SOLVATIVE

A SOLVATIVE WORK BOOK

The Right-Compute Architecture Blueprint

A self-scoring audit for teams building with AI. Sort your initiatives by the layer they actually belong in, score your organization against the four pillars of the Intelligent Systems Lifecycle, and see where the architecture gives out.

Twenty minutes. No tooling required. Work through it alone for a personal read, or with your architecture and operations leads for one that survives argument.

> **Why this exists.** Over the past two years the default move has been to apply a large language model to every problem and hope for favorable outcomes. The organizations getting durable results are doing something less glamorous: choosing the right AI tool for each specific task, and building the architecture that lets those tools work together. This workbook is how we test whether that is happening in a given environment.

Contents

SECTION ONE

The Layer Audit

Most **"AI initiatives"** are named after a technology rather than a task. This section renames them after the task.

A well-structured intelligent system is not one capability. It is distinct layers, each using a different class of AI to do a job it is actually suited for. Machine learning handles pattern recognition, classification, anomaly detection and forecasting over structured operational data. Large language models handle language: understanding, generation, summarization, conversation. Rule engines handle compliance checks, threshold alerts, and any business logic that must not be probabilistic. Specialized inference hardware handles the narrow, stable, latency-critical workloads where waiting is not an option.

Collapse that into "just use the LLM" and you lose both the intelligence and the reliability.

Step 1 — Use the decision matrix

For each AI initiative in your organization — live, in pilot, or on the roadmap — find the row that matches what the task actually does.

IF THE TASK IS...	THE RIGHT MODEL CLASS IS...	TYPICAL COMPUTE TARGET	GOVERNANCE REQUIREMENT
Predicting a numeric or categorical outcome from structured data (default risk, failure date, demand)	Supervised ML; gradient-boosted trees, logistic regression, tabular networks	Cloud training, cloud or on-prem inference	Back-testing, calibration, reproducibility, audit trail
Understanding, generating, or summarizing language	Large language model	Cloud API or hosted open-weight model	Output review, grounding, prompt and version control, PII handling
Enforcing a policy, threshold, or regulation	Rule engine / deterministic logic, not a model	Application runtime	Change control, traceability to the source policy
Detecting anomalies in continuous telemetry	ML; statistical or learned anomaly detection	Edge or stream processing	Drift monitoring, false-positive budget
Sub-millisecond inference on a stable model in a physical environment	Fixed, validated model	Edge accelerator or ASIC	Hardware-tied version control; re-validation on model change

IF THE TASK IS...	THE RIGHT MODEL CLASS IS...	TYPICAL COMPUTE TARGET	GOVERNANCE REQUIREMENT
Explaining a result, drafting the message, answering the follow-up question	LLM, downstream of the model that made the decision	Cloud API	Never let this layer make the decision it is describing

The most common finding. A prediction problem was assigned to an LLM because "AI" was the mandate and LLMs were the visible tool. Predicting whether a borrower repays a loan is supervised machine learning that needs models trained on repayment history, credit data and income trends, producing probabilities you can validate, back-test and audit. The LLM's real job starts after the prediction: writing the explanation, drafting the note to the loan officer, answering the customer's status question. Both tools are correct. The order matters.

Your Initiative Grid

List every initiative currently described internally as "AI." Tag each with the layer it belongs in. Mark the last column when the tag you assigned does not match the technology currently planned or deployed.

INITIATIVE	WHAT THE TASK ACTUALLY DOES	CORRECT MODEL CLASS	CURRENTLY BUILT WITH	MISMATCH?

Tally. Initiatives listed: _____ Mismatched: _____ Mismatch rate: _____%

In the assessments we run, the mismatch rate is typically between 40 and 70 percent, and it is rarely a modeling failure. It is an architecture that never asked the question. If your rate is low, the harder test is Section 2: correct model choices still fail when the data cannot reach them or nobody owns them after launch.

SECTION TWO

The ISL Readiness Scorecard

Twenty questions across the four pillars of the Intelligent Systems Lifecycle. Answer yes only where you could show the evidence to an auditor today.

One point per yes. Be strict as a generous score only produces a comfortable number and no useful action.

Pillar 1 · Strategy & Architecture

Defining the operational architecture intelligent systems need before choosing what to build with.

QUESTION	Y	N
We have a documented target architecture for intelligent systems, not just a list of tools and vendors.	☐	☐
We can state, for each workload, whether it runs in cloud, at the edge, or in a hybrid arrangement and why.	☐	☐
Interoperability between operational technology and enterprise systems is a design requirement, not a later integration project.	☐	☐
Security architecture for AI systems; data boundaries, model access, secrets is defined at the platform level.	☐	☐
A named owner is accountable for the architecture as a whole, not only for individual projects within it.	☐	☐

Pillar 2 · Data & Integration

Connecting operational data across infrastructure and enterprise systems so intelligence has something to work with.

QUESTION	Y	N
Telemetry from equipment, vehicles, devices or sensors reaches our analytical systems without manual export.	☐	☐
ERP, service, and product data are integrated with operational data rather than reconciled after the fact.	☐	☐
Our integrations are event-driven or streaming where latency matters, not batch by default.	☐	☐
Operational data is normalized to shared definitions across systems; an asset, a fault, a customer mean one thing.	☐	☐
We know the lineage of the data feeding each production model and could trace a given prediction back to its inputs.		

Pillar 3 · Intelligence Layer

Deploying decision systems on the appropriate model class and the appropriate compute; right-compute architecture.

QUESTION	Y	N
Every AI initiative has a documented rationale for its model class, tested against the Section 1 matrix.	☐	☐
Prediction problems are solved with models that produce validated, calibrated probabilities.	☐	☐
Deterministic logic; compliance, thresholds, policy is handled by rule engines, not inferred by models.	☐	☐
Latency-critical inference runs close to where the decision is needed, rather than round-tripping to a general-purpose cloud endpoint.	☐	☐
Language models sit downstream of decisions; explaining, drafting, conversing rather than making them.	☐	☐

Pillar 4 · Operationalization & Governance

The lifecycle work that keeps intelligent systems trustworthy after launch.

QUESTION	Y	N
Every production model has a named owner responsible for its performance over time.	☐	☐
We monitor for drift and degradation and would detect it before a user or regulator did.	☐	☐
Model versions, training data and deployment history are tracked and reproducible.	☐	☐
There is a defined gate an AI system must pass before it reaches production, and it has stopped something.	☐	☐
Regulatory and compliance obligations for AI outputs are mapped to specific controls in the system.	☐	☐

What Your Score Means

PILLAR	YOUR SCORE	WEAKEST SIGNAL
1 · Strategy & Architecture	/ 5	
2 · Data & Integration	/ 5	
3 · Intelligence Layer	/ 5	
4 · Operationalization & Governance	/ 5	
Total	**/ 20**	

0–8 · Fragmented

AI exists as projects, not as a system. Individual efforts may be strong, but data is siloed, model choices are driven by availability rather than fit, and nothing governs what happens after launch. The binding constraint is architecture, not talent. Adding more pilots here reliably makes the situation worse.

9–14 · Connected

Data moves and several systems work well. What is usually missing is the governance layer and consistent discipline about model class. This is the most common band, and the most fragile: systems are load-bearing before anyone owns their lifecycle.

15–20 · Governed

You are operating an intelligent platform rather than a portfolio of experiments. The opportunity shifts from remediation to extension; connecting more infrastructure, pushing intelligence closer to the edge, and turning operational capability into digital revenue channels.

Reading the shape, not just the total

The pattern across pillars matters more than the number. Three we see constantly:

- **High on 3, low on 2.** Sophisticated models starved of operational data. The models are not the problem; the integration layer is. This is the single most expensive misdiagnosis in enterprise AI.
- **High on 1 and 2, low on 4.** Well-built systems with no lifecycle owner. Works beautifully for roughly eighteen months, then degrades quietly and is discovered by an auditor, a customer, or an incident.
- **Low on 1, scattered elsewhere.** Capability without a target architecture. Every new initiative costs more than the last because nothing composes.

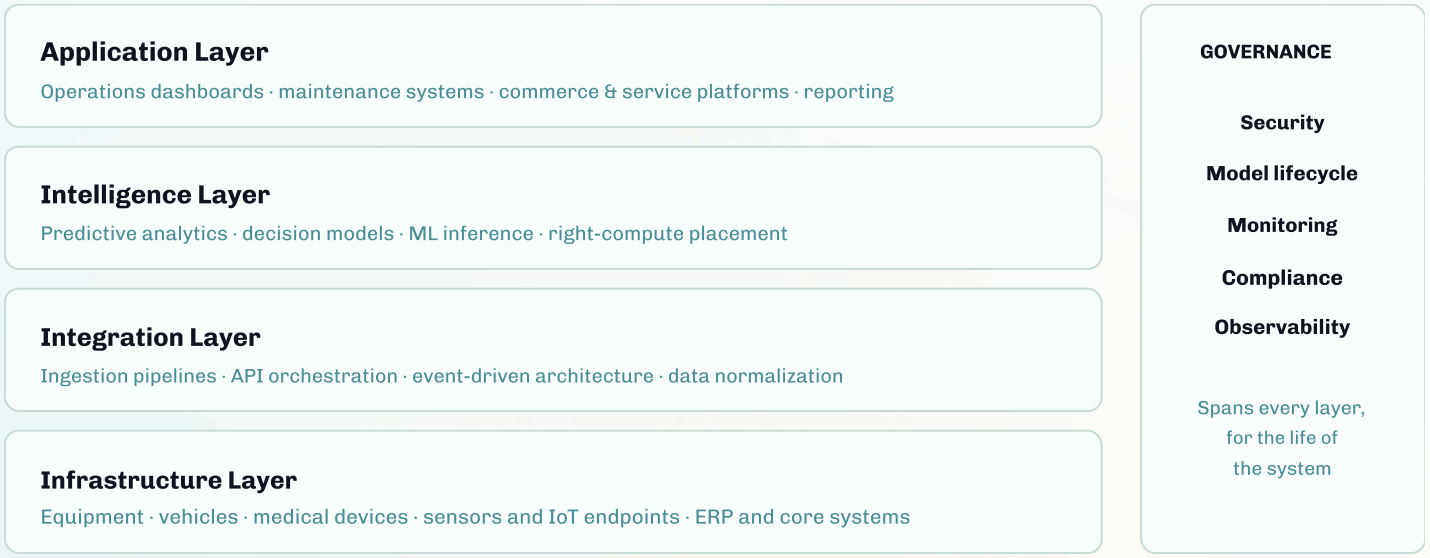
One question worth sitting with. If a model in production began degrading tomorrow, how long until someone noticed, and who would it be? Organizations that cannot answer that in one sentence do not have a governance gap in the abstract; they have an unowned system running the business.

SECTION THREE

The Reference Architecture

What the answers in Section 2 are describing: the five layers of an Intelligent Operational Platform.

An Intelligent Operational Platform is a unified environment connecting operational infrastructure, enterprise systems, intelligent decision systems and digital revenue channels, so an organization can observe, analyze and optimize the business in real time rather than in retrospect. Five layers, governed across their whole lifecycle by the ISL framework.



Read bottom to top for how data moves; read the governance column as the thing that determines whether any of it is still working in three years.

Why the layers are separated

Because each one fails differently. Infrastructure fails by going dark. Integration fails by going stale. Intelligence fails by being confidently wrong. Applications fail by being ignored. Governance fails silently and takes the other four with it. Architectures that treat "AI" as a single box cannot diagnose which of those is happening, which is why remediation so often means rebuilding rather than repairing.

Where your score lands on the diagram

WEAKEST PILLAR	WHAT IT USUALLY MEANS IN THE ARCHITECTURE
Strategy & Architecture	No target state. Layers were assembled by procurement order rather than design, so each addition increases coupling.
Data & Integration	The integration layer is thin or batch-only. Intelligence above it is working from a picture of yesterday.
Intelligence Layer	Model class and compute placement were inherited from whatever was easiest to procure, not from the workload.
Operationalization & Governance	The governance column does not exist as an owned function. Systems are in production without a lifecycle.

SECTION FOUR

Five Anti-Patterns

The failure modes we encounter most. Each one is reasonable at the moment the decision is made, which is why they survive review.

1 · The LLM as Universal Solvent

SYMPTOM A language model is assigned a prediction, scoring or classification task because "AI" was the mandate.

Language models predict tokens from preceding context. That makes them exceptional at understanding and generating language and a poor default for high-stakes tabular prediction, where calibrated probabilities, reproducibility, validation and auditability are the requirement. The fix is rarely to remove the LLM. It is to move it downstream, where explaining the decision is genuinely a language problem.

---

2 · The Ungoverned Pilot

SYMPTOM A successful proof of concept quietly becomes load-bearing without ever passing a production gate.

Pilots are built to demonstrate, not to endure, no drift monitoring, no version control, no named owner. Nothing goes wrong for a year or so. Then the data distribution moves, and the first person to notice is a customer or a regulator. Every pilot needs a defined path either into governed production or into deliberate retirement.

---

3 · The Disconnected Dashboard

SYMPTOM Substantial analytics investment, no measurable change in operational decisions.

Intelligence that terminates in a chart is intelligence that depends on someone happening to look. When the insight does not flow back into a workflow, a work order, a threshold or an alert, the system is describing operations rather than participating in them. Value appears at the point where analysis triggers action.

---

4 · Probabilistic Compliance

SYMPTOM A model is used to determine whether a rule was met.

Compliance checks, thresholds and policy logic must be deterministic, traceable and explainable to an auditor by construction. Inferring them is a category error that surfaces at the worst possible moment. Rule engines are unglamorous and exactly right for this, and pairing one with a model; the rule decides, the model explains and is usually the strongest available design.

---

5 · Premature Silicon

SYMPTOM Specialized inference hardware committed to before the model has stabilized.

Specialized accelerators and ASICs earn their place by removing the overhead between question and answer decisive for edge computing, real-time industrial applications, and anywhere latency is harmful. The tradeoff is flexibility: hardware tied to a model version cannot be upgraded without new hardware. That is a sound trade for stable, production-proven models, and an expensive one for a model still being refined. The discipline is knowing which you have.

SECTION FIVE

What To Do With Your Score

The point of the exercise is a sequence, not a number.

The order that works

1. **Fix mismatches before adding initiatives.** Every initiative in the wrong layer is compounding cost. Re-tagging is cheaper than rebuilding, and often reveals that two projects are actually one.
2. **Close the integration layer next.** Model quality is capped by data reach. Sophisticated intelligence on stale or partial data underperforms modest intelligence on current, complete data every time.
3. **Stand up governance before scale, not after.** Governance retrofitted onto ten production systems costs several times what it costs designed into one platform.
4. **Then extend.** Once the first three hold, connected infrastructure stops being an operational cost and starts supporting service revenue, digital commerce and predictive maintenance offerings.

Talk it through with us

If your score raised questions or the shape across pillars looks uneven in a way you did not expect ; we run a 45-minute architecture review against your completed Blueprint. No preparation beyond the pages you have just filled in.

You leave with a prioritized view of which layer is actually constraining you, and what the first ninety days of work would involve. It is a working session, not a pitch.

About this workbook

Solvative designs and builds Intelligent Operational Platforms, integrated systems that connect infrastructure, operations and revenue channels while embedding intelligence directly into the systems that run the business. Every platform we build is designed and governed through the Intelligent Systems Lifecycle framework described here, across manufacturing and industrial systems, healthcare and medical infrastructure, logistics and fleet operations, agriculture, and industrial distribution and B2B commerce.

Our philosophy has always been architecture before tools. The most successful organizations are not the ones that adopt the newest technologies; they are the ones that choose the right technologies, applied in the right context, for the right reasons.