



SEPT 22 - 24, 2026 | NAIROBI, KENYA

The Evolution of Platform Engineering: Building IDPs for Humans and AI Agents



Lakmal Warusawithana

VP, Distinguished Engineer & GM, Choreo Business Unit
OpenChoreo Maintainer

Cloud-Native Gave Us Power, And Complexity

- Kubernetes, service meshes, GitOps, observability stacks, gateways, secrets management, policy engines.
- We didn't remove complexity, we moved it from the infrastructure team to the developer.
- Deployments, Services, Ingress, network policies, telemetry, secrets, CI, cloud IAM... just to ship an app.

Developers want to deploy applications. They don't want to assemble infrastructure.



Platform Engineering Changed The Interface

- It isn't just another infrastructure layer — it changes the interface developers interact with.
- Platform engineers work with Kubernetes primitives.
- Developers work with application abstractions.
- *"I have a service. This is the API. I need a database. Deploy to dev. Promote to prod."*

The platform translates that intent into infrastructure.



The Internal Developer Platform

Self-Service

Developers don't file infrastructure tickets.

Golden Paths

Common patterns, encoded into the platform.

Automation

CI/CD, deployment & observability, automatically.

Governance

Freedom, without bypassing controls.

*The objective isn't to hide everything.
It is to expose the right abstraction to
the right user.*

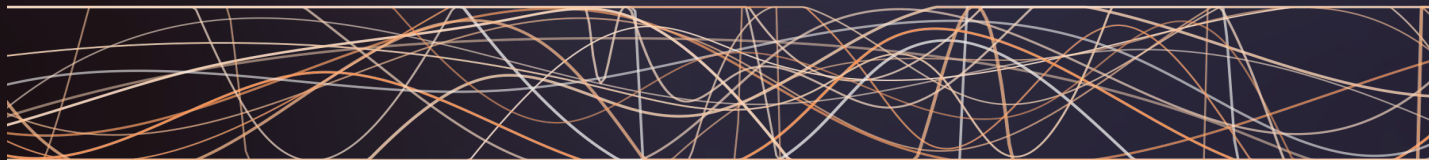


The Messy Middle

Every company building an IDP eventually builds some version of this layer.

Developer Portal

The Messy Middle



Underlying Systems

GitHub Actions

ArgoCD

Kubernetes

Prometheus

HashiCorp Vault

Gateway

OpenSearch

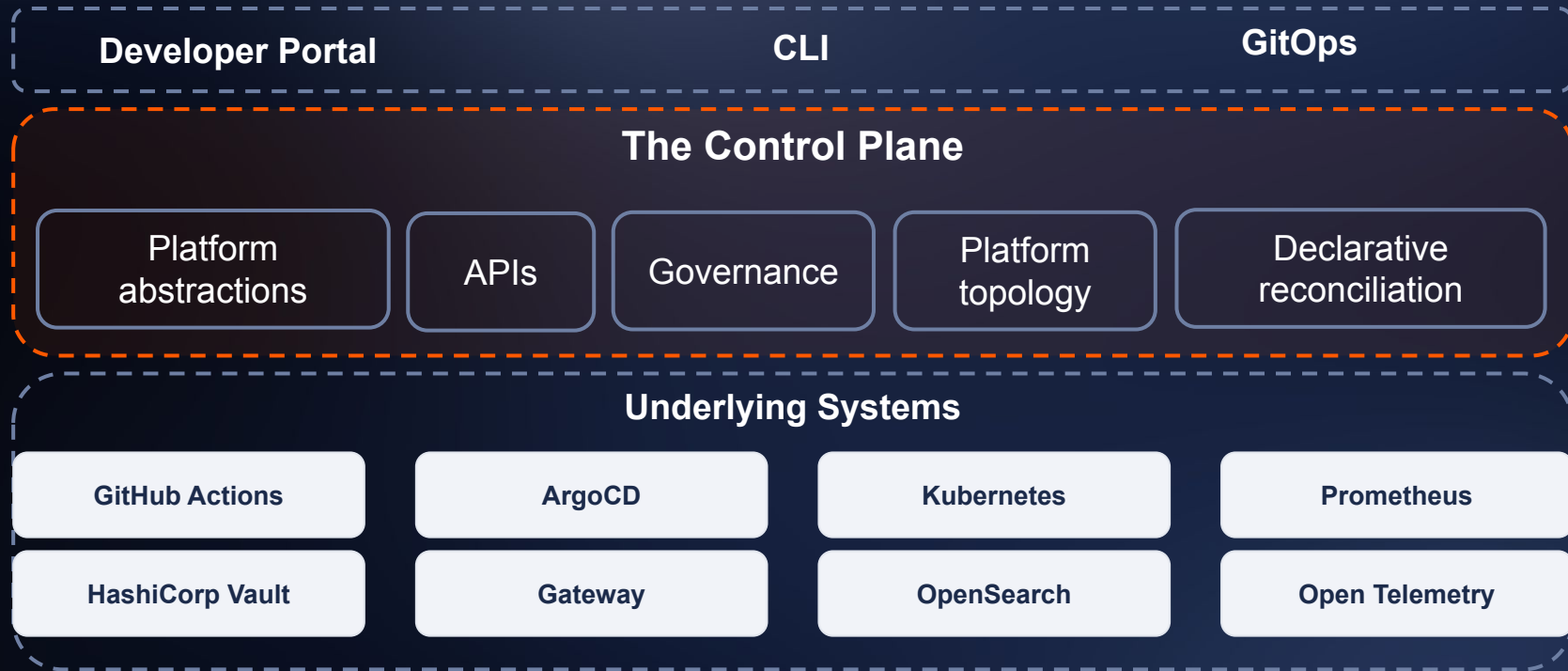
Open Telemetry

A portal is the front door. The platform is everything behind that door.



The Control Plane Becomes Critical

Replace custom integration logic with a common platform model and declarative reconciliation.



Our Evolution At WSO2

V1

Build Everything

Custom developer experience, tightly coupled to cloud infrastructure.

V2

Separate The Platform

Clearer APIs. Control and data planes separated.
Cloud-independent.

V3

OpenChoreo

A real control plane with reconciliation, Kubernetes as system of record.



A New Platform User Arrived

We spent years building platforms for developers. Then, suddenly, we had another consumer: AI agents.

- Agents increasingly write code and open pull requests
- Run tests and deploy software
- Investigate incidents and query telemetry
- Change configuration

The developer is no longer the only actor interacting with the software delivery system.



Humans And Agents

Humans

- Portal
- CLI
- Git
- API

Agents

- Programmatic interfaces
- Context
- MCP

These shouldn't become two separate platforms, the same platform abstractions power both.



Don't Give The Agent Your Kubernetes Cluster

The problem isn't capability. It's uncontrolled execution.

- Too Many Paths to the Same Outcome
- With raw Kubernetes access, an agent can accomplish the same task through many different operations, with very different consequences.
- Skills Don't Eliminate Complexity
- Skills and instructions help, but overlapping use cases, conflicting guidance and evolving infrastructure make them difficult to maintain.
- Low-Level Access, High-Risk Actions
- Broad credentials expose destructive operations, privilege escalation paths and unintended changes.

Give agents platform abstractions, not infrastructure credentials.



Agents Are Also Workloads

Like Applications

- Identity · Deployment
- Configuration · Resources
- Network access
- Observability · Security
- Lifecycle management

New Concerns

- Tools · Models
- Memory & context
- MCP servers
- Permissions
- Sandboxing

The platform has to manage applications, resources, and now agents.



Same Platform, New Requirements

This doesn't mean throwing away everything we've learned about platform engineering.

- RBAC · Least privilege · Policy enforcement
- Identity · Audit trails
- Observability · Network isolation

The principles haven't changed. The participants have.



Let's look at a practical implementation





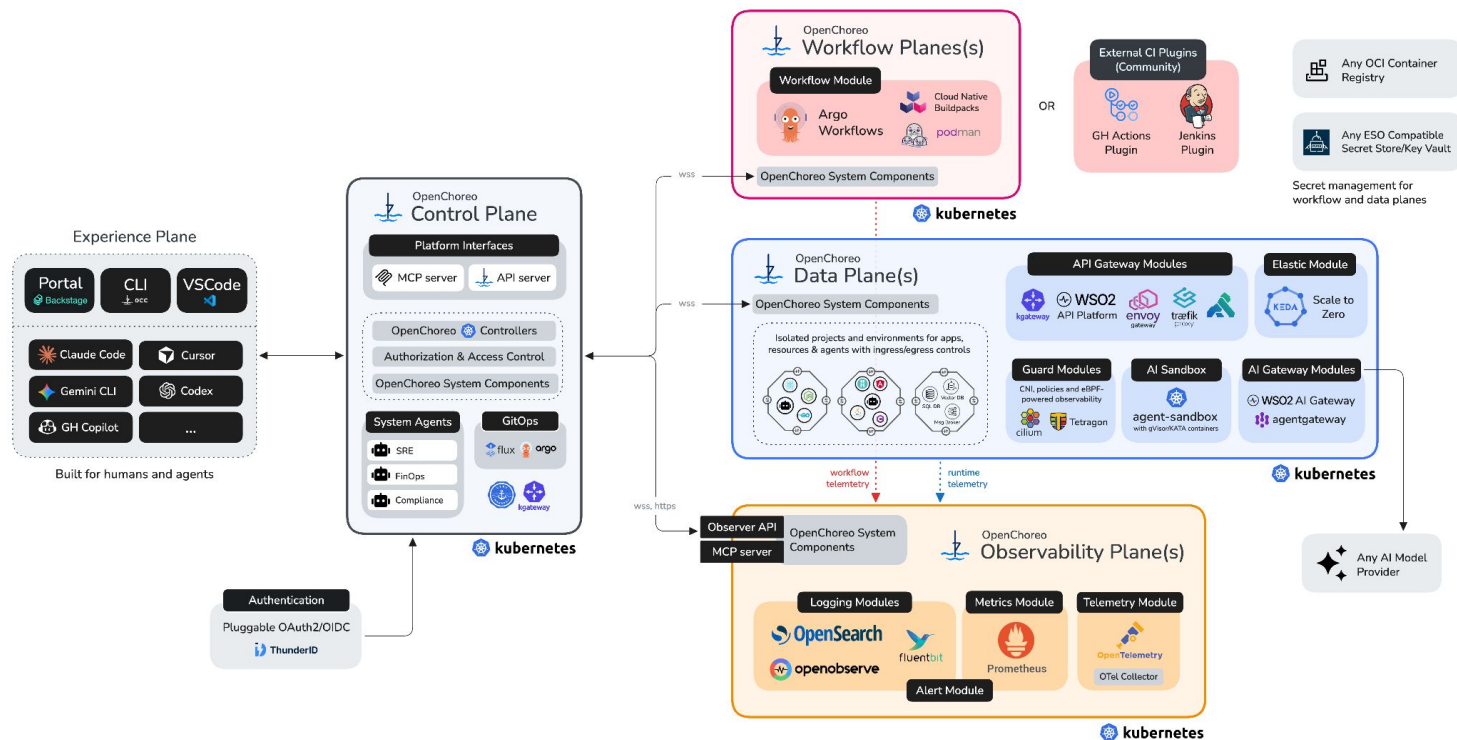
OpenChoreo is a CNCF Sandbox project.

**Go from zero to platform, or take what you have further.
One IDP for humans and agents.**

Made with ♥ at [WSO2](https://wso2.com).



What This Looks Like In OpenChoreo



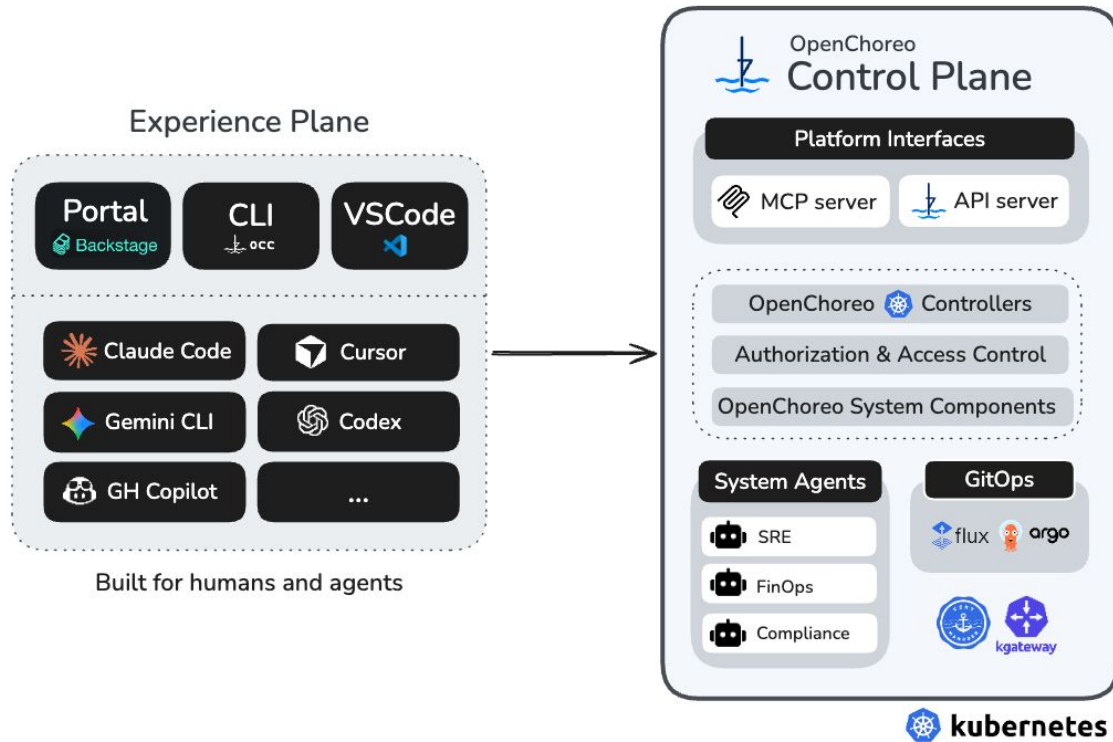
Humans + Agents

Developer Portal, CLI, API and GitOps for humans.

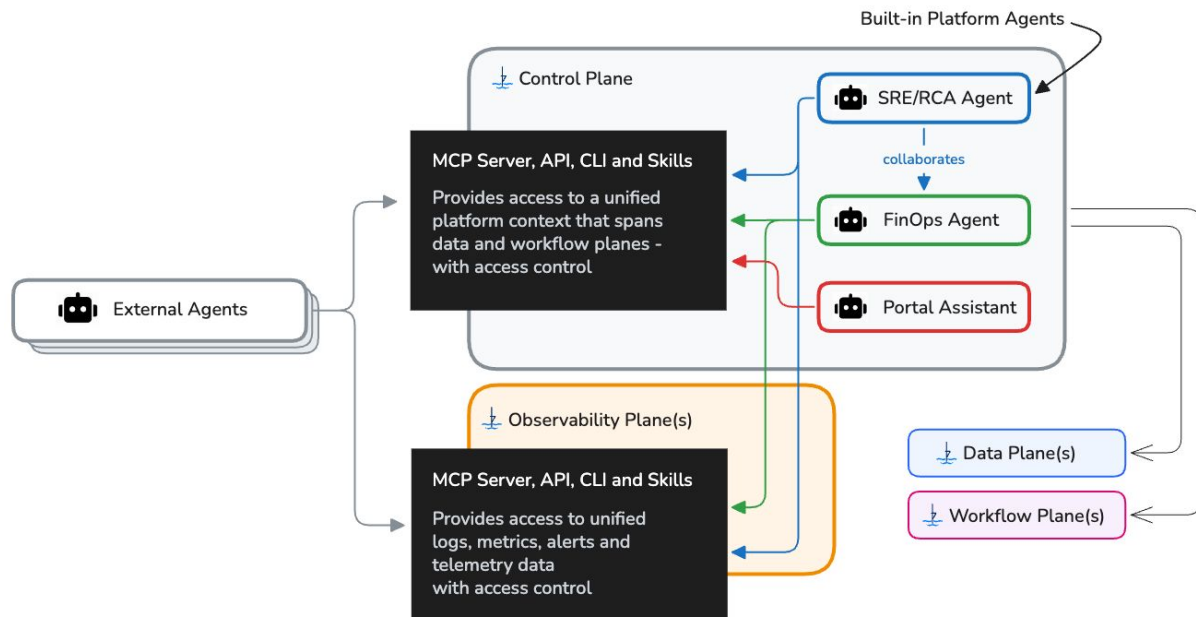
MCP for agents.

Both reach the same control plane — so an agent can ask:

- “What’s running in production?”
- “What changed before this incident?”
- “Promote this version.”



Agents Inside The Platform



Agentic Capabilities

SRE Agent

Incident RCA & remediation

FinOps Agent

Cost-aware reasoning

Portal Assistant

Context-aware assistant

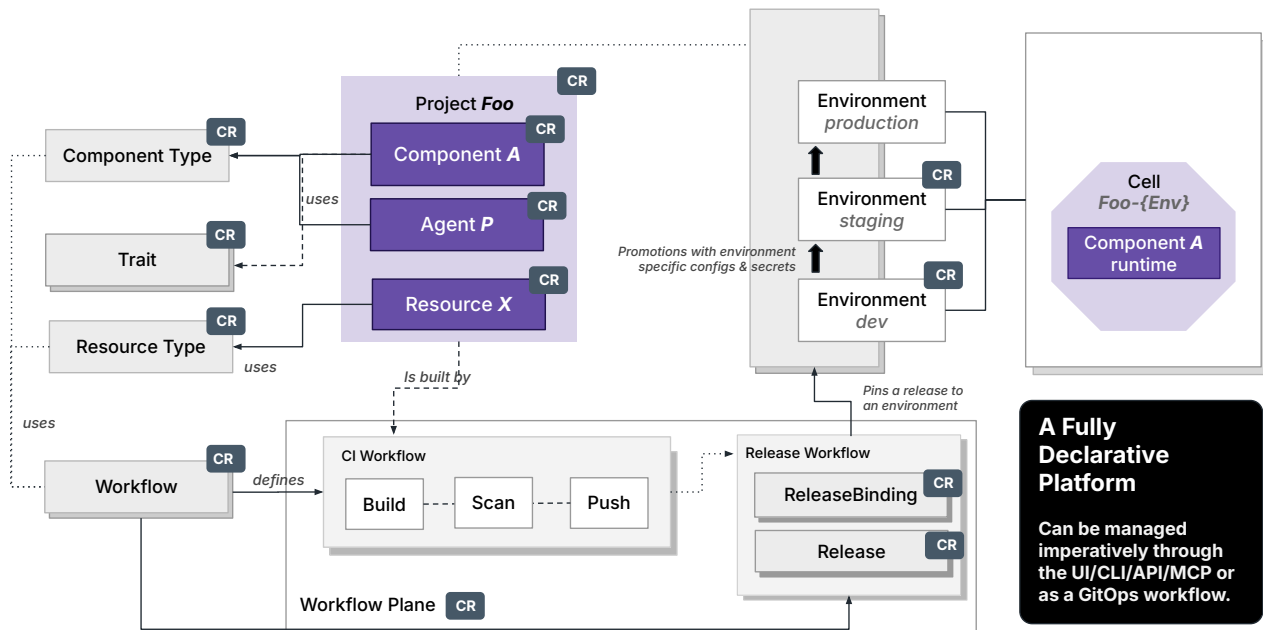


One Set Of Platform Abstractions

Humans and agents use different interfaces, but converge on the same platform abstractions:

- Data plane · Environment · Pipeline
- Project · Component · Resource · Agent
- Workflow · Releases

The platform becomes the common language between humans, agents and infrastructure.



Agentic Doesn't Mean Ungoverned

Identity

Who is this agent?

Least Privilege

What is it allowed to do?

Policy

Under what conditions?

Audit

What exactly did it do?

Observability

What happened as a result?

Isolation

Where can it execute?

AI doesn't remove governance. It makes governance more important.



How OpenChoreo Actually Governs Agents

Agents Get Real Identity

Every agent is provisioned its own identity through the platform's identity provider (Thunder), never a shared credential or borrowed token.

RBAC Extends To MCP Tools

MCP tool calls run through the same Casbin-based RBAC/ABAC engine as human users, with 80+ granular permissions, a tool can only do what the role allows.

Abstraction, Not Raw Infra

Agents call the same platform APIs and golden paths platform engineers defined for humans, never kubectl or cloud credentials directly.

Every Change, Reconciled & Logged

Agent and human actions both flow through the control plane's declarative reconciliation, a full, queryable history of what changed.

OTel Traces Power Agent Evals

Zero-code OpenTelemetry auto-instrumentation feeds the Observability Plane; those same traces drive built-in evaluators for accuracy, safety, reasoning and tool use.

Cells, Sandboxes & Zero Trust

Cell-based network isolation (deny-by-default, gateway-enforced) plus tiered agent sandboxing, runc, gVisor, or per-agent Kata VMs.

This isn't governance bolted onto agents. It's the same control plane humans have always gone through.

Built To Extend: AI & Agent Modules

AI Gateway

Manages LLM providers and model access as a first-class platform capability not a bespoke integration per team.

MCP Gateway

Exposes platform and tool operations to agents over MCP, enforced by the same RBAC every human API call goes through.

Agent Manager

Installs as an extension on top of OpenChoreo, agent lifecycle, identity and evaluation, on top of the same platform

Agent Sandbox Module

Runs every agent as a sandboxed pod, with tiered isolation from runc up to per-agent Kata VMs.

Secrets & Networking

OpenBao for runtime secrets, Cilium for eBPF-powered network policy, both drop in as ecosystem modules.

Evaluation Engine

Continuous agent evals, accuracy, safety, reasoning, tool use, as another pluggable capability, not a one-off script.

OpenChoreo philosophy, new AI capability doesn't mean a new platform. It means one more module on the same platform



SEPT 22 - 24, 2026 | NAIROBI, KENYA

Go from zero to platform, or take what you have further.

One IDP for humans and agents.

Build it once.

Let every participant use it
safely.