the exact artifact, configuration, version, operating envelope, and decision state examined. A passing result establishes only the tested property and coverage. A failed or indeterminate result remains part of the authoritative record. Where independent reproduction has not occurred, the result is not represented as independently established.

76.6 Selected Propositions

P76.1. The validity of public disclosure, protected intellectual property, research publication, and licensing boundaries cannot exceed the scope of its weakest mandatory dependency.

P76.2. A descriptive or advisory representation does not become an execution constraint until it controls the relevant consequence path.

P76.3. Component-level validity does not compose automatically across interfaces, timing, authority, failure, or deployment change.

P76.4. Evidence of absence is not inferred from absence of evidence; unresolved mandatory state remains unresolved.

76.7 Governance Status and Confidence Type

STRUCTURE. Specified as part of the integrated federated architecture.
EXECUTION REALITY. Requires a domain-specific implementation and complete path inventory.
FAILURE POINT. Any surviving bypass, stale authority, omitted dependency, unmodeled timing path, or unsupported evidence link narrows or defeats the claim.
GOVERNANCE STATUS. M1 unless a chapter-specific case study states a higher internal implementation maturity; no institutional compliance determination is inferred.
CONFIDENCE TYPE. Definitional for the architecture, deductive only under stated assumptions, implementation-dependent for runtime and hardware, institutionally imported for authority, and evidence-dependent for deployment claims.

Chapter 77

Governance and Law Volume Synthesis and Closure

The unified normative governance object, master governance predicate, rights, risk, evidence, dependencies, change, incident, assurance, and disclosure.

77.1 Objective

The objective of this chapter is to define governance and law volume synthesis and closure as a bounded component of execution-bound governance. The analysis connects normative source, authorized interpretation, rights and duties, hard enforcement, risk and evidence, supplier and data dependency, and remedy. The decisive test is not whether the system can describe the relevant rule, but whether the rule changes the set of reachable consequence-bearing transitions. The governance claim is a structured translation claim and not an independent legal determination; current primary sources and competent institutional interpretation remain mandatory.

77.2 Structural Model

A compact representation used in this chapter is:

$$GovernanceClaim \Rightarrow Governed \wedge EvidenceClosed$$

The structural object is represented as a typed relation rather than an informal checklist. Identity, object, context, authority, version, time, fault state, and evidence are separated because a change in any one of them can create a different transition. The architecture therefore rejects implicit substitution: a permission for one object is not a permission for another; a proof for one model is not a proof for a deployment; an approval before mutation is not approval after mutation. Unknown mandatory state is represented explicitly and does not default to ordinary allow.

The chapter’s principal structural commitments are:

Normative Source - retained as an explicit state or dependency.

Authorized Interpretation - retained as an explicit state or dependency.

Rights And Duties - retained as an explicit state or dependency.

Hard Enforcement - retained as an explicit state or dependency.

Risk And Evidence - retained as an explicit state or dependency.

Supplier And Data Dependency - retained as an explicit state or dependency.

Remedy - retained as an explicit state or dependency.

77.3 Execution Reality

Execution reality begins at the last point from which the protected consequence can still be prevented. Upstream analysis, documentation, monitoring, or recommendation may improve decision quality, but it becomes governance only when the consequence path depends on the resulting state. The chapter therefore treats direct capability, alternate interfaces, administrative paths, recovery mechanisms, stale state, and external dependencies as part of the governed object. A system is not credited with a property merely because a dashboard displays it; the relevant gate, controller, institution, or physical interface must enforce or evidence it.

77.4 Failure Point and Counter-Construction

The claim collapses under any in-scope counter-construction such as:

legal text is reduced to a policy label

risk score overrides a prohibition

supplier access becomes shadow authority

a certificate is generalized beyond its scope

These failures are not edge cases external to the theory. They are the primary falsification instruments. Where one of them remains reachable, the correct result is a narrowed claim, an unresolved state, or a failed invariant - not a narrative assertion that the architecture is nevertheless governed.

77.5 Verification and Evidence Obligations

verify the current authoritative source

trace interpretation to executable control

test supplier, data, change, incident, and remedy paths

bound every assurance statement to scope

Evidence must remain bound to the exact artifact, configuration, version, operating envelope, and decision state examined. A passing result establishes only the tested property and coverage. A failed or indeterminate result remains part of the authoritative record. Where independent reproduction has not occurred, the result is not represented as independently established.

77.6 Selected Propositions

P77.1. The validity of governance and law volume synthesis and closure cannot exceed the scope of its weakest mandatory dependency.

P77.2. A descriptive or advisory representation does not become an execution constraint until it controls the relevant consequence path.

P77.3. Component-level validity does not compose automatically across interfaces, timing, authority, failure, or deployment change.

P77.4. Evidence of absence is not inferred from absence of evidence; unresolved mandatory state remains unresolved.

77.7 Governance Status and Confidence Type

STRUCTURE. Specified as part of the integrated federated architecture.
EXECUTION REALITY. Requires a domain-specific implementation and complete path inventory.
FAILURE POINT. Any surviving bypass, stale authority, omitted dependency, unmodeled timing path, or unsupported evidence link narrows or defeats the claim.
GOVERNANCE STATUS. M1 unless a chapter-specific case study states a higher internal implementation maturity; no institutional compliance determination is inferred.
CONFIDENCE TYPE. Definitional for the architecture, deductive only under stated assumptions, implementation-dependent for runtime and hardware, institutionally imported for authority, and evidence-dependent for deployment claims.

PART VIII - APPLIED SYSTEMS, ORGANIZATIONAL DEPLOYMENT, AND DOMAIN IMPLEMENTATION

Part VIII instantiates the federated theory in applied domains. The purpose is not to claim universal implementation but to show how the same core can be configured without losing domain-specific authority, safety, evidence, and consequence semantics.

Part-level claim boundary. The material in this part is specified at M1 unless a bounded case study explicitly records internal implementation evidence. No part-level statement independently widens the system-level claim.

Chapter 78

From Federated Theory to Deployable Governed Systems

A domain-instantiation method that preserves boundaries, authority, control, runtime, hardware, evidence, and governance correspondence.

78.1 Objective

The objective of this chapter is to define from federated theory to deployable governed systems as a bounded component of execution-bound governance. The analysis connects domain boundary, use-case decomposition, authority map, execution gate, evidence design, deployment maturity, and case continuity. The decisive test is not whether the system can describe the relevant rule, but whether the rule changes the set of reachable consequence-bearing transitions. The applied-system claim is a case-bounded instantiation and does not generalize from laboratory configuration, demonstrator success, or one organization to an entire sector.

78.2 Structural Model

A compact representation used in this chapter is:

$$DomainInstance = Core \oplus Control \oplus Runtime \oplus Hardware \oplus Authority \oplus Governance \ominus$$

The structural object is represented as a typed relation rather than an informal checklist. Identity, object, context, authority, version, time, fault state, and evidence are separated because a change in any one of them can create a different transition. The architecture therefore rejects implicit substitution: a permission for one object is not a permission for another; a proof for one model is not a proof for a deployment; an approval before mutation is not approval after mutation. Unknown mandatory state is represented explicitly and does not default to ordinary allow.

The chapter’s principal structural commitments are:

Domain Boundary - retained as an explicit state or dependency.

Use-Case Decomposition - retained as an explicit state or dependency.

Authority Map - retained as an explicit state or dependency.

Execution Gate - retained as an explicit state or dependency.

Evidence Design - retained as an explicit state or dependency.

Deployment Maturity - retained as an explicit state or dependency.

Case Continuity - retained as an explicit state or dependency.

78.3 Execution Reality

Execution reality begins at the last point from which the protected consequence can still be prevented. Upstream analysis, documentation, monitoring, or recommendation may improve decision quality, but it becomes governance only when the consequence path depends on the resulting state. The chapter therefore treats direct capability, alternate interfaces, administrative paths, recovery mechanisms, stale state, and external dependencies as part of the governed object. A system is not credited with a property merely because a dashboard displays it; the relevant gate, controller, institution, or physical interface must enforce or evidence it.

78.4 Failure Point and Counter-Construction

The claim collapses under any in-scope counter-construction such as:

domain boundaries are copied from an organization chart

case state is lost at handover

prototype evidence is promoted to deployment evidence

local convenience silently expands authority

These failures are not edge cases external to the theory. They are the primary falsification instruments. Where one of them remains reachable, the correct result is a narrowed claim, an unresolved state, or a failed invariant - not a narrative assertion that the architecture is nevertheless governed.

78.5 Verification and Evidence Obligations

instantiate the full architecture for one real workflow

neutralize lab-specific identifiers before release

preserve public-safe versus protected boundaries

run staged internal and independent validation

Evidence must remain bound to the exact artifact, configuration, version, operating envelope, and decision state examined. A passing result establishes only the tested property and coverage. A failed or indeterminate result remains part of the authoritative record. Where independent reproduction has not occurred, the result is not represented as independently established.

78.6 Selected Propositions

P78.1. The validity of from federated theory to deployable governed systems cannot exceed the scope of its weakest mandatory dependency.

P78.2. A descriptive or advisory representation does not become an execution constraint until it controls the relevant consequence path.

P78.3. Component-level validity does not compose automatically across interfaces, timing, authority, failure, or deployment change.

P78.4. Evidence of absence is not inferred from absence of evidence; unresolved mandatory state remains unresolved.

78.7 Governance Status and Confidence Type

STRUCTURE. Specified as part of the integrated federated architecture.
EXECUTION REALITY. Requires a domain-specific implementation and complete path inventory.
FAILURE POINT. Any surviving bypass, stale authority, omitted dependency, unmodeled timing path, or unsupported evidence link narrows or defeats the claim.
GOVERNANCE STATUS. M1 unless a chapter-specific case study states a higher internal implementation maturity; no institutional compliance determination is inferred.
CONFIDENCE TYPE. Definitional for the architecture, deductive only under stated assumptions, implementation-dependent for runtime and hardware, institutionally imported for authority, and evidence-dependent for deployment claims.

Chapter 79

Boundary Selection, Use-Case Decomposition, and Consequence Mapping

Selecting system, object, authority, data, execution, physical, and institutional boundaries before implementation.

79.1 Objective

The objective of this chapter is to define boundary selection, use-case decomposition, and consequence mapping as a bounded component of execution-bound governance. The analysis connects domain boundary, use-case decomposition, authority map, execution gate, evidence design, deployment maturity, and case continuity. The decisive test is not whether the system can describe the relevant rule, but whether the rule changes the set of reachable consequence-bearing transitions. The applied-system claim is a case-bounded instantiation and does not generalize from laboratory configuration, demonstrator success, or one organization to an entire sector.

79.2 Structural Model

A compact representation used in this chapter is:

$$\Omega_{app} = \Omega_{system} \cap \Omega_{authority} \cap \Omega_{data} \cap \Omega_{execution} \cap \Omega_{fault}$$

The structural object is represented as a typed relation rather than an informal checklist. Identity, object, context, authority, version, time, fault state, and evidence are separated because a change in any one of them can create a different transition. The architecture therefore rejects implicit substitution: a permission for one object is not a permission for another; a proof for one model is not a proof for a deployment; an approval before mutation is not approval after mutation. Unknown mandatory state is represented explicitly and does not default to ordinary allow.

The chapter’s principal structural commitments are:

Domain Boundary - retained as an explicit state or dependency.

Use-Case Decomposition - retained as an explicit state or dependency.

Authority Map - retained as an explicit state or dependency.

Execution Gate - retained as an explicit state or dependency.

Evidence Design - retained as an explicit state or dependency.

Deployment Maturity - retained as an explicit state or dependency.

Case Continuity - retained as an explicit state or dependency.

79.3 Execution Reality

Execution reality begins at the last point from which the protected consequence can still be prevented. Upstream analysis, documentation, monitoring, or recommendation may improve decision quality, but it becomes governance only when the consequence path depends on the resulting state. The chapter therefore treats direct capability, alternate interfaces, administrative paths, recovery mechanisms, stale state, and external dependencies as part of the governed object. A system is not credited with a property merely because a dashboard displays it; the relevant gate, controller, institution, or physical interface must enforce or evidence it.

79.4 Failure Point and Counter-Construction

The claim collapses under any in-scope counter-construction such as:

domain boundaries are copied from an organization chart

case state is lost at handover

prototype evidence is promoted to deployment evidence

local convenience silently expands authority

These failures are not edge cases external to the theory. They are the primary falsification instruments. Where one of them remains reachable, the correct result is a narrowed claim, an unresolved state, or a failed invariant - not a narrative assertion that the architecture is nevertheless governed.

79.5 Verification and Evidence Obligations

instantiate the full architecture for one real workflow

neutralize lab-specific identifiers before release

preserve public-safe versus protected boundaries

run staged internal and independent validation

Evidence must remain bound to the exact artifact, configuration, version, operating envelope, and decision state examined. A passing result establishes only the tested property and coverage. A failed or indeterminate result remains part of the authoritative record. Where independent reproduction has not occurred, the result is not represented as independently established.

79.6 Selected Propositions

P79.1. The validity of boundary selection, use-case decomposition, and consequence mapping cannot exceed the scope of its weakest mandatory dependency.

P79.2. A descriptive or advisory representation does not become an execution constraint until it controls the relevant consequence path.

P79.3. Component-level validity does not compose automatically across interfaces, timing, authority, failure, or deployment change.

P79.4. Evidence of absence is not inferred from absence of evidence; unresolved mandatory state remains unresolved.

79.7 Governance Status and Confidence Type

STRUCTURE. Specified as part of the integrated federated architecture.
EXECUTION REALITY. Requires a domain-specific implementation and complete path inventory.
FAILURE POINT. Any surviving bypass, stale authority, omitted dependency, unmodeled timing path, or unsupported evidence link narrows or defeats the claim.
GOVERNANCE STATUS. M1 unless a chapter-specific case study states a higher internal implementation maturity; no institutional compliance determination is inferred.
CONFIDENCE TYPE. Definitional for the architecture, deductive only under stated assumptions, implementation-dependent for runtime and hardware, institutionally imported for authority, and evidence-dependent for deployment claims.

Chapter 80

KWI LIAN and Kyber as an Institutional Workflow Case Study

A public-safe case architecture for local document workflows, human commit, case state, governance metadata, and LAN-based continuity.

80.1 Objective

The objective of this chapter is to define kwi lian and kyber as an institutional workflow case study as a bounded component of execution-bound governance. The analysis connects domain boundary, use-case decomposition, authority map, execution gate, evidence design, deployment maturity, and case continuity. The decisive test is not whether the system can describe the relevant rule, but whether the rule changes the set of reachable consequence-bearing transitions. The applied-system claim is a case-bounded instantiation and does not generalize from laboratory configuration, demonstrator success, or one organization to an entire sector.

80.2 Structural Model

A compact representation used in this chapter is:

Case Request → LIAN Gate → Human Commit → Kyber Action → Case Evidence

The structural object is represented as a typed relation rather than an informal checklist. Identity, object, context, authority, version, time, fault state, and evidence are separated because a change in any one of them can create a different transition. The architecture therefore rejects implicit substitution: a permission for one object is not a permission for another; a proof for one model is not a proof for a deployment; an approval before mutation is not approval after mutation. Unknown mandatory state is represented explicitly and does not default to ordinary allow.

The chapter’s principal structural commitments are:

Domain Boundary - retained as an explicit state or dependency.

Use-Case Decomposition - retained as an explicit state or dependency.

Authority Map - retained as an explicit state or dependency.

Execution Gate - retained as an explicit state or dependency.

Evidence Design - retained as an explicit state or dependency.

Deployment Maturity - retained as an explicit state or dependency.

Case Continuity - retained as an explicit state or dependency.

80.3 Execution Reality

Execution reality begins at the last point from which the protected consequence can still be prevented. Upstream analysis, documentation, monitoring, or recommendation may improve decision quality, but it becomes governance only when the consequence path depends on the resulting state. The chapter therefore treats direct capability, alternate interfaces, administrative paths, recovery mechanisms, stale state, and external dependencies as part of the governed object. A system is not credited with a property merely because a dashboard displays it; the relevant gate, controller, institution, or physical interface must enforce or evidence it.

80.4 Failure Point and Counter-Construction

The claim collapses under any in-scope counter-construction such as:

domain boundaries are copied from an organization chart

case state is lost at handover

prototype evidence is promoted to deployment evidence

local convenience silently expands authority

These failures are not edge cases external to the theory. They are the primary falsification instruments. Where one of them remains reachable, the correct result is a narrowed claim, an unresolved state, or a failed invariant - not a narrative assertion that the architecture is nevertheless governed.

80.5 Verification and Evidence Obligations

instantiate the full architecture for one real workflow

neutralize lab-specific identifiers before release

preserve public-safe versus protected boundaries

run staged internal and independent validation

Evidence must remain bound to the exact artifact, configuration, version, operating envelope, and decision state examined. A passing result establishes only the tested property and coverage. A failed or indeterminate result remains part of the authoritative record. Where independent reproduction has not occurred, the result is not represented as independently established.

80.6 Selected Propositions

P80.1. The validity of kwi lian and kyber as an institutional workflow case study cannot exceed the scope of its weakest mandatory dependency.

P80.2. A descriptive or advisory representation does not become an execution constraint until it controls the relevant consequence path.

P80.3. Component-level validity does not compose automatically across interfaces, timing, authority, failure, or deployment change.

P80.4. Evidence of absence is not inferred from absence of evidence; unresolved mandatory state remains unresolved.

80.7 Governance Status and Confidence Type

STRUCTURE. Specified as part of the integrated federated architecture.
EXECUTION REALITY. Requires a domain-specific implementation and complete path inventory.
FAILURE POINT. Any surviving bypass, stale authority, omitted dependency, unmodeled timing path, or unsupported evidence link narrows or defeats the claim.
GOVERNANCE STATUS. M1 unless a chapter-specific case study states a higher internal implementation maturity; no institutional compliance determination is inferred.
CONFIDENCE TYPE. Definitional for the architecture, deductive only under stated assumptions, implementation-dependent for runtime and hardware, institutionally imported for authority, and evidence-dependent for deployment claims.

Chapter 81

Public Administration and Case-Based Decision Workflows

Case ownership, statutory or institutional mandate import, document generation, multi-principal review, handover, and contestability.

81.1 Objective

The objective of this chapter is to define public administration and case-based decision workflows as a bounded component of execution-bound governance. The analysis connects domain boundary, use-case decomposition, authority map, execution gate, evidence design, deployment maturity, and case continuity. The decisive test is not whether the system can describe the relevant rule, but whether the rule changes the set of reachable consequence-bearing transitions. The applied-system claim is a case-bounded instantiation and does not generalize from laboratory configuration, demonstrator success, or one organization to an entire sector.

81.2 Structural Model

The structural object is represented as a typed relation rather than an informal checklist. Identity, object, context, authority, version, time, fault state, and evidence are separated because a change in any one of them can create a different transition. The architecture therefore rejects implicit substitution: a permission for one object is not a permission for another; a proof for one model is not a proof for a deployment; an approval before mutation is not approval after mutation. Unknown mandatory state is represented explicitly and does not default to ordinary allow.

The chapter’s principal structural commitments are:

Domain Boundary - retained as an explicit state or dependency.

Use-Case Decomposition - retained as an explicit state or dependency.

Authority Map - retained as an explicit state or dependency.

Execution Gate - retained as an explicit state or dependency.

Evidence Design - retained as an explicit state or dependency.

Deployment Maturity - retained as an explicit state or dependency.

Case Continuity - retained as an explicit state or dependency.

81.3 Execution Reality

Execution reality begins at the last point from which the protected consequence can still be prevented. Upstream analysis, documentation, monitoring, or recommendation may improve decision quality, but it becomes governance only when the consequence path depends on the resulting state. The chapter therefore treats direct capability, alternate interfaces, administrative paths, recovery mechanisms, stale state, and external dependencies as part of the governed object. A system is not credited with a property merely because a dashboard displays it; the relevant gate, controller, institution, or physical interface must enforce or evidence it.

81.4 Failure Point and Counter-Construction

The claim collapses under any in-scope counter-construction such as:

domain boundaries are copied from an organization chart

case state is lost at handover

prototype evidence is promoted to deployment evidence

local convenience silently expands authority

These failures are not edge cases external to the theory. They are the primary falsification instruments. Where one of them remains reachable, the correct result is a narrowed claim, an unresolved state, or a failed invariant - not a narrative assertion that the architecture is nevertheless governed.

81.5 Verification and Evidence Obligations

instantiate the full architecture for one real workflow

neutralize lab-specific identifiers before release

preserve public-safe versus protected boundaries

run staged internal and independent validation

Evidence must remain bound to the exact artifact, configuration, version, operating envelope, and decision state examined. A passing result establishes only the tested property and coverage. A failed or indeterminate result remains part of the authoritative record. Where independent reproduction has not occurred, the result is not represented as independently established.

81.6 Selected Propositions

P81.1. The validity of public administration and case-based decision workflows cannot exceed the scope of its weakest mandatory dependency.

P81.2. A descriptive or advisory representation does not become an execution constraint until it controls the relevant consequence path.

P81.3. Component-level validity does not compose automatically across interfaces, timing, authority, failure, or deployment change.

P81.4. Evidence of absence is not inferred from absence of evidence; unresolved mandatory state remains unresolved.

81.7 Governance Status and Confidence Type

STRUCTURE. Specified as part of the integrated federated architecture.
EXECUTION REALITY. Requires a domain-specific implementation and complete path inventory.
FAILURE POINT. Any surviving bypass, stale authority, omitted dependency, unmodeled timing path, or unsupported evidence link narrows or defeats the claim.
GOVERNANCE STATUS. M1 unless a chapter-specific case study states a higher internal implementation maturity; no institutional compliance determination is inferred.
CONFIDENCE TYPE. Definitional for the architecture, deductive only under stated assumptions, implementation-dependent for runtime and hardware, institutionally imported for authority, and evidence-dependent for deployment claims.

Chapter 82

Clinical and Health-Related Decision Support

Decision support without autonomous medical authority, evidence quality, uncertainty, clinician mandate, consent, escalation, and hard stop conditions.

82.1 Objective

The objective of this chapter is to define clinical and health-related decision support as a bounded component of execution-bound governance. The analysis connects domain boundary, use-case decomposition, authority map, execution gate, evidence design, deployment maturity, and case continuity. The decisive test is not whether the system can describe the relevant rule, but whether the rule changes the set of reachable consequence-bearing transitions. The applied-system claim is a case-bounded instantiation and does not generalize from laboratory configuration, demonstrator success, or one organization to an entire sector.

82.2 Structural Model

The structural object is represented as a typed relation rather than an informal checklist. Identity, object, context, authority, version, time, fault state, and evidence are separated because a change in any one of them can create a different transition. The architecture therefore rejects implicit substitution: a permission for one object is not a permission for another; a proof for one model is not a proof for a deployment; an approval before mutation is not approval after mutation. Unknown mandatory state is represented explicitly and does not default to ordinary allow.

The chapter’s principal structural commitments are:

Domain Boundary - retained as an explicit state or dependency.

Use-Case Decomposition - retained as an explicit state or dependency.

Authority Map - retained as an explicit state or dependency.

Execution Gate - retained as an explicit state or dependency.

Evidence Design - retained as an explicit state or dependency.

Deployment Maturity - retained as an explicit state or dependency.

Case Continuity - retained as an explicit state or dependency.

82.3 Execution Reality

Execution reality begins at the last point from which the protected consequence can still be prevented. Upstream analysis, documentation, monitoring, or recommendation may improve decision quality, but it becomes governance only when the consequence path depends on the resulting state. The chapter therefore treats direct capability, alternate interfaces, administrative paths, recovery mechanisms, stale state, and external dependencies as part of the governed object. A system is not credited with a property merely because a dashboard displays it; the relevant gate, controller, institution, or physical interface must enforce or evidence it.

82.4 Failure Point and Counter-Construction

The claim collapses under any in-scope counter-construction such as:

domain boundaries are copied from an organization chart

case state is lost at handover

prototype evidence is promoted to deployment evidence

local convenience silently expands authority

These failures are not edge cases external to the theory. They are the primary falsification instruments. Where one of them remains reachable, the correct result is a narrowed claim, an unresolved state, or a failed invariant - not a narrative assertion that the architecture is nevertheless governed.

82.5 Verification and Evidence Obligations

instantiate the full architecture for one real workflow

neutralize lab-specific identifiers before release

preserve public-safe versus protected boundaries

run staged internal and independent validation

Evidence must remain bound to the exact artifact, configuration, version, operating envelope, and decision state examined. A passing result establishes only the tested property and coverage. A failed or indeterminate result remains part of the authoritative record. Where independent reproduction has not occurred, the result is not represented as independently established.

82.6 Selected Propositions

P82.1. The validity of clinical and health-related decision support cannot exceed the scope of its weakest mandatory dependency.

P82.2. A descriptive or advisory representation does not become an execution constraint until it controls the relevant consequence path.

P82.3. Component-level validity does not compose automatically across interfaces, timing, authority, failure, or deployment change.

P82.4. Evidence of absence is not inferred from absence of evidence; unresolved mandatory state remains unresolved.

82.7 Governance Status and Confidence Type

STRUCTURE. Specified as part of the integrated federated architecture.
EXECUTION REALITY. Requires a domain-specific implementation and complete path inventory.
FAILURE POINT. Any surviving bypass, stale authority, omitted dependency, unmodeled timing path, or unsupported evidence link narrows or defeats the claim.
GOVERNANCE STATUS. M1 unless a chapter-specific case study states a higher internal implementation maturity; no institutional compliance determination is inferred.
CONFIDENCE TYPE. Definitional for the architecture, deductive only under stated assumptions, implementation-dependent for runtime and hardware, institutionally imported for authority, and evidence-dependent for deployment claims.

Chapter 83

Industrial Edge, Maintenance, and Critical-Infrastructure Workflows

Local operation, offline continuity, maintenance authority, firmware and configuration control, degraded modes, and physical consequence boundaries.

83.1 Objective

The objective of this chapter is to define industrial edge, maintenance, and critical-infrastructure workflows as a bounded component of execution-bound governance. The analysis connects domain boundary, use-case decomposition, authority map, execution gate, evidence design, deployment maturity, and case continuity. The decisive test is not whether the system can describe the relevant rule, but whether the rule changes the set of reachable consequence-bearing transitions. The applied-system claim is a case-bounded instantiation and does not generalize from laboratory configuration, demonstrator success, or one organization to an entire sector.

83.2 Structural Model

The structural object is represented as a typed relation rather than an informal checklist. Identity, object, context, authority, version, time, fault state, and evidence are separated because a change in any one of them can create a different transition. The architecture therefore rejects implicit substitution: a permission for one object is not a permission for another; a proof for one model is not a proof for a deployment; an approval before mutation is not approval after mutation. Unknown mandatory state is represented explicitly and does not default to ordinary allow.

The chapter’s principal structural commitments are:

Domain Boundary - retained as an explicit state or dependency.

Use-Case Decomposition - retained as an explicit state or dependency.

Authority Map - retained as an explicit state or dependency.

Execution Gate - retained as an explicit state or dependency.

Evidence Design - retained as an explicit state or dependency.

Deployment Maturity - retained as an explicit state or dependency.

Case Continuity - retained as an explicit state or dependency.

83.3 Execution Reality

Execution reality begins at the last point from which the protected consequence can still be prevented. Upstream analysis, documentation, monitoring, or recommendation may improve decision quality, but it becomes governance only when the consequence path depends on the resulting state. The chapter therefore treats direct capability, alternate interfaces, administrative paths, recovery mechanisms, stale state, and external dependencies as part of the governed object. A system is not credited with a property merely because a dashboard displays it; the relevant gate, controller, institution, or physical interface must enforce or evidence it.

83.4 Failure Point and Counter-Construction

The claim collapses under any in-scope counter-construction such as:

domain boundaries are copied from an organization chart

case state is lost at handover

prototype evidence is promoted to deployment evidence

local convenience silently expands authority

These failures are not edge cases external to the theory. They are the primary falsification instruments. Where one of them remains reachable, the correct result is a narrowed claim, an unresolved state, or a failed invariant - not a narrative assertion that the architecture is nevertheless governed.

83.5 Verification and Evidence Obligations

instantiate the full architecture for one real workflow

neutralize lab-specific identifiers before release

preserve public-safe versus protected boundaries

run staged internal and independent validation

Evidence must remain bound to the exact artifact, configuration, version, operating envelope, and decision state examined. A passing result establishes only the tested property and coverage. A failed or indeterminate result remains part of the authoritative record. Where independent reproduction has not occurred, the result is not represented as independently established.

83.6 Selected Propositions

P83.1. The validity of industrial edge, maintenance, and critical-infrastructure workflows cannot exceed the scope of its weakest mandatory dependency.

P83.2. A descriptive or advisory representation does not become an execution constraint until it controls the relevant consequence path.

P83.3. Component-level validity does not compose automatically across interfaces, timing, authority, failure, or deployment change.

P83.4. Evidence of absence is not inferred from absence of evidence; unresolved mandatory state remains unresolved.

83.7 Governance Status and Confidence Type

STRUCTURE. Specified as part of the integrated federated architecture.
EXECUTION REALITY. Requires a domain-specific implementation and complete path inventory.
FAILURE POINT. Any surviving bypass, stale authority, omitted dependency, unmodeled timing path, or unsupported evidence link narrows or defeats the claim.
GOVERNANCE STATUS. M1 unless a chapter-specific case study states a higher internal implementation maturity; no institutional compliance determination is inferred.
CONFIDENCE TYPE. Definitional for the architecture, deductive only under stated assumptions, implementation-dependent for runtime and hardware, institutionally imported for authority, and evidence-dependent for deployment claims.

Chapter 84

Automotive Advisory Intelligence and Execution-Bound Integration

Advisory optimization above existing controllers, gateway mediation, Co-Chip enforcement, driver authority, and vehicle validation envelopes.

84.1 Objective

The objective of this chapter is to define automotive advisory intelligence and execution-bound integration as a bounded component of execution-bound governance. The analysis connects domain boundary, use-case decomposition, authority map, execution gate, evidence design, deployment maturity, and case continuity. The decisive test is not whether the system can describe the relevant rule, but whether the rule changes the set of reachable consequence-bearing transitions. The applied-system claim is a case-bounded instantiation and does not generalize from laboratory configuration, demonstrator success, or one organization to an entire sector.

84.2 Structural Model

The structural object is represented as a typed relation rather than an informal checklist. Identity, object, context, authority, version, time, fault state, and evidence are separated because a change in any one of them can create a different transition. The architecture therefore rejects implicit substitution: a permission for one object is not a permission for another; a proof for one model is not a proof for a deployment; an approval before mutation is not approval after mutation. Unknown mandatory state is represented explicitly and does not default to ordinary allow.

The chapter’s principal structural commitments are:

Domain Boundary - retained as an explicit state or dependency.

Use-Case Decomposition - retained as an explicit state or dependency.

Authority Map - retained as an explicit state or dependency.

Execution Gate - retained as an explicit state or dependency.

Evidence Design - retained as an explicit state or dependency.

Deployment Maturity - retained as an explicit state or dependency.

Case Continuity - retained as an explicit state or dependency.

84.3 Execution Reality

Execution reality begins at the last point from which the protected consequence can still be prevented. Upstream analysis, documentation, monitoring, or recommendation may improve decision quality, but it becomes governance only when the consequence path depends on the resulting state. The chapter therefore treats direct capability, alternate interfaces, administrative paths, recovery mechanisms, stale state, and external dependencies as part of the governed object. A system is not credited with a property merely because a dashboard displays it; the relevant gate, controller, institution, or physical interface must enforce or evidence it.

84.4 Failure Point and Counter-Construction

The claim collapses under any in-scope counter-construction such as:

domain boundaries are copied from an organization chart

case state is lost at handover

prototype evidence is promoted to deployment evidence

local convenience silently expands authority

These failures are not edge cases external to the theory. They are the primary falsification instruments. Where one of them remains reachable, the correct result is a narrowed claim, an unresolved state, or a failed invariant - not a narrative assertion that the architecture is nevertheless governed.

84.5 Verification and Evidence Obligations

instantiate the full architecture for one real workflow

neutralize lab-specific identifiers before release

preserve public-safe versus protected boundaries

run staged internal and independent validation

Evidence must remain bound to the exact artifact, configuration, version, operating envelope, and decision state examined. A passing result establishes only the tested property and coverage. A failed or indeterminate result remains part of the authoritative record. Where independent reproduction has not occurred, the result is not represented as independently established.

84.6 Selected Propositions

P84.1. The validity of automotive advisory intelligence and execution-bound integration cannot exceed the scope of its weakest mandatory dependency.

P84.2. A descriptive or advisory representation does not become an execution constraint until it controls the relevant consequence path.

P84.3. Component-level validity does not compose automatically across interfaces, timing, authority, failure, or deployment change.

P84.4. Evidence of absence is not inferred from absence of evidence; unresolved mandatory state remains unresolved.

84.7 Governance Status and Confidence Type

STRUCTURE. Specified as part of the integrated federated architecture.
EXECUTION REALITY. Requires a domain-specific implementation and complete path inventory.
FAILURE POINT. Any surviving bypass, stale authority, omitted dependency, unmodeled timing path, or unsupported evidence link narrows or defeats the claim.
GOVERNANCE STATUS. M1 unless a chapter-specific case study states a higher internal implementation maturity; no institutional compliance determination is inferred.
CONFIDENCE TYPE. Definitional for the architecture, deductive only under stated assumptions, implementation-dependent for runtime and hardware, institutionally imported for authority, and evidence-dependent for deployment claims.

Chapter 85

Document-Centric Self-Proving Workflows

Document identity, content fingerprinting, human commit, embedded governance metadata, audit portability, and limits of self-contained evidence.

85.1 Objective

The objective of this chapter is to define document-centric self-proving workflows as a bounded component of execution-bound governance. The analysis connects domain boundary, use-case decomposition, authority map, execution gate, evidence design, deployment maturity, and case continuity. The decisive test is not whether the system can describe the relevant rule, but whether the rule changes the set of reachable consequence-bearing transitions. The applied-system claim is a case-bounded instantiation and does not generalize from laboratory configuration, demonstrator success, or one organization to an entire sector.

85.2 Structural Model

A compact representation used in this chapter is:

$$h_D = H\left(Canon\left(Document, Transaction, Decision, Version\right)\right)$$

The structural object is represented as a typed relation rather than an informal checklist. Identity, object, context, authority, version, time, fault state, and evidence are separated because a change in any one of them can create a different transition. The architecture therefore rejects implicit substitution: a permission for one object is not a permission for another; a proof for one model is not a proof for a deployment; an approval before mutation is not approval after mutation. Unknown mandatory state is represented explicitly and does not default to ordinary allow.

The chapter’s principal structural commitments are:

Domain Boundary - retained as an explicit state or dependency.

Use-Case Decomposition - retained as an explicit state or dependency.

Authority Map - retained as an explicit state or dependency.

Execution Gate - retained as an explicit state or dependency.

Evidence Design - retained as an explicit state or dependency.

Deployment Maturity - retained as an explicit state or dependency.

Case Continuity - retained as an explicit state or dependency.

85.3 Execution Reality

Execution reality begins at the last point from which the protected consequence can still be prevented. Upstream analysis, documentation, monitoring, or recommendation may improve decision quality, but it becomes governance only when the consequence path depends on the resulting state. The chapter therefore treats direct capability, alternate interfaces, administrative paths, recovery mechanisms, stale state, and external dependencies as part of the governed object. A system is not credited with a property merely because a dashboard displays it; the relevant gate, controller, institution, or physical interface must enforce or evidence it.

85.4 Failure Point and Counter-Construction

The claim collapses under any in-scope counter-construction such as:

domain boundaries are copied from an organization chart

case state is lost at handover

prototype evidence is promoted to deployment evidence

local convenience silently expands authority

These failures are not edge cases external to the theory. They are the primary falsification instruments. Where one of them remains reachable, the correct result is a narrowed claim, an unresolved state, or a failed invariant - not a narrative assertion that the architecture is nevertheless governed.

85.5 Verification and Evidence Obligations

instantiate the full architecture for one real workflow

neutralize lab-specific identifiers before release

preserve public-safe versus protected boundaries

run staged internal and independent validation

Evidence must remain bound to the exact artifact, configuration, version, operating envelope, and decision state examined. A passing result establishes only the tested property and coverage. A failed or indeterminate result remains part of the authoritative record. Where independent reproduction has not occurred, the result is not represented as independently established.

85.6 Selected Propositions

P85.1. The validity of document-centric self-proving workflows cannot exceed the scope of its weakest mandatory dependency.

P85.2. A descriptive or advisory representation does not become an execution constraint until it controls the relevant consequence path.

P85.3. Component-level validity does not compose automatically across interfaces, timing, authority, failure, or deployment change.

P85.4. Evidence of absence is not inferred from absence of evidence; unresolved mandatory state remains unresolved.

85.7 Governance Status and Confidence Type

STRUCTURE. Specified as part of the integrated federated architecture.
EXECUTION REALITY. Requires a domain-specific implementation and complete path inventory.
FAILURE POINT. Any surviving bypass, stale authority, omitted dependency, unmodeled timing path, or unsupported evidence link narrows or defeats the claim.
GOVERNANCE STATUS. M1 unless a chapter-specific case study states a higher internal implementation maturity; no institutional compliance determination is inferred.
CONFIDENCE TYPE. Definitional for the architecture, deductive only under stated assumptions, implementation-dependent for runtime and hardware, institutionally imported for authority, and evidence-dependent for deployment claims.

Chapter 86

Multi-Agent, Long-Horizon, and Cross-System Governance

Delegation, shared context, constraint inheritance, distributed admissibility, authority gaps, and common-parent correlation.

86.1 Objective

The objective of this chapter is to define multi-agent, long-horizon, and cross-system governance as a bounded component of execution-bound governance. The analysis connects domain boundary, use-case decomposition, authority map, execution gate, evidence design, deployment maturity, and case continuity. The decisive test is not whether the system can describe the relevant rule, but whether the rule changes the set of reachable consequence-bearing transitions. The applied-system claim is a case-bounded instantiation and does not generalize from laboratory configuration, demonstrator success, or one organization to an entire sector.

86.2 Structural Model

A compact representation used in this chapter is:

$$A_{child} \preceq A_{parent}, A_{global} = \bigcap_i A_i \cap A_{interfaces}$$

The structural object is represented as a typed relation rather than an informal checklist. Identity, object, context, authority, version, time, fault state, and evidence are separated because a change in any one of them can create a different transition. The architecture therefore rejects implicit substitution: a permission for one object is not a permission for another; a proof for one model is not a proof for a deployment; an approval before mutation is not approval after mutation. Unknown mandatory state is represented explicitly and does not default to ordinary allow.

The chapter’s principal structural commitments are:

Domain Boundary - retained as an explicit state or dependency.

Use-Case Decomposition - retained as an explicit state or dependency.

Authority Map - retained as an explicit state or dependency.

Execution Gate - retained as an explicit state or dependency.

Evidence Design - retained as an explicit state or dependency.

Deployment Maturity - retained as an explicit state or dependency.

Case Continuity - retained as an explicit state or dependency.

86.3 Execution Reality

Execution reality begins at the last point from which the protected consequence can still be prevented. Upstream analysis, documentation, monitoring, or recommendation may improve decision quality, but it becomes governance only when the consequence path depends on the resulting state. The chapter therefore treats direct capability, alternate interfaces, administrative paths, recovery mechanisms, stale state, and external dependencies as part of the governed object. A system is not credited with a property merely because a dashboard displays it; the relevant gate, controller, institution, or physical interface must enforce or evidence it.

86.4 Failure Point and Counter-Construction

The claim collapses under any in-scope counter-construction such as:

domain boundaries are copied from an organization chart

case state is lost at handover

prototype evidence is promoted to deployment evidence

local convenience silently expands authority

These failures are not edge cases external to the theory. They are the primary falsification instruments. Where one of them remains reachable, the correct result is a narrowed claim, an unresolved state, or a failed invariant - not a narrative assertion that the architecture is nevertheless governed.

86.5 Verification and Evidence Obligations

instantiate the full architecture for one real workflow

neutralize lab-specific identifiers before release

preserve public-safe versus protected boundaries

run staged internal and independent validation

Evidence must remain bound to the exact artifact, configuration, version, operating envelope, and decision state examined. A passing result establishes only the tested property and coverage. A failed or indeterminate result remains part of the authoritative record. Where independent reproduction has not occurred, the result is not represented as independently established.

86.6 Selected Propositions

P86.1. The validity of multi-agent, long-horizon, and cross-system governance cannot exceed the scope of its weakest mandatory dependency.

P86.2. A descriptive or advisory representation does not become an execution constraint until it controls the relevant consequence path.

P86.3. Component-level validity does not compose automatically across interfaces, timing, authority, failure, or deployment change.

P86.4. Evidence of absence is not inferred from absence of evidence; unresolved mandatory state remains unresolved.

86.7 Governance Status and Confidence Type

STRUCTURE. Specified as part of the integrated federated architecture.
EXECUTION REALITY. Requires a domain-specific implementation and complete path inventory.
FAILURE POINT. Any surviving bypass, stale authority, omitted dependency, unmodeled timing path, or unsupported evidence link narrows or defeats the claim.
GOVERNANCE STATUS. M1 unless a chapter-specific case study states a higher internal implementation maturity; no institutional compliance determination is inferred.
CONFIDENCE TYPE. Definitional for the architecture, deductive only under stated assumptions, implementation-dependent for runtime and hardware, institutionally imported for authority, and evidence-dependent for deployment claims.

Chapter 87

Applied-Systems Synthesis and Deployment Pattern Library

Reusable domain patterns, anti-patterns, evidence templates, maturity gates, and the boundary between demonstrator and operational deployment.

87.1 Objective

The objective of this chapter is to define applied-systems synthesis and deployment pattern library as a bounded component of execution-bound governance. The analysis connects domain boundary, use-case decomposition, authority map, execution gate, evidence design, deployment maturity, and case continuity. The decisive test is not whether the system can describe the relevant rule, but whether the rule changes the set of reachable consequence-bearing transitions. The applied-system claim is a case-bounded instantiation and does not generalize from laboratory configuration, demonstrator success, or one organization to an entire sector.

87.2 Structural Model

The structural object is represented as a typed relation rather than an informal checklist. Identity, object, context, authority, version, time, fault state, and evidence are separated because a change in any one of them can create a different transition. The architecture therefore rejects implicit substitution: a permission for one object is not a permission for another; a proof for one model is not a proof for a deployment; an approval before mutation is not approval after mutation. Unknown mandatory state is represented explicitly and does not default to ordinary allow.

The chapter’s principal structural commitments are:

Domain Boundary - retained as an explicit state or dependency.

Use-Case Decomposition - retained as an explicit state or dependency.

Authority Map - retained as an explicit state or dependency.

Execution Gate - retained as an explicit state or dependency.

Evidence Design - retained as an explicit state or dependency.

Deployment Maturity - retained as an explicit state or dependency.

Case Continuity - retained as an explicit state or dependency.

87.3 Execution Reality

Execution reality begins at the last point from which the protected consequence can still be prevented. Upstream analysis, documentation, monitoring, or recommendation may improve decision quality, but it becomes governance only when the consequence path depends on the resulting state. The chapter therefore treats direct capability, alternate interfaces, administrative paths, recovery mechanisms, stale state, and external dependencies as part of the governed object. A system is not credited with a property merely because a dashboard displays it; the relevant gate, controller, institution, or physical interface must enforce or evidence it.

87.4 Failure Point and Counter-Construction

The claim collapses under any in-scope counter-construction such as:

domain boundaries are copied from an organization chart

case state is lost at handover

prototype evidence is promoted to deployment evidence

local convenience silently expands authority

These failures are not edge cases external to the theory. They are the primary falsification instruments. Where one of them remains reachable, the correct result is a narrowed claim, an unresolved state, or a failed invariant - not a narrative assertion that the architecture is nevertheless governed.

87.5 Verification and Evidence Obligations

instantiate the full architecture for one real workflow

neutralize lab-specific identifiers before release

preserve public-safe versus protected boundaries

run staged internal and independent validation

Evidence must remain bound to the exact artifact, configuration, version, operating envelope, and decision state examined. A passing result establishes only the tested property and coverage. A failed or indeterminate result remains part of the authoritative record. Where independent reproduction has not occurred, the result is not represented as independently established.

87.6 Selected Propositions

P87.1. The validity of applied-systems synthesis and deployment pattern library cannot exceed the scope of its weakest mandatory dependency.

P87.2. A descriptive or advisory representation does not become an execution constraint until it controls the relevant consequence path.

P87.3. Component-level validity does not compose automatically across interfaces, timing, authority, failure, or deployment change.

P87.4. Evidence of absence is not inferred from absence of evidence; unresolved mandatory state remains unresolved.

87.7 Governance Status and Confidence Type

STRUCTURE. Specified as part of the integrated federated architecture.
EXECUTION REALITY. Requires a domain-specific implementation and complete path inventory.
FAILURE POINT. Any surviving bypass, stale authority, omitted dependency, unmodeled timing path, or unsupported evidence link narrows or defeats the claim.
GOVERNANCE STATUS. M1 unless a chapter-specific case study states a higher internal implementation maturity; no institutional compliance determination is inferred.
CONFIDENCE TYPE. Definitional for the architecture, deductive only under stated assumptions, implementation-dependent for runtime and hardware, institutionally imported for authority, and evidence-dependent for deployment claims.

PART IX - VALIDATION, FALSIFICATION, AND INDEPENDENT REPRODUCTION

Part IX defines how the theory can fail. It treats negative results, counterexamples, coverage, independent reproduction, and unresolved states as first-class research outputs rather than administrative inconveniences.

Part-level claim boundary. The material in this part is specified at M1 unless a bounded case study explicitly records internal implementation evidence. No part-level statement independently widens the system-level claim.

Chapter 88

Validation Architecture and Evidence Maturity

A unified evidence program from M0 conceptuality to M6 institutional determination, with claim-specific coverage and ceilings.

88.1 Objective

The objective of this chapter is to define validation architecture and evidence maturity as a bounded component of execution-bound governance. The analysis connects claim-specific test object, coverage, counterexample, independent reproduction, evidence package, maturity ceiling, and unresolved state. The decisive test is not whether the system can describe the relevant rule, but whether the rule changes the set of reachable consequence-bearing transitions. The validation claim distinguishes proof, bounded safety, statistical support, internal test, independent reproduction, institutional assessment, unknown, and out-of-scope results.

88.2 Structural Model

A compact representation used in this chapter is:

$$M\left(C_{system}\right) \leq \min_{i} M\left(E_{i}\right)$$

The structural object is represented as a typed relation rather than an informal checklist. Identity, object, context, authority, version, time, fault state, and evidence are separated because a change in any one of them can create a different transition. The architecture therefore rejects implicit substitution: a permission for one object is not a permission for another; a proof for one model is not a proof for a deployment; an approval before mutation is not approval after mutation. Unknown mandatory state is represented explicitly and does not default to ordinary allow.

The chapter’s principal structural commitments are:

Claim-Specific Test Object - retained as an explicit state or dependency.

Coverage - retained as an explicit state or dependency.

Counterexample - retained as an explicit state or dependency.

Independent Reproduction - retained as an explicit state or dependency.

Evidence Package - retained as an explicit state or dependency.

Maturity Ceiling - retained as an explicit state or dependency.

Unresolved State - retained as an explicit state or dependency.

88.3 Execution Reality

Execution reality begins at the last point from which the protected consequence can still be prevented. Upstream analysis, documentation, monitoring, or recommendation may improve decision quality, but it becomes governance only when the consequence path depends on the resulting state. The chapter therefore treats direct capability, alternate interfaces, administrative paths, recovery mechanisms, stale state, and external dependencies as part of the governed object. A system is not credited with a property merely because a dashboard displays it; the relevant gate, controller, institution, or physical interface must enforce or evidence it.

88.4 Failure Point and Counter-Construction

The claim collapses under any in-scope counter-construction such as:

tests exercise only favorable paths

failed traces are omitted

the tested artifact differs from deployment

absence of failure is reported as proof

These failures are not edge cases external to the theory. They are the primary falsification instruments. Where one of them remains reachable, the correct result is a narrowed claim, an unresolved state, or a failed invariant - not a narrative assertion that the architecture is nevertheless governed.

88.5 Verification and Evidence Obligations

pre-register properties and failure criteria

capture full evidence including negative results

repeat on independent artifacts and environments

report proved, refuted, bounded, unknown, and out-of-scope distinctly

Evidence must remain bound to the exact artifact, configuration, version, operating envelope, and decision state examined. A passing result establishes only the tested property and coverage. A failed or indeterminate result remains part of the authoritative record. Where independent reproduction has not occurred, the result is not represented as independently established.

88.6 Selected Propositions

P88.1. The validity of validation architecture and evidence maturity cannot exceed the scope of its weakest mandatory dependency.

P88.2. A descriptive or advisory representation does not become an execution constraint until it controls the relevant consequence path.

P88.3. Component-level validity does not compose automatically across interfaces, timing, authority, failure, or deployment change.

P88.4. Evidence of absence is not inferred from absence of evidence; unresolved mandatory state remains unresolved.

88.7 Governance Status and Confidence Type

STRUCTURE. Specified as part of the integrated federated architecture.
EXECUTION REALITY. Requires a domain-specific implementation and complete path inventory.
FAILURE POINT. Any surviving bypass, stale authority, omitted dependency, unmodeled timing path, or unsupported evidence link narrows or defeats the claim.
GOVERNANCE STATUS. M1 unless a chapter-specific case study states a higher internal implementation maturity; no institutional compliance determination is inferred.
CONFIDENCE TYPE. Definitional for the architecture, deductive only under stated assumptions, implementation-dependent for runtime and hardware, institutionally imported for authority, and evidence-dependent for deployment claims.

Chapter 89

Formal Verification and Mechanized Proof Plan

Model scope, proof obligations, theorem provers, model checkers, certificates, trusted bases, and negative-result preservation.

89.1 Objective

The objective of this chapter is to define formal verification and mechanized proof plan as a bounded component of execution-bound governance. The analysis connects claim-specific test object, coverage, counterexample, independent reproduction, evidence package, maturity ceiling, and unresolved state. The decisive test is not whether the system can describe the relevant rule, but whether the rule changes the set of reachable consequence-bearing transitions. The validation claim distinguishes proof, bounded safety, statistical support, internal test, independent reproduction, institutional assessment, unknown, and out-of-scope results.

89.2 Structural Model

A compact representation used in this chapter is:

$$PO=(ID,Assumptions,Model,Property,Method,Certificate,Status)$$

The structural object is represented as a typed relation rather than an informal checklist. Identity, object, context, authority, version, time, fault state, and evidence are separated because a change in any one of them can create a different transition. The architecture therefore rejects implicit substitution: a permission for one object is not a permission for another; a proof for one model is not a proof for a deployment; an approval before mutation is not approval after mutation. Unknown mandatory state is represented explicitly and does not default to ordinary allow.

The chapter’s principal structural commitments are:

Claim-Specific Test Object - retained as an explicit state or dependency.

Coverage - retained as an explicit state or dependency.

Counterexample - retained as an explicit state or dependency.

Independent Reproduction - retained as an explicit state or dependency.

Evidence Package - retained as an explicit state or dependency.

Maturity Ceiling - retained as an explicit state or dependency.

Unresolved State - retained as an explicit state or dependency.

89.3 Execution Reality

Execution reality begins at the last point from which the protected consequence can still be prevented. Upstream analysis, documentation, monitoring, or recommendation may improve decision quality, but it becomes governance only when the consequence path depends on the resulting state. The chapter therefore treats direct capability, alternate interfaces, administrative paths, recovery mechanisms, stale state, and external dependencies as part of the governed object. A system is not credited with a property merely because a dashboard displays it; the relevant gate, controller, institution, or physical interface must enforce or evidence it.

89.4 Failure Point and Counter-Construction

The claim collapses under any in-scope counter-construction such as:

tests exercise only favorable paths

failed traces are omitted

the tested artifact differs from deployment

absence of failure is reported as proof

These failures are not edge cases external to the theory. They are the primary falsification instruments. Where one of them remains reachable, the correct result is a narrowed claim, an unresolved state, or a failed invariant - not a narrative assertion that the architecture is nevertheless governed.

89.5 Verification and Evidence Obligations

pre-register properties and failure criteria

capture full evidence including negative results

repeat on independent artifacts and environments

report proved, refuted, bounded, unknown, and out-of-scope distinctly

Evidence must remain bound to the exact artifact, configuration, version, operating envelope, and decision state examined. A passing result establishes only the tested property and coverage. A failed or indeterminate result remains part of the authoritative record. Where independent reproduction has not occurred, the result is not represented as independently established.

89.6 Selected Propositions

P89.1. The validity of formal verification and mechanized proof plan cannot exceed the scope of its weakest mandatory dependency.

P89.2. A descriptive or advisory representation does not become an execution constraint until it controls the relevant consequence path.

P89.3. Component-level validity does not compose automatically across interfaces, timing, authority, failure, or deployment change.

P89.4. Evidence of absence is not inferred from absence of evidence; unresolved mandatory state remains unresolved.

89.7 Governance Status and Confidence Type

STRUCTURE. Specified as part of the integrated federated architecture.
EXECUTION REALITY. Requires a domain-specific implementation and complete path inventory.
FAILURE POINT. Any surviving bypass, stale authority, omitted dependency, unmodeled timing path, or unsupported evidence link narrows or defeats the claim.
GOVERNANCE STATUS. M1 unless a chapter-specific case study states a higher internal implementation maturity; no institutional compliance determination is inferred.
CONFIDENCE TYPE. Definitional for the architecture, deductive only under stated assumptions, implementation-dependent for runtime and hardware, institutionally imported for authority, and evidence-dependent for deployment claims.

Chapter 90

Software, Runtime, and Distributed-System Test Program

State-machine tests, concurrency, crash, replay, capability bypass, message duplication, configuration drift, and recovery.

90.1 Objective

The objective of this chapter is to define software, runtime, and distributed-system test program as a bounded component of execution-bound governance. The analysis connects claim-specific test object, coverage, counterexample, independent reproduction, evidence package, maturity ceiling, and unresolved state. The decisive test is not whether the system can describe the relevant rule, but whether the rule changes the set of reachable consequence-bearing transitions. The validation claim distinguishes proof, bounded safety, statistical support, internal test, independent reproduction, institutional assessment, unknown, and out-of-scope results.

90.2 Structural Model

A compact representation used in this chapter is:

$$TestSpace = States \times Events \times Faults \times Versions \times Timings$$

The structural object is represented as a typed relation rather than an informal checklist. Identity, object, context, authority, version, time, fault state, and evidence are separated because a change in any one of them can create a different transition. The architecture therefore rejects implicit substitution: a permission for one object is not a permission for another; a proof for one model is not a proof for a deployment; an approval before mutation is not approval after mutation. Unknown mandatory state is represented explicitly and does not default to ordinary allow.

The chapter’s principal structural commitments are:

Claim-Specific Test Object - retained as an explicit state or dependency.

Coverage - retained as an explicit state or dependency.

Counterexample - retained as an explicit state or dependency.

Independent Reproduction - retained as an explicit state or dependency.

Evidence Package - retained as an explicit state or dependency.

Maturity Ceiling - retained as an explicit state or dependency.

Unresolved State - retained as an explicit state or dependency.

90.3 Execution Reality

Execution reality begins at the last point from which the protected consequence can still be prevented. Upstream analysis, documentation, monitoring, or recommendation may improve decision quality, but it becomes governance only when the consequence path depends on the resulting state. The chapter therefore treats direct capability, alternate interfaces, administrative paths, recovery mechanisms, stale state, and external dependencies as part of the governed object. A system is not credited with a property merely because a dashboard displays it; the relevant gate, controller, institution, or physical interface must enforce or evidence it.

90.4 Failure Point and Counter-Construction

The claim collapses under any in-scope counter-construction such as:

tests exercise only favorable paths

failed traces are omitted

the tested artifact differs from deployment

absence of failure is reported as proof

These failures are not edge cases external to the theory. They are the primary falsification instruments. Where one of them remains reachable, the correct result is a narrowed claim, an unresolved state, or a failed invariant - not a narrative assertion that the architecture is nevertheless governed.

90.5 Verification and Evidence Obligations

pre-register properties and failure criteria

capture full evidence including negative results

repeat on independent artifacts and environments

report proved, refuted, bounded, unknown, and out-of-scope distinctly

Evidence must remain bound to the exact artifact, configuration, version, operating envelope, and decision state examined. A passing result establishes only the tested property and coverage. A failed or indeterminate result remains part of the authoritative record. Where independent reproduction has not occurred, the result is not represented as independently established.

90.6 Selected Propositions

P90.1. The validity of software, runtime, and distributed-system test program cannot exceed the scope of its weakest mandatory dependency.

P90.2. A descriptive or advisory representation does not become an execution constraint until it controls the relevant consequence path.

P90.3. Component-level validity does not compose automatically across interfaces, timing, authority, failure, or deployment change.

P90.4. Evidence of absence is not inferred from absence of evidence; unresolved mandatory state remains unresolved.

90.7 Governance Status and Confidence Type

STRUCTURE. Specified as part of the integrated federated architecture.
EXECUTION REALITY. Requires a domain-specific implementation and complete path inventory.
FAILURE POINT. Any surviving bypass, stale authority, omitted dependency, unmodeled timing path, or unsupported evidence link narrows or defeats the claim.
GOVERNANCE STATUS. M1 unless a chapter-specific case study states a higher internal implementation maturity; no institutional compliance determination is inferred.
CONFIDENCE TYPE. Definitional for the architecture, deductive only under stated assumptions, implementation-dependent for runtime and hardware, institutionally imported for authority, and evidence-dependent for deployment claims.

Chapter 91

Hardware, HIL, and Physical Fault-Injection Program

Clock, reset, power, CDC, line faults, bypasses, actuator feedback, environmental corners, and production variation.

91.1 Objective

The objective of this chapter is to define hardware, hil, and physical fault-injection program as a bounded component of execution-bound governance. The analysis connects claim-specific test object, coverage, counterexample, independent reproduction, evidence package, maturity ceiling, and unresolved state. The decisive test is not whether the system can describe the relevant rule, but whether the rule changes the set of reachable consequence-bearing transitions. The validation claim distinguishes proof, bounded safety, statistical support, internal test, independent reproduction, institutional assessment, unknown, and out-of-scope results.

91.2 Structural Model

A compact representation used in this chapter is:

$$\Omega_{HIL} = PlantModel \times Hardware \times Timing \times Faults \times Environment$$

The structural object is represented as a typed relation rather than an informal checklist. Identity, object, context, authority, version, time, fault state, and evidence are separated because a change in any one of them can create a different transition. The architecture therefore rejects implicit substitution: a permission for one object is not a permission for another; a proof for one model is not a proof for a deployment; an approval before mutation is not approval after mutation. Unknown mandatory state is represented explicitly and does not default to ordinary allow.

The chapter’s principal structural commitments are:

Claim-Specific Test Object - retained as an explicit state or dependency.

Coverage - retained as an explicit state or dependency.

Counterexample - retained as an explicit state or dependency.

Independent Reproduction - retained as an explicit state or dependency.

Evidence Package - retained as an explicit state or dependency.

Maturity Ceiling - retained as an explicit state or dependency.

Unresolved State - retained as an explicit state or dependency.

91.3 Execution Reality

Execution reality begins at the last point from which the protected consequence can still be prevented. Upstream analysis, documentation, monitoring, or recommendation may improve decision quality, but it becomes governance only when the consequence path depends on the resulting state. The chapter therefore treats direct capability, alternate interfaces, administrative paths, recovery mechanisms, stale state, and external dependencies as part of the governed object. A system is not credited with a property merely because a dashboard displays it; the relevant gate, controller, institution, or physical interface must enforce or evidence it.

91.4 Failure Point and Counter-Construction

The claim collapses under any in-scope counter-construction such as:

tests exercise only favorable paths

failed traces are omitted

the tested artifact differs from deployment

absence of failure is reported as proof

These failures are not edge cases external to the theory. They are the primary falsification instruments. Where one of them remains reachable, the correct result is a narrowed claim, an unresolved state, or a failed invariant - not a narrative assertion that the architecture is nevertheless governed.

91.5 Verification and Evidence Obligations

pre-register properties and failure criteria

capture full evidence including negative results

repeat on independent artifacts and environments

report proved, refuted, bounded, unknown, and out-of-scope distinctly

Evidence must remain bound to the exact artifact, configuration, version, operating envelope, and decision state examined. A passing result establishes only the tested property and coverage. A failed or indeterminate result remains part of the authoritative record. Where independent reproduction has not occurred, the result is not represented as independently established.

91.6 Selected Propositions

P91.1. The validity of hardware, hil, and physical fault-injection program cannot exceed the scope of its weakest mandatory dependency.

P91.2. A descriptive or advisory representation does not become an execution constraint until it controls the relevant consequence path.

P91.3. Component-level validity does not compose automatically across interfaces, timing, authority, failure, or deployment change.

P91.4. Evidence of absence is not inferred from absence of evidence; unresolved mandatory state remains unresolved.

91.7 Governance Status and Confidence Type

STRUCTURE. Specified as part of the integrated federated architecture.
EXECUTION REALITY. Requires a domain-specific implementation and complete path inventory.
FAILURE POINT. Any surviving bypass, stale authority, omitted dependency, unmodeled timing path, or unsupported evidence link narrows or defeats the claim.
GOVERNANCE STATUS. M1 unless a chapter-specific case study states a higher internal implementation maturity; no institutional compliance determination is inferred.
CONFIDENCE TYPE. Definitional for the architecture, deductive only under stated assumptions, implementation-dependent for runtime and hardware, institutionally imported for authority, and evidence-dependent for deployment claims.

Chapter 92

Human-Factors, Authority, and Interface Validation

Mandate, competence, display-commit equivalence, intervention margin, workload, accessibility, consent, and handoff.

92.1 Objective

The objective of this chapter is to define human-factors, authority, and interface validation as a bounded component of execution-bound governance. The analysis connects claim-specific test object, coverage, counterexample, independent reproduction, evidence package, maturity ceiling, and unresolved state. The decisive test is not whether the system can describe the relevant rule, but whether the rule changes the set of reachable consequence-bearing transitions. The validation claim distinguishes proof, bounded safety, statistical support, internal test, independent reproduction, institutional assessment, unknown, and out-of-scope results.

92.2 Structural Model

A compact representation used in this chapter is:

$$HumanValidity = Mandate \wedge Competence \wedge Information \wedge Time \wedge Control$$

The structural object is represented as a typed relation rather than an informal checklist. Identity, object, context, authority, version, time, fault state, and evidence are separated because a change in any one of them can create a different transition. The architecture therefore rejects implicit substitution: a permission for one object is not a permission for another; a proof for one model is not a proof for a deployment; an approval before mutation is not approval after mutation. Unknown mandatory state is represented explicitly and does not default to ordinary allow.

The chapter’s principal structural commitments are:

Claim-Specific Test Object - retained as an explicit state or dependency.

Coverage - retained as an explicit state or dependency.

Counterexample - retained as an explicit state or dependency.

Independent Reproduction - retained as an explicit state or dependency.

Evidence Package - retained as an explicit state or dependency.

Maturity Ceiling - retained as an explicit state or dependency.

Unresolved State - retained as an explicit state or dependency.

92.3 Execution Reality

Execution reality begins at the last point from which the protected consequence can still be prevented. Upstream analysis, documentation, monitoring, or recommendation may improve decision quality, but it becomes governance only when the consequence path depends on the resulting state. The chapter therefore treats direct capability, alternate interfaces, administrative paths, recovery mechanisms, stale state, and external dependencies as part of the governed object. A system is not credited with a property merely because a dashboard displays it; the relevant gate, controller, institution, or physical interface must enforce or evidence it.

92.4 Failure Point and Counter-Construction

The claim collapses under any in-scope counter-construction such as:

tests exercise only favorable paths

failed traces are omitted

the tested artifact differs from deployment

absence of failure is reported as proof

These failures are not edge cases external to the theory. They are the primary falsification instruments. Where one of them remains reachable, the correct result is a narrowed claim, an unresolved state, or a failed invariant - not a narrative assertion that the architecture is nevertheless governed.

92.5 Verification and Evidence Obligations

pre-register properties and failure criteria

capture full evidence including negative results

repeat on independent artifacts and environments

report proved, refuted, bounded, unknown, and out-of-scope distinctly

Evidence must remain bound to the exact artifact, configuration, version, operating envelope, and decision state examined. A passing result establishes only the tested property and coverage. A failed or indeterminate result remains part of the authoritative record. Where independent reproduction has not occurred, the result is not represented as independently established.

92.6 Selected Propositions

P92.1. The validity of human-factors, authority, and interface validation cannot exceed the scope of its weakest mandatory dependency.

P92.2. A descriptive or advisory representation does not become an execution constraint until it controls the relevant consequence path.

P92.3. Component-level validity does not compose automatically across interfaces, timing, authority, failure, or deployment change.

P92.4. Evidence of absence is not inferred from absence of evidence; unresolved mandatory state remains unresolved.

92.7 Governance Status and Confidence Type

STRUCTURE. Specified as part of the integrated federated architecture.
EXECUTION REALITY. Requires a domain-specific implementation and complete path inventory.
FAILURE POINT. Any surviving bypass, stale authority, omitted dependency, unmodeled timing path, or unsupported evidence link narrows or defeats the claim.
GOVERNANCE STATUS. M1 unless a chapter-specific case study states a higher internal implementation maturity; no institutional compliance determination is inferred.
CONFIDENCE TYPE. Definitional for the architecture, deductive only under stated assumptions, implementation-dependent for runtime and hardware, institutionally imported for authority, and evidence-dependent for deployment claims.

Chapter 93

Governance, Legal, Data, and Supplier Validation

Source-to-control traceability, applicability, rights, audit, procurement, purpose limitation, change, incident, and remedy.

93.1 Objective

The objective of this chapter is to define governance, legal, data, and supplier validation as a bounded component of execution-bound governance. The analysis connects claim-specific test object, coverage, counterexample, independent reproduction, evidence package, maturity ceiling, and unresolved state. The decisive test is not whether the system can describe the relevant rule, but whether the rule changes the set of reachable consequence-bearing transitions. The validation claim distinguishes proof, bounded safety, statistical support, internal test, independent reproduction, institutional assessment, unknown, and out-of-scope results.

93.2 Structural Model

A compact representation used in this chapter is:

$$Source \leftrightarrow Requirement \leftrightarrow Control \leftrightarrow Evidence \leftrightarrow Remedy$$

The structural object is represented as a typed relation rather than an informal checklist. Identity, object, context, authority, version, time, fault state, and evidence are separated because a change in any one of them can create a different transition. The architecture therefore rejects implicit substitution: a permission for one object is not a permission for another; a proof for one model is not a proof for a deployment; an approval before mutation is not approval after mutation. Unknown mandatory state is represented explicitly and does not default to ordinary allow.

The chapter’s principal structural commitments are:

Claim-Specific Test Object - retained as an explicit state or dependency.

Coverage - retained as an explicit state or dependency.

Counterexample - retained as an explicit state or dependency.

Independent Reproduction - retained as an explicit state or dependency.

Evidence Package - retained as an explicit state or dependency.

Maturity Ceiling - retained as an explicit state or dependency.

Unresolved State - retained as an explicit state or dependency.

93.3 Execution Reality

Execution reality begins at the last point from which the protected consequence can still be prevented. Upstream analysis, documentation, monitoring, or recommendation may improve decision quality, but it becomes governance only when the consequence path depends on the resulting state. The chapter therefore treats direct capability, alternate interfaces, administrative paths, recovery mechanisms, stale state, and external dependencies as part of the governed object. A system is not credited with a property merely because a dashboard displays it; the relevant gate, controller, institution, or physical interface must enforce or evidence it.

93.4 Failure Point and Counter-Construction

The claim collapses under any in-scope counter-construction such as:

tests exercise only favorable paths

failed traces are omitted

the tested artifact differs from deployment

absence of failure is reported as proof

These failures are not edge cases external to the theory. They are the primary falsification instruments. Where one of them remains reachable, the correct result is a narrowed claim, an unresolved state, or a failed invariant - not a narrative assertion that the architecture is nevertheless governed.

93.5 Verification and Evidence Obligations

pre-register properties and failure criteria

capture full evidence including negative results

repeat on independent artifacts and environments

report proved, refuted, bounded, unknown, and out-of-scope distinctly

Evidence must remain bound to the exact artifact, configuration, version, operating envelope, and decision state examined. A passing result establishes only the tested property and coverage. A failed or indeterminate result remains part of the authoritative record. Where independent reproduction has not occurred, the result is not represented as independently established.

93.6 Selected Propositions

P93.1. The validity of governance, legal, data, and supplier validation cannot exceed the scope of its weakest mandatory dependency.

P93.2. A descriptive or advisory representation does not become an execution constraint until it controls the relevant consequence path.

P93.3. Component-level validity does not compose automatically across interfaces, timing, authority, failure, or deployment change.

P93.4. Evidence of absence is not inferred from absence of evidence; unresolved mandatory state remains unresolved.

93.7 Governance Status and Confidence Type

STRUCTURE. Specified as part of the integrated federated architecture.
EXECUTION REALITY. Requires a domain-specific implementation and complete path inventory.
FAILURE POINT. Any surviving bypass, stale authority, omitted dependency, unmodeled timing path, or unsupported evidence link narrows or defeats the claim.
GOVERNANCE STATUS. M1 unless a chapter-specific case study states a higher internal implementation maturity; no institutional compliance determination is inferred.
CONFIDENCE TYPE. Definitional for the architecture, deductive only under stated assumptions, implementation-dependent for runtime and hardware, institutionally imported for authority, and evidence-dependent for deployment claims.

Chapter 94

Adversarial Falsification and Counter-Construction

Assumption extraction, bypass discovery, common-mode failure, stale authority, late veto, evidence gaps, and collapse tests.

94.1 Objective

The objective of this chapter is to define adversarial falsification and counter-construction as a bounded component of execution-bound governance. The analysis connects claim-specific test object, coverage, counterexample, independent reproduction, evidence package, maturity ceiling, and unresolved state. The decisive test is not whether the system can describe the relevant rule, but whether the rule changes the set of reachable consequence-bearing transitions. The validation claim distinguishes proof, bounded safety, statistical support, internal test, independent reproduction, institutional assessment, unknown, and out-of-scope results.

94.2 Structural Model

A compact representation used in this chapter is:

$$\exists \tau \in \Omega : \neg P(\tau) \Rightarrow \neg \forall \tau \in \Omega P(\tau)$$

The structural object is represented as a typed relation rather than an informal checklist. Identity, object, context, authority, version, time, fault state, and evidence are separated because a change in any one of them can create a different transition. The architecture therefore rejects implicit substitution: a permission for one object is not a permission for another; a proof for one model is not a proof for a deployment; an approval before mutation is not approval after mutation. Unknown mandatory state is represented explicitly and does not default to ordinary allow.

The chapter’s principal structural commitments are:

Claim-Specific Test Object - retained as an explicit state or dependency.

Coverage - retained as an explicit state or dependency.

Counterexample - retained as an explicit state or dependency.

Independent Reproduction - retained as an explicit state or dependency.

Evidence Package - retained as an explicit state or dependency.

Maturity Ceiling - retained as an explicit state or dependency.

Unresolved State - retained as an explicit state or dependency.

94.3 Execution Reality

Execution reality begins at the last point from which the protected consequence can still be prevented. Upstream analysis, documentation, monitoring, or recommendation may improve decision quality, but it becomes governance only when the consequence path depends on the resulting state. The chapter therefore treats direct capability, alternate interfaces, administrative paths, recovery mechanisms, stale state, and external dependencies as part of the governed object. A system is not credited with a property merely because a dashboard displays it; the relevant gate, controller, institution, or physical interface must enforce or evidence it.

94.4 Failure Point and Counter-Construction

The claim collapses under any in-scope counter-construction such as:

tests exercise only favorable paths

failed traces are omitted

the tested artifact differs from deployment

absence of failure is reported as proof

These failures are not edge cases external to the theory. They are the primary falsification instruments. Where one of them remains reachable, the correct result is a narrowed claim, an unresolved state, or a failed invariant - not a narrative assertion that the architecture is nevertheless governed.

94.5 Verification and Evidence Obligations

pre-register properties and failure criteria

capture full evidence including negative results

repeat on independent artifacts and environments

report proved, refuted, bounded, unknown, and out-of-scope distinctly

Evidence must remain bound to the exact artifact, configuration, version, operating envelope, and decision state examined. A passing result establishes only the tested property and coverage. A failed or indeterminate result remains part of the authoritative record. Where independent reproduction has not occurred, the result is not represented as independently established.

94.6 Selected Propositions

P94.1. The validity of adversarial falsification and counter-construction cannot exceed the scope of its weakest mandatory dependency.

P94.2. A descriptive or advisory representation does not become an execution constraint until it controls the relevant consequence path.

P94.3. Component-level validity does not compose automatically across interfaces, timing, authority, failure, or deployment change.

P94.4. Evidence of absence is not inferred from absence of evidence; unresolved mandatory state remains unresolved.

94.7 Governance Status and Confidence Type

STRUCTURE. Specified as part of the integrated federated architecture.
EXECUTION REALITY. Requires a domain-specific implementation and complete path inventory.
FAILURE POINT. Any surviving bypass, stale authority, omitted dependency, unmodeled timing path, or unsupported evidence link narrows or defeats the claim.
GOVERNANCE STATUS. M1 unless a chapter-specific case study states a higher internal implementation maturity; no institutional compliance determination is inferred.
CONFIDENCE TYPE. Definitional for the architecture, deductive only under stated assumptions, implementation-dependent for runtime and hardware, institutionally imported for authority, and evidence-dependent for deployment claims.

Chapter 95

Independent Reproduction, Assurance Cases, and Claim Adjudication

Reproduction levels, assessor independence, public versus protected evidence, defeaters, surveillance, and claim status.

95.1 Objective

The objective of this chapter is to define independent reproduction, assurance cases, and claim adjudication as a bounded component of execution-bound governance. The analysis connects claim-specific test object, coverage, counterexample, independent reproduction, evidence package, maturity ceiling, and unresolved state. The decisive test is not whether the system can describe the relevant rule, but whether the rule changes the set of reachable consequence-bearing transitions. The validation claim distinguishes proof, bounded safety, statistical support, internal test, independent reproduction, institutional assessment, unknown, and out-of-scope results.

95.2 Structural Model

A compact representation used in this chapter is:

$$Reproduction \in \{FR0, FR1, FR2, FR3, FR4, FR5, FR6\}$$

The structural object is represented as a typed relation rather than an informal checklist. Identity, object, context, authority, version, time, fault state, and evidence are separated because a change in any one of them can create a different transition. The architecture therefore rejects implicit substitution: a permission for one object is not a permission for another; a proof for one model is not a proof for a deployment; an approval before mutation is not approval after mutation. Unknown mandatory state is represented explicitly and does not default to ordinary allow.

The chapter's principal structural commitments are:

Claim-Specific Test Object - retained as an explicit state or dependency.

Coverage - retained as an explicit state or dependency.

Counterexample - retained as an explicit state or dependency.

Independent Reproduction - retained as an explicit state or dependency.

Evidence Package - retained as an explicit state or dependency.

Maturity Ceiling - retained as an explicit state or dependency.

Unresolved State - retained as an explicit state or dependency.

95.3 Execution Reality

Execution reality begins at the last point from which the protected consequence can still be prevented. Upstream analysis, documentation, monitoring, or recommendation may improve decision quality, but it becomes governance only when the consequence path depends on the resulting state. The chapter therefore treats direct capability, alternate interfaces, administrative paths, recovery mechanisms, stale state, and external dependencies as part of the governed object. A system is not credited with a property merely because a dashboard displays it; the relevant gate, controller, institution, or physical interface must enforce or evidence it.

95.4 Failure Point and Counter-Construction

The claim collapses under any in-scope counter-construction such as:

tests exercise only favorable paths

failed traces are omitted

the tested artifact differs from deployment

absence of failure is reported as proof

These failures are not edge cases external to the theory. They are the primary falsification instruments. Where one of them remains reachable, the correct result is a narrowed claim, an unresolved state, or a failed invariant - not a narrative assertion that the architecture is nevertheless governed.

95.5 Verification and Evidence Obligations

pre-register properties and failure criteria

capture full evidence including negative results

repeat on independent artifacts and environments

report proved, refuted, bounded, unknown, and out-of-scope distinctly

Evidence must remain bound to the exact artifact, configuration, version, operating envelope, and decision state examined. A passing result establishes only the tested property and coverage. A failed or indeterminate result remains part of the authoritative record. Where independent reproduction has not occurred, the result is not represented as independently established.

95.6 Selected Propositions

P95.1. The validity of independent reproduction, assurance cases, and claim adjudication cannot exceed the scope of its weakest mandatory dependency.

P95.2. A descriptive or advisory representation does not become an execution constraint until it controls the relevant consequence path.

P95.3. Component-level validity does not compose automatically across interfaces, timing, authority, failure, or deployment change.

P95.4. Evidence of absence is not inferred from absence of evidence; unresolved mandatory state remains unresolved.

95.7 Governance Status and Confidence Type

STRUCTURE. Specified as part of the integrated federated architecture.
EXECUTION REALITY. Requires a domain-specific implementation and complete path inventory.
FAILURE POINT. Any surviving bypass, stale authority, omitted dependency, unmodeled timing path, or unsupported evidence link narrows or defeats the claim.
GOVERNANCE STATUS. M1 unless a chapter-specific case study states a higher internal implementation maturity; no institutional compliance determination is inferred.
CONFIDENCE TYPE. Definitional for the architecture, deductive only under stated assumptions, implementation-dependent for runtime and hardware, institutionally imported for authority, and evidence-dependent for deployment claims.

Chapter 96

Integrated Demonstrator and Benchmark Program

A staged public-safe demonstrator spanning document workflow, runtime gate, evidence ledger, hardware signal, and controlled fault campaigns.

96.1 Objective

The objective of this chapter is to define integrated demonstrator and benchmark program as a bounded component of execution-bound governance. The analysis connects claim-specific test object, coverage, counterexample, independent reproduction, evidence package, maturity ceiling, and unresolved state. The decisive test is not whether the system can describe the relevant rule, but whether the rule changes the set of reachable consequence-bearing transitions. The validation claim distinguishes proof, bounded safety, statistical support, internal test, independent reproduction, institutional assessment, unknown, and out-of-scope results.

96.2 Structural Model

The structural object is represented as a typed relation rather than an informal checklist. Identity, object, context, authority, version, time, fault state, and evidence are separated because a change in any one of them can create a different transition. The architecture therefore rejects implicit substitution: a permission for one object is not a permission for another; a proof for one model is not a proof for a deployment; an approval before mutation is not approval after mutation. Unknown mandatory state is represented explicitly and does not default to ordinary allow.

The chapter’s principal structural commitments are:

Claim-Specific Test Object - retained as an explicit state or dependency.

Coverage - retained as an explicit state or dependency.

Counterexample - retained as an explicit state or dependency.

Independent Reproduction - retained as an explicit state or dependency.

Evidence Package - retained as an explicit state or dependency.

Maturity Ceiling - retained as an explicit state or dependency.

Unresolved State - retained as an explicit state or dependency.

96.3 Execution Reality

Execution reality begins at the last point from which the protected consequence can still be prevented. Upstream analysis, documentation, monitoring, or recommendation may improve decision quality, but it becomes governance only when the consequence path depends on the resulting state. The chapter therefore treats direct capability, alternate interfaces, administrative paths, recovery mechanisms, stale state, and external dependencies as part of the governed object. A system is not credited with a property merely because a dashboard displays it; the relevant gate, controller, institution, or physical interface must enforce or evidence it.

96.4 Failure Point and Counter-Construction

The claim collapses under any in-scope counter-construction such as:

tests exercise only favorable paths

failed traces are omitted

the tested artifact differs from deployment

absence of failure is reported as proof

These failures are not edge cases external to the theory. They are the primary falsification instruments. Where one of them remains reachable, the correct result is a narrowed claim, an unresolved state, or a failed invariant - not a narrative assertion that the architecture is nevertheless governed.

96.5 Verification and Evidence Obligations

pre-register properties and failure criteria

capture full evidence including negative results

repeat on independent artifacts and environments

report proved, refuted, bounded, unknown, and out-of-scope distinctly

Evidence must remain bound to the exact artifact, configuration, version, operating envelope, and decision state examined. A passing result establishes only the tested property and coverage. A failed or indeterminate result remains part of the authoritative record. Where independent reproduction has not occurred, the result is not represented as independently established.

96.6 Selected Propositions

P96.1. The validity of integrated demonstrator and benchmark program cannot exceed the scope of its weakest mandatory dependency.

P96.2. A descriptive or advisory representation does not become an execution constraint until it controls the relevant consequence path.

P96.3. Component-level validity does not compose automatically across interfaces, timing, authority, failure, or deployment change.

P96.4. Evidence of absence is not inferred from absence of evidence; unresolved mandatory state remains unresolved.

96.7 Governance Status and Confidence Type

STRUCTURE. Specified as part of the integrated federated architecture.
EXECUTION REALITY. Requires a domain-specific implementation and complete path inventory.
FAILURE POINT. Any surviving bypass, stale authority, omitted dependency, unmodeled timing path, or unsupported evidence link narrows or defeats the claim.
GOVERNANCE STATUS. M1 unless a chapter-specific case study states a higher internal implementation maturity; no institutional compliance determination is inferred.
CONFIDENCE TYPE. Definitional for the architecture, deductive only under stated assumptions, implementation-dependent for runtime and hardware, institutionally imported for authority, and evidence-dependent for deployment claims.

Chapter 97

Validation-Volume Synthesis and Closure

The master evidence predicate, weakest-link rule, unresolved-state discipline, and the transition from research architecture to assessable system.

97.1 Objective

The objective of this chapter is to define validation-volume synthesis and closure as a bounded component of execution-bound governance. The analysis connects claim-specific test object, coverage, counterexample, independent reproduction, evidence package, maturity ceiling, and unresolved state. The decisive test is not whether the system can describe the relevant rule, but whether the rule changes the set of reachable consequence-bearing transitions. The validation claim distinguishes proof, bounded safety, statistical support, internal test, independent reproduction, institutional assessment, unknown, and out-of-scope results.

97.2 Structural Model

A compact representation used in this chapter is:

$$ValidatedClaim = Scoped \wedge Tested \wedge Reproduced \wedge DeploymentBound \wedge \neg OpenMand$$

The structural object is represented as a typed relation rather than an informal checklist. Identity, object, context, authority, version, time, fault state, and evidence are separated because a change in any one of them can create a different transition. The architecture therefore rejects implicit substitution: a permission for one object is not a permission for another; a proof for one model is not a proof for a deployment; an approval before mutation is not approval after mutation. Unknown mandatory state is represented explicitly and does not default to ordinary allow.

The chapter’s principal structural commitments are:

Claim-Specific Test Object - retained as an explicit state or dependency.

Coverage - retained as an explicit state or dependency.

Counterexample - retained as an explicit state or dependency.

Independent Reproduction - retained as an explicit state or dependency.

Evidence Package - retained as an explicit state or dependency.

Maturity Ceiling - retained as an explicit state or dependency.

Unresolved State - retained as an explicit state or dependency.

97.3 Execution Reality

Execution reality begins at the last point from which the protected consequence can still be prevented. Upstream analysis, documentation, monitoring, or recommendation may improve decision quality, but it becomes governance only when the consequence path depends on the resulting state. The chapter therefore treats direct capability, alternate interfaces, administrative paths, recovery mechanisms, stale state, and external dependencies as part of the governed object. A system is not credited with a property merely because a dashboard displays it; the relevant gate, controller, institution, or physical interface must enforce or evidence it.

97.4 Failure Point and Counter-Construction

The claim collapses under any in-scope counter-construction such as:

tests exercise only favorable paths

failed traces are omitted

the tested artifact differs from deployment

absence of failure is reported as proof

These failures are not edge cases external to the theory. They are the primary falsification instruments. Where one of them remains reachable, the correct result is a narrowed claim, an unresolved state, or a failed invariant - not a narrative assertion that the architecture is nevertheless governed.

97.5 Verification and Evidence Obligations

pre-register properties and failure criteria

capture full evidence including negative results

repeat on independent artifacts and environments

report proved, refuted, bounded, unknown, and out-of-scope distinctly

Evidence must remain bound to the exact artifact, configuration, version, operating envelope, and decision state examined. A passing result establishes only the tested property and coverage. A failed or indeterminate result remains part of the authoritative record. Where independent reproduction has not occurred, the result is not represented as independently established.

97.6 Selected Propositions

P97.1. The validity of validation-volume synthesis and closure cannot exceed the scope of its weakest mandatory dependency.

P97.2. A descriptive or advisory representation does not become an execution constraint until it controls the relevant consequence path.

P97.3. Component-level validity does not compose automatically across interfaces, timing, authority, failure, or deployment change.

P97.4. Evidence of absence is not inferred from absence of evidence; unresolved mandatory state remains unresolved.

97.7 Governance Status and Confidence Type

STRUCTURE. Specified as part of the integrated federated architecture.
EXECUTION REALITY. Requires a domain-specific implementation and complete path inventory.
FAILURE POINT. Any surviving bypass, stale authority, omitted dependency, unmodeled timing path, or unsupported evidence link narrows or defeats the claim.
GOVERNANCE STATUS. M1 unless a chapter-specific case study states a higher internal implementation maturity; no institutional compliance determination is inferred.
CONFIDENCE TYPE. Definitional for the architecture, deductive only under stated assumptions, implementation-dependent for runtime and hardware, institutionally imported for authority, and evidence-dependent for deployment claims.

PART X - ORIGINAL CONTRIBUTIONS, LIMITS, RESEARCH AGENDA, AND CONCLUSION

Part X states the contribution, its limits, and the program required to move from an integrated M1 architecture to independently assessed systems. It closes the manuscript without converting open obligations into solved claims.

Part-level claim boundary. The material in this part is specified at M1 unless a bounded case study explicitly records internal implementation evidence. No part-level statement independently widens the system-level claim.

Chapter 98

Original Contributions, Novelty, and Academic Positioning

The contribution set, relation to systems and control theory, architectural novelty, and limits of priority and originality claims.

98.1 Objective

The objective of this chapter is to define original contributions, novelty, and academic positioning as a bounded component of execution-bound governance. The analysis connects original contribution, scope conservation, non-claim discipline, research program, independent scrutiny, and execution-bound governability. The decisive test is not whether the system can describe the relevant rule, but whether the rule changes the set of reachable consequence-bearing transitions. The synthesis preserves the distinction between an original architectural contribution and a completed deployment-level proof.

98.2 Structural Model

A compact representation used in this chapter is:

$$Contribution=(NovelObject,Formalization,Architecture,Enforcement,Validation,$$

The structural object is represented as a typed relation rather than an informal checklist. Identity, object, context, authority, version, time, fault state, and evidence are separated because a change in any one of them can create a different transition. The architecture therefore rejects implicit substitution: a permission for one object is not a permission for another; a proof for one model is not a proof for a deployment; an approval before mutation is not approval after mutation. Unknown mandatory state is represented explicitly and does not default to ordinary allow.

The chapter's principal structural commitments are:

Original Contribution - retained as an explicit state or dependency.

Scope Conservation - retained as an explicit state or dependency.

Non-Claim Discipline - retained as an explicit state or dependency.

Research Program - retained as an explicit state or dependency.

Independent Scrutiny - retained as an explicit state or dependency.

Execution-Bound Governability - retained as an explicit state or dependency.

98.3 Execution Reality

Execution reality begins at the last point from which the protected consequence can still be prevented. Upstream analysis, documentation, monitoring, or recommendation may improve decision quality, but it becomes governance only when the consequence path depends on the resulting state. The chapter therefore treats direct capability, alternate interfaces, administrative paths, recovery mechanisms, stale state, and external dependencies as part of the governed object. A system is not credited with a property merely because a dashboard displays it; the relevant gate, controller, institution, or physical interface must enforce or evidence it.

98.4 Failure Point and Counter-Construction

The claim collapses under any in-scope counter-construction such as:

novelty is claimed without comparison

limitations are hidden

open problems are presented as solved

protected detail is used to shield unsupported claims

These failures are not edge cases external to the theory. They are the primary falsification instruments. Where one of them remains reachable, the correct result is a narrowed claim, an unresolved state, or a failed invariant - not a narrative assertion that the architecture is nevertheless governed.

98.5 Verification and Evidence Obligations

compare contributions against prior fields

state unresolved dependencies

define a falsifiable program of work

maintain claim and disclosure integrity

Evidence must remain bound to the exact artifact, configuration, version, operating envelope, and decision state examined. A passing result establishes only the tested property and coverage. A failed or indeterminate result remains part of the authoritative record. Where independent reproduction has not occurred, the result is not represented as independently established.

98.6 Selected Propositions

P98.1. The validity of original contributions, novelty, and academic positioning cannot exceed the scope of its weakest mandatory dependency.

P98.2. A descriptive or advisory representation does not become an execution constraint until it controls the relevant consequence path.

P98.3. Component-level validity does not compose automatically across interfaces, timing, authority, failure, or deployment change.

P98.4. Evidence of absence is not inferred from absence of evidence; unresolved mandatory state remains unresolved.

98.7 Governance Status and Confidence Type

STRUCTURE. Specified as part of the integrated federated architecture.
EXECUTION REALITY. Requires a domain-specific implementation and complete path inventory.
FAILURE POINT. Any surviving bypass, stale authority, omitted dependency, unmodeled timing path, or unsupported evidence link narrows or defeats the claim.
GOVERNANCE STATUS. M1 unless a chapter-specific case study states a higher internal implementation maturity; no institutional compliance determination is inferred.
CONFIDENCE TYPE. Definitional for the architecture, deductive only under stated assumptions, implementation-dependent for runtime and hardware, institutionally imported for authority, and evidence-dependent for deployment claims.

Chapter 99

Limitations, Non-Claims, and Epistemic Boundaries

Scope restrictions, unmodeled pathways, institutional import, legal uncertainty, empirical gaps, implementation gaps, and protected detail.

99.1 Objective

The objective of this chapter is to define limitations, non-claims, and epistemic boundaries as a bounded component of execution-bound governance. The analysis connects original contribution, scope conservation, non-claim discipline, research program, independent scrutiny, and execution-bound governability. The decisive test is not whether the system can describe the relevant rule, but whether the rule changes the set of reachable consequence-bearing transitions. The synthesis preserves the distinction between an original architectural contribution and a completed deployment-level proof.

99.2 Structural Model

A compact representation used in this chapter is:

$$ClaimScope \subseteq Model \cap FaultEnvelope \cap EnforcementBoundary \cap EvidenceCoverage$$

The structural object is represented as a typed relation rather than an informal checklist. Identity, object, context, authority, version, time, fault state, and evidence are separated because a change in any one of them can create a different transition. The architecture therefore rejects implicit substitution: a permission for one object is not a permission for another; a proof for one model is not a proof for a deployment; an approval before mutation is not approval after mutation. Unknown mandatory state is represented explicitly and does not default to ordinary allow.

The chapter’s principal structural commitments are:

Original Contribution - retained as an explicit state or dependency.

Scope Conservation - retained as an explicit state or dependency.

Non-Claim Discipline - retained as an explicit state or dependency.

Research Program - retained as an explicit state or dependency.

Independent Scrutiny - retained as an explicit state or dependency.

Execution-Bound Governability - retained as an explicit state or dependency.

99.3 Execution Reality

Execution reality begins at the last point from which the protected consequence can still be prevented. Upstream analysis, documentation, monitoring, or recommendation may improve decision quality, but it becomes governance only when the consequence path depends on the resulting state. The chapter therefore treats direct capability, alternate interfaces, administrative paths, recovery mechanisms, stale state, and external dependencies as part of the governed object. A system is not credited with a property merely because a dashboard displays it; the relevant gate, controller, institution, or physical interface must enforce or evidence it.

99.4 Failure Point and Counter-Construction

The claim collapses under any in-scope counter-construction such as:

novelty is claimed without comparison

limitations are hidden

open problems are presented as solved

protected detail is used to shield unsupported claims

These failures are not edge cases external to the theory. They are the primary falsification instruments. Where one of them remains reachable, the correct result is a narrowed claim, an unresolved state, or a failed invariant - not a narrative assertion that the architecture is nevertheless governed.

99.5 Verification and Evidence Obligations

compare contributions against prior fields

state unresolved dependencies

define a falsifiable program of work

maintain claim and disclosure integrity

Evidence must remain bound to the exact artifact, configuration, version, operating envelope, and decision state examined. A passing result establishes only the tested property and coverage. A failed or indeterminate result remains part of the authoritative record. Where independent reproduction has not occurred, the result is not represented as independently established.

99.6 Selected Propositions

P99.1. The validity of limitations, non-claims, and epistemic boundaries cannot exceed the scope of its weakest mandatory dependency.

P99.2. A descriptive or advisory representation does not become an execution constraint until it controls the relevant consequence path.

P99.3. Component-level validity does not compose automatically across interfaces, timing, authority, failure, or deployment change.

P99.4. Evidence of absence is not inferred from absence of evidence; unresolved mandatory state remains unresolved.

99.7 Governance Status and Confidence Type

STRUCTURE. Specified as part of the integrated federated architecture.
EXECUTION REALITY. Requires a domain-specific implementation and complete path inventory.
FAILURE POINT. Any surviving bypass, stale authority, omitted dependency, unmodeled timing path, or unsupported evidence link narrows or defeats the claim.
GOVERNANCE STATUS. M1 unless a chapter-specific case study states a higher internal implementation maturity; no institutional compliance determination is inferred.
CONFIDENCE TYPE. Definitional for the architecture, deductive only under stated assumptions, implementation-dependent for runtime and hardware, institutionally imported for authority, and evidence-dependent for deployment claims.

Chapter 100

Research Agenda and Program of Work

Formalization, control, runtime, hardware, human factors, governance, demonstrators, independent replication, and publication sequence.

100.1 Objective

The objective of this chapter is to define research agenda and program of work as a bounded component of execution-bound governance. The analysis connects original contribution, scope conservation, non-claim discipline, research program, independent scrutiny, and execution-bound governability. The decisive test is not whether the system can describe the relevant rule, but whether the rule changes the set of reachable consequence-bearing transitions. The synthesis preserves the distinction between an original architectural contribution and a completed deployment-level proof.

100.2 Structural Model

A compact representation used in this chapter is:

$$ResearchProgram = Formal \rightarrow Prototype \rightarrow FaultCampaign \rightarrow IndependentReproduct$$

The structural object is represented as a typed relation rather than an informal checklist. Identity, object, context, authority, version, time, fault state, and evidence are separated because a change in any one of them can create a different transition. The architecture therefore rejects implicit substitution: a permission for one object is not a permission for another; a proof for one model is not a proof for a deployment; an approval before mutation is not approval after mutation. Unknown mandatory state is represented explicitly and does not default to ordinary allow.

The chapter’s principal structural commitments are:

Original Contribution - retained as an explicit state or dependency.

Scope Conservation - retained as an explicit state or dependency.

Non-Claim Discipline - retained as an explicit state or dependency.

Research Program - retained as an explicit state or dependency.

Independent Scrutiny - retained as an explicit state or dependency.

Execution-Bound Governability - retained as an explicit state or dependency.

100.3 Execution Reality

Execution reality begins at the last point from which the protected consequence can still be prevented. Upstream analysis, documentation, monitoring, or recommendation may improve decision quality, but it becomes governance only when the consequence path depends on the resulting state. The chapter therefore treats direct capability, alternate interfaces, administrative paths, recovery mechanisms, stale state, and external dependencies as part of the governed object. A system is not credited with a property merely because a dashboard displays it; the relevant gate, controller, institution, or physical interface must enforce or evidence it.

100.4 Failure Point and Counter-Construction

The claim collapses under any in-scope counter-construction such as:

novelty is claimed without comparison

limitations are hidden

open problems are presented as solved

protected detail is used to shield unsupported claims

These failures are not edge cases external to the theory. They are the primary falsification instruments. Where one of them remains reachable, the correct result is a narrowed claim, an unresolved state, or a failed invariant - not a narrative assertion that the architecture is nevertheless governed.

100.5 Verification and Evidence Obligations

compare contributions against prior fields

state unresolved dependencies

define a falsifiable program of work

maintain claim and disclosure integrity

Evidence must remain bound to the exact artifact, configuration, version, operating envelope, and decision state examined. A passing result establishes only the tested property and coverage. A failed or indeterminate result remains part of the authoritative record. Where independent reproduction has not occurred, the result is not represented as independently established.

100.6 Selected Propositions

P100.1. The validity of research agenda and program of work cannot exceed the scope of its weakest mandatory dependency.

P100.2. A descriptive or advisory representation does not become an execution constraint until it controls the relevant consequence path.

P100.3. Component-level validity does not compose automatically across interfaces, timing, authority, failure, or deployment change.

P100.4. Evidence of absence is not inferred from absence of evidence; unresolved mandatory state remains unresolved.

100.7 Governance Status and Confidence Type

STRUCTURE. Specified as part of the integrated federated architecture.
EXECUTION REALITY. Requires a domain-specific implementation and complete path inventory.
FAILURE POINT. Any surviving bypass, stale authority, omitted dependency, unmodeled timing path, or unsupported evidence link narrows or defeats the claim.
GOVERNANCE STATUS. M1 unless a chapter-specific case study states a higher internal implementation maturity; no institutional compliance determination is inferred.
CONFIDENCE TYPE. Definitional for the architecture, deductive only under stated assumptions, implementation-dependent for runtime and hardware, institutionally imported for authority, and evidence-dependent for deployment claims.

Chapter 101

Final Thesis: Structural Sovereignty as Execution-Bound Governability

The integrated conclusion, the master invariant, the conditions for bounded claims, and the distinction between advisory governance and governed execution.

101.1 Objective

The objective of this chapter is to define final thesis: structural sovereignty as execution-bound governability as a bounded component of execution-bound governance. The analysis connects original contribution, scope conservation, non-claim discipline, research program, independent scrutiny, and execution-bound governability. The decisive test is not whether the system can describe the relevant rule, but whether the rule changes the set of reachable consequence-bearing transitions. The synthesis preserves the distinction between an original architectural contribution and a completed deployment-level proof.

101.2 Structural Model

A compact representation used in this chapter is:

$$Exec(\tau) \Rightarrow Permit^{\dot{c}}(\tau)$$

The structural object is represented as a typed relation rather than an informal checklist. Identity, object, context, authority, version, time, fault state, and evidence are separated because a change in any one of them can create a different transition. The architecture therefore rejects implicit substitution: a permission for one object is not a permission for another; a proof for one model is not a proof for a deployment; an approval before mutation is not approval after mutation. Unknown mandatory state is represented explicitly and does not default to ordinary allow.

The chapter's principal structural commitments are:

Original Contribution - retained as an explicit state or dependency.

Scope Conservation - retained as an explicit state or dependency.

Non-Claim Discipline - retained as an explicit state or dependency.

Research Program - retained as an explicit state or dependency.

Independent Scrutiny - retained as an explicit state or dependency.

Execution-Bound Governability - retained as an explicit state or dependency.

101.3 Execution Reality

Execution reality begins at the last point from which the protected consequence can still be prevented. Upstream analysis, documentation, monitoring, or recommendation may improve decision quality, but it becomes governance only when the consequence path depends on the resulting state. The chapter therefore treats direct capability, alternate interfaces, administrative paths, recovery mechanisms, stale state, and external dependencies as part of the governed object. A system is not credited with a property merely because a dashboard displays it; the relevant gate, controller, institution, or physical interface must enforce or evidence it.

101.4 Failure Point and Counter-Construction

The claim collapses under any in-scope counter-construction such as:

novelty is claimed without comparison

limitations are hidden

open problems are presented as solved

protected detail is used to shield unsupported claims

These failures are not edge cases external to the theory. They are the primary falsification instruments. Where one of them remains reachable, the correct result is a narrowed claim, an unresolved state, or a failed invariant - not a narrative assertion that the architecture is nevertheless governed.

101.5 Verification and Evidence Obligations

compare contributions against prior fields

state unresolved dependencies

define a falsifiable program of work

maintain claim and disclosure integrity

Evidence must remain bound to the exact artifact, configuration, version, operating envelope, and decision state examined. A passing result establishes only the tested property and coverage. A failed or indeterminate result remains part of the authoritative record. Where independent reproduction has not occurred, the result is not represented as independently established.

101.6 Selected Propositions

P101.1. The validity of final thesis: structural sovereignty as execution-bound governability cannot exceed the scope of its weakest mandatory dependency.

P101.2. A descriptive or advisory representation does not become an execution constraint until it controls the relevant consequence path.

P101.3. Component-level validity does not compose automatically across interfaces, timing, authority, failure, or deployment change.

P101.4. Evidence of absence is not inferred from absence of evidence; unresolved mandatory state remains unresolved.

101.7 Governance Status and Confidence Type

STRUCTURE. Specified as part of the integrated federated architecture.
EXECUTION REALITY. Requires a domain-specific implementation and complete path inventory.
FAILURE POINT. Any surviving bypass, stale authority, omitted dependency, unmodeled timing path, or unsupported evidence link narrows or defeats the claim.
GOVERNANCE STATUS. M1 unless a chapter-specific case study states a higher internal implementation maturity; no institutional compliance determination is inferred.
CONFIDENCE TYPE. Definitional for the architecture, deductive only under stated assumptions, implementation-dependent for runtime and hardware, institutionally imported for authority, and evidence-dependent for deployment claims.

APPENDICES

Appendix A - Consolidated Notation

Symbol

Meaning

z_t

Federated governed system state at time t

x_t

Physical, logical, or domain state

τ

Candidate transition

A

Admissibility relation or set

A_H

Human or institutional authority object

B

Execution or consequence boundary

F

Declared fault model

Ω

Claim or operating envelope

CE

Commit-enable signal

$HALT_N$

Active-low non-maskable halt signal

$M(C)$

Evidence maturity of claim C

E

Evidence set or evidence relation

U_G

Governed admissible control set

K

Safe or viable state set

$W^{¿}$

Winning region of a governance game

T_{irr}

Time at which the consequence becomes irreversible

$\preceq$

Authority or capability attenuation relation

Appendix B - Formal Correction Ledger

The source corpus contained several useful but over-broad formulations. The integrated edition adopts the following corrected forms.

Damped state equation. The expression

$$\dot{S} = R - \gamma S$$

has equilibrium $S^{\dot{c}} = R/\gamma$ only when R is constant and the remaining assumptions of the scalar model hold.

Halt semantics. A signal whose value one means safe or enabled is a permission gate, not a halt variable. Polarity and dominance are declared explicitly.

Spectral stability. $\rho(A) < 1$ is appropriate for discrete linear or locally linearised dynamics. Continuous local asymptotic stability requires eigenvalues with negative real parts; nonlinear claims require an identified Jacobian, linearisation, Lyapunov, or barrier argument.

Structural load. $L(t) = L_0 + \int \phi - S(t)$ is not necessarily monotone when shedding or recovery subtracts load. Monotonicity is never assumed without an explicit derivative condition.

Authority continuity. Authority is preserved through authenticated valid transitions, not through the unrealistic identity $A(t_1, v_1) = A(t_2, v_2)$ across all time and versions.

Load-resilient governance. The claim is $C(L) \geq C_{min}$ inside a certified envelope, not $\partial C / \partial L = 0$ universally.

Unsafe-state language. “Unsafe states are never entered” is limited to states classified as non-admissible inside the declared model, enforcement boundary, and fault envelope.

Confidence thresholds. No fixed scalar confidence threshold is treated as deterministic governance without calibration, consequence semantics, and a validated decision relation.

Diagnostic model self-report. Model responses to questionnaires are exploratory behavioural evidence, not proof of architecture or authority.

Validation language. “Confirmed” or “passed” has no academic force without a defined object, method, expected result, observed result, scope, and evidence.

Cryptographic representation. Full and truncated digests are distinguished. A truncated display is never represented as the complete hash value.

Licensing. CC BY 4.0 is suitable for documentation but does not automatically license software, data, model weights, or hardware design.

Appendix C - Public-Safe Architecture Register

Interlink Bridge

Interlink Bridge is the independent research interface under which the architecture is developed. In this manuscript it functions as an authorship and research-programme context, not as a claim of regulatory or institutional authority.

LIAN - Logical Integrity Architecture Node

LIAN is represented as a pre-execution integrity and admissibility layer. It consumes authority, object, context, version, state, and evidence predicates before a transition may reach the commit boundary. LIAN is not a model and does not generate its own mandate.

PoA Stack

The public-safe PoA Stack is treated as four interface classes:

PoA-01 Reachability: whether a consequence path exists.

PoA-02 Presence: whether a principal, object, and context are validly present.

PoA-03 Commit Integrity: whether the approved and committed transitions are identical.

PoA-04 Structural Reciprocity: whether authority, responsibility, and evidence remain mutually bound.

SLI - Structural Logic Integrity Plane

SLI is the runtime governance plane that maps structural state into typed decisions such as allow, delay, throttle, handoff, and halt. Its load model is bounded and non-monotonic where recovery or shedding occurs. No public claim is made that one scalar load metric captures all governance state.

Governed Co-Chip

The Co-Chip is an architectural hardware role outside the principal compute plane. It consumes a bounded permit representation, validates local health and version state, controls a protected effect path, asserts a non-maskable halt, and emits independent evidence. No fabricated production ASIC or certified vehicle integration is claimed.

KWI LIAN and Kyber

KWI LIAN and Kyber provide an applied local-workflow case study involving document operations, case state, explicit human commit, governance metadata, local infrastructure, and multi-client handover. The case supports internal implementation evidence for bounded components but does not establish universal deployment or legal compliance.

Appendix D - Consolidated Verification Outcome Vocabulary

Permitted result states are:

PROVED - deductively established within a specified formal model.

REFUTED - a valid counterexample defeats the stated claim.

BOUNDED SAFE - established for a finite horizon or envelope.

ABSTRACTLY SAFE - established in a sound abstraction, subject to correspondence.

STATISTICALLY SUPPORTED - supported by a stated statistical method and sample.

INTERNALLY TESTED - repeatable internal evidence exists.

INDEPENDENTLY REPRODUCED - a separate party reproduces the result.

INDETERMINATE - available evidence cannot determine the result.

INVALID TEST - the method, object, or environment does not support the conclusion.

NOT EXECUTED - planned but not run.

OUT OF SCOPE - outside the declared claim envelope.

BLOCKED - a dependency prevents execution of the test.

“No counterexample found” is not synonymous with “proved”.

Appendix E - Integrated Validation Matrix

Layer

Primary object	**Principal evidence**	**Typical falsifier**
Formal		
Governed transition model		
	proof certificate, model-checking trace	
		reachable non-permitted transition
Control		
Plant-governor-actuator system		
	invariant, barrier, timing, reachability evidence	
		empty feasible set or late intervention
Runtime		
Transaction and consequence path		
	state-machine, concurrency, crash, capability tests	
		alternate API or duplicate consequence
Hardware		
Physical gate and signal path		
	RTL, timing, board, fault, HIL evidence	
		maskable halt or physical bypass
Human authority		
Principal, mandate, commit		
	identity, mandate, interface, timing evidence	
		nominal approval without control
Governance		
Source-to-control mapping		
	source, interpretation, control, audit, remedy	
		policy text with no enforcement
Applied system		
Domain instance		
	end-to-end case evidence	
		boundary or authority mismatch
Deployment		
Active artefact and context		
	attestation, configuration, drift monitoring	
		unvalidated version or topology

Appendix F - Research and Publication Governance

The publication track follows these rules:

No external release is made without explicit author approval of the exact artefact.

Public material contains no protected bypass-sensitive mechanism.

Protected status never converts an unverified claim into a verified one.

Documentation, software, data, models, and hardware artefacts receive separate licence treatment.

Every release records version, date, claim envelope, maturity, supersession state, and full digest.

Zenodo or similar archival registration establishes a persistent public record, not independent validation.

Appendix G - Selected Internal Research Records

The following DOI records are relevant to the broader Interlink Bridge research programme and should be checked against the current Zenodo metadata before formal citation:

10.5281/zenodo.18980560

10.5281/zenodo.19048265

10.5281/zenodo.19076268

10.5281/zenodo.20171212

10.5281/zenodo.21569313

10.5281/zenodo.21610222

10.5281/zenodo.21667112

Appendix H - Selected Bibliography

Alur, R., and Dill, D. L. (1994). A theory of timed automata. *Theoretical Computer Science*, 126(2), 183-235.

Ames, A. D., Xu, X., Grizzle, J. W., and Tabuada, P. (2017). Control barrier function based quadratic programs for safety critical systems. *IEEE Transactions on Automatic Control*, 62(8), 3861-3876.

Ashby, W. R. (1956). *An Introduction to Cybernetics*. Chapman & Hall.

Baier, C., and Katoen, J.-P. (2008). *Principles of Model Checking*. MIT Press.

Bainbridge, L. (1983). Ironies of automation. *Automatica*, 19(6), 775-779.

Bertalanffy, L. von. (1968). *General System Theory*. George Braziller.

Clarke, E. M., Grumberg, O., and Peled, D. A. (1999). *Model Checking*. MIT Press.

Endsley, M. R. (1995). Toward a theory of situation awareness in dynamic systems. *Human Factors*, 37(1), 32-64.

Herlihy, M. P., and Wing, J. M. (1990). Linearizability: A correctness condition for concurrent objects. *ACM Transactions on Programming Languages and Systems*, 12(3), 463-492.

Khalil, H. K. (2002). *Nonlinear Systems* (3rd ed.). Prentice Hall.

Lamport, L. (1978). Time, clocks, and the ordering of events in a distributed system. *Communications of the ACM*, 21(7), 558-565.

Leveson, N. G. (2011). *Engineering a Safer World*. MIT Press.

NIST (2023). *Artificial Intelligence Risk Management Framework (AI RMF 1.0)*. National Institute of Standards and Technology. https://www.nist.gov/itl/ai-risk-management-framework

Parasuraman, R., Sheridan, T. B., and Wickens, C. D. (2000). A model for types and levels of human interaction with automation. *IEEE Transactions on Systems, Man, and Cybernetics - Part A*, 30(3), 286-297.

Ramadge, P. J., and Wonham, W. M. (1987). Supervisory control of a class of discrete event processes. *SIAM Journal on Control and Optimization*, 25(1), 206-230.

Rawlings, J. B., Mayne, D. Q., and Diehl, M. (2017). *Model Predictive Control: Theory, Computation, and Design* (2nd ed.). Nob Hill.

Regulation (EU) 2024/1689 of the European Parliament and of the Council of 13 June 2024 laying down harmonised rules on artificial intelligence. Official EUR-Lex record: https://data.europa.eu/eli/reg/2024/1689/oj

Saltzer, J. H., and Schroeder, M. D. (1975). The protection of information in computer systems. *Proceedings of the IEEE*, 63(9), 1278-1308.

Senke, A. (2026). *Source Corpus 2026 - Structural Sovereignty and Execution Governance*. Internal 250-page research corpus, Interlink Bridge.

Senke, A. (2026). *Structural Sovereignty: Constraint as Substrate - Architectural Conditions for Governable Systems at Scale*. Independent architectural research manuscript.

Senke, A. (2026). *Hardware-Enforced Admissibility Boundaries for Automotive AI Execution: A Structural Governance Architecture*. Public-safe research paper and presentation manuscript.

Tabuada, P. (2009). *Verification and Control of Hybrid Systems: A Symbolic Approach*. Springer.

Wiener, N. (1948). *Cybernetics: Or Control and Communication in the Animal and the Machine*. MIT Press.

W3C (2013). *PROV-O: The PROV Ontology*. W3C Recommendation. https://www.w3.org/TR/prov-o/

ISO 26262 series (2018). *Road vehicles - Functional safety*. International Organization for Standardization. Current official catalogue: https://www.iso.org/

ISO 21448:2022. *Road vehicles - Safety of the intended functionality*. International Organization for Standardization. Current official catalogue: https://www.iso.org/standard/77490.html

Appendix I - Open Research Programme

The manuscript remains open in the following mandatory areas:

Mechanized end-to-end proof of the federated permit across formal, control, runtime, hardware, authority, and governance layers.

Complete consequence-path discovery under administrative, recovery, supplier, and physical bypass conditions.

Demonstration of safe-state reachability under bounded disturbance, delay, and actuator degradation.

Independent reproduction of the public-safe runtime and hardware claims.

Human-factors validation of intervention margin, workload, consent, accessibility, and display-commit equivalence.

Current jurisdiction-specific legal mapping by competent counsel or institution.

Supplier-exit, state-portability, and external-permission dependency testing.

Public-safe Co-Chip demonstrator with protected implementation review.

Multi-agent authority and constraint-composition proofs.

Independent academic critique and adversarial falsification.

Appendix J - Final Claim Boundary

This manuscript establishes an integrated M1 architecture and a bounded research programme. Selected software and workflow components may possess internal M2-M3 evidence, but the complete federated system has not been independently reproduced, independently assessed, legally determined, fabricated as production hardware, or certified for vehicle or other safety-critical deployment.

The strongest justified final statement is therefore:

> *Structural Sovereignty is a candidate federated theory of governability in which authority, admissibility, feasibility, mediation, physical enforcement, evidence, and remedy jointly determine whether a consequence-bearing transition can exist. The theory is specified in a falsifiable and cross-disciplinary form. Deployment-level validity remains conditional on the discharge of the proof, implementation, validation, authority, and institutional obligations stated throughout this manuscript.*