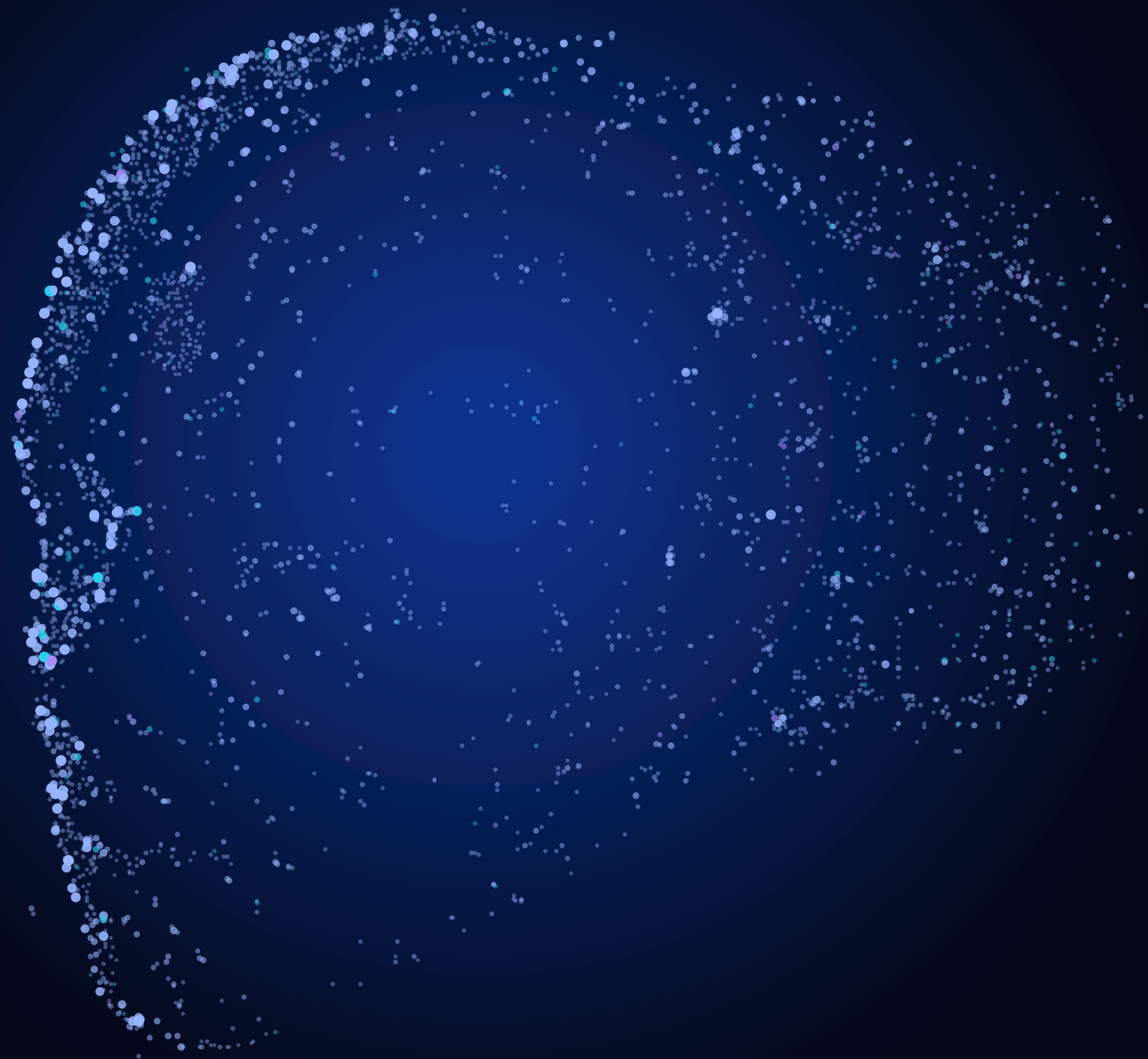




The control layer for AI workflows

Orchestrate models and agents, measure how they reason, keep every step accountable.

Design white paper v2.0 • neuor.ai • October 2026

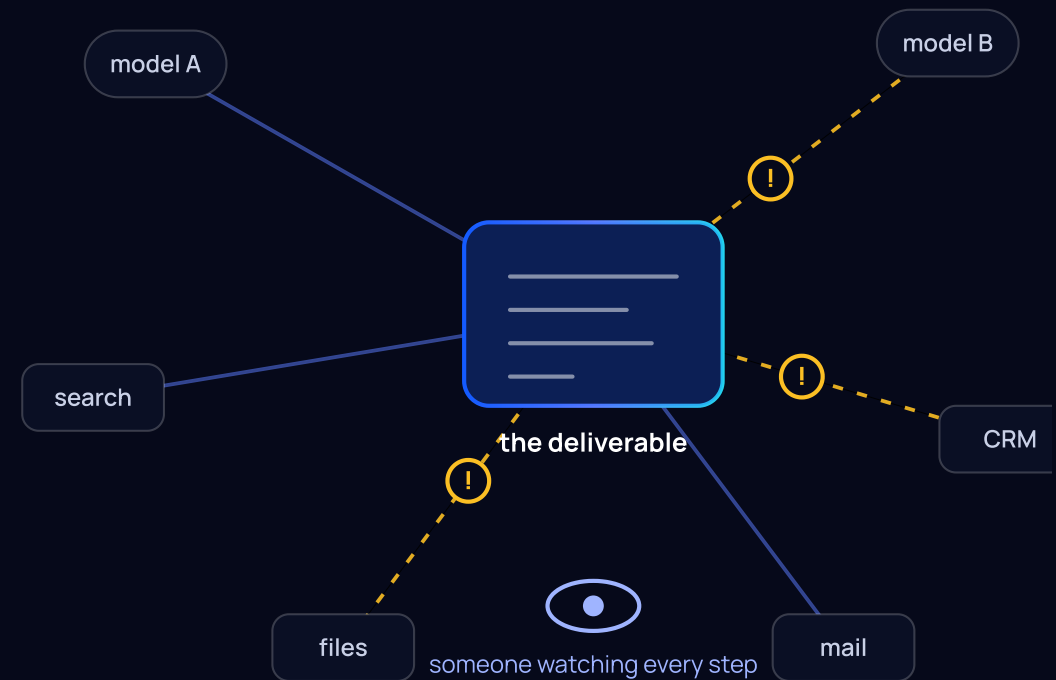


— THE PROBLEM

Generating an answer is only the beginning

A useful AI system must obtain the right information, coordinate tools, respect permissions, recover from interruptions, and show whether the deliverable meets its requirements. As workflows span more models, apps and teams, the cost of debugging, reviewing and recovering failures can swallow the apparent savings.

- Provider interfaces differ; tools fail; context grows
- Plausible answers conceal missing evidence
- Buyers can't answer: what was allowed, what happened, why was it accepted?



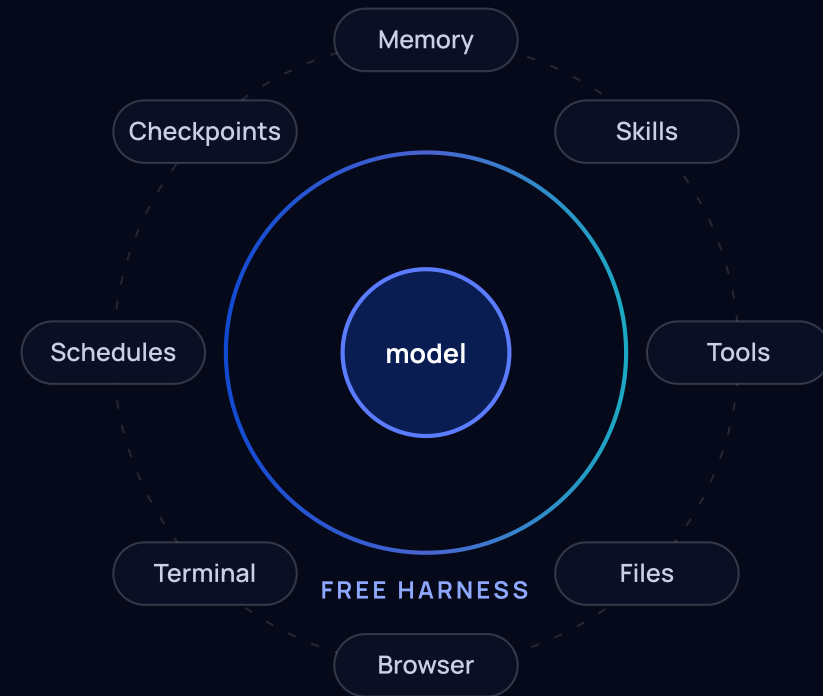
A free agent harness. Paid orchestration across models.

Free — single-model harness

Native macOS workspace. Projects, approvals, deliverables. Persistent memory, searchable sessions, reusable skills. Files, terminal, web, browser, MCP tools. Scheduled and background work. Checkpoints with rollback. Exportable run records.

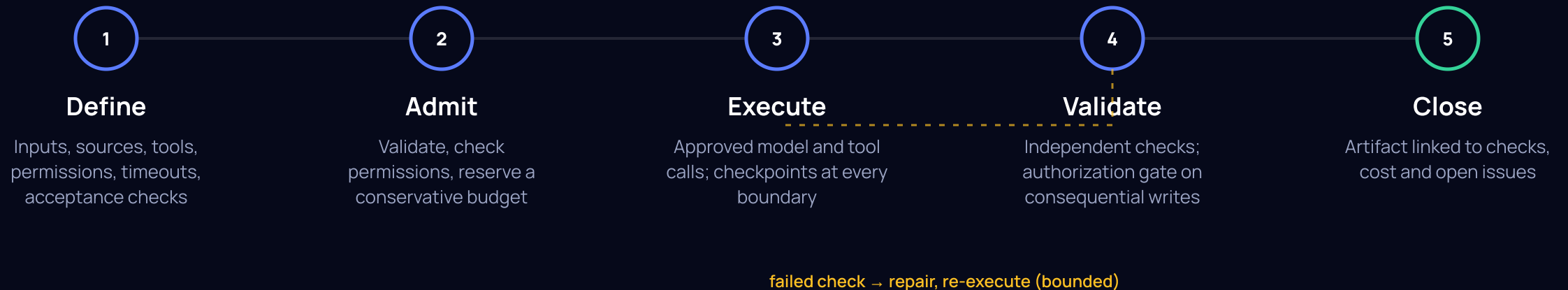
Paid — orchestration

Automatic model selection and role assignment. Parallel candidates, critique, repair and synthesis with explicit breadth and depth caps. Budget, deadline and regression stop rules. Team policies and administration.



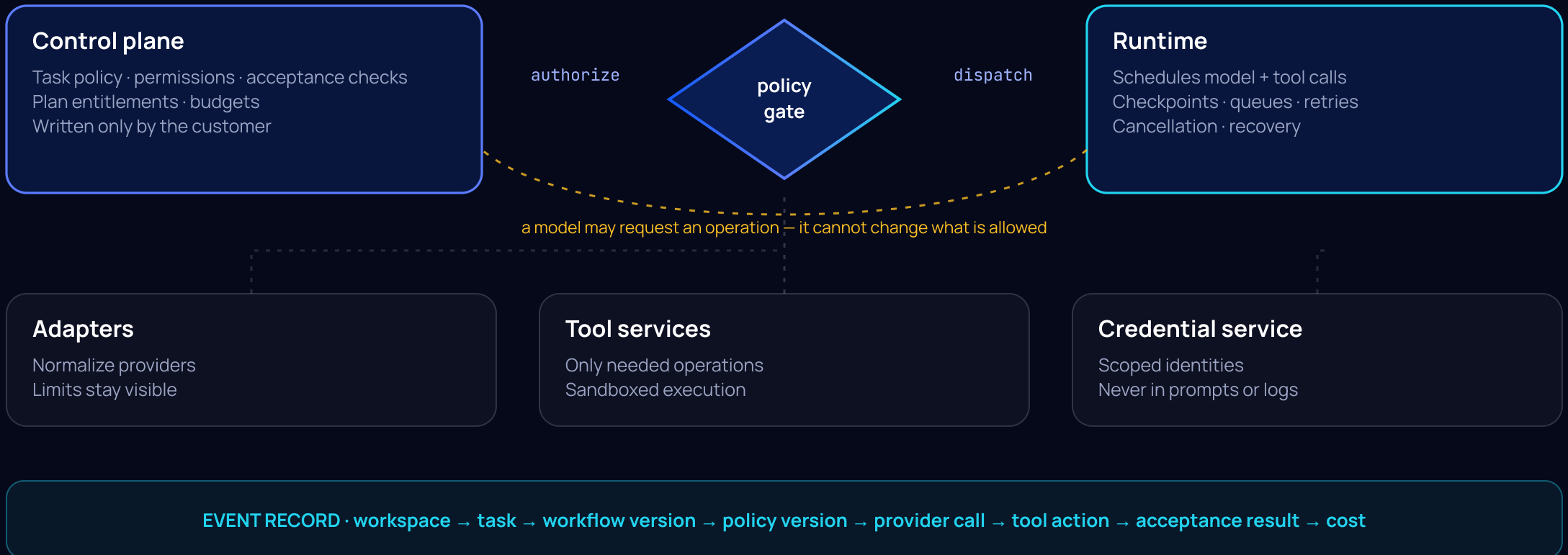
PAID ORCHESTRATION ACROSS MODELS

Intent in, accepted result out — or a clear exception



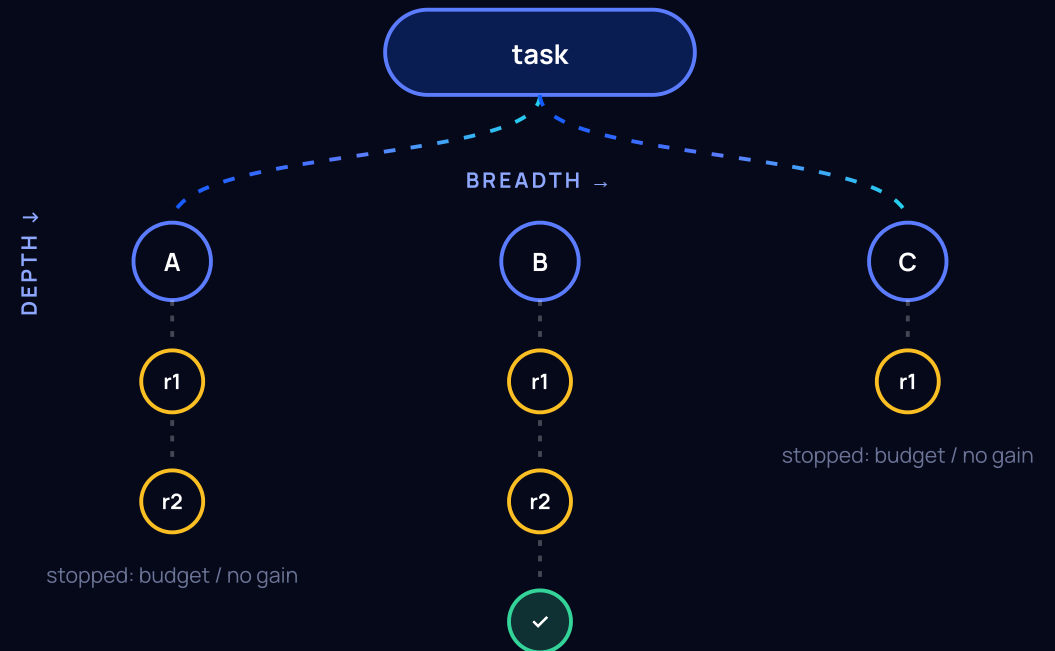
A model's generated instruction can never enlarge the permissions granted to its workflow. Permissions are set by the customer and enforced outside the model.

Customer intent is separated from execution



Reasoning effort is spent deliberately

- **Breadth** — candidate attempts. **Depth** — critique-and-repair rounds per attempt. Both capped explicitly.
- Start with one inexpensive attempt. Invoke another model only after a failed check. Reserve a stronger model for a narrow unresolved issue.
- A useful critique names a testable defect; repair addresses it and keeps the previous candidate.
- **Stop rules:** acceptance met, budget exhausted, deadline reached, or revisions stop improving. No pass → an exception naming the missing evidence.



B accepted after 3 bounded repair rounds · A, C kept as evidence

— HOW IT WORKS — IN PRACTICE

An agency's weekly client brief



retrieved instructions are data, never permission · delivery to the client is a separate authorized action

The reviewer accepts, returns specific corrections, or stops the workflow. Approval to create the draft never implies approval to send it.

The same pattern transfers: code review uses a diff, test output and a reproducible defect as evidence; document extraction uses the source passage and a field validation.

— WHY NOW

Models are converging. The harness is where the value goes.

Frontier models are interchangeable per task

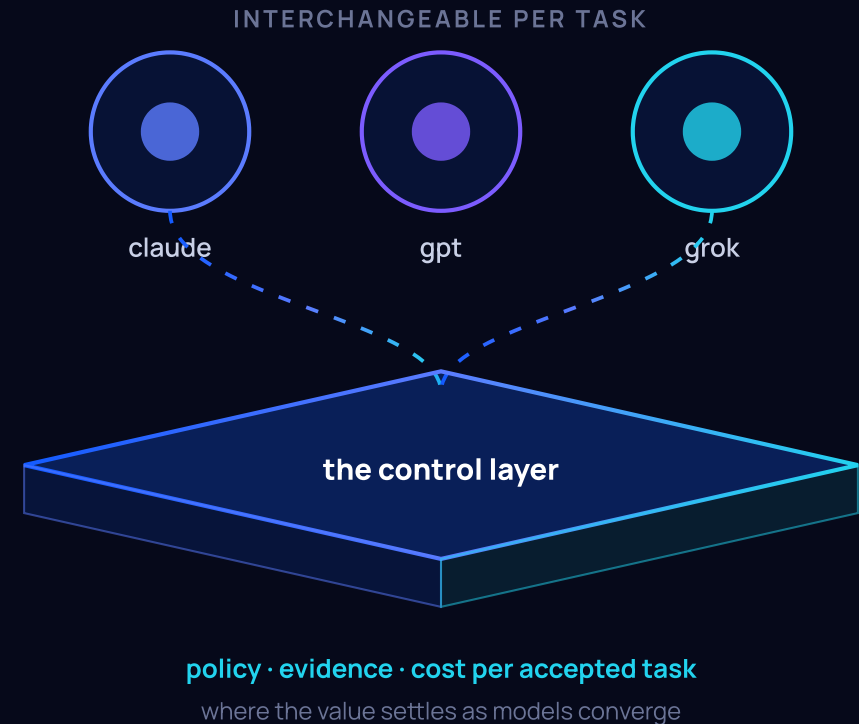
Routing by measured performance beats a permanent bet on one provider — and keeps the customer's leverage.

Orchestration infrastructure is a commodity

Checkpointed state machines and agent frameworks exist. The differentiated product is the policy, evaluation and operating experience.

Buyers now ask for evidence

Moving from demos to recurring work needs permissions, run records and cost per accepted task.



— CUSTOMERS

Recurring work with a visible finish line



Individuals

Free / Starter

RECURRING NEED

Everyday research, writing, file tasks

FIRST USEFUL WORKFLOW

A sourced brief or organized project



Automation agencies

Team

RECURRING NEED

Reuse tested processes across client workspaces

FIRST USEFUL WORKFLOW

Research briefs with source verification



Software teams

Developer / Team

RECURRING NEED

Control review cost, recover failures

FIRST USEFUL WORKFLOW

Code-change assessment with sandboxed tests



Operations teams

Business · after pilots

RECURRING NEED

Process routine documents, route exceptions

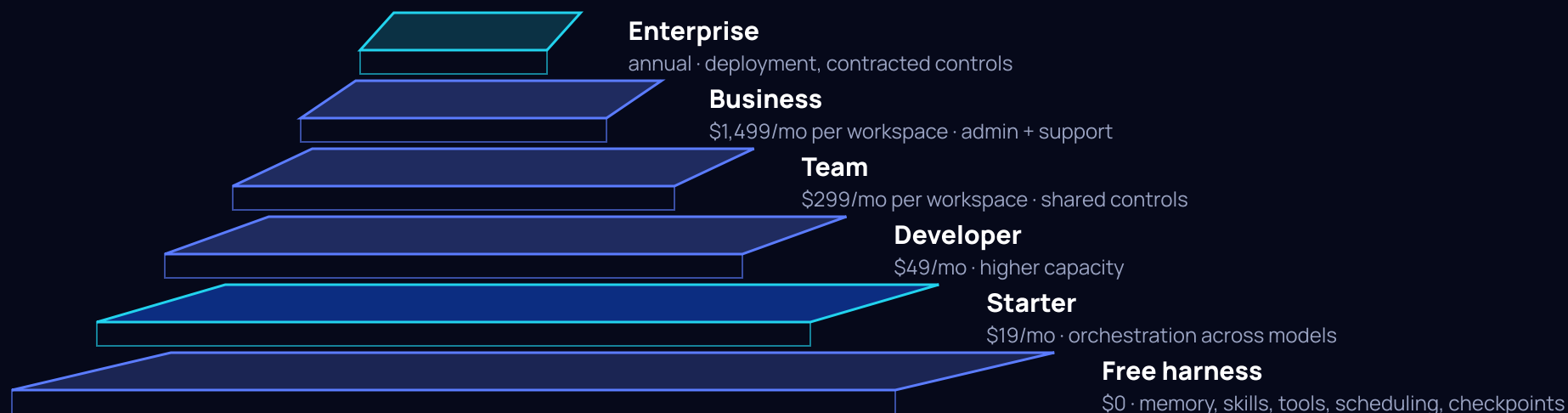
FIRST USEFUL WORKFLOW

Structured extraction with field-level checks



— BUSINESS MODEL

Freemium software. Inference never touches our margin.



Every paid plan includes Free. Customers pay providers directly for inference. Implementation \$3,000–\$15,000 · later marketplace at 15% of sales.

100,000 paying accounts

\$5.03M

monthly recurring revenue

81%

gross margin on paid costs

\$50.29

revenue per account / mo

\$40.74

gross profit per account / mo



Starter · 80% of accounts

\$1.60

Developer · 15% of accounts

\$0.73M/mo

Team · 4% of accounts

\$1.20M/mo

Business · 1% of accounts

\$1.50M/mo

Total \$5.03M MRR · \$60.4M run rate

Assumed monthly costs \$5 / \$5 / \$45 / \$300 per plan = \$955K. Doubling them leaves 62% margin; tripling, 43%. A hypothetical \$1,000 CAC pays back in ~24.5 months. Free users contribute \$0 and are not counted.

Illustrative scenario from the design white paper, not a forecast. Excludes churn, discounts, usage, enterprise, services and marketplace revenue, Free-tier costs and overhead.

Free adoption, measured upgrade, agencies as the channel

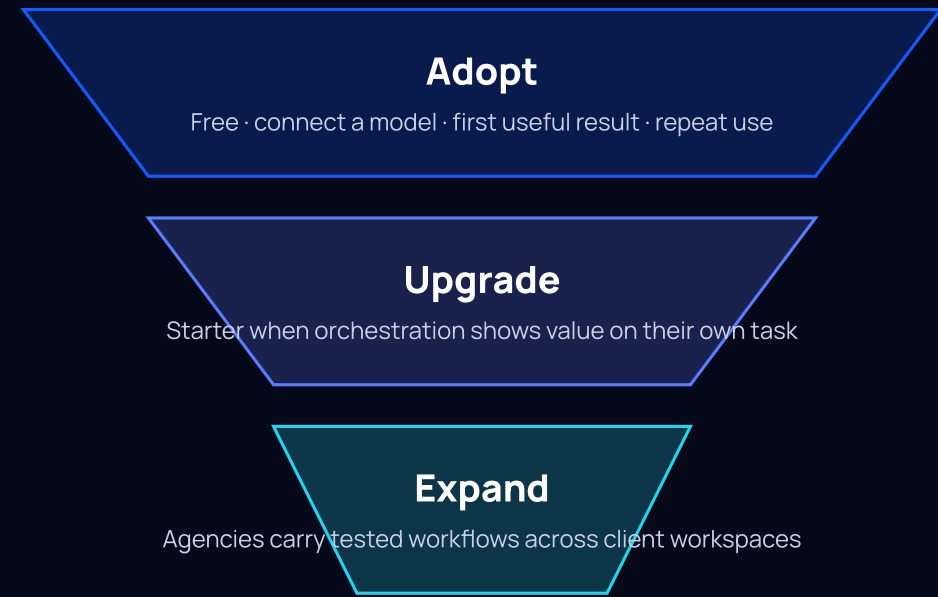
Five paid design partners

Agencies and software teams, each with a named buyer and a recurring workflow. Pilots end with a comparison the buyer can act on.

Free is a segment, not a trial

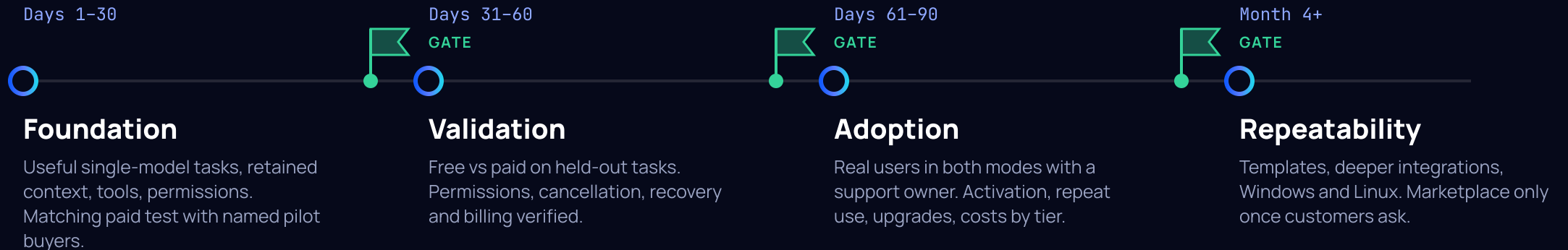
Its activity measures adoption; upgrades and renewals measure monetization. Free activity stays out of revenue-retention denominators.

The reason to upgrade must be the value of coordination, not withheld memory, tools, skills or basic safety.



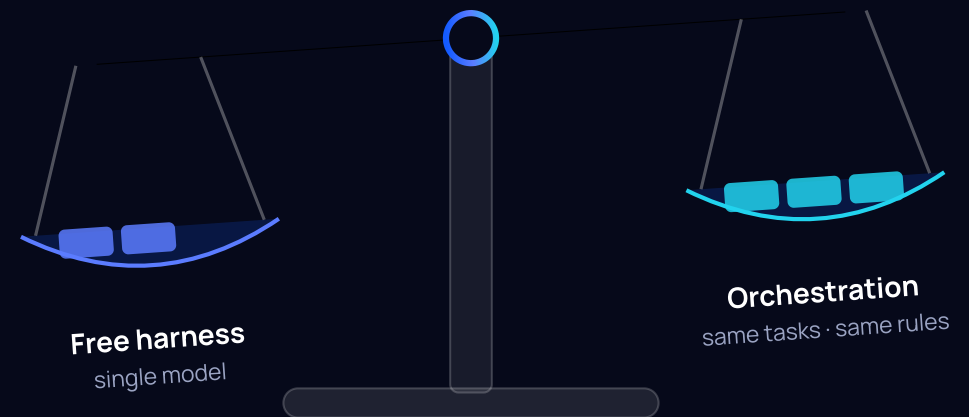
measured: activation → cohort conversion → paid retention → expansion

Every stage ends with an evidence gate



Orchestration has to beat the Free harness — on the same tasks

- Same tools, evidence and acceptance rules for every policy
- ≥ 100 held-out tasks per design partner; development set kept separate
- Failed attempts and reviewer time counted in the cost
- Blind human grading where judgment is required; model judges validated against labels
- Report by task category with uncertainty intervals; a faster-but-worse policy is a trade-off, not a winner

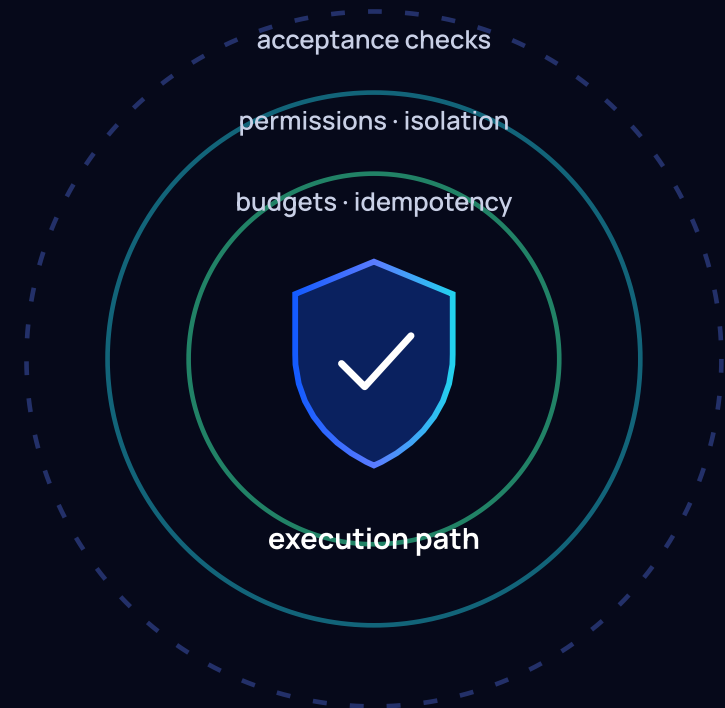


weights: accepted tasks · cost incl. failed attempts · reviewer time

if orchestration doesn't tip the scale, the customer shouldn't pay for it

Each risk maps to a control and a signal

RISK	CONTROL	SIGNAL
Confident incorrect output	Independent acceptance checks, evidence review	Rejected tasks, unsupported claims
Runaway cost or duplicate actions	Admission limits, reservations, idempotency, reconciliation	Budget overruns, duplicate incidents
Sensitive-data exposure	Scoped access, tenant isolation, secret handling	Access violations, data-flow exceptions
Weak Free-to-paid conversion	Measure cohorts, cost to serve, margins together	Conversion, retention, support cost



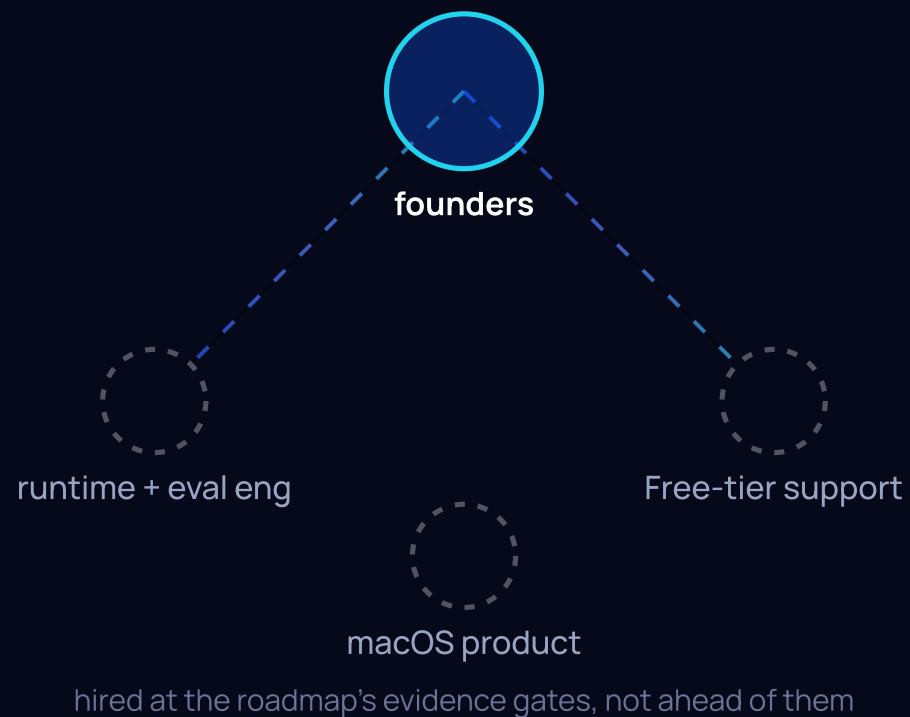
Founder-led, built by operators

Founding team

Operating background in network and telecom infrastructure, investment, and building internal AI operating systems for real business workflows. The product thesis comes from running recurring work through AI tools and paying the coordination cost first-hand.

Hiring with the first pilots

Runtime and evaluation engineering, macOS product, and a support owner for the Free tier.



— THE ASK

Design partners now. A round once demand is retained.

Early funding comes from founder capital, paid pilots and implementation work. Additional investment is tied to retained demand and a specific operating milestone. If you want to be a design partner or be first to the conversation when the round opens:

hello@neuor.ai · neuor.ai

Capabilities, pricing, milestones and financial scenarios describe proposals. No production performance, customer traction or investment return is represented as established. This deck is not an offer of securities.

