Microservices

Dr. Bambang Purnomosidi D.. P.
Praxis Academy

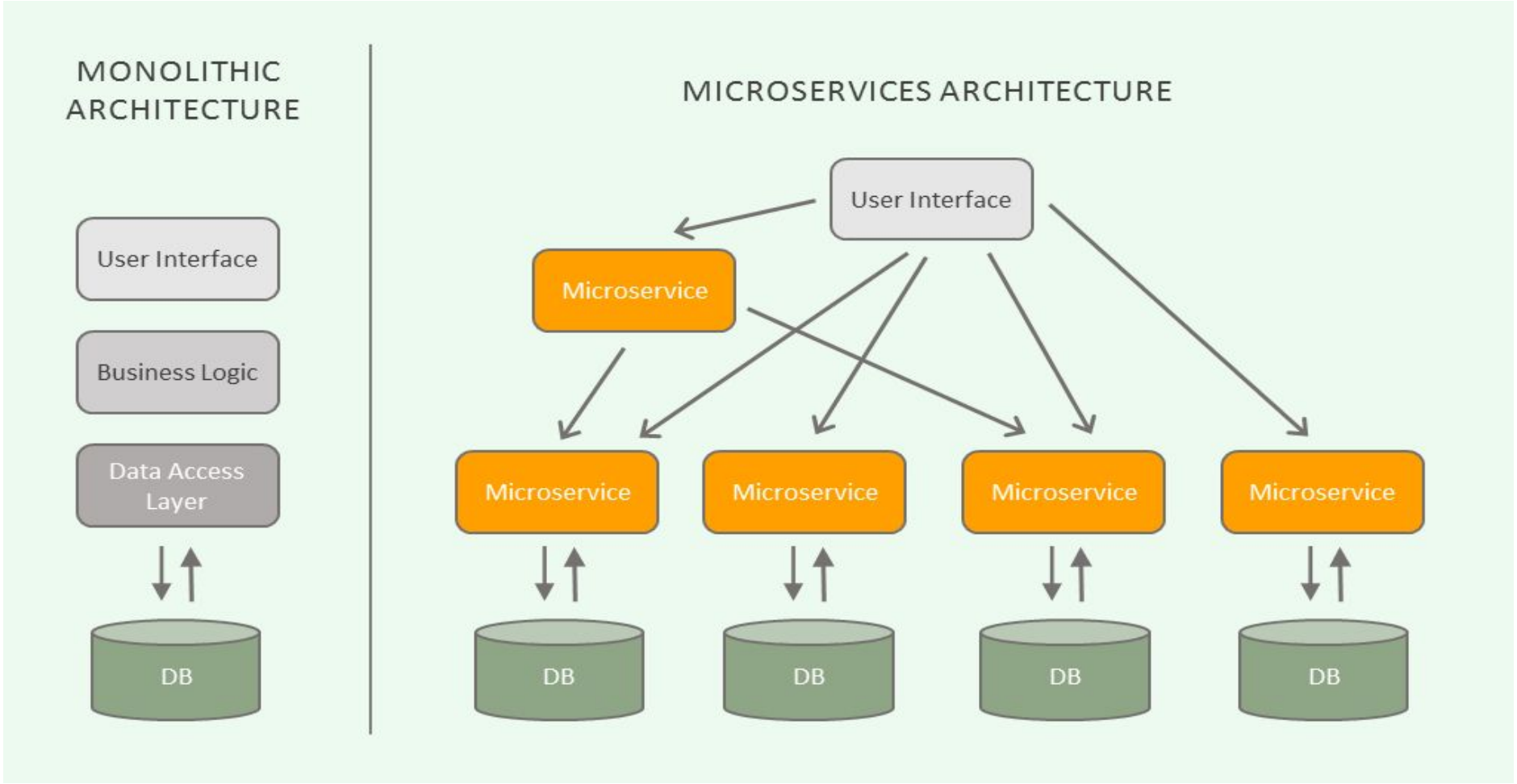
Agenda

1. What Are Microservices?
2. Reasons for Using Microservices
3. Microservices and SOA
4. Designing Microservices
5. Services Development
6. Inter-Service Communication
7. Microservices Integration
8. Microservices Deployment
9. Service Mesh
10. Challenges

What Are Microservices?

An architectural style that structures an application as a collection of services that are:

- Highly maintainable and testable
- Loosely coupled
- Independently deployable
- Organized around business capabilities
- Owned by a small team



Source: DZone (https://dzone.com/articles/what-are-microservices-actually)

Reasons for Using Microservices

Technical and Organizational Benefits:

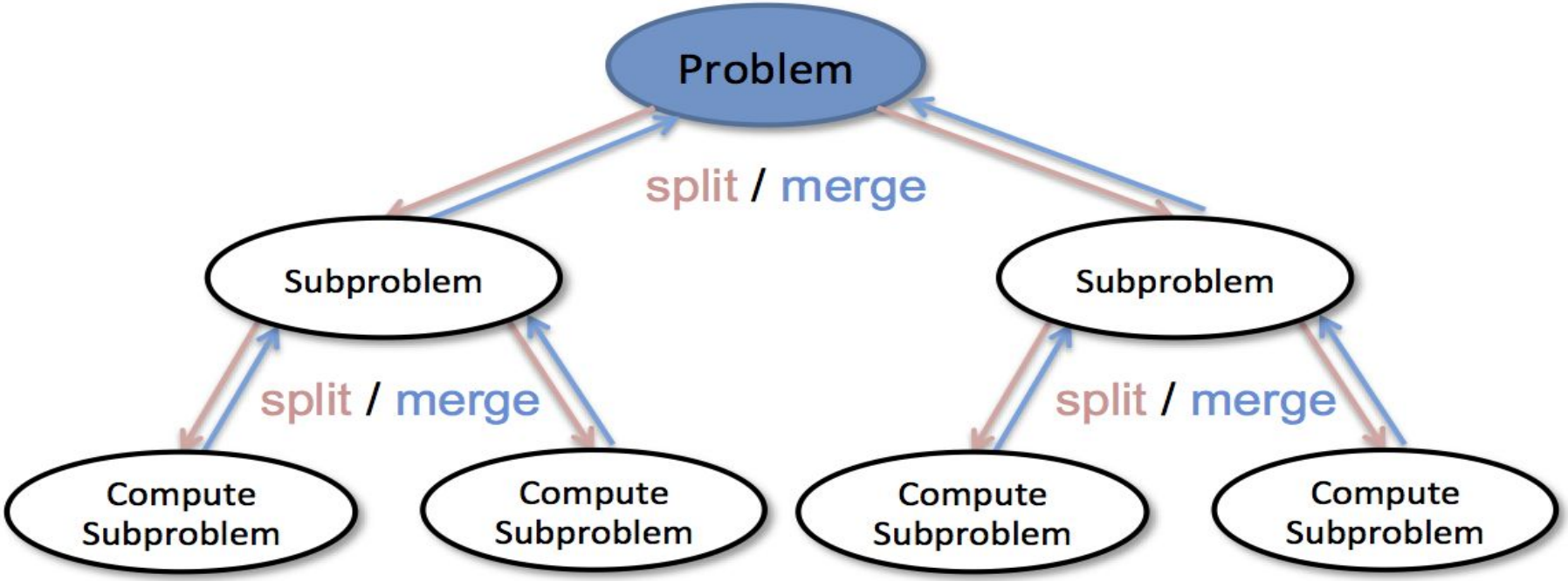
- Loosely-coupled services: easy to be replaced, easy to handle legacy system - lead to sustainable software development
- Continuous Delivery pipeline, from commit - acceptance tests - capacity tests - explorative tests - production
- Scaling
- Robust
- Free technology choice
- Independence but integrated
- Strong modularization
- Parallel development

Microservices and SOA

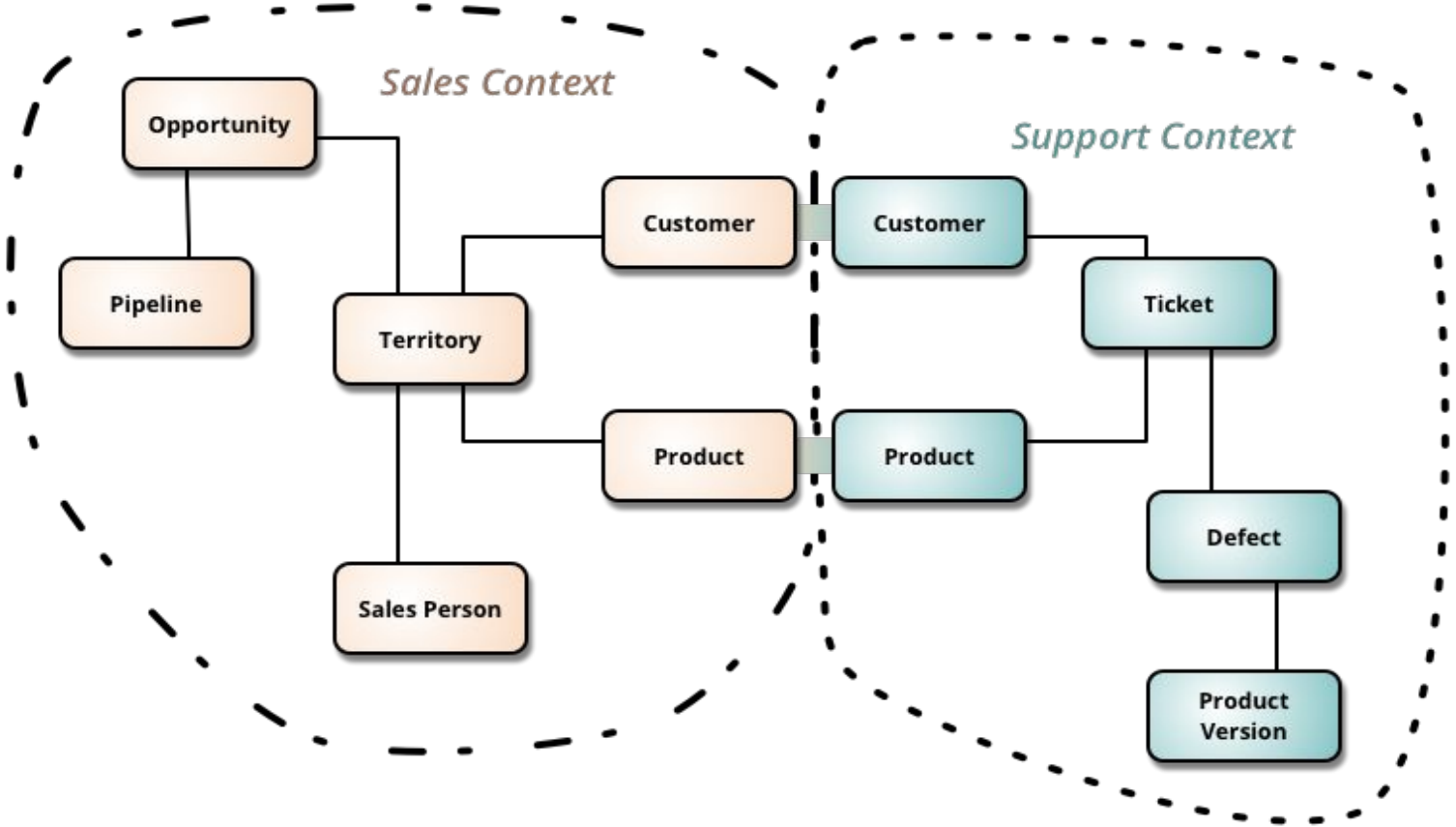
- Conceptually both are the same
- SOA - Web Services - Microservices
- The difference is on the tooling and microservices mantra: “smart endpoints, dumb pipes”.

Designing Microservices

- DDD (Domain Driven Design): divide and conquer: (source: https://bigdata.oden.utexas.edu/project/divide-conquer-methods-for-big-data-analytics/)



- Bounded Context (source:https://martinfowler.com/bliki/BoundedContext.html)



Services Development

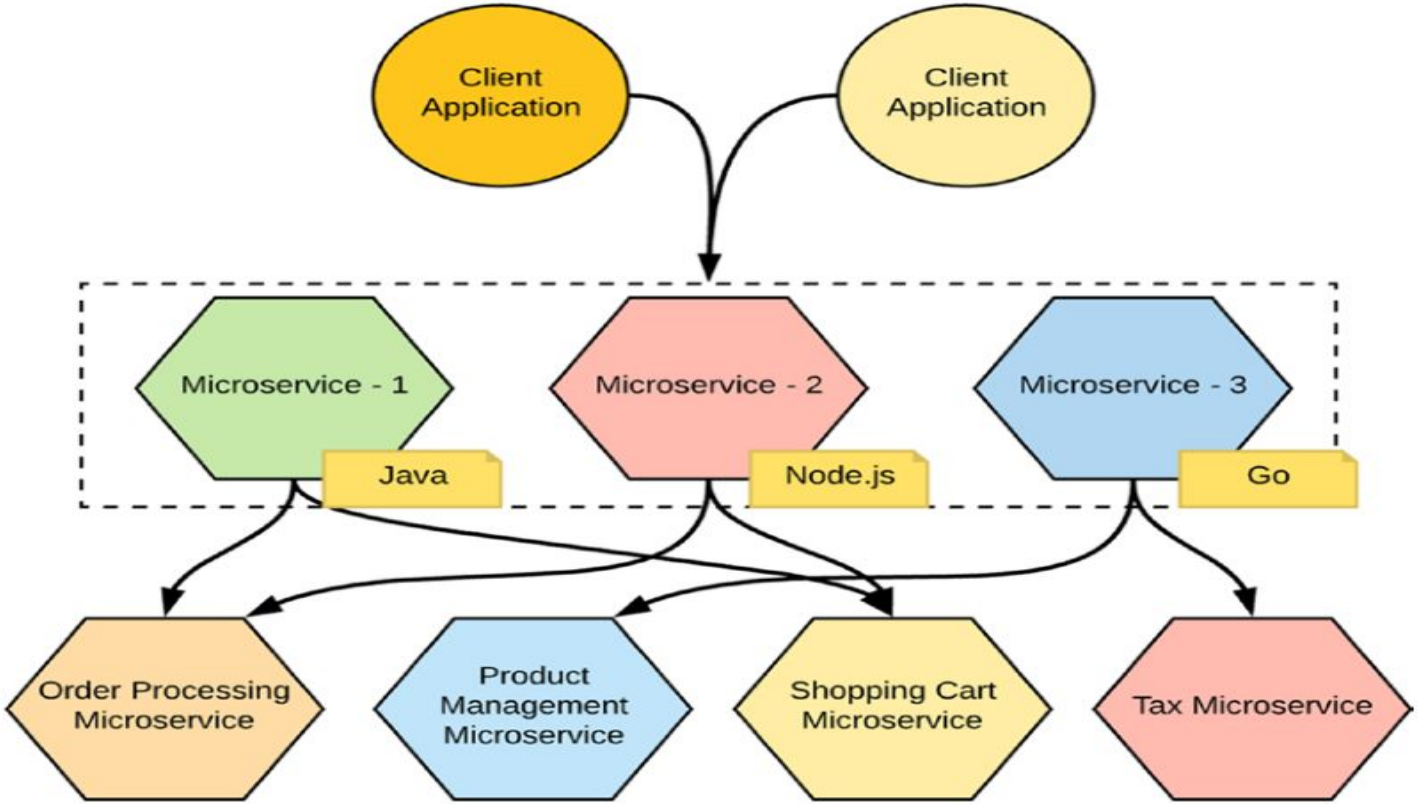
- Development tools and frameworks:
  - Micronaut
  - DropWizard
  - Spring Boot
  - Vert.x

Inter-Service Communication

- HTTP-based
  - REST
  - GraphQL
  - gRPC
- Messaging: Apache Kafka, AMQP

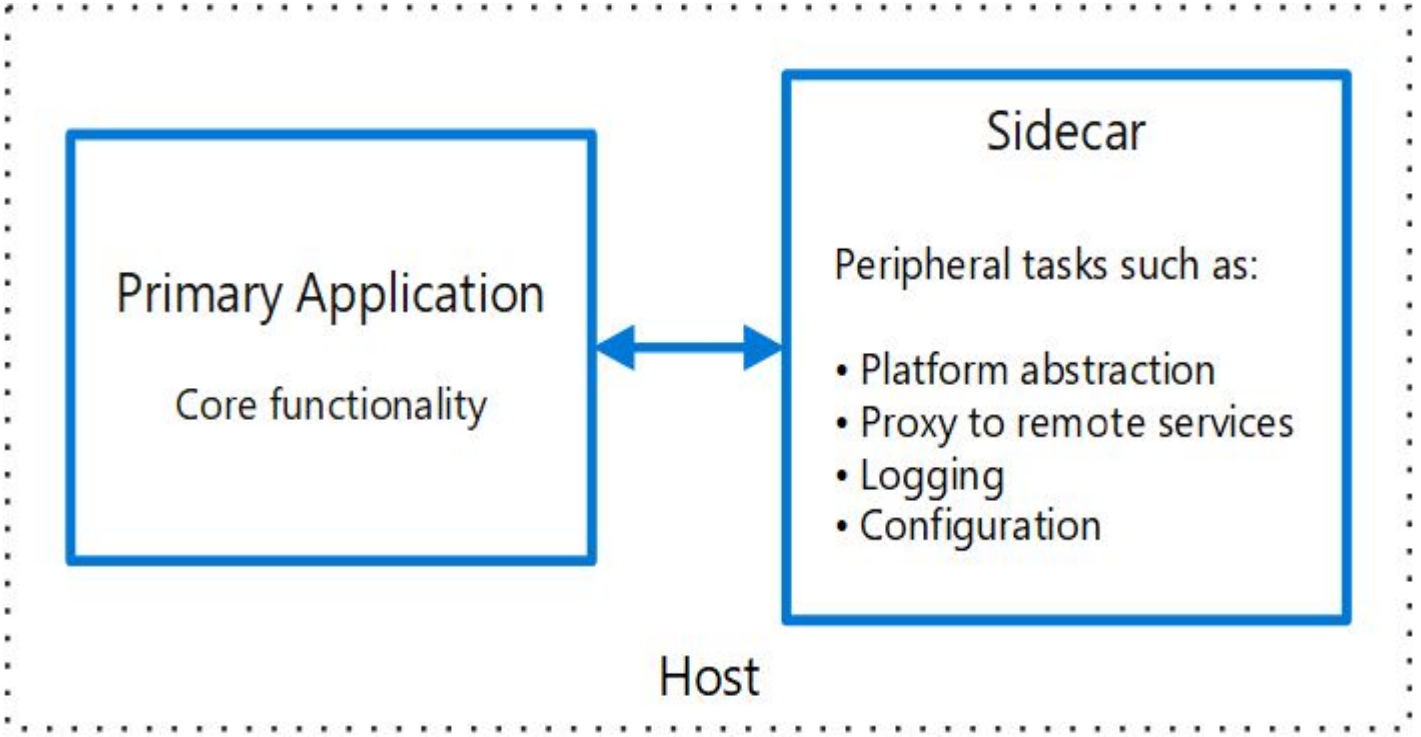
Microservices Integration

Smart endpoints, dumb pipes

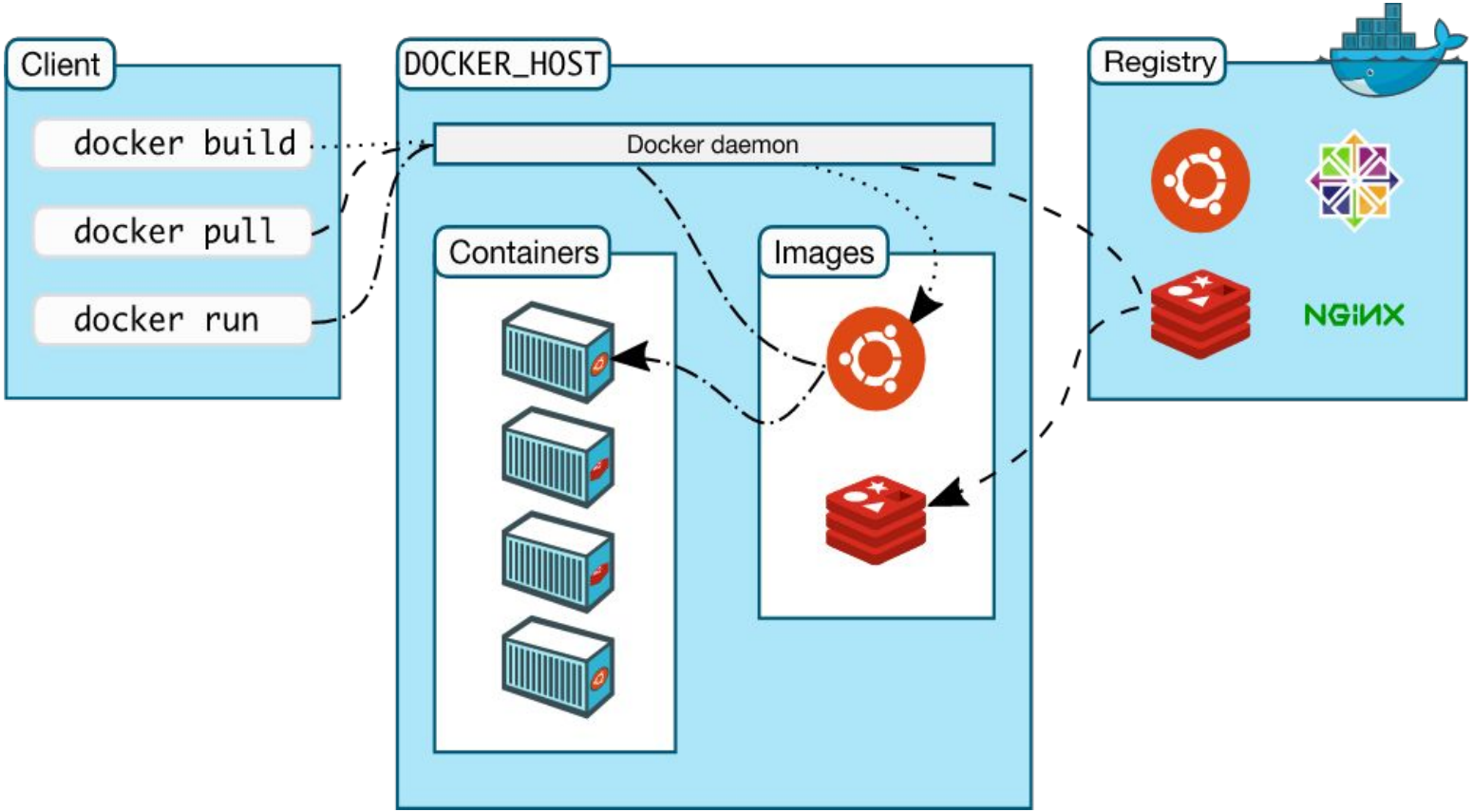


Microservices Deployment

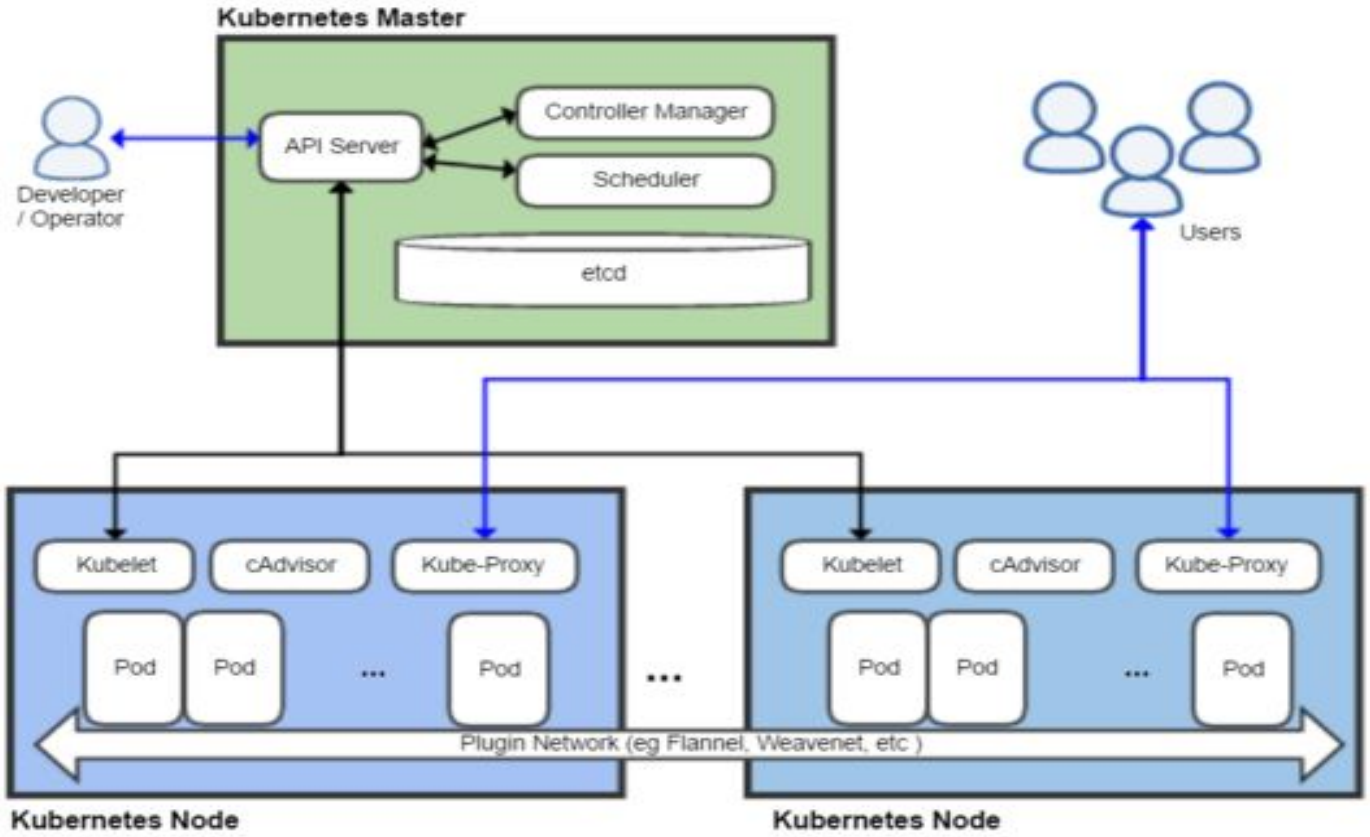
Service Mesh: inter-service communication infrastructure using **sidecar pattern**.



Container technology: docker (images, containers, management, orchestration) and kubernetes (management and orchestration).



Kubernetes


Kubernetes Master
Developer / Operator
API Server
Controller Manager
Scheduler
etcd
Users
Kubelet
cAdvisor
Kube-Proxy
Pod
Pod
...
Pod
Kubelet
cAdvisor
Kube-Proxy
Pod
Pod
...
Pod
Plugin Network (eg Flannel, Weavenet, etc)
Kubernetes Node
Kubernetes Node

More?

Enterprise Full Stack Application Developer program at **Praxis Academy**

https://s.id/56lzy.