20TH ANNIVERSARY EDITION

WSO2CON ASIA

PLATFORMLESS MODERNIZATION

Control Ingress and Egress Across Multiple API Gateways with WSO2 API Manager



20TH ANNIVERSARY EDITION

WSO2CON ASIA

PLATFORMLESS MODERNIZATION



Malintha Amarasinghe
Senior Technical Lead
WSO2



Krishan Wijesena
Technical Lead
WSO2



Introduction & Lab Overview

Introduction & Lab Overview

- **Objective of the Lab**
  - Understand the need for unified API management by exploring the challenges of managing ingress and egress across multiple API gateways.
  - Learn how to integrate decentralized API gateways into a single control plane for consistent security, governance, and observability
  - Future of Ingress and Egress API Management
- **Use Case Overview**
  - Demonstrate features in WSO2 API Manager to achieve a Unified API Management Experience.
  - Show how Centralized secret/token management by Egress API Gateway
  - The role of Egress API Gateway in AI APIs

The Growing Complexity of API Ecosystems

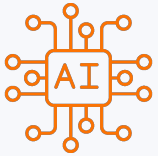
APIs Are Everywhere, Organizations Struggle to Scale



Different Security & Governance compliance of different teams.



Multi-Cloud and Hybrid Complexity



Egress Complexity Amplified by AI and LLM Integrations



To comply with various protocols and platforms



These challenges make us rethink the requirements–

Should we centralize for control, or decentralize for flexibility?

Centralized

- ⊕ Single Control over API
- ⊕ Central Governance & Security
- ⊕ Central Traffic Routing
- ⊖ Slow onboarding and flexibility
- ⊖ Single point of failure
- ⊖ Scalability & Operational issues

Decentralized

- ⊕ Independent Control for Teams
- ⊕ Team-Specific Policies
- ⊕ Flexible Processes
- ⊖ Lack of Standardization & Security
- ⊖ Observability & Monitoring
- ⊖ Increased Operational Overhead



What if there is a middle path ?

What if we could strike the right balance between control and flexibility?

Unified Control Plane with Federated Gateways

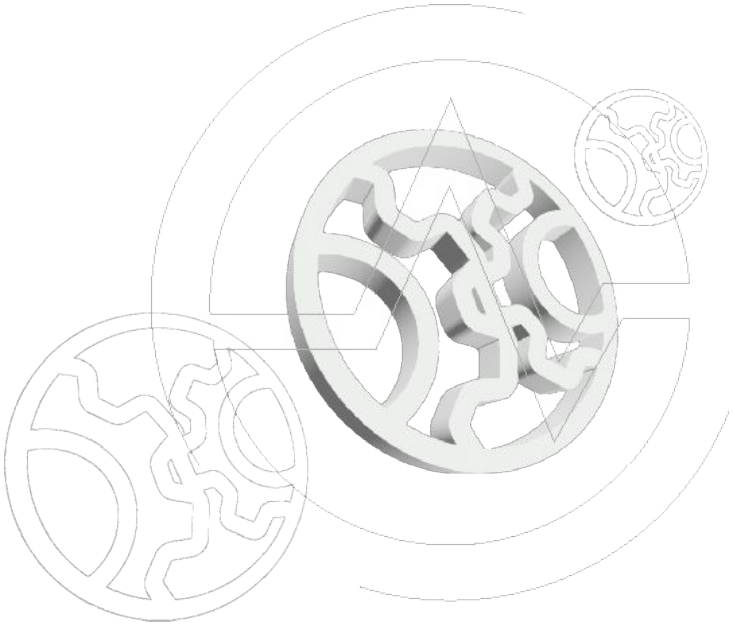
A Unified Control Plane for API Gateways

What is an API Control Plane (ACP)?

A centralized system to manage multiple API gateways, gateway deployments and enforce policies

Key Capabilities of ACP:

- Unified ingress and egress management
- Governance across all environments
- Observability and analytics
- Policy Management – Security, rate limits, authentication, Mediation
- API Discovery – Centralized registry for all APIs.



Gateway Federation and Multi-Gateway Management

Gateway Federation allows multiple API gateways to work together under a shared governance model while maintaining local autonomy.



Key Capabilities of Gateway Federation:

- Across each team, region, or business unit
- Configure, operate, and optimize its own API gateway to meet its specific needs
- Complying with governance and policies centrally.

One product – open source or SaaS

WSO2 API Manager

WSO2 API Control Plane

WSO2 Kubernetes Gateway

K8s-native design, lightweight, Envoy based

WSO2 Immutable Gateway

Offline mode, immutability, edge gateway

WSO2 Universal Gateway

Inbuilt mediation, range of protocols

Third-Party Gateways

AWS API Gateway, Solace, and more

Bijira by WSO2

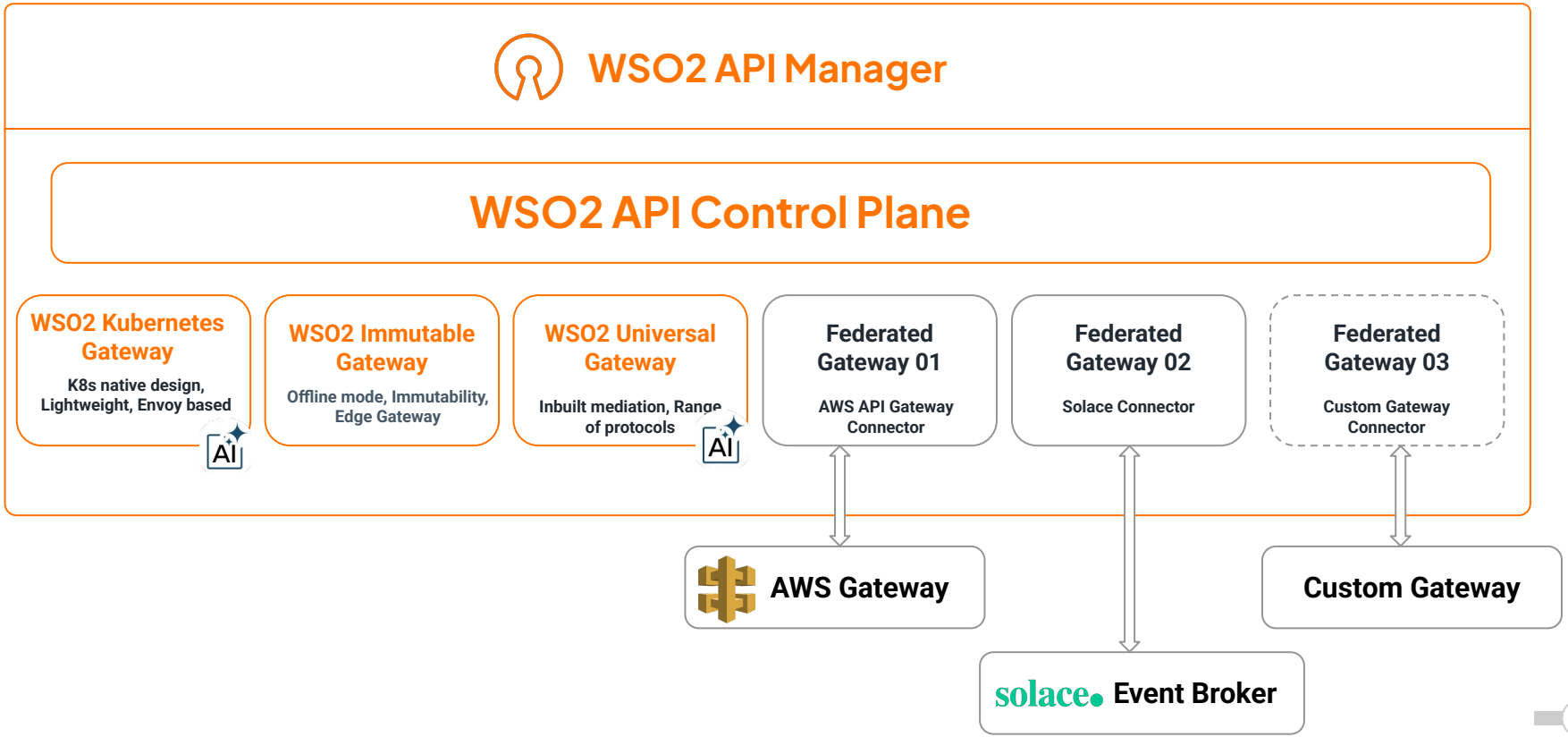
AI-native, fully-powered SaaS API and MCP Server management

Powered by WSO2 API Manager

Built on Choreo

AI: Design / Testing / Search / Governance

How We Extend Gateway Support: Connector Framework





Demo 01

Behavior of the Unified API Control Plane (ACP) with Federated Gateways

The Scenario

Company: ForBizTrust, A fast-growing financial and banking institution with services operating across multiple clouds and regions.

Challenge: They have different API gateways across AWS, and on-prem data centers. Managing policies, security, and traffic is fragmented.



ForBizTrust Financial and Banking Institution

Corporate & SME Service Team	Treasury & Risk Management Team	Wealth Management & Investment Service Team
Requires diverse protocol and mediation support across 2 regions	Service run on AWS; need a GW closer to services	Services run on K8s; need a K8s-native API gateway.	Legacy gateway required until migration
WSO2 Universal Gateway Europe APAC	AWS Gateway	WSO2 Kubernetes Gateway	Custom Gateway

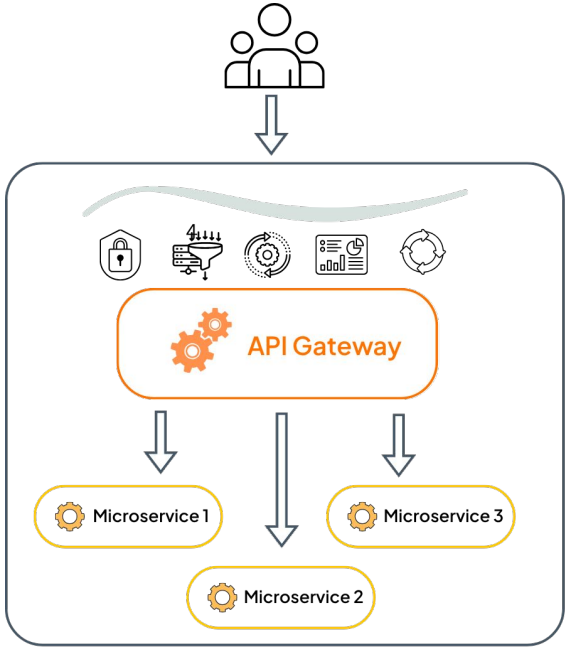
WSO2 API Control Plane

Ingress vs Egress API Management

Ingress API Management

APIs or requests coming **into** your system from external sources

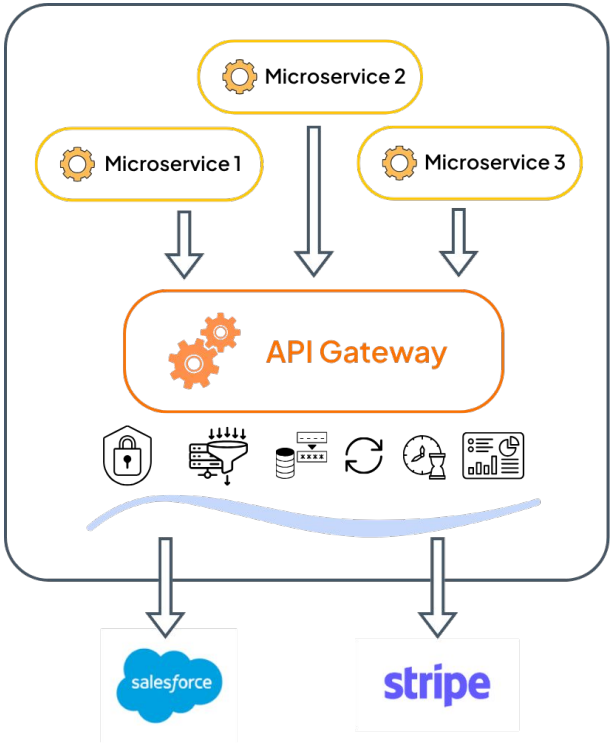
- Ingress API Management focuses on:
  - Securing external access to your APIs
  - Enforcing rate limiting, authentication, throttling
  - Applying request/response transformation policies
  - Logging, auditing, and monitoring incoming traffic
  - API versioning and lifecycle management
- Example: An external mobile app calling your **/payments** API.



Egress API Management

APIs or requests going **out** of your system to external services

- Egress API Management focuses on:
  - Securing and controlling outbound calls
  - Managing credentials (e.g., API keys, OAuth2 tokens)
  - Observability into third-party dependencies
  - Filtering/masking sensitive data in outbound requests
  - Applying resilience patterns (retries, timeouts, circuit breakers)
- Example: Your internal system calling Stripe or OpenAI's API.



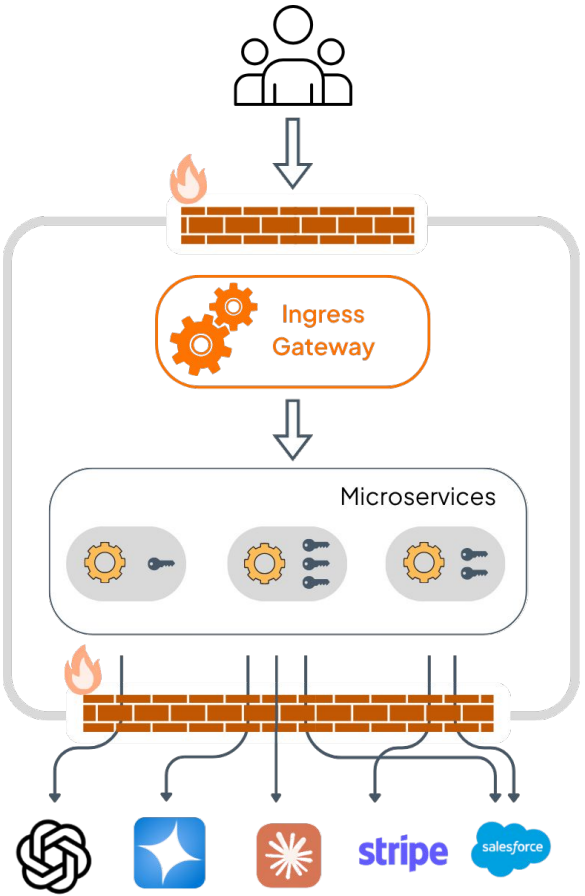


Everyone secures Ingress —

but Egress is often ignored until something **breaks**

Without Egress API Management

- Credential Leakage & Insecure Calls
- No Control Over External Dependencies
- Compliance Violations
- Unregulated use of paid services
- Poor Reliability
- Lack of Visibility

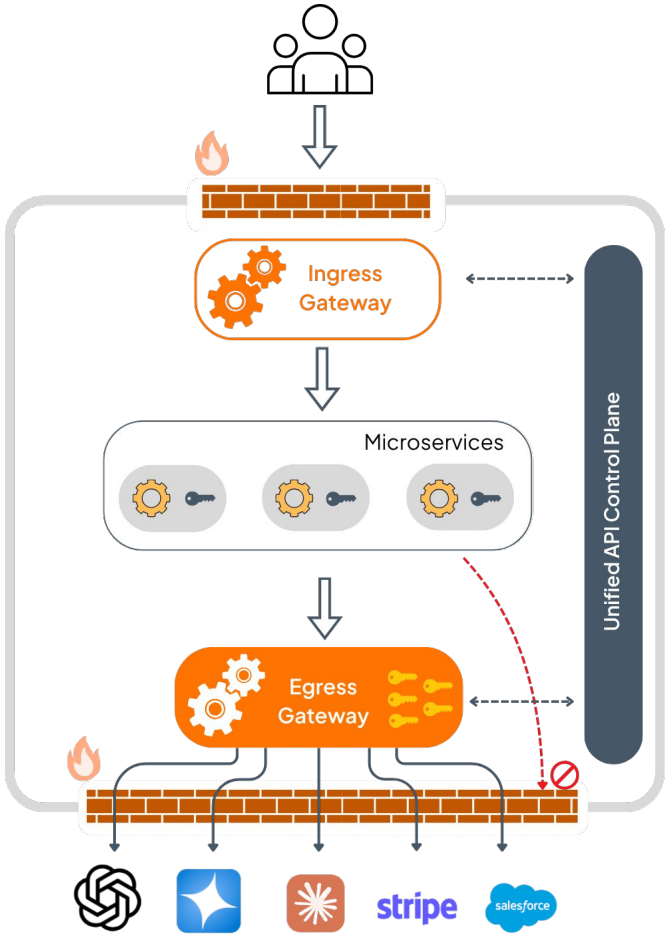


How Egress Gateway Solves This

FIXED
- Credential Leakage & Insecure Calls
  - Centralized secret/token management, encryption

FIXED
- No Control Over External Dependencies
  - Block unapproved domains, and apply egress policies

FIXED
- Compliance Violations
  - Enforce data masking, redact PIIs



How Egress Gateway Solves This

FIXED

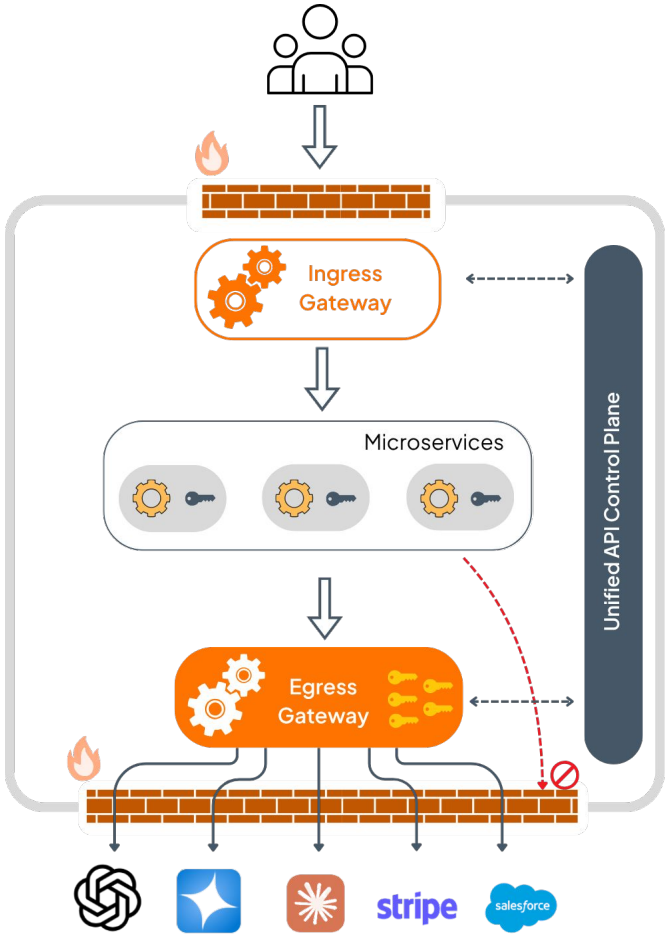
- Unregulated use of paid services
  - Do rate limits, cost monitoring on 3rd party API usage

FIXED

- Poor Reliability
  - Introduce retries, timeouts, and circuit breakers

FIXED

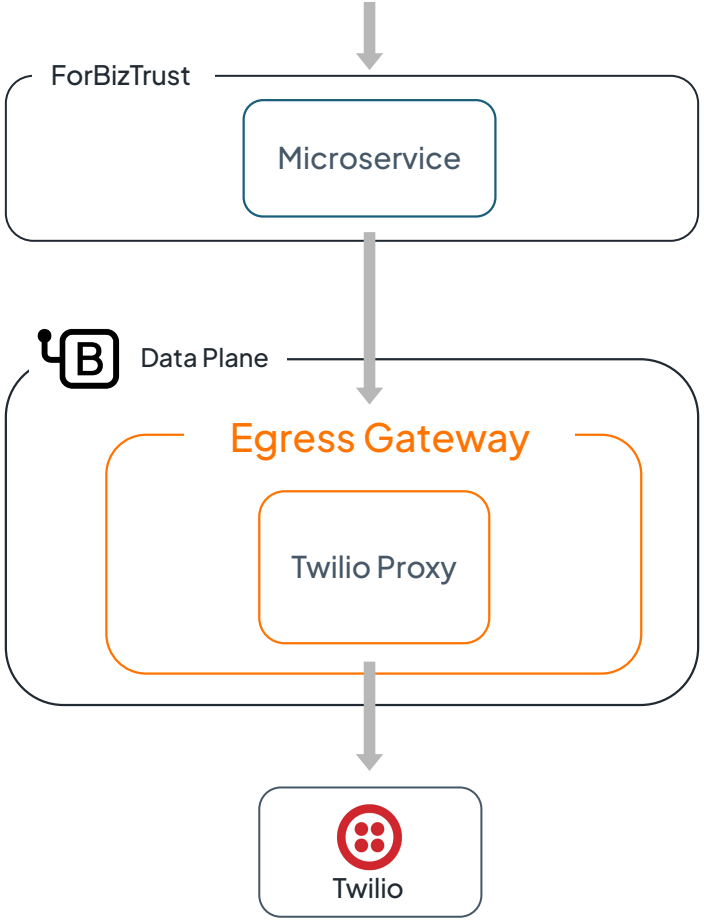
- Lack of Visibility
  - Enable logging, and insights for all outbound API calls





Demo 02

Centralized secret/token management by Egress API Gateway


ForBizTrust
Microservice
Data Plane
Egress Gateway
Twilio Proxy
Twilio



Why Egress API Management is More Important Than Ever

With the rise of **AI and LLM APIs**, almost every modern application now includes some form of AI functionality — often starting right from the **design phase**.

🧠 These apps heavily rely on external AI services like **OpenAI, Azure AI, and Anthropic**

Roles of an Egress Gateway for AI APIs

- **Guardrails & Data Policies** – Redact PII, filter toxic/forbidden content, validate prompts/responses.
- **Token-Based Cost & Rate Limiting** – Enforce quotas using prompt + completion tokens to keep spend in check.
- **Semantic Caching** – Return cached answers for similar prompts to cut latency and cost.
- **Intelligent Model Routing & Failover** – Choose the best model/provider dynamically (cost, latency, capability) with automatic fallback.
- **Track API/Token Usage statistics** – Every outbound call, parse token counts from request/response - Aggregate & Visualize

Egress API Management: WSO2 AI Gateway

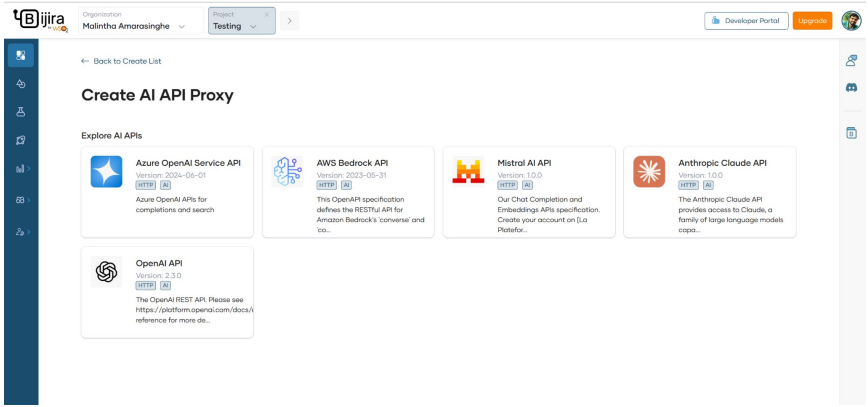
WSO2 AI Gateway governs, optimizes, and secures Generative AI API usage with flexible deployment and multi-provider support for seamless AI integration.

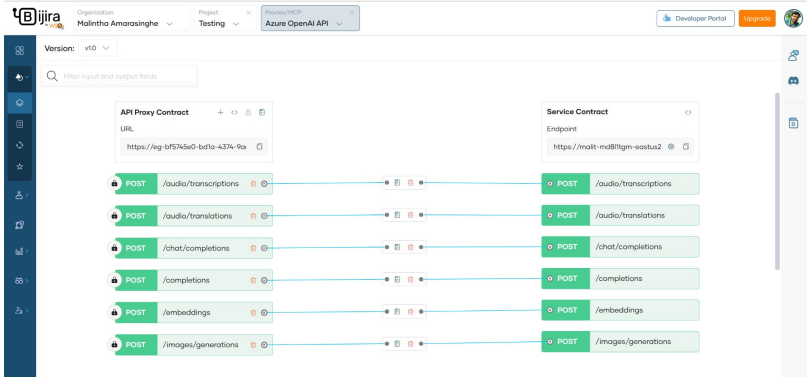
- Multi-provider support
- Token-based rate limiting
- Adaptive request handling
- AI-specific analytics
- AI guardrails
  - 3 types: Regex based, AI Inbuilt, 3rd Party
  - PII, Content Safety, and more
- Semantic Caching

WSO2 Kubernetes Gateway

WSO2 Universal Gateway

Bijira by WSO2


Organization
Malintha Amarasinghe
Project
Testing
Developer Portal
Upgrade
Back to Create List
Create AI API Proxy
Explore AI APIs
Azure OpenAI Service API
Version: 2024-06-01
HTTP
AI
Azure OpenAI APIs for completions and search
AWS Bedrock API
Version: 2023-05-31
HTTP
AI
This OpenAPI specification defines the RESTful API for Amazon Bedrock's 'converse' and 'co...
Mistral AI API
Version: 1.0.0
HTTP
AI
Our Chat Completion and Embeddings APIs specification. Create your account on [La Platefor...
Anthropic Claude API
Version: 1.0.0
HTTP
AI
The Anthropic Claude API provides access to Claude, a family of large language models capa...
OpenAI API
Version: 2.3.0
HTTP
AI
The OpenAI REST API. Please see https://platform.openai.com/docs/a reference for more de...


Organization
Malintha Amarasinghe
Project
Testing
Azure OpenAI API
Developer Portal
Upgrade
Version: v1.0
Filter input and output fields
API Proxy Contract
URL
https://eg-bf5745e0-bd1a-4374-9a
Service Contract
Endpoint
https://malit-md81ltgm-eastus2
POST /audio/transcriptions
POST /audio/translations
POST /chat/completions
POST /completions
POST /embeddings
POST /images/generations


Dashboards
Egress Overview
Full screen
Share
Duplicate
Reset
Edit
Filter your data using KQL syntax
Last 6 days
Refresh
Navigation Bar
Back
Egress Overview
Egress Traffic
Egress Errors
Egress Latency
2,807
Total Token Usage
762
Prompt Token Usage
2,045
Completion Token Usage
AI Token Usage
Hour
Day per day
20 - 236.4
236.4 - 452.8
452.8 - 669.2
669.2 - 885.6
> 885.6



Demo 03

The role of Egress API Gateway in AI/LLM APIs

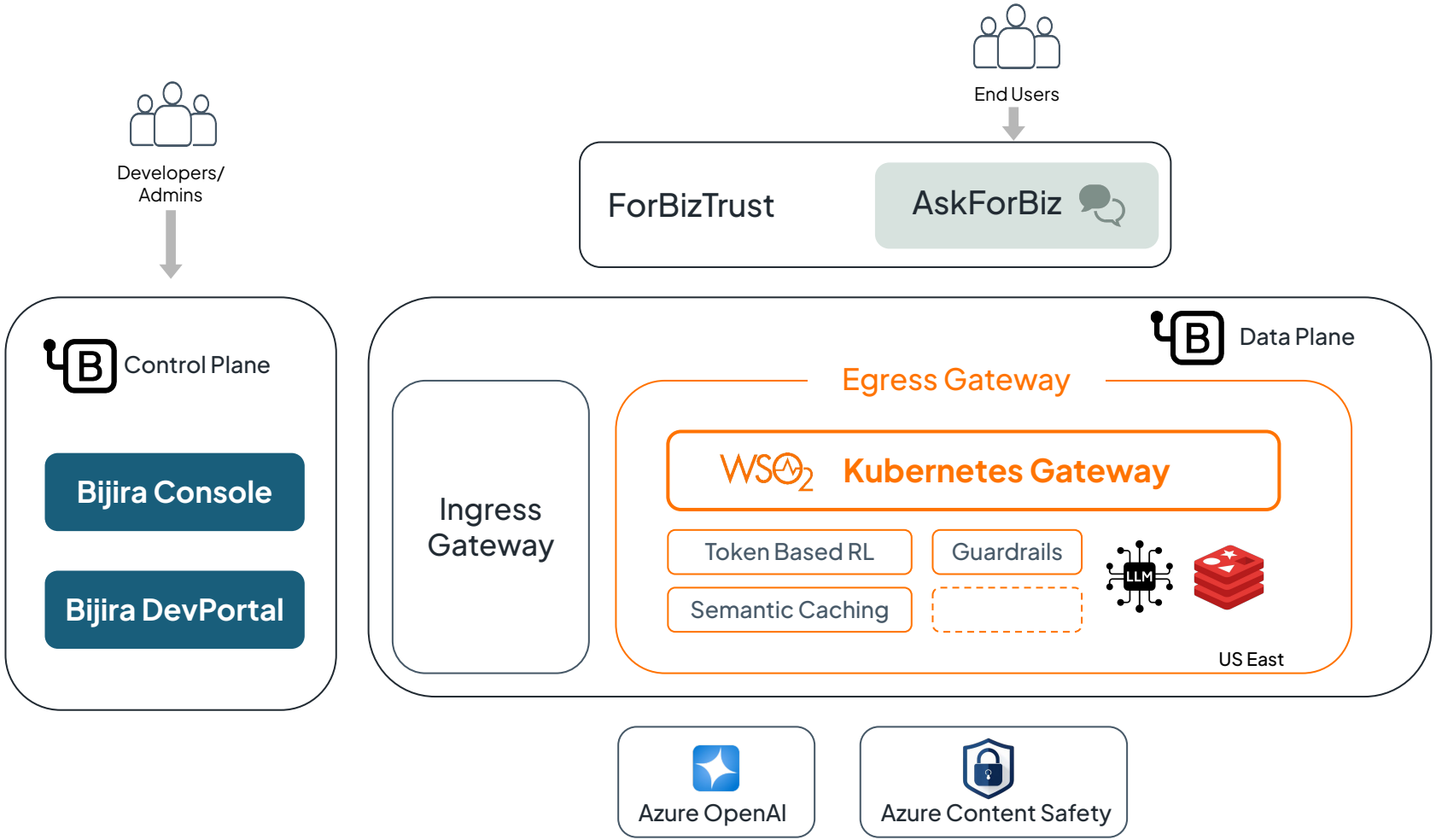
The Scenario

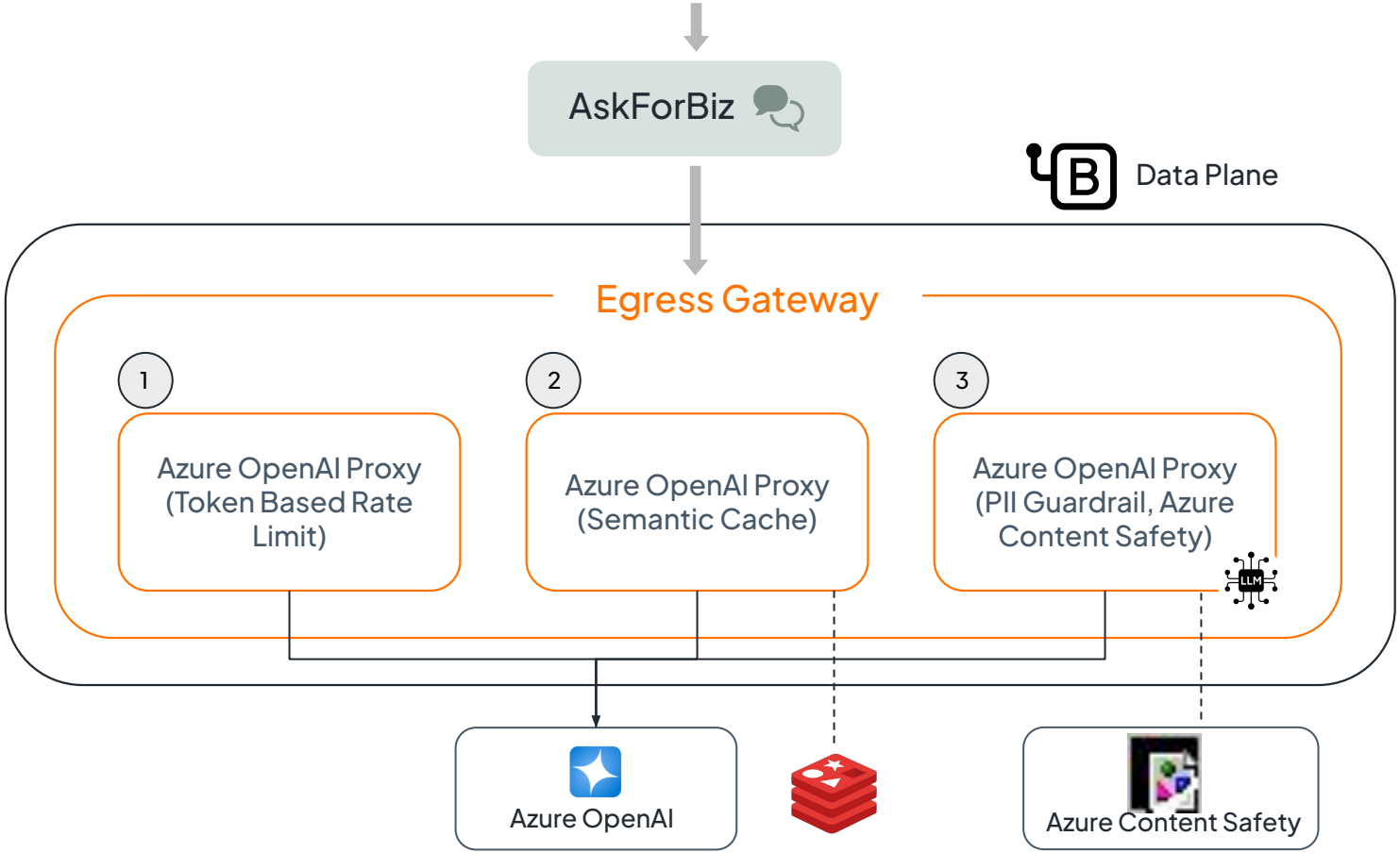
Challenge: You are the Chief Solution Architect at **ForBizTrust,** building an AI-powered customer assistant - **AskForBiz –** for retail banking. The goal is to embed GenAI into customer support and transaction insights.

Critical Architecture Mandates:

- ForBizTrust can't risk exposing their OpenAI Key
- Tight control over AI usage - specially tokens, not just requests
- They can't afford to leak customer PII
- They need to reduce LLM latency and cost for repetitive FAQ-style queries



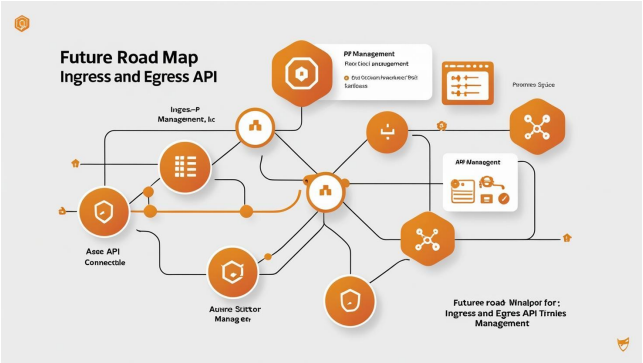

End Users
Developers/
Admins
ForBizTrust
AskForBiz
Control Plane
Bijira Console
Bijira DevPortal
Data Plane
Ingress
Gateway
Egress Gateway
WSO2 Kubernetes Gateway
Token Based RL
Guardrails
Semantic Caching
LLM
US East
Azure OpenAI
Azure Content Safety


AskForBiz
Data Plane
Egress Gateway
1
Azure OpenAI Proxy (Token Based Rate Limit)
2
Azure OpenAI Proxy (Semantic Cache)
3
Azure OpenAI Proxy (PII Guardrail, Azure Content Safety)
Azure OpenAI
Azure Content Safety

Future Roadmap

Future Roadmap

- More first class federated gateway types support (eg: Azure, Kong)
- In case there are changes in federated gateways our connectors will be versioned to bridge them.
- Multi Vendor Routing Support for AI Gateway
- Additional Egress/AI API types will be added in the future



Key Takeaways and Next Steps

Key Takeaways and Next Steps

- **Unified API management simplifies operations** – Centralizing ingress and egress governance reduces complexity across multiple API gateways.
- **Consistent security and policies** – A unified control plane ensures uniform enforcement of authentication, rate limiting, and compliance rules.
- **Enhanced observability and analytics** – Real-time monitoring across all API gateways provides better insights for performance and security.
- **Scalability & future-proofing** – A unified approach makes it easier to integrate new gateways, cloud services, and AI-based APIs.
- <ingress/egress>

Upcoming API Management Sessions

- **Conference Day 3 (Thursday, July 31st)**

WSO2 API Manager Roadmap and Vision →

Christopher Davey
Vice President & General Manager - APIM, WSO2

Bijira: API PaaS for the Cloud and AI-Native Era →

Pubudu Gunatilaka
Associate Director & Head of Architecture, APIM BU, WSO2

Managing APIs Across Federated Gateways →

Thilini Shanika
Associate Director and Head of Engineering, APIM BU, WSO2

What You Can Do with the AI Gateway →

Arshardh Ifthikar
Technical Lead, WSO2

Modern Enterprise Architecture with APIs →

Vidura Gamini Abhaya
Vice President, Solutions Architecture, WSO2

Customer Panel: Journeys in Enterprise Modernization Through API Management →

Question Time!




Thank you!
20TH ANNIVERSARY EDITION
WSO2CONASIA
PLATFORMLESS MODERNIZATION