ORACLE

Oracle Fusion Claw: Secure Runtime for Enterprise AI Execution

How Fusion Claw isolates, constrains, and governs AI-generated execution

October 2026, Version 1.0
Copyright © 2026, Oracle and/or its affiliates
Public

Executive summary

This is the companion paper to “Delegated Authority for Agents in Oracle AI Agent Studio” which focuses on the know-your-agent (KYA) framework for agents operating in Oracle Fusion Cloud Applications. KYA asks four questions:

1. Who is acting?
2. On whose behalf?
3. Within what limits?
4. With what accountability?

This paper addresses the next question: What happens when that authorized agent is given a runtime, tools, files, generated code, and the ability to execute work?

Oracle Fusion Claw allows an AI agent to do more than reason and invoke individual tools. It provides an execution environment in which the agent can generate execution logic, run commands, process files, use tools, and continue working toward an outcome.

A conventional agent typically reasons and invokes a bounded tool or API. Fusion Claw can take on a larger outcome, generate execution logic, process significant amounts of data and files, perform computation, and continue working over time. That changes the security boundary. The enterprise is no longer securing only an API invocation. It is securing an AI-directed execution environment.

That creates a different security problem. The model and everything it consumes must be treated as untrusted, including prompts, attachments, restored files, downloaded content, skills, generated commands, tool output, web content, and model responses.

The architecture starts with a simple assumption: The model may be wrong, manipulated, or malicious. A model decision is therefore never itself a security decision. The security boundary is the set of platform controls around the model—not the model, its prompt, its generated code, or the container alone.

KYA governs authority.

Fusion Claw governs execution.

Fusion governs the transaction.

The AI can use the computer. It does not control the security boundary around the computer.

We are not trying to make the model trusted. We are building a trusted execution environment around an untrusted model.

Isolate the execution

Fusion Claw gives the AI somewhere to perform substantial work without giving it control over the environment governing that work.

For the multi-tenant execution path, a dedicated runtime is associated with a logical conversation or outcome. The isolation unit is the conversation or outcome, not each individual task. Follow-up work may intentionally reuse the same runtime within that boundary.

Copyright © 2026, Oracle and/or its affiliates / Public

If the runtime has been recycled, the approved state can be rehydrated into a replacement so the work can continue. Hydration restores the approved state. It does not re-derive or broaden authority.

Reasoning, generated code, working files, tools, computation, and task execution are inside the execution environment. Identity, credentials, security policy, network policy, lifecycle controls, authorization, and authoritative enterprise systems are outside the agent's control.

Multi-tenant runtimes execute as non-root, use a read-only root filesystem, do not mount the host file system or container runtime socket, and use an ephemeral workspace.

A Kubernetes pod is a meaningful workload boundary. It is not an inner process sandbox between every process inside it. The common-container path is a shared service with workspace controls, not a dedicated tenant process boundary.

Pod deletion discards process memory and the ephemeral workspace. It does not delete durable task records, outputs, or approved archives maintained outside the runtime for continuity.

Security objective: Isolate the place where the AI works from the controls governing that work.

Keep authority outside the runtime

Giving an agent more computing capability does not give it more business authority.

Delegated user identity is preserved for user-owned Fusion work. Application identity is limited to approved platform operations and is not a substitute for user authority. The runtime does not accept work until identity is validated. Fusion roles, data security, delegated authority, business rules, approvals, and transaction controls remain authoritative when work returns to Fusion.

Moving execution into a container does not move authority into the container.

KYA establishes what authority has been delegated. Fusion Claw constrains execution within that authority. Fusion remains the final enforcement point for business operations.

Successful execution inside the runtime is therefore not the same as authorization to perform a business transaction. Follow-up work retains the applicable ownership, tenant, and authority context.

The same separation applies to credentials. Secret values do not cross the orchestrator boundary. Trusted platform components resolve credential references and hold the required material. The agent process does not inherit standing credentials, private keys, or client secrets.

The agent gets the ability to perform approved work, not possession of the credentials that make the platform trusted.

Greater execution capability does not confer greater authority. User authority remains the ceiling.

Copyright © 2026, Oracle and/or its affiliates / Public

Constrain the execution

Fusion Claw is an execution environment, but it is not an unrestricted shell.

- **What can execute.** The platform owns the executable and security-sensitive configuration. Task input cannot supply an arbitrary command or weaken security hooks, writable roots, credential sources, or proxy policy.
- **What it can access.** Inputs, working state, outputs, and temporary files are separated. Inputs are treated as untrusted. Only intentional outputs are returned. Paths are checked, and symbolic links cannot redirect access outside the assigned workspace.
- **Where it can communicate.** Destination policy is applied outside the agent process. Web search can be controlled independently. Convincing the model that it should contact a destination does not create network authority.

Consider a simple failure case. An untrusted file tells the agent to read a credential path and send the workspace to an outside host. The command is rejected, the path is rejected, the destination is not authorized, and Fusion never receives a business action.

Model proposes execution → Runtime evaluates it → Permitted execution runs → Fusion independently authorizes business actions

A skill is guidance, not a gate. Fusion Claw may validate a script, but the script does not authorize itself.

Command checks, script inspection, path controls, and pre-execution policy checks are defense in depth. They do not prove every possible behavior of arbitrary code. A saved script or binary can contain behavior not visible in its launch command. Likewise, destination controls constrain network behavior but are not a claim of complete egress isolation. Together, these controls are intended to limit the blast radius of incorrect or malicious model behavior by constraining the execution environment around it.

Assume the model can be manipulated

The architecture does not depend on making the large language model (LLM) immune to prompt injection. Instead, it starts with the stronger threat assumption: Assume the model is wrong, manipulated, or malicious.

The model may encounter malicious instructions in prompts, attachments, files, web content, skills, downloaded content, tool responses, or restored working state. Those instructions may convince the model that it should read a protected file, execute a prohibited command, expose a credential, or communicate with an unapproved destination. The security architecture still determines what it can execute, what it can access, where it can communicate, and what Fusion will ultimately allow it to do.

> **Prompt injection becomes less dangerous when convincing the model is not the same thing as controlling the runtime.**

This does not eliminate prompt-injection risk. It limits the consequences by making model reasoning independent from the controls governing execution.

Copyright © 2026, Oracle and/or its affiliates / Public

Defense in depth

The security model can be viewed as a set of independent boundaries around an untrusted model. No individual layer is sufficient.

Identity and delegated authority
↓
Authenticated runtime
↓
Credential isolation
↓
Command and code controls
↓
File system isolation
↓
Network and egress controls
↓
Fusion authorization and transaction controls
↓
Audit and evidence

Fusion Claw security does not depend on trusting the model. It depends on independent controls around the model.

Test and evidence the boundary

A governed execution environment needs evidence of both authority and execution.

- **Authority evidence** establishes who acted, on whose behalf, and under what delegated authority.
- **Runtime evidence** establishes what occurred while that authority was being exercised, including the task and conversation, follow-up relationships, security decisions and rejections, lifecycle events, and execution results.

Evidence records what happened. It does not create authority.

Logs and evidence must not contain bearer tokens, private keys, client secrets, or full prompt and response bodies.

The security boundary is tested adversarially, not only through expected-use testing. More than 5,000 adversarial tests exercise the defined controls across command execution, credential access, file system escape, path traversal, prompt injection, network bypass, authentication, authorization, and cross-task isolation. Concurrent tests specifically look for cross-task data exposure.

Testing provides evidence that the defined control set behaves as intended. It does not prove that no bypass can exist outside that control set.

What this paper does not claim

1. The agent process is **not** fully sandboxed.
2. A pod is **not** an inner process sandbox.
3. Destination controls are **not** complete egress isolation.
4. Command validation does **not** determine the behavior of every saved script or binary.

Copyright © 2026, Oracle and/or its affiliates / Public

ORACLE

Secure execution without trusting the model

Enterprise AI is moving from assistance toward execution. More complex and challenging enterprise work requires more than individual tool calls. An AI may need to reason over a problem, generate execution logic, process data and files, perform computation, and continue working toward an outcome. The more execution capability the AI receives, the more important it becomes that authority and security remain outside its control.

- KYA governs authority.
- Fusion Claw governs execution.
- Fusion governs the transaction.

AI reasons and writes the execution logic. The Fusion Claw runtime constrains where and how that logic can execute. Fusion remain authoritative for identity, data, authorization, and transactions.

We are not trying to make the model trusted. We are building a trusted execution environment around an untrusted model.

Connect with us

Call +**1.800.ORACLE1** or visit **oracle.com**. Outside North America, find your local office at: **oracle.com/contact**.

Copyright © 2026, Oracle and/or its affiliates. This document is provided for information purposes only, and the contents hereof are subject to change without notice. This document is not warranted to be error-free, nor subject to any other warranties or conditions, whether expressed orally or implied in law, including implied warranties and conditions of merchantability or fitness for a particular purpose. We specifically disclaim any liability with respect to this document, and no contractual obligations are formed either directly or indirectly by this document. This document may not be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without our prior written permission.

Oracle, Java, MySQL, and NetSuite are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.