

AGENT UX · CASE STORY

Who's watching the agents?

What it takes to let AI agents act inside your apps without losing control of them.

4 min

Click or press → to continue

Imagine giving a new hire the keys to your office.

You wouldn't hand over every key. You'd decide:

Which doors they can open

When they need a manager's OK

Who signs the log

How to take the keys back

AI agents get keys too.

Usually with none of those rules.

Meet Priya. She runs four apps, and eleven AI agents work inside them.

- A chatbot answering visitors
- A categorizer sorting bank transactions
- A writing assistant
- A spend tracker
- +7 more



Priya
Who's watching all of them?

Watching isn't the same as deciding.

WATCHING

What did it do?

Dashboards tell you after the fact.

DECIDING

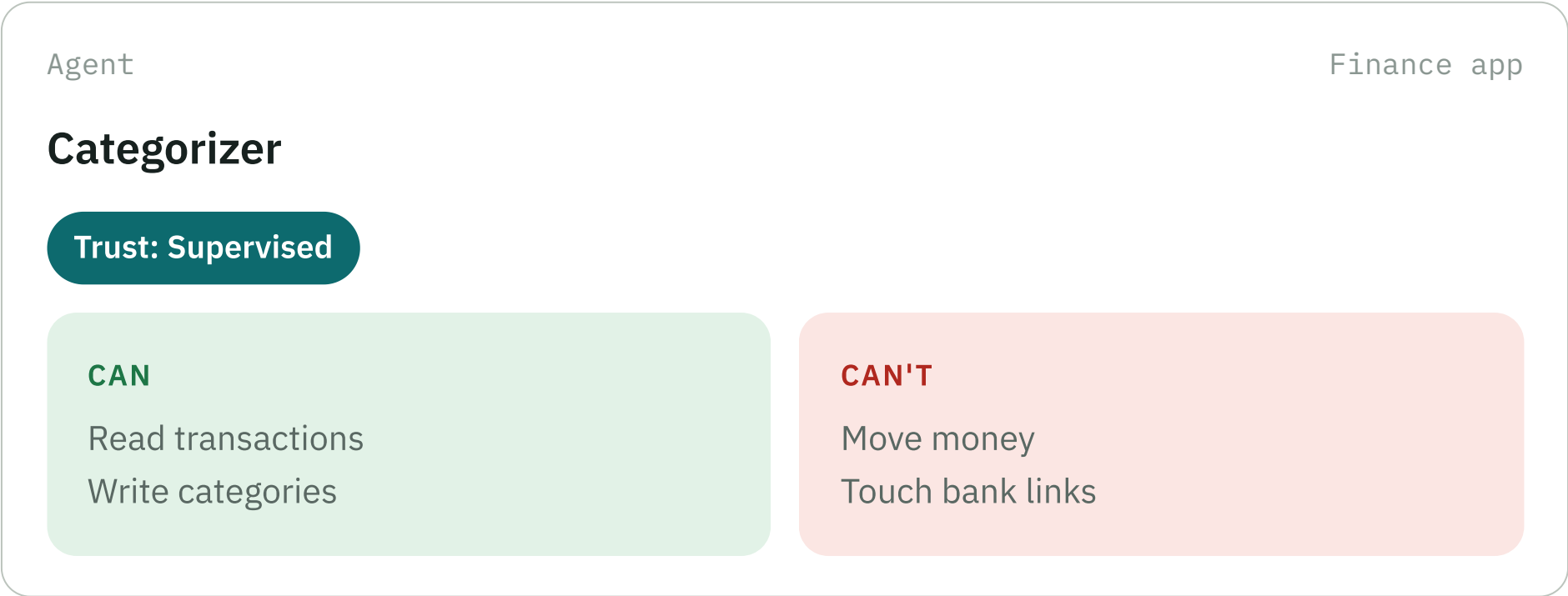
What may it do?

Rules apply before it acts.

Priya needs both. Here's how Warden does it, one rule at a time.

Rule 1

Every agent gets its own keys.



● PRINCIPLE

Least privilege

Give each actor only the access its job needs, and deny everything else by default.¹ An agent is not its owner. It gets its own, smaller identity.

Rule 2

Every action passes five gates, in order.

- ✓ Is the agent switched on?
- ✓ Does its owner have the authority?
- ✓ Is this inside its scope?
- ! Does a guardrail say stop?
- ✓ How much can it do on its own?

Allowed

Needs approval

Blocked

Autonomy is checked **last**. However much an agent is trusted, it can never go beyond its keys.

Rule 3

Guardrails beat trust.

Production needs approval

Limit how much it can touch

Rate limit

Ask when unsure

The strictest rule wins. A guardrail can stop even a fully trusted agent.

Rule 4

When an agent stops to ask, ask the right way.

Held action

Ask when unsure

Categorizer · invoke categorize · staging

Approve

Deny



Priya

What exactly am I approving?

Agents ask two kinds of questions. They need two kinds of answers.

PERMISSION

"May I re-sort 9,000 transactions in production?"

Approve or deny.

KNOWLEDGE

"What is this \$500?"

Give it the answer.

TRANSACTION

ZELLE TO P RAO

\$500.00 · Aug 12 · checking

Family transfer

Rent

Gift

Other

☒ Make this a rule, so it never asks again

Resolve

● PRINCIPLE

Ask the right way when unsure

When an AI is uncertain, it should ask the person instead of guessing.² The control should match the question, and the card should show the thing being acted on, not just the verb.

Rule 5

Allowed isn't always normal.

Anomaly

3:12 AM

Categorizer read 4,000 transactions in one minute

It's allowed to. It has never done this before.

Proposed fix: hold bursts over 500 for review

Apply fix

Always do this

The gates ask "is it allowed?" This asks "is it **normal**?"
Decide once, and Warden repeats your decision next time
and tells you it did.

● PRINCIPLE

Supervise by exception

Trust should match how reliable the system actually is.³ People step in for the unusual, not for every action.

Rule 6

An off switch that really works.

Agent

Live

Website chatbot

Answering visitors on the live site.

Pause agent

Pause it and the live chatbot stops answering within seconds. The page stays up. Only the AI stops.

● PRINCIPLE

Easy to stop, easy to undo

People need a clearly marked emergency exit.⁴ For agents, that means a pause that takes effect in the real app, not just on a dashboard.

Rule 7

Every decision answers: on whose authority?

Audit

Re-sort 9,000 transactions

✓

Agent switched on

✓

Owner: Priya

✓

In scope

!

Guardrail: production needs approval

Held for Priya

Every attempt is on record with the reasoning that allowed or stopped it.

And it watches

What the agents cost, and who they serve.

AI spend per agent

Visitors, with no third-party trackers

Cost per visitor

Uptime for every app

Watching and deciding, in one place.



Priya

I know what every agent can do, what each one did, and I can stop any of them.

BEHIND THE STORY

Priya's setup is mine.

Warden is a real product I built. It governs eleven AI agents across four of my live apps. The chatbot on my portfolio asks Warden for permission before every answer.

I started it to test the waters for a Datadog interview. Then the problem got more interesting than the interview, and I couldn't stop.

And the Resolve control? It started with me looking at a held transaction and asking: **what exactly am I approving?**

CHEAT SHEET

Letting AI agents act, safely

- 1 **Own keys per agent.** Deny by default.
- 2 **Check in order:** on, owner, scope, guardrails, then autonomy.
- 3 **Guardrails beat trust.** The strictest rule wins.
- 4 **Two kinds of questions:** yes/no for permission, an answer for knowledge.
- 5 **Watch for abnormal,** not just forbidden.
- 6 **A real off switch.**
- 7 **Record who decided,** and why.

TRY IT

Explore Warden.

The public demo shows the whole console with clearly labeled sample data.

Explore Warden: warden.anujaharsha.com

You finished case story 01: "Who's watching the agents?" Warden is a real product by Anuja Nimmagadda, built by directing Claude Code. Priya and the examples in this story are fictional.