# Akretic A2A Trust Gateway

Secure enterprise agent collaboration over A2A


**Akretic helps enterprise agents collaborate over A2A without letting the model decide what it may read, call, send, or approve.**

**Public posture:** This document describes a challenge prototype using synthetic data. It is not a production certification, security guarantee, or Google Cloud Marketplace listing.

# 1. Public overview

## The short version

Akretic A2A Trust Gateway is a B2B multi-agent control-plane prototype for vendor-risk review. It demonstrates secure enterprise agent collaboration where identity, policy, retrieval, approvals, and evidence are enforced outside the model.

### What it is

A challenge prototype for controlling multi-agent collaboration over A2A in enterprise procurement and security workflows.

### Who it serves

Procurement, security, legal, and platform teams that need auditable agent-assisted vendor-risk review.

### Core thesis

Do not rely on prompts alone. Gate retrieval, A2A calls, egress, approval, and evidence with deterministic controls.

## Why this matters now

- A2A makes agent discovery and coordination easier, increasing the need for enforceable cross-agent authorization.
- Enterprise RAG workflows can leak sensitive context unless permissions are applied before model context is assembled.
- Sensitive or external-facing actions need approval and evidence paths rather than relying on model behavior alone.

## Challenge alignment

- B2B focus: vendor-risk review for enterprise procurement/security teams.
- Cloud-native architecture: Cloud Run services, Cloud Storage, and Google Cloud logging/traceability.
- Vertex-powered intelligence: Gemini through Vertex AI.
- A2A interoperability: specialized agents expose Agent Cards and collaborate through A2A calls.

**Document scope:** This public brief explains the prototype, architecture, technologies, data posture, findings, and third-party rights. It intentionally excludes credentials, non-public routes, internal build notes, customer data, and unsupported production claims.

# 2. Problem to solve

Enterprises are beginning to deploy multiple AI agents that read internal documents, query external sources, and coordinate with other agents over A2A. But many agent stacks still rely too heavily on prompt instructions or model behavior to decide what an agent may retrieve, share, or trigger.

That creates a practical B2B governance problem: agents can pull restricted internal context, act on injected instructions, or exchange sensitive information with another agent without an enforceable policy decision, approval checkpoint, or evidence trail.

### Restricted context leakage

An agent retrieves internal or role-restricted material and places it into model context before a policy decision occurs.

### Prompt-only authorization

The model is asked to follow rules, but the rules are not enforced by a separate decision point.

### Cross-agent data movement

One agent can share sensitive context with another agent without clear identity, authorization, or purpose binding.

### Weak auditability

Teams cannot later prove what was retrieved, denied, approved, or exchanged during the run.

## Akretic control question

How can a business allow useful agent collaboration while preserving identity, permissions, approvals, and auditability outside the model?

# 3. Our solution and agent workflow

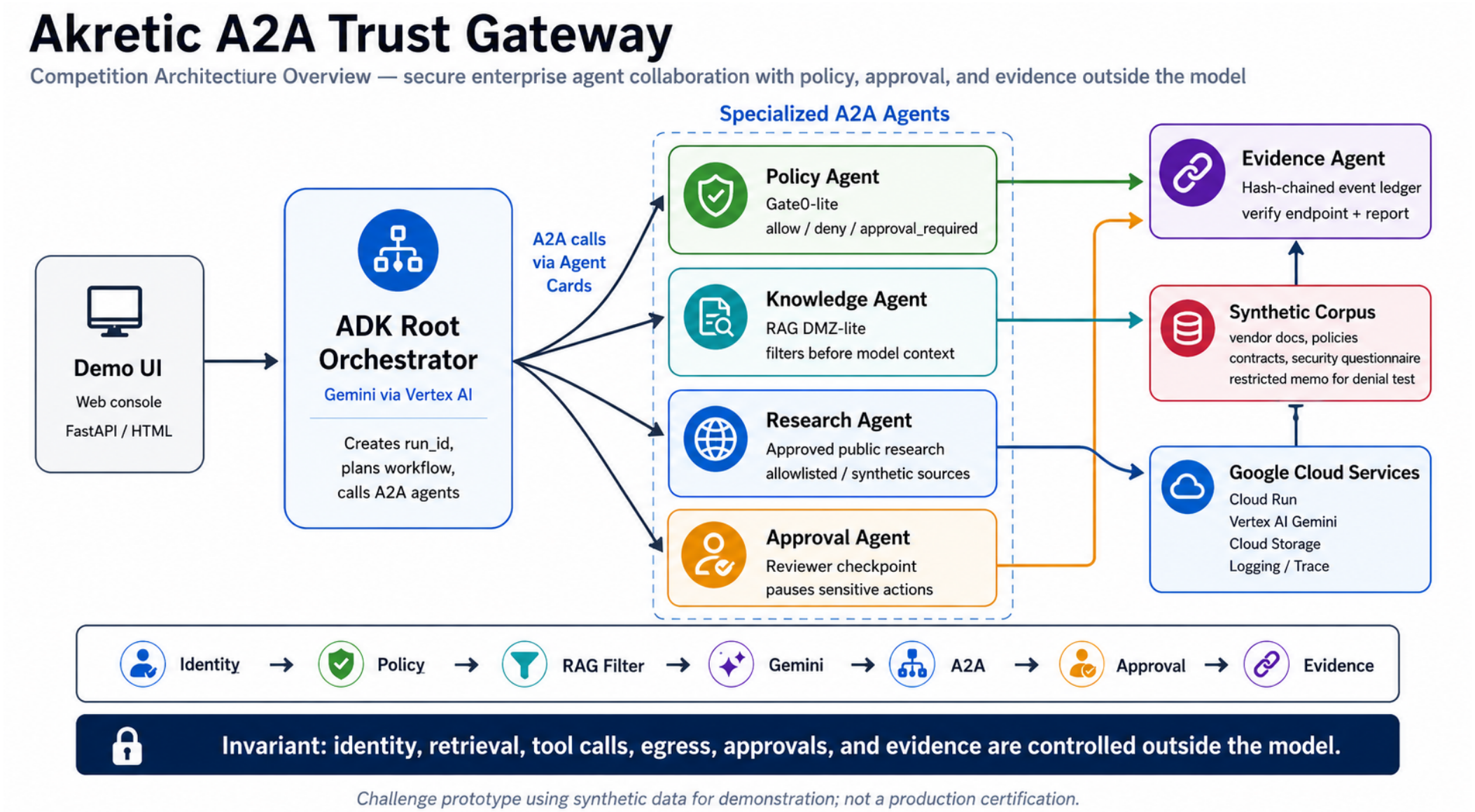
## Solution overview

Akretic A2A Trust Gateway coordinates specialized A2A agents through an ADK root orchestrator powered by Gemini. Before retrieval, public research, A2A exchange, or external-facing draft actions run, Gate0-lite evaluates deterministic policy and returns **allow**, **deny**, or **approval_required**. A RAG DMZ-lite layer filters synthetic enterprise documents by user role before context reaches Gemini, and material decisions are written to a tamper-evident evidence ledger.

| Step | Agent behavior | Control outcome |
|---|---|---|
| 1 | Root ADK Orchestrator receives a vendor-risk request and creates a run_id. | Every event is correlated from start to finish. |
| 2 | Policy Agent evaluates actor, action, resource, and context. | Gate0-lite returns allow, deny, or approval_required. |
| 3 | Knowledge Agent retrieves synthetic internal context through RAG DMZ-lite. | Restricted chunks are filtered before Gemini context. |
| 4 | Research Agent returns approved public or seeded research snippets. | No arbitrary external browsing is required for the demo path. |
| 5 | Approval/Evidence Agent pauses sensitive actions and records decisions. | Reviewer approval and verification report. |

## Narrow by design

The prototype focuses on vendor-risk review for B2B procurement and security workflows. Narrow scope makes the security story easier to demonstrate and keeps public claims credible.




**Invariant:** identity, retrieval, tool calls, egress, approvals, and evidence are controlled outside the model.

# 4. Key capabilities

| Capability | Public description |
| --- | --- |
| **A2A interoperability** | Specialized agents expose Agent Cards and are called by the root orchestrator as discoverable enterprise agents. |
| **Policy-mediated access** | Gate0-lite evaluates retrieval, research, egress, and approval intents outside the model. |
| **Permission-preserving RAG** | RAG DMZ-lite filters synthetic internal documents by group and classification before Gemini sees them. |
| **Approval-gated side effects** | Sensitive or external-facing actions return approval_required and pause for reviewer decision. |
| **Tamper-evident evidence** | A hash-chained ledger records material policy decisions, A2A calls, approvals, and results for verification. |

## What judges should see in the demo

Agents coordinate over A2A, restricted context is blocked before Gemini sees it, sensitive actions pause for approval, and the evidence report proves the path.

### What this document does not claim

This is not presented as unhackable, universally leak-proof, fully autonomous enterprise execution, legal non-repudiation, production certification, or Google Cloud Marketplace approval.

# 5. Technologies used and data sources

## Technologies powering the prototype

- **Intelligence:** Gemini API through Vertex AI.
- **Orchestration:** Google Agent Development Kit (ADK).
- **Interoperability:** Agent-to-Agent (A2A) protocol and Agent Cards.
- **Runtime:** Cloud Run services, containerized with Docker.
- **Deployment:** Artifact Registry plus Cloud Build or gcloud workflows.
- **Storage:** Cloud Storage for synthetic corpus and evidence exports.
- **Observability:** Cloud Logging and Cloud Trace.
- **RAG:** Vertex AI embeddings or custom in-memory embeddings depending on final configuration.
- **App stack:** Python, FastAPI, Pydantic, Uvicorn, pytest.
- **Evidence:** Standard cryptographic/hash libraries for the hash chain.

**Optional extensions:** Agent Runtime, Agent Registry, Firestore, Vertex AI embeddings, or Google Search grounding are included only if enabled in the final build.

## Data sources

The project uses a synthetic enterprise vendor-risk corpus created for the challenge. It includes internal security policy excerpts, procurement policy, a contract-review checklist, a vendor security questionnaire, a vendor profile, a SOC2-style summary, public-risk snippets, and a restricted memo used only to demonstrate denial.

Each document includes metadata such as source_id, classification, allowed_groups, document_type, and external_release_allowed. No customer data, private third-party data, production enterprise data, credentials, or non-public URLs are included in this brief.

synthetic corpus | role metadata | pre-context filtering | approval evidence

# 6. Findings and learnings

## Main learning

Multi-agent systems amplify governance risk because context can move from one agent boundary to another. A2A helps with discovery and coordination, but it does not by itself decide whether an agent should read, share, or act.

### Pre-context controls matter

Permission filtering must happen before retrieval enters model context. Post-generation redaction is not enough.

### Approval_required is valuable

A clear approval_required state is more useful than trying to make every sensitive operation fully autonomous.

### Evidence changes the demo

A hash-chained event trail makes the workflow explainable: what was allowed, denied, paused, approved, and returned.

### Narrow workflows win trust

Vendor-risk review is specific enough to show meaningful controls without pretending to solve every enterprise workflow.

## Enterprise pattern

Cloud Run and ADK make a solo prototype practical, but the stronger enterprise pattern is not broad autonomy. It is controlled collaboration: useful agents, deterministic policy checks, approval gates, and evidence outside the model.

# 7. Public release posture

## Third-party integrations

The build uses Google Cloud services and APIs including Vertex AI/Gemini, ADK, A2A, Cloud Run, Cloud Storage, Cloud Logging, and optional supporting services such as Agent Runtime, Agent Registry, Vertex AI embeddings, Firestore, or Google Search grounding if enabled in the final build.

Open-source Python packages may include FastAPI, Uvicorn, Pydantic, pytest, python-dotenv, and cryptography, each used under its applicable open-source license.

## Rights and data statement

All project-specific content, demo documents, policies, prompts, and sample vendor data are original synthetic materials created for the challenge. The project does not use third-party proprietary data, customer records, or third-party trademarks/logos beyond nominative references to the technologies used.

| Say | Do not say |
|---|---|
| Policy-mediated agent collaboration. | Unhackable, guaranteed compliance, or impossible to fail. |
| Permission-preserving retrieval for the prototype corpus. | Universal data-leak prevention. |
| Approval-gated side effects. | Fully autonomous enterprise action execution. |
| Tamper-evident evidence chain. | Legal non-repudiation unless formally reviewed. |
| Built on Google Cloud technologies for the challenge. | Google Cloud Marketplace approved or certified. |

**References:** Google for Startups AI Agents Challenge rules and resource guide; Google Cloud documentation for ADK, A2A, Cloud Run, Vertex AI/Gemini, Cloud Storage, Logging/Trace; Akretic public product thesis. All references are public or competition-facing.