Cloud Native API Management on Kubernetes

Workshop - 18/6/2020

Pubudu Gunatilaka - pubudug@wso2.com, @PubuduSpace (twitter)

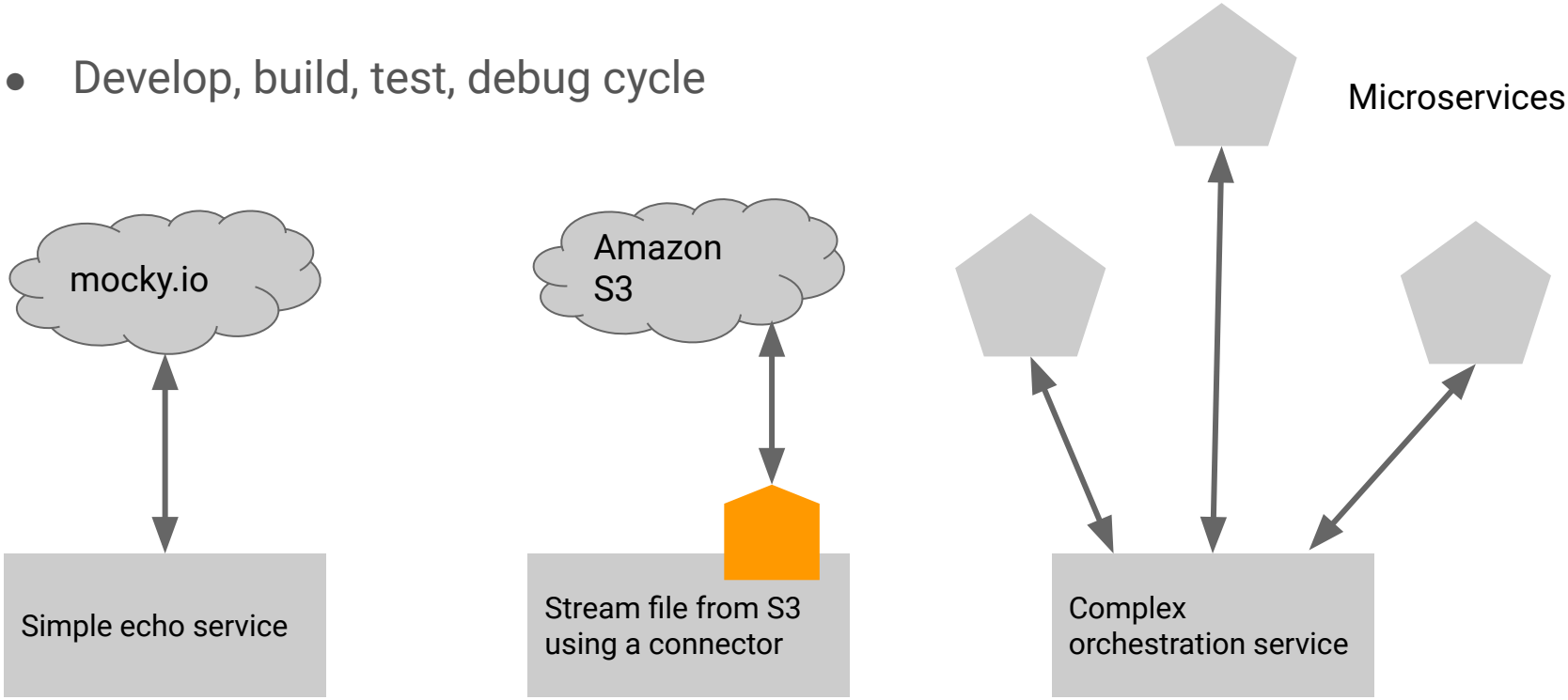
Technical Lead

Agenda

- Part 1 - Development cycle - backend services, integration services (dev, build, debug test cycle)
- Part 2 - Deploying services to kubernetes (dependent backend services)
- Part 3 - Creating a managed API for services - Publishing, token generation and accessing. CLI or through developer portal
- Part 4 - CI/CD pipeline integration and planning
- Part 5 - Kubernetes best practices

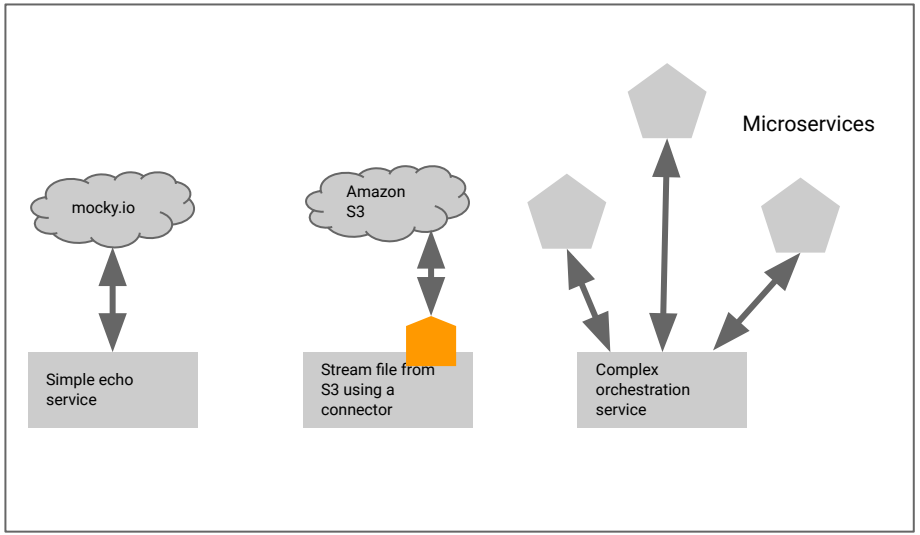
Part 1- Developing backend services

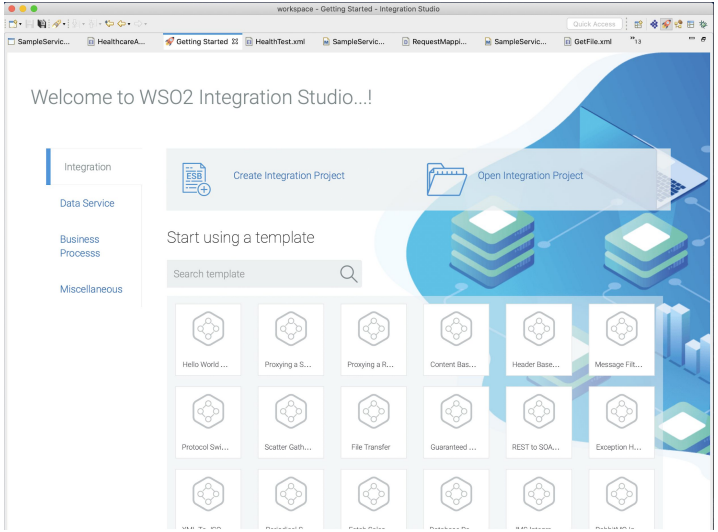
- Develop, build, test, debug cycle



Part 1- Developing backend services

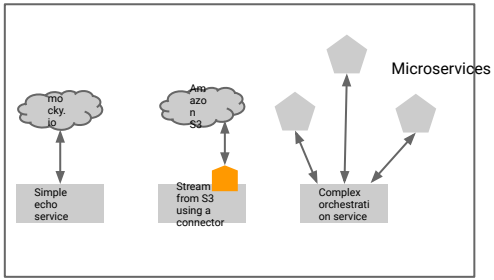
- Develop, build, test, debug cycle

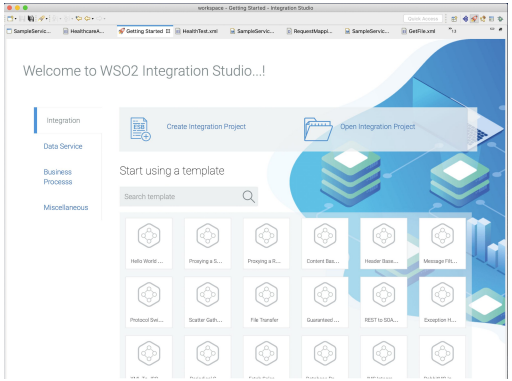




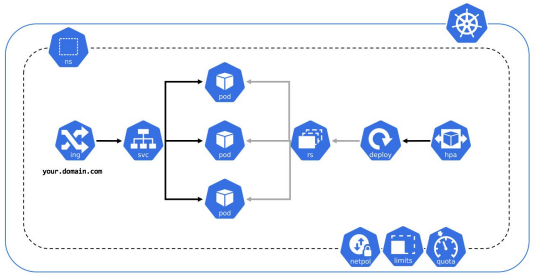
Part 1- Developing backend services

- Develop, build, test, debug cycle





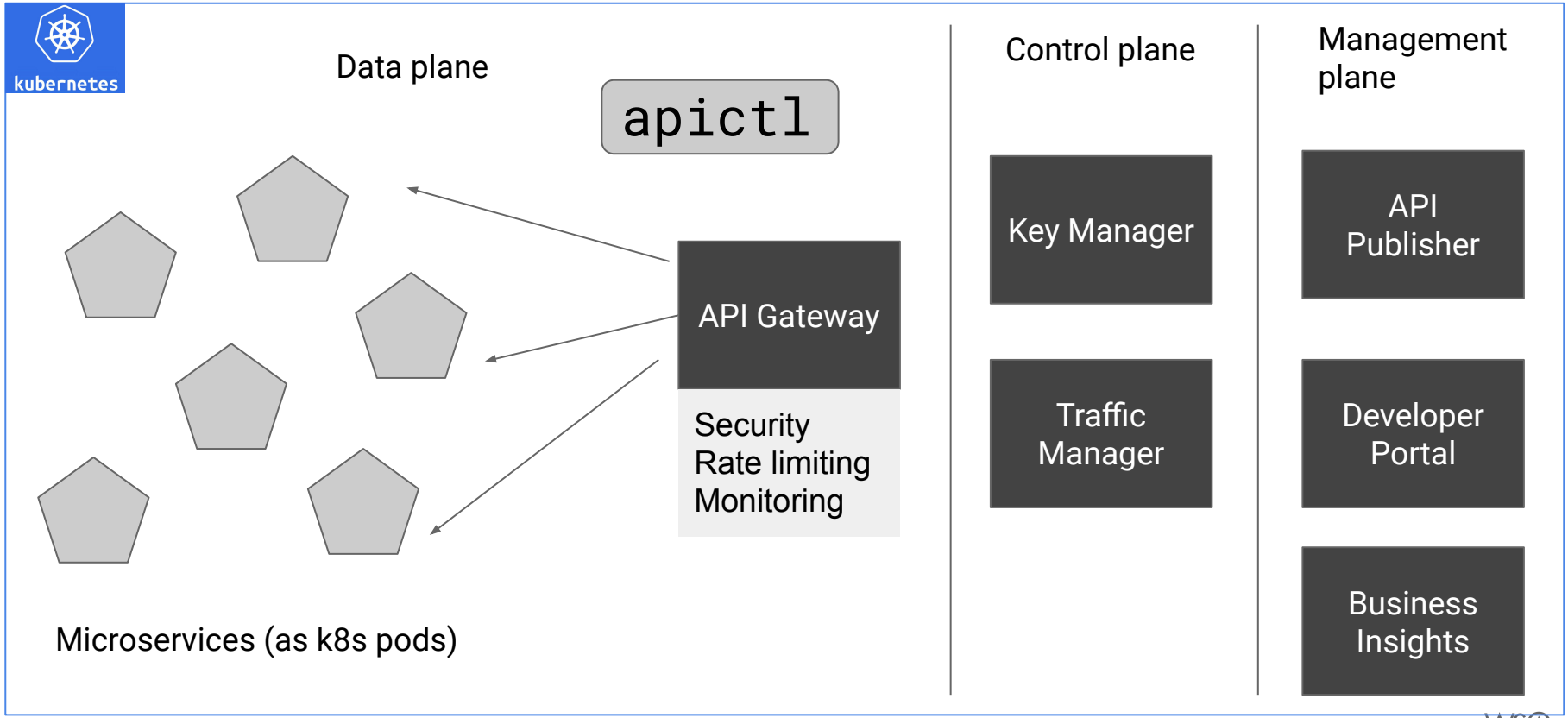
Kubernetes cluster



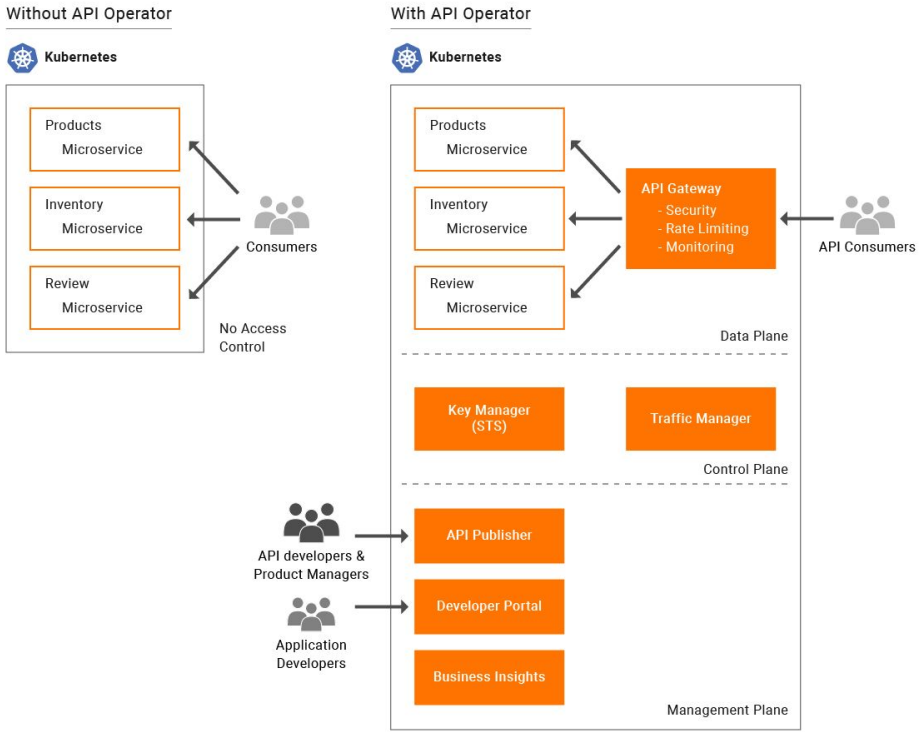
Part 2 - Deploying services to Kubernetes

- For dependent backend services - kubectl apply (we're assuming they're already available in a shared environment in real life)
- Containerizing and deploying integration services (Integration Studio + EI K8S Operator)
- Testing services (port numbers / URLs may vary depending on the setup)

Part 3 - Creating managed APIs for services



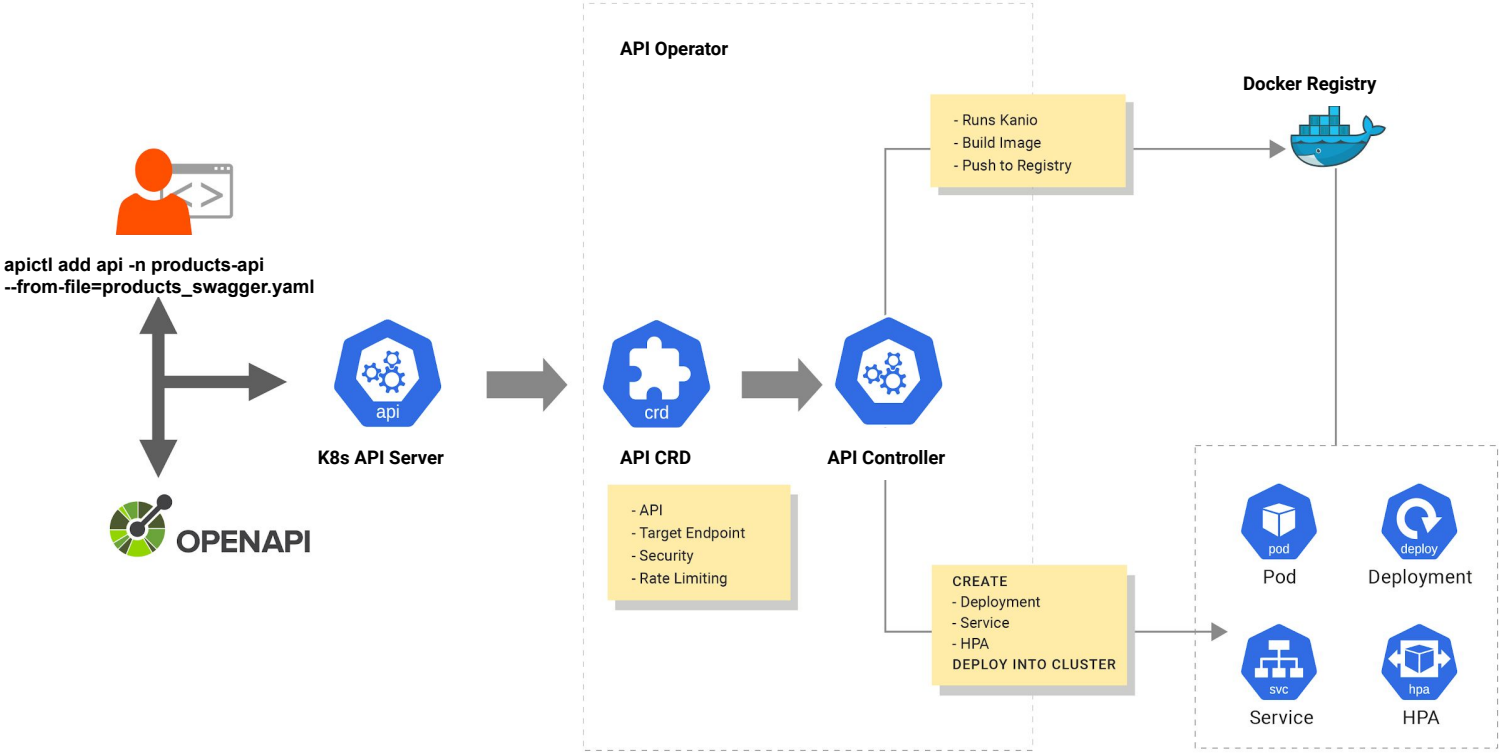
API Operator for Kubernetes



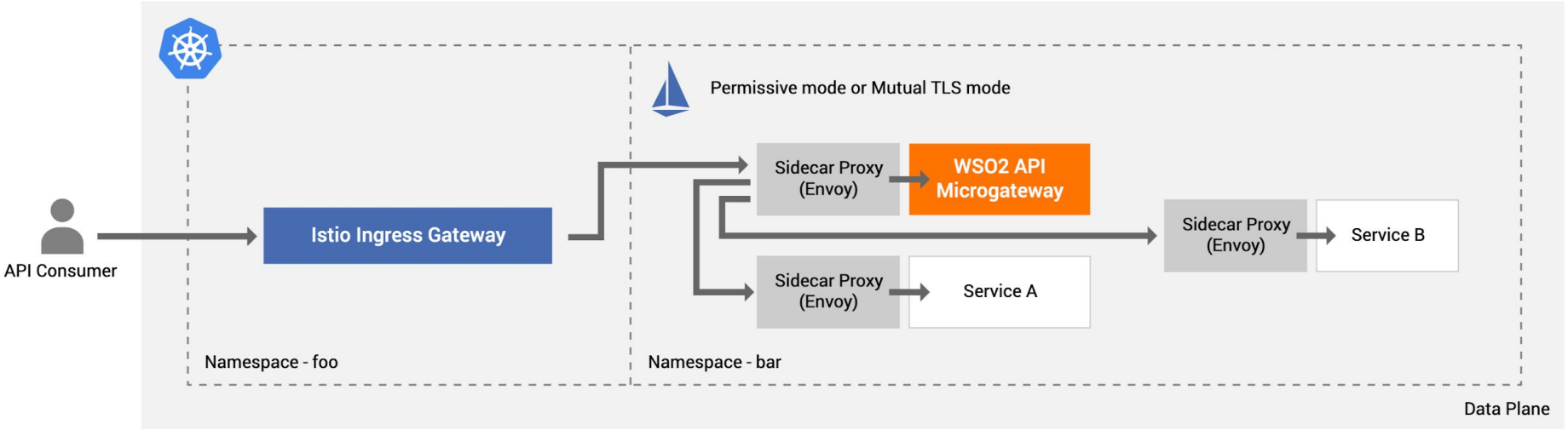
- Makes API a first class citizen in Kubernetes eco system
- Easier to deploy
- Imposese
  - Security
  - Throttling
  - Mediation
  - Monitoring

    For your microservices
- Build in deployment patterns
- Fully automated experience for cloud native API management
- Single source of truth

API Operator Overview

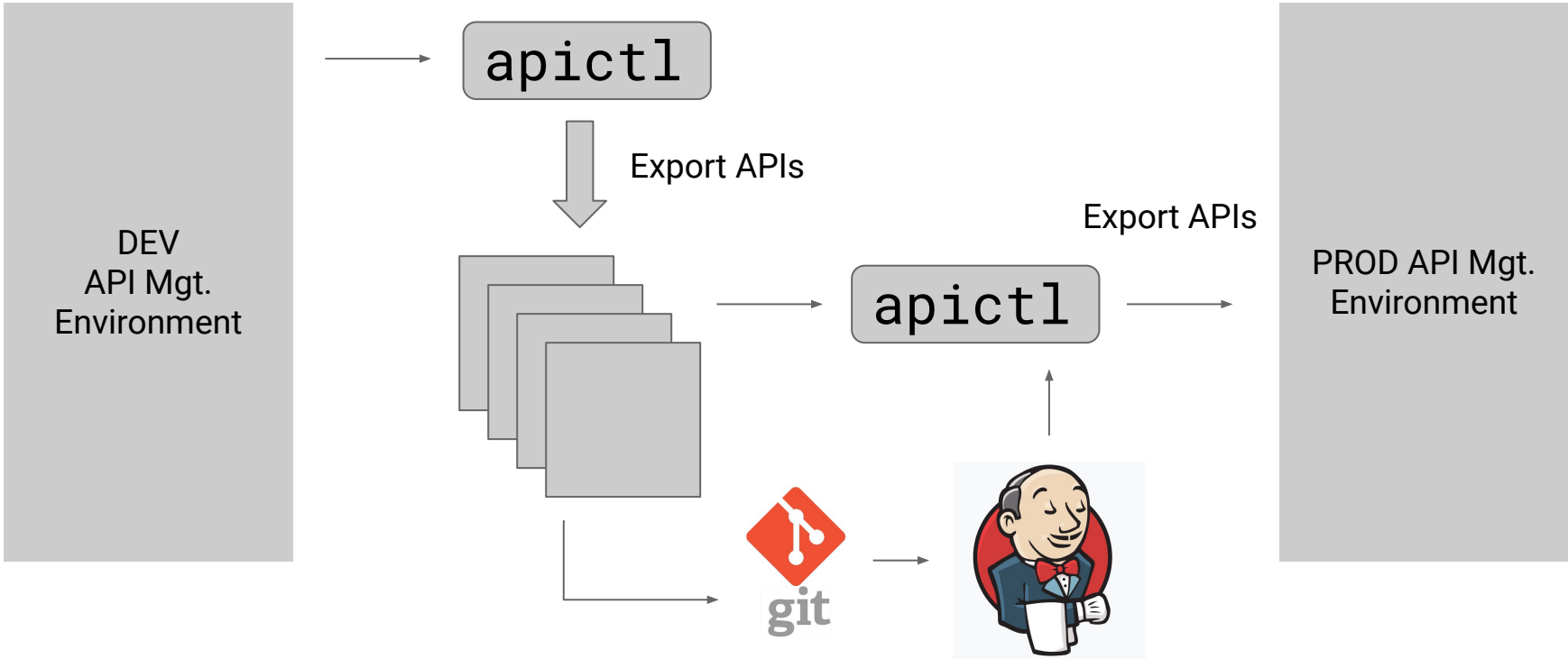


API Management for Istio Microservices



Article - https://hackernoon.com/kubernetes-api-operator-apply-api-management-for-istio-microservices-qs5e3yrq

Part 4 - CI/CD considerations / planning



Part 5 - Kubernetes best practices (some highlights)

- Use WSO2 official docker images (patched and production hardened) to build on top of it for packaging your binaries
- Maintain a private image registry
- Avoid NakedPods in production (not bound to a RS)
- Use officially released helm charts for production deployments
- Centralized log monitoring using a tool like ELK
- Use of probes in Kubernetes - Readiness and Liveness
- Define resource limits in the containers
- Target small container image sizes

THANK YOU

wso2.com