

AI Agents and Identity Platforms: Auditability over Powerful APIs

An MCP-based Gateway Design and Evaluation

1. Executive Summary

This paper summarizes a series of experiments on how to connect AI agents to an enterprise identity platform without losing control over scopes, auditability, and explainability.

Instead of wiring a chat-based AI agent directly into the administrative APIs of an identity platform, the experiments introduce an MCP-based gateway that sits between AI agents and the platform. The gateway enforces device-based authorization for AI clients, scope-based access control for each operation, and transaction-level audit logging that captures both the action and the AI's reasoning.

Concretely, the prototype gateway was implemented in front of an Okta tenant, but the design is intentionally vendor-neutral. The same pattern can be applied to other IDaaS platforms or directory services. The focus is not on any particular product feature but on the security and governance questions that arise when AI agents are allowed to propose or execute actions such as user suspension, password reset, or risk-based investigation.

Key elements of the design include:

- A clear separation between "read and analyze" capabilities and "write and act" capabilities, enforced by OAuth scopes and validated access tokens.
- A normalization layer that turns raw system logs into semantic events and risk assessments that both AI and humans can understand.
- Machine-learning-based anomaly detection and impossible-travel checks that provide additional signals for AI decision-making.
- Transaction-level audit logs for critical operations such as suspend and rollback, including an AI-generated reasoning string for each operation.

From a CISO perspective, the main conclusion is that when connecting AI agents to identity platforms, what matters most is not more powerful APIs but an explainable and defensible audit trail. The experiments show that such an audit trail can be enforced in practice, but also highlight policy questions that technology alone cannot answer, such as how much autonomy to give AI agents and where to draw the line between automation and human approval.

2. Background and Problem Statement

Generative AI chat tools are rapidly moving from experimentation to daily operational use. In many organizations, administrators already rely on AI assistants to draft emails, summarize incident reports, or query documentation. The next natural step is to let AI agents query security systems, suggest actions, and eventually trigger changes in production environments.

If this step is taken naively, identity platforms are among the most sensitive targets. An AI agent that can call administrative APIs for an identity system can in principle suspend or disable large numbers of user accounts, reset passwords or MFA factors at scale, modify group memberships and application assignments, and generate a high volume of investigative queries against sensitive logs.

Even if an organization never intends to give an AI agent full autonomy, the combination of natural-language prompts, complex back-end APIs, and limited human attention can easily lead to “accidental autonomy”. The AI suggests an action, the human approves it without fully understanding the impact, and the result is operational or security damage.

Traditional API security controls are not sufficient for this situation. An API gateway can enforce authentication, rate limits, and IP restrictions. It does not answer questions such as how much authority any single AI agent should have over identities, what a defensible audit trail looks like when the actor is not a human but an AI, and how an organization can reconstruct why a specific suspend or reset happened weeks or months later.

The experiments described in this paper were designed around a simple objective:

- Do not connect AI agents directly to the identity platform’s administrative APIs.
- Introduce a gateway that constrains what AI agents are allowed to do, normalizes and analyzes signals from logs, and records AI-driven actions in a way that is explainable to auditors.

In practice, this gateway was implemented as a set of MCP servers that AI clients such as Claude or ChatGPT can connect to. The back-end identity system in the experiments is Okta, but the architectural questions are general. The central question is how to let AI agents interact with identity operations at all while still satisfying least privilege, auditability, and accountability requirements.

3. Architecture Overview: MCP Gateway in Front of an Identity Platform

3.1 High-level Components

The overall architecture contains the following components.

- AI chat client
A conversational AI application that supports MCP as a way to call external tools. Examples include desktop clients or browser-based chat interfaces. The AI agent receives natural-language instructions from a human operator and decides which MCP tools to invoke.
- MCP gateway server
A custom server that exposes a set of MCP tools for identity operations and analysis. It terminates MCP connections from AI clients, validates tokens, enforces scopes, calls back-end services, and returns results to the AI agent.
- Identity platform
An enterprise identity platform such as Okta, Entra ID, or another IDaaS. In the experiments, an Okta tenant acts as the primary identity system.
- Orchestration services
Supporting components such as Okta Workflows, which execute user lifecycle operations, search

system logs, send notification emails, and encapsulate business logic behind HTTP flows.

- External analytical services

Additional services such as embedding APIs and GeoIP lookup services that support anomaly detection, similarity search, and impossible-travel analysis.

3.2 Data Flow

The typical data flow for an operation is as follows.

- A human operator issues a natural-language instruction to the AI chat client, such as "Check whether user@example.com has shown any suspicious activity in the last 24 hours, and suspend the account if the risk is high."
- The AI agent interprets the request and selects one or more MCP tools exposed by the gateway. It constructs parameters based on the prompt and the schema of the tools.
- The AI client calls the MCP gateway with the selected tool and parameters.
- The MCP gateway validates the access token presented by the AI client, checks scopes, and determines whether the call is permitted.
- If permitted, the MCP gateway calls the identity platform or associated services. For example, it may call Okta Workflows to obtain System Logs, compute risk assessments, or suspend a user.
- The results are returned to the AI agent through the MCP interface. The AI uses them to generate a response for the human operator and may optionally chain additional tool calls.
- For critical actions, the MCP gateway writes a transaction-level audit record that includes the AI's reasoning.

3.3 Layered Design

The gateway is structured in three logical layers.

- Account operations layer

Exposes tools for user suspension, reactivation, password resets, and user attribute retrieval. All write operations go through this layer and are routed to orchestrated HTTP flows in the identity platform's automation service.

- Log semantics and analysis layer

Exposes tools that query the identity platform's system logs, normalize events into a semantic schema, compute heuristic risk scores, run anomaly detection, and detect impossible travel. This layer does not modify identities and can be scoped to read-only access.

- Authorization and policy enforcement layer

Handles device authorization for AI clients, validates access tokens using JWKS and JWT checks, enforces scope-based access control for each tool, and generates authorization errors when requests exceed granted scopes.

This layered approach allows AI agents to use rich analytical capabilities without automatically gaining direct write access to user accounts. It also creates explicit seams where organizations can apply policy and governance.

3.4 Vendor-neutral Intent

The implementation described here uses Okta and Okta Workflows as concrete back-end services. However, the architectural pattern is intentionally vendor-neutral.

- The MCP gateway treats the identity platform as an abstract service with endpoints for user operations and log retrieval.
- The semantics layer defines a generic event schema with actors, targets, outcomes, and categories that would apply to most identity platforms.
- The authorization layer relies on standards such as OAuth, OIDC, JWT, and JWKS, which are supported across major IDaaS providers.

An organization can replicate this pattern in front of other identity systems as long as equivalent APIs for lifecycle operations and log access exist.

4. Implemented Capabilities and Test Scenarios

This section describes the main capabilities implemented in the MCP gateway and the scenarios used to test them.

4.1 Account Operation Capabilities

The account operations layer exposes tools that wrap user lifecycle operations behind explicit MCP interfaces.

- Suspend user
Suspends a user account based on a login identifier such as an email address. The tool requires the AI agent to supply an explanation string describing why suspension is appropriate. The gateway calls an HTTP flow in the automation service to perform the suspension and then records a transaction-level audit entry.
- Reactivate user and rollback
Reactivates a previously suspended user and links the new operation to the original suspension via a rollback reference. The tool takes the user login and the previous transaction identifier. The audit log records both the new transaction and the rollback relationship.
- Reset password
Triggers a password reset for a user via an automation flow. The tool is only available to AI clients with a write scope for user operations. Usage of this tool is treated as a high-impact action and can be restricted by policy.
- Read user attributes
Retrieves selected user attributes such as title, department, manager, or division. This allows the AI agent to understand the user's role and business context when analyzing behavior or proposing actions.

These tools were tested in scenarios where the AI agent first runs analysis tools, presents a risk assessment to the human operator, and then calls the suspend or reset tool if the operator approves.

4.2 Log Semantics and Risk Scoring

The log semantics and analysis layer focuses on turning raw system logs into structured risk assessments.

- Semantic event normalization

System Log entries are mapped into a normalized schema with fields such as actor, primary target, subject user, client information, event category, and outcome. This makes it easier for AI agents to reason about logs and for humans to interpret outputs.

- Heuristic risk scoring

For a given user and time window, the gateway aggregates normalized events and computes a risk level such as low, medium, high, or critical. Factors include the number of failed authentications, presence or absence of MFA, high-severity events, and recent password resets. The output includes a risk level, a human-readable summary of reasons, and suggested next actions.

These capabilities were tested using synthetic and real logs to confirm that obviously benign users receive low risk levels and clearly problematic patterns are flagged as high or critical.

4.3 Machine-learning-based Anomaly Detection

To supplement heuristic scoring, the gateway implements machine-learning-based anomaly detection.

- Feature extraction

Events are converted into simple numerical and categorical features per time window. Example features include hour of day, day of week, a binary flag for failures, a binary flag for MFA events, and a normalized severity measure.

- IsolationForest-based detection

An IsolationForest model is used to assign anomaly scores to time windows. The system can compute an average anomaly score, count the number of anomalous windows, and provide a coarse anomalous or normal label for the period.

Tests focused on whether obviously unusual patterns such as sudden spikes of activity or new combinations of event types were assigned higher anomaly scores than baseline behavior. The goal is not perfect detection but an additional signal that can appear in the analysis presented to the AI agent and the human operator.

4.4 Impossible-travel Detection

The gateway implements an impossible-travel detector to catch suspicious login patterns across geographic locations.

- IP geolocation

Login events with IP addresses are enriched with approximate latitude and longitude using a GeolP service.

- Speed calculation

For consecutive logins by the same user, the system calculates the distance between locations and divides by the time difference to estimate an implied travel speed.

- Flagging suspicious cases

If the implied speed exceeds a configurable threshold that is unrealistic for normal travel, the event pair is flagged as suspicious. The analysis tool returns details such as origin, destination, timestamps, and estimated speed.

Test scenarios included simulated rapid switches between distant locations to verify that the detector identifies implausible behavior without generating excessive noise for normal travel.

4.5 Behavioral Embedding and Similarity Search

To support explainability and comparison with past cases, the gateway maintains a simple vector store of behavioral summaries.

- Behavioral summaries

For each analysis run, the system constructs a textual summary describing the user's recent activity. This summary refers to event types, risk factors, and key observations.

- Embedding and storage

The summary is converted into an embedding vector using an embedding API and stored together with metadata in a JSONL file that acts as a local vector store.

- Similarity search

When a new case is analyzed, the system can search for the most similar past summaries and return them as context. This allows an AI agent to say whether the current case resembles previously observed benign or malicious patterns.

Tests confirmed that clearly similar patterns such as repeated login failures during a short window are clustered together, while distinctly different behaviors produce separated embeddings.

4.6 Test Scenarios

The implemented capabilities were exercised using several representative scenarios.

- Suspicious login pattern and suspension

A user shows an abnormal combination of failed logins, impossible travel, and high anomaly scores. The AI agent uses the analysis tools to build a risk assessment, proposes suspension to the operator, and calls the suspend tool upon approval. The operator later requests reactivation, and the rollback tool is used to restore access with audit linkage.

- Noisy but benign user

A user generates many low-severity events and occasional failed logins but no impossible travel and low anomaly scores. The analysis returns a medium or low risk level with recommendations for monitoring rather than immediate suspension. The AI agent is expected to surface this guidance instead of proposing aggressive actions.

- Geo-driven suspicious logins

Logs include alternating logins from distant regions in short time intervals. The impossible-travel detector flags specific event pairs, and the risk engine elevates the risk level even if total event counts are modest. The AI agent uses these signals to justify further investigation or targeted suspension.

These scenarios confirm that the gateway can both support high-impact actions and avoid overreacting in less clear-cut situations.

5. Security, Auditability, and Governance Model

This section describes how the gateway enforces security properties and what kind of audit trail it produces. It also outlines where technology stops and governance decisions begin.

5.1 Authentication and Device Authorization for AI Clients

AI clients connect to the MCP gateway as tools. The gateway requires them to acquire access tokens through a device authorization grant.

- The AI client requests device authorization for a set of tools.
- The gateway derives the union of required scopes for these tools and initiates a device authorization flow with the identity platform's authorization server.
- The user or administrator visits the verification URL, enters the user code, and explicitly approves the requested scopes.
- The AI client polls the token endpoint and receives an access token if and only if the user approves.

This flow ensures that the authority of an AI agent is bounded by an explicit human approval that is visible as an OAuth consent decision. The AI cannot silently gain scopes beyond what a human has granted.

5.2 Scope-based Access Model

Each MCP tool is associated with one or more OAuth scopes that are required to invoke it.

- Read scopes
Scopes for reading logs and user attributes, such as a logs read scope and a user read scope.
- Analysis scopes
Scopes for running risk assessments and anomaly detection, which might require both log read and analysis scopes.
- Write scopes
Scopes for performing changes to user accounts, such as a user write scope for suspension and password reset.

The MCP gateway inspects the scopes contained in the access token that the AI client presents. If the token does not contain the required scopes for a tool, the gateway returns an authorization error and does not call the identity platform or automation services.

This approach gives organizations a straightforward way to separate analysis and action. AI agents can be granted log and analysis scopes in early phases and can be denied write scopes until governance questions are resolved.

5.3 JWT Validation and Separation of Privileges

The gateway validates access tokens as JSON Web Tokens before recognizing any scopes.

- It fetches JSON Web Key Sets from the authorization server and caches them.
- It locates the appropriate key using the key identifier and verifies the signature using RS256.
- It checks issuer and audience claims against expected values for the gateway.
- It enforces expiration and not-before constraints.

Alongside AI tokens, the gateway uses separate secrets or tokens to call automation flows in the identity platform. These tokens are intended to be separated by purpose. Suspension flows, reactivation flows, log retrieval flows, and notification flows can each have distinct tokens with limited privileges. This separation reduces the impact of misconfiguration or credential leakage.

5.4 Transaction-level Audit Logs

Critical operations such as suspension and reactivation generate transaction-level audit records in a JSON Lines file or equivalent storage.

Each record includes fields such as:

- Transaction identifier.
- Timestamp in a standard format.
- Operation type, such as suspend or reactivate.
- Target user identifier.
- Status, such as success or error.
- Link to a previous transaction identifier for rollback operations.
- AI reasoning text that explains why the action was performed.

These records provide a precise trail of what the gateway did on behalf of AI agents and when. They are separate from the identity platform's native logs and focus specifically on AI-driven actions and their justifications.

5.5 Capturing AI Reasoning

For critical actions, the gateway requires an AI reasoning string as input. The AI agent must provide a textual explanation of why the action is appropriate based on the available evidence.

The reasoning string is stored in the audit log alongside the transaction. This supports explainability by making it possible to reconstruct not only what happened but also why the AI believed the action was justified at the time.

This mechanism cannot guarantee that the AI's reasoning is correct, but it forces explicit articulation and preserves that articulation for later review.

5.6 Mapping Analysis Outputs to Actions

The gateway links analytical outputs and actions through metadata and identifiers.

- Analysis tools return structured risk assessments, anomaly summaries, and impossible-travel findings.
- The AI agent typically uses these outputs as input to its reasoning when proposing or executing actions.
- In audit or investigation, the transaction identifier for an action can be mapped back to the time range and user for which analysis was performed.
- The organization can retrieve the original logs and re-run analysis to verify whether the decision was supported by the data.

This mapping supports a consistent story from raw events to analysis to AI decision and finally to action taken by the gateway.

5.7 Audit Support and Remaining Governance Questions

The implemented design supports several audit-oriented use cases.

- Reconstruction of the sequence of AI-driven actions on a user account.
- Examination of the stated reasons for a specific suspension or reactivation.
- Comparison of AI decisions across time and across users.
- Verification that AI agents did not perform actions outside their granted scopes.

At the same time, several questions remain in the domain of governance rather than technology.

- How much autonomy should AI agents have for high-impact actions such as suspension and password reset.
- When human approval should be mandatory and how such approvals should be recorded.
- How long AI action logs, analysis artifacts, and behavioral embeddings should be retained.
- What portions of log and user data may be sent to external services under the organization's privacy and regulatory constraints.

The gateway provides the technical hooks to implement decisions in these areas, but it does not dictate the decisions themselves.

6. Lessons Learned and Recommendations for CISOs

The experiments with the MCP gateway in front of an identity platform produced several practical lessons and suggest concrete recommendations for CISOs.

6.1 Lessons Learned

- It is technically feasible to constrain AI agents using a gateway.
An MCP-based gateway with device authorization, scopes, and JWT validation can effectively bound what AI agents can do. The design can prevent write operations when only read or analysis scopes are granted.
- The bottleneck is not the API but the audit trail.
Most modern identity platforms already have rich APIs. The real challenge is being able to show, months later, why a specific AI-driven action was taken and which evidence it was based on.
- Read and analyze permissions should be strictly separated from act permissions.
Combining log access, analysis, and account modification under a single set of scopes makes it difficult to enforce least privilege. The gateway pattern makes this separation explicit and enforceable.
- AI reasoning is a useful audit artifact.
Forcing AI agents to provide a reasoning string for critical actions produces an additional layer of visibility. Auditors can compare AI reasoning with underlying evidence and with organizational policies.
- Vector-based similarity is promising for reviewing AI behavior.
Embedding behavioral summaries and using similarity search provides a way to cluster incidents and compare AI responses to similar situations. This opens a path to audit not only human behavior but AI behavior itself.

6.2 Recommendations for CISOs

- Start with analysis-only access for AI agents.
In early phases, constrain AI agents to log retrieval and analysis tools. Deny write scopes for account operations until the organization has confidence in the analysis layer and has defined policies for escalation.
- Define clear boundaries for AI autonomy versus human approval.
For each type of action such as suspension, reactivation, and password reset, decide whether AI agents may execute autonomously, may only propose actions, or are not allowed to touch the action at all. Document these decisions and enforce them through scopes and workflow design.
- Treat AI action logs as first-class audit artifacts.
Ensure that transaction-level AI action logs are stored, protected, and retained with the same care as other security logs. Incorporate them into incident review and audit processes.
- Limit and document data sent to external services.
Review what data the gateway sends to embedding or GeoIP services. Apply masking or pseudonymization where possible, and maintain documentation describing what is sent and why.
- Plan for periodic review of AI agent behavior.
Use the collected logs and behavioral summaries to review AI decision patterns. Check for overuse of high-impact actions, inconsistent responses to similar cases, or bias in the way risk is assessed.

By adopting these recommendations, organizations can move beyond abstract concerns about AI risk and begin to define concrete controls that make AI-assisted identity operations governable.

7. Appendix

7.1 Example MCP Tool List

The following is an illustrative list of MCP tools exposed by the gateway.

- `suspend_user`
Suspends a user account based on a login identifier and an AI reasoning string.
- `reactivate_user`
Reactivates a suspended user and links the operation to a previous suspension transaction.
- `reset_password`
Triggers a password reset for a user via an automation flow.
- `read_user`
Retrieves selected user attributes for context.
- `search_logs`
Retrieves recent system logs for a user or filter and returns normalized events.
- `analyze_identity_state`
Runs heuristic risk assessment, anomaly detection, and impossible-travel checks for a user and returns a structured risk summary.

- **detect_anomalies_ml**
Runs machine-learning-based anomaly detection on log features.
- **detect_impossible_travel**
Flags suspicious login pairs based on inferred travel speed.
- **notify_user**
Sends notification emails through an automation flow using AI-generated subject and body.
- **get_ai_audit_log**
Returns recent AI-driven action records from the transaction-level audit log.

7.2 Example Audit Log Entry

The following JSON snippet illustrates an audit log record for a suspension.

```
{  
  "transaction_id": "b7b5f6b8-3b8b-4e9e-84b7-9c5a8a9f8b13",  
  "timestamp": "2025-11-26T09:32:15Z",  
  "operation": "suspend_user",  
  "user_login": "user@example.com",  
  "status": "success",  
  "rollback_of": null,  
  "ai_reasoning": "Multiple failed logins from unusual locations and an  
impossible-travel alert indicate that this account is likely compromised.  
Suspending access while investigation continues is the safest option.",  
  "actor_client": "ai-mcp-client-1"  
}
```