Unified Enterprise AI Orchestration

A Meta-Orchestrator Architecture Built on NVIDIA NeMo

Enterprise Architecture Specification

Abstract

This whitepaper describes an enterprise "meta-orchestrator" built on the NVIDIA NeMo Agent Toolkit that unifies native AI agents across major SaaS platforms, exposes a standard connector/MCP-style interface for new sources, and uses AI to semi-automate the onboarding of additional systems.

1 Executive Summary

Large enterprises now run multiple AI-augmented platforms—ServiceNow, Salesforce Agentforce, Atlassian Intelligence, Microsoft 365 Copilot, Oracle Expense Assistant, SailPoint AI, ProcessUnity AI—each with its own agents, data stores, and workflows [5–11]. Without a unifying layer, this produces "agentic sprawl": users must know which bot to talk to, admins duplicate policies, and cross-system workflows remain brittle.

The proposed architecture introduces:

- A NeMo Agent Toolkit-based meta-orchestrator that acts as a single conversational and API front-door for all enterprise AI capabilities.
- A connector contract that wraps each platform's APIs and native AI agents as NeMo tools, plus an optional Model Context Protocol (MCP) style interface for external consumers.
- An AI-assisted connector generator that can ingest API descriptions or documentation and propose new connector configs, gracefully handling tenant-specific schemas and minimizing manual engineering effort.

2 Architecture Overview

The design defines a four-layer architecture: HelpBot UI, Orchestrator, Domain/Platform Integrations, and Foundation Data/LLMs. NeMo Agent Toolkit becomes the implementation backbone of Layers 2 and 3, while NeMo Microservices/NIM host the models used by the orchestrator workflows.

2.1 Layer 1: HelpBot UI

A web "single pane of glass" with a conversational core, My To-Do, My Approvals, IT updates, Vendor tasks, and My Day calendar, all driven by a single chat entry point.

2.2 Layer 2: NeMo Orchestrator (Meta-Agent)

A NeMo workflow (e.g., `react_agent`) that receives user intents, calls domain agents and platform tools, and enforces approvals and governance.

2.3 Layer 3: Domain and Platform Agents

Domain agents (IT, HR, Finance, Vendor Risk) are NeMo workflows with limited tool sets, wrapping each SaaS platform's APIs and, where appropriate, native AI agents such as Now Assist or Agentforce. *Note: To mitigate latency cascades during agent-to-agent delegation, deterministic REST API calls are prioritized over recursive LLM reasoning where feasible.*

2.4 Layer 4: Data and Models

A Vector DB with curated enterprise embeddings, plus LLMs and embedders hosted via NeMo Microservices/NIM for orchestration, RAG, and summarization. NeMo Agent Toolkit expresses this as one or more YAML workflow configs with functions (tools), LLMs, embedders, and workflow sections, run via `nat serve` to expose an HTTP API consumed by the HelpBot UI.

3 NeMo Agent Toolkit Role

NeMo Agent Toolkit provides a framework-agnostic orchestration layer that can sit on top of existing LangChain, LangGraph, AutoGen agents, or custom Python tools, adding observability, profiling, and evaluation across multi-agent workflows.

Workflows are defined declaratively in YAML—listing functions (tools), LLMs, embedders, and a top-level workflow type—and can be run as long-lived services via `nat serve` for low-latency inference. Key capabilities relevant to this design include:

- **Functions as tools:** Any HTTP endpoint, Python function, or existing agent can be wrapped as a NeMo function and exposed to the orchestrator, with descriptions and JSON schemas controlling tool use.
- **A2A (Agent-to-Agent) delegation:** Agent Toolkit supports connecting agents together so a meta-agent can call domain workflows as if they were tools, enabling the "meta-orchestrator plus domain agents" pattern.
- **Observability and profiling:** Built-in integration with OpenTelemetry, Phoenix, Langfuse, and the NeMo Agent Toolkit Profiler lets you trace how each platform tool and agent contributes to latency and quality.

4 Platform AI Agents: Capabilities and Lifecycle

This section summarizes how to manage the built-in AI agents in each platform and what data you can extract from them or from the underlying systems.

4.1 ServiceNow: Now Assist and ITSM Agents

Capabilities: Now Assist adds generative skills across ITSM, CSM, HR, and other apps. "Now Assist for ITSM" provides an AI agent collection that can interpret incidents, recommend resolutions, and summarize work.
Management and operation: Admins use the Now Assist Admin console to enable plugins, configure knowledge bases, and define data boundary rules.
Data extraction: Operational data is accessible via REST Table APIs (e.g., `/api/now/table/incident`). Now Assist-generated summaries are persisted on records and can be indexed into the vector store.
Orchestrator integration: For complex ITSM workflows, expose a tool `servicenow_now_assist` that calls a Virtual Agent entrypoint. For explicit operations (e.g., approvals), define NeMo HTTP tools that hit REST APIs directly.

4.2 Salesforce: Agentforce

Capabilities: Agentforce operates on trusted CRM data to perform tasks like triage, case resolution, and sales planning, guided by the Atlas Reasoning Engine.
Management and operation: Architects use Agentforce Studio to define agent instructions, topics, accessible data, and allowed actions before publishing.
Data extraction: Data surfaces through REST and SOQL APIs. Agentforce outputs can be logged internally, exported, and mapped into the orchestrator's RAG indexes.
Orchestrator integration: Create NeMo tools that call specific Agentforce API endpoints. For cross-system flows, NeMo orchestrates direct REST calls and queries Agentforce only for CRM-specific subplans.

4.3 Atlassian: Atlassian Intelligence

Capabilities: Embedded virtual teammate for generation, rewriting, and summarization in Jira and Confluence, alongside low-code JSM virtual agents in Slack/Teams.
Management and operation: Configured via organization/product level toggles and a low-code flow editor for JSM knowledge sources.
Data extraction: Jira issues and Confluence pages are accessible via REST APIs. AI-generated content lives in standard fields, which can be indexed directly.
Orchestrator integration: Treat Atlassian AI primarily as UX augmentation. Use NeMo to integrate via REST APIs for cross-system reasoning and summarization.

4.4 Microsoft 365: Copilot and Agents

Capabilities: AI assistance across the 365 suite, supporting declarative agents and custom engine agents via the M365 Agents Toolkit.
Management and operation: Managed via Copilot Studio and the Extensibility portal, mapping plugins and Graph connectors to security groups.
Data extraction: Core data is accessible through Microsoft Graph REST endpoints.
Orchestrator integration: Expose the meta-orchestrator as a Copilot plugin so users can access enterprise-wide workflows from within M365. Rely on Graph APIs as NeMo tools.

4.5 Oracle: Expense Assistant

Capabilities: A conversational skill within Fusion Applications Digital Assistant (FA Digital Assistant) for creating, modifying, and checking expenses.
Management and operation: Configured via the Digital Assistant Platform. Authentication utilizes SSO, sometimes requiring step-up authentication (e.g., OTP).
Data extraction: Expense data is exposed through Oracle Fusion Expenses REST APIs.
Orchestrator integration: Route expense intents through the NeMo orchestrator directly to Expenses REST APIs to mirror supported flows, unifying the UX.

4.6 SailPoint: AI-Driven Identity Security

Capabilities: Analytics for Access Insights and Recommendations, plus Harbor Pilot—an AI agent for exploring identity data and building workflows.
Management and operation: Enabled per tenant via the SailPoint admin console.
Data extraction: Accessible via REST and SCIM APIs. AI risk scores and recommendations are stored as metadata on identities and entitlements.
Orchestrator integration: Treat Harbor Pilot as an in-product agent. The meta-orchestrator focuses on workflow APIs for cross-system provisioning flows.

4.7 ProcessUnity: AI for Third-Party Risk

Capabilities: Automates vendor risk assessments using Evidence Evaluator and Predictive Analytics based on Global Risk Exchange content.
Management and operation: Configured inside the TPRM UI using feature flags and weight tuning.
Data extraction: Vendors, assessments, and AI-evaluated evidence scores are exposed via API endpoints.
Orchestrator integration: Define tools like `processunity_get_vendor_risk` to return normalized risk summaries, correlating vendor risk with tickets and access rights in NeMo.

5 Unified Connector and MCP-Style Abstraction

To avoid bespoke code for each platform, define a connector contract that both NeMo and external MCP clients can utilize. Each connector implements:

- **Metadata:** Name, version, domain, supported intents, and capability level.
- **Tools:** Logical names, natural-language descriptions, JSON input/output schemas, and endpoint templates.
- **Runtime Interface:** An `invoke` handler that executes HTTP calls and normalizes results, and an optional `delegate_agent` function for native AI delegation.

Your orchestrator can host an MCP server that maps MCP tools to NeMo tools, mapping MCP resources to underlying data sources, and exposing a single endpoint to client LLMs safely.

6 Data Extraction and RAG Across Platforms

The orchestrator should maintain a unified enterprise knowledge layer combining:

- Transactional data (tickets, cases, expenses, identities) via connectors.
- Content data (articles, policies, contracts) from Confluence, SharePoint, and ServiceNow KB.
- Agent-generated content (Now Assist summaries, Agentforce plans) tagged with source IDs.

NeMo's embedders section points at NIM-hosted embedding models, defining RAG tools as functions that run vector search and return relevant passages to orchestrator workflows.

7 AI-Assisted Onboarding of New Sources

To accelerate the addition of new platforms, an AI-powered connector generator workflow can be implemented in NeMo. Recognizing the complexity of real-world enterprise architectures, this process must accommodate tenant-specific customizations:

[1] **API Analysis & Discovery:** A NeMo workflow ingests the OpenAPI spec and clusters endpoints. Crucially, it dynamically queries platform metadata APIs (e.g., Salesforce SOQL describe) to supplement static specs with tenant-specific custom fields and composite object requirements.

[2] **Schema Generation:** The agent writes candidate JSON schemas for inputs and outputs aligned with NeMo's functions format and MCP tool schemas.

[3] **Config Synthesis:** It drafts a YAML fragment including base URLs, HTTP methods, and auth structures.

[4] **Human-in-the-Loop Review:** A platform owner reviews the generated config. *This step is critical for resolving composite API requirements and non-standard authentication flows that AI cannot reliably infer from documentation alone.* Synthetic tests (`nat eval`) are run to validate the connector.

[5] **Promotion:** Once approved, the connector enters the Tool Registry for meta-orchestrator consumption.

8 Operational Model and Governance

To run this architecture at enterprise scale, specific governance paradigms are required:

- **Identity and Access:** Use SSO (OIDC/SAML) for the HelpBot and store only per-platform OAuth tokens or delegated credentials. *Handling asynchronous step-up authentication (e.g., Oracle's one-time PIN) requires a robust session-state abstraction layer to pause and resume meta-workflows without breaking the conversational UI.*
- **Approvals and Human-in-the-Loop:** Implement a policy engine that tags high-risk actions (e.g., sensitive access grants) and forces a Confirm step in the UI before NeMo executes the tool.
- **Observability:** Enable NeMo Agent Toolkit's OpenTelemetry integration and Profiler to trace LLM calls, tool invocations, and external agent delegations centrally.
- **Versioning and Testing:** Treat connectors as versioned artifacts. Utilize NeMo's `nat eval` to run regression suites before promoting changes to production environments.

9 References

[1] https://github.com/NVIDIA/NeMo-Agent-Toolkit
[2] https://docs.nvidia.com/nemo/agent-toolkit/latest/
[3] Enterprise-AI-Assistant_-Design-Spec.pdf
[4] Enterprise-AI-Assistant-Design_Intercontinental_confidential.pdf
[5] https://cirra.ai/articles/salesforce-agentforce-ai-agents
[6] https://www.servicenow.com/docs/r/xanadu/intelligent-experienc platform-now-assist-landing.html
[7] https://www.valiantys.com/en/resources/atlassian-intelligence/
[8] https://learn.microsoft.com/nl-be/microsoft-365-copilot/extensibility/
[9] https://docs.oracle.com/en/cloud/saas/financials/26a/fawde/overview-of-expense-assistant.html
[10] https://www.sailpoint.com/products/ai-driven-identity-security
[11] https://www.processunity.com/third-party-risk-management/processunity-ai/
[12] https://docs.nvidia.com/nemo/agent-toolkit/1.3/workflows/about/index.html
[13] https://docs.nvidia.com/nemo/microservices/latest/about/core-concepts/inference.html
[14] https://docs.nvidia.com/nemo/microservices/latest/about/index.html
[15] https://docs.nvidia.com/nemo/microservices/index.html
[16] https://github.com/NVIDIA/NeMo-Skills/blob/main/docs/basics/inference.md
[17] https://docs.nvidia.com/nemo/agent-toolkit/1.2/workflows/run-workflows.html
[18] https://docs.nvidia.com/nemo/agent-toolkit/latest/index.html

[19] https://developer.nvidia.com/nemo-agent-toolkit
[20] https://www.youtube.com/watch?v=yrqdvBLAI3k
[21] https://www.servicenow.com/community/s/cgfwn76974/attachments/cgfwn76974/now-assist-blog/28/1/ServicenowAssistCalculation.pdf
[22] https://www.servicenow.com/docs/bundle/yokohama-it-service-management-page/product/now-assist-itsm/concept/now-assist-itsm.html
[23] https://www.youtube.com/watch?v=EowWsoAf4wM
[24] https://www.youtube.com/watch?v=Nytvguldb6E
[25] https://www.salesforce.com/agentforce/
[26] https://developer.salesforce.com/docs/ai/agentforce/overview
[27] https://www.salesforceben.com/how-does-salesforces-agentforce-work/
[28] https://developer.salesforce.com/docs/ai/agentforce/guide/get-started.html
[29] https://www.easesolutions.com/lp-atlassian-ai
[30] https://www.eesel.ai/blog/atlassian-intelligence-ai-jira-confluence
[31] https://almarise.com/en/atlassian-intelligence-how-ai-is-transforming-team-efficiency-and-project-management/
[32] https://support.atlassian.com/organization-administration/docs/atlassian-intelligence-features-in-confluence/
[33] https://www.youtube.com/watch?v=uzzazDawRfA
[34] https://www.youtube.com/watch?v=sgYlj_AS8D4
[35] https://www.youtube.com/watch?v=RNGpzB5ql5k
[36] https://www.youtube.com/watch?v=TMj5NGFWVdM
[37] https://docs.oracle.com/en/cloud/saas/financials/25a/faiex/overview-of-setting-up-expense-assistant.html
[38] https://www.ateam-oracle.com/howtos-oracle-fusion-applications-agent
[39] https://cedricleruth.com/configuring-oracle-digital-assistant-
[40] https://documentation.sailpoint.com/saas/help/ai/index.html
[41] https://documentation.sailpoint.com/saas/help/ai/iiq/index.html
[42] https://www.bls360.com/insights/sailpoint-ai-driven-identity-
[43] https://cybersectools.com/tools/processunity-third-party-risk-
[44] https://www.processunity.com/resources/blogs/deploying-an-ai-powered-third-party-risk-management-program/
[45] https://www.youtube.com/watch?v=BOxkDs4b7DE
[46] https://www.processunity.com
[47] https://info.processunity.com
[48] https://www.linkedin.com/posts/processunity-unite2025-tprm-ai-activity-7325984639649472513-DnXn
[49] https://nvidia.github.io/NeMo-Skills/basics/inference/