FROM POC TO PRODUCTION

A Practical Framework for Building Production-Ready Agentic Systems

What it takes to move agentic systems out of pilot and into regulated production — and why 95% never get there.

JULIE PACHECO
VP Client Services

BISHOY YOUSSEF
Senior Solutions Architect

START INSIDE THE PROBLEM

95% of enterprise GenAI pilots never reach production. Not because the model failed.
(MIT 2025)

65% cite agentic complexity as top barrier, 2 quarters running
(KPMG 2025)

44% of orgs stuck at PoC stage
(Avanade 2026)

Most teams aren't stuck because they couldn't build a POC. They're stuck because they did.

The model wasn't the problem. **Everything around it** was — the workflow design, the governance, the evals, the team structure.

integrant

Four questions about your current system.

Question 01/ 04

Was your governance baseline set (Approved Models, Cloud, Logging) before you started building?

integrant

- **Four questions about your current system.**

Question 02/ 04

Did you map the workflow before you selected the model or did the model selection come first?

integrant

Four questions about your current system.

Question 03/ 04

If your system misbehaves tonight, can you reconstruct what it retrieved, decided, and returned?

- **Four questions about your current system.**

Question 04/ 04

When a prompt changes, how do you know behavior got better before users see it?

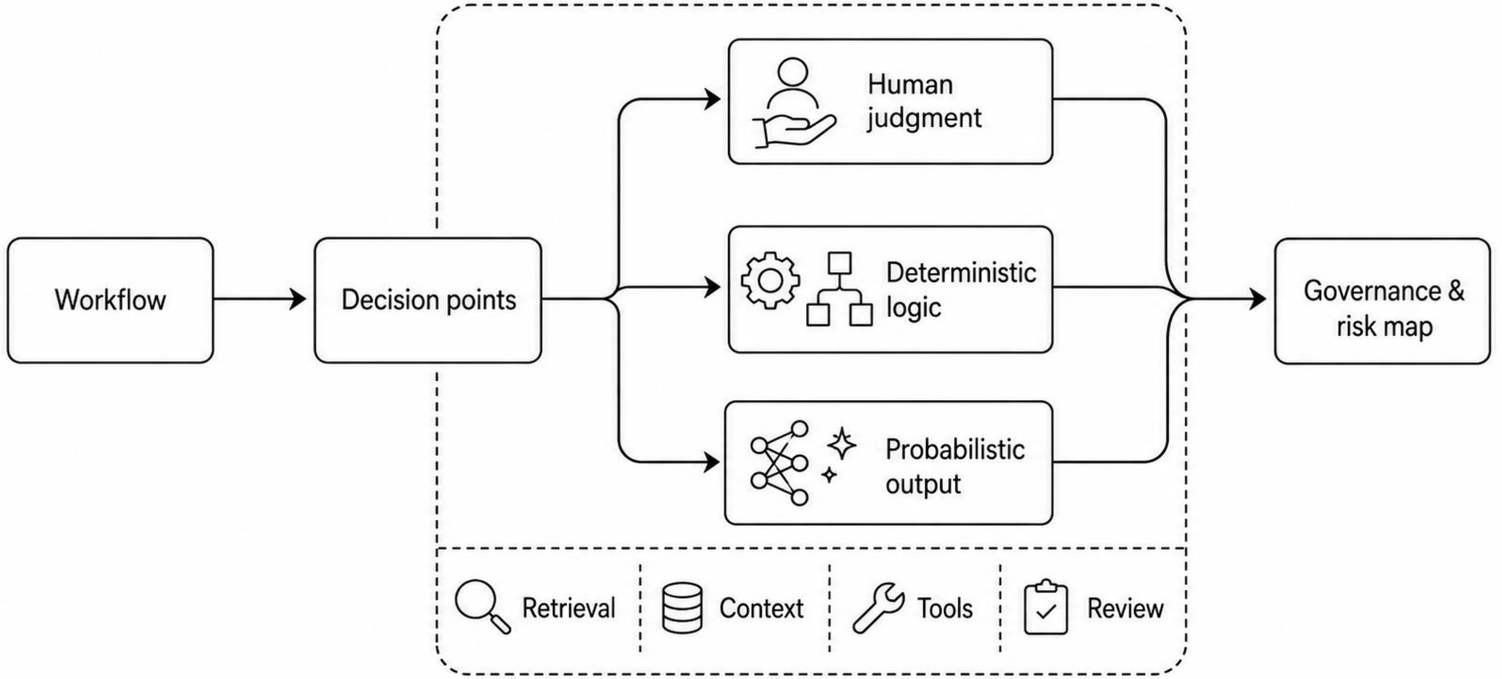
integrant

STAGE 01 · THE FRAMEWORK

01 Workflow Decomposition

Map every decision point in the workflow **before** you select a model.

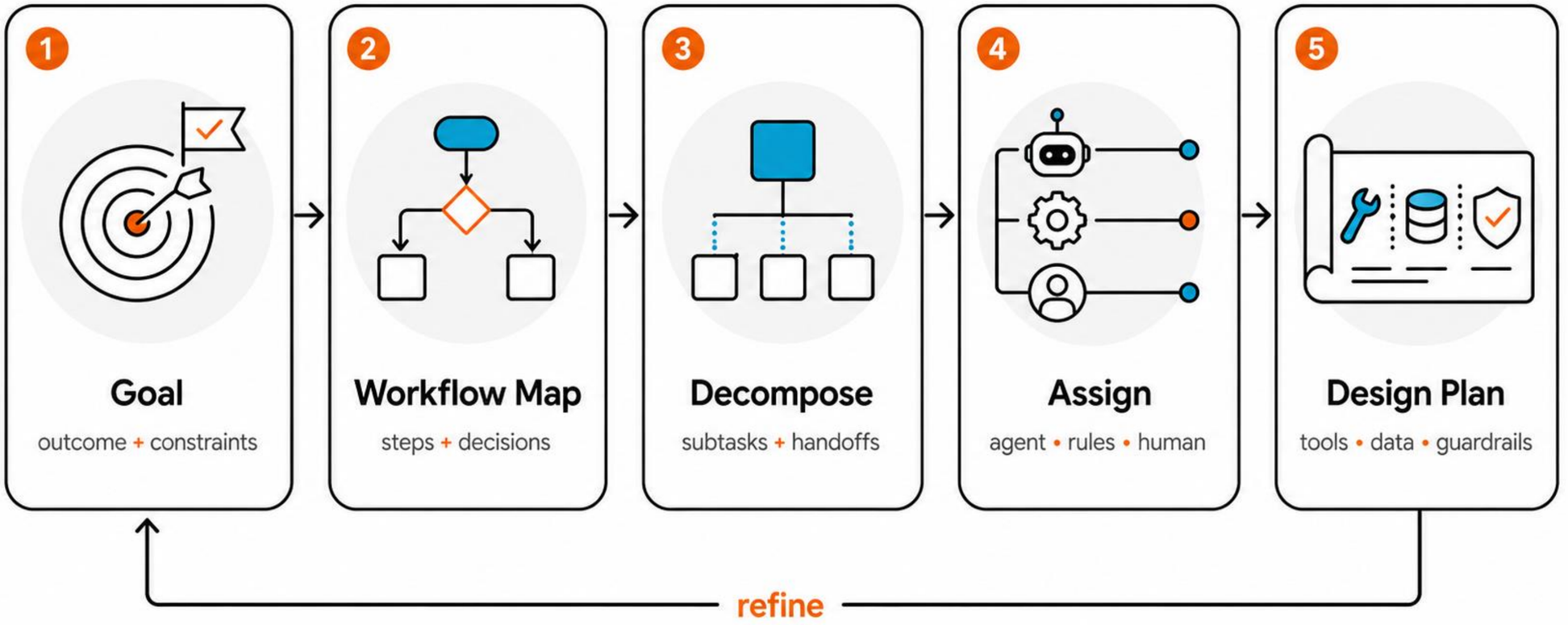
- → Where **human judgment** is required.
- → Where **deterministic logic** is sufficient.
- → Where **probabilistic output** is acceptable — and where it isn't.



Workflow Decomposition:

Agentic Systems Are Not Just "LLM Calls"

Every workflow contains human judgment, deterministic logic, and probabilistic output — map it before selecting a model



Feedback loops & human oversight at key points — designed in from the start, not bolted on after the first incident

STAGE 02 · THE FRAMEWORK

02 Architecture for Ownership

Every decision is explicit, documented, and **owned** — not buried inside a vendor abstraction.

- → **Governance baseline first.** Approved models, cloud, logging — before sprint one.
- → **Audit logging, validation, decision trails** — designed in, not bolted on.
- → **Prompts in source control.** Reviewed, tested, promoted, rollback-able.
- → **Raise your own ceiling.** Rented abstraction = the vendor's roadmap becomes yours.

WHAT OWNERSHIP LOOKS LIKE

When something breaks at 2 a.m., someone in your org can answer the question.

When behavior degrades, you can name **what changed and when** . Without it, prompts drift silently.

ANSWERS QUESTION 01

GOVERNANCE & PLATFORM FOUNDATION

Must be in place before any agentic workflow can be safely deployed

ENVIRONMENTS & ACCESS

- Segregated (Dev / Test / Stage / Prod)
- Data classification & residency
- Least privilege access (RBAC)
- Change control & approvals
- Secrets & key management

DEV → TEST → STAGE → PROD

SOURCE CONTROL & CI/CD

- Version everything — (code, prompts, configs, tools, schemas)
- Pull requests & peer reviews
- Policy as code
- Reproducible builds
- Promotion gates between environments

REPO → CI → TESTS → DEPLOY

OBSERVABILITY & MONITORING

- Traces across agent steps & tools
- Metrics — (latency, cost, success rate)
- Logs with PII redaction
- Alerts & SLOs
- Audit trails — (who did what, when)

EVENTS → COLLECT → STORE → DASHBOARDS & ALERTS

EVALUATION & QUALITY

- Offline evals — datasets, benchmarks
- Online evals — shadow / A/B testing
- Regression & guardrail testing
- Safety, fairness, toxicity checks
- Human-in-the-loop reviews

DATASETS → TEST → REPORTS

GOVERNANCE CONTROLS Policies · Risk & Compliance · Documentation · Approvals · Auditability

Runtime & Integration Layer

Many moving parts to integrate, secure, and operate — every component is an explicit architectural decision, not a platform default

Model Providers

LLM & Embeddings

Azure OpenAI · Claude
Gemini · Mistral
Fine-tuned models

Vector Stores

Semantic retrieval at scale

Azure Cognitive Search
Pinecone · pgvector
Hybrid search

Tools & APIs

Internal & External

Veeva · SharePoint
Smartsheet · Databricks
n8n · REST APIs

Data Sources

Structured & Unstructured

ERP · LIMS · SharePoint
RSS · Web scraping
Regulatory DBs

Memory & State

Short & Long-term context

Session memory
User profiles & history
Conversation context

Guardrails & Queues

Safety & Async Orchestration

Content filters
Async job queues
Rate limiting

EVERY INTEGRATION IS A DECISION — NOT A DEFAULT

· No rip and replace · Connect to existing infrastructure
· Security-scoped per source · Observable across every hop · Owned by your team

integrant

STAGE 03 · THE FRAMEWORK

03 EVALUATIONS & Performance

Measuring output quality & capturing user feedback

→ **Eval suites with regression gates.** “Code runs” ≠ “output in bounds.”

→ **Performance Metrics & Feedback**
Measuring production performance is not optional

→ **Security & Compliance**, a misbehaving agent isn't a post-mortem. It's a compliance event.

A THREAT WORTH KNOWING

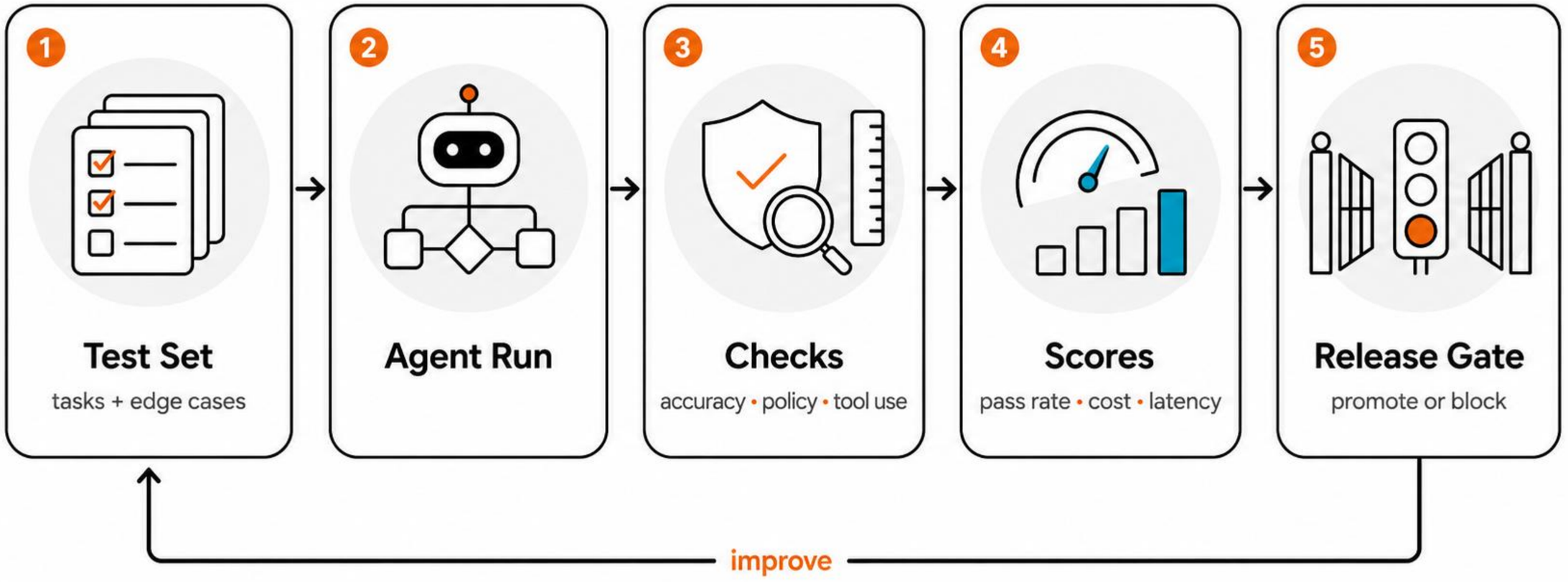
Indirect prompt injection.

An instruction embedded in a document the agent reads during normal operation. One Veeva record is enough. The document looks normal. The agent sees an instruction.

ANSWERS QUESTION 03

Evals in actions

Every change must pass our evals checks and improve the score, and every new feature must have new tests in evals



STAGE 04 · THE FRAMEWORK

04 THE POD THAT SHIPS IT

Production agentic systems require a pod structure — not a **standard** dev team.

→ **Lead Architect**
Owns workflow decomposition and domain context. Governs agent output.

→ **Principal Engineer**
Builds the system. Prompts, retrieval, tools, technical quality.

→ **Eval & Quality Owner**
The role most internal teams are missing. Suites, gates, observability.

WHY THIS MATTERS

Strong engineers stall when no one owns the agentic discipline specifically.

Decomposition, eval, observability — distinct from model integration. It has to be someone's job , not everyone's assumption.

ANSWERS QUESTION 04

Pod Roles:

Different Expertise, Shared Accountability

The agentic engineering discipline is specific and rare — it has to be someone's named job, not a shared assumption

ROLE 1

Architect

Design it right

- Define target architecture & patterns
- Data flows, trust boundaries, and controls
- Model / tool selection strategy
- Governance, risk, and compliance alignment
- Scalability, reliability, and cost guardrails

ROLE 2

Principal Engineer

Build it right

- Platform & workflow implementation
- CI/CD, environment strategy, infrastructure
- Observability, reliability, and performance
- Security hardening & integrations
- Operability & incident readiness

ROLE 3

Eval & Quality Lead

Prove it's good enough, always

- Evaluation strategy & datasets
- Offline / online evals & benchmarking
- Guardrails, safety, and policy testing
- Human evals & red teaming
- Quality gates & continuous monitoring

WHY THIS MATTERS Strong engineers stall without dedicated ownership of the agentic discipline — decomposition, eval, observability must be a named role.

Shared SME Roles:

SMEs Working Across Pods and Teams

Add something here

SHARED ROLE 1

Data Engineer

Source it right

SHARED ROLE 2

Data Scientist

Model it

SHARED ROLE 3

Product Owner

Shape it and build a roadmap

SHARED ROLE 4

DevOps/Infra

Deploy and monitor

SHARED ROLE 5

Metrics

Measure for ROI

WHY THIS MATTERS

The three core roles define how systems are built. These five extend what they're built on. Without them, data quality, user adoption, deployment reliability, and production measurement all become shared assumptions rather than owned responsibilities.

integrant

Case Study

Commercial-stage rare disease biotech. FDA-approved drug. Three more in submission.

→ **Rich library of ideas and use cases.** 100s collected by IT from SMEs.

→ **400+ users.** Four complex agents deployed to hundreds of users across multiple groups including Compliance, R&D, Finance, and others.

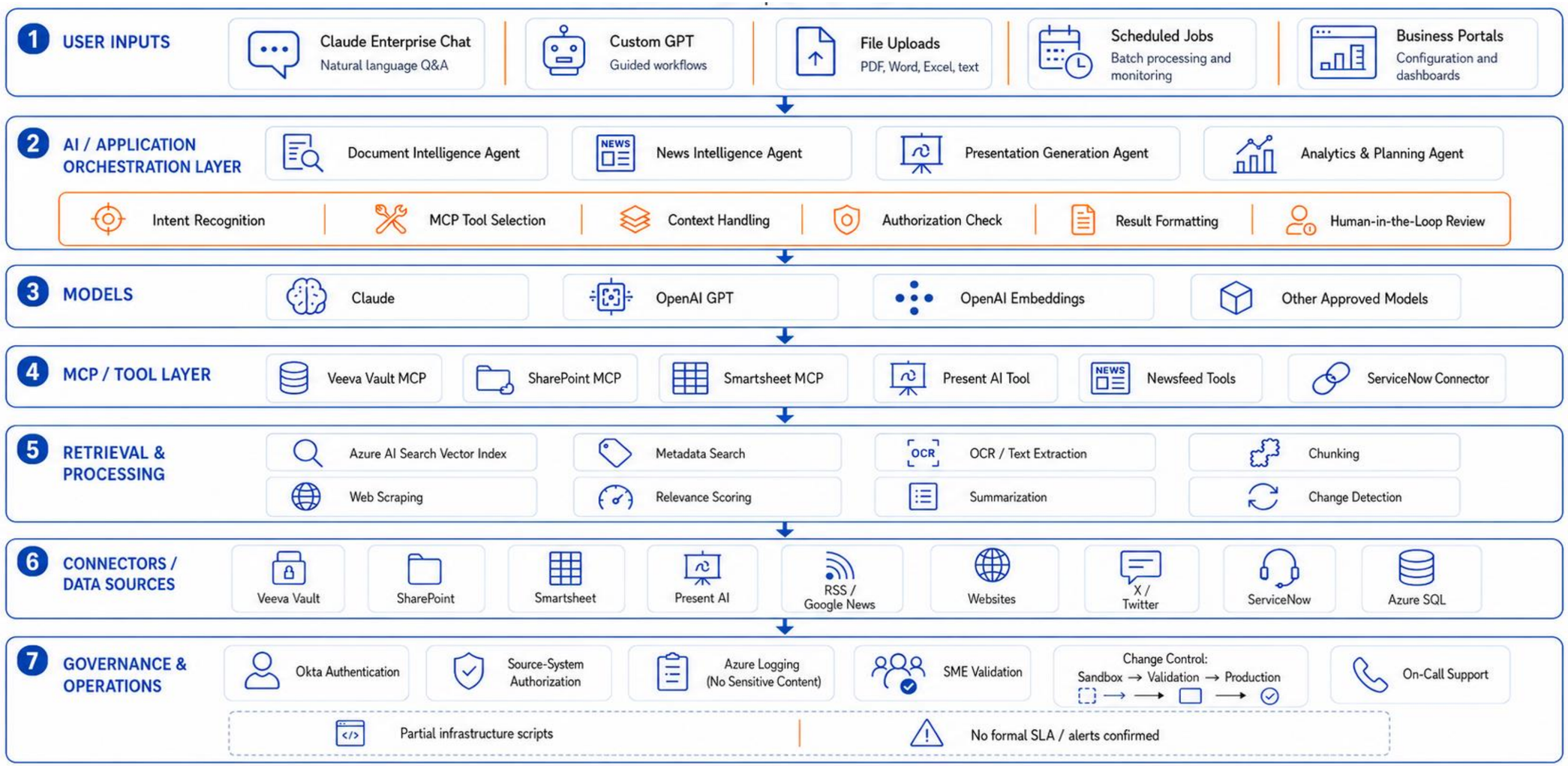
→ **Iterative approach** to prioritizing and adding use cases for the existing agents. Building PoCs for new agents every 4-8 weeks.

Balance of Buy vs. Build

Approach to selecting best-fit technologies, custom building components, and architecting a platform that evolves, grows, and scales.

integrant

Case Study



Why POCs Don't Reach Production

The failure isn't the model — it's what wasn't built around it. Eight failure modes, five principles that close them.

⚠ THE EIGHT FAILURE MODES

- ⚠ Insufficient governance and auditability
- ⚠ Inadequate observability and evals
- ⚠ Unclear ownership and operating model
- ⚠ Workflow logic not decomposed or tested
- ⚠ Security, data, and compliance gaps
- ⚠ No regression testing as things evolve
- ⚠ High operational cost or fragility
- ⚠ Lack of trust in outputs and system behavior

✓ WHAT PRODUCTION REQUIRES INSTEAD

Governance by design
Policies, controls, and audit trails built in from sprint one — not retrofitted after the first incident

Decomposition by intent
Every decision point mapped before a model is selected — human, deterministic, or probabilistic

Evaluation by default
Quality bar defined before first deploy — every prompt change runs through a regression gate

Ownership by role
The agentic discipline has a named owner — not shared across a standard dev pod by default

Trust by evidence
Observable, traceable, auditable — stakeholders Can see what the system did and why

FROM POC TO PRODUCTION REQUIRES:

- Governance by design
- Decomposition by intent
- Evaluation by default
- Ownership by role
- Trust by evidence

integrant

Three questions to answer before you choose a path.

01

Does your workflow need domain-specific judgment?

Then a horizontal platform will abstract exactly the logic that makes it work. **You need architecture you control.**

02

Does your environment need audit trails or regulatory defensibility?

Governance can't be retrofitted. Your engineering approach has to **start** there — not end there.

03

Do you have internal depth to own the agentic discipline?

Not just model integration. The question isn't *build or buy* — it's whether your partner has a **repeatable practice**.

integrant

THE TAKEAWAY

The teams getting agentic systems into regulated production aren't the ones who found the best platform. **They treated this as what it is — an architectural discipline — and built accordingly.**

FIND US AT THE CONFERENCE.

info@integrant.com | **www.integrant.com** | **BOOTH #308**