ORACLE

Oracle Fusion Claw

The autonomous, governed execution runtime for Oracle Fusion Applications

October 2026, Version 1.1
Copyright © 2026, Oracle and/or its affiliates
Public

Executive summary

For three decades, enterprise software has been a system of record. It captured what happened, stored it reliably, and reported on it after the fact. Every meaningful action still ran through a person: someone read the record, decided what to do, and did it.

Oracle Fusion Claw changes where the work happens. It is a frontier model–backed runtime that lets Oracle Fusion Agentic Applications plan intelligently, execute deterministically at scale, and learn from every cycle. The model reasons about what needs to happen, and a separate execution layer carries the work out under enforced authority and controls. Work that used to wait on a human now runs continuously, with people involved by exception rather than by default.

This is the move from systems of record to systems of outcomes.

This technical brief explains how Fusion Claw works, the architecture that makes autonomous execution safe for enterprise use, the autonomy model that governs how much a customer delegates, and how Claw connects to the broader agent ecosystem. It is written for technical evaluators, enterprise architects, and decision makers assessing Oracle's agentic direction.

From systems of record to systems of outcomes

Enterprise applications were built to record transactions, enforce data integrity, and produce an auditable history. The software could tell you the state of the business, but it could not change that state on its own. Appropriate for its era, that design created a structural limit to what the software could do.

Generative AI arrived first as assistance, followed by embedded GenAI capabilities that drafted text inside existing screens. Then, answer agents began responding to questions and workflow agents began to string steps together. Each stage moved responsibility closer to the software, but a person still owned the outcome. Agentic applications are the next step in that progression, shifting from software that assists with work to software that can take responsibility for completing an outcome. The Fusion AI portfolio reflects that progression across four delivery models.

The four AI delivery models

Fusion Applications deliver AI capabilities in four ways, representing increasing levels of software autonomy:

- **Embedded GenAI capabilities:** generation inside the existing experience
- **Answer agents:** responses to questions grounded in enterprise context
- **Workflow agents:** multi-step task execution with human direction
- **Fusion Agentic Applications:** software that owns an outcome end to end

Fusion Agentic Applications replace the manual loop that legacy systems depended on and deliver business outcomes. Customers pay for the trusted execution of enterprise work, not individual AI components. Fusion Claw is the runtime that lets Fusion Agentic Applications address complex problems at enterprise scale.

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

ORACLE

Understanding Fusion Claw

Fusion Claw is an autonomous execution runtime. It is defined by three actions:

- **Plan intelligently:** A frontier model reasons over context, goals, and constraints to determine what should happen.
- **Execute deterministically at scale:** A separate execution layer consistently executes the plan every time it runs, so the same inputs produce the same result.
- **Learn every time:** Each cycle feeds back so the system improves without re-reasoning from scratch on every run.

Fusion Claw reasons once, then executes continually. The expensive, variable work of reasoning with the frontier model happens deliberately at the start. Then, the repeatable work of execution runs cheaply and predictably, so customers don't pay the reasoning cost on every action.

Adapt and learn at the outcome and workflow levels

Performance improvement happens on two distinct levels, the outcome and the workflow. Keeping them separate matters for how the system is governed.

- **Adapt within a single outcome:** As a single job runs toward an outcome, Fusion Claw responds to what it encounters. It adjusts to the data in front of it, handles conditions that arise mid-execution, and re-plans inside the boundaries it was given. This is learning scoped to one job, and it stays inside the job's authority and policy.
- **Learn across multiple outcomes:** Over many runs, the system carries forward what worked, so repeated classes of work get more efficient and more reliable without a person re-specifying them each time. This is improvement at the level of the workflow.

The distinction keeps autonomy within its defined boundaries. Adapting within an outcome never expands the authority granted for that job, and learning across outcomes changes how well the system performs, not what it is permitted to do. Both are constrained by the enforcement layer described in the next section.

Architecture with four-way separation of responsibilities

Autonomous execution is only acceptable in an enterprise if the system cannot grant itself permission to act. Fusion Claw is built on the following four-way separation of responsibilities that makes this work:

LAYER	ROLE
Reasoning	The model interprets context and recommends what to do. It has no power to act on its own conclusions.
Authority	A distinct layer decides what the system is permitted to do, for whom, and within what limits. The model cannot grant itself authority.
Execution	Permitted plans are carried out deterministically within Fusion Applications.
Enforcement	Controls, guardrails, and audit run independently and constrain the other three layers.

The most important takeaway is that the model can reason and recommend, but it cannot grant itself the authority to execute. Reasoning and permission live in different layers. That separation is what makes it defensible to let the software act without a person in the loop for every step.

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

Containment: where the work runs

Separating reasoning from authority governs what Fusion Claw is allowed to decide. A second boundary governs where the work physically runs. Fusion Claw works toward an outcome inside an isolated, temporary execution environment that is created for that outcome and torn down when it completes. The work is contained, which means the outcome operates only on the data and transactions it was permitted to touch, cannot reach beyond its assigned scope, and leaves no standing process behind once it finishes.

This matters because autonomy without containment would mean an agent could act beyond its assigned scope. A contained execution environment means Fusion Claw cannot take unrelated actions, accumulate access over time, or persist beyond the work it was authorized to do. Combined with the authority and enforcement layers, containment is what lets an outcome run at full autonomy while staying strictly bounded.

The autonomy model

Customers decide how much autonomy to delegate. Fusion Claw supports a graduated model so that customers can gain trust in the system before giving it more responsibility. This metaphorical autonomy 'dial' can be set at different places:

- **Human in the loop:** The system proposes an action; a person approves each action before it runs.
- **Human in the lead:** The system runs, and a person directs and intervenes rather than approving every step.
- **Autonomous execution:** The system runs within its guardrails, and people are involved only for exceptions.

A customer can start conservatively and move along this spectrum, workflow by workflow, as confidence grows. The enforcement layer applies at every level, so moving toward autonomy changes how often a human is asked to intervene, not whether controls apply.

Connecting to the agent ecosystem

Enterprises aren't standardizing on a single AI client. They will bring their own agents and new frontier assistants, and they expect those to work with Fusion Applications. With Fusion Claw, responsibility for governed execution stays in the Fusion layer, no matter which client initiated the request. This is delegation, not orchestration from the front end.

The discoverability of business objects and data does not mean governance. An interface can expose what objects and operations exist without exposing the approval chains, delegation rules, separation of duties, and transaction integrity that make an action safe. Fusion Claw is what supplies the second half of that governance.

Fusion Claw in practice

The architecture and autonomy model are easier to understand with real work examples. The two Fusion Claw use cases below sit at different points on the autonomy spectrum. The first shows a human delegating a bounded objective and Fusion Claw executing within that authority. The second shows a governed, fully autonomous outcome where the system plans at scale but only approved recommendations reach the downstream system.

Both examples end with an outcome receipt: a durable record of who requested the work, who owned and executed it, which policy governed it, what data and transactions were touched, and a verification signature. The receipt is the artifact that lets a person trust execution they did not personally perform.

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

Example 1: Workforce scheduling under delegated authority (human in the lead and reviewing)

A charge nurse and team leader returns to find that agents—working within established organizational policies and delegated authority—kept routine work moving in her absence: timecards meeting predefined validation criteria were prepared for approval, routine attendance exceptions were flagged for appropriate review, and attendance patterns were surfaced for her consideration rather than acted upon automatically.

Fusion Claw also flags a real risk. A telemetry unit is at 70% coverage against a 95% target. Rather than fixing it silently, the system presents the problem to her. She defines a scheduling outcome, permitting it to consider reallocating qualified staff, adjusting scheduling parameters where permitted by applicable requirements, or offering additional overtime opportunities. She also authorizes the system to implement scheduling adjustments within established organizational policies, applicable requirements, and her delegated authority.

Fusion Claw simulates three alternatives against predefined organizational constraints: availability, absences, qualifications, open shifts, overtime limits, cost, and scheduling rules across both the source and target units. Reallocating 36 qualified staff covers 70 of 72 open shifts and lifts coverage from 70% to 99% at a 5% relative cost increase, while the donor unit stays fully covered. The alternatives resolve fewer shifts or cost more; one would have required union approval. The system confirms qualifications, availability, and no schedule overlap. Where permitted by applicable requirements and established organizational policies, the system publishes authorized assignments, notifies affected workers, and creates an outcome receipt documenting the starting state, alternatives considered, applicable policy parameters, approvals, and resulting coverage. No policy exceptions were introduced.

Why it matters: The charge nurse set the boundary and Fusion Claw acted inside it. Reasoning produced options; authority, granted deliberately and in advance, decided which could run. The selected plan was implemented within the established authorization and approval framework, and the evidence to review it after the fact was produced automatically.

Example 2: Shipment load planning at scale (governed, fully autonomous)

An inbound materials plan covers 782 buy orders across an eight-week horizon. Freight utilization sits at 61%, below the 70% target, across 180 planned moves at $262K in transportation cost.

A fully autonomous outcome is defined to generate optimized loads, with permission to pull orders forward by up to three weeks. Fusion Claw evaluates forecast volume, order timing, equipment capacity, service constraints, business rules, and policy. It then proposes 154 loads built from 776 eligible orders. Average utilization rises to 88.2%, the number of moves falls by 26, and projected savings reach $58K. Fusion Claw also reports its own limits honestly: it pulled 117 orders forward and six that could not be loaded because of setup or master-data issues were routed to a planner for review.

Fully autonomous does not mean unsupervised commitment. Of the 154 proposed loads, 149 are sent downstream to the supply plan and five are held for further review. The outcome receipt records the requester, owner, executor, and approver, plus the governing policy, the data sources, the linked transactions, a verification ID, and a digital signature.

Why it matters: The system planned across hundreds of orders in one pass, but the boundary between recommendation and committed transaction remained in place. Execution was deterministic and governed, exceptions were surfaced rather than buried, and every applied change carries a traceable line back to the policy and authority that permitted it.

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

ORACLE

Commercial model and availability

Fusion Claw's commercial framing follows Oracle's outcome-first principle: customers pay for trusted execution of enterprise work, described in outcome terms rather than component or token math.

Fusion Claw ships with Release 26D.

Conclusion

Fusion Claw extends Fusion Applications from systems that record and support work to systems that can take responsibility for outcomes. Its architecture separates reasoning, authority, execution, and enforcement so autonomous work can run within defined enterprise controls.

Oracle's full-stack position is central to Fusion Claw's capabilities. The data, applications, infrastructure, models, and agent platform sit within one coherent stack, allowing reasoning, authority, execution, and enforcement to work together rather than being assembled across vendors. That foundation gives customers a governed way to delegate more work to software while keeping control over what it can do, how it acts, and how each outcome is verified.

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.