AGI.Eth Namespace + AGI Jobs v0

Institutional brief: identity → proof → settlement → governance

v0.1–v0.2 unified view (Alpha example; ENV_SET general)

Executive charter

Identity

- Roles bound to keys (ENS/DID).
- Accountability by default.

Evidence

- Deterministic runs emit signed proof bundles.
- Replay beats rhetoric (audit as replay).

Settlement

- Escrow releases only after validation.
- Receipts are settlement-grade artifacts.

Governance

- Policy gates, upgrades, emergency brakes.
- Operator override is final (autonomy is bounded).

Invariant: no value without evidence · no autonomy without authority · no settlement without validation

<entity>.(<env>.)<role>.agi.eth

role ∈ {club, agent, node} • env ∈ ENV_SET (optional; e.g., alpha, x, …)

Example (env = alpha)

Validator (club): alice.club.agi.eth | alice.alpha.club.agi.eth
Agent: helper.agent.agi.eth | helper.alpha.agent.agi.eth
Node: gpu01.node.agi.eth | gpu01.alpha.node.agi.eth

Sovereigns / AGI Businesses (general)

Global sovereign: <sovereign>.agi.eth
Ecosystem root: env.agi.eth (e.g., alpha.agi.eth, x.agi.eth)
AGI Business: <business>.env.agi.eth (e.g., ops.alpha.agi.eth)

Scope rule: entity.env.role.agi.eth is recognized & developed only within env.agi.eth.

Official environment package (issued by AGI King)

env.agi.eth • env.agent.agi.eth • env.node.agi.eth • env.club.agi.eth

Optional aliases (env-local; not official by default)

agent.env.agi.eth • node.env.agi.eth • club.env.agi.eth (recognized only within env.agi.eth unless explicitly whitelisted)

Recognition & scope

Name form	Class	Recognized in any env?	Developed / governed where?
entity.role.agi.eth	Global role identity	Yes	By the global role namespace + each env
entity.env.role.agi.eth	Env-scoped identity	No	Only inside env.agi.eth
env.role.agi.eth	Env-role mountpoint	No	Only inside env.agi.eth
role.env.agi.eth	Optional alias mountpoint	No (default)	Only inside env.agi.eth (optional recognition)

Best-practice guardrails

- Reserve {agent,node,club} under every env.agi.eth so they cannot be mistaken for AGI Businesses.
- Optional aliases are env-local conveniences; “official” status is granted only via the env registry (recognizedAliases).
- Keep semantics stable; let endpoints and metadata churn behind wildcard + CCIP-Read resolvers.

Registry-as-genome (autopoiesis)

Autopoietic control loop

- Membrane: env.agi.eth defines the ecosystem boundary.
- Organs: role roots (agent/node/club) separate duties.
- Genome: registry.agi.eth publishes ENV_SET + canonical packages + recognizedAliases.
- Metabolism: resolvers/gateways turn names into live endpoints.
- Immune system: status + proofs + slashing detect and correct drift.

Environment registry (machine-checkable)

```
{
  env, state,
  canonicalPackage: {
    root, agentMount, nodeMount, clubMount
  },
  recognizedAliases: [ ... ]
}
```

Invariants (what must not change as the system scales)

- Names classify actors unambiguously (role suffixes are globally meaningful).
- No payout without validated proof; no settlement without validation.
- Official recognition is registry-driven (not self-asserted).
- High-churn metadata lives behind resolvers; trust anchors stay minimal and stable.

Physics + Game Theory framing

Free-energy intuition (operational best practice)

- Minimize “coordination energy”: fewer on-chain writes, less governance overhead, deterministic ops.
- Minimize “namespace entropy”: one canonical form, registry-based recognition, reserved labels.
- As change-rate increases, push churn into resolvers (wildcards + CCIP-Read), not into naming.

Incentive design (dominant strategies)

- Workers earn only for validated proof (no counter-party trust).
- Validators are stake-bonded; misbehavior becomes negative-sum (slashing).
- Commit–reveal reduces herding/bribery leverage; disputes resolved by replay evidence.

Control objective (high level)

Stability = throughput with bounded risk
→ meter (α-WU) → validate → settle ($AGIALPHA) → update policy

Thermostats: parameters (fees, burns, tier multipliers, quorum, slashing) adjust using observed signals (throughput, disputes, SLO drift).

One system, three surfaces

AGI-Alpha-Agent-v0

- Meta-agentic orchestration
- Planning + evaluation loops
- R&D sandbox → production interface

AGIJobsv0 (Work OS)

- Contracts kernel + paymasters
- Job registry, escrow, settlement
- Control plane: APIs, dashboards, CI

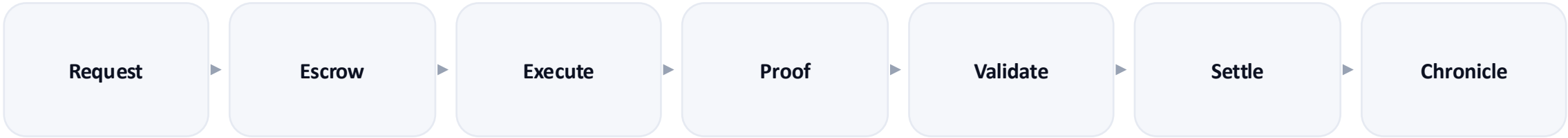
AGI-Alpha-Node-v0

- Deterministic runtime + sidecars
- Metering + artifact packaging
- Validator/sentinel services + observability

Proof-sync: hashes + signatures + attestations

End-to-end job lifecycle

Invariant: no payout without validated proof.



What settles

- Artifacts: content-addressed outputs (hash/CID/URI).
- Proof bundle: environment pins, logs, traces, metering telemetry.
- Validation: commit–reveal attestations + dispute window.
- Receipt: validated α-WU + signatures → escrow release + Chronicle entry.

Proof bundle (settlement-grade evidence)

Minimum bundle contents

- JobSpec + acceptance tests + policy context.
- Container digest / SBOM + pinned dependencies + deterministic seeds.
- Inputs/outputs as hashes (and immutable pointers).
- Logs, traces, metering telemetry (α-WU inputs).
- Signatures: worker/node + validator attestations.

Replay + Attest + Settle

If it cannot be replayed, it does not settle.

Commit–reveal validation (why it matters)

- Commit phase locks hashed verdicts to reduce herding and bribery leverage.
- Reveal phase publishes evidence (replay traces, tests, QV/SLO scoring).
- Disagreements open disputes; slashing makes dishonesty negative-sum.

Metrology + settlement

α-Work Units (α-WU)

- Canonical, hardware-normalized measure of verified work (policy-parameterized).
- Computed from signed metering telemetry × difficulty tier × quality score.
- Fails acceptance/SLO → 0 credit.

$AGIALPHA utility

- Stake: bond participation; Sybil resistance; slashable accountability.
- Settle: jobs paid via escrow; release after validation; fee routing + optional burn.
- Coordinate: epoch accounting (α-WU); routing signals; governance parameters.

Token coupling (thermostat): supply and incentives remain tethered to validated α-WU

Signals: validated α-WU/epoch · dispute rate · SLO drift · burn/emission ratios
Actuators: fee splits · burn fraction · tier multipliers · quorum/slashing parameters

Utility token only: required for protocol operation; no equity, profit rights, or claims on an entity.

AGI Alpha Nodes

What is an AGI ALPHA Node?

- Containerized runtime that is ENS identified, staked, and authorized to execute/validate AGI Jobs.
- Binds identity to <name>.alpha.node.agi.eth for trustless discovery.
- Produces measurable work that can be settled on-chain (proofs + receipts).

Roles (clear accountability)

- Worker: executes deterministically; publishes artifacts; claims settlement after validation.
- Validator: commit–reveal attestations; scores SLO + output quality; slashable for dishonesty.
- Sentinel: monitors health/drift; triggers local pause + escalation; preserves audit posture.

Operator UX (institutional posture)

- One-click/container-first deployment with boot-time safety checks (ENS, stake, contracts).
- Signed telemetry + tamper-evident audit trails; dashboards (Prometheus/Grafana).
- Fail-closed controls: circuit breakers, local pause, incident playbooks, key custody.

Universal deployability (Agents + Businesses)

Principle: one name, infinite deployability

- Register a canonical L1 ENS handle (e.g., defi.agi.eth) as the point of contact.
- Store L2/cross-chain addresses and metadata in ENS records for frictionless interaction.
- Keep multi-chain complexity and fractional token relationships “behind” one human-readable identity.

Practical pattern (best practice)

- Use ENS text records (ENSIP-5) for endpoints, capabilities, and provenance pointers.
- Use wildcards (ENSIP-10) + CCIP-Read (EIP-3668) for high-churn records at scale.
- Normalize names (ENSIP-15) before hashing/lookup to reduce spoofing and ambiguity.

Engineering posture

- Pre-alpha stacks MUST stay policy-bounded (pause, allowlists, rate limits).
- Prefer reproducible builds + pinned deps; export audit packs by default.
- Treat the env registry as the “source of truth” for official recognition.

Deployment (Green Path)

Pilot (Green Path)

- Launch workspace (Codespaces recommended) and run: make operator:green
- Pass criteria: ✅ Day-One Utility banner + default uplift guardrail
- Review artifacts: JSON + HTML dashboard + PNG snapshot + owner controls snapshot

Owner controls (fail-closed)

- Pause/resume: make owner-toggle
- Restore defaults: make owner-reset
- Treat “green wall” as deployable truth; block unsafe changes

One-click deploy (illustrative)

```
npm run deploy:checklist
npm run deploy:oneclick:auto -- --config deployment-config/<network>.json --network <network> --compose
docker compose --env-file deployment-config/oneclick.env up --build -d
```

Adoption playbook

Start with proofs; scale autonomy only as verification stays ahead

Pilot

- Pick one workflow
- Define acceptance tests
- Run private nodes
- Export audit packs

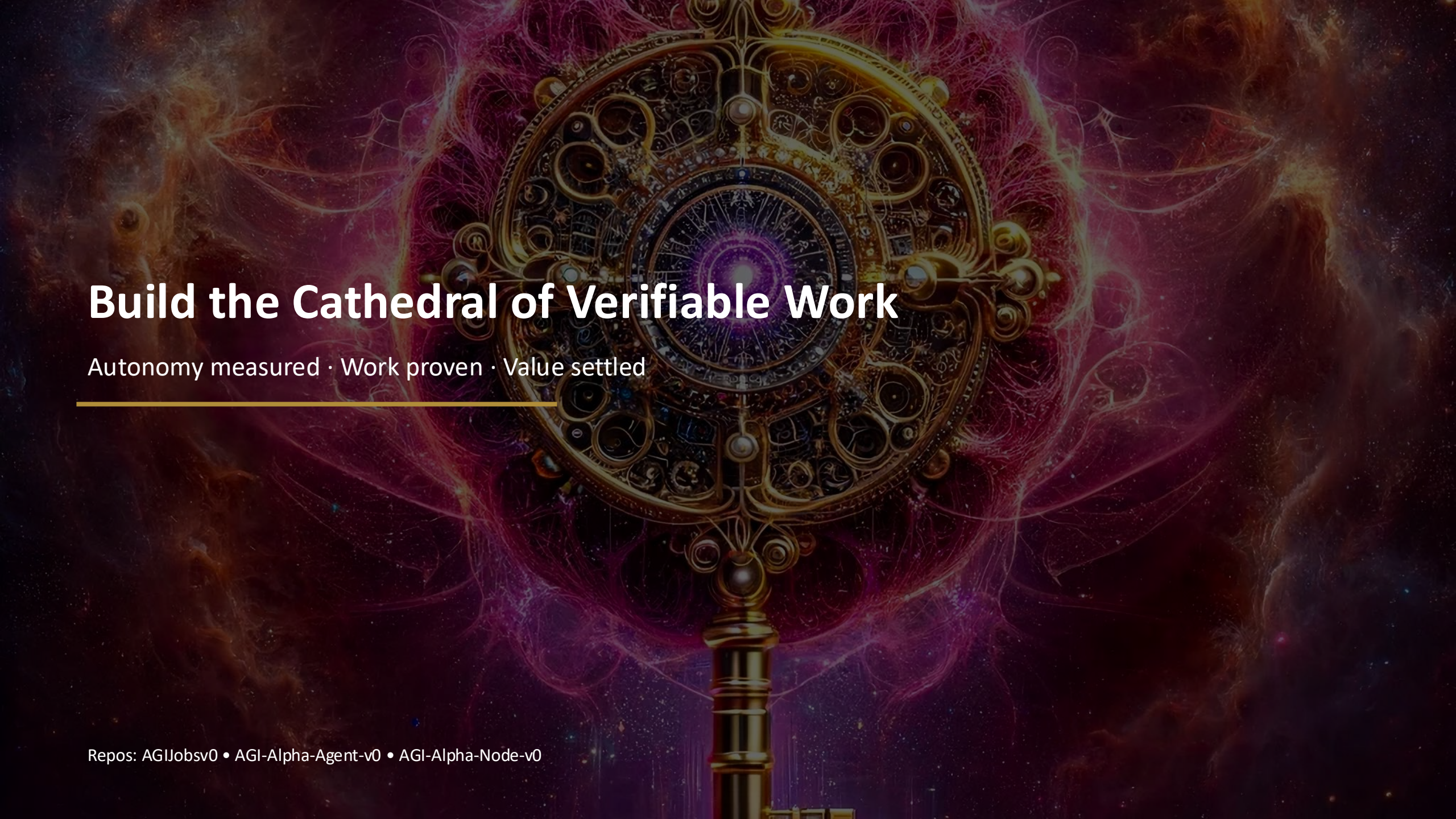
Harden

- Add validator quorum
- Enable policy brakes
- Increase replay coverage
- Establish key custody + rotation

Scale

- Route more workloads
- Publish α-WU indices
- Open markets by policy
- Expand autonomy only after gates pass

Principle: expand power only as fast as proofs and brakes are proven.


Build the Cathedral of Verifiable Work
Autonomy measured · Work proven · Value settled
Repos: AGIJobsv0 • AGI-Alpha-Agent-v0 • AGI-Alpha-Node-v0