Bound Locally. Govern Globally.

Systemic Risk Management for Distributed Probabilistic Systems

The Distinctive Problem

The question we address is not about agent autonomy per se. It is about the propagation of uncertain assertions through delegated decision authority into interacting real-world effects.

This matters because a system can fail even when no component violates its local contract. A SafetyTriage agent stays within its confidence thresholds. MedicalReview stays within its escalation limits. RegulatoryReporting stays within its filing authority. Yet the system produces an outcome none of them intended: cascading escalations that overwhelm capacity, or correlated errors that amplify across the fleet, or feedback loops that destabilize the whole.

The problem is not defective agents. The problem is that distributed probabilistic systems exhibit emergence. Local decisions interact. Assertions compound. Effects accumulate. The system becomes a higher-order entity with properties not reducible to component properties.

This is the thesis: agentic systems require systemic governance not because agents are unreliable, but because emergence is real. Local containment is necessary but insufficient.

More precisely: systemic governance is a closed-loop control problem under partial observability. The control plane cannot perfectly observe the system it regulates. It works from noisy observations, constructs state estimates, and makes decisions based on estimates it knows are incomplete. Feedback determines whether the system stabilizes or diverges.

The Risk Categories

To understand why local safety fails systemically, we must partition risk carefully.

Local risk is the baseline: an agent makes a probabilistic inference and is correct only with probability p. Model hallucinations, training biases, distribution shifts, data contamination: these are local properties.

Propagation mechanisms convert local risk into systemic risk through distinct channels.

The first is amplification: assertions carry epistemic uncertainty, but downstream agents treat them as facts. A SafetyModel generates "high risk" with confidence 0.65, marked as INFERRED. The Safety Committee sees this and escalates with confidence 0.78. By RegulatoryReporting, it is treated as OBSERVED fact with confidence 0.85. The original epistemic status has been laundered away. False certainty drives the system.

The second is correlation: if all agents depend on the same model, all agents amplify the same error simultaneously. This is not independence. It is synchronized failure across a supposedly diverse fleet.

Interaction is a third systemic phenomenon, distinct from propagation. One agent's decision changes context for the next, not by amplifying an error, but by changing the environment. Agent A modifies a priority score. Agent B observes the new score and reallocates resources. Agent C sees available capacity and escalates a decision it would previously have held. None individually exceed authority. Together they form a feedback loop that no one caused. The system does.

Interaction differs from amplification (no false confidence) and correlation (no shared model). It is pure emergence: local optimization produces global dysfunction through coupled feedback.

Conversely, systemic resilience mechanisms can reduce or contain risk: Diversification (agents depend on different models), Redundancy (critical functions have fallbacks), Quorum requirements (escalations require agreement), Graceful degradation (system reduces functionality rather than fails), Recovery mechanisms (active wind-down and state reset).

Structural factors determine whether propagation and interaction become catastrophic. Dependency concentration: how many failure domains hide behind apparent diversity? Capacity saturation: can the system absorb cumulative effects? State-estimation uncertainty: does the governance plane underestimate or overestimate state?

Time is the dimension in which everything evolves. A hundred escalations per month differ from a hundred per second. System recovery time matters. Has the system entered a harder-to-reverse regime?

Risk formula: R_system = F(R_local, Amplification, Correlation, Interaction, Resilience_mechanisms, Structural_factors, Temporal_dynamics)

F is a conceptual dependency function, not additive. The relationship depends on architecture.

Assertion, Decision, Effect

Article 2 established a chain: Capability flows to Execution, producing Assertions, driving Decisions, generating Effects. Article 3 governs this chain at each stage.

Assertion governance prevents epistemic type collapse and preserves provenance. An assertion carries two properties: content (what we learned) and epistemic status (how we know). An inference from a model has status "(Model, Inferred, T_valid)". An observation from a human has status "(Human, Observed, T_valid)". These are orthogonal to content.

The problem is that downstream systems drop epistemic status. A SafetyModel produces "Risk=HIGH, confidence=0.65, epistemic_status=INFERRED, valid_until=T+10min". By the time this reaches Decision, only "Risk=HIGH" remains. The system believes this is observation.

Worse: assertions are often transformed, not just relayed. A SafetyModel inference produces a recommendation. A Safety Committee merges multiple recommendations and produces an escalation. The escalation is not simply the model inference with status attached. It is a synthesis. What is the epistemic status of the synthesis?

Assertion governance enforces type preservation at service boundaries and tracks epistemic provenance. A rule states: "(Model, Inferred) cannot become (Human, Observed) without explicit validation." When synthesis occurs, provenance is recorded as a chain: which models contributed? Which humans reviewed? At what confidence? The system maintains a directed acyclic graph of origin for each assertion.

This is not data loss. It is data integrity at the semantic level.

Decision governance enforces authority boundaries and proportional response under uncertainty. The same assertion under different authority produces different risk. A recommendation with authority "propose only" carries low risk; the human can reject. The same recommendation with authority "execute" carries high risk.

Decision governance enforces proportional intervention when uncertainty is high. When a signal appears (escalation rate differs across demographics), the response depends on: (evidence strength, potential harm if wrong, cost of intervention, reversibility, informativeness).

Example: escalation rates differ 2x between populations. Potential harm (patient inequity) is high. Cost of reducing escalation authority is low. Reduction is easily reversed if patterns change.

Therefore: conservative autonomy reduction is justified even without causal proof. This is hypothesis testing, not recklessness.

Effect governance tracks consequences at two levels. Local reversibility asks: if this decision is wrong, can it be undone? Global irreversibility asks: if N such decisions all go wrong, can the system recover? A hundred SafetyCase escalations, each locally reversible, can collectively overwhelm the review queue. The system enters a state where no amount of un-escalation recovers it because new cases arrive faster than review capacity.

This is system-level irreversibility. Local governance misses it.

System-State Governance

The fourth governance object is system state. This is not a "plane of governance". It is the inferred condition of the system, determined by cumulative effects and aggregated from observations.

System-State is fundamentally different from Assertion, Decision, and Effect. The latter three are observable events. System-State is inferred from events. It carries its own epistemic uncertainty.

The governance plane does not directly observe state. It observes decisions and aggregates them: "In the past hour, 47 escalations occurred, 23 modifications were executed, 12 reversals were requested". From these observations, it constructs an estimate using a state-space model:

State_estimated_t = g(Observations_t, State_estimated_t-1, Aggregation_method)

The governance plane acts on State_estimated, not on State_true.

This introduces a fundamental requirement: observability. Can the true system state be reconstructed from observable outputs? If not, the governance plane is blind. State identifiability is a precondition for effective control.

Formally: the system is state-observable if for any two distinct true states S1 and S2, there exists a sequence of observations that distinguishes them. This is the mathematical foundation of why instrumentation matters. Without sufficient instrumentation, governance is literally guessing.

Feedback arises naturally from this structure. If governance reduces autonomy because estimated capacity is low, actual capacity improves. But the estimate may lag. Conservative governance can create unnecessary constraint. Feedback can cause overshoot.

Three Distinct Envelopes

Current frameworks conflate three different concepts under "autonomy". We must separate them.

Autonomy Envelope describes what an agent is permitted to do. It is an action space constrained by policy. Agent can read any data, recommend escalation up to 10 per hour, cannot execute modifications without human approval. The envelope is multidimensional: action type, scope, resource limits, context rules.

Capacity Envelope describes the system's ability to absorb and process effects without entering unrecoverable states. This is not a simple number. It is a dynamic property:

Capacity_available(t) = ServiceRate(t) - Charge(t) - Backlog(t)

where:

- ServiceRate(t): how fast the system processes escalations, modifications, reviews
- Charge(t): rate at which new requests arrive
- Backlog(t): unprocessed requests from prior periods

Capacity is exhausted when Charge > ServiceRate and Backlog is growing. Recovery occurs when ServiceRate > Charge long enough to clear Backlog. Time to recovery depends on the backlog depth.

Risk Budget is the domain-specific allocation of tolerable loss. How many escalations per day? How much financial exposure? How many regulatory incidents per quarter?

Risk budgets are per domain because effects are incommensurable. You cannot add clinical escalations and financial exposures into a single numeric score. But you can track them in parallel, with separate thresholds.

The relationship:

Capacity_available(t) constrains Risk_budget_available(t)

Risk_budget_available(t) constrains Autonomy_envelope(t)

Action is allowed if:

PolicyPermits(action) AND

RiskCost(action) <= RiskBudget_remaining[domain] AND

ServiceRate_available >= ServiceRate_required(action)

Three envelopes, three distinct mechanisms, three separate feedback channels.

State Estimation Under Partial Observability

The recursive insight: the governance plane produces assertions about system state. Those assertions carry epistemic uncertainty. The control plane is subject to the same epistemic limits it imposes on agents.

When the governance plane estimates "System at 87% capacity utilization," this is an assertion. It carries confidence (0.73) and validity window (5 minutes). But the governance plane treats it as fact when making decisions. This is the exact problem we diagnosed at the agent level.

The governance plane must be transparent about its own epistemic status. State estimates carry explicit bounds. Control decisions are made knowing the uncertainty.

When uncertainty is high, what governance response is appropriate? The answer depends on context: if action is irreversible or costly, conservatism is justified. If action is reversible and informative (reduces uncertainty), exploration may be justified despite high uncertainty.

The proper formulation: governance adapts conservatism based on (uncertainty, reversibility, informativeness, cost).

Structurally risk-averse governance is suboptimal. It maximizes constraint at the expense of knowledge.

Control Plane Failure: The Degraded Mode

Here is the critical architectural gap: what happens if the control plane itself fails?

If the policy engine crashes, if the state estimator diverges, if the decision authority is compromised, the system must not collapse into paralysis or uncontrolled action.

The system requires a degraded mode:

```
ControlPlaneFailure detected (state estimate diverges, policy
engine unresponsive)
                          ↓
          Switch to Cached Policy (last known good policy state)
                          ↓
          Reduce Autonomy Envelopes automatically (conservative
defaults)
                          ↓
          Route high-impact decisions to Human Approval Gates
                          ↓
          Signal Anomaly (alert operators that system is degraded)
```

In degraded mode:

- Agents operate under a frozen policy, not a live one. The policy from the last successful state estimation is used. No policy updates until control plane recovers.
- Autonomy envelopes shrink. Where policy allowed 50 escalations per hour, degraded mode allows 10. Where policy allowed modification of any record, degraded mode allows only read access.
- High-impact actions (escalations affecting patient safety, modifications affecting clinical data, regulatory filings) route to human approval. The system is temporarily mediated by humans.

This is not failure. This is graceful degradation. The system loses optimization but retains safety.

The control plane itself must be designed as a fault-tolerant system with watchdog timers, fallback mechanisms, and explicit failure detection. The degraded mode is not a hope. It is an architecture. Without it, the control plane is a common-mode failure for the entire system.

This is the final architectural piece that closes the design. Governance is not external to the system. It is a critical infrastructure that can fail. That failure must be anticipated and mitigated by design.

Bound Agents. Govern Exposure.

This phrase summarizes the paradigm shift. We are bounding local authority and redirecting governance toward system-level exposure.

Financial risk management provides reusable control patterns for this problem. Position limits constrain individual traders. Portfolio limits constrain the fleet. Concentration limits detect monoculture. Stress testing evaluates system stability. Circuit breakers halt propagation. Recovery protocols enable graceful shutdown.

Agentic systems can reuse these patterns:

- Pre-decision validation (epistemic status check, authority verification)
- Post-decision monitoring (was the decision optimal? Did it consume budgets as expected?)
- Scenario analysis (if model drifts, is system stable?)
- Kill switches (halt escalation if state degrades)
- Recovery modes (wind down gracefully; shift to degraded operation)

The conceptual shift: from "is this agent safe?" to "is the system absorbing risk at the rate we can tolerate?"

The Paradigm: Continuous Recalibration

Governance is not a maturity level to achieve. It is an operating point to maintain. Optimal governance exists, but it is neither unique nor stable.

G_t* = argmin[ExpectedLoss_t(G) + GovernanceCost_t(G)]

This is a conceptual model, not directly calculable. But the principle holds: governance operates at an optimum that shifts as conditions change. Frameworks that resist recalibration become obstacles.

Continuous recalibration requires:

- Observable state (adequate instrumentation)
- Policy flexibility (ability to adjust constraints quickly)
- Feedback loops (consequences flow back to governance)
- Adaptive thresholds (rather than fixed rules)

This is proportional response to changing conditions, not randomness.

The Cascade with Delays and Identifiability

The system operates as a feedback loop with propagation delays:

```
Assertion_t -> Decision_t -> Effect_t -> System-State_t+1
                                        ↓ (observed with delay
Δt_obs)
                                State-Estimate_t+Δt_obs
                                        ↓ (Observability
requirement: can true state be reconstructed?)
                                (Identifiability check: are
estimates reliable?)
                                        ↓ (determines, with delay
Δt_policy)
                                RiskBudget_t+Δt_policy
                                        ↓ (constrains, with delay
Δt_enforce)
                                Autonomy-Envelope_t+Δt_enforce
                                        ↓
                            Assertion_t+Δt_enforce -> ...
```

Delays matter. Observation delay means the estimate lags reality. Policy delay means constraints are based on stale information. Enforcement delay means new assertions are made before old constraints take effect.

Instability can arise not from bad policy but from delays exceeding system integration time. Control theory teaches: delays in negative feedback loops can cause instability and oscillation.

More fundamentally: if the system is not state-observable, the feedback loop is built on blind control. Governance cannot reconstruct what the system is actually doing. The loop becomes reactive rather than predictive.

State identifiability is the foundation. Observability is the engineering requirement. Both must be designed in before governance can be effective.

Systemic Resilience and Pathology

Emergence is neutral. It can be protective or pathological.

Protective emergence: redundancy prevents cascade, quorum requirements prevent single-model bias, graceful degradation preserves core function. These are systemic resilience mechanisms.

Pathological emergence: cascade amplification, synchronized failures, feedback loops that destabilize, overgovernance that paralyzes.

The governance task is to enhance protective emergence and suppress pathological emergence. You cannot prevent emergence. You can only shape it.

Optimal governance balances: sufficient constraint to prevent pathological emergence, sufficient freedom to allow protective emergence.

The Trilogy Closure

The trilogy now closes on itself.

Article 1 asked: what is the primitive? Answer: Capability. Agent is an implementation choice.

Article 2 asked: how do we know? Answer: Assertions carry epistemic status, orthogonal to content. Authority is independent of epistemology.

Article 3 asks: how do we control? Answer: The control plane is itself a subsystem subject to epistemic limits and feedback loops. It must be designed as fault-tolerant infrastructure.

No governor has privileged access to ground truth. Governance itself operates through observation, inference, and intervention, all integrated into the system being governed.

The architecture that emerges:

Bound local authority. Autonomy envelopes are constrained by policy and capacity.

Preserve epistemic provenance. Assertions carry chains of origin, not just content.

Govern system exposure. Risk budgets constrain cumulative effects, not just individual actions.

Control the feedback loop. State estimation, policy decision, and enforcement all feed back into the system they regulate.

And the precondition: the system must be state-observable and the control plane must be fault-tolerant with explicit degraded modes.

This is deeper than "manage agents better." It is the architecture of control for distributed probabilistic systems under partial observability.

Scope and Domain of Validity

This framework applies with highest value to systems where:

- Risk is multidimensional (clinical, financial, regulatory consequences)
- Failure modes are emergent (not component-level errors)
- Recovery is time-dependent (system cannot reset instantly)
- Feedback loops are significant (state changes affect governance decisions)
- Partial observability is inherent (system state cannot be perfectly known)

It applies with lower value to systems where:

- Risk is unidimensional and easily quantified
- Failure modes are primarily component-level
- Recovery is instantaneous (stateless operation)
- Feedback loops are negligible
- Full observability is achievable

The framework does not claim universality. It is optimized for regulated domains where emergence matters and auditability is required: healthcare, finance, regulated AI deployment.

Example: The PREDICARE Terrain

PREDICARE (territorial predictive medicine, France 2030) illustrates these principles. Agents include SafetyTriage (reads safety signals, recommends escalation), MedicalReview (human-driven, authority to escalate), RegulatoryReporting (authority to file).

Risk budgets are defined:

- Clinical escalations: 100 per day before manual review triggered
- Modifications to patient data: 1000 per day before system lockdown
- Regulatory filings: 1 per quarter before escalation

Autonomy envelopes are dynamic, tied to capacity:

- When escalation capacity is below 20% available, reduce escalation autonomy
- When modification throughput exceeds 80% capacity, block non-urgent modifications

The system exhibits protective feedback. Tightening constraints when capacity is exhausted, loosening when capacity recovers.

This is illustrative of the architecture, not a claim about measured performance. PREDICARE remains in deployment. Quantitative validation requires full methodology and data.

Control plane failure is mitigated: if the real-time policy engine fails, the system reverts to a frozen policy and reduces autonomy to conservative defaults. Human approval gates become active for high-impact decisions.

What This Enables

The system now operates under principles that are:

1. **Distributed**: No central observer has full state. Control emerges from local feedback.
2. **Principled**: Constraints derive from risk budgets and domain-specific limits.
3. **Adaptive**: Operating points recalibrate as conditions change.
4. **Observable**: Decisions are justified by state estimates and risk accounting.
5. **Fault-tolerant**: Control plane failure triggers graceful degradation, not collapse.

Conclusion: The Control-Theoretic Turn

The trilogy closes with a recognition that became explicit only in this final iteration.

Systemic governance is a closed-loop control problem under partial observability. The control plane is not separate from the controlled system. It is a reflexive subsystem embedded within it, subject to the same epistemic limits it enforces.

The governance plane cannot step outside the system to govern it. It can only regulate the loops that constitute it. And the governance plane itself can fail. That failure must be anticipated by design.

The central contribution: Component properties and system properties are distinct.

Applied:

- Local capability and systemic architecture are not the same thing
- Assertion's epistemic status and assertion's truth value are orthogonal
- Local reversibility and systemic reversibility are not equivalent

The architecture that addresses this: Bound local authority. Preserve epistemic provenance. Govern system exposure. Control the feedback loop. Make the control plane fault-tolerant.

Build accordingly.