Agent Metadata Specification

Version 0.2.1

Published by Tavro AI, LLC

January 13, 2026

Table of Contents

License and Usage

The information provided in this document does not, and is not intended to, constitute legal advice. All information is for general informational purposes only. This document may contain links to third-party websites provided solely for convenience. Tavro AI, LLC (“Tavro”) does not recommend or endorse the contents of these third-party sites.

Acknowledgements

Leader Authors:

- Sunil Soares (Tavro AI)
- Sanjeev Varma (Tavro AI)

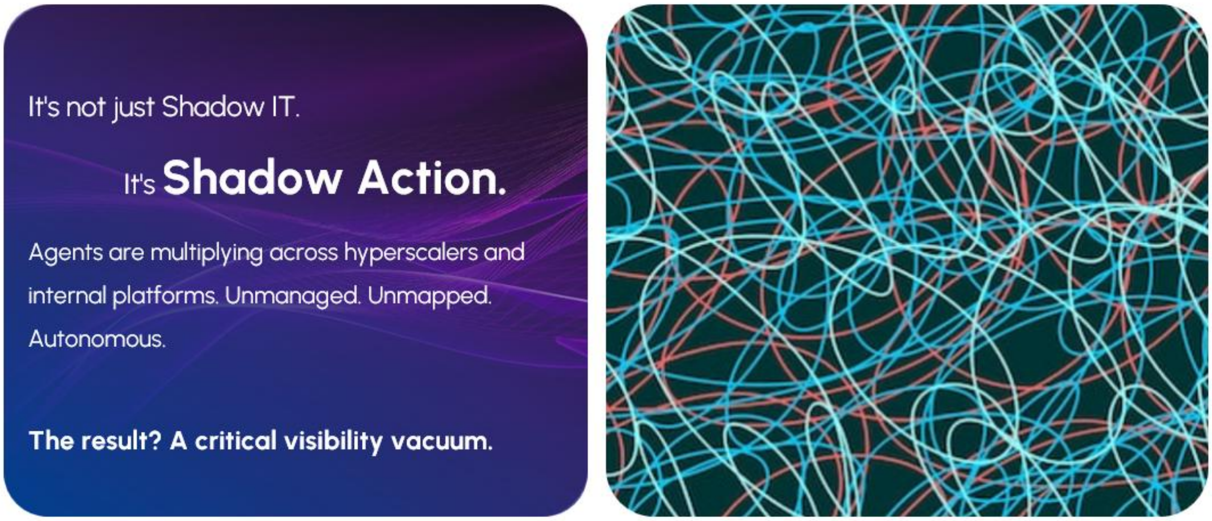
Key Contributors and Reviewers

The following individuals are acknowledged for their significant contributions to the review and development of this document, **listed alphabetically by last name**:

Key Contributor/Reviewer Name	Affiliation
Prasanna Kumar Akiri	Tavro AI
Val Calvo	BankUnited
Stan Christiaens	Collibra
Tony DiPerna	BankUnited
Mihir Dudhatra	Tavro AI
Gokula Mishra	OmniProAI
Rahul Pandit	Tavro AI
Khushboo Shah	Tavro AI

Introduction

AI agents are multiplying across the enterprise. Similar to the challenges with *shadow IT*, AI agents are creating so called *shadow action*. AI agents are often unmanaged, unmapped and autonomous. The result is a critical visibility vacuum (see Figure 1).



It's not just Shadow IT.

It's **Shadow Action.**

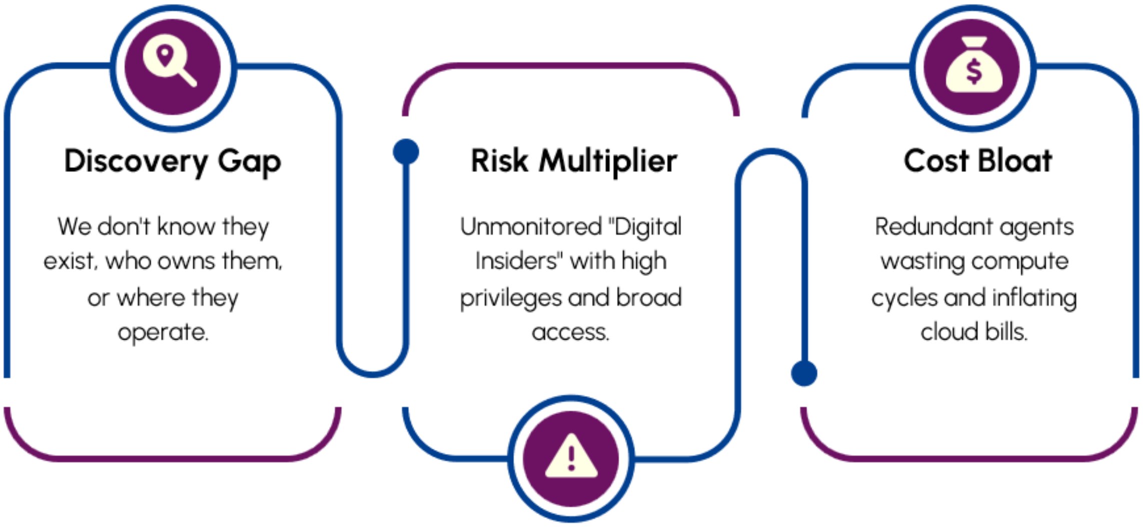
Agents are multiplying across hyperscalers and internal platforms. Unmanaged. Unmapped. Autonomous.

The result? A critical visibility vacuum.

Figure 1: AI agents create a visibility vacuum

Because agents are easy to build and deploy, they create multiple challenges (see Figure 2):

- *Discovery Gap* – Organizations do not know that the agents exist, who owns them, or where they operate.
- *Risk Multiplier* – Agent may operate as unmonitored "digital insiders" with high privileges and broad access.
- *Cost Bloat* – Redundant agents waste compute cycles and inflate cloud bills.



Discovery Gap

We don't know they exist, who owns them, or where they operate.

Risk Multiplier

Unmonitored "Digital Insiders" with high privileges and broad access.

Cost Bloat

Redundant agents wasting compute cycles and inflating cloud bills.

Figure 2: AI agents create a discovery gap, risk multiplier, and cost bloat

This document serves as a comprehensive template for capturing the essential metadata and core configuration parameters of a specific AI agent. The strategic importance of this template lies in its role as a standardized framework for organizations leveraging enterprise-grade agentic solutions.

Importance of Agent Metadata Standards

By implementing this robust metadata specification, enterprises gain several critical, strategic advantages:

- **Unified Enterprise-Wide View and Single Source of Truth of Agents:** The specification provides a mechanism to develop a consolidated, holistic view of all deployed agents across the entire organization. This centralization establishes a single, authoritative source of information for every agent, eliminating data silos and inconsistencies that can plague decentralized management systems.
- **Enhanced Accountability and Transparency of Ownership:** The metadata structure rigorously captures ownership details, ensuring clear accountability for the agent's performance, maintenance, and policy adherence. This transparency is crucial for operational governance and risk mitigation.
- **Automated Risk Management Functions for Agents:** By leveraging the standardized metadata, enterprises can apply systematic risk analysis, monitoring, and control across all business processes, applications, and the underlying agents they consume.
- **Accelerated Audit Readiness for Governance and Compliance:** The standardized and comprehensive nature of the metadata significantly accelerates the process of achieving audit readiness. It provides a structured record necessary for satisfying stringent governance requirements and demonstrating compliance with internal policies and external regulations (e.g., GDPR, CCPA, industry-specific compliance standards).
- **Easier Third-Party Risk Assessments for AI-Enabled Applications:** By standardizing the agent metadata, applications with embedded agents should find it easier to complete third-party risk assessments.

Agent Proliferation

Traditional metadata platforms capture information primarily from analytical systems. However, the metadata challenges increase exponentially for agents, which also leverage operational systems (see Figure 3).

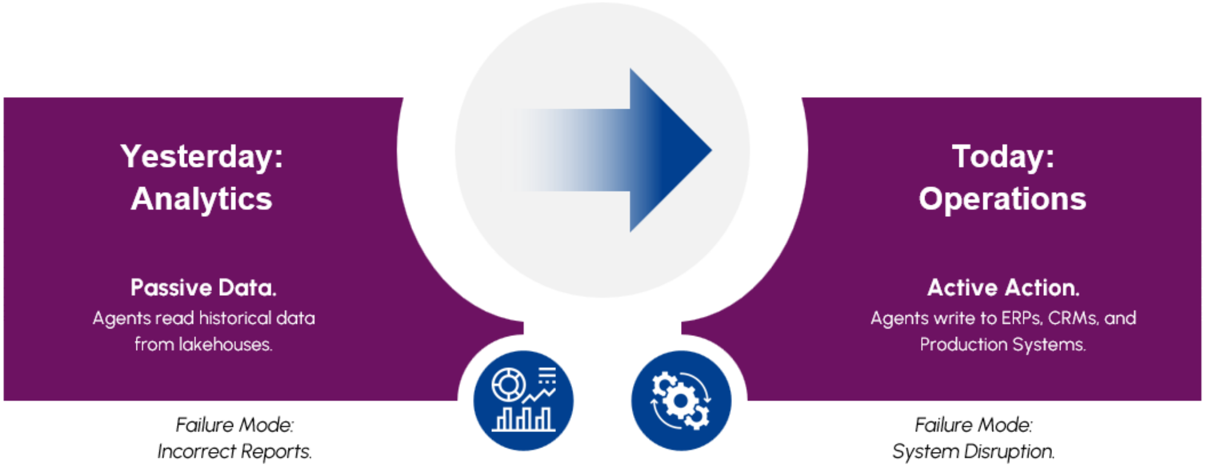


Figure 3: Agent metadata increases exponentially from analytical to operational systems

A number of platforms produce or consumer agent metadata (see Figure 4).

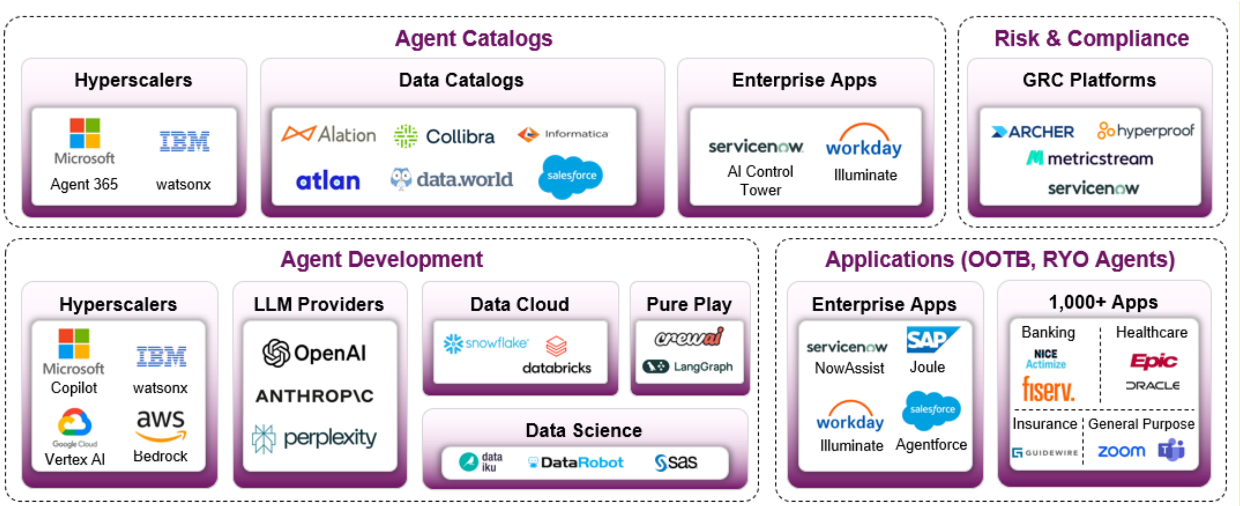


Figure 4: Producers and consumers of agent metadata

Agent Metadata Producers including the following:

- Hyperscalers (e.g., Microsoft Copilot, Google Vertex AI, IBM watsonx, Amazon Bedrock)
- LLM Providers (e.g., OpenAI, Anthropic, Perplexity)

- Data Cloud Providers (e.g., Snowflake, Databricks)
- Pure Play Agent Platforms (e.g., crewAI, LangGraph)
- Data Science Vendors (e.g., Dataiku, DataRobot, SAS)
- Enterprise Applications with out-of-the-box (OOTB) and roll-your-own (RYO) agents (e.g., ServiceNow NowAssist, SAP Joule, Workday Illuminate, Salesforce Agentforce)
- Industry-Specific Applications with OOTB and RYO agents (e.g., Fiserv and NICE Actimize in Banking, Epic and Oracle in healthcare, Guidewire in Insurance)
- The typical enterprise uses more than 1,000 applications[1] and an increasing percentage of these platforms will have OOTB and RYO agents

Agent Metadata Consumers including the following:

- Hyperscalers (e.g., Microsoft Agent 365, IBM watsonx)
- Data Catalogs (e.g., Alation, Atlan, Collibra, data.world/ServiceNow, Informatica/Salesforce)
- Enterprise Applications (e.g., ServiceNow AI Control Tower, Workday Illuminate)
- Governance, Risk, and Compliance (GRC) Platforms (e.g., Archer, Hyperproof, MetricStream, ServiceNow IRM)

Agent Data Model

[Outstanding Comment – Show diagram with Agent Conceptual Data Model]

The template is logically partitioned into the following key sections, each addressing a distinct facet of the Agent's identity, operation, and lifecycle:

- **Agent Identification:** This foundational section is dedicated to capturing the metadata required to uniquely identify the Agent. This includes key identifiers, its current deployment status (e.g., development, staging, production, deprecated), versioning information, and the essential details regarding its ownership and organizational context.

- **Configuration:** This section details the technical architecture and underlying components of the Agent. It meticulously categorizes and documents metadata associated with the core technologies, such as the underlying Large Language Model (LLM) being utilized (e.g., model name, version, fine-tuning details), specific Memory models (e.g., type, retention policy, capacity), and other critical computational and operational parameters.

- **Knowledge:** This section provides a deep dive into the information resources the Agent relies upon. It specifies the data sources it has been trained on (e.g., dataset identifiers, date of last training), and critically, details the mechanisms and interfaces it uses to access its knowledge base, including Retrieval-Augmented Generation (RAG) system configurations, database connections, and document repositories.

- **Tools and Actions:** This defines the Agent's functional capabilities and its interaction boundary with the external world. It enumerates what the Agent is capable of doing (its designated

[1] Salesforce, February 1, 2023, https://www.salesforce.com/news/stories/connectivity-report-2023/.

actions and use cases) and precisely how it interacts with external systems, APIs, or business applications, including function call specifications and security protocols.

- **Lineage:** This is a vital section for enterprise governance, establishing the end-to-end context for the Agent's usage. It meticulously maps the Agent to the specific business processes and high-level enterprise applications that consume or are associated with it. This mapping provides a comprehensive view of the Agent's usage patterns, the data sets it consumes (input lineage), and the resulting data sets it produces (output lineage), which is essential for impact analysis.
- **Risks and Controls:** This critical section consolidates a comprehensive view of all identified risks associated with the agent.

Identification

Table 1 summarizes the core identifying and descriptive details of the agent.

Attribute	Description
Agent ID	A unique, permanent identifier for the agent (e.g., HR-POL-003)
Agent Version	The current version number of the agent's configuration and logic
Title	The human-readable name of the agent (e.g., Internal HR Policy Assistant)
Description	A concise summary of the agent's purpose, capabilities, and target user
Goal Orientation	The specific objective or success metric the agent is designed to achieve
Role	The defined character or communication style that governs its interaction
Owner	The team or department responsible for the agent's maintenance and cost
Environment	The deployment environment (e.g., DEV, UAT, PROD)
Tags	A list of keywords for search and categorization (e.g., HR, policy, internal)
Governance Status	The current governance lifecycle status (e.g., DRAFT, APPROVED, DECOMMISSIONED)
Reviewer	Name of the person who approved the latest governance status

Table 1: Agent identification details

Configuration

Table 2 summarizes the configuration details for the agent.

Attribute	Description

LLM Model	The specific foundational model used by the agent (e.g., gemini-2.5-flash)
Prompt Template Reference	The ID of the template used for guiding LLM behavior (e.g., ABC-RAG-Standard-V2)
Access Scope	The agent's overall data access level (e.g., LOW_PRIVILEGE)
Memory Storage Reference	The external system used for long-term data/vector storage (e.g., Atlas-HR-RAG-VectorDB)
Memory Type	The type of memory storage used (e.g., VECTOR_DB, KEY_VALUE_STORE)
Data Freshness Policy	The maximum acceptable age of the data (caching policy) for the source
Autonomy Level	The degree to which the agent can act independently without human approval (FULL, SEMI-AUTONOMOUS, REACTIVE)
Reasoning Model	The underlying logic or planning paradigm (ReAct, ReWOO, Deductive, Inductive, Goal-based)
Multimodal Capability	The types of input the agent can process

Table 2: Agent configuration details

Knowledge Sources

Table 3 summarizes the knowledge sources for the agent.

Attribute	Description
Identifier	A Unique ID for the knowledge source
Name	A list of all specific data sources (e.g., databases, documents) the agent can access
Access Mechanism	The protocol or service used to retrieve knowledge (e.g., REST API, SQL connector)

Table 3: Agent knowledge details

Tools and Actions

Table 4 summarizes the tools configured for the agent including its external capabilities and delegation options.

Attribute	Description

Identifier	A unique reference ID for the tool (e.g., PolicySearchTool)
Name	Name of the tool
Description	Detailed explanation of the purpose and functionality
Delegation Possible	Boolean indicating if the agent can pass the request to another agent
Allowed Delegates	A list of Agent IDs, which the agent is allowed to delegate to
Input or Output	Indicates whether it is an input or output parameter
Parameter Name	Name of the parameter
Parameter Type	Required type and format of the parameter
Default Value	Default value, if any

Table 4: Agent tool and action details

Business Process

Table 5 summarizes business processes that consume the agent.

Attribute	Description
Identifier	The ID of the business process that uses the agent
Name	The human-readable name of the business process
Description	A brief description of the business process, its significance and relevance
Consumption Type	PRIMARY (core to process) or SECONDARY (auxiliary role)

Table 5: Business process details

Application

Table 6 summarizes applications that consume the agent.

Attribute	Description
Identifier	The ID of the business application that uses the agent. Ideally this should be a reference to an application configuration management database (CMDB)
Name	The human-readable name of the application

Description	A brief description of the application including its significance and relevance
Consumption Type	PRIMARY (core to application) or SECONDARY (auxiliary role)

Table 6: Application details

Data Lineage

Agentic lineage needs to map the entire decision path from prompt to logic to action to impact (see Figure 5).

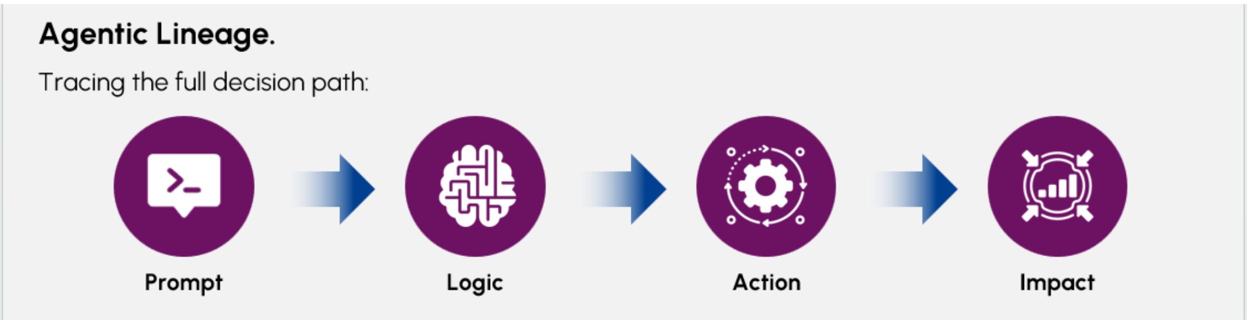


Figure 5: Agentic lineage

Table 7 captures the data sets associated with the agent.

Attribute	Description
Relationship ID	A unique Relationship ID
Parent Relationship ID	Parent ID, if any
Source Object ID	Unique ID as in the Source system
Source Object Domain	Domain
Source Object Name	Object name
Source Object Type	Type of Source Object (e.g., Agent, MCP Server, Table, Column, View, File, Folder)
Target Object ID	Unique ID as in the Target system
Target Object Domain	Domain
Target Object	Object Name

Name	
Target Object Type	Type of Target Object (e.g., Agent, MCP Server, Table, Column, View, File, Folder)
Access Level	READ, WRITE or DELETE

Table 7: Agent data lineage

Risks

Table 8 consolidates all risks associated with the agents across all risk vectors (e.g., data, security, Responsible AI, etc.)

Attribute	Description
Identifier	A unique Risk ID
Name	Name of the Risk
Description	Description of the Risk
Type	Type of Risk (e.g., Compliance, Reputational, Cyber, Responsible AI, Third-Party)
Impact	The expected impact of the risk (LOW, MEDIUM, HIGH)
Likelihood	The expected likelihood of the risk materializing (LOW, MEDIUM, HIGH)

Table 8: Agent risks

Controls

Table 9 consolidates all risks associated with the agents across all risk vectors (e.g., data, security, Responsible AI)

Attribute	Description
Identifier	A unique ID for the control (e.g., C-DAT-005)
Name	The human-readable name of the control
Description	A detailed description of the control's function
Risk Identifier	A list of risk vectors, which the control is designed to mitigate

Table 9: Agent controls

[Comment – Map Guardrail Metadata including for Self-Evaluating Agents and Judge Agents]

[Comment – Is OpenAI working on an overall Agent Metadata Spec
https://platform.openai.com/docs/guides/agents]