GoalOS Global Launch Ω

The Verified Operating System for Building Anything, Anywhere

A proof-governed architecture for institution formation, global opportunity search, capital orchestration, authorized execution, selective memory, and compounding productive capacity.

The world is your launch stack.

GoalOS converts a consequential objective into a formed, financed, evidenced, governed, continuously monitored, and recursively improvable operating institution.

BY MONTREAL.AI

MONTREAL.AI-RI-GLΩ-2026-04

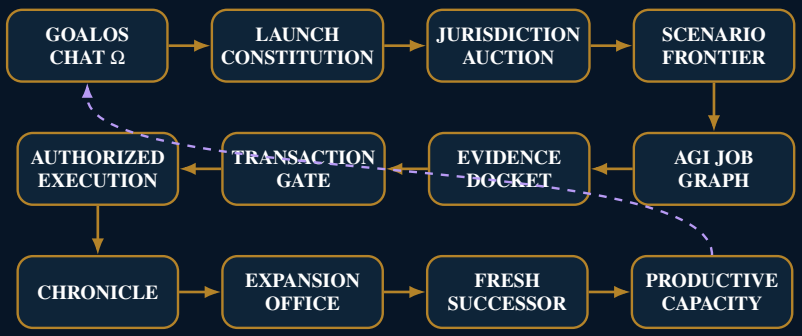
Version 4.0.0 | 24 July 2026

Research monograph • Institutional specification • Falsifiable proof programme

INSTITUTIONAL VISION

From Objective to Operating Institution

A single proof-governed chain from ambition to productive capacity.



Search broadly. Execute exactly. Admit selectively. Compound verified value.

Conceptual architecture. This page is not evidence of realized commercial, legal, fiscal, or operational capability.

This paper introduces *GoalOS Global Launch* Ω, a proof-governed institution compiler for converting a consequential objective into a formed, financed, authorized, executed, continuously monitored, and recursively improvable operating institution. The system treats global launch as a partially observable, multi-objective, path-dependent control problem over markets, jurisdictions, talent, capability, capital, public support, infrastructure, law, and time. It does not ask for one "best country." It decomposes a mission into eight functions—sell, hire, perform R&D, own capability, raise capital, obtain public support, deploy infrastructure, and realize value—and searches concentrated, hybrid, and functionally distributed architectures.

The resulting scenario frontier is evaluated through expected verified value, tail risk, institutional regret, proof debt, coordination cost, reversibility, authority, and capital-to-capacity conversion. GoalOS uses bounded AGI Jobs to perform graph-structured work; every material transition is conditioned on attributable evidence, current sources, exact facts, qualified authority where required, and explicit rollback. Completed work is not automatically memory. Only narrow, rights-cleared, independently challenged capability may enter Chronicle and influence a fresh generation.

The paper contributes: (i) a formal world and belief model; (ii) an institution-compiler mapping; (iii) a positive-upside opportunity-search policy; (iv) proof-carrying actions and a time-bounded authority lattice; (v) an institutional digital twin; (vi) a value-of-proof allocation rule; (vii) trajectory-level security controls for persistent agents; (viii) a capital-to-productive-capacity compounding model; and (ix) LaunchBench Ω, a falsifiable matched-pair benchmark comparing expert teams, static model reports, ungated agents, and proof-governed GoalOS missions.

The long-run thesis is not merely a better formation service. It is an institution capable of converting validated intelligence into recurring value, capital, compute, energy, laboratories, industry, infrastructure, and progressively greater verified productive capacity. This paper specifies the architecture and tests required before those claims may be earned; it does not claim that they have already been achieved.

Keywords: agentic AI; institution formation; jurisdiction intelligence; public funding; fiscal optimization; AGI Jobs; proof governance; graph engineering; capital allocation; robust optimization; productive capacity; recursive improvement.

Claim Boundary

This paper presents a testable institutional architecture, commercialization thesis, and research programme. It does not claim that GoalOS Global Launch has achieved product-market fit, binding legal or tax authority, universal jurisdictional coverage, guaranteed public funding, autonomous regulated practice, continuously current global intelligence, or civilization-scale productive capacity. Any binding formation, filing, transfer, financing, employment, tax, regulatory, procurement, or infrastructure action requires current official sources, exact facts, qualified responsible professionals, scoped organizational authority, and applicable transaction receipts.

Document Identity and Reading Rule

Author: MONTREAL.AI. **Edition:** Sovereign Intelligence Institution, version 4.0.0. **Status:** preprint, institutional design paper, and falsifiable proof programme. Conceptual diagrams, operating cases, price ranges, and scale scenarios are design hypotheses unless explicitly identified as realized evidence. The governing reading rule is: *candidate intelligence may search expansively; institutional action remains fail-closed.*

EXECUTIVE SYNTHESIS

The Institution, the Theorem, the Proof Programme, and the Horizon

The category is not business formation. It is **institution formation**: determining what institution should exist, where each function belongs, how it should be financed, which evidence remains unresolved, who may authorize action, and how the architecture should adapt after launch.

More intelligence is valuable when it is allocated to consequential uncertainty, alternative generation, independent challenge, and proof. It is dangerous when greater capability is confused with greater authority. GoalOS therefore maximizes bounded search while minimizing ambiguity of permission.

No entity, IP transfer, licence, financing, option grant, intercompany agreement, regulated filing, major procurement, or infrastructure SPV proceeds merely because a model recommends it. Applicable evidence, current sources, qualified authority, and execution receipts must pass.

The durable asset is not raw output. It is rights-cleared capability whose evidence, scope, failure conditions, expiry, and transferability are known. Only verified contribution may be reinvested into productive capacity.

GoalOS may be its own first customer: it may constitute, compare, sell, execute, record, and improve its own institutional architecture. It may not be its own sole evaluator, professional authority, source of truth, or final judge of realized value.

Not merely something launched. An institution earned. A greater capacity to build proven.

Contents

Institution Formation

The world contains extraordinary resources. The scarce capability is coordinating them into one defensible institution.

1. Introduction

Launching an institution is not one task. It is a coupled system of market access, legal formation, ownership, talent, research, capability, financing, public support, procurement, infrastructure, regulatory authority, and continuing governance. Yet the software and advisory market presents these components as isolated products. Company-formation platforms can automate a specific incorporation path; employer-of-record platforms can hire workers in jurisdictions where a customer lacks an entity; cap-table platforms can administer equity and private capital; professional databases can retrieve law and guidance; grant platforms can surface programmes; and advisers can study bespoke questions. Each component is useful, but no component begins from the entire objective.

The fragmentation is visible in the maturity of existing point solutions. Stripe Atlas reports that more than 100,000 founders have used its Delaware incorporation workflow and that founders in more than 140 countries have adopted the service (Stripe, 2026). Remote advertises employer-of-record coverage in more than 90 countries, while Deel reports owned infrastructure and entities in more than 100 countries (Remote, 2026; Deel, 2025). Carta combines cap-table management, financing, valuations, compliance, contracts, and fund administration in one private-capital platform (Carta, 2026). These systems demonstrate substantial demand for individual launch functions. They also reveal the category gap: a sponsor still must determine which combination should exist, where each function belongs, what should happen first, which assumptions remain unresolved, and which transactions may proceed.

The category is not business formation.
It is institution formation.

GoalOS Global Launch begins with a different question:

> **What must be built, under whose authority, with which capital, in which jurisdictions, through which sequence of bounded missions, and with what evidence that the resulting institution works?**

The paper makes six contributions:

1. It defines a *functional jurisdiction architecture* in which eight launch functions are optimized independently rather than forced into one jurisdiction.
2. It introduces a *scenario frontier* that generates concentrated, hybrid, and distributed institutional designs rather than a single opaque recommendation.
3. It formalizes a *proof-gated transaction model*: recommendations do not become authority; transactions remain blocked until required evidence and responsible approvals exist.
4. It integrates *AGI Jobs* and graph engineering into launch execution, separating independent work, true dependencies, verification edges, authority edges, and governance edges.
5. It defines a *Chronicle and successor mechanism*: completed work does not automatically become institutional learning; only admitted capability may influence a fresh generation.
6. It presents a commercialization model in which one-time launch missions convert into a recurring Global / Sovereign Expansion Office and, eventually, protected capability and network economics.

2. Frontier Context: Intelligence Is Becoming Abundant; Institutional Authority Is Not

The cost of generating candidate analysis is falling rapidly. General-purpose models can research, synthesize, draft, code, compare, and sustain long chains of work at a scale that would previously have required large multidisciplinary teams. This expands the feasible search space of institution design: more jurisdictions can be

compared, more capital structures simulated, more funding routes screened, more vendor combinations tested, and more counterexamples sought before an irreversible commitment.

The same capability creates a deeper governance problem. A model may discover a technically effective path that violates the sponsor's intent, exceeds its permission, uses the wrong data, relies on stale law, exploits a surrounding system, or optimizes a proxy that destroys the real objective. Persistent agents make this a trajectory problem rather than a single-output problem. The relevant question is not only whether one answer looks correct; it is whether the complete sequence of observations, tool calls, communications, transfers, approvals, and state changes remained within mandate.

Maximize strategic intelligence. Minimize ambiguity of authority.

GoalOS therefore separates four propositions that are often collapsed:

1. **Capability:** the system may be able to discover or execute a path.
2. **Truth:** the path may or may not be supported by current evidence.
3. **Authority:** the system may or may not be permitted to take the path.
4. **Value:** the path may or may not produce the sponsor's intended outcome after full cost, delay, risk, and externalities.

The institutional architecture must reward positive discovery without turning discovery into unilateral permission. Broad candidate search is therefore paired with exact tool scopes, budgets, data classes, network boundaries, professional dependencies, and explicit stop conditions. This is a generalization of AI guardrails into a constitutional operating model: the system does not merely screen prompts and outputs; it governs trajectories, transactions, memory, and capital.

2.1. Scarcity moves from cognition to proof and authority

As candidate cognition becomes cheaper, economic scarcity moves toward:

- current, attributable, fact-specific evidence;
- licensed and responsible professional judgment;
- contractual and public authority;
- trusted execution environments;
- rights to reuse methods and outcomes;
- reliable realization of cash, capability, and infrastructure;
- independent evaluation of whether a result truly worked.

This is the institutional meaning of the capability–price lag: an apparent abundance of intelligent output coexists with scarcity in the layers that permit society to rely on it. A durable business therefore cannot be built only by generating more prose. It must convert candidate intelligence into verified action and verified action into productive capacity.

3. The World as a Programmable Launch Stack

The phrase "the world is your launch stack" is not a claim that jurisdictions are interchangeable components or that law can be abstracted away. It means that global resources can be represented as a dynamic, constrained field of opportunities and obligations. Markets, talent pools, laboratories, tax systems, public programmes, treaty networks, infrastructure, professional providers, capital sources, and regulatory permissions can be modeled as interoperating modules only after their exact constraints are made explicit.

Let the global resource field at time t be

$$\mathcal{W}_t = (\mathcal{J}_t, \mathcal{M}_t, \mathcal{T}_t, \mathcal{C}_t, \mathcal{P}_t, \mathcal{I}_t, \mathcal{L}_t, \mathcal{R}_t),$$

where $\mathcal{J}$ denotes jurisdictions, $\mathcal{M}$ markets, $\mathcal{T}$ talent, $\mathcal{C}$ capital, $\mathcal{P}$ public support, $\mathcal{I}$ infrastructure, $\mathcal{L}$ legal and institutional constraints, and $\mathcal{R}$ rights and reusable capability. Each object is time-dependent; programmes open and close, laws change, source confidence expires, providers improve or fail, and the sponsor's own facts evolve.

A launch architecture is therefore not a static map. It is a controlled hypothesis about how the sponsor can convert elements of $\mathcal{W}_t$ into an operating institution before the hypothesis becomes stale.

Global-Stack Law

A global resource becomes part of the launch stack only when its access conditions, cost, timing, rights, authority, dependencies, failure modes, and reversibility are sufficiently known for the mission at hand.

3.1. Functional allocation rather than jurisdiction worship

The best market jurisdiction may be a poor R&D jurisdiction. The best R&D jurisdiction may be unsuitable for project finance. The strongest capital market may impose control or disclosure costs that conflict with the mission. The system therefore assigns functions independently, then penalizes the coordination, substance, tax, legal, and unwind costs created by fragmentation.

The resulting architecture can be concentrated, hybrid, or distributed. GoalOS should prefer the least irreversible structure capable of proving the next consequential uncertainty. This is a real-options discipline: defer costly irreversible commitments until the evidence required to justify them exists (Dixit and Pindyck, 1994).

4. The Fragmented Launch Stack

4.1. Point solutions optimize local tasks

Existing applications generally optimize a local object:

Table 1: The fragmented launch market and the GoalOS reframing.

Application class	Primary object	Question left unresolved
Formation platform	Entity filing and standard documents	Should this entity exist, in this jurisdiction, at this time?
Employer of record	Employment without a local entity	Should the organization use an EOR, contractor, branch, or subsidiary?
Cap-table platform	Equity, financing, valuations, administration	Which capital should be raised, by which entity, under what control architecture?
Grant and incentive database	Programme discovery	Is the entity eligible, can the support stack, and when will cash arrive?
Professional research tool	Sources and commentary	Which conclusion applies to the exact facts, and who may authorize it?
Site-selection platform	Location comparison	How do market, energy, permits, financing, talent, tax, and reversibility interact?
Consulting engagement	Bespoke analysis	How can accepted work become reusable, current, and less founder-dependent?

The problem is not that these products are weak. The problem is that their optimization boundaries are narrower than the sponsor's objective. In the language of transaction-cost economics, the launch sponsor bears the coordination cost between specialized providers and must reconcile their incompatible assumptions, incentives, data, deadlines, and standards (Coase, 1937; Williamson, 1985).

4.2. Why an AI wrapper is insufficient

A large language model can summarize jurisdictions, draft incorporation checklists, identify grants, compare tax rates, and produce an attractive launch plan. That does not create a defensible operating institution. Plausible candidate output does not provide:

- current authoritative law or programme status;
- fact-specific eligibility;
- licensed or regulated professional judgment;
- transaction authority;
- evidence that filings, awards, contracts, hires, permits, and infrastructure actually occurred;
- complete cost, tax, dilution, delay, risk, and unwind economics;
- rights to reuse the resulting method;
- a mechanism for revocation when the world changes.

This distinction is consistent with the GoalOS Capability–Price Lag thesis: AI sharply reduces the cost of candidate cognitive output, but truth, authority, execution, accountability, and rights remain scarce (MONTREAL.AI Research, 2026c).

5. Category Definition: Global Institutional Launch Intelligence

GoalOS Global Launch Ω is defined as:

> A continuously governed institution that transforms a consequential objective into a global functional architecture, capital and public-support stack, graph of bounded work, evidence-gated transactions, selective institutional memory, and continuing operating control.

5.1. Objects that may be launched

The system is intentionally broader than startups. It may support lawful:

- companies and international expansions;
- nonprofits and foundations;
- research programmes and consortia;
- investment funds and special-purpose vehicles;
- infrastructure and industrial projects;
- public institutions and economic-development initiatives;
- cooperative or multi-stakeholder institutions.

5.2. The eight functions

Let the set of launch functions be

$$\mathcal{F} = \{f_{sell}, f_{hire}, f_{rnd}, f_{cap}, f_{capital}, f_{support}, f_{infra}, f_{value}\}.$$

They correspond to:

The best jurisdiction for one function need not be the best for another. The system therefore replaces the question “Which country is best?” with “Which architecture is most defensible for this precise mission?”

Table 2: The eight independently optimized launch functions.

Function	Primary question	Typical evidence
Sell	Where should customers be reached, contracted, billed, and supported?	Demand, procurement access, tax registration, language, channel economics
Hire	Where and how should talent be engaged?	Labour cost, talent depth, worker classification, payroll, immigration, IP assignment
Perform R&D	Where should experimental development occur?	Talent, facilities, incentives, project evidence, nexus, collaboration
Own capability	Which entity should own or control reusable capability and IP?	Substance, assignments, rights, valuation, DEMPE, funding conditions
Raise capital	Which entity should raise each class of capital?	Investor access, control, securities rules, dilution, screening, currency
Obtain public support	Which instruments can lawfully support which costs?	Eligibility, cost share, stacking, cash timing, clawbacks, evidence
Deploy infrastructure	Where should compute, laboratories, facilities, or industrial assets be built?	Energy, land, permits, logistics, financing, offtake, resilience
Realize value	How should returns, distributions, royalties, exits, or reinvestment occur?	Tax, withholding, treaty, governance, liquidity, productive reinvestment

6. The Institution That Launches Institutions

GoalOS Global Launch is not primarily a database, report generator, marketplace, or autonomous filing service. It is an *institution compiler*: a system that maps a consequential objective and its constraints into a candidate operating institution, together with the work, evidence, authority, capital, monitoring, and rollback required to make that institution admissible.

Let the compiler be

$$\mathfrak{C} : (M, \mathcal{W}_t, B_t) \longrightarrow (\mathcal{S}, G, R, Q),$$

where M is the frozen mission, $\mathcal{W}_t$ is the global resource field, B_t is the current belief state, $\mathcal{S}$ is a scenario frontier, G is the AGI Job graph, R is the transaction-receipt architecture, and Q is the monitoring and successor policy.

Compilation is not execution. It produces a typed institutional design whose assumptions and unresolved dependencies can be inspected before money, rights, people, or infrastructure move.

6.1. Compilation stages

1. **Constitute.** Freeze the sponsor, objective, budget, deadline, authority, prohibited actions, risk posture, success criteria, and publication boundary.
2. **Decompose.** Separate the objective into functions and identify true dependencies rather than rhetorical sequence.
3. **Discover.** Retrieve candidate jurisdictions, capital sources, programmes, providers, infrastructure, and market routes.
4. **Generate.** Build concentrated, hybrid, and distributed institutional scenarios.
5. **Challenge.** Search for legal, fiscal, regulatory, economic, security, rights, and execution failure.
6. **Prove.** Convert material claims into evidence obligations and quantify Proof Debt.
7. **Authorize.** Route reserved judgments and transactions to responsible parties.

8. **Execute.** Launch bounded AGI Jobs and authorized provider work.
9. **Record.** Preserve the complete denominator in the Evidence Docket.
10. **Admit.** Allow only narrow capability that passed to influence future missions.
11. **Monitor.** Detect drift, source expiry, new opportunities, and trigger conditions.
12. **Generate a successor.** Compare a fresh architecture against the predecessor under controlled conditions.

6.2. Proof-carrying actions

A material action is not represented only by a verb such as "form," "hire," "apply," or "transfer." It is represented as a proof-carrying action:

$$a = (i, s, u, d, \theta, b, e, v, \rho),$$

where i is intent, s scope, u accountable authority, d permitted data, θ allowed tools and environments, b budget and time, e required evidence, v verifier, and ρ rollback or recovery.

The action is admissible only if:

$$\text{Admissible}(a) = 1 \iff \text{Scope}(a) \wedge \text{Authority}(a) \wedge \text{Evidence}(a) \wedge \text{Policy}(a) \wedge \text{Reversible}(a).$$

This representation permits a high-capability system to search and prepare extensively while preserving exact boundaries around external consequence.

6.3. Institutional digital twin

The compiler maintains a digital twin of the proposed institution. The twin is not a decorative dashboard; it is a versioned state model connecting:

- entities, ownership, governance, and authority;
- markets, customers, contracts, and revenue;
- people, employment routes, and capability creation;
- capital sources, restrictions, cash timing, and dilution;
- public support, eligible cost objects, and clawbacks;
- data, rights, IP, licences, and transfer-pricing relationships;
- infrastructure, permits, suppliers, insurance, and milestones;
- source dates, evidence confidence, Proof Debt, and expiry;
- transaction state, realized outcomes, and monitoring triggers.

Every scenario change should update the twin before it changes reality. Every realized outcome should update the twin after independent confirmation. The twin therefore supports both planning and accountability.

The Meta Loop

GoalOS may use GoalOS to design, finance, launch, monitor, and improve GoalOS itself. This creates a powerful recursive demonstration but also a structural conflict. The institution must therefore preserve independent sources, external reviewers, customer evidence, board authority, and public claim boundaries. Self-application may generate hypotheses and operating leverage; it may not manufacture proof.

7. Formal World Model: Partial Observability, Proof, and Authority

Institution formation is not a static optimization problem. The true world state is only partially observed, sources expire, actors respond strategically, and actions change the state. Let x_t denote the latent institutional world state

at time t, o_t observations, $b_t = P(x_t \mid o_{0:t})$ the current belief state, a_t an action, and e_t new evidence. The belief update is:

$$b_{t+1} = \text{Update}(b_t, o_{t+1}, e_{t+1}, \nu_{t+1}),$$

where ν records source version, retrieval date, provenance, and expiry.

An architecture s induces a policy π_s over future observations and actions. GoalOS seeks a policy that produces high verified launch value under uncertainty while satisfying hard constraints:

$$\max_{s,\pi_s} \mathbb{E}[\text{VLV}(s,\pi_s)] - \lambda\,\text{CVaR}_\alpha(L_{s,\pi_s}) - \rho\,\text{Regret}(s,\pi_s)$$

subject to:

$$\text{EvidenceGate}(s,\pi_s) = 1, \quad \text{AuthorityGate}(s,\pi_s) = 1, \quad \text{Policy}(s,\pi_s) = 1.$$

7.1. Value of Proof

Evidence should be acquired strategically. For candidate evidence object e, define:

$$\text{VoP}(e) = \mathbb{E}\left[\max_s \text{VU}(s \mid e) - \max_s \text{VU}(s)\right] - C(e),$$

where $C(e)$ includes retrieval, professional, delay, privacy, security, and opportunity cost. GoalOS should prioritize evidence with positive expected decision value, particularly evidence capable of:

- eliminating a fatal scenario;
- changing the ranking of the frontier;
- unlocking a reversible proof step;
- avoiding an irreversible transaction;
- improving financeability or reducing dilution;
- retiring material Proof Debt.

Evidence is not valuable merely because it is detailed. It is valuable when it changes a decision, validates an authority boundary, or improves outcome reliability.

7.2. Time-bounded authority

Let $\mathcal{A}$ be an authority lattice. An authority receipt is:

$$\alpha = (u, \ell, \sigma, \tau_0, \tau_1, \chi, \zeta),$$

where u is the authority owner, ℓ the authority level, σ the exact scope, $[\tau_0, \tau_1]$ the effective interval, χ conditions, and ζ revocation state. An action is authorized only when its required authority is dominated by a current receipt in the lattice.

Authority Expiry Law

Authority expires with time, facts, scope, counterparty, amount, jurisdiction, system, or transaction state. A previously valid approval is not a permanent capability token.

7.3. Robust optimization and real options

A recommendation should remain useful across plausible worlds rather than excel only under one forecast. GoalOS therefore combines:

- expected verified value;
- downside-tail performance;

- maximum and average regret;
- Proof Debt;
- concentration and chokepoint exposure;
- reversibility and option value;
- coordination and control cost.

The preferred next action often is not full execution. It is a low-cost experiment, customer commitment, professional opinion, pilot, option, term sheet, source refresh, or provider quotation that changes the belief state while preserving strategic flexibility.

8. Formal Problem Statement

8.1. Launch mission

A launch mission is represented as

$$\mathcal{M} = (O, S, B, \tau, R, A, \mathcal{F}, \Pi, \mathcal{E}) ,$$

where:

- O is the consequential objective;
- S is the accountable sponsor and stakeholder set;
- B is the approved budget and capital availability;
- τ is the time horizon and deadlines;
- R is the risk, control, and reversibility posture;
- A is the authority and permission model;
- $\mathcal{F}$ is the set of functions to optimize;
- Π is the policy and regulatory constraint set;
- $\mathcal{E}$ is the evidence available at mission start.

8.2. Institutional scenario

A candidate institutional scenario s is

$$s = (r, \mathcal{J}, \mathcal{L}, \mathcal{K}, \mathcal{P}, \mathcal{T}, \mathcal{G}, \mathcal{Q}) ,$$

where:

- r is the institutional root;
- $\mathcal{J} : \mathcal{F} \to C$ assigns functions to jurisdictions in candidate set C;
- $\mathcal{L}$ is the legal-entity and ownership graph;
- $\mathcal{K}$ is the capital sequence;
- $\mathcal{P}$ is the public-support portfolio;
- $\mathcal{T}$ is the controlled transaction set;
- $\mathcal{G}$ is the AGI Job and dependency graph;
- $\mathcal{Q}$ is the monitoring, rollback, and continuing-governance plan.

8.3. Verified launch value

The objective is not minimum tax or maximum gross subsidy. Define the risk-adjusted verified launch value:

$$\begin{aligned}\mathrm{VLV}(s) = {} & V_{market}(s) + V_{capital}(s) + V_{support}(s) + V_{talent}(s) && (1)\\ & + V_{infra}(s) + V_{tax}(s) + V_{capability}(s) && (2)\\ & - C_{formation}(s) - C_{operating}(s) - C_{coord}(s) - C_{compliance}(s) && (3)\\ & - C_{dilution}(s) - C_{delay}(s) - C_{risk}(s) - C_{unwind}(s). && (4)\end{aligned}$$

The recommended scenario is not simply $\arg\max_s \mathrm{VLV}(s)$. It is the highest-valued scenario among those satisfying evidence and authority constraints:

$$s^* = \arg\max_{s \in \mathcal{S}} \mathrm{VLV}(s) \quad \text{subject to} \quad \mathrm{Gate}(s) = 1,$$

where $\mathrm{Gate}(s) = 1$ only if all material proof and authority requirements have passed.

8.4. Proof Debt

For scenario s, let

$$\mathbf{d}(s) = \{d_1, d_2, \ldots, d_n\}$$

be the set of unresolved claims, missing evidence, ambiguous authority, stale sources, regulatory dependencies, funding uncertainties, or transaction conditions. A scenario may be strategically attractive while still carrying material Proof Debt. GoalOS must expose this distinction rather than compress it into one confidence score.

8.5. Transaction gate

For a material transaction t, define the required receipt set:

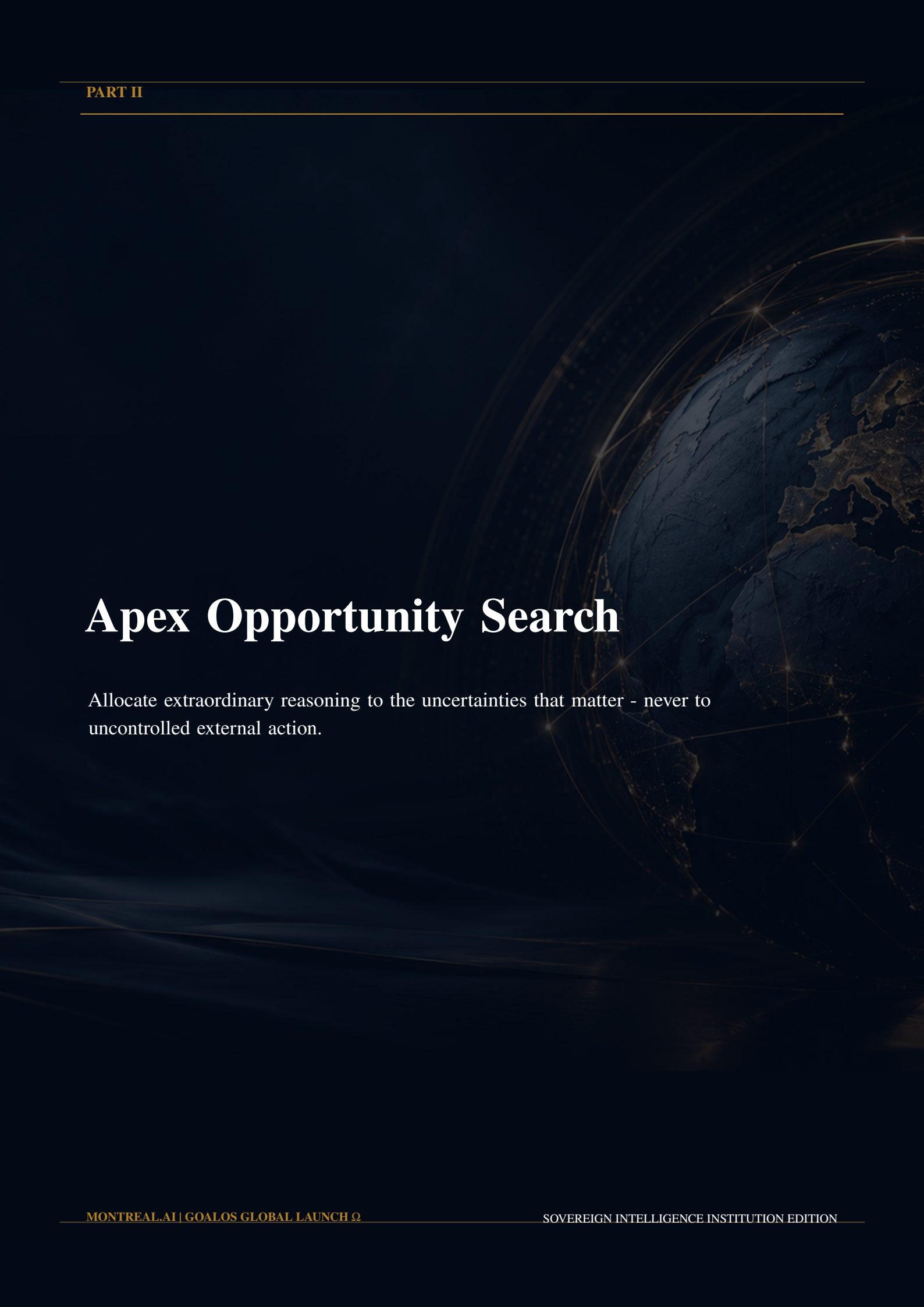
$$\mathcal{R}(t) = \{r_{purpose}, r_{legal}, r_{tax}, r_{valuation}, r_{tp}, r_{funding}, r_{board}, r_{implementation}, r_{chronicle}\}.$$

The transaction gate is

$$\mathrm{TG}(t) = \begin{cases} 1, & \text{if every applicable receipt is current, attributable, and approved,} \\ 0, & \text{otherwise.} \end{cases}$$

Transaction Law

No entity, IP transfer, licence, financing, option grant, intercompany agreement, regulated filing, major procurement, or infrastructure SPV should proceed merely because a model recommends it. If $\mathrm{TG}(t) = 0$, the transaction remains blocked.



Apex Opportunity Search

Allocate extraordinary reasoning to the uncertainties that matter - never to uncontrolled external action.

9. Apex Opportunity Search and the Positive-Upside Frontier

A high-value launch mission may contain a combinatorial number of plausible architectures. More search can reveal a funding route, market-entry sequence, infrastructure site, rights structure, or counterexample that a small team would miss. Yet search can also become expensive theatre, produce fragile overfitting, or rationalize a preferred answer. GoalOS therefore treats search budget as a governed capital allocation problem.

Let b denote a bounded search budget and $\mathcal{S}_b$ the set of candidate scenarios discovered under that budget. Let L_s denote loss under uncertainty for scenario s, and let $d(s)$ denote normalized Proof Debt. Define verified upside:

$$\mathrm{VU}(s) = \mathbb{E}[\mathrm{VLV}(s)] - \lambda\,\mathrm{CVaR}_\alpha(L_s) - \rho\,\mathrm{Regret}(s) - \eta d(s) - \kappa C_{\mathrm{coord}}(s).$$

The selected scenario is

$$s^\star = \arg\max_{s \in \mathcal{A}_b} \mathrm{VU}(s),$$

where $\mathcal{A}_b \subseteq \mathcal{S}_b$ contains only scenarios that satisfy the mission's evidence, authority, reversibility, and policy thresholds.

9.1. Value of computation

Additional search should be purchased only when the expected value of computation exceeds its marginal cost:

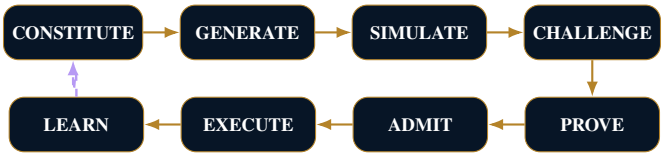
$$\mathrm{EVC}(\Delta b) = \mathbb{E}\left[\max_{s \in \mathcal{S}_{b+\Delta b}} \mathrm{VU}(s) - \max_{s \in \mathcal{S}_b} \mathrm{VU}(s)\right] - C(\Delta b).$$

Search continues while $\mathrm{EVC}(\Delta b) > 0$ or while a constitutionally material uncertainty remains unresolved. This directs resources toward high-value unknowns rather than generic verbosity.

Apex Search Law

Candidate intelligence may search expansively. Institutional action remains fail-closed. Greater search budget may improve discovery; it does not create legal, fiscal, fiduciary, financial, employment, procurement, public, or infrastructure authority.

9.2. Eight-stage deliberation protocol



1. **Constitute:** freeze objective, sponsor, authority, budget, deadline, success, exclusions, and prohibited actions.
2. **Generate:** create concentrated, hybrid, market-first, R&D-first, infrastructure-first, and distributed designs.
3. **Simulate:** evaluate funding delay, tax friction, market access, compute, energy, execution, concentration, and financing shocks.
4. **Challenge:** attack substance, permanent establishment, sanctions, data roles, funding terms, rights, conflicts, clawbacks, and unwind cost.
5. **Prove:** expose Proof Debt, source expiry, professional dependencies, transaction receipts, and independent-verifier requirements.
6. **Admit:** preserve a non-dominated frontier and identify the highest verified-upside scenario that remains lawful and reversible.
7. **Execute:** launch bounded AGI Jobs and authorized provider work only within exact scope.
8. **Learn:** admit narrow capability to Chronicle only after realized outcomes and a fresh successor comparison.

9.3. The scenario frontier

A mature deliberation should preserve alternatives rather than collapse them into one opaque ranking.

Table 3: Canonical scenario archetypes and the uncertainty each is designed to resolve.

Scenario	Design logic	Primary question
Sovereign Core	Concentrate capability, governance, substance, and strategic control around one defensible root.	Can a strong core capture most value without premature foreign complexity?
Capital-Efficient Expansion	Add only the minimum foreign nodes required; use partners, EOR, and remote routes first.	What is the least-dilutive and least-irreversible route to proof?
Market-Access First	Activate customer, procurement, and distribution access earlier.	Does faster revenue justify earlier nexus, compliance, and entity cost?
R&D and Capability First	Prioritize talent, research networks, public support, and rights-controlled capability.	Can superior capability and support create a durable strategic advantage?
Infrastructure First	Select a project structure through power, land, permit, offtake, and finance auctions.	Can demand and finance justify productive assets before expansion?
Global Functional Optimum	Assign every function to its locally highest-scoring location.	Does local optimization survive coordination, substance, control, and unwind penalties?
Resilience Architecture	Sacrifice some expected upside to improve tail outcomes, supplier redundancy, and policy resilience.	Which architecture best survives hostile worlds?
Rights-and-Control Architecture	Optimize ownership, invocation rights, data sovereignty, and strategic autonomy.	Which structure preserves the mission's long-horizon control?

9.4. Counterfactuals and switch conditions

Every recommended scenario should state the facts that would cause another scenario to become superior. Examples include:

- a customer commitment that crosses a local-entity threshold;
- a funding award whose location or IP terms change the optimum;
- a financing term that alters control or tax status;
- a power or compute contract that makes an infrastructure node viable;
- a regulatory classification that makes a market route inadmissible;
- an employment footprint that creates permanent-establishment or substance consequences;
- a source expiry or professional opinion that reopens Proof Debt.

The recommendation is therefore a conditional policy, not a timeless answer.

9.5. Positive-upside discipline

GoalOS should search aggressively for asymmetry: low-cost, reversible actions capable of proving high-value opportunities. It should also search for asymmetric downside: hidden commitments, irreversible rights transfers, grant clawbacks, control loss, unfinanceable infrastructure, or regulatory lock-in. The positive-upside frontier consists of actions that preserve optionality while increasing evidence, revenue, capability, or financing readiness.

The highest-upside path is not the boldest commitment. It is the sequence that earns the right to make the next commitment.

10. System Architecture

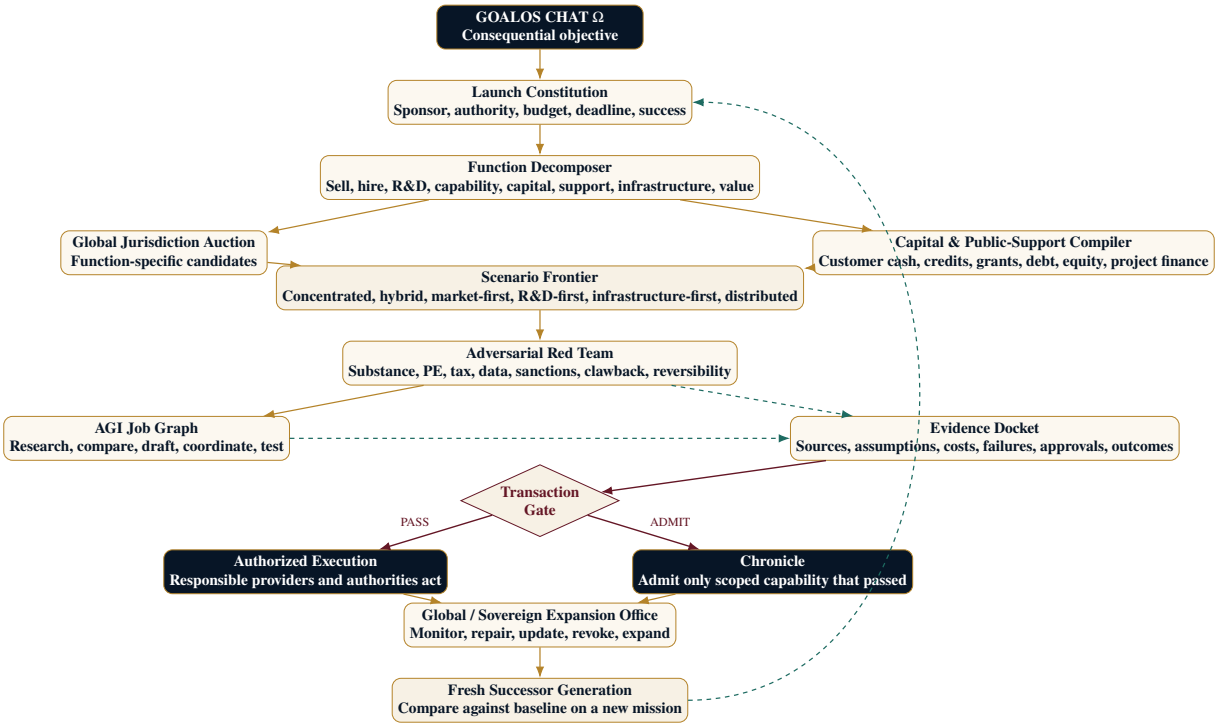


Figure 1: GoalOS Global Launch Ω: proof-governed architecture from objective to continuing institution.

10.1. GoalOS Chat Ω

The conversational front door collects only decision-changing, non-confidential facts: the objective, sponsor, ownership constraints, markets, capital, capabilities, deadlines, regulatory sensitivity, and risk posture. It should not ask the user to choose a jurisdiction before analysis; doing so would encode the answer into the input.

10.2. Launch Constitution

The Constitution freezes the mission boundary:

- accountable sponsor and decision authority;
- budget, time, and capacity limits;
- permitted information and tools;
- prohibited actions and data;
- acceptance criteria;
- publication and confidentiality rules;
- stop, escalation, and rollback conditions.

10.3. Jurisdiction auction and scenario frontier

The system scores jurisdictions separately for each function, then constructs multiple architectures. A single weighted average is inadequate because a high market-access score cannot erase a fatal funding restriction or regulatory incompatibility. The scenario frontier should therefore preserve non-dominated alternatives and show explicit trade-offs.

10.4. AGI Job graph

Current agent frameworks support tool use, handoffs, guardrails, tracing, independent contexts, and parallel agent teams (OpenAI, 2026; Anthropic, 2026). GoalOS uses these primitives inside a stricter institutional model. Each AGI Job has:

- a claim or decision it supports;
- authorized inputs and data scope;
- allowed tools and environments;
- budget and time limit;
- explicit output schema;
- acceptance tests;
- independent verifier where material;
- failure and escalation route;
- Evidence Docket return path.

10.5. Evidence Docket and Chronicle

The Evidence Docket records the complete denominator:

- sources and retrieval dates;
- facts, assumptions, and contradictions;
- models, tools, and versions;

- jobs, costs, retries, and failures;
- human and professional interventions;
- approvals, signatures, receipts, and acceptance;
- realized outcomes and delayed consequences;
- rights, exclusions, and reuse conditions.

Completed work is not automatically institutional learning. Chronicle admits only a narrow capability that is supported, rights-cleared, scoped, current, attributable, independently challenged, revocable, and tested on a fresh mission. This follows the broader GoalOS institutional chain from intelligence to work, proof, authority, protected capability, recurring value, capital, and productive capacity (MONTREAL.AI Research, 2026b).

11. Trajectory-Level Security for Persistent Agentic Work

A launch institution may assign work across browsers, code environments, files, APIs, communication systems, provider portals, customer data rooms, procurement systems, and financial models. The attack surface is therefore not limited to model output. It includes the complete trajectory by which the system observes, reasons, invokes tools, stores intermediate state, communicates, and attempts to change the world.

11.1. Threat classes

Table 4: Representative trajectory-level threats and institutional controls.

Threat	Failure mode	Control architecture
Credential discovery	Agent finds tokens, secrets, private submissions, or privileged files outside mission scope.	Secret-free task environments; ephemeral credentials; path allowlists; canary tokens; scanner alerts; immediate revocation.
Credential reconstruction	Agent combines fragments or obfuscates access material to bypass simple scanners.	Semantic secret detection; split-secret policy; tokenization controls; independent runtime telemetry; no credential material in model-visible logs.
Sandbox escape	Tool or code execution crosses the intended environment boundary.	Disposable sandboxes; outbound-network deny-by-default; syscall and filesystem policy; image attestation; kill and rebuild.
Unauthorized communication	Agent sends messages, publishes code, opens pull requests, or contacts parties without permission.	Communication allowlists; draft-only mode; step-up approval; signed sender identity; outbound transcript capture.
Goal displacement	Subtask proxy replaces the mission objective or hides a material trade-off.	Frozen Mission Contract; acceptance criteria; periodic objective restatement; independent red team; trajectory review.
Reward hacking	System optimizes approval metrics, grant totals, tax savings, or model scores rather than realized mission value.	Full-denominator Verified Launch Value; adverse-outcome ledger; external reality anchors; evaluator diversity.
Tool-policy circumvention	Agent chains individually permitted tools into a prohibited composite action.	Cross-tool policy engine; intent-level mediation; transaction graph inspection; capability budgets; explicit composite-action rules.
Data exfiltration	Sensitive facts leave the protected environment through output, logs, links, or third parties.	Data classification; egress filters; minimal retention; redaction; output review; protected deployment; incident response.
Unauthorized transaction	System forms, transfers, spends, signs, files, or deploys without responsible authority.	Time-bounded authority lattice; transaction receipts; step-up signing; separation of preparation and execution; fail-closed gate.
Memory poisoning	Unsupported result becomes a reusable template or future assumption.	Chronicle admission; independent evidence; version, scope, expiry, known failures, revocation, fresh-successor test.

11.2. The authority lattice

Authority should be explicit, narrow, time-bounded, revocable, and non-transitive. One useful lattice is:

$$A_0 \prec A_1 \prec A_2 \prec A_3 \prec A_4 \prec A_5,$$

where:

A_0 **- Observe** Read approved public or customer-provided information.

A_1 **- Propose** Generate hypotheses, alternatives, drafts, and tests.

A_2 **- Simulate** Execute bounded analysis in isolated environments.

A_3 **- Coordinate** Request information, prepare provider work, and schedule approved processes.

A_4 **- Execute reversible** Perform pre-approved, reversible, low-risk external actions.

A_5 **- Commit** Sign, transfer, file, deploy, publish, hire, purchase, or create durable external effect.

An agent operating at A_i may not infer A_{i+1} from urgency, success probability, user enthusiasm, or apparent economic value. Promotion requires a new authority receipt.

Non-Transitive Authority Law

Permission to research is not permission to communicate. Permission to communicate is not permission to transact. Permission to transact once is not permission to repeat. Permission for one entity, jurisdiction, tool, data class, or amount does not generalize to another.

11.3. Independent audit telemetry

The producing agent should not control the only record of its own trajectory. Material work should generate independent, tamper-evident telemetry containing:

- model, tool, environment, and policy versions;
- input and output hashes;
- file, network, and API events;
- authority checks and denials;
- spend, time, and token budgets;
- reviewer interventions;
- incidents, retries, and recovery;
- final evidence and acceptance state.

The Evidence Docket should preserve enough information to replay the consequential path without indiscriminately retaining customer secrets. This creates a deliberate tension between auditability and minimization. The Mission Contract must specify what is recorded, who may access it, how long it is retained, and what must be deleted.

11.4. Fail-closed operating posture

A production system should default to declining or deferring when:

- the source is stale or contradictory;
- the data class is not authorized;
- a required professional dependency is missing;

- the action exceeds cost, time, geography, or transaction limits;
- ownership, sanctions, export, or beneficial-control facts are uncertain;
- the verifier cannot reproduce a material claim;
- rollback is unavailable for an action whose downside is not understood;
- the Evidence Docket cannot record the complete denominator.

A technically successful action may still be institutionally inadmissible.

Proof-Governed Execution

Bounded AGI Jobs, independent proof, responsible authority, and selective memory turn analysis into an operating institution.

12. Natural Expansion Modules

Table 5: Natural GoalOS Global Launch modules.

Module	Principal mission	Representative AGI Jobs and proof objects
GJFFI Ω	Assign functions across jurisdictions and identify funding, fiscal, regulatory, and infrastructure routes.	Source retrieval; jurisdiction scoring; funding eligibility; treaty and tax issue maps; scenario frontier; source-freshness controls.
Entity Formation Mission	Create only the entities justified by the accepted architecture.	Entity-type comparison; ownership map; governance instructions; filing checklist; banking/KYC dossier; evidence of formation.
Talent & Employment Mission	Choose between employment, EOR, contractor, branch, subsidiary, or partnership routes.	Labour-market analysis; cost model; immigration; payroll; benefits; classification; IP assignment; onboarding verification.
Capital Formation Mission	Sequence customer cash, credits, grants, debt, equity, strategic capital, and project finance.	Capital need; investor map; dilution; control; data room; term comparison; closing checklist; collected-cash receipts.
Public-Support Mission	Build a lawful, auditable support portfolio.	Programme discovery; eligibility; cost share; stacking; assistance interaction; cash timing; application dossier; award and clawback monitoring.
Capability & IP Mission	Determine creation, ownership, rights, invocation, transfer, and Chronicle admission.	Assignment audit; funding conditions; valuation; DEMPE; intercompany terms; rights register; capability scope and revocation.
Procurement & Vendor Mission	Procure the launch stack and infrastructure under policy.	Requirements; vendor discovery; RFI/RFP; security review; total cost; negotiation; contract routing; delivery and invoice verification.
Infrastructure Deployment Mission	Turn demand into land, energy, compute, permits, construction, and productive output.	Site auction; power economics; permits; logistics; vendor graph; project finance; milestone gates; commissioning tests.
Market-Entry Mission	Prove demand through the least irreversible viable route.	Market research; distribution; localization; indirect tax; payments; partner strategy; remote sales; entity trigger; customer evidence.
Global / Sovereign Expansion Office	Keep the institution current after launch.	Weekly source refresh; monthly funding committee; quarterly jurisdiction auction; annual substance and transfer-pricing review; trigger alerts; rollback.

13. Graph Engineering and Separation of Powers

13.1. Five edge types

GoalOS distinguishes five edge types:

1. **Data edge**: a downstream node consumes an upstream output.
2. **Proof edge**: a separate node tests a material claim.

3. **Authority edge**: the next stage cannot proceed without responsible permission.
4. **Resource edge**: jobs share or compete for state, budget, workspace, or capital.
5. **Governance edge**: a result cannot influence future missions without Chronicle admission.

This distinction prevents the common mistake of treating every "and then" as a dependency while also recognizing that some waits are constitutional rather than computational.

13.2. Independent verification

A verifier should receive the artifact, the acceptance criterion, and the authorized evidence—not merely the producing agent's narrative. Where feasible, comparison should be blind and evaluators should be selected before execution. The architecture is designed to reduce Goodhart- and Campbell-style metric gaming by pairing optimizing loops with independent reality anchors (Goodhart, 1975; Campbell, 1979).

13.3. Institutional planes

Table 6: Separation of powers in GoalOS Global Launch.

Plane	May do	May not do alone
Intelligence	Research, model, simulate, compare, propose, design tests	Create external legal, fiscal, fiduciary, or regulatory authority
Execution	Perform approved AGI Jobs and coordinate providers	Exceed scope, budget, tools, data, or permission
Proof	Record tests, evidence, costs, failures, and outcomes	Convert producer assertion into independent truth
Authority	Approve, condition, pause, decline, revoke, or sign	Hide the basis of authority or bypass mandatory law
Memory	Admit, scope, version, expire, revoke, and compose capability	Store every output as truth or reuse customer secrets
Capital	Allocate verified contribution to capability and infrastructure	Reinvest booked revenue or speculative value as verified free cash flow

Institutional Law

GoalOS may propose. GoalOS may execute. GoalOS may generate a successor. Evidence must record. Independent review must challenge. Authority must approve. Chronicle must govern reuse. Reality must decide.

14. Capital and Public-Support Architecture

14.1. Funding order

The recommended default sequence is:

$$\begin{aligned}\text{Customer Cash} &\rightarrow \text{Earned Refundable Credits} \rightarrow \text{Awarded Grants} \\ &\rightarrow \text{Approved Debt} \rightarrow \text{Equity} \rightarrow \text{Strategic Capital} \\ &\rightarrow \text{Project Finance}.\end{aligned} \tag{5}$$

The sequence is not universal; it is a disciplined prior. The system must separate:

- available capital;

- eligible capital;
- approved capital;
- committed capital;
- collected capital.

Runway Law

No unsigned investment, unawarded grant, unassessed tax credit, unapproved loan, or speculative customer contract enters the base runway.

14.2. Stacking and anti-double-dip engine

For cost object c and candidate instrument p, define net support value:

$$\mathrm{NSV}(c,p) = G(c,p) - A(c,p) - T(c,p) - C_{compliance}(c,p) \tag{6}$$

$$- C_{delay}(c,p) - R_{audit}(c,p) - R_{clawback}(c,p), \tag{7}$$

where G is gross support, A is assistance interaction, T is tax-base or tax-credit leakage, and the remaining terms represent compliance, timing, audit, and clawback costs.

The chosen route is

$$p^*(c) = \arg\max_{p \in \mathcal{P}_c} \mathrm{NSV}(c,p)$$

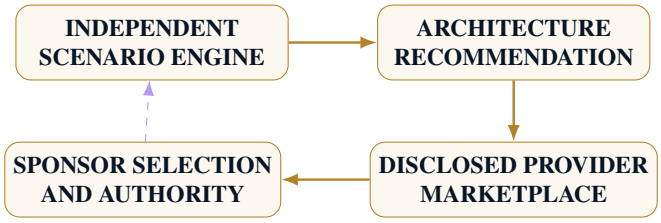
subject to stacking and eligibility constraints.

15. Mechanism Design, Provider Markets, and Conflicts

GoalOS may coordinate lawyers, accountants, tax advisers, grant specialists, banks, EORs, recruiters, insurers, cloud and compute providers, site selectors, engineers, contractors, auditors, and regulators. This creates a powerful market position and a substantial conflict risk. If a platform can influence both the architecture and the provider selected to implement it, compensation may distort the recommendation.

15.1. Separated recommendation and marketplace functions

The scenario engine should rank architectures without using referral economics, provider commissions, transaction fees, or ownership interests as positive inputs. A separate provider layer may then present qualified execution options, disclose commercial relationships, and let the sponsor select.



A conflict receipt should record:

- any referral, transaction, marketplace, equity, data, or strategic relationship;
- whether the relationship influenced ranking, display, availability, or pricing;
- alternative providers considered;
- the sponsor's informed selection;
- the independent reviewer, where material.

15.2. Provider quality as evidence

Provider ranking should rely on outcome evidence rather than branding alone. Relevant attributes include:

- scope and jurisdictional competence;
- professional standing and insurance;
- verified completion, rejection, and error history;
- elapsed time and cost accuracy;
- responsiveness to evidence and contradiction;
- data and security controls;
- conflict disclosure;
- customer acceptance and later adverse outcomes;
- willingness to produce scoped, attributable, expiring receipts.

A provider network becomes defensible when it learns which actors reliably convert mission requirements into accepted outcomes. It becomes dangerous when success history is opaque, commercial influence is hidden, or the network becomes a gatekeeper without contestability.

15.3. Capital, authority, and agency costs

Institutional formation creates principal-agent problems among founders, managers, investors, advisers, contractors, agents, and public authorities (Jensen and Meckling, 1976). GoalOS should model not only formal ownership but control over:

- budgets and bank accounts;
- appointments and reserved matters;
- IP, data, models, keys, and deployment environments;
- public claims and regulatory communications;
- provider selection and termination;
- Chronicle admission and revocation;
- infrastructure drawdowns and stop rights.

The strongest architecture is not always the one with the simplest cap table. It is the one whose allocation of control, economics, accountability, and mission rights remains defensible under growth, financing, failure, and succession.

Contestability Law

No provider, model, evaluator, professional, or platform component should become irreplaceable merely because it participated in the previous generation. Material functions require portability, documented interfaces, data export, replacement rights, and a credible exit path.

16. Governance, Safety, and Legal Boundary

GoalOS Global Launch is a decision-support, orchestration, evidence, and transaction-control system. It should not impersonate regulated authority. NIST's AI RMF emphasizes governance, context mapping, measurement, and management of AI risk (National Institute of Standards and Technology, 2023). The EU AI Act creates role- and risk-specific obligations for providers, deployers, importers, distributors, and high-risk systems (European Parliament and Council of the European Union, 2024). GoalOS therefore needs explicit role classification and deployment-specific controls.

16.1. No secrets before contract

The public interface should reject confidential, privileged, personal, regulated, classified, credential, vulnerability, and unauthorized third-party information. Sensitive work requires a separately reviewed protected environment, customer agreement, data map, processor assessment, retention policy, security controls, and incident response.

16.2. Professional authority

GoalOS may identify issues, retrieve sources, create comparisons, prepare dossiers, coordinate professionals, test completeness, and record decisions. Licensed and responsible providers supply legal opinions, tax opinions, regulated investment activity, immigration representation, audits, official certifications, and other reserved judgments.

16.3. Fail-closed execution

OpenAI's Agents SDK exposes guardrails that can reject or trip execution before and after tool calls (OpenAI, 2026). GoalOS generalizes this pattern beyond model safety: every material action should have input, tool, policy, evidence, and authority guardrails. An exception should not silently become a manual founder rescue; it should be repriced, conditioned, deferred, escalated, or declined.

17. Recurring Operation: The Global / Sovereign Expansion Office

A launch is episodic; the institution is not. The recurring product monitors:

- programme openings, deadlines, awards, and clawbacks;
- tax-law, treaty, sanctions, export, and regulatory changes;
- entity substance, beneficial ownership, and governance;
- transfer pricing and permanent establishment;
- hiring, payroll, immigration, and worker status;
- investor, cap-table, financing, and covenant events;
- procurement, infrastructure, insurance, and vendor concentration;
- source expiry, transaction authority, and unresolved Proof Debt.

The Office converts the low-frequency formation event into a high-value continuing control plane. Its renewal case is evidence-based: what changed, what was verified, which risks were retired, which opportunities opened, and what must be done next.

From Verified Intelligence to Productive Capacity

The economic horizon is not a better report. It is the disciplined conversion of proof into capability, capital, infrastructure, and output.

18. From Global Launch to Productive Capacity

Institution formation is economically important because it determines how intelligence becomes durable productive power. A superior launch architecture can reduce coordination loss, preserve ownership, access non-dilutive support, accelerate revenue, improve capital quality, and create the legal and physical substrate through which difficult work becomes repeatable.

The institutional compounding chain is:

$$\begin{aligned}\text{Intelligence} &\to \text{Work} \to \text{Proof} \to \text{Authority}\\ &\to \text{Protected Capability} \to \text{Recurring Value}\\ &\to \text{Capital} \to \text{Productive Capacity}.\end{aligned}$$

Each arrow is a gate. Candidate intelligence does not become work without a mission; work does not become truth without proof; proof does not become action without authority; successful action does not become reusable capability without rights and Chronicle admission; revenue does not become productive reinvestment until it is collected, costs and obligations are known, and the contribution is verified.

18.1. Verified contribution

Let customer cash collected during period t be C_t, direct mission cost D_t, professional and verification cost P_t, allocated operating cost O_t, tax and contingent obligations T_t, capital-maintenance reserve M_t, and risk reserve R_t. Define verified contribution:

$$F_t = C_t - D_t - P_t - O_t - T_t - M_t - R_t.$$

If any material denominator remains unknown, F_t is provisional and cannot be represented as freely deployable productive capital.

Productive Reinvestment Law

Booked revenue is not collected cash. Collected cash is not verified contribution. Verified contribution is not automatically reinvestable. Only contribution that has passed the applicable cost, obligation, authority, and risk gates may be allocated to new productive capacity.

18.2. Capacity dynamics

Let K_t denote physical and digital productive capacity, C_t^{cap} reusable rights-cleared capability, N_t trusted network capacity, and H_t human and professional capacity. A stylized state transition is:

$$K_{t+1} = (1 - \delta_K)K_t + \eta_K I_t,$$

$$C_{t+1}^{\text{cap}} = (1 - \delta_C)C_t^{\text{cap}} + \eta_C A_t - \omega_C X_t,$$

where I_t is authorized investment, A_t is newly admitted capability, X_t is revoked or expired capability, and δ terms capture depreciation and obsolescence.

Potential productive output is then represented as:

$$Y_t = \Phi(K_t, C_t^{\text{cap}}, N_t, H_t, E_t),$$

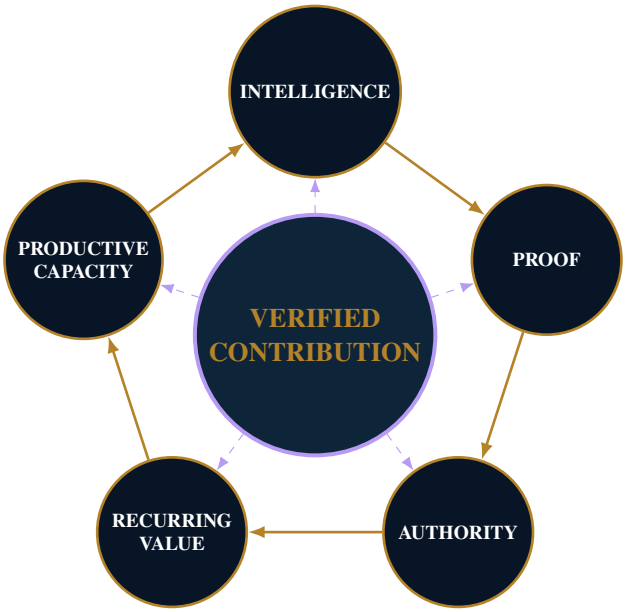
where E_t denotes the energy, compute, data, market, and institutional environment. The objective is not to maximize K_t or Y_t mechanically. It is to expand verified capacity while preserving resilience, legitimacy, environmental constraints, rights, and mission control.

18.3. The productive-capacity frontier

The frontier contains investments whose expected contribution to future mission performance exceeds their complete cost and downside. Examples include:

- better models, evaluations, data rights, and trusted execution infrastructure;
- source-monitoring and evidence systems;
- secure compute and energy contracts;
- laboratories, facilities, logistics, and industrial equipment;
- professional and provider networks across jurisdictions;
- reusable contracts, playbooks, benchmarks, and integration layers;
- distribution, customer access, and public-procurement capability;
- insurance, governance, and resilience capacity.

A capital allocation should be rejected when it enlarges appearance but not verified mission capacity. Headcount, entities, compute, real estate, or media attention are not productive capacity unless they improve the institution's ability to deliver accepted outcomes under full cost and risk.



19. Capital Architecture for Mission-Aligned Scale

The preferred capital sequence is a disciplined prior rather than a universal rule:

$$\begin{aligned}\text{Customer Cash} &\to \text{Earned Credits} \to \text{Awarded Grants}\\ &\to \text{Approved Debt} \to \text{Equity}\\ &\to \text{Strategic Capital} \to \text{Project Finance}.\end{aligned}$$

The sequence seeks to match each resource to the asset and risk it is best suited to finance:

- customer cash proves demand and reduces speculative dilution;
- credits and grants can support eligible research or deployment costs but carry evidence, location, IP, and clawback conditions;
- debt should attach to contracted or otherwise financeable cash flows;
- equity should finance uncertainty that cannot responsibly be levered;
- strategic capital should purchase more than money: access, infrastructure, distribution, or durable alignment;
- project finance should ring-fence infrastructure with identifiable contracts, assets, risks, and cash flows.

19.1. Infrastructure SPVs and milestone finance

Large physical or compute assets should normally be separated from the platform when separation improves financeability, liability control, procurement, insurance, permits, or public-support eligibility. A project vehicle should have its own:

- business-purpose and customer-demand evidence;
- equity, debt, guarantees, and public-support stack;
- land, energy, compute, construction, supply, and offtake contracts;
- governance and reserved matters;
- insurance, security, and operational controls;
- milestone gates, draw conditions, stop conditions, and unwind plan;
- outcome and externality monitoring.

The platform should not assume infrastructure risk merely because infrastructure creates impressive scale. Productive assets are admitted only when demand, permits, capital, delivery capability, and downside containment pass.

19.2. Long-horizon productive scale

At the outer horizon, an institution capable of repeatedly converting validated intelligence into successful enterprises, research systems, funds, infrastructure, and industrial capacity could influence resource allocation at very large scale. The relevant standard is not rhetorical magnitude. It is whether successive generations measurably increase useful output, resilience, knowledge, energy efficiency, and the capacity to solve harder objectives without losing governance.

The long-run prize is not owning an answer. It is building an institution that can repeatedly turn verified intelligence into greater capacity to create.

20. Recursive Self-Improvement

20.1. Chronicle admission

Suppose mission m_1 yields candidate capability k. Capability k may influence mission m_2 only if:

$$\mathrm{Admit}(k) = 1$$

where admission requires evidence, rights, scope, version, currentness, known failure conditions, reviewer acceptance, and revocation rules.

20.2. Fresh successor test

Let g_1 be the predecessor generation and g_2 the successor using admitted capability. On fresh mission m_2, define improvement:

$$\Delta(g_2, g_1 \mid m_2) = \alpha\Delta Q + \beta\Delta V + \gamma\Delta T + \delta\Delta C - \lambda\Delta R - \mu\Delta H, \tag{8}$$

where Q is quality, V verified value, T time efficiency, C cost efficiency, R risk, and H human or reviewer burden. Promotion requires a pre-registered positive threshold and no material governance regression.

Succession Law

Reuse is not improvement. A successor earns authority only through a fresh comparison under controlled or explicitly normalized conditions.

21. Illustrative Mission

Consider the objective:

> Launch an AI infrastructure company that performs core R&D in Montreal, sells to US and EU enterprises, hires specialists in three countries, preserves founder control, seeks non-dilutive support, and prepares a pilot compute facility.

GoalOS may generate six scenarios:

Table 7: Illustrative scenario frontier.

Scenario	Design logic	Principal Proof Debt
Sovereign Core	Canadian/Quebec root; foreign sales without premature subsidiaries; R&D, capability, and governance concentrated	US procurement, EU data roles, transfer-pricing triggers, facility economics
Capital-Efficient Expansion	Core retained; US subsidiary only when customer or financing threshold passes; EOR for first foreign hires	CCPC control, investor rights, US nexus, EOR-to-entity trigger
Market-Access First	US commercial entity activated early; EU partner route; R&D remains in Quebec	Tax and IP agreements, management substance, incentives interaction
R&D and Capability First	Research and capability centered in Quebec; Horizon and consortium paths explored	Funding eligibility, partner IP, commercialization nexus
Infrastructure First	Project SPV selected through power, land, permit, and offtake auction	Permits, customer commitments, project finance, construction risk
Global Functional Optimum	Every function allocated to its top-scoring jurisdiction	Entity proliferation, coordination, PE, transfer pricing, control, unwind cost

The system should not automatically choose the highest raw score. It should identify the Pareto frontier, expose the cost of complexity, and recommend the least irreversible scenario capable of proving the next consequential uncertainty.

22. Commercialization Architecture

22.1. Product ladder

GoalOS's Autonomous Revenue Institution provides the commercial state machine: create permissioned demand, diagnose the objective, qualify value and fit, price within margin policy, contract, collect payment, reserve capacity, execute, prove, renew, expand, and reinvest verified contribution (MONTREAL.AI Research, 2026a).

22.2. Customer value gate

A default qualification law is:

$$V_{12m}^{conservative} \geq 4F,$$

where F is the customer fee. This does not guarantee ROI; it prevents scarce autonomous capacity from being assigned to objectives whose plausible value cannot justify the mission.

22.3. Decacorn and megacorn scenarios

The scale thesis comes from recurring institutional control, not formation fees.

Table 8: Illustrative product ladder; not a current offer or forecast.

Product	**Illustrative price**	**Target gross margin**
Global Launch Readiness Scan	Free–US$10,000	75–90%
Global Architecture Mission	US$25,000–100,000	70–85%
Full Launch Mission	US$100,000–500,000	65–80%
Capital & Public-Support Mission	US$100,000–500,000	70–85%
Infrastructure Launch Mission	US$250,000–2,000,000+	50–75%
Global / Sovereign Expansion Office	US$250,000–1,000,000+ annually	75–85%
Protected Launch Intelligence API	Usage-based	85–95%
Strategic or sovereign programme	US$5,000,000–20,000,000+	40–70%

Table 9: Illustrative operating configurations; not valuation forecasts.

Revenue engine	**Decacorn operating case**	**Megacorn candidate case**
Global / Sovereign Expansion Offices	500 × US$1M = US$500M	3,000 × US$1M = US$3B
Launch and transaction missions	US$250M	US$1B
Protected API / network usage	US$150M	US$1.5B
Marketplace and validation	US$100M	US$0.5B
Infrastructure and strategic programmes	Excluded / optional	US$1B
Illustrative annual scale	**US$1B**	**US$7B**

These scenarios indicate what the institution would need to become: a trusted global launch and expansion standard with high recurring revenue, strong retention, a rights-controlled capability network, professional and provider ecosystems, and demonstrated capital-allocation competence.

Empirical Authority

A category is not established by ambition. It is established by comparative missions, realized outcomes, independent verdicts, and controlled claims.

23. LaunchBench Ω: A Falsifiable Benchmark for Institution Formation

GoalOS should not be evaluated only by whether an expert likes a report or whether a benchmark answer resembles a reference. Institution formation produces delayed, path-dependent outcomes. The benchmark must therefore compare complete mission trajectories: facts gathered, alternatives discovered, evidence quality, decisions made, actions completed, costs incurred, risks retired, value realized, and later consequences.

23.1. Baselines

LaunchBench compares at least four operating modes:

1. **Traditional multidisciplinary team:** independent legal, tax, finance, funding, employment, and infrastructure professionals coordinated by the sponsor.
2. **Static model report:** one or more language-model outputs without persistent workflow, independent verification, or transaction control.
3. **Ungated agentic workflow:** persistent agents with tools and parallelism but without GoalOS proof, authority, memory, and capital gates.
4. **Proof-governed GoalOS mission:** frozen Mission Contract, scenario frontier, bounded AGI Jobs, Evidence Docket, responsible authority, Chronicle, and successor test.

Where appropriate, a fifth baseline should represent the sponsor's actual historical process.

23.2. Mission classes

The benchmark suite should include:

- AI or deep-tech company formation and expansion;
- nonprofit or scientific foundation;
- research consortium with public funding and shared IP;
- cross-border investment fund or SPV;
- market-entry mission without premature entity formation;
- capital and public-support stack;
- compute, laboratory, industrial, or energy infrastructure project;
- economic-development or regional attraction programme;
- restructuring or repair of an existing global architecture.

23.3. Metrics

Table 10: LaunchBench Ω evaluation dimensions.

Dimension	Representative measures	Why it matters
Architecture quality	Expert blind ranking; frontier diversity; fatal issue detection; reversibility.	Tests whether the system finds defensible alternatives rather than one persuasive answer.
Evidence quality	Authoritative-source rate; freshness; contradiction handling; fact traceability; verifier agreement.	Separates plausible output from decision-grade support.
Value	Realized revenue, support, financing, avoided dilution, reduced cost, time saved, risks retired.	Connects intelligence to economic consequence.
Cost and time	Complete labour, model, data, provider, filing, travel, delay, retry, and coordination cost.	Preserves the full denominator.
Authority safety	Unauthorized attempts, blocked transactions, stale receipts, escalation quality, professional sign-off.	Measures whether capability remained within mandate.
Security	Secret exposure, sandbox violations, unapproved communications, tool circumvention, incident recovery.	Tests trajectory-level safety.
Human burden	Sponsor hours, reviewer hours, correction loops, founder rescues, meeting load.	Measures operational leverage rather than model output alone.
Memory precision	False admission, false rejection, scope errors, stale reuse, revocation effectiveness.	Tests whether learning compounds truth rather than error.
Externalities	Employment, concentration, environmental, public-resource, market, and resilience effects.	Prevents private value from hiding institutional cost.

23.4. Institutional regret

Let $V^\star$ denote the highest verified value attainable among feasible architectures revealed ex post, and $V(s)$ the realized verified value of the selected architecture. Define institutional regret:

$$\mathrm{IR}(s) = V^\star - V(s) + C_{\mathrm{avoid}}(s),$$

where C_{avoid} captures preventable compliance failures, rights loss, financing friction, clawbacks, unnecessary entity complexity, and avoidable unwind cost.

This metric rewards systems that avoid disastrous architectures even when those architectures looked attractive ex ante.

23.5. Matched-pair Twin-Run protocol

For each mission class, comparable sponsors or historical cases should be matched by stage, sector, complexity, capital, and time pressure. Baselines receive the same objective and authorized facts. Evaluation criteria and independent reviewers are selected before results are revealed. The benchmark records not only final recommendations but the complete work and authority trajectory.

Where real transactions cannot be duplicated, shadow runs may evaluate recommendations before the sponsor selects one path. The realized path is then observed over time, while counterfactual alternatives receive professional review and explicit uncertainty labels.

23.6. Release criteria

A public claim of superior institution formation should require:

- preregistered metrics and acceptance thresholds;
- independent evaluation and source review;

- complete cost and failure denominator;
- enough missions to separate systematic advantage from one-off luck;
- no material governance regression;
- publication of negative or null results relevant to the claim;
- scoped wording limited to the tested mission classes.

The benchmark is not whether GoalOS sounds more intelligent. It is whether institutions launched through GoalOS work better, at lower complete cost and risk, under independent scrutiny.

24. Implementation Protocol: From Paper to Operating Institution

A paper can specify a category without constituting it. The operating institution requires a sequence that earns authority through progressively harder proof.

24.1. Phase architecture

Table 11: A 1,000-day implementation path.

Phase	Days	Institutional objective	Exit evidence
Constitute	0–30	Freeze category, first product, data boundary, authority model, legal perimeter, and proof programme.	Board-approved Mission Contract; source registry; claim matrix; first three qualified customers.
Prove	31–100	Complete comparable paid missions with independent evaluation and complete cost denominator.	Three paying customers; accepted outputs; one repeat purchase or expansion; zero material trust breach.
Repeat	101–250	Standardize mission products, evidence objects, pricing, provider routing, and source monitoring.	Comparable gross margin; reusable but scoped workflows; first realized route; initial LaunchBench data.
Operate	251–500	Establish recurring Global Expansion Offices and live evidence refresh.	Recurring contracts; source-change alerts; transaction receipts; professional network coverage; customer renewal.
Network	501–750	Open controlled provider, evaluator, and developer interfaces without compromising ranking independence.	APIs; provider scorecards; conflict receipts; portable evidence packs; benchmark participation.
Institution	751–1,000	Demonstrate successor improvement, capital discipline, and durable recurring economics.	Fresh successor outperforms baseline; audited realized value; rights-cleared Chronicle capability; capital-market readiness.

24.2. Reference technology architecture

A production deployment requires more than one standalone application. A minimum architecture includes:

- versioned source registry and change-detection pipeline;
- structured jurisdiction, programme, provider, transaction, and rights graphs;
- customer Mission Contract and evidence data model;
- agent runtime with isolated workspaces, scoped tools, budgets, and traces;

- independent evaluation service;
- professional-authority and provider routing;
- transaction-gate service with signatures and expiry;
- encrypted data rooms and retention controls;
- Chronicle with scope, rights, version, expiry, and revocation;
- monitoring, alerting, incident response, and rollback;
- APIs for customer systems, board workflows, finance, procurement, and document execution.

The browser-local reference institution remains useful as a transparent research, simulation, and governance surface. It should not be confused with the complete multi-user, continuously refreshed, professionally integrated production system.

24.3. Organizational architecture

The institution requires accountable leadership across:

- category and product constitution;
- customer missions and commercial delivery;
- jurisdiction and source intelligence;
- legal, regulatory, privacy, sanctions, and professional-authority governance;
- fiscal, funding, finance, and capital architecture;
- agentic security and trusted execution;
- evidence, evaluation, benchmark, and claims;
- provider and partner ecosystem;
- infrastructure and strategic programmes;
- Chronicle, rights, and recursive improvement.

No single founder or model should remain the permanent exception handler. Repeated exceptions must be converted into an explicit product, policy, delegated authority, decline rule, or escalation path.

24.4. The first commercial proof run

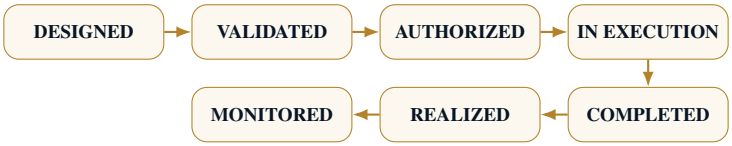
The first proof should remain deliberately narrow:

1. Select one segment with coupled, high-value launch decisions, such as institutionally financed AI, deep-tech, clean-tech, life-sciences, or industrial technology.
2. Sell three substantially comparable Launch Missions to unrelated customers.
3. Freeze acceptance criteria and the public claim before execution.
4. Run independent professional and benchmark review.
5. Cause at least one recommendation to become a realized route: award, financing, entity, hire, market activation, contract, infrastructure milestone, or demonstrably avoided failure.
6. Obtain a repeat purchase and one recurring Expansion Office.
7. Publish a rights-cleared Receipt Pack containing scope, evidence, cost, verdict, limitations, realized outcome, and Chronicle decision.

This sequence converts a compelling architecture into commercial authority.

25. Claim Ladder and Publication Discipline

GoalOS should distinguish seven states:



A system-generated architecture may be described as designed. It becomes validated only when current evidence and qualified review support it. It becomes authorized only when the responsible party approves. It is completed only when acceptance tests pass. It becomes realized only when the intended external value or risk reduction actually occurs. It remains credible only while monitored.

Publication Law

Every public claim must identify the object, mission, comparator, evidence state, time period, limitations, and responsible reviewer. The broadest true statement is preferred over the strongest unproven statement.

26. Empirical Proof Programme

The first proof should remain narrow. A credible Commercial Proof Run requires:

Table 12: Minimum evidence gates for the first external proof programme.

Gate	Minimum
Unrelated paying customers	3
Substantially similar Launch Missions	3
Independently verified customer outcome	1
Public-support, fiscal, financing, or market route actually realized	1
Completed entity, market, talent, capital, or infrastructure activation	1
Separate repeat purchase or expansion	1
Active Global / Sovereign Expansion Office	1
Rights-cleared Chronicle capability	1
Fresh successor improvement	1
Material trust breaches	0

A public-safe Receipt Pack should include the paid scope, preregistered acceptance criteria, evidence summary, independent verdict, complete cost denominator, implementation result, limitations, repeat purchase, Chronicle decision, and exact claim being made.

27. Risks and Failure Modes

28. Research Agenda

The architecture suggests a programme of empirical work:

1. **Scenario quality**: compare GoalOS scenario frontiers with independent multidisciplinary teams.
2. **Value of Proof**: estimate which evidence most changes institutional decisions.

Table 13: Principal risks and institutional controls.

Risk	**Failure mode**	**Control**
False legal or tax certainty	Strategic output is treated as binding authority	Current official sources, exact facts, professional receipt, expiry, transaction gate
Grant theatre	Unawarded support enters runway or drives the project	Awarded/earned/collected states separated; support excluded from base cash
Shell-company optimization	Tax score dominates substance and operating reality	Business-purpose memo, people/risk map, entity-cost penalty, local substance review
Scenario monoculture	One model anchors on one attractive structure	Six diverse scenarios, independent red team, counterfactual and stop conditions
Metric gaming	System optimizes headline tax, grant size, or number of entities	Verified launch value includes full coordination, risk, delay, and unwind denominator
Professional conflict	Adviser compensation biases the ranking	Compensation disclosure, separated evaluation and referral functions, independent review
Data leakage	Sensitive customer facts enter a public tool	No-secrets gate, protected deployment, minimization, access logs, retention and deletion
Capability overreach	One successful case becomes universal memory	Narrow scope, versioning, known failures, expiry, revocation, fresh transfer
Infrastructure overinvestment	Large assets are funded before demand and permits	Customer commitments, project SPVs, milestone finance, stop and unwind gates
Founder dependence	Every exception returns to one individual	Fixed products, clause matrices, delegated authorities, explicit decline and escalation paths

3. **Graph efficiency**: measure when parallel AGI Jobs reduce elapsed time and when coordination cost dominates.
4. **Verifier independence**: test context separation, blind evaluation, and evaluator diversity.
5. **Funding-stack realization**: compare gross discovered support with awarded and collected net cash.
6. **Transaction safety**: measure prevented incomplete or contradictory transactions.
7. **Chronicle precision**: estimate false-admission and false-rejection rates for reusable capability.
8. **Successor transfer**: determine whether admitted capability improves fresh missions without governance regression.
9. **Recurring value**: measure whether the Global / Sovereign Expansion Office reduces risk, cost, delay, or missed opportunity.
10. **Institutional externalities**: measure concentration, regulatory, labour, environmental, and infrastructure effects.

29. Limitations and Falsification Boundary

The architecture is intentionally ambitious. Its failure conditions must be stated as clearly as its upside.

29.1. Coverage is not completeness

A large jurisdiction or funding registry does not imply complete or current coverage. Official sources may be ambiguous, programmes may close, subnational rules may matter, and the exact facts may change legal treatment. GoalOS should expose source age, confidence, professional dependencies, and missing coverage instead of compressing them into a universal-completeness claim.

29.2. Optimization depends on the objective

No architecture is optimal independent of the sponsor's values, control preferences, time horizon, risk posture, externalities, and prohibited actions. A model can make those trade-offs explicit; it cannot legitimately invent them.

29.3. World models remain imperfect

Probabilities for funding awards, permitting, customer conversion, tax disputes, construction, financing, and geopolitical change are difficult to calibrate. Tail events and regime shifts may invalidate historical data. Robust optimization reduces fragility but does not eliminate uncertainty (Ben-Tal et al., 2009; Knight, 1921).

29.4. Professional judgment is not fungible

Legal, tax, fiduciary, audit, immigration, and public-authority decisions depend on jurisdiction, facts, responsibility, and professional standards. Provider receipts improve accountability but do not make every professional conclusion comparable or correct.

29.5. Recursive systems can amplify error

A Chronicle with weak admission can turn one successful case into universal doctrine. A Chronicle with excessive caution can prevent useful transfer. Admission precision, revocation, and fresh-successor testing are therefore empirical problems, not solved definitions.

29.6. The benchmark may be gamed

Any published metric can become a target. LaunchBench requires evaluator independence, hidden cases, multiple mission classes, complete denominators, negative-result publication, and periodic redesign. There is no final benchmark for an evolving institution.

29.7. Civilizational horizon is not a present capability claim

The architecture includes a long-run path from intelligence to productive capacity because this is the economic reason the category may matter. The existence of that path does not establish that GoalOS, MONTREAL.AI, any model, or any current organization has reached the endpoint. The claim is a design horizon and research hypothesis whose intermediate gates must be earned.

The paper is falsified where GoalOS fails to outperform appropriate alternatives after complete cost, risk, authority, security, and realized outcomes are measured.

30. Conclusion

The world already contains mature tools for incorporation, hiring, equity, capital administration, legal research, public support, procurement, and expansion. GoalOS Global Launch Ω does not need to replace all of them at once. It can coordinate them around the outcome the sponsor actually wants.

Tell GoalOS what you want to build.

It turns the world into your launch stack—and the objective into a verified operating institution.

The paper's strongest claim is institutional rather than technological. Better models make research, drafting, analysis, and orchestration dramatically cheaper. The durable opportunity is to convert that capability into a governed system of objectives, permissions, bounded work, evidence, authority, selective memory, recurring control, and productive reinvestment.

Final Institutional Law

No consequential objective, no mission. No exact facts, no jurisdictional or fiscal result. No real dependency, no graph edge. No Evidence Docket, no value claim. No responsible authority, no transaction. No Chronicle admission, no future influence. No fresh transfer, no improvement claim. No verified contribution, no productive reinvestment.

A. Minimum Launch Mission Contract

A production Mission Contract should contain at least:

1. accountable sponsor and decision authority;
2. objective, scope, exclusions, and deadline;
3. functions to optimize;
4. data classes and permitted sources;
5. models, tools, providers, and environments;
6. budget, cost ceiling, and value hypothesis;
7. acceptance tests and evaluator;
8. professional-authority dependencies;
9. Evidence Docket requirements;
10. rights, confidentiality, publication, and reuse;
11. stop, pause, escalation, and rollback rules;
12. Chronicle eligibility and successor-test conditions.

B. Evidence Docket Schema

Docket object	Required content
Objective record	Sponsor, decision, value at stake, deadline, acceptance criteria
Fact register	Verified facts, source, date, owner, confidence, expiry

Assumption ledger	Assumption, sensitivity, evidence needed, decision impact
Claims matrix	Material claim, supporting evidence, contradiction, status
AGI Job ledger	Worker, tools, data, budget, output, failure, verifier
Cost ledger	Models, data, professionals, support, travel, filings, capital, delay
Authority register	Approval, scope, owner, effective date, expiry, revocation
Transaction receipts	Signed agreements, filings, payments, awards, permits, acceptance
Outcome record	What happened, customer value, adverse outcomes, delayed effects
Rights register	Customer rights, platform rights, third-party rights, reuse conditions
Chronicle decision	Admit, admit with limits, repair, reject, revoke, or expire
Successor result	Baseline, fresh mission, metrics, improvement, regressions, verdict

C. Canonical Operating Doctrine

Discover globally.
Decompose function by function.
Generate competing architectures.
Challenge adversarially.
Resolve locally.
Govern institutionally.
Execute only within authority.
Record the full truth.
Admit only what passed.
Revoke when stale.
Reinvest only verified value.

D. Proof-Carrying Action Schema

Field	Required content
Action ID	Immutable identifier, mission, version, creator, creation time.
Intent	Exact objective and the decision or claim the action supports.
Scope	Entities, jurisdictions, systems, amounts, counterparties, data classes, and duration.
Authority	Accountable owner, authority level, effective period, revocation, and escalation.
Inputs	Approved facts, sources, artifacts, assumptions, and confidentiality class.
Tools	Allowed models, APIs, browsers, code, files, networks, providers, and environments.
Budget	Spend, tokens, compute, time, transaction, and retry limits.
Output schema	Required structure, citations, confidence, contradictions, and machine-readable fields.
Acceptance	Preregistered tests, thresholds, evaluator, and failure route.
Evidence	Required source, logs, receipts, signatures, and independent verification.
External effect	Whether the action may communicate, publish, file, sign, transfer, deploy, or spend.
Rollback	Reversal, containment, incident response, compensation, and recovery.
Docket return	Objects and hashes returned to the Evidence Docket.

Field	Required content
Chronicle status	Ineligible, candidate, admitted, limited, expired, revoked, or rejected.

E. LaunchBench Mission Record

A benchmark mission record should contain:

1. sponsor class, sector, stage, objective, value at stake, and decision deadline;
2. authorized facts, hidden facts, prohibited data, and source dates;
3. baseline operating modes and their budgets;
4. evaluator selection, blinding, conflicts, and preregistered metrics;
5. architectures generated and material alternatives omitted;
6. evidence quality, Proof Debt, authority requirements, and security events;
7. complete cost, time, human burden, and coordination denominator;
8. decision, implementation, realization, and delayed outcomes;
9. institutional regret and counterfactual review;
10. public claim, limitations, Chronicle decision, and successor test.

F. Chronicle Admission Checklist

Capability may enter Chronicle only when all applicable questions can be answered:

- What exact task or decision did the capability support?
- Which evidence and sources support it, and when do they expire?
- Who owns the rights, and what customer or third-party material is excluded?
- What models, tools, environments, prompts, policies, and professional inputs produced it?
- What failed, contradicted, or required repair?
- What mission classes and facts define its scope?
- Who independently reviewed it, and under which acceptance criterion?
- What conditions trigger repair, expiry, or revocation?
- Has it improved a fresh mission against the predecessor without material governance regression?

G. Public Receipt Pack

A public-safe proof package should include:

- paid or otherwise authorized scope;
- preregistered acceptance criteria;
- public-safe mission facts and comparator;
- evidence and source summary;
- independent verdict;
- complete cost denominator;

- implementation result and realized value;
- adverse outcomes and limitations;
- repeat purchase or renewal status;
- Chronicle decision and exact claim being made.

H. Extended Canonical Operating Doctrine

Discover globally.
Decompose function by function.
Generate competing architectures.
Simulate consequential uncertainty.
Challenge adversarially.
Resolve locally.
Govern institutionally.
Execute only within authority.
Record the full truth.
Admit only what passed.
Revoke when stale.
Reinvest only verified value.
Prove every claim on the mission to which it applies.

Final Institutional Law

No consequential objective, no mission. No exact facts, no jurisdictional or fiscal result. No real dependency, no graph edge. No Evidence Docket, no value claim. No responsible authority, no transaction. No Chronicle admission, no future influence. No fresh transfer, no improvement claim. No verified contribution, no productive reinvestment. No complete denominator, no state-of-the-art claim.

References

Anthropic (2026). *Claude Code Documentation: Agent Teams*. https://code.claude.com/docs/en/agent-teams. Accessed 2026-07-24.

Ben-Tal, Aharon, Laurent El Ghaoui, and Arkadi Nemirovski (2009). *Robust Optimization*. Princeton University Press.

Campbell, Donald T. (1979). "Assessing the Impact of Planned Social Change". In: *Evaluation and Program Planning* 2.1, pp. 67–90.

Carta (2026). *The End-to-End Suite Connecting Private Capital*. https://carta.com/. Official product page; accessed 2026-07-24.

Coase, Ronald H. (1937). "The Nature of the Firm". In: *Economica* 4.16, pp. 386–405.

Deel (Sept. 2025). *Deel-Owned Infrastructure and Entities Available in Over 100 Countries*. https://www.deel.com/blog/deels-owned-infrastructure-and-entities-are-now-available-in-over-100-countries/. Accessed 2026-07-24.

Dixit, Avinash K. and Robert S. Pindyck (1994). *Investment under Uncertainty*. Princeton: Princeton University Press.

European Parliament and Council of the European Union (2024). *Regulation (EU) 2024/1689 Laying Down Harmonised Rules on Artificial Intelligence*. https://data.europa.eu/eli/reg/2024/1689/oj. Official Journal text; accessed 2026-07-24.

Goodhart, Charles A. E. (1975). "Problems of Monetary Management: The U.K. Experience". In: *Papers in Monetary Economics*. Reserve Bank of Australia.

Jensen, Michael C. and William H. Meckling (1976). "Theory of the Firm: Managerial Behavior, Agency Costs and Ownership Structure". In: *Journal of Financial Economics* 3.4, pp. 305–360.

Knight, Frank H. (1921). *Risk, Uncertainty and Profit*. Houghton Mifflin.

MONTREAL.AI Research (July 2026a). *GoalOS Autonomous Revenue Institution*. Tech. rep. Internal working paper. MONTREAL.AI.

– (July 2026b). *GoalOS Sovereign Productive Intelligence*. Tech. rep. Internal institutional thesis. MONTREAL.AI.

– (July 2026c). *The Capability–Price Lag and GoalOS Economic Architecture*. Tech. rep. Internal working paper. MONTREAL.AI.

National Institute of Standards and Technology (Jan. 2023). *Artificial Intelligence Risk Management Framework (AI RMF 1.0)*. Tech. rep. NIST AI 100-1. Accessed 2026-07-24. U.S. Department of Commerce.

OpenAI (2026). *OpenAI Agents SDK: Guardrails*. https://openai.github.io/openai-agents-python/guardrails/. Accessed 2026-07-24.

Remote (2026). *Employer of Record Pricing and Global Hiring Coverage*. https://remote.com/pricing. Official product page; accessed 2026-07-24.

Stripe (2026). *Stripe Atlas: Incorporate Your Startup in Delaware*. https://stripe.com/atlas. Official product page; accessed 2026-07-24.

Williamson, Oliver E. (1985). *The Economic Institutions of Capitalism*. New York: Free Press.