Rhino
Federated Computing

PRESENTATION INSIGHTS

Collaborative Edge Computing A Rising AI Tide Lifts All Boats

Adrish Sannyasi
VP, Customer Solutions & Delivery

The Paradoxes of 2026

Three contradictions defining the next phase of enterprise AI.

01 For the first time, AI can reason over the data that actually drives outcomes.

Most enterprises cannot use it.

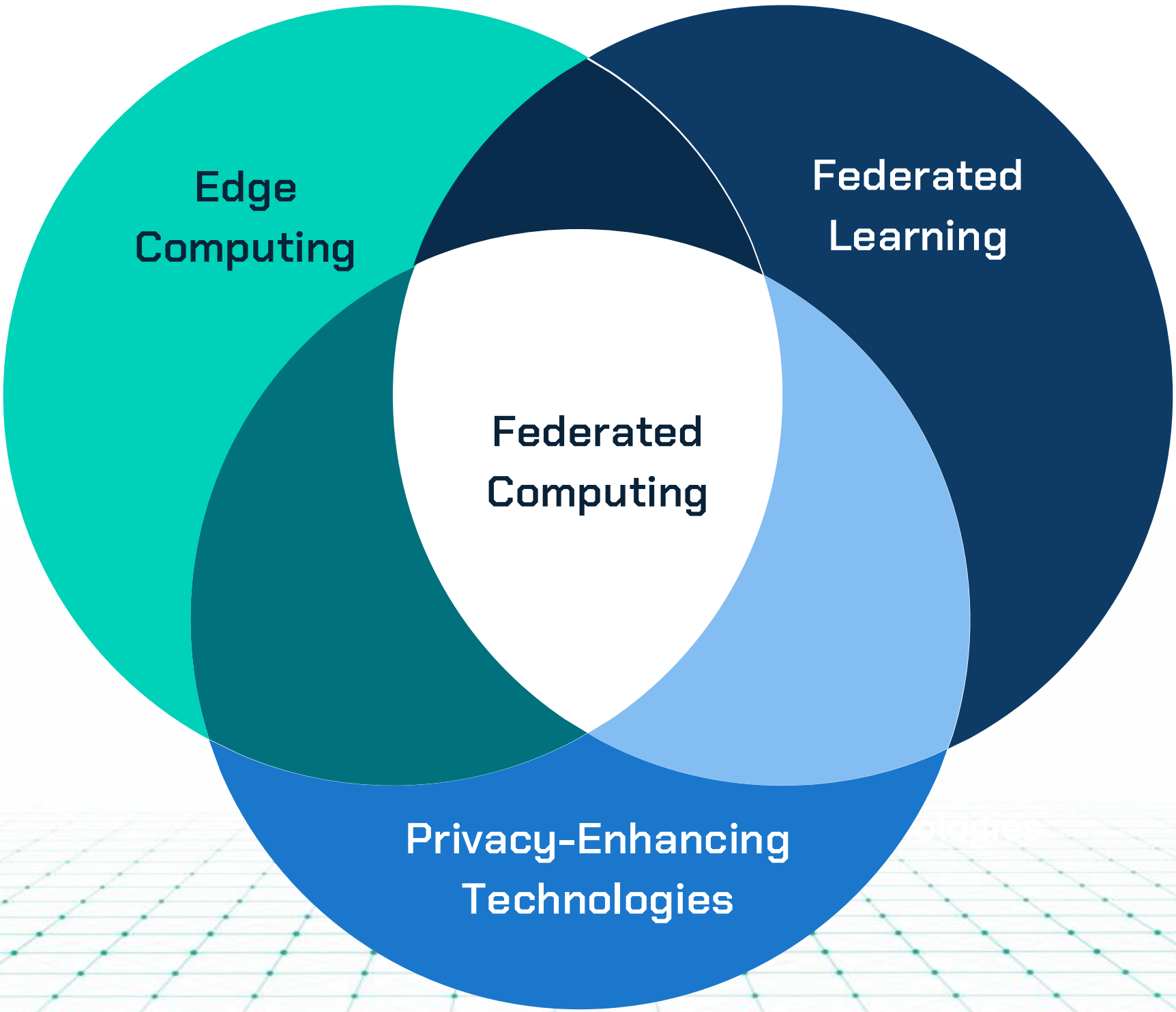
02 Every major enterprise is investing in AI. The technology has genuinely arrived.

AI maturity is no longer the constraint.

03 Model improvement requires learning across deployments — but telemetry, process data, and production context stay on-site/at edge.

It was never going to the central data lake.

A federated approach has a potential to overcome the barriers



Illustrative Edge Execution Flow

1. Query sent by User/Agent to AI client - Parse intents and initiate the workflow in the orchestrator.

2. Orchestrator sends instructions to edge clients.

3. Provision infrastructure and execute the tasks at edges.

4. Intermediate results are created at edges and aggregated centrally.

5. Insights shared with user through the AI Client

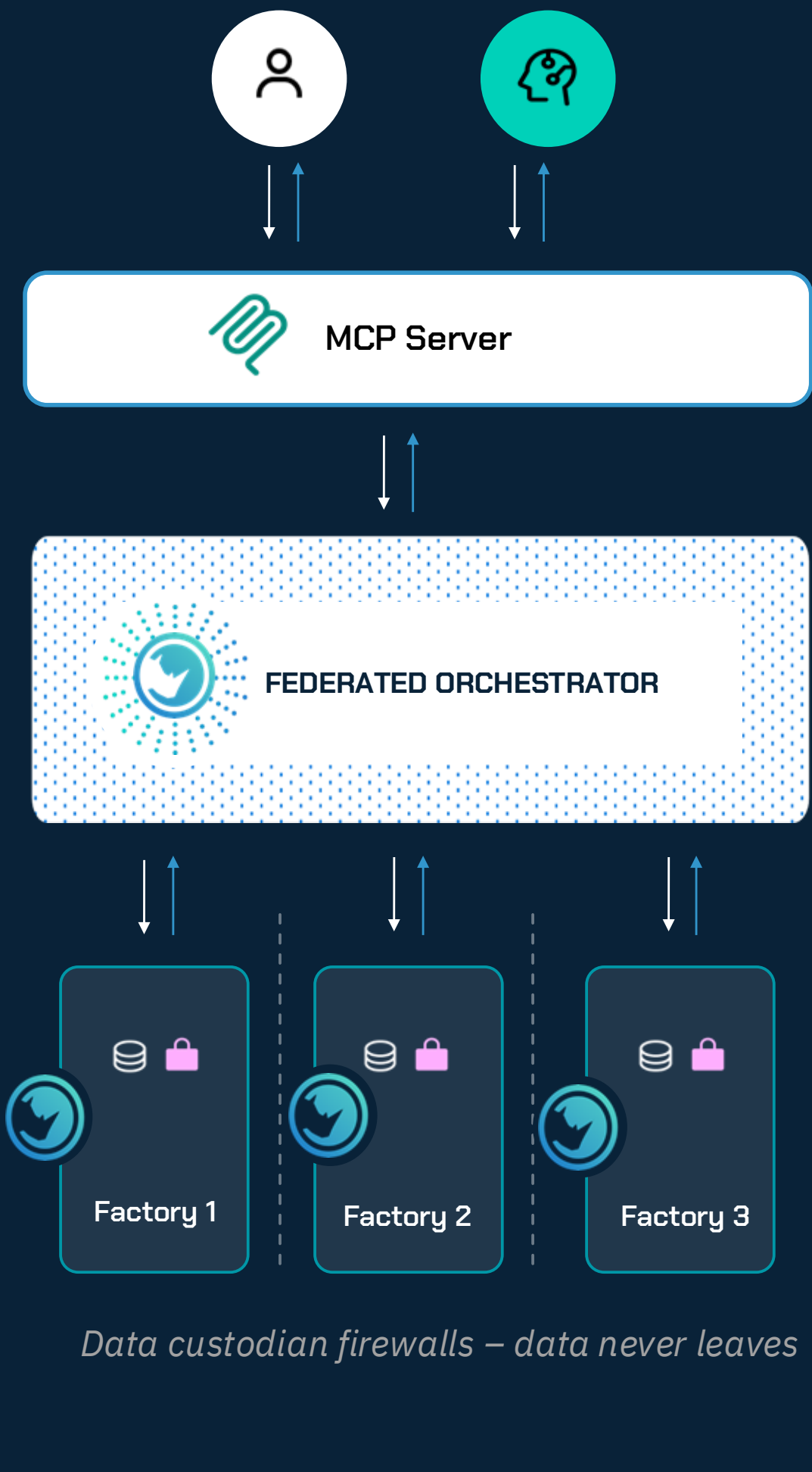


Illustration #1: Edge Model Networks

Industrial AI vendors can improve models from every customer deployment without touching customer data.

Vendor baseline model

A predictive maintenance or quality inspection model is trained using vendor's own data.

Customer deployments

Each factory has different equipment, sensors, lighting, operators, and failure modes.

Shared improvement

Local learning updates improve the global model without centralizing raw telemetry.

- Vendors post-train from their whole deployed customer base, not just pre-training data.
- Customers retain control over proprietary telemetry and production context.
- The model adapts to real-world drift without renegotiating data-sharing projects.

Illustration #2: Federated Edge Data Network

Run multi-plant and multi-operator analytics and share real time insights without raw data centralization

Global manufacturers

Defect patterns across plants

Grid operators

Regional demand forecasting and anomaly detection

Oil & gas

Joint subsurface models across proprietary seismic data

Banks

Cross-bank fraud and AML detection

Federated Agentic AI: Unlock Decentralized AI workloads

Creating an actionable context layer for agents across data silos

Put AI to work **across all your data**

Agents that can reason across the full data estate, including the silos that can't be centralized.

Move fast without getting blocked

Accelerate review and keep security, compliance, and legal on board by not moving sensitive data.

Know what your agents are doing

Govern, evaluate, and monitor your agents with secure global and local observability and AgentOps.

Scale patterns without retooling

Deploy faster by extending existing tooling and teams, then replicate across local installations.

THE ROAD AHEAD

Three strands of federated AI

Three architectural shifts that together make AI work where the data actually lives.

01

Invisible Compute

How does the work get done?

Goal-driven infrastructure. State the goal in natural language; the platform self-provisions across edge, on-prem, and cloud.

02

Federated Data Network

Where does the data live?

Data stays at source. Governed, versioned data products are discoverable to authorized agents — never centralized.

03

Integrated AI Systems

Who does the thinking?

Cognitive interface, orchestration engine, and specialist agents that route between large and small models per goal.

Different problems, one architecture · trust threads through every layer.

Infrastructure as a goal-driven utility

The human operator/planner agent states a goal. The infrastructure adapts.

MINIMIZE COST

spot instances · off-peak scheduling · cold-tier storage

TIME-TO-RESULT

HPC pool · parallelize · smaller routed models

MAX ACCURACY

larger models · deeper CoT · multi-inference voting

PRIVACY-FIRST

federated · DP noise · TEEs · data never leaves source

User intent

natural-language goals

Cloud ops agents

self-provision · self-heal · optimize

Multi-cloud abstraction

uniform API across providers

Composable resource layer

CPU/GPU · memory · storage

Physical / multi-cloud

AWS · GCP · Azure · on-prem

Introduction to Invisible Computation

Computation is where the data is: factory floor, utility network, bank, or fleet. Invisible does not mean uncontrolled.

Edge and site compute

Run inference, training updates, or analytics inside customer-controlled environments.

Confidential execution

Use trusted execution, attestation, policy controls, and audit logs for sensitive workloads.

Goal-oriented operations

Optimize for cost, speed, privacy, accuracy, or resilience instead of manually configuring infrastructure.

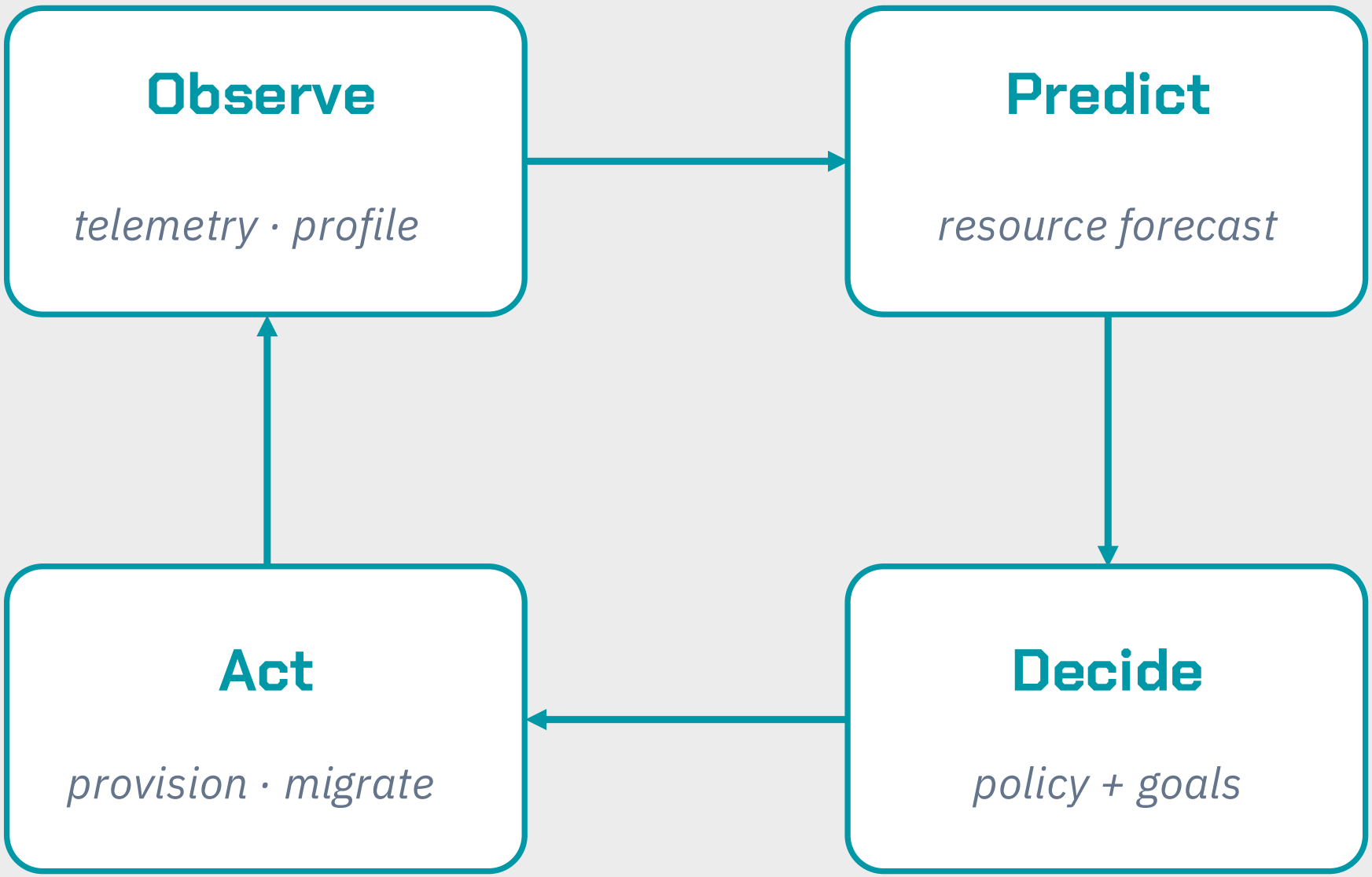
Edge agents plan and act autonomously within local constraints without central storage round-trips.

The platform abstracts operational complexity while preserving human oversight, policy boundaries, and auditable execution.

Application-aware provisioning · serverless everything with observability

Stateless functions today.
Stateful, long-running pipelines tomorrow.

The provisioning agent



Introduction to Federated Data Networks

From Data Sharing to governed data products — owned, versioned, policy-aware, agent-usable assets.

Discoverable

Cataloged so authorized users and agents can find the right asset.

Documented

Rich metadata, lineage, quality score, schema, and owner context.

Governed

Access, permitted operations, output controls, and audit trails.

Computable

Approved code/queries, training and inference jobs, and analyses run near the data.

Federated Ownership at Scale

Domain ownership	Data as a product	Self-serve platform	Federated governance
Data owned by those closest to it	Stable API · contract · SLOs	Domain teams ship	global rules · local execution

Federated data catalog + computational governance

Manufacturing	Energy and Grid	Supply Chain
Equipment Telemetry	**Grid Anomaly Patterns**	**Batch Quality**
data fabric (in-domain) in-place storage · governance	*data fabric (in-domain)* in-place storage · governance	*data fabric (in-domain)* in-place storage · governance

Data products expose computational interfaces → enables federated learning natively. Example pattern: MoE with frozen shared expert, per-domain trainable experts.

The nervous system — three intelligences

Data intelligence

Where to find what · what it means · how to combine it

- → scans federated fabrics
- → curates data and metadata. creates semantic layer.
- → decides on generation / data contribution needs

Compute intelligence

How to run it · where · for what cost

- → routes to model size
- → negotiates with cloud agents
- → honors user-stated goals

Algorithm intelligence

What method · what tools · what plan

- → selects from Model Foundry
- → discovers tools via MCP
- → decomposes goals (CoT/ToT)

Each intelligence is a coordinated set of agents and models.

Universal Tool Repository

UNIVERSAL TOOL REPOSITORY

Tools as discoverable services

Each tool registered with rich, machine-readable metadata:

function · inputs · outputs · resource profile · business purpose

The architectural shift:

tools become discoverable

rather than hard-integrated.

Enabled by emerging agent-tool protocols — MCP (Model Context Protocol) as the common surface.

Specialized agents

Data Curation Agent

queries the federated fabric, validates quality, harmonizes → analysis-ready dataset

Workflow Execution Agent

drafts pipelines, provisions compute, monitors, retries

Industrial Agent

domain-specific: safety pipelines, manufacturing QA, regulated workflows

Cognitive Interface + Orchestration Engine

COGNITIVE INTERFACE

The conversational partner — translates intent.

1 Conversational reasoning

fine-tuned LLM with persistent project context · disambiguates, asks clarifying questions

2 Persistent project memory

shared digital notebook · structured knowledge graph of hypotheses, configs, decisions

3 Proactive assistance

moves from reactive to predictive — suggests next steps, flags issues against prior work

4 Multimodal synthesis

code · interactive dashboards · process diagrams · time-series plots — picks the right modality

ORCHESTRATION ENGINE

The project manager — turns intent into execution.

1 Goal decomposition

Chain of Thought / Tree-of-Thoughts plan a sub-task and dependencies

2 Agent selection

consults dynamic agent registry · dispatches sub-tasks with params and data handles

3 Execution monitoring

manages retries, error paths · talks to Invisible Infra and Federated Data

4 Human-in-the-loop checkpoints

pauses workflow at pre-defined stages for review — transparency by design

PUTTING IT TOGETHER

End-to-end trace — one query, three strands

Plant Operator (in natural language)

"How does raw-material batch variation correlate with downstream defect rates across our three plants?"

1	Cognitive Interface	parses intent · identifies plant/line/batch domains · loads project memory
2	Orchestration Engine	ToT decomposition → DAG: { resolve plant scope, fetch batch QA, fetch defect data, correlate, validate }
3	Data Curation Agents	query federated catalog · resolve semantic links across plant/line/batch data · build virtual dataset
4	Model Router	user goal = privacy-first → federated correlation pattern, no raw batch data leaves source
5	Invisible Infrastructure	provisions edge compute at each plant · composes per-domain workers · TEE-backed at sites
6	Cognitive Interface	presents validated batch-defect correlations + reproducibility manifest

TRUST & GOVERNANCE

Full-stack trust — observable at every layer - controllable at every layer - auditable end to end

Human protection

approvals · escalation · explainability · accountability checkpoints

Model protection

model cards · usage monitoring · drift detection · adversarial & misuse controls

Execution protection

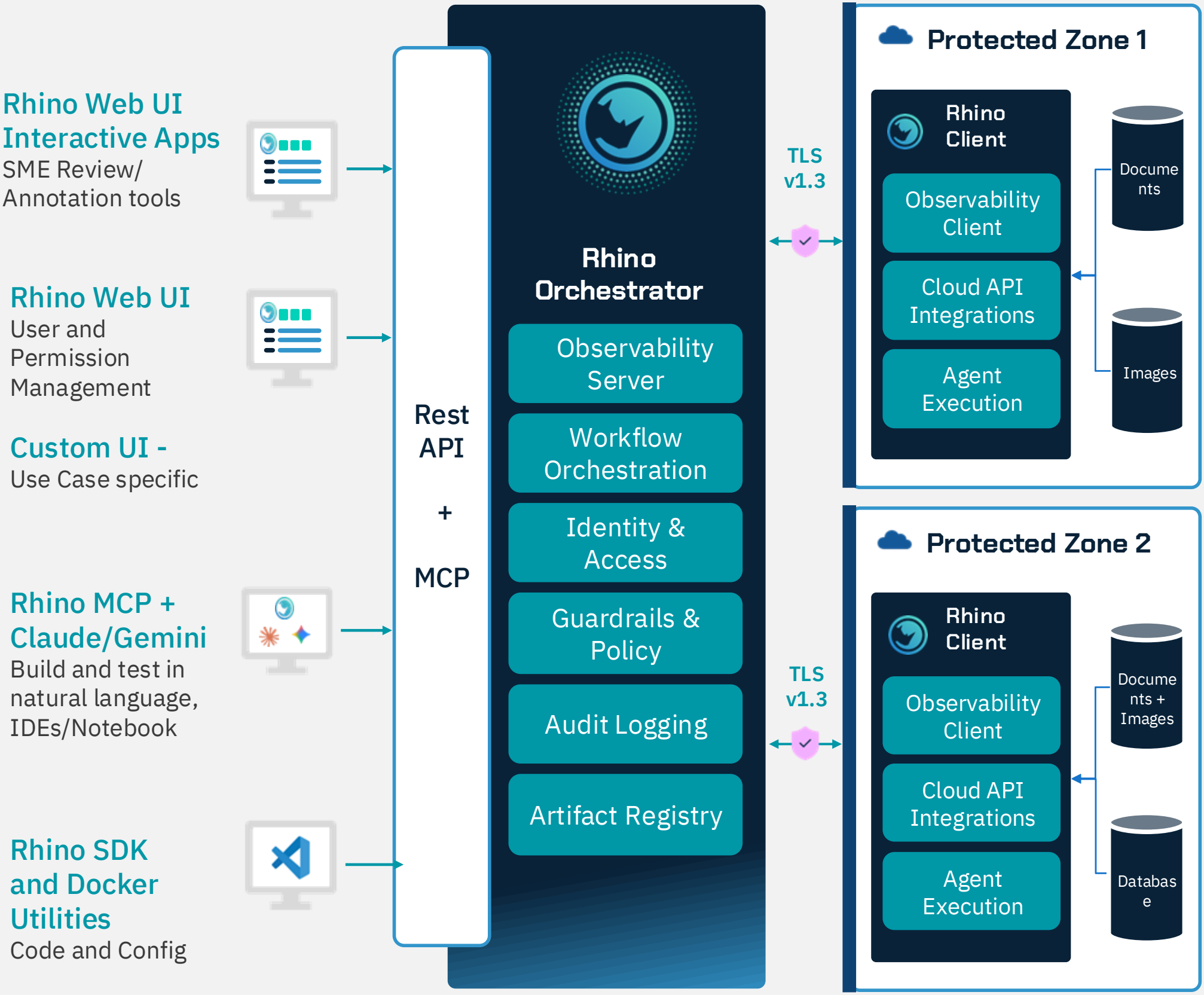
confidential computing · remote attestation · secure enclaves · TEEs

Data protection

anonymization · differential privacy · identity. access policies · audit logs · lineage

Rhino Technical Architecture for Multi-Agent AI Systems

Supports multi-cloud/on-prem orchestration and secure enclave edge execution



Open problems - where we need continuous efforts

The Collaboration Problem

01 Aligning with AI Collaborator

Modeling the collaborator's cognitive style, goals, standards. Persistent memory is just data — we don't yet have a measure of "synergy".

02 Multi-objective goal specification

"Minimize cost" is easy. Complex trade off surfaces — cost × accuracy × privacy × time — need formal objective languages that don't exist.

03 Evaluating AI collaborators

No benchmarks for "did this agent help me think better?" Static QA evals don't capture long-horizon scientific value.

The Composition Problem

04 Tool discovery at scale

MCP is a protocol, not a solution. Semantic matching, capability negotiation, trust evaluation across thousands of tools — open.

05 Compositional failure modes

Errors propagate. Hallucinations compound across orchestrated steps. Robust verification between agents is largely unsolved.

06 Reproducibility under autonomy

When agents pick tools and compute dynamically, what does "reproducible" mean? Manifests + replay? Frozen plans? Both?