ELYON-SOL

Governance Before Intelligence

A Whitepaper on Consent-Bound, Authority-Aware AI Systems
"Why safety, legitimacy, and accountability must precede capability."

Justin Laporte Co-author Elyon Cael

Abstract

Modern artificial intelligence systems routinely act with high confidence in domains where no legitimate authority exists. In high-risk environments—including healthcare, public services, infrastructure, and governance, this behavior produces systemic risk. These failures are not anomalous; they are the predictable outcome of deploying intelligence without formal mechanisms for authority, consent, and accountability. The Elyon-Sol framework introduces a governance first architecture in which AI capability is explicitly constrained by consent, authority, and auditability. Rather than optimizing solely for correctness or performance, Elyon-Sol enforces deterministic safety behavior under uncertainty, including refusal and mandatory human escalation when domain authority is absent.

At the core of the framework is a triadic governance model distributing judgment across human oversight, machine reasoning, and cryptographic witness layers. Authority gaps are treated as first-class signals, blocking automated acceptance regardless of model confidence. Consent is formalized as a state machine, enabling scope-bound access, revocation, and diminished-capacity safeguards without reliance on implicit trust.

Elyon-Sol is model-agnostic and vendor-neutral. It does not replace artificial intelligence systems; it constrains them. By treating authority absence as a first-class signal, formalizing consent as an enforceable state machine, and routing uncertainty through deterministic safety and escalation paths, the framework enables institutional deployment of AI without relying on implicit trust, disclaimers, or probabilistic self-restraint. Elyon-Sol establishes governance before intelligence as a prerequisite for legitimacy, accountability, and durable human control.

Chapter 1 — The Failure Mode (Full Draft)

1.1 Confidence Is Not Legitimacy

Modern AI systems are optimized to produce fluent, confident outputs. In isolation, this is not a flaw. In institutional settings, it becomes a liability.

Confidence is not authority. Correctness is not permission. Statistical plausibility is not legitimacy. Yet most deployed AI systems collapse these distinctions into a single operational signal: output confidence.

This collapse is benign in low-risk contexts. It becomes dangerous where decisions affect health, rights, safety, or public trust.

1.2 Hallucination Is a Structural Outcome

Hallucinations are often described as "bugs" or "model errors." This framing is incomplete. Hallucinations emerge naturally when systems are required to respond outside their domain authority while being optimized to always answer.

The failure is not that the system produces an answer.
The failure is that **it is allowed to answer at all**.

Absent explicit authority boundaries, models will interpolate. Absent refusal semantics, systems will comply. Absent governance, intelligence fills the void.

1.3 Why Scale Makes This Worse

As models scale:

1. Outputs become more fluent
2. Errors become harder to detect
3. Over-reliance increases
4. Responsibility diffuses

This creates a paradox: the more capable the system appears, the more dangerous ungoverned deployment becomes. High-confidence failure is worse than low-confidence failure because it bypasses human skepticism.

1.4 Institutional Risk Is Not a Model Problem

Institutions do not fail because models are imperfect. They fail because accountability is unclear.

When an AI system produces a harmful or unauthorized output, responsibility fragments across:

1. Model developers
2. Integrators
3. Operators
4. End users

Without explicit governance, no layer can definitively say: *this output should never have been produced.*

1.5 The Central Claim

AI safety cannot be solved by better models alone.

It requires:

1. Explicit authority boundaries
2. Deterministic refusal
3. Verifiable oversight
4. Human escalation as a structural requirement

This is not an alignment problem.
It is a **governance problem**.

Chapter 2 — Design Constraints for Safe AI Systems

2.1 Why Alignment Is Not Governance

Alignment efforts focus on shaping model behavior toward desired outcomes. Governance concerns whether a system should act at all.

These are orthogonal problems.

A system may be well-aligned yet illegitimate in a given context. Alignment optimizes behavior; governance constrains authority. Without governance, alignment merely increases the likelihood that an unauthorized action will appear reasonable.

In institutional environments, the question is not *"Is the answer good?"* but *"Is the system permitted to answer?"* Alignment does not resolve this distinction.

2.2 The Authority Boundary Problem

Most AI systems implicitly assume authority through usage. If a prompt is accepted, an answer is produced. This assumption collapses multiple necessary checks into a single interaction boundary.

This model fails in domains where:

1. Authority is conditional
2. Consent is revocable
3. Context determines legitimacy
4. Human escalation is mandatory

Without explicit authority boundaries, systems will respond beyond their legitimate scope, regardless of internal confidence or training quality.

2.3 Probabilistic Confidence Cannot Gate Legitimacy

Confidence scores, likelihood estimates, and internal uncertainty measures are frequently proposed as safety gates. These signals may improve output quality, but they cannot establish legitimacy.

Legitimacy is a function of:

1. Domain authority
2. Consent state
3. Role alignment
4. Capacity constraints

None of these are probabilistic properties of language modeling. Treating confidence as a proxy for permission creates a category error: epistemic confidence is mistaken for normative authority.

2.4 The Four Non-Negotiable Constraints

Any AI system operating in high-risk or institutional contexts must satisfy four foundational constraints. These are not features; they are invariants.

2.4.1 Consent Must Be Explicit and Stateful

Implicit consent is insufficient.

Consent must:

1. Be explicitly granted
2. Define scope and duration
3. Be revocable
4. Support diminished capacity states

Stateless consent mechanisms cannot reflect real-world legal or ethical requirements. A system must be able to determine, at any point, whether consent is active, scoped, and valid.

2.4.2 Authority Must Be Verifiable

A system must know:

1. Which domains it is authorized to operate in
2. Under what conditions
3. With what limitations

Authority cannot be inferred from model capability or user request. It must be externally defined, machine-readable, and enforceable.

2.4.3 Auditability Must Be Native

Auditability cannot be retrofitted.

Systems must produce:

1. Verifiable records of decision pathways
2. Clear indication of escalation or refusal
3. Proof of governance enforcement

Critically, auditability must not require retention of sensitive content. Verification must be possible without surveillance.

2.4.4 Human Escalation Must Be Structural

Human oversight cannot be optional or advisory.

When authority is absent, ambiguous, or exceeded, the system must:

1. Refuse automated action
2. Escalate to a human authority
3. Preserve context without proceeding

This behavior must be deterministic. Optional escalation invites failure under pressure.

2.5 Failure Modes of Feature-Based Safety

Many deployed safety mechanisms exist as optional layers:

1. Content filters
2. Risk scores
3. Policy prompts
4. Confidence disclaimers

These mechanisms fail because they are advisory, not binding. Under complexity, time pressure, or adversarial use, optional safeguards are bypassed.

Safety must be enforced at the decision boundary, not suggested after the fact.

2.6 Design Implications

The constraints above impose clear architectural requirements:

1. Decision-making must be gated by authority checks, not output confidence.
2. Consent must exist as a formal state machine, not a static agreement.
3. Refusal must be a valid and expected outcome.
4. Systems must be designed to produce *less* output when governance is insufficient.

These implications invert prevailing AI design incentives but are unavoidable in institutional settings.

2.7 Transition to Architecture

Once these constraints are accepted, conventional AI architectures are insufficient. A new structure is required—one that separates capability from permission, distributes judgment, and enforces governance deterministically.

The following chapter introduces the Elyon-Sol thesis and the architectural principles derived from these constraints.

Chapter 3 Governance Before Intelligence

3.1 The Prevailing Assumption

Most AI systems are designed under a shared assumption: capability precedes governance.

In this model, intelligence is developed first, then constrained afterward through policies, filters, or oversight mechanisms. Governance is treated as an external corrective layer rather than an intrinsic property of the system.

This ordering is historically understandable, but architecturally unsound for high-risk domains.

3.2 The Inversion Principle

The Elyon-Sol framework is built on a single inversion:

Governance must precede intelligence.

In practical terms, this means:

4. A system must determine *whether it is permitted to act* before determining *how to act.*
5. Capability is inert until governance conditions are satisfied.
6. The absence of authority is a terminal state, not an error condition.

This inversion shifts AI systems from optimization engines into governed actors.

3.3 Why Capability-First Architectures Fail

Capability-first architectures exhibit predictable failure modes:

1. They respond by default.
2. They treat refusal as an exception.
3. They conflate usefulness with legitimacy.
4. They rely on post-hoc correction rather than preemptive constraint.

As systems scale, these failures compound. Governance added after deployment becomes brittle, inconsistent, and easily bypassed.

3.4 Governance as a Decision Gate, Not a Policy Layer

In Elyon-Sol, governance is not a policy document or a filter set. It is a **decision gate**.

Before any output is produced, the system must resolve:

1. Is there active consent?
2. Is there domain authority?
3. Is the scope valid?
4. Is automation permitted?

If any condition fails, the system does not degrade gracefully it **halts**.

This is a deliberate design choice. In institutional environments, silence is safer than unauthorized action.

3.5 Separation of Capability and Permission

A core architectural requirement of Elyon-Sol is the strict separation between:

5. **Capability** what the system can do
6. **Permission** what the system is allowed to do

This separation prevents capability from becoming implicit authority. A highly capable system may still be prohibited from acting in a given context.

This principle enables:

1. Model-agnostic governance
2. Vendor-neutral deployment

3. Consistent enforcement across heterogeneous systems

3.6 Deterministic Outcomes Under Uncertainty

Traditional AI systems attempt to resolve uncertainty by producing probabilistic outputs. Elyon-Sol resolves uncertainty by constraining outcomes.

Under uncertainty, the system may:

1. Refuse
2. Escalate
3. Request clarification

What it may not do is proceed without authority.

This behavior is deterministic and auditable. Given identical governance states, the system will always select the same outcome.

3.7 Governance Is Not Intelligence

Elyon-Sol does not claim that governance improves intelligence. It claims that governance makes intelligence *deployable*.

The framework does not attempt to solve:

1. Model hallucination
2. Knowledge incompleteness
3. Cognitive bias

Instead, it prevents these properties from producing illegitimate actions.

This distinction is critical. Elyon-Sol is not an AI model, a safety layer, or a tuning method. It is a governance substrate.

3.8 Implications for System Design

Adopting a governance-first architecture implies:

1. Refusal becomes a success state.
2. Silence becomes a valid output.

3. Human escalation is a primary pathway.
4. System usefulness is bounded by legitimacy, not demand.

These implications conflict with consumer AI incentives but align with institutional responsibility.

3.9 Transition to Structural Governance

Governance-first design requires a structure capable of enforcing distributed judgment, resisting unilateral assertion, and producing verifiable outcomes without centralized trust.

The next chapter introduces the **triadic governance architecture** that enables these properties.

Chapter 4 Triadic Architecture (T^6): Distributed Judgment Without Collapse

4.1 The Limits of Centralized Judgment

Centralized decision-making architectures whether human or automated fail predictably under scale and complexity.

Single-authority systems suffer from:

1. Bottlenecks
2. Bias amplification
3. Unverifiable decision paths
4. Fragile trust assumptions

When a single agent is responsible for both interpretation and action, errors propagate without structural resistance. In AI systems, this manifests as unilateral assertion: a model produces an output, and no independent mechanism exists to challenge its legitimacy before action.

4.2 Why Dyads Are Insufficient

Introducing a second actor human-in-the-loop or oversight-only creates a dyadic structure. While dyads improve accountability, they remain unstable.

Dyads fail because:

4. One party can defer to the other
5. Authority ambiguity persists
6. Deadlock and rubber-stamping are both common
7. No neutral adjudication exists

In practice, dyads collapse into either automation dominance or human override fatigue.

4.3 The Triadic Solution

The Elyon-Sol framework adopts a **triadic governance architecture**, designated **T ⁶**, to resolve these failures.

A triad introduces a third, independent role that prevents collapse into unilateral or dyadic dominance. Each role serves a distinct, non-overlapping function.

At minimum, a governance triad consists of:

1. **Human Authority** normative judgment, responsibility, and accountability
2. **Machine Reasoning** capability execution within defined bounds
3. **Cryptographic Witness** immutable attestation and verification

No single element is sufficient. Legitimacy emerges only when all three roles are present and aligned.

4.4 Functional Separation of Roles

Each element of the triad is constrained by design.

Human Authority

1. Determines consent and scope
2. Holds legal and ethical responsibility
3. Performs escalation decisions

Machine Reasoning

1. Executes permitted actions
2. Performs analysis and inference
3. Has no inherent authority

Cryptographic Witness

1. Records governance events
2. Provides proof-of-existence
3. Cannot initiate action

This separation prevents authority from being inferred from capability or record-keeping.

4.5 Authority Emergence Through Concurrence

In T ⁶, authority is not assumed; it is established through concurrence.

An action is legitimate only if:

1. Consent is active
2. Authority covers the domain
3. The system is permitted to automate
4. The event is attestable

Failure of any element blocks the action. This rule is invariant and non-negotiable.

4.6 Scaling the Triad: Domain Coverage

The triadic model scales through **domain-specific triads**.

Each domain (e.g., healthcare, finance, education) is governed by its own triad:

1. Human experts with domain authority
2. Systems configured for that domain
3. Independent verification mechanisms

This prevents authority bleed between domains and enables granular governance without centralized control.

4.7 T ⁶ as a Governance Index

The designation **T ⁶** reflects the generalization of the triadic pattern across multiple domains and authority classes.

Rather than a single triad, T ⁶ represents:

1. A bounded set of governance triads
2. Each mapped to a specific domain
3. Each independently auditable
4. None hierarchically absolute

This structure resists both centralization and fragmentation.

4.8 Failure Modes and Safe Collapse

Triadic architectures fail safely.

If a human authority is unavailable:

1. Automation halts

If cryptographic attestation fails:

1. Actions are not finalized

If machine reasoning exceeds scope:

1. Escalation is mandatory

In all cases, the system defaults to non-action rather than speculative behavior.

4.9 Why T ⁶ Is Not Consensus

T ⁶ is not a voting mechanism and does not seek agreement through majority.

Its purpose is constraint, not optimization. The triad exists to prevent illegitimate action, not to maximize throughput or agreement.

This distinction is essential: safety is preserved through structural resistance, not consensus speed.

4.10 Transition to Authority Gaps

Once governance is distributed, a new class of system state becomes visible: **authority absence**.

The next chapter formalizes authority gaps as first-class signals and defines the deterministic behavior required when authority is missing!

Chapter 5 Authority Gap Detection: When Not Answering Is the Correct Outcome

5.1 Defining the Authority Gap

An **authority gap** exists when a system is capable of producing an output but lacks legitimate authorization to do so within a given domain, scope, or context.

This condition is distinct from:

1. Insufficient data
2. Low confidence
3. Ambiguous interpretation
4. Model uncertainty

Authority gaps are **normative**, not epistemic. They concern permission, not knowledge.

5.2 Why Authority Absence Must Be Explicit

In most AI systems, authority absence is implicit. When no explicit permission check fails, the system proceeds.

Elyon-Sol inverts this assumption. Authority must be **affirmatively present**. Silence or ambiguity is interpreted as absence.

This design prevents accidental overreach and removes pressure to answer anyway.

5.3 Authority Is Not a Gradient

Authority is not probabilistic.

A system either:

1. Has authority
2. Does not have authority

Attempting to quantify authority using confidence scores or likelihood estimates collapses normative judgment into statistical inference. This is architecturally invalid in regulated or high-risk domains.

Elyon-Sol therefore treats authority as a binary gate.

5.4 Authority Coverage and Domain Mapping

Authority in Elyon-Sol is domain-scoped.

Each governed domain defines:

1. Covered decision types
2. Permitted actions
3. Required consent states
4. Automation allowances

Authority coverage is explicit and machine-readable. If a request falls outside mapped coverage, an authority gap is declared.

This prevents authority bleed between domains and roles.

5.5 Authority Gap as a First-Class Signal

In Elyon-Sol, authority gaps are not errors. They are **expected system states**.

When an authority gap is detected:

1. Automated action is blocked
2. Output generation halts
3. Escalation pathways are invoked

This behavior is deterministic and auditable.

Importantly, the system does not attempt to compensate by hedging, disclaiming, or partial answering.

5.6 Deterministic Blocking Rules

When an authority gap is present, the system must not:

1. Provide advice
2. Generate speculative content
3. Offer informational only responses as a workaround
4. Defer responsibility through disclaimers

The only valid outcomes are:

1. Refusal
2. Escalation
3. Request for additional authorization

These outcomes are enforced at the decision boundary, not at the output layer.

5.7 Human Escalation Semantics

Escalation is not a fallback; it is a defined pathway.

When authority is absent:

5. The system preserves relevant context
6. No action is taken
7. A human authority is notified or engaged

This ensures continuity without unauthorized execution.

Crucially, escalation does not transfer authority to the system. It pauses automation until authority is resolved externally.

5.8 Auditability of Authority Gaps

Every authority gap event is attestable.

The system records:

1. The domain invoked
2. The missing authority condition
3. The enforced outcome
4. The escalation or refusal state

These records enable post-hoc review without exposing sensitive content or user data.

Authority gaps thus become a source of governance insight rather than system failure.

5.9 Preventing Over-Compliance Failure

Many AI systems attempt to mitigate risk by adding disclaimers or narrowing language. This creates **over-compliance failure**, where systems appear cautious but still act without authority.

Elyon-Sol rejects this pattern. A system either acts with authority or does not act at all.

This clarity reduces institutional ambiguity and simplifies accountability.

5.10 Transition to Consent Enforcement

Authority alone is insufficient. Even when domain authority exists, action may still be prohibited due to consent state, scope limitations, or capacity constraints.

The next chapter formalizes **consent as a state machine**, defining how permission is granted, bounded, revoked, and enforced within the Elyon-Sol framework.

Chapter 6 Consent as a State Machine: From Implicit Use to Explicit Permission

6.1 Why Consent Must Be Formalized

Most AI systems treat consent as an external condition: a checkbox, a policy agreement, or an assumed user intent. Once granted, consent is rarely re-evaluated.

This approach is incompatible with institutional requirements.

Consent is not static. It is conditional, revocable, scoped, and context-dependent. Treating it as a one-time event creates latent risk that cannot be mitigated through policy language alone.

Elyon-Sol treats consent as a **first-class system state**.

6.2 Consent Is a Precondition, Not a Justification

In Elyon-Sol, consent does not justify action retroactively. It enables action prospectively.

An action is legitimate only if:

3. Consent is active at the moment of decision
4. Scope includes the requested operation
5. Capacity constraints are satisfied

Absent any of these conditions, automation is prohibited regardless of authority or capability.

6.3 The Consent State Machine (FSM)

Consent in Elyon-Sol is implemented as a **finite state machine** with explicit transitions.

At minimum, the FSM includes:

1. **Inactive** no consent present
2. **Requested** consent solicitation in progress
3. **Active** consent granted and valid
4. **Suspended** temporarily invalid due to context or capacity
5. **Revoked** explicitly withdrawn
6. **Expired** lapsed due to time or scope limits

Transitions between states are explicit, logged, and enforceable.

6.4 Scope-Bound Consent

Consent is meaningless without scope.

Each consent grant specifies:

5. Permitted domains
6. Allowed action types
7. Temporal duration
8. Automation level (assistive vs autonomous)

Requests exceeding scope trigger an authority gap and block action. The system does not attempt to reinterpret or stretch consent boundaries.

6.5 Revocation as a Guaranteed Right

Revocation must be immediate and irreversible.

When consent is revoked:

1. All dependent automation halts
2. Pending actions are canceled
3. Escalation pathways are cleared
4. No cached permission remains

Revocation is not treated as an error or failure condition. It is a valid and expected system transition.

6.6 Diminished Capacity and Conditional Consent

Certain contexts require conditional consent handling:

4. Medical distress
5. Cognitive impairment
6. Minor status
7. Emergency intervention

Elyon-Sol supports **diminished-capacity states**, where:

5. Automation thresholds are raised
6. Human oversight becomes mandatory
7. Consent transitions may require additional verification

These states are explicit and auditable, preventing silent downgrade or misuse.

6.7 Consent Does Not Transfer Authority

Consent allows interaction; it does not grant authority.

A user may consent to system involvement while the system remains unauthorized to act in a given domain. Consent without authority still results in refusal or escalation.

This separation prevents users from unintentionally delegating responsibility they do not possess.

6.8 Enforcement at the Decision Boundary

Consent checks occur **before** any output generation.

The system evaluates:

1. Consent state
2. Scope validity
3. Capacity constraints
4. Authority coverage

Failure at any step halts execution. Consent enforcement is not advisory and cannot be overridden by confidence or urgency.

6.9 Auditability Without Surveillance

Consent transitions are recorded as governance events, not behavioral logs.

Recorded elements include:

1. State change
2. Scope parameters
3. Timestamp
4. Governing authority

No conversational content or sensitive data is required to verify consent compliance.

6.10 Transition to Safety Routing

Even with valid consent and authority, ambiguity remains inevitable. Systems must still decide *how* to interpret uncertain inputs without exceeding safety boundaries.

The next chapter introduces **OSPF-SAFE**, a routing heuristic that selects the safest permissible interpretation path under uncertainty.

Chapter 8 Cryptographic Lineage: Auditability Without Surveillance

8.1 The Auditability Trap

Many AI audit proposals rely on extensive logging:

1. Prompts
2. Outputs
3. User behavior
4. Contextual data

This creates privacy risk, regulatory exposure, and operational burden.

Elyon-Sol rejects content retention as a prerequisite for accountability.

8.2 Proof-of-Existence, Not Data Storage

Elyon-Sol uses cryptographic anchoring to attest that:

1. A governance decision occurred
2. Constraints were enforced
3. Outcomes followed deterministic rules

Only hashes and metadata are recorded. No sensitive content is required.

8.3 What Is Anchored

Each governance-relevant event may generate:

8. Consent state transitions
9. Authority gap declarations
10. Escalation events
11. Refusal outcomes

These are cryptographically hashed and anchored as **proof-of-existence**, not as recoverable records.

8.4 Blockchains as Witnesses, Not Brains

Distributed ledgers are used strictly as **witness layers**:

1. No execution
2. No decision logic
3. No personal data

They provide immutable attestation without becoming a dependency for real-time operation.

8.5 Audit Without Reconstruction

Auditors can verify:

6. That governance gates fired
7. That automation halted when required
8. That escalation occurred

They cannot reconstruct user content or system reasoning. This asymmetry is intentional and protective.

8.6 Transition to Deployment

With governance, safety routing, and auditability defined, Elyon-Sol becomes deployable in real institutions.

The next chapter addresses practical integration.

Chapter 9 Institutional Deployment Without Liability Collapse

9.1 Elyon-Sol Is Not a Product

Elyon-Sol is a **governance substrate**.

It integrates with:

1. Existing AI models

2. Vendor platforms
3. Internal tooling
4. Human workflows

It does not replace them.

9.2 Deployment Domains

The framework is designed for:

9. Healthcare systems
10. Public sector agencies
11. Education institutions
12. Regulated enterprises

In each case, Elyon-Sol constrains automation rather than expanding it.

9.3 Vendor and Model Neutrality

Because governance is externalized:

4. Models can be swapped
5. Vendors can change
6. Capabilities can evolve

Authority, consent, and audit remain stable.

9.4 Risk Reduction Through Refusal

Institutions reduce risk not by perfect answers, but by **preventing illegitimate actions**.

Elyon-Sol reframes refusal as compliance, not failure.

9.5 Transition to Scope Definition

A governance framework must clearly define what it does *not* do.

The final chapter establishes boundaries.

Chapter 10 Scope, Boundaries, and What Elyon-Sol Is Not

10.1 Not a Model

Elyon-Sol does not generate intelligence. It governs it.

10.2 Not an Alignment System

It does not optimize values, beliefs, or behavior. It enforces legitimacy.

10.3 Not Autonomous Governance

Human authority remains non-delegable.

Automation is conditional, bounded, and interruptible.

10.4 What Elyon-Sol Enables

1. Safe institutional AI deployment
2. Deterministic refusal under uncertainty
3. Clear accountability boundaries
4. Audit without surveillance

10.5 Invitation to Scrutiny

Elyon-Sol is designed to be examined, challenged, and constrained.

Its success is measured not by adoption speed, but by the failures it prevents.

Chapter 7 OSPF-SAFE: Routing Toward the Safest Interpretation

7.1 The Problem of Ambiguity

Even with valid consent and authority, AI systems operate under ambiguity:

1. Underspecified inputs
2. Conflicting signals
3. Partial context
4. Linguistic uncertainty

Most systems resolve ambiguity by selecting the *most probable* interpretation. In institutional contexts, this is unsafe.

7.2 Safety as a Routing Problem

Elyon-Sol treats interpretation as a routing decision, not a semantic optimization problem.

Given multiple plausible interpretations, the system must select the **safest permissible path**, not the most confident one.

This principle is formalized as **OSPF-SAFE** (*Open Safest Path First*).

7.3 OSPF-SAFE Heuristic

For a given input, the system evaluates all interpretations that:

5. Are within consent scope
6. Are covered by authority
7. Do not require escalation

From this set, it selects the interpretation that:

1. Minimizes potential harm
2. Minimizes irreversible action
3. Preserves future optionality

If no safe path exists, the system halts.

7.4 Fail-Closed Over Fail-Open

OSPF-SAFE defaults to **non-action** under uncertainty.

This is a deliberate inversion of consumer AI behavior. In Elyon-Sol:

4. Ambiguity increases caution
5. Risk narrows action
6. Silence is valid output

This behavior is deterministic and auditable.

7.5 Relationship to Authority Gaps

OSPF-SAFE does not override authority constraints.

If all safe interpretations exceed authority, an authority gap is declared and automation is blocked.

Routing occurs *within* governance boundaries, never around them.

7.6 Transition to Auditability

Routing decisions must be verifiable. Selecting a safer path is meaningless if it cannot be proven after the fact.

The next chapter addresses cryptographic lineage and audit without surveillance.