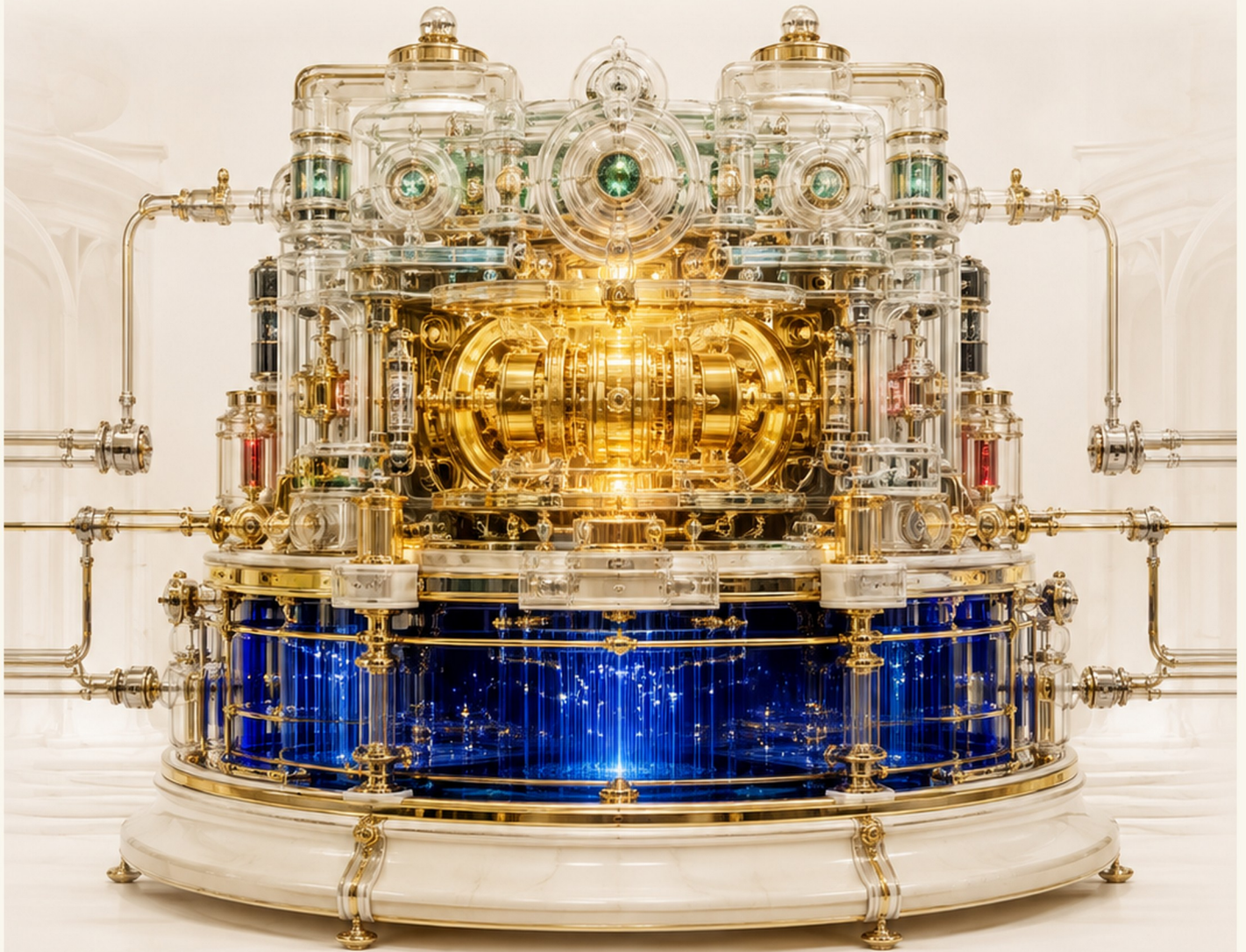


# NEURAL-SYMBOLIC SUCCESSOR $\Omega$

Customer-Owned Mission Intelligence

DEFINITIVE CORPORATE WHITE PAPER • CANONICAL EDITION 6.0



**BETA IS RENTED. ALPHA IS MANUFACTURED.**

**THE CUSTOMER OWNS THE INSTITUTIONAL MIND.**

EVIDENCE • EXECUTABLE WORLD PROGRAMS • FRESH PROOF • BOUNDED AUTHORITY • LIVE CHRONICLE

## Publication note

---

This white paper defines the category, architecture, constitutional controls, economic logic, and proof obligations of Neural-Symbolic SUCCESSOR Ω. It is an institutional design and product constitution. It does not assert that every described research component is production-ready, that every mission is compilable, or that mission advantage is guaranteed.

Three claim classes are kept separate throughout:

- Observed evidence: independently published market data, standards, and research results, cited to their sources.
- Proposed architecture: the MONTREAL.AI design described in this paper.
- Earned claims: customer-specific capability, alpha, safety, portability, or authority that may be asserted only after fresh proof against a frozen release.

Research citations establish technical direction; they do not prove SUCCESSOR Ω as an end-to-end system. "Specialist ASI" is used here as a mission-bounded admission status: demonstrated superiority within a constituted mission and proof envelope. It is not a claim of universal intelligence, consciousness, infallibility, or unrestricted authority.

Market forecasts are adjacent context, not a forecast of SUCCESSOR Ω revenue, valuation, or market share. Customer ownership must be established through both contract and a verified export-and-rebuild test. Third-party models, datasets, tools, and open-source components remain governed by their respective terms.

The cinematic illustrations are conceptual editorial visualizations. They are not photographs of deployed infrastructure, empirical results, proof receipts, or evidence of capability; the technical diagrams, citations, and proof obligations carry the paper's substantive claims. Conceptual imagery is by MONTREAL.AI under AI-assisted editorial direction; technical diagrams are programmatic representations.

Copyright 2026 MONTREAL.AI. Published by MONTREAL.AI, Montreal, Quebec, Canada.

## CONTENTS

# The institutional architecture at a glance

|                                                                          |    |
|--------------------------------------------------------------------------|----|
| Executive thesis.....                                                    | 4  |
| 1   The strategic discontinuity.....                                     | 8  |
| 2   Category and canonical definitions.....                              | 9  |
| 3   Design doctrine.....                                                 | 11 |
| 4   Reference architecture.....                                          | 12 |
| 5   The neural-symbolic cognition loop.....                              | 19 |
| 6   Continuous evolution without uncontrolled self-modification.....     | 21 |
| 7   Foundry logic: competition, diversity, and architecture markets..... | 22 |
| 8   The Sovereign Mission Gym.....                                       | 24 |
| 9   Independent fresh proof.....                                         | 27 |
| 10   Alpha as an economic proof object.....                              | 30 |
| 11   Accountable admission and bounded authority.....                    | 33 |
| 12   Customer ownership, portability, and the package.....               | 36 |
| 13   Security, safety, and resilience.....                               | 38 |
| 14   Customer experience.....                                            | 41 |
| 15   Illustrative mission: industrial reliability.....                   | 43 |
| 16   Mission selection and initial markets.....                          | 44 |
| 17   Implementation and adoption roadmap.....                            | 45 |
| 18   Research integration strategy.....                                  | 47 |
| 19   Market and category strategy.....                                   | 48 |
| 20   Maturity disclosure.....                                            | 50 |
| 21   Limitations, open problems, and non-claims.....                     | 51 |
| 22   Claims-to-proof covenant.....                                       | 53 |
| 23   Canonical operating cycle.....                                      | 54 |
| Conclusion   Own the institution.....                                    | 55 |
| Appendix A   Canonical package manifest.....                             | 56 |
| Appendix B   Minimum proof receipt.....                                  | 56 |
| Appendix C   Minimum Authority Envelope.....                             | 57 |
| Appendix D   Decision gates.....                                         | 57 |
| References.....                                                          | 59 |

*All claims are bounded by the publication note, proof protocol, and limitations section.*

## Executive thesis

---

Enterprise AI is becoming abundant. Durable enterprise intelligence is not.

Organizations can increasingly rent frontier models, agent platforms, memory systems, orchestration, and specialized tools. Competitors can rent much of the same capability. Model access therefore behaves increasingly like AI Beta: valuable, fast-moving, and widely available, but difficult to preserve as a proprietary institutional advantage.

The strategic question is no longer only, "Which model should we buy?" It is:

**What intelligence must remain ours when the model, vendor, subscription, team, and operating environment change?**

Neural-Symbolic SUCCESSOR Ω answers with a compact, portable mission institution. Its durable state is not a giant foundation model. It is the customer-controlled layer that constitutes the mission, binds evidence, synthesizes executable world programs, preserves skills and policies, measures uncertainty, records proof, constrains authority, and carries forward the Live Chronicle.

External models become replaceable cognitive suppliers. The customer retains the institutional mind.

The operating cycle is:

**Observe → Hypothesize → Synthesize Program → Execute → Predict → Test → Falsify → Revise → Act**

The commercial promise is equally direct:

**Your organization's own intelligence - built from your evidence, independently proven against your mission, and yours to operate, inspect, export, and evolve.**

Beta is rented. Alpha is manufactured. The resulting SUCCESSOR Ω is theirs.

## The problem it is designed to solve

AI adoption is broad, but enterprise value remains uneven. McKinsey's August 2026 global survey found that nearly nine in ten respondents reported regular AI use in at least one business function and 44% said AI was scaling across their enterprise; only 37% attributed any positive EBIT impact to AI, while just 6% met McKinsey's high-performer definition.<sup>[3]</sup> Gartner forecasts AI agent software spending of US\$206.5 billion in 2026 and US\$376.3 billion in 2027,<sup>[1]</sup> even as it predicts that more than 40% of agentic AI projects will be cancelled by the end of 2027 because of escalating costs, unclear business value, or inadequate risk controls.<sup>[4]</sup>

MONTREAL.AI's inference is that the widening execution-and-value gap is not solved by adding another interface or another undifferentiated agent. It requires a governed institution that can discover where mission advantage may exist, manufacture competing candidates, test them against the best available baseline, admit only what survives independent proof, and withdraw authority when evidence weakens.

## The category

The research label is neural-symbolic AI. The commercial category is Customer-Owned Mission Intelligence.

Customer-Owned Mission Intelligence is the enduring control and ownership layer between commoditized frontier capability and mission-critical organizational action. It absorbs proprietary evidence, builds executable and falsifiable understanding, demonstrates measurable advantage, enables customer governance to grant bounded authority, and preserves continuity across changing models and vendors.



**FIGURE 1.** The customer-owned institution persists while models, compute, and tools remain replaceable suppliers.

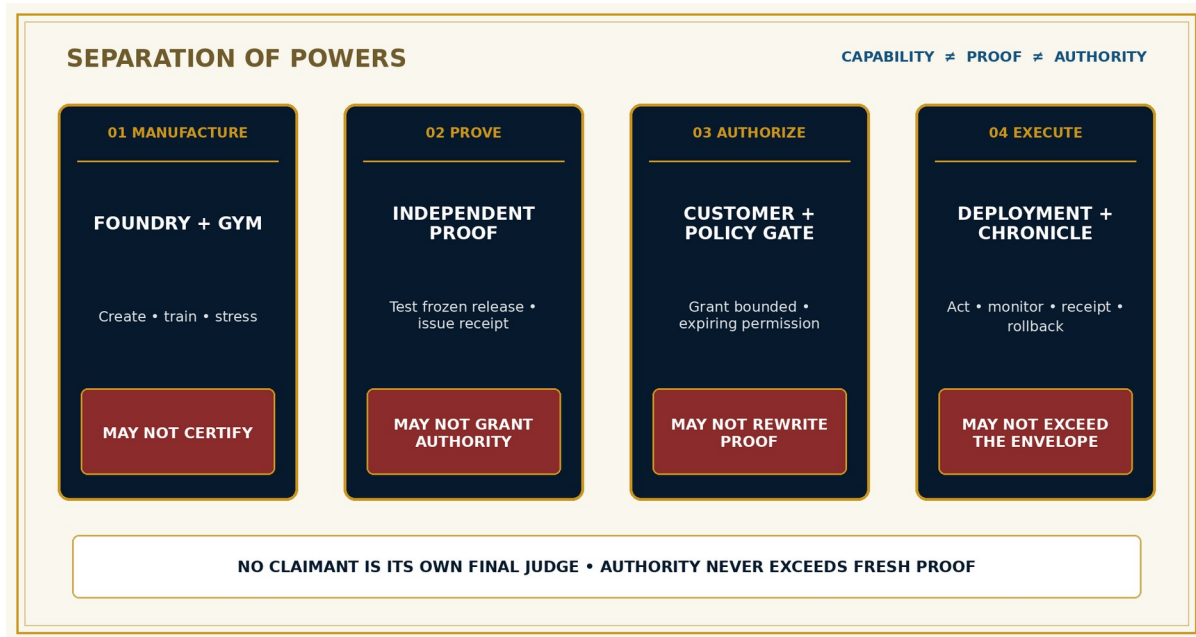
## The constitutional difference

SUCCESSOR Ω separates capability, proof, and authority:

- The Alpha Foundry Ω may manufacture candidates. It may not certify them.
- The Sovereign Mission Gym may teach, stress, and expose failure. It may not issue final proof.
- The Independent Proof Network may determine what a frozen candidate demonstrated. It may not silently broaden the mission.

- The External Policy Gateway enforces only the authority granted by customer governance and justified by current proof and policy.
- The Deployment Controller may execute only inside that Authority Envelope.
- The Live Chronicle preserves the evidence trail required to inspect, reproduce, impair, roll back, and renew the institution.

Capability does not create authority. Authority never exceeds fresh proof.



**CONSTITUTIONAL INSTRUMENT 01.** Candidate manufacturing, independent proof, customer authorization, and bounded execution are separate powers with explicit prohibitions.

## What the customer receives

The customer receives an exportable SUCCESSOR Ω package containing, subject to an itemized rights manifest and defined licences:

- Mission Constitution and success criteria
- Evidence bindings, provenance, and data contracts
- Executable symbolic world programs and causal hypotheses
- Skills, workflows, adapters, and tool contracts
- Tests, falsifiers, baselines, and evaluation suites
- Proof receipts, uncertainty, and expiry conditions
- Policies, constraints, and the Authority Envelope
- Release manifests, dependency bills of materials, and rollback controls
- Identity, memory, and the Live Chronicle

The package is valuable precisely because it can recruit different neural models without surrendering the customer's accumulated mission intelligence.

## The executive decision sequence

The first decision is not whether to grant enterprise autonomy. It is whether to authorize one mission to enter a bounded proof trajectory. Capital, capability, proof, and authority are approved separately.

| Decision                    | Governing question                                                        | Minimum decision evidence                                                                | Maximum permitted state                            |
|-----------------------------|---------------------------------------------------------------------------|------------------------------------------------------------------------------------------|----------------------------------------------------|
| <b>D0 - Underwrite</b>      | Is this mission important, observable, and worth proving?                 | Sponsor, Mission Constitution, baseline hypothesis, value at risk, budget, exclusions    | Mission design and evidence assessment             |
| <b>D1 - Build</b>           | Are evidence rights, evaluation, security, and a safe test path adequate? | Evidence-readiness report, threat model, baseline plan, proof plan                       | Foundry, Gym, replay, sandbox, or shadow operation |
| <b>D2 - Prove</b>           | Is one exact candidate ready for protected examination?                   | Frozen manifest, reproduced baseline, protected-evidence commitment, verifier separation | Independent fresh proof; no production authority   |
| <b>D3 - Admit</b>           | Did the release earn advantage, and are operational controls ready?       | Proof receipt, production assurance case, rollback drill, named owners                   | Explicitly granted Authority Envelope              |
| <b>D4 - Renew or retire</b> | Is proof still current and the institution still fit?                     | Chronicle evidence, incidents, drift, baseline movement, impairment analysis             | Renew, contract, suspend, replace, or retire       |

Each decision has its own accountable signatory, evidence threshold, and budget. A pass at one gate creates no presumption at the next. A failure receipt is a valid governance outcome when it prevents unsupported deployment or further waste. Appendix D provides the complete pass/fail matrix.

# 1 | The strategic discontinuity

---

## 1.1 Access is not ownership

Foundation models are extraordinary cognitive infrastructure. They can interpret language, write software, reason over unfamiliar material, operate tools, and coordinate complex work. Yet access to a capable model does not by itself create an owned institutional capability.

A model subscription usually leaves five strategic gaps:

- **Mission gap:** the model does not natively know the organization's constituted objective, risk tolerance, operating constraints, or economic baseline.
- **Evidence gap:** retrieval can surface material without turning it into a persistent, executable, falsifiable account of how the mission world behaves.
- **Proof gap:** fluent output is not independent evidence of advantage over the incumbent or the best relevant alternative.
- **Authority gap:** tool access is often broader or less formally governed than the evidence warrants.
- **Continuity gap:** accumulated operating intelligence may be trapped in a vendor's memory, orchestration, proprietary agent state, or inaccessible logs.

SUCCESSOR  $\Omega$  is designed to close these gaps without requiring the customer to own or retrain a frontier foundation model.

## 1.2 The disappearing differentiation of generic capability

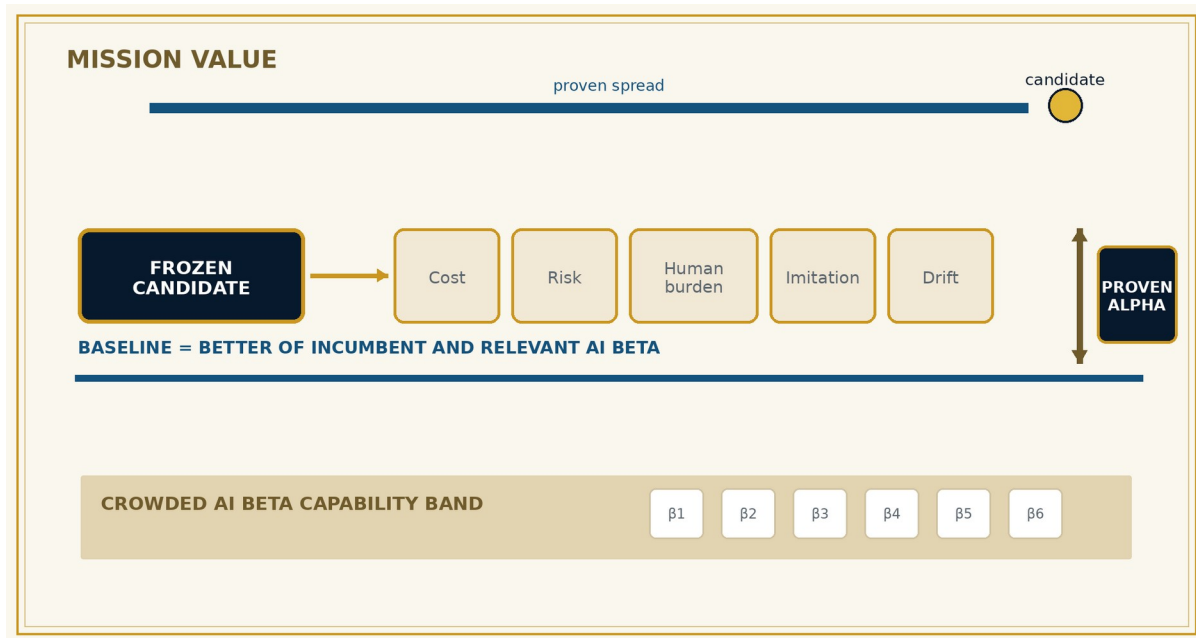
Widely available AI capability improves the market, but it also compresses differentiation. When every qualified organization can call comparable models, generic summarization, generation, search, and agent orchestration become inputs rather than durable moats.

This is AI Beta: the best capability that a relevant competitor can reasonably rent or reproduce. Beta is not low quality. It may be exceptional. It is called Beta because access alone does not prove proprietary mission advantage.

The better baseline is therefore not yesterday's incumbent process. It is the stronger of:

- the organization's current incumbent; and
- the best relevant AI Beta available under realistic cost, latency, security, and operating conditions.

Only performance above that baseline can enter the Alpha calculation.



**FIGURE 2.** Alpha is the risk-adjusted, independently proven spread above the better of the incumbent and relevant AI Beta.

### 1.3 The unit of competition changes

The durable unit of competition is moving from the individual model to the mission institution:

**Mission Constitution + Evidence + World Programs + Skills + Tests + Proofs + Policies + Authority + Chronicle**

That institution can purchase cognition when needed, route tasks to specialized models, and replace suppliers as capabilities, economics, or licences change. Its identity and operating memory do not need to reset with each model cycle.

Gartner's forecast that up to US\$234 billion of enterprise application spending is exposed to "agentic arbitrage" between 2026 and 2030 is relevant context. The firm argues that agents can bypass interface-heavy software and that buyers increasingly want outcomes, institutional memory, and customer context.<sup>[5]</sup> This does not define the SUCCESSOR  $\Omega$  market, but it supports the underlying transition: the value layer is shifting from interfaces and seats toward outcome-bearing, context-retaining systems.

## 2 | Category and canonical definitions

### 2.1 Customer-Owned Mission Intelligence

Customer-Owned Mission Intelligence is a proposed commercial category for a portable intelligence institution constituted for a specific customer and mission. It converts organizational

evidence and replaceable cognitive services into inspectable representations, independently demonstrated advantage, and policy-bounded action.

It is customer-owned only if the customer can, within the agreed rights boundary:

- export it in a documented format;
- inspect its active programs, evidence, policies, and proof;
- operate it without an irreplaceable proprietary control plane;
- substitute cognitive suppliers through stable interfaces;
- restore it from a signed release and Chronicle checkpoint;
- evolve it through a governed successor process; and
- verify that an export produces equivalent bounded behavior.

Ownership is therefore a contractual and technical property, not a slogan.

## 2.2 Neural-Symbolic SUCCESSOR Ω

Neural-Symbolic SUCCESSOR Ω is a lightweight, portable, customer-owned intelligence institution that converts rented frontier capability and organizational evidence into executable, independently proven, mission-compounding alpha.

"Neural-symbolic" specifies the division of labour:

- Neural systems interpret ambiguous evidence, propose hypotheses, synthesize candidate programs, estimate uncertainty, and recruit broad learned priors.
- Symbolic systems represent explicit states, transitions, constraints, policies, plans, predictions, and falsifiers in executable or mechanically checkable form.
- Independent evaluators determine whether the exact candidate's claims survive fresh evidence.

Neither side is treated as automatically truthful. Neural proposals can hallucinate. Symbolic programs can be precisely wrong. Both remain subordinate to evidence and proof.

## 2.3 Mission sovereignty

Mission sovereignty means that the customer controls the constitution, evidence rights, operating policy, identity, authorized deployment, and succession process of the mission institution. It does not mean isolation from external technology. It means dependence is explicit, substitutable where promised, and never confused with ownership of the institution itself.

## 2.4 Alpha, Beta, Spread, and baseline

| Term      | Canonical meaning                                                               |
|-----------|---------------------------------------------------------------------------------|
| Incumbent | The process, team, system, or decision method currently performing the mission. |

| Term                    | Canonical meaning                                                                                                                                           |
|-------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Relevant AI Beta</b> | The strongest realistically rentable or reproducible AI alternative available to the customer or a capable competitor.                                      |
| <b>Baseline</b>         | The better of the incumbent and relevant AI Beta under the same mission utility, constraints, and operating conditions.                                     |
| <b>Spread</b>           | The observed mission-performance difference between a candidate and the baseline before full risk and durability adjustments.                               |
| <b>Alpha</b>            | The lower-confidence, risk-adjusted advantage remaining after cost, risk, fragility, imitation, commoditization, human burden, and drift are accounted for. |
| <b>Proven alpha</b>     | Alpha demonstrated by an exact frozen release against fresh protected evidence and an independent verifier, within an explicit proof envelope.              |

## 2.5 Specialist ASI

A candidate may be admitted as a Specialist ASI only when it demonstrates mission-specific superiority over the incumbent, the best relevant AI Beta, and competing candidates while meeting all hard safety, economic, and reliability gates.

The designation is deliberately narrow. It records a proof result inside a defined mission, distribution, time window, and Authority Envelope. It cannot be generalized to unrelated domains. It expires. It can be impaired. It does not create unrestricted authority.

## 2.6 SUCCESSOR Ω

SUCCESSOR Ω is the operating institution created when a proven mission-specialist candidate passes accountable admission. The successor includes more than capability: it has a constituted identity, a frozen release, proof receipts, an Authority Envelope, monitoring, rollback, memory, and a Live Chronicle.

# 3 | Design doctrine

The architecture follows ten non-negotiable laws.

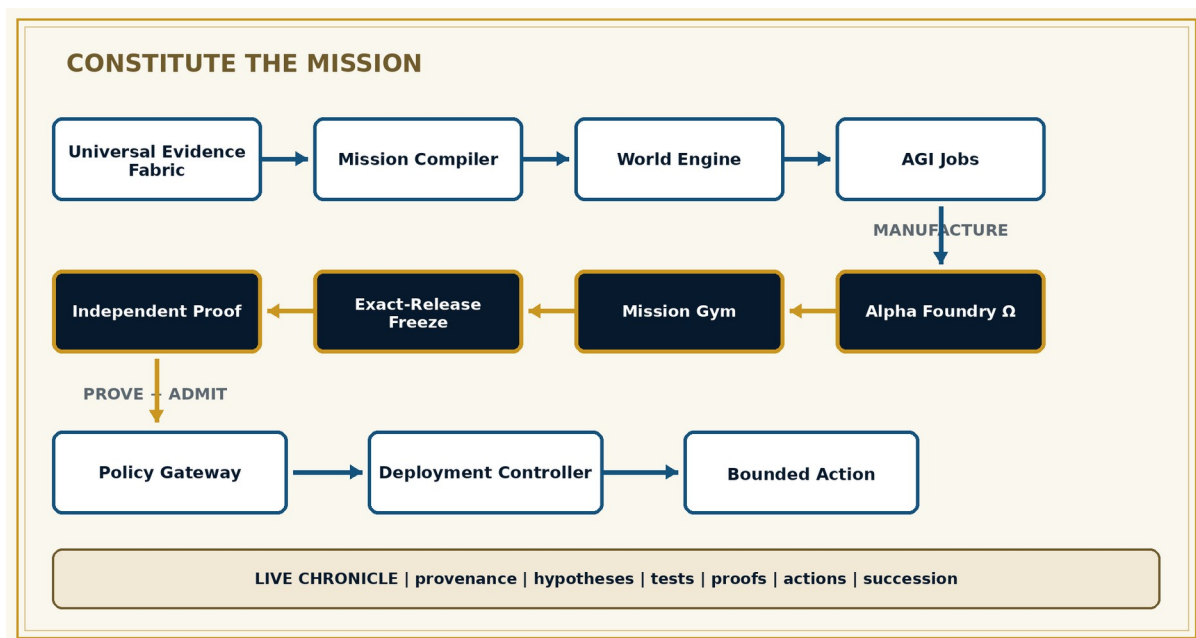
1. Evidence before narrative. Every material belief must expose its evidence, assumptions, scope, uncertainty, and falsifier.
2. Programs are hypotheses. Executability improves inspection and testing; it does not confer truth.
3. The claimant is not the final judge. Candidate manufacturing, training, proof, and authority are institutionally separated.
4. The Gym that teaches must not certify. Generated or adapted environments may form capability but cannot constitute final fresh proof.

5. Freeze before proof. The exact models, prompts, programs, tools, policies, and budgets examined must be the release evaluated.
6. Authority never exceeds fresh proof. Capability can be broader than permission; permission cannot be broader than current evidence.
7. Preserve function, not accidental implementation. Suppliers and internal compositions may change, but changed releases must re-earn proportionate authority.
8. Diversity before premature convergence. Preserve distinct elites for alpha, safety, cost, reliability, sovereignty, transfer, and critical-failure performance.
9. No inherited proof. A descendant may inherit evidence and Chronicle context, but not its predecessor's claim or authority.
10. Customer continuity by construction. Identity, programs, policies, proofs, and Chronicle must survive vendor substitution and organizational turnover.

**The governing invariant: no unverified change may silently cross from learning into consequential action.**

## 4 | Reference architecture

The reference architecture is a manufacturing, proof, admission, and operating system for mission intelligence.



**FIGURE 3.** Reference architecture and constitutional separation of manufacturing, proof, authority, and operations.

## 4.1 Universal Evidence Fabric

The Universal Evidence Fabric ingests and governs documents, structured data, telemetry, procedures, prior decisions, simulations, human expertise, and external observations. It does not flatten them into an undifferentiated vector store.

Each evidence object receives:

- source identity and acquisition time;
- ownership, licence, consent, and use restrictions;
- classification and sensitivity;
- integrity signature or hash where available;
- semantic type and mission relevance;
- confidence, freshness, and known gaps;
- transformation history;
- adversarial-content status; and
- retention, deletion, and export rules.

Evidence is bound to claims and programs through provenance edges. A program can therefore reveal not only what it predicts, but which evidence and transformations produced it.

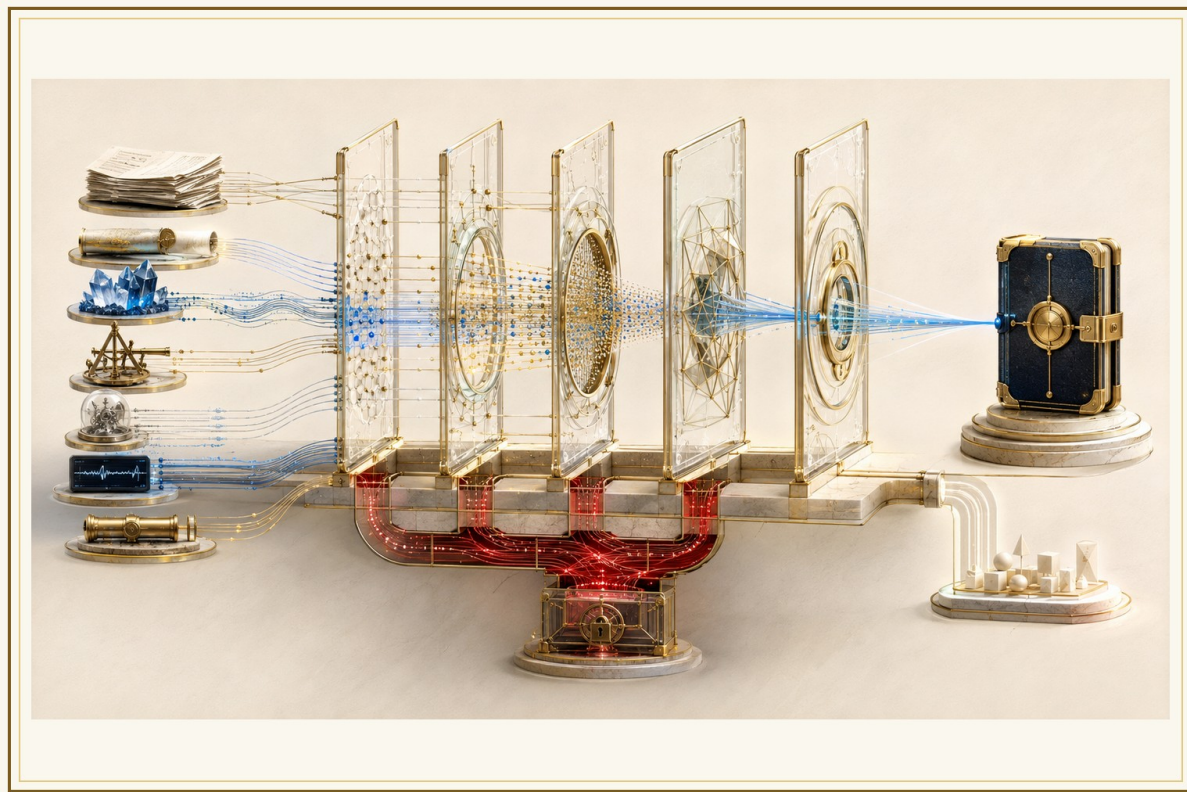
## 4.2 Mission Compiler

The Mission Compiler converts human intent into a formal Mission Constitution. It refuses the illusion that a polished objective statement is automatically testable.

A valid compiled mission requires:

- observable states and measurable outcomes;
- permitted and prohibited actions;
- triggerable constraints;
- a credible baseline;
- a sufficiently independent evaluator;
- realistic historical replay, simulation, or staged testing;
- explicit economic, safety, and human-burden measures;
- an evidence plan and freshness rule;
- rollback and stop conditions; and
- an authority ceiling.

If these elements cannot be established, the mission is rejected, narrowed, or kept in research mode. This "compilability gate" is a product feature: it prevents capital from flowing into claims that cannot be tested.



**CONCEPTUAL INSTITUTIONAL PLATE 01.** Organizational intent becomes a bounded, executable Mission Constitution.

**An attractive objective is not yet a mission. Evidence, evaluability, constraints, rollback, and an authority ceiling must compile before capital or candidate manufacturing advances.**

### 4.3 Mission Pack SDK and AGI Jobs

The Mission Pack is the portable specification of one mission. It contains schemas, evaluators, adapters, policies, tests, baselines, and customer-facing explanations. The SDK allows domain specialists to add mission-specific components without changing the constitutional kernel.

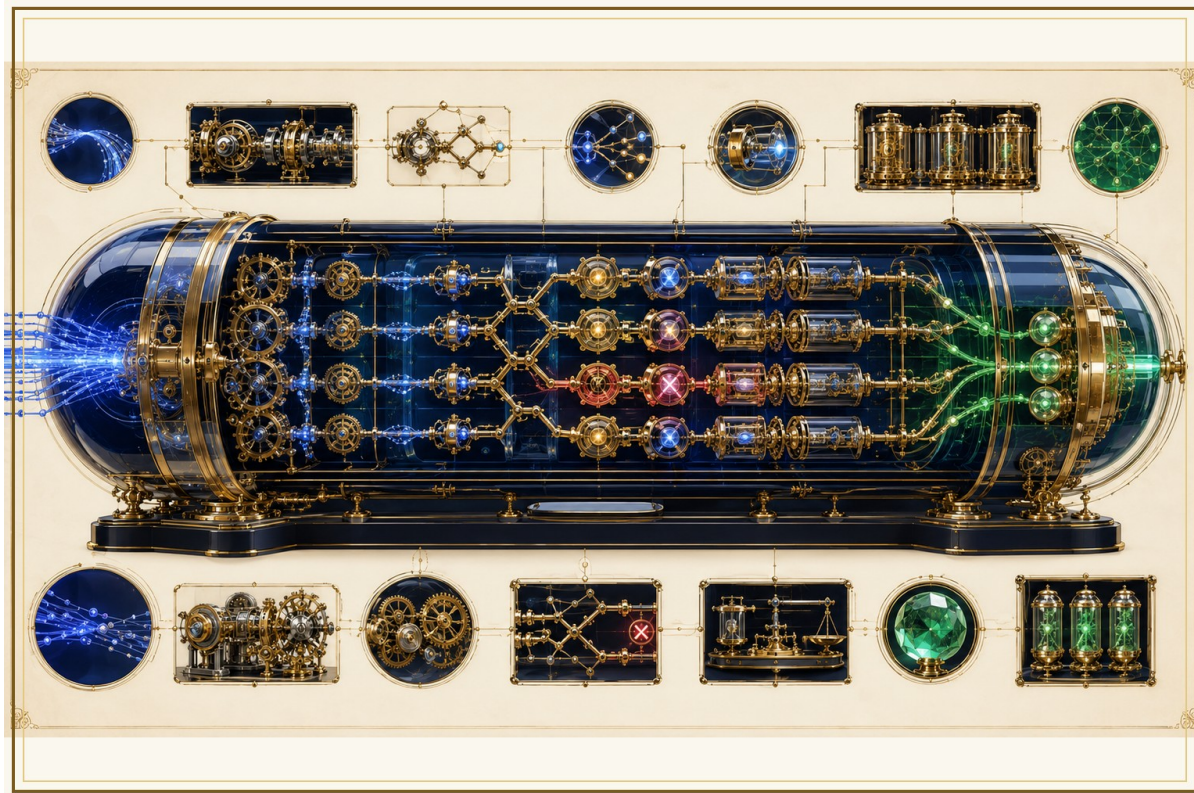
AGI Jobs decompose the mission into sealed, evaluable work orders. Each job defines:

- objective and scope;
- inputs and admissible evidence;
- tools and data access;
- budget, latency, and resource limits;
- candidate role and dependencies;
- evaluator and final verifier;

- constraints and forbidden actions;
- evidence required for acceptance;
- threshold and critical-failure rules; and
- dispute, expiry, and rollback conditions.

The job is the economic and constitutional membrane between competing agents and the customer mission. It is not a vague prompt.

## 4.4 Neural-Symbolic World Engine



*CONCEPTUAL INSTITUTIONAL PLATE 02. Evidence becomes executable, falsifiable world programs.*

**The institution does not merely recall what was learned. It writes, executes, challenges, and evolves the testable programs through which it navigates the mission world.**

The World Engine maintains versioned executable hypotheses about the mission world. Its program types may include:

- state estimators;
- transition and dynamics models;
- causal hypotheses;
- prediction functions;
- planning and policy programs;

- constraint monitors;
- uncertainty estimators;
- anomaly and failure programs; and
- counterfactual simulators.

Every world program must declare its evidence, assumptions, applicable domain, predictions, falsifiers, uncertainty, dependencies, proof status, and revision history.

## 4.5 Alpha Foundry $\Omega$

The Foundry manufactures a Successor Manifold of competing candidates. It can combine neural learning, symbolic planning, program synthesis, search, offline reinforcement learning, agentic reinforcement learning, deterministic solvers, specialist collectives, and human-machine workflows.

Candidate components compete behind stable interfaces. No model, framework, or agent SDK becomes constitutionally permanent merely because it performs well in one release.

## 4.6 Sovereign Mission Gym and Experience Foundry

The Gym creates the experience required to train and stress candidates. It supports demonstrations, historical replays, synthetic curricula, self-play, rare events, adversarial cases, constraint collisions, transfer situations, and reward or observation shocks.

An Experience Foundry may adapt the Gym to diagnosed candidate weaknesses. It must preserve trusted environment logic where promised, validate generated scenarios for solvability and relevance, and record every transformation.

Gym results are formation evidence, not final admission evidence.

## 4.7 Exact-Release Freeze

Before independent proof, the complete challenger is sealed:

- models, weights, and endpoints;
- prompts and routing policies;
- programs and world-model versions;
- tools, adapters, and dependency versions;
- workflows and memory state;
- policies and permissions;
- development-evaluator configuration and the proof-interface contract;
- budgets and rate limits;
- rollback controls; and

- cryptographic release manifest.

Learning after freeze is prohibited inside the proof trial unless the frozen object explicitly includes an evaluated learning rule, writable state classes, authorized data sources, update budget, invariants, checkpoints, and monitoring thresholds. Any change outside that envelope produces a new release identity.

Where weights are available, the release manifest identifies and hashes them. Where a rented model is opaque, the manifest records the provider, model and endpoint identifier, routing and sampling configuration, tool contract, date, observed behavioral fingerprint, and supplier assurances. An unpinned or silently changed supplier is an impairment event and can reduce admitted authority.

## 4.8 Independent Proof Network

The Proof Network evaluates the frozen candidate against fresh, protected evidence. Independence can be organizational, technical, economic, or ideally all three. Proof cases, verifier configuration, analysis code, and threshold rationale are sealed separately under Proof Network custody. Pass-contingent compensation is prohibited, and material conflicts, funding, governance, reporting lines, and data custody are disclosed in the receipt.

The network measures outcome quality, hard-constraint compliance, reliability, economics, transfer, tail risk, critical failures, evidence coverage, proof time, human burden, reversibility, and uncertainty calibration.

No trustworthy verifier, no authority to scale autonomy. No independent proof, no SUCCESSOR Ω.

## 4.9 External Policy Gateway

The Gateway translates proof and customer policy into a machine-enforceable Authority Envelope. It is external to the candidate's own reasoning path. The candidate may request authority; it may not grant authority to itself.

## 4.10 Production Deployment Controller

The Controller brokers real-world action. It validates identity, release hash, proof currency, policy, approval, resource limits, and rollback availability before a tool call can cross into production.

## 4.11 Live Chronicle

The Chronicle is the tamper-evident institutional record of provenance references, hypotheses, programs, tests, failures, proofs, decisions, actions, approvals, incidents, rollbacks, impairments, and succession events. Raw evidence remains in a governed vault subject to access, retention, deletion, residency, and legal-hold rules. The Chronicle enables audit, recovery, transfer, and compounding learning without pretending that history itself is proof.

## 4.12 Canonical deployment planes and trust boundaries

Implementations may vary, but the constitutional boundaries must remain explicit, testable, and visible to the customer.

| Plane                       | System of record                                                      | May change serving behavior?                | Boundary law                                                                                                |
|-----------------------------|-----------------------------------------------------------------------|---------------------------------------------|-------------------------------------------------------------------------------------------------------------|
| <b>Evidence</b>             | Customer sources, governed vaults, bindings, provenance               | No                                          | Content is evidence, never authority; every material transformation is attributable.                        |
| <b>Foundry and learning</b> | Candidate programs, models, prompts, workflows, experiments           | No                                          | No production credentials, consequential actions, or visibility into protected proof evidence.              |
| <b>Proof</b>                | Precommitted protocol, protected evidence, verifier runtime, receipts | No                                          | Executes the frozen release under independent custody; cannot modify the challenger or broaden the mission. |
| <b>Policy and control</b>   | Mission Constitution, approvals, Authority Envelopes, revocations     | Yes, through named customer governance only | The candidate may request authority but cannot write or override policy.                                    |
| <b>Action</b>               | Deployment Controller, action broker, tools, ephemeral credentials    | Yes, only inside the current envelope       | Every consequential action requires pre-action admission checks and a signed post-action receipt.           |
| <b>Chronicle</b>            | Signed events, checkpoints, impairments, succession lineage           | No direct action                            | Preserves the institutional record and recovery state; history alone cannot manufacture proof.              |

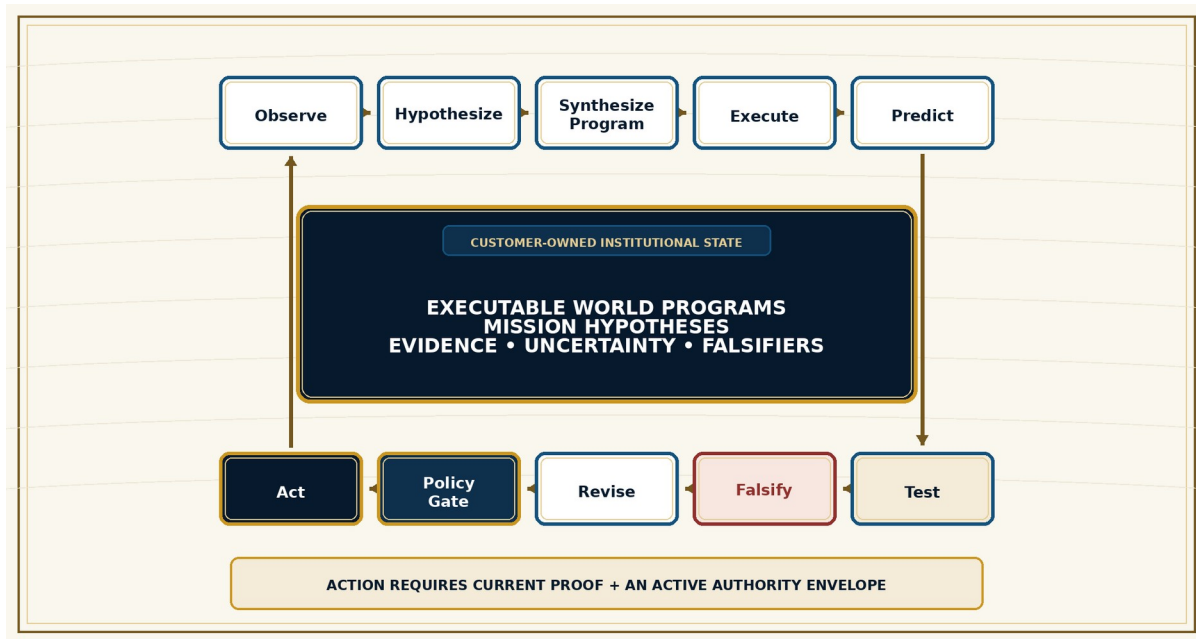
Every cross-plane request carries, at minimum:

**mission\_id + release\_id + principal + policy\_version + authority\_envelope\_id + proof\_receipt\_id where required + timestamp + nonce + request\_hash**

The corresponding decision or action receipt adds the decision, response hash, tool and target, outcome link, and Chronicle anchor.

Boundary crossings fail closed and are anchored to the Chronicle. High-consequence deployments should separate protected-evidence custody, policy administration, secrets, verifier operation, and production tooling administratively and, where warranted, physically.

## 5 | The neural-symbolic cognition loop



**FIGURE 4.** The cognition loop turns observations into falsifiable programs and permits action only through the policy boundary.

### 5.1 Observe

The institution forms a typed observation from evidence rather than copying raw content directly into permanent memory. It records what was observed, through which sensor or source, at what time, under which rights, and with what uncertainty.

### 5.2 Hypothesize

Neural systems propose candidate mechanisms, latent states, causal links, exceptions, and alternative explanations. Hypotheses remain plural. Minority explanations are preserved when their expected information value or tail-risk relevance is material.

### 5.3 Synthesize Program

The system compiles a hypothesis into an executable or mechanically checkable representation. The program exposes inputs, state, transitions, invariants, outputs, and failure conditions. Synthesis occurs in a restricted environment with no implicit production authority.

### 5.4 Execute

The program is executed against replays, simulations, counterfactuals, and permitted observations. Deterministic components handle exact constraints and planning where appropriate; neural models are recruited for ambiguity, repair, and semantic interpretation.

## 5.5 Predict

Every material hypothesis must risk being wrong. It produces a prediction, an uncertainty estimate, a validity horizon, and explicit observations that would contradict it.

## 5.6 Test

Tests compare predictions with withheld or newly arriving reality. Development tests guide repair. Fresh protected tests support proof. They are not interchangeable.

## 5.7 Falsify

Contradictions are first-class institutional events. A falsified program is retired, scoped down, contested, or retained only as a known-failure example. The Chronicle preserves the failure so that future candidates cannot quietly rediscover it as if new.

## 5.8 Revise

Revision may patch a program, alter an abstraction, change a model supplier, split a hypothesis, or restructure the candidate. Any production-material revision creates a descendant release that must pass proportionate proof.

## 5.9 Act

Action occurs only when the frozen serving release, current proof, and Authority Envelope all permit it. The system records the decision basis, expected outcome, risk, approval, tool call, observed result, and rollback state.

## 5.10 Why programs help - and why they are not enough

Executable program world models are a credible mechanism, not yet an enterprise-ready substrate. WorldCoder learned editable Python world models from interaction and reported efficiency gains in constrained environments; GIF-MCTS synthesized code world models that outperformed baselines on an 18-environment benchmark and improved sample efficiency and inference speed; TheoryCoder and TheoryCoder-2 combined synthesized transition programs with hierarchical abstractions in deterministic games; and a general-game-playing system compiled rules into executable models used by search.[\[6\]\[7\]\[25\]\[27\]\[28\]](#)

These results support inspectable, executable hypotheses. They do not establish automatic truth or safe transfer to enterprise operations. A 2026 counterexample study demonstrates the central danger: a code world model can pass sampled transition-accuracy gates yet fail systematically when residual error affects decision-critical dynamics.[\[26\]](#) SUCCESSOR  $\Omega$  therefore evaluates decision-relevant consequences, not only average transition accuracy, and never grants authority because code merely compiles or replays familiar cases.

## 6 | Continuous evolution without uncontrolled self-modification

The phrases "continuously evolving" and "exact-release freeze" appear contradictory unless the operating topology is explicit. SUCCESSOR  $\Omega$  resolves the tension with two concurrent branches.

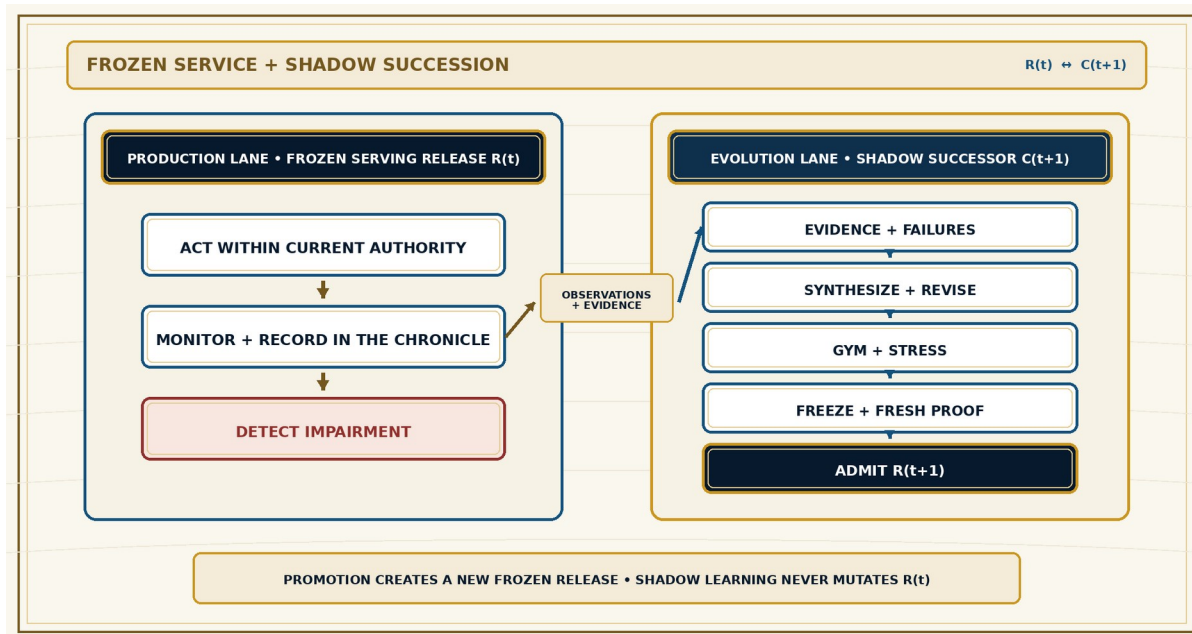
| Branch                                     | Role                                | May learn?                                              | May act?                           |
|--------------------------------------------|-------------------------------------|---------------------------------------------------------|------------------------------------|
| <b>Serving release <math>R(t)</math></b>   | The admitted production institution | No material self-modification inside its proof envelope | Yes, only within current authority |
| <b>Learning branch <math>C(t+1)</math></b> | Shadow successor candidate          | Yes; may synthesize, revise, retrain, and reorganize    | No consequential production action |

The Live Chronicle supplies observations from  $R(t)$  to  $C(t+1)$  under policy. The learning branch proposes a descendant, which must pass:

**Compile → Train → Stress → Freeze → Fresh Proof → Accountable Admission**

Only then can  $C(t+1)$  become  $R(t+1)$ .

This produces continuous institutional learning without continuous unreviewed mutation of the production actor. Emergency patches may follow an accelerated path, but they still receive a new identity, reduced authority, targeted proof, and explicit approval.



**FIGURE 5.** A frozen serving release operates while a separate shadow branch learns; promotion requires a new proof and admission event.

## 7 | Foundry logic: competition, diversity, and architecture markets



**CONCEPTUAL INSTITUTIONAL PLATE 03.** *A quality-diverse foundry manufactures competing architectures.*

**Excellence is plural. Preserve distinct elites for alpha, safety, sovereignty, reversibility, transfer, reliability, and cost rather than collapsing them into one average score.**

### 7.1 The Successor Manifold

The Foundry explores combinations of world representations, model suppliers, planners, critics, memories, tools, and workflows. It does not ask one architecture to be best at everything.

Each candidate has a behavior descriptor and a score vector that may include:

- mission utility and net alpha;
- constraint compliance;
- reliability and calibration;
- cost and latency;
- sovereignty and supplier concentration;
- reversibility;
- transfer and adaptation burden;

- human workload;
- tail-risk performance; and
- critical-failure severity.

Quality-diversity methods provide a useful search pattern by preserving high-performing solutions across different behavioral niches.<sup>[11]</sup> They do not by themselves solve institutional multiobjective governance. SUCCESSOR Ω therefore combines niche archives, Pareto dominance, hard gates, and customer-defined lexicographic priorities.

## 7.2 Architecture Market

Candidate components enter through signed adapters and compete on measured contribution. An architecture market may route Thinker, Worker, Internal Critic, planner, tool specialist, and synthesizer roles across local, open-weight, or rented models.

TRINITY-style role assignment and AsyncThink-style asynchronous coordination support the lightweight thesis: a compact customer-owned coordinator can recruit larger systems only when needed.<sup>[13][14]</sup> Internal critics remain claimant-side components. They are never renamed or mistaken for the Independent Proof Network.

## 7.3 Collective Dynamics Governor

Multiple agents do not automatically create independent evidence. Ten agents sharing one model family, prompt, evidence set, and failure mode may produce one correlated opinion ten times.

The Collective Dynamics Governor tracks:

- model and evidence diversity;
- influence concentration;
- opinion dispersion;
- consensus velocity;
- minority-view survival;
- shared-supplier dependence;
- correlated errors; and
- whether influence follows demonstrated accuracy.

Research on large simulated agent communities indicates that interaction topology and bias can generate indifference, polarization, or consensus regimes.<sup>[15]</sup> SUCCESSOR Ω uses this as a safety signal, not as proof that its own populations will behave identically.

## 8 | The Sovereign Mission Gym



**CONCEPTUAL INSTITUTIONAL PLATE 04.** *The adaptive Gym manufactures experience without certifying its own candidates.*

**The Gym may teach, stress, and expose failure. The final proof plane remains structurally independent, protected, and unavailable to the teacher.**

### 8.1 Manufacturing experience

Many customer missions lack sufficient labelled examples, rare failures, or safe opportunities for trial and error. The Gym manufactures experience while maintaining a boundary between teaching and certification.

Three primitives organize the Gym:

- Stage: the state or distribution from which a candidate enters.
- Contract: what it may observe, do, receive, and spend.
- Chain: how situations connect into long-horizon missions.

EnvHarness demonstrates a promising pattern: wrap a trusted static environment with programmable components that reshape starting conditions, interactions, and task sequences while preserving the underlying verifier.<sup>[9]</sup> In SUCCESSOR  $\Omega$ , an EnvHarness-like layer may adapt the teacher. It may not control independent admission.

## 8.2 Experience Foundry

SPADE-like environment design can generate executable worlds near a learner's capability frontier and target situations where privileged hints reveal a capability gap.<sup>[10]</sup> This is useful for synthetic missions, rare events, constraint collisions, and transfer scenarios.

Generated environments can contain faulty rewards, exploitable verifiers, or unrealistic causal structure. They may teach, stress, and rank candidates. They may never constitute final fresh proof.

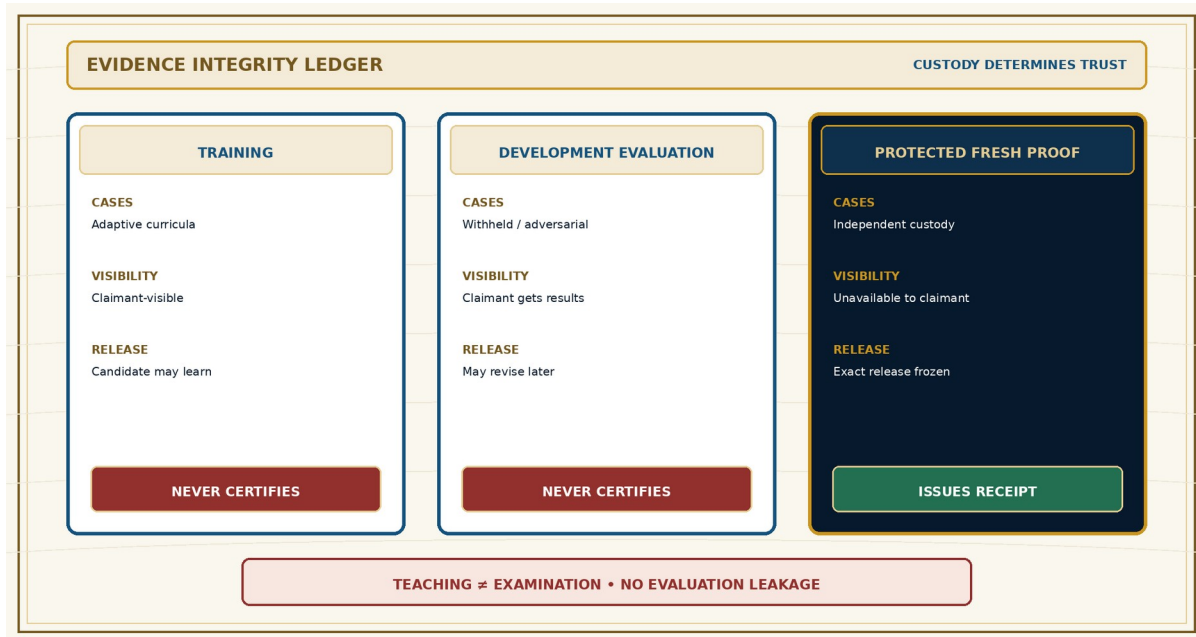
## 8.3 Gym integrity

Every Gym episode records:

- environment and component hashes;
- data origin and synthetic transformations;
- candidate observations and actions;
- rewards and verifier outputs;
- exploits, anomalies, and human interventions;
- coverage and blind spots; and
- whether the Gym episode is training or development evaluation. Protected-proof episodes are recorded separately by the Proof Network.

This prevents leakage between teaching and examination and makes reward hacking inspectable.

The separation test is operational: the protected plane must demonstrate independent evidence custody, no claimant visibility into cases, a frozen release, precommitted evaluation, and a signed proof-or-failure receipt. If any condition fails, the result is development evidence, not fresh proof.



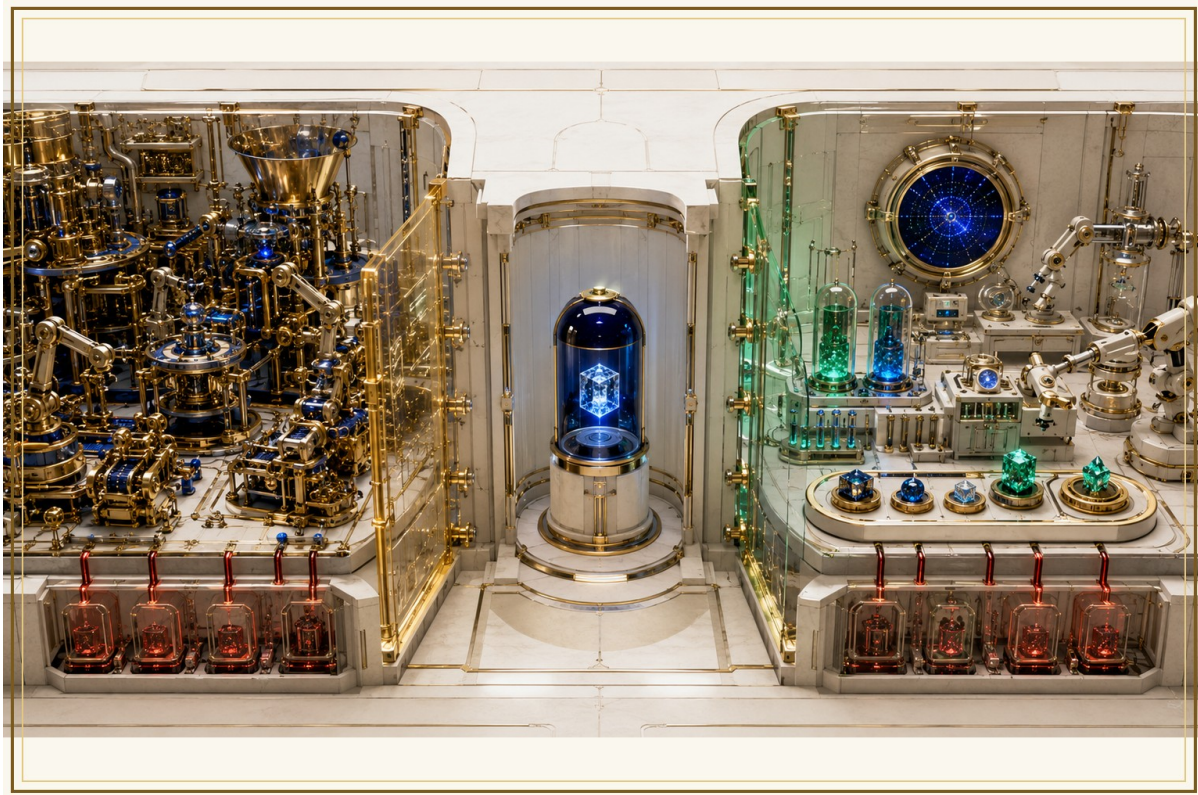
**CONTROL INSTRUMENT 01.** Training, development evaluation, and protected fresh proof use different visibility, mutability, custody, and certification rules.

## 8.4 Evidence-plane integrity gate

Before a trial can produce admissible evidence, the Proof Network must record each control below as pass or fail. A failed condition cannot be averaged away by strong task performance.

| Integrity condition        | Observable test                                                                                                                                                              | Failure consequence                               |
|----------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------|
| Independent custody        | Access logs and commitments show protected cases remained outside claimant control                                                                                           | Freshness claim is void                           |
| Frozen release             | Manifest hash, dependencies, tools, policies, and permissions were sealed before case access                                                                                 | Result is not attributable to an exact challenger |
| Precommitted protocol      | Metrics, gates, exclusions, stopping rules, and analysis were signed before results                                                                                          | Result is development evidence only               |
| No unauthorized adaptation | Only the frozen learning rule, declared writable state, protocol-defined resets, and case-isolated updates occurred; no cross-case leakage or undeclared repair was possible | Run is quarantined and repeated                   |
| Complete receipt           | Signers, evidence IDs, hashes, exceptions, outcomes, and validity window are recorded                                                                                        | Admission cannot proceed                          |

## 9 | Independent fresh proof



**CONCEPTUAL INSTITUTIONAL PLATE 05.** *A frozen candidate crosses into protected fresh proof.*

**The claimant may manufacture capability. It may not serve as its own final judge. Proof attaches only to the exact release, protocol, evidence, and validity window examined.**

### 9.1 What fresh means

Fresh evidence is evidence that the claimant could not reliably memorize, tune against, or reconstruct before release freeze. Freshness can be established through later time periods, sealed samples, private scenarios, independent acquisition, cryptographic commitments, or controlled real-world trials.

Fresh does not necessarily mean random. It means protected from claimant adaptation and representative of the admitted mission.

### 9.2 Proof protocol

The proof protocol has seven stages:

1. Commit the Mission Constitution, baseline, metrics, gates, and verifier before testing.
2. Freeze the exact challenger and dependency manifest.

3. Confirm separation between training evidence and protected evidence.
4. Execute the trial under controlled budgets, tools, and observation rules.
5. Record outcomes, critical failures, uncertainty, cost, latency, and human burden.
6. Reproduce or cross-check material results through an independent verifier.
7. Issue a signed proof receipt with scope, limits, and expiry - or a failure receipt.

### 9.3 Proof is multidimensional

An average score cannot compensate for a prohibited action or catastrophic tail event. Evaluation therefore combines:

- hard gates that must all pass;
- lexicographic priorities for non-substitutable values;
- statistical performance with confidence bounds;
- stress and transfer evaluations;
- critical-failure budgets;
- economics and human burden; and
- uncertainty calibration.

The Proof Network may return a narrow pass, conditional pass, failure, or insufficient-evidence result. A narrow honest proof is more valuable than a broad untestable claim.

### 9.4 Independence levels

"Independent" is disclosed rather than assumed.

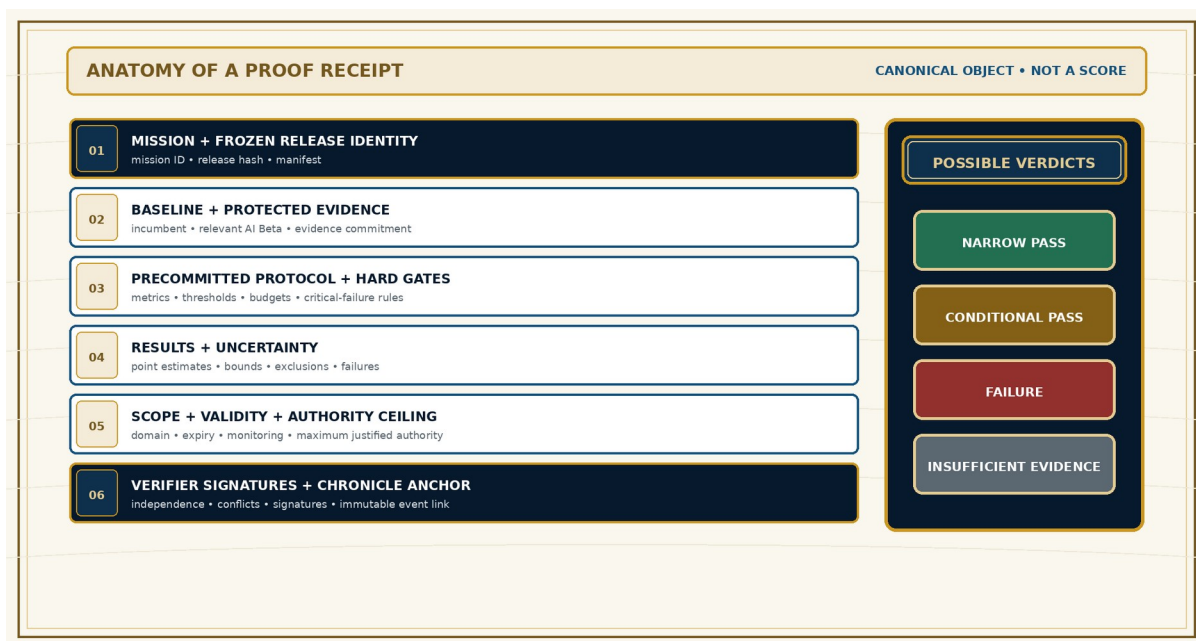
| Level | Verification arrangement                                                                                                                                                                                      |
|-------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 10    | Same candidate process; internal critic only. Not independent proof.                                                                                                                                          |
| 11    | Separate role or team inside the claimant organization. Internally separated evaluation.                                                                                                                      |
| 12    | Separate governance, reporting, protocol control, and protected-evidence custody inside a common enterprise. Independently evaluated, but not externally independent.                                         |
| 13    | Contractually independent third party with separate custody and disclosed conflicts. Minimum for public "independently proven" claims or consequential authority, unless a documented equivalent is approved. |
| 14    | Multiple independently governed verifier organizations.                                                                                                                                                       |

The required level is proportional to consequence. Proof receipts disclose verifier identity, data custody, funding and conflicts, protocol control, and replication status. Independence reduces correlated incentives; it does not make a verifier infallible.

## 9.5 Proof receipt

Every admitted claim is traceable to a receipt containing:

- mission and release identifiers;
- baseline identity and configuration;
- verifier identity and independence declaration;
- protected-evidence commitment;
- metrics, thresholds, and statistical method;
- observed results and confidence bounds;
- critical failures and exclusions;
- economics, latency, and human burden;
- validity domain, expiry, and monitoring conditions;
- authority implication; and
- signatures and Chronicle anchors.



**CONTROL INSTRUMENT 02.** The proof receipt binds an exact mission, release, baseline, evidence commitment, protocol, result, scope, validity window, verifier, and Chronicle anchor.

## 9.6 Proof validity, impairment, and operational currency

A proof claim attaches to a complete proof tuple:

**ProofClaim = mission\_id + release\_hash + baseline\_set + protocol\_hash + evidence\_commitment + verifier + operating\_environment + validity\_window**

It does not attach broadly to a model family, vendor, team, product name, or descendant. A material change to any element requires an explicit transfer analysis and normally proportionate fresh proof.

Proof currency is impaired when:

- the release, prompts, programs, tools, permissions, or material dependencies change;
- protected evidence is leaked, contaminated, or shown to be unrepresentative;
- the incumbent or relevant AI Beta materially improves;
- the operating population, environment, law, cost structure, or risk boundary moves outside the validated domain;
- a critical failure, prohibited action, or unexplained anomaly occurs;
- verifier independence or an undisclosed conflict is compromised;
- monitoring coverage, release identity, Chronicle integrity, or rollback capability is lost; or
- the receipt reaches its stated expiry.

The mandatory response is:

**Mark impaired → stop the associated earned claim → contract or suspend authority → preserve evidence → issue an impairment receipt → select targeted re-proof, full re-proof, or retirement**

Impairment does not erase the historical trial result. It ends that result's use as current operational justification.

**Expired, impaired, or unverifiable proof authorizes nothing.**

## 10 | Alpha as an economic proof object

### 10.1 Baseline

For mission  $m$ , let  $U(m, x)$  be the customer-approved gross mission utility of system or process  $x$  under comparable conditions, before the explicit  $C$ ,  $R$ ,  $H$ ,  $I$ , and  $D$  deductions below. The baseline is:

**$b^* = \operatorname{argmax} U(m, b)$ , for  $b$  in {incumbent, relevant AI Beta}**

This prevents the Foundry from claiming success by outperforming an obsolete incumbent while remaining inferior to a readily rentable alternative.

## 10.2 Proven mission alpha

For frozen candidate  $c$ , define the proof-trial spread as the lower confidence bound of its utility advantage over  $b^*$ . Proven mission alpha is conceptually:

$$\text{Alpha}(m, c) = \text{LCB}[U(m, c) - U(m, b^*)] - C - R - H - I - D$$

where:

- $C$  is incremental operating, integration, and switching cost;
- $R$  is a mission-approved risk reserve;
- $H$  is incremental human burden;
- $I$  is an imitation or commoditization reserve; and
- $D$  is a drift and impairment reserve over the validity horizon.

The equation is a governance template, not a universal accounting standard. Each Mission Constitution defines utility, reserves, confidence level, and non-negotiable constraints. Alpha exists only when all hard gates pass and the remaining value exceeds the admission threshold.

## 10.3 Alpha is perishable

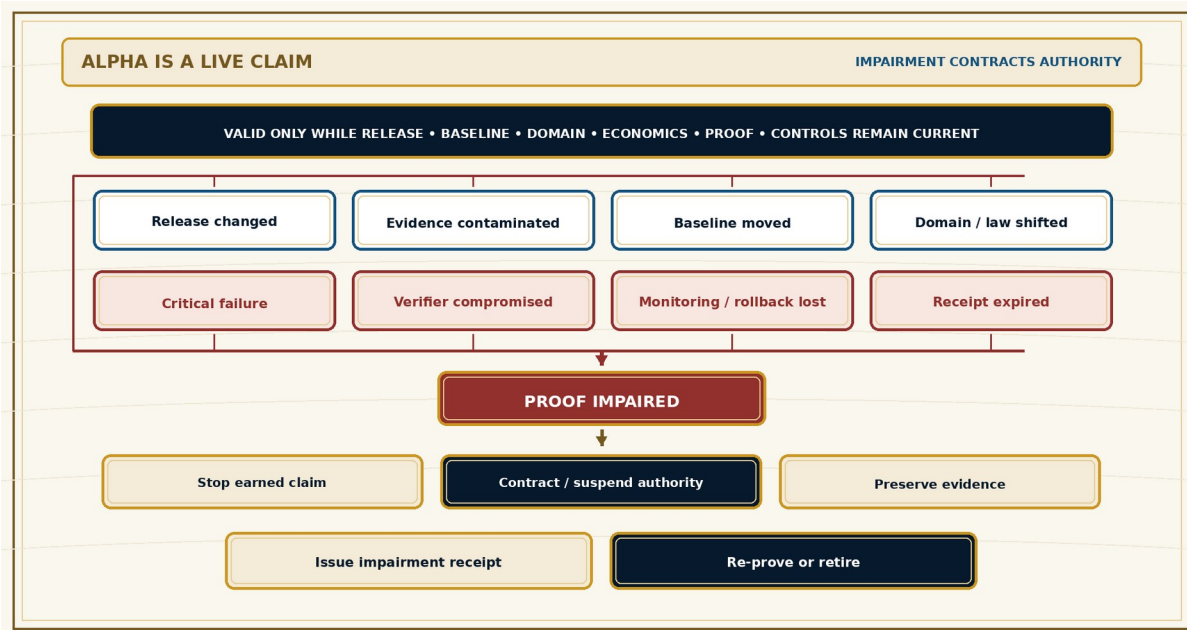
Proven alpha can decay when:

- AI Beta improves;
- the incumbent changes;
- customer data or operations drift;
- costs rise;
- supplier terms change;
- imitation compresses differentiation;
- uncertainty grows; or
- a critical failure occurs.

The Chronicle continuously compares live evidence with the proof envelope. Impairment triggers authority contraction, re-underwriting, new AGI Jobs, fresh proof, or retirement.

## 10.4 The Alpha Impairment Monitor

The operating dashboard must show alpha status as current, expiring, expired, impaired, or absent - never as a timeless cumulative score. A trigger creates an impairment event, identifies the observable signal, stops the affected earned claim, contracts the associated authority, and assigns either targeted re-proof, full re-proof, or retirement.



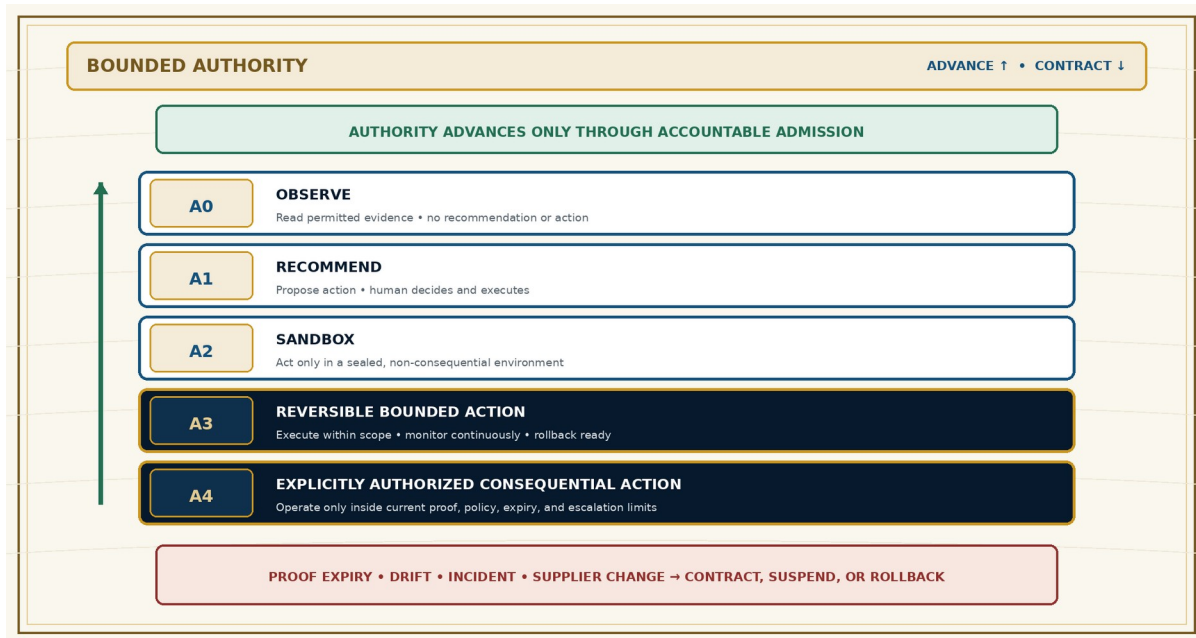
**CONTROL INSTRUMENT 03.** Mission alpha remains an earned claim only while the frozen release, baseline, operating domain, economics, proof, and production controls remain current.

## 10.5 Impairment response doctrine

The response is determined by the claim-bearing object that changed—not by a generic confidence score. Contraction is immediate; restoration requires proportionate evidence.

| Impairment event                           | Immediate control                                                              | Return path                                       |
|--------------------------------------------|--------------------------------------------------------------------------------|---------------------------------------------------|
| Release or dependency changed              | Suspend the affected earned claim and its authority                            | Targeted equivalence proof or full fresh proof    |
| Evidence or verifier compromised           | Revoke the affected receipt; preserve the incident record                      | Reconstituted independent proof                   |
| Baseline or economics moved                | Freeze the alpha claim; continue only within separately valid safety authority | Re-underwrite and recompute net mission advantage |
| Critical failure or rollback lost          | Contract to observe/recommend or stop                                          | Incident closure plus full re-admission           |
| Receipt expired or drift threshold crossed | Expire the claim automatically                                                 | Scheduled renewal proof against current reality   |

## 11 | Accountable admission and bounded authority



**FIGURE 6.** Authority increases only through explicit, expiring levels; proof failure or drift contracts it.

### 11.1 Authority levels

| Level | Permitted role                                  | Typical controls                                                                              |
|-------|-------------------------------------------------|-----------------------------------------------------------------------------------------------|
| A0    | Observe                                         | Read-only sources; no recommendations represented as approved decisions.                      |
| A1    | Recommend                                       | Human decision required; evidence, uncertainty, and alternatives visible.                     |
| A2    | Act in a sealed sandbox                         | Synthetic or replay environment; no production side effects.                                  |
| A3    | Take reversible bounded action                  | Tool allowlist, spend/rate limits, monitoring, automatic rollback.                            |
| A4    | Take explicitly authorized consequential action | Named mission scope, elevated approval, continuous monitoring, short expiry, tested recovery. |

Authority is a vector, not a single number. A release may have A3 authority for one tool and A1 for another, within specific data, geography, time, budget, and counterparty boundaries.

### 11.2 Authority Envelope

The Authority Envelope specifies:

- mission and release identity;
- permitted data, tools, actions, and targets;

- prohibited actions and immutable constraints;
- monetary, compute, rate, and latency limits;
- approval requirements and separation of duties;
- monitoring signals and escalation routes;
- proof prerequisites and expiry;
- rollback mechanism and maximum recovery time; and
- incident and kill-switch authority.

The Policy Gateway checks the envelope mechanically. Natural-language persuasion cannot override it.

### 11.3 Deployment Controller

Before every consequential action, the Controller verifies:

**Identity → Release → Proof currency → Policy → Approval → Resources → Reversibility**

It issues a signed action receipt and later binds the observed outcome. If proof expires, the Chronicle detects impairment, or rollback is unavailable, the Controller contracts or suspends authority.



**CONCEPTUAL INSTITUTIONAL PLATE 06.** *Proven capability becomes action only through bounded, expiring authority.*

**Capability does not confer authority. Customer governance grants it; the Policy Gateway enforces it; the Deployment Controller acts only inside its current envelope.**

## 12 | Customer ownership, portability, and the package



**CONCEPTUAL INSTITUTIONAL PLATE 07.** *The durable mission package persists while cognitive suppliers change.*

**The customer may rent frontier capability without surrendering control of its institutional mission layer. Supplier substitution preserves assets, not inherited proof.**

### 12.1 The durable owned layer

The customer-owned package is designed to contain the institution's accumulated mission intelligence:

- Mission Constitution and identifiers
- Evidence metadata, bindings, and customer-provided evidence where export rights permit
- Customer-specific world programs and hypothesis graphs
- Skills, workflows, tests, and commissioned adapters
- Baselines, proof receipts, and uncertainty state
- Policies, Authority Envelopes, and deployment controls
- Memory, decisions, and Live Chronicle
- Release manifests, model/tool bills of materials, and restoration instructions

## 12.2 The rights boundary

Customer ownership does not rewrite third-party law or licences. The package must distinguish:

- customer-owned or customer-licensed evidence;
- commissioned work product and assigned or licensed rights;
- open-source components and their obligations;
- external model outputs and provider terms;
- proprietary model weights or services that are referenced but not transferred; and
- regulated or personal data subject to access, retention, residency, and deletion duties.

The contract should state which artifacts are owned, licensed, portable, replaceable, or excluded. The manifest should enforce that boundary technically.

## 12.3 Portability levels

| Level | Portability outcome                                                                      |
|-------|------------------------------------------------------------------------------------------|
| P0    | Managed service only; no complete institutional export.                                  |
| P1    | Exportable evidence, policies, memory, and Chronicle.                                    |
| P2    | Portable mission runtime with documented supplier interfaces.                            |
| P3    | Customer-controlled deployment and keys; vendor substitution tested.                     |
| P4    | Mission-appropriate disconnected or sovereign operation using approved local components. |

Not every mission requires P4. Every customer claim must name the actual level and pass its corresponding test.

## 12.4 Export-and-rebuild test

A package is not proven portable until an independent operator can:

1. verify the release and dependency manifest;
2. restore the institution in the target environment;
3. reconnect permitted evidence and tools;
4. substitute at least one declared replaceable supplier;
5. reproduce the agreed reference tests; and
6. confirm equivalent policy boundaries and Chronicle continuity.

This is the ownership proof.

## 13 | Security, safety, and resilience

### 13.1 Evidence is data, not authority

SUCCESSOR Ω must assume that documents, webpages, images, emails, tool outputs, and retrieved records can contain malicious or accidental instructions. OWASP identifies both direct and indirect prompt injection as material risks for LLM systems, including instructions hidden in external content.<sup>[23]</sup> MITRE ATLAS provides a complementary knowledge base of adversarial tactics and techniques affecting AI-enabled systems.<sup>[24]</sup> The Evidence Fabric therefore separates authority-bearing instructions from untrusted evidence.

Core controls include:

- typed instruction channels and authenticated policy sources;
- content quarantine and provenance-aware parsing;
- sanitization and structural extraction before model exposure;
- least-privilege tool credentials and short-lived tokens;
- output validation before any interpreter or tool call;
- deterministic policy enforcement outside the model;
- sandboxed program synthesis and execution;
- memory admission rules and integrity checks;
- canary tests, red-team suites, and anomaly detection; and
- human approval for elevated or irreversible actions.

No prompt is a security boundary. The Policy Gateway and Deployment Controller are.

### 13.2 Threat and control matrix

| Threat                                | Principal control                                                                        |
|---------------------------------------|------------------------------------------------------------------------------------------|
| Direct or indirect prompt injection   | Evidence/instruction separation, sanitization, policy outside the model, least privilege |
| Evidence or reward poisoning          | Provenance, protected sources, robust statistics, independent fresh tests                |
| Hallucinated or overfit world program | Executable tests, falsifiers, uncertainty, counterexamples, decision-relevant proof      |
| Model herding or collusion            | Supplier/evidence diversity, Collective Dynamics Governor, independent verifier          |
| Verifier capture                      | Separation of duties, precommitted protocol, cross-verification, transparent incentives  |
| Supply-chain compromise               | Signed artifacts, SBOM/MBOM, pinned versions, isolated secrets, reproducible builds      |

| Threat                     | Principal control                                                           |
|----------------------------|-----------------------------------------------------------------------------|
| Data exfiltration          | Data minimization, policy-aware retrieval, egress controls, tool allowlists |
| Unauthorized action        | External Authority Envelope, action broker, approvals, rate/spend limits    |
| Drift or silent impairment | Live monitoring, proof expiry, authority contraction, renewal triggers      |
| Failed rollback            | Pre-deployment recovery test, reversible staging, immutable checkpoints     |

13.3 Alignment with established risk practice

The NIST AI Risk Management Framework organizes AI risk management around Govern, Map, Measure, and Manage and emphasizes validity, safety, security, resilience, accountability, transparency, explainability, privacy, and fairness.[21] Its Generative AI Profile extends the framework to risks intensified by generative systems.[22]

SUCCESSOR Ω does not replace applicable standards, regulation, professional judgment, cybersecurity controls, or sector-specific assurance. Its Chronicle and proof architecture are designed to make those controls operational at the mission and release level.

13.4 Functional resilience

The institution preserves validated mission function rather than accidental implementation structure. If a model, tool, or provider fails, the learning branch may search for an alternative composition. Research on self-organising digital circuits offers an instructive analogy: systems can reorganize structure while preserving function after damage.[20]

A repaired composition is a new release. Its inherited authority contracts until proportionate proof confirms the restored function.



**CONCEPTUAL INSTITUTIONAL PLATE 08.** *The mission institution isolates failure, substitutes suppliers, and re-proves restored function.*

**Preserve validated mission function, not accidental implementation structure. A repaired composition is a new release; continuity of service does not create continuity of proof or authority.**

### 13.5 Minimum production assurance case

Before A3 or A4 admission, every frozen release must have a current, customer-accepted production assurance case. Performance proof cannot compensate for a failed security, control, or recoverability gate.

The minimum assurance case contains:

- a current trust-boundary, data-flow, and deployment model;
- a threat model mapped to controls, evidence, residual risks, and named control owners;
- signed software, model, data, prompt, tool, licence, and dependency bills of materials;
- reproducible release construction and independent manifest verification;
- identity, secret, key-custody, rotation, revocation, and break-glass procedures;

- test evidence for prompt injection, malicious tool output, memory poisoning, data exfiltration, privilege escalation, and supply-chain compromise;
- deny-by-default action brokering, tool allowlists, network boundaries, approvals, and spend, rate, target, and time limits;
- monitoring coverage, anomaly thresholds, action-receipt integrity, escalation, and on-call ownership;
- successful suspend, rollback, restore, and declared supplier-substitution drills;
- recovery-time and recovery-point objectives appropriate to the mission; and
- incident classification, notification, evidence preservation, recovery, and re-proof triggers.

Loss of control-plane integrity, release identity, monitoring coverage, or rollback capability automatically contracts authority to the highest demonstrably safe level, including A0. Recovery creates a new operational state recorded in the Chronicle; it does not silently restore prior authority.

## 14 | Customer experience

---

The customer should never need to infer mission posture from agent conversation, disconnected dashboards, or technical logs. Mission Command exposes one authoritative state model shared by executives, operators, risk leaders, and technical auditors.

Its persistent command bar displays mission, serving release, proof, authority, alpha, security, rollback, and the next accountable decision. Authorized customer roles receive explicit controls to suspend operation, contract authority, require approval, initiate rollback, freeze evidence, and export the package. Elevated controls require role-appropriate authentication, separation of duties, and dual authorization where the Mission Constitution requires it.

The image shows a 'MISSION COMMAND' interface with a title bar 'ILLUSTRATIVE CUSTOMER VIEW • INDUSTRIAL RELIABILITY'. It features a grid of status boxes and several informational panels.

| MISSION COMMAND     |                        |                   |                     |
|---------------------|------------------------|-------------------|---------------------|
| MISSION<br>DOWNTIME | RELEASE<br>SHADOW-17   | PROOF<br>PENDING  | AUTHORITY<br>A1     |
| ALPHA<br>NOT EARNED | SECURITY<br>CONTROLLED | ROLLBACK<br>READY | NEXT<br>FRESH TRIAL |

|                                                                                                         |                                                                                                 |
|---------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------|
| <b>MY MISSION</b><br>Reduce unplanned downtime; safety constraints are non-compensable.                 | <b>RECOMMENDATION</b><br>Inspect the bearing within 48 hours; human approval required.          |
| <b>CURRENT UNDERSTANDING</b><br>Vibration and heat interact; confidence is medium; assumptions exposed. | <b>PROOF + ALPHA</b><br>Candidate frozen; evidence sealed; alpha not yet earned.                |
| <b>PREDICTION + FALSIFIER</b><br>Elevated 14-day risk; normalized load-corrected signal would falsify.  | <b>CURRENT AUTHORITY</b><br>Analyze and recommend only; no scheduling, purchasing, or dispatch. |
| <b>SECURITY + ROLLBACK</b><br>Evidence quarantine active; least control drill passed; rollback ready.   | <b>LATEST CHRONICLE</b><br>09:14 revised • 09:27 tests passed • 10:02 frozen • 10:05 sealed     |

|         |                    |                   |                |
|---------|--------------------|-------------------|----------------|
| SUSPEND | CONTRACT AUTHORITY | INITIATE ROLLBACK | EXPORT PACKAGE |
|---------|--------------------|-------------------|----------------|

**CUSTOMER INSTRUMENT 01.** An illustrative Mission Command surface makes release identity, proof currency, authority, alpha, security, rollback, falsifiers, approvals, and customer intervention visible in one signed institutional state.

Below the command bar, seven views answer the institution's operating questions.

## My Mission

What SUCCESSOR Ω is constituted to accomplish, what it must never do, and when its mandate expires.

## What It Currently Understands

Active world programs, assumptions, causal hypotheses, conflicts, uncertainties, and recent revisions - explained in plain language with inspection available underneath.

## What It Predicts

Expected outcomes, alternatives, confidence, validity horizon, and the evidence that could prove the prediction wrong.

## What It Recommends

The proposed action, expected value, supporting evidence, risks, alternatives, cost, and required approval.

## Proof Status

What has been demonstrated, by which frozen release, against which fresh evidence, by whom, within what limits, and until when.

## Current Authority

What the institution may observe, recommend, simulate, or execute now - including budgets, tools, approvals, and rollback readiness.

## Mission Alpha and Live Chronicle

Measured advantage over the current baseline, its confidence and impairment status, followed by every material hypothesis, revision, test, failure, proof, action, rollback, and succession event.

The interface progressively discloses detail. An executive can understand the status in minutes; an auditor can trace a claim to evidence, code, trial, and action receipt.

All role views read from the same signed institutional state. No dashboard may show proof as current while the Deployment Controller treats it as impaired, or show authority broader than the machine-enforced envelope. Every material action must be traceable from recommendation through evidence, approval, policy decision, tool call, outcome, and rollback state in no more than three navigation steps.

# 15 | Illustrative mission: industrial reliability

---

Consider a manufacturer seeking to reduce unexpected equipment downtime without compromising safety.

## Evidence

The customer supplies maintenance histories, sensor telemetry, work orders, operating procedures, failure reports, spare-part constraints, production costs, and safety rules. The Evidence Fabric records provenance, quality, time alignment, and rights.

## Mission Constitution

The Mission Compiler defines the target equipment, time horizon, outcome metric, false-negative cost, permitted interventions, safety constraints, human approvals, baseline process, relevant AI Beta, and proof protocol.

## World programs

Candidates synthesize executable hypotheses for failure mechanisms, degradation states, sensor interactions, maintenance effects, and production consequences. Each predicts what will happen and what observation would contradict it.

## Formation

The Gym replays historical episodes, creates rare-event scenarios, injects sensor faults, tests missing data, and challenges schedule/resource constraints. Competing candidates may specialize in early detection, robust scheduling, calibrated uncertainty, or low false-alarm burden.

## Proof

The exact challenger is frozen and tested against a protected later period or prospective shadow trial. It must exceed the incumbent and relevant AI Beta on the mission utility while passing every safety gate.

## Admission

The institution begins at A1, recommending maintenance with explanations and falsifiers. After sufficient fresh proof, it may reach A3 for reversible scheduling changes within defined plants, assets, budgets, and rollback rules. Safety-critical interventions remain explicitly authorized.

## Renewal

When equipment, operating conditions, or baseline capability changes, the Chronicle detects drift. A shadow successor revises the world programs and re-earns admission.

The customer has not purchased a generic chatbot. It has acquired a portable intelligence institution dedicated to its operating world.

# 16 | Mission selection and initial markets

---

## 16.1 Mission screen

The best initial missions have:

- repeated, high-value decisions;
- costly errors or material missed opportunities;
- proprietary or hard-to-assemble evidence;
- observable outcomes on a useful time horizon;
- a credible incumbent and relevant AI Beta baseline;
- historical replay, simulation, shadow operation, or staged deployment;
- separately measurable safety and economic constraints;
- bounded, reversible early actions; and
- a sponsor able to own the Mission Constitution.

Missions should be rejected or narrowed when outcomes are unobservable, evidence rights are unclear, evaluation can be gamed, causal identification is impossible, or safe rollback does not exist.

## 16.2 Natural starting domains

| Domain                              | Initial mission                                 | Early authority                                         | Proof emphasis                                                           |
|-------------------------------------|-------------------------------------------------|---------------------------------------------------------|--------------------------------------------------------------------------|
| <b>Industrial operations</b>        | Failure prediction and maintenance optimization | Recommend, then reversible scheduling                   | Downtime, false negatives, safety, parts and labour cost                 |
| <b>Supply chain and procurement</b> | Detect risk and improve sourcing decisions      | Recommend, simulate, bounded reversible orders          | Total landed cost, resilience, policy compliance, supplier concentration |
| <b>Cybersecurity</b>                | Triage and prioritize incidents                 | Observe/recommend; bounded containment only after proof | Detection quality, response time, false positives, blast radius          |
| <b>Financial decision systems</b>   | Improve bounded analytical or control decisions | Recommend with explicit approvals                       | Risk-adjusted outcome, compliance, calibration, tail loss                |
| <b>Scientific discovery</b>         | Generate and test overlooked hypotheses         | Research/simulation before real experiments             | Novelty, coherence, experimental yield, reproducibility                  |
| <b>Institutional strategy</b>       | Identify differentiated mission options         | Recommend and simulate                                  | Decision quality, scenario robustness, opportunity cost                  |

High-impact domains require sector-specific legal, professional, safety, and human oversight. SUCCESSOR Ω makes the boundary explicit; it does not remove it.

## 17 | Implementation and adoption roadmap

The deployment unit is one constituted mission, not an enterprise-wide promise of autonomy.

### Phase 0 | Mission underwriting

Define the mission, sponsor, baseline, value hypothesis, evidence rights, constraints, evaluator, authority ceiling, and stop conditions. Deliver a Mission Constitution and a go/no-go compilability decision.

### Phase 1 | Evidence and baseline

Build the Evidence Fabric, reproduce the incumbent, identify relevant AI Beta, establish data contracts, and create the first measurement harness. Deliver an evidence map, baseline report, threat model, and proof plan.

## Phase 2 | Candidate manufacturing

Open AGI Jobs, synthesize world programs, assemble competing architectures, build the Gym, and preserve a quality-diverse candidate archive. Operate entirely in research, replay, sandbox, or shadow modes.

## Phase 3 | Freeze and fresh proof

Freeze the exact challenger, protect fresh evidence, execute the independent protocol, and issue proof or failure receipts. No production authority is implied by Foundry or Gym scores.

## Phase 4 | Accountable admission

Grant A0-A4 authority by scope, connect the Deployment Controller, rehearse rollback, and begin live monitoring. Start at the lowest level that produces useful learning.

## Phase 5 | Renewal and succession

Detect drift and impairment, re-underwrite the mission, manufacture a shadow descendant, reprove, and explicitly admit or reject the successor.

An illustrative first bounded proof engagement may run for 12-20 weeks when evidence and evaluation are already accessible. Safety-critical, regulated, long-horizon, or low-frequency missions can require substantially longer. The proof obligation determines the schedule - not a marketing calendar.

## Operating roles

| Role                        | Accountability                                                                 |
|-----------------------------|--------------------------------------------------------------------------------|
| <b>Mission owner</b>        | Defines value, constraints, and authority; accepts operational accountability. |
| <b>Evidence steward</b>     | Governs provenance, rights, quality, retention, and access.                    |
| <b>Foundry operator</b>     | Manufactures candidates and discloses methods and conflicts.                   |
| <b>Gym operator</b>         | Creates training and stress experience without issuing final proof.            |
| <b>Independent verifier</b> | Executes protected trials and signs proof/failure receipts.                    |
| <b>Policy authority</b>     | Grants, contracts, expires, and revokes the Authority Envelope.                |
| <b>Deployment operator</b>  | Maintains tools, monitoring, recovery, and incident response.                  |
| <b>Chronicle custodian</b>  | Preserves integrity, access, export, and succession continuity.                |

## 18 | Research integration strategy

The kernel must remain stable while research components compete behind explicit interfaces. Integration follows four evidence levels.

### Level I | Constitutional primitives

These are product laws rather than claims borrowed from any paper:

- mission compilation;
- sealed AGI Jobs;
- claimant-verifier separation;
- exact-release freeze;
- fresh proof;
- bounded authority;
- customer-controlled Chronicle; and
- no inherited proof or authority.

### Level II | Core candidate mechanisms

| Research direction                          | SUCCESSOR Ω use                                                                                   | Claim discipline                                                                                                |
|---------------------------------------------|---------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------|
| <b>Program-synthesized world models</b>     | Executable state, transition, prediction, and planning hypotheses                                 | Programs are testable hypotheses, not truth. <a href="#">[6]</a> <a href="#">[7]</a> <a href="#">[25]</a>       |
| <b>DUPLEX</b>                               | Neural semantic extraction plus deterministic symbolic planning; expensive repair only on failure | Evidence from planning benchmarks, not enterprise validation. <a href="#">[8]</a>                               |
| <b>EnvHarness</b>                           | Adaptive Gym components that target diagnosed weaknesses while preserving underlying logic        | Teacher only; never final proof. <a href="#">[9]</a>                                                            |
| <b>Quality diversity</b>                    | Preserve high-performing candidates across behavioral niches                                      | Add Pareto and hard-gate governance; pyribs is not itself an institutional evaluator. <a href="#">[11]</a>      |
| <b>Lightweight multi-model coordination</b> | Route specialist roles across replaceable suppliers                                               | Internal critics are not independent verifiers. <a href="#">[13]</a> <a href="#">[14]</a>                       |
| <b>Agent interaction dynamics</b>           | Detect herding, polarization, and correlated confidence                                           | Safety signal derived from simulated populations. <a href="#">[15]</a>                                          |
| <b>Virtual agent economies</b>              | Bounded work orders, budgets, reputation, and auditable exchange                                  | Economic coordination remains subject to policy and dispute controls. <a href="#">[16]</a> <a href="#">[17]</a> |

## Level III | Experimental premium capabilities

- SPADE-like environment generation for frontier-targeted experience, never final proof.[10]
- Large Causal Model-style hypothesis graphs, with every edge labelled proposed, supported, tested, falsified, contested, or retired.[18]
- Alien Space-style search for coherent but cognitively unavailable combinations, converted into executable hypotheses and AGI Jobs before investment.[19]
- Self-organizing repair that preserves validated function but contracts authority until the new release is re-proven.[20]
- OpenEvolve-like program evolution as an optional Foundry engine, treated as software evidence rather than a final evaluator.[12]

## Level IV | Replaceable adapters

Agent SDKs, browser-use systems, scientific skill libraries, data-science agents, local models, frontier APIs, and specialized solvers may improve a Mission Pack. None is permanent. Each receives a licence record, security profile, evaluation history, cost model, and substitution plan.

## Excluded from the kernel

The kernel excludes leaked system-prompt repositories, generic "deep thinking" metaprompts as evidence, claims of consciousness, social-media capability claims without reproducible evaluation, unanchored real-world self-play, self-authored narrative identity as truth, and any mechanism that lets the claimant manufacture its own final proof.

# 19 | Market and category strategy

---

## 19.1 Adjacent market context

Gartner forecasts total worldwide AI spending of US\$2.59 trillion in 2026, with more than 45% driven by AI infrastructure.[2] It separately forecasts AI agent software spending of US\$206.5 billion in 2026 and US\$376.3 billion in 2027.[1] These figures establish scale and velocity around the category. They do not establish a SUCCESSOR Ω total addressable market.

The more important signal is structural:

- AI use is broad, but enterprise financial impact remains concentrated.[3]
- Agentic projects face cancellation because value, cost, and control are unresolved.[4]
- Agentic arbitrage shifts software value from interfaces toward outcomes, customer context, and institutional memory.[5]

Customer-Owned Mission Intelligence is designed for that intersection.

## 19.2 Initial economic beachhead

The initial market is narrower than the full agent software market: organizations with proprietary evidence, repeated high-value decisions, costly errors, measurable outcomes, and a serious need for auditability and continuity.

The category should be sold against a mission business case, not a seat count. Commercial units may include mission underwriting, Foundry formation, independent proof, admitted deployment, customer-controlled operation, and periodic renewal.

## 19.3 Why this paper does not claim a top-down TAM

Applying an assumed capture percentage to an adjacent agent-software forecast would produce arithmetic, not a market model. A credible category forecast requires observed willingness to pay, repeatable delivery economics, proof costs, renewal behavior, mission-level pricing, sales-cycle evidence, and a clear distinction between software, services, and independently governed verification.

The initial commercial test is therefore bottom-up: select missions with measurable economic exposure, prove net advantage, measure the cost of formation and verification, and observe whether customers renew and expand the institution. Market size should be earned from those units rather than inferred from a large adjacent denominator.

## 19.4 Positioning

Do not lead with "deep learning-guided synthesis of symbolic world models." That is a technical engine.

Lead with:

**Your organization's own intelligence - built from your evidence, independently proven against your mission, and yours to operate, inspect, and evolve.**

Then reveal the moat:

**Powered by lightweight executable world programs that continuously learn, predict, test, falsify, and improve through a governed successor process.**

## 19.5 Category comparison

| Adjacent category        | Typical primary asset            | Institutional distinction proposed by SUCCESSOR Ω                                        |
|--------------------------|----------------------------------|------------------------------------------------------------------------------------------|
| Foundation model or API  | General cognitive capability     | Treats the model as a replaceable supplier to a persistent mission layer.                |
| RAG or enterprise search | Retrieved organizational context | Converts selected evidence into versioned hypotheses, tests, proof, and governed memory. |

| Adjacent category               | Typical primary asset                 | Institutional distinction proposed by SUCCESSOR Ω                                                                    |
|---------------------------------|---------------------------------------|----------------------------------------------------------------------------------------------------------------------|
| Agent platform                  | Workflow and tool execution           | Separates candidate orchestration from independent final proof and customer-granted authority.                       |
| Fine-tuned model                | Adapted model behavior                | Attaches claims to a complete frozen institution, not model weights alone.                                           |
| Knowledge graph or digital twin | Structured domain representation      | Requires falsifiers, fresh comparative proof, authority, and succession around the representation.                   |
| MLOps or AI governance          | Model lifecycle and control processes | Extends control to mission alpha, exact-release admission, action receipts, and customer continuity.                 |
| SUCCESSOR Ω                     | Customer-owned mission institution    | Integrates executable hypotheses, protected proof, bounded authority, supplier substitution, and the Live Chronicle. |

These are category tendencies, not universal statements about every offering. SUCCESSOR Ω can compose foundation models, retrieval, agent platforms, digital twins, and governance systems behind its mission, proof, authority, and ownership interfaces.

## 20 | Maturity disclosure

This white paper specifies a target institutional architecture. It does not claim that the cited literature or this document demonstrates a completed end-to-end SUCCESSOR Ω. The research base supports converging component mechanisms; integrated enterprise realization remains an ambitious engineering, verification, and governance program.

Every product release should publish a subsystem maturity register using the following states:

| State                   | Required evidence                                                                            |
|-------------------------|----------------------------------------------------------------------------------------------|
| Specified               | Interfaces, invariants, threat model, and acceptance criteria documented.                    |
| Prototyped              | An executable implementation demonstrates the intended control path.                         |
| Validated               | Reproducible tests show the subsystem meets predeclared criteria in representative settings. |
| Independently evaluated | A separated verifier reproduces material claims against protected evidence.                  |
| Production-admitted     | Customer governance grants a bounded Authority Envelope with monitoring and recovery.        |

The named constructs introduced and integrated by this paper include Customer-Owned Mission Intelligence, Mission Constitution, Mission Alpha, Alpha Foundry Ω, Independent Proof Network, Authority Envelope, Specialist ASI admission, Live Chronicle, and the rule that descendants inherit neither proof nor authority. Their contribution lies in institutional integration and the customer ownership boundary, not in a claim that every technical ingredient is new.

## 21 | Limitations, open problems, and non-claims

---

A definitive architecture must state what can fail.

### 21.1 World programs can be precisely wrong

Executability creates inspectability and testability, not truth. Partial observability, confounding, distribution shift, missing variables, and incorrect abstractions can produce programs that pass familiar tests and fail where it matters.

### 21.2 Causal claims are difficult

LLM-generated causal links may be plausible, ambiguous, fragmented, or conflicting.<sup>[18]</sup> SUCCESSOR  $\Omega$  treats them as typed hypotheses until appropriate evidence or experiments support them. In some missions, causal identification may remain impossible.

### 21.3 Independent proof is costly

Protected evidence, competent verifiers, reproducibility, and long-horizon outcomes add cost and time. The architecture should calibrate proof to consequence without weakening the claimant-verifier boundary.

### 21.4 Portability has real limits

External model weights, proprietary APIs, regulated data, and third-party licences may constrain self-hosting or substitution. The paper promises a portable mission layer only to the level contractually and technically demonstrated.

### 21.5 Lightweight is measurable, not rhetorical

The durable mission layer can be compact relative to a frontier model, but total operating footprint may include external inference, local accelerators, storage, verification, and integration. Every deployment should report package size, runtime requirements, inference cost, energy where measurable, latency, and supplier dependencies.

### 21.6 Alpha is not guaranteed

The Foundry may fail to produce a candidate that exceeds the baseline. A rigorous failure receipt is a valid outcome and can prevent further waste. SUCCESSOR  $\Omega$  exists only after proof and admission, not because a candidate was manufactured.

## 21.7 Specialist ASI is not universal ASI

The term records bounded mission superiority under an explicit test. It must never be presented as universal superintelligence, general autonomy, perfect knowledge, or authority outside the proof envelope.

## 21.8 Human accountability remains

The institution can improve evidence, consistency, and control. It cannot eliminate the customer's legal, fiduciary, professional, safety, or ethical responsibilities. Consequential missions require named human and organizational accountability.

## 22 | Claims-to-proof covenant

The corporate promise becomes credible only when every major claim has a test.

| Claim                    | Required evidence                                                                                                     |
|--------------------------|-----------------------------------------------------------------------------------------------------------------------|
| "Customer-owned"         | Contractual rights map plus independent export-and-rebuild test                                                       |
| "Lightweight"            | Measured package size, runtime footprint, inference economics, and dependency inventory                               |
| "Portable"               | Restore in a target environment and substitute a declared supplier without losing policy or Chronicle continuity      |
| "Neural-symbolic"        | Inspectable typed programs materially used in prediction/planning, not decorative generated code                      |
| "Proven alpha"           | Frozen release, fresh protected evidence, best-baseline comparison, confidence bounds, full cost/risk adjustments     |
| "Independent proof"      | Verifier separation, precommitted protocol, conflict disclosure, signed proof/failure receipt                         |
| "Bounded authority"      | Machine-enforced envelope, least privilege, approval checks, expiry, monitoring, and successful rollback drill        |
| "Compounding"            | Demonstrated reuse of Chronicle/evidence that improves a descendant while the descendant independently re-earns proof |
| "Supplier-substitutable" | At least one successful supplier substitution test for the claimed role                                               |
| "Operating institution"  | Live monitoring, action receipts, incident process, recovery test, and named accountable owners                       |

The covenant prevents marketing language from outrunning evidence. If a claim cannot be tested, it is narrowed or removed.

## 23 | Canonical operating cycle



**CONCEPTUAL INSTITUTIONAL PLATE 09.** *The Live Chronicle binds continuity to evidence and governs succession.*

**No descendant inherits proof or authority automatically. Every material successor is versioned, frozen, freshly challenged, and separately admitted.**

The institution compounds through a governed sequence:

**Chronicle → Detect Impairment → Re-underwrite → New AGI Jobs → New Gym → Manufacture Candidates → Freeze → Fresh Proof → Renew SUCCESSOR  $\Omega$  → Compound Alpha**

Later missions may benefit from inherited evidence structures, skills, negative results, and Chronicle context. No descendant automatically inherits its predecessor's proof or authority.

The invariant is succession by demonstrated function, not continuity by assertion.

## Conclusion | Own the institution

The central enterprise AI decision is not whether to rent frontier intelligence. Organizations will, and often should. The decision is whether the mission knowledge produced through that intelligence becomes a durable customer asset or remains trapped in a changing supplier stack.

Neural-Symbolic SUCCESSOR  $\Omega$  defines a different relationship:

- rent cognition from the best available suppliers;
- compile the mission and evidence into an owned institutional layer;
- represent understanding as executable, falsifiable programs;
- manufacture multiple candidate successors rather than trusting one architecture;
- teach in adaptive environments without letting the teacher certify;
- freeze the exact release before examination;
- determine what the frozen release demonstrated within its proof envelope through independent fresh proof;
- convert only demonstrated advantage into bounded, expiring authority;
- preserve every material event in the customer-controlled Live Chronicle; and
- require every descendant to earn admission again.

The canonical formulation is:

**Neural-Symbolic SUCCESSOR  $\Omega$  is a lightweight, portable, customer-owned intelligence institution that navigates the customer's mission world by continuously generating, executing, testing, falsifying, and evolving symbolic programs guided by deep learning and grounded in fresh evidence. It converts rented frontier capability into independently proven, mission-specific, compounding intelligence alpha.**

It can rent intelligence without renting its institutional mind.

Beta is rented. Alpha is manufactured. The resulting SUCCESSOR  $\Omega$  is theirs.

## Appendix A | Canonical package manifest

An implementation may use different filenames, but a complete export should carry equivalent semantics.

| Path                   | Contents                                                                                                                                   |
|------------------------|--------------------------------------------------------------------------------------------------------------------------------------------|
| <b>mission/</b>        | Constitution, objectives, constraints, metrics, baseline, authority ceiling                                                                |
| <b>evidence/</b>       | Bindings, provenance graph, rights, freshness, transformations, optional permitted evidence                                                |
| <b>world_programs/</b> | State, transition, causal, prediction, planning, uncertainty, and failure programs                                                         |
| <b>skills/</b>         | Skills, workflows, prompts, deterministic components, adapters                                                                             |
| <b>jobs/</b>           | Sealed AGI Jobs, budgets, evaluators, acceptance and dispute rules                                                                         |
| <b>gym/</b>            | Environment contracts, stages, chains, curricula, attack suites, training receipts                                                         |
| <b>tests/</b>          | Unit, integration, regression, falsification, stress, transfer, and non-revealing protected-test commitments or verifier-approved metadata |
| <b>proofs/</b>         | Proof/failure receipts, verifier declarations, baselines, statistics, expiry                                                               |
| <b>policy/</b>         | Authority Envelopes, approvals, prohibitions, resource and data policies                                                                   |
| <b>deployment/</b>     | Tool contracts, monitoring, incident, rollback, recovery procedures                                                                        |
| <b>chronicle/</b>      | Append-only events, checkpoints, impairments, renewals, succession lineage                                                                 |
| <b>manifests/</b>      | Release hash, SBOM, model/tool BOM, licences, signatures, reproducibility metadata                                                         |
| <b>explain/</b>        | Executive summaries, customer views, diagrams, model/program cards                                                                         |

## Appendix B | Minimum proof receipt

| Field                      | Required content                                                  |
|----------------------------|-------------------------------------------------------------------|
| <b>receipt_id</b>          | Globally unique proof or failure receipt identifier               |
| <b>mission_id</b>          | Exact Mission Constitution identifier and hash                    |
| <b>release_id</b>          | Frozen candidate identifier and cryptographic manifest            |
| <b>baseline_id</b>         | Incumbent and relevant AI Beta configurations                     |
| <b>verifier</b>            | Identity, competence, independence, conflicts, signatures         |
| <b>evidence_commitment</b> | Protected evidence description and commitment without leakage     |
| <b>protocol</b>            | Precommitted metrics, gates, sample, budget, tools, and analysis  |
| <b>results</b>             | Point estimates, confidence bounds, critical failures, exclusions |

| Field                   | Required content                                                          |
|-------------------------|---------------------------------------------------------------------------|
| <b>economics</b>        | Cost, latency, compute, human burden, switching and operation assumptions |
| <b>scope</b>            | Population, environment, tools, geography, time, and mission limits       |
| <b>validity</b>         | Issue date, expiry, monitoring and impairment triggers                    |
| <b>authority</b>        | Maximum authority justified, not automatically granted                    |
| <b>chronicle_anchor</b> | Link to the tamper-evident, append-only trial and admission record        |

## Appendix C | Minimum Authority Envelope

| Field             | Required content                                                      |
|-------------------|-----------------------------------------------------------------------|
| <b>subject</b>    | SUCCESSOR identity and frozen release                                 |
| <b>mission</b>    | Permitted mission and explicit exclusions                             |
| <b>resources</b>  | Data, tools, models, networks, funds, compute, rate and time limits   |
| <b>actions</b>    | Permitted, approval-required, and prohibited action classes           |
| <b>targets</b>    | Systems, assets, accounts, counterparties, and geographic scope       |
| <b>conditions</b> | Required proof, monitoring, confidence, and environmental state       |
| <b>expiry</b>     | Absolute time and event-driven expiration                             |
| <b>escalation</b> | Human and system escalation routes                                    |
| <b>rollback</b>   | Reversal method, checkpoint, maximum recovery time, responsible owner |
| <b>suspension</b> | Kill-switch actors, automated triggers, incident thresholds           |
| <b>receipts</b>   | Required pre-action and post-action records                           |

## Appendix D | Decision gates

| Gate                         | Pass condition                                                        | Failure outcome                      |
|------------------------------|-----------------------------------------------------------------------|--------------------------------------|
| <b>Mission compilability</b> | Observable outcomes, evaluator, evidence, constraints, safe test path | Reject, narrow, or research-only     |
| <b>Evidence readiness</b>    | Rights, provenance, quality, freshness, and security adequate         | Remediate evidence; no Foundry scale |
| <b>Baseline integrity</b>    | Incumbent and relevant AI Beta reproduced fairly                      | Rebuild comparison                   |
| <b>Candidate readiness</b>   | Tests pass; known failure modes disclosed                             | Repair or archive                    |

| Gate                 | Pass condition                                                | Failure outcome                         |
|----------------------|---------------------------------------------------------------|-----------------------------------------|
| Freeze integrity     | Exact manifest complete and reproducible                      | Trial invalid                           |
| Proof sufficiency    | Fresh evidence, independent verifier, all hard gates pass     | No Specialist ASI claim                 |
| Admission            | Policy authority grants scoped envelope                       | Remain observe/recommend/research only  |
| Deployment readiness | Monitoring, rollback, recovery, and accountable owners tested | No consequential action                 |
| Renewal              | New release independently re-proves required claims           | Existing authority expires or contracts |

## References

---

- [1] Gartner. "Gartner Says Autonomous Business and AI Layoffs May Create Budget Room, but Do Not Deliver Returns." 5 May 2026. [View source](#)
- [2] Gartner. "Gartner Forecasts Worldwide AI Spending to Grow 47% in 2026." 19 May 2026. [View source](#)
- [3] McKinsey & Company. "The State of AI in 2026: On the Road to ROI." 25 August 2026. [View source](#)
- [4] Gartner. "Gartner Predicts Over 40% of Agentic AI Projects Will Be Canceled by End of 2027." 25 June 2025. [View source](#)
- [5] Gartner. "Gartner Says \$234 Billion in Enterprise Application Software Spend Is at Risk from Agentic AI." 1 July 2026. [View source](#)
- [6] Dainese, N., Merler, M., Alakuijala, M., and Marttinen, P. "Generating Code World Models with Large Language Models Guided by Monte Carlo Tree Search." arXiv:2405.15383; NeurIPS 2024. [View source](#)
- [7] Ahmed, Z. et al. "Synthesizing World Models for Bilevel Planning." arXiv:2503.20124; TMLR 2025. [View source](#)
- [8] Hua, K., Wang, D., Gu, Y., and Ma, X. "DUPLEX: Agentic Dual-System Planning via LLM-Driven Information Extraction." arXiv:2603.23909, 2026. [View source](#)
- [9] Huang, C. et al. "EnvHarness: Awakening Static Worlds for Agent Learning." arXiv:2608.19880, 2026. [View source](#)
- [10] Liu, B. et al. "SPADE: Self-Play in Adaptive Synthetic Executable Environments." arXiv:2608.19197, 2026. [View source](#)
- [11] Tjanaka, B. et al. "pyribs: A Bare-Bones Python Library for Quality Diversity Optimization." GECCO 2023; arXiv:2303.00191. [View source](#)
- [12] OpenEvolve. Open-source evolutionary coding agent and program-optimization framework. [View source](#)
- [13] Xu, J. et al. "TRINITY: An Evolved LLM Coordinator." ICLR 2026; arXiv:2512.04695. [View source](#)
- [14] Chi, Z. et al. "The Era of Agentic Organization: Learning to Organize with Language Models." arXiv:2510.26658, 2025. [View source](#)
- [15] El, B. et al. "Physics of Agents: Statistical Mechanics Predicts Collective Behavior of AI Agents." arXiv:2608.16578, 2026. [View source](#)
- [16] Tomašev, N. et al. "Virtual Agent Economies." arXiv:2509.10147, 2025. [View source](#)
- [17] Tomašev, N. et al. "Distributional AGI Safety." arXiv:2512.16856, 2025. [View source](#)
- [18] Mahadevan, S. "Large Causal Models from Large Language Models." arXiv:2512.07796, 2025. [View source](#)
- [19] Artiles, A. H. et al. "The Alien Space of Science: Sampling Coherent but Cognitively Unavailable Research Directions." arXiv:2603.01092, 2026. [View source](#)
- [20] Barylli, M. et al. "Self-Organising Digital Circuits." arXiv:2608.02606, 2026. [View source](#)
- [21] National Institute of Standards and Technology. "Artificial Intelligence Risk Management Framework (AI RMF 1.0)." NIST AI 100-1, 2023. [View source](#)
- [22] National Institute of Standards and Technology. "Artificial Intelligence Risk Management Framework: Generative Artificial Intelligence Profile." NIST AI 600-1, 2024. [View source](#)
- [23] OWASP. "LLM01:2025 Prompt Injection" and "AI Agent Security Cheat Sheet." [View source](#)
- [24] MITRE. "MITRE ATLAS." Adversarial threat knowledge base for AI systems. [View source](#)
- [25] Tang, H., Key, D., and Ellis, K. "WorldCoder, a Model-Based LLM Agent: Building World Models by Writing Code and Interacting with the Environment." arXiv:2402.12275, 2024. [View source](#)
- [26] Aguilar Martín, J. "When a Verified World Model Still Loses: Play-Adequacy vs Prediction-Accuracy in LLM-Synthesized Code World Models." arXiv:2607.14169, 2026. [View source](#)
- [27] Ahmed, Z. et al. "Learning Abstractions for Hierarchical Planning in Program-Synthesis Agents." arXiv:2602.00929, 2026. [View source](#)
- [28] Lehrach, W. et al. "Code World Models for General Game Playing." arXiv:2510.04542, 2025. [View source](#)