

# API Security Challenges for Cloud-Native Architects

---

 Pierre Versali

 [pierre-versali.bitbucket.io](https://pierre-versali.bitbucket.io)

APISEC  
CON

apidays



# Pierre Versali

Cloud-Native Software Architect  
Principal Consultant | Team Coach



Agenda



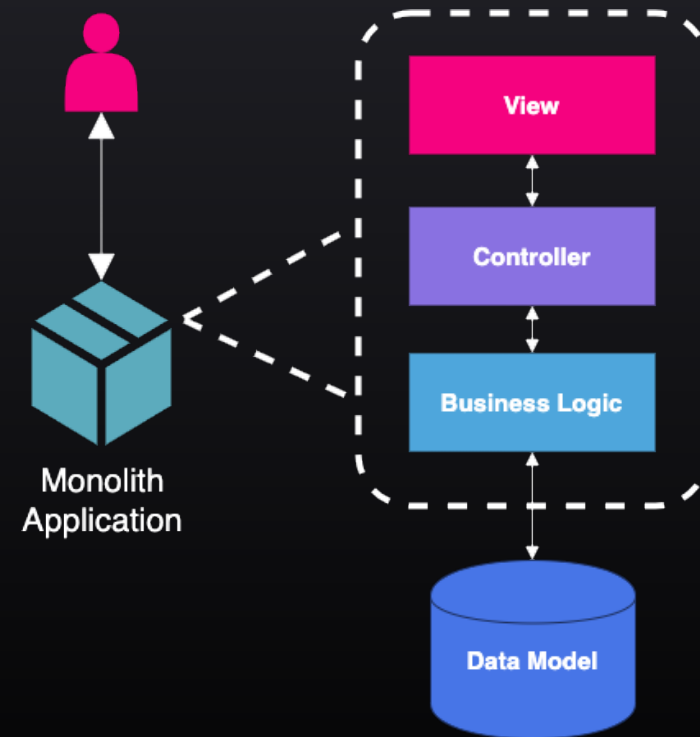
Software **Architecture**

**Cloud-Native** Software

**API Security** Challenges

# Software Architecture

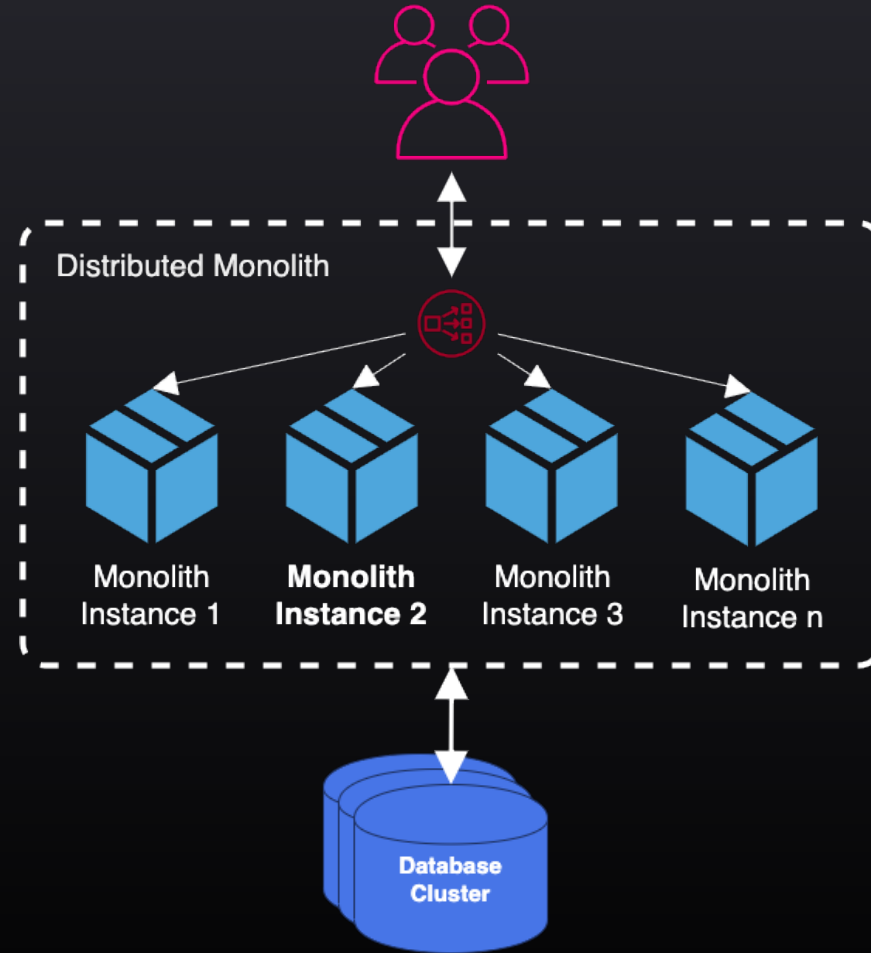
## Monolith Application



# Software Architecture



## Monolith Application **Distributed Monoliths**

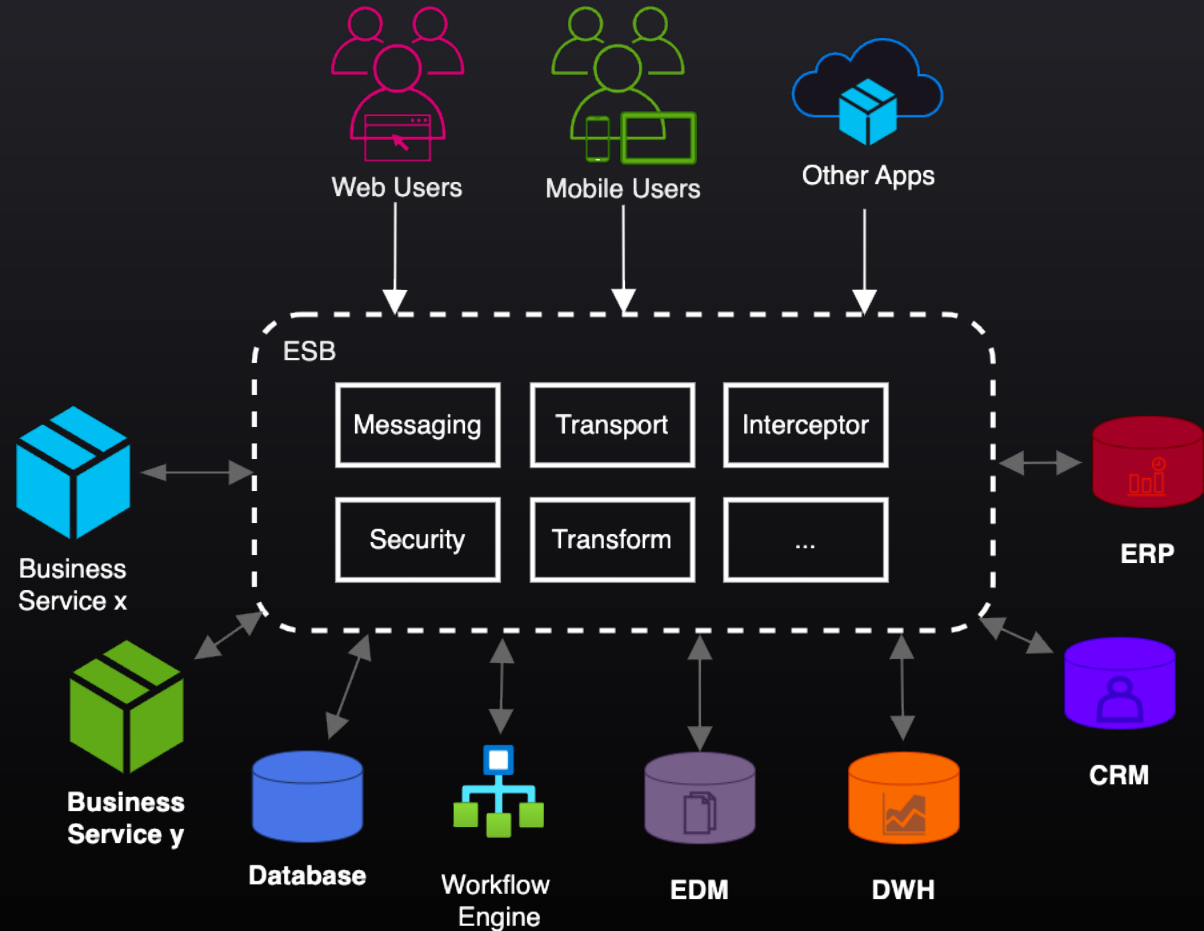


# Software Architecture

Monolith

Distributed Monoliths

**Service Oriented Architecture**



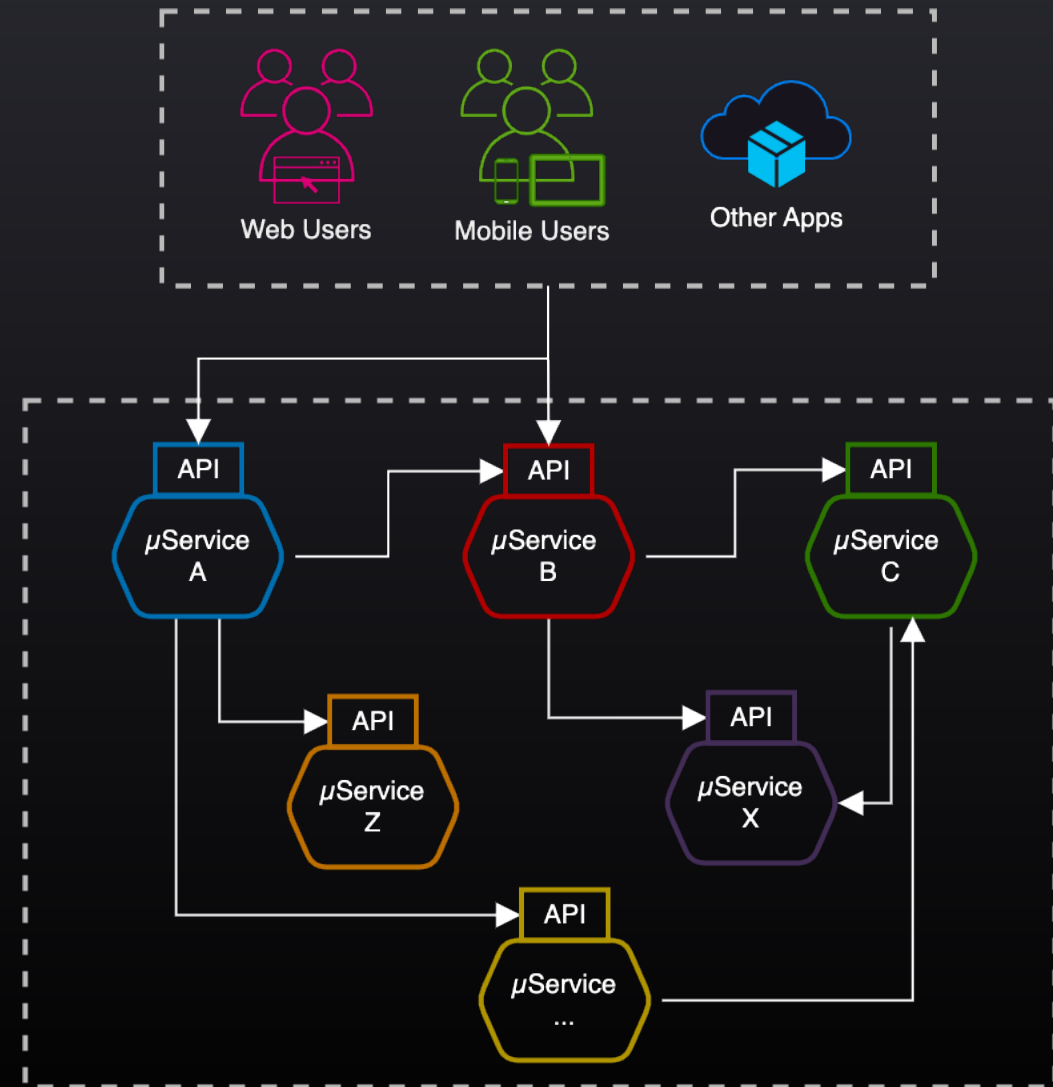
# Software Architecture

Monolith

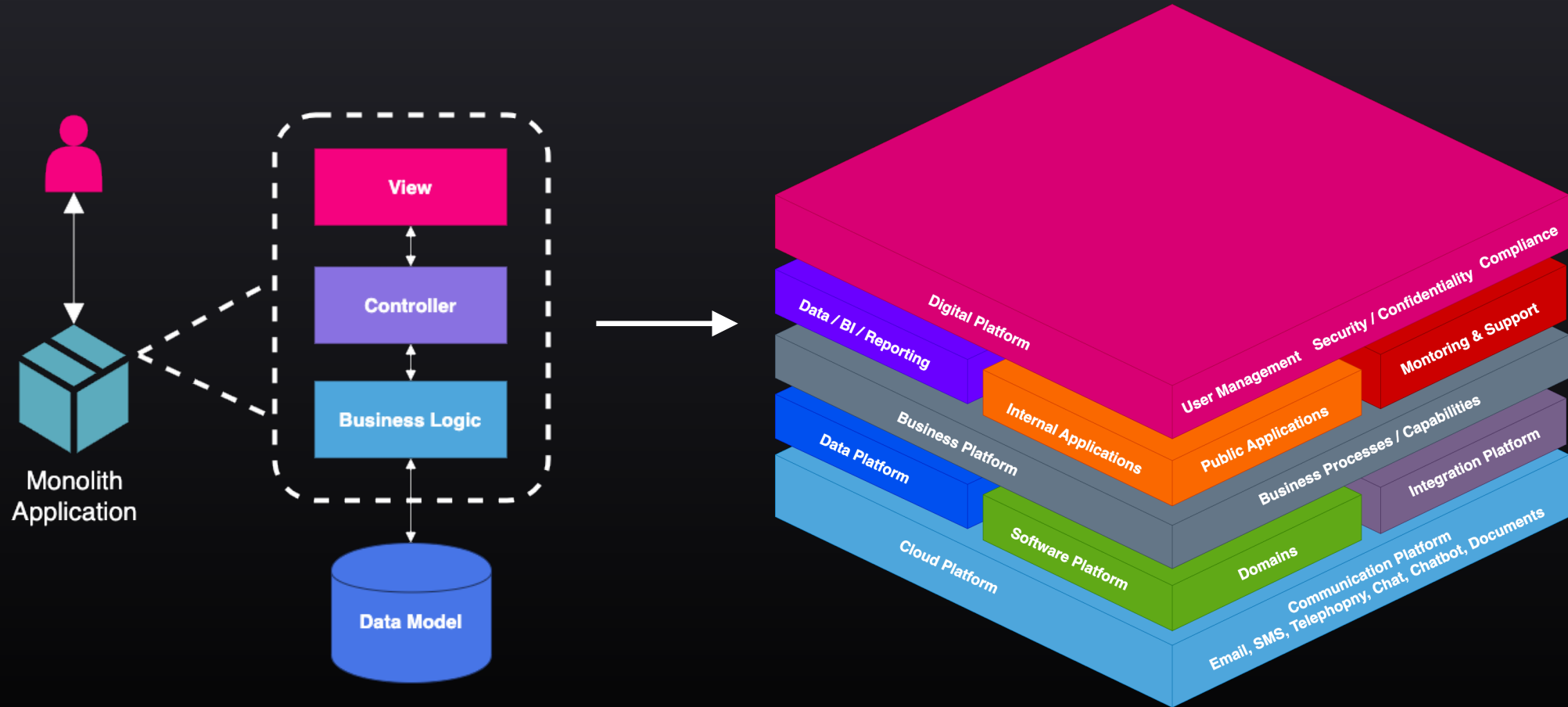
Distributed Monoliths

Service Oriented Architecture

**Microservices**

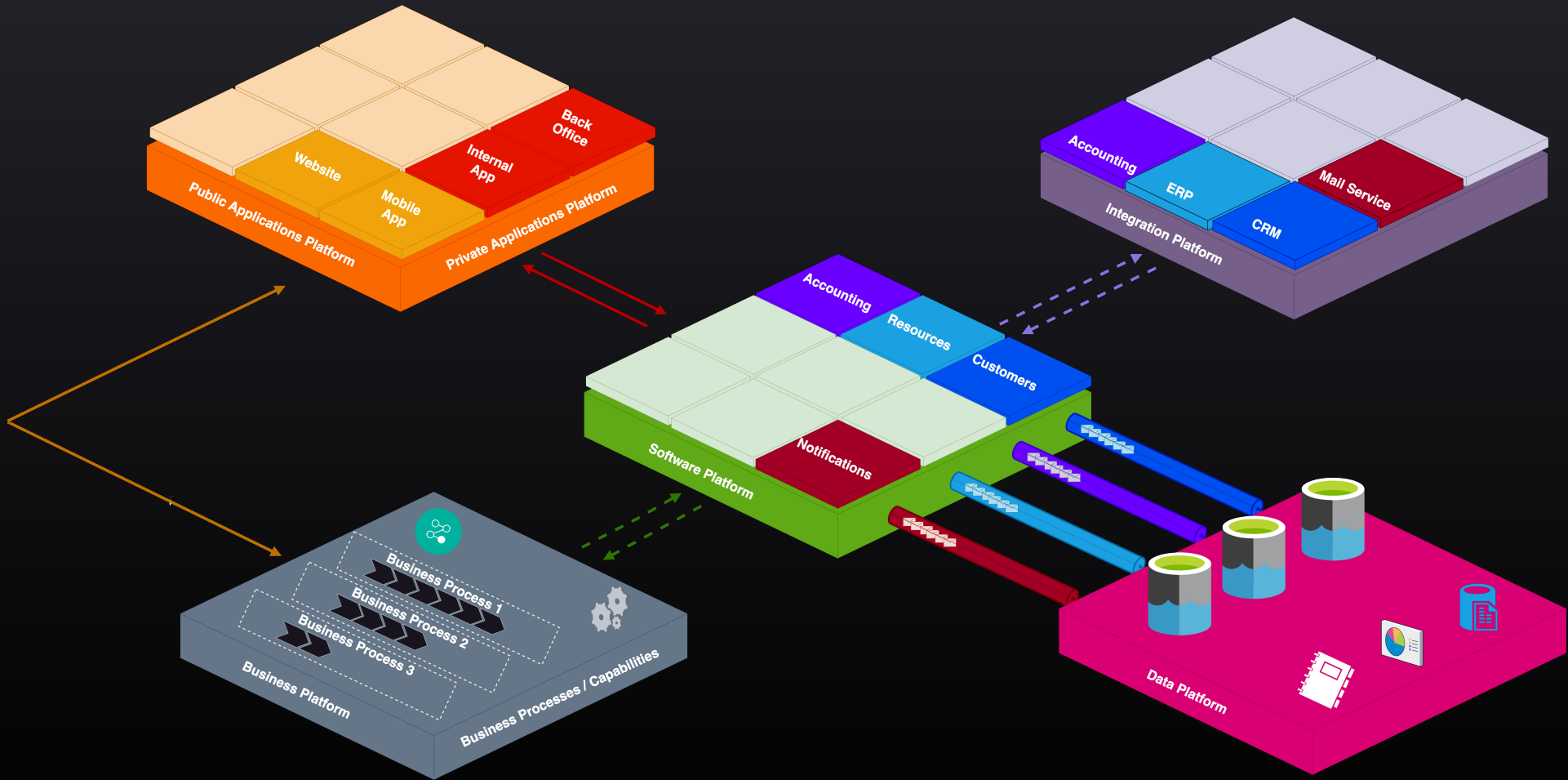


# From Monolith Application to Digital Platforms

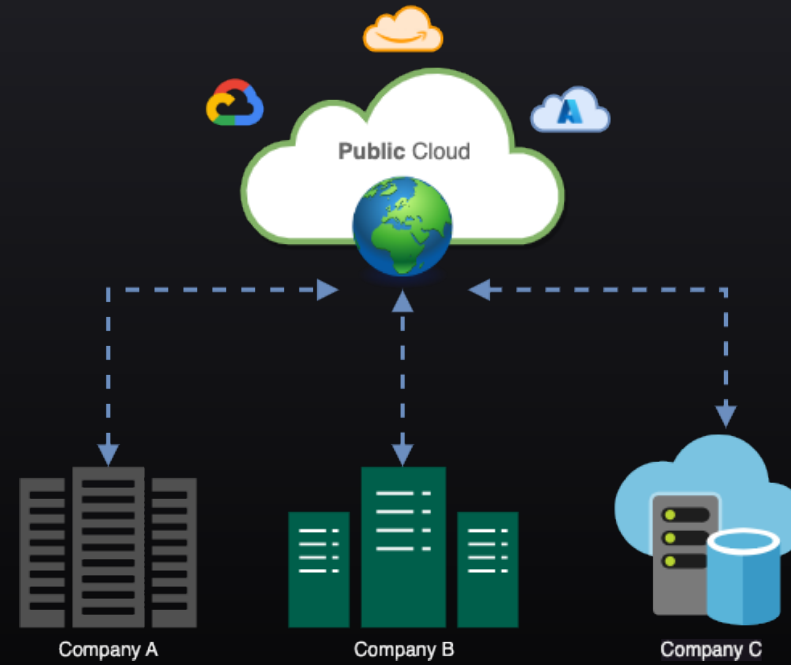




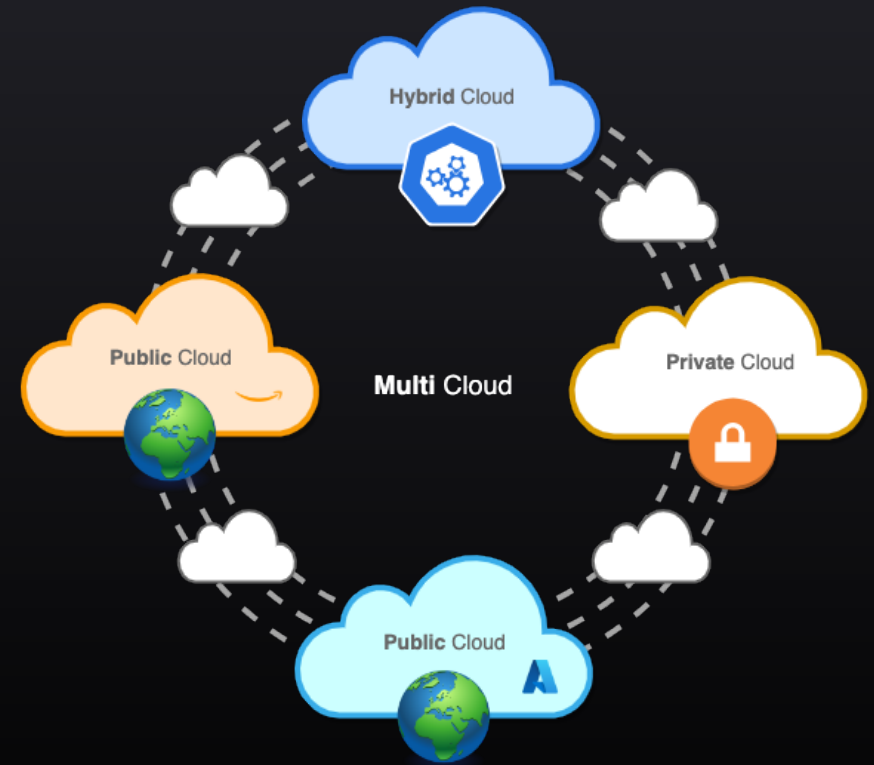
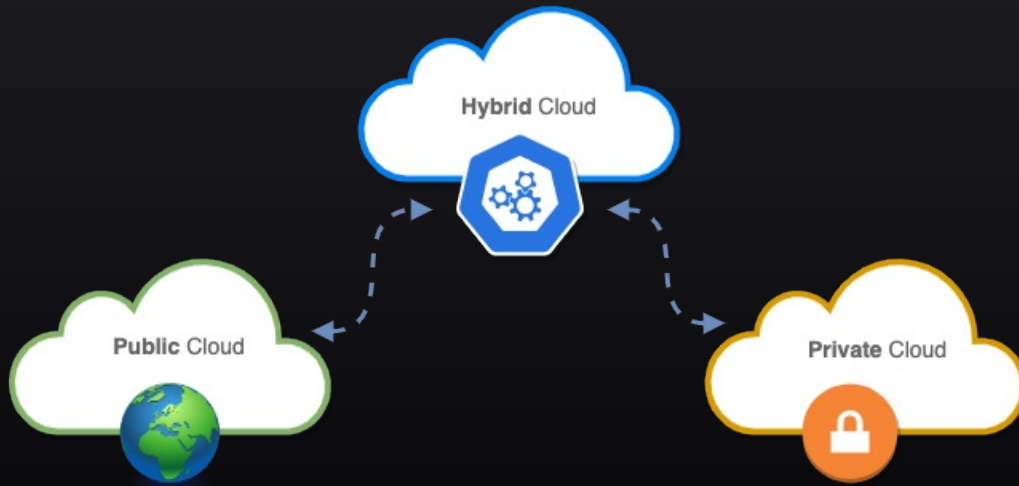
# Digital Platform – Data Flows



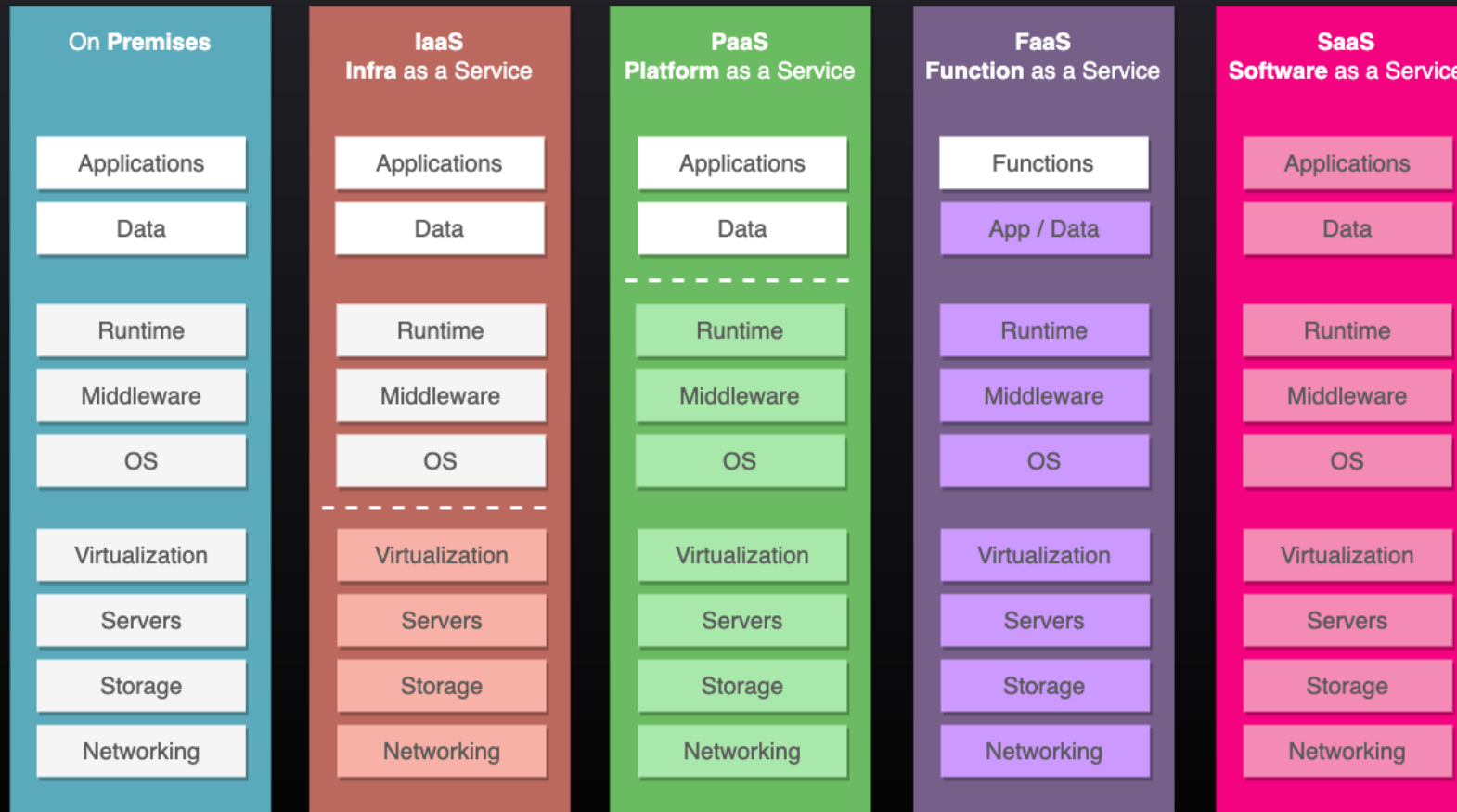
# Cloud Models



# Cloud Models



# Evolution of Delivery Models



## Benefits of Cloud



**Strategic**



**Efficient**



**Secure**



**Flexible**



**Cost-effective**

# Cloud-native Software



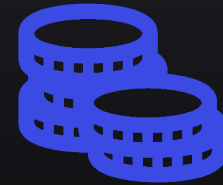
# Cloud-native Software



**HIGHLY-AVAILABLE**

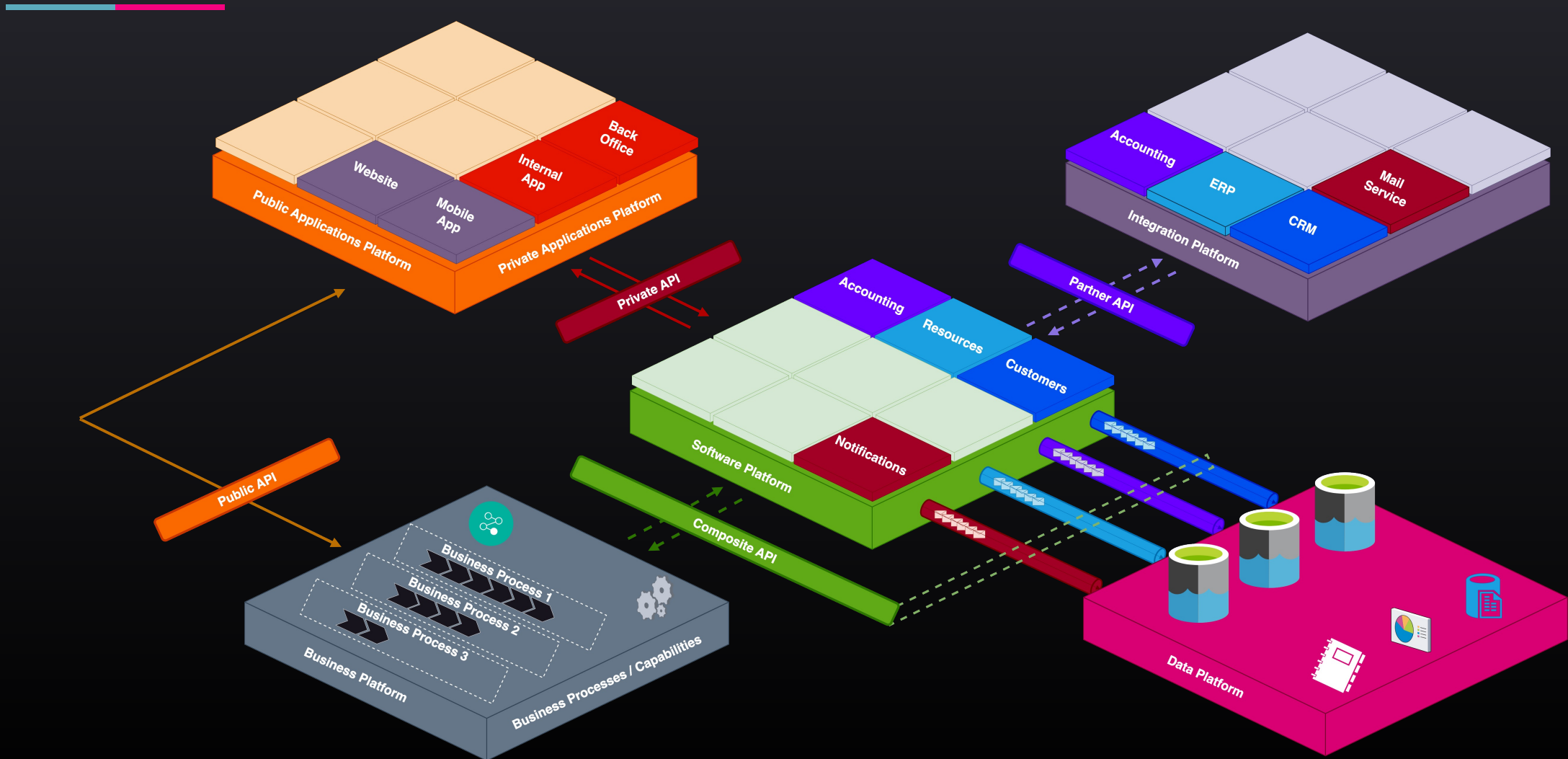


**SCALABLE**



**COST-EFFECTIVE**

# API – Application Programming Interface





## API Benefits

- ✓ Loose **Coupling**
- ✓ **Integration**
- ✓ **Collaboration**
- ✓ **Standardization**
- ✓ Developer **Experience**
- ✓ **Testability**



## API – Protocols | Message Formats | Specification

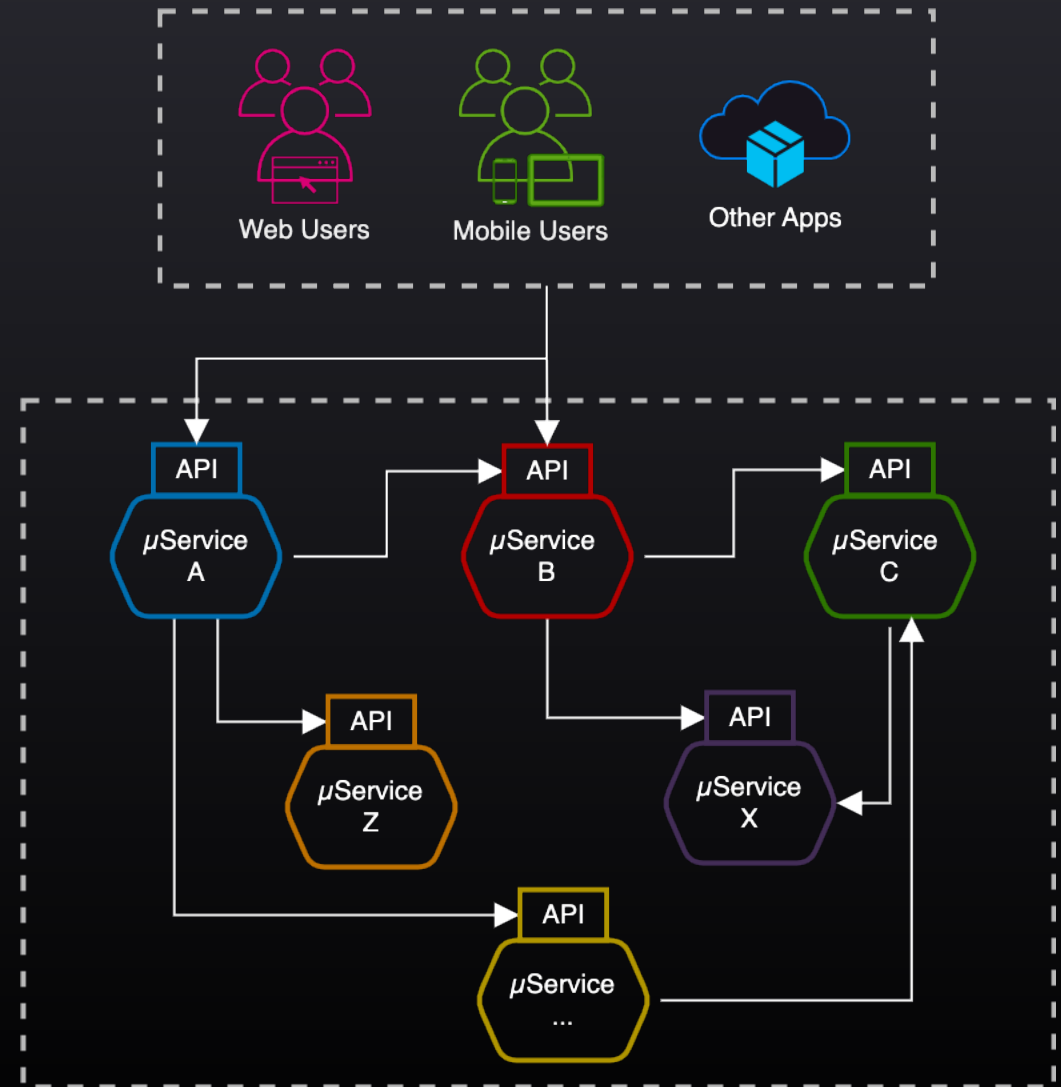
API	Protocol	Message Format	Specification
Synchronous	SOAP	XML	SOAP
	<b>REST over HTTP</b>	<b>JSON</b>	<b>OpenAPI</b>
	GraphQL	GraphQL	GraphQL
	gRPC over HTTP/2	Protobuf	gRPC
Asynchronous	Event Broker Pub / Sub Kafka / MQTT	JSON Protobuf Avro Thrift	AsyncAPI
	WebSockets		

Standards and Best Practices

## Microservices **Benefits**

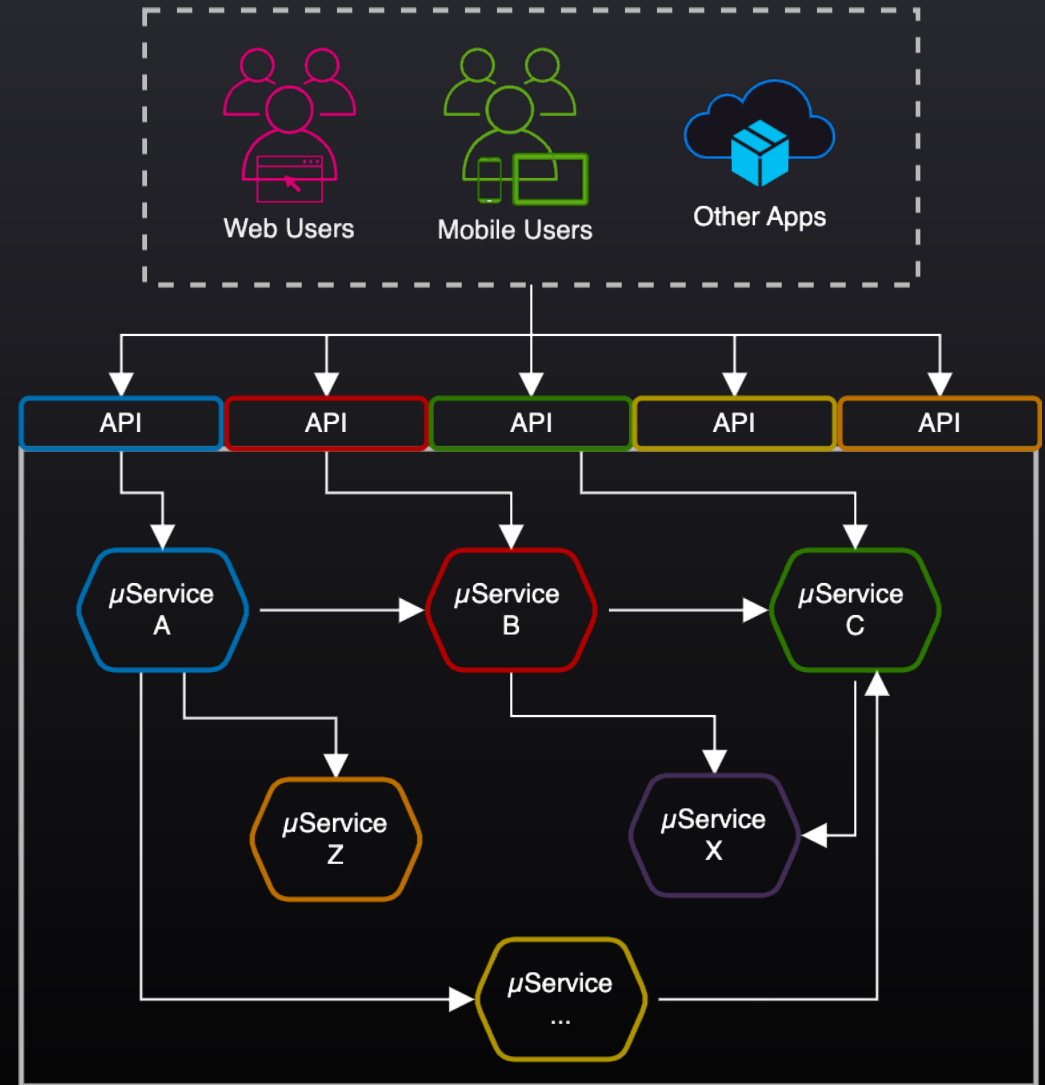


- ✓ Separation of **Concern**
- ✓ Diversity in **Technology Stack**
- ✓ **Isolation**
- ✓ **Reusability**
- ✓ **Flexibility / Scalability**
- ✓ **Reliability**

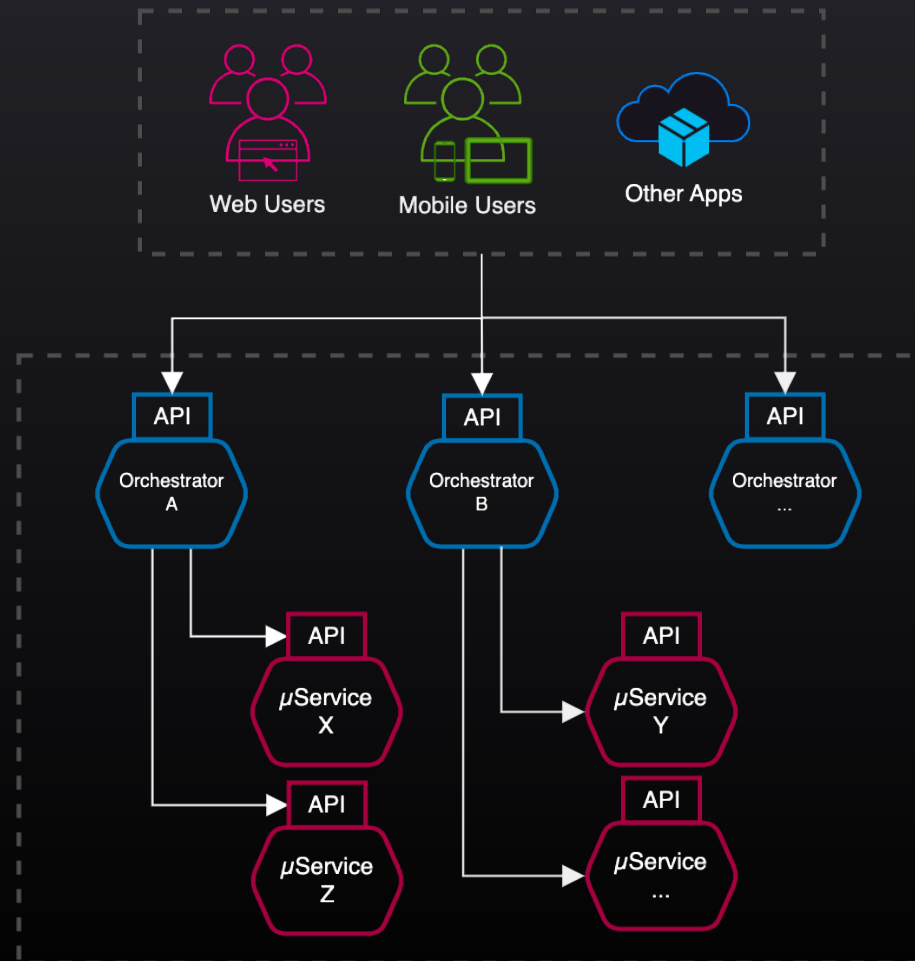


## Microservices concerns

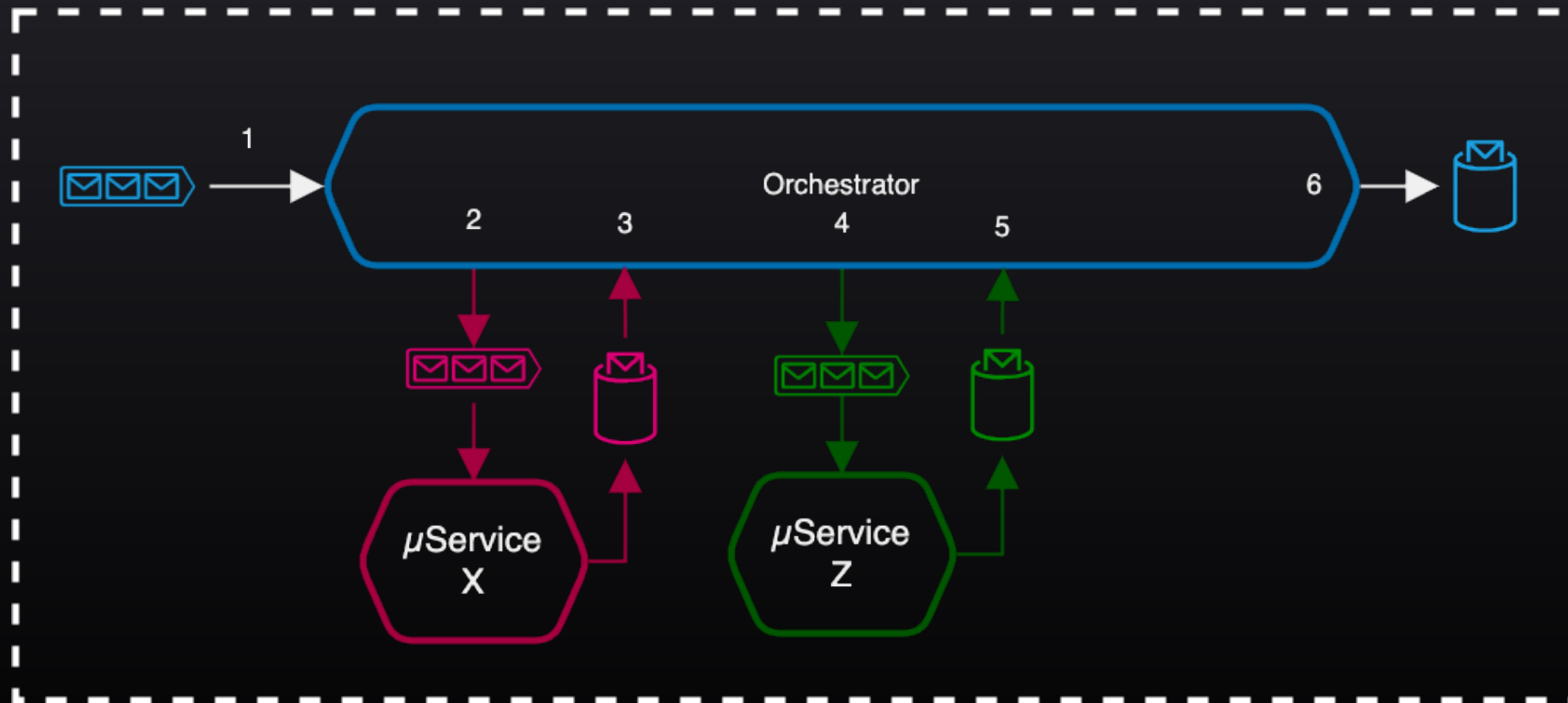
- Complexity
- Security
- Performance
- Evolutivity
- Deployment
- Data Consistency
- Resilience
- Fault Tolerance
- ...



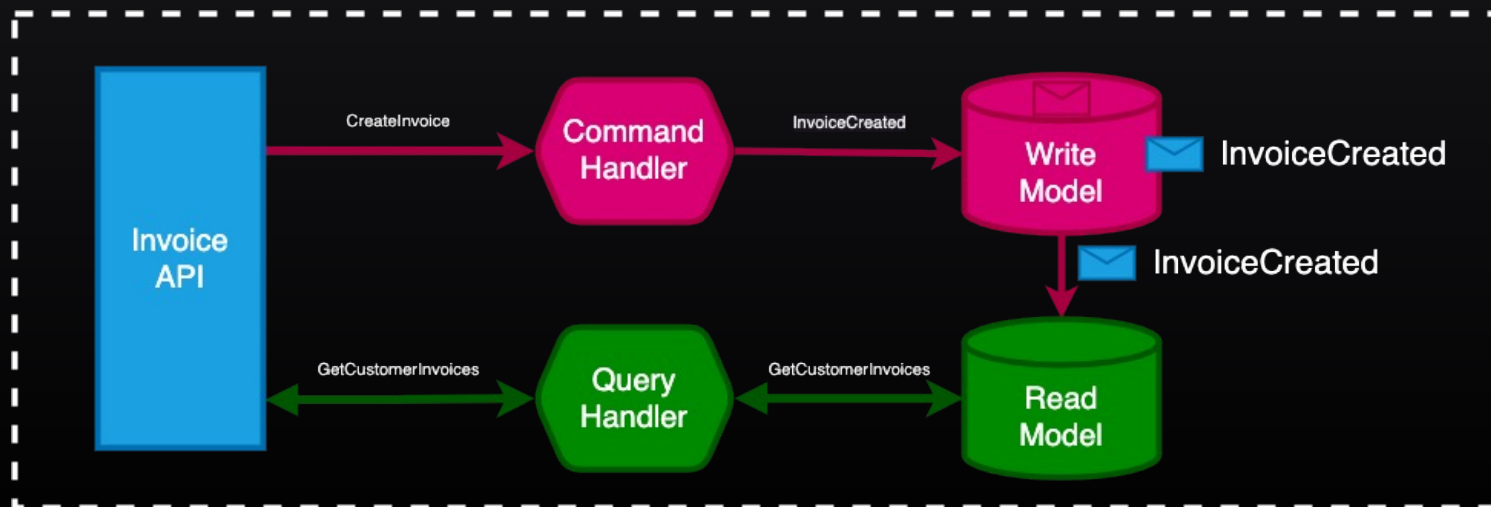
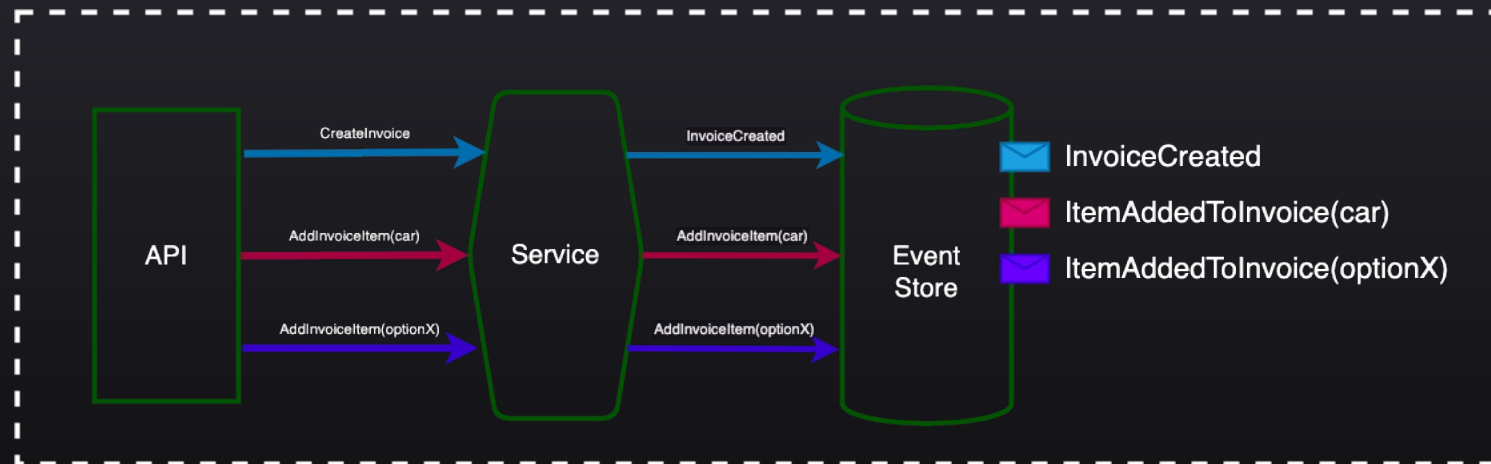
# Service Orchestration



# Reactive Programming / Message-Driven



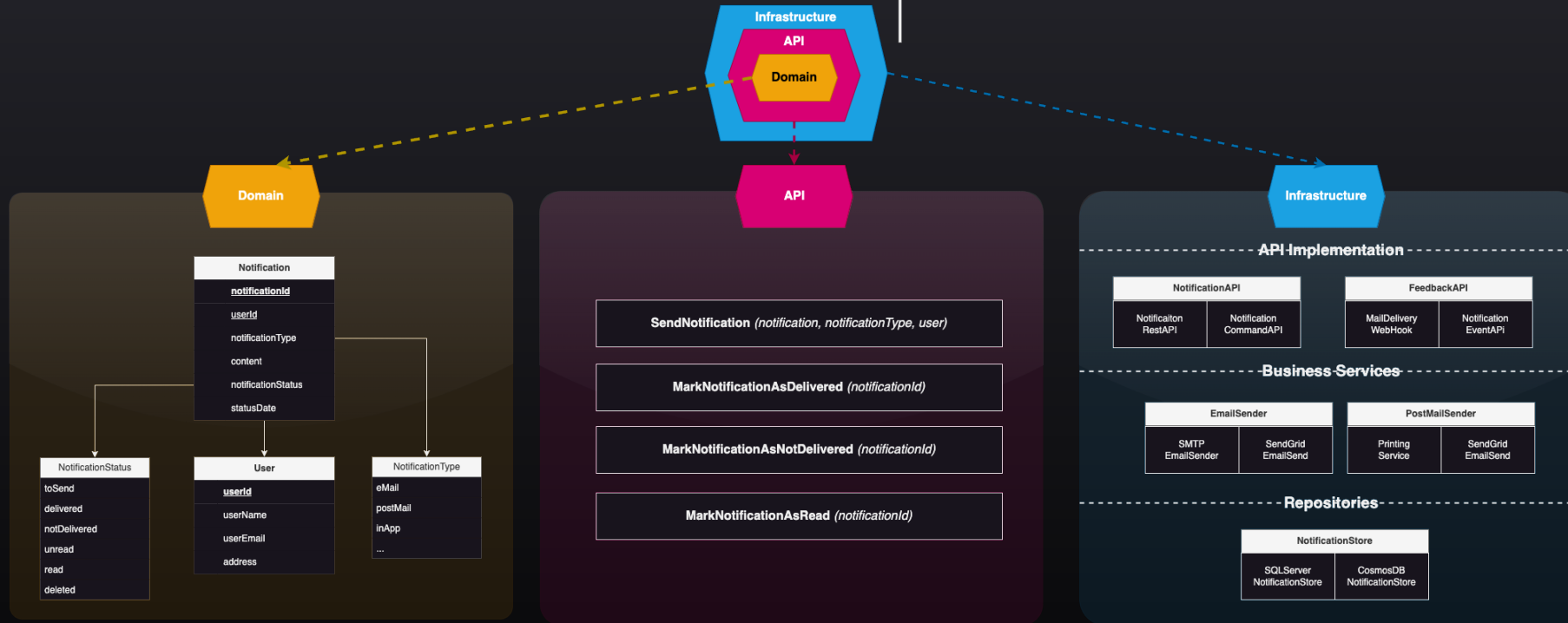
# Event Sourcing & CQRS



# Domain-Driven Design



Microservice  
Packaged and deployed independently  
*Isolation and scalability*



**Ubiquitous Language**  
Business Objects & Business Operations

**Bounded Context**  
*Independent from all other business concepts in the system*  
Loose coupling

**No technical concern here!!!**

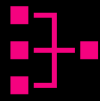
**Services abstractions & Implementations**

**Abstracts services from their implementation(s)**  
*Easier to maintain and evolve*

**Dedicated Data Store**  
*Improves isolation and scalability*



## Microservices **problems**



### **API Conversation Pattern**

Synch  
Graph  
Async  
Messaging (Pub/Sub)



### **Processes Data Consistency**

Eventual Consistency  
Choreography / Orchestration  
Event-Sourcing  
CQRS



### **Fault-Tolerance**

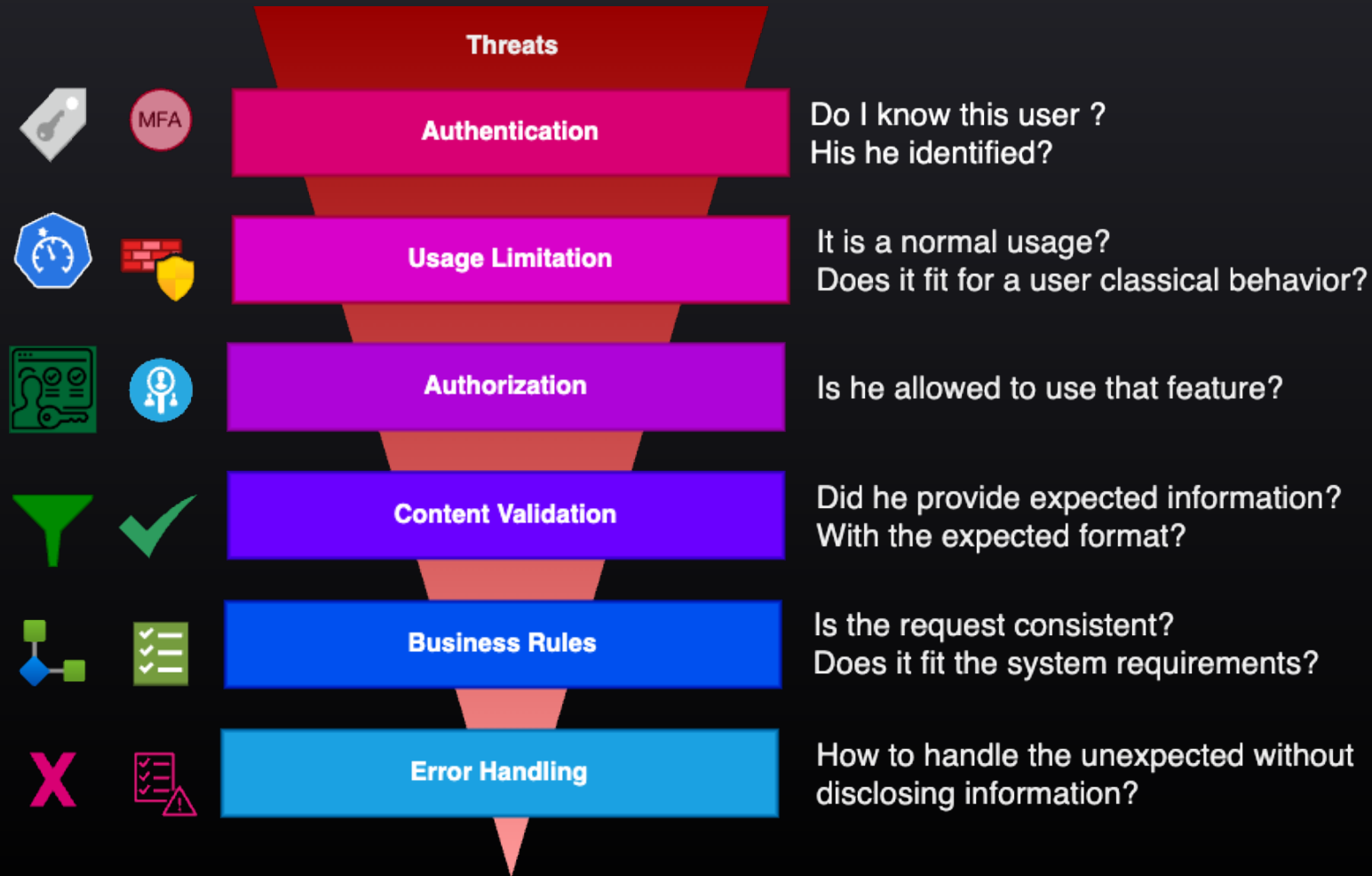
Process Management  
State Management  
Retry / Rollback



### **Data storage**

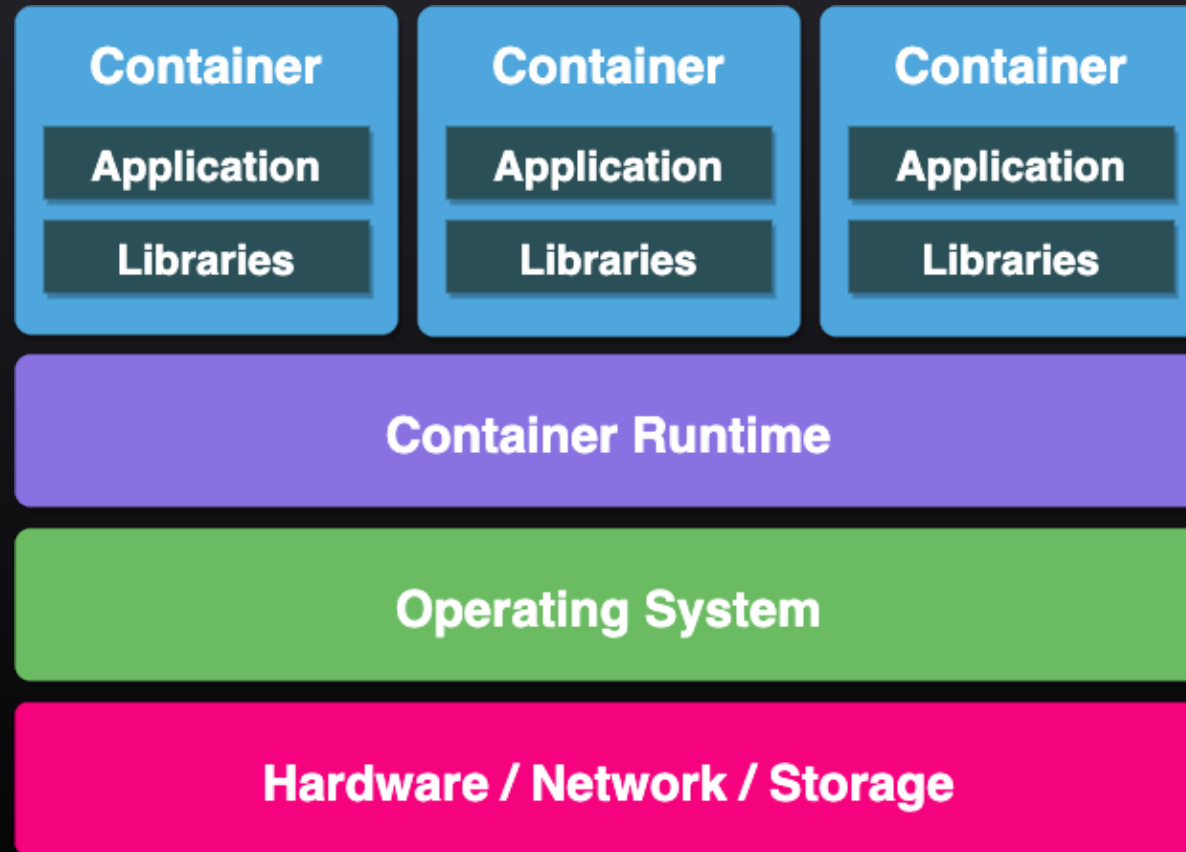
Relational Data  
Key-Value  
Event-Driven

# Identity and Access Management

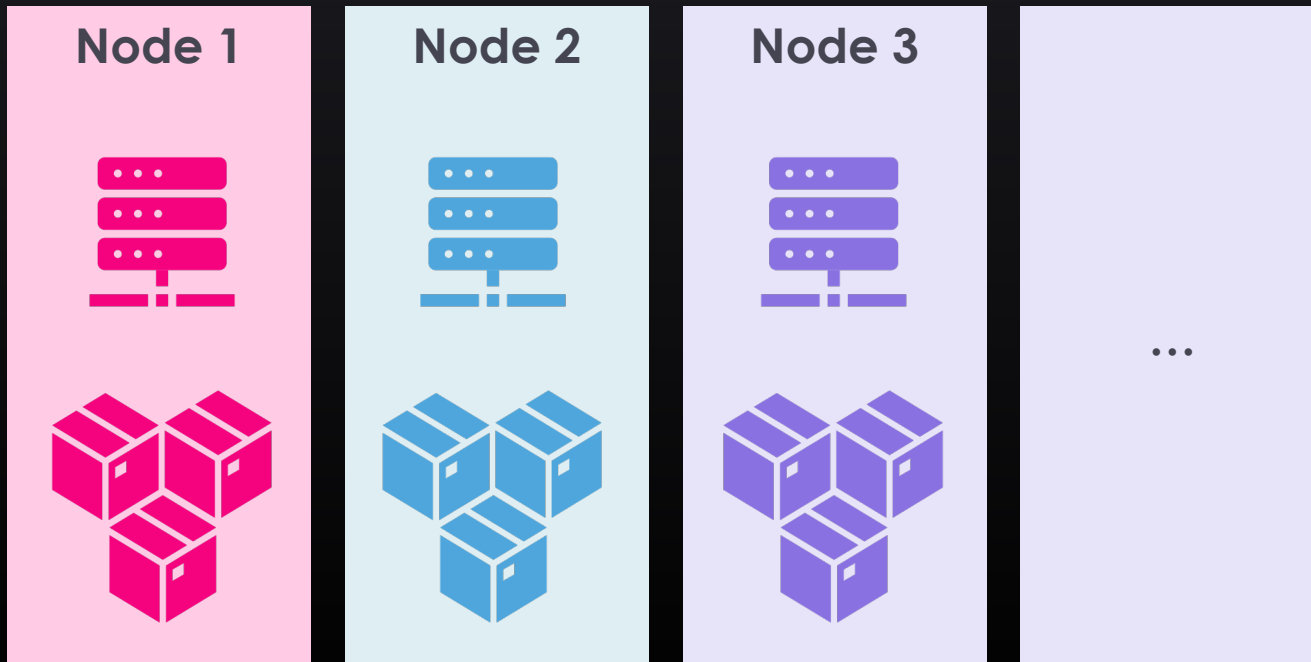







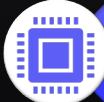


# Containers

---

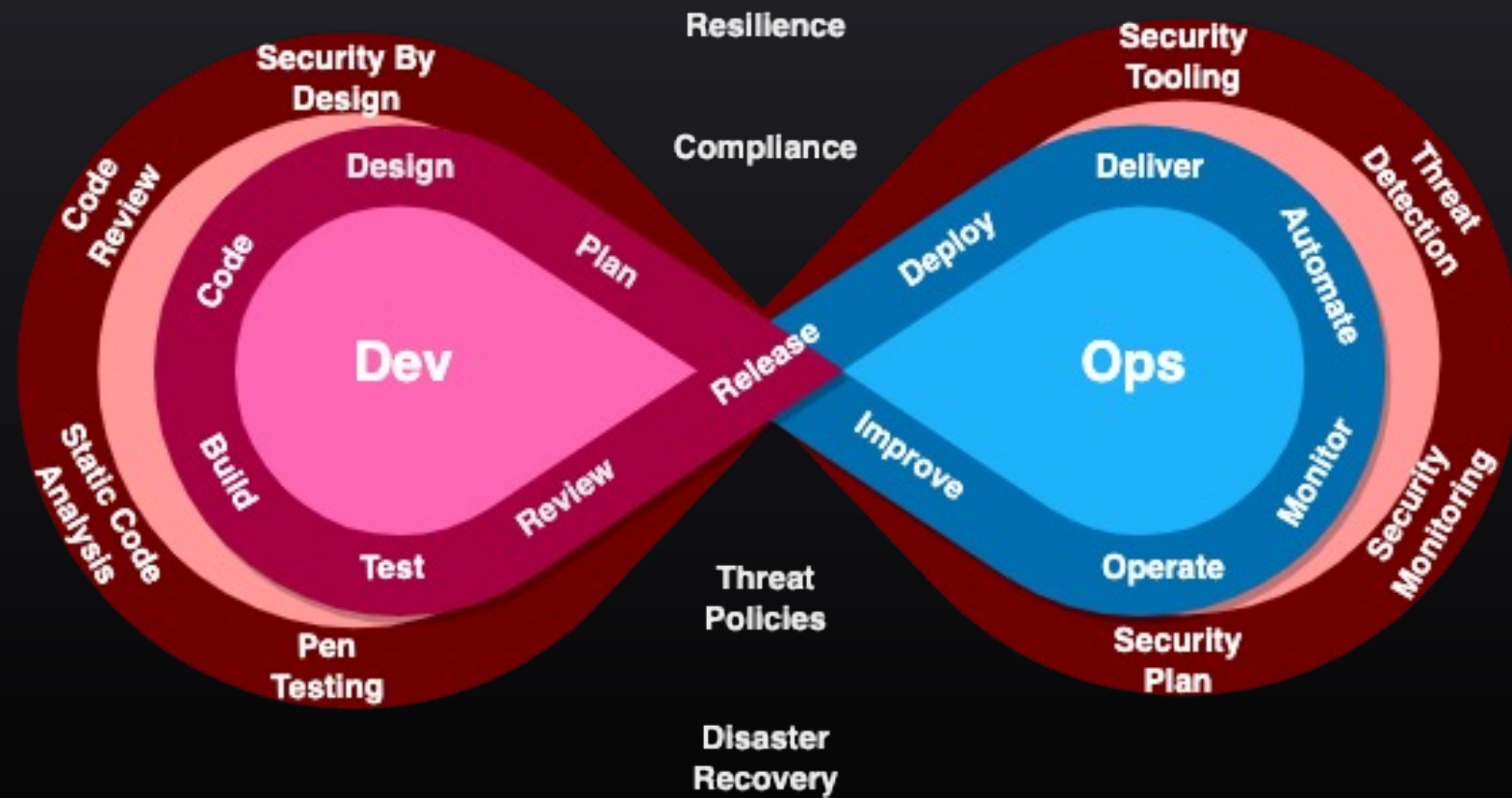
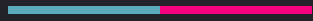


# Container Orchestration

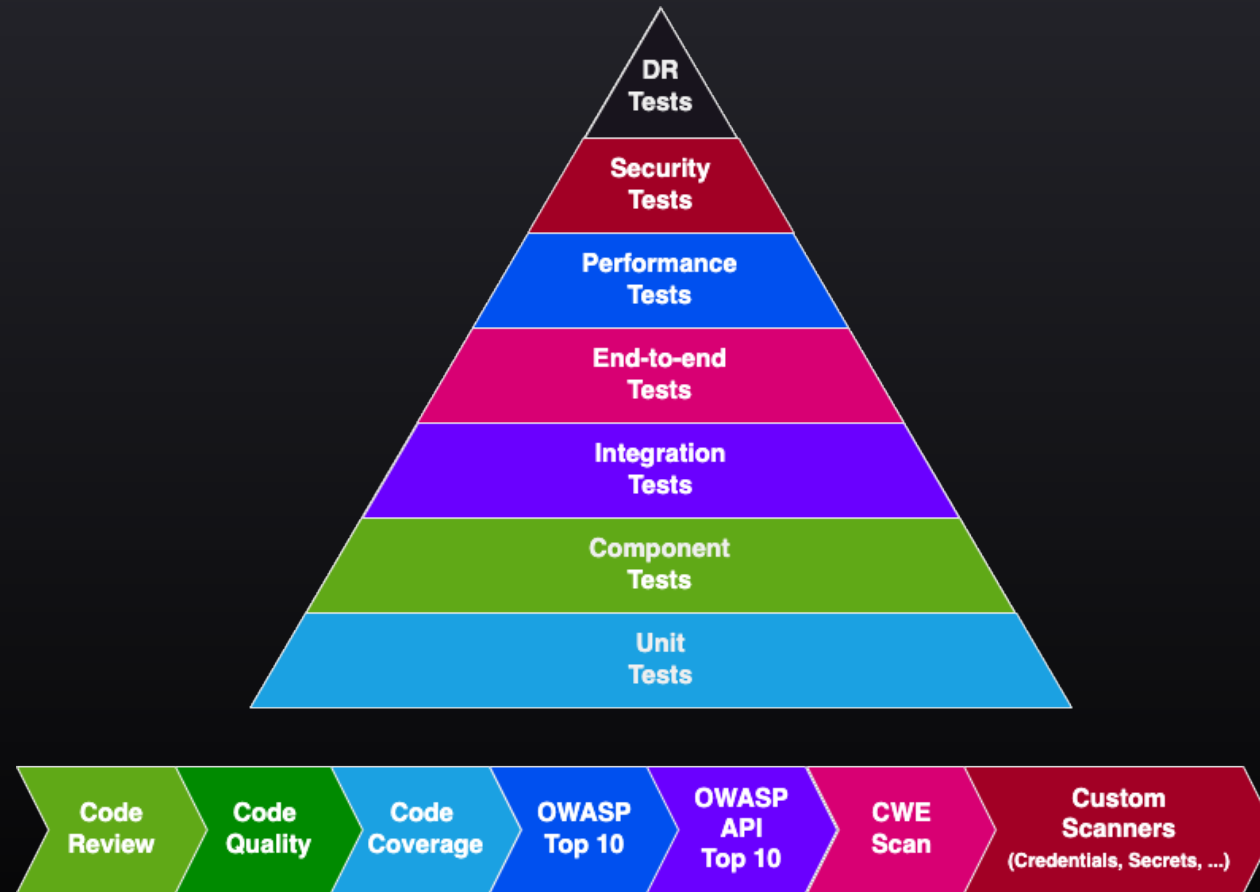


-  Configuration
-  Availability
-  Provisioning
-  Scaling
-  Automation
-  Resource Allocation
-  Load Balancing
-  Health Monitoring

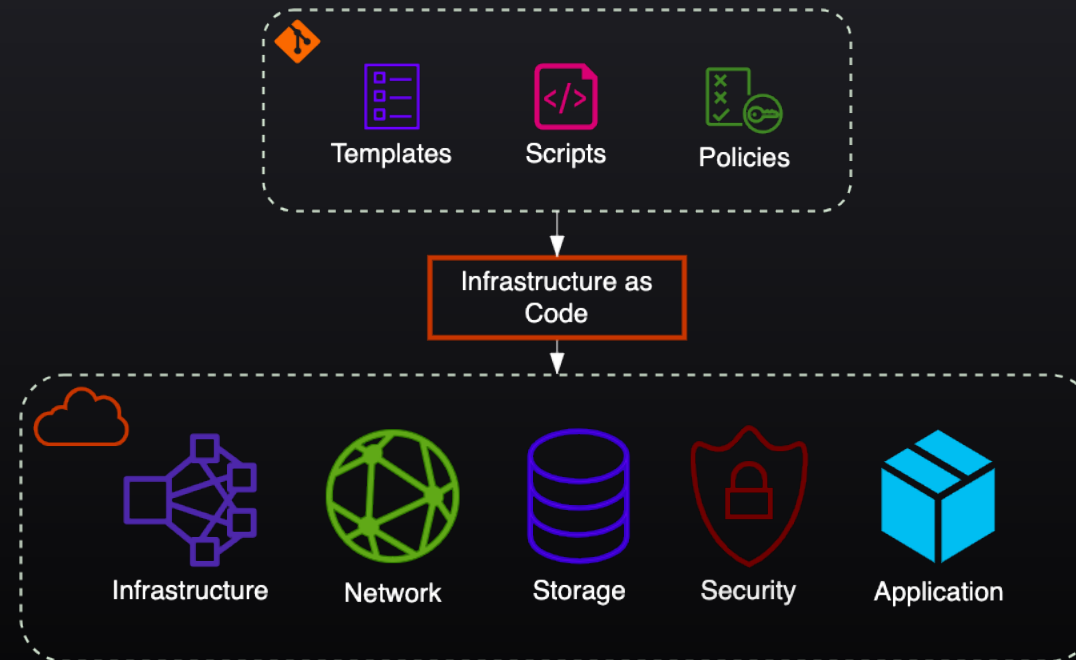
# DevSecOps



# Quality Gate & Test Automation



# Infrastructure as Code



# Infrastructure as Code – Going further

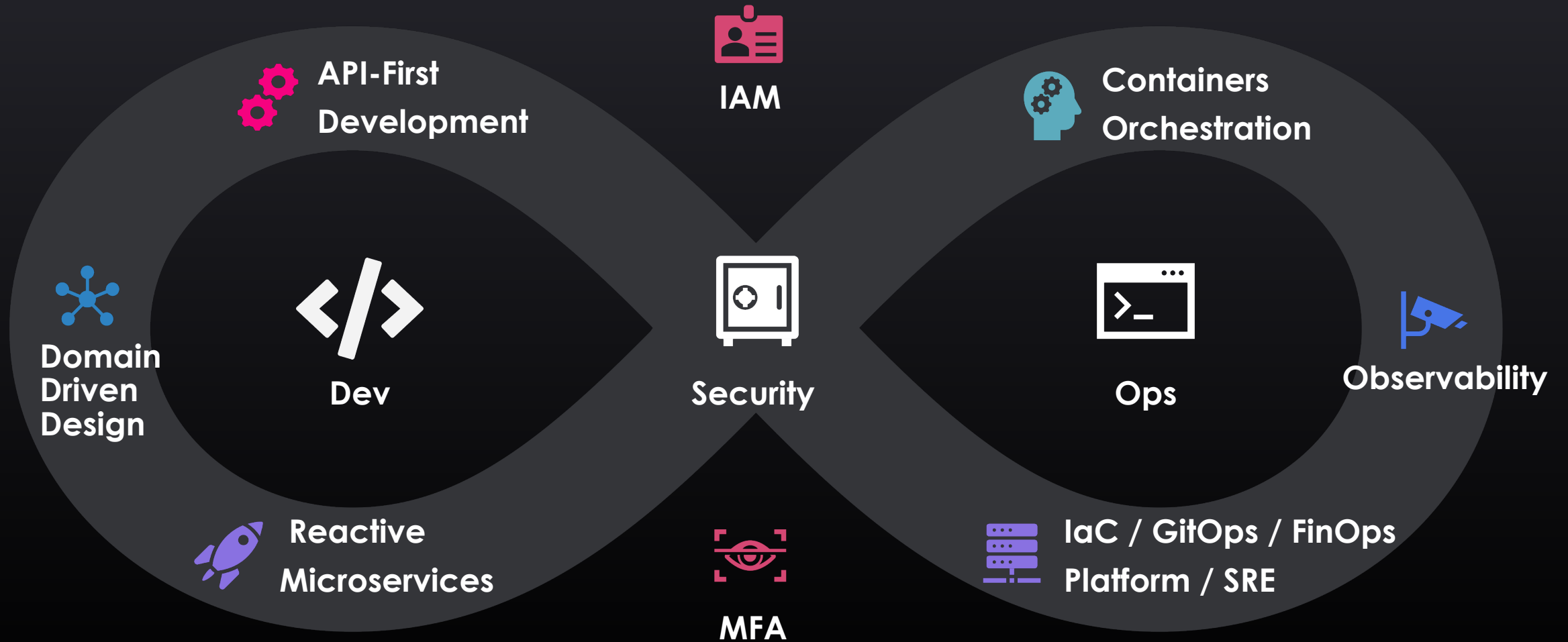
- Key Vault
- GitOps
- Platform Engineering
- SRE
- FinOps



**Rafa Nadal**  
French Open Champion  
2005 2006  
2007 2008  
2010 2001  
2012 2013  
2014 2017  
2018 2019  
2020  
tennishead



# Cloud-native Software on Steroids



## Cloud-native Software **Benefits**



Accelerate  
**delivery**



Speed-up  
**time-to-market**



Enable  
**modularity**

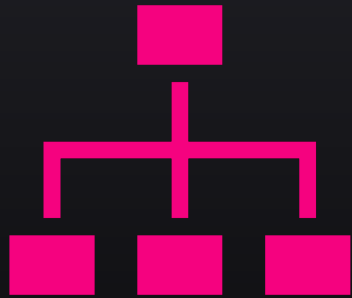


Enable  
**Agility**



Continuous  
**Delivery**

## Takeaway



There is not a single or clear way for designing Software Architecture



Software Architecture is at the edge between **Business Goals, Functional Requirements, Hardware Capabilities** and... **your Budget!**

## Takeaway

---

- Architecture decisions are **tough**
- Architecture decisions always come with **trade-offs**
- Architecture decisions always require **effort** (and sometimes **pain**)
- Architecture decisions require **compromise**
- Architecture decisions should always be **balanced**
- Changes require to **adapt**

Takeaway

“

*What problem are  
you trying to solve?*

”

*A Software Architect*



Takeaway



*Here are the  
options: ...*



*A Software Architect*



*Any question?* 🙋

*Thank you!* 🙏

 Pierre Versali

 [pierre-versali.bitbucket.io](https://pierre-versali.bitbucket.io)

APISEC  
CON

apidays