



a-AGI Insight v1

Beyond Human Foresight

A zero-data, offline-first, meta-agentic forecasting demo: map AGI-driven sector phase transitions and actionable decision frontiers.

What this demo delivers

MODE

Offline-first

Runs with or without API keys

ENGINE

Meta-Agentic

Falls back to local tree-search runtime

OUTPUT

Decision frontier

3-D scatter: effectiveness × risk × complexity

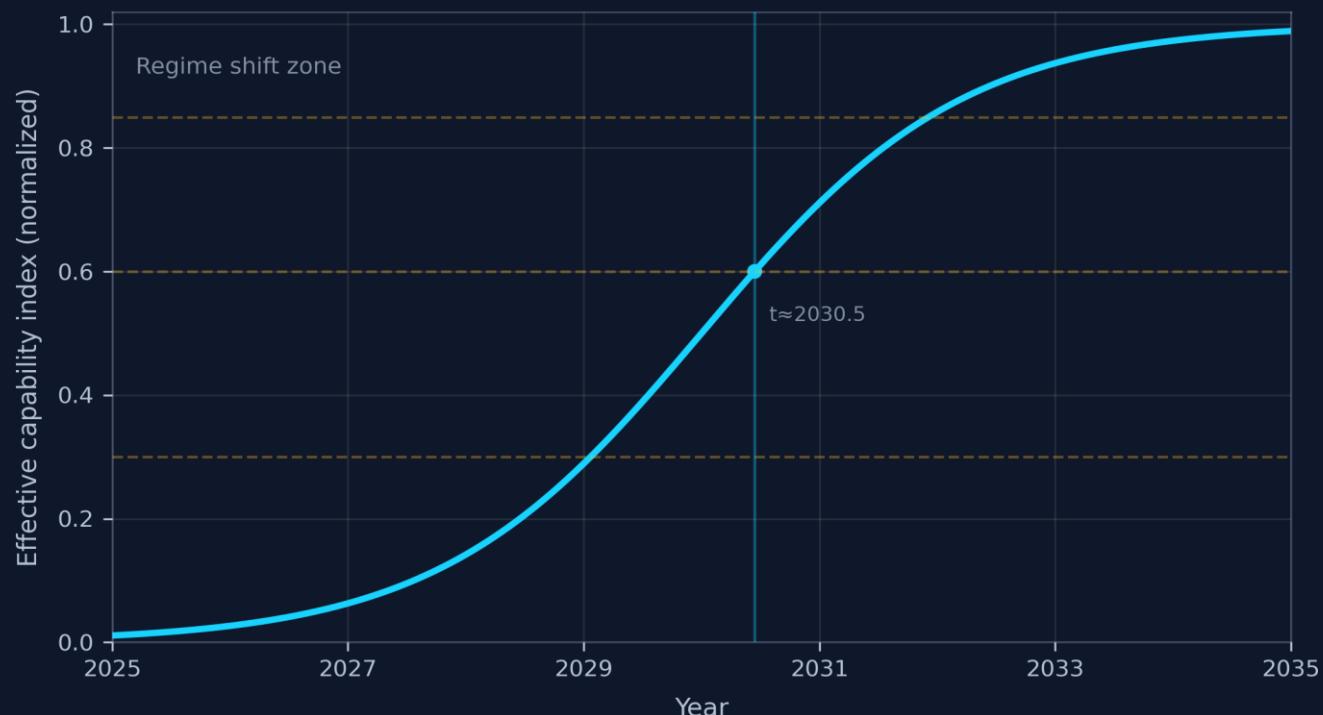
In one run, you obtain:

- A calibrated capability curve and scenario controls (episodes, curve parameters, offline mode).
- A population of candidate innovations evaluated across competing objectives (effectiveness, risk, complexity).
- Operational outputs: first affected step/year, live agent status, and exportable results for decision playbooks.

AGI impact is not linear — it arrives as phase change

This demo operationalizes “tipping points” as thresholds on a capability trajectory, then explores response options under competing objectives.

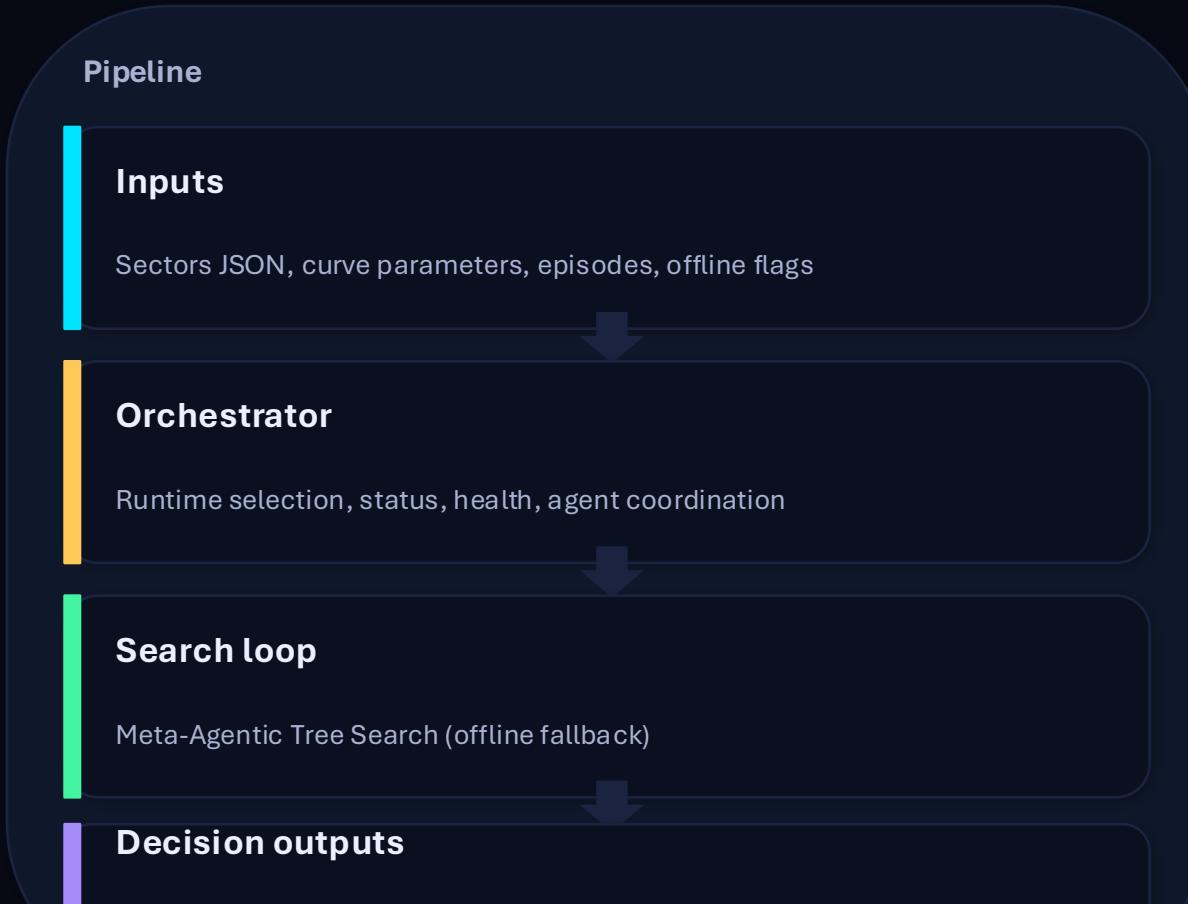
Capability curve (illustrative)



Interpretation

- Thresholds represent domain susceptibility (automation → design → discovery).
- Crossing a threshold is a regime shift: decisions must change, not just pace.
- The demo’s role: explore the option space, then surface robust strategies.

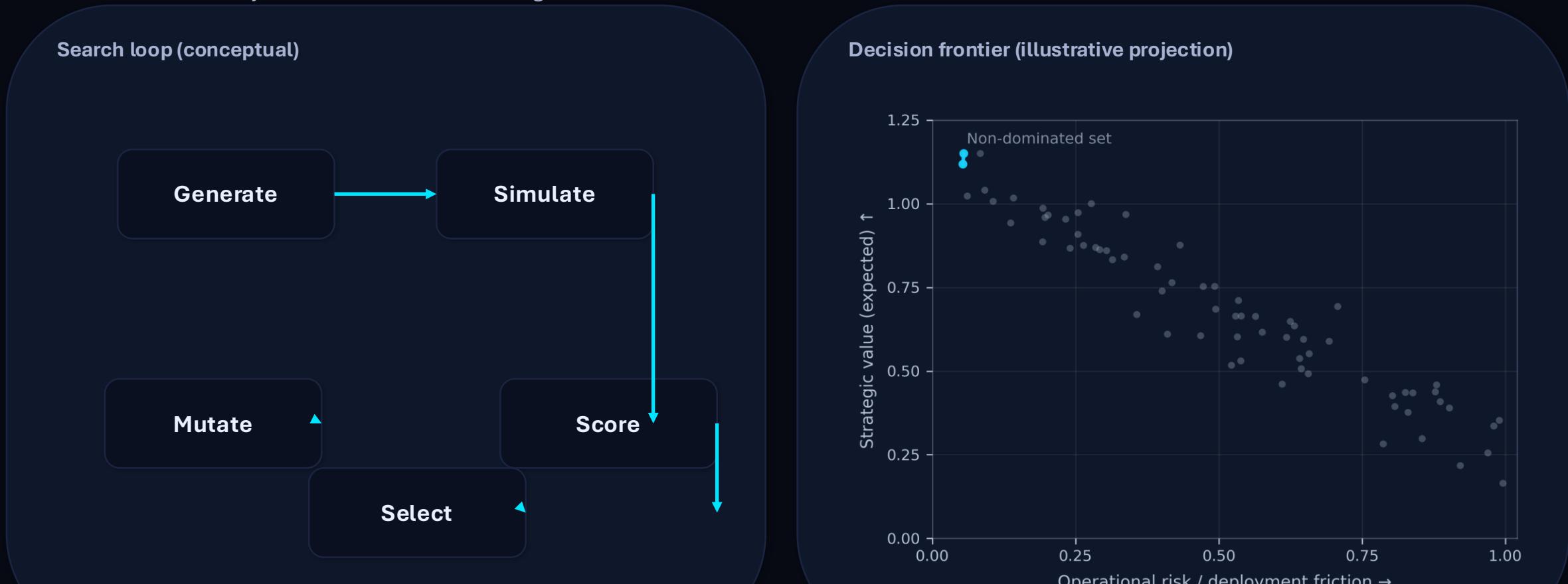
a-AGI Insight v1: meta-agentic orchestration around an offline-capable search loop



- CLI and web modes; can run headless.
- Health/status endpoints + agent heartbeats.
- Local audit ledger for traceability; optional pinning via gateway.

Offline-capable search loop: explore → evaluate → retain the frontier

The demo automatically falls back to a local Meta-Agentic Tree Search when API credentials are absent.



Thermodynamic lens: stability vs uncertainty as scenario controls

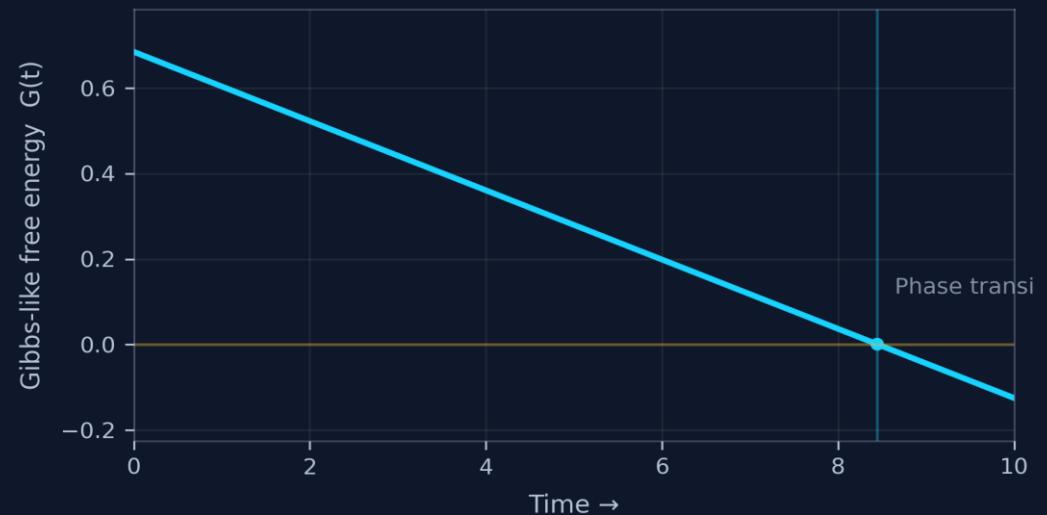
The demo exposes explicit “energy” and “entropy” knobs. A useful interpretation is a free-energy threshold that marks regime change.

Model sketch (interpretation)

$$G = E - T \cdot S$$

- E (energy): structural stability / barriers to change.
- S (entropy): uncertainty / degrees of freedom exploited by new capability.
- T: capability “temperature” — increases over time via the configured curve.
- Crossing $G \leq 0$ is a natural marker for “phase transition”.

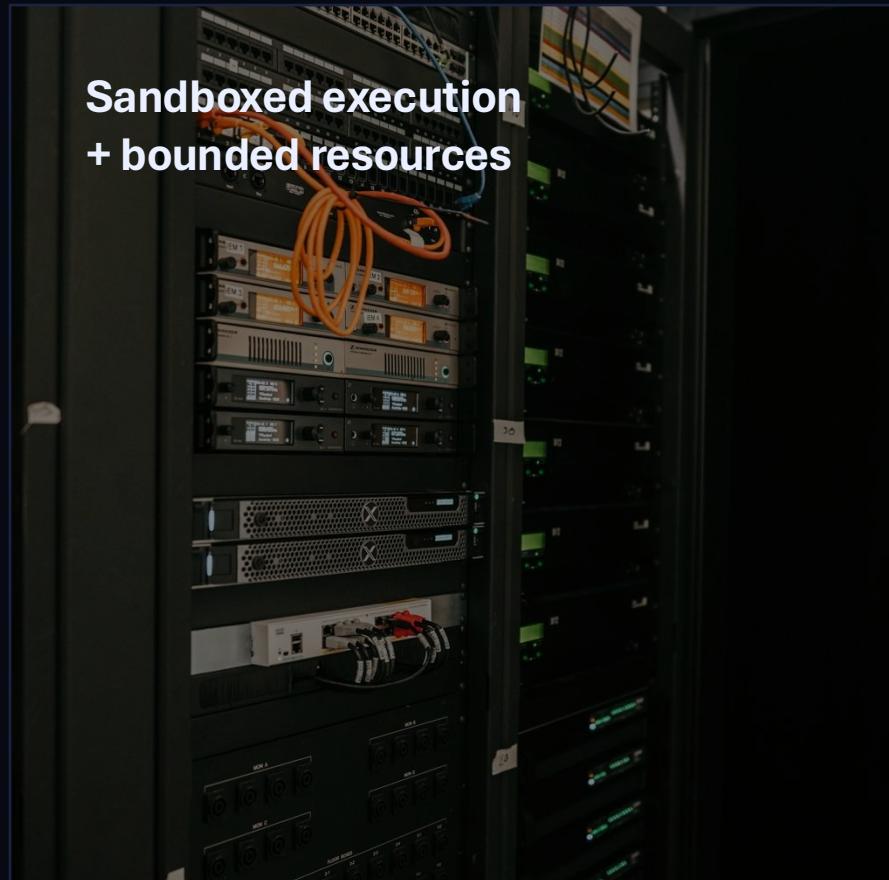
Phase marker (illustrative)



“Energy” and “entropy” are first-class CLI parameters for the Insight simulation.

Operational control surface: status, health, tokens, TLS, sandbox limits

A practical system is one you can deploy, observe, constrain, and audit.



Deployment & runtime controls (from the demo)

`AGI_INSIGHT_OFFLINE`

Force local inference models

`AGI_INSIGHT_LEDGER_PATH`

Local audit ledger path (audit.db)

`AGI_INSIGHT_BUS_CERT / KEY / TOKEN`

Secure gRPC transport and auth

`API_TOKEN`

Bearer token required by REST API

`SANDBOX_CPU_SEC / SANDBOX_MEM_MB`

Resource limits for sandboxed code

The deliverable is a decision surface, not a single forecast

Interpret outputs as: “what becomes possible” + “what becomes fragile” + “which moves dominate under constraints.”

Disruption map

- Affected step/year output (CLI + dashboard).
- Attach to sector definitions for repeatable runs.
- Use as an early-warning system.

Option frontier

- Population evaluated across objectives.
- 3-D scatter in dashboard: effectiveness × risk × complexity.
- Pick robust points, not “best” points.

Operational trace

- Audit ledger persisted to a local DB path.
- Health, status, and metrics endpoints for observability.
- Offline runs available by design.

A reproducible run is a compact contract: parameters → outputs → ledger

1

Define sectors

Provide a sectors JSON (sample file path is included in the repo).

alpha-ag-i-insight-v1 --episodes 5

2

Select runtime

Use hosted models when configured; otherwise the demo falls back to the offline search loop.

3

Run episodes

CLI wrapper: alpha-ag-i-insight-v1 --episodes N (or python -m ...).

4

Inspect status

Use /status or agents-status to monitor heartbeats and restart counts.

5

Export decisions

Review the final population scatter, affected steps/years, and persist the ledger.

Runs anywhere: local search loop + optional local inference

Guarantee no network access with one flag; package includes guidance for wheelhouse installs and cached assets.

Guarantee offline execution

```
AGI_INSIGHT_OFFLINE=1 alpha-agl-insight-v1 --episodes 5
```

- If OPENAI_API_KEY is empty, the demo uses offline mode automatically.
- When the host cannot reach the internet, demos continue using cached data; optional downloads are skipped.
- Offline guidance includes wheelhouse builds and local model weights.

Air-gapped operational profile

- Prebuild a local wheel cache (wheels/) for installation.
- Optional: set LLAMA_MODEL_PATH to a local .gguf weight for inference.
- Build browser assets to enable offline web demo caching.

Dashboard & API: monitor, steer, and validate runs in real time

The repo documents a web workflow (uvicorn + React client + Docker Compose) and defines service health criteria.

Stack (as described)

FastAPI orchestrator

REST: /healthz, /status, /metrics

Web client (React)

npm ci → npm run dev/build

Compose/Helm

Web UI at :8080; API at :8000

Streamlit/minimal UI

Fallback UI for headless servers

“Healthy when...”

- :8000/healthz returns 200 for orchestrator.
- :8000/status exposes heartbeats and restart counts.
- :8080 returns 200 for web container.

Live plot:
3-D scatter

Operational paths: Compose, Helm, and edge runner

The repository documents resource-restricted agents and persistent storage for the audit ledger.

Docker Compose

```
cp .env.sample .env
docker compose up --build
# dashboard:
http://localhost:8080
```

Helm (Kubernetes)

```
helm upgrade --install
alpha-demo \
./infrastructure/helm-
chart \
--values
./infrastructure/helm-
chart/values.yaml \
--set env.RUN_MODE=web
# optional audit-ledger
persistence:
#   --set
persistence.enabled=true \
#   --set
persistence.size=5Gi
```

Edge runner

```
python edge_runner.py \
--agents
manufacturing,energy \
--cycle 60 --loglevel
info
# forwards to:
alpha_factory_v1.edge_runner
```

Constrain power: isolate execution, require tokens, encrypt transport, log everything

Sandbox bounds

CPU and memory limits for sandboxed code paths.

API tokens

REST API requires API_TOKEN; fail fast when unchanged.

Secure bus

gRPC bus can use cert/key/token; TLS can be enforced.

Network isolation

Agents can run without outbound network access (Compose hardening).

Observability

Metrics at /metrics; traces and dashboards described in repo.

Governance hooks

Repo documents governance & compliance primitives and policy runbook.

Game theory lens: when the payoff matrix changes, equilibria move

Use the demo outputs to identify when incumbency advantages invert and which strategic moves remain dominant under uncertainty.

Decision patterns to test

- Commitment timing: pre-position capabilities before thresholds are crossed.
- Cooperation vs competition: where coordination becomes a dominant response.
- Information asymmetry: who sees the regime shift first, and how fast do others adapt?
- Option value: choose frontiers that preserve maneuverability under shock.

Concrete hook to the demo

- Use the 3-D scatter to select strategies that dominate along a constraint (e.g., low risk at fixed effectiveness).
- Use affected steps/years as the “move horizon” for investment and policy sequencing.
- Use the ledger + parameters to run adversarial “what-if” counterfactuals.

Hamiltonian lens: optimize value under invariants

The demo's multi-objective population can be treated as an energy landscape: local maxima, global frontiers, and constraint-induced trajectories.

Conceptual mapping

$$H(x) = -V(x) + \lambda_1 \cdot \text{Risk}(x) + \lambda_2 \cdot \text{Complexity}(x)$$

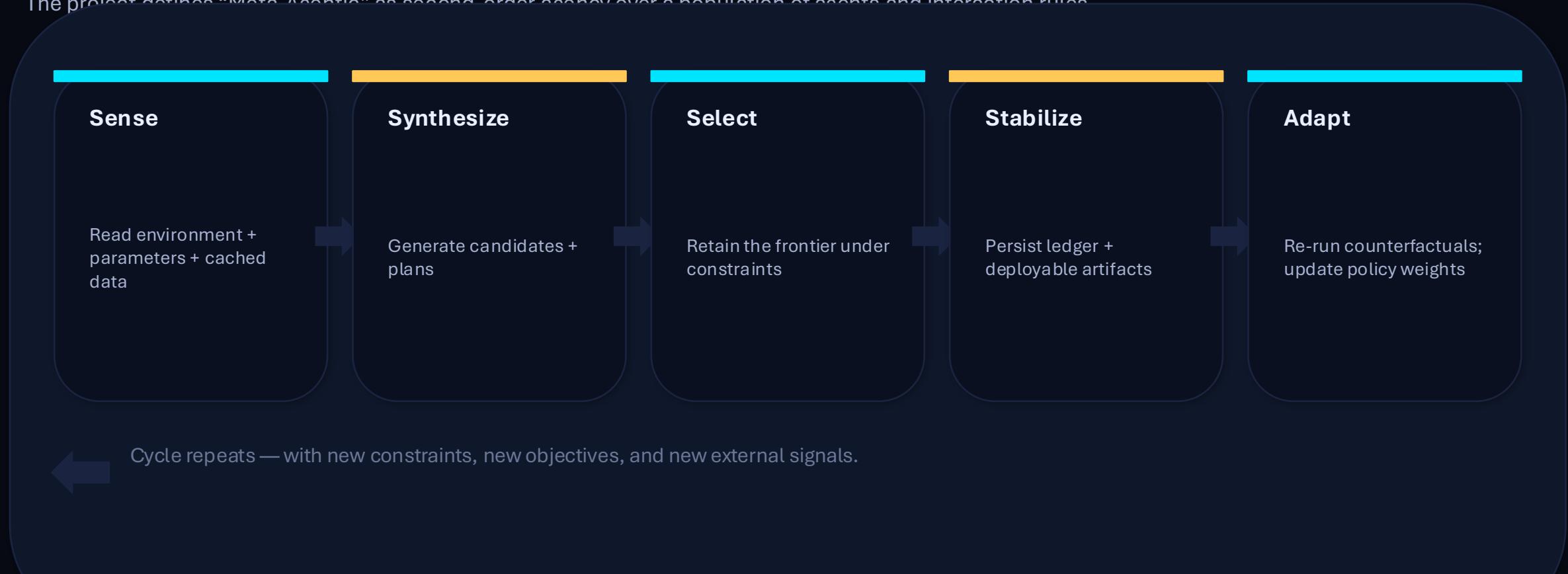
- x : a strategy or innovation candidate.
- V : value / effectiveness; penalties encode constraints.
- Vary λ to sweep policies: “risk-tight” vs “speed-first”.
- The frontier is the set of non-dominated solutions across λ .

Why this matters

- It turns debates (“should we move fast?”) into parameterized trade-offs.
- It makes governance measurable: choose λ values, then test stability.
- It supports portfolio thinking: allocate across multiple frontier points.

Meta-agentic systems behave like living organizations: they generate, select, and reconfigure

The project defines “Meta-Agentic” as second-order agency over a population of agents and interaction rules.



Why this matters: compounding advantage accrues to superior foresight

A practical foresight engine shifts decision latency from months to minutes... and makes counterfactual testing routine.

Capital allocation

Invest before thresholds; exit brittle positions before shocks.

Industrial policy

Sequence training, safety, and standards against the move horizon.

Product strategy

Select frontier points aligned to risk appetite and time-to-ship.

Security posture

Run air-gapped, token-gated, observable deployments as baseline.

Scientific & energy regimes

Target high-leverage discovery loops; treat compute/energy as a civilizational resource.

This is a demo forecasting engine — treat outputs as structured hypotheses

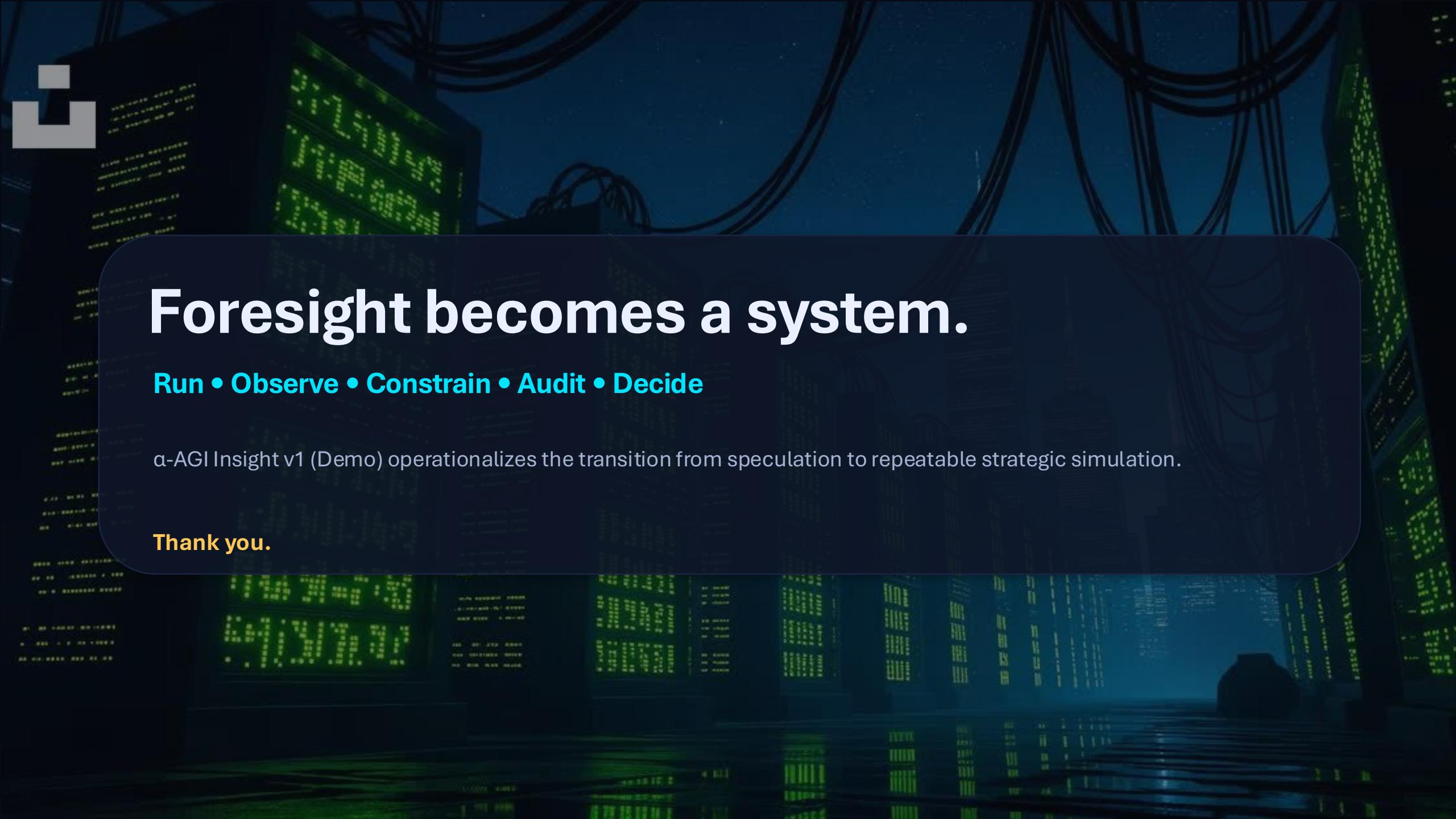
Rigor comes from repeatability: parameter sweeps, adversarial counterfactuals, and audited runs — not from a single run.

Best practices

- Run ensembles: vary curve, energy/entropy, mutation/crossover rates.
- Use offline mode for reproducible air-gapped runs.
- Log and compare results via the audit ledger path.
- Treat 3-D scatter as a negotiation space between stakeholders.

Common failure modes

- Overconfidence in a single trajectory (ignore uncertainty).
- Optimizing for one metric while silently increasing another (risk/complexity).
- Skipping governance: running without tokens/TLS in sensitive contexts.
- Treating the demo as “truth” rather than a disciplined generator of hypotheses.



Foresight becomes a system.

Run • Observe • Constrain • Audit • Decide

a-AGI Insight v1 (Demo) operationalizes the transition from speculation to repeatable strategic simulation.

Thank you.