



May 20 - 22, 2026 | Austin, Texas, USA

Hybrid Architecture and Multi-Gateway Control



WSO2con
NORTH AMERICA

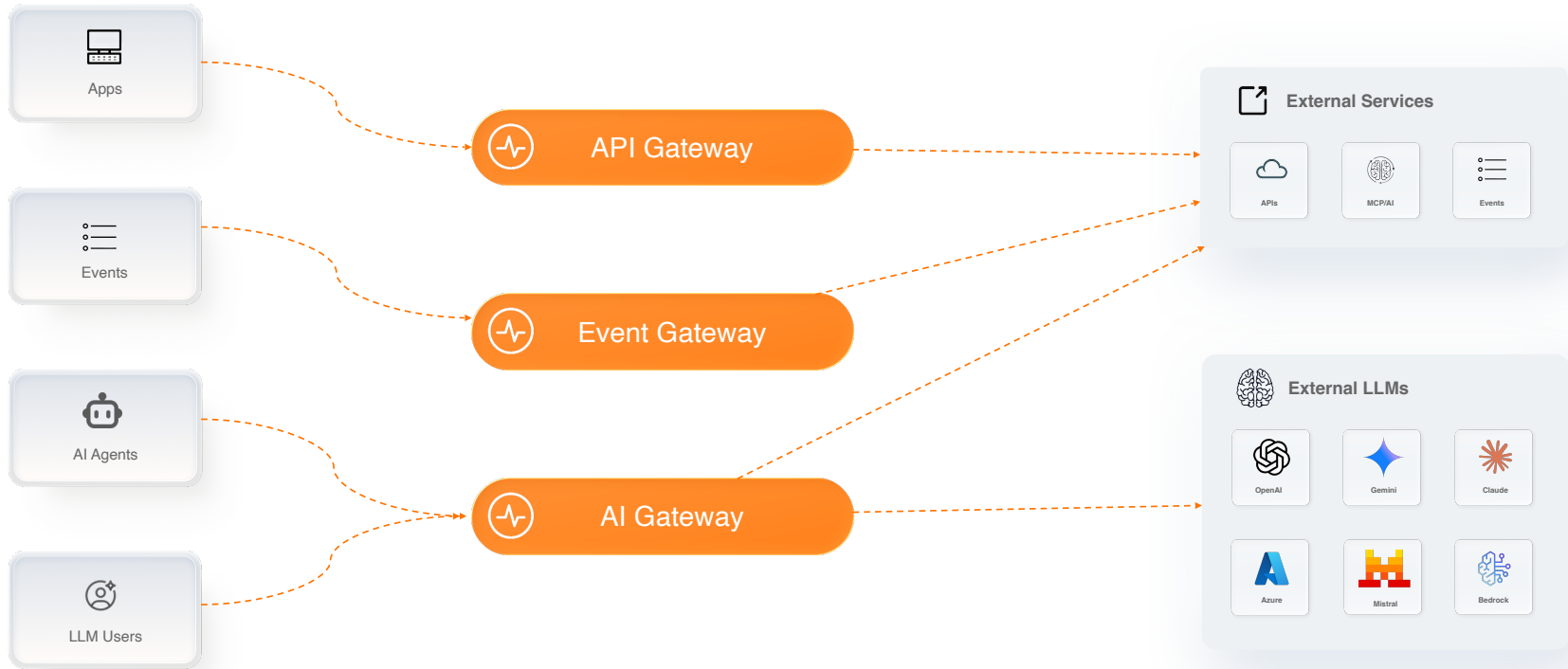
May 20 - 22, 2026 | Austin, Texas, USA



Pubudu Gunatilaka
Director of Engineering
WSO2



The Single Gateway Model is Reaching Its Limits



Modern Architectures Demand More



Infrastructure Complexity

- Multi Cloud
- Hybrid (Cloud + On-prem)
- Edge Deployments



Compliance & Performance

- Data residency
- Low latency
- Regional routing



API Diversity

- REST APIs
- Event APIs
- AI / Agent APIs



Organizational Scale


- Team autonomy
- Business-unit ownership
- Independent release cycles



Hybrid Architecture

Hybrid Architecture


Combining multiple deployment models to meet business and technical needs.



Cloud + On-prem

Purpose
Mix agility with legacy integrations


Example
SaaS + Internal systems



Managed + Self Hosted

Purpose
Balance convenience and control

Example
Managed gateways + Kubernetes native gateways



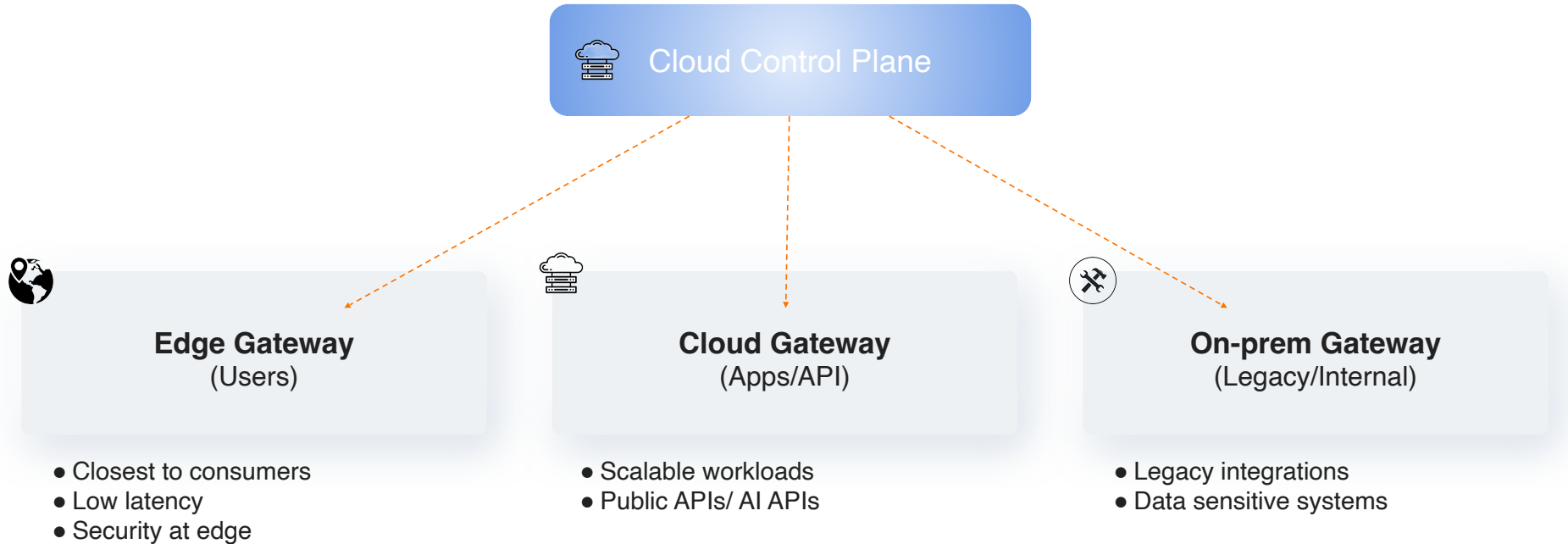
Edge + Regional

Purpose
Improved latency and compliance

Example
Country/region specific deployments

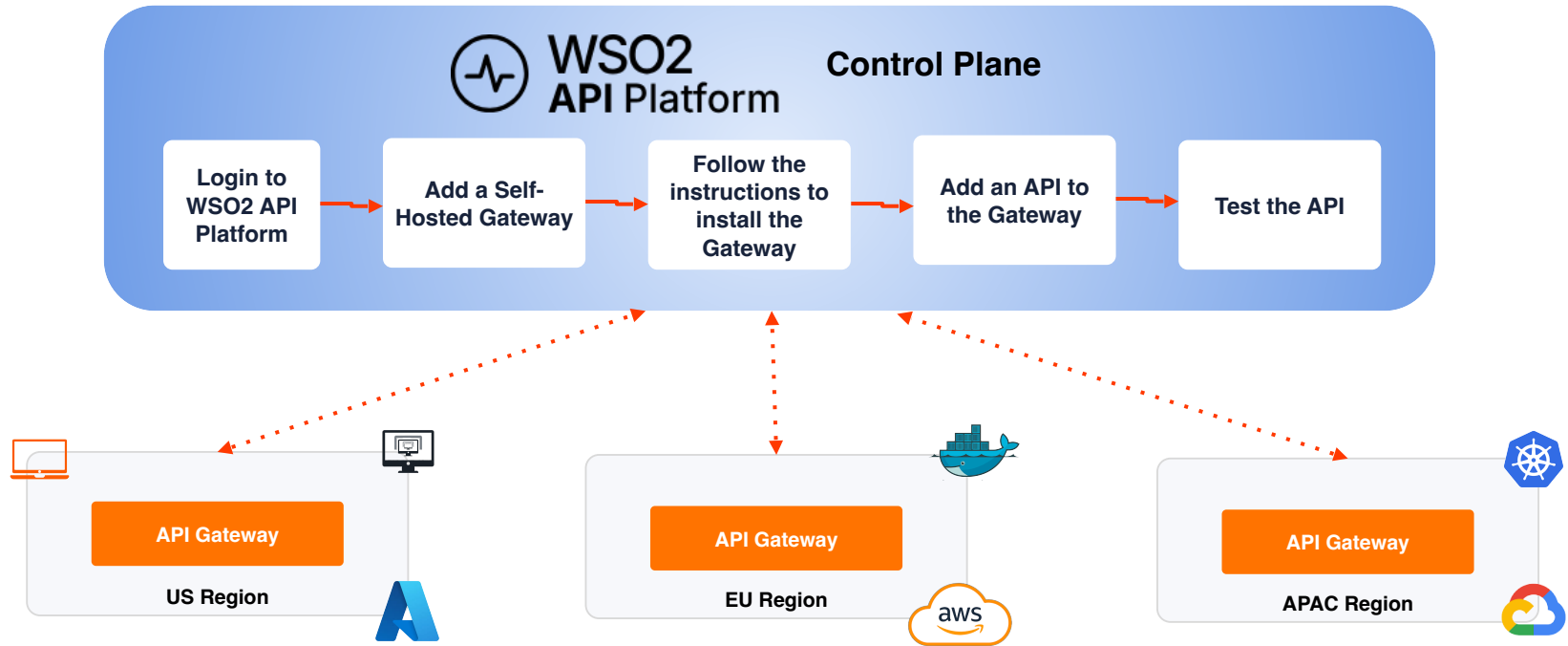


A Distributed Runtime with Centralized Control



Introducing Self-Hosted Gateways for the WSO2 API Platform

Self Hosted Gateways



Self Hosted Gateways...

WSO2 API Platform pubudugunatlaka

← Back to Gateways

Add Self-Hosted Gateway

General Details

Type

API Gateway AI Gateway Event Gateway Beta

Name Description (Optional)

Configurations

URL Associated Environment

Back Add

Quick Start Virtual Machine Docker Kubernetes

Prerequisites

- cURL installed
- unzip installed
- Docker installed and running
- Docker Compose installed

Step 1: Download the Gateway

Run the command in your terminal to download the gateway:

```
curl -sLO https://github.com/wso2/api-platform/releases/download/gateway/v1.1.0/wso2apip-api-gateway-1.1.0.zip && \
unzip wso2apip-api-gateway-1.1.0.zip
```

Step 2: Configure the Gateway

Run this command to create wso2apip-api-gateway-11.0/configs/keys.env with the required environment variables:

```
cat > wso2apip-api-gateway-1.1.0/configs/keys.env << 'ENVFILE'
MOESIF_KEY=<your-moesif-key>
GATEWAY_CONTROL_PLANE_HOST=connect.bijira.dev
GATEWAY_REGISTRATION_TOKEN=<your-gateway-token>
ENVFILE
```

Step 3: Start the Gateway

1. Navigate to the gateway folder.

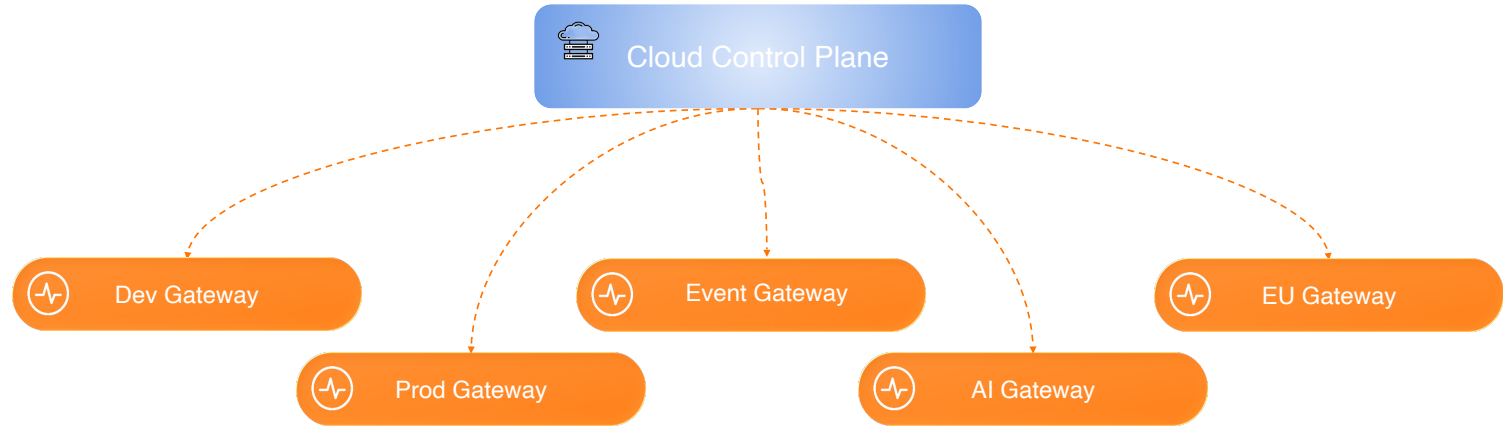
```
cd wso2apip-api-gateway-1.1.0
```

2. Run this command to start the gateway using the configs/keys.env file created in Step 2:

```
docker compose --env-file configs/keys.env up
```



Self Hosted Gateways...



Environment Based

- Dev / Test / Prod
- Regional Deployments



Workload Based

- REST / GraphQL APIs
- Event / Streaming APIs
- AI / MCP APIs



Team Based

- Business unit ownership
- Independent release cycles

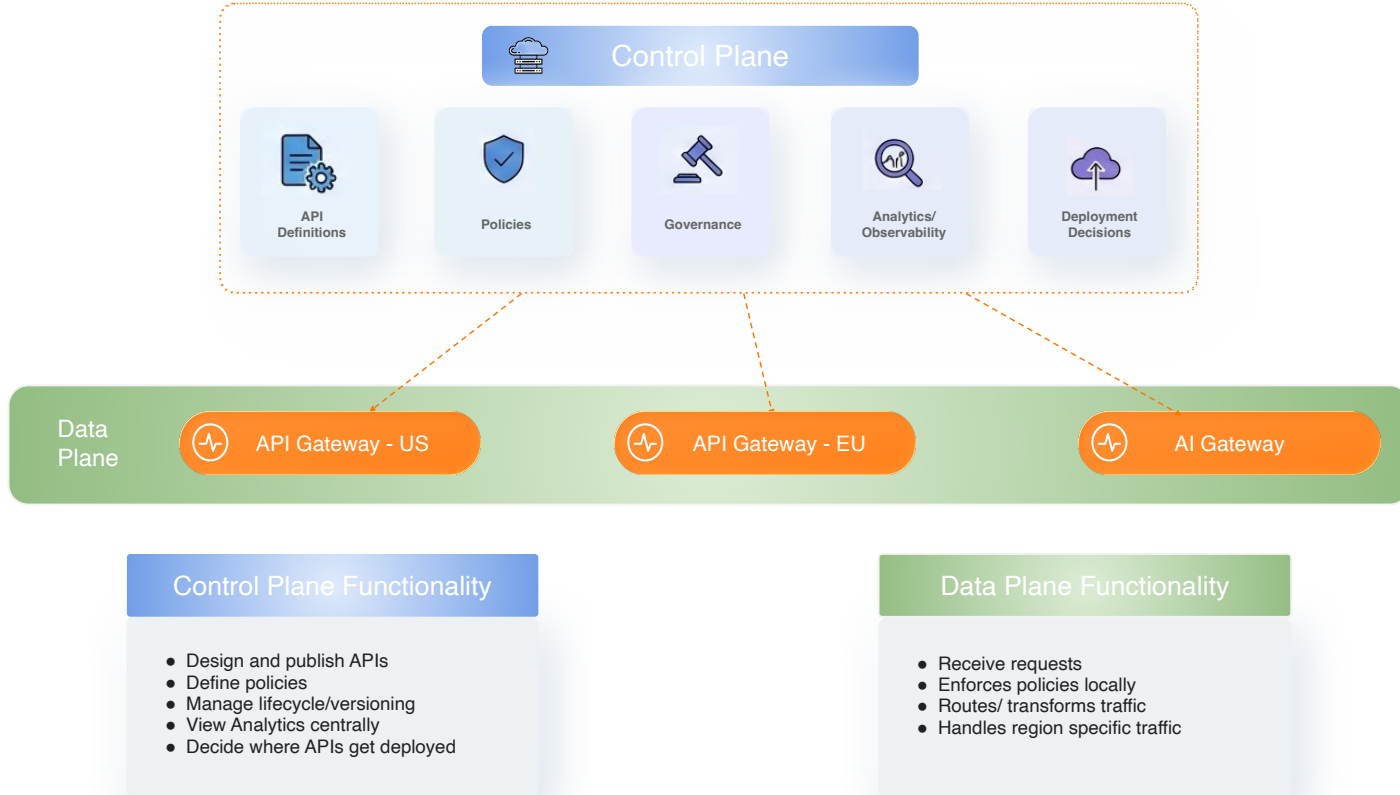


Specialized Gateways

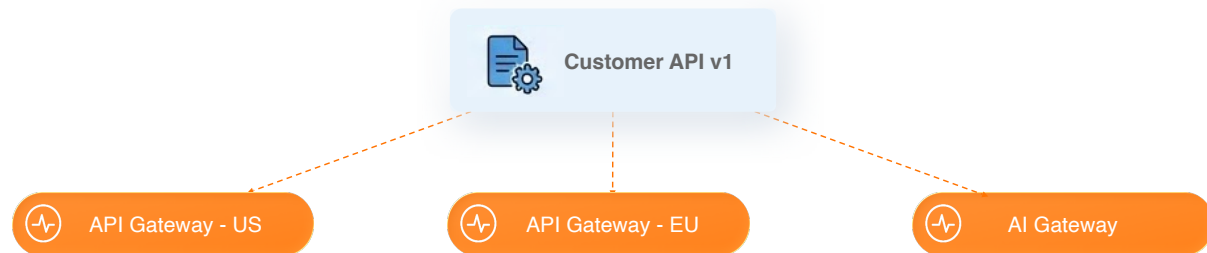
- AI Gateways
- Event Gateways
- Internal only Gateway



Centralized Control, Decentralized Execution



One API Definition, Many Deployments



Define once, apply everywhere

Security

Rate Limits

AI Guardrails

Any gateway policy

Selective Deployments

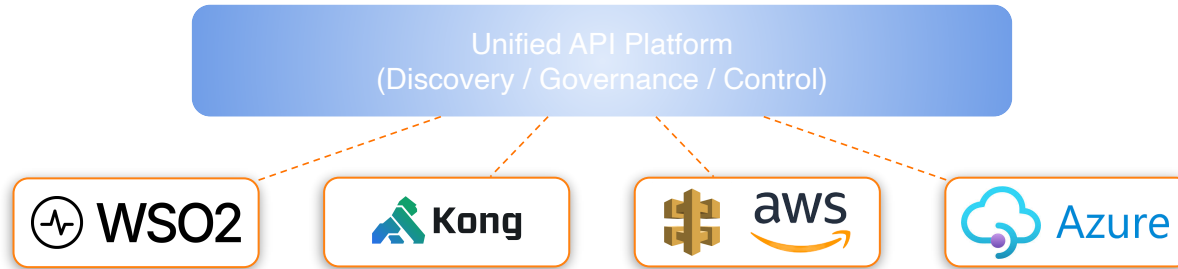
- Region based
 - US only
 - EU only
- Environment based
 - Dev / Prod
- Gateway type based
 - AI Gateway
 - Event Gateway



Govern Traditional APIs and AI APIs under the same Platform



Gateway Federation: Unifying Distributed API Gateways



Plug Any Gateway

Federate existing gateway solutions instead of replacing them.

- Avoid vendor lock-in
- Preserve existing investments

Central API Discovery

Single discovery point for API consumers:

- Unified API Catalog & Developer Portal
- Search APIs across federated gateways
- One place to discover, subscribe, and access APIs

Unified Access Layer

- Consistent API Catalog
- Unified governance and lifecycle visibility
- Standardized onboarding experience

Unified Visibility

Observe APIs across gateways

- Centralized Analytics
- Centralized Observability

Use Cases

Use Case: Multi-Region SaaS APIs



Global APIs with local compliance and performance



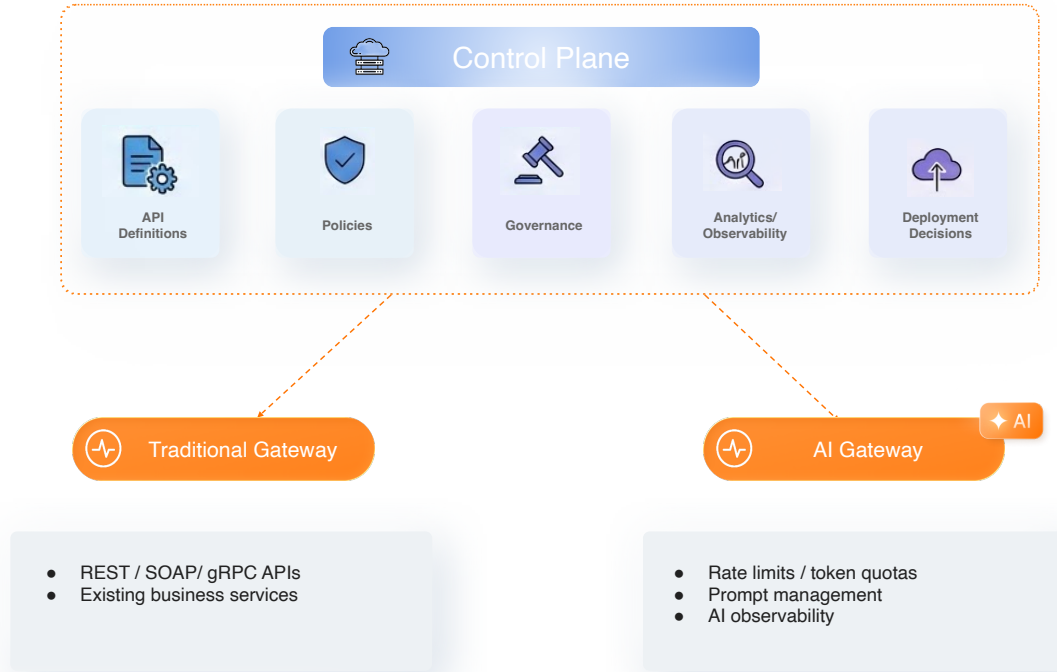
Use Case: Hybrid Enterprise Integration



Modernize externally, integrate internally



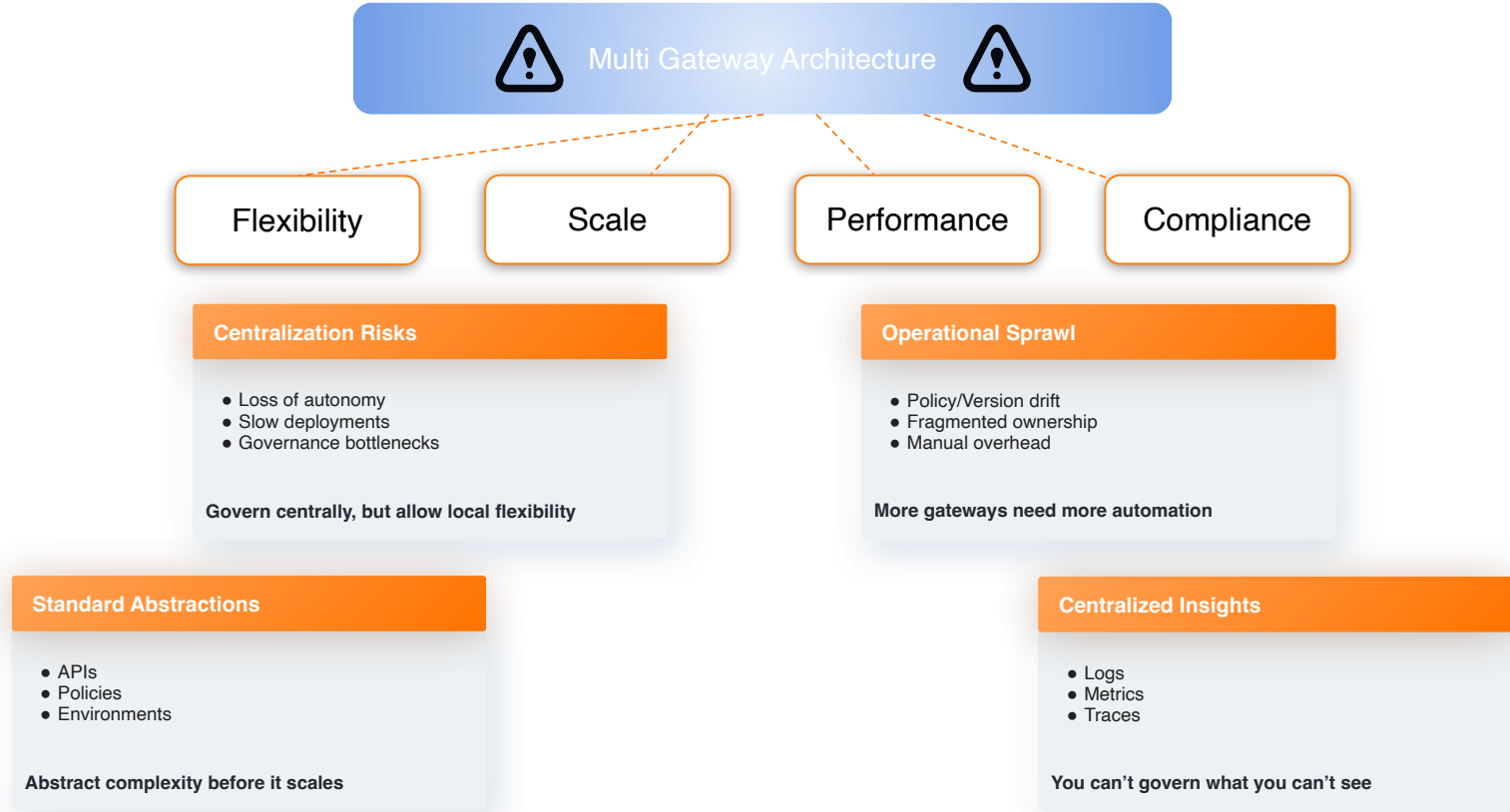
Use Case: AI + Traditional APIs



Separate runtimes, unified governance



Lessons Learned & Common Pitfalls



Key Takeaways



Hybrid Architectures are the New Reality

Cloud, on-prem, edge, and AI runtimes now coexist.



Centralized Control Enables Distributed Gateways

Govern APIs, policies, and deployments without centralizing traffic.



Gateway Federation Reduces Fragmentation

Unify discovery, governance, and visibility across heterogeneous gateways.



Future Gateways Will Be AI-Aware and Policy-Driven

API gateways are evolving into intelligent distributed runtimes.



A dark blue background featuring a silhouette of a city skyline at night. The skyline includes various skyscrapers and a prominent tower with a spherical top. The sky is filled with small white stars and a few larger, brighter stars. In the top left and bottom right corners, there are decorative wavy lines in shades of blue and pink. A thin orange horizontal line runs across the middle of the page.

Question Time!

An orange circle containing a white question mark, positioned on the orange horizontal line.

?



May 20 - 22, 2026 | Austin, Texas, USA

Thank You!

