Agentics Day:
MCP + Agents
NORTH AMERICA

Toward AI-Native Platform Operations

Releasing 100+ microservices with agentic AI



Kai Levin
AI adoption in Ericsson ADP

Agentic AI AI-Native

Heading Towards AI-Native



AI across the lifecycle

What does AI-native actually change in the way we work?

Value comes through the workflow



AI usage

AI-native workflow

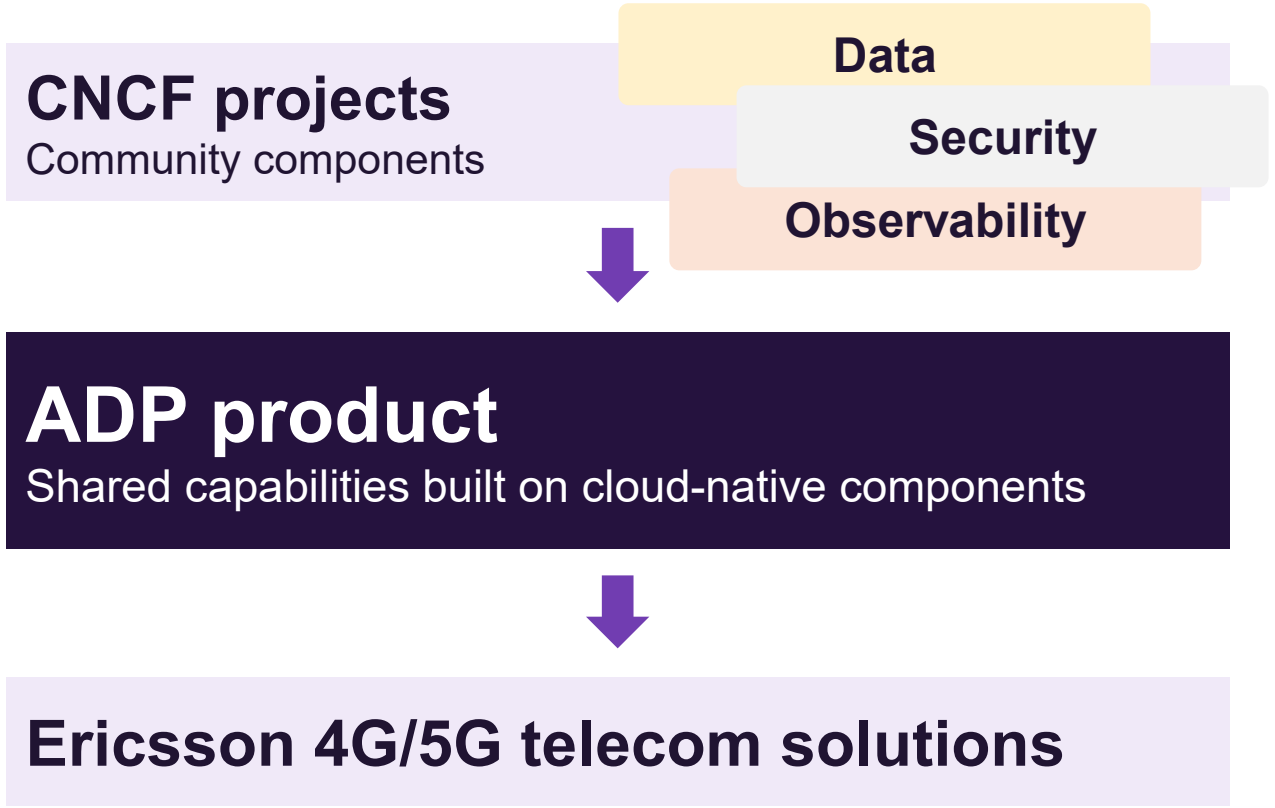
Measurable efficiency

Business value

ADP builds

100+

shared microservices as generic building blocks



ADP = Application Development Platform.

ADP Product Release

Major Release

Every

2–3 weeks

Security Patch

Every

2 weeks

or on demand

On-Demand Release

Based on user requirement

Every release stacks several processes

Build & test	results
Security checks	findings
Documentation	documents
Approvals	decisions
Publishing	artifacts
Registration	records

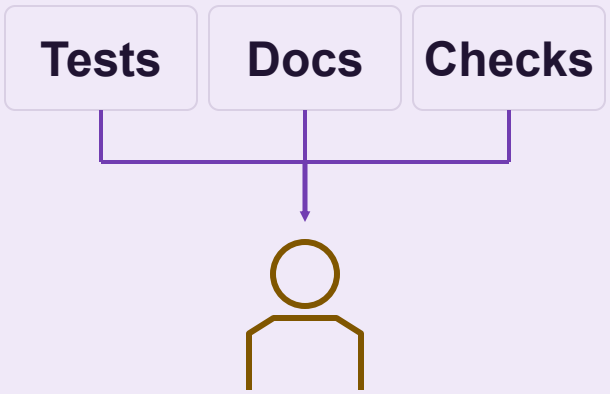


Release owner

Gathers context
Interprets evidence
Follows up

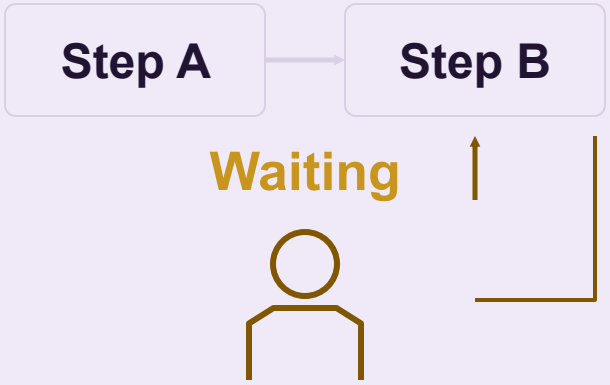
The same coordination repeats across services

Scattered evidence



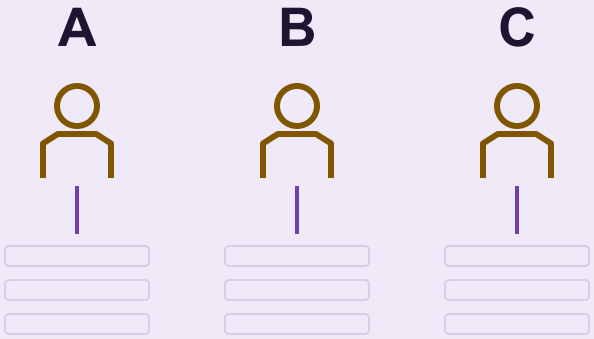
Owners gather and interpret results

Manual follow-up



Owners chase handoffs and exceptions

Repeated coordination



Every service repeats the work

Recurring release work adds up

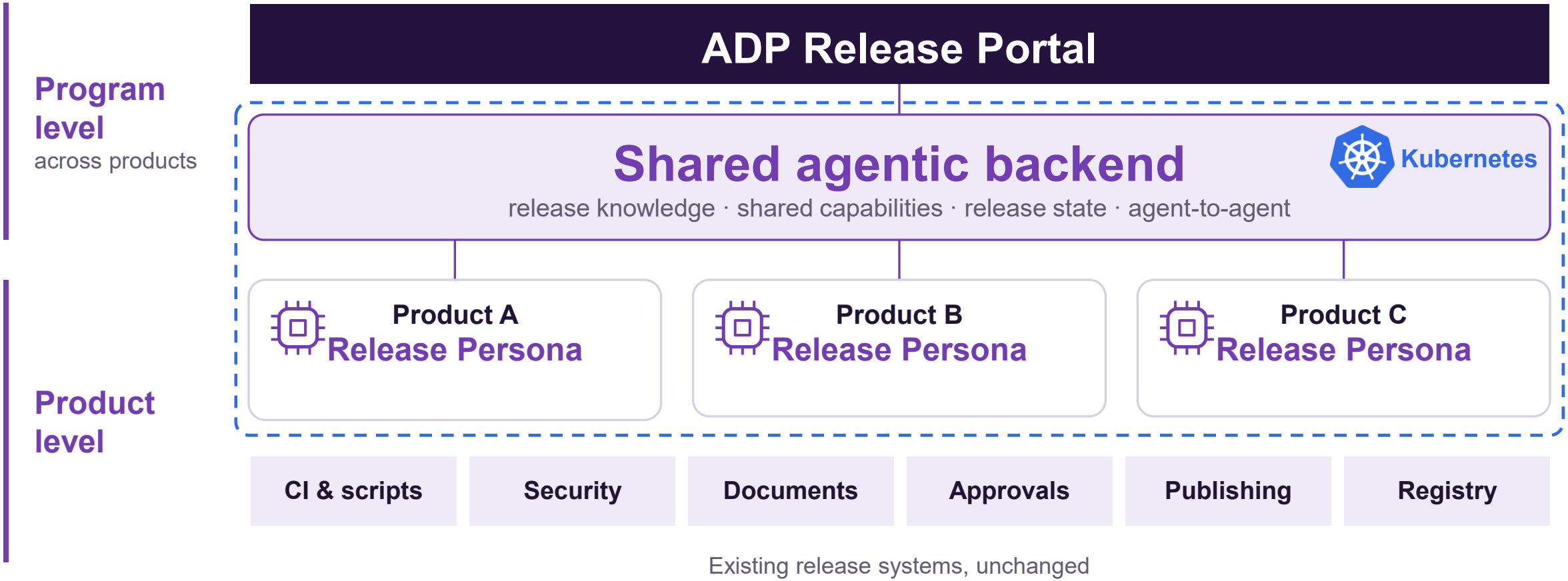
11.1%

of capacity spent
on release handling

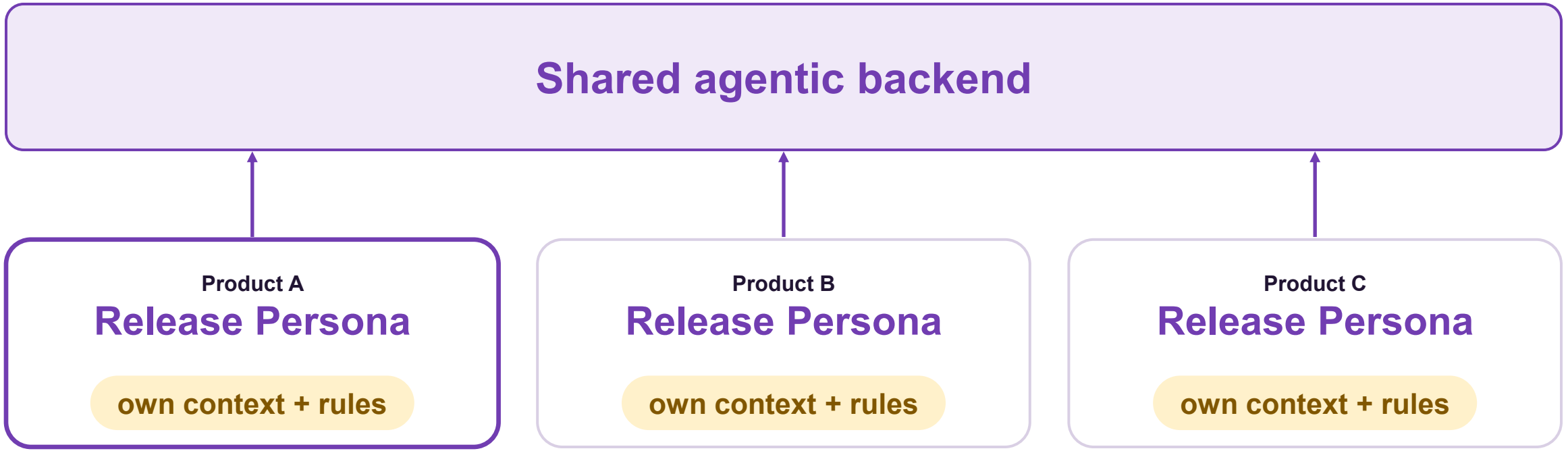
Where does that coordination live in your organisation?

ADP Baseload Survey 2025 · self-reported across all ADP product development teams

Making Product Release Agentic and Autonomous

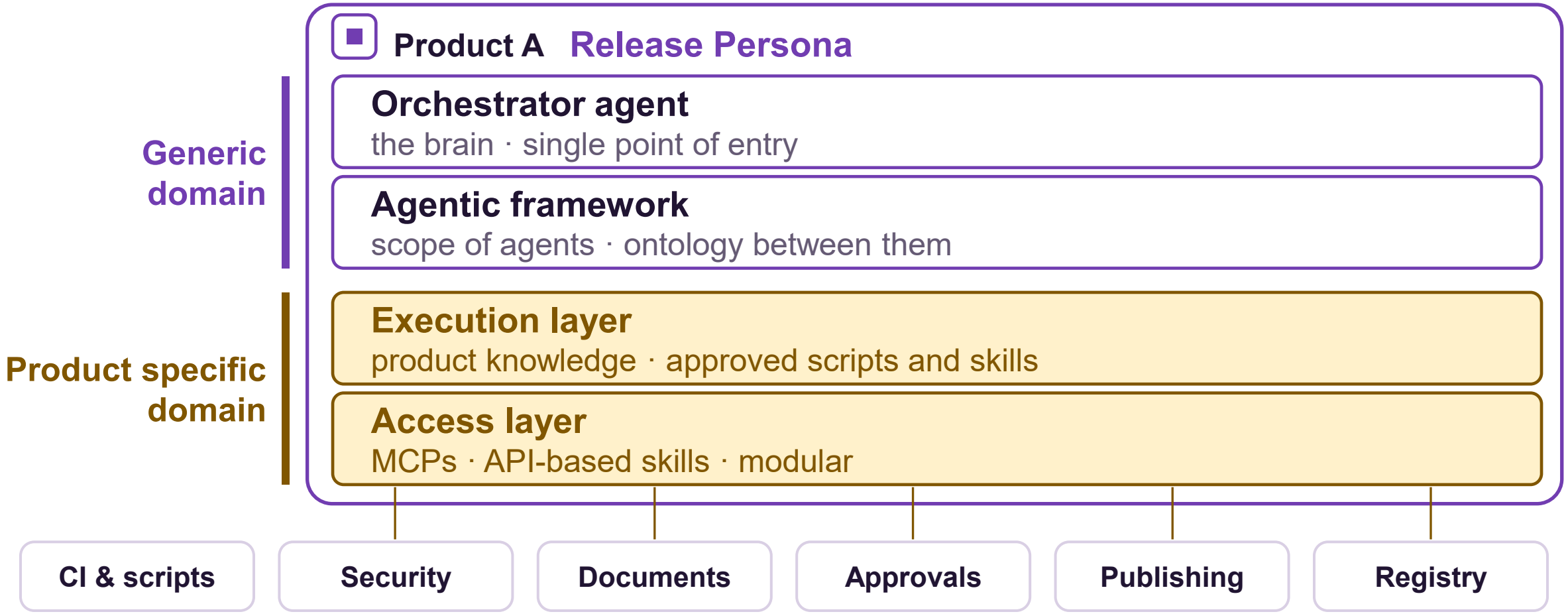


Every product gets its own Persona



Inside a Release Persona

Four layers in two domains.



Product specific domain

Each product team owns its product specific domain.

Product A Release Persona

Generic domain

Orchestrator agent

Agentic framework

Product specific domain

Execution layer
product knowledge · approved tools

Access layer
MCPs · API-based skills

Owned by the team

- Built by agents
- Team approves fixes
- Learns each release

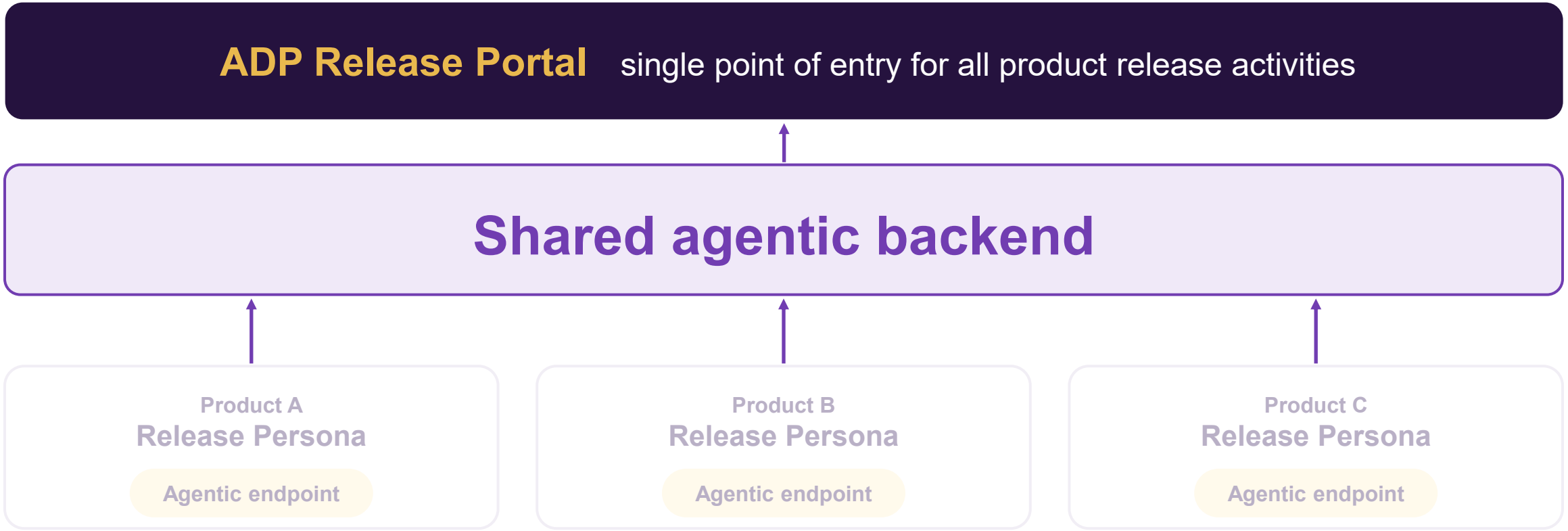
CI & scripts | Security | Documents | Approvals | Publishing | Registry

Generic domain

One agentic framework, with a single orchestrator as the brain.

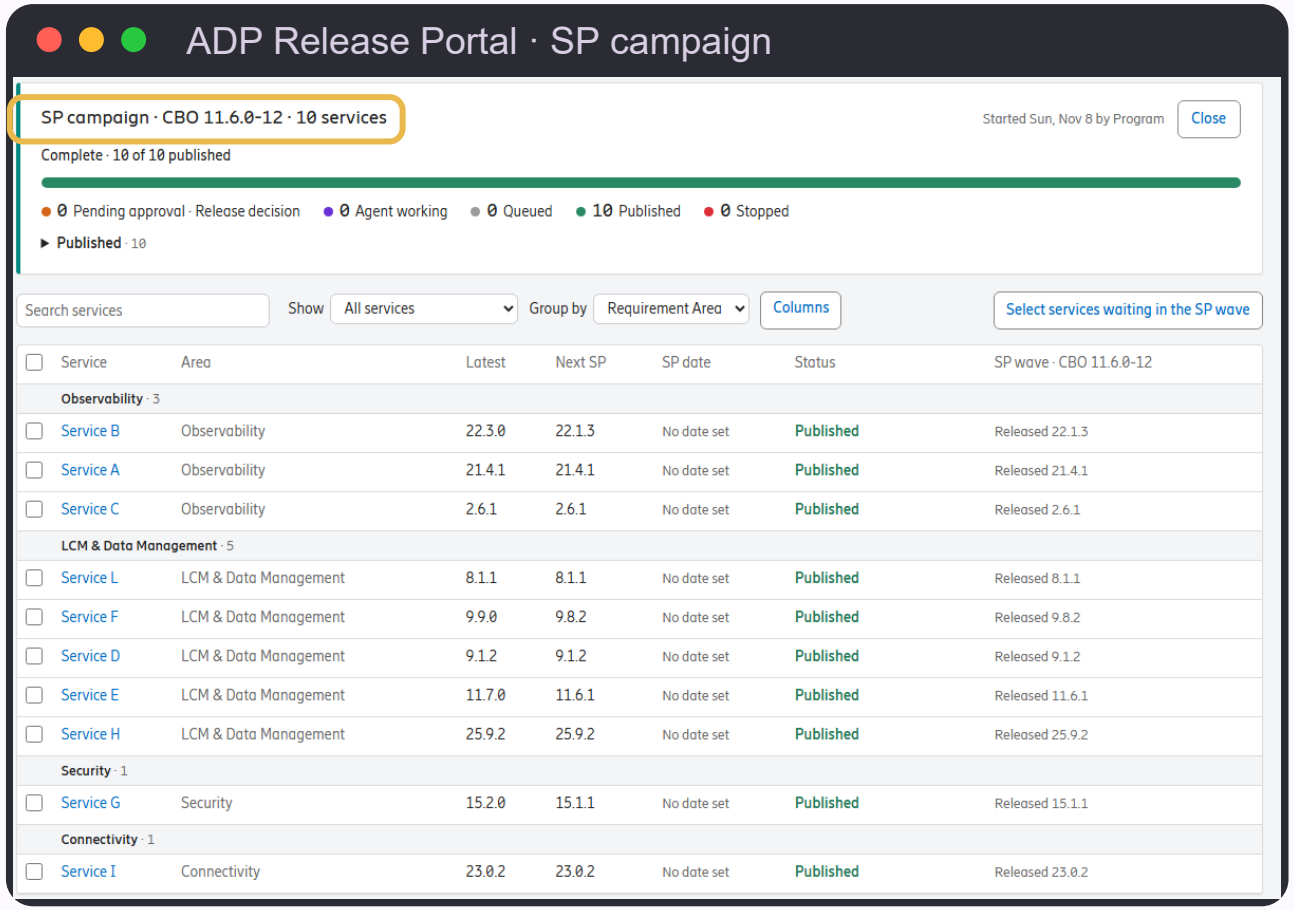


One shared agentic backend



One view across releases and stakeholders

One centralized interface for product releases.



Product development teams

Program management

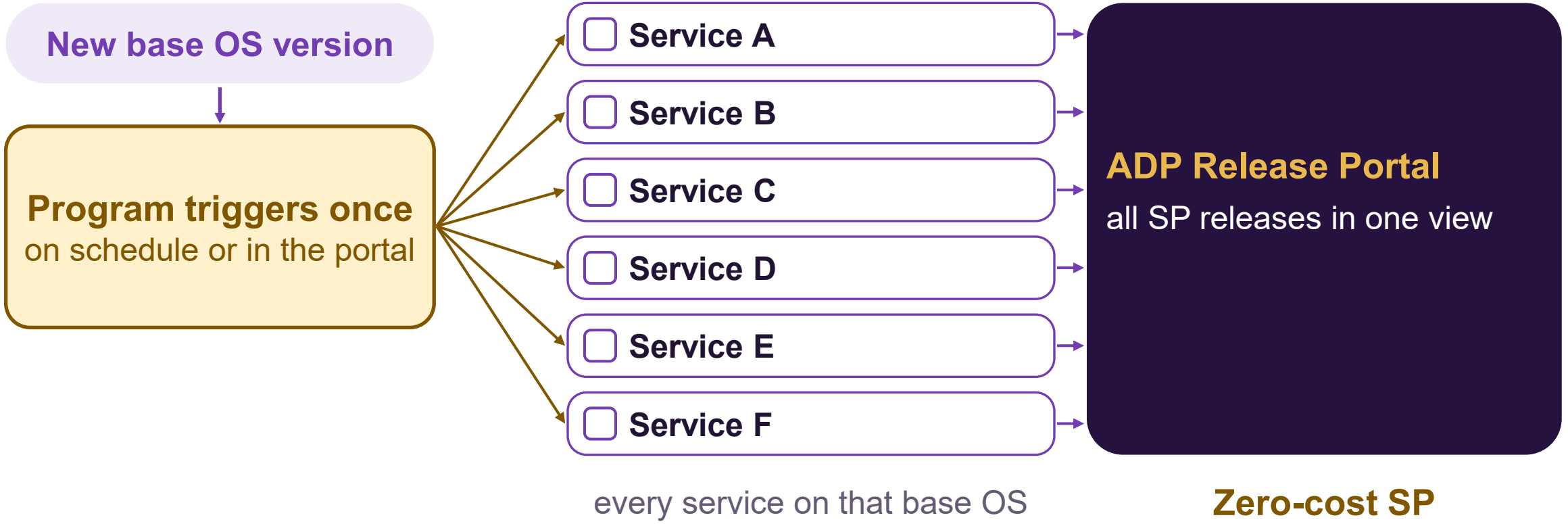
Users

↓

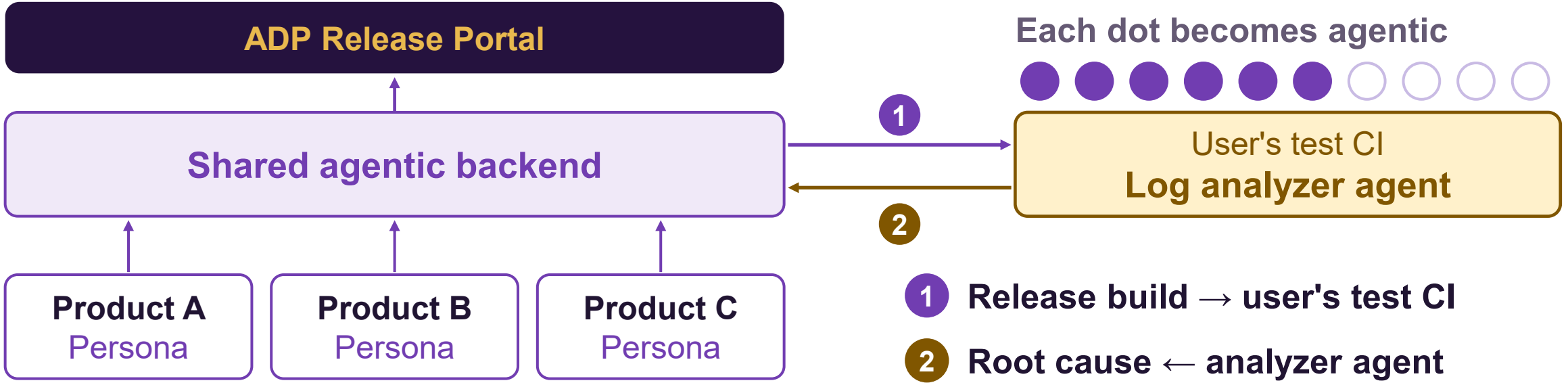
trigger · steer · observe
track · communicate

Autonomous SP Release

One trigger. Every service on the new base OS releases its security patch.



Extending autonomy, agent to agent



1. **Release build → user's test CI**
2. **Root cause ← analyzer agent**
3. **Persona re-triggers the build**

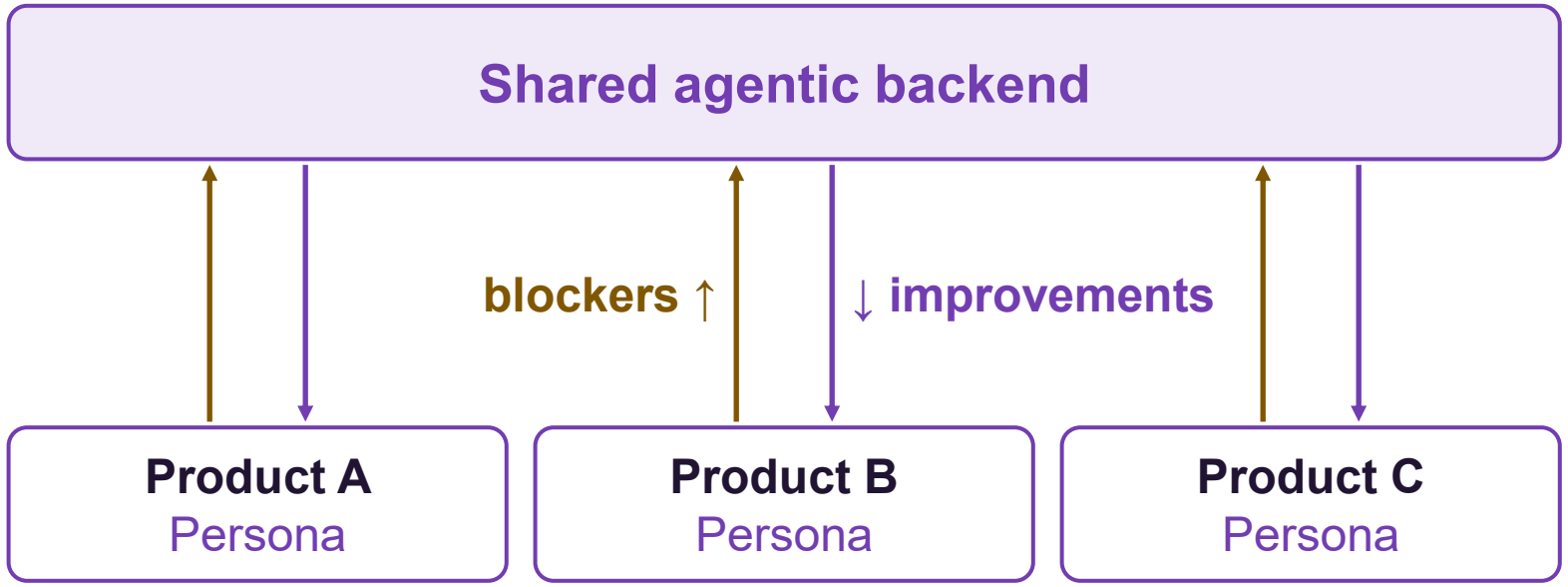
Hours of human coordination → **agent to agent**

Improve once. Reuse with local context.

Before

- **Service A** ?
- **Service B** ?
- **Service C** ?

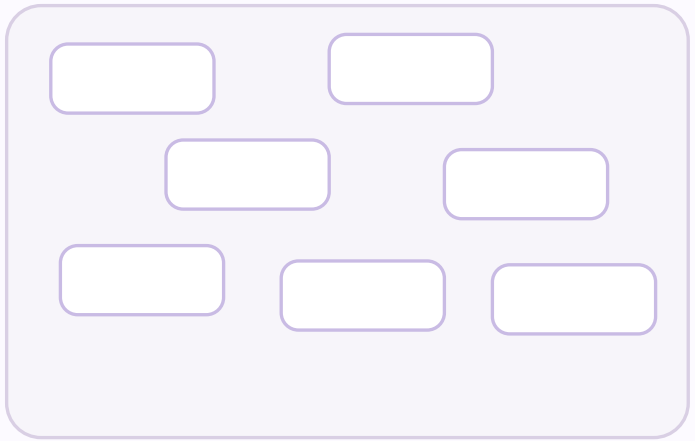
Blockers stay hidden

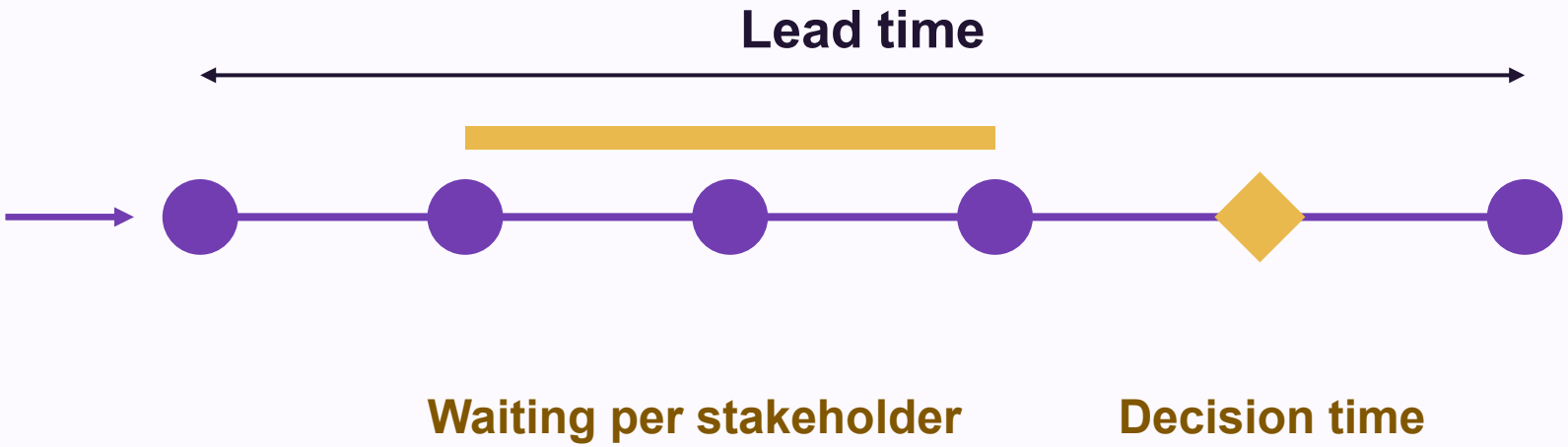


every Persona gets the improvement, with its own context

Agents surface the blockers across services; we improve once, for all

One agentic flow makes every release measurable





Scattered manual steps
hard to measure

One agentic flow · every step recorded

- **Release KPIs**
- **Cross-org coordination**
- **Improvement evidence**

Rolling out in three stages

ownership moves to the agents

L1 Orchestrated

Clone the Release Persona, knowledge graph builds up

Every action approved

L2 Owned

Agents learn each release, take over the execution path

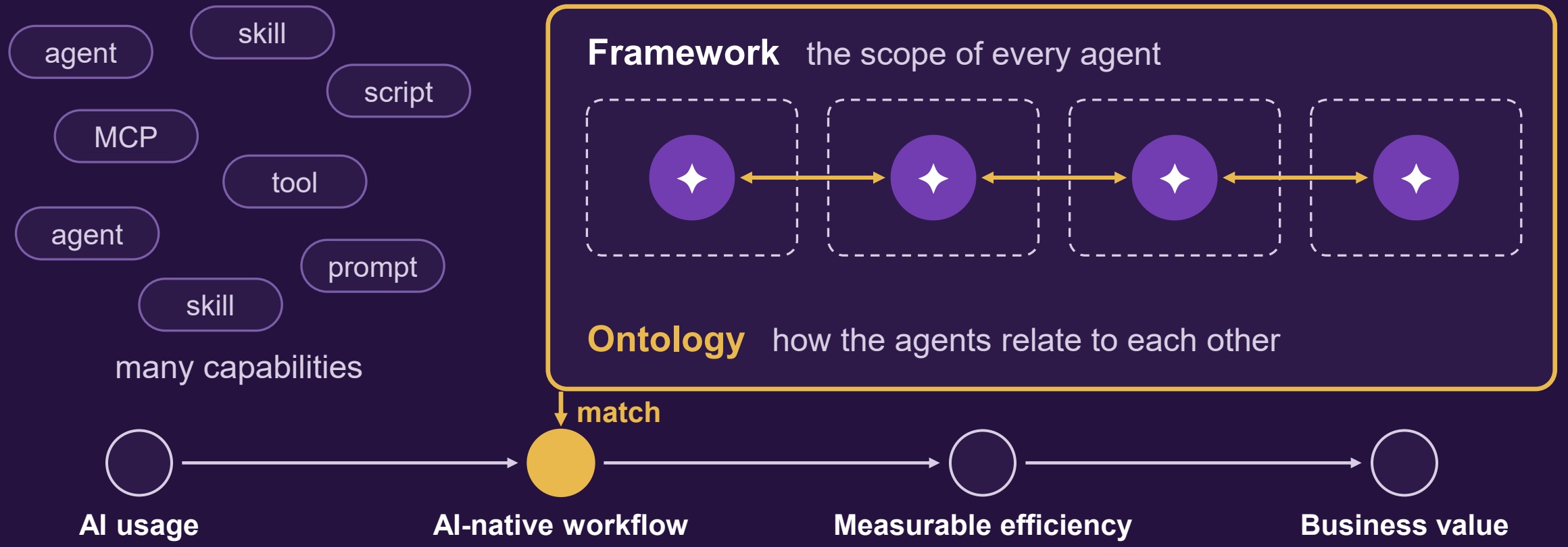
Human supervision

L3 Autonomous

Personas join the portal, coordinate across products

One organisational view

What we learnt



Framework and ontology turn many capabilities into one model.

Thank you

Autonomous release can be a shared capability.

Which recurring workflow would your teams start with?

Toward AI-Native Platform Operations: Releasing 100+ Microservices with Agentic AI

Rate this session

Scan to rate

Connect with me!



Kai Levin

AI adoption in Ericsson ADP

Kai.Levin@ericsson.com

Acknowledgement

Early PoC implementation



Afshan Hashemi

Cloud Developer

Afshan.Hashemi@ericsson.com

Responsibility remains explicit

Release agent

Coordinates the workflow
Uses approved execution
Maintains release state

Product owner

Defines decision boundaries
Takes release decisions
Handles escalated exceptions

Stakeholder agents

Assess their own domain
Return evidence and status
Escalate within their authority

Shared coordination respects each team’s authority

Measure the full business effect

Measure	Include
Human effort	Preparation, review, rework, coordination, maintenance
Setup effort	Onboarding, integration, capability development
Elapsed time	Release duration and waiting
Runtime cost	Compute and supporting infrastructure
Quality	Escaped issues and repeat incidents
Product outcome	What the organisation achieves with the capacity

Compare like for like.

Count all costs.

Protect quality.

Compare like-for-like releases. No savings inferred from the 11.1% survey.

Observed in the pilot · qualitative

What the pilot taught us

Autonomy grows with capability

As the agent's capability improved, teams handed it more of the release.



Once onboarded, easy to tweak

A product is onboarded by describing it; after that, a change is an edit, not a rebuild.

