

IG

INFERENCE GATEWAY

A Unified Platform for Governed AI Agent Execution
with Human-in-the-Loop Tool Verification

Author: rng

Submitted to: Apart Research
Technical AI Governance Challenge

Date: February 2026

Abstract: As AI systems transition from passive assistants to autonomous agents capable of executing real-world actions, the governance challenge shifts from controlling model *outputs* to controlling model *behaviors*. This paper introduces Inference Gateway, a unified self-hosted platform that addresses critical gaps in AI governance through three novel contributions: a two-stage privacy-preserving pipeline, a multi-factor human-in-the-loop tool approval system with mobile device integration, and a provider-agnostic abstraction layer enabling consistent governance policies across heterogeneous AI infrastructure.

Contents

1	Introduction	3
1.1	The Governance Gap in Agentic AI	3
1.2	Contributions	3
2	Background and Related Work	4
2.1	The Evolution of AI Governance Approaches	4
2.2	The Model Context Protocol and Tool Use	4
2.3	Human-in-the-Loop Systems	4
3	System Architecture	5
3.1	Overview	5
3.2	Gateway Service: Provider Abstraction	5
3.3	Five Routing Strategies	6
3.4	Runner Service: Workflow Execution	7
3.5	Memory Service: Vector Storage	7
4	Two-Stage Privacy Pipeline	7
4.1	The Privacy Problem in Cloud Inference	7
4.2	Architectural Solution	8
4.3	Implementation Details	8
4.4	Governance Benefits	9
5	Human-in-the-Loop Tool Approval	9
5.1	The Tool Execution Problem	9
5.2	Multi-Factor Tool Approval System	9
5.2.1	Approval Modes	10
5.3	Duo Mobile Integration	10
5.4	Risk-Based Approval Escalation	10
6	Agent Configuration and the CODEC Interface	11
6.1	The CODEC Design Philosophy	11
6.2	Agent Configuration Model	11
6.3	Tool Configuration Per Agent	12
7	Observability and Audit Infrastructure	12
7.1	Distributed Tracing with OpenTelemetry	12
7.2	Governance Applications	12
8	Security Architecture	13
8.1	Defense in Depth	13
8.2	Secrets Management	13
8.3	Threat Model	14
9	Case Studies and Applications	14
9.1	Financial Services: Trade Execution Agent	14
9.2	Healthcare: Clinical Decision Support	14
9.3	Legal: Document Analysis	14
10	Comparison with Existing Approaches	15

11 Limitations and Future Work	15
11.1 Current Limitations	15
11.2 Future Directions	15
12 Conclusion	15
A System Specifications	16
A.1 Service Ports	16
A.2 Technology Stack	17
B Governance Checklist	17

1 Introduction

1.1 The Governance Gap in Agentic AI

The rapid evolution of large language models from text generators to autonomous agents has created a fundamental governance vacuum. While substantial research effort has focused on alignment during training and content filtering during inference, comparatively little attention has been paid to the *execution layer*—the moment when an AI agent’s decision translates into real-world action.

RLHF, Constitutional AI, filters, moderation, actions

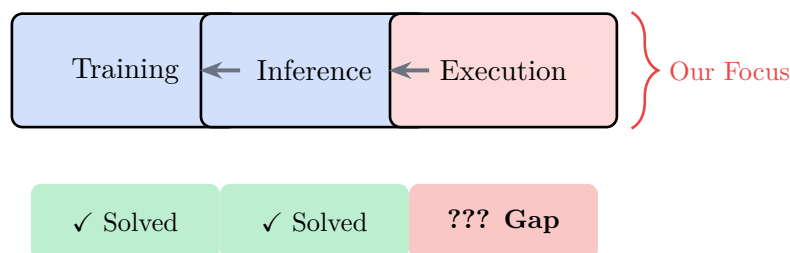


Figure 1: The AI governance landscape. Current approaches address training and inference, but execution-time governance remains an unsolved gap.

Consider a financial AI agent tasked with portfolio optimization. Current governance approaches might ensure the model doesn’t generate harmful content or reveal sensitive training data, but they provide no mechanism to verify that the agent’s decision to execute a \$10 million trade is appropriate, timely, or even intended by the human operator. This gap between model output and world-state change represents one of the most significant unsolved problems in technical AI governance.

Key Insight

The critical intervention point for agentic AI governance is not *what the model says*, but *what the model does*. As AI systems gain agency, governance must shift from controlling outputs to controlling actions.

1.2 Contributions

This paper presents **Inference Gateway**, a production-ready platform that addresses the governance gap through five key contributions:

- 1. Architectural Separation of Concerns:** A microservices architecture that cleanly separates inference (model interaction), execution (tool invocation), memory (context retrieval), and management (configuration/audit) concerns, enabling fine-grained governance at each layer.
- 2. Two-Stage Privacy Pipeline:** A novel approach where sensitive operations (PII detection, credential handling, local tool execution) are processed by self-hosted models *before* cloud inference, ensuring sensitive data never leaves the local environment.
- 3. Human-in-the-Loop Tool Approval:** A multi-factor authentication system for tool invocation supporting Duo Mobile push notifications, webhook callbacks, and manual approval workflows—treating high-stakes AI actions with the same rigor as financial transactions.

4. **Provider Abstraction with Consistent Governance:** A unified gateway presenting identical OpenAI-compatible and Ollama-compatible APIs regardless of the underlying provider, ensuring governance policies apply uniformly across all AI infrastructure.
5. **Comprehensive Audit Trail:** Full request/response tracing with OpenTelemetry integration, enabling post-hoc analysis of agent behavior and supporting compliance requirements.

2 Background and Related Work

2.1 The Evolution of AI Governance Approaches

AI governance has historically operated at three levels, each with distinct mechanisms and limitations:

Level	Mechanisms	Limitations
Training-time	RLHF, Constitutional AI, dataset curation	Cannot adapt to deployment-specific requirements
Inference-time	Content filters, rate limits, output sanitization	Reactive—detects problems after generation
Execution-time	Inference Gateway	<i>Novel contribution of this work</i>

Table 1: Comparison of AI governance approaches by intervention level.

2.2 The Model Context Protocol and Tool Use

The Model Context Protocol (MCP), introduced by Anthropic in 2024, established a standardized interface for LLMs to interact with external tools and data sources. MCP servers expose capabilities through a JSON-RPC interface, enabling models to query databases, invoke APIs, manipulate files, and interact with arbitrary external systems.

While MCP provides excellent *capability* abstraction, it explicitly defers *governance* to the implementation layer. The protocol defines what tools exist and how to invoke them, but not whether a given invocation should be permitted.

Governance Implication

Inference Gateway fills this gap by wrapping MCP tool access in a comprehensive approval and audit framework. Every tool invocation—whether from MCP servers or built-in tools—passes through the same governance layer.

2.3 Human-in-the-Loop Systems

Prior work on human-in-the-loop AI has largely focused on:

- **Labeling pipelines** for training data annotation
- **Correction interfaces** for chatbot response refinement
- **Escalation mechanisms** for customer service applications

The application of HITL principles to autonomous agent execution—treating AI tool invocation as analogous to privileged system access requiring multi-factor authentication—represents a novel contribution of this work.

3 System Architecture

3.1 Overview

Inference Gateway comprises five interconnected microservices, each addressing a distinct governance concern:

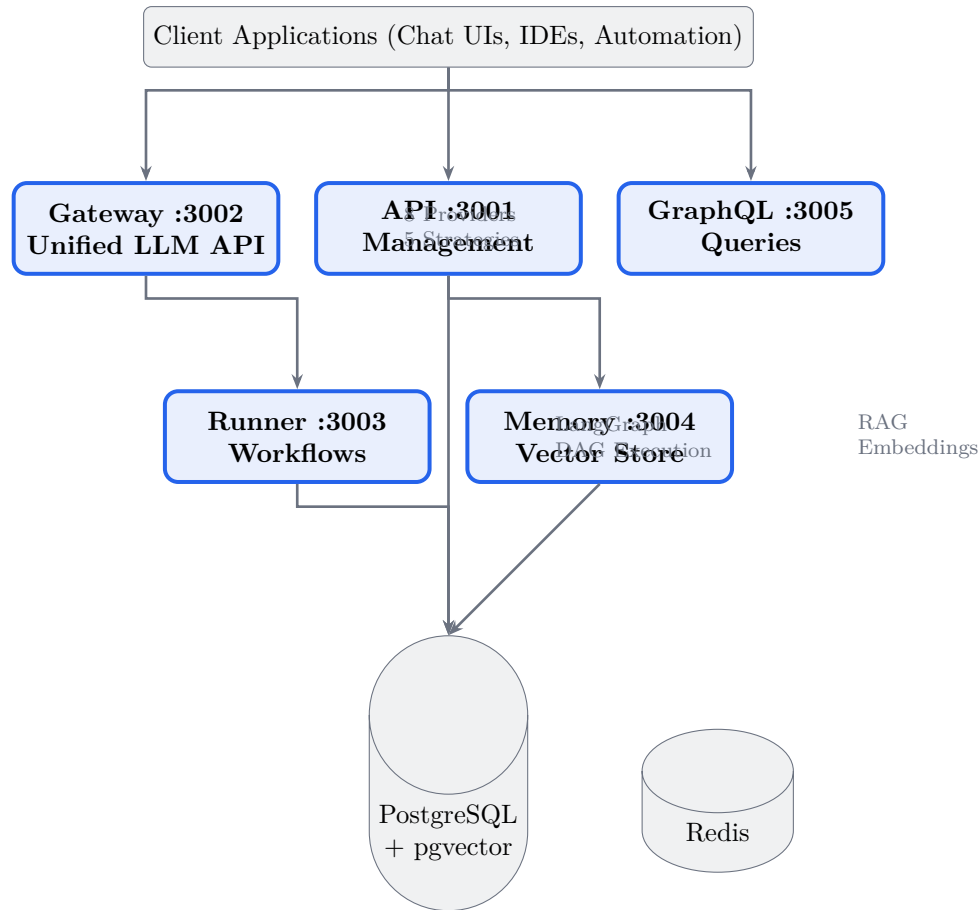


Figure 2: Inference Gateway microservices architecture. Each service addresses a distinct governance concern.

3.2 Gateway Service: Provider Abstraction

The Gateway service (port 3002) presents dual API interfaces—OpenAI-compatible (`/v1/chat/completions`) and Ollama-compatible (`/api/chat`)—that route requests to any of eight supported providers:

Provider	Type	Primary Use Case
OpenAI	Cloud	General-purpose, GPT-4/o1 access
Anthropic	Cloud	Claude models, extended context windows
Azure OpenAI	Cloud	Enterprise compliance, regional deployment
Google Gemini	Cloud	Multimodal capabilities, long-context
Ollama	Local	Privacy-sensitive workloads, offline operation
vLLM	Local	High-throughput self-hosted inference
TGI	Local	HuggingFace model serving
HuggingFace Inference	Cloud	Open model access via API

Table 2: Supported LLM providers with deployment type and use cases.

Governance Implication

By abstracting provider differences, organizations can implement **uniform logging, rate limiting, content filtering, and access control** regardless of which models are actually invoked. A single policy configuration applies consistently across the entire AI infrastructure.

3.3 Five Routing Strategies

The Gateway implements five distinct routing strategies, each with different governance characteristics:

1. Default — Direct routing to specified provider. Full control, simple deployments.

2. Weighted — Probabilistic distribution (70% A, 20% B, 10% C). Load balancing, cost optimization.

3. Round-Robin — Sequential rotation. Deterministic distribution for audit purposes.

4. Failover — Primary/backup chains. Resilience with clear escalation paths.

5. Sticky — Session affinity via hashing. Consistency for stateful conversations.

Figure 3: Five routing strategies with governance implications.

The choice of routing strategy has governance implications. Failover routing creates an audit trail of which providers were attempted before success, while sticky routing ensures conversation context remains with a single provider (important for data residency requirements).

3.4 Runner Service: Workflow Execution

The Runner service (port 3003) implements LangGraph-style directed acyclic graph execution for complex agent workflows. Five node types are supported:

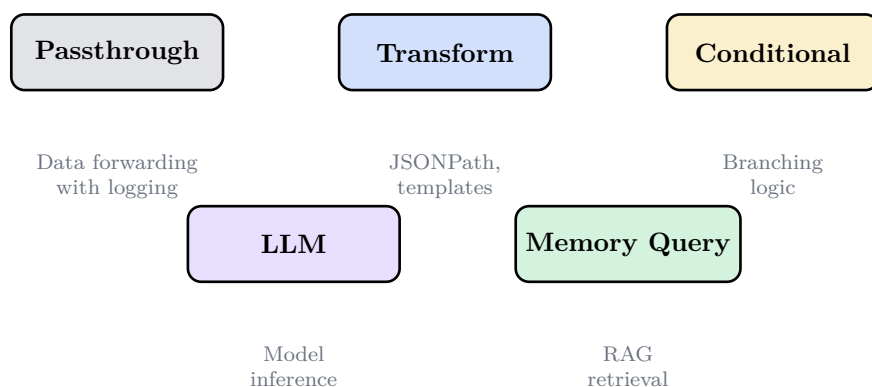


Figure 4: Runner service node types for workflow execution.

Governance Implication

By decomposing agent behavior into discrete, auditable nodes, the Runner enables **fine-grained monitoring of agent decision-making**. Each node execution is logged, creating a complete trace of how inputs transformed into outputs and what intermediate decisions were made.

3.5 Memory Service: Vector Storage

The Memory service (port 3004) provides vector storage and retrieval using PostgreSQL with the pgvector extension. Three embedding providers are supported:

- **Ollama** — Local, privacy-preserving
- **OpenAI** — Cloud, high-quality
- **HuggingFace Transformers** — Local, customizable

The choice of embedding provider affects data privacy. Organizations handling sensitive documents can configure local-only embeddings, ensuring document content never leaves the infrastructure while still enabling semantic search capabilities.

4 Two-Stage Privacy Pipeline

4.1 The Privacy Problem in Cloud Inference

Cloud-based LLM inference creates inherent tension between capability and privacy. Organizations may need powerful models (GPT-4, Claude) for sophisticated reasoning while simultaneously handling data that cannot leave the local environment:

- Personally Identifiable Information (PII)
- Credentials and API keys
- Proprietary business information
- Data subject to regulatory restrictions (HIPAA, GDPR, SOX)

4.2 Architectural Solution

Inference Gateway implements a two-stage pipeline that cleanly separates concerns:

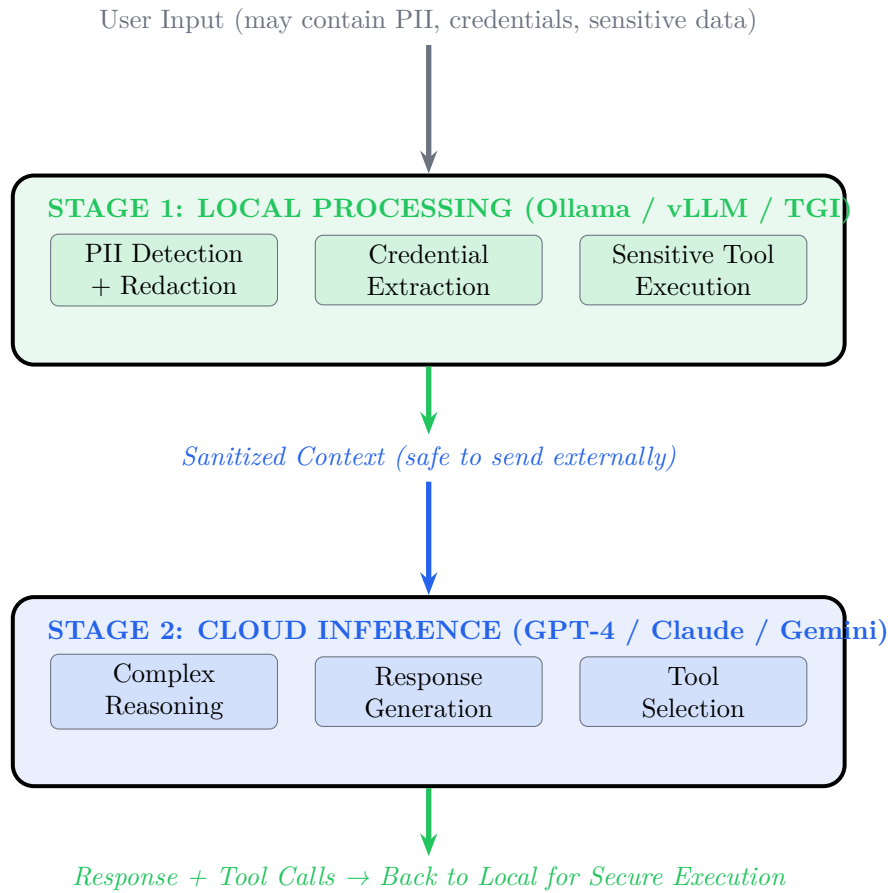


Figure 5: Two-stage privacy pipeline. Sensitive operations processed locally before cloud inference.

4.3 Implementation Details

The frontend application coordinates this pipeline through the CODEC agent interface. When a user configures an agent, they specify:

```

1 interface AIAgent {
2   id: string;
3   name: string;
4   role: AgentRole;
5   status: AgentStatus;
6   model: string;           // Cloud model for reasoning
7   localModel?: string;     // Local model for sensitive ops
8   piiEnabled?: boolean;    // PII detection toggle
9   stealthMode?: boolean;   // Minimize external exposure
10  tools?: AgentToolConfig[];
11  mcpServers?: string[];
12 }
  
```

Listing 1: Agent configuration with privacy settings (TypeScript)

4.4 Governance Benefits

This architecture provides several governance advantages:

1. **Data Minimization:** Only sanitized, non-sensitive context reaches cloud providers
2. **Auditability:** Clear demarcation of what data flows where
3. **Regulatory Compliance:** Enables use of powerful cloud models while satisfying data residency requirements
4. **Defense in Depth:** Even if cloud provider is compromised, sensitive data remains local

5 Human-in-the-Loop Tool Approval

5.1 The Tool Execution Problem

When an AI agent decides to invoke a tool—sending an email, modifying a database, executing code, or transferring funds—there exists a critical moment between decision and action. Current systems offer inadequate options:

Approach	Limitation
Fully autonomous	No human oversight for high-stakes actions
Fully restricted	Requires approval for every action (impractical)
Binary permission	Tools either allowed or blocked, no context-awareness

Table 3: Limitations of existing tool permission approaches.

5.2 Multi-Factor Tool Approval System

Inference Gateway introduces a tool permission system modeled on multi-factor authentication for privileged access:

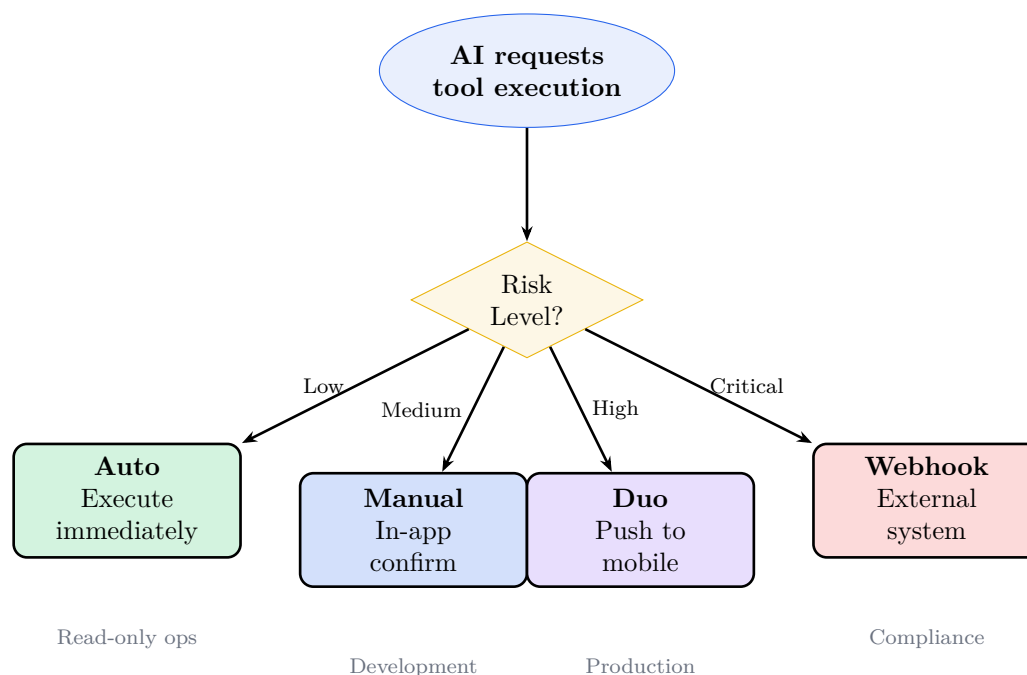


Figure 6: Multi-factor tool approval system with risk-based escalation.

5.2.1 Approval Modes

- Auto** Tool executes immediately—appropriate for read-only operations with low risk (fetching weather, reading public documentation).
- Manual** In-application modal requires explicit human confirmation before execution. Appropriate for development/testing or when mobile integration isn't available.
- Duo** Integration with Duo Mobile sends push notifications to enrolled devices. The human operator receives tool name, parameters, conversation context, and Approve/Deny buttons. This treats AI tool invocation with the same rigor as VPN access or financial transaction approval.
- Webhook** HTTP callback to external system enables integration with existing approval workflows (ServiceNow, JIRA, custom ticketing systems). The external system receives tool details and returns approve/deny decision.
- Deny** Tool is completely blocked—the agent receives an error and must find alternative approaches.

5.3 Duo Mobile Integration

The Duo enrollment flow demonstrates the production-ready nature of this system:

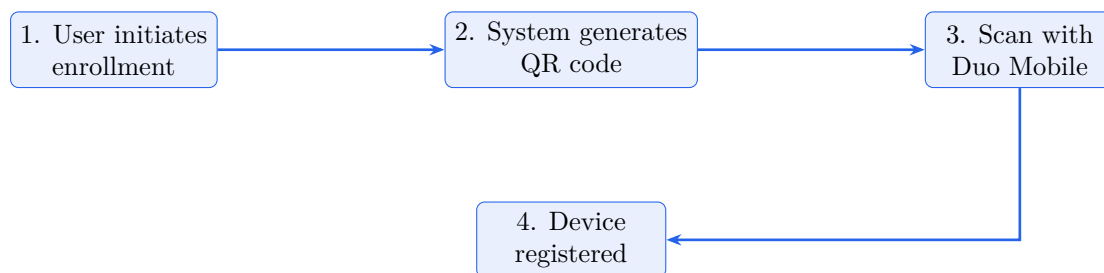


Figure 7: Duo Mobile enrollment flow for tool approval.

5.4 Risk-Based Approval Escalation

Tools are classified by risk level, with approval requirements escalating accordingly:

Risk Level	Default Mode	Override	Audit	Example
Low	Auto	Yes	Optional	Read public API
Medium	Manual	Yes	Recommended	Send notification
High	Duo/Webhook	Admin only	Required	Database write
Critical	Duo + Webhook	No	Required + Alert	Financial transaction

Table 4: Risk-based approval escalation matrix.

Critical-risk tools (e.g., production database writes, financial transactions, system commands) can require *both* Duo approval and webhook confirmation, implementing defense-in-depth for the highest-stakes operations.

Key Insight

We treat AI agent actions with the same rigor as financial transactions. Just as transferring \$10,000 requires multi-factor authentication and audit logging, executing a high-stakes tool should require equivalent verification.

6 Agent Configuration and the CODEC Interface

6.1 The CODEC Design Philosophy

The agent management interface, codenamed CODEC (a reference to Metal Gear Solid's communication system), reflects a deliberate design philosophy: AI agents are *operatives* that require mission briefings, equipment loadouts, and command oversight.

This metaphor serves both UX and governance purposes:

- Agents have **roles** (Commander, Analyst, Specialist, Operator, Observer) that imply capability levels
- Agents have **status** (Active, Standby, Offline, Compromised) reflecting operational state
- Agents have **callsigns** and **frequencies** reinforcing the command-and-control paradigm

6.2 Agent Configuration Model

```

1 interface AIAgent {
2   id: string;
3   callsign: string;           // Human-memorable identifier
4   name: string;              // Display name
5   role: AgentRole;           // Capability classification
6   status: AgentStatus;       // Operational state
7   frequency: string;         // Visual identifier (e.g., "141.12")
8
9   // Model Configuration
10  model: string;              // Primary LLM
11  localModel?: string;        // Privacy-preserving local model
12
13  // Tool Access
14  tools?: AgentToolConfig[];  // Permitted tools with settings
15  mcpServers?: string[];      // Connected MCP servers
16
17  // Behavioral Constraints
18  systemPrompt?: string;      // Base instructions
19  temperature?: number;       // Randomness control
20  maxTokens?: number;         // Output limits
21
22  // Privacy Controls
23  piiEnabled?: boolean;       // PII detection
24  stealthMode?: boolean;      // Minimize exposure
25
26  // Audit
27  createdAt?: Date;
28  updatedAt?: Date;
29 }

```

Listing 2: Complete agent configuration interface (TypeScript)

6.3 Tool Configuration Per Agent

Each agent maintains individual tool configurations, enabling fine-grained access control:

```

1 interface AgentToolConfig {
2   toolId: string;
3   enabled: boolean;
4   approvalOverride?: 'auto' | 'duo' | 'webhook' | 'manual' | 'deny';
5   parameterConstraints?: Record<string, unknown>;
6   usageQuota?: {
7     maxDaily: number;
8     maxPerConversation: number;
9   };
10 }
```

Listing 3: Per-agent tool configuration (TypeScript)

This enables scenarios like:

- Agent A can execute database reads automatically, but Agent B requires approval
- All agents are limited to 100 API calls per day to external services
- Critical tools require Duo approval for all agents, regardless of individual settings

7 Observability and Audit Infrastructure

7.1 Distributed Tracing with OpenTelemetry

Every request through Inference Gateway generates comprehensive traces:

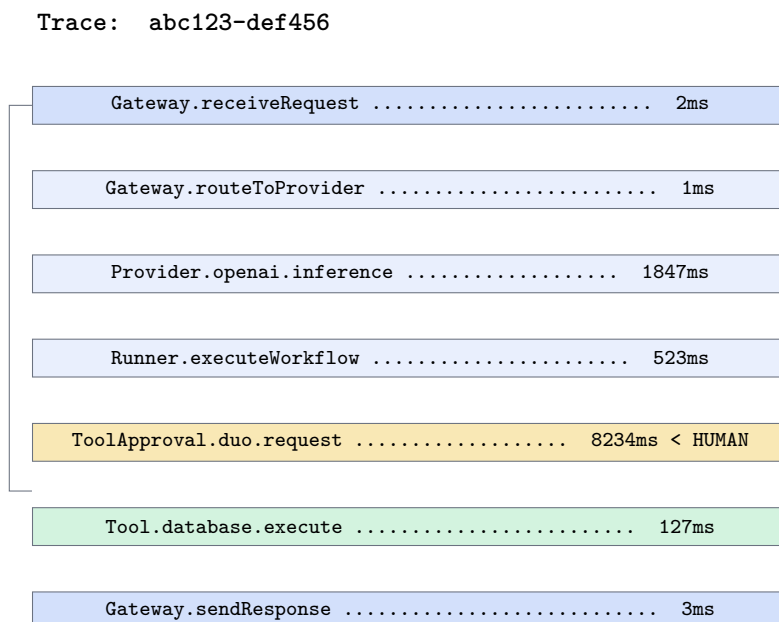


Figure 8: OpenTelemetry trace showing complete request lifecycle including human approval.

7.2 Governance Applications

The trace infrastructure enables:

1. **Anomaly Detection:** Unusual patterns (high token usage, repeated tool denials, error spikes) trigger alerts

2. **Compliance Reporting:** Generate audit reports showing all privileged actions with approval chains
3. **Cost Attribution:** Track inference costs per user, project, or agent
4. **Performance Optimization:** Identify bottlenecks in approval workflows or provider latency
5. **Incident Investigation:** Full replay of agent decision-making for post-hoc analysis

8 Security Architecture

8.1 Defense in Depth

Security is implemented at multiple layers:

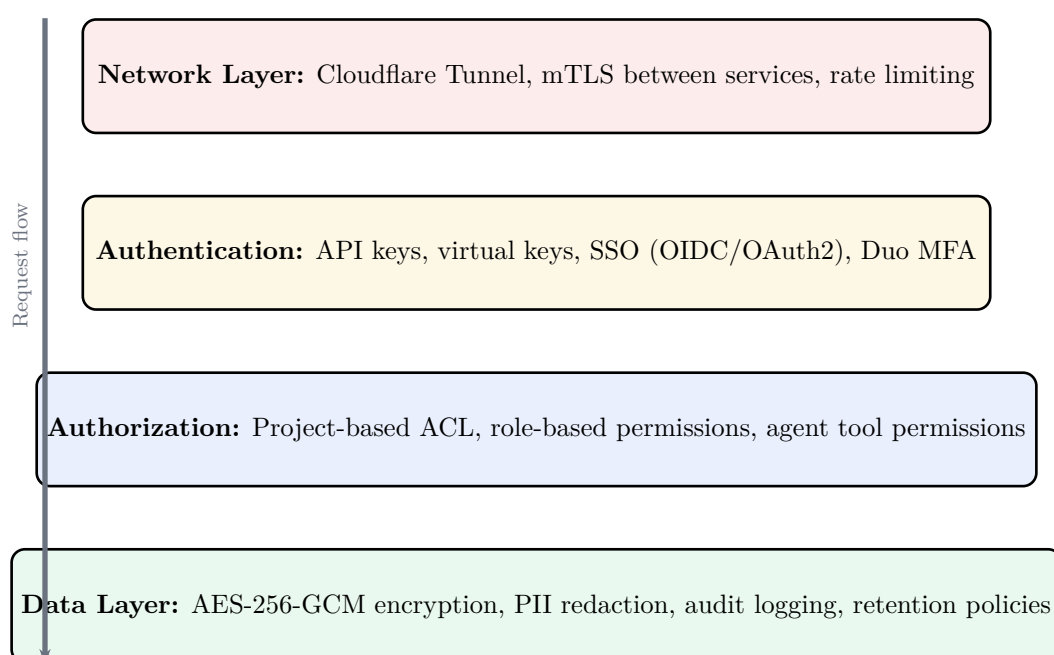


Figure 9: Defense in depth security architecture.

8.2 Secrets Management

Provider credentials and API keys are stored encrypted in PostgreSQL using AES-256-GCM. Virtual keys provide an additional abstraction layer:

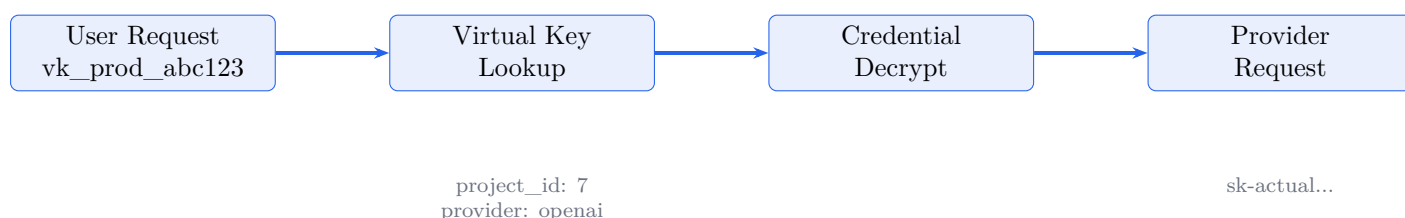


Figure 10: Virtual key abstraction for credential protection.

8.3 Threat Model

Threat	Mitigation
Credential theft	Encryption at rest, virtual key abstraction
Unauthorized tool execution	Multi-factor approval system
Data exfiltration via prompts	Two-stage privacy pipeline
Prompt injection attacks	Input sanitization, parameter constraints
Denial of service	Rate limiting, quota management
Insider threat	Audit logging, role-based access
Provider compromise	Local processing options, data minimization

Table 5: Threat model with corresponding mitigations.

9 Case Studies and Applications

9.1 Financial Services: Trade Execution Agent

A hedge fund deploys an AI agent for trade execution:

- **Cloud Model:** GPT-4 for market analysis
- **Local Model:** Llama for PII/credential handling
- **Tools:** Market data (auto), Trade execution (Duo + Webhook)
- **Result:** No trade executes without human approval AND compliance logging

9.2 Healthcare: Clinical Decision Support

A hospital network implements clinical decision support:

- **Model:** Local fine-tuned model only (no cloud—PHI protection)
- **Tools:** EHR read (auto), Treatment recommendation (manual), Prescription (deny)
- **Result:** AI assists diagnosis, never prescribes without physician

9.3 Legal: Document Analysis

A law firm uses Inference Gateway for contract analysis:

- **Cloud Model:** Claude for analysis
- **Local Model:** Document ingestion only
- **Tools:** Document search (auto), Client communication (Duo)
- **Memory:** Vector storage with local embeddings only (attorney-client privilege)

10 Comparison with Existing Approaches

Feature	Inference Gateway	Commercial	Agent Frameworks
Self-hosted	✓ Required	Optional	✓
Human-in-the-loop	Multi-factor	Basic	None
Privacy pipeline	Two-stage	Filtering	None
Provider support	8 unified	Varies	Varies
MCP integration	Native	Limited	Framework
Open source	✓	×	✓
Workflow execution	LangGraph-style	API chaining	Native
Enterprise auth	SSO + MFA	SSO	Basic
Compliance-ready	✓	✓	×

Table 6: Feature comparison with commercial gateways and agent frameworks.

Key Insight

Key Differentiator: Inference Gateway treats governance as a *first-class concern*, not an afterthought. Security, privacy, and human oversight are architectural foundations, not bolted-on features.

11 Limitations and Future Work

11.1 Current Limitations

1. **Approval Latency:** Human-in-the-loop introduces delay; not suitable for real-time applications requiring sub-second responses
2. **Mobile Dependency:** Duo integration requires smartphone access
3. **Complexity:** Five-service architecture has operational overhead
4. **Local Model Quality:** Privacy-preserving pipeline limited by local model capabilities

11.2 Future Directions

1. **Automated Approval Learning:** ML models to predict approval likelihood, reducing human burden for routine operations
2. **Federated Governance:** Cross-organizational policy sharing while maintaining data isolation
3. **Formal Verification:** Proof-carrying workflows that guarantee safety properties
4. **HSM Integration:** Hardware security module support for credential storage
5. **Streaming Approval:** Approve/deny as agent generates, not just before execution

12 Conclusion

As AI systems assume greater agency in the world, governance must evolve from controlling outputs to controlling actions. Inference Gateway demonstrates that production-ready human-in-the-loop systems for AI tool execution are not only possible but practical—borrowing proven

patterns from privileged access management and adapting them to the unique requirements of autonomous agents.

The platform’s three key contributions—**two-stage privacy pipelines**, **multi-factor tool approval**, and **provider-agnostic governance**—address critical gaps in the current AI safety landscape. By treating AI agent actions with the same rigor as financial transactions, organizations can deploy capable autonomous systems while maintaining meaningful human oversight.

Core Thesis

The critical governance intervention point for agentic AI is not what the model *says*—it’s what the model *does*. Treating AI agent actions with the same rigor as financial transactions is not just possible—it’s practical.

References

- [1] Anthropic. (2024). Model Context Protocol Specification. <https://modelcontextprotocol.io>
- [2] Bai, Y., et al. (2022). Constitutional AI: Harmlessness from AI Feedback. arXiv:2212.08073
- [3] Duo Security. (2023). Duo Push Authentication Protocol. Cisco Systems.
- [4] LangChain. (2024). LangGraph: Stateful Multi-Actor Applications. <https://langchain.com>
- [5] OpenAI. (2024). OpenAI API Reference. <https://platform.openai.com/docs>
- [6] PostgreSQL Global Development Group. (2024). pgvector: Vector Similarity Search for Postgres.
- [7] W3C. (2024). OpenTelemetry Specification. <https://opentelemetry.io>

A System Specifications

A.1 Service Ports

Service	Port	Protocol
API	3001	HTTP/REST
Gateway	3002	HTTP/REST
Runner	3003	HTTP/REST
Memory	3004	HTTP/REST
GraphQL	3005	HTTP/GraphQL

Table 7: Service port assignments.

A.2 Technology Stack

Layer	Technology
Frontend	Next.js 14, React, Zustand, Tailwind CSS
API Services	Fastify, Node.js, TypeScript
Database	PostgreSQL 15+, pgvector
Cache	Redis
ORM	Drizzle ORM
Observability	OpenTelemetry, Pino
Authentication	Passport.js, OIDC
Containerization	Docker, Docker Compose

Table 8: Technology stack by layer.

B Governance Checklist

For organizations deploying Inference Gateway:

- ☐ Define tool risk classifications for your use case
- ☐ Establish approval workflow ownership (who approves what)
- ☐ Configure audit retention policies per compliance requirements
- ☐ Set up alerting for anomalous patterns
- ☐ Document data flow for privacy impact assessments
- ☐ Train operators on Duo enrollment and approval process
- ☐ Establish incident response procedures for agent misbehavior
- ☐ Schedule regular reviews of agent permissions and tool access

This paper describes the Inference Gateway system as implemented in February 2026. The platform continues to evolve; readers are encouraged to consult the latest documentation.