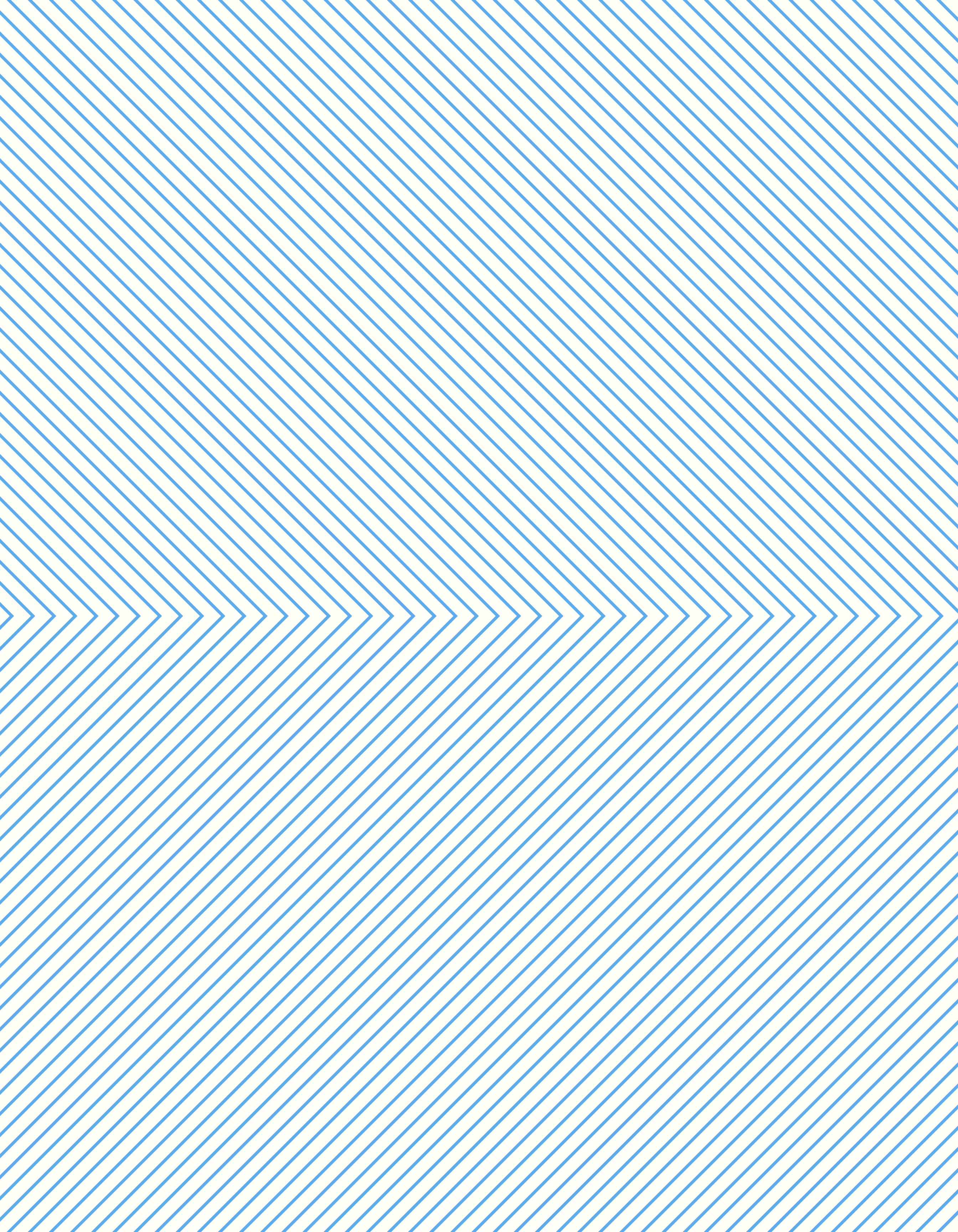
Palantir

How Institutions Make Their Decision Systems Digital, Sovereign, and Compounding

Institutional Sovereignty in the Age of AI



Every Institution Already Has a Decision System

Every institution already has a decision system. Human judgment, software, policies, machines, and processes continuously turn what the institution knows into actions in the world—and validate those decisions through review, correction, escalation, and observation of what happened.

Much of that system remains analog, fragmented across applications, or carried as tradecraft in people's heads. A decision made in one place becomes an email, phone call, spreadsheet update, API request, machine command, or manual entry somewhere else. The institution performs the work and learns from the result, but it cannot reliably connect what was known, how options were evaluated, who was authorized to decide, which action followed, and whether it worked.

Models are a new source of reasoning that can participate in this system. They become valuable when joined to the institution's operating context, decision rights, actions, and existing validation loops.

AI needs validation. The pace and ceiling of AI transformation are bounded by how much trustworthy, consequential validation an institution can capture from its operations and connect to outcomes. The enterprise already produces this evidence through normal work; the opportunity is to make those analog validation loops digital, executable, and capable of compounding.

That requires more than the contractual choice to use another model. It requires the operational ability to preserve the workflow's state, logic, permissions, actions, quality bar, and outcomes while components change. Frontier intelligence remains extraordinarily valuable. Open and adapted models can be valuable too. But the institution's durable advantage lies in its operating context, decisions, exceptions, actions, outcomes, and validation loops—assets that no provider possesses on its behalf.

Own the layers where institutional intelligence compounds; treat the changing components as substitutable.

How the Ontology Changes the Physics of Compounding

Digitizing a decision requires four connected capabilities: data, the state and evidence used to understand it; logic, the policies and computational processes used to evaluate options; action, the governed ability to execute a choice; and security, the decision rights governing who may know, propose, approve, execute, or override. The Ontology binds these capabilities around the institution's operational objects and preserves their lineage through action and outcome:

- → **Objects** represent the customers, orders, referrals, animals, parts, assets, cases, and other entities the business reasons about
- → **Links** represent their operational relationships and dependencies
- → **Properties and States** represent what is known and where work currently stands
- → **Functions and Other Logic Assets** represent the rules, calculations, forecasts, optimizers, and transformations that humans and agents may use
- → **Action Types** represent the governed choices humans and agents may stage or execute, including write-back to operational systems
- → **Permissions and Policies** encode who may know, propose, approve, execute, or override
- → **Decision Lineage and Action History** preserve what was known, what was proposed, who decided, which action followed, and what changed
- → **Outcomes** show whether the decision worked
- → **Workflow and Release Controls** preserve which version of the system produced the result

Together, these elements capture not only enterprise data, but also **decision data**: the context surrounding a decision, the options evaluated, the judgment exercised, and the downstream implications of the committed choice. Decision lineage connects that evidence to the relevant version of data, logic, workflow, actor, action, and outcome.

How the Ontology Changes the Physics of Compounding Cont.

In this way, to borrow an architecture claim from *Attention is All You Need*, we have seen that *Ontology is All You Need*. The Transformer did not contain only attention; it reorganized sequence modeling around attention rather than recurrence or convolution.

The corresponding claim here is that an institution need not encode its full operating knowledge in any model's weights or reconstruct it for every provider. When objects, state, logic, decision rights, actions, and outcomes are represented in the Ontology, each model can perform a bounded role against the same governed operational system. **The Ontology is sufficient as the shared architecture through which the institution's decision system remains coherent as people, software, agents, and models change.**

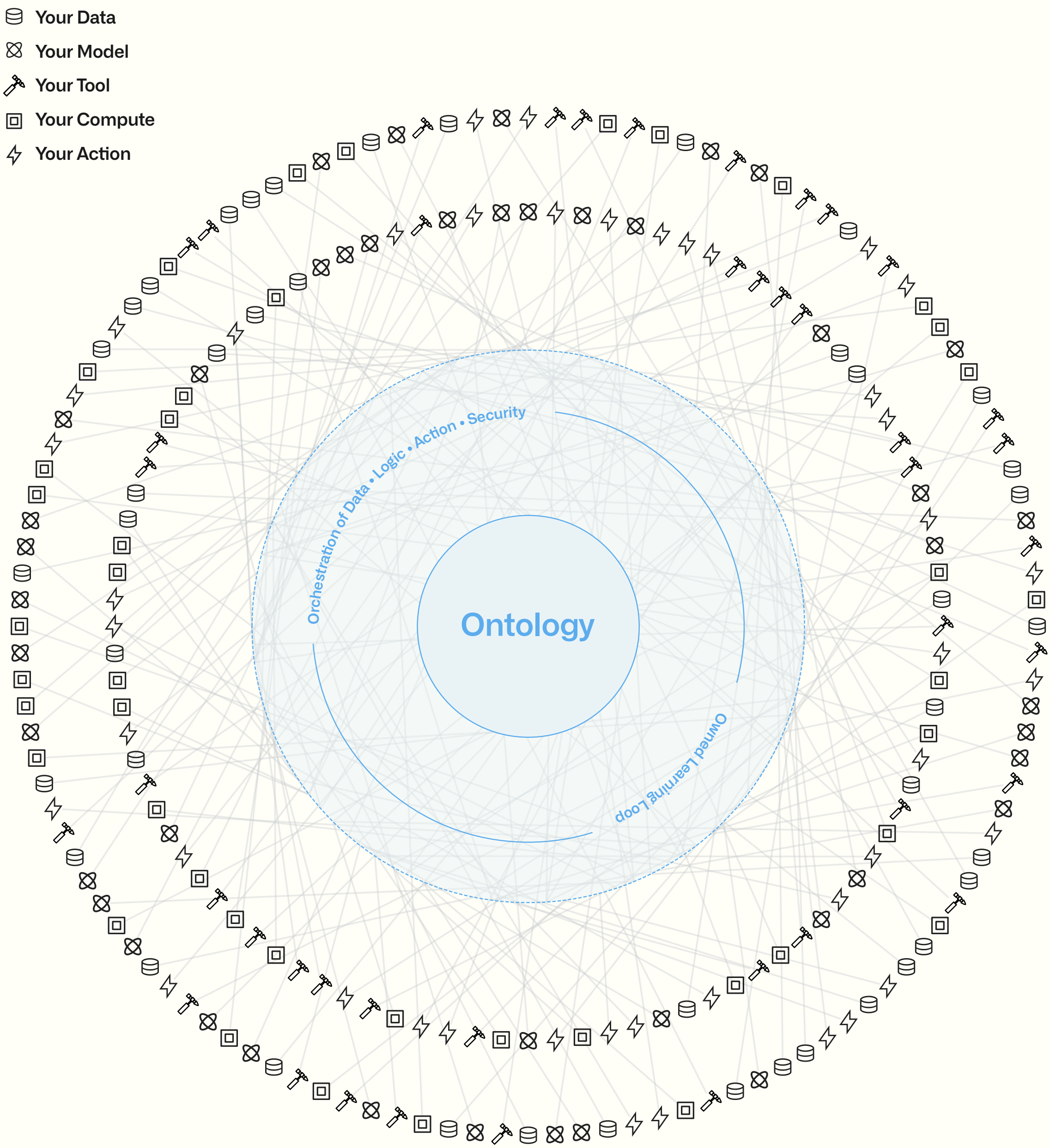
The Ontology does more than provide context to a model. It integrates data, logic, action, and security across the systems in which the institution operates, giving humans and agents a shared operational foundation. It connects inference to a governed decision, the decision to execution, execution to a state change, and the state change to an outcome. Closing that action loop is what distinguishes an operational system from an analytical one.

That connection produces **operational alpha**: the advantage created when the institution's policies, exceptions, tradecraft, and outcomes become reusable by every future workflow.

A task-specific model may become a **precipitate**: a useful artifact produced from the exhaust of the institution's live decision system. The model can be owned, deployed, replaced, and retrained. But the primary asset is the operational vent—the governed workflow that keeps producing new evidence as the institution and its environment change.

The precipitate is valuable. The institution's ability to produce the next model, rule, evaluation, or workflow improvement is more valuable.

Fig. 1 Ontology Is All You Need



Smelt the Workflow, Then Transform It

AI needs validation. A future-state system built apart from real users and operations has no reliable way to prove that it is better than the present. Institutions nevertheless fear that improving today's process will lower the ceiling on tomorrow's. The opposite failure is more dangerous: an exquisite "AI-native" system developed without the users, operational constraints, actions, and outcomes required to validate it.

A proven transformation strategy is to **smelt, then replace, then ship**. Create value inside live work. Digitize its state, logic, decision rights, actions, and outcomes. Replace bounded pieces under the constraint of real-world validation. Keep shipping toward the larger architecture.

In the age of AI, smelting means turning a live, multimodal, cross-system process into governed digital state, actions, and validation loops. It requires the classic work of implementation:

- → Integrate the systems, data, and modalities in which the process occurs
- → Decompose the work into states, decisions, actors, actions, and transitions
- → Make explicit the logic and practices held in people's heads, and bind them alongside deterministic rules, models, optimizers, and simulations
- → Give humans and agents software leverage inside the actual workflow
- → Capture the resulting proposals, decisions, execution, exceptions, and outcomes
- → Govern data, logic, actions, tools, and resulting telemetry with the same decision rights and security policies
- → Validate every replacement against real users and real operational results

This is not a mandate to preserve a weak process forever; it is how the institution builds the **rock-hard foundation for exponential transformation**. The as-is is not the destination. It is the validated surface from which every larger replacement can be tested, improved, and trusted.

The work people choose to perform with scarce human capacity is a strong signal of value. Start there, then use abundant AI labor to raise the ceiling. Decisions made weekly can be reconsidered hourly. Work performed for the top ten percent of customers, parts, or cases can extend to every relevant object. Every customer, order, pallet, machine, or case can have an advocate continually evaluating its current state and the actions available on its behalf.

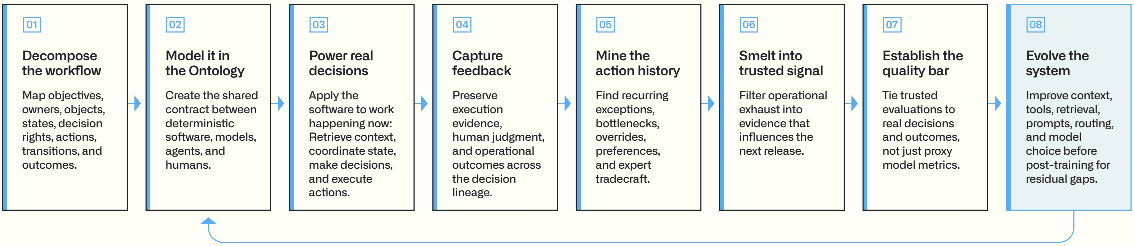
That is not AI plugging a gap at the edge of existing software. It is the institution gradually reorganizing operations around the objects it serves and the outcomes it exists to produce—while remaining grounded in work it can observe, measure, and improve today.

No Skipped Steps: From Workflow to Learning

A model cannot compensate for a workflow the institution has not made legible. Nor can a trace store turn peripheral model activity into evidence about the business.

The sequence matters. Each step expands the amount of consequential validation the institution can generate; skipping one leaves later optimization disconnected from the work and outcomes it is meant to improve.

Fig. 2 No Skipping Steps: From Workflow to Learning



1. Decompose the operational workflow

Identify the objective, owner, objects, states, decision rights, actions, transitions, and downstream outcomes. Locate the places where humans exercise judgment and where work crosses systems or modalities. The unit of design is not the prompt. It is the consequential workflow.

2. Model it in the Ontology

Represent the relevant data as objects, properties, links, and state. Bind the deterministic and probabilistic logic used to evaluate decisions. Define the actions humans and agents may stage or execute, including write-back. Govern all of it with the permissions, policies, and release controls that encode decision rights and preserve outcome links. The Ontology becomes the shared contract between deterministic software, models, agents, and humans.

3. Give the live process software leverage

The workflow must help people do work today. If it is not used to retrieve context, make decisions, execute actions, and coordinate state across systems, its exhaust will not represent the beating heart of the institution. Real use creates the constraints that an isolated future-state design lacks.

No Skipped Steps: From Workflow to Learning Cont.

4. Capture feedback across the workflow

Preserve three different classes of evidence as decision data, connected through end-to-end decision lineage:

- → **System execution evidence:** inputs, outputs, tool calls, retries, errors, model identity, latency, cost, and released workflow version
- → **Human judgment:** accepted, modified, rejected, overridden, or escalated, including the correction and rationale
- → **Operational outcome:** what happened after the decision and action

A trace says what the system attempted. Human judgment says whether someone trusted it. An outcome says whether it worked. None can substitute for the others.

5. Mine the action history

Action logs reveal repeated exceptions, bottlenecks, overrides, preferences, and expert tradecraft. These are the institution's operational secrets: not merely what policy says should happen, but how good operators exercise judgment under real conditions.

Those patterns can become governed context, deterministic rules, evaluation slices, or candidate model behaviors.

6. Smelt activity into trusted signal

Operational exhaust is not automatically a training set. The Ontology helps filter weak signal by connecting a suggestion to the actor who reviewed it, the action that followed, the state that changed, and the eventual outcome.

The institution must still exclude leakage, duplicates, low-trust labels, confounders, stale policy, and activity unrelated to the objective. The goal is not to collect the most traces. It is to determine which evidence deserves to influence the next release.

No Skipped Steps: From Workflow to Learning Cont.

7. Establish an outcome-correlated quality bar

General benchmarks tell the institution where to look. They do not tell it whether a production change is safe or useful.

Begin with “proper noun” cases the workflow owner understands. Measure the stage-level behavior that must remain reliable, but connect it to the end-to-end result the institution actually values. A supply-chain workflow should not merely predict shortages accurately; its evaluation should correlate with idle material, backorders, fulfillment, or another operational consequence. **A model metric is useful only when the institution can show how it predicts or protects a consequential workflow outcome.**

Ten trusted evaluations tied to real decisions can be more valuable than thousands of tests measuring an elegant proxy.

---

8. Evolve the system before adapting the model

Once the institution owns the signal and quality bar, it can improve the full workflow with AIP Evolve: context, semantic retrieval, deterministic logic, tools, prompts, routing, model choice, and human interaction.

This is **system tuning**—improving the whole AI workflow inside the customer boundary. **Model adaptation** or **post-training** refers specifically to changing model behavior through methods such as SFT, LoRA, preference tuning, distillation, or reinforcement learning.

If a bounded, model-specific residual remains after system tuning and model selection, adaptation becomes a measured intervention. The live workflow provides a vent through which permitted, outcome-linked signal can become training data. A new model may precipitate from that exhaust.

Fine-tuning is therefore neither the default nor a novelty. It is one powerful lever the institution has earned the right to use.

Case Study

A Hospital Digitizes the Patient-Access Validation Loop

Every day, thousands of patients request specialist care by sending referrals to a hospital system. If the referring provider is outside the hospital's network, the referral arrives via fax as a PDF. That difference in transport can determine how long the patient waits before the hospital even begins scheduling care.

The hospital employs dozens of FTEs to manually process and ingest tens of thousands of documents into their EHR each day. On a typical weekday, roughly a thousand of these are external referrals. The team prioritizes them, but processing still takes one to two business days.

Entry into the EHR does not complete the workflow. Additional staff perform outreach across several responsibilities, including thousands of outbound calls each day. The process generally operates with another one-to-two-day backlog, and the number of outreach attempts is limited by manpower constraints.

At any given time, tens of thousands of orders may be waiting in scheduling queues. For a new patient referred to a specialist because of a real health concern, the combined process can approach a week before the first successful scheduling interaction. The patient-access problem spans intake, routing, outreach, response, scheduling, and escalation—not document processing alone.

The hospital already operates validation loops throughout this process. Intake specialists verify document interpretation and EHR entry. Encoded urgency determines who must be contacted first. Routing policy determines the appropriate service line. Patient response reveals whether the outreach path worked. Appointment creation reveals whether access was completed. Elapsed time shows whether the service-level objective was met.

Deep implementation makes those analog loops one digital decision system. The significance is not extraction by itself: clinical urgency trapped in a PDF becomes governed state that can prioritize work and trigger downstream scheduling without manual re-entry. The Ontology connects the patient and referral; source document and facts; service line, urgency, provider and routing decision; outreach, scheduling or escalation; elapsed time and final disposition; and the people, policies, workflow and model

Case Study

A Hospital Digitizes the Patient-Access Validation Loop Cont.

The pilot targets referral processing, patient contact and scheduling beginning in approximately fifteen minutes—potentially before the patient leaves the referring provider's office—rather than after several days of intake and outreach queues. The deeper proof point is that the same Ontology can power multiple outreach and scheduling workflows, capture how patients respond, and preserve a human path for the minority of cases requiring logistical or social support. Speed is evidence that the decision system has become operational; the reusable state, actions and feedback are what make it compound.

Daily quality control inside the intake process produced approximately 5,000 human-validated cases. Those cases were created to keep a consequential workflow transition trustworthy, not to manufacture a training dataset. An early experiment adapted a small model on a subset of those reviewed referrals and evaluated it against held-out cases.

Metric	Fine-tuned 1B model	Baseline LLM
Accuracy	90.8%	91.4%
F1	93.9%	93.9%
Referral recall	97.2%	91.2%
Missed referrals	16	50
False positives	55	17

The adapted model approached the incumbent classifier while changing the operational tradeoff: it missed fewer referrals but allowed more non-referrals through. That is not a model leaderboard result. It is a decision for the workflow owner about which error is safer and how the downstream process absorbs it. The threshold, negative examples, serving configuration, and connection to downstream disposition still require owner agreement before deployment.

The experiment also replaced a large rules prompt and generated explanation with a bounded classification contract. Total LLM tokens fell by roughly 46 percent and output tokens by roughly 99 percent. That becomes increasingly important as the workflow expands toward approximately 80 document types and potentially an order of magnitude more volume. Production throughput and latency remain infrastructure questions rather than conclusions from a single model run.

The adapted model is one precipitate from the patient-access system. The compounding asset is the live workflow that continues to generate validated interpretations, channel responses, scheduling outcomes, exceptions, costs and new evidence as the hospital expands.

The Workflow Is the Learning System

The institution's decision system is the compounding asset. A governed workflow makes one part of that system digital and creates value and validation at every stage.

Stage	Capability created	Compounding value
Decompose and integrate the workflow	Shared operational state across systems and modalities	The institution can see and improve the work that matters
Encode state, actions, and decision rights in the Ontology	Governed execution and memory	Business intent survives component changes
Capture judgment, action, and outcome	Auditable operational evidence	The institution learns from real decisions
Mine and filter the action history	Trusted context, rules, eval slices, and training candidates	Tacit tradecraft becomes reusable
Establish outcome-correlated AIP Evals	An owned quality bar	Changes become testable rather than speculative
Use AIP Evolve and system tuning	Better context, logic, prompts, routing, tools, and model choice	Most improvements arrive without retraining
Choose the model and runtime	Hosted, registered, open, or adapted models compared under one contract	Real model liquidity and deployment choice
Adapt a bounded residual	A task-specific model artifact	Specialized behavior where the evidence justifies it

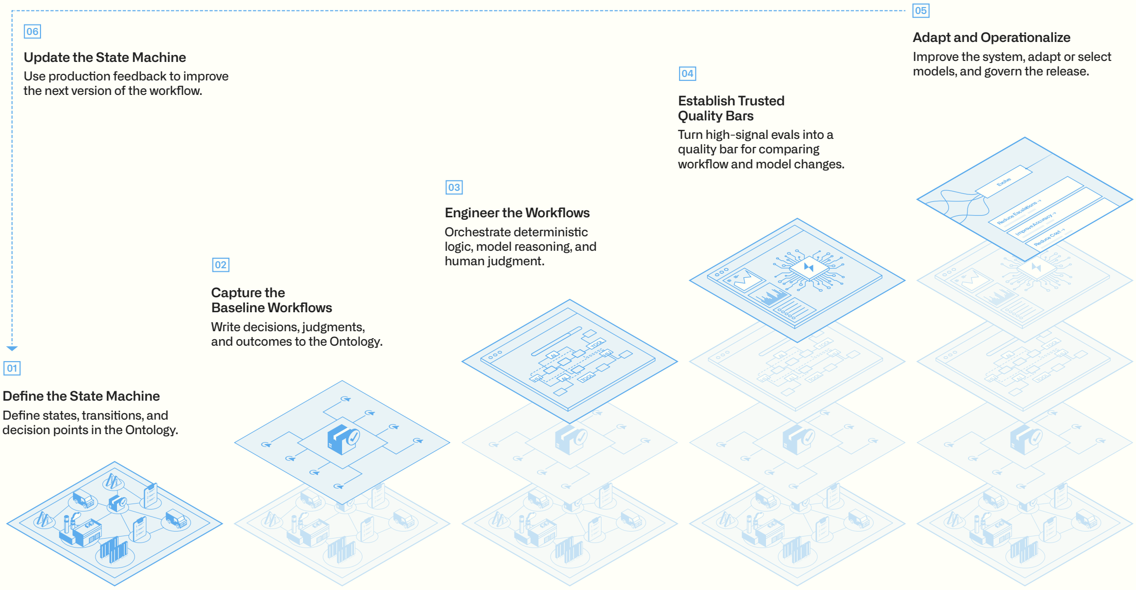
The Workflow Is the Learning System Cont.

A trainable model without a trusted evaluation is not an asset. A large trace corpus without judgment, action, or outcome is not a training set. A fine-tune that compensates for bad retrieval or an unstable tool contract is not durable improvement. A model that cannot be promoted, observed, compared, and rolled back under the same governance as the workflow is not sovereign capability.

There are valid reasons not to train. The use case may not require it. Better context or deterministic logic may close the gap. The data may be insufficient. The economics may not justify the operating burden. Those should be specific, evidence-backed conclusions—not artifacts of missing compute, missing experience, or a belief that prompt engineering must always be enough.

The normal outcome is not that every workflow produces a model. It is that every important workflow becomes observable, measurable, portable, and improvable. The few workflows that still need adaptation become obvious, and everything adaptation requires is already present.

Fig. 3 Building the Learning System



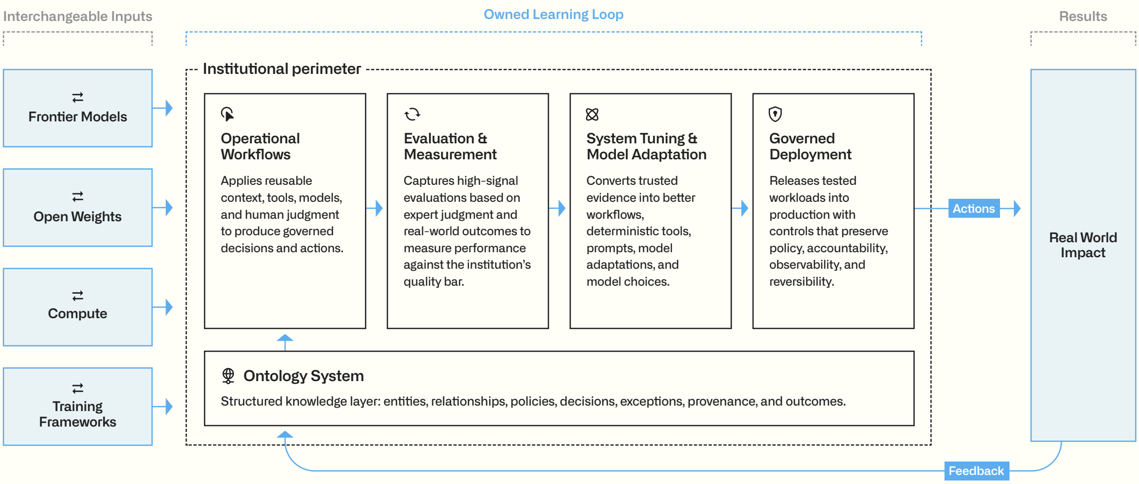
Compounding Inside the Boundary

The enterprise value of this architecture is not a fine-tuned model sitting in a registry. It is the accumulation of decision-making capability the institution can continue to exercise.

A correction made by one expert becomes a test case for the next release. A recurring failure becomes deterministic logic. An accepted exception becomes governed context for another workflow. A downstream outcome changes which examples are considered trustworthy. A model migration can be evaluated against the same quality bar rather than negotiated from memory.

This is operational alpha. It is not generic model intelligence. It is the distinctive advantage created when the institution's policies, exceptions, tradecraft, and outcomes become machine-legible and operationally reusable.

Fig. 4 Compounding Inside the Boundary



Cheap and abundant model intelligence strengthens the thesis. As generation becomes easier to buy, the scarce asset is the governed system that knows what the generation means, what it is allowed to do, whether it worked, and how the result should change the next decision.

The asset belongs to the institution. By connecting its data, logic, decision rights, actions and outcomes, the Ontology changes how quickly and reliably that asset can compound.

Compounding Inside the Boundary Cont.

Models, providers, prices, hardware, and training techniques will continue to change. Sovereignty requires both the choice to change them and the demonstrated ability to preserve the institution's workflow while doing so:

→ Can the institution replace a model without reconstructing its business context?

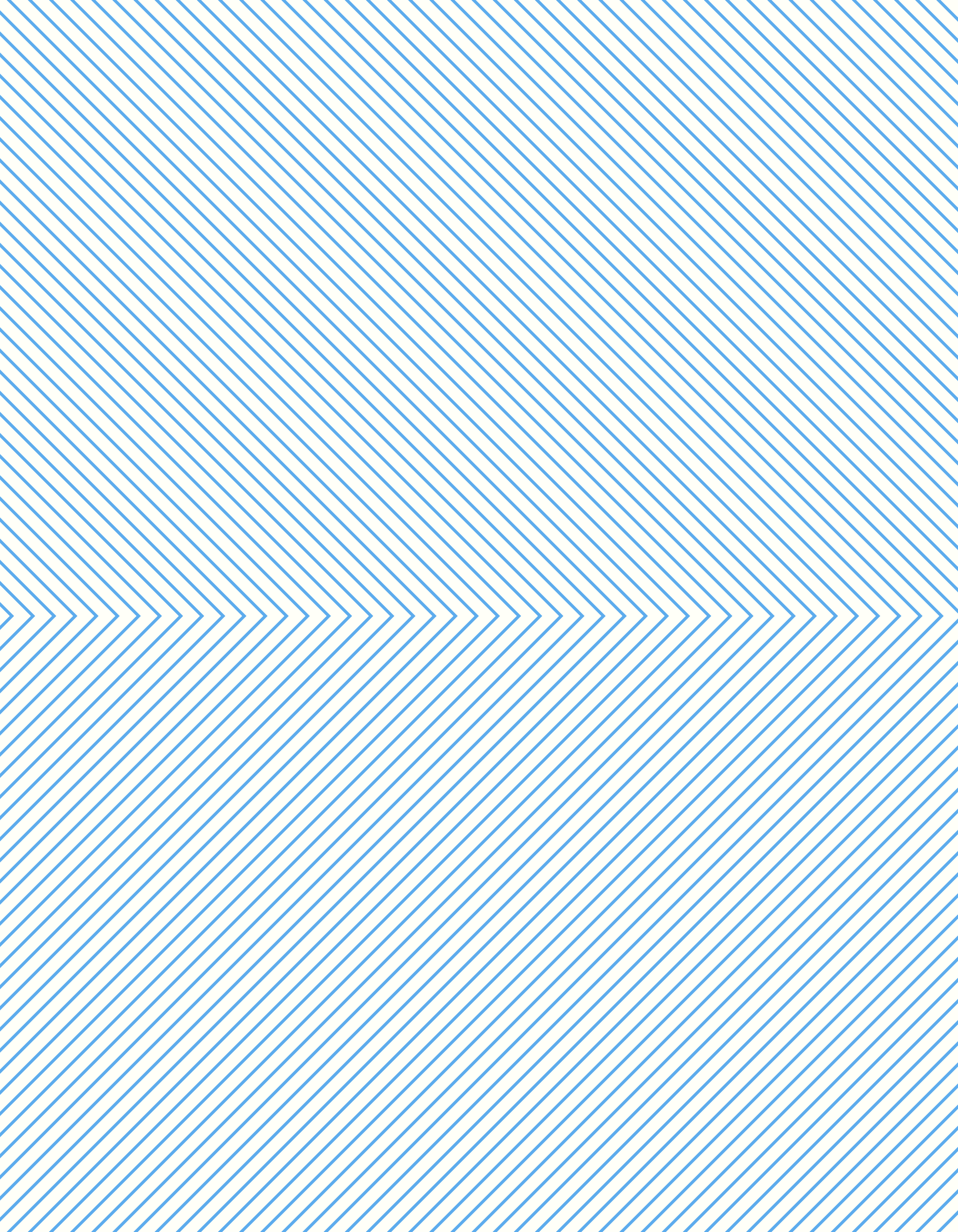
→ Can it preserve decision rights, permissions, and the quality bar?

→ Can it reroute, fail over, or roll back when a component changes?

→ Can it keep operating while the next model is evaluated or adapted?

→ Does the institution own the evidence required to improve again?

Fine-tuning is one possible act inside this system. A fine-tuned model may be a valuable artifact produced by it. **Sovereignty is the institution's practiced ability to preserve and improve its decision-making capability while every technical component changes around it.**



Palantir