20TH ANNIVERSARY EDITION

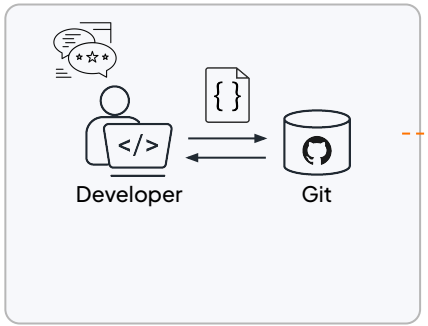
WSO2CON ASIA

PLATFORMLESS MODERNIZATION

Engineering the Backbone: How We Built Our Internal Developer Platform

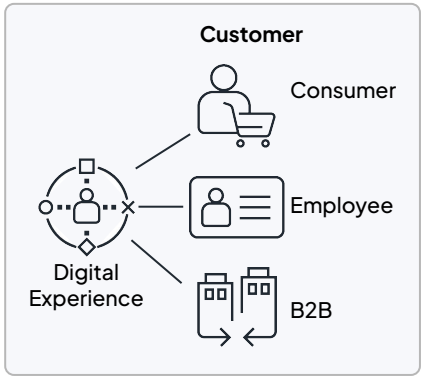
Lakmal Warusawithana
VP & Distinguished Engineer
WSO2

The reality of software manufacturing in the cloud native era



You write code. You want to get it to the hands of the customer ASAP.

- Code analysis, Unit testing, Vulnerability scanning, Secret scanning
- Container image build, Image tagging and versioning, Image signing, Build artifact upload to registry
- Integration testing, End-to-end testing
- Promotion between environments (dev → stage → prod)
- Zero Trust security enforcement, Log and metrics pipeline setup, Alerting rule updates, Audit trail capture



These are the kinds of repetitive, cross-cutting tasks that platform teams aim to standardize, abstract, and automate into a platform

Internal developer platform

The internal developer platform, as we see it

Establish your **enterprise software engineering** practice

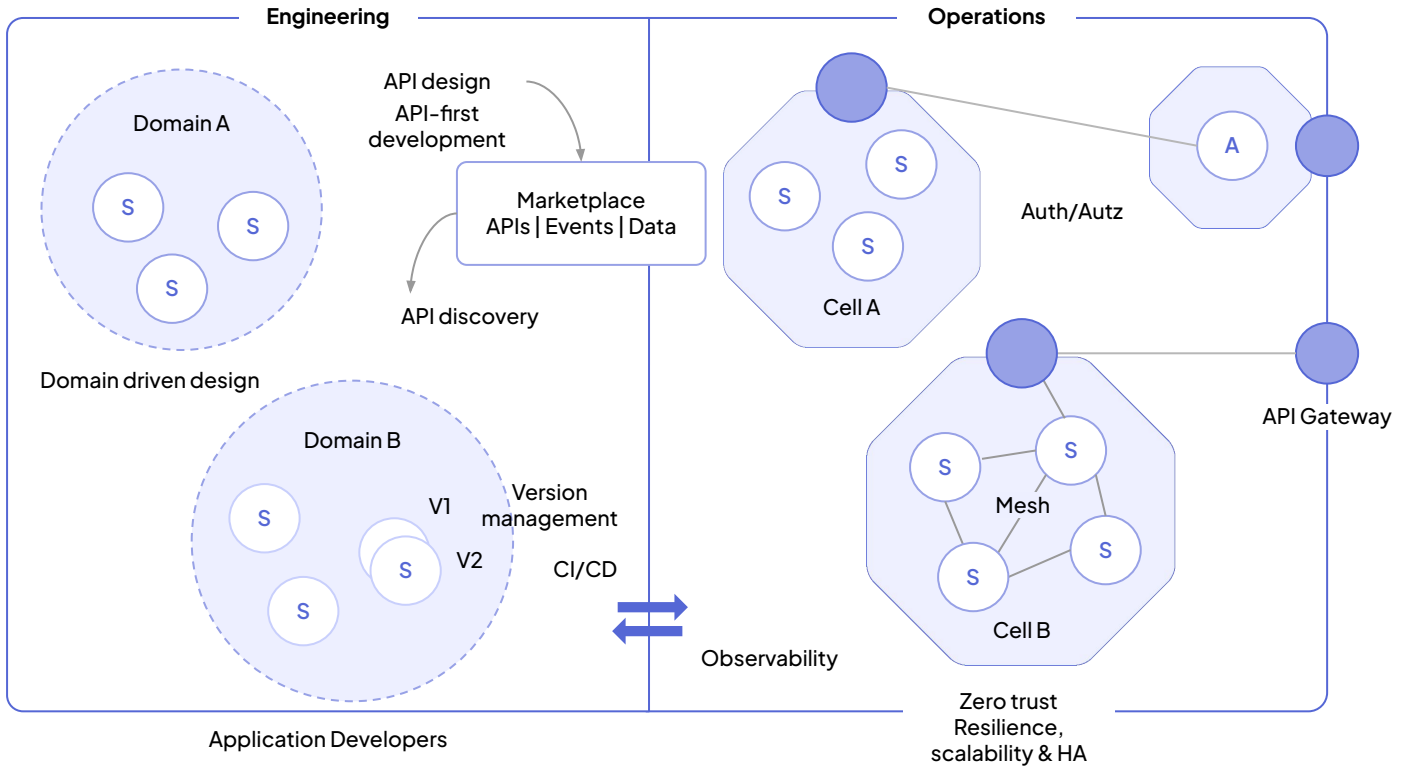
- API-first development
- Domain driven design
- Microservice architecture
- Test driven development
- Automated DevOps
- Secure by default
- Version management
- Iterative architecture
- AI augmented engineering

Refine modern **software delivery & operations**

- Developer self-service
- Containerization & Kubernetes
- CI/CD & GitOps
- Resource optimization
- Multi-cloud, multi-environment
- Observability & alerting
- Config & secret management
- Resiliency, scalability & HA
- Operational & business insights

Our journey in implementing Choreo: An internal developer platform as a service

Ultimate experience



Key challenges faced in architecting and implementing Choreo

- **Bridging development and operations**

  Aligning software engineering, delivery, and operational workflows into a seamless, unified experience.

- **Scalability across dimensions**

  Ensuring scalability not just at runtime, but also across engineering processes and support operations.

- **Multi-cloud and multi-region support**

  Building capabilities to operate reliably across multiple cloud providers and geographic regions.

- **Run anywhere**

  Providing the flexibility to deploy workloads close to the data source for optimal performance and compliance.

- **Accelerated feature delivery**

  Improving development velocity and deployment frequency to support faster, more reliable feature releases.

Key challenges faced in architecting and implementing Choreo

- **Compliance adherence**

  Ensuring alignment with regulatory, security, and industry-specific compliance requirements.

- **Built-in zero trust security**

  Embedding zero-trust principles as a default across the platform architecture.

- **Enterprise-grade governance**

  Providing strong governance frameworks to support enterprise-level requirements and policy enforcement.

- **Observability and monitoring**

  Offering comprehensive visibility into system behavior, performance, and health through integrated tools.

- **Cost optimization**

  Reducing infrastructure and cloud operational costs through efficient resource utilization and automation.

Key technical strategies for building Choreo

#1: Separation of control plane and data plane

Challenges overcome:

- Scalability
- Separation of management and runtime traffic

#2: Adherence to the right architecture

Challenges overcome:

- Bridging development and operations
- Scalability
- Compliance adherence
- Built-in zero trust security
- Enterprise governance

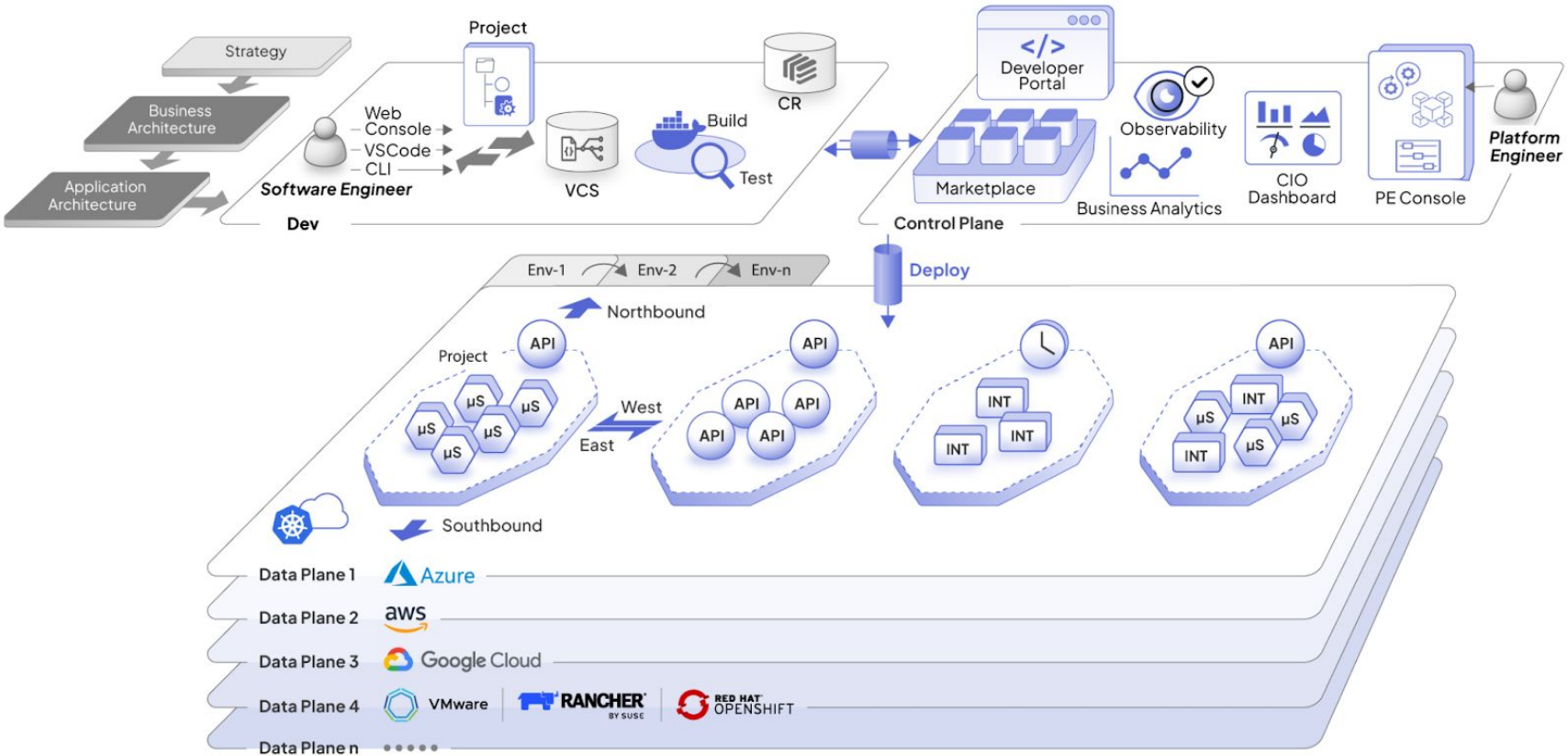
#3: Adopting a cloud-provider-agnostic technology stack

Challenges overcome:

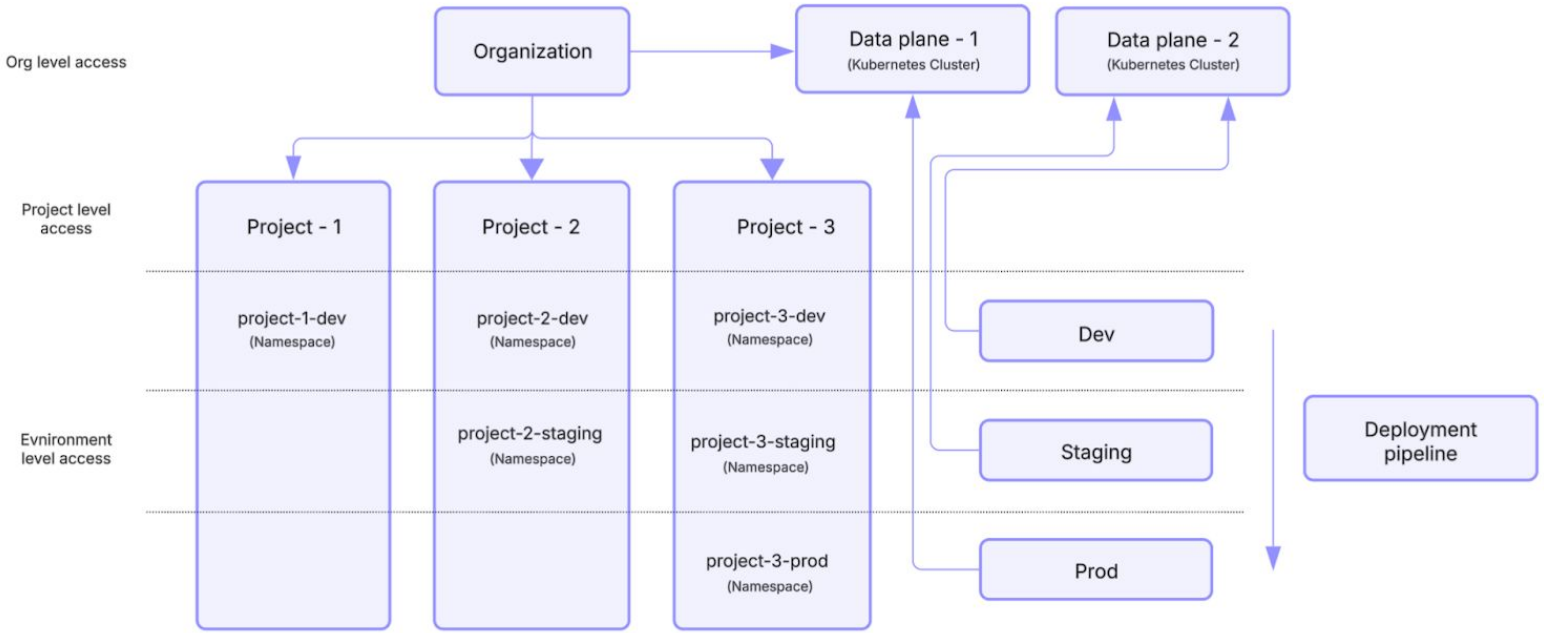
- Multi-cloud and Multi-regional support
- Run anywhere
- Scalability
- Increased feature release velocity
- Cost optimization
- Built-in zero trust security
- Enterprise governance
- Observability and monitoring

Key architectural patterns and abstractions in Choreo

Control plane and data plane separation

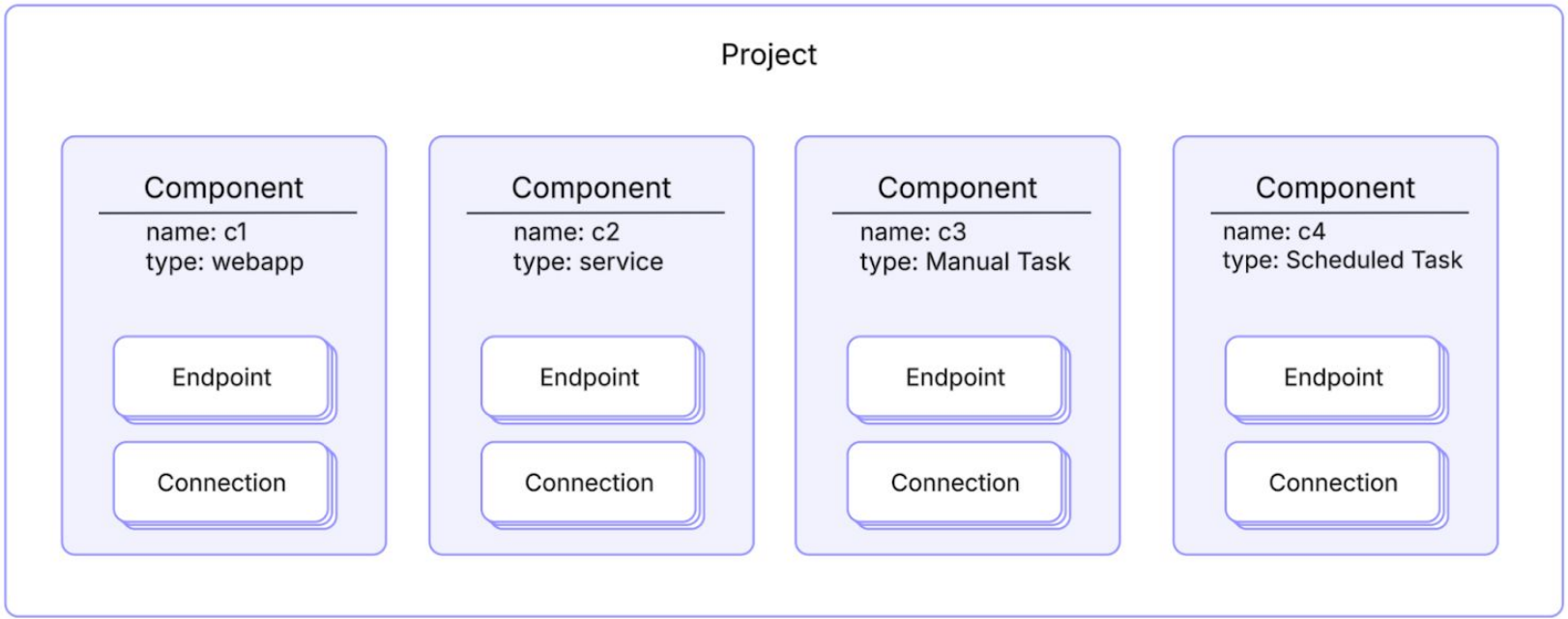


Right-level of abstractions



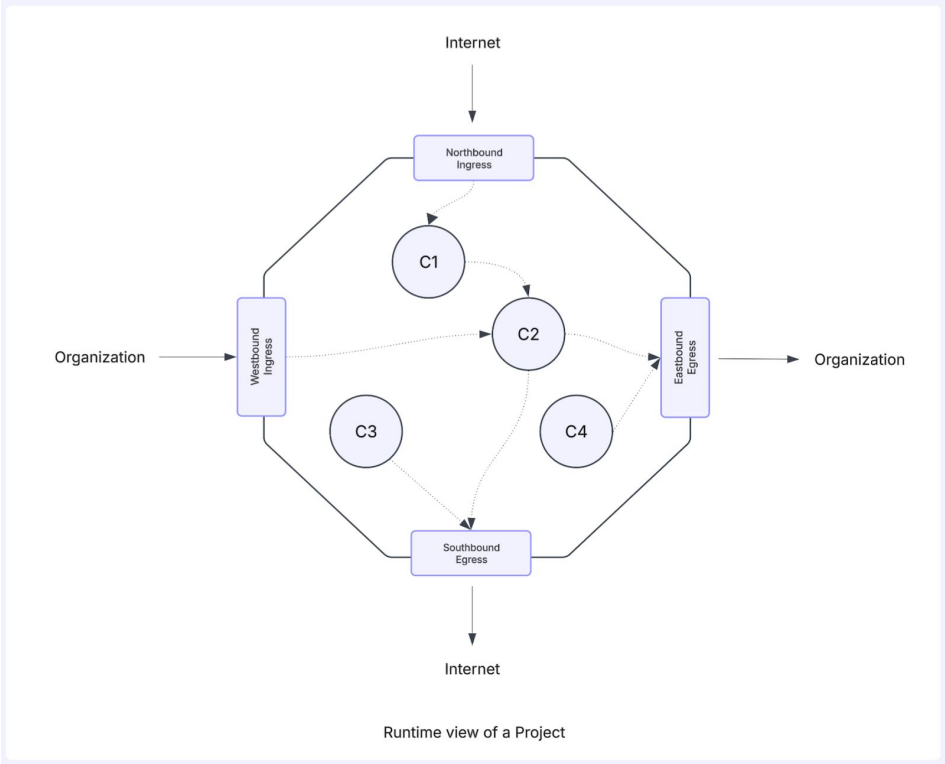
Platform Abstractions

Right-level of abstractions…



Development Abstractions

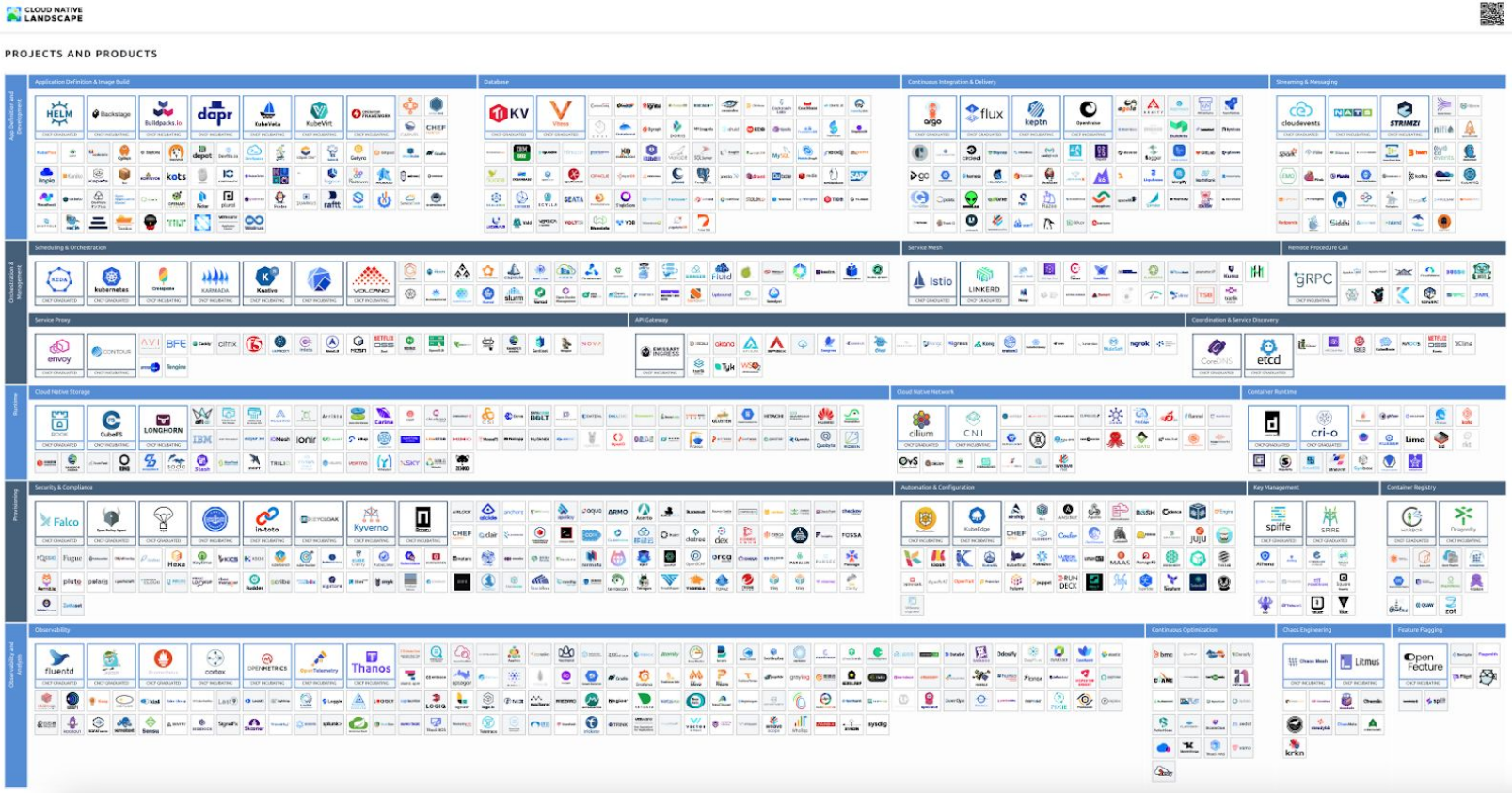
Cell-based Architecture (CBA)



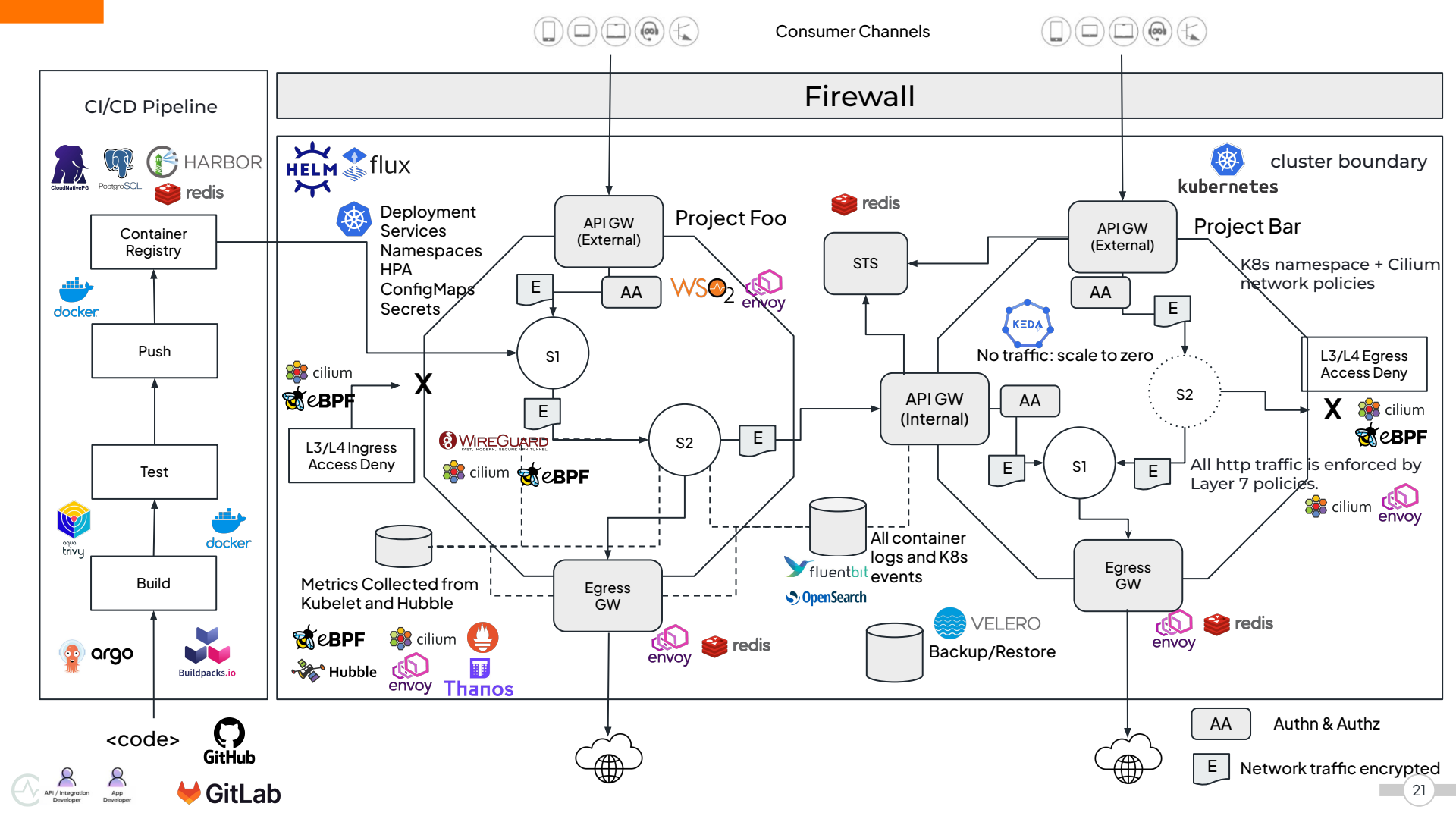
https://github.com/wso2/reference-architecture/blob/master/reference-architecture-cell-based.md

Cloud-provider-agnostic stack and CNCF

The cloud native landscape is very complex



Cloud native tools used in the current implementation of Choreo

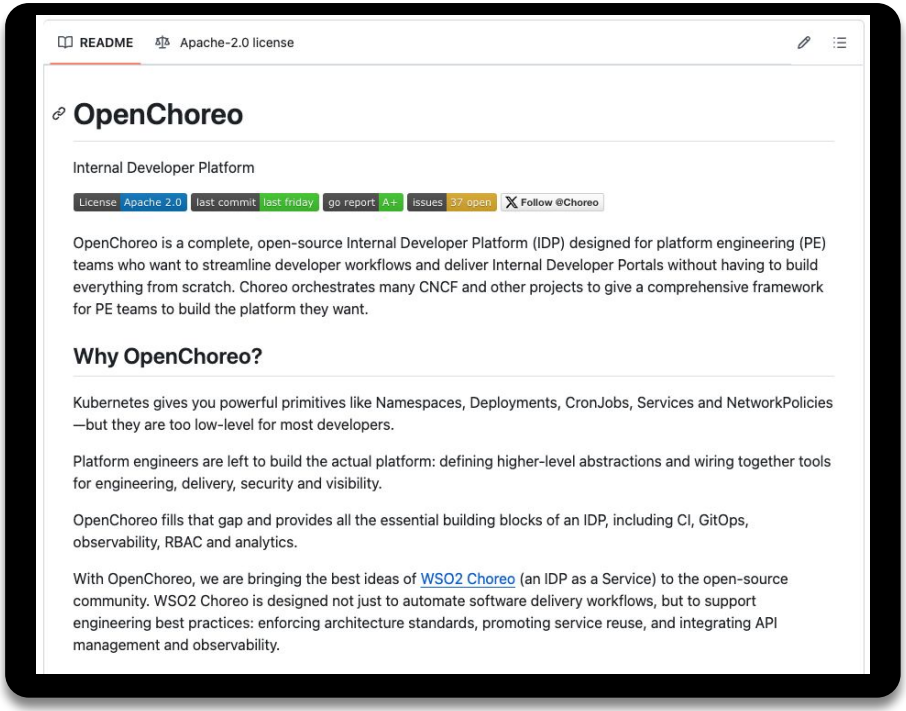

Consumer Channels
Firewall
CI/CD Pipeline
Container Registry
Push
Test
Build
<code>
GitHub
GitLab
HELM
flux
Deployment
Services
Namespaces
HPA
ConfigMaps
Secrets
API GW (External)
Project Foo
E
AA
S1
S2
WSO2
envoy
cilium
eBPF
L3/L4 Ingress Access Deny
WireGuard
Metrics Collected from Kubelet and Hubble
Hubble
Thanos
Egress GW
redis
STS
API GW (Internal)
fluentbit
OpenSearch
All container logs and K8s events
VELERO
Backup/Restore
cluster boundary
kubernetes
Project Bar
K8s namespace + Cilium network policies
KEDA
No traffic: scale to zero
L3/L4 Egress Access Deny
All http traffic is enforced by Layer 7 policies.
Authn & Authz
Network traffic encrypted

Choreo journey...so far

Choreo engineering facts

- Integrated 20+ CNCF projects
- Single unified console for all functionalities
- Over 60 team members working for 4+ years
- 20+ core platform engineers
- 15+ dedicated SREs
- Support for all major cloud providers, including Azure, GCP, AWS, and on-premises. The only prerequisites are a dedicated Kubernetes cluster, networking, and storage infrastructure.

Choreo is going open source!



https://github.com/openchoreo/openchoreo

Question Time!


Thank you!
20TH ANNIVERSARY EDITION
WSO2CONASIA
PLATFORMLESS MODERNIZATION