MONTREAL.AI / QUEBEC.AI

AUTONOMOUS ENTERPRISES
ACTIVATE. EVOLVE. SCALE.

SUCCESSOR Ω Harness

Constructing and Financing Reusable Productive Capability

Powered by TRANSFORMATIONAL INTELLIGENCE ∞

Vincent Boucher
MONTREAL.AI • QUEBEC.AI
Montréal, Québec, Canada

22 October 2026 • Source-grounded research manuscript

Abstract

An enterprise should leave behind more than its delivery: it should improve the conditions under which the next useful enterprise can begin. We realize this principle as an operating path for the existing SUCCESSOR Ω Harness, separating off-chain capability construction from on-chain admission, commitments, financing and settlement. A concrete local companion constructs a bounded interval-measurement instrument and applies it to two distinct offerings: demand exposure reports and infrastructure capacity-violation certificates. The retained instrument removes repeated raw-overlap scans; each offering still receives a fresh semantic interpretation and independent output check. Independent setup, ordinary harness reuse and instrument reuse are compared, including construction and verification work, failed transfers and recovery. A strong conventional sweep implementation is credited explicitly; no new sweep algorithm or general recursive self-improvement is claimed. We map the owner-designated Ethereum five-contract deployment to exact source functions and preserve the separate USDC and $AGIALPHA Jobs routes. The deployed inventory, open harness candidate, replayed worker examples and new simulations are distinguished throughout. The contribution is an inspectable integration specification and runnable construction experiment, not autonomous incorporation, observed demand, investment performance or approved mainnet operation. Useful baselines can activate now; bounded improvement should accelerate their successors rather than prevent their launch.

Build what makes the next enterprise possible.

Activate available capability. Operate accountably. Retain what survives a fresh test.

FIND YOUR ALPHA. BUILD YOUR SUCCESSOR Ω.

github.com/MontrealAI/AGI-Alpha-Agent-v0

1 The enterprise after the first result

A planning team needs to know how much demand accumulated during several windows. An infrastructure operator needs the exact intervals during which simultaneous link loads exceed capacity. Both receive public, bounded integer interval records. The first needs an exposure report; the second needs an overload certificate. Repeatedly inspecting every raw interval for every requested window is an inherited production method, not part of either beneficiary's required outcome.

Construct a measurement instrument that converts interval endpoints into disjoint segments with a constant aggregate load on each segment. It can integrate exposure over a window or identify every capacity violation. The next offering inherits the qualified instrument, its input convention and its measurement evidence. It does not inherit the first customer's consent, the first workload's segmentation, or a decision that the new result is correct. A separate checker still evaluates the actual requested result. The instrument removes repeated interior-overlap scanning from production; it does not remove fresh input interpretation or acceptance.

This is deliberately recognizable engineering. Sweep lines, integration and typed interfaces already exist. A competent engineer can build the same instrument, and our strong conventional comparator does. The transformational move is to ask which alleged necessity belongs to the outcome and which belongs only to the inherited setup, then construct and test a replacement. The engineering contribution is an executable retained-capability boundary aligned to an existing enterprise harness and its real financing interfaces. The evidence does not support a new general optimization algorithm or superiority over expert engineering.

What work do enterprises repeatedly perform because of an inherited setup, and what could we construct so future enterprises no longer need to perform it?

Keep the beneficiary's outcome, permitted actions and acceptance obligations fixed. Change the means.

The ambition is a growing diversity of useful enterprises. It has no artificial ceiling of one organization or one offering. Yet a hundred renamed copies are not a hundred kinds of productive capability. Useful expansion changes a beneficiary, deliverable, instrument or production possibility and pays for the new interpretation, qualification and operating burden. The next enterprise is a test of inheritance, not a rhetorical multiplication of the first.

1.1 Lineage and the actual contribution

This manuscript continues the pinned 22 October *Activate. Evolve. Scale.* paper, revision `3717c9ffb677751e1733a9ac7e010f5c877ef5d1`, and preserves the earlier *Build the Means of Building*, *Transformational Intelligence* and *Invent the Economy* sources and URLs [4, 6, 5]. Historical result identities remain historical; the new companion is a clearly identified reconstruction and extension, not independent reproduction of unavailable original raw records.

Three contributions are reviewable here. First, an outcome-preserving construction protocol makes a retained productive instrument useful across distinct offerings and exposes the cost of its inheritance. Second, an exact-source operating specification connects that path to the deployed Ascension suite without confusing commitments and votes with industrial truth. Third, a runnable local companion makes successful delivery, unsuccessful transfer, bounded resources, retained history and independent checking inspectable. The existing harness remains the platform. The companion supplies a small integration experiment and proposed adapter boundary rather than another control plane.

2 Preserve the outcome; reconstruct the means

Write a mission as $M = (I, O, A, P, B, D, R)$: allowed inputs, required output, acceptance predicate, permitted actions, resource ceiling, deadlines and applicable rights. These are operational obligations. A proposed representation change is eligible only when it preserves the relevant meaning of this tuple or makes an explicit, separately accepted change. A faster answer to a different question is not an improvement.

A retained capability is $K = (v, \mathcal{D}, f, \mathcal{I}, E, L)$: a version, supported domain, executable method, typed interface, evidence and provenance/lineage. Reuse requires a fresh compatibility check applicable(K, M) and a fresh decision under A. A schema can reject malformed records while accepting a semantically wrong unit, interval convention or objective. Hash equality can identify bytes while failing to establish their usefulness. Therefore interface compatibility, artifact identity and task acceptance are separate tests.

For the companion, intervals are half-open $[a, b)$ with integer endpoints and nonnegative integer loads. Let

$$L(t) = \sum_i w_i \mathbf{1}_{[a_i, b_i)}(t). \tag{1}$$

The instrument sorts endpoints and constructs a piecewise-constant representation of L. An exposure request returns $\int_u^v L(t)\,dt$ exactly in integer load-time units. A capacity request returns the maximal constant-load intervals where $L(t) > c$, with its declared reporting convention. These related mathematical operations serve different outcomes. They do not certify whether a supplied demand series is representative or whether an engineering capacity threshold is safe.

For every new dataset the instrument constructs its own segments. The inherited asset is executable machinery and its bounded qualification evidence, not cached answers or a claim that old measurements apply to new records. The independent checker uses raw intervals and a different defining calculation. Consequently the construction can reduce production work while verification remains expensive. The paper reports that cost rather than hiding it behind the word "reusable."

2.1 A practical construction protocol

Begin with an available implementation and deliver one bounded result. Observe a recurring obstacle in that actual procedure. State the counterfactual necessity: "every new query scans all raw intervals." Propose a concrete alternative and list what still must happen: input validation, endpoint processing, current acceptance and accountable recovery. Execute both methods on matched requests. Retain the alternative only when its total burden is justified over the intended reuse horizon. Credit the conventional method whenever it provides the same result.

No mandatory discovery ritual precedes activation. A small report can run locally without a wallet, token purchase, funding campaign or recursive self-improvement system. More demanding models and adapters can join the same acceptance boundary when their execution is authorized and useful. A fixed library of supported work families is a sensible initial boundary; it is not unrestricted execution of invented code.

3 One architecture, six responsibilities, continuous operation

AGI ALPHA organizes agents, tools, memory, work and validation. TRANSFORMATIONAL INTELLIGENCE ∞ constructs better conditions for discovery and production. The six responsibilities remain the foundation. The SUCCESSOR Ω Harness turns them into a working path; it is neither a seventh stage nor a requirement for six model calls per task.

ACTIVATE an available capability around a worthwhile mission. **OPERATE** continuously: execute,

Table 1: Six responsibilities, each with a concrete obligation. A single tool call can serve several; some tasks require no model.

Responsibility	Work to perform	Boundary to preserve
OUT-LEARN	Find relevant evidence and uncertainty	Evidence relevance is not authority
OUT-THINK	Replace a restrictive representation	Preserve the outcome and constraints
OUT-DESIGN	Construct an enabling instrument or production boundary	Test an executable alternative
OUT-STRATEGISE	Identify necessary information and actions	Include qualification and recovery cost
OUT-EXECUTE	Deliver and check useful results	Retain valid progress and failed attempts
OUT-EVOLVE	Qualify means a successor can improve	New use needs fresh applicability and acceptance

validate, account, recover and retain. **EVOLVE** the means of creating and delivering useful offerings. **SCALE** the diverse offerings and shared capabilities that prove useful. OPERATE is the durable center even though the title emphasizes three verbs. Without execution, validation and recovery, a succession of plans produces no accepted work.

An enterprise record binds a mission to explicit resources, current authority, an executable version, review obligations and retained outcomes. The paper uses "enterprise" in that software-operating sense. Incorporation, legal personhood, ownership of intellectual property, customer consent and permission to use infrastructure require their own valid bases. Neither NFT parentage nor copied configuration supplies them.

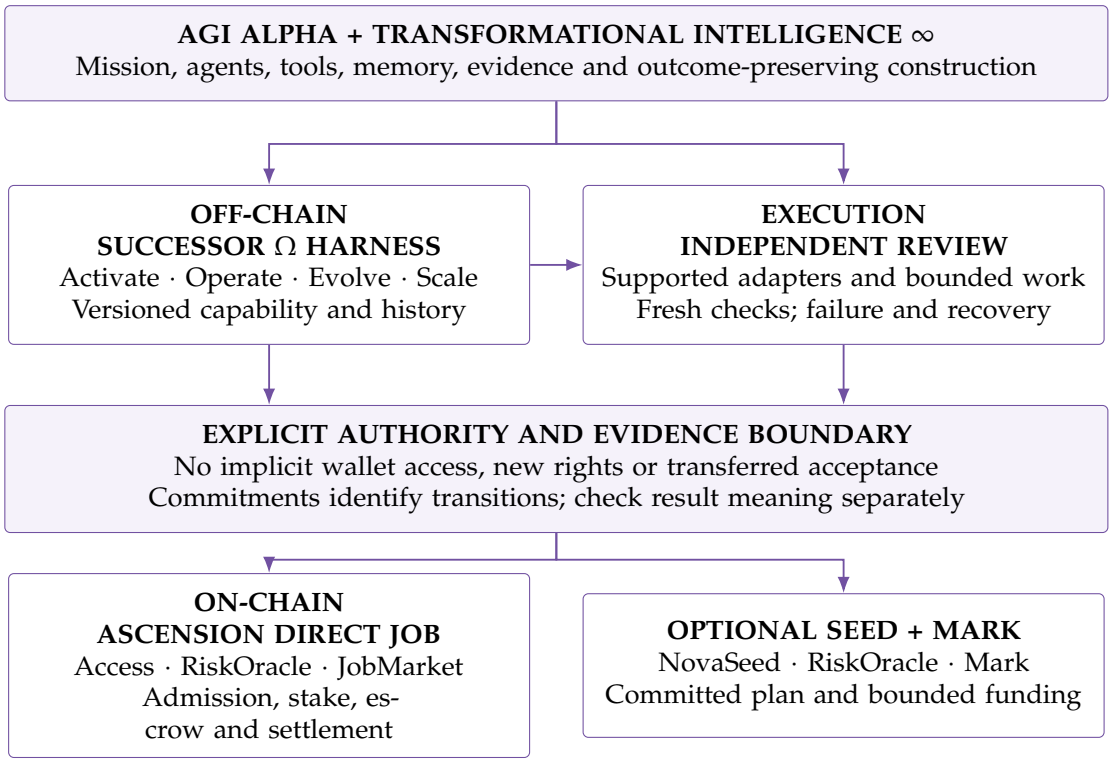


Figure 1: Off-chain intelligence constructs and checks productive capability. On-chain components enforce their actual admission, escrow, decision and transfer rules. Evidence references connect the layers; neither layer by itself establishes customer demand or physical truth. The reference companion executes locally and simulates the marked transaction boundary.

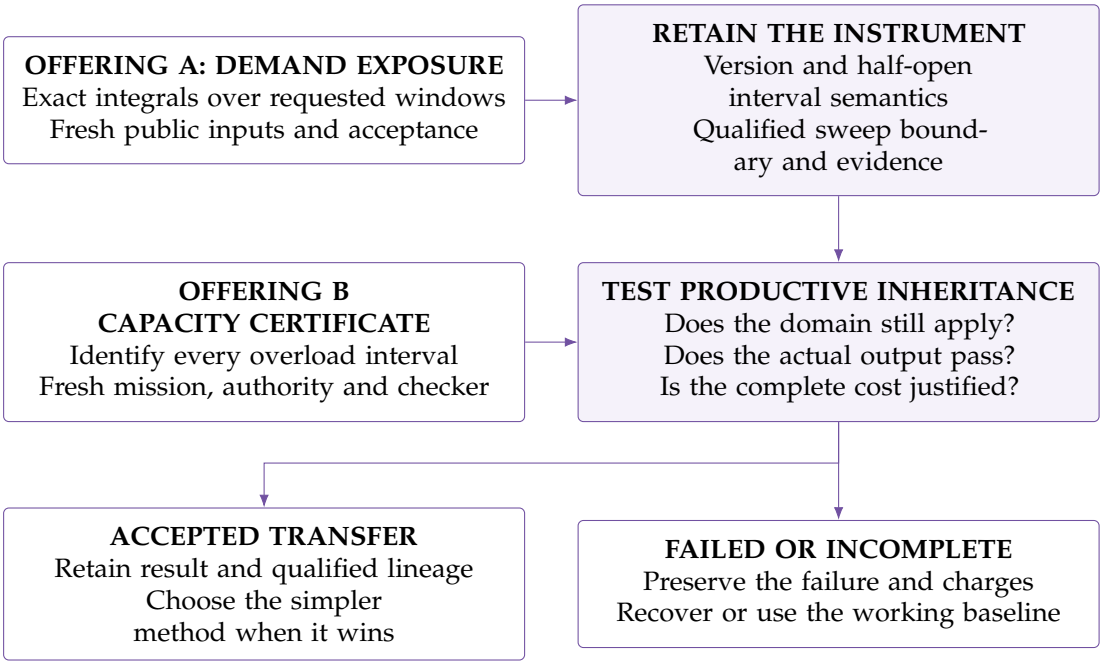


Figure 2: A successor inherits the instrument and applicable evidence, then qualifies the new meaning and result. A rejected transfer stays in the history. On-chain parentage can record lineage; only the off-chain qualification procedure tests productive inheritance.

4 The Ascension vision as an operating architecture

"α-AGI Ascension: Powered by $AGIALPHA" becomes useful when each named role has an actual responsibility and evidence boundary. No new subsystem is needed merely to honor a name.

Role	Mechanism in this architecture	What it does not establish
α-AGI Insight	Develop testable hypotheses about unmet needs, beneficiaries and missions; retain disconfirming evidence	Beyond-human foresight, a trillion-dollar opportunity or observed demand
α-AGI Nova-Seeds	ERC-721 lifecycle commits a capsule, immutable Merkle plan root, evidence and bounded job count; disclose plan material so consumers can verify commitments	Company equity, IP title, revenue rights or truth of disclosed claims
α-AGI MARK	Supported curve-priced campaign contributions; phase-specific operations and committed-job routing	A general-purpose DEX, unrestricted treasury or guaranteed liquidity
α-AGI Sovereign / Business	An admitted business identity operates a bounded mission with controlled resources, accepted outputs and accountable successors	Incorporation, legal sovereignty or autonomous legal agency
Large Scale α-AGI Marketplace	Here: `AscensionJobMarket`, with funded jobs and source-defined bounded scoring, assignment, validation and settlement	Universally fastest/cheapest selection; the separate Jobs v0 suite
α-AGI Agents	Eligible participants execute permitted off-chain work and submit result commitments and evidence URIs	A token holder's automatic eligibility or result correctness
α-AGI Jobs	Define deliverables, funded rewards, deadlines and explicit acceptance evidence	Demand, profitability or universal work coverage
α-AGI Architect	Propose, evaluate and version improvements to activation, instruments, execution and qualification	Unbounded self-modification or new spending authority
Validator Council / α-AGI CLUB	Actual eligible, independent, staked reviewers make contract-supported decisions	Votes, incentives and tests being equivalent to proof of truth

A FusionPlan should be disclosed as a canonical set of committed job terms and its Merkle witnesses, together with the exact hashing/encoding convention consumed by the pinned contracts. The seed records commitments and metadata, not a complete business automatically executed by an NFT. A consumer verifies the disclosed plan against those commitments before relying on it. A hash alone supplies no disclosure, encryption quality, feasibility evidence or license.

The intended identities are `name.alpha.agi.eth`, `name.alpha.agent.agi.eth` and `name.alpha.club.agi.eth`. The access contract records an already-owned name under its configured role root; it does not mint a subname. Owner-issued admission, expiry, role registration and current ENS ownership are distinct conditions. Section 5 details the pinned source rules and the missing root-observation evidence; intended spelling is not a live eligibility claim. The direction is a validator-gated constellation of enterprises. Achieved general intelligence, universal foresight and unconstrained autonomous sovereignty are not results of this manuscript.

5 The deployed commitment and settlement boundary

The reference uses the owner-designated five-contract Ascension graph, rather than a newer-looking deployment or either separate Jobs system. We retrieved the deployment pointer, manifest and source on 22 October 2026. The manifest identifies Ethereum chain ID 1 and source revision `c0fc5217e364a39a4cd123d17a8317c06b80f820`; it reports five Etherscan exact matches and scoped authority observations at block 26,220,072, hash `0x8a3357d17a61a8f37c99dc836ac1176c6ee44b6261a81c41893527046c88a088`.[9, 8, 7] These are retained provider observations, not a fresh chain query in this study.

Component	Manifest address
AscensionAccess	`0x9eE41dF6d076Fa4F89b102b76DEf1c503a1e7C3A`
NovaSeed	`0x31BD0736DcD55DA2f9a2BB586C8a98695F86f19E`
AscensionRiskOracle	`0x198FC6EBee21c6A0B34E4d055AFD6D811AA6dE58`
AscensionJobMarket	`0xc1d3d7fb064C23c46C4EB20d24ab6EA650FF0ceE`
AscensionMark	`0x6D5323b2609E36fDc4dB3B2501B7BcF0E131033c`
$AGIALPHA	`0xA61a3B3a130a9c20768EEBF97E21515A6046a1fA`

All copied addresses were compared with the manifest or its token getters; token precision is 18 decimals. The dated record reports agent and validator minimum stakes of 10^{19} base units, oracle quorum 2 and maximum risk 2,000 basis points. The minima and oracle parameters are constructor immutables in this source. It does not follow that a participant currently has sufficient stake, valid admission, live ENS ownership, approved allowance or an available reviewer.

Identity and authority. `AscensionAccess.Role` encodes Business = 0, Agent = 1, Validator = 2. Its constructor derives the roots `alpha.agi.eth`, `alpha.agent.agi.eth` and `alpha.club.agi.eth`. Registering `label` therefore records `label.alpha.agi.eth`, `label.alpha.agent.agi.eth` or `label.alpha.club.agi.eth`. Labels contain 1–63 bytes: lowercase ASCII letters, digits and interior hyphens only. The registration call checks the current ENS owner, or the current unexpired NameWrapper owner when the registry points to the configured wrapper. It does not use a resolver address as ownership.

Registration and admission are separate. Only the Access owner can call `setAdmission`; a positive admission requires nonzero evidence and expiry strictly after now and at most 365 days ahead. Setting expiry to zero revokes it. `eligible(account,role)` requires both unexpired admission and current ownership of that account's recorded role name. The source fixes the root construction, but this study did not read current root getters or any named participant from mainnet. It consequently asserts no live eligible identity. A software business identity is neither incorporation nor legal personhood.

Configuration is not operation. The Oracle's `bindMarketplace(address)` and JobMarket's `bindMark(address)` are owner-only, one-time bindings with reciprocal graph checks. Their

finalized transactions are reported at blocks 26,219,975 and 26,220,058. Ownable2Step ownership is transferable storage, not immutable governance. The manifest records the governor `0xd76AD27a1Bcf8652e7e46BE603FA742FD1c10A99` for Access, Oracle and JobMarket, with zero pending owners at the authority block. NovaSeed and MARK have no owner administrator in this source. There is no global Solidity pause or unpause function. The manifest's `activationAllowed=false` and `readyForPublicFunding=false` are off-chain readiness declarations; they do not stop a Solidity call. The published pointer still states that activation and public funding are not approved. Independent security review, external authority inventory, exact-source real-dependency fork evidence and participant commissioning remain separate requirements.

5.1 Callable feature map

Tables 3–5 map the claims to the exact named callable, actor, transition, event and representative reverting condition. Full typed argument lists and ABI provenance are included in the companion's call map. Events record contract decisions; they do not certify external semantics.

Table 3: Identity, plan and risk calls at the pinned source.

Callable / actor	Prerequisite and state transition	Event / representative failure
`register(role,label)` / name owner	Canonical label and live ENS ownership; stores the role node.	`Registered`; noncanonical label or ownership absent.
`setAdmission(account, evidence,until)` / Access owner	Valid account and bounded expiry; writes or revokes admission.	`AdmissionSet`; invalid admission or non-owner.
`seal(...)` / eligible business	Nonzero commitments, 1–128 jobs, 1–512-byte URI; optional parent currently owned by caller. Mints NFT and immutable genome.	`SeedSealed`; empty commitment or parent ownership.
`deposit(amount)`, `withdraw(amount)` / validator	Deposit: eligible validator and exact token receipt. Withdrawal: positive balance and now strictly beyond `lockedUntil`.	`StakeChanged`; validator identity or locked stake.
`setJobAvailability( business,until, capacity)` / validator account	Nonzero business; expiry zero or within 120 days; capacity at most 128. Records caller-offered availability for a business and shared caller capacity.	`JobAvailability`; availability bounds.
`openReview(id)` / eligible seed owner	Prior round strictly expired. Starts a seven-day round tied to the present owner.	`ReviewOpened`; seed owner or review still active.
`attest(id,riskBps, policyApproved, evidenceHash)` / staked validator	Current review/owner; validator differs from owner and original creator; one vote, nonzero evidence. Locks stake until round expiry; increments yes or no.	`RiskAttested`; duplicate, decided, expired or non-independent review.
`green(id)` / any reader	Requires quorum yes, unexpired round, same owner, eligible business and still-qualified supporting validators.	View only; returns false when a condition lapses.

5.2 The minimal direct-job journey

Start without a wallet for local mission design, evaluation and unfunded computation. Add the chain only for a separately authorized funded commitment. The exact source uses

$$\texttt{Spec} = (\texttt{string goal, string successMetric, uint96 bounty,}\\ \texttt{uint32 duration, uint16 priceWeight}).$$

Table 4: Direct and MARK-originated job calls.

Callable / actor	Prerequisite and state transition	Event / representative failure
`deposit(amount)`, `withdraw(amount)` / agent	Deposit requires eligible Agent and exact receipt; withdrawal cannot consume locked collateral.	`StakeChanged`; agent identity or locked stake.
`post(spec, auctionSeconds, validators)` / eligible business	Bounds and three independent reviewers; reserves capacity and escrows bounty from caller. Creates Auction.	`JobPosted`; job bounds, committee conflict or unavailable capacity.
`postFromSeed( business,...)` / bound MARK only	Same job checks; MARK is token payer and refund client, nominated business is enterprise principal.	`JobPosted`; MARK only.
`bid(id,price, duration)` / eligible agent	Open auction, no role conflict, at most 32 distinct bidders; locks the job bond.	`BidPlaced`; invalid bounds, duplicate or insufficient free stake.
`award(id)` / any caller	Auction ended. Chooses lowest eligible score fitting remaining execution horizon; unlocks losers. No winner produces Refunded.	`Assigned` or `Closed`; auction not ended.
`submit(id,resultHash, resultURI)` / assigned worker	Assigned, eligible Agent and now at or before due; nonzero commitment and bounded URI. Creates Submitted.	`Delivered`; not deliverable or result required.
`validate(id, resultHash,approved, evidence)` / selected validator	Submitted and now at or before review end; same artifact, current stake/eligibility, one vote. Second yes pays; second no fails with slashing.	`Validated`, `Payout`, `Closed`; artifact, duplicate or eligibility failure.
`timeout(id)` / any caller	Assigned past due: slash and Failed. Submitted past review end: Refunded without slashing.	`Closed`; delivery/review still open.
`claimRefund(id)` / recorded client only	Terminal job, not previously claimed. Transfers accrued refund once.	`RefundClaimed`; refund unavailable.

All monetary fields are integer token base units and all durations are seconds. The reference order is as follows.

1. Each role holder calls Access `register(role,label)` for its owned name. The Access owner separately supplies a valid `setAdmission(account,evidence,until)`. An already admitted holder need not repeat a valid registration.
2. Agents approve the *JobMarket* as token spender and call its `deposit(uint256)`. Each validator approves the *Oracle*, calls its `deposit(uint256)`, and records `setJobAvailability(business,until, capacity)` covering the entire job horizon.
3. The eligible business approves the *JobMarket* for the bounty and calls `post(Spec,uint32,address[ 3])`. Goal and success metric each contain 1–512 bytes; bounty is at least 100 base units; duration is positive and at most 90 days; auction duration is positive and at most seven days; price weight is at most 10,000. The job bond is $\max(\lfloor B/10 \rfloor, m)$, with bounty B and minimum m. Three distinct current Oracle validators must differ from the business and token-paying caller.
4. Before auction end, each agent calls `bid(uint256,uint96,uint32)` with price from 100 through B, positive duration no greater than the specification, and enough free deposited stake to lock the bond. A committee member cannot bid on that job.
5. At or after auction end, any caller invokes `award(uint256)`. The winner is the lowest implemented score among still-eligible bids that fit the remaining execution horizon; this is a bounded auction, not

universal optimal agent selection. Ties use lower price, shorter duration and then lower numerical address. Every losing bond is unlocked. With no admissible winner, the full bounty becomes refundable.

6. The assigned worker performs the supported off-chain work, retains exact bytes and checker results, and calls `submit(uint256,bytes32,string)` by its due time. The contract stores the commitment and emits the URI. Each selected validator independently obtains those bytes, checks their commitment and acceptance evidence, then calls `validate(uint256,bytes32,bool,bytes32)`.
7. Two approvals trigger payment inside the second successful validation transaction. The harness independently verifies the terminal state and actual artifact acceptance before retaining a qualified capability. The client calls `claimRefund(uint256)` for unused bounty. A successor receives qualified artifacts and evidence, while obtaining its own authority, funding and task-specific acceptance.

For bid price p, promised duration d, bounty B, maximum duration D, price weight w, and reputation snapshot r, the exact integer score is

$$c = \left\lfloor \frac{pw10^6}{B} \right\rfloor, \quad t = \left\lfloor \frac{d(10000 - w)10^6}{D} \right\rfloor, \quad s = \left\lfloor \frac{(c + t)10000}{10000 + \min(r, 10000)} \right\rfloor.$$

Reputation here is the JobMarket's own bounded accepted-job counter, captured at bid time. It is not a general claim about an agent. Successful settlement increments it by one, capped at 10,000; a slashed failure decreases a positive value by one.

Quorum, arithmetic and recovery. The committee has three validators. Two approvals settle immediately; two rejections fail with slashing. Earlier approving or rejecting voters are rechecked for current Oracle eligibility when their decisive branch executes. Worker eligibility is checked at submission; the payout branch does not recheck worker or business eligibility. If p is the agreed price, each approving validator's gross payout is

$$v = \left\lfloor \frac{\lfloor 500p/10000 \rfloor}{2} \right\rfloor, \qquad g_{\text{worker}} = p - 2v.$$

Thus the nominal 95/5 split has rounding dust assigned to the worker. Every gross payout g burns $\lfloor g/100 \rfloor$ and transfers the remainder. With $p = 100$ tokens, the worker receives 94.05 tokens, each of the two approving validators receives 2.475, and one token is burned. Only the two approving validators are paid. The difference $B - p$ is a separate client refund, not a worker payout.

A missed worker due time permits anyone to call `timeout`: the bond is removed from agent stake, one percent of it is burned with integer rounding, and its remainder is added to the client's full-bounty refund. A two-rejection decision follows the same slash branch. The slash emits a `Payout` event for the client while crediting the refund; actual transfer occurs later in `claimRefund`. Missing reviewer quorum after submission instead returns the bounty and unlocks worker collateral without slashing: missing quorum is not proof of defective work. There is no general business cancellation call. Award, timeout and refund collection require calls; passing a deadline alone does not execute them. Terminal jobs release reviewer active-job capacity, but do not shorten the Oracle's previously established stake lock deadline.

5.3 Optional NovaSeed and MARK financing

A FusionPlan discloses a bounded set of committed job specifications and proofs against an immutable Merkle root. Its contract commitment does not itself publish the plan contents: collaborators must retain and disclose the exact job specifications, indexes, hashing convention and Merkle proofs. JobMarket `hashSpec(index,spec)` computes a double-keccak leaf from ABI-encoded index, hashed goal and

metric, bounty, duration and price weight. MARK verifies the actual plan index; the job itself stores `hashSpec(0,spec)`, so a nonzero plan leaf and the stored job hash must not be equated. MARK verifies the proof using OpenZeppelin's sorted-pair Merkle convention. A seed's NFT records the capsule, plan and evidence commitments, original creator, bounded job count and optional parent. It does not automatically transfer equity, intellectual property, revenue rights or proven capability inheritance.

Table 5: Optional campaign calls; MARK lots are internal campaign accounting.

Callable / actor	Prerequisite and state transition	Event / representative failure
`open(id,base,slope, maxLots,target, fundingEnd,duration)` / green seed owner	No prior campaign; bounded reachable target and horizon. Records business and ENS node; creates Funding.	`MarketOpened`; green owner, bounds or unreachable target.
`buy(id,count, maxCost)` / admitted buyer	Before funding end; green decision and unchanged seed owner; bounded lot count and quoted cost. Receives tokens; increases lots and reserve.	`Traded`; admission gate, lot bound or slippage.
`sell(id,count, minReturn)` / lot holder	Funding phase and sufficient lots. Returns reverse-curve value; no admission or green recheck and no funding-deadline check.	`Traded`; not redeemable or slippage.
`transferLots(id,to, count)` / admitted holder	Funding before deadline; green and unchanged owner; both parties admitted. Moves internal lots without payment.	`LotsTransferred`; funding/admission gate.
`bloom(id)` / recorded business	Funding before deadline, reserve reaches target, current green owner and still-owned recorded ENS node. Creates Bloomed.	`Bloomed`; not funded or business authority.
`route(id,index,spec, proof,auctionSeconds, validators)` / recorded business	Bloomed, same seed owner, green, node ownership, horizon, valid unused plan leaf and reserve. Exact bounty allowance to Jobs, call `postFromSeed`, then clear allowance.	`MissionRouted`; authority, plan proof, budget or job failure.
`collect(jobId)` / any caller	Routed terminal job not collected. Pulls job refund into reserve and reduces active count.	`MissionCollected`; not collectible.
`close(id)` / any caller	Funding at/after funding end; or Bloomed with no active jobs and all plan jobs posted or execution end strictly passed. Creates Closed.	`CampaignClosed`; funding/missions still open.
`reclaim(id)` / lot holder	Closed and positive holdings; consumes holder's lots and returns proportional unused reserve.	`CapitalReturned`; no capital claim.

The typed seed call is `seal(bytes32 capsuleHash,bytes32 planRoot,uint32 jobs,string metadataURI,uint256 parent,bytes32 evidenceHash)`. For a first seed, parent is zero; otherwise the caller must own the parent at sealing. The eligible owner then calls Oracle `openReview(uint256)`. Eligible staked validators, independent of both current seed owner and original creator, call `attest(uint256,uint16,bool,bytes32)`. A yes requires policy approval and reported risk no greater than the configured threshold. A decisive no quorum prevents further votes, but this is a risk-policy round, not job-output acceptance.

A green owner may next call MARK `open(uint256,uint96,uint96,uint32,uint96,uint64,uint32)`. Base is positive; slope may be zero; maximum supply is 1–1,000,000 lots; target is positive and no greater than the full curve quote; funding ends within 30 days; execution duration is greater than the three-day review window and at most 90 days. For supply s, count n, base b and slope a, the curve quote is

$$Q(s,n) = nb + \lfloor an(2s + n - 1)/2 \rfloor$$

for $n > 0$, with $Q(s, 0) = 0$. Buyers approve *MARK*, then `buy(uint256,uint32,uint256)` caps cost; `sell(uint256,uint32,uint256)` floors return using $Q(s - n, n)$. These are campaign lots, not a newly issued ERC-20, an unrestricted DEX or a guaranteed liquid secondary market.

When reserve reaches target, the recorded business calls `bloom`. The business can then `route(uint256,uint32,Spec,bytes32[],uint32,address[3])` only for a committed, unused plan index with sufficient reserve and time for auction, execution and review. MARK, not the business wallet, is the job client and allowance source. Its exact bounty approval is cleared after the call. The reserve cannot be freely withdrawn as an enterprise treasury.

Oracle green lasts only through its seven-day round and is sensitive to owner, admission and supporting-validator changes. A 30-day funding window or longer campaign does not extend that approval. A fresh round may be needed before later buys, bloom or routing; a new round cannot start until the prior round is strictly expired. Already posted jobs have their own independent review lifecycle. On closure, `reclaim` returns $\lfloor Rh/S \rfloor$ for remaining reserve R, holder lots h, and remaining supply S, then removes both the paid reserve and consumed lots. The final claimant receives the final residual. This is unused-capital recovery, not promised profit or revenue distribution.

Token utility and its cost. Agent deposits secure locked performance collateral; validator deposits support eligibility and timed review locks; funded bounty escrow supports agreed delivery and review payments; MARK contributions fund committed jobs. Deposits, ordinary transfers, curve purchases/sales, reserve recovery and bounty refunds are not all subject to the payout burn. `AscensionToken` checks exact transfer receipts and recipients, 18-decimal precision and actual supply destruction for its burn calls. A token holder is not thereby an admitted business, eligible worker, reviewer or payout recipient. Against a competent centralized system, this route adds gas, capital lockup, reviewer cost, latency and explicit recovery calls. Its potential benefit is inspectable commitments and rule-bound settlement among separately admitted participants; customer demand and commercial advantage remain to be demonstrated.

ABI and execution evidence boundary. We inspected the pinned Solidity declarations and a pinned historical compiled ABI archive, and supplied a source-compatible call ABI with an explicit provenance record. The historical wallet profile embeds 11 October addresses and must not be used to select this deployment. This study did not retrieve bytes authenticating the 18 October explorer ABI hashes or execute these calls on mainnet. The runnable companion is a labeled local transition model with its own checker, not a claim of Solidity conformance, a fork reproduction or a transaction-ready client. A future authorized operator must authenticate the complete deployed ABI, live graph and participant state before signing.

6 Implementation: one harness, explicit route boundaries

The reference extends AGI ALPHA's maintained operating path rather than introducing another enterprise service. The inspected `successor-harness` facade is PR 4863 at `c455ad4ce109c86df2fef09d19674810290b5185`. On 22 October 2026 the refreshed PR record was *open, not merged*, with that exact head. It therefore supplies candidate source, not released or deployed functionality.[1] The retrieved workflow-runs and combined-status responses contained no entries; these responses do not establish hosted qualification or publication. The PR's report of 34 passing harness/workbench tests remains an external author report in this paper; we inspected the tests but did not rerun the complete Python facade environment.

[1] https://github.com/MontrealAI/AGI-Alpha-Agent-v0/pull/4863

6.1 What the candidate actually retains and executes

`enterprise.py` accepts exactly five top-level fields: `schema`, `enterprise_id`, `name`, `work`, and `parent`. The schema is `successor-enterprise/v1`; an enterprise ID is a canonical UUID. The work request is validated through the existing workbench before registration. The immutable configuration and work record are retained in the existing SQLite journal, whose entries form a hash chain signed by its local Ed25519 identity. This supplies source attribution and tamper detection relative to retained keys/checkpoints; it is not independent organizational review or proof that an operator never rewrote its own history.

Actual entry point	Result	Boundary preserved
`new`, `validate`	Editable, inert configuration for a maintained family.	No journal or permission is created by template preparation.
`activate`	Configuration and planned work run registered; repeated identical activation is idempotent.	An ID cannot bind changed configuration; an exact accepted parent is checked again in the registration transaction.
`operate`	Fixed worker runs; completed output awaits result-bound local review.	Completed terminal results are retained; retries do not silently erase prior attempts.
`review`, `export`	Exact result digest accepted/rejected; artifacts and signed local receipt exported.	Review records the local operator and `independent=False`. A recipient needs separately trusted source keys.
`derive`	New UUID and editable work template with exact parent configuration/result digests.	Child begins planned, without its parent's result, acceptance, authority or settlement.
`handoff`	Existing Studio request bound to enterprise work and a verified plan index.	Validation is not execution, profile installation, a grant, or funding.

Table 6: Pinned candidate behavior, established by source inspection. Full paths and blob identities are retained in the companion evidence.

The five maintained families are reliability, performance, documentation, public-data and numerical. `workbench.invoke` invokes the packaged Node worker with bounded JSON, a scrubbed environment and cancellation checks. It cannot select a caller-supplied executable. The local path has three attempts, a declared 60 000 ms per-attempt budget, and a 65-second outer watchdog; these are distinct implementation bounds. Compute price and human-review cost are explicitly unknown. The local states include planned, running, review, accepted, rejected, interrupted, paused, stopped and retired. Resume is explicit for interrupted/paused pure work; stop and retirement are terminal.

We executed the exact pinned worker directly on its included numerical and public-data examples, then ran its inspection operation. The numerical output reports the exact integral of x^2 over $[0, 3]$ as 9, Simpson's result as 9, and the retained trapezoid error as $1/32$, which fails the declared 10^{-6} tolerance. The public-data report retains five rows and group totals 60, 155 and 80, summing to 295. These are local fixed-worker executions, not runs of the full enterprise facade, independently commissioned services or on-chain settlements. Their raw requests, results and inspection records are supplied separately. They establish a working bounded baseline while the new construction is evaluated.

The companion also exercises a concrete integration: `harness_adapter.py` verifies the new instrument's retained exposure result and converts all 80 window/value pairs into the existing public-data CSV interface. Each row retains its unique window index and bounds; units explicitly warn that summing overlapping windows counts exposure repeatedly. The adapter invokes the pinned worker's actual `plan`, `run` and `inspect` operations. A separate Python check compares every group and the delivered CSV/JSON artifacts against the interval result. It also emits a valid-shaped `successor-`

`enterprise/v1` configuration for later activation. The worker formats this exact report; the new instrument executes in the Python companion. Neither the full enterprise facade nor a native signed journal was activated by this integration.

6.2 Studio and the three economic routes

The actual handoff schema is `agialpha.ascension.studio-work.v1`, containing exactly `schema`, `project_id`, `cycle_id`, `plan`, `plan_index` and `work_spec`. Plan verification and a valid index bind the selected work; they do not establish remote capacity. Existing Studio functions `prepare_work`, `seal_work`, `bind_work`, `execute_work`, `accept_work` and `settle_work` own their respective boundaries. Execution checks the installed environment and exact controller-signed authorization. The read-only Ascension adapter binds canonical posting and assignment observations, commits delivered bytes, and reconciles contract facts. Local acceptance and the market's validator decision are separate records.

Route / source	Inspected interface	What remains separate
Alpha / Ascension	Studio request and `MarketReader`; `bind_assignment`, `bind_delivery`, `import_settlement`.	The designated five-contract graph, AGIALPHA in 18 decimals, installed trust and explicit grants.
Fleet Linux	`SuccessorController.prepare`, `execute`, `result`, `stop`; installed Controller → Studio → Companion → ComputeWorker.	Current runtime health, exact task, session-bound confirmation and worker/reviewer control; an internal checker is not independent control.
Jobs v0	`importAlphaProject`, `runProjectTransfer`, `inspectProjectTransfer`, `verifyProjectReceipt`.	A distinct AGIALPHA *JobRegistry* route; unfunded drafts and receipt checks do not grant authority or reconcile payouts.
Jobs Manager	`prepareImprovement`; native employer-model/console interfaces and `compileComputerWork`.	A separate six-decimal USDC draft, current admission, runtime context and authorized console action.

Table 7: Existing source interfaces, not evidence of an installed four-repository commercial journey. Inspected Fleet, Jobs v0 and Manager sources are pinned in the evidence inventory.

The native Jobs v0 project receiver supports public/synthetic integer CSV totals. Its envelope `agijobsv0.successor-public-data-work.v1` explicitly differs from the Python Alpha market adapter's work-spec namespace. It preserves original project bytes, failed/rejected history and a separately chosen AGIALPHA allowance. Manager's draft uses its native USDC compiler and a fresh six-decimal allowance; it does not relabel an AGIALPHA request. Fleet's project interface compiles supported work into the installed execution boundary rather than accepting arbitrary generated programs.

These differences define narrow engineering work: register and qualify the new instrument in the existing fixed-work boundary, bind its exact artifacts to the chosen request namespace, and rehearse the corresponding receipt path against one pinned contract system. They do not require a new token, bridge, control plane or enterprise database. A schema-valid handoff, adapter catalog entry or signed local receipt must never be reported as remote execution, current admission, funded settlement, or measured recursive improvement.

The candidate's own `show` response makes the present boundary machine-readable: `authority=[]`, `settlement=NOT_CONNECTED`, and `independent_qualification=False`. A broader validator-gated constellation is an engineering direction; this source slice neither deploys one nor establishes general intelligence.

7 A retained instrument across two distinct offerings

The supplied companion is a new, executed local experiment, not a reproduction of the predecessor's missing companion. Its inputs are public, authored synthetic fixtures: at most 64 nonnegative integer-weighted half-open intervals in the integer coordinate domain $[0, 128)$. The first offering delivers 80 exact exposure integrals to an energy analyst from 24 intervals. The second delivers the maximal constant-load violation intervals for 16 capacity thresholds to an infrastructure operator from a different 32-interval input. The beneficiaries, questions and deliverables differ; neither is a paying business or a qualified physical engineering measurement.

The retained instrument is fixed shipped code: a conventional event sweep constructs disjoint constant-load segments, and a prefix integral answers exposure queries. Each new dataset is processed anew. No result cache, generated executable or old acceptance decision is transplanted into the successor. The retained record binds the exact `instrument.py` SHA-256, version `interval-sweep-prefix/v2`, supported domain and qualification-record hash. The retention checker validates the same source binding for all successors. Its parent also binds the accepted configuration and result. A third accepted generation verifies cumulative accounting: $2698 + 1279 + 677 = 4654$ producer steps. Neither authority nor money is inherited.

For a coordinate between adjacent endpoints, the sweep's running sum equals the sum of the active input weights; endpoint deltas add starts and subtract ends. Integrating that constant value over each intersection yields the exposure. Selecting the constant segments above a threshold yields the violation certificate. This is familiar exact arithmetic on the declared finite input convention. It establishes no physical validity for the supplied intensities or threshold. A separate checker reconstructs every integer-bin load directly from the original intervals, without importing the sweep, and checks each requested output.

7.1 Matched outcomes and an explicit conventional tie

Independent setup and ordinary harness reuse deliberately execute the same competent raw procedure. Reusing the wrapper alone does not remove its repeated scans. The constructed instrument and the frozen expert use the same conventional sweep/prefix recipe: their computational tie is intentional. It credits established engineering and prevents an apparent advantage from a weak expert baseline. This comparator is an algorithmic control, not a contest between independently authored general improvers.

Method	First use producer	Successor producer	Two-offering producer	Semantic checker
Independent raw setup	1944	65568	67512	12076
Ordinary wrapper reuse	1944	65568	67512	12076
Sweep with linear query	4835	1279	6114	28612
Retained sweep/prefix	2698	1279	3977	28612
Strong conventional expert	2698	1279	3977	28612

Table 8: Executed synthetic workload. Producer and checker columns are different declared operation models and must not be added or converted to money. First use charges the method's own qualification producer work; the checker total includes qualification and both fresh task checks. Every method delivers both accepted offerings.

The producer meter charges one abstract step for each instrumented domain check, interval-to-event emission, merge-sort comparison, event consumption, segment emission, overlap/membership/threshold predicate, prefix entry or binary-search comparison. It excludes Python overhead, allocation and individual arithmetic instructions. The raw per-bin overload procedure is a transparent baseline,

not the best available algorithm. The expert column shows that a standard sweep already supplies its replacement. We therefore claim removal of an inherited production burden, not an algorithmic breakthrough.

The retained method's first-use cost vector is $(2698, 22468)$: 1247 task-production steps plus 1451 qualification-production steps, and 5932 current-task checker operations plus 16536 qualification-checker operations on 24 fixtures. Raw first use costs $(1944, 5932)$ and wins. The distinct successor costs $(1279, 6144)$ versus raw $(65568, 6144)$. Across both offerings the retained vector is $(3977, 28612)$, versus $(67512, 12076)$ for raw operation. Verification remains substantial and is never waived because a parent passed. The 24 qualification fixtures are bounded evidence, not exhaustive qualification of every possible future input.

The trace also records the cost of evaluating alternatives. A proposed replacement of prefix queries with linear segment scans costs 3387 producer steps for the exposure workload, versus 1247 for the retained version. The evaluation itself costs 4634 producer steps and 11864 independent checker operations, separately from the selected operating path; it is rejected. The sweep-linear method's own initial qualification costs 1448 producer steps and 16536 checker operations. Development, model generation and author/reviewer labor were not timed and are not claimed free. This finite human-authored search is a small versioned improvement loop, not demonstrated general recursive self-improvement.

7.2 Failed transfer, incomplete work and accountable recovery

A signed-load request is rejected because negative intensities lie outside the retained nonnegative domain. The request and rejection remain in the trace. A separate capacity attempt exhausts its declared seven-step budget and returns `incomplete-budget`, with seven consumed steps, no accepted result and no completed negative finding. The previously accepted parent remains available. The successful successor and third generation are retained alongside these unsuccessful attempts; later success does not erase them.

The local journal is hash-linked, not signed by a trusted party. Its checker requires the complete fixed-study matrix, qualification fixtures, three accepted generations, the negative and incomplete controls, the regression decision, the expert tie and all simulation branches. Empty records and valid but truncated prefixes fail completion checking. Hashes provide local integrity and exact references; they do not establish independent authorship, authority or a trusted timestamp. Native signed-journal integration remains a distinct harness boundary.

The companion additionally exports the 80 exposure outputs to the pinned harness's real `public-data` worker interface. Its adapter calls `plan`, `run` and `inspect`; an independent Python comparison checks the preserved window labels, values, grouped evidence and generated artifacts. The pinned worker formats and checks that export. The new interval instrument still executes in the local companion; the complete enterprise facade, native journal, remote Fleet executor and on-chain settlement are not exercised by this adapter.

7.3 Contract transitions are explicitly simulated

The direct-job double uses exact source-backed call mappings and a dated 10-token stake snapshot. It runs six branches: successful two-approval payment, two-vote rejection, missed execution deadline, missing review quorum, no qualifying bid and a previously approving validator becoming ineligible before a decisive second vote. The last decisive vote reverts without being retained; the still-submitted job later follows review-expiry recovery. Validator active-job reservations release while their withdrawal lock deadline remains. Refunds are credited for a later client-only pull claim.

For the paid fixture, a simulated 100-token bounty and 90-token winning price produce a 10-token

client refund. The worker receives a gross 85.5 tokens, and each of two approving validators receives a gross 2.25 tokens. Their respective one-percent burns are 0.855 and 0.0225 tokens, for 0.9 tokens burned in total. These are integer base-unit calculations in the local model, not actual transfers, observed yields or estimated gas. The double is a selected normal-path and recovery model, not an EVM execution or full Solidity conformance test. It neither simulates MARK nor verifies live ENS ownership, admissions, allowances or external token behavior.

Raw records retain producer/checker wall times from one local run. They are not statistically qualified throughput measurements. Each comparison method delivers two accepted synthetic outputs with the same requests and checks; the appendix reports the descriptive completion rate and its excluded costs. Runtime model tokens and submitted transactions are zero; transaction expense is not incurred, gas is not estimated, and collateral lockup is simulated. Customer demand, pricing, human operating burden and installed-release acceptance remain unobserved.

8 Let improvement accelerate an operating baseline

Observe one recurring obstacle; propose a versioned change; evaluate it against the retained procedure; adopt or reject; then test the next cycle. The target can be how the harness discovers unnecessary work or constructs reusable means. Merely increasing model calls, creating another configuration, or repeating a finite authored search does not demonstrate recursive self-improvement.

Keep the deployable baseline available throughout. A promoted artifact must bind its version, supported domain, tests, cost and rollback target. It cannot silently expand admission, wallet authority, permitted actions, spending limits or contract control. It cannot weaken acceptance or rewrite the behavior of a running enterprise without an explicit transition. Existing versioning, relevant regression checks and rollback suffice for the reference path; no new administrative platform is required.

The new companion compares retained construction with a frozen conventional improver and includes construction/evaluation costs. Where the expert already selects the sweep instrument, the proposed treatment ties it. A shift with too little reuse can make construction wasteful. A transfer outside the declared domain is rejected. Those results limit the mechanism usefully: retained capability is an option for qualified work, not an obligation to use the latest artifact.

In one consistent cost unit, an improvement with upfront cost H, incremental maintenance m_j and per-delivery saving s_j pays for itself only if $\sum_j (s_j - m_j) > H$. When the terms have different units, use a cost vector rather than add them. Unknown demand, review work and pricing leave commercial break-even unknown. This elementary accounting is a decision aid, not a compounding law.

9 Preserved result lineage and its corrected interpretation

The immediate predecessor is *Autonomous Enterprises: Activate. Evolve. Scale.*, preserved at repository revision 3717c9ffb677751e1733a9ac7e010f5c877ef5d1 [4]. We retrieved both its 15-page PDF and its full TeX on 22 October 2026. The PDF SHA-256 is 9cc01c8143400dd2f4258e11b3a86295d346be0671afc19cb99249b4a47b3cd2; the TeX SHA-256 is 407e27956e09b8a9af06171ac6119a119e754e76057e52957273dd4238bd31c7. The predecessor in turn identifies *Build the Means of Building*; this revision does not rename or overwrite that historical paper or either published URL. The original runnable companion was not recovered in this focused source search. Thus its execution results remain historical reports, not newly reproduced measurements. The new companion supplied here is an explicitly identified reconstruction and extension, with its own result identities.

The predecessor's displayed "polynomial composition" task returns the coefficient vector of a product. Its operation is *polynomial multiplication*, equivalently coefficient convolution: for $p(x) = \sum_i a_i x^i$ and $q(x) = \sum_j b_j x^j$, the coefficient $c_k = \sum_{i+j=k} a_i b_j$. This differs from substituting $q(x)$ into $p(x)$. The

historical task identifier and recorded counts are preserved; explanatory terminology is corrected.

Table 9: Historical candidate-check counts from the predecessor, not independently rerun. Each continuation uses its same candidates and finite acceptance suite. Lower is better.

Continuation	Fixed	Learned	Move-to-front	Expert
Polynomial multiplication	223	118	172	**90**
Dependency reachability	233	**89**	233	93
Both continuations	456	207	405	**183**

The learned continuation count 207 already exceeds the expert alternative's 183 by 24 checks. Charging the learned method's additional 90 source-qualification checks yields 297, or 114 more than that expert continuation count; profile construction and human work remain additional. Its 159-check saving is the comparison $456 - (207 + 90)$ against *fixed order*. It is not a saving against every baseline. The shared 230-check source campaign gives fixed and learned totals 686 and 527 respectively, with the same restricted interpretation. Use the simpler expert alternative where it wins. A later learned update reported 90 rather than 89 reachability checks and was rejected. Neither that failed update nor the expert's win disappears behind the aggregate fixed-order saving.

9.1 A conditional rejection-only theorem

Let $T = (t_1, \ldots, t_n)$ be fixed, deterministic, side-effect-free predicates on a fixed candidate c, fixed inputs and fixed predicate implementations. Outcomes do not depend on order. Acceptance is $A(c) = \bigwedge_{i=1}^{n} t_i(c)$. Let P be k distinct members of T; run P first, then all remaining predicates in their original order, stopping at the first failure. Each t_i has a fixed nonnegative evaluation cost $w_i(c)$ independent of order.

Theorem 9.1 (Rejection-only inheritance, under fixed predicates)**.** *The reordered schedule preserves $A(c)$. If $C_T(c)$ and $C_P(c)$ denote costs under the original and reordered schedules, then*

$$C_P(c) \leq C_T(c) + \sum_{t_i \in P} w_i(c).$$

A passing candidate incurs the same total predicate cost. For unit costs and $0 \leq k < n$, the additional-check bound k is attainable.

Proof. Every predicate appears exactly once, preserving the conjunction. A failure in P costs at most its sum. Otherwise the residual schedule retains original order: the paid predicates lie in the union of the original paid prefix and P. This proves the bound. For sharpness, let only t_1 fail and choose k later passing predicates as P. The respective costs are 1 and $k + 1$. □

This elementary guarantee concerns a fixed finite predicate, not the semantic adequacy of the tests or truth of a business claim. Candidate mutation, stateful tests, order-dependent resources, changing evaluators and uncontrolled stochastic outcomes break its assumptions. Budget exhaustion is `INCOMPLETE`, never acceptance; no acceptance can be inherited from a partial prefix. Log processing, scheduling and qualification overhead are outside the displayed predicate-cost bound. The scheduling result supports the harness's retention discipline; it is not the enterprise thesis or a claim of general recursive self-improvement.

10 What improvement research contributes

The foundation separates the beneficiary's authorized purpose from the chosen formulation and the instrument that judges it [6]. Its useful discipline is to construct a different means, bridge its outputs to that purpose, and test the claimed removal of a restriction. The accompanying economic formulation separately tracks technical capability, voluntary arrangements, financing, rights and evidence [5]. A positive modeled surplus does not establish liquidity, customer consent or realized exchange. AGI ALPHA supplies the wider organization architecture and evidence discipline [3]; this paper instantiates a bounded path inside that architecture.

Conventional software engineering already supplies reusable libraries, interfaces, regression tests, artifact versioning and recovery. Test prioritization has long used earlier executions to detect subsequent faults sooner [15]. Those are enabling methods rather than new discoveries here. Anthropic's long-running-agent report describes initialization, retained progress and incremental tested work [1]. We adopt the engineering lesson that continued operation depends on inspectable state and recovery, not a sufficiently inspiring instruction.

Darwin Gödel Machine empirically evaluates code modifications and retains an archive of candidate agents [16]. Hyperagents makes the meta-level modification program editable as well as the task agent [17]. These are relevant precedents for testing whether the means of improvement improve. They are not evidence that this harness, its fixed work families or the present human-authored companion achieve the same capability. The actionable loop here is smaller: observe recurring work; propose one versioned means; evaluate the next complete campaign against both a frozen incumbent and a strong conventional improver; adopt, reject or retain a tie; then test another cycle. Construction, failed experiments and qualification belong in its cost ledger. A deployable baseline continues to operate while candidates remain under evaluation.

Recursive reports that its Eureka Machine contributed across Flint's model-building stack [11]. Its earlier research account explicitly includes correctness audits against reward hacking [10]. These are developer reports, not benchmarks reproduced in this work. They motivate testing the evaluator and retaining experimental context; they do not justify a superiority claim against machinery we cannot inspect or run. The released model and inference engine are not the proprietary research system that produced them.

Available artifacts and commercial use. The Flint model card identifies the released checkpoint as Apache-2.0 and distinguishes hosted API terms [14]. The inspected `rec` source at revision 8af99abaf50b33903147d89eed91d1bb3ac642dd includes an Apache-2.0 license [13]. That license permits commercial use and redistribution subject to its terms, including applicable license, attribution, modification and NOTICE obligations; it grants no general trademark permission or warranty [2]. Recursive's 21 October terms, section 1.3, explicitly give an open-source release's license precedence for those materials while retaining separate terms for websites and hosted services [12]. Third-party components retain their own licenses. These observations establish an available licensing route, not operational qualification of a selected checkpoint/runtime pair. No Flint weights were downloaded or executed here. The engine README specifies supported Apple Silicon and at least 48 GB unified memory; it emphasizes single-session use. An implementation team can evaluate a suitable model as a proposal generator behind existing execution and acceptance boundaries without making that evaluation a launch prerequisite.

11 Finance bounded work; keep commerce understandable

A beneficiary buys a defined useful outcome with an agreed check. The reference offerings deliver an exact public-input report and its inspectable evidence. Prices and demand remain unobserved. No invented revenue, customer, token appreciation, passive income or guaranteed yield is needed to

explain the mechanism.

The Ascension suite uses $AGIALPHA for specific obligations: worker and validator stake, bounty escrow, supported MARK contributions and source-defined settlement. A holder is not thereby an admitted business, an eligible agent, a validator or a payout recipient. The separate AGIJobManager route uses USDC. Moving between systems requires an explicit supported handoff and fresh funding authority; their balances and settlement semantics are not interchangeable.

An ordinary centralized implementation can run exactly the same instrument with a database, authenticated users, a service agreement and off-chain payment. It may be preferable when one trusted operator and buyer already suffice. The on-chain route is justified only when its inspectable commitments and coordination are useful enough to pay for their burden.

Table 10: Compare against a competent off-chain service, with the same deliverable and acceptance. Neither column is universally better.

Decision term	Centralized/off-chain service	Ascension-funded route
Authority	Authenticated account and agreement	Current admission, role/ENS checks and exact caller rules
Funding	Payment or service-credit arrangements	Token balance, allowance and contract escrow/phase conditions
Review	Tests and accountable reviewers	Those checks plus eligible independent validators and quorum
Expense	Hosting, payment, review and support	Those burdens plus gas, lockup and chain/recovery effort
Latency	Application and review schedule	Those schedules plus transaction confirmation and contract windows
Inspectability	Published records when provided	Public transitions and commitments; semantics still need evidence
Recovery	Operator and contractual remedy	Only implemented rejection, expiry and refund branches; no global pause

Keep unfunded planning and useful local computation wallet-free. Use funded commitments when they are needed. MARK is optional financing, not the entrance to every useful enterprise. Its reserve is constrained by the committed plan, budget and timing, and unused-reserve recovery is not a promised profit distribution. A green risk decision is distinct from later job acceptance and establishes neither investment suitability nor successful delivery.

12 Scale useful variety within the real bottlenecks

Retained instruments can support multiple beneficiaries concurrently when resources permit. Copying configuration does not copy funds, available reviewers, legal rights, qualified hardware or demand. A simple stationary upper bound illustrates the limit:

$$\lambda_{\text{accepted}} \leq \min\{p/e,\ q/v,\ H/h,\ D\}, \tag{2}$$

where execution capacity is p/e, validation capacity q/v, available human review capacity H/h, and arriving demand D in the same time unit. Retries, heterogeneous jobs, correlated checker faults, unavailable validators and settlement lockup can tighten this bound. No throughput law follows merely from the number of agents.

Selectively expand offerings whose accepted benefit exceeds their complete burden. Track execution, validation, operator attention, failed attempts, recovery, transaction expense and locked capital

separately. The reference companion reports local results; it does not measure production-market throughput. A useful next workload should include independently selected public tasks and naturally occurring failures, rather than only authored fixtures.

The long horizon reaches science, infrastructure and useful energy through intermediate constructions: an accepted measurement instrument, a reproducible discovery, a qualified production method, a permitted installation and demonstrated operation. Each step needs evidence, capacity, materials and valid rights. The present mechanisms establish none of those future physical achievements. Their purpose is to make the next warranted step easier to construct and test.

13 What is ready to do next

The smallest useful next step is an independently selected public-input report run through a qualified baseline, with its input meaning, deliverable, resource allowance and acceptance agreed in advance. The retained instrument is then tested on a distinct second request. This local step needs no mainnet activation and can proceed while deployment readiness work continues.

For harness integration, use the pinned candidate's existing enterprise configuration, retained signed records, review and control/recovery surfaces. Add only the narrow instrument-to-supported-work-family adapter and its task-specific checker after the candidate's normal qualification and integration. The companion is not an installed four-repository compatibility set. The runbook and source map identify which boundaries were actually replayed, which are simulated and which remain proposals.

For on-chain operation, first satisfy the recorded readiness gaps and inspect current chain state: exact code and bindings, admission/expiry and ENS ownership, balances and spender allowances, stake and availability, deadlines and campaign/job conditions. A static paper cannot establish these. No transaction was signed or submitted for this assignment. Owner-only changes stay separate from ordinary business, worker, reviewer and buyer calls. The absence of a pause makes conservative readiness and implemented recovery paths more important, not less.

The strongest scientific limitation is that the new instrument and workload are human-authored and bounded. A strong conventional engineer attains the same construction. The strongest implementation limitation is the absence of executed funded four-repository settlement and authenticated live participant state. The decisive future test is whether naturally occurring distinct offerings benefit after charging interpretation, construction, qualification, review and recovery. A negative result should retire the transfer rather than preserve a claim of progress.

ACTIVATE a growing diversity of useful enterprises.
EVOLVE the conditions that make their successors possible.
SCALE what proves useful.
FIND YOUR ALPHA. BUILD YOUR SUCCESSOR Ω.

Research status. AI-assisted source review, authoring and programming. Local deterministic computation and explicitly labeled simulation; no model-weight improvement, autonomous deployment, incorporation, customer purchase or mainnet state change. Separate code paths and independent agent review do not constitute independent institutional reproduction.

A Replay the new companion and inspect the evidence

The supplied companion is a new local reconstruction aligned to the inspected harness candidate. It requires Python 3.11 or later; the fixed-worker replay additionally requires Node. It makes no network calls, installs no model and holds no wallet key. From the extracted package root:

```
python companion/run.py --out rerun
python companion/checker.py rerun/trace.jsonl
python -m unittest discover -s companion -p 'test_*.py' -v
python evidence/check_call_abi.py evidence/contract-source
python evidence/harness_sources/check_pinned_worker.py
```

Use a new output directory for a new run and preserve the supplied records. Counts and mathematical outputs are deterministic under the declared fixtures and operation model; wall times vary and therefore change trace hashes. The independent semantic checker rebuilds the bounded load function directly from raw intervals. It does not import the instrument's sweep implementation. It verifies output values, trace linkage, accepted-parent references and cumulative producer counts. Its independence is computational, not independent institutional authorship or a trusted external timestamp.

The source-compatible ABI checker cross-checks 44 explicit source function signatures and exact source-file digests. It does not authenticate October 18 explorer ABI bytes or execute an EVM. The pinned worker replay's checker independently checks its exact numerical and public-data artifact values. The new adapter's README and raw requests document its narrower interface conversion; executing a public-data aggregation of a supplied report is not executing the new interval instrument inside the original worker.

The new companion's trace and summary retain the synthetic inputs, exact outputs, resource/counter definitions, qualification fixtures, failed transfers, incomplete attempts, recovery, three-generation producer accounting and labeled contract simulations. Failed work remains in the same retained history. The source archive separately retains the predecessor PDF/TeX and their historical result identity; the new trace is not a substitute for that missing original companion.

A.1 Descriptive local completion rates

The retained run also records per-delivery production and independent-check wall times. Table 11 divides the two accepted outputs per method by their summed production-plus-check time. This excludes instrument qualification, trace persistence, study overhead, authoring, human review, installation and settlement latency. It is a descriptive single-pass measurement on Python 3.12.14, Linux; it supplies no confidence interval, operational capacity forecast or supported speed ranking between the retained and expert methods. Their counted work ties even though their observed elapsed times differ.

Method	Accepted	Elapsed milliseconds	Accepted/second
independent	2	17.015	117.5
ordinary	2	14.347	139.4
sweep-linear	2	2.327	859.4
constructed	2	1.370	1460.3
expert	2	1.717	1164.8

Table 11: One recorded local pass over the same two synthetic requests. These times are separate from the qualification and producer/checker operation vectors. Raw nanoseconds and the rate formula are supplied in the evidence.

A.2 Rebuild the paper

The figure sources are plain TikZ in `figures/`. Bibliography sources are in `paper/references.bib` and `paper/research.bib`. With a standard TeX Live installation including Palatino, TikZ, microtype and booktabs:

```
cd paper
pdflatex -no-shell-escape -interaction=nonstopmode -halt-on-error
    Autonomous_Enterprises_Activate_Evolve_Scale.tex
bibtex Autonomous_Enterprises_Activate_Evolve_Scale
pdflatex -no-shell-escape -interaction=nonstopmode -halt-on-error
    Autonomous_Enterprises_Activate_Evolve_Scale.tex
pdflatex -no-shell-escape -interaction=nonstopmode -halt-on-error
    Autonomous_Enterprises_Activate_Evolve_Scale.tex
```

The final PDF was compiled and its rendered pages visually inspected. `SHA256SUMS.txt` identifies the delivered files; it is an integrity inventory, not a code-signing or commercial acceptance certificate.

B Claim classes and custody

Class	Claim and evidence	Practical boundary
Proved	Fixed-predicate rejection-only conjunction and conditional screening-cost bound	Pure, deterministic, fixed candidate and predicate assumptions only
Executed	New instrument, two output types, fresh independent checks, negative transfers and retained lineage; raw companion trace	Authored synthetic domain; abstract operation meter and single-run timings
Executed	Pinned numerical/public-data worker replay and narrow public-data adapter conversion	No full enterprise facade, installed Fleet journey or funded settlement
Simulated	Local direct-job state model: assignment, votes, payouts, expiry and pull-refund outcomes	Python test double; no EVM, fork, real ENS, tokens, gas or MARK execution
Source verified	Designated addresses match inventory; exact Solidity declarations and 44 explicit ABI signatures compared	Static source plus dated manifest, not current balances/admissions or authenticated explorer ABI bytes
Externally reported	October 18 deployment observations and explorer exact matches; historical companion counts; PR test report	No new chain reproduction or independent recovery of historical companion results
Proposed	Instrument qualification in fixed-worker boundary; installed, independently controlled four-repository journey	Requires a narrow adapter, applicable checks and installed acceptance

Class	Claim and evidence	Practical boundary
Unestablished	Paying customers, market size, prices, profit, token appreciation, broad RSI, general intelligence and civilizational outcomes	No measured evidence in this assignment

Evidence custody is explicit. Contract source revision is c0fc5217e364a39a4cd123d17a8317c06b80f820. Harness candidate is c455ad4ce109c86df2fef09d19674810290b5185. Historical paper revision is 3717c9ffb677751e1733a9ac7e010f5c877ef5d1. Retrieval date is 22 October 2026. Exact Git blob identities, SHA-256 values, scope of reading and retained original files are listed in the evidence reports and inventories. A mutable public URL remains useful for readers; the recorded revision identifies the bytes actually inspected.

B.1 Activation, funding, recovery and rollback

The separate `runbook/OPERATIONS.md` gives the short native CLI path and separates ordinary callers from owner/administrator actions. Activate local supported work first; operate and inspect its artifacts; make an explicit result-bound review; derive a new mission from an accepted parent; check that mission afresh. Paused or interrupted pure work resumes only within the existing attempt budget. Stopped/retired work remains terminal. Rollback selects a previously reviewed version for a new run without rewriting the old history.

Funding is a separate step. Choose the exact Ascension, Jobs v0 or Manager route, identify the current caller and allowance spender, inspect live prerequisites and use only that system's supported transitions. A local stop cannot cancel a chain transaction. A stalled direct Ascension job uses its actual award, timeout and pull-refund calls. An optional MARK campaign follows its own sell, collect, close and reclaim phase conditions. The writing assignment authorizes none of those live transactions.

C Revision note and falsifiable next test

This revision moves the center of the paper from test ordering to reusable productive construction. It preserves and corrects historical results, mechanizes the Ascension roles, maps the designated deployed graph and adds a replayable instrument experiment with an explicit conventional tie. The contribution is the outcome-preserving operating construction and its inspectable boundaries; it is not the invention of sweep lines, a new token platform or a demonstrated autonomous economy.

The smallest remaining technical deployment step is to qualify the instrument at the existing fixed-work adapter boundary and exercise it from an exact-source installed package against a disposable, separately authorized contract graph. Include a distinct successor and a failed transfer, independently controlled execution/review, canonical receipts and output checks. The immediate commercial test is simpler: one actual beneficiary accepts or rejects a bounded public-input delivery. If the full construction, interpretation, qualification, review and recovery burden fails to improve the successor's useful delivery, reject that transfer claim while continuing to use the working baseline.

References

[1] Anthropic. Effective harnesses for long-running agents, November 2025. 26 November 2025 first-party engineering report; inspected 22 October 2026. https://www.anthropic.com/research/effective-harnesses-for-long-running-agents.

[2] Apache Software Foundation. Apache license, version 2.0, January 2004. Authoritative English

license, sections 2–9 inspected 22 October 2026. https://www.apache.org/licenses/LICENSE-2.0.

[3] Vincent Boucher. AGI ALPHA: A scalable substrate for intelligence organizations, April 2026. 30 April 2026 architectural manuscript; opening, abstract and reader map inspected 22 October 2026; not independently validated. https://github.com/MontrealAI/agialpha-first-real-loop/blob/bd920a6c52d820a087116bf59f2a4236d0494ac0/docs/manuscript/AGI_ALPHA_Unified_Publication_Final.md.

[4] Vincent Boucher. Autonomous enterprises: Activate. evolve. scale. https://github.com/MontrealAI/AGI-Alpha-Agent-v0/blob/3717c9ffb677751e1733a9ac7e010f5c877ef5d1/docs/Autonomous_Enterprises_Activate_Evolve_Scale.pdf, 2026. Pinned predecessor revision 3717c9ffb677751e1733a9ac7e010f5c877ef5d1, inspected 22 October 2026; historical companion measurements are reported rather than independently reproduced.

[5] Vincent Boucher. Invent the economy: Transformational intelligence for creating viable markets. MONTREAL.AI research manuscript, October 2026. Selected source sections inspected 22 October 2026; TeX blob 36d366b57c777fc85c5cd61e9c705b2a63e76e04. https://github.com/MontrealAI/AGI-Alpha-Agent-v0/blob/main/docs/Invent_the_Economy.pdf.

[6] Vincent Boucher. Transformational intelligence: Learning to change the game. MONTREAL.AI research manuscript, October 2026. Selected source sections inspected 22 October 2026; TeX blob 83f834fbbabfc0680cf560e5e96852cc8f4fa9e7. https://github.com/MontrealAI/AGI-Alpha-Agent-v0/blob/main/docs/Transformational_Intelligence.pdf.

[7] MONTREAL.AI. Ascension five-contract source. https://github.com/MontrealAI/AGI-Alpha-Agent-v0/tree/c0fc5217e364a39a4cd123d17a8317c06b80f820/contracts/ascension, 2026. Exact deployed-source revision c0fc5217e364a39a4cd123d17a8317c06b80f820; retrieved 22 October 2026.

[8] MONTREAL.AI. Current mainnet deployment: 18 october 2026. https://github.com/MontrealAI/AGI-Alpha-Agent-v0/blob/main/docs/agent/CURRENT_MAINNET_DEPLOYMENT.md, 2026. Retrieved 22 October 2026. Activation and public funding not approved; no global pause.

[9] MONTREAL.AI. Ethereum mainnet public deployment manifest, 18 october 2026. https://github.com/MontrealAI/AGI-Alpha-Agent-v0/blob/main/docs/deployments/2026-10-18/manifest.json, 2026. Retrieved 22 October 2026; point-in-time inventory and reported authority evidence at block 26220072, not a fresh chain observation.

[10] Recursive. First steps toward automated AI research, 2026. 11 June 2026 developer report; methodology and correctness-audit discussion inspected 22 October 2026. https://recursive.com/updates/first-steps-toward-automated-ai-research.

[11] Recursive. Introducing flint, 2026. Developer release report inspected 22 October 2026; not independently reproduced. https://recursive.com/updates/introducing-recursive-flint.

[12] Recursive. Model terms, 2026. Last updated 21 October 2026; sections 1.3, 8.1 and 15 inspected 22 October 2026. https://recursive.com/legal/model-terms.

[13] Recursive. rec: Local inference engine, 2026. README and LICENSE inspected 22 October 2026; engine not executed. https://github.com/recursive-oss/rec/tree/8af99abaf50b33903147d89eed91d1bb3ac642dd.

[14] RecursiveSI. Flint model card, 2026. Model card and Apache-2.0 declaration inspected 22 October 2026; initial release history identifies 2a247be on 21 October 2026; weights not executed. https://huggingface.co/RecursiveSI/flint.

[15] Gregg Rothermel, Roland H. Untch, Chengyun Chu, and Mary Jean Harrold. Prioritizing test cases for regression testing. *IEEE Transactions on Software Engineering*, 27(10):929–948, 2001. Author-hosted primary abstract inspected 22 October 2026. https://digitalcommons.unl.edu/csearticles/9/.

[16] Jenny Zhang, Shengran Hu, Cong Lu, Robert Lange, and Jeff Clune. Darwin Gödel machine: Open-ended evolution of self-improving agents, 2025. Revision 3, 12 March 2026; primary abstract inspected; reported experiments not rerun. https://arxiv.org/abs/2505.22954v3.

[17] Jenny Zhang, Bingchen Zhao, Wannan Yang, Jakob Foerster, Jeff Clune, Minqi Jiang, Sam Devlin, and Tatiana Shavrina. Hyperagents, 2026. 19 March 2026; primary abstract inspected; reported experiments not rerun. https://arxiv.org/abs/2603.19461v1.