SUSE Rancher, OpenChoreo

SUSE Rancher Prime and WSO2 Developer Platform for OpenChoreo

AI-Native Platform Engineering Stack by SUSE and WSO2

SUSE Rancher Prime
WSO2 OpenChoreo

Lakmal Warusawithana, Vice President & Distinguished Engineer (WSO2)
Jivaji Ihare, Partner Solution Architect (SUSE)
Terry Smith, Director of Partner Ecosystem Solution Innovation (SUSE)

SUSE

SUSE Rancher Prime and WSO2 Developer Platform for OpenChoreo

AI-Native Platform Engineering Stack by SUSE and WSO2

Date: 2026-04-03

Summary

This guide presents a reference architecture that combines SUSE® Rancher Prime and the WSO2 Developer Platform for OpenChoreo to build an enterprise internal developer platform. SUSE Rancher manages Kubernetes clusters while OpenChoreo provides developer self-service, CI/CD, observability, governance, AI-native operational agents, a WSO2 AI/API Gateway, and a WSO2 Agent Manager for agent lifecycle management.

Contents

1 Introduction

Organizations increasingly adopt Kubernetes as the foundation for building and operating cloud-native applications. However, Kubernetes by itself does not provide the full developer experience, governance model, or operational automation needed to support enterprise platform engineering. To address this gap, many organizations build internal developer platforms that give developers self-service workflows while allowing platform teams to maintain security, compliance, and operational control. As AI-driven software delivery becomes more common, these platforms must also support AI agents as first-class participants in software delivery and operations. This reference configuration describes a layered solution that combines:

- SUSE® Rancher Prime for enterprise Kubernetes lifecycle management
- WSO2 Developer Platform for OpenChoreo for developer self-service and platform abstractions
- WSO2 AI/API Gateway for secure API and AI traffic mediation
- WSO2 Agent Manager for managing the lifecycle of AI agents
- OpenChoreo Security and OpenChoreo Observability modules for governance and operational visibility

Together, these technologies provide:

- Centralized Kubernetes cluster management
- Self-service application delivery for developers
- GitOps, API, CLI, UI, and MCP-based platform access
- Built-in CI/CD automation
- Integrated observability
- Policy-driven governance
- AI-native operational automation
- Managed lifecycle support for enterprise AI agents

The solution is designed for organizations that want to build a modern internal developer platform on Kubernetes while supporting both human users and AI agents through a governed and extensible platform architecture.

1.1 Scope

This guide provides a reference architecture and deployment guidance for building an internal developer platform using SUSE Rancher Prime and WSO2 Developer Platform for OpenChoreo, bundled with WSO2 AI/API Gateway and WSO2 Agent Manager. The following aspects are covered:

- Business challenge and value delivered by the solution
- Layered solutioOpenChoreo securityn architecture
- Core software and infrastructure components, and their roles and functions
- High-level deployment guidance for a proof-of-concept environment
- Validation of the deployed solution

The guide does not cover:

- Detailed product installation instructions
- Hardware vendor specific sizing and implementation guidance
- Advanced production tuning for very large scale deployments
- Detailed cost modeling or licensing guidance

For detailed, component-specific installation and configuration steps, refer to the provided links to official product documentation.

1.2 Audience

This guide is intended for technical professionals who build, operate, and evolve cloud-native platforms. Typical readers include platform engineers, Kubernetes administrators, DevOps engineers, site reliability engineers, cloud architects, enterprise architects, security architects, and AI platform engineers. These roles are commonly responsible for designing, deploying, and operating internal developer platforms and Kubernetes-based application environments across development, testing, and production systems.

To follow this guide successfully, you should have a basic understanding of Kubernetes concepts and operations, familiarity with containers and cloud-native applications, and experience with CI/CD workflows. You should also have general knowledge of API management, ingress and

networking patterns, and observability concepts such as logs, metrics, and traces. Basic familiarity with AI agents or agent-based workflows is helpful, as this reference architecture includes AI-native capabilities that allow agents to interact with the platform through governed interfaces.

2 Business aspect

Organizations need to improve developer productivity without losing control over security, governance, and operations. In many enterprises, Kubernetes adoption has been successful at the infrastructure level. However, the developer experience still depends on fragmented tools, manual processes, and platform teams that must bridge the gaps between infrastructure, application delivery, and operational support.

At the same time, enterprises are beginning to adopt AI agents to assist with development, operations, architecture review, incident response, and cost optimization. These new agent-driven workflows create additional requirements for security, lifecycle management, observability, and governed access to platform capabilities.

This solution addresses both needs through a layered architecture. SUSE Rancher Prime provides centralized Kubernetes management and operational consistency. WSO2 Developer Platform for OpenChoreo provides a self-service developer platform with platform abstractions, built-in workflows, and AI-native interfaces. WSO2 AI/API Gateway and WSO2 Agent Manager extend the solution to support governed AI interactions and managed agent lifecycles. OpenChoreo Security and OpenChoreo Observability modules provide runtime governance, operational visibility, and integrated platform insights across the environment.

2.1 Business challenge

Key challenges faced by organizations include:

- **Developer productivity and platform complexity**
  Developers often depend on platform teams to provision environments, configure pipelines, expose APIs, and troubleshoot deployments. This slows delivery and creates repeated operational overhead.
- **Fragmented platform tooling**
  Many platform stacks combine separate tools for cluster management, CI/CD, observability, API exposure, and AI integration. Without a consistent platform model, teams must integrate and maintain these tools independently.

- **Difficulty governing AI-native workflows**
  As enterprises adopt AI copilots and agents, they need a secure way to expose platform capabilities to those agents. They also need visibility into how agents operate, what systems they access, and how lifecycles are controlled.

- **Operational burden on SRE and platform teams**
  SREs and platform engineers spend significant time diagnosing issues across multiple clusters, services, APIs, and workflows. These tasks become more complex when AI agents and automation are added to the environment.

- **Need for enterprise-ready Kubernetes operations**
  Organizations need a reliable and repeatable way to manage Kubernetes clusters, enforce operational standards, and support multiple environments across development, testing, and production.

2.2 Business value

With this solution, organizations gain:

- **Unified enterprise platform**
  SUSE Rancher Prime delivers centralized Kubernetes lifecycle management and operational consistency. This gives you a strong foundation for running an internal developer platform across multiple environments.

- **Faster developer self-service**
  WSO2 Developer Platform for OpenChoreo gives developers self-service access to platform capabilities through the portal, API, CLI, GitOps, and MCP interfaces. Developers can deploy and manage workloads without working directly with low-level infrastructure details.

- **Governed AI-native platform operations**
  The solution supports built-in and external AI agents as first-class users of the platform. With MCP-based access, AI agents can safely interact with platform services using structured interfaces instead of ad hoc integrations.

- **Secure AI and API exposure**
  The WSO2 AI/API Gateway provides a controlled gateway layer for APIs and AI-facing interactions. This helps you enforce authentication, policy, mediation, and traffic governance consistently.

- **Managed agent lifecycle**
  The WSO2 Agent Manager adds lifecycle management for AI agents. This supports the onboarding, operation, governance, and evolution of agents that participate in platform workflows.

- **Better operational visibility**
  OpenChoreo observability capabilities provide richer insight into system behavior, application health, and agent-driven operations.

- **Extensible modular architecture**
  The solution is modular by design. You can adopt the full stack or extend selected layers over time as your platform engineering and AI strategy matures.

3 Architectural overview

The solution is illustrated by the diagram below.

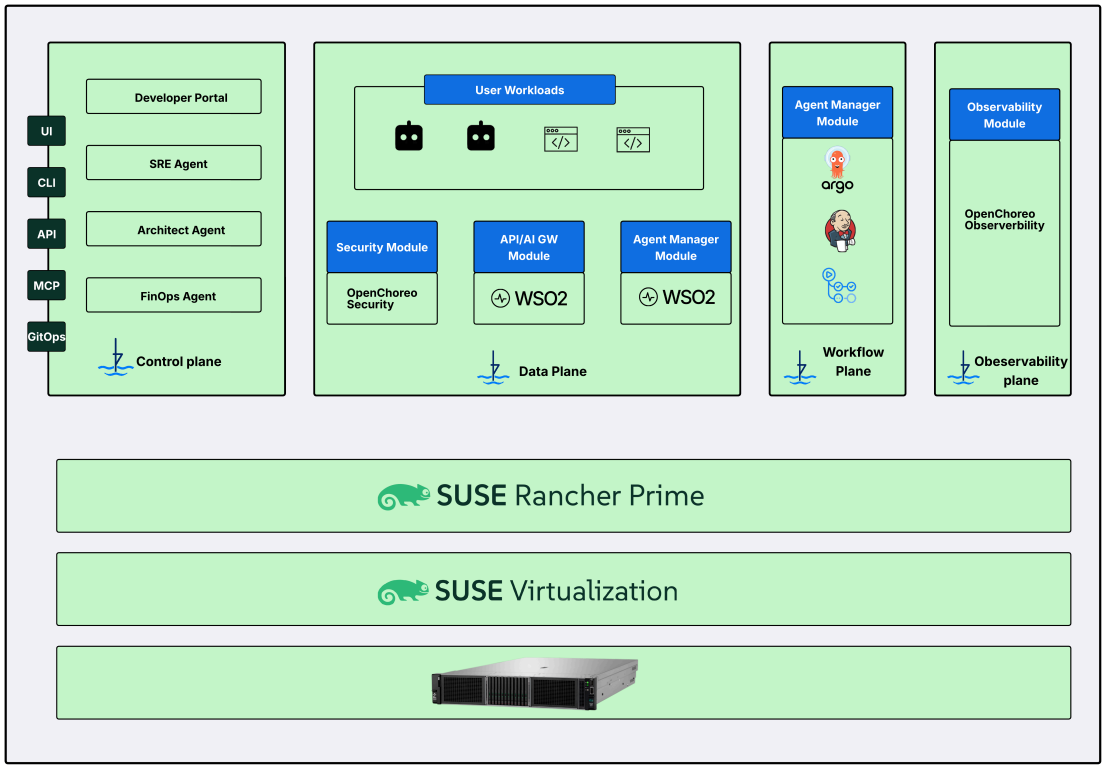

FIGURE 1: SUSE-WSO2 ARCHITECTURE

This is a layered, cloud-native solution that can be deployed in your data center, in public clouds, or anywhere you can run Kubernetes. For on-premises, you can choose SUSE Virtualization to provide a modern, consistent infrastructure foundation that enables you to run virtual machine and container workloads side-by-side.

SUSE Rancher Prime is an enterprise, cloud-native platform offering unified, multi-cluster Kubernetes management. It provides streamlined and consistent operations, centralized security, policy, and user management, along with a secure supply chain, trusted delivery, and enterprise lifecycle and support. In the solution, SUSE Rancher Prime provides essential management for the Kubernetes clusters that host the WSO2 Developer Platform control, data, workflow, and observability planes.

The control plane provides multiple interfaces to platform users and tools, including:

- UI
- CLI
- API
- MCP
- GitOps

The control plane also hosts the Developer Portal (powered by Backstage) and persona-oriented AI agents such as:

- SRE Agent
- Architect Agent
- FinOps Agent

The data plane runs user workloads and platform modules. In this architecture, the data plane includes:

- Security Module based on OpenChoreo capabilities
- Optional WSO2 AI/API Gateway Module
- Optional WSO2 Agent Manager Module

The workflow plane hosts build and automation capabilities, including integration with tools such as Argo and Jenkins. The observability plane hosts observability services, including the OpenChoreo observability module.

Key components of the solution include:

Software

- WSO2 Developer Platform for OpenChoreo – Developer self-service platform and control plane
- optional WSO2 AI/API Gateway – Secure API and AI gateway capabilities
- optional WSO2 Agent Manager – AI agent lifecycle management
- OpenChoreo Security – Security and policy capabilities for the platform
- OpenChoreo Observability – Monitoring and observability support
- Workflow Tooling – Build and delivery automation
- SUSE Rancher Prime – Centralized Kubernetes cluster lifecycle management and governance
- optional SUSE Virtualization - Cloud-native virtualization for consistent infrastructure foundation

Compute Platform

- Underlying infrastructure – Compute, network, and storage resources in on-premises or cloud environments

Each key solution component is described below, starting at the platform layer and moving downward to the compute platform.

3.1 WSO2 Developer Platform for OpenChoreo

WSO2 Developer Platform for OpenChoreo is a Kubernetes-native internal developer platform that enables platform teams to provide self-service workflows while maintaining governance and operational control. It offers a developer portal, declarative platform abstractions, CI/CD integration, environment management, and observability-aware workflows.

In this solution, OpenChoreo acts as the primary developer platform layer. It exposes interfaces through UI, CLI, API, MCP, and GitOps so that both human users and AI agents can interact with the platform.

Key functions include:

- Developer self-service
- Environment and workload management
- Application lifecycle workflows
- Policy-driven platform abstractions
- AI-native interaction through MCP and agent integrations

Recommended version: Latest WSO2 OpenChoreo release (https://wso2.com/engineering-platform/openchoreo/)

3.2 WSO2 AI/API Gateway

The WSO2 AI/API Gateway provides a managed gateway layer for APIs and AI-related traffic. It can mediate, secure, and govern how services and agents interact with platform capabilities and user workloads.

In this architecture, the gateway runs as a dedicated module in the data plane. It helps centralize access control and policy enforcement for service-to-service, user-to-service, and agent-to-platform interactions.

Key functions include:

- API exposure and governance
- Authentication and authorization enforcement
- Traffic mediation
- Policy-driven AI and API access control
- Secure connectivity between consumers and workloads

Recommended version: Latest supported version
Website: WSO2 API Gateway (https://wso2.com/api-platform/)

3.3 WSO2 Agent Manager

The WSO2 Agent Manager provides lifecycle management for AI agents. It helps organizations manage how agents are onboarded, configured, governed, and operated across the platform.

In this solution, Agent Manager complements OpenChoreo's AI-native model by managing agents that participate in development and operations workflows, including SRE, architecture, and FinOps use cases.

Key functions include:

- Agent registration and lifecycle control
- Agent configuration and governance
- Management of agent interactions with platform services
- Support for enterprise operational workflows driven by AI agents

Recommended version: Latest supported version
Website: WSO2 Agent Platform (https://openchoreo.dev/modules/)

3.4 OpenChoreo Security

The OpenChoreo Security provides security capabilities for the developer platform environment across all planes. In this architecture, it is positioned in the data plane to provide policy and security controls alongside application and agent-facing services.

Key functions include:

- Security policy enforcement
- Runtime security integration
- Platform guardrails for workloads and services

Recommended version: Latest supported version
Website: OpenChoreo Security (https://openchoreo.dev/)

3.5 OpenChoreo Observability plane

The OpenChoreo Observability plane provides visibility into applications, platform services, and supporting components. In this architecture, it operates in the observability plane and supports platform teams, SREs, and AI-assisted workflows.

Key functions include:

- Metrics collection
- Logging and tracing support
- Platform and workload visibility
- Operational insight for human users and AI agents

Recommended version: Latest supported version
Website: OpenChoreo Observability plane (https://openchoreo.dev/)

3.6 OpenChoreo Workflow Plane

The workflow plane supports build and delivery automation. The diagram shows tools such as Argo and Jenkins, which can be used to implement build pipelines, deployment automation, and workflow execution required by the platform.

Key functions include:

- Build automation
- Delivery workflows
- Integration with platform CI/CD processes

Website: OpenChoreo Workflow Plane (https://openchoreo.dev/)

3.7 SUSE Rancher Prime

SUSE Rancher Prime is an enterprise, cloud-native platform offering unified, multi-cluster Kubernetes management. It provides streamlined and consistent operations, centralized security, policy, and user management, along with a secure supply chain, trusted delivery, and enterprise lifecycle and support.

A Kubernetes cluster, managed by SUSE Rancher Prime, hosts the control plane, data plane, workflow plane, observability plane, and user workloads. A conformant Kubernetes distribution, such as RKE2, is recommended.

SUSE Rancher provides the cluster management foundation for the entire solution, including all platform and workload services.

Key functions include:

- Cluster lifecycle management
- Access control
- Multi-cluster governance
- Policy enforcement
- Operational consistency across environments

Recommended version: Latest supported version
Website: SUSE Rancher Prime (https://www.suse.com/products/rancher/)
Documentation: SUSE Rancher Prime Documentation (https://documentation.suse.com/cloudnative/rancher-manager/)

3.8 SUSE Virtualization

SUSE Virtualization provides a modern, cloud-native virtualization platform that enables organizations to run Kubernetes clusters on top of bare-metal infrastructure with a consistent and supported foundation. It is recommended for on-premise deployments where virtualization is required to host the underlying infrastructure.

For cloud or hosted environments, SUSE Virtualization is optional, and the SUSE Rancher Prime and WSO2 solution can run on any supported infrastructure or public cloud provider.

Recommended version: SUSE Virtualization latest supported version (https://www.suse.com/suse-harvester/support-matrix/all-supported-versions/)
Documentation: SUSE Virtualization Documentation (https://documentation.suse.com/cloudnative/virtualization/)

4 Deployment overview

The solution is designed to be flexible and be deployed in on-premises environments, private cloud platforms, and supported public cloud providers. This allows organizations to implement the platform according to their operational, regulatory, and cost requirements.

For a proof-of-concept deployment, you should prepare infrastructure with sufficient compute, storage, and networking resources for the control plane, data plane, workflow plane, and observability components in a Kubernetes cluster managed by SUSE Rancher Prime.

Before installing the platform components, ensure that you have met the following prerequisites:

- A supported Kubernetes cluster, such as RKE2, managed by SUSE Rancher Prime
- Persistent storage available for platform services and workloads
- Ingress and load balancing configured for external access
- DNS resolution for platform endpoints
- Access to container image registries
- Network connectivity between control plane, data plane, and workflow components

You can deploy the solution on infrastructure hosted in your own data center or on a cloud provider. However, the Kubernetes environment must be supported by SUSE Rancher and meet the resource requirements of the platform components.

The software components provide the foundational layer for all higher-level services. For a proof-of-concept deployment, you should prepare:

- Kubernetes worker and control-plane capacity appropriate for platform services and sample workloads
- Ingress and DNS support
- Persistent storage
- Cluster networking
- Access to image registries
- Identity and access integration as required

4.1 SUSE Rancher Prime

To install SUSE Rancher Prime follow below high-level steps:

1. Install SUSE Rancher Prime.
2. Create or import the target Kubernetes cluster.
3. Configure cluster access, authentication, and governance.
4. Confirm that the cluster is ready to host platform services.

Detailed guidance for various deployment scenarios are available in SUSE Rancher: Installation and upgrade (https://documentation.suse.com/cloudnative/rancher-manager/latest/en/installation-and-upgrade/installation-and-upgrade.html)

4.2 WSO2 Developer Platform for OpenChoreo

Deploy the WSO2 Developer Platform for OpenChoreo into the Rancher-managed Kubernetes cluster by following the installation procedure described in the OpenChoreo documentation. OpenChoreo can run on any conformant Kubernetes cluster, including clusters provisioned and managed by SUSE Rancher Prime, and can be deployed on on-premises infrastructure or supported cloud providers.

The complete WSO2 OpenChoreo[installation guide] is available.

The following high-level steps summarize the installation flow. Refer to the linked documentation for the detailed commands and configuration values.

1. Ensure that the target cluster is accessible using kubectl and Helm. Refer to WSO2 OpenChoreo environment prerequisites (https://openchoreo.dev/docs/getting-started/try-it-out/on-your-environment/#prerequisites).
2. Install required dependencies such as ingress, certificate management, and storage support. Refer to WSO2 OpenChoreo installation prerequisites (https://openchoreo.dev/docs/getting-started/try-it-out/on-your-environment/#install-prerequisites).
3. Deploy the OpenChoreo control plane components into the Kubernetes cluster. Refer to Steps to install OpenChoreo (https://openchoreo.dev/docs/getting-started/try-it-out/on-your-environment/#install-openchoreo).
4. Confirm that all platform services are running and accessible. Refer to Verify installation (https://openchoreo.dev/docs/getting-started/try-it-out/on-your-environment/#verify-installation).
5. Open the portal endpoint and confirm that the platform UI is available. Refer to access the OpenChoreo Portal (https://openchoreo.dev/docs/getting-started/try-it-out/on-your-environment/#access-the-portal).
6. Deploy a sample workload to validate the build, deploy, and runtime workflow. Refer to Deploy a sample application (https://openchoreo.dev/docs/getting-started/try-it-out/on-your-environment/#deploy-a-sample).

7. Confirm that the workload runs in the data plane and that logs, metrics, and traces are available. Refer to Verify deployment (https://openchoreo.dev/docs/getting-started/try-it-out/on-your-environment/#verify).

8. Enable additional modules.

After the base platform is installed, optional modules such as the AI/API Gateway, Agent Manager, workflow integrations, and observability extensions can be installed according to the solution architecture.

Refer to the OpenChoreo documentation for supported deployment models, configuration options, and production recommendations.

4.3 WSO2 AI/API Gateway

Deploy the WSO2 AI/API Gateway module into the data plane by following the installation procedure provided in the OpenChoreo community modules repository. The gateway module is installed as an OpenChoreo module and integrated with the platform using the standard module integration mechanism.

The complete installation guide for the gateway module is available. see the refference documentation (https://github.com/openchoreo/community-modules/tree/main/gateway-wso2-api-platform)

The following high-level steps summarize the installation flow. Refer to the linked documentation for the exact commands and configuration values.

1. Confirm that the OpenChoreo data plane is installed and that the Kubernetes cluster is accessible using kubectl and Helm.

2. Install the WSO2 API Platform gateway module using the Helm chart provided in the module repository.

3. Provide the required configuration values for routing, security policies, and service exposure according to the module documentation.

4. Ensure the module is registered with the OpenChoreo data plane so that the control plane can create Gateway and routing resources.

5. Confirm that the gateway controller and related services are running in the Kubernetes cluster.

6. Deploy a sample service and confirm that requests are routed through the gateway and enforced by configured policies.

7. Additional features such as policy enforcement, AI traffic mediation, and API governance can be configured using the gateway settings supported by the module.

Refer to the module documentation (https://github.com/openchoreo/community-modules/tree/main/gateway-wso2-api-platform) for supported configuration options and production deployment recommendations.

4.4 WSO2 Agent Manager and built-in agents

Configure AI and agent capabilities for the platform by enabling Model Context Protocol (MCP) support, configuring the WSO2 Agent Manager module, and registering platform agents such as the SRE agent. These features allow AI agents to interact with the platform through governed interfaces while maintaining security, observability, and lifecycle control.

1. Configure Agent Manager
   Install and configure the WSO2 Agent Manager module to manage the lifecycle of AI agents used by the platform. Agent Manager is responsible for registering agents, managing agent configuration, and controlling how agents interact with platform services. Ensure that the Agent Manager module is deployed in the data plane and integrated with the OpenChoreo control plane. Refer to WSO2 Agent Manager: configurations (https://openchoreo.dev/modules/)

2. Configure MCP integration

   a. Enable Model Context Protocol (MCP) support so that AI agents and external tools can interact with the platform through governed interfaces.

   b. MCP allows agents to access platform APIs, observability data, and deployment workflows using structured requests instead of direct cluster access.
      Refer to WSO2 Agent Manager: Configure MCP integration (https://openchoreo.dev/docs/next/ai/overview/#mcp-servers)

   c. Verify that MCP endpoints are reachable and that authentication and authorization policies are correctly configured.

3. Configure SRE agent.

a. Configure the SRE agent to enable AI-assisted operational workflows such as incident investigation, observability analysis, and runtime diagnostics.

b. The SRE agent uses MCP and observability data to analyze logs, metrics, and traces and provide operational insights to platform engineers.
Refer to WSO2 Agent Manager: Configure SRE agent (https://openchoreo.dev/docs/next/ai/overview/#sre-agent)

4.5 Workflow and observability integrations

Integrate workflow tooling and observability services with the platform by following the OpenChoreo workflow documentation and the community module installation guides. Workflow capabilities are configured through the OpenChoreo workflow feature, and observability services are installed as platform modules in the data plane.

The following high-level steps summarize the integration flow. Refer to the linked documentation for the exact commands and configuration values.

1. Enable workflow capabilities and create workflows using the OpenChoreo workflow feature (https://openchoreo.dev/docs/user-guide/workflows/creating-workflows/).
2. Install the OpenSearch logs (https://github.com/openchoreo/community-modules/tree/main/observability-logs-opensearch) module using the provided Helm chart and module configuration.
3. Install the tracing module (https://github.com/openchoreo/community-modules/tree/main/observability-tracing-opensearch) based on OpenSearch and configure it for the data plane.
4. Install the Prometheus metrics (https://github.com/openchoreo/community-modules/tree/main/observability-metrics-prometheus) module and configure metrics collection for platform and workloads.
5. Confirm that logs, metrics, and traces are collected for platform services, workloads, and agent operations.
6. Create and run a workflow to verify that build, deployment, or automation tasks execute successfully.

For production deployments, refer to the official documentation for sizing, security hardening, high availability, backup, and disaster recovery guidance for workflow tooling and observability modules.

5 Validation

Validate successful deployment of the solution using the following checks.

1. Confirm that SUSE Rancher shows the target Kubernetes cluster in a healthy state.
2. Confirm that OpenChoreo control plane services are running.
3. Access the developer portal successfully.
4. Verify UI, CLI, API, MCP, and GitOps access paths as applicable.
5. Deploy a sample application or component using one of the examples from the OpenChoreo example catalog (https://openchoreo.dev/docs/learn-from-examples/examples-catalog/).
6. Route traffic through the WSO2 AI/API Gateway.
7. Register or activate a sample agent through WSO2 Agent Manager.
8. Confirm logs, metrics, and traces are visible in the observability stack.
9. Verify that a sample operational workflow can be performed by a human user and, where applicable, by an AI agent.

Successful validation demonstrates that the platform supports developer self-service, governed API access, observability, and managed AI-agent participation.

6 Summary

This reference configuration describes a layered architecture for building a secure, AI-native, internal developer platform on Kubernetes. It features SUSE Rancher Prime to provide the operational foundation for Kubernetes. This is combined with WSO2 OpenChoreo, providing the developer platform layer and WSO2 AI/API Gateway and WSO2 Agent Manager to extend the platform to support governed API exposure and managed AI agent lifecycles. Additionally, OpenChoreo Security and observability modules strengthen runtime governance and operational insight.

As next steps, you can evaluate production sizing, high availability, enterprise security integration, explore the OpenChoreo Platform Engineering Guide (https://openchoreo.dev/docs/next/category/platform-engineer-guide/), and additional agent-driven workflows based on your platform strategy.

7 Frequently Asked Questions (FAQ)

1. **What role does SUSE Rancher Prime play in this solution?**
   SUSE Rancher Prime provides enterprise Kubernetes lifecycle management, cluster governance, and a consistent operational foundation for the solution.

2. **What does WSO2 Developer Platform for OpenChoreo add on top of Kubernetes?**
   OpenChoreo adds the developer platform layer, including self-service workflows, developer portal capabilities, workload abstractions, CI/CD integration, environment management, and AI-native interaction models.

3. **Why are WSO2 AI/API Gateway and WSO2 Agent Manager bundled in this architecture?**
   The WSO2 AI/API Gateway provides secure and governed API and AI-facing traffic management. The WSO2 Agent Manager manages the lifecycle of AI agents that operate within the platform.

4. **How do AI agents interact with the platform?**
   Agents can interact with the platform through governed interfaces such as MCP (https://openchoreo.dev/docs/next/category/working-with-ai/) and API-based integrations. This allows them to participate in workflows while remaining subject to platform policies and lifecycle controls.

5. **Is this solution limited to developer workflows?**
   No. The architecture supports developers, platform engineers, SREs, architects, FinOps users, and AI agents that assist these personas.

6. **Can this architecture be extended?**
   Yes. The architecture is modular and can be extended with additional workflow, security, observability, and AI capabilities over time.

7. **Why prefer RKE2 as the underlying Kubernetes engine for the clusters?**

RKE2 (Rancher Kubernetes Engine 2) is SUSE's next-generation, fully conformant Kubernetes distribution. It is our preferred engine because it is built from the ground up for enterprise security and compliance. Unlike older engines, it entirely removes the Docker dependency (using containerd instead) and provides a hardened, secure-by-default environment essential for running mission-critical WSO2 API Gateways and Agent Modules.

8. **How does RKE2 improve the security posture of the Data and Control planes?**
   Security is built into RKE2's default configuration. It is designed to pass the CIS Kubernetes Benchmark out of the box with minimal configuration. Furthermore, it supports FIPS 140-2 compliance and adheres to DISA STIG requirements. This ensures that the WSO2 Security Modules and API/AI Gateways operate on a defense-in-depth foundation.

9. **How does RKE2 integrate with the "SUSE Rancher Prime" layer shown in the architecture?**
   RKE2 is a first-class citizen within the SUSE Rancher Prime ecosystem. Choosing RKE2 ensures seamless, automated cluster lifecycle management (provisioning, upgrading, and scaling) directly from the SUSE Rancher UI or via GitOps workflows. This native integration reduces operational overhead for the SRE and FinOps Agents managing the control plane.

8 Legal notice

9 GNU Free Documentation License

0. PREAMBLE

The purpose of this License is to make a manual, textbook, or other functional and useful document "free" in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondarily, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of "copyleft", which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The "Document", below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as "you". You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A "Modified Version" of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A "Secondary Section" is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The "Invariant Sections" are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The "Cover Texts" are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A "Transparent" copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not "Transparent" is called "Opaque".

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The "Title Page" means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, "Title Page" means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text.

A section "Entitled XYZ" means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as "Acknowledgements", "Dedications", "Endorsements", or "History".) To "Preserve the Title" of such a section when you modify the Document means that it remains a section "Entitled XYZ" according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

2. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

3. COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.

B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.

C. State on the Title page the name of the publisher of the Modified Version, as the publisher.

D. Preserve all the copyright notices of the Document.

E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.

F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.

G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.

H. Include an unaltered copy of this License.

I. Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.

J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.

K. For any section Entitled "Acknowledgements" or "Dedications", Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.

L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.

M. Delete any section Entitled "Endorsements". Such a section may not be included in the Modified Version.

N. Do not retitle any existing section to be Entitled "Endorsements" or to conflict in title with any Invariant Section.

O. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section Entitled "Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties—for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

5. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled "History" in the various original documents, forming one section Entitled "History"; likewise combine any sections Entitled "Acknowledgements", and any sections Entitled "Dedications". You must delete all sections Entitled "Endorsements".

6. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

7. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an "aggregate" if the copyright resulting from the compilation is not used to limit the legal rights of the compilation's users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document's Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

8. TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all

Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled "Acknowledgements", "Dedications", or "History", the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

9. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided for under this License. Any other attempt to copy, modify, sublicense or distribute the Document is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

10. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See http://www.gnu.org/copyleft/.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License "or any later version" applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation.

ADDENDUM: How to use this License for your documents

```
Copyright (c) YEAR YOUR NAME.
   Permission is granted to copy, distribute and/or modify this document
   under the terms of the GNU Free Documentation License, Version 1.2
```

```
   or any later version published by the Free Software Foundation;
   with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts.
   A copy of the license is included in the section entitled “GNU
   Free Documentation License”.
```

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts, replace the “ with…Texts.” line with this:

```
with the Invariant Sections being LIST THEIR TITLES, with the
   Front-Cover Texts being LIST, and with the Back-Cover Texts being LIST.
```

If you have Invariant Sections without Cover Texts, or some other combination of the three, merge those two alternatives to suit the situation.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.