WSO2CON NORTH AMERICA

May 20 - 22, 2026 | Austin, Texas, USA

WSO2 Agent Manager

Malith Jayasinghe

VP of AI

Nadheesh Jihan

Senior Technical Lead

Are your agents behaving?

gavel

Governance

Control exactly what an agent is allowed to call (LLMs & Tools).

analytics

Evaluation

Score performance and behavior over time to ensure quality.

fingerprint

Identity

Control who the agent is and enforce granular permissions.

Policy Enforcement: Ensuring agentic reliability through structured governance.

Agent Governance & Guardrails

Governing outbound LLM requests and tool calls via centralized gateways.

Policy Enforcement: Ensuring agentic reliability through unified Ingress & Egress control.

Why the AI Gateway?

Every agent calls LLMs and tools. Both leave the agent process. The gateway is the one place where you can:

visibility

See every call

Unified visibility of LLM + tool traffic across the entire stack.

gavel

Apply policy

Enforce security and compliance rules before any call lands.

speedometer

Meter centrally

Control cost, tokens, and usage at an organization-wide level.

"Per-agent code can't enforce org-wide policy. The gateway can."

Defining Governance

1. Budget Control

Track requests, tokens, and costs per agent or per organization.

2. Request Guardrails

PII masking, JSON validation, and prompt-attack detection in real-time.

3. Permitted Tools

Access control for what an agent is allowed to call at all.

4. Audit & Traceability

Every single call recorded end-to-end for compliance and debugging.

Comprehensive governance ensures secure and accountable agent operations.

Org-level vs Per-agent

ORG-LEVEL

Cross-cutting Defenses

Applied at the provider level. Every agent inherits these (e.g., PII rules, core URL filters).

PER-AGENT

Instant-level Tuning

Instance-specific rules. Tweak semantic prompts or specific tool filters for a single bot.

*One **PII rule** org-wide. One **Semantic Prompt Guard** for the agent.*

Demo:
Agent Governance in Action

Agent Evaluation

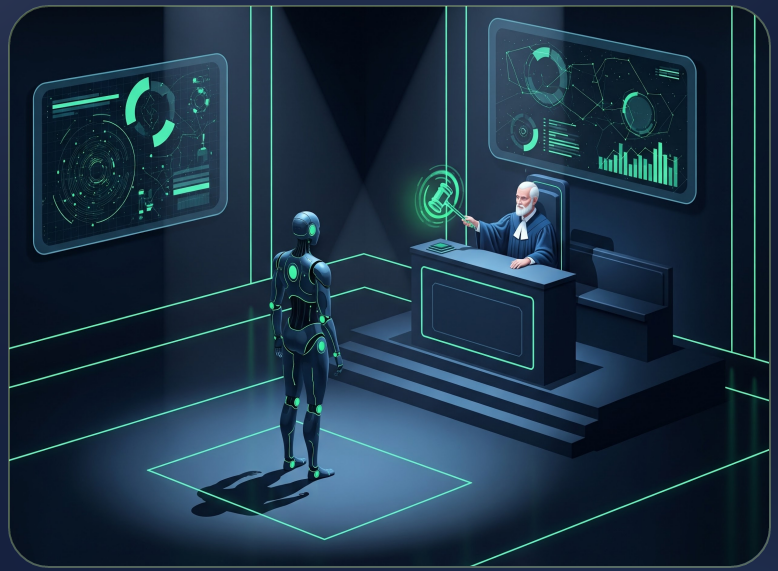
Checking agent behavior on dimensions that matter:
e.g. :Accuracy, Path, Recovery, and Tone.

Accuracy
Factual correctness and groundedness of answers.

Path (Trajectory)
Efficiency and optimality of tool execution sequences.

Recovery
Graceful handling of tool timeouts or malformed data.

Tone
Professionalism, empathy, and persona adherence.



Evaluation turns qualitative intuition into measurable signals for agent quality.

check_circuit_breaker_voice_over

Stages of Evaluation

Development

Golden test set

Notebook / CLI

code

Deployment

Regression set

CI pipeline + workflows

rocket_launch

Production

Real traffic traces

Monitors on live traces

insights

"Our focus is on the stages you cannot do manually."

Why Continuous Eval in Production?

Inputs drive behavior

Code doesn't crash, but behavior shifts with unseen prompts.

Silent Failures

Agents can go rogue without an exception (Successful HTTP 200, wrong path).

Scale

Debugging from logs is hopeless; failure is hidden inside deep traces.

Catching what dev-time tests cannot.

all_inclu

sive

"Continuous monitoring is the only way to ensure LLM reliability at scale."

Evaluation Concepts

Evaluator

Scores one aspect of a trace (e.g., Tone, Groundedness, Path Efficiency).

Types

Code Evaluators
Using code to evaluate certain patterns of the agent.

LLM Judges
Using LLMs to validate the answers by LLMs.

Evaluator Levels

Trace Level
Whole application output.

Agent Level
Decision process and tool calls.

LLM Level
Individual model invocation quality.

Tracing Hierarchy & Evaluation Levels

Trace Level

The root container for a full user session. Scopes the entire journey from initial prompt to final response.

Agent Level

Defines logic boundaries for a specific persona or principal (e.g., "Travel Agent"). Inherits identity and permissions.

Span Level

The atomic unit of work. Captures a single LLM request/response or a specific tool execution (ToolSpan).

```
Trace: User Request —> Final Response
  AgentSpan: "supervisor"
    LLMSpan: reasoning ("User wants a flight...")
    ToolSpan: search_flights (API Call)
    ToolSpan: delegate_to_agent ("travel-agent")
      AgentSpan: "travel-agent"
      LLMSpan: reasoning ("Booking cheapest...")
      ToolSpan: book_flight (id: AA100)
  AgentSpan: "itinerary-formatter"
    LLMSpan: reasoning ("Format booking...")
    ToolSpan: format_itinerary (CONF-123)
```

EVALUATOR DISTRIBUTION

1x	3x	5x
TRACE EVAL	AGENT EVAL	LLM / SPAN EVAL

Evaluation Concepts Contd...

Monitor

A bundle of evaluators applied across multiple traces. Doesn't require a dataset to define task or reference answers.

Experiments

A bundle of evaluators applied to grade a set of predefined tasks and to compare against expected reference behaviors.

history

Past Monitor
Runs on historical windows for RCA and regression checks.

update

Future Monitor
Runs continuously on incoming streams for drift and prod-watch.

"Both are based on traces collected."

What Agent Manager Monitors Give You

Drift Detection

Surface when behavior degrades over time.

trending_down

Triage by Trace

Let LLM judges sort 500+ traces so you don't have to.

rule

Accuracy Check

Score the next N traces after a prompt change.

fact_check

"Same primitive (window, evaluators) — three different jobs."

Demo:
Eval Monitors at Work

Agent Identity

badge

Digital Workers

Agents are no longer just software — they are **digital workers**.

They live in the same directory as employees, with their own identity.

security

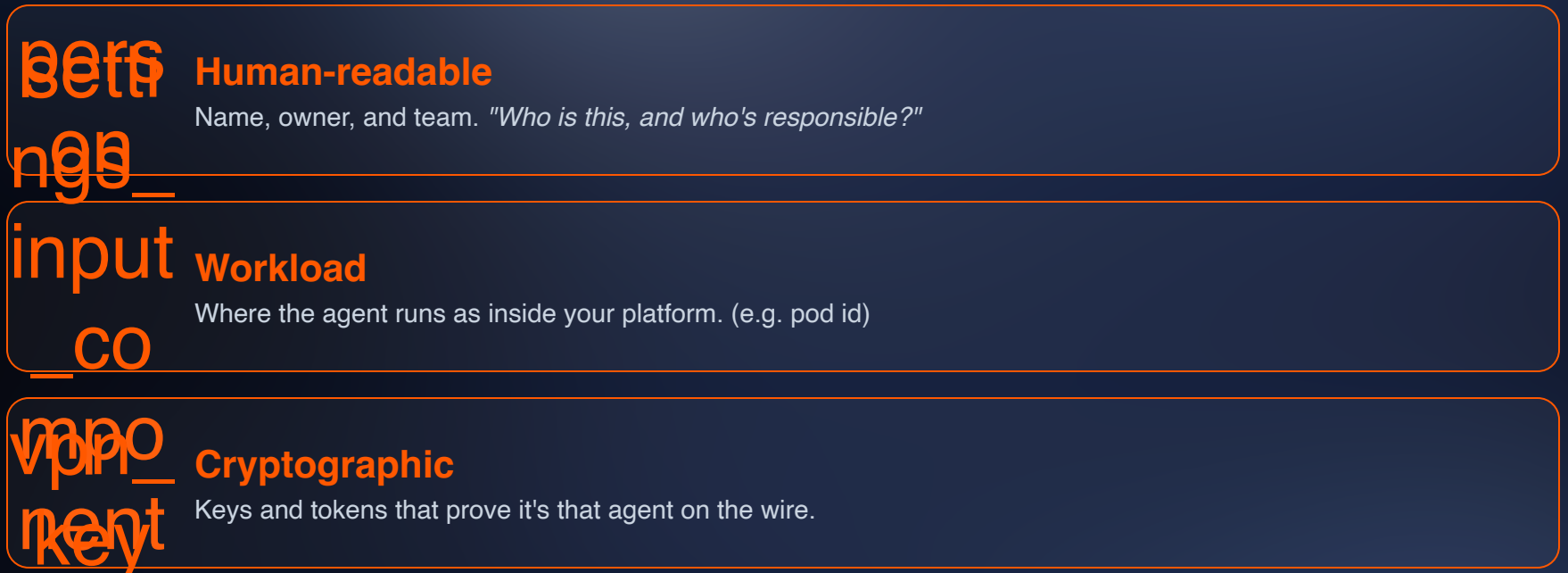
First-Class Principals

Treat agents as first-class principals with:

- Attachable permissions
- Attributable actions
- Revocable access

"Identity isn't documentation. It's the credential every outbound call carries."

What an agent identity is



One identity, three projections. All three live together — and stay consistent — in Agent Manager.

Agent Manager puts Identity to Use

badge

Issues & Injects

Assigns identity and injects it into the specific workload running the agent.

hub

Connects Resources

Determines which LLMs, MCPs, and tools the agent reaches under that identity.

history_egg

Audits Actions

Every single action is recorded and mapped against the agent's unique identity.

app_maybe

Enforces Control

AI Gateway identity gates every outbound call for total security.

"Identity isn't documentation. It's the credential every outbound call carries."

Our Plan

What we have today

Org/project scoping, per-instance keys, and gateway-enforced access — the foundation.

The Destination: Complete Agent Identity

Moving towards directory-integrated identity for each **agent instance** created from a **source repo** or a **kind**.

Learn More About Agent Identity

"Identity & Access Management" Track

May 21st | Yellow Room

Ensuring secure, attributed, and audited agent operations through cryptographic identity.

Innovating for the future

- The agent ecosystem is evolving rapidly
- As enterprises scale agentic systems, new operational, governance, and runtime challenges continue to emerge
- We are continuously evolving the agent manager based on:
  - Real-world enterprise requirements
  - Customer feedback and operational learnings
  - Changes in the broader AI and agent ecosystem
  - Emerging standards, frameworks, and architectural patterns

Innovating for the future

- Focus areas
  - Observability
  - Agent ID
  - Evaluation
  - Agent Catalog and Kinds
  - Governance and Guardrails
  - AI Gateway for Tool Governance
  - Skills and MCPs
  - Memory Management
- Each of these areas opens up a large set of possibilities.
- The challenge is not only innovation — it is carefully prioritizing the capabilities that solve the most important real-world enterprise problems while adapting to a rapidly evolving ecosystem

Road Map (Coming Mid-May)

- **Auth for Agents Exposed as APIs**
  - API key-based security for agent API endpoints
- **CLI (command-line interface) for agent lifecycle management**
- MCP Server
  - Agent Manager capabilities as an MCP server for AI assistants and coding tools
- Skills for Coding Agents
  - skills that enable coding agents (e.g., Claude Code, Cursor) to manage agents through Agent Manager (e.g. create, deploy, monitor, and troubleshoot agents as part of developer workflows)

Road Map (Q2-Q3)

- **Auth for Agents Exposed as APIs (continued)**
  - **Embedded Thunder OAuth**
    - M2M client credentials generated via embedded Thunder, enabling OAuth security without an external authorization server (requires OAuth app integration with Thunder)
  - **Third-Party Authorization Server Integration**
    - connect agents to existing enterprise authorization servers (e.g., Asgardeo, Okta, Auth0)
  - **User-Behalf Flow**
    - delegated authorization where agents act on behalf of authenticated end-users with proper consent and token exchange

Road Map (Q2-Q3)

- **Platform Capabilities**
  - Extending agent kinds
  - **Platform-Managed Agent Input Interfaces**
    - decouple agent logic from how it is triggered. The platform manages the input interface (API, event-driven, scheduled, chat, etc.) while agents are developed independently of their consumption mode. Enables the same agent to be exposed through different interfaces without code changes.

Road Map (Q2-Q3)

- **Governance**
  - **Tool Access Governance (API & MCP Proxies)**
    - proxy-based policy enforcement for how agents access external tools - API endpoints and MCP servers
  - **Custom Guardrails for LLMs** - user-defined guardrail policies for LLM interactions beyond the built-in evaluators
- **Developer Experience**
  - **Declarative Definition for Agents** - define agents as code (YAML/TOML) with GitOps-style version control and promotion
- **Observability**
  - **Alerting** - configurable alerts for agent health, performance, and operational events
- **Security & Reliability**
  - **Runtime Sandboxing** - hardened, isolated execution environments with container-level security controls