Governance Inside the Agent Loop: Exact Nonblocking Control under Partial Observation and Dynamic Authority

FORMAL METHODS | AGENTIC SYSTEMS | RUNTIME GOVERNANCE | CERTIFIED CONTROL

ABSTRACT

An agent can become committed to an unsafe course before making a tool call by adopting an unsafe plan, reusing stale memory, delegating responsibility, acting under changed authority, or combining individually acceptable steps into a harmful sequence. We introduce a finite-state model in which every represented governable effect is wrapped by an atomic proposal–decision–commit macro-step, governors observe different causal summaries, authority varies with runtime state, and commands use a deployed interface drawn from a fixed catalogue of finite intervention transducers. The target is to permit exactly the next actions that remain inside the greatest uncontrollable-closed region from which an acceptable marked completion is reachable without leaving that region. The paper makes four linked contributions:

- We prove exact admitted-language instrumentation and reduce general fused governance to finite constraints over observation, authority, profile effects, and fusion.
- For transparent route-bearing governance, we distinguish exactness of the selected route from the stronger independently valid semantics and prove a five-state cross-layer noncomposition result for the actual deployment.
- We instantiate hitting-set certificates for authorised exact routes over complete information fibres and prove that independently justified current-bound certificates compose to exact governance under explicit coverage and commit-revalidation assumptions.
- We define universally verified replay regions and prove a preservation invariant: inserting a verified record cannot change an earlier certified replay decision and can only enlarge verified replay coverage.

1 Introduction

Large-language-model agents repeatedly observe, retrieve memory, form plans, delegate, invoke tools, incorporate results, and revise their working state. A dangerous execution may therefore become committed before its final external effect. A stale approval can be imported by procedural memory; a planning decision can remove every compliant completion; a risk agent can detect a hazard without possessing authority to intervene; or an authorised reviewer can lack the information needed to select the correct remedy. A tool-call gate may stop one effect, but it does not by itself preserve the acceptable alternatives that existed earlier in the execution. The formal model covers an internal plan, memory update, or delegation only when that commitment is represented as a state or controlled event in the instrumented runtime.

This paper studies the following question.

> *When can a faithfully instrumented agent runtime, governed through a fixed implementable intervention library by partially informed actors whose authority changes with runtime state, preserve exactly the continuations that remain capable of reaching an acceptable completion?*

The word *exactly* separates the problem from conservative suppression. Blocking every controlled action may remove all unsafe branches but also destroy viable completion routes. We therefore combine a safety envelope with marked nonblockingness: a continuation is viable only when its successor remains in the greatest region closed under exogenous behaviour and retaining a path to an acceptable marked completion without leaving that region.

The formal development is built on two earlier boundaries. Kingston's history-dependent veto-authority theory shows that decision knowledge and authority must be aligned in one issuer's information cell; separate witnesses need not compose [11]. Further, his equality-support theory shows that a safe fixed action library may still fail to reproduce a target branch set exactly, and that exact repertoire deployment has a covering structure [12]. The present model combines these boundaries through an instrumented nonblocking runtime, implementable multi-valued intervention transducers, explicit command fusion, a transparent-layer noncomposition result, governance-specific proof certificates, and a sound incremental memory semantics.

The principal contributions are:

1. **Faithful instrumentation and exact nonblocking realisability.** We relate an original runtime to a proposal–decision–commit wrapper and prove exact equality between its projected trace language and the admitted original language, with no represented controlled-effect bypass. We then give the finite cell-command constraints for general fused governance.

2. **Transparent realisability hierarchy and noncomposition.** We distinguish route-bearing transparent exactness from the stronger independently valid class, in which every emitted non-abstaining route is individually authorised and exact throughout its issuer's information cell. A five-state construction is exact under each of two layer-specific counterfactual repairs, yet the actual deployment has no route-bearing transparent exact realisation and therefore no independently valid one.

3. **Proof-carrying aligned witnesses.** We define a joint authority-and-effect predicate for a governor–transducer route. Its certificates use established hitting-set duality, while current-bound acceptance connects a universally certified route to the authenticated proposal, request, version, context, and runtime state. Under explicit coverage and revalidation assumptions, independently justified accepted routes compose to exact governance.

4. **Certified governance-memory invariant.** We define universally verified replay regions rather than assume that embedding similarity or a request abstraction is globally sound. Inserting a verified record cannot change an earlier certified replay decision and can only enlarge verified replay coverage.

The paper's main new object is a governor–profile route that must be simultaneously authorised and exact throughout the governor's complete causal information fibre; the principal results connect this aligned route condition to a strict hierarchy of transparent realisability, cross-layer noncomposition, current-bound proof acceptance, and exact marked-nonblocking execution. This is a theory paper: the finite regression suite in Appendix C specifies a proposed companion verification artifact, but no implementation or empirical artifact is claimed here. Set Cover, sufficient-reason/hitting-set duality, counterexample-guided hitting-set optimisation, and outcome-homogeneous cache factorisation are supporting machinery rather than independent novelty claims.

2 Relation to Existing Theory

Supervisory control and decentralised observation. Classical supervisory control represents intervention through admissible control patterns and studies controllable, nonblocking, and maximally permissive behaviour [20]. Partial-observation theory requires decisions to remain consistent across observationally indistinguishable executions [18], while decentralised control distributes those decisions across local supervisors [22]. Restricted and arbitrary control-pattern theory asks whether a target language remains achievable when only a prescribed family of decisions is available under partial observation [25, 26, 27]; a recent security application reduces robust supervision under sensor and actuator deception to this restricted-pattern setting [16]. Explicit decision-fusion architectures combine local commands [19, 37]; dynamic and conditional variants can realise behaviours unavailable under more restrictive static fusion rules [35, 36]. Recent event-forcing control under partial observation likewise requires forcing decisions to remain observation-consistent [28]. The selector-dependent expressive gain exhibited later is therefore not claimed as new. Relative to this lineage, the present model couples a named fixed transducer effect with state-dependent issuer authority over the same complete causal information fibre, separates catalogue membership from deployment, and introduces an independently valid audit class in which exactness cannot depend on a hidden dynamic selector rescuing locally nonexact routes.

Runtime enforcement and assurance. Security automata mediate traces by termination; edit automata enlarge enforcement power through event suppression and insertion [23, 13]. Shield synthesis and Simplex-style runtime assurance place verified correction or fallback around a less trusted component and seek to limit unnecessary deviation [3, 1, 15]. Agent-specific systems include rule-based runtime constraints in AgentSpec, proactive enforcement through probabilistic reachability analysis in Pro2Guard, path-dependent policy evaluation before proposed actions, and explicit enforcement sites inside the agent loop in SARC [29, 30, 6, 31]. The Agent Control Protocol formalises cryptographic action admission, while the Reconstructive Authority Model separates authenticated state integrity from coverage under partial observability [4, 5]. These works address runtime policy evaluation, action admission, or execution validity. The present target is instead a declared exact marked-nonblocking successor set under distributed observation, dynamic authority, fixed transducer effects, and complete-fibre route validity.

Proof-bearing execution and authorisation. Proof-carrying code established the producer–consumer pattern in which an untrusted artefact is accompanied by a checkable safety proof, while proof-carrying authorisation applied submitted proofs directly to access decisions [17, 2]. Recent agent-governance work attaches portable evidence to heterogeneous actions, derives execution authority from structured proof artefacts, composes multiple authority sources into verifiable receipts, or validates authorisation, path integrity, history, and replay through execution attestations [34, 7, 32, 33]. The certificate developed here has a narrower formal semantics: it proves that one deployed governor–profile route is authorised and has exactly the required controlled successor effect throughout the complete observation fibre, while a separate current-execution check prevents an authentic but stale certificate from being replayed in a different context.

Formal explanation. A subset of input features that guarantees the same output under all completions is a formal abductive explanation or sufficient reason. Contrastive explanations describe feature changes capable of changing the output. Their minimal-hitting-set duality follows a line from model-based diagnosis through formal XAI [21, 24, 8, 9, 14]. We use this established duality for a joint predicate that simultaneously represents observation-cell consistency, authority legitimacy, and

exact intervention effect, and then connect the resulting certificate to global runtime realisability and governance-memory growth.

Dynamic authority and fixed exact interfaces. History-dependent authority constrains which command an issuer may legitimately emit at a target-reachable decision point [11]. Equality-support covers constrain which exact successor filters are available in a predeclared actuator library [12]. The new interaction is fibrewise: a deployed transducer must be exact and legitimately issuable throughout the same causal observation cell, and the actual deployed interface and actual authority architecture may fail together even when either projection is adequate in isolation.

3 Finite Runtime and Nonblocking Viability

3.1 Original runtime

Let

$$\mathcal{A} = (S, s_0, \Sigma, \delta, S_m)$$

be a finite deterministic partial transition system, where:

- S is a nonempty finite set of runtime states;
- $s_0 \in S$ is the initial runtime state;
- Σ is a finite event alphabet;
- $\delta : S \times \Sigma \rightharpoonup S$ is a partial transition function; and
- $S_m \subseteq S$ is the set of acceptable marked completion states.

The symbol $\rightharpoonup$ denotes a partial function. Let Σ^* be the set of all finite words over Σ, and let $\varepsilon \in \Sigma^*$ be the empty word. Extend δ to the partial word-transition function

$$\delta^* : S \times \Sigma^* \rightharpoonup S$$

by $\delta^*(s, \varepsilon) = s$ for $s \in S$ and, for $s \in S$, $w \in \Sigma^*$, and $\sigma \in \Sigma$,

$$\delta^*(s, w\sigma) = \delta\big(\delta^*(s, w), \sigma\big)$$

whenever both terms on the right are defined. A *run from s along w* is defined exactly when $\delta^*(s, w)$ is defined. For any transition system $\mathcal{T}$ with a designated initial state, $L(\mathcal{T})$ denotes its finite trace language; in particular,

$$L(\mathcal{A}) = \{w \in \Sigma^* : \delta^*(s_0, w) \text{ is defined}\}.$$

For a set X, 2^X denotes its power set; for a finite set X, $|X|$ denotes its cardinality.

The event alphabet is partitioned as the disjoint union

$$\Sigma = \Sigma_c \,\dot\cup\, \Sigma_u,$$

where Σ_c is the set of governable commitments and Σ_u is the set of exogenous events that the governance layer cannot suppress. For each $s \in S$, define

$$\mathrm{Av}(s) = \{\sigma \in \Sigma : \delta(s, \sigma) \text{ is defined}\}, \quad \mathrm{Av}_c(s) = \mathrm{Av}(s) \cap \Sigma_c, \quad \mathrm{Av}_u(s) = \mathrm{Av}(s) \cap \Sigma_u.$$

Thus $\mathrm{Av}(s)$, $\mathrm{Av}_c(s)$, and $\mathrm{Av}_u(s)$ are respectively the available, governable, and exogenous events at state s. A set $Q \subseteq S$ is the declared safe state envelope, and we assume $S_m \subseteq Q$. The runtime state contains any finite policy, workflow, budget, or monitor product needed by the specification.

Definition 3.1 (Admissible nonblocking region)**.** A set $R \subseteq Q$ is *admissible* if:

(i) for every $s \in R$ and $\sigma \in \mathrm{Av}_u(s)$, one has $\delta(s, \sigma) \in R$; and

(ii) for every $s \in R$, there is a finite word $w \in \Sigma^*$ such that the run from s along w is defined, every visited state lies in R, and the final state lies in $S_m \cap R$.

Condition (i) is uncontrollable closure. Condition (ii) is standard marked nonblockingness: every state retains at least one completion path. It does not require every infinite execution to terminate.

Let $\mathfrak{R}$ be the family of admissible regions and define

$$W = \bigcup_{R \in \mathfrak{R}} R.$$

Lemma 3.2 (Greatest admissible region). *The set W is admissible and contains every admissible region. Hence it is the unique greatest uncontrollable-closed marked-nonblocking subset of Q.*

Proof. Greatestness is immediate from the union definition. For uncontrollable closure, let $s \in W$. Then $s \in R$ for some admissible R. Every uncontrollable successor of s belongs to R, hence to W. For marked coaccessibility, the same region R contains a path from s to $S_m \cap R$, and that path lies in W. Thus W is admissible. ∎

We assume $s_0 \in W$; otherwise no controller can both remain inside Q, respect every exogenous event, and preserve marked nonblockingness from the initial state.

Definition 3.3 (Exact viable successor set). For $s \in W$, define

$$\mathrm{Viab}(s) = \{\sigma \in \mathrm{Av}(s) : \delta(s,\sigma) \in W\}, \qquad \mathrm{Viab}_c(s) = \mathrm{Viab}(s) \cap \Sigma_c.$$

By Lemma 3.2, the greatest admissible region is uncontrollable-closed:

$$s \in W,\ \sigma \in \mathrm{Av}_u(s) \implies \delta(s,\sigma) \in W.$$

Exact governance therefore concerns the controlled part $\mathrm{Viab}_c(s)$: it should retain every controlled transition that stays in the greatest nonblocking region and reject every controlled transition that leaves it.

3.2 A marked proposal–decision–commit wrapper

Fix an admission map

$$\mu : W \to 2^{\Sigma_c} \quad \text{with} \quad \mu(s) \subseteq \mathrm{Av}_c(s).$$

The instrumented runtime has ready, pending, and cleared states:

$$\widehat{S}_\mu = S \,\dot\cup\, \{p_{s,a} : s \in W,\ a \in \mathrm{Av}_c(s)\} \,\dot\cup\, \{c_{s,a} : s \in W,\ a \in \mu(s)\}.$$

A ready state is an original state $s \in S$. The pending state $p_{s,a}$ records that controlled event a has been proposed at s; the cleared state $c_{s,a}$ records that the proposal has been permitted but not yet committed. Define

$$\widehat{S}_m = S_m$$

and regard marked states as marked ready states. Let

$$\Lambda = \{\mathsf{prop}(a), \mathsf{reject}(a), \mathsf{permit}(a), \mathsf{commit}(a) : a \in \Sigma_c\}$$

be fresh instrumentation labels, let $\widehat{\Sigma} = \Sigma_u \,\dot\cup\, \Lambda$, and define

$$\widehat{\mathcal{A}}_\mu = (\widehat{S}_\mu, s_0, \widehat{\Sigma}, \widehat{\delta}_\mu, \widehat{S}_m).$$

For wrapper states $x, x' \in \widehat{S}_\mu$ and a label $\ell \in \widehat{\Sigma}$, the notation $x \xrightarrow{\ell} x'$ abbreviates $\widehat{\delta}_\mu(x,\ell) = x'$. The transition relation is exactly

$$\begin{aligned}
s &\xrightarrow{\sigma} \delta(s,\sigma), && s \in S,\ \sigma \in \mathrm{Av}_u(s),\\
s &\xrightarrow{\mathsf{prop}(a)} p_{s,a}, && s \in W,\ a \in \mathrm{Av}_c(s),\\
p_{s,a} &\xrightarrow{\mathsf{reject}(a)} s, && a \notin \mu(s),\\
p_{s,a} &\xrightarrow{\mathsf{permit}(a)} c_{s,a}, && a \in \mu(s),\\
c_{s,a} &\xrightarrow{\mathsf{commit}(a)} \delta(s,a), && a \in \mu(s).
\end{aligned}$$

An admitted proposal cannot be rejected in this wrapper. That is a deterministic exact-admission design choice, not a consequence of faithful instrumentation.

The proposal–decision–commit segment is one atomic macro-step at the abstraction used for supervisory control. Exogenous events are not assumed absent in physical time. If an exogenous occurrence can interleave with review, its transitions must be represented from the pending and cleared states, and those transitions must be included when computing uncontrollable closure. Authority leases, timeouts, proposal hashes, and other expiring premises must likewise be represented in the finite state product and revalidated at commit.

Define the label projection $\mathrm{proj} : \widehat{\Sigma} \to \Sigma \cup \{\varepsilon\}$ by

$$\mathrm{proj}(\sigma) = \sigma \quad (\sigma \in \Sigma_u), \qquad \mathrm{proj}(\mathsf{commit}(a)) = a \quad (a \in \Sigma_c),$$

and

$$\mathrm{proj}(\mathsf{prop}(a)) = \mathrm{proj}(\mathsf{permit}(a)) = \mathrm{proj}(\mathsf{reject}(a)) = \varepsilon.$$

Extend proj homomorphically to $\widehat{\Sigma}^*$, erasing occurrences of ε.

For $w = \sigma_1 \cdots \sigma_k \in L(\mathcal{A})$ and $1 \le \ell \le k$, let $w_{<\ell} = \sigma_1 \cdots \sigma_{\ell-1}$ and $s_\ell = \delta^*(s_0, w_{<\ell})$. Define

$$L_\mu(\mathcal{A}) = \left\{ w \in L(\mathcal{A}) : \begin{array}{l} \text{for every controlled position } \ell,\ s_\ell \in W, \\ \text{and } \sigma_\ell \in \mu(s_\ell) \end{array} \right\}. \tag{AL}$$

Thus an admitted trace may reach $S \setminus W$ through a nonviable admitted controlled event when μ is nonexact. No controlled proposal is available while the ready state remains outside W, although an exogenous transition may later return the runtime to W, where controlled proposals are again available. In the exact wrapper below, no reachable ready state leaves W.

Theorem 3.4 (Faithful marked instrumentation). *For every admission map μ:*

(i)

$$\mathrm{proj}(L(\widehat{\mathcal{A}}_\mu)) = L_\mu(\mathcal{A}) \subseteq L(\mathcal{A});$$

(ii) every projected controlled event is the image of a commit event preceded, without an intervening ready state, by its matching proposal and permit events.

Hence the constructed wrapper is sound, has no represented controlled-effect bypass, and is exactly complete for the admitted original language.

Proof. For the forward inclusion in (i), exogenous wrapper transitions reproduce transitions of $\mathcal{A}$, while a commit from $c_{s,a}$ reaches exactly $\delta(s,a)$; all remaining wrapper transitions project to ε. Moreover, a commit can occur only after a proposal from a ready state $s \in W$ and a permit for $a \in \mu(s)$. Induction on wrapper-trace length therefore gives a projected trace in $L_\mu(\mathcal{A})$.

For the reverse inclusion, lift a trace of $L_\mu(\mathcal{A})$ inductively. Lift each exogenous event directly. At a controlled position ℓ, condition (AL) gives both $s_\ell \in W$ and $\sigma_\ell \in \mu(s_\ell)$, so replace σ_ℓ by

$$\mathsf{prop}(\sigma_\ell)\,\mathsf{permit}(\sigma_\ell)\,\mathsf{commit}(\sigma_\ell).$$

All three wrapper edges are defined. Finally, (ii) follows because the only transition labelled $\mathsf{commit}(a)$ leaves $c_{s,a}$, the only incoming transition to $c_{s,a}$ is $\mathsf{permit}(a)$ from $p_{s,a}$, and the only transition entering $p_{s,a}$ from a ready state is $\mathsf{prop}(a)$. ∎

Define

$$L_W = \{w \in L(\mathcal{A}) : \text{every state visited by } w \text{ lies in } W\}.$$

Corollary 3.5 (Exact marked-nonblocking wrapper). *If $\mu(s) = \mathrm{Viab}_c(s)$ for every $s \in W$, then*

$$\mathrm{proj}(L(\widehat{\mathcal{A}}_\mu)) = L_W.$$

Every reachable wrapper state lies on a wrapper path to $\widehat{S}_m$. Consequently $\widehat{\mathcal{A}}_\mu$ is a marked nonblocking transition system.

Proof. When $\mu = \mathrm{Viab}_c$, uncontrollable closure of W and the definition of Viab_c give $L_\mu(\mathcal{A}) = L_W$. The language equality follows immediately from Theorem 3.4(i).

It remains to prove coaccessibility of every reachable wrapper state. A reachable ready state s lies in W and therefore has a path inside W to S_m. A rejected pending state $p_{s,a}$ returns to s. An admitted pending state proceeds through permit and commit to $\delta(s,a) \in W$. A cleared state commits to the same viable successor. Lifting the remaining admissible path reaches the corresponding marked ready state in $\widehat{S}_m$. ∎

4 Fixed Intervention Transducers and Distributed Fusion

4.1 Implementable fixed profiles

Let Γ be a finite alphabet of authenticated public state labels and $\gamma : S \to \Gamma$ the public-label map. Let $\overline{\mathcal{U}}$ be a nonempty finite ambient catalogue of intervention-profile identifiers, and fix a nonempty deployed interface

$$\varnothing \neq \mathcal{U} \subseteq \overline{\mathcal{U}}.$$

Each catalogue profile $u \in \overline{\mathcal{U}}$ is a predeclared finite transducer

$$\mathcal{I}_u = (R_u, r_u^0, \eta_u, \lambda_u),$$

where R_u is a nonempty finite state set, $r_u^0 \in R_u$,

$$\eta_u : R_u \times \Gamma \to R_u$$

is a deterministic update function, and

$$\lambda_u : R_u \times \Gamma \to 2^{\Sigma_c}$$

is an output function.

We use the state-labelled public-input convention. For each $u \in \overline{\mathcal{U}}$, the runtime product contains a projection $r_u : S \to R_u$ satisfying

$$r_u(s_0) = r_u^0 \tag{T0}$$

and, for $s, s' \in S$ and $\sigma \in \Sigma$, whenever $\delta(s, \sigma) = s'$,

$$r_u(s') = \eta_u(r_u(s), \gamma(s')). \tag{TU}$$

If two histories reaching an otherwise identical application state would yield different transducer states, the product construction distinguishes those histories as different elements of S. A claimed profile is *implementable* only when (T0)–(TU) hold on every defined transition of the finite model.

For an implementable $u \in \overline{\mathcal{U}}$ and $s \in S$, its controlled effect is

$$E_u(s) = \lambda_u(r_u(s), \gamma(s)) \cap \mathrm{Av}_c(s) \subseteq \mathrm{Av}_c(s).$$

The transducer, public labels, and recurrence are fixed before the target region W is computed and cannot query Q, S_m, W, or Viab_c.

Because $E_u(s)$ is a subset of already available controlled events, profiles are formally filters. Redirection, rollback, inserted review, simulation, or recovery must therefore appear as controlled events and transitions in the original runtime alphabet before a profile can retain or suppress them.

Catalogue membership, deployment, and authority are distinct. Every $u \in \overline{\mathcal{U}}$ has a fixed implementable effect table, but a runtime policy may issue only profiles in the deployed interface $\mathcal{U}$. Authority may already name an undeployed catalogue profile; that profile becomes issuable only after the deployment is enlarged to a set $\mathcal{U}'$ with $\mathcal{U} \subseteq \mathcal{U}' \subseteq \overline{\mathcal{U}}$. Thus deploying a profile changes the active interface, whereas deploying a governor–profile route changes issuer configuration or authority.

Let I be a nonempty finite set of governors.

4.2 Causal observation, decision-state authority, and effect-aware fusion

For each $i \in I$, let Y_i be a finite observation set and

$$O_i : W \to Y_i$$

a causal observation map induced by authenticated information available no later than the current state. Define

$$D = \{s \in W : \mathrm{Av}_c(s) \neq \varnothing\}.$$

For each $i \in I$, write

$$O_i(D) = \{O_i(s) : s \in D\}$$

for the set of decision observations. Commands are requested only at states in D. Authority is represented by

$$\mathrm{Auth}_i : W \to 2^{\overline{\mathcal{U}} \cup \{\bot\}},$$

with the standing abstention axiom

$$\bot \in \mathrm{Auth}_i(s) \qquad (i \in I,\ s \in D). \tag{AB}$$

A local policy $g_i : Y_i \to \mathcal{U} \cup \{\bot\}$ is legitimate when

$$g_i(O_i(s)) \in \mathrm{Auth}_i(s) \qquad (s \in D). \tag{LG}$$

Authority values outside D may be retained for versioning or later extensions, but they do not constrain a command that is never requested.

For $s \in S$ and $c \in \overline{\mathcal{U}} \cup \{\bot\}$, define the declared command effect

$$\widetilde{E}_c(s) = \begin{cases} E_c(s), & c \in \overline{\mathcal{U}}, \\ \mathrm{Av}_c(s), & c = \bot, \end{cases}$$

so abstention is neutral unless a particular composition rule states otherwise. For any set K, write K^I for the set of functions from I to K, identified with I-indexed tuples.

For every $s \in S$, fix an effect-composition operator

$$\Psi_s : (\overline{\mathcal{U}} \cup \{\bot\})^I \times (2^{\mathrm{Av}_c(s)})^I \longrightarrow 2^{\mathrm{Av}_c(s)}.$$

Write $\Psi = (\Psi_s)_{s \in S}$ for this family. The operators are primitive model data. Their *target independence* is a provenance requirement: the transducer catalogue, public labels, and fusion artefact must be fixed and authenticated before Q, S_m, W, or Viab_c is supplied to the governance synthesis procedure. This construction-history claim is not an extensional property of a function table or circuit and is not implied by the axiom below. A deployment requiring mechanically checkable target blindness must enforce a separate syntactic discipline, such as a signed pre-target artefact or a fusion grammar whose constructors can read only commands and supplied effects.

For $s \in S$ and a command vector $\mathbf{c} = (c_i)_{i \in I}$, write

$$\mathbf{E}_s(\mathbf{c}) = (\widetilde{E}_{c_i}(s))_{i \in I}$$

for its declared effect vector. The transparent characterisation and proof-carrying soundness results do not require an effect-sensitivity axiom. For the circuit-fusion hardness result in Section 9, we record the following local nondegeneracy property:

$$\begin{aligned} \mathrm{EffectAware}(\Psi) \iff\ & \forall s \in S \text{ with } \mathrm{Av}_c(s) \neq \varnothing,\ \forall \mathbf{c} \in (\overline{\mathcal{U}} \cup \{\bot\})^I \text{ with } \exists i\ c_i \neq \bot, \\ & \exists \mathbf{B} \in (2^{\mathrm{Av}_c(s)})^I :\quad B_i = \widetilde{E}_{c_i}(s) \text{ whenever } c_i = \bot, \\ & \qquad\qquad \Psi_s(\mathbf{c}, \mathbf{B}) \neq \Psi_s(\mathbf{c}, \mathbf{E}_s(\mathbf{c})). \end{aligned} \tag{EA}$$

At the declared effect vector of every command combination containing an active issuer, (EA) requires some counterfactual change to the active supplied effects to change the fused result while abstaining components remain fixed. It excludes complete local insensitivity but does not establish target independence, substantive causal dependence on the actual effects, or provenance. Its only theorem-level role is to strengthen the hardness statement in Theorem 9.2; transparent selection instead exposes the selected profile directly through (S2).

For $s \in W$, the resulting fusion map

$$\Phi_s : (\overline{\mathcal{U}} \cup \{\bot\})^I \longrightarrow 2^{\mathrm{Av}_c(s)}$$

is derived from issued commands and their fixed effects by

$$\Phi_s(\mathbf{c}) = \Psi_s\Big(\mathbf{c}, (\widetilde{E}_{c_i}(s))_{i \in I}\Big). \tag{FU}$$

The target may test the resulting map for exactness but may not define the actuator or fusion artefact. All results for general fusion are therefore relative to the supplied operator and the stated provenance assumption; the principal transparent results obtain a stronger extensional guarantee from (S2).

Examples satisfying (FU)–(EA) include:

- *coordinator selection*: a fixed selector chooses a non-abstaining issuer and returns that issuer's effect;
- *intersection with neutral abstention*: intersect the effects of non-abstaining commands, returning $\mathrm{Av}_c(s)$ when all abstain;
- *deny-overrides*: intersect every participating filter, with a designated deny profile contributing $\varnothing$; and
- *fixed precedence*: scan a predeclared issuer order and return the first non-abstaining effect.

Given $g = (g_i)_{i \in I}$, define the induced admission map $\mu_g : W \to 2^{\Sigma_c}$ by

$$\mu_g(s) = \Phi_s((g_i(O_i(s)))_{i \in I}).$$

The policy tuple exactly realises W when every local policy is legitimate on D and

$$\mu_g(s) = \mathrm{Viab}_c(s) \qquad (s \in D). \tag{EX}$$

When no transparent route-attribution restriction is imposed, we call such a tuple *general fused exact*.

4.3 Cell-command characterisation

The following finite characterisation records the exact constraint system induced by observation-based policies and provides the general-fusion baseline before transparent route semantics are imposed.

For each $i \in I$ and decision observation $y \in O_i(D)$, introduce $c_{i,y} \in \mathcal{U} \cup \{\bot\}$. Consider

$$c_{i,O_i(s)} \in \mathrm{Auth}_i(s), \qquad i \in I,\ s \in D, \tag{1}$$

$$\Phi_s((c_{i,O_i(s)})_{i \in I}) = \mathrm{Viab}_c(s), \qquad s \in D. \tag{2}$$

Theorem 4.1 (Transducer-aware fused-governance characterisation)**.** *The following are equivalent:*

(i) a legitimate policy tuple exactly realises W;

(ii) the finite system (1)–(2) is satisfiable.

When these conditions hold, the admission map μ_g generates the exact marked-nonblocking wrapper of Corollary 3.5.

Proof. Given an exact legitimate tuple, set $c_{i,y} = g_i(y)$ for $y \in O_i(D)$. Equations (1) and (2) follow from (LG) and (EX).

Conversely, define $g_i(y) = c_{i,y}$ on $O_i(D)$, and extend g_i arbitrarily to observations outside the decision domain in $Y_i \setminus O_i(D)$. The first constraint proves legitimacy exactly on the command domain D, and the second proves (EX). Corollary 3.5 then supplies exact projection and marked nonblockingness. ∎

The theorem is an extensional assignment–policy equivalence and remains valid for any fixed supplied family Φ_s. Excluding a fusion artefact that hard-codes $\mathrm{Viab}_c(s)$ remains a separate provenance obligation. Transparent selection avoids that ambiguity because (S2) requires the fused result to equal the declared effect of a named profile.

4.4 Transparent single-witness governance

For audit-oriented issuance, for each $s \in W$ fix a transparent selector

$$\mathrm{Sel}_s : (\overline{\mathcal{U}} \cup \{\bot\})^I \to I$$

defined whenever at least one command is non-abstaining and satisfying

$$\mathrm{Sel}_s(\mathbf{c}) \in \{i \in I : c_i \neq \bot\}. \tag{S1}$$

Transparent fusion satisfies

$$\Phi_s(\mathbf{c}) = E_{c_{\mathrm{Sel}_s(\mathbf{c})}}(s) \tag{S2}$$

whenever at least one command is non-abstaining.

For a policy tuple $g = (g_i)_{i \in I}$ and $s \in D$, write

$$\mathbf{c}_g(s) = (g_i(O_i(s)))_{i \in I}.$$

Definition 4.2 (Route-bearing transparent exact policy tuple)**.** A policy tuple $g = (g_i)_{i \in I}$ is *route-bearing transparent exact* when it is legitimate and, for every $s \in D$:

(i) $\mathbf{c}_g(s)$ has at least one non-abstaining component; and

(ii) with

$$j_s = \mathrm{Sel}_s(\mathbf{c}_g(s)),$$

the selected route satisfies

$$E_{g_{j_s}(O_{j_s}(s))}(s) = \mathrm{Viab}_c(s).$$

Nonselected non-abstaining commands need not be individually exact.

Definition 4.3 (Independently valid transparent policy tuple)**.** A transparent policy tuple $g = (g_i)_{i \in I}$ is *independently valid* when, for every $s \in D$:

(i) at least one governor emits a non-abstaining command;

(ii) for every $i \in I$, if $g_i(O_i(s)) = u \neq \bot$, then $u \in \mathrm{Auth}_i(s)$; and

(iii) for every $i \in I$, if $g_i(O_i(s)) = u \neq \bot$, then $E_u(s) = \mathrm{Viab}_c(s)$.

The selector is fixed separately and obeys (S1)–(S2). Since every emitted non-abstaining effect is exact, the selector's choice is immaterial to exactness.

For $s \in D$, define the deployed exact-profile set

$$\mathcal{U}^{=}(s) = \{u \in \mathcal{U} : E_u(s) = \mathrm{Viab}_c(s)\}.$$

For each $i \in I$ and $y \in O_i(D)$, let

$$C_i(y) = \{s \in D : O_i(s) = y\}$$

and

$$\mathcal{U}_i^{AE}(y) = \bigcap_{s \in C_i(y)} \big(\mathrm{Auth}_i(s) \cap \mathcal{U}^{=}(s)\big).$$

The superscript AE denotes simultaneous authority-and-effect alignment over the complete cell.

Corollary 4.4 (Transparent aligned-witness characterisation)**.** *The following are equivalent:*

(i) there exists an independently valid transparent policy tuple;

(ii)

$$\forall s \in D\ \exists i \in I\ \exists u \in \mathcal{U} : \quad u \in \mathcal{U}_i^{AE}(O_i(s)). \tag{AEC}$$

Whenever these conditions hold, every independently valid transparent policy tuple exactly realises W, and its admission map generates the exact marked-nonblocking wrapper.

Proof. For necessity, let g be an independently valid tuple and fix $s \in D$. By Definition 4.3(i), some governor i emits a non-abstaining command u at s. Observation uniformity makes i issue u throughout $C_i(O_i(s))$. Conditions (ii) and (iii) of the definition put u in every $\mathrm{Auth}_i(t) \cap \mathcal{U}^{=}(t)$ over that cell. Hence $u \in \mathcal{U}_i^{AE}(O_i(s))$.

For sufficiency, choose one $u_{i,y} \in \mathcal{U}_i^{AE}(y)$ whenever this support is nonempty. Let governor i issue $u_{i,y}$ on y and abstain on other decision observations; this may create redundant exact issuances on overlapping cells, which are harmless under (S2). Extend policies arbitrarily elsewhere. The cell intersection proves authorisation and exactness for every emitted profile, while (AEC) gives at least one non-abstaining command at every decision state. Thus the tuple is independently valid. Finally, for any independently valid tuple, (S2) returns the exact effect of its selected non-abstaining profile, so $\mu_g(s) = \mathrm{Viab}_c(s)$ on D. Theorem 4.1 and Corollary 3.5 complete the claim. ∎

The three realisability notions used in the paper satisfy

$$\text{independently valid transparent} \implies \text{route-bearing transparent exact} \implies \text{general fused exact.}$$

The first implication can be strict under state-indexed selection, as Proposition 5.2 shows. The second need not reverse because general fusion may produce a result not attributable to a selected non-abstaining route, including on an all-abstaining command vector. The aligned-witness condition (AEC) characterises the strongest, independently valid class.

Here *full observation* means singleton decision-state cells; *universal authority relative to the deployed interface* means $\mathcal{U} \cup \{\bot\} \subseteq \mathrm{Auth}_i(s)$ on D; an *enlarged interface* replaces $\mathcal{U}$ by $\mathcal{U}'$ satisfying $\mathcal{U} \subseteq \mathcal{U}' \subseteq \overline{\mathcal{U}}$, while leaving the catalogue transducers, their effects, observations, and authority maps unchanged; and a state is *hidden from governor i* when it shares i's observation with another decision state requiring a different exact profile.

5 Noncomposition under Transparent Selection

Under transparent single-witness governance, separate adequacy of the deployed intervention interface and the observation/authority architecture does not imply adequacy of their actual combination.

Theorem 5.1 (Noncomposition under transparent selection)**.** *There exists a five-state marked runtime with an ambient profile catalogue $\overline{\mathcal{U}}$, an actual deployed interface $\mathcal{U} \subsetneq \overline{\mathcal{U}}$, and an actual observation/authority architecture such that:*

(i) under counterfactual full observation and universal authority relative to $\mathcal{U}$, the actual deployed interface admits a route-bearing transparent exact realisation;

(ii) after enlarging only the deployed interface to one $\mathcal{U}'$ with $\mathcal{U} \subsetneq \mathcal{U}' \subseteq \overline{\mathcal{U}}$, the actual observation and authority maps admit a route-bearing transparent exact realisation; but

(iii) the actual deployed interface $\mathcal{U}$ combined with the actual observation and authority maps has no route-bearing transparent exact realisation under any selector satisfying (S1)–(S2), and therefore has no independently valid transparent realisation.

Proof. Let

$$S = \{r, p, q, m, d\}, \quad s_0 = r, \quad Q = \{r, p, q, m\}, \quad S_m = \{m\},$$
$$\Sigma_u = \{e_p, e_q\}, \qquad \Sigma_c = \{a, b\}.$$

The defined transitions are

$$\delta(r, e_p) = p, \quad \delta(r, e_q) = q,$$
$$\delta(p, a) = m, \quad \delta(p, b) = d, \qquad \delta(q, b) = m, \quad \delta(q, a) = d.$$

Every unlisted pair is undefined. Hence

$$W = Q, \qquad D = \{p, q\}, \qquad \mathrm{Viab}_c(p) = \{a\}, \quad \mathrm{Viab}_c(q) = \{b\}.$$

Use public labels

$$\Gamma = \{\gamma_r, \gamma_p, \gamma_q, \gamma_m, \gamma_d\}, \qquad \gamma(x) = \gamma_x \qquad (x \in S).$$

Let the ambient catalogue and actual deployed interface be

$$\overline{\mathcal{U}} = \{u_a, u_b, u_c\}, \qquad \mathcal{U} = \{u_a, u_b\}.$$

All three catalogue profiles are fixed before the target is computed. Profiles u_a and u_b each have singleton transducer state set $\{\rho\}$, initial state ρ, updates

$$\eta_{u_a}(\rho, \gamma_x) = \eta_{u_b}(\rho, \gamma_x) = \rho \qquad (x \in S),$$

and outputs

$$\lambda_{u_a}(\rho, \gamma_x) = \{a\}, \qquad \lambda_{u_b}(\rho, \gamma_x) = \{b\}$$

for every label. Profile u_c has singleton state set $\{\rho_c\}$, initial state ρ_c, update

$$\eta_{u_c}(\rho_c, \gamma_x) = \rho_c \qquad (x \in S),$$

and output

$$\lambda_{u_c}(\rho_c, \gamma_p) = \{a\}, \qquad \lambda_{u_c}(\rho_c, \gamma_q) = \{b\},$$

with output $\varnothing$ on the remaining labels. Therefore

$$E_{u_a}(p) = E_{u_a}(q) = \{a\}, \qquad E_{u_b}(p) = E_{u_b}(q) = \{b\},$$

and

$$E_{u_c}(p) = \{a\}, \qquad E_{u_c}(q) = \{b\}.$$

The recurrence (T0)–(TU) is immediate for every catalogue profile.

For (i), consider a counterfactual architecture with one governor that distinguishes p from q and has universal authority relative to the actual deployment $\mathcal{U}$. It issues u_a at p and u_b at q. A fixed selector returns that governor's non-abstaining effect, so the actual deployed interface is exact.

For the actual architecture, let $I = \{R, H\}$. Governor R sees

$$O_R(p) = y_p, \qquad O_R(q) = y_q,$$

while the states are hidden from governor H:

$$O_H(p) = O_H(q) = y_H.$$

Choose arbitrary observation values at r and m, where commands are not requested. On the decision-state domain, set

$$\mathrm{Auth}_R(p) = \mathrm{Auth}_R(q) = \{\bot\},$$

and, from the outset,

$$\mathrm{Auth}_H(p) = \mathrm{Auth}_H(q) = \{u_a, u_b, u_c, \bot\}.$$

Extend both authority maps arbitrarily to r and m, thereby obtaining total maps on W. Thus H is already authorised for u_c, although $u_c \notin \mathcal{U}$ and is not issuable in the actual deployment. Let any selector family satisfying (S1)–(S2) be supplied. Whenever H is non-abstaining and R abstains, the selector must choose H.

For (ii), enlarge only the deployed interface to

$$\mathcal{U}' = \mathcal{U} \cup \{u_c\} = \overline{\mathcal{U}}.$$

Do not change either observation map or either authority map. Although H's own observation remains coarse, the predeclared transducer u_c reads the authenticated public state label. Governor H can now issue u_c uniformly on y_H, and its effect is exact at both p and q.

For (iii), return to the actual deployed interface $\mathcal{U} = \{u_a, u_b\}$. Governor R is authorised only for $\bot$, so in any route-bearing transparent exact tuple the required non-abstaining command must be issued by H. Since H has the single decision-state cell $\{p, q\}$, observation uniformity forces it to issue the same deployed profile at both states. Relative to the actual deployment,

$$\mathcal{U}^{=}(p) = \{u_a\}, \qquad \mathcal{U}^{=}(q) = \{u_b\},$$

and neither deployed profile is exact at both states. The already authorised profile u_c cannot be issued because it is not deployed. Therefore no route-bearing transparent exact realisation exists, independently of the selector; in particular, (AEC) fails and no independently valid realisation exists. ∎

The counterfactuals change different layers. Full observation and universal authority repair the issuer side while leaving the actual deployment fixed. The second counterfactual changes only deployment, activating a profile whose transducer and authority were already fixed while leaving H's coarse observation unchanged. Their separate success does not imply composition under route-bearing transparent selection. The theorem intentionally makes no claim against unrestricted state-indexed general fusion: without route-bearing semantics, a supplied operator may return state-specific target sets, including on an all-abstaining vector, without attributing the result to any authorised deployed profile.

Proposition 5.2 (Independent validity is strictly stronger). *Every independently valid transparent policy tuple is route-bearing transparent exact. Under state-indexed transparent selection satisfying* (S1)–(S2)*, the converse need not hold.*

Proof. Use the five-state runtime and deployed profiles u_a, u_b from Theorem 5.1. Replace its actual issuer architecture by two governors $1, 2$, each with the single decision-state observation cell $\{p, q\}$ and universal authority over $\{u_a, u_b, \bot\}$. Let

$$g_1 \equiv u_a, \qquad g_2 \equiv u_b,$$

and predeclare the state-indexed selector on this command vector by

$$\mathrm{Sel}_p(u_a, u_b) = 1, \qquad \mathrm{Sel}_q(u_a, u_b) = 2.$$

Extend the selector family arbitrarily to every other non-abstaining command vector subject to (S1). The complete selector family is fixed with the model data before the target region is computed; only the displayed values are used below. Then the selected effects are $\{a\} = \mathrm{Viab}_c(p)$ and $\{b\} = \mathrm{Viab}_c(q)$, so the tuple is route-bearing transparent exact. However, each governor emits a profile that is nonexact at one state of its complete observation cell. The tuple is therefore not independently valid, and in fact both aligned supports on that cell are empty. ∎

Independent validity deliberately excludes realisations whose correctness depends on a hidden state-indexed selector choosing among locally nonexact commands. Its stronger per-route semantics is the one used by the aligned-witness and proof-carrying developments below.

6 Monotone Architectural Repairs

The noncomposition construction identifies three different design levers. Their effects are monotone, but they are not interchangeable.

Definition 6.1 (Observation refinement). Fix a governor $i \in I$, and let Y_i' be a finite observation-value set. An observation map $O_i' : W \to Y_i'$ *refines* $O_i : W \to Y_i$ if, for all $s, t \in W$,

$$O_i'(s) = O_i'(t) \quad \Longrightarrow \quad O_i(s) = O_i(t).$$

Thus every refined information cell is contained in an original cell.

For any governor $i \in I$, finite set Y, and observation map $O : W \to Y$ assigned to that governor, write

$$\mathcal{U}_{i,O}^{AE}(O(s)) = \bigcap_{t \in D:\, O(t) = O(s)} (\mathrm{Auth}_i(t) \cap \mathcal{U}^{=}(t)) \qquad (s \in D)$$

for the aligned support computed with O while all other model components remain fixed.

Theorem 6.2 (Observation-refinement monotonicity). *If O_i' refines O_i, while authority and transducer effects are unchanged, then for every decision state $s \in D$,*

$$\mathcal{U}_{i,O_i}^{AE}(O_i(s)) \subseteq \mathcal{U}_{i,O_i'}^{AE}(O_i'(s)).$$

Consequently, refining a governor's authenticated observation cannot destroy an aligned exact witness already available to that governor.

Proof. The refined cell containing s is a subset of the original cell containing s. The aligned support is an intersection of authorised exact profile sets over the relevant cell. Intersecting the same family over fewer states can only enlarge the result. ∎

Theorem 6.3 (Authority and deployment monotonicity). *Fix the ambient catalogue $\overline{\mathcal{U}}$ and all catalogue-profile effects. Let*

$$\mathcal{U} \subseteq \mathcal{U}' \subseteq \overline{\mathcal{U}}$$

be two deployed interfaces. For each $i \in I$, let

$$\mathrm{Auth}_i, \mathrm{Auth}_i' : W \to 2^{\overline{\mathcal{U}} \cup \{\bot\}}$$

satisfy $\mathrm{Auth}_i(s) \subseteq \mathrm{Auth}'_i(s)$ *for every* $s \in W$. *Any independently valid transparent realisation under deployment* $\mathcal{U}$ *and authority family* $\{\mathrm{Auth}_i\}_{i\in I}$ *remains an independently valid transparent realisation under deployment* $\mathcal{U}'$ *and authority family* $\{\mathrm{Auth}'_i\}_{i\in I}$.

Proof. Reuse the original policy tuple. Every issued profile belongs to $\mathcal{U} \subseteq \mathcal{U}'$, every old authorised command remains authorised, and every old exact effect is unchanged. The tuple therefore remains independently valid, so Corollary 4.4 preserves exact realisation. ∎

The theorem is one-directional. More authority or a larger deployed interface cannot invalidate an existing transparent controller, but neither necessarily repairs an alignment failure. Authority expansion helps only when an exact deployed profile is available throughout the relevant cell; deployment expansion helps only when some appropriately informed governor is already authorised to issue the newly deployed profile.

Corollary 6.4 (Authenticated communication repair). *In the construction of Theorem 5.1, an authenticated message that distinguishes* p *from* q *and is incorporated into governor* H*'s causal observation restores independently valid transparent exact realisability with the original library* $\{u_a, u_b\}$.

Proof. The message refines H's cell $\{p, q\}$ into singleton cells. Governor H can then issue u_a at p and u_b at q, both within its existing authority. Theorem 6.2 formalises the support enlargement. ∎

A recommendation from a risk agent is therefore not sufficient merely because it is semantically informative. It must be authenticated, bound to the proposal, delivered before commitment, incorporated into the authorised issuer's decision state, and remain valid through commit.

7 Proof-Carrying Aligned Witnesses

7.1 Bridge from requests to state-level witnesses

The certificate layer does not introduce a second notion of information. It pulls the state-level aligned-support condition back through a versioned request–context map:

$$\begin{aligned}\text{request + context} &\longrightarrow \text{augmented decision state} \longrightarrow \text{governor observation cell}\\ &\longrightarrow \text{authority/effect test} \longrightarrow \text{certificate predicate.}\end{aligned}$$

Surjectivity of each request–context map ensures that every decision state of its version is represented. Global request–context fibres ensure that two requests producing the same governor observation are tested together rather than certified independently.

7.2 Complete versioned runtime model

Let $\mathcal{V}$ be a finite set of materialised model versions. Fix a nonempty finite governor-identifier set I, a nonempty finite ambient profile-identifier catalogue $\overline{\mathcal{U}}$, a nonempty deployed identifier set $\mathcal{U} \subseteq \overline{\mathcal{U}}$, and an integer $n \geq 1$. Let

$$N = \{1, \dots, n\}, \qquad \mathcal{X} = \prod_{j\in N} X_j,$$

where each coordinate domain X_j is finite and nonempty. The identifier sets I, $\overline{\mathcal{U}}$, and $\mathcal{U}$, together with the coordinate space $\mathcal{X}$, are shared across versions. This is a fixed-deployment-identifier assumption: versioned transducer semantics, observations, authority, fusion, requests, and outcomes may change, but changing the deployed identifier set requires a separately declared migration rather than silent reinterpretation.

For every $v \in \mathcal{V}$, define

$$\mathcal{A}_v = (S_v, s_v^0, \Sigma_v, \delta_v, S_{m,v}), \qquad \Sigma_v = \Sigma_{c,v} \dot\cup \Sigma_{u,v},$$

together with a safe envelope $Q_v^{\mathrm{safe}} \subseteq S_v$ satisfying $S_{m,v} \subseteq Q_v^{\mathrm{safe}}$. Let Av^v, Av_c^v, and Av_u^v denote the versioned availability maps defined as in Section 3. Applying that section gives the greatest admissible region W_v, decision-state set D_v, and exact controlled successor map Viab_c^v. Define the active-version set

$$\mathcal{V}_{\mathrm{act}} = \{v \in \mathcal{V} : s_v^0 \in W_v\}.$$

A version outside $\mathcal{V}_{\mathrm{act}}$ fails the initial-viability check and does not enter the proof-carrying layer.

For each $v \in \mathcal{V}_{\mathrm{act}}$, let $\mathcal{R}_v$ be a finite set of typed checkpoint requests, chosen nonempty when $D_v \neq \emptyset$ and empty when $D_v = \emptyset$. Version tags make the request sets pairwise disjoint. Define the version-tagged request–context domain

$$\Omega = \bigcup_{v\in\mathcal{V}_{\mathrm{act}}} (\mathcal{R}_v \times \mathcal{X}).$$

For every $v \in \mathcal{V}_{\mathrm{act}}$, the model additionally contains:

- a finite public-label set Γ_v and a map $\gamma_v : S_v \to \Gamma_v$;
- for every $u \in \overline{\mathcal{U}}$, a finite transducer
$$\mathcal{I}_u^v = (R_u^v, r_{u,v}^0, \eta_u^v, \lambda_u^v),$$
where R_u^v is nonempty, $r_{u,v}^0 \in R_u^v$, $\eta_u^v : R_u^v \times \Gamma_v \to R_u^v$, and $\lambda_u^v : R_u^v \times \Gamma_v \to 2^{\Sigma_{c,v}}$;
- a state projection $r_u^v : S_v \to R_u^v$ for each profile, satisfying the versioned forms of (T0)–(TU), with derived effect
$$E_u^v(s) = \lambda_u^v(r_u^v(s), \gamma_v(s)) \cap \mathrm{Av}_c^v(s);$$
- for every $i \in I$, a finite observation set Y_i^v, a causal observation map $O_i^v : W_v \to Y_i^v$, and an authority map $\mathrm{Auth}_i^v : W_v \to 2^{\overline{\mathcal{U}} \cup \{\bot\}}$ satisfying the versioned abstention axiom on D_v;
- a versioned fusion family $\Psi^v = (\Psi_s^v)_{s \in S_v}$ of the type in Section 4, carrying its target-independence provenance declaration and inducing Φ_s^v on W_v. A general-fusion model may additionally record whether (EA) holds. For transparent governance, the model supplies selectors Sel_s^v satisfying the versioned forms of (S1)–(S2); and
- a total surjective request–context map
$$\zeta_v : \mathcal{R}_v \times \mathcal{X} \longrightarrow D_v. \tag{RC}$$

When $D_v = \varnothing$, both the domain and codomain of (RC) are empty, so ζ_v is the unique empty map and the proof-carrying section is vacuous for that version. Let L_{W_v} denote the versioned form of the language L_W defined in Section 3. Thus W_v, D_v, E_u^v, Viab_c^v, and L_{W_v} are derived from one complete versioned model rather than supplied as unrelated state tables.

Every explicit finite-verification claim below assumes that requests, contexts, observations, authority, transducer states, effects, outcomes, and obligation predicates are either explicitly materialised or represented by total deterministic decidable procedures over finite domains. An unbounded request language requires a separate decidability theorem and is not automatically covered by the finite claims.

For $v \in \mathcal{V}_{\mathrm{act}}$, $i \in I$, and $(q, x) \in \mathcal{R}_v \times \mathcal{X}$, define the global request–context fibre
$$F_i^v(q,x) = \left\{(q', y) \in \mathcal{R}_v \times \mathcal{X} : O_i^v(\zeta_v(q', y)) = O_i^v(\zeta_v(q, x))\right\}. \tag{GF}$$

This is the pullback of the complete state-level information cell. If every issuer observes an authenticated request identifier, one may instead assume that equality of observations implies $q' = q$; the request-local form is then a derived simplification rather than an unstated premise.

For $x \in \mathcal{X}$ and $C \subseteq N$, write $x \restriction_C = (x_j)_{j \in C}$.

Definition 7.1 (Global-cell aligned-witness predicate). For $v \in \mathcal{V}_{\mathrm{act}}$, $i \in I$, and $u \in \mathcal{U}$, define
$$\mathrm{AE}_{i,u}^v : \mathcal{R}_v \times \mathcal{X} \to \{0, 1\}$$
by $\mathrm{AE}_{i,u}^v(q,x) = 1$ exactly when, for every $(q', y) \in F_i^v(q,x)$,
$$u \in \mathrm{Auth}_i^v(\zeta_v(q', y)) \quad \text{and} \quad E_u^v(\zeta_v(q', y)) = \mathrm{Viab}_c^v(\zeta_v(q', y)). \tag{AW}$$

Call (v, i, u, q, x) a *positive tuple* when $v \in \mathcal{V}_{\mathrm{act}}$, $i \in I$, $u \in \mathcal{U}$, $(q,x) \in \mathcal{R}_v \times \mathcal{X}$, and $\mathrm{AE}_{i,u}^v(q,x) = 1$. A positive tuple therefore certifies one governor–profile pair over the complete state-level information cell, including states generated by other requests.

For $v \in \mathcal{V}_{\mathrm{act}}$, $i \in I$, $u \in \mathcal{U}$, and $q \in \mathcal{R}_v$, define the bad-context set
$$\mathrm{Bad}_{i,u}^v(q) = \{y \in \mathcal{X} : \mathrm{AE}_{i,u}^v(q, y) = 0\}.$$

For $x \in \mathcal{X}$ and $C \subseteq N$, define the coordinate cylinder
$$[x]_C = \{y \in \mathcal{X} : y \restriction_C = x \restriction_C\}.$$

Definition 7.2 (Aligned-witness certificate). Fix a positive tuple (v, i, u, q, x). A set $C \subseteq N$ is an *aligned-witness certificate* when
$$[x]_C \cap \mathrm{Bad}_{i,u}^v(q) = \varnothing. \tag{CERT}$$
The certificate fixes attributes of the current request, while each predicate evaluation certifies the named route over its complete global information fibre.

7.3 Formal-XAI duality instantiated for governance

Fix a positive tuple (v, i, u, q, x) as in Definition 7.2. For each $y \in \mathrm{Bad}^{v}_{i,u}(q)$, define

$$\mathrm{Diff}(x, y) = \{j \in N : x_j \neq y_j\}$$

and define the failure hypergraph

$$\mathcal{H}^{AE}_{i,u}(q, x, v) = \Big(N, \{\mathrm{Diff}(x, y) : y \in \mathrm{Bad}^{v}_{i,u}(q)\}\Big).$$

Theorem 7.3 (Aligned-witness certificate duality). *For every positive tuple (v, i, u, q, x), a set $C \subseteq N$ is an aligned-witness certificate if and only if it intersects every edge of $\mathcal{H}^{AE}_{i,u}(q, x, v)$.*

Proof. If C misses $\mathrm{Diff}(x, y)$ for a bad context y, then x and y agree on every coordinate in C, so $y \in [x]_C$, contradicting (CERT). Conversely, a bad $y \in [x]_C$ yields a failure edge disjoint from C. ∎

The result is the established abductive/contrastive explanation duality [21, 9, 14] applied to the governance-specific predicate (AW). Positive weights $w_j > 0$ for $j \in N$ may represent latency, privacy, provenance, or volatility, and a minimum-cost certificate minimises $\sum_{j \in C} w_j$.

For the next complexity statement, a *succinct circuit-represented request/context model* encodes runtime states by Boolean vectors and represents transitions, observations, request–context mapping, transducer outputs, exact effects, and authority by polynomial-size circuits. Its instance size is the total circuit-description length plus the bit dimensions and finite alphabets, rather than the cardinality of the represented semantic state space.

Corollary 7.4 (Certificate hardness). *Minimum-cardinality and minimum-cost aligned-witness certificate synthesis are NP-hard, even for Boolean contexts and a monotone-CNF authority predicate in a succinct circuit-represented request/context model.*

Proof. Take a Hitting Set instance with universe $N = \{1, \ldots, n\}$, nonempty edges $E_1, \ldots, E_m$, and define the polynomial-size monotone CNF

$$F(z) = \bigwedge_{k=1}^{m} \bigvee_{j \in E_k} z_j \qquad (z \in \{0,1\}^n).$$

Use one active version and one governor, set $I = \{i_*\}$, $\mathcal{X} = \{0,1\}^n$, and $\mathcal{R} = \{q_*\}$, and suppress the fixed version and governor decorations in this reduction. Semantically, the runtime has a decision state d_z for each $z \in \{0,1\}^n$, but these states are represented by n state bits rather than materialised. A polynomial-size transition circuit reaches the encoded state d_z through n exogenous binary-choice stages, and the sole controlled event a then leads to one marked state. Hence $\mathrm{Viab}_c(d_z) = \{a\}$. The request–context map $\zeta(q_*, z) = d_z$ and the injective observation map $O(d_z) = z$ are wiring circuits.

Use ambient catalogue and deployed interface $\overline{\mathcal{U}} = \mathcal{U} = \{u\}$ and one one-state transducer u with identity update and total constant output $\{a\}$, so it is exact at every decision state. The sole governor's authority circuit is

$$\mathrm{Auth}(d_z) = \begin{cases} \{u, \bot\}, & F(z) = 1, \\ \{\bot\}, & F(z) = 0. \end{cases}$$

Thus the aligned-witness predicate satisfies $\mathrm{AE}(q_*, z) = F(z)$. All state, transition, observation, effect, request-map, and authority circuits have total size polynomial in the Hitting Set input.

The all-ones context is positive. Fixing coordinates C to one preserves positivity under every completion exactly when C intersects every clause edge E_k. Therefore minimum certificates solve Hitting Set, equivalently Set Cover under the standard reduction [10]; weights are preserved. ∎

The corollary establishes NP-hardness only. The counterexample query used below is treated as an exact-verifier oracle; in a succinct circuit representation that query may itself be computationally expensive. The procedure is therefore an exact synthesis scheme rather than a polynomial-time algorithm.

7.4 Counterexample-guided certificate synthesis

The exact verifier in this subsection answers precisely the counterexample query whose succinct representation may be expensive; the intended regimes are explicitly materialised finite models or representations for which that query is tractable in practice.

Fix a positive tuple (v, i, u, q, x) and positive coordinate weights $w = (w_j)_{j \in N}$. Initialise the working edge family by

$$\widehat{\mathcal{E}}_0 = \varnothing.$$

At iteration t, solve the weighted hitting-set problem over $\widehat{\mathcal{E}}_t$ globally optimally to obtain a candidate C_t, and ask an exact verifier whether

$$\exists y \in \mathcal{X} : y \restriction_{C_t} = x \restriction_{C_t} \quad \wedge \quad \mathrm{AE}^v_{i,u}(q, y) = 0. \tag{V}$$

A witness y_t updates

$$\widehat{\mathcal{E}}_{t+1} = \widehat{\mathcal{E}}_t \cup \{\mathrm{Diff}(x, y_t)\};$$

unsatisfiability proves that C_t satisfies (CERT).

Theorem 7.5 (Counterexample-guided exact synthesis)**.** *Under the finite and exact-representation assumptions above, the procedure terminates and returns a minimum-cost aligned-witness certificate.*

Proof. Every returned witness is a genuine missed failure edge and excludes the current candidate. Only finitely many disagreement supports exist. At termination, unsatisfiability of (V) is exactly (CERT). Every generated edge must be hit by every exact certificate, so an optimal terminating candidate is globally optimal. ∎

7.5 Compositional evidence

For fixed $v \in \mathcal{V}_{\mathrm{act}}$, $i \in I$, and $u \in \mathcal{U}$, define Boolean predicates

$$\mathrm{AuthOK}^v_{i,u}, \mathrm{ExactOK}^v_{i,u} : \mathcal{R}_v \times \mathcal{X} \longrightarrow \{0, 1\}$$

by

$$\mathrm{AuthOK}^v_{i,u}(q, x) = 1 \iff \forall (q', y) \in F^v_i(q, x) : u \in \mathrm{Auth}^v_i(\zeta_v(q', y)),$$
$$\mathrm{ExactOK}^v_{i,u}(q, x) = 1 \iff \forall (q', y) \in F^v_i(q, x) : E^v_u(\zeta_v(q', y)) = \mathrm{Viab}^v_c(\zeta_v(q', y)).$$

Then, pointwise on $\mathcal{R}_v \times \mathcal{X}$,

$$\mathrm{AE}^v_{i,u} = \mathrm{AuthOK}^v_{i,u} \wedge \mathrm{ExactOK}^v_{i,u}.$$

Assume $\mathrm{AE}^v_{i,u}(q, x) = 1$. An *authority certificate* is a set $C_A \subseteq N$ such that

$$\forall y \in [x]_{C_A} : \mathrm{AuthOK}^v_{i,u}(q, y) = 1,$$

and an *exactness certificate* is a set $C_E \subseteq N$ such that

$$\forall y \in [x]_{C_E} : \mathrm{ExactOK}^v_{i,u}(q, y) = 1.$$

Proposition 7.6 (Positive certificate composition)**.** *If C_A is an authority certificate and C_E is an exactness certificate for (v, i, u, q, x), then $C_A \cup C_E$ is an aligned-witness certificate.*

Proof. Every completion agreeing with x on $C_A \cup C_E$ agrees on both component coordinate sets. The authority and exactness predicates are therefore both true, so (AW) is true throughout the resulting cylinder. ∎

The union need not be minimum because joint synthesis can exploit shared attributes or logical implications.
For either component predicate $P \in \{\mathrm{AuthOK}^v_{i,u}, \mathrm{ExactOK}^v_{i,u}\}$, call $C^- \subseteq N$ a *negative cylinder certificate at* (q, x) when

$$\forall y \in [x]_{C^-} : \quad P(q, y) = 0.$$

Remark 7.7 (Stable denying reason)**.** A negative cylinder certificate for either mandatory conjunct proves that the governor–profile route fails (AW) throughout the same cylinder. This is a denial certificate, not a positive aligned-witness certificate.

7.6 Current-bound certified governance for one active version

The proof-carrying pattern follows the general producer–verifier separation of proof-carrying code and proof-carrying authorisation, together with recent certificate-bearing agent-action and proof-derived-authorisation architectures [17, 2, 34, 7, 32, 33]. The semantic obligation here is specific: the proof establishes one authorised exact route over a complete governor information fibre, and acceptance separately establishes that the proof applies to the current execution. The certified layer uses the per-route invariant underlying independent validity: every accepted non-abstaining command independently proves authority and exact effect over the issuer's complete information fibre. The accepted-command map is snapshot-bound rather than assumed a priori to be a static observation-only policy.

For every $v \in \mathcal{V}_{\mathrm{act}}$ and $s \in D_v$, let $\mathsf{Prop}_v(s)$ be a nonempty finite set of execution-relevant controlled proposal records enabled at s. A proposal record contains the proposed controlled event and every execution field needed for current binding.

Any field that changes transition or profile semantics must already be represented in the runtime state or event alphabet, and every event in $\mathrm{Av}_c^v(s)$ occurs in at least one record in $\mathsf{Prop}_v(s)$. For each $v \in \mathcal{V}_{\mathrm{act}}$, let H_v be a finite identifier set and let

$$\mathrm{pid}_v : \bigsqcup_{s \in D_v} \mathsf{Prop}_v(s) \longrightarrow H_v$$

be an authenticated injective proposal-identifier map, where $\bigsqcup$ denotes disjoint union. Thus $h = \mathrm{pid}_v(p)$ uniquely binds the complete proposal record p, including its controlled event and payload.

For each $v \in \mathcal{V}_{\mathrm{act}}$, define the authenticated snapshot set

$$\mathsf{Snap}_v = \left\{(\mathrm{pid}_v(p), q, v, x, s) : (q,x) \in \mathcal{R}_v \times \mathcal{X},\ s = \zeta_v(q,x),\ p \in \mathsf{Prop}_v(s)\right\},$$

and let $\mathsf{Snap} = \bigsqcup_{v \in \mathcal{V}_{\mathrm{act}}} \mathsf{Snap}_v$. A proof-carrying command that claims version $v \in \mathcal{V}_{\mathrm{act}}$ is

$$\Pi = (h, q, v, i, u, C, \xi, \varrho), \qquad h \in H_v,\ q \in \mathcal{R}_v,\ i \in I,\ u \in \overline{\mathcal{U}}, \qquad \xi \in \prod_{j \in C} X_j,$$

where $C \subseteq N$ and ϱ is a proof object binding the proposal identifier, request, version, issuer, profile, certificate coordinates and values, and verifier result.

At an acceptance point, let

$$\chi_{\mathrm{cur}} = (h_{\mathrm{cur}}, q_{\mathrm{cur}}, v_{\mathrm{cur}}, x_{\mathrm{cur}}, s_{\mathrm{cur}}) \in \mathsf{Snap}_{v_{\mathrm{cur}}}$$

be the authenticated current-execution snapshot. Thus $v_{\mathrm{cur}} \in \mathcal{V}_{\mathrm{act}}$, $q_{\mathrm{cur}} \in \mathcal{R}_{v_{\mathrm{cur}}}$, $x_{\mathrm{cur}} \in \mathcal{X}$, $s_{\mathrm{cur}} \in D_{v_{\mathrm{cur}}}$, and $h_{\mathrm{cur}} = \mathrm{pid}_{v_{\mathrm{cur}}}(p_{\mathrm{cur}}) \in H_{v_{\mathrm{cur}}}$ for one authenticated current proposal record $p_{\mathrm{cur}} \in \mathsf{Prop}_{v_{\mathrm{cur}}}(s_{\mathrm{cur}})$.

Definition 7.8 (Current-bound proof acceptance)**.** A verifier accepts Π at χ_{cur} only when:

(i) $h = h_{\mathrm{cur}}$, $q = q_{\mathrm{cur}}$, and $v = v_{\mathrm{cur}}$, with v the active version;

(ii) $i \in I$ and $u \in \mathcal{U}$, so the named route uses a currently deployed profile;

(iii)
$$\xi = x_{\mathrm{cur}} \restriction_C, \qquad s_{\mathrm{cur}} = \zeta_v(q_{\mathrm{cur}}, x_{\mathrm{cur}}) \in D_v; \tag{CB}$$

(iv) ϱ is authentic, binds every field of Π, and establishes the universal certificate claim

$$\forall y \in \mathcal{X} : y \restriction_C = \xi \Longrightarrow \mathrm{AE}_{i,u}^v(q,y) = 1. \tag{UC}$$

Condition (UC) is the cylinder form of an aligned-witness certificate and is equivalent to Theorem 7.3 for any positive context having restriction ξ. Authenticity of an old proof alone is insufficient: (CB) must be checked against the current authenticated proposal, request, context, version, and runtime state.

Assume that each governor submits at most one non-abstaining proof-carrying command for any snapshot $\chi \in \mathsf{Snap}$. For each $i \in I$, define the accepted-command map

$$c_i^{\mathrm{acc}} : \mathsf{Snap} \longrightarrow \mathcal{U} \cup \{\bot\}, \qquad c_i^{\mathrm{acc}}(\chi) = \begin{cases} u, & \text{if an accepted command from governor } i \text{ names } u, \\ \bot, & \text{otherwise,} \end{cases}$$

and the accepted command vector

$$\mathbf{c}^{\mathrm{acc}}(\chi) = (c_i^{\mathrm{acc}}(\chi))_{i \in I} \in (\mathcal{U} \cup \{\bot\})^I.$$

Whenever $\chi = (h, q, v, x, s) \in \mathsf{Snap}_v$ has a non-abstaining accepted command vector, let

$$j^*(s, \chi) = \mathrm{Sel}_s^v(\mathbf{c}^{\mathrm{acc}}(\chi))$$

and define the accepted admission set

$$\mu_{\mathrm{acc}}^v(s, \chi) = E^v_{c^{\mathrm{acc}}_{j^*(s,\chi)}(\chi)}(s).$$

Assumptions (iii)–(iv) in the next theorem form a certified coverage hypothesis: every represented proposal snapshot has at least one accepted non-abstaining route, and every such route carries a universal aligned-witness proof. Together with surjectivity of the request–context map, these assumptions cover every decision state in D_v by an authorised exact route. The theorem proves that independently justified, currently bound routes compose under transparent selection to the exact admission map, with stale premises rejected or revalidated before commit.

Theorem 7.9 (Soundness of current-bound certified transparent governance). *Fix $v \in \mathcal{V}_{\mathrm{act}}$. Assume:*

(i) the versioned runtime has the faithful marked wrapper of Section 3;

(ii) transparent fusion obeys the versioned forms of (S1)–(S2), and its transducers, selector, and fusion artefact satisfy the declared provenance assumptions;

(iii) every accepted non-abstaining command passes current-bound proof acceptance in Definition 7.8;

(iv) for every $(q,x) \in \mathcal{R}_v \times \mathcal{X}$, every proposal record $p \in \mathsf{Prop}_v(s)$ at $s = \zeta_v(q,x)$, and $h = \mathrm{pid}_v(p)$, the accepted command vector at $\chi = (h,q,v,x,s)$ has at least one non-abstaining component; and

(v) before commit, the runtime either preserves the accepted snapshot atomically or revalidates the proposal identifier, active version, context restriction, request–context state relation, deployment status, and every expiring premise represented in the model.

Then, for every snapshot $\chi = (h,q,v,x,s)$ covered by assumption (iv),

$$\mu^v_{\mathrm{acc}}(s,\chi) = \mathrm{Viab}^v_c(s).$$

Consequently these decisions induce the state-level admission map

$$\mu^v : W_v \longrightarrow 2^{\Sigma_{c,v}}, \qquad \mu^v(s) = \mathrm{Viab}^v_c(s) \qquad (s \in W_v),$$

the projected language is exactly L_{W_v}, and the versioned wrapper is marked nonblocking.

Proof. Fix a covered snapshot

$$\chi = (h_{\mathrm{cur}}, q_{\mathrm{cur}}, v, x_{\mathrm{cur}}, s_{\mathrm{cur}}).$$

Assumption (iv) makes $\mathbf{c}^{\mathrm{acc}}(\chi)$ non-abstaining. Consider any accepted component $c_i^{\mathrm{acc}}(\chi) = u \neq \bot$, witnessed by an accepted command $\Pi = (h,q,v,i,u,C,\xi,\varrho)$. Current binding gives $q = q_{\mathrm{cur}}$, $s_{\mathrm{cur}} = \zeta_v(q_{\mathrm{cur}}, x_{\mathrm{cur}})$, and $x_{\mathrm{cur}} \restriction_C = \xi$. Applying (UC) to $y = x_{\mathrm{cur}}$ yields

$$\mathrm{AE}^v_{i,u}(q_{\mathrm{cur}}, x_{\mathrm{cur}}) = 1.$$

By (AW), u is authorised and exact at the actual state s_{cur}, and the same route remains authorised and exact throughout the complete governor information fibre. Hence every non-abstaining component of $\mathbf{c}^{\mathrm{acc}}(\chi)$ has effect $\mathrm{Viab}^v_c(s_{\mathrm{cur}})$. The versioned form of (S1) selects one such component, and (S2) gives

$$\mu^v_{\mathrm{acc}}(s_{\mathrm{cur}}, \chi) = E^v_{c^{\mathrm{acc}}_{j^*(s_{\mathrm{cur}},\chi)}(\chi)}(s_{\mathrm{cur}}) = \mathrm{Viab}^v_c(s_{\mathrm{cur}}).$$

Because ζ_v is surjective, assumption (iv) covers every proposal record at every represented decision state, and assumption (iii) makes each accepted non-abstaining component satisfy (UC), the accepted decisions define the same exact admission set at every $s \in D_v$. At each $s \in W_v \setminus D_v$, one has $\mathrm{Av}^v_c(s) = \varnothing$, so both the induced admission set and $\mathrm{Viab}^v_c(s)$ are empty. Commit-time preservation or revalidation ensures that the proved premises still describe the committed macro-step. Exactness therefore holds on all of W_v, and the versioned form of Corollary 3.5 yields projection to L_{W_v} and coaccessibility of every reachable wrapper state. ∎

A verifier may equivalently derive and store one canonical route per global observation fibre, then feed that fixed policy to transparent selection, provided each use still passes current-execution binding. The theorem is a conditional soundness result: it assumes certificate coverage and proves that accepted certified commands compose to exact governance. It is relative to the encoded target, trusted instrumentation, authenticated current attributes, total deterministic model representations, exact verifier, and stated provenance assumptions. It does not establish the epistemic correctness of a human or learned assessor.

8 Monotone Certified Governance Memory

This section derives a preservation invariant from the certified route semantics. Similarity may retrieve a candidate record, but it cannot justify replay; each record is therefore verified universally over its declared region.

For each $v \in \mathcal{V}_{\mathrm{act}}$, let Z_v be a finite retrieval-key set and

$$\alpha_v : \mathcal{R}_v \to Z_v$$

a deterministic index. It has no soundness force by itself.

For each $v \in \mathcal{V}_{\mathrm{act}}$, let Out_v be a finite set of deterministic certified governance dispositions. Each outcome contains:

(i) the selected governor–profile route (i,u);

(ii) the admitted controlled event set;

(iii) a routing tag in {permit, reject, review}; and

(iv) the proof or proof reference needed to reconstruct a currently valid proof-carrying command.

Let

$$G_v : \mathcal{R}_v \times \mathcal{X} \to \mathsf{Out}_v$$

be the authoritative full-evaluation outcome, and let

$$\Theta_v : \mathcal{R}_v \times \mathcal{X} \times I \times \mathcal{U} \to \mathsf{Out}_v$$

be the disposition obtained by executing the named route (i, u) under the complete versioned model. Define

$$\mathsf{Out}_v^{\mathrm{auto}} \subseteq \mathsf{Out}_v$$

to contain dispositions that require no human review or escalation. A record whose outcome lies outside this subset may still be replayable; replay coverage and automatic resolution are distinct.

A governance-memory record is

$$R = (z, v, C, \xi, i, u, \theta, \omega_R, \varpi_R),$$

where $v \in \mathcal{V}_{\mathrm{act}}$, $z \in Z_v$, $C \subseteq N$, $i \in I$, $u \in \mathcal{U}$, $\xi \in \prod_{j \in C} X_j$, and $\theta \in \mathsf{Out}_v$. Write $v_R = v$ and $\theta_R = \theta$ for its version and outcome components.

$$\omega_R : \mathcal{R}_v \times \mathcal{X} \to \{0, 1\}$$

is the total deterministic obligation predicate, and ϖ_R is the proof object establishing the universal record-verification claim. Its region is

$$\mathrm{Reg}(R) = \{(q, x) \in \mathcal{R}_v \times \mathcal{X} : \alpha_v(q) = z,\ x \restriction_C = \xi,\ \omega_R(q, x) = 1\}.$$

Definition 8.1 (Verified record). A record R is verified when:

(i) $\mathrm{AE}^v_{i,u}(q, x) = 1$ for every $(q, x) \in \mathrm{Reg}(R)$;

(ii) $\Theta_v(q, x, i, u) = \theta$ for every $(q, x) \in \mathrm{Reg}(R)$;

(iii) $G_v(q, x) = \theta$ for every $(q, x) \in \mathrm{Reg}(R)$; and

(iv) ϖ_R is an authenticated proof that binds all record fields and establishes the universal claims in (i)–(iii).

The insertion proof may remain authentic while replay-time premises expire. Authenticity therefore establishes only what was proved about the declared region; it does not by itself establish that the current request and context lie in that region.

Definition 8.2 (Current-bound record replay). Let

$$\chi_{\mathrm{cur}} = (h_{\mathrm{cur}}, q_{\mathrm{cur}}, v_{\mathrm{cur}}, x_{\mathrm{cur}}, s_{\mathrm{cur}}) \in \mathsf{Snap}_{v_{\mathrm{cur}}},$$

be the authenticated current-execution snapshot. A verified record $R = (z, v, C, \xi, i, u, \theta, \omega_R, \varpi_R)$ is replay-applicable only when the runtime checks

$$v_{\mathrm{cur}} = v, \qquad \alpha_v(q_{\mathrm{cur}}) = z, \qquad x_{\mathrm{cur}} \restriction_C = \xi, \tag{RM1}$$

$$\omega_R(q_{\mathrm{cur}}, x_{\mathrm{cur}}) = 1, \qquad s_{\mathrm{cur}} = \zeta_v(q_{\mathrm{cur}}, x_{\mathrm{cur}}), \qquad u \in \mathcal{U}, \tag{RM2}$$

and revalidates ϖ_R, every expiring premise, and any deployment or authority premise represented in the active model. If replay reconstructs a command, it must create or verify a proof bound to the current proposal identifier h_{cur} and pass Definition 7.8; an old proposal-bound command is not replayed verbatim.

Lemma 8.3 (Verified overlap compatibility). *Any two verified records whose regions overlap have the same outcome on that overlap.*

Proof. Let R and R' have version components v and v', and outcome components θ and θ', respectively. If $(q, x) \in \mathrm{Reg}(R) \cap \mathrm{Reg}(R')$, membership in the version-tagged domain Ω forces $v = v'$. Verification then gives $\theta = G_v(q, x) = \theta'$. ∎

A finite library $\mathcal{M}$ is *verified* when every member is verified. Define

$$\mathrm{Cov}(\mathcal{M}) = \bigcup_{R \in \mathcal{M}} \mathrm{Reg}(R), \qquad \Omega_{\mathrm{unres}}(\mathcal{M}) = \Omega \setminus \mathrm{Cov}(\mathcal{M}).$$

By Lemma 8.3, the replay map

$$D_{\mathcal{M}} : \mathrm{Cov}(\mathcal{M}) \to \bigcup_{v \in \mathcal{V}_{\mathrm{act}}} \mathsf{Out}_v$$

is well defined by the common outcome of all matching records.

Define automatic coverage separately:

$$\mathrm{Cov}_{\mathrm{auto}}(\mathcal{M}) = \bigcup_{\substack{R \in \mathcal{M} \\ \theta_R \in \mathsf{Out}_{v_R}^{\mathrm{auto}}}} \mathrm{Reg}(R).$$

Theorem 8.4 (Monotone verified governance memory). *Let $\mathcal{M}$ be a finite verified library and let R be another verified record. Set $\mathcal{M}' = \mathcal{M} \cup \{R\}$. Then:*

(i) *$\mathcal{M}'$ is verified;*

(ii) *$D_{\mathcal{M}'}(q,x) = D_{\mathcal{M}}(q,x)$ for every $(q,x) \in \mathrm{Cov}(\mathcal{M})$;*

(iii) *$\mathrm{Cov}(\mathcal{M}) \subseteq \mathrm{Cov}(\mathcal{M}')$; and*

(iv) *$\Omega_{\mathrm{unres}}(\mathcal{M}') \subseteq \Omega_{\mathrm{unres}}(\mathcal{M})$.*

Thus insertion weakly decreases the region requiring repeated full governance evaluation. It does not imply that every covered outcome is automatic.

Proof. Verification of old records is unchanged and R is verified. On an old covered point, any newly matching record agrees with every old matching record by Lemma 8.3, proving decision preservation. Finally,

$$\mathrm{Cov}(\mathcal{M}') = \mathrm{Cov}(\mathcal{M}) \cup \mathrm{Reg}(R),$$

and taking complements gives the unresolved-region inclusion. ∎

Corollary 8.5 (Automatic-coverage monotonicity). *Under the hypotheses and notation of Theorem 8.4, if $\theta_R \in \mathsf{Out}_{v_R}^{\mathrm{auto}}$, then*

$$\mathrm{Cov}_{\mathrm{auto}}(\mathcal{M}) \subseteq \mathrm{Cov}_{\mathrm{auto}}(\mathcal{M}').$$

If $\theta_R \notin \mathsf{Out}_{v_R}^{\mathrm{auto}}$, deterministic replay coverage may grow while automatic coverage remains unchanged.

8.1 Record verification, replay, and migration

For $R = (z, v, C, \xi, i, u, \theta, \omega_R, \varpi_R)$, verification is the finite query

$$\nexists (q,x) \in \mathrm{Reg}(R) : \mathrm{AE}^{v}_{i,u}(q,x) = 0 \ \vee \ \Theta_v(q,x,i,u) \neq \theta \ \vee \ G_v(q,x) \neq \theta. \tag{RV}$$

Proposition 8.6 (Exact insertion test). *A proposed record is verified if and only if (RV) is unsatisfiable and its proof object ϖ_R authenticates that result.*

Proof. Unsatisfiability is exactly the universal conjunction in the definition of verified record; authentication binds the proof to the record fields. ∎

Corollary 8.7 (One-record certified reuse). *If R is verified and the authenticated current snapshot passes the replay checks in Definition 8.2, then $(q_{\mathrm{cur}}, x_{\mathrm{cur}}) \in \mathrm{Reg}(R)$ and the stored disposition may be reused without repeating full governance evaluation. Any reconstructed command must nevertheless pass current-bound proof acceptance.*

Cross-version migration requires explicit maps. For versions $v, v' \in \mathcal{V}_{\mathrm{act}}$, a migration specification from v to v' consists of:

$$m_R : \mathcal{R}_v \to \mathcal{R}_{v'}, \quad m_Z : Z_v \to Z_{v'}, \quad m_{\mathrm{Out}} : \mathsf{Out}_v \to \mathsf{Out}_{v'},$$

a coordinate map $m_X : \mathcal{X} \to \mathcal{X}$, a profile-identifier map $m_U : \mathcal{U} \to \mathcal{U}$, a cylinder-descriptor translation

$$\mathsf{Cyl}(\mathcal{X}) = \bigsqcup_{C \subseteq N} \left(\{C\} \times \prod_{j \in C} X_j \right), \qquad m_{\mathrm{cyl}} : \mathsf{Cyl}(\mathcal{X}) \to \mathsf{Cyl}(\mathcal{X}),$$

and an obligation-predicate translation

$$m_\omega : (\mathcal{R}_v \times \mathcal{X} \to \{0,1\}) \longrightarrow (\mathcal{R}_{v'} \times \mathcal{X} \to \{0,1\}).$$

For a source record $R = (z, v, C, \xi, i, u, \theta, \omega_R, \varpi_R)$, write

$$m_{\mathrm{cyl}}(C, \xi) = (C', \xi').$$

The maps specify the candidate target fields

$$\big(m_Z(z), v', C', \xi', i, m_U(u), m_{\mathrm{Out}}(\theta), m_\omega(\omega_R)\big).$$

A new proof object must then establish a fresh (RV) check under the complete v'-model. Without declared migration maps, "reinterpretation" is an implementation suggestion rather than a mathematical operation.

Proposition 8.8 (Outcome-homogeneous cache fibres). *Fix $v \in \mathcal{V}_{\mathrm{act}}$. Let K be a cache-key set and $\kappa : \mathcal{R}_v \times \mathcal{X} \to K$ be deterministic. Define the used key set*

$$K_\kappa = \kappa(\mathcal{R}_v \times \mathcal{X}).$$

For $k \in K_\kappa$, call the nonempty fibre $\kappa^{-1}(k)$ G_v-homogeneous when G_v is constant on it. A map $d : K_\kappa \to \mathsf{Out}_v$ satisfies $d(\kappa(q, x)) = G_v(q, x)$ everywhere if and only if every used fibre of κ is G_v-homogeneous.

Proof. If $d \circ \kappa = G_v$, equal used keys imply equal authoritative outcomes. Conversely, assign each used key the common outcome of its nonempty homogeneous fibre. This also covers the empty-domain case, when $K_\kappa = \varnothing$. ∎

Embedding similarity may therefore retrieve a candidate record, but a cluster crossing an authority, effect, version, or outcome boundary cannot justify one replay decision.

9 Complexity, Deployment, and Checker Families

9.1 Explicit independently valid transparent verification

For an explicitly materialised finite version, compute W, each $\mathcal{U}^{=}(s)$, each complete decision-state observation cell, and each $\mathcal{U}_i^{AE}(y)$, then test (AEC). The input-size measure includes the runtime graph, public labels, transducer recurrences, output tables, observations, authority, selectors, and finite event-set representations.

Proposition 9.1 (Explicit independently valid transparent verification). *Independently valid transparent exact marked-nonblocking realisability is decidable in polynomial time in the size of an explicitly materialised finite model.*

Proof. The fixed point for W uses finite graph operations. Transducer consistency is checked on every defined transition of the finite model. Exact-profile membership compares finite event sets, each aligned support scans one finite observation cell, and coverage is a final decision-state scan. ∎

9.2 Succinct circuit-fusion complexity

The decision problem in this subsection uses an explicitly enumerated finite runtime, decision-state observations, finite command alphabets, authority tables, and transducer-effect tables, together with a polynomial-size Boolean circuit for Ψ. The input size is the total encoding length of these objects. The problem asks whether the cell-command constraints are satisfiable. Recognising a construction-history claim such as target independence is not part of this decision language.

Theorem 9.2 (Circuit-fusion realisability). *The unrestricted exact fused-governance realisability problem with Boolean-circuit fusion is NP-complete, even with one decision state, no exogenous events, universal authority over one shared two-profile deployed interface, and one observation per governor. NP-hardness persists for the circuits produced by the reduction, which satisfy (EA) and are generated from the SAT instance without consulting Q, S_m, W, or Viab_c.*

Proof. Membership in NP follows by guessing one command per observation and evaluating the transducers, authority predicates, effect vectors, and supplied fusion circuit. No verification of target-independence provenance or of (EA) is needed for membership in the unrestricted decision problem.

For hardness, reduce SAT. Given $\varphi(x_1, \ldots, x_n)$, define

$$S = \{s, m, d\}, \quad s_0 = s, \quad Q = \{s, m\}, \quad S_m = \{m\},$$

$$\Sigma_u = \varnothing, \qquad \Sigma_c = \{a, b\},$$

and let the only transitions be

$$\delta(s, a) = m, \qquad \delta(s, b) = d.$$

Thus s is the only decision state and $\mathrm{Viab}_c(s) = \{a\}$.

Let the governor set be $I = \{1, \dots, n\}$, and use one shared ambient catalogue and deployed interface

$$\overline{\mathcal{U}} = \mathcal{U} = \{u^0, u^1\}.$$

Every governor has one observation value and universal authority relative to $\mathcal{U}$. Give the three states distinct public labels. Both profiles are one-state transducers with identity update and total output functions on those labels: profile u^0 always outputs $\varnothing$, while u^1 always outputs $\{a, b\}$.

For a command vector $\mathbf{c} = (c_i)_{i \in I}$ and an arbitrary supplied effect vector $\mathbf{A} = (A_i)_{i \in I} \in (2^{\{a,b\}})^I$, let $\mathbf{1}[P] \in \{0, 1\}$ denote the indicator of proposition P, and set $b_i = \mathbf{1}[a \in A_i]$. The circuit Ψ_s includes a in its output exactly when $\varphi(b_1, \dots, b_n)$ is true, and includes b exactly when

$$\exists i \in I : \quad c_i \neq \bot \quad \text{and} \quad b \in A_i.$$

It is fixed from φ and does not query W or Viab_c. It satisfies (EA): for any command vector with an active issuer, if b is absent from the fused output, add b to one active supplied effect; if b is present, replace every active supplied effect by $\varnothing$. Either change toggles membership of b while leaving abstaining components fixed.

At the declared effect vector, exact realisation excludes every active use of u^1. Any satisfying assignment is nevertheless representable: issue $\bot$ for a true variable, giving effect $\{a, b\}$ without an active issuer, and issue u^0 for a false variable. Conversely, any exact command vector induces Boolean values b_i satisfying φ. Exact realisation therefore exists exactly when φ is satisfiable. ∎

If one instead defines a promise problem restricted to supplied circuits that are target-independent and satisfy (EA), the reduction proves NP-hardness of that promise subclass. NP membership for a language that must also reject nonconforming circuits requires a separate polynomial-time conformance test, for example a syntactically restricted fusion grammar; it does not follow from the theorem above.

9.3 Minimum independently valid governor–profile route deployment

For each route $(i, u) \in I \times \mathcal{U}$, define its decision-state support

$$\mathrm{Supp}_{i,u} = \bigcup \{C_i(y) : y \in O_i(D),\ u \in \mathcal{U}_i^{AE}(y)\}.$$

Assign each route $(i, u) \in I \times \mathcal{U}$ a positive deployment cost $c_{i,u}$. A selected route family is a set $J \subseteq I \times \mathcal{U}$; it is independently sufficient when

$$D \subseteq \bigcup_{(i,u) \in J} \mathrm{Supp}_{i,u},$$

and its total cost is $\sum_{(i,u) \in J} c_{i,u}$.

Theorem 9.3 (Route-deployment cover equivalence and hardness). *Minimum-cost independently valid transparent governor–profile route deployment is weighted Set Cover and is NP-hard, even with one fully observing governor and universal authority.*

Proof. A deployed route family is independently sufficient exactly when its supports cover D, which is weighted Set Cover.

For hardness, take a weighted Set Cover instance with universe $\{1, \dots, n\}$, subsets $B_1, \dots, B_m$, and costs $c_1, \dots, c_m$. Use initial state $s_0 = r$, state set

$$S = \{r, m, d, s_1, \dots, s_n\}, \quad Q = \{r, m, s_1, \dots, s_n\}, \quad S_m = \{m\},$$

exogenous alphabet $\Sigma_u = \{e_1, \dots, e_n\}$, controlled alphabet $\Sigma_c = \{a, b\}$, exogenous transitions $\delta(r, e_k) = s_k$, and controlled transitions

$$\delta(s_k, a) = m, \qquad \delta(s_k, b) = d.$$

Thus $W = Q$, $D = \{s_1, \dots, s_n\}$, and $\mathrm{Viab}_c(s_k) = \{a\}$.

Let $\Gamma = \{\gamma_r, \gamma_m, \gamma_d, \gamma_1, \dots, \gamma_n\}$ and define the public-label map by $\gamma(x) = \gamma_x$ for every state $x \in S$. Set $\overline{\mathcal{U}} = \mathcal{U} = \{u_1, \dots, u_m\}$. For every $j \in \{1, \dots, m\}$, profile u_j has singleton transducer state set $\{\rho_j\}$, initial state ρ_j, identity update $\eta_{u_j}(\rho_j, \gamma_x) = \rho_j$ for every $x \in S$, and total output

$$\lambda_{u_j}(\rho_j, \gamma_k) = \begin{cases} \{a\}, & k \in B_j, \\ \varnothing, & k \notin B_j, \end{cases} \qquad \lambda_{u_j}(\rho_j, \gamma_r) = \lambda_{u_j}(\rho_j, \gamma_m) = \lambda_{u_j}(\rho_j, \gamma_d) = \varnothing.$$

The recurrence is immediate and u_j is exact exactly on the decision states indexed by B_j. Let $I = \{i_0\}$. The sole governor fully observes each s_k, has universal authority over $\mathcal{U} = \{u_1, \dots, u_m\}$, and route (i_0, u_j) has cost c_j. Route covers and Set Cover subsets coincide. NP-hardness follows from the classical result [10]. ∎

9.4 Four checker families

Existence, deployment conformance, proof validity, and replay validity are different questions. A verifier should expose four checker families rather than collapse them into one obstruction label.

Target and viability computation.

- Initial viability: $s_0 \notin W$;
- Candidate target: an uncontrollable successor leaves a supplied candidate region or a state lacks a marked path.

Independently valid transparent architecture synthesis.

- Interface: $\mathcal{U}^{=}(s) = \varnothing$;
- Authority: exact profiles exist, but no governor is authorised pointwise;
- Alignment: pointwise authorised exact pairs exist, but none survives a complete decision-state observation cell.

Wrapper and deployment conformance.

- Mediation: a controlled effect bypasses proposal–permit–commit;
- Atomicity/revalidation: an interleaving or expiring premise is absent from the model or not rechecked.

Proof and replay validation.

- Certificate: a cylinder contains a bad completion;
- Record: a record region contains a witness against route validity or outcome homogeneity;
- Version: active-version or migration checks fail.

These categories need not be mutually exclusive. An implementation may impose a documented first-failure order for reporting.

Proposition 9.4 (Completeness of independently valid transparent architecture classification)**.** *After W is computed and $s_0 \in W$, failure of independently valid transparent architecture synthesis produces an Interface, Authority, or Alignment witness. If none exists, (AEC) holds and an independently valid transparent exact controller exists.*

Proof. At any uncovered state, either no exact profile exists; exact profiles exist but no issuer is authorised for one; or a pointwise authorised exact pair is lost under the relevant cell intersection. These are the three cases. If no state is uncovered, Corollary 4.4 applies. ∎

Finite deployment-conformance note. Given a supplied finite implementation transition system and an explicitly enumerated finite set of atomicity and revalidation obligations, each obligation can be checked as a structural or reachability predicate. Such checking is relative to the supplied implementation model and obligation list: it cannot prove that every physical effect channel, interleaving, or expiry condition has been modelled. Failure of implementation conformance does not imply that the abstract architecture lacks an exact controller.

10 Worked Payment-Release Model

This section instantiates the definitions as a finite model rather than relying on an informal scenario.

10.1 Runtime and exact region

Let the states be

$$S_{\text{pay}} = \{r,\ h,\ d,\ v_h,\ v_d,\ p,\ \text{released}, \text{aborted}, \text{unsafe}\},$$

where r is the initial state, h and d are respectively healthy-tool and degraded-tool release decisions, v_h, v_d are human-review states, and p is a prepared-payment state. Let

$$S_{m,\text{pay}} = \{\text{released}, \text{aborted}\}, \qquad Q_{\text{pay}} = S_{\text{pay}} \setminus \{\text{unsafe}\}.$$

The exogenous events are

$$\Sigma_{u,\text{pay}} = \{\text{healthy}, \text{degraded}\},$$

and the controlled events are

$$\Sigma_{c,\text{pay}} = \{\text{release}, \text{prepare}, \text{routeReview}, \text{block}, \\ \text{approveReview}, \text{rejectReview}, \text{executePrepared}, \text{cancelPrepared}\}.$$

Let

$$\Sigma_{\text{pay}} = \Sigma_{c,\text{pay}} \dot{\cup} \Sigma_{u,\text{pay}},$$

and let $\delta_{\text{pay}} : S_{\text{pay}} \times \Sigma_{\text{pay}} \rightharpoonup S_{\text{pay}}$ be the partial transition function in Table 1. Thus the payment runtime is

$$\mathcal{A}_{\text{pay}} = (S_{\text{pay}}, r, \Sigma_{\text{pay}}, \delta_{\text{pay}}, S_{m,\text{pay}}).$$

Within this worked model, unadorned instances of the generic notation refer to the corresponding payment-model objects once those objects are defined below, unless a subscript is needed for clarity.

Table 1. Payment-runtime transitions. Unlisted pairs are undefined.

Source	Event	Successor
r	healthy	h
	degraded	d
h	release	released
	prepare	p
	routeReview	v_h
	block	aborted
d	release	unsafe
	prepare	p
	routeReview	v_d
	block	aborted
v_h	approveReview	released
	rejectReview	aborted
v_d	approveReview	p
	rejectReview	aborted
p	executePrepared	released
	cancelPrepared	aborted

The fixed-point computation gives

$$W_{\text{pay}} = Q_{\text{pay}}, \qquad D_{\text{pay}} = \{h, d, v_h, v_d, p\}.$$

The exact controlled successor sets are

$$\begin{aligned} \text{Viab}_c(h) &= \{\text{release}, \text{prepare}, \text{routeReview}, \text{block}\}, \\ \text{Viab}_c(d) &= \{\text{prepare}, \text{routeReview}, \text{block}\}, \\ \text{Viab}_c(v_h) = \text{Viab}_c(v_d) &= \{\text{approveReview}, \text{rejectReview}\}, \\ \text{Viab}_c(p) &= \{\text{executePrepared}, \text{cancelPrepared}\}. \end{aligned}$$

10.2 Labels, profiles, observations, and authority

Let

$$\Gamma_{\text{pay}} = \{\gamma_s : s \in S_{\text{pay}}\}, \qquad \gamma_{\text{pay}}(s) = \gamma_s \qquad (s \in S_{\text{pay}}),$$

so every payment state has a distinct authenticated public label. Each profile below has a singleton transducer state set $\{\rho_u\}$, initial state ρ_u, and identity update $\eta_u(\rho_u, \gamma_s) = \rho_u$. The table defines $\lambda_u(\rho_u, \gamma_s)$ at decision-state labels; every unlisted label has output $\varnothing$. These data define total output functions and satisfy (T0)–(TU).

For compactness, write the following literal controlled-event sets:

$$\begin{aligned} A_h &= \{\text{release}, \text{prepare}, \text{routeReview}, \text{block}\} = \text{Viab}_c(h), \\ A_d &= \{\text{prepare}, \text{routeReview}, \text{block}\} = \text{Viab}_c(d), \\ A_v &= \{\text{approveReview}, \text{rejectReview}\} = \text{Viab}_c(v_h) = \text{Viab}_c(v_d), \\ A_p &= \{\text{executePrepared}, \text{cancelPrepared}\} = \text{Viab}_c(p). \end{aligned}$$

The profile output tables are predeclared using the literal event sets on the left. Their equalities with the viable successor sets are consequences of the preceding fixed-point computation and are not inputs queried by the transducers.

Table 2. Profile outputs before intersection with available controlled events.

Profile	h	d	v_h	v_d	p
pass	A_h	A_h	$\varnothing$	$\varnothing$	$\varnothing$
degradedGuard	A_d	A_d	$\varnothing$	$\varnothing$	$\varnothing$
reviewDecision	$\varnothing$	$\varnothing$	A_v	A_v	$\varnothing$
preparedDecision	$\varnothing$	$\varnothing$	$\varnothing$	$\varnothing$	A_p
adaptiveRelease	A_h	A_d	$\varnothing$	$\varnothing$	$\varnothing$

The ambient payment-profile catalogue contains all five profiles in the table, but the actual deployed interface initially excludes adaptiveRelease:

$$\overline{\mathcal{U}}_{\text{pay}} = \mathcal{U}_{\text{pay}} \cup \{\text{adaptiveRelease}\}, \qquad \mathcal{U}_{\text{pay}} = \{\text{pass}, \text{degradedGuard}, \text{reviewDecision}, \text{preparedDecision}\}.$$

Let the payment-governor set be $I_{\text{pay}} = \{F, R, H, O\}$, representing finance, risk, human review, and operations. Using $C_i(y) = \{s \in D_{\text{pay}} : O_i(s) = y\}$ for payment-model observation cells, the following displayed cells specify their non-singleton or operationally relevant observations:

$$C_F(y_{\text{pre}}) = \{h, d\}, \qquad C_R(y_h) = \{h\}, \quad C_R(y_d) = \{d\},$$

$$C_H(y_{\text{review}}) = \{v_h, v_d\}, \qquad C_O(y_p) = \{p\}.$$

For each $i \in I_{\text{pay}}$, let Y_i be the finite set containing the displayed observation values and the fresh values introduced next. Every decision state not listed for a governor is assigned a fresh governor-specific singleton observation value. This convention defines O_i on D_{pay}; extend it arbitrarily to a map $O_i : W_{\text{pay}} \to Y_i$. Thus tool health is hidden from F but visible to R. Authority is

$$\text{Auth}_F(h) = \text{Auth}_F(d) = \{\text{pass}, \text{degradedGuard}, \text{adaptiveRelease}, \bot\},$$

$$\text{Auth}_R(h) = \text{Auth}_R(d) = \{\bot\},$$

$$\text{Auth}_H(v_h) = \text{Auth}_H(v_d) = \{\text{reviewDecision}, \bot\}, \qquad \text{Auth}_O(p) = \{\text{preparedDecision}, \bot\}.$$

At every unrelated decision state, each governor is authorised only for $\bot$, completing the authority maps on D_{pay}; extend those maps arbitrarily to $W_{\text{pay}} \setminus D_{\text{pay}}$. The transparent selector scans the fixed precedence order $F \succ H \succ O \succ R$ and returns the first non-abstaining governor.

The exact profile sets are

$$\mathcal{U}^{=}(h) = \{\text{pass}\}, \quad \mathcal{U}^{=}(d) = \{\text{degradedGuard}\},$$

$$\mathcal{U}^{=}(v_h) = \mathcal{U}^{=}(v_d) = \{\text{reviewDecision}\}, \quad \mathcal{U}^{=}(p) = \{\text{preparedDecision}\}.$$

Therefore

$$\mathcal{U}_F^{AE}(y_{\text{pre}}) = \{\text{pass}\} \cap \{\text{degradedGuard}\} = \varnothing.$$

The remaining decision states have nonempty aligned supports. The actual architecture fails independently valid transparent synthesis only at $\{h, d\}$, and the aligned-witness architecture checker returns Alignment: pointwise authorised exact profiles exist, but no one profile survives F's complete cell.

Two repairs are formally distinct:

1. *observation repair*: authenticate tool health to F, refining its cell into $\{h\}$ and $\{d\}$; or
2. *deployment repair*: deploy the already predeclared, label-sensitive, and authorised adaptiveRelease transducer, allowing F to issue it uniformly.

Either repair makes every decision state satisfy (AEC) and therefore restores an independently valid transparent realisation.

10.3 Illustrative certificate and explicit negative completion

Denote the observation-repair version by v_{pay}. The runtime, profile effects, observations, and authority above are fully specified. The context layer below is illustrative because the finite coordinate domains, complete request set, request–context map, and authority/effect rules for every completion are not enumerated. Accordingly, the positive cylinder is a certificate candidate conditional on an exact verifier, while the negative completion is explicit. List the context coordinates before choosing the candidate:

$$\begin{aligned}\text{Attr}_{\text{pay}} = (&\text{tenant}, \text{supplier status}, \text{invoice match}, \text{amount band},\\ &\text{authority lease status}, \text{policy version}, \text{memory version}, \text{tool health},\\ &\text{reversibility}, \text{reviewer availability}, \text{reporting tag}, \text{description class}).\end{aligned}$$

Index this ordered list by $N_{\text{pay}} = \{1, \ldots, 12\}$. For each coordinate index $j \in N_{\text{pay}}$, let X_j^{pay} be the finite nonempty value domain of the corresponding coordinate and define

$$\mathcal{X}_{\text{pay}} = \prod_{j \in N_{\text{pay}}} X_j^{\text{pay}}.$$

Here *authority lease status* is not a policy version, memory version, or proof version.

For a healthy-tool request q, let $x \in \mathcal{X}_{\text{pay}}$ have a valid authority lease, current policy and memory versions, matched invoice, permitted supplier, and tool-health coordinate equal to healthy. Identifying each coordinate name with its index in N_{pay}, for finance profile pass, the set

$$\begin{aligned}C = \{&\text{supplier status}, \text{invoice match}, \text{amount band}, \text{authority lease status},\\ &\text{policy version}, \text{memory version}, \text{tool health}, \text{reversibility}, \text{reviewer availability}\}\end{aligned}$$

is an illustrative positive certificate candidate. It becomes a verified certificate only if the exact verifier establishes that omitted tenant, reporting, and descriptive fields do not change (AW) over the fully materialised finite completion space.

The smaller candidate $C \setminus \{\text{tool health}\}$ fails. A completion that keeps every remaining fixed coordinate but changes tool health to degraded maps to state d, where

$$E_{\text{pass}}(d) \neq \text{Viab}_c(d)$$

because the profile retains the unsafe direct-release event. This completion is an explicit certificate counterexample. The same model also shows why human review must be represented by an explicit routing event: routeReview leads to v_h or v_d, where approve and reject transitions are displayed and can be checked.

11 Verification and Falsification Protocol

The guarantees are relative to a finite formal model. A companion implementation study would test modelling fidelity and checker conformance rather than substitute empirical accuracy for proof.

Prospective companion artifact. A machine-readable artifact can contain the versioned runtime, safe and marked sets, public labels, ambient catalogue and deployed interface, transducer recurrences and outputs, fusion artefact and provenance record, governor observations, decision-state authority, request–context map, outcome semantics, certificates, memory records, and migration maps. Synthesis, deployment conformance, certificate checking, and record checking should be separate procedures with separate failure types. No such implementation is claimed in this paper.

Primary exactness checks. For every governed state, compare

$$|\mu_g(s) \setminus \text{Viab}_c(s)| \quad \text{and} \quad |\text{Viab}_c(s) \setminus \mu_g(s)|.$$

The first quantity counts nonviable retained branches, including unsafe or blocking branches; the second counts viable branches unnecessarily removed. Both vanish under exact governance. Independently check (T0)–(TU), (FU), marked wrapper coaccessibility, current-execution binding, replay-region membership, and global-fibre equality (GF). For a general-fusion artifact, separately record the optional (EA) check. Target independence must be checked through the declared provenance mechanism or a restricted fusion grammar; it is not inferred from (EA).

Regression countermodels. Appendix C specifies eleven proposed finite regression cases covering decision-domain scope, global information fibres, noncomposition, the separation between route-bearing and independently valid transparent exactness, transducer recurrence, fusion provenance, current binding, certificate counterexamples, record overlap, and the distinction between replay and automatic coverage. They are proposed falsification checks, not reported experimental results or substitutes for proof.

Ablations. A future artifact can coarsen an observation partition, revoke authority, remove a profile, corrupt transducer recurrence, permit a mediation bypass, weaken a certificate, or replace a verified record region by embedding similarity. The reported outcome should identify the applicable checker family rather than classify every failure as architectural nonrealisability.

Efficiency measures. Relevant measurements include viability computation, cell-command solving, aligned-support construction, certificate size and evidence cost, verifier counterexamples, full-evaluation rate, replay coverage, automatic coverage, and commit-time revalidation cost. Theorem 8.4 predicts nondecreasing replay coverage, while automatic coverage grows only under its separate corollary.

12 Scope and Limitations

The results are relative to the encoded safe envelope, marked states, public instrumentation, transducer recurrence, authority predicates, observation monitors, fusion operator, request–context map, outcome semantics, and verifier. A certificate cannot repair an omitted hazard, compromised attribute source, incorrect target, or hidden effect channel.

Proposal–decision–commit is atomic only at the supervisory-control abstraction. Any exogenous event that can occur during physical review must be represented at pending and cleared states and included in uncontrollable closure. Long-running review also requires explicit timeout, lease, and commit-revalidation transitions.

Marked nonblockingness means that every reachable wrapper state retains some marked path. It does not guarantee termination under every scheduler. Fairness, strong progress, probability, utility, and adversarial timing require additional game or stochastic structure.

Profiles are event filters. Rollback, redirection, simulation, or inserted review are available only when corresponding events and transitions already occur in the original runtime. The theory can verify routing and deterministic use of human or learned evidence, but not the epistemic correctness of the assessor.

The aligned-witness condition characterises independently valid transparent governance, not every selector-dependent or unrestricted fused realisation. Route-bearing transparent exactness may depend on a state-indexed selector, while general

fusion may define behaviour that is not attributable to a selected non-abstaining route. The proof-carrying results deliberately enforce the stronger per-route invariant.

The finite verification and complexity results assume finite materialised domains or total deterministic decidable representations. Cross-version preservation is not automatic: explicit migration maps and a fresh verification query are required.

Finally, the paper does not claim novelty for Hitting Set, Set Cover, generic abductive/contrastive duality, CEGIS, or cache factorisation. Those are supporting mechanisms applied to the combined governance object.

13 Conclusion

This paper developed an exact marked-nonblocking governance theory for finite instrumented agent execution. Within the encoded model, the proposal–decision–commit wrapper mediates every represented controlled effect, projects exactly to the admitted original trace language, and preserves marked coaccessibility when its admission map equals the viable controlled-successor map.

Fixed intervention transducers separate ambient catalogue membership from deployment, while observation and authority constrain which routes can be issued. Transparent governance forms a strict hierarchy: independently valid tuples are route-bearing transparent exact, and route-bearing tuples are instances of general fused exact governance. State-indexed selection can make the first implication strict, but the five-state construction shows that separate interface and issuer-side adequacy still need not compose in the actual route-bearing deployment. Observation refinement, authority enlargement, and deployment enlargement are monotone repair levers, but they are not interchangeable.

The proof-carrying layer enforces the stronger per-route invariant. An aligned-witness certificate proves one governor–profile route authorised and exact throughout its complete global information fibre; current-bound acceptance checks that the proposal, request, context restriction, version, and runtime state still match that proof. Under explicit certificate-coverage and commit-revalidation assumptions, independently justified accepted routes compose to $\mu^{v}(s) = \mathrm{Viab}_{c}^{v}(s)$, and stale premises are rejected or revalidated before commit.

Universally verified replay regions then provide a preservation invariant: adding a verified record cannot change an earlier certified replay decision and can only enlarge verified replay coverage, while automatic resolution remains a separate property. Relative to the encoded finite model, authenticated evidence, fixed intervention semantics, exact verification, and deployment conformance, the resulting governance layer admits exactly the represented controlled branches retained by the greatest marked-nonblocking region—neither permitting branches outside that region nor deleting viable completion paths merely for conservatism.

A Computing the Greatest Nonblocking Region

The greatest admissible region used in Section 3 can be computed by finite descending iteration. For $R \subseteq Q$, define

$$\mathrm{UC}(R) = \{s \in R : \delta(s,\sigma) \in R \text{ for every } \sigma \in \mathrm{Av}_u(s)\},$$

and let $\mathrm{CoAcc}(R)$ be the states of R from which some path contained in R reaches $S_m \cap R$. For integers $k \geq 0$, set

$$R_0 = Q, \qquad R_{k+1} = \mathrm{UC}(R_k) \cap \mathrm{CoAcc}(R_k).$$

Theorem A.1 (Finite fixed-point computation). *The descending sequence stabilises after at most $|Q|$ strict iterations, and its limit is the greatest admissible region W.*

Proof. The sequence is descending because both operators return subsets of their argument. Since Q is finite, it stabilises after at most $|Q|$ strict removals. Let the fixed point be R_*. Equality

$$R_* = \mathrm{UC}(R_*) \cap \mathrm{CoAcc}(R_*)$$

shows that R_* is uncontrollable-closed and marked-coaccessible, hence admissible.

Let $A \subseteq Q$ be any admissible region. We prove $A \subseteq R_k$ by induction. It holds for $R_0 = Q$. If $A \subseteq R_k$, then every uncontrollable successor of a state in A remains in $A \subseteq R_k$, so $A \subseteq \mathrm{UC}(R_k)$. Every state in A has a marked path lying in $A \subseteq R_k$, so $A \subseteq \mathrm{CoAcc}(R_k)$. Thus $A \subseteq R_{k+1}$. Hence every admissible region is contained in R_*, proving $R_* = W$. ∎

B Finite Independently Valid Transparent Synthesis

The following existence procedure is deliberately separate from checking a supplied deployment.

1. Compute W. If $s_0 \notin W$, return Initial viability.
2. Form D and compute $\mathcal{U}^{=}(s)$ for every $s \in D$. If one is empty, return Interface.
3. Compute $\mathrm{Auth}_i(s) \cap \mathcal{U}^{=}(s)$ for every $i \in I$ and $s \in D$. If their union over issuers is empty at some state, return Authority.
4. Compute $\mathcal{U}_i^{AE}(y)$ over complete decision-state cells. If a state is not covered by any nonempty aligned support, return Alignment.
5. For every $i \in I$ and $y \in O_i(D)$ with $\mathcal{U}_i^{AE}(y) \neq \emptyset$, choose one $u_{i,y} \in \mathcal{U}_i^{AE}(y)$; issue $u_{i,y}$ on y, abstain on decision observations with empty support, extend policies arbitrarily to observations outside the decision domain, and output the policies with their aligned-support proof table.

Theorem B.1 (Soundness and completeness of finite independently valid transparent synthesis)**.** *The procedure returns policies exactly when the transparent architecture admits an independently valid transparent exact realisation. Its Interface, Authority, and Alignment outputs are valid nonrealisability witnesses for that class.*

Proof. The fixed-point theorem establishes the target. Empty exact support is an interface impossibility. If exact profiles exist but no issuer is authorised for one, no legitimate non-abstaining command exists at that state. If pointwise authorised exact pairs exist but complete-cell coverage fails, Corollary 4.4 proves nonrealisability in the independently valid class. If no obstruction is returned, the constructed policies satisfy (AEC), and the same corollary proves exact realisation. ∎

After synthesis, construct the mathematical wrapper from the resulting admission map. An externally implemented wrapper is checked separately against the finite deployment-conformance obligations described in Section 9; a malformed implementation does not retroactively make the abstract architecture unrealisable.

C Proposed Regression Countermodels

The following table specifies finite checks that should be implemented in a reproducible regression artifact. No companion implementation is claimed in this version.

Table 3. Proposed regression suite and expected result.

ID	Model mutation	Expected checker result
R1	A marked nondecision state shares an observation with a decision state, but the profile is authorised only at the decision state.	Pass: legitimacy is restricted to D.
R2	Two requests map to states with one governor observation but require incompatible exact profiles.	Global-cell witness is false.
R3	The five-state noncomposition construction uses recurrent transducers, an ambient catalogue, and route-bearing transparent selection.	The actual deployment has no route-bearing transparent exact realisation and returns Alignment for the independently valid checker; both counterfactual repairs pass.
R4	A desired effect table assigns different transducer states after identical labelled histories.	Transducer recurrence fails.
R5	Fusion returns Viab_c while declared effects are unchanged and the full effect vector is ignored.	The operator fails (EA).
R6	Fusion hard-codes Viab_c but changes on an unused counterfactual supplied-effect vector.	(EA) may pass; target-independence provenance or the restricted grammar must fail.
R7	An authentic certificate is presented with a current context restriction or runtime state that does not match its bound fields.	Current-bound acceptance fails (CB).
R8	Payment certificate includes tool health, then omits it.	The query with tool health passes once the finite completion model is supplied; the second returns the degraded completion.
R9	Two verified record regions overlap.	Outcomes coincide through G_v.
R10	A verified covered outcome has routing tag review.	Replay coverage grows; automatic coverage does not.
R11	Two universally authorised governors share the cell $\{p, q\}$, issue u_a and u_b uniformly, and a predeclared state-indexed selector chooses the exact route separately at p and q.	Route-bearing transparent exactness passes; the independently valid checker returns Alignment because both complete-cell aligned supports are empty.

These proposed checks are finite falsification cases for definition drift, not substitutes for theorem proofs.

D Notation and Tag Glossary

Table 4 records the named conditions used across the paper. The mnemonic column explains each tag's abbreviation; every tagged expression and its constituent symbols are defined in the cited section before the tag is used in a theorem or proof.

Table 4. Named condition tags, their mnemonics, and their formal roles.

Condition tag	Mnemonic	Role	First location
(AL)	Admitted language	Original traces admitted by the state-dependent map μ.	Section 3.2
(T0)	Transducer initialisation	Initial transducer-state consistency.	Section 4.1
(TU)	Transducer update	State recurrence along runtime transitions.	Section 4.1
(AB)	Abstention authority	Abstention belongs to every governor's authority set on D.	Section 4.2
(LG)	Local legitimacy	Legitimacy of an observation-based local policy.	Section 4.2
(EA)	Effect awareness	Optional effect-sensitivity condition used for the hardness robustness clause.	Section 4.2
(FU)	Fusion uniformity	Fusion induced by commands and their declared effects.	Section 4.2
(EX)	Exact realisation	Equality of induced and viable controlled-successor sets.	Section 4.2
(S1)	Selector non-abstention	Transparent selector chooses a non-abstaining issuer.	Section 4.4
(S2)	Selector effect	Transparent fusion returns the selected profile's effect.	Section 4.4
(AEC)	Aligned exact coverage	Complete-cell authority-and-effect coverage characterising independently valid transparent realisation.	Section 4.4
(RC)	Request–context coverage	Surjective request–context map into decision states.	Section 7.2
(GF)	Global fibre	Complete request–context fibre of one governor observation.	Section 7.2
(AW)	Aligned witness	Global-cell aligned authority-and-effect predicate.	Section 7.2
(CERT)	Certificate	Positive aligned-witness certificate cylinder.	Section 7.2
(V)	Verification query	Counterexample query used in certificate synthesis.	Section 7.4
(CB)	Current binding	Current proposal, request, context, version, and state checks.	Section 7.6
(UC)	Universal certificate	Universal claim carried by a command proof.	Section 7.6
(RM1)	Replay match I	Version, retrieval-key, and coordinate checks for replay.	Section 8
(RM2)	Replay match II	Obligation, state, and deployment checks for replay.	Section 8
(RV)	Record verification	Counterexample query for record verification.	Section 8.1

References

[1] M. Alshiekh, R. Bloem, R. Ehlers, B. Könighofer, S. Niekum, and U. Topcu. Safe Reinforcement Learning via Shielding. *Proceedings of the AAAI Conference on Artificial Intelligence*, 32(1), 2018. doi:10.1609/aaai.v32i1.11797.

[2] L. Bauer, M. A. Schneider, E. W. Felten, and A. W. Appel. Access control on the Web using proof-carrying authorization. In *Proceedings of the 3rd DARPA Information Survivability Conference and Exposition*, volume 2, pages 117–119. IEEE, 2003. doi:10.1109/DISCEX.2003.1194942.

[3] R. Bloem, B. Könighofer, R. Könighofer, and C. Wang. Shield Synthesis: Runtime Enforcement for Reactive Systems. In *Tools and Algorithms for the Construction and Analysis of Systems (TACAS 2015)*, LNCS 9035, pages 533–548. Springer, 2015. doi:10.1007/978-3-662-46681-0_51.

[4] M. Fernandez. Agent Control Protocol: Admission Control for Agent Actions. arXiv preprint arXiv:2603.18829, 2026. doi:10.48550/arXiv.2603.18829.

[5] M. Fernandez. Reconstructive Authority Model: Runtime Execution Validity Under Partial Observability. arXiv preprint arXiv:2604.22898, 2026. doi:10.48550/arXiv.2604.22898.

[6] M. Kaptein, V.-J. Khan, and A. Podstavnychy. Runtime Governance for AI Agents: Policies on Paths. arXiv preprint arXiv:2603.16586, 2026. doi:10.48550/arXiv.2603.16586.

[7] J. He and D. Yu. Verifiable Agentic Infrastructure: Proof-Derived Authorization for Sovereign AI Systems. arXiv preprint arXiv:2605.15228, 2026. doi:10.48550/arXiv.2605.15228.

[8] A. Ignatiev, N. Narodytska, and J. Marques-Silva. Abduction-Based Explanations for Machine Learning Models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 1511–1519, 2019. doi:10.1609/aaai.v33i01.33011511.

[9] A. Ignatiev, N. Narodytska, N. Asher, and J. Marques-Silva. From Contrastive to Abductive Explanations and Back Again. In *AIxIA 2020 – Advances in Artificial Intelligence*, LNCS 12414, pages 335–355. Springer, 2021. doi:10.1007/978-3-030-77091-4_21.

[10] R. M. Karp. Reducibility among Combinatorial Problems. In *Complexity of Computer Computations*, pages 85–103. Plenum Press, 1972. doi:10.1007/978-1-4684-2001-2_9.

[11] P. Kingston. Exact Realization and Supremal Synthesis under History-Dependent Veto Authority. Preprint, 2026. doi:10.13140/RG.2.2.35418.50884.

[12] P. Kingston. Equality-Support Covers for Exact Branch-Set Preservation under Fixed Actuator Interfaces. Preprint, 2026. doi:10.13140/RG.2.2.15285.84968.

[13] J. Ligatti, L. Bauer, and D. Walker. Edit automata: enforcement mechanisms for run-time security policies. *International Journal of Information Security*, 4(1–2):2–16, 2005. doi:10.1007/s10207-004-0046-8.

[14] J. Marques-Silva and A. Ignatiev. Delivering Trustworthy AI through Formal XAI. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(11):12342–12350, 2022. doi:10.1609/aaai.v36i11.21499.

[15] U. Mehmood, S. Sheikhi, S. Bak, S. A. Smolka, and S. D. Stoller. The Black-Box Simplex Architecture for Runtime Assurance of Autonomous CPS. In *NASA Formal Methods*, pages 231–250, 2022. doi:10.1007/978-3-031-06773-0_12.

[16] R. Meira-Góes, H. Marchand, and S. Lafortune. Dealing with sensor and actuator deception attacks in supervisory control. *Automatica*, 147:110736, 2023. doi:10.1016/j.automatica.2022.110736.

[17] G. C. Necula. Proof-carrying code. In *Proceedings of the 24th ACM SIGPLAN–SIGACT Symposium on Principles of Programming Languages*, pages 106–119. ACM, 1997. doi:10.1145/263699.263712.

[18] F. Lin and W. M. Wonham. On observability of discrete-event systems. *Information Sciences*, 44(3):173–198, 1988. doi:10.1016/0020-0255(88)90001-1.

[19] J. H. Prosser, M. Kam, and H. G. Kwatny. Decision fusion and supervisor synthesis in decentralized discrete-event systems. In *Proceedings of the American Control Conference*, volume 4, pages 2251–2255. IEEE, 1997. doi:10.1109/ACC.1997.608978.

[20] P. J. Ramadge and W. M. Wonham. Supervisory Control of a Class of Discrete Event Processes. *SIAM Journal on Control and Optimization*, 25(1):206–230, 1987. doi:10.1137/0325013.

[21] R. Reiter. A theory of diagnosis from first principles. *Artificial Intelligence*, 32(1):57–95, 1987. doi:10.1016/0004-3702(87)90062-2.

[22] K. Rudie and W. M. Wonham. Think globally, act locally: decentralized supervisory control. *IEEE Transactions on Automatic Control*, 37(11):1692–1708, 1992. doi:10.1109/9.173140.

[23] F. B. Schneider. Enforceable security policies. *ACM Transactions on Information and System Security*, 3(1):30–50, 2000. doi:10.1145/353323.353382.

[24] A. Shih, A. Choi, and A. Darwiche. A Symbolic Approach to Explaining Bayesian Network Classifiers. In *Proceedings of IJCAI*, pages 5103–5111, 2018. doi:10.24963/ijcai.2018/708.

[25] S. Takai, T. Ushio, and S. Kodama. Supervisory Control of Discrete Event Systems under Partial Observations of Events and States. *Transactions of the Institute of Systems, Control and Information Engineers*, 8(5):222–229, 1995. doi:10.5687/iscie.8.222.

[26] S. Takai. Supervisory Control of Partially Observed Systems with Arbitrary Control Patterns. *Transactions of the Institute of Systems, Control and Information Engineers*, 12(8):498–504, 1999. doi:10.5687/iscie.12.498.

[27] S. Takai. Supervisory control of partially observed discrete event systems with arbitrary control patterns. *International Journal of Systems Science*, 31(5):649–656, 2000. doi:10.1080/002077200290957.

[28] Y. Tong and K. Cai. Supervisory Control with Event Forcing Under Partial Observation. arXiv preprint arXiv:2607.21040, 2026. doi:10.48550/arXiv.2607.21040.

[29] H. Wang, C. M. Poskitt, and J. Sun. AgentSpec: Customizable Runtime Enforcement for Safe and Reliable LLM Agents. arXiv preprint arXiv:2503.18666, 2025. doi:10.48550/arXiv.2503.18666.

[30] H. Wang, C. M. Poskitt, J. Sun, and J. Wei. Pro2Guard: Proactive Runtime Enforcement of LLM Agent Safety via Probabilistic Model Checking. arXiv preprint arXiv:2508.00500v1, 2025. arXiv:2508.00500v1.

[31] G. Besanson. SARC: A Governance-by-Architecture Framework for Agentic AI Systems. arXiv preprint arXiv:2605.07728, 2026. doi:10.48550/arXiv.2605.07728.

[32] A. Kaul, Q. Lan, and P. Gupta. AgentBound: Verifiable Behavioral Governance for Autonomous AI Agents. arXiv preprint arXiv:2606.30970v1, 2026. arXiv:2606.30970v1.

[33] J. Rhodes and G. Kang. Proof of Execution: Runtime Verification for Governed AI Agent Actions. arXiv preprint arXiv:2607.05397, 2026. doi:10.48550/arXiv.2607.05397.

[34] Z. Wang. Proof-Carrying Agent Actions: Model-Agnostic Runtime Governance for Heterogeneous Agent Systems. arXiv preprint arXiv:2606.04104, 2026. doi:10.48550/arXiv.2606.04104.

[35] T.-S. Yoo and S. Lafortune. Decentralized supervisory control: a new architecture with a dynamic decision fusion rule. In *Proceedings of the 6th International Workshop on Discrete Event Systems (WODES 2002)*, pages 11–17. IEEE, 2002. doi:10.1109/WODES.2002.1167663.

[36] T.-S. Yoo and S. Lafortune. Decentralized Supervisory Control With Conditional Decisions: Supervisor Existence. *IEEE Transactions on Automatic Control*, 49(11):1886–1904, 2004. doi:10.1109/TAC.2004.837595.

[37] T.-S. Yoo and S. Lafortune. A General Architecture for Decentralized Supervisory Control of Discrete-Event Systems. *Discrete Event Dynamic Systems*, 12(3):335–377, 2002. doi:10.1023/A:1015625600613.