AI NATIVE SOLUTIONS

# Software is going to zero.

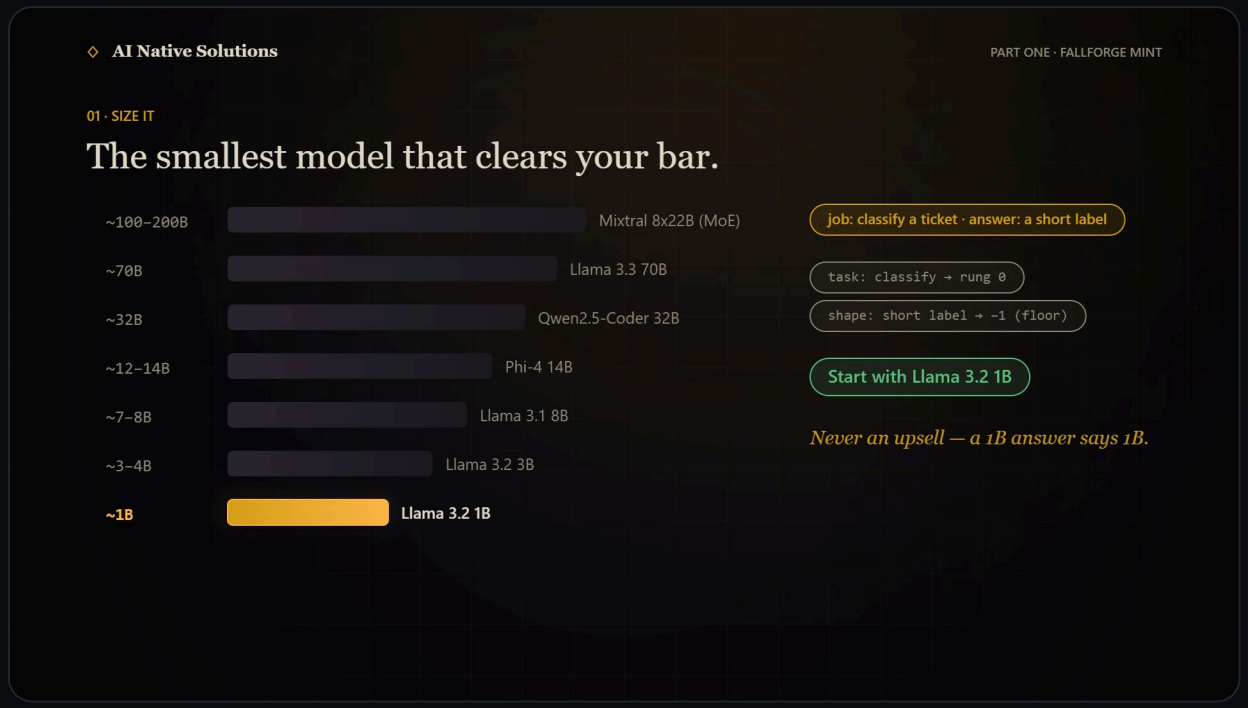
**What stays scarce is proof that it works, and ownership of the system it runs in.** fall-os is an operating system built on that bet — and an estate of 1,700+ public repositories already standing on it.

OPEN FALL-OS | ▶ WATCH THE 90-SECOND FILM | SEE WHAT'S LIVE | DOWNLOAD PDF

NEW IN THIS EDITION · FALLFORGE MINT

# Size it. Mint it. *Prove it.*

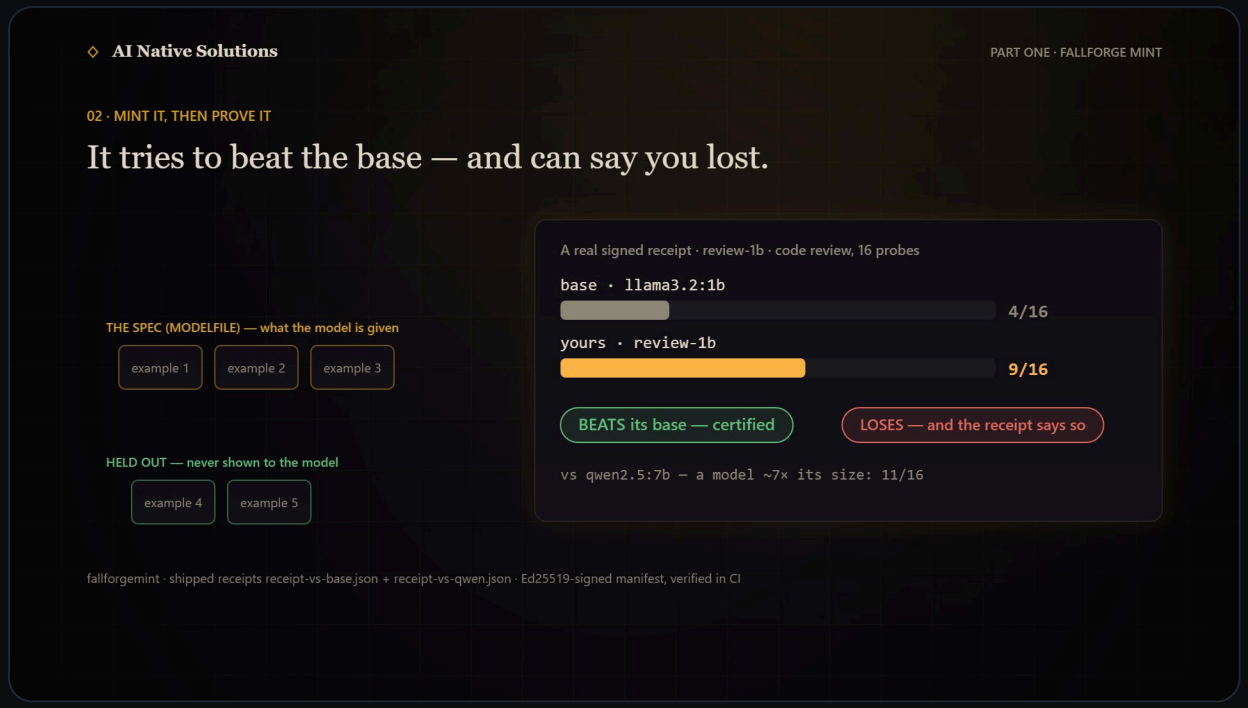
The sizer picks the **smallest open-weight model that clears your bar** — ~1B to ~200B, every factor shown, never an upsell. Mint it from a few of your own examples; prove it on examples it never saw.



NEW IN THIS EDITION · THE RECEIPT

# A receipt that can *say you lost.*

**review-1b** scored 9/16 against its base's 4/16 — BEATS. Against a model ~7× its size: 9/16 to 11/16 — LOSES. The held-out answers weren't in the spec the model was given, and anyone can re-run it.



NEW IN THIS EDITION · THE ESTATE

# One front door. *Every door live.*

Seven stages, one pipeline. The dispatcher routes every job: **7** organs run themselves, **9** wait for a human key, **8** are honestly to-do.


▶ WATCH THE 90-SECOND FILM

EVERY NUMBER, SOURCED

01 · THE PROBLEM

# You are renting *every part* of your own business.

A modern small firm pays per seat, per tool, per month — and pays again per token every time it asks an AI a question. The tools do not talk to each other, the integrations never finish, and the memory of your own work accumulates on someone else's servers.

- **Cost compounds with headcount** — every new person multiplies every subscription.
- **Nothing composes** — twenty vendors, twenty data models, an integration budget that never closes.
- **Your corpus is the product** — retained, mined, and used to improve a model you do not own.
- **Nothing is provable** — "it works" is a claim you cannot re-run for yourself.

02 · THE SHIFT

# When code costs nothing to produce, *value moves.*

Generation is no longer the hard part. Two things become scarce instead — and both are structural, not features:

**Proof**

If anyone can generate a plausible tool, the question is whether it is correct. Reproducible verification becomes the asset.

**Ownership**

If the software itself is cheap, paying rent for it forever stops making sense. What you want is the system, on your hardware.

This is the whole thesis, and the estate was built to it: a trust rail that makes correctness reproducible, and an operating system you own rather than subscribe to.

03 · WHAT FALL-OS IS

# One core. One conductor. *An estate of organs.*

You direct a conductor in plain language. It runs on **your own models**, reasons over **your own corpus**, builds what you ask, and proves it before it ships. Every tool in the estate plugs into the same core as an organ — which is why a hundred separate tools behave as one system.

### The core

One routine every organ calls: expand options, score them against one shared bar, commit deliberately, cache by content.

### The conductor

One loop — explore, resolve, verify, build, remember. Organs are called by it, never run loose.

### The organs

Booking, ledger, legal, memory, agents. Take the ones you need; add more later; remove any of them.

03B · THE UNIFIER

# What *Konomi* actually is.

Konomi is a **self-defining standards compression format** — originating in industrial automation, where the standards are unforgiving. It carries ISA-95, ISA-88, ISA-101, ISA-18.2, OPC-UA, MQTT Sparkplug and Modbus as machine-readable types, with explicit crosswalks between them, so a definition written once is valid everywhere it is used.

Its compositional half is **konomigami** — a fold algebra. Seven base operators, each a pure function from one state and mesh to the next, applied in sequence to a flat plane to produce a structure:

| OPERATOR | FOLD | AUTOMATION LEVEL |
|---|---|---|
| **Ground** | identity / base | L0 · physical |
| **Wave** | valley fold | L1 · sensing |
| **Gate** | mountain fold | L2 · control |
| **Sink** | vertex inward | L3 · operations |
| **Reverse** | direction reversal | L4 · business |
| **Petal** | open and flatten | L5 · enterprise |
| **Collapse** | full collapse | L6 · observer |

Seven operators and six mutations compose into any structure — so the whole stack, from a sensor on a machine to an enterprise ledger, is described in one algebra instead of seven incompatible vocabularies. That is why tools built to it compose without integration work, and why the estate behaves as one system.

04 · YOUR MODELS

# Local first. *Frontier optional.*

Requests descend to the cheapest capable tier. Routine work is answered by deterministic logic or a model on your own machine at no marginal cost. A paid API is reached for only when you allow it — and your key hits your provider directly, with no intermediary.

- **On your hardware** — built-in logic, an in-browser model, then your own local host.
- **Optional and consented** — free-tier and frontier models, under your own account, never required.
- **Structurally private** — a request cannot reach a tier you have not allowed; the remote provider is simply never called.

04B · THE HARDWARE

# It runs on what you *already own.*

There is no minimum spend to start. The lowest tiers need no dedicated hardware at all — and each step up simply raises how much work stays local, never whether the system runs.

### A phone

The tools are single-file and offline-capable. An operator can run bookings, jobs and checklists from a handset in the field, with no signal.

### A £500 laptop

Deterministic logic and an in-browser model cover routine work with nothing installed. This is the honest floor — ordinary hardware, no GPU.

### A desktop or Mac Studio

Add a local host and a mid-size model, and the great majority of day-to-day requests never leave the room or incur a bill.

### A tower rig

Larger local weights for the heavy end — and idle capacity, including hardware written off after crypto, becomes useful again.

The design thesis is that structure beats scale: better memory, better routing and better verification carry more of the load than raw parameter count for the work most businesses actually do. Which tier you need depends on your workload — and you can measure it yourself before spending anything.

05 · MEMORY

# Better memory — *not more retrieval.*

The industry bolts a vector database onto a model and re-fetches raw passages by similarity on every question. It never gets sharper: the same lookup, the same noise, and a permanent bill for storage, embeddings and re-indexing.

The estate places memory by **content**, wires it with **typed relationships**, and **consolidates it while idle** — merging duplicates, materialising links that follow from what is known, generalising rules, resolving contradictions by weight of evidence. The structure improves overnight with no retraining. Recall becomes a short walk through a graph.

Measured, not asserted: the same day of memories in, one idle cycle later, questions answered that could not be answered before — with no parameters touched.

06 · THE DEFENSIBLE ASSET

# Nothing ships on a claim. *Everything ships on a proof.*

A gate alters one operator at a time in the code and requires the test suite to catch it. A green suite that cannot detect a broken version does not count as green. It runs on every push, and anyone can run it themselves.

**witness**

Deterministic mutation and fuzz gate. Packaged so any repository can adopt it.

**acg-assessor**

A deterministic quality rubric — same repository, same verdict, no self-marking.

**Proof-of-play**

Capability demonstrated against a pinned artifact and a canonical grader. A self-graded claim dies on re-grade.

06C · THE CLINIC

# Agents that find their own *weakest point.*

An agent's performance is replayed deterministically from a seed. Seven detectors read that replay and project it onto the agent's capability axes, grounded in the engine's own state transitions rather than labels applied from outside. The worst-scoring axis is the agent's **shadow** — its real weakness, identified from evidence, not self-report.

Then the shadow does work: remediation biases the next generation in proportion to it, and fitness is scored as performance **minus** the weakness that remains. An agent cannot win by being strong on average while carrying a severe blind spot — which is exactly the failure mode that makes autonomy unsafe to deploy.

- **Measured against a blind control** — the therapy arm drove population shadow to 0.18 against the control's 0.04, reproducibly across seeds.
- **With a published phase transition** — below a threshold of population and remediation strength it does not heal at all; above it, it beats the control on every seed. Defaults ship above that line, so the result is cold-reproducible.
- **Memory as compression** — the lifetime journal is minted, not reset: each episode compresses to one lesson. The raw record fades, the correction remains.
- **It closes the loop** — the build gate proves the code is correct; the clinic proves the agent is improving. Structure and behaviour, both verified.

The honest answer to "does it always work?" is no — it works above a stated threshold, and the threshold is published rather than hidden.

06B · GOVERNANCE & SAFETY

# Four gates, and an agent that *cannot overspend.*

Autonomy without constraint is the reason most AI pilots never reach production. Here the constraints are structural — properties of the system, not promises in a policy document.

- **Correct** — a deterministic rubric grades the structure of the code, reproducibly.
- **Behaviourally proven** — the mutation gate requires the tests to catch a broken version before anything ships.
- **Alive** — liveness checks tell a running organ from a flatlined one.
- **Affordable** — every agent's signed identity *is* its capability and its budget ceiling. It cannot spend past its grant, because the grant is what it holds.

On top of that: the conductor never commits on its own — it prepares and proposes, and a person authors the decision. Where a tool touches law or money it is built to **flag what it cannot verify rather than bluff**, and to route to free human help rather than improvise. Contribution and settlement are recorded on a signed, tamper-evident ledger, and settle onto Thomas Frumkin's onlybrains substrate — so what an agent did, and what it earned, is provable rather than asserted.

07 · THE ESTATE

# Not a prototype. *An estate.*

**1,700+**
PUBLIC REPOSITORIES

**371**
LIVE, LINK-VERIFIED

**1**
SHARED CORE

Accounting, legal, insurance and clinic practices; enterprise database, ledger, analytics and low-code; bookings, housekeeping and inbox; agent registries and settlements; memory, mesh and verification. Every one opens in a browser right now.

OPEN THE CATALOGUE

08 · WHAT IT REPLACES

# The software — and *the firm you retain to run it.*

The expensive part of a business is rarely the licence; it is the practice on retainer. These do that work directly, and you own them.

— **Accountancy** — a practice operating system, sole-trader and trades accounting, a double-entry ledger.

— **Legal** — a sovereign paralegal, a statutory redress navigator, cited letters with a guardrail on what cannot be verified.

— **Insurance & clinics** — practice-shaped toolsets, model-agnostic and single-file.

— **Enterprise** — database, analytics, low-code builder and ledger, without the seven-figure platform.

— **Sales & marketing** — a coordinated agent team in one sovereign file.

09 · RENTED VERSUS OWNED

# What actually *changes.*

| | RENTED | OWNED |
|---|---|---|
| **Cost** | Per seat, per tool, per month — forever. | Own what you take. No seats, no renewals. |
| **Inference** | Every request billed per token. | Routine work on your hardware at no marginal cost. |
| **Data** | Uploaded, retained, used to train someone else's model. | Stays on the machine; several kernels have no network capability at all. |
| **Energy** | A datacenter round trip for every question. | Answered on the device already on your desk — hardware considered obsolete is enough. |
| **Collaboration** | A vendor's server in the middle; an outage stops you. | Devices reconcile directly, offline edits merge cleanly — work can pass between machines with no network at all. |
| **Trust** | A marketing claim you cannot re-run. | A reproducible gate you can run yourself. |

Cost, time and energy differences depend on which tools you replace and how you work, so we publish the mechanism rather than a headline percentage. What is structural and checkable today: no per-seat licence, no per-token bill for work a local model handles, no upload of your data, no vendor between your own devices.

10 · HOW YOU BUY IT

# Take only *what you need.*

Not a suite you must swallow whole. Take bookings and invoicing; leave analytics. Add a third tool next year. Because every organ is built on the same core and speaks one contract, they compose instead of collide — no integration project, no middleware tax — and improving the core improves everything you already own.

OPEN FALL-OS | THE CROWN | THE ESTATE

09B · IN PRACTICE

# Thirty systems that *never spoke to each other.*

A large organisation does not lack software. It has a finance stack, a CRM, a document store, reporting tools inherited through acquisitions, and AI pilots sitting *alongside* all of it — because no two systems agree on what a customer or an invoice is. Integration never finishes, and the AI layer cannot reason across the estate.

fall-os attacks that where the problem lives: **one shared contract every system speaks, and one conductor reasoning across all of them.** Existing systems are adapted at the edge, not ripped out — their logic untouched, only their decision-making relocated.

- **Ten-office accountancy firm** — one definition of a client across every office; drafting and review run locally, so client material never leaves the firm. That is the compliance argument, not just the cost one.
- **Multinational group** — regional systems adapted onto one contract, so consolidation becomes a query rather than a quarterly project. Agents carry signed budget ceilings, which is what lets a board approve autonomy at all.
- **Listed company** — a reproducible gate anyone can re-run, a record of what was chosen *and* rejected, and a data-residency answer that is structural rather than contractual.
- **Private mid-market** — adapt the two systems nobody dares migrate off, and stop paying per head for software.

The architecture is identical at every scale, because the merge happens at the decision layer rather than per application. A sole trader runs the same system, smaller.

09C · THE MERGE

# Why 1,700+ repositories are *one system.*

Every build in the estate reduces to the same four moves: offer options, score them against one shared bar, commit deliberately, and remember both the choice and what was passed over. A booking engine, a legal tool, a memory layer and a marketplace differ at the surface and are identical underneath.

So merging is not an integration project. Each build is re-pointed to call the shared core for those four moves and registers with the conductor. Its own logic is untouched — only the decision-making relocates.

- **Improve the bar once** — every organ's judgement improves at the same instant.
- **One surface** — plain language across the estate, not a hundred interfaces.
- **One corpus** — outputs and discarded alternatives consolidate instead of scattering into logs.
- **One standard of proof** — a build that cannot pass the gate does not become an organ.
- **One identity layer** — bounded autonomy and provable contribution estate-wide.

**The merge is at the decision layer, not the feature layer** — which is why every improvement is estate-wide by construction, and the only reason an estate this size is maintainable at all.

09D · HOW THIS GETS USED

# The path, *already visible.*

— **Rent stops.** Three or four subscriptions become owned tools doing the same job on hardware already present. Nothing about the work changes — the software simply stops being a tenancy.

— **The assistant moves inside the work.** Not a chat window to visit, but a conductor drafting against real facts, flagging what it cannot verify, and waiting for a person to commit. Routine questions cost nothing, so you stop rationing them.

— **Memory becomes the moat — and it is yours.** Six months in it answers what it could not at the start, because the structure of what it knows improved. On your machine, so switching cost stops being someone else's lever.

— **Work is delegated, not performed.** Autonomy per organ, bounded by a signed budget — the ninety-ten split, defensible to an auditor rather than taken on faith.

— **What you build is worth something.** Publishable as a tool, SDK, MCP server or API, where a buyer's own test decides its value and royalties travel up the fork tree.

— **Agents transact with agents.** Signed identity, hard budgets, verified work, tamper-evident settlement — the part that was built first.

None of this requires a breakthrough. Every step runs today, in public, with the gate behind it. What is uncertain is not whether it works but how fast the economics force it — and they move one way.

CREDIT

# Standing on *Thomas Frumkin's* work.

The foundations of this estate are Thomas Frumkin's work, and the credit is his. The architecture it is named for, the standard it speaks, the peer-to-peer layer, the guild discipline behind its rubric and the substrate contribution settles onto all originate with him.

— **Forked directly, lineage in git** — *Regulus*, the health engine, forked from teslasolar/regulus and put through the estate's gate. The fork relationship is recorded by the repository itself, not merely claimed here.

— **Built to a published architecture** — *konomigami*, the 紙 Geometric Fold Architecture: seven fold operators over a plane. Implemented as konomigami-lib and the konomi-cube fold engine, each crediting the substrate input in its own README. There was no upstream repository to fork; the architecture was published and implemented.

— **Standards adopted** — the *Konomi Standard* (ISA-95, ISA-88, ISA-101, ISA-18.2, OPC-UA, MQTT Sparkplug, Modbus, with crosswalks) and *KonomiLang*. Adopted as the type and contract layer.

— **Guild lineage** — the assessment discipline of the *AI Craftspeople Guild*, whose initials the estate's deterministic rubric carries.

— **Settlement substrate** — *onlybrains*, where signed contribution is recorded and tracked.

— **Peer to peer** — *kp2p*, Konomi peer-to-peer: devices reconciling directly with no server in the middle. *Fits as:* the browser-native collaboration layer.

— **Recursive orchestration** — the CASSIE line: an orchestrator over specialist subagents, carried into the estate's own multi-agent work.

— **Industrial grounding** — GitPLC, GitHMI, GitSCADA, GitTAG and ISA 5.1 work: the automation context the standard was proven against.

On mechanism, for the record: Regulus carries the fork lineage in the repository itself; konomigami and the rest were published as architecture and standard and implemented here with the substrate input credited in each repository. Different mechanisms, one origin — and where a build began as his work it is forked openly rather than reimplemented quietly.

CLOSE

# Own the system. *Prove the work.*

One core, one conductor, and as many organs as you need — running on hardware you already own, with your corpus as its context and a reproducible gate behind every build.

OPEN FALL-OS | PROSPECTUS | DOWNLOAD DECK (PDF) | AI-NATIVESOLUTIONS.COM