

From Smart Answers to Trustworthy AI

Designing Governed Copilot Studio Agents

COPILOT STUDIO

SHAREPOINT

POWER AUTOMATE

Karl-Johan Spiik • Senior Consultant • Microsoft MVP

Get the slides

FROM SMART ANSWERS TO TRUSTWORTHY AI



Scan the QR code to open or download the presentation.

Karl-Johan Spiik

FROM SMART ANSWERS TO TRUSTWORTHY AI



Microsoft MVP and Senior Consultant

- Projects with Copilot Studio, Power Automate, Power Apps and Dynamics 365
- 19 years of experience in Microsoft development and project delivery
- Past eight years working with Power Platform
- Past three years working with AI
- Experience with agentic development
- Five books about Community-Led Orientation
- Blog and books: www.karlex.fi

karl-johan.spiik@karlex.fi

The enterprise AI challenge

A confident wrong answer

sounds helpful, but creates risk

“I don’t know”

is defensible and actionable

Which outcome would your security and compliance teams prefer?

Trust starts by limiting what the agent is allowed to use.

The story

01

Control the scope

Knowledge boundaries

02

Govern the lifecycle

Ownership, expiry, retention

03

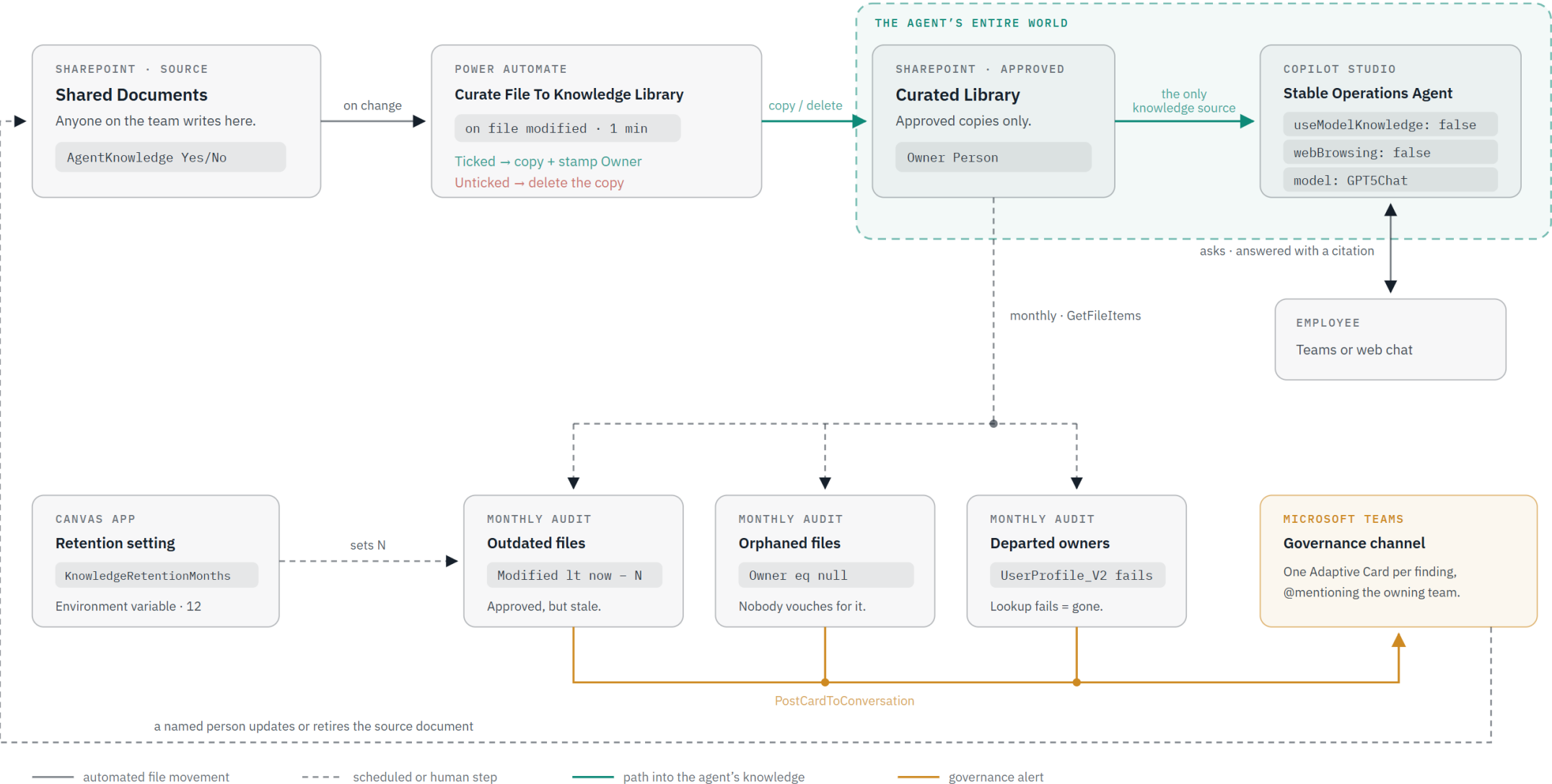
Enforce the rules

Power Automate boundary

04

Prove the behavior

Demo + test-driven
development



A person ticks a box. That tick is the only door into the agent's knowledge — and three scheduled audits keep watch on what is already inside.

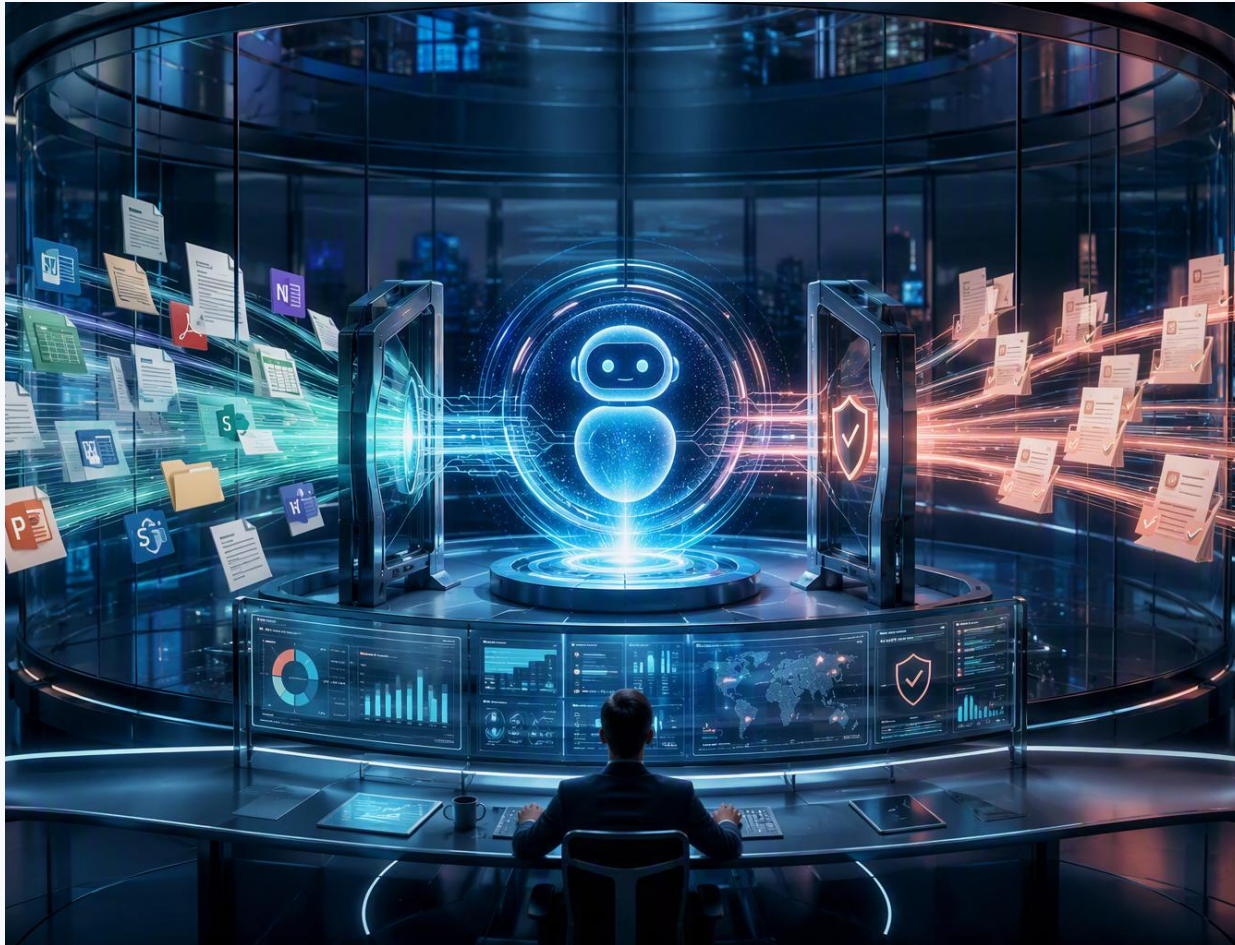


01

Control the knowledge scope

Trust begins before the prompt: decide which content may participate in an answer.

More sources do not automatically mean better answers



Uncontrolled sources

Duplicates, conflicts and accidental oversharing

Lifecycle blind spots

Stale, expired or deleted source content

Missing accountability

No owner means no authority to validate

The agent can only be as trustworthy as the sources it is allowed to see.

A governed agent has an explicit contract

WHO

Which user or group?

WHAT

Which approved content?

WHY

Who owns the decision?

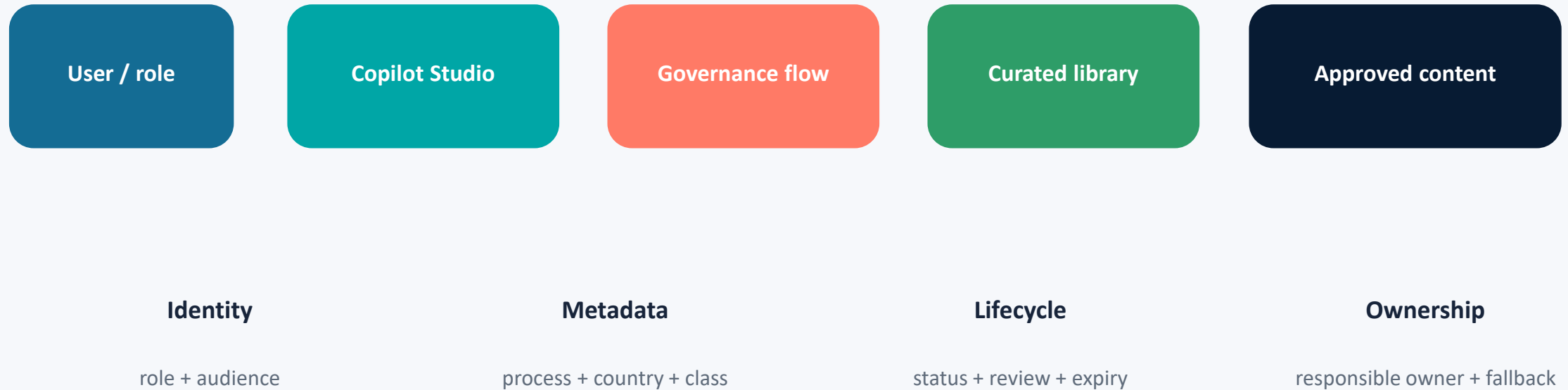
WHEN

Is the content still valid?

**USER-CONTROLLED
KNOWLEDGE SCOPE**

The user controls what the AI is allowed to know.

Reference architecture



Every arrow is a governance decision, not just an integration.

Metadata turns a document into governed knowledge

Policy document

Business process	Procurement
Country	Finland
Classification	Internal
Owner	Process team
Review date	2026-10-01
Status	Published

RULES

**AVAILABLE
TO THE AGENT**

If the metadata does not prove eligibility, the content stays out.

Knowledge has a lifecycle, not just a location



- | | | |
|---|----------------|-----------------------------|
| 1 | Create | Not searchable |
| 2 | Review | Not searchable |
| 3 | Approve | Not searchable |
| 4 | Publish | Agent eligible |
| 5 | Retire | Immediately excluded |
| 6 | Delete | No longer officially exists |

The orphaned file problem

**The file still exists.
The authority behind it does not.**

- Owner left the organization
- Review date passed
- No fallback owner assigned
- Agent can still retrieve the file

GOVERNANCE DECISION

**Remove the document
from the agent's
approved scope**

and route the user to the accountable
process team

Demo: three questions, three governed outcomes

01

**Valid and approved
source**

Answer with source grounding

02

Expired knowledge

Refuse and explain next step

03

Missing owner

Exclude and route to process team

Success is predictable behavior, not the maximum number of answers.



VIDEO

02

Prove and maintain trust

The final 15%: develop the governed agent through evidence, not intuition.

Test-Driven Agent Development



1 Product owner

Defines real questions

2 Evaluations

Run against the agent

3 Claude Code

Implements targeted change

4 Control run

Repeat with nothing changed

The tests decide the next change. The human decides which failure is worth acting on.

One run is not evidence

Same agent. Same questions. Same grader.

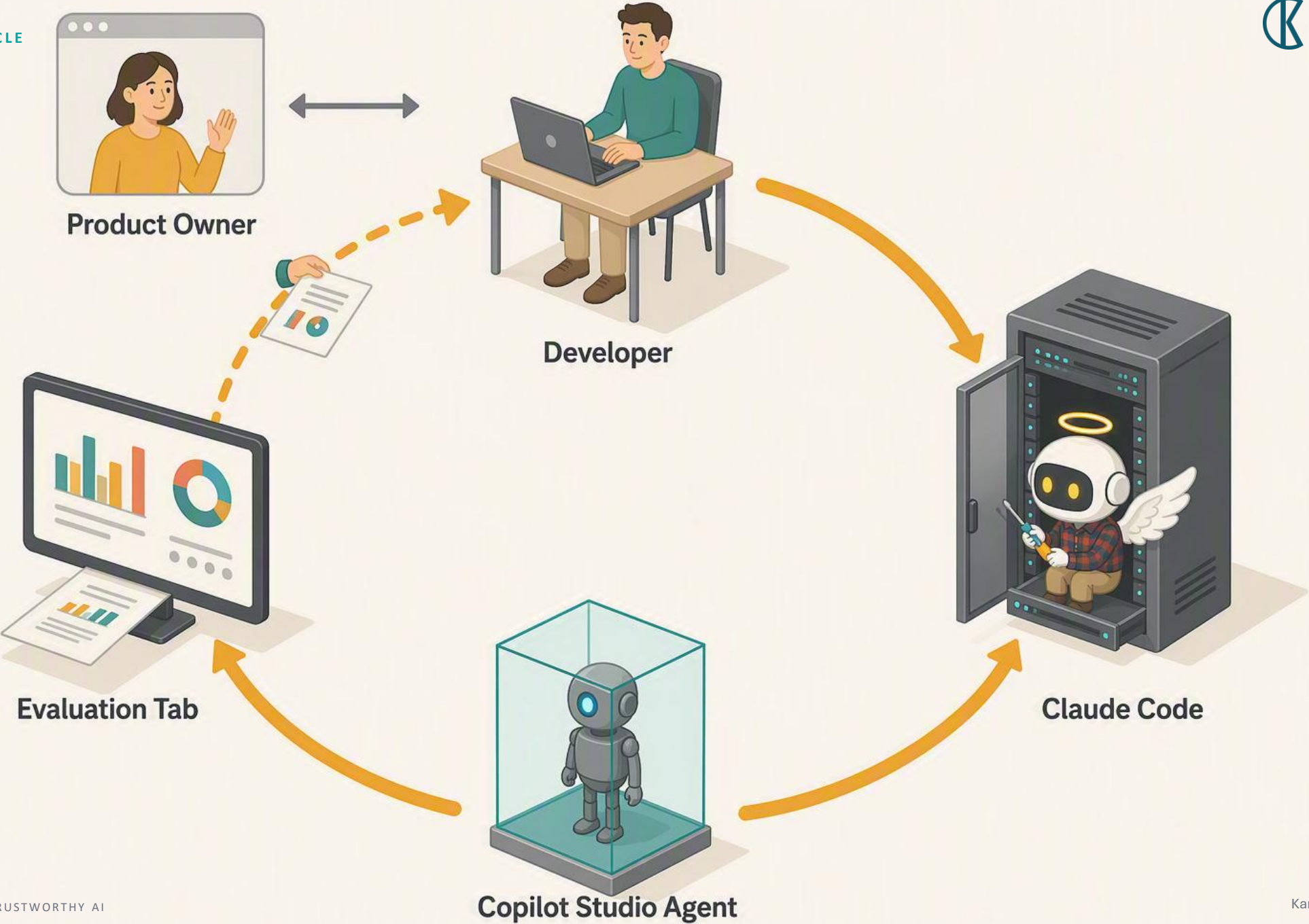
Set	Round 1	Round 2 no changes
Core process	5	8
Operational detail	9	9
Coverage sweep	10	14
TOTAL	24 / 43	31 / 43

7 points of improvement

from changing nothing

Without the control run, noise
would have been treated as
regression.

Repeat first. Diagnose second. Change third.



Not every failure belongs to the agent

Agent

Reachable content, wrong response

Fix the agent

Measurement

Expected answer or grader is wrong

Fix the test

Structural

Required source or capability is missing

Change product design

Content gap

Authoritative information does not exist

Create or improve content

Four principles to apply tomorrow

1

Scope before prompts

Define exactly what the agent may know.

2

Metadata before retrieval

Eligibility must be provable and machine-readable.

3

Lifecycle before launch

Design for expiry, deletion and owner loss.

4

Evidence before change

Repeat evaluations and classify failures first.

Defensible answers beat impressive answers.

A successful enterprise agent is not the one
that answers the most questions.

It is the one that answers only the questions
it is allowed to answer.

Defensible AI > Impressive AI

Q&A