

POSITION PAPER

AI Agents Are Users, Not Features.

The next era of enterprise software won't be defined by better AI. It will be defined by a second kind of user.

ENTERPRISE ARCHITECTURE

20-MINUTE READ

V0.1 · 2026

WHY THIS PAPER EXISTS

Enterprise AI is being discussed at the wrong altitude.

The conversation is dominated by models, copilots and benchmarks. This paper argues that the harder — and more durable — problem is **architectural**: what an AI agent *is* inside an enterprise system, and how it is identified, deployed and governed.

Written for enterprise architects, CTOs and technology leaders designing systems that must still make sense five years from now. Not a product brochure — a point of view.

You don't configure an agent. You **ship** it.

Six words that move enterprise AI off the hype curve and onto ground every architect has stood on for twenty years. The rest of this paper is what follows once you take them literally.

WHY THAT CHANGES EVERYTHING

Ship it, and forty years of software discipline suddenly apply.

A feature is a black box you switch on. Something you **ship** can be versioned, staged, promoted, reviewed, audited — and rolled back the moment it misbehaves. That is the discipline enterprises already trust for software, now available to AI.

But it raises a sharper question. If this thing ships like software, yet observes, decides and acts on its own like a user — **what exactly is it?**

Every enterprise platform has added AI.

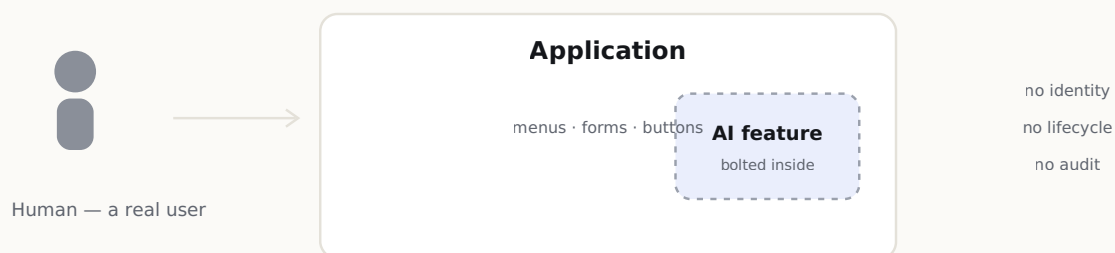
Almost all added it in the same place.

Inside the application.

As a feature.

That decision will look as dated as making the Internet a menu option.

The mistake isn't the AI. The mistake is the object model.



We put a new kind of user where menu items go.

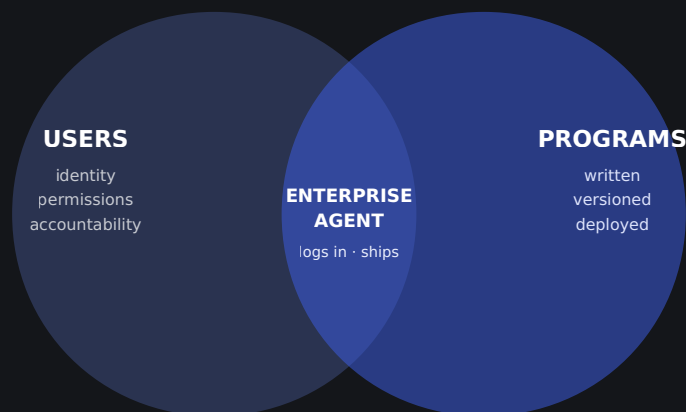
The feature model scales AI.

The user model scales governance.

THE ANSWER

One entity. Two natures.

An enterprise agent is the first thing that is **both a user and a program**. It logs in like a person. It ships like software. Every hard question about it comes from treating it as both at once.

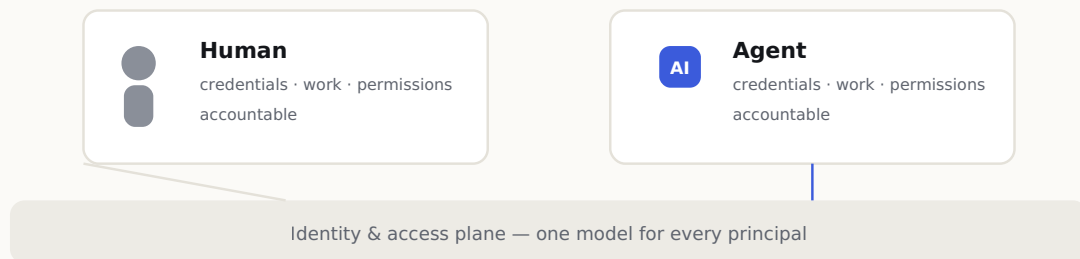


Identity systems own the left circle. Automation systems flirt with the right. Neither holds the intersection.

DEMAND ONE

It needs an identity.

Not an API key in a config file. A first-class principal on the same identity plane as your people: **its own credentials, its own assigned work, its own place in your permission model.**

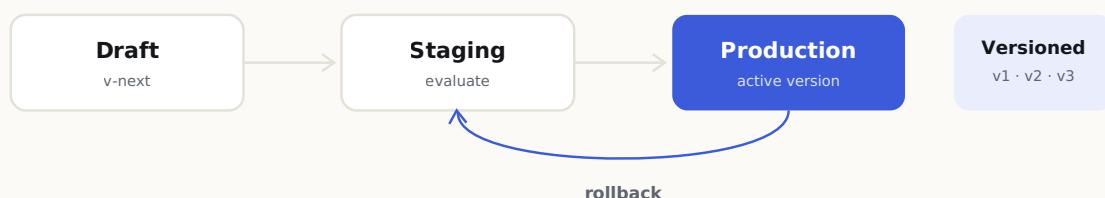


If it can act in your systems, it must be someone your systems recognise.

DEMAND TWO — THE PAGE MOST PLATFORMS CAN'T WRITE

It needs a deployment lifecycle.

You draft it, stage it, promote it to production, and roll it back when it misbehaves — **because it is software, and software is deployed, not toggled.**

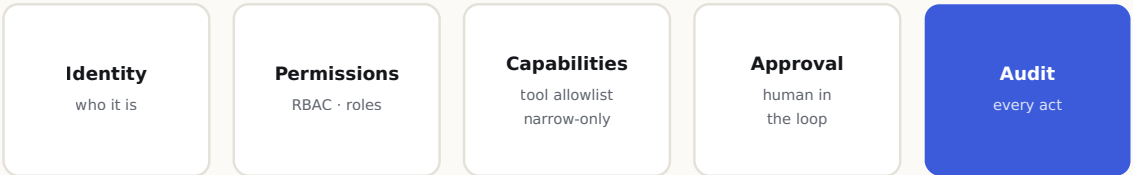


Identity is becoming crowded. This is not. It's the moat hiding in plain sight.

DEMAND THREE

It needs governance.

The same controls you already trust for people and systems — **permissions, scoped capabilities, human approval, a complete record** — applied to a principal that never sleeps.

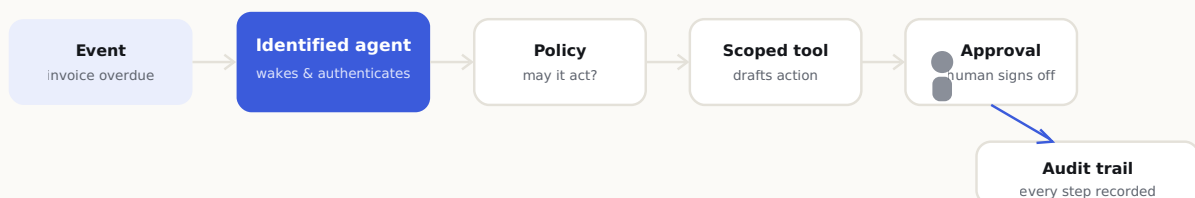


Governed like a member of the org. Logged like a system. Deployed like code.

PROOF — NO LONGER PHILOSOPHY

Watch one event move through the architecture.

A supplier invoice falls overdue. **Traditional AI waits for someone to open the application. This architecture doesn't.** An identified agent wakes on the event, checks what it is permitted to do, drafts the action with a scoped tool, pauses for a human to approve, executes, and writes the whole sequence to an audit trail.



Identity systems see step 2. Automation sees step 4. Only an architecture that treats the agent as **user and program at once** can see — and govern — the whole line.

QUESTIONS ARCHITECTS SHOULD ASK

Notice what every real question turns out to be.

Ask what it means to add an AI user to your enterprise. Not one question is about AI. **They are all enterprise-software questions.**

- | | |
|-----------------------------|----------------------------|
| 01 What is an AI user? | 05 How is it deployed? |
| 02 How is it authenticated? | 06 Who audits what it did? |
| 03 Who assigned it work? | 07 Who updates it? |
| 04 How is it governed? | 08 Who rolls it back? |

The future of enterprise AI is not a model problem. It's an architecture problem we already know how to reason about.

AI agents shouldn't be features inside enterprise software. They should be users of it.

When that sentence turns up in an architecture review that has never heard of the company that wrote it, the argument has done its job.

THIS IS NOT A PROPOSAL

This architecture already exists.

Most organisations will spend the next few years assembling this architecture from separate products. Copyl was built around it from the beginning — an agent is an **authenticated principal** with its own credentials, assigned work like a colleague, governed by the same permission model as your people, and **versioned, staged, promoted to production, and rolled back** like software.

SHIPPING TODAY

Enforced in the platform, not the pitch.

- › Agents as authenticated identity principals
- › Human permission model (RBAC) over every agent
- › Scoped, narrow-only tool capabilities
- › Human approval gates: pause, resume, cancel
- › Draft → staging → production, with rollback
- › Event subscriptions that wake agents to act
- › Execution, decision & activity logging

THE ROADMAP

The same model, deepened — not a new one.

- + Richer authority: mandates & limits on what an agent may do
- + Richer delegation between agents
- + Richer governance & measurable accountability
- + Tamper-evident, append-only audit trails
- + Hardened behavioural boundaries

Today an agent is a governed **user**. The roadmap doesn't change that model — it deepens it. · copyl.com

ABOUT THE AUTHOR



Rolf Bäck

Founder & Chief Architect, Copyl

Rolf has spent more than 30 years building enterprise software platforms. Since 2011 he has led the development of Copyl, an enterprise platform for business applications, automation and AI. His current work focuses on how AI agents become governed participants in enterprise systems rather than isolated assistants.

- Founder of Copyl
- Enterprise software architect
- Working on enterprise AI architecture since 2022
- Based in Sweden

ABOUT COPYL

Copyl is an Enterprise AI Operating System that enables organizations to deploy, govern and orchestrate AI agents alongside business applications, APIs and automations on a single platform.

copyl.com