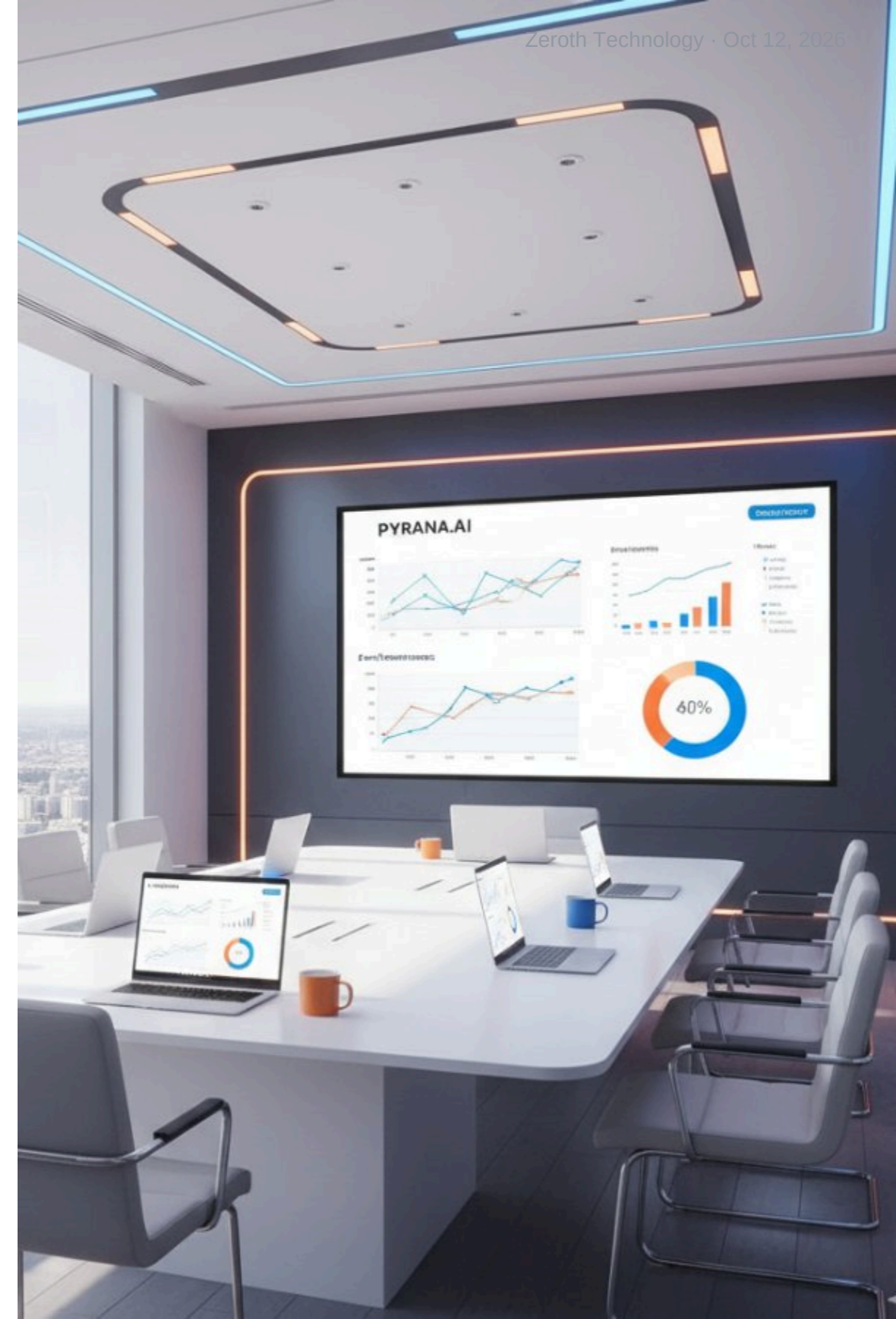


PYRANA® Overview

Agentic applications, not agents.

October 2026



Introducing PYRANA

Production-grade AI applications, not chat boxes. Rapidly built around your processes, informed by your knowledge, controlled by the platform.

Outcome driven

Applications, not a text box

Chat makes every user responsible for knowing what to ask. Agentic applications are shaped around the task and bring the work to the person, so results depend on the process, not on who is prompting.

Compounding Scale

Every application accelerates the next

Take one use case from pilot to production, extend it across teams and sites, then add the next. The foundation and the business knowledge are shared, so every application builds on the last.

Risk Mitigating

Guardrails outside the model

Permissions, approvals, and limits are enforced by the platform, not by a prompt. Agents inherit the access and authority of the people they work for, and every action has an audit-ready history.

Introducing PYRANA

Powerful, AI-powered enterprise applications you can trust

THREE KINDS OF SOFTWARE, AND WHERE PYRANA SITS



Systems of record

Software that keeps track. People do all the work.



Copilots and chat tools

Software that helps one person at a time. The result depends on that person's skill and on what they thought to ask.



Agentic applications: PYRANA

Software that carries the work. Agents do the analysis inside an application built around the process; people decide inside it.

HOW IT WORKS: EVERY APPLICATION IS CONFIGURATION ON ONE PLATFORM

Month-end close

Contract review

Order exceptions

+ the next one

SHARED ONCE, REUSED BY EVERY APPLICATION

Business knowledge

System connections

Identity and permissions

Approval rules

Audit record

PYRANA HARNESS: ONE RUNTIME FOR EVERY AGENT RUN

Durable

Saves each step. Resumes after a crash or a long approval wait.

Any model

Chosen per application. Your keys, your data, no training.

Governed

Identity, approvals, limits, and the record enforced on every call.

WHAT MAKES IT DIFFERENT



Outcome driven

Applications, not a text box. Shaped around the task, so results depend on the process, not on who is prompting.



Compounding scale

Every application accelerates the next.

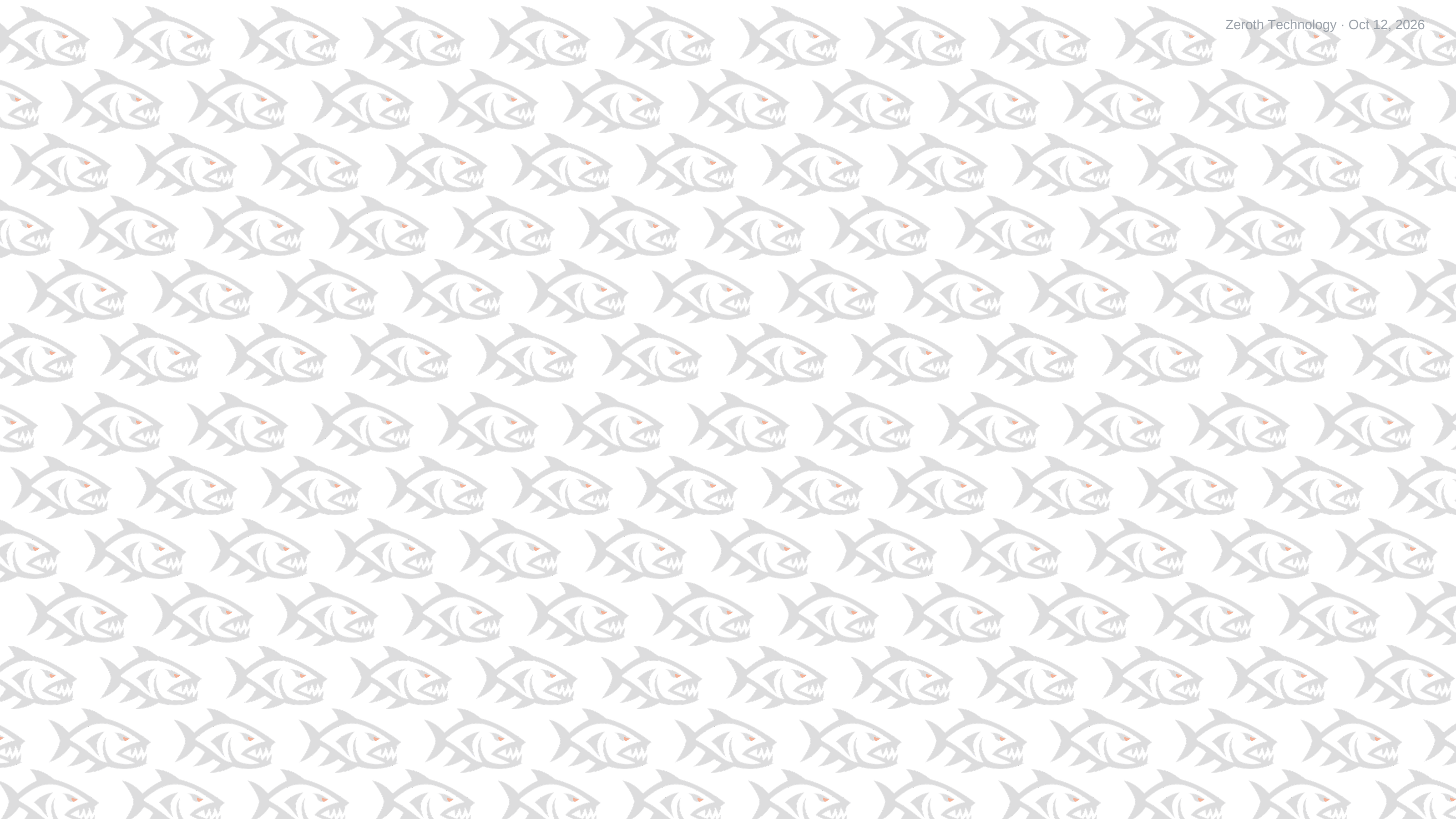
Foundation and business knowledge are shared, so each one builds on the last.



Risk mitigating

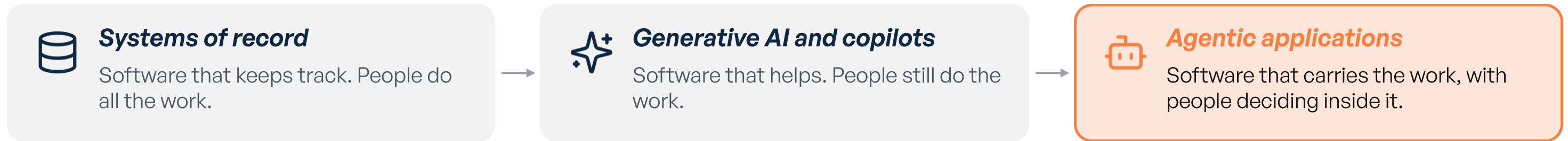
Guardrails outside the model. The platform enforces permissions, approvals, and limits; every action has an audit-ready history.

The hard part is already built. Each application is configuration on top: the process, the pages, and the knowledge, not the plumbing.

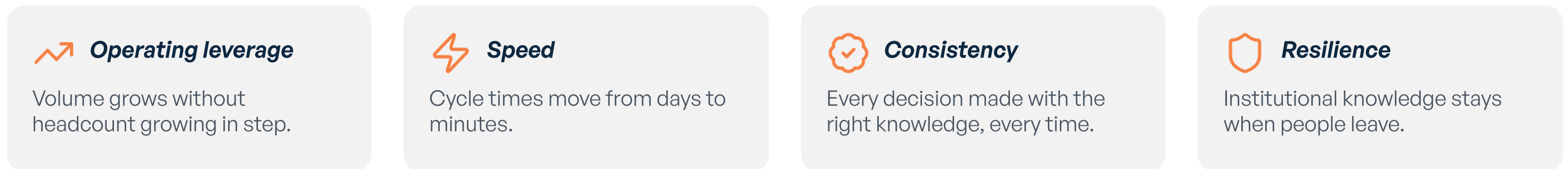


Why Agentic AI, and Why Now

When software does the work, the economics of every function change



WHAT IT CHANGES FOR THE BUSINESS



Why now: the models are ready. What separates the winners is how safely, and how fast, they put them to work inside the business. Early movers are already compounding the advantage.

What an Agentic Application Is

People and agents share the same pages, data, and knowledge, under one identity and one record

YOUR BUSINESS people, systems, knowledge, customers

THE APPLICATION the job to be done and the rules around it

who may do
what

what needs
approval

which limits
apply

the record of
every action



THE AGENT a participant, not the application

Reads the documents and systems, works out what to do, acts within the limits, asks a person when the policy says so, and cites what it used.

It never sets its own rules. It holds only the access the application grants, including its own credentials to outside systems, and every action is on the record.



What agents do

- Gather the facts from documents and systems
- Analyze and recommend
- Act within the limits you set
- Ask a person when the rules say so
- Remember, and improve with use



What your people do

- Set the rules and the limits
- Approve what matters
- Handle exceptions and judgment calls
- Spend time on customers, strategy, and relationships

Everyone, human or agent, passes the same gate: same identity, same authorization, same record.



Working in the background

Agents run the process end to end and bring people only the exceptions.

Example: the monthly financial review across every business unit, delivered in minutes instead of days.



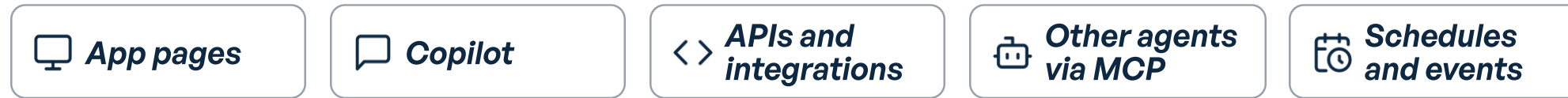
Working in the application's pages

People work in the application's own pages. The copilot sits beside the page for questions about what is on screen and for personal pages; it is not the primary interface.

Example: a buyer completing a procurement request on a page, with the policy applied as they go.

Meet PYRANA

Build, run, and govern agentic applications on one platform, private tenant or your own infrastructure



THE GATE every call, human or agent, passes eight steps before anything runs: identity, App version, authorization, approval policy. Fails closed. The record is written before dispatch.

AGENTIC APPLICATIONS

One App configuration each: surfaces, targets, tool sources, resources, access, knowledge. Examples below.

Financial close and review

Collections

Procurement

Marketing content

IT project management



Pyrana Harness

Runtime and orchestration: planning, tool use, subagents, durable workflows, approval policies, checkpoint and resume. Five lanes: direct, agent, workflow, batch, event. A model assigned per agent.



cortIQ context engine

Documents and tacit knowledge become typed, classified, evidence-backed Context Units in a knowledge graph. Agents retrieve and cite them; every run accounts for what it used.

ENTERPRISE FOUNDATION SaaS on a dedicated private tenant, or your own infrastructure · SSO and authorization in layers · tenant isolation · one execution record per call · your data never trains external models · model per agent

WHAT MAKES IT DIFFERENT



One authorization model for people and agents

Identity, authorization, approvals, and audit enforced once, on every door. Each agent carries its own identity and credentials, inside and out.



Context engineering, built in

A shared context engine with source-level provenance. Knowledge compounds across apps with use.



Applications, not agents

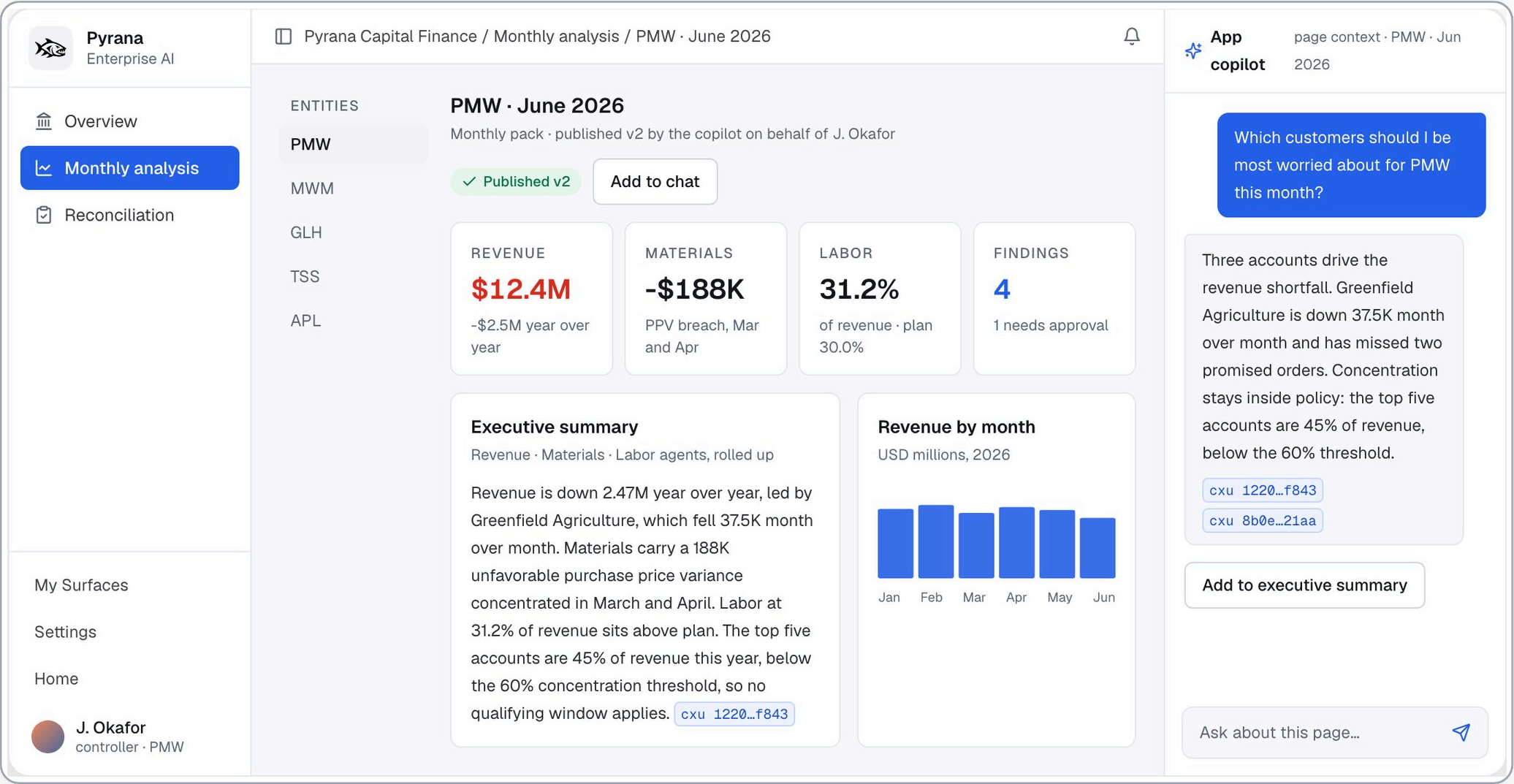
The basics are already built and live in the platform. Teams build the process, the pages, and the knowledge, not the plumbing.

The platform is built once. Each application is configuration, and value arrives in weeks.



What a Team Sees

Elegant, user-friendly application pages. The copilot sits beside the page; it is not the primary interface.



WHAT THE PAGE SHOWS

Pages people work in

Entities, KPIs, an executive summary, and a chart on one App page. Not a chat window.

Copilot beside the page

Auxiliary: questions about what is on screen, and personal pages. The page stays primary.

Every figure cites its unit

Numbers carry Context Unit references. The run that produced them is on the record.

ILLUSTRATIVE the monthly run published the pack; the controller asks the copilot about one account and revises the summary in place. Pyrana Capital is a synthetic portfolio.

Why PYRANA

Because the hard part is already built



A framework

You get an agent. Security, authorization, and storage are still yours to build, and to maintain as agents multiply.



Vertical applications

One finished product per job, each with its own identity, permissions, and data. Nothing between them. CIOs do not want twelve of them.



A platform: PYRANA

Identity, authorization, storage, orchestration, approvals, and audit built once. Every application composes on it as configuration, with people inside the loop.

SPEED TO VALUE



The basics are already built

Identity, authorization, audit, storage, orchestration, approvals, and context live in the platform. A team builds the process, pages, and knowledge.



About a week

to stand up the App configuration, with clean data available. The frontend experience is the main effort.



About six weeks

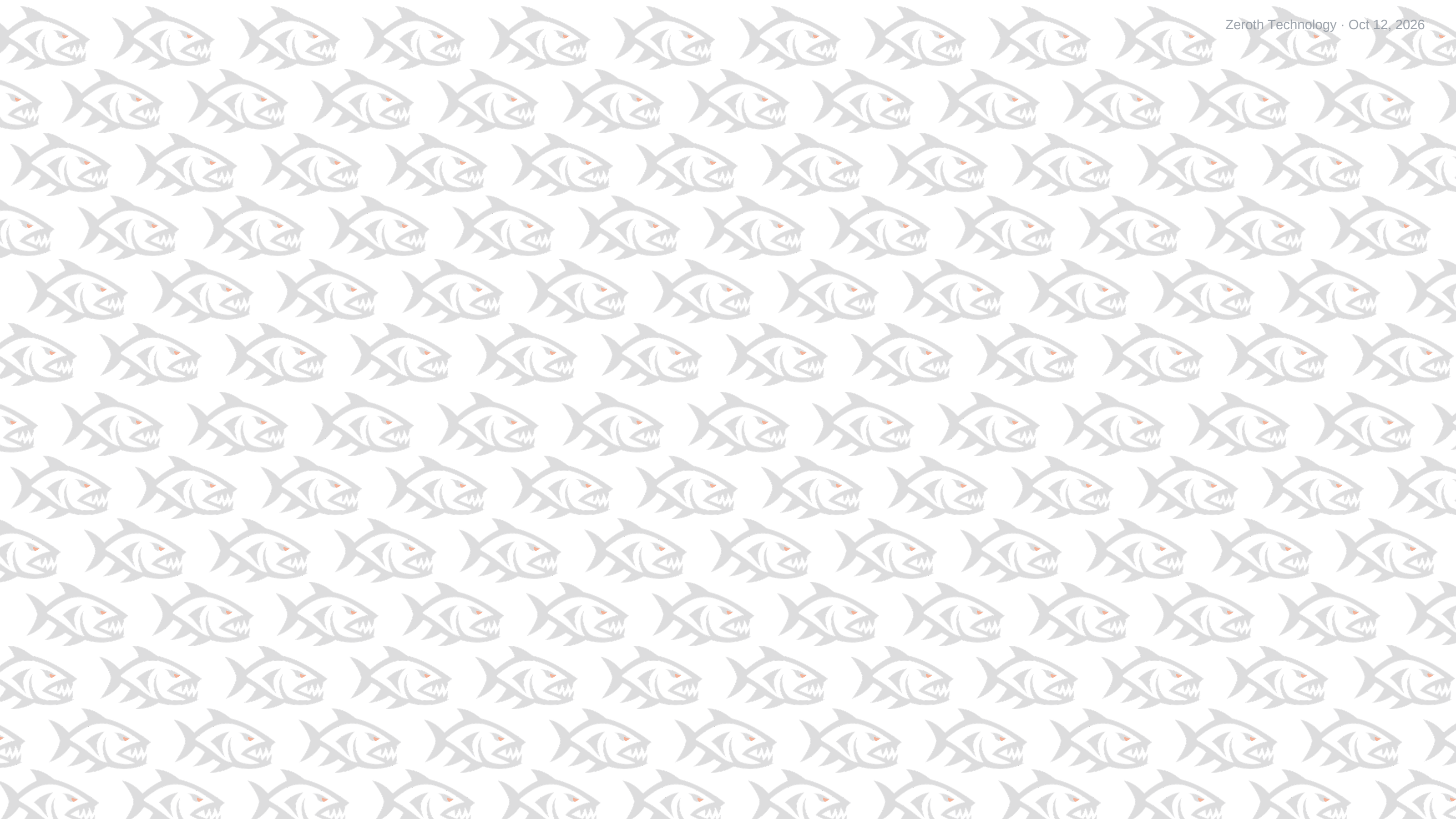
from decision to a first App in production, with real identities and approvals. Confirmed per engagement.



Less each time

Every App after the first inherits the identity model, access profiles, approval lanes, audit record, and Context Units.

Built and delivered by experienced operators, in production with enterprise clients in regulated industries today.



Pilots Are Easy. Production Is the Problem.

Frameworks and vertical applications both leave the hard part undone

VERTICAL APPLICATIONS

One vertical AI product per function. Each arrives with its own identity model, permission semantics, audit format, and storage.



Finance AI

- ⛔ own identity
- ⛔ own permissions
- ⛔ own audit log
- ⛔ own storage



Procurement AI

- ⛔ own identity
- ⛔ own permissions
- ⛔ own audit log
- ⛔ own storage



IT Ops AI

- ⛔ own identity
- ⛔ own permissions
- ⛔ own audit log
- ⛔ own storage

Context stays trapped in each silo. Integration glue is rebuilt for every pair of systems and governed by none of them.

A FRAMEWORK

Agent frameworks and cloud agent services solve the visible part: a model, a prompt, and some tools.

 **The agent**

what the demo shows

below the waterline

Fine-grained authorization

Tenant isolation

Durable execution

Human approval policy

Idempotent external effects

Audit, secrets, versioning

The hard part sits underneath every application. It is expensive, security critical, and usually bolted on after the first incident.

40%+

of agentic AI projects will be canceled by the end of 2027, citing cost, unclear value, and inadequate risk controls

Gartner, 2025

50%

of enterprise AI agents operate in isolation: no shared data, no coordination, no handoffs between them

Salesforce, 2026

21%

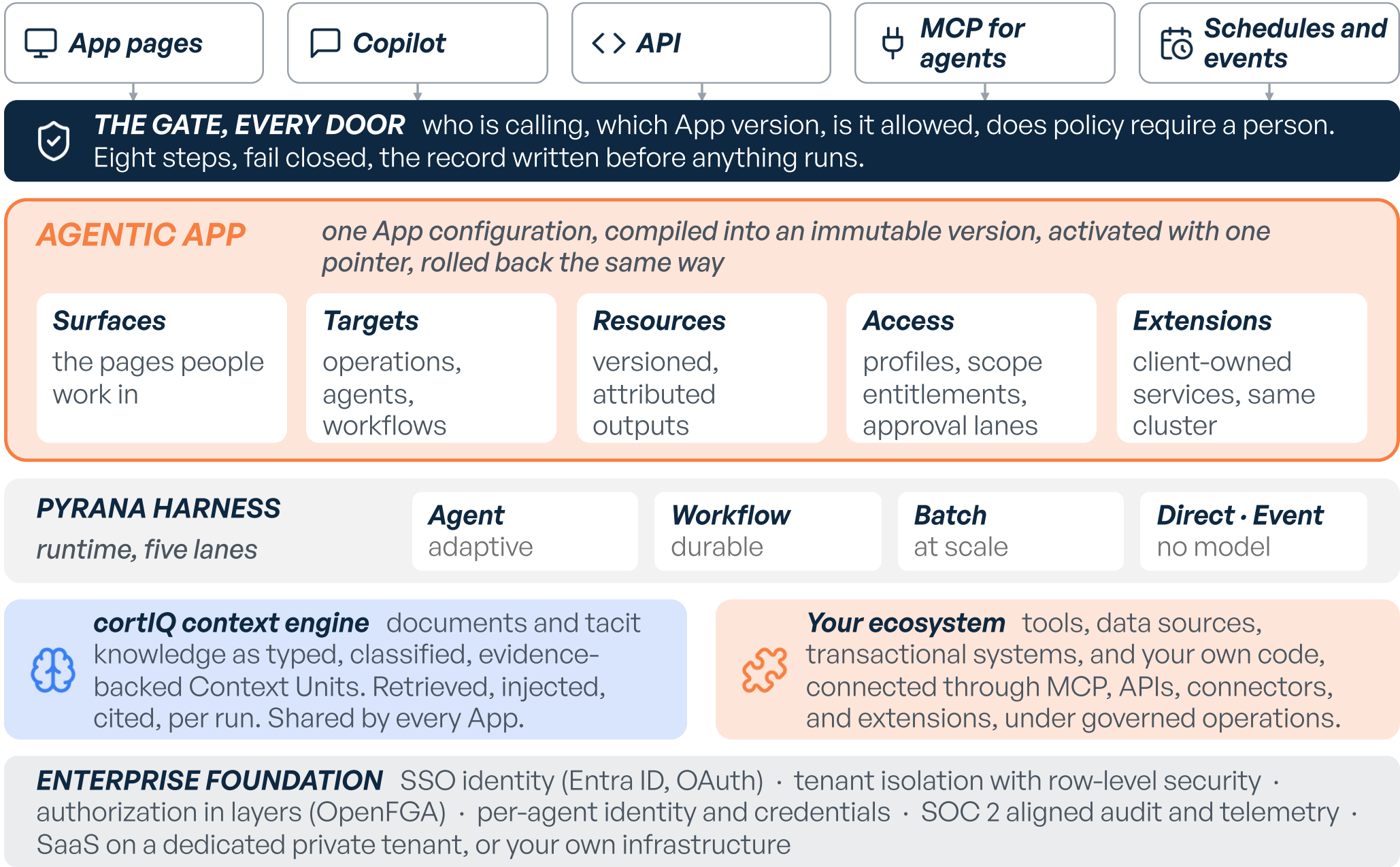
of organizations have a mature governance model for agentic AI

Deloitte, April 2026

The third way: a platform that does the hard part once. Applications as configuration, agents where they help, people deciding inside.

PYRANA: The Agentic Application Platform

Build governed enterprise applications where agents are participants, not the product



One identity.
One authorization model.
One audit trail.
One shared context.

- Faster**
The basics are already built. New applications are configuration. Value in weeks.
- Safer**
Governance is built once and enforced on every call, human or agent.
- Compounding**
Context, outputs, and learning are shared across Apps.

SaaS on a dedicated private tenant, or deployed into your infrastructure. In production with enterprise clients today.



Agentic Apps: One Application, Many Doors

One set of rules for every person, agent, workflow, and system in the application



Why not assemble it from components?

COMPONENT BUILD

Permissions

× Every door grows its own lock; policy drifts.

Agent authority

× Agents call any tool they can see.

Outputs

× Trapped in the run or thread that made them.

Your domain code

× Forks the platform or leaks into shared code.

Change control

× Config edits go live unversioned.

AGENTIC APP

✓ One target, one gate, every door.

✓ Agents hold their own identity, inside and out.

✓ Versioned, citable resources with provenance.

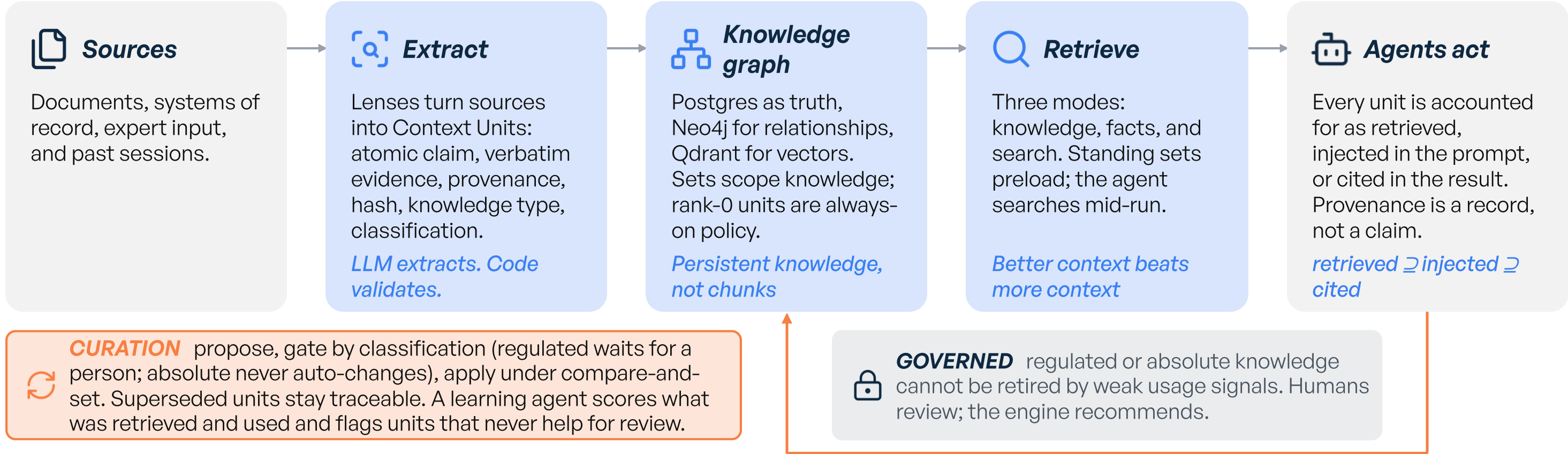
✓ Runs as a governed extension in your cluster.

✓ Compile, version, activate, roll back.

Agents are participants in the app, not the app itself.

cortlQ: Company Knowledge, Typed and Cited

Documents and tacit knowledge become Context Units: typed, classified, evidence-backed



BEYOND RETRIEVAL	Typical RAG or agent memory	cortlQ
Unit of knowledge	Text chunks sized for the embedding model	Atomic claims with evidence, provenance, and a citable hash
Retrieval	Similarity search over one index	Graph, vector, keyword fused and reranked; policy always on
Provenance	A log of documents retrieved	Retrieved, injected, and cited tracked per unit, per run
Improvement	Static index; re-embed to change behavior	Learns from use under governance; protected knowledge stays
Memory	Conversation transcript replay	Session memory over temporal, semantic, and reference links

An Agent Harness Built for Enterprise Work

A proprietary, model-agnostic runtime with durability and human oversight built in



Native tool calling and subagents

Plans, calls tools, delegates bounded work to named subagents, and reads results through scoped views.



Durable execution

Temporal-backed workflows and dynamic DAGs. Suspend, checkpoint, and resume after any interruption.



Human approval as policy

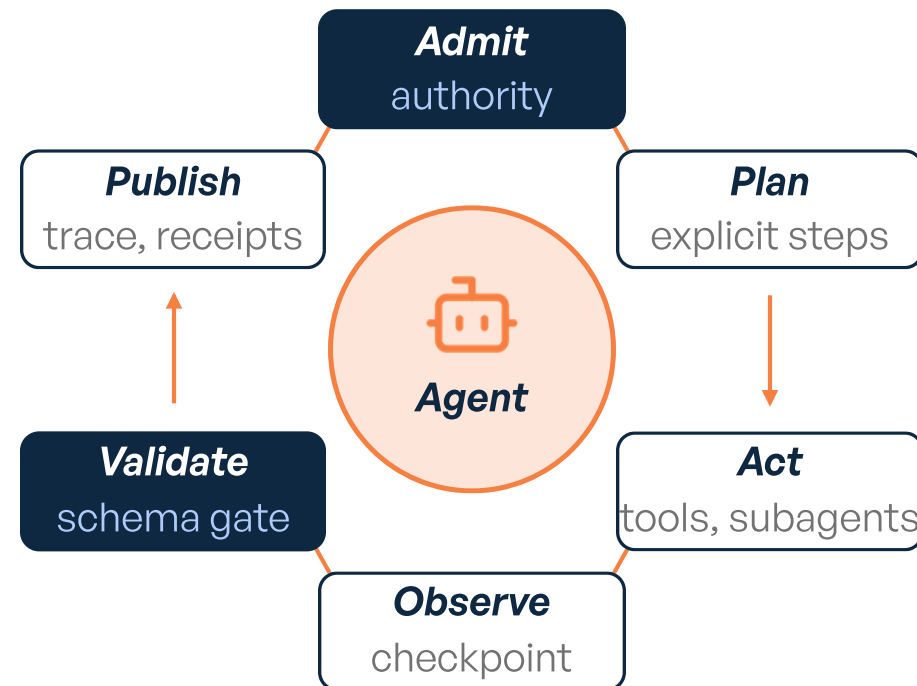
Typed conditions, named approval lanes, frozen invocations, revalidation before dispatch. Never pays twice.



Model agnostic

A model is assigned per agent through provider adapters and never switched mid-run. Bring your own keys.

ONE GOVERNED LOOP



Every agent has its own identity, in the platform and in the systems it touches: one agent may hold SharePoint site A, another only site B. Every child call re-enters admission.



Does the job, every time

Deterministic checks confirm the required output, structure, and evidence before an agent can finish.



Plugs into your ecosystem

Your tools, data, and transactional systems via MCP, APIs, connectors. Your code as an extension.



Explicit state and virtual file system

Plan and knowledge change only through tools. Path-based access to files and context, no raw credentials.



One causal trace

Admission, model calls, tools, approvals, effects, outputs: one identity. Evaluate and audit any run.

FIVE LANES behind one gate. Direct runs deterministic work with no model; the four below run the rest.

Agent adaptive, with tools

Workflow durable, long running

Batch many entities at once

Event schedules, webhooks

Comparing Agentic Solution Approaches

All three get one agent live. They diverge on every application after it.

APPROACH	WHAT ARRIVES	EVERY APPLICATION AFTER THE FIRST	SECURITY AND AUTHORIZATION	CONTEXT AND KNOWLEDGE	CONTROL AND LOCK-IN
 A framework Agent SDKs and cloud agent stacks: AgentCore, Foundry, Vertex, LangChain	An agent. Security, authorization, and storage are still yours to build.	Rebuilt per agent, or a platform team maintaining what the framework left out.	The enterprise's to build, and to maintain as agents multiply.	Per-agent memory. Cross-app knowledge is another build.	Full control, full burden. Primitives tied to one cloud.
 Vertical applications One product per job: Sierra, Harvey, Jasper, Wonderful	One finished application per job, each with its own identity, permissions, and data.	Another product, another identity model, another audit format.	Each vendor's own. Nothing between them.	Trapped inside each product. Re-onboarded for every vendor.	Vendor cloud, vendor data model, vendor's choice of models.
 A platform PYRANA	The foundation, built once: identity, authorization, storage, orchestration, audit.	One App configuration. Same identity, authorization, audit, and context. Basics never redone.	One model, decided on the backend in layers, shared by every application.	cortIQ shared and cited across every app, learning from use.	Your tenant or your infrastructure, your data, your tools, any supported model.

Agent control planes (Agent 365, Agent Fabric, AI Control Tower) govern agents under any approach and build no applications.

The foundational engineering is done once, in the platform. That is what value in weeks depends on.

Questions Buyers Ask

The short answers. The site carries the long ones.



Hosting

SaaS on a dedicated private tenant, or deployment into your own infrastructure for large enterprises. Temporal workers provide durability and horizontal scaling in both.



Portability and lock-in

One YAML configuration per application; agent instructions in plain language; declarative workflows; Context Units are your data. Nothing is opaque.



Data layer

No copy of your data is required. Agents query live through tool sources: REST, ODBC, MCP, Databricks, Snowflake, Drive, OneDrive. Cleansing runs in a code interpreter or an ETL layer upstream.

MORE



Models

A model is assigned per agent through provider adapters and never switched mid-run. Bring your own keys.



Evals

Every run's traces, tokens, and tool errors are captured as JSON today. Deterministic checks first, then LLM-as-judge.



API

A full administrative API: push and pull configuration, start runs, read runs and traces. Any agent can work with an App over its MCP server, as the user.



Time to build

About a week for the App configuration with clean data; about six weeks from decision to a first App in production, confirmed per engagement.

Security review: questionnaires answered with mechanisms and documents, on request.