Outerlimit

Outerlimit and Microsoft Agent 365

Enhanced security and governance for Enterprise agentic AI estates.

Outerlimit
info@outerlimit.com

Contents

Introduction

The recent launch of Microsoft Agent 365 provides organizations with a strong foundation for agent identity, discovery, and compliance in Microsoft-native environments. Assigning Microsoft agents an identity by default is a significant step forward and Microsoft is well placed to lead here, since it is building upon existing Entra controls enterprises already trust and run at scale.

That said, agent identity is only part of the AI governance problem. While the ability to identify an agent is important, visibility over precisely what an agent is doing, and more importantly whether that action should be permitted, are critical to delivering end-to-end governance.

Agent 365 extends existing Microsoft capabilities rather than introducing a new control model. Conditional Access provides control at the access layer, Purview at the data layer, and Defender on detected behavior. Each capability is a mature, well-understood control, but none answer the question;

> *"Should this specific tool call be allowed to execute in the moment, with specific parameters, and in this execution chain."*

It is precisely this capability gap that Outerlimit solves by providing a deterministic enforcement point at the tool boundary, where every tool call is evaluated at runtime before the target system is affected regardless of which cloud, framework, or identity provider the calling agent sits behind.

As a result, Agent 365 and Outerlimit are complementary, not competing. Agent 365 surfaces what agents exist and how they are behaving, while Outerlimit answers whether a specific action may execute under specific conditions.

Agent 365 Overview

With Agent 365 and Agent ID, agents are now assigned first-class identities. Every Microsoft agent is given its own identity within Entra. Microsoft agents, including those built within Copilot Studio and Foundry, are now assigned identities by default, without any manual work from the creator. Agent identities live alongside users, applications, and service principals within Entra. Now that agents have their own identity, customers can leverage existing Microsoft controls against them, including conditional access, lifecycle governance, and risk detection. This provides a strong agent identity foundation and is beneficial for customers, given these are already well-established and familiar controls.

It is worth being precise about what this foundation covers. Identity and access governance answer who an agent is and what it's permitted to reach, but not what it does once it starts calling tools. Conditional access can block a sign-in, but it doesn't inspect the content of a tool call an already-authenticated agent makes moments later.

Outside of agent identity, Microsoft has also worked to improve visibility of agent assets. Microsoft now provide an Agent Registry which was recently consolidated into the Agent 365 admin center, which builds an inventory of agents alongside dashboards and usage telemetry. As you'd expect, this works well within the Microsoft ecosystem: Copilot Studio and Foundry agents arrive in the registry automatically, complete with their human owners attached.

Visibility is often the first challenge enterprises face as AI adoption accelerates, and a native registry is a genuinely useful starting point. That said, it is a registry of registered agents which depends on agents being built through, or explicitly onboarded to, the Microsoft stack. Agents built on other infrastructure, spun up by individual teams outside a formal build process, or reached only via third-party MCP servers, are not surfaced. In reality, that unregistered layer, commonly called "shadow AI", is exactly where the highest uncertainty lives and the majority of enterprises are very rarely building agents in a single platform.

None of this is a shortcoming so much as a natural boundary of scope. By design, Agent 365 and Agent ID are Microsoft's answer for agents built on Microsoft's platform. However, as frontier models and agent frameworks continue to advance at pace, today’s agentic footprint is scattered through a myriad of different platforms. Agents get built on AWS, GCP, and custom infrastructure, they call out to MCP servers that have never heard of Entra and they chain into one another across vendor boundaries in ways no single ecosystem's registry was designed to see.

Outerlimit closes this visibility and control gap, by discovering what's running regardless of where it was built, and governing what agents are authorized to do at the moment they call a tool.

The Enforcement Gap

Agent 365 gives agents an identity, so we know who they are. The question Agent 365 **does not answer** is what a specific agent is allowed to do under pre-defined conditions. The three Microsoft controls that exist today answer a different question:

1. **Conditional Access is an access-time gate:** Entra evaluates whether an agent identity may access a tool at authentication and token issuance. It does not sit on the tool-call path. An agent authorized to make payments is authorized, at the token level, to make any payment its scope permits.

2. **Purview is a classification layer:** It answers whether data is sensitive, and whether it's allowed to move, not what a specific tool call is about to do.

3. **Defender is reactive:** It is built to recognise anomalous patterns and block on what it detects, which by design means it identifies a behavior after it has been exhibited.

None of these controls evaluates an individual tool call at execution against policy, identity, parameters, and context, with a deterministic allow or deny before the target system is affected. That is the control Outerlimit provides.

Enforcement with Outerlimit

Complimentary to these Microsoft capabilities, **Outerlimit Enforce** provides:

- **Per-call pre-execution enforcement:** Every tool call is evaluated before it runs: calling identity, tool path, action parameters, context, and required approvals, resulting in deterministic controls whatever framework built the agent. Enforcement runs in your environment and fails closed if validation cannot complete, execution is denied.

- **Policy-as-code:** Policy intent is compiled into a structured, versioned, inspectable representation, governed through standard change control rather than portal toggles.

- **Zero credential exposure:** Agents never hold a real standing credential. Credentials are recovered inside the enforcement pipeline per invocation, only when identity, policy, and approvals validate. A compromised agent does not inherit execution

capability.

- **Multi-agent chain integrity:** Modern agent workflows are rarely stand-alone. Multi-agent chains are becoming increasingly commonplace, where tool calls arrive at the end of a delegation chain with one agent invoking another. Outerlimit evaluates the full provenance of that chain including which agent initiated the agent, which agent is executing the call, and whether each hop was itself authorized. A delegation handoff doesn't create a blind spot at the tool boundary; the policy decision accounts for the whole chain and therefore maintains integrity of the entire chain before actions are authorized.

- **Consent as a policy primitive:** High-stakes tool calls can require explicit consent, bound to the specific action parameters, before execution proceeds.

- **Low-friction, multi-cloud coverage:** The enforcement point deploys as a gateway, proxy, sidecar, or SDK — no modification to the agent or MCP server. Discovery ingests what the Agent 365 registry finds and augments it with AWS Bedrock, Anthropic Claude, and CrowdStrike Falcon telemetry.

None of this competes with Agent 365. It answers a question Agent 365, Conditional Access, Purview, and Defender aren't built to answer.

The Visibility Gap

By default, Agent 365's registry provides an inventory of agents on Microsoft platforms including Copilot Studio and Foundry. Agents built on other platforms need to be added deliberately. AWS Bedrock and Google Vertex AI agents sync as inventory entries in preview, but with no runtime governance attached, and agents built on other frameworks such as LangChain aren't in scope at all unless manually integrated one at a time.

The same pattern exists at the MCP layer. Global Secure Access and Defender for Cloud Apps can spot traffic to known SaaS MCP servers and generative AI applications, and Defender for Endpoint now discovers supported local agents and their MCP configurations on onboarded devices. That's genuine progress. However, both paths lean on a catalogue and a managed endpoint. This means that an internal MCP server that isn't a recognised SaaS app, a server reached from a contractor laptop, or a personal device, sits outside what either capability was designed to surface. This is a particular problem for customers using other EDR products, for example CrowdStrike over Microsoft Defender.

Visibility with Outerlimit

Outerlimit Discover provides the following extended capabilities:

- **Vendor-agnostic discovery**: Outerlimit's inventory isn't scoped to a single cloud, framework, or identity provider. It covers agents built on Microsoft, AWS, Google Cloud, or custom infrastructure, as well as frameworks such as LangChain.
- **Complements Agent 365**: For agents that are in the Microsoft estate, Outerlimit ingests the Agent Registry and Entra Agent ID as a source rather than duplicating it, so Microsoft-native agents are displayed correctly attributed alongside everything Microsoft's registry doesn't reach.
- **Multi-EDR correlation**: Shadow MCP discovery isn't tied to a single endpoint stack. Outerlimit integrates with EDR platforms including CrowdStrike Falcon, so shadow agent and MCP activity surfaces regardless of which EDR an organisation has deployed.
- **A governed path for MCP, not just a detection path**: Where Microsoft and EDR telemetry stop at finding shadow MCP usage, Outerlimit provides a route to bring discovered MCP servers behind a managed gateway. Once identified, a server can be onboarded into policy scope rather than left as a flagged risk with no remediation path. Discover turns into control, not just an alert.

Capability Summary

Capability	Agent 365 (E7)	Outerlimit
Agent discovery	Native registry (third-party sync in preview)	Ingests the Agent 365 registry; augments with AWS, Anthropic, CrowdStrike, etc.
Agent identity	Native (Entra Agent ID)	Integrates with Entra
Purview / DLP compliance	Native	Not provided
Threat detection	Yes (some in preview)	Not provided
Per-call tool enforcement	Not provided	Core capability
Policy-as-code governance	Limited	Core capability
Credential isolation from agents	Not provided	Per-invocation substitution
Consent per action	Not provided	Policy primitive
Failure mode at the tool boundary	Detection gap defaults to execution	Fail-closed: defaults to denial

Summary

Agent 365 and Outerlimit address fundamentally different problems. Agent 365 provides identity, compliance, posture, and fleet observability across the agents it can see. Outerlimit provides runtime enforcement and policy governance at the tool boundary. The Agent Registry is a discovery source for Outerlimit, not a competing inventory. What it finds flows into the Outerlimit governance inventory, augmented by discovery from the platforms, frameworks and endpoints Microsoft does not cover.

Customers already leveraging Microsoft E7 benefit from Agent 365 at no additional license cost. Outerlimit co-exists by becoming the enforcement layer on top, applied to the tool calls where impact is highest based on operational and security risk specific to each individual customer. This approach is agnostic by design, enforcement operates at the tool boundary, where an agent's intent becomes an action. Outerlimit is independent of everything upstream: the platform the agent runs on, the framework that built it, and the model that drives it.

Outerlimit provides the security and authorization layer for Agentic AI, offering a single control and observability plane across your agentic ecosystem.

As enterprises evolve their adoption of AI and give agents access to tools via APIs, databases and MCP servers, exponential risk is introduced into their environment. Shadow AI and the unpredictability of agents create new attack surfaces and vulnerabilities. To quantify and manage this risk, the ability to discover and control an expanding agentic footprint becomes critical.

Outerlimit closes this visibility and control gap, enabling enterprises to discover and observe their agentic activity, quantify risk, and enforce run-time controls at the tool execution layer.