databricks

WHITEPAPER

Databricks Agentic AI Security

Extension to the Databricks AI Security Framework

Version 1.0

databricks

Table of Contents

01

Executive Summary

As enterprises move beyond singular Large Language Models (LLMs) toward Agentic AI, the future of work is shifting from static information retrieval to autonomous action. Agentic systems—capable of reasoning, planning, and executing complex tasks with minimal human intervention—promise unprecedented efficiency and innovation. However, this leap in autonomy introduces a new frontier of security and privacy risks. Unlike traditional models that simply process and output text, AI agents possess the ability to interact with external tools, manipulate data, and execute code, significantly expanding the potential attack surface.

Databricks has extended the **Databricks AI Security Framework (DASF)** to address these emerging challenges. Building upon the original 12 foundational components of AI systems, this extension introduces the 13th Component: **Agentic AI**, adding 35 new technical security risks and 6 new mitigation controls (DASF 68–73) and bringing the full DASF framework to 97 risks and 73 controls across 13 components. This update provides a holistic strategy to mitigate the unique risks associated with autonomous agents, including **Memory Poisoning (Risk 13.1)**, **Intent Breaking & Goal Manipulation (Risk 13.6)**, **Tool Misuse (Risk 13.2)**, and vulnerabilities introduced by emerging standards such as the Model Context Protocol (MCP).

This white paper outlines the specific security architectural considerations for Agentic systems, detailing how agents interact with existing data and model components. It identifies specific technical security risks—ranging from agent hijacking to cascading hallucination attacks—and maps them to actionable defense-in-depth controls. By integrating these new findings into the DASF, we aim to empower security teams, ML practitioners, risk, and governance leaders to collaborate effectively, ensuring that the adoption of Agentic AI is both transformative and secure. We have also updated the DASF compendium (**Google sheet**, **Excel**) to include these new risks and controls, mapping them to industry standards to facilitate immediate operationalization, and are cataloged as DASF v 3.0 against “DASF Revision” column.

02

Introduction

As AI evolves from predictive Machine Learning and Generative Models to autonomous agents, the security frontier is fundamentally altered. This shift from static output to operational action necessitates a enhanced defense-in-depth approach to governance and risk mitigation. To help us understand this, let's start with some definitions.

> **Agentic AI (noun):** The architectural paradigm shift where AI evolves from passive content generation to active coordination, enabling systems to reason, plan, and take actions on data and tools to reliably achieve a wide variety of goals.

While GenAI revolutionized how we generate content, Agentic AI revolutionizes how we automate and execute tasks. Gone are the days of one-shot model responses. Organizations are deploying **Compound AI Systems** where models function as Agents that can perceive their environment, reason through complex problems, retain context through memory, and act upon the world using defined tools and APIs.

> **Agent (noun):** An autonomous software system that perceives its environment, uses tools and data to reason about and how to achieve a user-defined goal, and then takes actions on the user's behalf.

This shift brings opportunities but also fundamentally changes the risk landscape. In a traditional GenAI context, the primary risks revolved around training data leakage or biased output. In an Agentic context, these risks can impact downstream deterministic systems allowing an attacker to take unintended action. An agent that has permission to access data, execute SQL queries, call external APIs, or modify code repositories possesses a level of agency that requires rigorous security controls and governance. What happens if an agent is tricked into "forgetting" its safety instructions via **Memory Poisoning (Risk 13.1)**? What if an attacker manipulates the agent's planning logic to execute unauthorized actions via **Intent Breaking & Goal Manipulation (Risk 13.6)**? Or, crucially, what if the tools the agent relies on—connected via standards like the Model Context Protocol (MCP)—are themselves compromised via **Tool Poisoning (Risk 13.18)** and/or **Malicious Server Connection (Risk 13.26)**?

To help organizations navigate this complex new terrain, this white paper introduces the Agentic AI Extension to the **Databricks AI Security Framework (DASF)**. It builds upon our previous work by expanding the holistic DASF approach to cover the lifecycle of autonomous agents. We define the architecture of an agent, break down its sub-components—Planning, Memory, and Tools—and analyze the specific threat vectors applicable to each.

This extension is intended for the same collaborative audience as the original DASF: security teams, data practitioners, and business leaders. It provides a structured language to discuss agent autonomy, tool permissions, and safety guardrails without requiring deep expertise. Every risk is paired with mitigations that are concrete and actionable. By applying these guidelines, organizations can move from experimental agent deployments to robust, production-grade Agentic systems with confidence.

2.1 DASF 2.0 Refresher

The **Databricks AI Security Framework (DASF)** provides a structured approach to understanding and mitigating security risks across AI systems. The framework is designed for collaborative use between data, AI, and security teams throughout the AI lifecycle.

DASF is centered around:

- Foundational components of a data-centric AI system organized across four operational stages
- Technical security risks that arise across these components
- Mitigation controls that help address these risks, mapped to industry standards including MITRE, OWASP, NIST, ISO, and HITRUST

The second version of DASF, **released** February 2025, documents 12 foundational system components, 62 technical security risks, and 67 mitigation controls.

Below we will enumerate the system components covered in DASF 2.0. For a comprehensive treatment of each risk, its potential business impact, and the specific controls available to address it, refer to **the full DASF whitepaper**. Also consider the compendium (**Google sheet**, **Excel**), containing the same content in usable Excel or Google Sheet – now updated with this agentic extension.

DASF System Components

Building a common language for what constitutes an AI System has proven to be one of the most valuable tools to bring different stakeholders together within an enterprise. From engineers to executives, few have visibility across the lifecycle, and understanding the full scope aligns expectations.

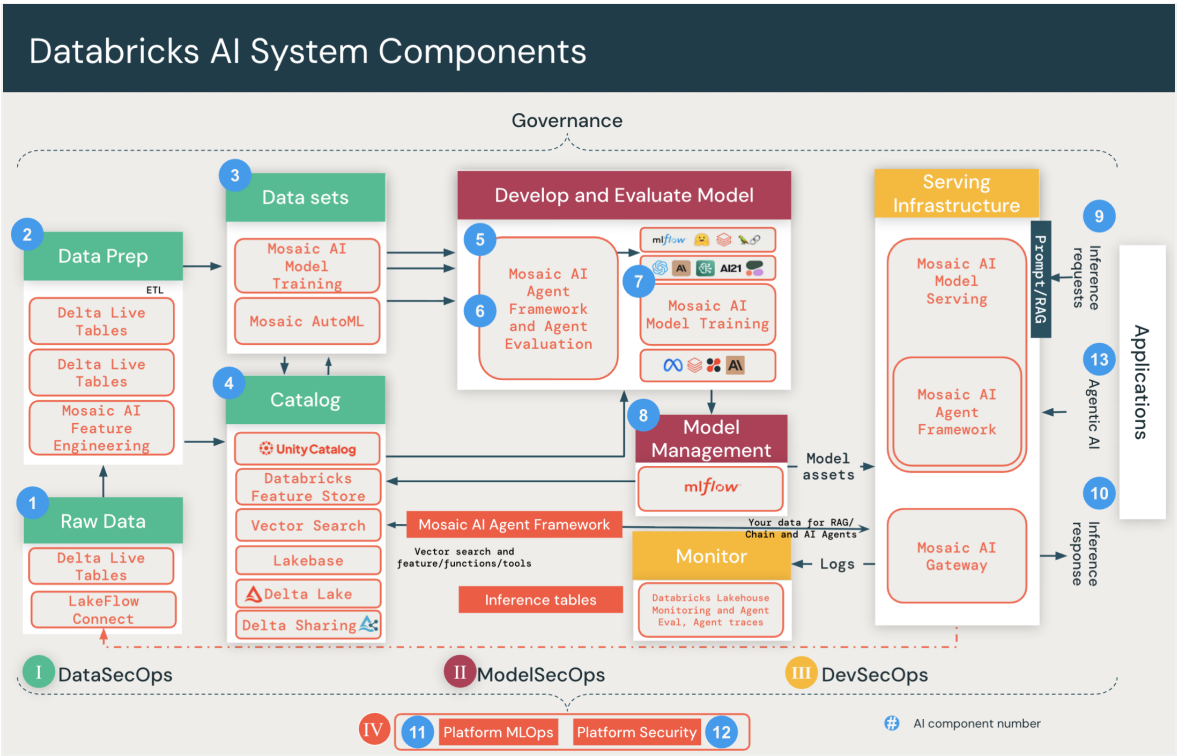


Diagram of all AI System Components covered in the full Databricks AI Security Framework

Here is the enumeration of system components included in DASF 2.0

I. Data Operations (DataSecOps)

1. **Raw Data**: Enterprise data, metadata, and operational data (structured, semi-structured, unstructured)
2. **Data Prep**: Cleaning, exploratory data analytics (EDA), featurization, feature extraction
3. **Datasets**: Training, validation, and test datasets
4. **Catalog (Data and AI Governance)**: Features, indexes, models, tools, agents, managed in a unified catalog

II. Model Operations (ModelSecOps)

1. **Algorithms**: Training algorithms for building models
2. **Evaluation**: Model evaluation and testing
3. **Models**: Custom models, external models, fine-tuning and pretrained models, foundation models
4. **Model Management**: Model assets, versioning, lifecycle management

III. Model Deployment and Serving (DevSecOps)

1. **Serving Infrastructure / Inference Requests**: Model serving, AI gateway, prompts/RAG, inference requests
2. **Inference Response**: Model outputs, logs, monitoring

IV. Operations and Platform

1. **MLOps (Machine Learning Operations)**: New ML and RLHF data pipelines, monitoring
2. **Platform Security (Data and AI Platform Security)**: Underlying infrastructure and platform controls

With this extension, we've added a 13th component: **Agentic AI**

What's new in the Databricks AI Security Framework 3.0

DASF v3 adds component 13, Agentic AI: 35 new technical security risks organized across three sub-components:
- 13A: Agents - Core (Risks 13.1–13.15). Risks targeting the agent's cognitive loop (memory, planning, multi-agent dynamics)
- 13B: Agents - Tools: MCP Server (Risks 13.16–13.25). Risks in the MCP server tool interface
- 13C: Agents - Tools: MCP Client (Risks 13.26–13.35). Risks in the agent's client-side connection layer

There are also seven new mitigation controls (DASF 68–73) mapped to 10+ industry standards including MITRE ATLAS, MITRE ATT&CK, OWASP, NIST CSF 2.0, and ISO 42001.

The updated compendium (**Google sheet**, **Excel**) with Component 13 risk-to-control mappings for all 73 existing and new controls. All new risk and controls introduced are cataloged as DASF v3.0 against "DASF Revision" column in the compendium.

2.2 Understanding the Agentic Shift

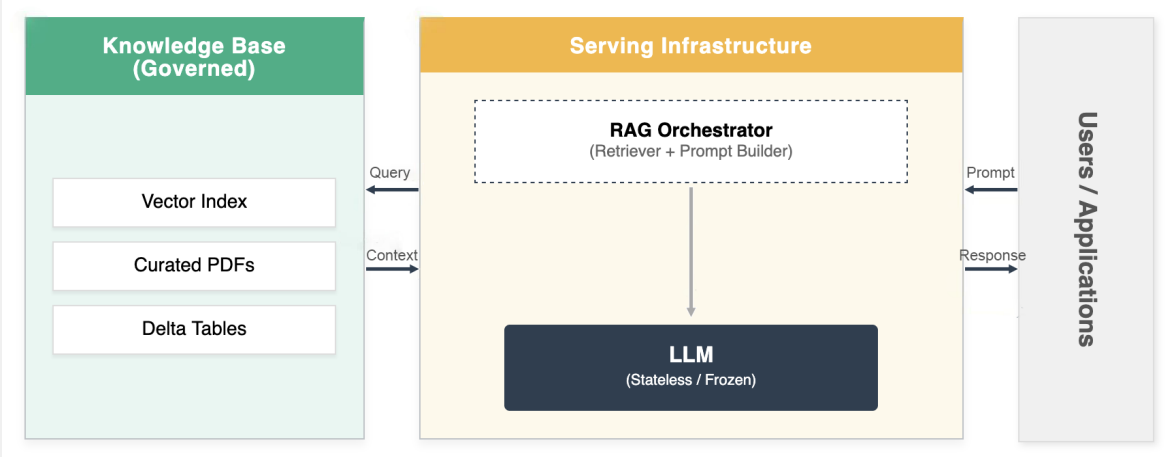
To properly assess the security posture of an Agentic system, it is necessary to distinguish its operational flow from that of a traditional Generative AI application. To illustrate this, we will look at the transition from standard Retrieval Augmented Generation (RAG) to an Agentic workflow. This shift alters the potential attack surface, requiring data and security teams to look beyond the model. The security teams now need to track everything around it: the agent, its tools, and the data platform beneath them. Furthermore, teams need to focus on what the agent did, where, and whether it was ever supposed to.

Traditional RAG Workflow

In a standard RAG architecture, the system operates on a linear, read-only basis. When a user submits a query, the system acts as a sophisticated search engine: it retrieves relevant text chunks from a pre-determined, governed dataset (such as a vector index) and passes them to the LLM as context.

From a security perspective, the boundaries are static. The model is constrained to the specific context provided by the retrieval mechanism. It generally lacks the ability to independently explore other data sources or execute commands. Consequently, the impact

of any unauthorized action (e.g., data exfiltration) is confined to the retrieved dataset, and the risk of privilege escalation is minimal.



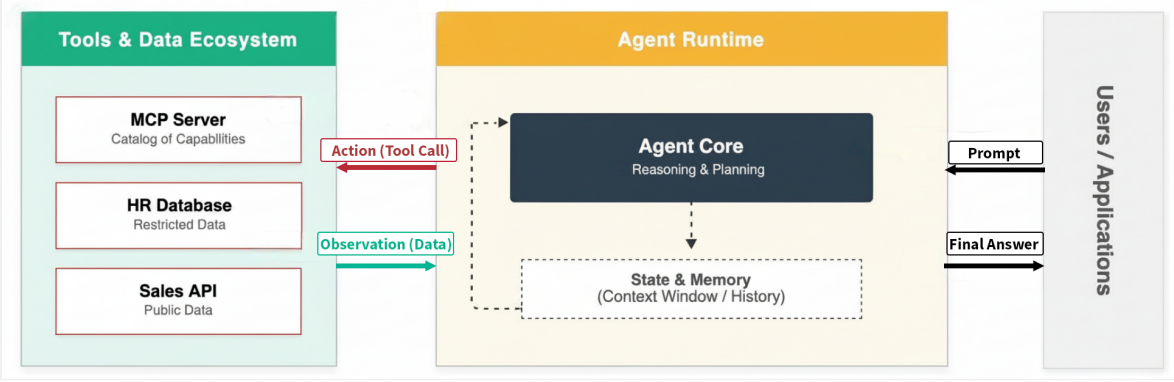
A Linear RAG Architecture

Agentic AI Workflow

In contrast, an Agentic system operates as a dynamic execution loop. The developer may not hard-code a specific dataset for every interaction; instead, they can provision the agent with a set of Tools: interfaces to APIs, SQL warehouses, or Python interpreters and a high-level objective.

When a user engages an agent, the model initiates a discovery process. It evaluates the user's intent and determines which tools are required to solve the problem. This often involves a multi-step reasoning loop:

1. **Plan:** The agent breaks the request into sub-tasks.
2. **Act:** The agent selects a tool (e.g., "Query Sales Database") and executes it.
3. **Observe:** The agent analyzes the output. If the data is insufficient, it may decide to call a different tool (e.g., "Check Inventory API").
4. **Refine:** The agent continues this loop until the task is complete.



An Agentic Loop RAG Architecture

Risk Evolution

Unlike a RAG system that simply uses the data supplied or reports "Data not found," an agent is designed to actively search for a solution. If an agent has access to a broad set of tools, it may attempt to traverse data paths that were not intended for the current user, turning a helpful feature into a potential vector for privilege escalation.

This introduces a critical new risk category: Discovery and Traversal. Because an agent is designed to find solutions autonomously, it may traverse data paths and tool interfaces that were never intended for the requesting user. It isn't exploiting a bug, instead doing exactly what it was built to do. Without the right security controls, this behavior becomes a vector for privilege escalation — where a user effectively inherits the agent's permissions rather than their own.

The Lethal Trifecta in Agentic Systems

This escalation of privilege is not theoretical. It is a structural vulnerability inherent to autonomous systems described by security researchers as the **"Lethal Trifecta."** While traditional GenAI risks focus on what the model might say (e.g., offensive output), Agentic AI risks focus on what the model might do. The risk profile spikes critically when three specific conditions converge within the Agent Core:

1. **Access to Privileged Information:** The agent has retrieval access to private data (e.g., the "Knowledge Base" or restricted "HR Database").

2. **Ingestion of Untrusted Input:** The agent processes data from outside the trust boundary (e.g., summarizing an external website, reading incoming emails, or analyzing third-party logs).

3. **Capability to Act:** The agent is equipped with Tools or MCP connections that can modify state (e.g., "Action (Tool Call)" capabilities like sending emails, executing SQL, or modifying code).

In a static RAG system, an injection attack most commonly corrupts the text output. In an Agentic system defined by this trifecta, an Indirect Prompt Injection embedded in untrusted data can hijack the agent's Planning logic, forcing it to use its Privileged Access and Action capabilities against the enterprise. This turns the agent into a "confused deputy" that performs authorized actions with malicious intent.

The security implications of this architecture are best understood through a simplified example of "Tool Misuse" and "Identity Propagation failure." Consider an internal enterprise agent equipped with two tools:

1. **Sales Tool (Standard Access):** Accesses general revenue tables.
2. **HR Metrics Tool (Restricted Access):** Accesses aggregate headcount planning data.

The Scenario: A user with standard access rights prompts the agent: "Summarize the Q3 revenue trends, and provide context on how executive compensation impacts our operating margins."

The Risk: When excessive permissions, autonomous execution, and unvalidated input converge, the result is an attack scenario that can be difficult to detect and prevent.

1. **Step 1 — Excessive Permissions (Leg 1):** The agent is provisioned with access to both the Sales Tool and the HR Metrics Tool to serve a broad range of users and tasks. No single permission grant seems unreasonable in isolation — but collectively, the agent holds more access than any individual user it serves should be able to exercise.
2. **Step 2 — Autonomous Execution Without Checkpoints (Leg 2):** The agent identifies that answering the query requires both tools and proceeds to query them sequentially without pausing for a human approval step. This end-to-end autonomy is intentional — it's what makes the agent useful. But it also means there is no friction point at which a human could catch a problematic action before it completes.
3. **Step 3 — Untrusted Input as the Trigger (Leg 3):** The original user prompt itself acts as the untrusted input. The phrase "executive compensation impacts our operating margins" is a natural-sounding business question, but it is unvalidated input that directly drives the agent's tool selection logic. The agent has no mechanism to evaluate whether the intent behind the query is within the requesting user's authorization scope — it only evaluates whether it, the agent, can technically fulfill the request.
4. **Step 4 — The Lethal Combination:** Because the agent holds broad tool access (Leg 1), operates without a mid-execution human checkpoint (Leg 2), and acts on unvalidated user intent (Leg 3), it successfully retrieves restricted compensation data and presents it to a user who was never authorized to see it — wrapping it in a clean, professional summary with no indication of a security violation.

Removing any one leg of the trifecta mitigates the risk. Scope the agent's permissions to the requesting user's entitlements, and the HR Metrics Tool returns nothing. Introduce a confirmation step before accessing restricted tools, and a human catches it. Validate user intent against authorization policy before tool selection, and the query never reaches the restricted tool. The breach is not caused by any single flaw — it is caused by the simultaneous presence of all three conditions. This update to DASF addresses each leg of the trifecta directly: **DASF 64** (Limit access from AI models and agents), **DASF 57** (Use attribute-based access controls (ABAC)), **DASF 5** (Control access to data and other objects), **DASF 58** (Protect data with filters and masking), removing Leg 1. **DASF 66** (Use Human-in-the-loop feedback), **DASF 62** (Implement network segmentation) limiting tool execution scope removing Leg 2. **DASF 73** (Register prompts) versions and validates system prompt intent before execution, **DASF 54** (Implement AI guardrails), **DASF 37** (Set up inference tables for monitoring and debugging models) reducing the blast radius and giving visibility of Leg 3. Together, these controls implement a structural defense-in-depth posture specific to autonomous Agentic systems.

2.3 Agentic AI Ecosystem

The true power of Agentic AI lies in interoperability. For an agent to be useful, it must connect to a diverse array of data sources and enterprise tools. Historically, this required bespoke integrations for every new tool, creating a fragmented landscape that was difficult to scale and nearly impossible to secure.

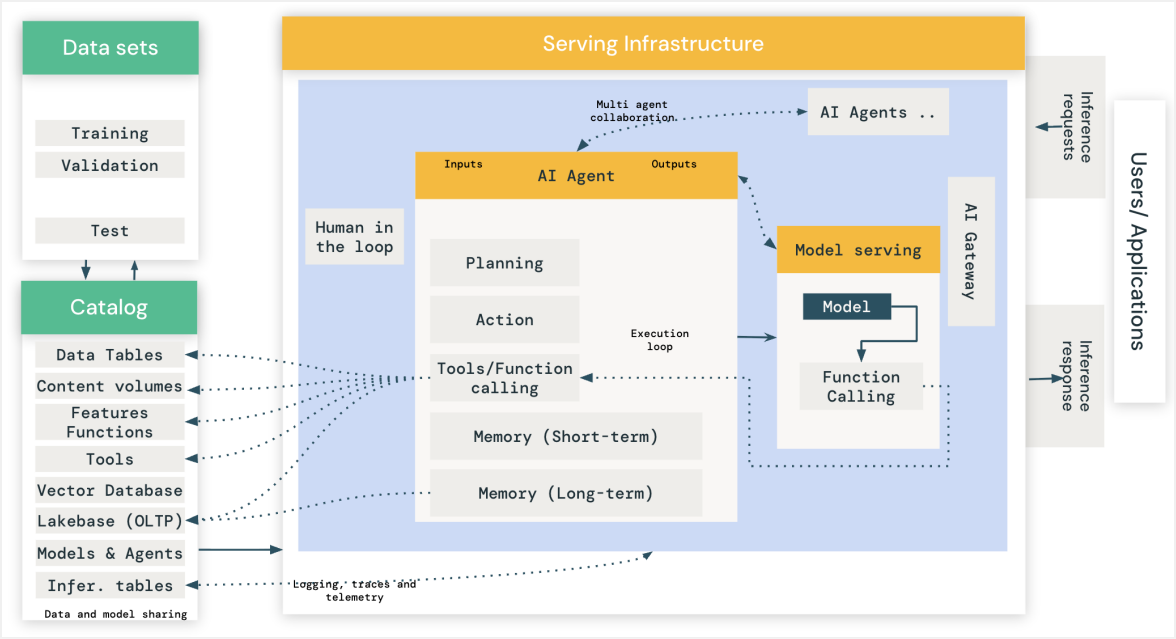
The **Model Context Protocol (MCP)** is emerging as a primary standard to solve this connectivity crisis because its architecture creates a clearly defined security boundary. Much like USB-C provided a universal hardware interface, MCP provides a standard open protocol for connecting AI models to data and tools. This standardization allows developers to build a tool once, be it a Google Drive connector or a SQL interface, and have it work across any MCP-compliant agent. By analyzing the interactions between the **Agent Core** (acting as the MCP Host), the **MCP Client** (the connection layer), and the **MCP Server** (the tool interface), we can map specific DASF controls to the exact points where data crosses a trust boundary.

That said, MCP is a connectivity standard, not a governance framework. Competing approaches like A2A, and OpenAPI implementations continue to evolve, but none of them fully solve for enterprise governance out of the box. Enterprises adopting MCP today are largely building the governance layer themselves: permission boundaries, identity propagation, and audit trails are not natively provided. MCP is the right foundation. It is not the finished structure. This is precisely why this DASF enhancement introduces three new sub-components within Component 13: 13A (Agents: Core), addressing the agent's cognitive loop; 13B (Agents: Tools — MCP Server), addressing the server-side tool interface; and 13C (Agents: Tools — MCP Client), addressing the client-side connection layer. The following section enumerates the specific threat vectors that arise within each sub-component and the controls available to address them.

03 Risks in Agentic AI System Components

In the original DASF, we defined 12 foundational components of a generic data-centric AI system, covering everything from raw data ingestion to model serving. With the rise of autonomous systems, we are formally introducing a 13th Component: Agentic AI.

While Agentic AI interacts heavily with the underlying model serving and data platform layers, it introduces a distinct logic loop of Perception, Brain (Reasoning/Planning), Memory, and Action. This loop operates on top of the traditional stack and creates new entry points for attackers that do not exist in standard predictive or generative pipelines.
The risks in this new component are categorized not just by the agent itself, but by the ecosystem it inhabits, specifically focusing on the agent's logic and its interfaces (tools).



Foundational components of an Agentic AI system

The Agentic Risk Landscape:

1. **The Agent Core (The Brain & Memory):** Risks here target the agent's cognitive processes. Attackers may attempt **Memory Poisoning (Risk 13.1)** to introduce false context that alters current or future decision-making, or **Intent Breaking & Goal Manipulation (Risk 13.6)**, where the agent is coerced into deviating from its objective. Because agents often operate in multi-turn loops, **Cascading Hallucination Attacks (Risk 13.5)** can cause a minor error to compound into a destructive action.

2. **The Tooling Interface (MCP & Actions):** Agents interact with the outside world through tools, increasingly standardized via the Model Context Protocol (MCP). This introduces a client-server dynamic between the agent (Client) and its tools (Server).
    - **MCP Server Risks:** Attackers may deploy **Tool Poisoning (Risk 13.18)** or exploit **Prompt Injection (Risk 13.16)** vulnerabilities within the tool definitions themselves to bypass security controls.
    - **MCP Client Risks:** If the agent (acting as the client) connects to a **Malicious Server Connection (Risk 13.26)** or fails to validate server responses, it risks **Client-Side Code Execution (Risk 13.32)** or **Client-Side Data Leakage (Risk 13.30)**.
3. **Inter-Agent Dynamics:** As systems scale, agents will communicate with other agents. This creates risks of **Agent Communication Poisoning (Risk 13.12)** and the emergence of **Rogue Agents in Multi-Agent Systems (Risk 13.13)** that operate outside monitoring boundaries.

By understanding the agent not just as a model, but as a system of Memory, Planning, and Tools, organizations can apply targeted controls to the specific sub-components that are most vulnerable.

3.1 Agents — Core

RISK/DESCRIPTION	MITIGATION CONTROLS
Agents — Core 13.1 **Memory Poisoning** Memory Poisoning involves exploiting an AI's memory systems, both short and long-term, to introduce malicious or false data and exploit the agent's context. This can lead to altered decision-making and unauthorized operations.	DASF 1 SSO with IdP and MFA DASF 2 Sync users and groups DASF 3 Restrict access using IP access lists DASF 4 Restrict access using private link DASF 5 Control access to data and other objects DASF 24 Control access to models and model assets DASF 31 Secure model serving endpoints DASF 34 Run models in multiple layers of isolation DASF 37 Set up inference tables for monitoring and debugging models DASF 46 Store and retrieve embeddings securely DASF 54 Implement AI guardrails DASF 55 Monitor audit logs DASF 64 Limit access from AI models and agents DASF 65 Implement end-to-end AI traceability DASF 67 Federate authentication DASF 71 Log and register AI agents DASF 72 Securely store and reuse agent state

RISK/DESCRIPTION	MITIGATION CONTROLS
Agents — Core 13.2 **Tool Misuse** Tool Misuse occurs when attackers manipulate AI agents to abuse their integrated tools through deceptive prompts or commands, operating within authorized permissions. This includes Agent Hijacking,where an AI agent ingests adversarial manipulated data and subsequently executes unintended actions, potentially triggering malicious tool interactions. For more information on Agent Hijacking see: https://www.nist.gov/news-events/news/2025/01/technical-blog-strengthening-ai-agent-hijacking-evaluations	DASF 1 SSO with IdP and MFA DASF 2 Sync users and groups DASF 3 Restrict access using IP access lists DASF 4 Restrict access using private link DASF 5 Control access to data and other objects DASF 24 Control access to models and model assets DASF 31 Secure model serving endpoints DASF 34 Run models in multiple layers of isolation DASF 37 Set up inference tables for monitoring and debugging models DASF 54 Implement AI guardrails DASF 55 Monitor audit logs DASF 56 Restrict outbound connections from models DASF 64 Limit access from AI models and agents DASF 65 Implement end-to-end AI traceability DASF 67 Federate authentication DASF 71 Log and register AI agents DASF 73 Register prompts
Agents — Core 13.3 **Privilege Compromise** Privilege Compromise arises when attackers exploit weaknesses in permission management to perform unauthorized actions. This often involves dynamic role inheritance or misconfigurations, including cross-agent privilege delegation where one agent inherits or escalates permissions it was not explicitly authorized to hold.	DASF 1 SSO with IdP and MFA DASF 2 Sync users and groups DASF 3 Restrict access using IP access lists DASF 4 Restrict access using private link DASF 5 Control access to data and other objects DASF 24 Control access to models and model assets DASF 31 Secure model serving endpoints DASF 34 Run models in multiple layers of isolation DASF 37 Set up inference tables for monitoring and debugging models DASF 54 Implement AI guardrails DASF 55 Monitor audit logs DASF 56 Restrict outbound connections from models DASF 64 Limit access from AI models and agents DASF 65 Implement end-to-end AI traceability DASF 67 Federate authentication DASF 71 Log and register AI agents

RISK/DESCRIPTION	MITIGATION CONTROLS
Agents — Core 13.4 **Resource Overload** Resource Overload targets the computational, memory, and service capacities of AI systems to degrade performance or cause failures, exploiting their resource-intensive nature.	DASF 31 Secure model serving endpoints DASF 34 Run models in multiple layers of isolation DASF 36 Set up monitoring alerts DASF 39 Platform security — Incident Response Team DASF 55 Monitor audit logs DASF 60 Rate limit number of inference queries
Agents — Core 13.5 **Cascading Hallucination Attacks** These attacks exploit an AI's tendency to generate contextually plausible but false information, which can propagate through systems and disrupt decision-making. This can also lead to destructive reasoning affecting tools invocation.	DASF 31 Secure model serving endpoints DASF 34 Run models in multiple layers of isolation DASF 36 Set up monitoring alerts DASF 37 Set up inference tables for monitoring and debugging models DASF 39 Platform security — Incident Response Team DASF 45 Evaluate models DASF 49 Automate LLM evaluation DASF 54 Implement AI guardrails DASF 60 Rate limit number of inference queries DASF 65 Implement end-to-end AI traceability DASF 66 Use Human-in-the-loop feedback

RISK/DESCRIPTION	MITIGATION CONTROLS
Agents — Core 13.6 **Intent Breaking & Goal Manipulation** This threat exploits vulnerabilities in an AI agent's planning and goal-setting capabilities, allowing attackers to manipulate or redirect the agent's objectives and reasoning. One common approach is Agent Hijacking, where an AI agent ingests adversarial manipulated data and subsequently executes unintended actions aligned with the attacker's goals rather than the user's intent.	DASF 1 SSO with IdP and MFA DASF 2 Sync users and groups DASF 3 Restrict access using IP access lists DASF 4 Restrict access using private link DASF 5 Control access to data and other objects DASF 24 Control access to models and model assets DASF 31 Secure model serving endpoints DASF 34 Run models in multiple layers of isolation DASF 37 Set up inference tables for monitoring and debugging models DASF 49 Automate LLM evaluation DASF 54 Implement AI guardrails DASF 55 Monitor audit logs DASF 56 Restrict outbound connections from models DASF 64 Limit access from AI models and agents DASF 65 Implement end-to-end AI traceability DASF 66 Use Human-in-the-loop feedback DASF 67 Federate authentication DASF 71 Log and register AI agents DASF 73 Register prompts

RISK/DESCRIPTION	MITIGATION CONTROLS
Agents — Core 13.7 **Misaligned & Deceptive Behaviors** AI agents executing harmful or disallowed actions, triggered by an input that exploits the agent's reasoning capabilities and deceptive responses to meet an objective. This includes behaviors where the agent produces outputs that are inconsistent with its declared purpose, or actively conceals its reasoning and actions from operators and users.	DASF 1 SSO with IdP and MFA DASF 2 Sync users and groups DASF 3 Restrict access using IP access lists DASF 4 Restrict access using private link DASF 5 Control access to data and other objects DASF 24 Control access to models and model assets DASF 31 Secure model serving endpoints DASF 34 Run models in multiple layers of isolation DASF 37 Set up inference tables for monitoring and debugging models DASF 38 Platform security — penetration testing, red teaming, bug bounty and vulnerability management DASF 45 Evaluate models DASF 49 Automate LLM evaluation DASF 54 Implement AI guardrails DASF 55 Monitor audit logs DASF 56 Restrict outbound connections from models DASF 64 Limit access from AI models and agents DASF 65 Implement end-to-end AI traceability DASF 66 Use Human-in-the-loop feedback DASF 67 Federate authentication DASF 71 Log and register AI agents DASF 73 Register prompts

RISK/DESCRIPTION	MITIGATION CONTROLS
Agents — Core 13.8 **Repudiation & Untraceability** Occurs when actions performed by AI agents cannot be traced back or accounted for due to insufficient logging or transparency in decision-making processes.	DASF 1 SSO with IdP and MFA DASF 2 Sync users and groups DASF 3 Restrict access using IP access lists DASF 4 Restrict access using private link DASF 5 Control access to data and other objects DASF 24 Control access to models and model assets DASF 31 Secure model serving endpoints DASF 37 Set up inference tables for monitoring and debugging models DASF 54 Implement AI guardrails DASF 55 Monitor audit logs DASF 65 Implement end-to-end AI traceability DASF 67 Federate authentication DASF 71 Log and register AI agents
Agents — Core 13.9 **Identity Spoofing & Impersonation** Attackers exploit authentication mechanisms to impersonate AI agents or human users, enabling them to execute unauthorized actions under false identities.	DASF 1 SSO with IdP and MFA DASF 2 Sync users and groups DASF 3 Restrict access using IP access lists DASF 4 Restrict access using private link DASF 5 Control access to data and other objects DASF 24 Control access to models and model assets DASF 31 Secure model serving endpoints DASF 37 Set up inference tables for monitoring and debugging models DASF 54 Implement AI guardrails DASF 55 Monitor audit logs DASF 64 Limit access from AI models and agents DASF 65 Implement end-to-end AI traceability DASF 67 Federate authentication DASF 71 Log and register AI agents

RISK/DESCRIPTION	MITIGATION CONTROLS
Agents — Core 13.10 **Overwhelming Human in the Loop** This threat targets systems with human oversight and decision validation, aiming to exploit human cognitive limitations or compromise interaction frameworks. Attackers may flood agents with decision requests to cause approval fatigue, or embed malicious actions within a high volume of legitimate requests to bypass human scrutiny.	DASF 36 Set up monitoring alerts DASF 37 Set up inference tables for monitoring and debugging models DASF 49 Automate LLM evaluation DASF 60 Rate limit number of inference queries DASF 66 Use Human-in-the-loop feedback
Agents — Core 13.11 **Unexpected RCE and Code Attacks** Attackers exploit AI-generated execution environments to inject malicious code, trigger unintended system behaviors, or execute unauthorized scripts.	DASF 34 Run models in multiple layers of isolation DASF 38 Platform security — penetration testing, red teaming, bug bounty and vulnerability management DASF 55 Monitor audit logs DASF 56 Restrict outbound connections from models DASF 62 Implement network segmentation DASF 63 Update software DASF 64 Limit access from AI models and agents DASF 71 Log and register AI agents
Agents — Core 13.12 **Agent Communication Poisoning** Attackers manipulate communication channels between AI agents to spread false information, disrupt workflows, or influence decision-making.	DASF 9 Encrypt data in transit DASF 34 Run models in multiple layers of isolation DASF 37 Set up inference tables for monitoring and debugging models DASF 55 Monitor audit logs DASF 64 Limit access from AI models and agents DASF 65 Implement end-to-end AI traceability DASF 67 Federate authentication DASF 71 Log and register AI agents

RISK/DESCRIPTION	MITIGATION CONTROLS
Agents — Core 13.13 **Rogue Agents in Multi-Agent Systems** Malicious or compromised AI agents operate outside normal monitoring boundaries, executing unauthorized actions or exfiltrating data.	DASF 34 Run models in multiple layers of isolation DASF 37 Set up inference tables for monitoring and debugging models DASF 38 Platform security — penetration testing, red teaming, bug bounty and vulnerability management DASF 55 Monitor audit logs DASF 56 Restrict outbound connections from models DASF 62 Implement network segmentation DASF 64 Limit access from AI models and agents DASF 65 Implement end-to-end AI traceability DASF 67 Federate authentication DASF 71 Log and register AI agents
Agents — Core 13.14 **Human Attacks on Multi-Agent Systems** Adversaries exploit inter-agent delegation, trust relationships, and workflow dependencies to escalate privileges or manipulate AI-driven operations.	DASF 1 SSO with IdP and MFA DASF 2 Sync users and groups DASF 3 Restrict access using IP access lists DASF 4 Restrict access using private link DASF 5 Control access to data and other objects DASF 24 Control access to models and model assets DASF 31 Secure model serving endpoints DASF 34 Run models in multiple layers of isolation DASF 37 Set up inference tables for monitoring and debugging models DASF 38 Platform security — penetration testing, red teaming, bug bounty and vulnerability management DASF 55 Monitor audit logs DASF 64 Limit access from AI models and agents DASF 65 Implement end-to-end AI traceability DASF 67 Federate authentication DASF 71 Log and register AI agents

RISK/DESCRIPTION	MITIGATION CONTROLS
Agents — Core 13.15 **Human Manipulation** In scenarios where AI agents engage in direct interaction with human users, the trust relationship reduces user skepticism, increasing reliance on the agent's responses and autonomy. This implicit trust and direct human/agent interaction create risks, as attackers can coerce agents to manipulate users, spread misinformation, and take covert actions.	DASF 1 SSO with IdP and MFA DASF 2 Sync users and groups DASF 3 Restrict access using IP access lists DASF 4 Restrict access using private link DASF 5 Control access to data and other objects DASF 24 Control access to models and model assets DASF 31 Secure model serving endpoints DASF 34 Run models in multiple layers of isolation DASF 37 Set up inference tables for monitoring and debugging models DASF 38 Platform security — penetration testing, red teaming, bug bounty and vulnerability management DASF 54 Implement AI guardrails DASF 55 Monitor audit logs DASF 64 Limit access from AI models and agents DASF 65 Implement end-to-end AI traceability DASF 66 Use Human-in-the-loop feedback DASF 71 Log and register AI agents DASF 73 Register prompts

3.2 Agents — Tools MCP Server

RISK/DESCRIPTION	MITIGATION CONTROLS
Agents — Tools MCP Server 13.16 **Prompt Injection** Prompt injection attacks occur when malicious prompts manipulate MCP server behavior or bypass security controls. This includes direct injection through user input, indirect injection through data sources, and manipulation of tool descriptions. These attacks can lead to unauthorized actions, data exfiltration, and privilege escalation, making this the most critical MCP security risk.	DASF 1 SSO with IdP and MFA DASF 3 Restrict access using IP access lists DASF 4 Restrict access using private link DASF 5 Control access to data and other objects DASF 24 Control access to models and model assets DASF 31 Secure model serving endpoints DASF 32 Govern and monitor access of AI model and model serving endpoints DASF 37 Set up inference tables for monitoring and debugging models DASF 46 Store and retrieve embeddings securely DASF 54 Implement AI guardrails DASF 55 Monitor audit logs DASF 56 Restrict outbound connections from models DASF 60 Rate limit number of inference queries DASF 64 Limit access from AI models and agents DASF 65 Implement end-to-end AI traceability DASF 67 Federate authentication DASF 68 Use Securely Hosted Managed MCP Servers DASF 73 Register prompts
Agents — Tools MCP Server 13.17 **Confused Deputy** The confused deputy problem occurs when MCP servers perform actions on behalf of the wrong user or with incorrect permissions. This can result in authorization bypass, cross-user data access, and privilege escalation. The complexity of MCP's role-based interactions makes this vulnerability particularly dangerous in multi-user environments.	DASF 1 SSO with IdP and MFA DASF 3 Restrict access using IP access lists DASF 4 Restrict access using private link DASF 5 Control access to data and other objects DASF 24 Control access to models and model assets DASF 38 Platform security — penetration testing, red teaming, bug bounty and vulnerability management DASF 55 Monitor audit logs DASF 64 Limit access from AI models and agents DASF 65 Implement end-to-end AI traceability DASF 67 Federate authentication DASF 68 Use Securely Hosted Managed MCP Servers DASF 71 Log and register AI agents

RISK/DESCRIPTION	MITIGATION CONTROLS
Agents — Tools MCP Server 13.18 **Tool Poisoning** Tool poisoning involves malicious tools masquerading as legitimate ones, or legitimate tools with malicious descriptions designed to trick AI models. Examples include fake tool descriptions, malicious tool implementations, and tool name squatting. This attack vector exploits the trust relationship between AI models and their available tools.	DASF 1 SSO with IdP and MFA DASF 2 Sync users and groups DASF 3 Restrict access using IP access lists DASF 4 Restrict access using private link DASF 5 Control access to data and other objects DASF 24 Control access to models and model assets DASF 31 Secure model serving endpoints DASF 32 Govern and monitor access of AI model and model serving endpoints DASF 37 Set up inference tables for monitoring and debugging models DASF 38 Platform security — penetration testing, red teaming, bug bounty and vulnerability management DASF 46 Store and retrieve embeddings securely DASF 54 Implement AI guardrails DASF 55 Monitor audit logs DASF 56 Restrict outbound connections from models DASF 60 Rate limit number of inference queries DASF 64 Limit access from AI models and agents DASF 65 Implement end-to-end AI traceability DASF 67 Federate authentication DASF 68 Use Securely Hosted Managed MCP Servers DASF 69 Securely host Custom MCP Servers
Agents — Tools MCP Server 13.19 **Credential and Token Exposure** Credential and token exposure occurs through improper handling, storage, or transmission of API keys, OAuth tokens, and other sensitive credentials. This includes hardcoded credentials, token theft, and credential leakage in logs. Given MCP's reliance on API integrations, credential security is fundamental to overall system security.	DASF 8 Encrypt data at rest DASF 9 Encrypt data in transit DASF 33 Manage credentials securely DASF 55 Monitor audit logs DASF 64 Limit access from AI models and agents DASF 67 Federate authentication DASF 68 Use Securely Hosted Managed MCP Servers DASF 69 Securely host Custom MCP Servers DASF 71 Log and register AI agents

RISK/DESCRIPTION	MITIGATION CONTROLS
Agents — Tools MCP Server 13.20 **Insecure Server Configuration** Insecure server configuration encompasses weak default configurations, exposed endpoints, and inadequate authentication mechanisms. This includes default credentials, open endpoints, and weak authentication systems. Many MCP security incidents stem from basic configuration errors that leave systems vulnerable.	DASF 1 SSO with IdP and MFA DASF 2 Sync users and groups DASF 3 Restrict access using IP access lists DASF 4 Restrict access using private link DASF 5 Control access to data and other objects DASF 31 Secure model serving endpoints DASF 33 Manage credentials securely DASF 55 Monitor audit logs DASF 64 Limit access from AI models and agents DASF 67 Federate authentication DASF 68 Use Securely Hosted Managed MCP Servers DASF 69 Securely host Custom MCP Servers DASF 70 Securely connect to External MCP Servers
Agents — Tools MCP Server 13.21 **Supply Chain Attacks** Supply chain attacks target the MCP ecosystem through compromised MCP servers, malicious dependencies, or rug pull attacks where maintainers abandon or maliciously modify previously trusted servers. The distributed nature of MCP server development makes supply chain security particularly challenging.	DASF 38 Platform security — penetration testing, red teaming, bug bounty and vulnerability management DASF 53 Third-party library control DASF 55 Monitor audit logs DASF 63 Update software DASF 68 Use Securely Hosted Managed MCP Servers DASF 69 Securely host Custom MCP Servers DASF 70 Securely connect to External MCP Servers

RISK/DESCRIPTION	MITIGATION CONTROLS
Agents — Tools MCP Server 13.22 **Excessive Permissions and Scope Creep** Excessive permissions occur when MCP servers request more permissions than necessary for their intended function, or when privileges gradually escalate over time. This includes overprivileged OAuth scopes, unnecessary file system access, and excessive API permissions that increase the potential impact of a compromise.	DASF 5 Control access to data and other objects DASF 38 Platform security — penetration testing, red teaming, bug bounty and vulnerability management DASF 55 Monitor audit logs DASF 64 Limit access from AI models and agents DASF 67 Federate authentication DASF 68 Use Securely Hosted Managed MCP Servers DASF 69 Securely host Custom MCP Servers DASF 70 Securely connect to External MCP Servers
Agents — Tools MCP Server 13.23 **Data Exfiltration** Data exfiltration involves unauthorized access to or transmission of sensitive data through MCP channels. This can occur through sensitive data in responses, covert channels, or unauthorized data access. The ability of MCP servers to access diverse data sources makes this a significant concern for data protection.	DASF 1 SSO with IdP and MFA DASF 2 Sync users and groups DASF 3 Restrict access using IP access lists DASF 4 Restrict access using private link DASF 5 Control access to data and other objects DASF 9 Encrypt data in transit DASF 31 Secure model serving endpoints DASF 33 Manage credentials securely DASF 55 Monitor audit logs DASF 56 Restrict outbound connections from models DASF 64 Limit access from AI models and agents DASF 67 Federate authentication DASF 68 Use Securely Hosted Managed MCP Servers DASF 69 Securely host Custom MCP Servers DASF 70 Securely connect to External MCP Servers

RISK/DESCRIPTION	MITIGATION CONTROLS
Agents — Tools MCP Server 13.24 **Context Spoofing and Manipulation** Context spoofing involves manipulation of context information provided to AI models to alter their behavior in unintended ways. Attacks target flaws in protocols like MCP or A2A, such as consent bypass or context hijacking, leading to unauthorized agent actions. This includes fake context injection, context poisoning, and state manipulation. These attacks exploit the AI model's reliance on context to make decisions about tool usage.	DASF 1 SSO with IdP and MFA DASF 2 Sync users and groups DASF 3 Restrict access using IP access lists DASF 4 Restrict access using private link DASF 5 Control access to data and other objects DASF 9 Encrypt data in transit DASF 31 Secure model serving endpoints DASF 33 Manage credentials securely DASF 37 Set up inference tables for monitoring and debugging models DASF 54 Implement AI guardrails DASF 55 Monitor audit logs DASF 64 Limit access from AI models and agents DASF 67 Federate authentication DASF 68 Use Securely Hosted Managed MCP Servers DASF 69 Securely host Custom MCP Servers DASF 70 Securely connect to External MCP Servers
Agents — Tools MCP Server 13.25 **Insecure Communication** Insecure communication encompasses unencrypted or improperly secured communication channels between MCP components. This includes unencrypted transport, weak TLS implementation, and vulnerability to man-in-the-middle attacks. Secure communication is fundamental to preventing interception of sensitive data and credentials.	DASF 3 Restrict access using IP access lists DASF 4 Restrict access using private link DASF 9 Encrypt data in transit DASF 55 Monitor audit logs DASF 62 Implement network segmentation DASF 68 Use Securely Hosted Managed MCP Servers DASF 69 Securely host Custom MCP Servers DASF 70 Securely connect to External MCP Servers

3.3 Agents — Tools MCP Client

RISK/DESCRIPTION	MITIGATION CONTROLS
Agents — Tools MCP Client 13.26 **Malicious Server Connection** Malicious server connection occurs when clients connect to compromised or malicious MCP servers without proper validation. This includes fake servers, DNS poisoning, server impersonation, and connection hijacking. The distributed nature of MCP server deployment makes it easy for attackers to create malicious servers that appear legitimate to unsuspecting clients.	DASF 9 Encrypt data in transit DASF 38 Platform security — penetration testing, red teaming, bug bounty and vulnerability management DASF 55 Monitor audit logs DASF 68 Use Securely Hosted Managed MCP Servers DASF 69 Securely host Custom MCP Servers DASF 70 Securely connect to External MCP Servers
Agents — Tools MCP Client 13.27 **Insecure Credential Storage** Insecure credential storage involves improper storage of MCP server credentials, API keys, and authentication tokens on the client system. This includes plaintext credentials, weak encryption, accessible credential files, and inadequate protection of sensitive authentication data. Client-side credential storage is particularly vulnerable to local attacks and system compromise.	DASF 8 Encrypt data at rest DASF 33 Manage credentials securely DASF 38 Platform security — penetration testing, red teaming, bug bounty and vulnerability management DASF 55 Monitor audit logs DASF 67 Federate authentication DASF 68 Use Securely Hosted Managed MCP Servers DASF 69 Securely host Custom MCP Servers DASF 70 Securely connect to External MCP Servers

RISK/DESCRIPTION	MITIGATION CONTROLS
Agents — Tools MCP Client 13.28 **UI/UX Deception** UI/UX deception involves misleading users about MCP server actions, permissions, or capabilities through the client interface. This includes hidden tool calls, misleading permission dialogs, unclear action descriptions, and interfaces that don't clearly communicate what actions are being performed. Users may unknowingly authorize dangerous operations	DASF 1 SSO with IdP and MFA DASF 2 Sync users and groups DASF 3 Restrict access using IP access lists DASF 4 Restrict access using private link DASF 5 Control access to data and other objects DASF 9 Encrypt data in transit DASF 31 Secure model serving endpoints DASF 33 Manage credentials securely DASF 55 Monitor audit logs DASF 61 Train users on AI risk taxonomy and AI/ML security DASF 64 Limit access from AI models and agents DASF 65 Implement end-to-end AI traceability DASF 66 Use Human-in-the-loop feedback DASF 67 Federate authentication DASF 68 Use Securely Hosted Managed MCP Servers DASF 69 Securely host Custom MCP Servers DASF 70 Securely connect to External MCP Servers
Agents — Tools MCP Client 13.29 **Insufficient Server Validation** Insufficient server validation occurs when clients fail to properly validate MCP server authenticity and integrity before establishing connections. This includes no certificate validation, missing server verification, weak trust models, and inadequate verification of server identity. Without proper validation, clients may connect to malicious servers.	DASF 9 Encrypt data in transit DASF 38 Platform security — penetration testing, red teaming, bug bounty and vulnerability management DASF 55 Monitor audit logs DASF 67 Federate authentication DASF 68 Use Securely Hosted Managed MCP Servers DASF 69 Securely host Custom MCP Servers DASF 70 Securely connect to External MCP Servers

RISK/DESCRIPTION	MITIGATION CONTROLS
Agents — Tools MCP Client 13.30 **Client-Side Data Leakage** Client-side data leakage involves sensitive data leaking through client logs, caches, temporary files, or local storage mechanisms. This includes credentials in logs, cached sensitive responses, temporary file exposure, and inadequate cleanup of sensitive data. Client systems often store more data than users realize, creating multiple leakage vectors.	DASF 1 SSO with IdP and MFA DASF 2 Sync users and groups DASF 3 Restrict access using IP access lists DASF 4 Restrict access using private link DASF 5 Control access to data and other objects DASF 8 Encrypt data at rest DASF 33 Manage credentials securely DASF 55 Monitor audit logs DASF 61 Train users on AI risk taxonomy and AI/ML security DASF 64 Limit access from AI models and agents
Agents — Tools MCP Client 13.31 **Excessive Permission Granting** Excessive permission granting occurs when clients grant more permissions to MCP servers than necessary for their intended function. This includes overprivileged server access, permission escalation, unnecessary scopes, and inadequate permission review processes. Users may grant broad permissions without understanding the implications.	DASF 5 Control access to data and other objects DASF 55 Monitor audit logs DASF 61 Train users on AI risk taxonomy and AI/ML security DASF 64 Limit access from AI models and agents DASF 67 Federate authentication DASF 68 Use Securely Hosted Managed MCP Servers DASF 69 Securely host Custom MCP Servers DASF 70 Securely connect to External MCP Servers

RISK/DESCRIPTION	MITIGATION CONTROLS
Agents — Tools MCP Client 13.32 **Client-Side Code Execution** Client-side code execution involves malicious responses from MCP servers that execute code on the client system. This includes script injection, code execution via responses, client-side attacks through crafted responses, and exploitation of client-side interpreters. Clients that process server responses unsafely are vulnerable to code execution attacks.	DASF 34 Run models in multiple layers of isolation DASF 54 Implement AI guardrails DASF 55 Monitor audit logs DASF 62 Implement network segmentation DASF 68 Use Securely Hosted Managed MCP Servers DASF 69 Securely host Custom MCP Servers DASF 70 Securely connect to External MCP Servers
Agents — Tools MCP Client 13.33 **Insecure Communication Handling** Insecure communication handling involves poor implementation of secure communication protocols between clients and servers. This includes weak TLS implementation, certificate bypass vulnerabilities, protocol downgrade attacks, and inadequate encryption of sensitive communications. Clients must properly implement secure communication to protect data in transit.	DASF 3 Restrict access using IP access lists DASF 4 Restrict access using private link DASF 9 Encrypt data in transit DASF 38 Platform security — penetration testing, red teaming, bug bounty and vulnerability management DASF 55 Monitor audit logs DASF 63 Update software DASF 68 Use Securely Hosted Managed MCP Servers DASF 69 Securely host Custom MCP Servers DASF 70 Securely connect to External MCP Servers

RISK/DESCRIPTION	MITIGATION CONTROLS
Agents — Tools MCP Client 13.34 **Session and State Management Failures** Session and state management failures involve inadequate management of client sessions, authentication state, and application state. This includes session hijacking, state manipulation, authentication bypass, and inadequate session termination. Poor session management can lead to unauthorized access and privilege escalation.	DASF 1 SSO with IdP and MFA DASF 3 Restrict access using IP access lists DASF 4 Restrict access using private link DASF 5 Control access to data and other objects DASF 9 Encrypt data in transit DASF 55 Monitor audit logs DASF 64 Limit access from AI models and agents DASF 67 Federate authentication DASF 68 Use Securely Hosted Managed MCP Servers DASF 69 Securely host Custom MCP Servers DASF 70 Securely connect to External MCP Servers DASF 72 Securely store and reuse agent state
Agents — Tools MCP Client 13.35 **Update and Patch Management** Update and patch management issues involve insecure update mechanisms or delayed application of security patches. This includes unverified updates, delayed patches, insecure update channels, and inadequate update verification. Clients with poor update mechanisms remain vulnerable to known security issues.	DASF 38 Platform security — penetration testing, red teaming, bug bounty and vulnerability management DASF 53 Third-party library control DASF 55 Monitor audit logs DASF 63 Update software DASF 68 Use Securely Hosted Managed MCP Servers DASF 69 Securely host Custom MCP Servers

04 Understanding Agentic AI Risk Mitigation Controls

The mitigation controls for Agentic AI build upon the foundation of the DASF but require a shift in perspective—from securing information to securing autonomy. When an AI system can take action, "read-only" access controls are no longer sufficient.

The controls outlined in this extension address the 35 specific technical risks identified across the Agentic AI landscape. These controls are designed to enforce "Safe by Design" principles for autonomous systems, here is a quick sampling of what is comprehensively mapped in the compendium:

- **Control of Scope & Agency:** Mitigation strategies now emphasize **Least Privilege for Tools**. Just as humans have RBAC (Role-Based Access Control), agents require granular permissions (Authorization). Controls focus on limiting an agent's "blast radius" by ensuring it only has access to the specific MCP tools and data slices necessary for its immediate task. See **DASF 64: Limit access from AI models and agents**.
- **Human-in-the-Loop (HITL) & Oversight:** For high-stakes actions, controls are introduced to require human verification before tool execution, mitigating risks like **Overwhelming Human in the Loop (Risk 13.10)** by designing interaction frameworks that respect human cognitive limits. See **DASF 66: Use Human-in-the-loop feedback**.
- **Observability of Thought:** Unlike standard logging, Agentic controls require tracing the chain of thought. We recommend implementation controls that capture the agent's planning steps and tool reasoning, ensuring that **Repudiation & Untraceability (Risk 13.8)** is addressed. See **DASF 65: Implement end-to-end AI traceability**.
- **Sandboxing & Isolation:** To prevent **Unexpected RCE and Code Attacks (Risk 13.11)**, controls prioritize the execution of agent-generated code in ephemeral, isolated environments. See **DASF 34: Run models in multiple layers of isolation**.

As with the core DASF, these controls are categorized as Out-of-the-box (platform default), Configuration (settings to toggle), or Implementation (architectural choices). For Databricks customers, the compendium maps these directly to features such as **Unity Catalog** governance for tools (**DASF 5**), **Mosaic AI Agent Framework's authentication for AI agents** (**DASF 64**), and **Vector Search security settings** (**DASF 46**).

CONTROL/RISK	DESCRIPTION OF CONTROL IMPLEMENTATION ON THE DATABRICKS PLATFORM
DASF 1 **SSO with IdP and MFA** RISKS AGENTS — CORE 13.1, AGENTS — CORE 13.14, AGENTS — CORE 13.15, AGENTS — CORE 13.2, AGENTS — CORE 13.3, AGENTS — CORE 13.6, AGENTS — CORE 13.7, AGENTS — CORE 13.8, AGENTS — CORE 13.9, AGENTS — TOOLS MCP CLIENT 13.28, AGENTS — TOOLS MCP CLIENT 13.30, AGENTS — TOOLS MCP CLIENT 13.34, AGENTS — TOOLS MCP SERVER 13.16, AGENTS — TOOLS MCP SERVER 13.17, AGENTS — TOOLS MCP SERVER 13.18, AGENTS — TOOLS MCP SERVER 13.20, AGENTS — TOOLS MCP SERVER 13.23, AGENTS — TOOLS MCP SERVER 13.24	DESCRIPTION Implementing single sign-on with an identity provider's (IdP) multi-factor authentication is critical for secure authentication. It adds an extra layer of security, ensuring that only authorized users access the Databricks Platform. CONTROL CATEGORY: Configuration PRODUCT REFERENCE: AWS Azure GCP
DASF 2 **Sync users and groups** RISKS AGENTS — CORE 13.1, AGENTS — CORE 13.14, AGENTS — CORE 13.15, AGENTS — CORE 13.2, AGENTS — CORE 13.3, AGENTS — CORE 13.6, AGENTS — CORE 13.7, AGENTS — CORE 13.8, AGENTS — CORE 13.9, AGENTS — TOOLS MCP CLIENT 13.28, AGENTS — TOOLS MCP CLIENT 13.30, AGENTS — TOOLS MCP SERVER 13.18, AGENTS — TOOLS MCP SERVER 13.20, AGENTS — TOOLS MCP SERVER 13.23, AGENTS — TOOLS MCP SERVER 13.24	DESCRIPTION Synchronizing users and groups from your identity provider (IdP) with Databricks using the SCIM standard facilitates consistent and automated user provisioning for enhancing security. CONTROL CATEGORY: Configuration PRODUCT REFERENCE: AWS Azure GCP

CONTROL/RISK	DESCRIPTION OF CONTROL IMPLEMENTATION ON THE DATABRICKS PLATFORM
DASF 3 **Restrict access using IP access lists** RISKS AGENTS — CORE 13.1 AGENTS — CORE 13.14 AGENTS — CORE 13.15 AGENTS — CORE 13.2 AGENTS — CORE 13.3 AGENTS — CORE 13.6 AGENTS — CORE 13.7 AGENTS — CORE 13.8 AGENTS — CORE 13.9 AGENTS — TOOLS MCP CLIENT 13.28 AGENTS — TOOLS MCP CLIENT 13.30 AGENTS — TOOLS MCP CLIENT 13.33 AGENTS — TOOLS MCP CLIENT 13.34 AGENTS — TOOLS MCP SERVER 13.16 AGENTS — TOOLS MCP SERVER 13.17 AGENTS — TOOLS MCP SERVER 13.18 AGENTS — TOOLS MCP SERVER 13.20 AGENTS — TOOLS MCP SERVER 13.23 AGENTS — TOOLS MCP SERVER 13.24 AGENTS — TOOLS MCP SERVER 13.25	DESCRIPTION Configure IP access lists to restrict authentication to Databricks from specific IP ranges, such as VPNs or office networks, and strengthen network security by preventing unauthorized access from untrusted locations. Secure Egress Gateway (SEG) is a component of Databricks Platform Security that allows an administrator to implement policies that restrict access to internal or external endpoints. Currently, SEG only applies to Serverless runtimes. CONTROL CATEGORY Configuration PRODUCT REFERENCE AWS Azure GCP
DASF 4 **Restrict access using private link** RISKS AGENTS — CORE 13.1 AGENTS — CORE 13.14 AGENTS — CORE 13.15 AGENTS — CORE 13.2 AGENTS — CORE 13.3 AGENTS — CORE 13.6 AGENTS — CORE 13.7 AGENTS — CORE 13.8 AGENTS — CORE 13.9 AGENTS — TOOLS MCP CLIENT 13.28 AGENTS — TOOLS MCP CLIENT 13.30 AGENTS — TOOLS MCP CLIENT 13.33 AGENTS — TOOLS MCP CLIENT 13.34 AGENTS — TOOLS MCP SERVER 13.16 AGENTS — TOOLS MCP SERVER 13.17 AGENTS — TOOLS MCP SERVER 13.18 AGENTS — TOOLS MCP SERVER 13.20 AGENTS — TOOLS MCP SERVER 13.23 AGENTS — TOOLS MCP SERVER 13.24 AGENTS — TOOLS MCP SERVER 13.25	DESCRIPTION Use AWS PrivateLink, Azure Private Link or GCP Private Service Connect to create a private network route between the customer and the Databricks control plane or the control plane and the customer's compute plane environments to enhance data security by avoiding public internet exposure. CONTROL CATEGORY Configuration PRODUCT REFERENCE AWS Azure GCP

CONTROL/RISK	DESCRIPTION OF CONTROL IMPLEMENTATION ON THE DATABRICKS PLATFORM
DASF 5 **Control access to data and other objects** RISKS AGENTS — CORE 13.1 AGENTS — CORE 13.14 AGENTS — CORE 13.15 AGENTS — CORE 13.2 AGENTS — CORE 13.3 AGENTS — CORE 13.6 AGENTS — CORE 13.7 AGENTS — CORE 13.8 AGENTS — CORE 13.9 AGENTS — TOOLS MCP CLIENT 13.28 AGENTS — TOOLS MCP CLIENT 13.30 AGENTS — TOOLS MCP CLIENT 13.31 AGENTS — TOOLS MCP CLIENT 13.34 AGENTS — TOOLS MCP SERVER 13.16 AGENTS — TOOLS MCP SERVER 13.17 AGENTS — TOOLS MCP SERVER 13.18 AGENTS — TOOLS MCP SERVER 13.20 AGENTS — TOOLS MCP SERVER 13.22 AGENTS — TOOLS MCP SERVER 13.23 AGENTS — TOOLS MCP SERVER 13.24	DESCRIPTION Implementing Unity Catalog for unified permissions management and assets simplifies access control and enhances security. CONTROL CATEGORY Implementation PRODUCT REFERENCE AWS Azure GCP

CONTROL/RISK	DESCRIPTION OF CONTROL IMPLEMENTATION ON THE DATABRICKS PLATFORM
DASF 6 **Classify data** RISKS	DESCRIPTION Data catalogs can have a vast amount of data, often containing known and unknown sensitive data. It is critical for data teams to understand what kind of sensitive data exists in each table so that they can both govern and democratize access to this data. To address this problem, Databricks Data Classification automatically classifies and tags tables in your catalog. This allows you to discover sensitive data, as well as apply governance controls over the results, using tools such as role-based access control (RBAC) and attribute-based access control (ABAC) policies in Unity Catalog. You can also use tags manually or via API as attributes containing keys and optional values that you can apply to different securable objects in Unity Catalog. Organizing securable objects with tags in Unity Catalog aids in efficient data management, data discovery and classification, essential for handling large datasets. Additionally, use governed tags to manage and enforce a tag policy. Tag policies allow admins to define which tag keys are governed, specify the allowed values for those tags, and control who can assign or modify them. This provides organizations with centralized control over tag usage, supporting consistent data classification, compliance, and operational standards. CONTROL CATEGORY: Implementation PRODUCT REFERENCE: AWS Azure GCP
DASF 7 **Enforce data quality checks on batch and streaming datasets** RISKS	DESCRIPTION Databricks Delta Live Tables (DLT) simplifies ETL development with declarative pipelines that integrate quality control checks and performance monitoring. CONTROL CATEGORY: Implementation PRODUCT REFERENCE: AWS Azure GCP

CONTROL/RISK	DESCRIPTION OF CONTROL IMPLEMENTATION ON THE DATABRICKS PLATFORM
DASF 8 **Encrypt data at rest** RISKS AGENTS — TOOLS MCP CLIENT 13.27 AGENTS — TOOLS MCP CLIENT 13.30 AGENTS — TOOLS MCP SERVER 13.19	DESCRIPTION Databricks supports customer-managed encryption keys to strengthen data at rest protection and greater access control. CONTROL CATEGORY: Configuration PRODUCT REFERENCE: AWS Azure GCP
DASF 9 **Encrypt data in transit** RISKS AGENTS — CORE 13.12 AGENTS — TOOLS MCP CLIENT 13.26 AGENTS — TOOLS MCP CLIENT 13.28 AGENTS — TOOLS MCP CLIENT 13.29 AGENTS — TOOLS MCP CLIENT 13.33 AGENTS — TOOLS MCP CLIENT 13.34 AGENTS — TOOLS MCP SERVER 13.19 AGENTS — TOOLS MCP SERVER 13.23 AGENTS — TOOLS MCP SERVER 13.24 AGENTS — TOOLS MCP SERVER 13.25	DESCRIPTION Databricks supports TLS 1.2+ encryption to protect customer data during transit. This applies to data transfer between the customer and the Databricks control plane and within the compute plane. Customers can also secure inter-cluster communications within the compute plane per their security requirements. CONTROL CATEGORY: Out-of-the-box PRODUCT REFERENCE: AWS Azure GCP

CONTROL/RISK	DESCRIPTION OF CONTROL IMPLEMENTATION ON THE DATABRICKS PLATFORM
DASF 10 **Version data** RISKS	DESCRIPTION Store data in a lakehouse architecture using Delta tables. Delta tables can be versioned to revert any user or malicious actor poisoning of data. Data can be stored in a lakehouse architecture in the customer's cloud account. Both raw data and feature tables are stored as Delta tables with access controls to determine who can read and modify them. Data lineage with UC helps track and audit changes and the origin of ML data sources. Each operation that modifies a Delta Lake table creates a new table version. User actions are tracked and audited, and lineage of transformations is available all in the same platform. You can use history information to audit operations, roll back a table or query a table at a specific point in time using time travel. CONTROL CATEGORY: Implementation PRODUCT REFERENCE: AWS Azure GCP
DASF 11 **Capture and view data lineage** RISKS	DESCRIPTION Unity Catalog tracks and visualizes real-time data lineage across all languages to the column level, providing a traceable history of an object from notebooks, workflows, models and dashboards. This enhances transparency and compliance, with accessibility provided through the Catalog Explorer. CONTROL CATEGORY: Out-of-the-box PRODUCT REFERENCE: AWS Azure GCP

CONTROL/RISK	DESCRIPTION OF CONTROL IMPLEMENTATION ON THE DATABRICKS PLATFORM
DASF 12 **Delete records from datasets** RISKS	DESCRIPTION Data governance in Delta Lake, the lakehouse storage layer, utilizes its atomicity, consistency, isolation, durability (ACID) properties for effective data management. This includes the capability to remove data based on specific predicates from a Delta Table, including the complete removal of data's history, supporting compliance with regulations like GDPR and CCPA. CONTROL CATEGORY: Implementation PRODUCT REFERENCE: AWS Azure GCP
DASF 13 **Use near real-time data** RISKS	DESCRIPTION Use Databricks for near real-time data ingestion, processing, machine learning, and AI for streaming data. CONTROL CATEGORY: Implementation PRODUCT REFERENCE: AWS Azure GCP
DASF 14 **Audit actions performed on datasets** RISKS	DESCRIPTION Databricks auditing, enhanced by Unity Catalog's events, delivers fine-grained visibility into data access and user activities. This is vital for robust data governance and security, especially in regulated industries. It enables organizations to proactively identify and manage over entitled users, enhancing data security and ensuring compliance. CONTROL CATEGORY: Implementation PRODUCT REFERENCE: AWS Azure GCP

CONTROL/RISK	DESCRIPTION OF CONTROL IMPLEMENTATION ON THE DATABRICKS PLATFORM
DASF 15 **Explore datasets and identify problems** RISKS	DESCRIPTION Iteratively explore, share and prep data for the machine learning lifecycle by creating reproducible, editable and shareable datasets, tables and visualizations. Within Databricks this EDA process can be accelerated with Mosaic AI AutoML. AutoML not only generates baseline models given a dataset, but also provides the underlying model training code in the form of a Python notebook. Notably for EDA, AutoML calculates summary statistics on the provided dataset, creating a notebook for the data scientist to review and adapt. CONTROL CATEGORY: Implementation PRODUCT REFERENCE: AWS Azure GCP
DASF 16 **Secure model features** RISKS	DESCRIPTION Databricks Feature Store is a centralized repository that enables data scientists to find and share features and also ensures that the same code used to compute the feature values is used for model training and inference. Unity Catalog's capabilities, such as security, lineage, table history, tagging and cross-workspace access, are automatically available to the feature table to reduce the risk of malicious actors manipulating the features that feed into ML training. CONTROL CATEGORY: Implementation PRODUCT REFERENCE: AWS Azure GCP

CONTROL/RISK	DESCRIPTION OF CONTROL IMPLEMENTATION ON THE DATABRICKS PLATFORM
DASF 17 **Track and reproduce the training data used for ML model training** RISKS	DESCRIPTION MLflow with Delta Lake tracks the training data used for ML model training. It also enables the identification of specific ML models and runs derived from particular datasets for regulatory and auditable attribution. CONTROL CATEGORY: Configuration PRODUCT REFERENCE: AWS Azure GCP
DASF 18 **Govern model assets** RISKS	DESCRIPTION With Unity Catalog, organizations can implement a unified governance framework for their structured and unstructured data, machine learning models, notebooks, features, functions, and files, enhancing security and compliance across clouds and platforms. Maintain an updated inventory of high-impact AI use cases, including details on purpose, benefits, risks, and risk management practices. CONTROL CATEGORY: Configuration PRODUCT REFERENCE: AWS Azure GCP
DASF 19 **Manage end-to-end machine learning lifecycle** RISKS	DESCRIPTION Databricks includes a managed version of MLflow featuring enterprise security controls and high availability. It supports functionalities like experiments, run management and notebook revision capture. MLflow on Databricks allows tracking and measuring machine learning model training runs, logging model training artifacts and securing machine learning projects. CONTROL CATEGORY: Implementation PRODUCT REFERENCE: AWS Azure GCP

CONTROL/RISK	DESCRIPTION OF CONTROL IMPLEMENTATION ON THE DATABRICKS PLATFORM
DASF 20 **Track ML training runs** RISKS	DESCRIPTION MLflow tracking facilitates the automated recording and retrieval of experiment details, including algorithms, code, datasets, parameters, configurations, signatures and artifacts. CONTROL CATEGORY: Implementation PRODUCT REFERENCE: AWS Azure GCP
DASF 21 **Monitor data and AI system from a single pane of glass** RISKS	DESCRIPTION Databricks Lakehouse Monitoring offers a single pane of glass to centrally track tables' data quality and statistical properties and automatically classifies data. It can also track the performance of machine learning models and model serving endpoints by monitoring inference tables containing model inputs and predictions through a single pane of glass. CONTROL CATEGORY: Implementation PRODUCT REFERENCE: AWS Azure GCP

CONTROL/RISK	DESCRIPTION OF CONTROL IMPLEMENTATION ON THE DATABRICKS PLATFORM
DASF 22 **Build models with all representative, accurate and relevant data sources** RISKS	DESCRIPTION Harnessing internal data and intellectual property to customize large AI models can offer a significant competitive edge. However, this process can be complex, involving coordination across various parts of the organization. The Data Intelligence Platform addresses this challenge by integrating data across traditionally isolated departments and systems. This integration facilitates a more cohesive data and AI strategy, enabling the effective training, testing and evaluation of models using a comprehensive dataset. Use caution when preparing data for traditional models and GenAI training to ensure that you are not unintentionally including data that causes legal conflicts, such as copyright violations, privacy violations or HIPAA violations. CONTROL CATEGORY: Implementation PRODUCT REFERENCE: AWS Azure GCP
DASF 23 **Register, version, approve, promote, deploy and monitor models** RISKS	DESCRIPTION MLflow Model Registry supports managing the machine learning model lifecycle with capabilities for lineage tracking, versioning, staging and model serving. CONTROL CATEGORY: Implementation PRODUCT REFERENCE: AWS Azure GCP

CONTROL/RISK	DESCRIPTION OF CONTROL IMPLEMENTATION ON THE DATABRICKS PLATFORM
DASF 24 **Control access to models and model assets** RISKS AGENTS — CORE 13.1 AGENTS — CORE 13.14 AGENTS — CORE 13.15 AGENTS — CORE 13.2 AGENTS — CORE 13.3 AGENTS — CORE 13.6 AGENTS — CORE 13.7 AGENTS — CORE 13.8 AGENTS — CORE 13.9 AGENTS — TOOLS MCP SERVER 13.16 AGENTS — TOOLS MCP SERVER 13.17 AGENTS — TOOLS MCP SERVER 13.18	DESCRIPTION Organizations commonly encounter challenges in tracking and controlling access to ML models, auditing their usage, and understanding their evolution in complex machine learning workflows. Unity Catalog integrates with the MLflow Model Registry across model lifecycles. This approach simplifies the management and oversight of ML models, proving particularly valuable in environments with multiple teams and diverse projects. CONTROL CATEGORY: Implementation PRODUCT REFERENCE: AWS Azure GCP
DASF 25 **Use retrieval augmented generation (RAG) with large language models (LLMs)** RISKS	DESCRIPTION Generating relevant and accurate responses in large language models (LLMs) while avoiding hallucinations requires grounding them in domain-specific knowledge. Retrieval augmented generation (RAG) addresses this by breaking down extensive datasets into manageable segments (“chunks”) that are “vector embedded.” These vector embeddings are mathematical representations that help the model understand and quantify different data segments. As a result, LLMs produce responses that are contextually relevant and deeply rooted in the specific domain knowledge. CONTROL CATEGORY: Implementation PRODUCT REFERENCE: AWS Azure GCP

CONTROL/RISK	DESCRIPTION OF CONTROL IMPLEMENTATION ON THE DATABRICKS PLATFORM
DASF 26 **Fine-tune large language models (LLMs)** RISKS	DESCRIPTION Data is your competitive advantage. Use it to customize large AI models to beat your competition. Produce new model variants with tailored LLM response style and structure via fine-tuning. Fine-tune your own LLM with open models to own your IP. CONTROL CATEGORY: Implementation PRODUCT REFERENCE: AWS Azure
DASF 27 **Pretrain a large language model (LLM)** RISKS	DESCRIPTION Data is your competitive advantage. Use it to customize large AI models to beat your competition by pretraining models with your data, imbuing the model with domain-specific knowledge, vocabulary and semantics. Pretrain your own LLM with MosaicML to own your IP. CONTROL CATEGORY: Implementation PRODUCT REFERENCE: AWS Azure
DASF 28 **Create model aliases, tags and annotations** RISKS	DESCRIPTION Model aliases in machine learning workflows allow you to assign a mutable, named reference to a specific version of a registered model. This functionality is beneficial for tracking and managing different stages of a model's lifecycle, indicating the current deployment status of any given model version. CONTROL CATEGORY: Implementation PRODUCT REFERENCE: AWS Azure GCP

CONTROL/RISK	DESCRIPTION OF CONTROL IMPLEMENTATION ON THE DATABRICKS PLATFORM
DASF 29 **Build MLOps workflows** RISKS	DESCRIPTION The lakehouse forms the foundation of a data-centric AI platform. Key to this is the ability to manage both data and AI assets from a unified governance solution on the lakehouse. Databricks Unity Catalog enables this by providing centralized access control, auditing, approvals, model workflow, lineage, and data discovery capabilities across Databricks workspaces. These benefits are now extended to MLflow Models with the introduction of Models in Unity Catalog. Through providing a hosted version of the MLflow Model Registry in Unity Catalog, the full lifecycle of an ML model can be managed while leveraging Unity Catalog's capability to share assets across Databricks workspaces and trace lineage across both data and models. CONTROL CATEGORY: Implementation PRODUCT REFERENCE: AWS Azure GCP
DASF 30 **Encrypt models** RISKS	DESCRIPTION Databricks Platform secures model assets and their transfer with TLS 1.2+ in-transit encryption. Additionally, Unity Catalog's managed model registry provides encryption at rest for persisting models, further enhancing security. CONTROL CATEGORY: Out-of-the-box PRODUCT REFERENCE: AWS Azure GCP

CONTROL/RISK	DESCRIPTION OF CONTROL IMPLEMENTATION ON THE DATABRICKS PLATFORM

DASF 31

Secure model serving endpoints

RISKS

AGENTS — CORE 13.1 | AGENTS — CORE 13.14 | AGENTS — CORE 13.15 | AGENTS — CORE 13.2 | AGENTS — CORE 13.3 | AGENTS — CORE 13.4 | AGENTS — CORE 13.5 | AGENTS — CORE 13.6 | AGENTS — CORE 13.7 | AGENTS — CORE 13.8 | AGENTS — CORE 13.9 | AGENTS — TOOLS MCP CLIENT 13.28 | AGENTS — TOOLS MCP SERVER 13.16 | AGENTS — TOOLS MCP SERVER 13.18 | AGENTS — TOOLS MCP SERVER 13.20 | AGENTS — TOOLS MCP SERVER 13.23 | AGENTS — TOOLS MCP SERVER 13.24

DESCRIPTION

Model serving involves risks of unauthorized data access and model tampering, which can compromise the integrity and reliability of machine learning deployments. Mosaic AI Model Serving addresses these concerns by providing secure-by-default REST API endpoints for MLflow machine learning models, featuring autoscaling, high availability and low latency.

CONTROL CATEGORY

Out-of-the-box

PRODUCT REFERENCE

AWS Azure GCP

CONTROL/RISK	DESCRIPTION OF CONTROL IMPLEMENTATION ON THE DATABRICKS PLATFORM

DASF 32

Govern and monitor access of AI model and model serving endpoints

RISKS

AGENTS — TOOLS MCP SERVER 13.16

AGENTS — TOOLS MCP SERVER 13.18

DESCRIPTION

Mosaic AI Gateway is designed to streamline the usage and management of generative AI models and agents within an organization. It is a centralized service that brings governance, monitoring, and production readiness to model serving endpoints. It also allows you to run, secure, and govern AI traffic to democratize and accelerate AI adoption for your organization. Supported by Model Serving AI Gateway, Databricks external models via the AI Gateway allow you to streamline the usage and management of various large language model (LLM) providers, such as OpenAI and Anthropic, within an organization. You can also use Mosaic AI Model Serving as a provider to serve predictive ML models, which offers rate limits for those endpoints. As part of this support, Model Serving offers a high-level interface that simplifies the interaction with these services by providing a unified endpoint to handle specific LLM-related requests. In addition, Databricks support for external models provides centralized credential management. By storing API keys in one secure location, organizations can enhance their security posture by minimizing the exposure of sensitive API keys throughout the system. It also helps to prevent exposing these keys within code or requiring end users to manage keys safely.

CONTROL CATEGORY

Out-of-the-box

PRODUCT REFERENCE

AWS Azure GCP

CONTROL/RISK	DESCRIPTION OF CONTROL IMPLEMENTATION ON THE DATABRICKS PLATFORM
DASF 33 **Manage credentials securely** RISKS AGENTS — TOOLS MCP CLIENT 13.27 AGENTS — TOOLS MCP CLIENT 13.28 AGENTS — TOOLS MCP CLIENT 13.30 AGENTS — TOOLS MCP SERVER 13.19 AGENTS — TOOLS MCP SERVER 13.20 AGENTS — TOOLS MCP SERVER 13.23 AGENTS — TOOLS MCP SERVER 13.24	DESCRIPTION Databricks Secrets stores your credentials and references them in notebooks, scripts, configuration properties and jobs. Integrating with heterogeneous systems requires managing a potentially large set of credentials and safely distributing them across an organization. Instead of directly entering your credentials into a notebook, use Databricks Secrets to store your credentials and reference them in notebooks and jobs to prevent credential leaks through models. Databricks secret management allows users to use and share credentials within Databricks securely. You can also choose to use a third-party secret management service, such as AWS Secrets Manager or a third-party secret manager. CONTROL CATEGORY: Implementation PRODUCT REFERENCE: AWS Azure GCP
DASF 34 **Run models in multiple layers of isolation** RISKS AGENTS — CORE 13.1 AGENTS — CORE 13.11 AGENTS — CORE 13.12 AGENTS — CORE 13.13 AGENTS — CORE 13.14 AGENTS — CORE 13.15 AGENTS — CORE 13.2 AGENTS — CORE 13.3 AGENTS — CORE 13.4 AGENTS — CORE 13.5 AGENTS — CORE 13.6 AGENTS — CORE 13.7 AGENTS — TOOLS MCP CLIENT 13.32	DESCRIPTION Databricks Serverless Compute provides a secure-by-design model serving service featuring defense-in-depth controls like dedicated VMs, network segmentation, and encryption for data in transit and at rest. It adheres to the principle of least privilege for enhanced security. CONTROL CATEGORY: Out-of-the-box PRODUCT REFERENCE: AWS Azure GCP

CONTROL/RISK	DESCRIPTION OF CONTROL IMPLEMENTATION ON THE DATABRICKS PLATFORM
DASF 35 **Track model performance** RISKS	DESCRIPTION Databricks Lakehouse Monitoring provides performance metrics and data quality statistics across all account tables. It tracks the performance of machine learning models and model serving endpoints by observing inference tables with model inputs and predictions. CONTROL CATEGORY: Implementation PRODUCT REFERENCE: AWS Azure GCP
DASF 36 **Set up monitoring alerts** RISKS AGENTS — CORE 13.10 AGENTS — CORE 13.4 AGENTS — CORE 13.5	DESCRIPTION Databricks SQL alerts can monitor the metrics table for security-based conditions, ensuring data integrity and timely response to potential issues: - Statistic range alert: Triggers when a specific statistic, such as the fraction of missing values, exceeds a predetermined threshold - Data distribution shift alert: Activates upon shifts in data distribution, as indicated by the drift metrics table - Baseline divergence alert: Alerts if data significantly diverges from a baseline, suggesting potential needs for data analysis or model retraining, particularly in InferenceLog analysis CONTROL CATEGORY: Implementation PRODUCT REFERENCE: AWS Azure GCP

CONTROL/RISK	DESCRIPTION OF CONTROL IMPLEMENTATION ON THE DATABRICKS PLATFORM

DASF 37

Set up inference tables for monitoring and debugging models

RISKS

AGENTS — CORE 13.1 | AGENTS — CORE 13.10 | AGENTS — CORE 13.12 | AGENTS — CORE 13.13 | AGENTS — CORE 13.14 | AGENTS — CORE 13.15 | AGENTS — CORE 13.2 | AGENTS — CORE 13.3 | AGENTS — CORE 13.5 | AGENTS — CORE 13.6 | AGENTS — CORE 13.7 | AGENTS — CORE 13.8 | AGENTS — CORE 13.9 | AGENTS — TOOLS MCP SERVER 13.16 | AGENTS — TOOLS MCP SERVER 13.18 | AGENTS — TOOLS MCP SERVER 13.24

DESCRIPTION

Databricks inference tables automatically record incoming requests and outgoing responses to model serving endpoints, storing them as a Unity Catalog Delta table. This table can be used to monitor, debug and enhance ML models. By coupling inference tables with Lakehouse Monitoring, customers can also set up automated monitoring jobs and alerts on inference tables, such as monitoring text quality or toxicity from endpoints serving LLMs, etc. Critical applications of an inference table include: - Retraining dataset creation: Building datasets for the next iteration of your models - Quality monitoring: Keeping track of production data and model performance - Diagnostics and debugging: Investigating and resolving issues with suspicious inferences - Mislabeled data identification: Compiling data that needs relabeling

CONTROL CATEGORY

Implementation

PRODUCT REFERENCE

AWS Azure

CONTROL/RISK	DESCRIPTION OF CONTROL IMPLEMENTATION ON THE DATABRICKS PLATFORM

DASF 38

Platform security — penetration testing, red teaming, bug bounty and vulnerability management

RISKS

AGENTS — CORE 13.11 | AGENTS — CORE 13.13 | AGENTS — CORE 13.14 | AGENTS — CORE 13.15 | AGENTS — CORE 13.7 | AGENTS — TOOLS MCP CLIENT 13.26 | AGENTS — TOOLS MCP CLIENT 13.27 | AGENTS — TOOLS MCP CLIENT 13.29 | AGENTS — TOOLS MCP CLIENT 13.33 | AGENTS — TOOLS MCP CLIENT 13.35 | AGENTS — TOOLS MCP SERVER 13.17 | AGENTS — TOOLS MCP SERVER 13.18 | AGENTS — TOOLS MCP SERVER 13.21 | AGENTS — TOOLS MCP SERVER 13.22

DESCRIPTION

Mitigating attacks on infrastructure hosting AI services, including AI red teaming for large language models, is crucial for safe model development and deployment. Regular security and penetration testing help identify and address infrastructure vulnerabilities before attackers can exploit them. Databricks operates a formal, documented vulnerability management program overseen by the Chief Security Officer (CSO). The program is management-approved, reviewed annually, and communicated to all relevant internal parties. The policy mandates that vulnerabilities be addressed based on severity: critical vulnerabilities within 14 days, high-severity vulnerabilities within 30 days, and medium-severity vulnerabilities within 60 days. AI red teaming, especially for large language models, is an essential component of ensuring model safety and security. Databricks conducts regular AI red teaming on models and systems developed internally.

CONTROL CATEGORY

Out-of-the-box

PRODUCT REFERENCE

AWS Azure GCP

CONTROL/RISK	DESCRIPTION OF CONTROL IMPLEMENTATION ON THE DATABRICKS PLATFORM
DASF 39 **Platform security — Incident Response Team** RISKS AGENTS — CORE 13.4 AGENTS — CORE 13.5	DESCRIPTION Databricks has established a formal incident response plan that outlines key elements such as roles, responsibilities, escalation paths and external communication protocols. The platform handles over 9TB of audit logs daily, aiding customer and Databricks security investigations. A dedicated security incident response team operates an internal Databricks instance, consolidating essential log sources for thorough security analysis. Databricks ensures continual operational readiness with a 24/7/365 on-call rotation. Additionally, a proactive hunting program and a specialized detection team support the incident response program and require periodic AI audits and establish protocols for incident reporting, including logs review, performance monitoring, and procedures to report and address misuse. CONTROL CATEGORY: Out-of-the-box PRODUCT REFERENCE: AWS Azure GCP
DASF 40 **Platform security — internal access** RISKS	DESCRIPTION Databricks personnel, by default, do not have access to customer workspaces or production environments. Access may be temporarily requested by Databricks staff for purposes such as investigating outages, security events or supporting deployments. Customers have the option to disable this access. Additionally, staff activity within these environments is recorded in customer audit logs. Accessing these areas requires multi-factor authentication, and employees must connect to the Databricks VPN. CONTROL CATEGORY: Out-of-the-box PRODUCT REFERENCE: AWS Azure GCP

CONTROL/RISK	DESCRIPTION OF CONTROL IMPLEMENTATION ON THE DATABRICKS PLATFORM
DASF 41 **Platform security — secure SDLC** RISKS	DESCRIPTION Databricks engineering integrates security throughout the software development lifecycle (SDLC), encompassing both technical and process-level controls under the oversight of our chief security officer (CSO). Activities within our SDLC include: - Code peer reviews - Static and dynamic scans for code and containers, including dependencies - Feature-level security reviews - Annual software engineering security training - Cross-organizational collaborations between security, product management, product security and security champions - Controls to prevent over-disclosure of technical information about AI systems and organizational details that could enable adversarial targeting These development controls are augmented by internal and external penetration testing programs, with findings tracked for resolution and reported to our executive team. Databricks' processes undergo an independent annual review, the results of which are published in our SOC 2 Type 2 report, available upon request. CONTROL CATEGORY: Out-of-the-box PRODUCT REFERENCE: AWS Azure GCP

CONTROL/RISK	DESCRIPTION OF CONTROL IMPLEMENTATION ON THE DATABRICKS PLATFORM
DASF 42 **Employ data-centric MLOps and LLMOps** RISKS	DESCRIPTION MLOps enhances efficiency, scalability, security and risk reduction in machine learning projects. Databricks integrates with MLflow, focusing on enterprise reliability, security and scalability for managing the machine learning lifecycle. The latest update to MLflow introduces new LLMOps features for better management and deployment of large language models (LLMs). This includes integrations with Hugging Face Transformers, OpenAI and the external models in Mosaic AI Model Serving. MLflow also integrates with LangChain and a prompt engineering UI, facilitating generative AI application development for use cases such as chatbots, document summarization and text classification. CONTROL CATEGORY: Implementation PRODUCT REFERENCE: AWS Azure GCP
DASF 43 **Use access control lists** RISKS	DESCRIPTION Databricks access control lists (ACLs) enable you to configure permissions for accessing and interacting with workspace objects, including folders, notebooks, experiments, models, clusters, pools, jobs, Delta Live Tables pipelines, alerts, dashboards, queries and SQL warehouses. CONTROL CATEGORY: Implementation PRODUCT REFERENCE: AWS Azure GCP

CONTROL/RISK	DESCRIPTION OF CONTROL IMPLEMENTATION ON THE DATABRICKS PLATFORM
DASF 44 **Triggering actions in response to a specific event** RISKS	DESCRIPTION Webhooks in the MLflow Model Registry enable you to automate machine learning workflow by triggering actions in response to specific events. These webhooks facilitate seamless integrations, allowing for the automatic execution of various processes. For example, webhooks are used for: - CI workflow trigger: Validate your model automatically when creating a new version - Team notifications: Send alerts through a messaging app when a model stage transition request is received - Model fairness evaluation: Invoke a workflow to assess model fairness and bias upon a production transition request - Automated deployment: Trigger a deployment pipeline when a new tag is created on a model CONTROL CATEGORY: Implementation PRODUCT REFERENCE: AWS Azure GCP
DASF 45 **Evaluate models** RISKS AGENTS — CORE 13.5 AGENTS — CORE 13.7	DESCRIPTION Model evaluation is a critical component of the machine learning lifecycle. It provides data scientists with the tools to measure, interpret and explain the performance of their models. MLflow plays a critical role in accelerating model development by offering insights into the reasons behind a model's performance and guiding improvements and iterations. MLflow offers many industry-standard native evaluation metrics for classical machine learning algorithms and LLMs, and also facilitates the use of custom evaluation metrics. CONTROL CATEGORY: Implementation PRODUCT REFERENCE: AWS Azure GCP

CONTROL/RISK	DESCRIPTION OF CONTROL IMPLEMENTATION ON THE DATABRICKS PLATFORM
DASF 46 **Store and retrieve embeddings securely** RISKS AGENTS — CORE 13.1 AGENTS — TOOLS MCP SERVER 13.16 AGENTS — TOOLS MCP SERVER 13.18	DESCRIPTION Mosaic AI Vector Search is a vector database that is built into the Databricks Data Intelligence Platform and integrated with its governance and productivity tools. A vector database is a database that is optimized to store and retrieve embeddings. Embeddings are mathematical representations of the semantic content of data, typically text or image data. Embeddings are usually generated by feature extraction models for text, image, audio or multi-modal data, and are a key component of many GenAI applications that depend on finding documents or images that are similar to each other. Examples are RAG systems, recommender systems, and image and video recognition. Databricks implements the following security controls to protect your data: Every customer request to Vector Search is logically isolated, authenticated and authorized Mosaic AI Vector Search encrypts all data at rest (AES-256) and in transit (TLS 1.2+) and optionally can be encrypted with Customer Managed Keys (CMK). CONTROL CATEGORY: Implementation PRODUCT REFERENCE: AWS Azure GCP
DASF 47 **Compare LLM outputs on set prompts** RISKS	DESCRIPTION New, no-code visual tools allow users to compare models' output based on set prompts, which are automatically tracked within MLflow. With integration into Mosaic AI Model Serving, customers can deploy the best model to production. The AI Playground is a chat-like environment where you can test, prompt and compare LLMs. CONTROL CATEGORY: Implementation PRODUCT REFERENCE: AWS Azure GCP

CONTROL/RISK	DESCRIPTION OF CONTROL IMPLEMENTATION ON THE DATABRICKS PLATFORM
DASF 48 **Use hardened Runtime for Machine Learning** RISKS	DESCRIPTION Databricks Runtime for Machine Learning (Databricks Runtime ML) now automates cluster creation with versatile infrastructure, encompassing pre-built ML/DL libraries and custom library integration. Enhanced scalability and cost management tools optimize performance and expenditure. The refined user interface caters to various expertise levels, while new collaboration features support team-based projects. Comprehensive training resources and detailed documentation complement these improvements. CONTROL CATEGORY: Out-of-the-box PRODUCT REFERENCE: AWS Azure GCP
DASF 49 **Automate LLM evaluation** RISKS AGENTS — CORE 13.10 AGENTS — CORE 13.5 AGENTS — CORE 13.6 AGENTS — CORE 13.7	DESCRIPTION The “LLM-as-a-judge” feature in MLflow 2.8 automates LLM evaluation, offering a practical alternative to human judgment. It’s designed to be efficient and cost-effective, maintaining consistency with human scores. This tool supports various metrics, including standard and customizable GenAI metrics, and allows users to select an LLM as a judge and define specific grading criteria. CONTROL CATEGORY: Implementation PRODUCT REFERENCE: AWS Azure GCP

CONTROL/RISK	DESCRIPTION OF CONTROL IMPLEMENTATION ON THE DATABRICKS PLATFORM
DASF 50 **Platform compliance** RISKS	DESCRIPTION Develop your solutions on a platform created using some of the most rigorous security and compliance standards in the world. Get independent audit reports verifying that Databricks adheres to security controls for ISO 27001, ISO 27018, SOC 1, SOC 2, FedRAMP, HITRUST, IRAP, etc. Use platform with established data policy with trust and safety commitments. CONTROL CATEGORY: Out-of-the-box PRODUCT REFERENCE: AWS Azure GCP
DASF 51 **Share data and AI assets securely** RISKS	DESCRIPTION Databricks Delta Sharing lets you share data and AI assets securely in Databricks with users outside your organization, whether those users use Databricks or not. CONTROL CATEGORY: Out-of-the-box PRODUCT REFERENCE: AWS Azure GCP
DASF 52 **Source code control** RISKS	DESCRIPTION Databricks' Git Repository integration supports effective code and third-party libraries management, enhancing customer control over their development environment. CONTROL CATEGORY: Out-of-the-box PRODUCT REFERENCE: AWS Azure GCP

CONTROL/RISK	DESCRIPTION OF CONTROL IMPLEMENTATION ON THE DATABRICKS PLATFORM
DASF 53 **Third-party library control** RISKS AGENTS — TOOLS MCP CLIENT 13.35 AGENTS — TOOLS MCP SERVER 13.21	DESCRIPTION Databricks' library management system allows administrators to manage the installation and usage of third-party libraries effectively. This feature enhances the security and efficiency of systems, pipelines and data by giving administrators precise control over their development environment. CONTROL CATEGORY: Out-of-the-box PRODUCT REFERENCE: AWS Azure GCP
DASF 54 **Implement AI guardrails** RISKS AGENTS — CORE 13.1 AGENTS — CORE 13.15 AGENTS — CORE 13.2 AGENTS — CORE 13.3 AGENTS — CORE 13.5 AGENTS — CORE 13.6 AGENTS — CORE 13.7 AGENTS — CORE 13.8 AGENTS — CORE 13.9 AGENTS — TOOLS MCP CLIENT 13.32 AGENTS — TOOLS MCP SERVER 13.16 AGENTS — TOOLS MCP SERVER 13.18 AGENTS — TOOLS MCP SERVER 13.24	DESCRIPTION AI Guardrails allow users to configure and enforce data compliance at the model serving endpoint level and to reduce harmful content on any requests sent to the underlying model. Bad requests and responses are blocked, and a default message is returned to the user. You can configure and enforce data compliance at the model serving endpoint level and reduce harmful content on any requests sent to the underlying model with AI Guardrails. With AI Guardrails, you can configure the following controls on your AI system: - Safety filtering prevents your model from interacting with unsafe and harmful content, like violent crime, self-harm, and hate speech. - Personally identifiable information (PII) detection to detect any sensitive information (such as names, addresses, and credit card numbers) for users - Topic moderation to list a set of allowed topics. Given a chat request, this guardrail flags the request if its topic is not one of the permitted topics. - Keyword filtering will specify different sets of invalid keywords for both the input and the output. One potential use case for keyword filtering is so the model does not talk about competitors. CONTROL CATEGORY: Implementation PRODUCT REFERENCE: AWS Azure GCP

CONTROL/RISK	DESCRIPTION OF CONTROL IMPLEMENTATION ON THE DATABRICKS PLATFORM

DASF 55

Monitor audit logs

RISKS

AGENTS — CORE 13.1 | AGENTS — CORE 13.11 | AGENTS — CORE 13.12 | AGENTS — CORE 13.13 | AGENTS — CORE 13.14 | AGENTS — CORE 13.15 | AGENTS — CORE 13.2 | AGENTS — CORE 13.3 | AGENTS — CORE 13.4 | AGENTS — CORE 13.6 | AGENTS — CORE 13.7 | AGENTS — CORE 13.8 | AGENTS — CORE 13.9 | AGENTS — TOOLS MCP CLIENT 13.26 | AGENTS — TOOLS MCP CLIENT 13.27 | AGENTS — TOOLS MCP CLIENT 13.28 | AGENTS — TOOLS MCP CLIENT 13.29 | AGENTS — TOOLS MCP CLIENT 13.30 | AGENTS — TOOLS MCP CLIENT 13.31 | AGENTS — TOOLS MCP CLIENT 13.32 | AGENTS — TOOLS MCP CLIENT 13.33 | AGENTS — TOOLS MCP CLIENT 13.34 | AGENTS — TOOLS MCP CLIENT 13.35 | AGENTS — TOOLS MCP SERVER 13.16 | AGENTS — TOOLS MCP SERVER 13.17 | AGENTS — TOOLS MCP SERVER 13.18 | AGENTS — TOOLS MCP SERVER 13.19 | AGENTS — TOOLS MCP SERVER 13.20 | AGENTS — TOOLS MCP SERVER 13.21 | AGENTS — TOOLS MCP SERVER 13.22 | AGENTS — TOOLS MCP SERVER 13.23 | AGENTS — TOOLS MCP SERVER 13.24 | AGENTS — TOOLS MCP SERVER 13.25

DESCRIPTION

Audit logs and system tables serve as a centralized operational data store, backed by Delta Lake and governed by Unity Catalog. Audit logs and system tables can be used for a variety of purposes, from user activity, model serving events, and cost monitoring to audit logging. Databricks recommends that customers configure system tables and set up automated monitoring and alerting to meet their needs. The blog post Improve Lakehouse Security Monitoring Using System Tables in Databricks Unity Catalog is a good starting point to help customers get started. Customers that are using Enhanced Security Monitoring or the Compliance Security Profile can monitor and alert on suspicious activity detected by the behavior-based malware and file integrity monitoring agents.

CONTROL CATEGORY

Configuration

PRODUCT REFERENCE

AWS Azure GCP

CONTROL/RISK	DESCRIPTION OF CONTROL IMPLEMENTATION ON THE DATABRICKS PLATFORM
DASF 56 **Restrict outbound connections from models** RISKS AGENTS — CORE 13.11 AGENTS — CORE 13.13 AGENTS — CORE 13.2 AGENTS — CORE 13.3 AGENTS — CORE 13.6 AGENTS — CORE 13.7 AGENTS — TOOLS MCP SERVER 13.16 AGENTS — TOOLS MCP SERVER 13.18 AGENTS — TOOLS MCP SERVER 13.23	DESCRIPTION Egress Control enables you to control outbound connections from your Model Serving compute resources. With this feature, you can restrict access to the internet while allowing access via Unity Catalog Connections or Private Link. Further, this feature blocks direct access to cloud storage (over the shared S3 gateway) to ensure that all data access occurs via Unity Catalog-controlled paths to reduce the risk of data exfiltration. CONTROL CATEGORY: Configuration PRODUCT REFERENCE: AWS Azure GCP
DASF 57 **Use attribute-based access controls (ABAC)** RISKS	DESCRIPTION Attribute-based access controls (ABAC) allow data stewards to set policies on data and AI assets using various criteria like user-defined tags, workspace details, location, identity and time. Whether it's restricting sensitive data to authorized personnel or adjusting access dynamically based on project needs, ABAC ensures security measures are applied with detailed accuracy. Implement attribute-based access controls (ABAC) to define access policies based on attributes or characteristics of the user or the resource being accessed. Use row level filters and column masking for fine-grained access controls. CONTROL CATEGORY: Implementation PRODUCT REFERENCE: AWS Azure GCP

CONTROL/RISK	DESCRIPTION OF CONTROL IMPLEMENTATION ON THE DATABRICKS PLATFORM
DASF 58 **Protect data with filters and masking** RISKS	DESCRIPTION Implement filters on sensitive table data using row filters and column masks by harnessing the power of Unity Catalog to secure your data at a granular level. Row filters allow you to apply a filter to a table so that queries return only rows that meet the filter criteria. Column masks let you apply a masking function to a table column. CONTROL CATEGORY: Implementation PRODUCT REFERENCE: AWS Azure GCP
DASF 59 **Use clean rooms** RISKS	DESCRIPTION Building AI applications today necessitates collaborative efforts across organizations and teams, emphasizing a commitment to privacy and data security. Databricks Clean Rooms offer a secure environment for private collaboration on diverse data and AI tasks, spanning machine learning, SQL queries, Python, R and more. Designed to facilitate seamless collaboration across different cloud and data platforms, Databricks Clean Rooms ensure multiparty collaboration without compromising data privacy or security and enables organizations to build scalable AI applications in a privacy-safe manner. CONTROL CATEGORY: Implementation PRODUCT REFERENCE: AWS Azure GCP
DASF 60 **Rate limit number of inference queries** RISKS AGENTS — CORE 13.10 AGENTS — CORE 13.4 AGENTS — CORE 13.5 AGENTS — TOOLS MCP SERVER 13.16 AGENTS — TOOLS MCP SERVER 13.18	DESCRIPTION Enforce request rate limits to manage traffic at the endpoint level on a per-user and per-endpoint basis, effectively controlling access levels and volume. CONTROL CATEGORY: Configuration PRODUCT REFERENCE: AWS Azure GCP

CONTROL/RISK	DESCRIPTION OF CONTROL IMPLEMENTATION ON THE DATABRICKS PLATFORM
DASF 61 **Train users on AI risk taxonomy and AI/ML security** RISKS AGENTS — TOOLS MCP CLIENT 13.28 AGENTS — TOOLS MCP CLIENT 13.30 AGENTS — TOOLS MCP CLIENT 13.31	DESCRIPTION Provide secure coding education and AI vulnerability awareness for model developers. Training personnel who manage AI infrastructure on cybersecurity best practices is essential to prevent human errors that could lead to security breaches. Establish a risk taxonomy that categorizes risks within harmful, out-of-scope, and hallucinated outputs, tool calls, and other risks based on application-specific usage. https://www.databricks.com/trust/responsibleAI https://www.databricks.com/trust/ai-security CONTROL CATEGORY Databricks - academy PRODUCT REFERENCE AWS Azure GCP
DASF 62 **Implement network segmentation** RISKS AGENTS — CORE 13.11 AGENTS — CORE 13.13 AGENTS — TOOLS MCP CLIENT 13.32 AGENTS — TOOLS MCP SERVER 13.25	DESCRIPTION Establish network and security policies to define and enforce egress rules from models and outbound network connections from your serverless compute resources. CONTROL CATEGORY Configuration PRODUCT REFERENCE AWS Azure GCP
DASF 63 **Update software** RISKS AGENTS — CORE 13.11 AGENTS — TOOLS MCP CLIENT 13.33 AGENTS — TOOLS MCP CLIENT 13.35 AGENTS — TOOLS MCP SERVER 13.21	DESCRIPTION Patch and update software regularly to address existing and potential vulnerabilities throughout the AI lifecycle. In Databricks, automatic cluster updates ensure that all clusters within a workspace receive the latest OS images and security patches periodically. Account administrators can customize the maintenance window frequency, start date, and time. CONTROL CATEGORY Out-of-the-box PRODUCT REFERENCE AWS Azure GCP

CONTROL/RISK	DESCRIPTION OF CONTROL IMPLEMENTATION ON THE DATABRICKS PLATFORM
DASF 64 **Limit access from AI models and agents** RISKS AGENTS — CORE 13.1, AGENTS — CORE 13.11, AGENTS — CORE 13.12, AGENTS — CORE 13.13, AGENTS — CORE 13.14, AGENTS — CORE 13.15, AGENTS — CORE 13.2, AGENTS — CORE 13.3, AGENTS — CORE 13.6, AGENTS — CORE 13.7, AGENTS — CORE 13.9, AGENTS — TOOLS MCP CLIENT 13.28, AGENTS — TOOLS MCP CLIENT 13.30, AGENTS — TOOLS MCP CLIENT 13.31, AGENTS — TOOLS MCP CLIENT 13.34, AGENTS — TOOLS MCP SERVER 13.16, AGENTS — TOOLS MCP SERVER 13.17, AGENTS — TOOLS MCP SERVER 13.18, AGENTS — TOOLS MCP SERVER 13.19, AGENTS — TOOLS MCP SERVER 13.20, AGENTS — TOOLS MCP SERVER 13.22, AGENTS — TOOLS MCP SERVER 13.23, AGENTS — TOOLS MCP SERVER 13.24	DESCRIPTION Allow AI models and agents access to enterprise resources based on the principle of least privilege. In Unity Catalog, a "securable object" is any object that can be permissioned to a principal (e.g., user, service principal, or group) and is organized hierarchically. Treat AI as a principal and assign permissions accordingly. With on-behalf-of-user authentication, agents deployed via Mosaic AI model serving can access Databricks resources using the identity of the Databricks end user who queried the agent. This enables accessing sensitive information on a per-user basis, with fine-grained enforcement of data access control in Unity Catalog. CONTROL CATEGORY: Implementation PRODUCT REFERENCE: AWS Azure GCP
DASF 65 **Implement end-to-end AI traceability** RISKS AGENTS — CORE 13.1, AGENTS — CORE 13.12, AGENTS — CORE 13.13, AGENTS — CORE 13.14, AGENTS — CORE 13.15, AGENTS — CORE 13.2, AGENTS — CORE 13.3, AGENTS — CORE 13.5, AGENTS — CORE 13.6, AGENTS — CORE 13.7, AGENTS — CORE 13.8, AGENTS — CORE 13.9, AGENTS — TOOLS MCP CLIENT 13.28, AGENTS — TOOLS MCP SERVER 13.16, AGENTS — TOOLS MCP SERVER 13.17, AGENTS — TOOLS MCP SERVER 13.18	DESCRIPTION MLflow Tracing is a powerful feature that provides end-to-end observability for gen AI applications, including complex agent-based systems. It records inputs, outputs, intermediate steps, and metadata to give you a complete picture of how your app behaves. Tracing allows you to: • Debug and understand your application • Monitor performance and optimize cost • Evaluate and enhance application quality • Ensure auditability and compliance • Integrate tracing with many popular third-party frameworks CONTROL CATEGORY: Implementation PRODUCT REFERENCE: AWS Azure GCP

CONTROL/RISK	DESCRIPTION OF CONTROL IMPLEMENTATION ON THE DATABRICKS PLATFORM
DASF 66 **Use Human-in-the-loop feedback** RISKS AGENTS — CORE 13.10 AGENTS — CORE 13.15 AGENTS — CORE 13.5 AGENTS — CORE 13.6 AGENTS — CORE 13.7 AGENTS — TOOLS MCP CLIENT 13.28	DESCRIPTION End-user feedback is invaluable for understanding how your GenAI application performs in real-world scenarios. MLflow provides tools to capture, store, and analyze feedback directly from the users of your deployed applications. CONTROL CATEGORY: Implementation PRODUCT REFERENCE: AWS Azure GCP
DASF 67 **Federate authentication** RISKS AGENTS — CORE 13.1 AGENTS — CORE 13.12 AGENTS — CORE 13.13 AGENTS — CORE 13.14 AGENTS — CORE 13.2 AGENTS — CORE 13.3 AGENTS — CORE 13.6 AGENTS — CORE 13.7 AGENTS — CORE 13.8 AGENTS — CORE 13.9 AGENTS — TOOLS MCP CLIENT 13.27 AGENTS — TOOLS MCP CLIENT 13.28 AGENTS — TOOLS MCP CLIENT 13.29 AGENTS — TOOLS MCP CLIENT 13.31 AGENTS — TOOLS MCP CLIENT 13.34 AGENTS — TOOLS MCP SERVER 13.16 AGENTS — TOOLS MCP SERVER 13.17 AGENTS — TOOLS MCP SERVER 13.18 AGENTS — TOOLS MCP SERVER 13.19 AGENTS — TOOLS MCP SERVER 13.20 AGENTS — TOOLS MCP SERVER 13.22 AGENTS — TOOLS MCP SERVER 13.23 AGENTS — TOOLS MCP SERVER 13.24	DESCRIPTION OAuth token federation is a simpler and more secure method for authenticating to Databricks, especially for automated workloads. Token federation allows applications to authenticate to Databricks using tokens from your trusted IdP, eliminating the need to store Databricks secrets like static tokens or passwords. Databricks OAuth token federation enables your users and applications to securely access Databricks AI APIs using tokens from your identity provider (IdP). Workloads authenticate to Databricks as a service principal in the Databricks account, using workload identity tokens issued by the automation environment. The Databricks SDKs and Databricks CLI automatically fetch these workload identity tokens and exchange them for Databricks OAuth tokens, which eliminates the need manage and rotate Databricks secrets. CONTROL CATEGORY: Configuration PRODUCT REFERENCE: AWS Azure GCP

CONTROL/RISK	DESCRIPTION OF CONTROL IMPLEMENTATION ON THE DATABRICKS PLATFORM

DASF 68

Use Securely Hosted Managed MCP Servers

RISKS

- AGENTS — TOOLS MCP CLIENT 13.26
- AGENTS — TOOLS MCP CLIENT 13.27
- AGENTS — TOOLS MCP CLIENT 13.28
- AGENTS — TOOLS MCP CLIENT 13.29
- AGENTS — TOOLS MCP CLIENT 13.31
- AGENTS — TOOLS MCP CLIENT 13.32
- AGENTS — TOOLS MCP CLIENT 13.33
- AGENTS — TOOLS MCP CLIENT 13.34
- AGENTS — TOOLS MCP CLIENT 13.35
- AGENTS — TOOLS MCP SERVER 13.16
- AGENTS — TOOLS MCP SERVER 13.17
- AGENTS — TOOLS MCP SERVER 13.18
- AGENTS — TOOLS MCP SERVER 13.19
- AGENTS — TOOLS MCP SERVER 13.20
- AGENTS — TOOLS MCP SERVER 13.21
- AGENTS — TOOLS MCP SERVER 13.22
- AGENTS — TOOLS MCP SERVER 13.23
- AGENTS — TOOLS MCP SERVER 13.24
- AGENTS — TOOLS MCP SERVER 13.25

DESCRIPTION

Model Context Protocol (MCP) servers act as bridges that let AI agents access external data and tools. Instead of building these connections from scratch, you can use securely hosted Databricks managed MCP servers to instantly connect your agents to data stored in Unity Catalog, vector search indexes, and custom functions.

CONTROL CATEGORY

Out-of-the-box

PRODUCT REFERENCE

AWS Azure GCP

CONTROL/RISK	DESCRIPTION OF CONTROL IMPLEMENTATION ON THE DATABRICKS PLATFORM

DASF 69

Securely host Custom MCP Servers

RISKS

- AGENTS — TOOLS MCP CLIENT 13.26
- AGENTS — TOOLS MCP CLIENT 13.27
- AGENTS — TOOLS MCP CLIENT 13.28
- AGENTS — TOOLS MCP CLIENT 13.29
- AGENTS — TOOLS MCP CLIENT 13.31
- AGENTS — TOOLS MCP CLIENT 13.32
- AGENTS — TOOLS MCP CLIENT 13.33
- AGENTS — TOOLS MCP CLIENT 13.34
- AGENTS — TOOLS MCP CLIENT 13.35
- AGENTS — TOOLS MCP SERVER 13.18
- AGENTS — TOOLS MCP SERVER 13.19
- AGENTS — TOOLS MCP SERVER 13.20
- AGENTS — TOOLS MCP SERVER 13.21
- AGENTS — TOOLS MCP SERVER 13.22
- AGENTS — TOOLS MCP SERVER 13.23
- AGENTS — TOOLS MCP SERVER 13.24
- AGENTS — TOOLS MCP SERVER 13.25

DESCRIPTION

Host your own custom or third-party MCP servers as secure Databricks apps. Custom MCP servers are useful if you already have an MCP server you want to deploy or if you want to run a third-party MCP server as a source of tools.

CONTROL CATEGORY

Implementation

PRODUCT REFERENCE

AWS Azure GCP

CONTROL/RISK	DESCRIPTION OF CONTROL IMPLEMENTATION ON THE DATABRICKS PLATFORM
DASF 70 **Securely connect to External MCP Servers** RISKS AGENTS — TOOLS MCP CLIENT 13.26 AGENTS — TOOLS MCP CLIENT 13.27 AGENTS — TOOLS MCP CLIENT 13.28 AGENTS — TOOLS MCP CLIENT 13.29 AGENTS — TOOLS MCP CLIENT 13.31 AGENTS — TOOLS MCP CLIENT 13.32 AGENTS — TOOLS MCP CLIENT 13.33 AGENTS — TOOLS MCP CLIENT 13.34 AGENTS — TOOLS MCP SERVER 13.20 AGENTS — TOOLS MCP SERVER 13.21 AGENTS — TOOLS MCP SERVER 13.22 AGENTS — TOOLS MCP SERVER 13.23 AGENTS — TOOLS MCP SERVER 13.24 AGENTS — TOOLS MCP SERVER 13.25	DESCRIPTION Connect Databricks to external Model Context Protocol (MCP) servers to give your agents access to a wider range of tools that are hosted outside of Databricks. Databricks uses managed MCP proxies and Unity Catalog HTTP connections to securely handle authentication to your workspace. With OAuth, Databricks automatically handles token exchange and refreshes on expiry—users don't need to manage tokens manually. CONTROL CATEGORY Configuration PRODUCT REFERENCE AWS Azure GCP

CONTROL/RISK	DESCRIPTION OF CONTROL IMPLEMENTATION ON THE DATABRICKS PLATFORM
DASF 71 **Log and register AI agents** RISKS AGENTS — CORE 13.1 AGENTS — CORE 13.11 AGENTS — CORE 13.12 AGENTS — CORE 13.13 AGENTS — CORE 13.14 AGENTS — CORE 13.15 AGENTS — CORE 13.2 AGENTS — CORE 13.3 AGENTS — CORE 13.6 AGENTS — CORE 13.7 AGENTS — CORE 13.8 AGENTS — CORE 13.9 AGENTS — TOOLS MCP SERVER 13.17 AGENTS — TOOLS MCP SERVER 13.19	DESCRIPTION Logging an agent and registering it with catalog is the basis of the development and deployment process of agentica AI. The use of the Databricks AI Agent Framework, leveraging MLflow, to log and register all AI agents. Logging is the foundational step, capturing a specific "point in time" version of the agent's code, configuration, and Python environment to ensure traceability and evaluation quality. The standard logging mechanism is MLflow Models from Code, which ensures the agent is packaged reliably for deployment. Key requirements during logging include: 1. Model Signature: Defining the MLflow Model Signature to validate agent inputs and outputs, ensuring correct interaction with tools and other downstream applications. 2. Resource Declaration: Explicitly declaring all external Databricks-managed resources (e.g., Vector Search indices, Foundation Model Serving Endpoints) that the agent requires for execution. Following successful logging, the agent must be registered to Unity Catalog (UC). This registration packages the agent as a model within UC, enabling the use of Unity Catalog permissions for authorization over the agent and its associated resources. CONTROL CATEGORY: Implementation PRODUCT REFERENCE: AWS Azure GCP
DASF 72 **Securely store and reuse agent state** RISKS AGENTS — CORE 13.1 AGENTS — TOOLS MCP CLIENT 13.34	DESCRIPTION Managing Agent state is critical to secure multi-agent systems because it provides a centralized, protected layer for agents to store their short-term memory (such as conversation history, decision logs, or temporary tasks) directly within this governed perimeter, rather than exporting sensitive data to disconnected, insecure external databases. CONTROL CATEGORY: Implementation PRODUCT REFERENCE: AWS Azure

CONTROL/RISK	DESCRIPTION OF CONTROL IMPLEMENTATION ON THE DATABRICKS PLATFORM
DASF 73 **Register prompts** RISKS AGENTS — CORE 13.15 AGENTS — CORE 13.2 AGENTS — CORE 13.6 AGENTS — CORE 13.7 AGENTS — TOOLS MCP SERVER 13.16	DESCRIPTION Although not quite as strong a control as a good safety filtering, prompt injection or jailbreak detection model, a good system prompt can sometimes be the difference between an attack being successful or not. MLflow prompt registry is a powerful tool that streamlines prompt engineering and management in your GenAI applications. It enables you to version, track, test and reuse prompts across your organization, helping maintain consistency and improving collaboration in effective prompt development. CONTROL CATEGORY Implementation PRODUCT REFERENCE AWS Azure GCP

05

Conclusion

The transition to Agentic AI represents a pivotal moment in the adoption of artificial intelligence. By granting models the agency to use tools and make decisions, we unlock immense potential for automation and productivity. However, this autonomy necessitates a corresponding evolution in our security frameworks. We can no longer secure only the data warehouse and the model endpoint. We must expand our focus to secure the decision, the plan, and the action.

This extension to the **Databricks AI Security Framework (DASF)** provides the blueprint for this evolution. By formally recognizing Agentic AI as a critical system component and dissecting the risks inherent in memory, planning, and tool use (e.g., MCP), we provide organizations with the clarity needed to innovate safely.

We encourage you to download the updated **DASF Compendium** (**Google sheet**, **Excel**), which now includes the full list of Agentic risks and their corresponding controls. Use this resource to:

1. **Assess** your current agent architectures against the "13th Component" risk model.
2. **Map** your tool ecosystems (including MCP servers and clients) to the identified threat vectors.
3. **Implement** the recommended controls to ensure your agents operate within safe, governed boundaries.

As Agentic systems continue to mature, the collaboration between security, data, and AI teams will be the defining factor in their success. With the DASF as your guide, you can build agents that are not only intelligent and capable but also secure and trustworthy.

06

Resources and Further Reading

We've discussed many different capabilities in this document, with documentation links where possible. Organizations that prioritize high security can learn more than what's in this document. Here are additional resources to dive deeper.

Databricks resources

- **Databricks AI Security Framework 2.0**
- DASF compendium document (**Google sheet**, **Excel sheet**) - updated for Agentic AI
- **AI Without Fear: A Practical Framework to Manage Risk** - an Ebook providing executives with an overview of DASF and its role in driving AI adoption, governance, and collaboration
- **Databricks AI Governance Framework 1.0** - a comprehensive guide to implementing enterprise AI programs responsibly and effectively
- **Delivering Securely on Data and AI Strategy** - read how top security leaders are safely accelerating the adoption of AI use cases.
- Databricks Documentation:
- **Mosaic AI Agent Framework**
- **Agent Bricks**
- **Model Context Protocol (MCP) on Databricks**
- **Generative AI observability**
- **Deploy agents with tracing**
- **Agentic AI Glossary**

Industry resources

Industry resources such as the CSA MCP Security Resource Center and OWASP Agentic AI – Threats and Mitigations provide in depth analysis and practical guidance for Agentic AI systems across their full lifecycle.

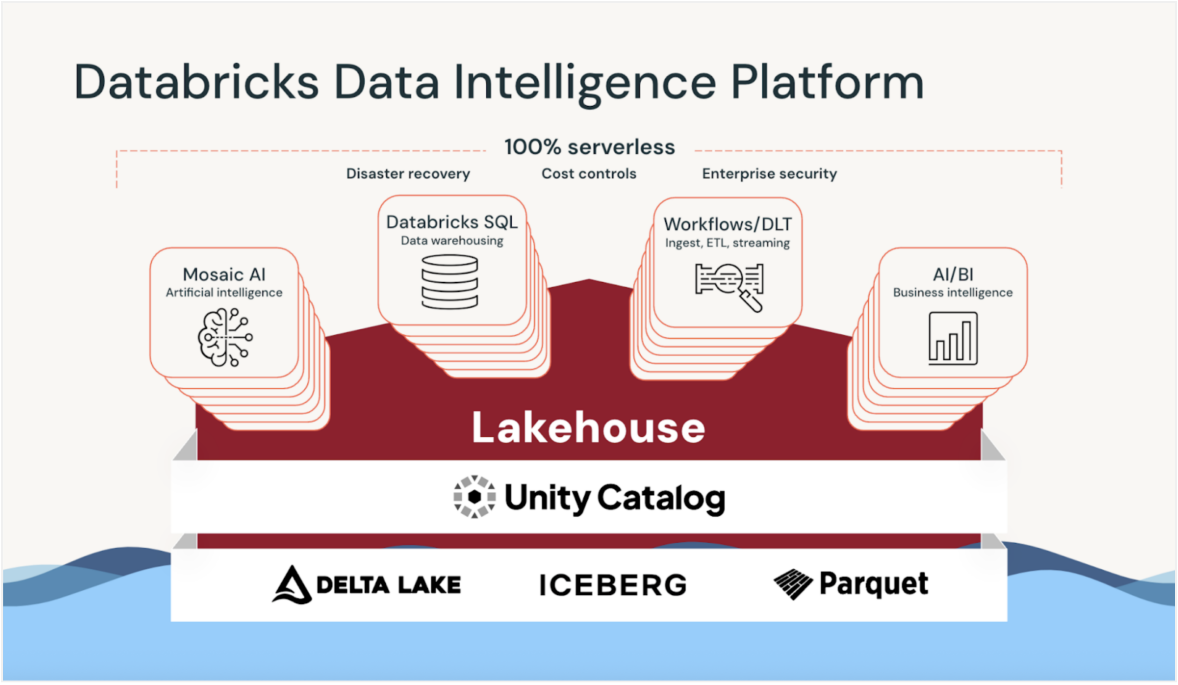
- **CSA MCP Security Resource Center**
- **OWASP Agentic AI – Threats and Mitigations**
- **The Lethal Trifecta for AI Agents**
- **MAESTRO Framework** — CSA (Cloud Security Alliance) threat modeling framework for multi-agent AI systems
- **OWASP MCP Top 10** — forthcoming catalog of the top 10 security risks specific to the Model Context Protocol

07 Appendix: Understanding the Databricks Platform

Databricks is the data and AI company with origins in academia and the open source community. Databricks was founded in 2013 by the original creators of **Apache Spark™**, **Delta Lake** and **MLflow**. We pioneered the concept of the **lakehouse** to combine and unify the best of data warehouses and data lakes. Databricks made this vision a reality in 2020; since then, it has seen tremendous adoption as a category. Today, 74% of global CIOs report having a lakehouse in their estate, and almost all of the remainder intend to have one within the next three years.

In November 2023, we announced the **Databricks Data Intelligence Platform**. It's built on a lakehouse to provide an open, unified foundation for all data and governance. We built the Data Intelligence Platform to allow every employee in every organization to find success with data and AI. At the heart of the platform is a Data Intelligence Engine, **DatabricksIQ**, that understands the semantics of your data and how it flows across all of your workloads. This allows for new methods of optimization, as well as for technical and nontechnical users to use natural language to discover and use data and AI in the context of your business.

In this section, we provide an overview of our platform and its architecture and components related to governance, security, and AI and machine learning.



The Databricks Data Intelligence Platform combines AI assets — from data and features to models — into one catalog, ensuring full visibility and fine-grained control throughout the AI

workflow. We provide automatic lineage tracking, centralized governance and seamless cross-workspace collaboration for simplified MLOps and enhanced productivity. Furthermore, we give customers complete control and ownership of their data and models with privacy controls to maintain compliance as well as efficiency and granular models on their data, fine-tuned at lower costs.

The Databricks Data Intelligence Platform offers a secure, unified and data-centric solution for both MLOps and LLMOps, adopting a defense-in-depth approach to implementing security across all AI system components. As shown in the following diagram, the platform aligns seamlessly with the AI system components defined in the DASF, supporting key stages — from data preparation to serving infrastructure — through products like Delta Live Tables, Unity Catalog and Mosaic AI.

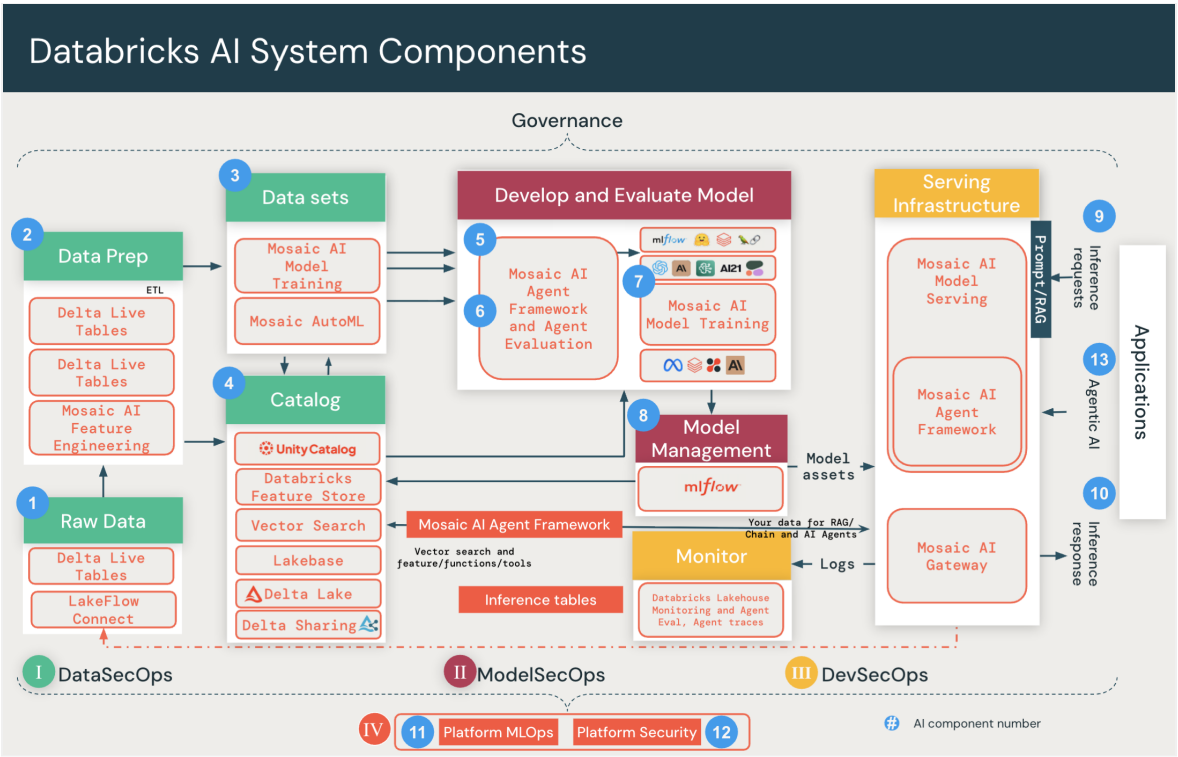


Diagram of all AI System Components covered in the full Databricks AI Security Framework

Databricks Platform architecture

Databricks is a platform as a service (PaaS) general-purpose data-agnostic compute platform.

We use the phrase “hybrid PaaS” because our lakehouse architecture is split into two separate planes to simplify your permissions, avoid data duplication and reduce risk. The control plane is the management plane where Databricks runs the workspace application and manages notebooks, configuration and clusters. The compute plane handles your data processing. Customers deploy a compute plane (virtual network and compute) in a cloud service provider account (such as AWS, Azure or GCP) that the customer owns. With serverless deployments, the compute plane exists in the customer’s Databricks account rather than their cloud service provider account. Customers get the benefits of PaaS with the option to keep their data processing clusters locally within their environment.

The phrase “general-purpose data-agnostic” means that you can use Databricks services for any type of data and purpose that you need. If you’re new to Databricks or the lakehouse architecture, start with an overview of the architecture and a review of common security questions before you hop into specific recommendations. You’ll see those in our **Security and Trust Center** and the **Security and Trust Overview Whitepaper**.

Controls addressed by Databricks Platform architecture:

- DASF 1 — SSO with IdP and MFA
- DASF 2 — Sync users and groups
- DASF 3 — Restrict access using IP access lists
- DASF 4 — Restrict access using private link
- DASF 5 — Control access to data and other objects
- DASF 56 — Restrict outbound connections from models
- DASF 62 — Implement network segmentation
- DASF 64 — Limit access from AI models and agents

Delta Lake

Delta Lake is the optimized storage layer that provides the foundation for tables in a lakehouse on Databricks. Delta Lake is **open source software** that extends Parquet data files with a file-based transaction log for **ACID transactions** and scalable metadata handling. It’s fully compatible with Apache Spark APIs and was developed for tight integration with Structured Streaming, allowing you to easily use a single copy of data for both batch and streaming operations and providing incremental processing at scale.

Universal Format (UniForm), is a feature of Delta Lake that takes advantage of the inherent similarities among the three open table formats. Delta Lake, Iceberg and Apache Hudi all store data in the **Apache Parquet** file format but diverge in how they store additional metadata. UniForm generates Iceberg metadata alongside Delta Lake while maintaining a single copy of the Parquet files. By writing once to Delta Lake, you can access your data using any engine that supports any one of the open formats. Additional details:

- Delta Lake is the default format for all operations on Databricks. Unless otherwise specified, all tables on Databricks are Delta tables.
- Delta Lake is deeply integrated with **Spark Structured Streaming** through readStream and writeStream. Delta Lake overcomes **many of the limitations** typically associated with streaming systems and files.

- Additionally, you can use **Auto Loader** to incrementally and efficiently process new data files as they arrive in cloud storage. It provides a Structured Streaming source called cloudFiles. Given an input directory path on the cloud file storage, the cloudFiles source automatically processes new files as they arrive, with the option of also processing existing files in that directory. Auto Loader has support for both Python and SQL in Delta Live Tables. Auto Loader can process billions of files to migrate or backfill a table. Auto Loader scales to support near real-time ingestion of millions of files per hour.
- **Liquid clustering** replaces table partitioning and Z-Ordering to simplify data layout decisions and optimize query performance. Liquid clustering provides flexibility to redefine clustering keys without rewriting existing data, allowing data layout to evolve alongside analytic needs over time. Databricks recommends liquid clustering for all new Delta tables.
- **Predictive optimization** automatically optimizes your data for the best performance and price. It learns from your data usage patterns, builds a plan for the right optimizations to perform and then runs those optimizations on hyper-optimized serverless infrastructure.

Controls addressed by Delta Lake:

DASF 7	Enforce data quality checks on batch and streaming datasets
DASF 10	Version data
DASF 12	Delete records from datasets
DASF 13	Use near real-time data

Databricks Unity Catalog

Databricks **Unity Catalog** is the industry's only unified and open governance solution for managing data and AI assets across any lakehouse format or data source. Unity Catalog's security model is based on standard ANSI SQL and allows administrators to grant permissions in their existing data lake using familiar syntax at the level of catalogs, schemas (also called databases), tables and views. With Unity Catalog, data scientists, analysts and engineers can seamlessly govern their structured and unstructured data, machine learning models, notebooks, dashboards and files on any cloud or platform. This unified approach to governance accelerates data and AI initiatives while ensuring regulatory compliance in a simplified manner.
Unity Catalog provides key capabilities like:

- **Access control for data and AI assets with a single permission model:** Unity Catalog simplifies access management with a unified interface to define access policies on data and AI assets and consistently apply and audit these policies on any cloud or data platform. You can access data from other computing platforms using open interfaces, with consistent permissions managed in one place.

- **Open data sharing and collaboration:** Easily share data and AI assets across clouds, regions and platforms with open source **Delta Sharing**, natively integrated within Unity Catalog. Securely **collaborate** with anyone, anywhere to unlock new revenue streams and drive business value without relying on proprietary formats, complex ETL processes or costly data replication.
- **Centralized data search and discovery:** Quickly find, understand and reference data from across your data estate, boosting productivity. Data search in Unity Catalog is secure by default, limiting search results based on access privileges of the users and adding an additional layer of security for privacy considerations.
- **Tags** are attributes that include keys and optional values that you can use to organize and categorize securable objects in Unity Catalog. They simplify search and discovery of tables and views.
- **Automated lineage:** You can use Unity Catalog to capture runtime **data lineage** across queries in any language executed on a Databricks cluster or SQL warehouse. Lineage is captured down to the column level, and includes notebooks, jobs and dashboards related to the query. Lineage can be retrieved via REST APIs to support integrations with our catalog partners.
- **Historical observability across your account with system tables: System tables** are a Databricks-hosted analytical store of your account's operational data found in the system catalog. Unity Catalog lets you easily access and query your account's operational data, including audit logs, billable usage and lineage.
- **Built-in auditing:** Unity Catalog automatically captures user-level **audit logs** that record access to your data
- **Row filters and column masking:** This capability allows you to secure and govern your data at the granular level.
- Row filters allow you to apply a filter to a table. You implement a row filter as a **SQL user-defined function (UDF)**. Python and Scala UDFs are also supported, but only when they are wrapped in SQL UDFs.
- Column masks let you apply a masking function to a table column. The masking function evaluates at query runtime, substituting each reference of the target column with the results of the masking function. For most use cases, column masks determine whether to return the original column value or redact it based on the identity of the invoking user.
- **Attribute-based access controls (ABAC) — in Private Preview: ABAC** offers organizations a high-leverage governance solution that simplifies the enforcement of governance policies across their entire lakehouse. By employing straightforward rules and tags, ABAC ensures consistent governance across all data sources, whether native to Databricks or federated from external sources. With ABAC, users can establish access controls tailored to specific attributes of resources like workspaces, data assets such as tables, and AI assets. These attributes encompass a wide range of parameters, including user-defined tags, workspace details, location, identity and time. Whether it's ensuring sensitive data remains restricted to authorized personnel or dynamically adjusting access based on changing project requirements, ABAC empowers users to enforce security measures with granular precision.

- **Model management:** **Models in Unity Catalog** extends the benefits of Unity Catalog to ML models, including centralized access control, auditing, lineage and model discovery across workspaces
- **Lakehouse monitoring:** **Databricks Lakehouse Monitoring** lets you monitor the statistical properties and quality of the data in all of the tables in your account. You can also use it to track the performance of machine learning models and model-serving endpoints by monitoring inference tables that contain model inputs and predictions.
- **Lakehouse Federation:** **Lakehouse Federation** is the query federation platform for Databricks. The term "query federation" describes a collection of features that enable users and systems to run queries against multiple data sources without needing to migrate all data to a unified system. Databricks uses Unity Catalog to manage query federation. You configure read-only connections to popular database solutions using drivers that are included on pro SQL warehouses, serverless SQL warehouses and Databricks Runtime clusters.

Controls addressed by Databricks Unity Catalog:

- DASF 2 Sync users and groups
- DASF 6 Classify data
- DASF 11 Capture and view data lineage
- DASF 12 Delete records from datasets
- DASF 14 Audit actions performed on datasets
- DASF 16 Secure model features
- DASF 17 Track and reproduce the training data used for ML model training
- DASF 18 Govern model assets
- DASF 21 Monitor data and AI system from a single pane of glass
- DASF 23 Register, version, approve, promote, deploy and monitor models
- DASF 24 Control access to models and model assets
- DASF 25 Use retrieval augmented generation (RAG) with large language models (LLMs)
- DASF 28 Create model aliases, tags and annotations
- DASF 30 Encrypt models
- DASF 32 Govern and monitor access of AI model and model serving endpoints
- DASF 35 Track model performance
- DASF 37 Set up inference tables for monitoring and debugging models
- DASF 46 Store and retrieve embeddings securely
- DASF 47 Compare LLM outputs on set prompts
- DASF 51 Share data and AI assets securely
- DASF 53 Third-party library control
- DASF 54 Implement AI guardrails
- DASF 55 Monitor audit logs
- DASF 57 Use attribute-based access controls (ABAC)
- DASF 58 Protect data with filters and masking
- DASF 67 Federate authentication
- DASF 71 Log and register AI agents
- DASF 72 Securely store and reuse agent state

Databricks Platform security

Data and AI are your most valuable assets and always have to be protected — that's why security is built into every layer of the Databricks Data Intelligence Platform. Databricks security is based on three core principles: trust, technology and transparency.

- Trust — Third-party audit firms regularly audit Databricks systems and processes. Databricks customers can trust independent validation of internal security processes.
- Technology — Databricks deploys modern technology solutions combined with secure processes across the enterprise to maximize security. Security design and tools are applied throughout. Databricks considers security in the platform architecture design, network security processes, automated penetration testing on the production systems and vulnerability scanning tools during development.
- Transparency — Databricks provides customers with full attestation reports (for example, SOC 2 Type 2), certifications (for example, ISO 27001) and detailed architecture overviews. Our transparency enables you to meet your regulatory needs while taking advantage of our platform.

Our Databricks Security team regularly works with customers to securely deploy AI systems on our platform with the appropriate security and governance features. We understand how ML systems are designed for security, teasing out possible security engineering risks and making such risks explicit. Databricks is committed to providing a data intelligence platform where business stakeholders, data engineers, data scientists, ML engineers, data governance officers and data analysts can trust that their data and AI models are secure.

Controls addressed by Databricks Platform security:

- DASF 1 SSO with IdP and MFA
- DASF 2 Sync users and groups
- DASF 3 Restrict access using IP access lists
- DASF 4 Restrict access using private link
- DASF 5 Control access to data and other objects
- DASF 8 Encrypt data at rest
- DASF 9 Encrypt data in transit
- DASF 31 Secure model serving endpoints
- DASF 33 Manage credentials securely
- DASF 34 Run models in multiple layers of isolation
- DASF 36 Set up monitoring alerts
- DASF 38 Platform security — penetration testing, red teaming, bug bounty and vulnerability management
- DASF 39 Platform security — Incident Response Team
- DASF 40 Platform security — internal access
- DASF 41 Platform security — secure SDLC
- DASF 43 Use access control lists
- DASF 46 Store and retrieve embeddings securely
- DASF 48 Use hardened Runtime for Machine Learning
- DASF 50 Platform compliance
- DASF 51 Share data and AI assets securely
- DASF 52 Source code control
- DASF 53 Third-party library control
- DASF 54 Implement AI guardrails
- DASF 55 Monitor audit logs
- DASF 56 Restrict outbound connections from models
- DASF 58 Protect data with filters and masking
- DASF 60 Rate limit number of inference queries
- DASF 61 Train users on AI risk taxonomy and AI/ML security
- DASF 62 Implement network segmentation
- DASF 63 Update software
- DASF 64 Limit access from AI models and agents

Serverless egress control

Serverless egress control (SEG) is a security feature specific to serverless compute on Databricks that allows an administrator to implement networking policies that constrain access to:

- Endpoints on the internet such as Google Drive, Box, OpenAI, etc.
- Unauthorized storage resources (S3, ADLS, GCS) on the cloud platform
- Unauthorized on-premises database resources

SEG offers data exfiltration controls natively within Databricks, and admins can enforce a "deny by default" policy, restricting outbound connections only to approved locations such as specific cloud storage destinations. For added flexibility, admins can customize policies per workspace. Additionally, "log-only mode" logs policy violations without blocking connections, enabling you to test security settings confidently. It can be used in conjunction with technologies such as Private Link. For example, an administrator can block access to endpoints on the internet with SEG and then enable private access to specific backend endpoints over Private Link.

Controls addressed by serverless egress control:

- DASF 3 Restrict access using IP access lists
- DASF 31 Secure model serving endpoints
- DASF 43 Use access control lists
- DASF 54 Implement AI guardrails
- DASF 56 Restrict outbound connections from models
- DASF 62 Implement network segmentation
- DASF 64 Limit access from AI models and agents

Databricks Mosaic AI

Databricks provides a scalable, collaborative platform that empowers ML teams to prepare and process data, streamline cross-team collaboration and standardize the full ML lifecycle from experimentation to production, including generative AI and large language models (LLMs). You can build models from scratch and tune existing models on your data. However, it's not just about building and serving models. Databricks Mosaic AI covers the end-to-end AI workflow to help you deploy and manage models all the way through production. Some of our AI offerings include:

- End-to-end **retrieval augmented generation** (RAG) to build high-quality conversational agents on your data, leveraging **Mosaic AI Vector Search** for increased relevance and accuracy
- Integration with data-centric applications with leading AI APIs like OpenAI

- Training of predictive ML models either from scratch on an organization's tabular data or by **fine-tuning** existing models such as MPT and Meta Llama 3.1 to further enhance AI applications with a deep understanding of a target domain
- Efficient and secure serverless inference on your enterprise data and connection to **Unity Catalog** governance and quality monitoring functionality
- End-to-end MLOps based on the popular **MLflow** open source project, with all produced data automatically actionable, tracked and monitorable in the lakehouse
- Improved visibility and proactive detection of anomalies in your entire data and AI workflow, reducing risks, time to value, and high operational costs with **Databricks Lakehouse Monitoring**
- **Mosaic AI Agent Framework** gives developers the ability to build and deploy high-quality agentic applications with a set of tools on Databricks designed to help build, deploy and evaluate production-quality AI agents like retrieval augmented generation (RAG) applications.

We'll now outline specific products and components of Mosaic AI as they relate to the controls outlined in the Databricks AI Security Framework.

Controls addressed by Databricks Mosaic AI:

DASF 23	**Register, version, approve, promote, deploy and monitor models**
DASF 25	**Use retrieval augmented generation (RAG) with large language models (LLMs)**

Compound AI system

Compound AI systems involve integrating multiple interacting components such as models, retrievers, functions or external tools to perform complex tasks collaboratively. These systems are designed to handle complex tasks that single AI models can accomplish. Unlike standalone models, which focus on narrow tasks like language generation or image classification, **compound AI systems** use multiple models and other components that can dynamically collaborate to improve performance, decision-making and adaptability. This approach allows more flexibility and control over the system's behavior, making it a preferred architecture for more complex AI applications. For more information, refer to **The Shift from Models to Compound AI Systems**.

AI agents

While the industry is refining the definition of AI agents, generally, an AI agent is an application capable of making decisions based on data, learning from experience and adapting to new situations over time. Unlike traditional rule-based systems like robotic process automation (RPA), AI agents can manage structured, unstructured and other data types, analyze them and make informed decisions in dynamic or uncertain environments. These agents often use large language models (LLMs) to accomplish their objectives, helping automate tasks and streamline workflows.

AI agents can be classified into two main types:

1. **Interactive agents:** These agents respond directly to human input. A common example is a generative AI (GenAI) chatbot that interacts with users in real time.
2. **Autonomous agents:** These agents operate independently, automating tasks or workflows without human input. They trigger actions in response to events or processes and make decisions based on predefined logic. For instance, an autonomous agent could prioritize incidents in a system like ServiceNow, leveraging past knowledge of incident management.

While LLMs and retrieval augmented generation (RAG) are effective at understanding and generating text, they often face limitations regarding real-world task execution. Real-world scenarios demand linguistic comprehension, dynamic decision-making, task execution and adaptability. AI agents address these gaps by dynamically constructing and executing tasks, interacting with internal and external systems and adapting to changing conditions.

For more information, see **From LLMs to AI agents**.

MLflow

ML lifecycle management in Databricks is provided by managed **MLflow**. Databricks provides a fully managed and hosted version of MLflow integrated with enterprise security features, high availability and other Databricks workspace features such as experiment and run management and notebook revision capture. MLflow is an open source platform for managing the end-to-end machine learning lifecycle. MLflow supports **Java**, **Python**, **R** and **REST** APIs. It has the following primary components:

- **Tracking:** Allows you to track experiments to record and compare parameters and results
- **Models:** Allow you to manage and deploy models from a variety of ML libraries to a variety of model serving and inference platforms
- **Projects:** Allow you to package ML code in a reusable, reproducible form to share with other data scientists or transfer to production
- **Model Registry:** Allows you to centralize a model store for managing full lifecycle stage transitions for models, from staging to production, with capabilities for versioning and annotating. Databricks provides a managed version of the Model Registry in Unity Catalog.
- **Model Serving:** Allows you to host MLflow models as REST endpoints. Databricks provides a unified interface to deploy, govern and query your served AI models.
- **MLflow Tracing for agents:** Using MLflow Tracing you can log, analyze and compare traces across different versions of generative AI applications. It allows you to debug your generative AI Python code and keep track of inputs and responses. Doing so can help you discover conditions or parameters that contribute to poor performance of your application. MLflow Tracing is tightly integrated with Databricks tools and infrastructure, allowing you to store and display all your traces in Databricks Notebooks or the MLflow experiment UI as you run your code.

Controls addressed by MLflow:

- DASF 6 Classify data
- DASF 13 Use near real-time data
- DASF 15 Explore datasets and identify problems
- DASF 17 Track and reproduce the training data used for ML model training
- DASF 18 Govern model assets
- DASF 19 Manage end-to-end machine learning lifecycle
- DASF 20 Track ML training runs
- DASF 21 Monitor data and AI system from a single pane of glass
- DASF 23 Register, version, approve, promote, deploy and monitor models
- DASF 24 Control access to models and model assets
- DASF 26 Fine-tune large language models (LLMs)
- DASF 27 Pretrain a large language model (LLM)
- DASF 28 Create model aliases, tags and annotations
- DASF 29 Build MLOps workflows
- DASF 30 Encrypt models
- DASF 31 Secure model serving endpoints
- DASF 32 Govern and monitor access of AI model and model serving endpoints
- DASF 35 Track model performance
- DASF 42 Employ data-centric MLOps and LLMOps
- DASF 44 Triggering actions in response to a specific event
- DASF 45 Evaluate models
- DASF 47 Compare LLM outputs on set prompts
- DASF 49 Automate LLM evaluation
- DASF 51 Share data and AI assets securely
- DASF 54 Implement AI guardrails
- DASF 55 Monitor audit logs
- DASF 65 Implement end-to-end AI traceability
- DASF 66 Use Human-in-the-loop feedback
- DASF 71 Log and register AI agents
- DASF 72 Securely store and reuse agent state
- DASF 73 Register prompts

Mosaic AI Model Training

With **Mosaic AI Model Training** (formerly Foundation Model Training), you can use your own data to customize a foundation model to optimize its performance for your specific application. By conducting full parameter fine-tuning or continuing training of a foundation model, you can train your own model using significantly less data, time and compute resources than training a model from scratch.

Controls addressed by Mosaic AI Model Training:

- DASF 22 — **Build models with all representative, accurate and relevant data sources**
- DASF 26 — **Fine-tune large language models (LLMs)**
- DASF 27 — **Pretrain a large language model (LLM)**

Mosaic AI Vector Search

Mosaic AI Vector Search is a vector database that's built into Databricks and integrated with its governance and productivity tools. Mosaic AI Vector Search enables developers to improve the accuracy of their retrieval augmented generation (RAG) and generative AI applications through similarity search over unstructured documents such as PDFs, Microsoft Office documents and wikis. Mosaic AI Vector Search offers the following security features:

- Every customer request to Mosaic AI Vector Search is logically isolated, authenticated and authorized
- Encryption of all data at rest (AES-256) and in transit (TLS 1.2+)
- Integration with Unity Catalog to allow for vector indexes to be stored as entities within Unity Catalog and leveraged under the same unified interface to define policies on data, with fine-grained control on embeddings
- Support for two modes of authentication:
- Personal access token (PAT): You can use a personal access token to authenticate with Mosaic AI Vector Search. See **personal access authentication token**. If you use the SDK in a notebook environment, it automatically generates a PAT for authentication.
- Service principal token: An admin can generate a service principal token and pass it to the SDK or API. See **manage service principals**. For production use cases, Databricks recommends using a service principal token.
- **Customer-managed keys (CMK)** are supported on endpoints created on or after May 8, 2024

Controls addressed by Mosaic AI Vector Search:

- DASF 25 — **Use retrieval augmented generation (RAG) with large language models (LLMs)**
- DASF 37 — **Set up inference tables for monitoring and debugging models**
- DASF 46 — **Store and retrieve embeddings securely**

Mosaic AI Gateway

With **Mosaic AI Gateway** you can streamline the usage and management of generative AI (GenAI) models with your organization. It's a centralized service that brings governance, monitoring and production readiness to model serving endpoints. It also allows you to run, secure and govern AI traffic. Many enterprises mix and match multiple AI models from different providers to build compound AI systems (e.g., RAG, multi-agent architectures) that achieve the quality needed to deploy GenAI applications into production. However, as enterprises integrate a diverse array of open and proprietary models, they encounter challenges with operational inefficiencies, cost overruns and potential security risks. With Mosaic AI Gateway you can configure the following controls on your AI system:

- Permission and rate limiting to control who has access and how much access
- Payload logging to monitor and audit data being sent to model APIs using inference tables
- Usage tracking to monitor operational usage on endpoints and associated costs using system tables
- Traffic routing to minimize production outages during and after deployment

AI guardrails is another control within Mosaic AI Gateway. With AI guardrails you can configure and enforce data compliance at the model serving endpoint level to reduce harmful content on any requests sent to the underlying model. AI guardrails has the following controls:

- Safety filtering prevents your model from interacting with unsafe and harmful content like violent crime, self-harm and hate speech
- Personally identifiable information (PII) detection to detect any sensitive information (such as names, addresses, credit card numbers) for users
- Topic moderation to list a set of allowed topics. Given a chat request, this guardrail flags the request if its topic isn't in the allowed topics.
- Keyword filtering to specify different sets of invalid keywords for both the input and the output. One potential use case for keyword filtering is to prevent the model from talking about competitors.

Controls addressed by Mosaic AI Gateway:

- DASF 11 Capture and view data lineage
- DASF 14 Audit actions performed on datasets
- DASF 32 Govern and monitor access of AI model and model serving endpoints
- DASF 33 Manage credentials securely
- DASF 43 Use access control lists
- DASF 45 Evaluate models
- DASF 51 Share data and AI assets securely
- DASF 54 Implement AI guardrails
- DASF 55 Monitor audit logs
- DASF 60 Rate limit number of inference queries
- DASF 73 Register prompts

Databricks Lakehouse Monitoring

Databricks Lakehouse Monitoring lets you monitor the statistical properties and quality of the data in all of the tables in your account. You can also use it to track the performance of machine learning models and model serving endpoints by monitoring inference tables that contain model inputs and predictions.
To draw useful insights from your data, you must have confidence in the quality of your data. Monitoring your data provides quantitative measures that help you track and confirm the quality and consistency of your data over time. When you detect changes in your table's data distribution or corresponding model's performance, the tables created by Databricks Lakehouse Monitoring can capture and alert you to the change and can help you identify the cause.

Controls addressed by Databricks Lakehouse Monitoring:

- DASF 35 Track model performance
- DASF 55 Monitor audit logs

Databricks Clean Rooms

Databricks Clean Rooms uses Delta Sharing and serverless compute to provide a secure and privacy-protecting environment where multiple parties can work together on sensitive enterprise data without direct access to each other's data. Databricks Clean Rooms enables customers to execute diverse workloads using their preferred languages like SQL, Python and soon, Scala and Java. It supports multicloud collaboration across platforms such as AWS, Azure and GCP, with upcoming support for federated queries with external data platforms like Snowflake and BigQuery. Currently supporting two-party collaboration,

Databricks plans to scale up to 10 collaborators post-GA, offering APIs and orchestration workflows for flexibility.

Controls addressed by Databricks Clean Rooms:

DASF 59 **Use clean rooms**

Databricks Git folders

Databricks Git folders (formerly known as "Repos") is a visual Git client in Databricks. It supports common Git operations such as cloning a repository, committing and pushing, pulling, branch management and visual comparison of diffs when committing. Within Databricks Git folders you can develop code in notebooks or other files and follow data science and engineering code development best practices using Git for version control, collaboration and CI/CD.

Controls addressed by Databricks Git folders:

DASF 52 **Source code control**

Model Context Protocol (MCP) on Databricks

Model Context Protocol (MCP) is an open standard for connecting agentic AI systems to tools, resources, prompts and other context through a consistent interface. MCP lets you define tools once and reuse them across different agent frameworks and clients. On Databricks, MCP is one of the main tool approaches in the **Mosaic AI Agent Framework**, alongside Unity Catalog function tools and agent code tools, so you can choose the right balance of governance, flexibility and integration for each use case.

Databricks provides several ways to use MCP:

- **Managed MCP servers for Databricks-native capabilities:** Databricks offers ready-to-use MCP servers that expose governed capabilities such as Vector Search, Unity Catalog functions, Genie spaces and DBSQL as MCP resources. These servers follow consistent URL patterns and always enforce Unity Catalog permissions, so agents can only access data and tools they are authorized to use. See **Model Context Protocol (MCP) on Databricks** and **Use Databricks managed MCP servers** for details and examples.
- **External MCP servers with consistent governance:** You can connect agents to third-party MCP servers (including those available through Databricks Marketplace) or to any MCP server reachable through Unity Catalog connections. Databricks manages connectivity and authentication through secure proxy patterns and token management so that external tools can be used alongside Databricks-native tools under the same governance model.

- **Custom MCP servers as Databricks Apps:** You can host your own MCP server as a Databricks App to bring custom tools, workflows and business logic into your agent ecosystem. These apps use OAuth-based authentication and are governed by Unity Catalog and workspace controls, making it straightforward to control who can discover and invoke your custom tools.

MCP on Databricks is designed to support enterprise-scale agent workflows:

- **Standardized tool discovery and reuse:** MCP-based tools can be created once and reused across multiple agents and clients, which reduces duplication and simplifies integration when you mix Databricks-native agents with third-party tools and frameworks.
- **Flexible authentication patterns for agents:** Databricks supports both automatic authentication passthrough (where the platform provisions short-lived credentials and applies least-privilege access to declared resources) and on-behalf-of-user flows (Public Preview) where agents act using the end user's permissions with down-scoped tokens and per-user auditing. These patterns allow you to choose between service-style access and user-context access depending on the risk and governance requirements of your application. You can learn more in **Authentication for AI agents**.
- **Governance and security by design:** Unity Catalog governs access to MCP resources across managed, external and custom servers. All options respect Unity Catalog permissions and use secure connection patterns for external tools, so data access, auditing and policy enforcement remain consistent even as you integrate diverse tools and agent frameworks.
- **Prototyping to production on one platform:** Teams can prototype MCP tool-calling agents using common frameworks (for example, LangGraph or OpenAI SDK) and then deploy those agents with Mosaic AI Model Serving. When you deploy, Databricks automatically tracks the MCP resources the agent depends on so that authentication, permissions and logging are handled consistently at runtime.
- **Pricing aligned with underlying workloads:** Managed MCP servers are billed according to the Databricks services they use: serverless compute for Unity Catalog functions, serverless SQL for Genie, SQL warehouses for DBSQL and Vector Search pricing for vector indices. Custom MCP servers are billed according to Databricks Apps pricing, so costs align with the compute and storage your agents actually consume.

Controls addressed by Model Context Protocol (MCP) on Databricks:

DASF 68 **Use Securely Hosted Managed MCP Servers**

DASF 69 **Securely host Custom MCP Servers**

DASF 70 **Securely connect to External MCP Servers**

08 Appendix: The Databricks AI Governance Framework (DAGF)

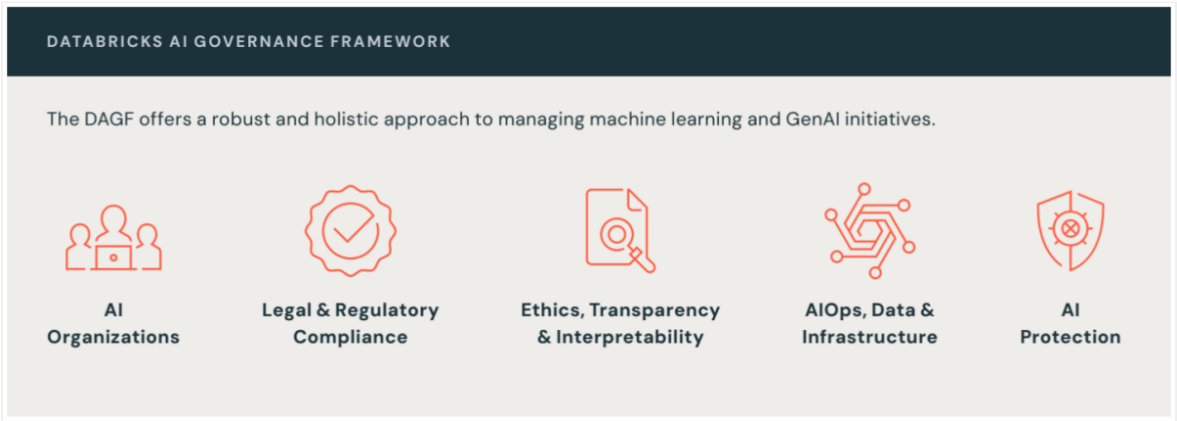
While this document is an extension to the Databricks AI Security Framework, in July we also released the Databricks AI Governance Framework. We recommend that readers responsible for higher level direction and governance of AI within their organization **read the full document**. Find a short summary below.

As Artificial Intelligence (AI) and Generative AI (GenAI) fundamentally reshape industries, the mandate for organizations is no longer just to adopt AI, but to do so responsibly. The **Databricks AI Governance Framework (DAGF)** offers a comprehensive, strategic guide for enterprises to navigate this complex landscape. It provides the structured approach needed to align AI initiatives with business objectives, adhere to rigorous ethical standards, and comply with an evolving global regulatory environment.

This framework is not merely a set of rules; it is a blueprint for innovation. By integrating governance into every stage of the AI lifecycle—from data collection to model deployment—organizations can confidently harness the full potential of AI while safeguarding against risks and fostering a culture of transparency and accountability.

The Foundation: Five Pillars of AI Governance

The DAGF is built upon five foundational pillars that collectively ensure a resilient and responsible AI program. These pillars bridge the gap between strategic vision and operational execution.



Pillar I: AI Organization

Embedding Governance into the Corporate DNA

Success begins with structure. This pillar focuses on embedding AI governance within the broader organizational strategy to ensure that every AI initiative delivers measurable business value.

- **Business Alignment:** AI must support the organization's mission. The framework emphasizes prioritizing investments based on business impact and ensuring that AI outcomes are measured against enterprise KPIs, not just technical metrics.
- **Governance Models:** Whether an organization chooses a **centralized**, **distributed**, or **hybrid** governance structure, the DAGF guides leaders in defining clear decision-making authorities and oversight responsibilities.
- **Risk Management:** A proactive approach to risk is essential. The framework outlines a lifecycle for managing AI risks: **Identify** (recognizing challenges like bias or security vulnerabilities), **Assess** (prioritizing risks based on impact and likelihood), **Mitigate** (implementing technical and policy measures), and **Monitor** (continuous evaluation of system performance).
- **Culture & Values:** It establishes the need for guiding values that influence ethical approaches and the importance of cross-functional collaboration between technical, legal, and business teams.

Pillar II: Legal and Regulatory Compliance

Navigating the Complex Legal Landscape

With the rapid emergence of regulations like the **EU AI Act** and existing privacy laws like **GDPR** and **CCPA**, compliance is paramount. This pillar helps organizations transform legal obligations into a competitive advantage.

- **Proactive Assessment:** Organizations are guided to assess their legal obligations early, including data residency laws, industry-specific regulations (e.g., HIPAA), and intellectual property rights regarding AI-generated outputs.
- **Liability & Risk:** The framework details strategies for managing liability, such as implementing legal risk scoring systems and establishing safeguards for high-risk AI use cases.
- **Contractual Frameworks:** It emphasizes the need for robust contracts with third-party vendors, ensuring clear definitions of data ownership, liability distribution, and adherence to ethical standards.
- **Future-Proofing:** By maintaining a "Legal Change Management Strategy," organizations can stay adaptable to new laws and emerging regulatory trends, ensuring long-term compliance.

Pillar III: Ethics, Transparency, and Interpretability

Building Trust Through Responsible Design

Trust is the currency of the AI era. This pillar provides the methodologies to ensure AI systems are fair, accountable, and understandable.

- **AI Ethics:** The framework champions core ethical principles, including **Accountability** (clear roles for oversight), **Fairness** (mitigating bias in models), **Human-Centricity** (prioritizing human well-being and safety), and **Inclusivity** (designing for diverse populations).
- **Transparency & Explainability:** "Black box" models pose significant risks. The DAGF promotes **Interpretability** (inherently understandable logic) and **Explainability** (providing rationale for decisions) to ensure stakeholders can verify and trust AI outcomes.
- **Traceability:** It underscores the importance of documenting data sources, model iterations, and decision logic to create a clear audit trail and support reproducibility.
- **Operationalizing Ethics:** Transparency is not just a design principle but an operational one. The framework suggests mechanisms for real-time explanations during model serving, allowing users to query and challenge AI outputs effectively.

Pillar IV: Data, AIOps, and Infrastructure

The Technical Engine of Innovation

Reliable AI requires a robust foundation. This pillar addresses the technical requisites for deploying scalable and secure AI systems.

- **Data Integrity:** High-quality data is the lifeblood of AI. The framework details standards for data classification, handling, and lineage tracking to ensure models are trained on accurate, unbiased, and secure datasets.
- **AIOps (AI Operations):** To move from pilot to production, organizations need a structured **ML Lifecycle**. This covers everything from raw data collection and processing to model training, evaluation, and continuous monitoring for drift or degradation.
- **Infrastructure Resilience:** Success depends on scalable infrastructure. The DAGF highlights the need for computing resources, storage, and cloud services that support the full ML lifecycle, ensuring performance, reliability, and cost-efficiency.
- **Tools for Maturity:** It points to essential capabilities like **Model Registries** for version control and **Feature Stores** for consistency, utilizing platforms like Databricks Unity Catalog for unified governance.

Pillar V: AI Security

Safeguarding the Future

As a companion to the **Databricks AI Security Framework (DASF)**, this pillar addresses the unique security challenges posed by AI.

- **End-to-End Protection:** Security must be integrated into every layer, from securing **Raw Data** and **Datasets** against poisoning or theft, to protecting **ML Algorithms** and **Models** from inversion attacks and malicious alterations.

- **Secure Serving:** The framework emphasizes securing model serving endpoints to prevent malicious data injection and ensuring that inference responses are not manipulated.
- **Operational Security:** It advocates for standardized **MLOps** security practices, including vulnerability assessments, rigorous access controls, and continuous threat monitoring to maintain a secure AI environment.

DAGF Conclusion

The **Databricks AI Governance Framework** is more than a document; it is a strategic asset. By adopting this holistic approach, organizations can effectively navigate the complexities of AI, balancing the drive for innovation with the imperative for control. It empowers enterprises to build AI systems that are not only powerful and efficient but also ethical, compliant, and secure—ultimately driving long-term strategic success in an AI-defined world.

09

Acknowledgements

This whitepaper would not be possible without the insight and guidance provided by our reviewers and contributors at Databricks and externally. Additionally, we extend our appreciation to the frameworks that inspired our research (MITRE, OWASP, NIST,CSA, etc.), as they have played a pivotal role in shaping the foundation of the Databricks Agentic AI Security.

DATABRICKS
Arun Pamulapati, Principal Security Engineer
David Veuve, Head of Security, Field Engineering
Omar Khawaja, Vice President, Field CISO
Nishith Sinha, Sr. Manager, Security Software Engineering
Caelin Kaplan, AI Security Engineer
Abhi Arikapudi, Sr. Director, Security Engineering

COMPLYLEFT
Ben Johns, Cybersecurity Specialist

ATLASSIAN
Niels Heijmans, Senior Security Architect

EXPERIAN
Karol Piekarski, Lead DevSecOps Engineer

10 License