# Reactive Microservices

VERSION 1.0

**Microservices** are an architecture and an approach to building applications. Microservices are distributed and loosely coupled, small and autonomous services that work together, collections of small and isolated services each of which owns their data.

# Monoliths

- All-in-one, all-or-nothing
- Difficult to scale
- Difficult to understand
- Difficult to maintain

@itrjwyss

AMBASSADOR

@itrjwyss

Oracle Groundbreaker Ambassador

@itrjwyss

AMBASSADOR

@itrjwyss

Oracle Groundbreaker Ambassador

@itrjwyss

AMBASSADOR

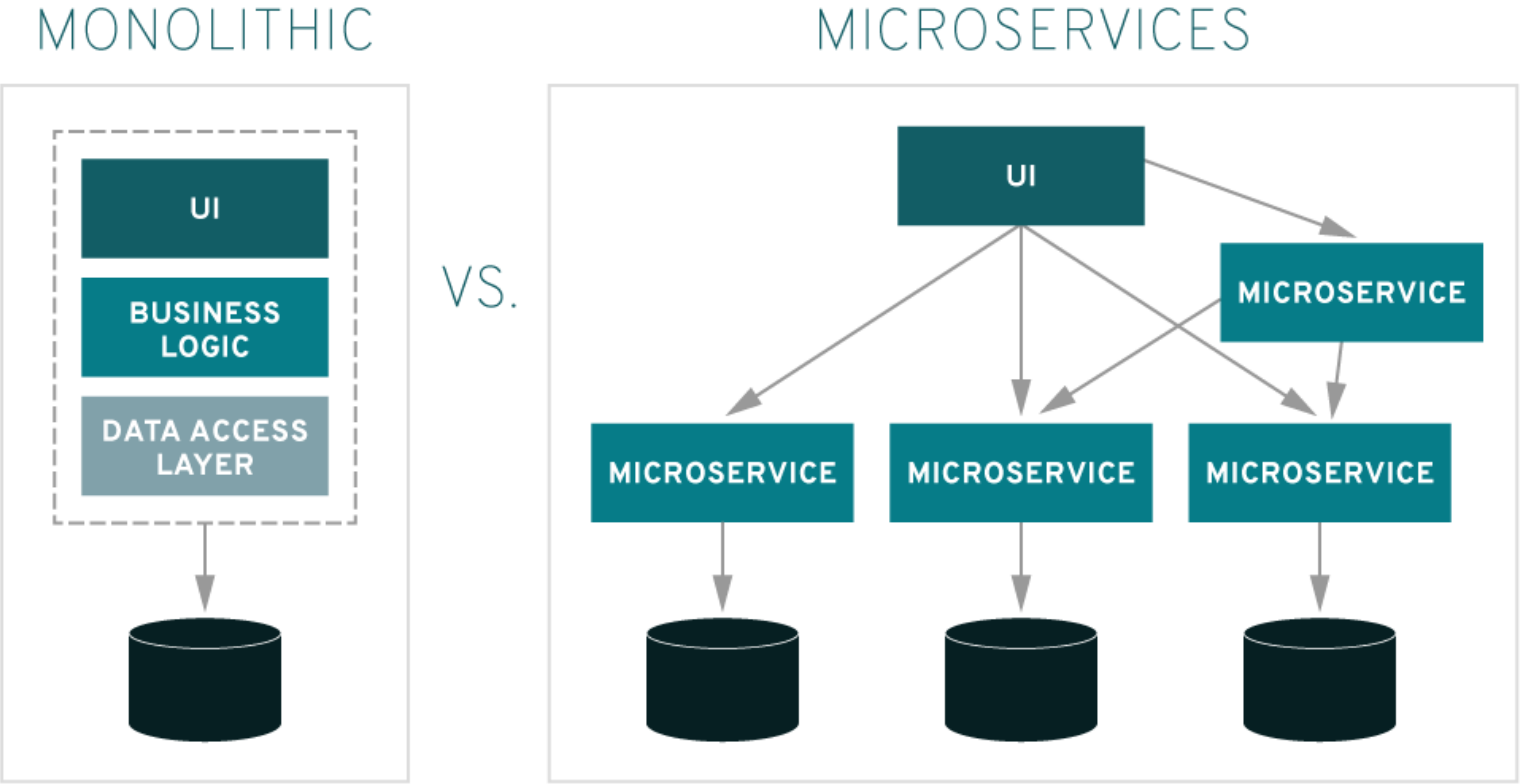
@itrjwyss

Oracle Groundbreaker Ambassador

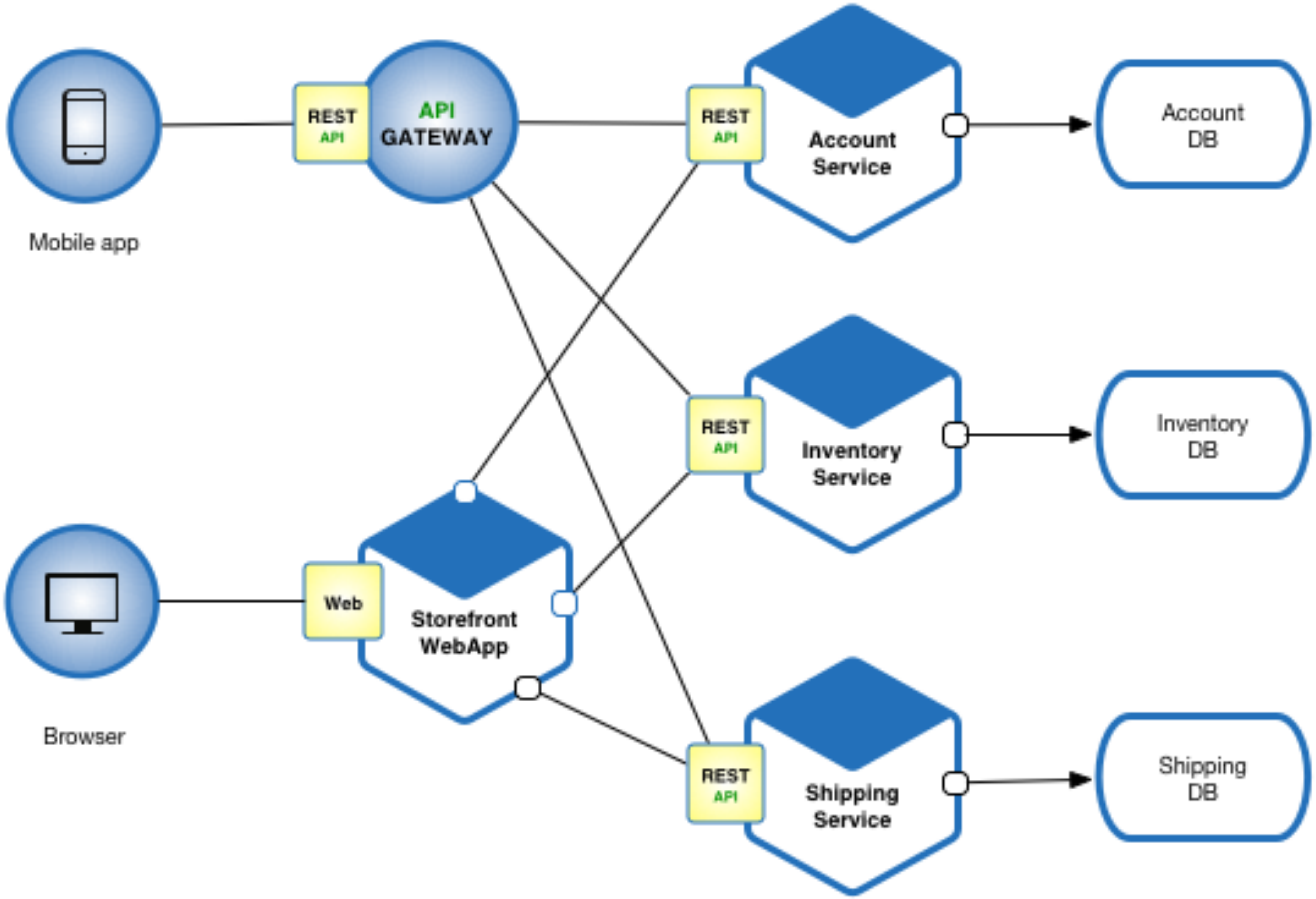
Microservices

# Divide and Conquer

@itrjwyss



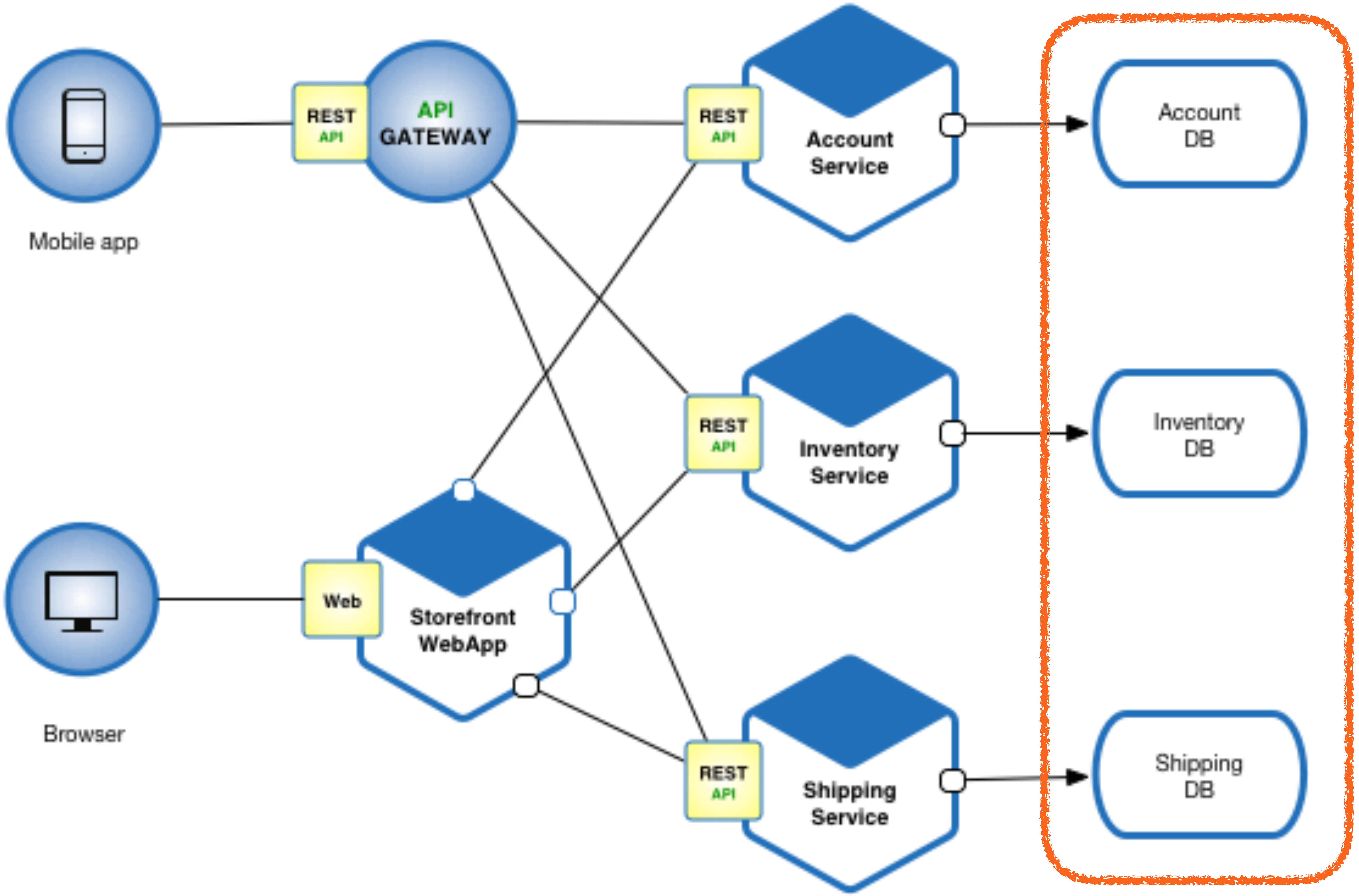
*https://www.redhat.com/en/topics/microservices/what-are-microservices*



https://microservices.io/patterns/microservices.html

@itrjwyss

AMBASSADOR



https://microservices.io/patterns/microservices.html

@itrjwyss

@itrjwyss

Oracle
Groundbreaker
Ambassador

# Characteristics

- Small in size
- Messaging Enabled
- Bounded by contexts
- Autonomously developed
- Independently deployable
- Decentralized
- Built and released with automated processes

@itrjwyss

AMBASSADOR

# Microservices, Really?

- A huge developer team
- New team members must quickly become productive
- The application must be easy to understand and modify
- Devops, CI/CD
- Satisfy scalability and availability
- Take advantage of emerging technologies

# Challenges

- Building
- Testing
- Versioning
- Deployment
- Logging
- Monitoring
- Debugging
- Conectivity

@itrjwyss

AMBASSADOR

# Mercedes Wyss

@itrjwyss

**Community Leader**

Devs+502 & JDuchess Chapter Guatemala

**Ex-JUG Member**

Guatemala Java Users Group (GuateJUG)

**Chief Technology Officer (CTO) at Produactivity**

Full Stack Developer

**Auth0 Ambassador &**
**Oracle Groundbreaker Ambassador**

Duke's Choice Award

2016 Winner

Java

AMBASSADOR

Oracle Groundbreaker Ambassador

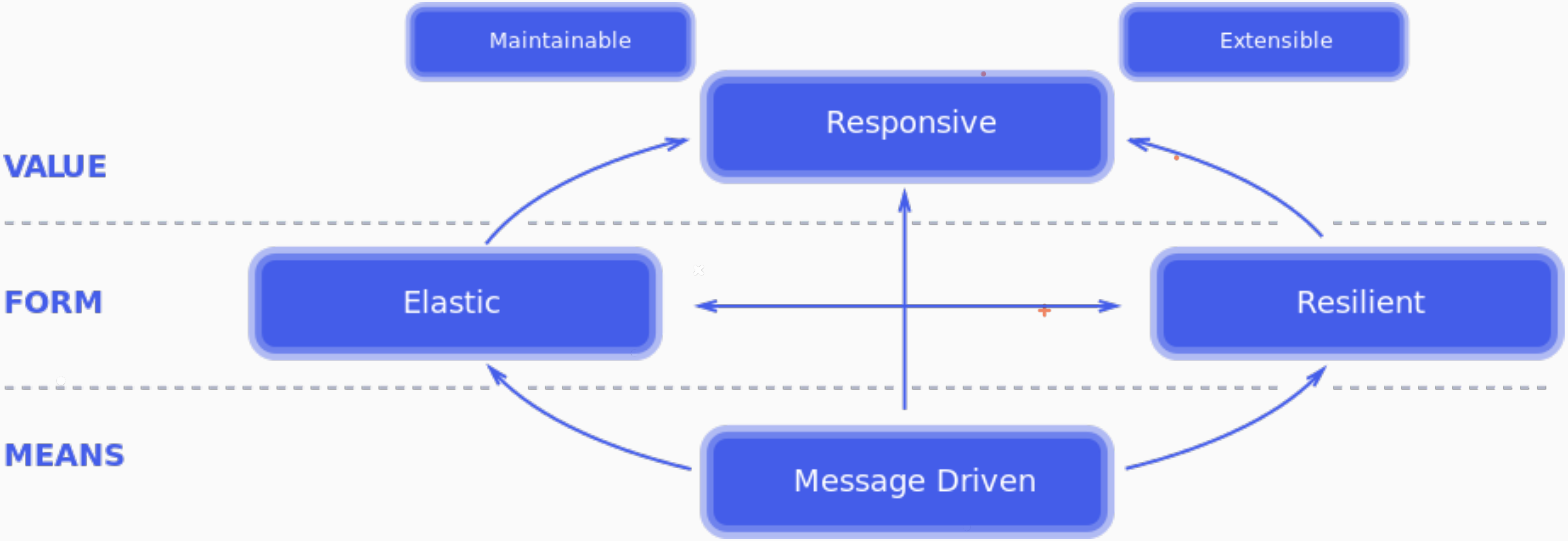
Devs +502

Comunidad Desarrolladores en Tecnologías < Google > en Guatemala

JDuchess

Guatemala

# Reactive Manifesto



@itrjwyss

Oracle Groundbreaker Ambassador

# Responsive

@itrjwyss

AMBASSADOR

# Resilient

@itrjwyss

Oracle Groundbreaker Ambassador

# Elastic

@itrjwyss

AMBASSADOR

# Message Driven

@itrjwyss

Oracle Groundbreaker Ambassador

# Reactive

- Reactive is a set of design principles
- Mean one of three things:
  - ✓ Reactive Systems (architecture and design)
  - ✓ Reactive Programming (declarative event-based)
  - ✓ Functional Reactive Programming

# Functional Reactive Programming

- Call FRP
- React to data streams using the functional paradigm
- Is not a utility or a library

# Reactive Programming

- Subset of asynchronous programming
- Discrete steps can be executed in an asynchronous and non-blocking
- Is event-driven
- Emphasis on the flow of data rather than the flow of control.
- Two styles (Callback-based, Declarative)

@itrjwyss

# Benefits

- Increased utilization of computing resources on multicore and multi-CPU hardware.
- Increased performance by reducing serialization points as per Amdahl’s Law and, by extension, Günther’s Universal Scalability Law.
- Provide a simple and maintainable approach to dealing with asynchronous and non-blocking computation and I/O.
- Typically removes the need for explicit coordination between active components.

# Event-driven vs. Message-driven

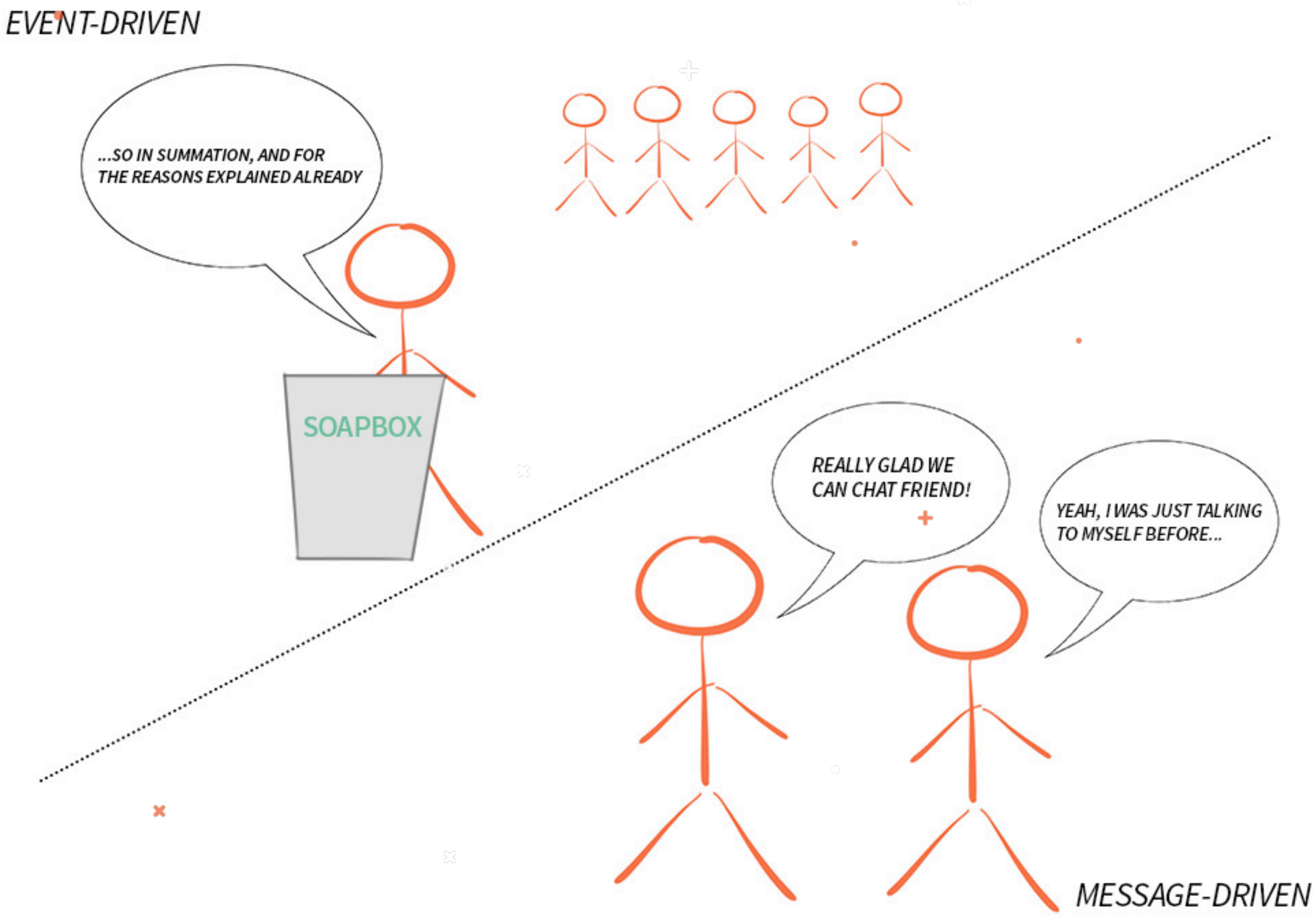
- Reactive Programming (computation through ephemeral data flow chains), event-driven

- Reactive Systems (resilience and elasticity through the communication and coordination), message-driven

- Messages are inherently directed, events are not.

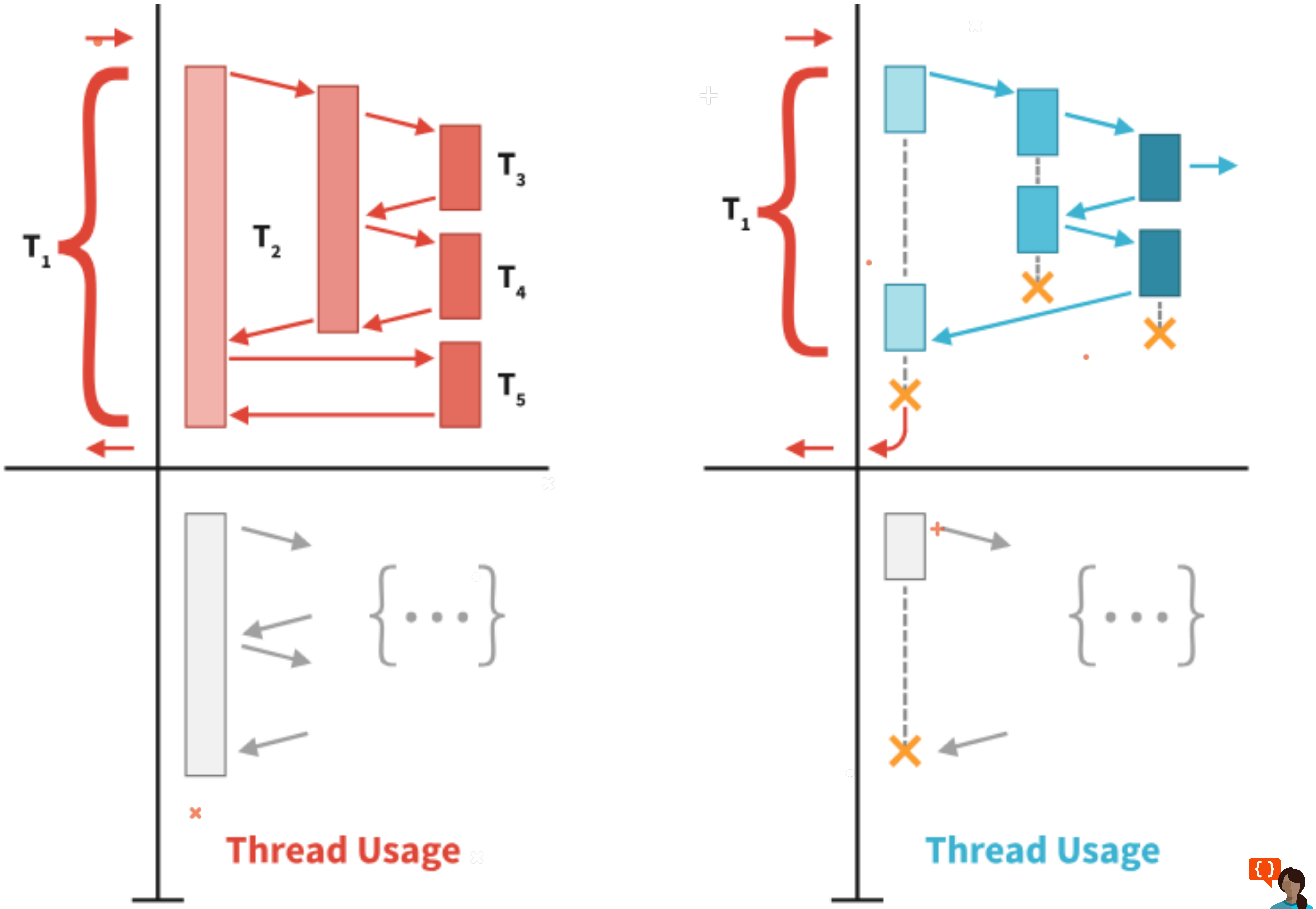
@itrjwyss

A **message** is an item of data that is sent to a specific destination. An **event** is a signal emitted by a component upon reaching a given state.


EVENT-DRIVEN
...SO IN SUMMATION, AND FOR THE REASONS EXPLAINED ALREADY
SOAPBOX
REALLY GLAD WE CAN CHAT FRIEND!
YEAH, I WAS JUST TALKING TO MYSELF BEFORE...
MESSAGE-DRIVEN

@itrjwyss

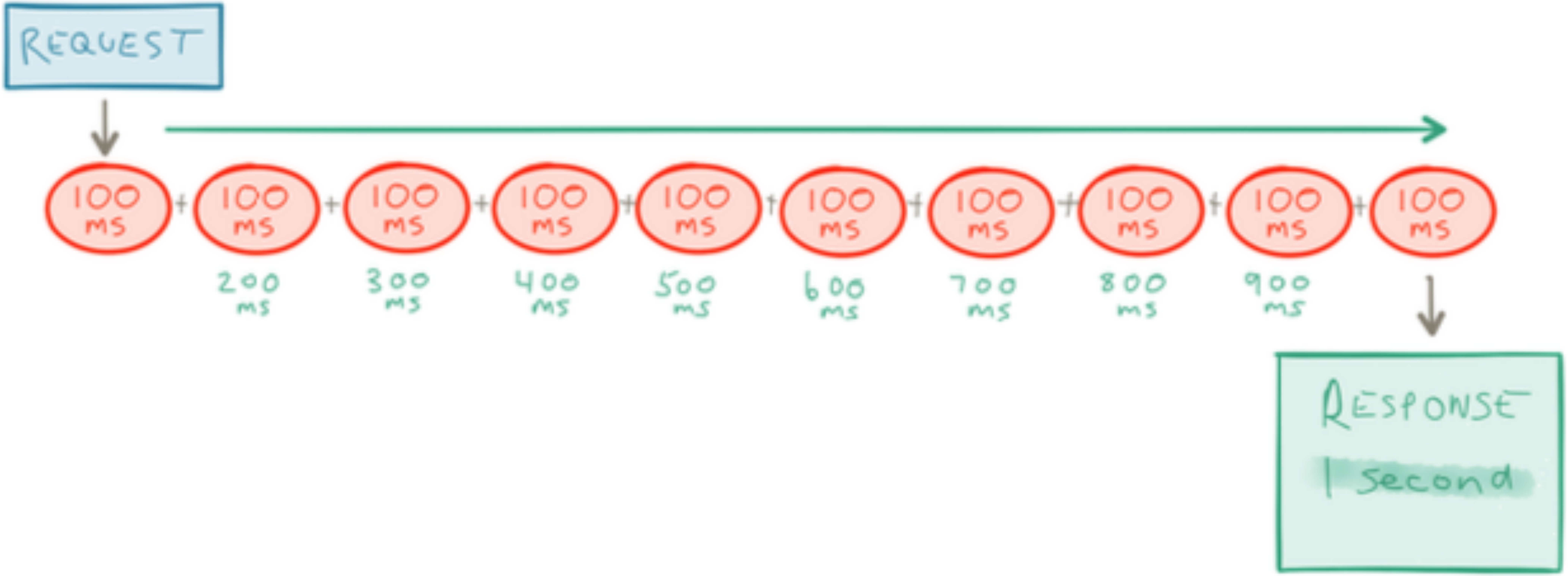
AMBASSADOR



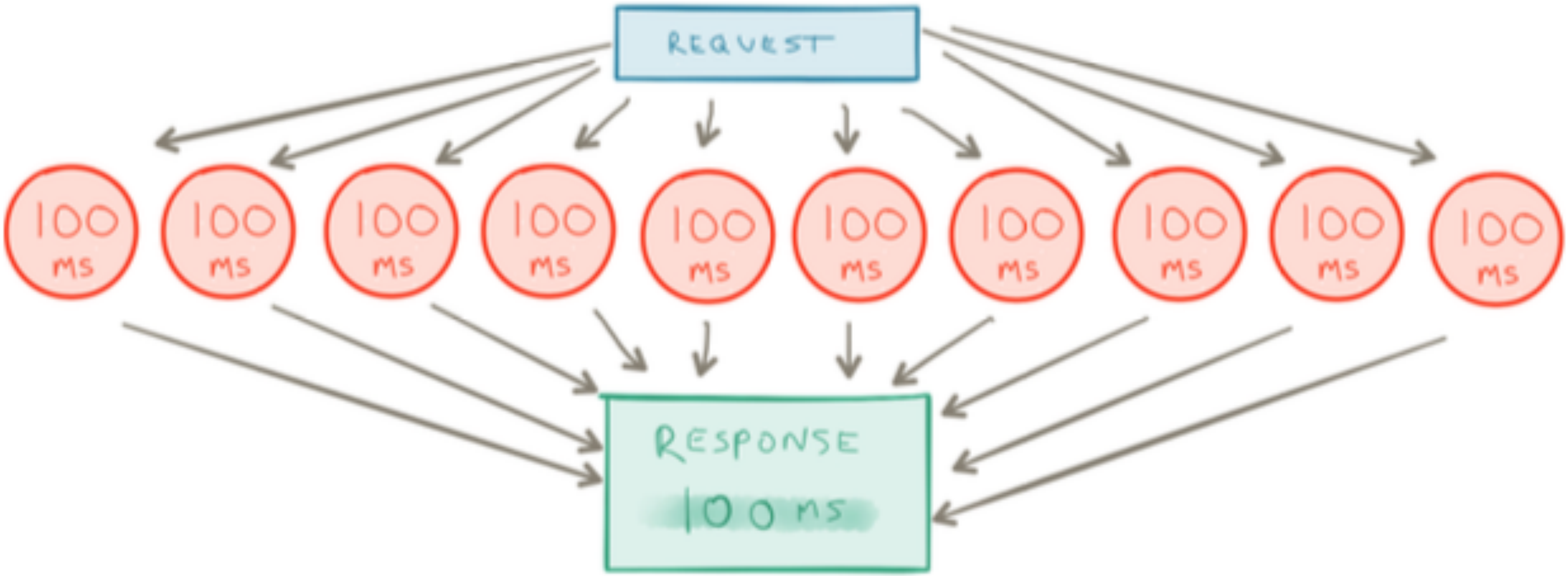
@itrjwyss

Oracle
Groundbreaker
Ambassador

# Synchronous

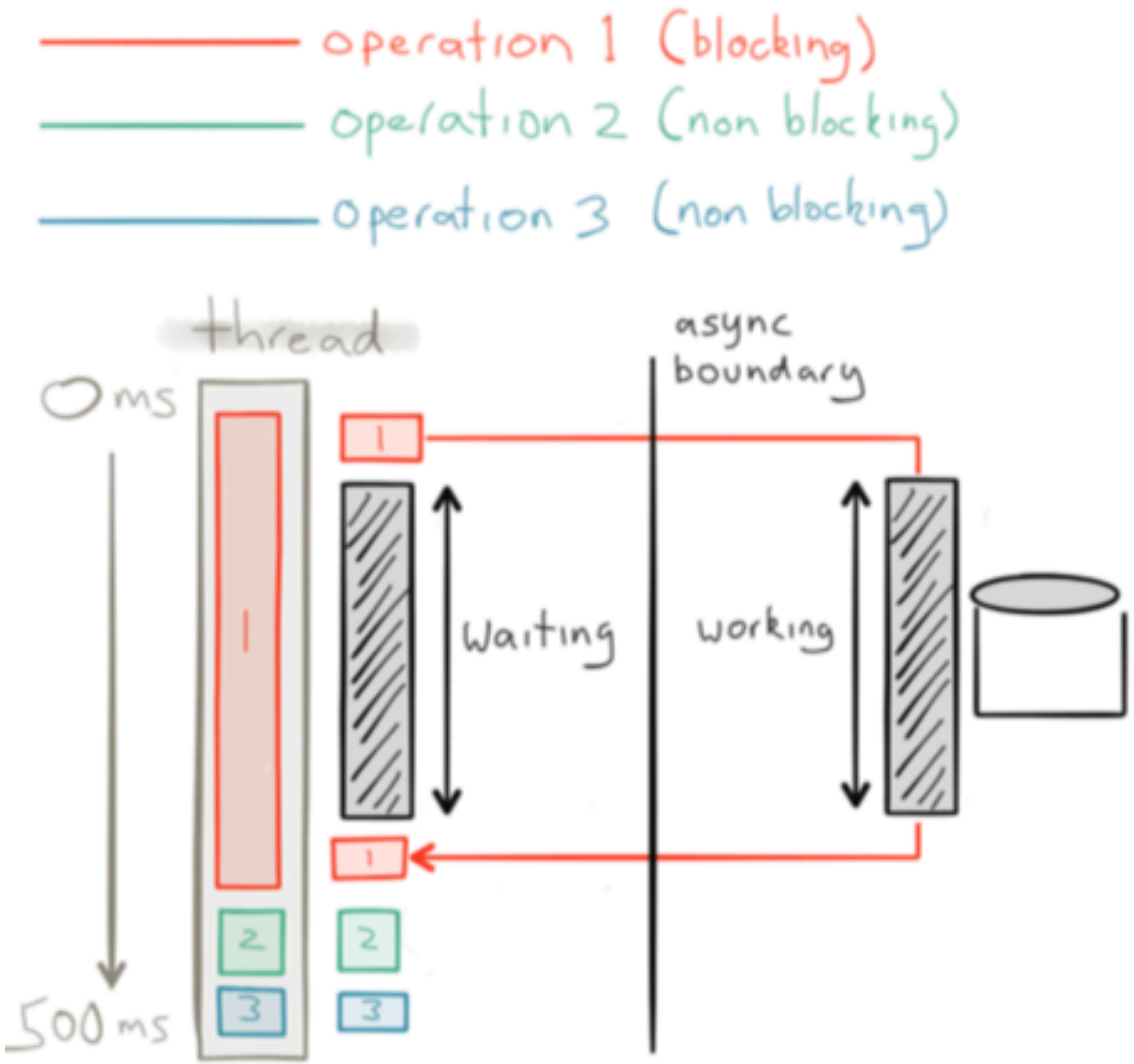


# Asynchronous



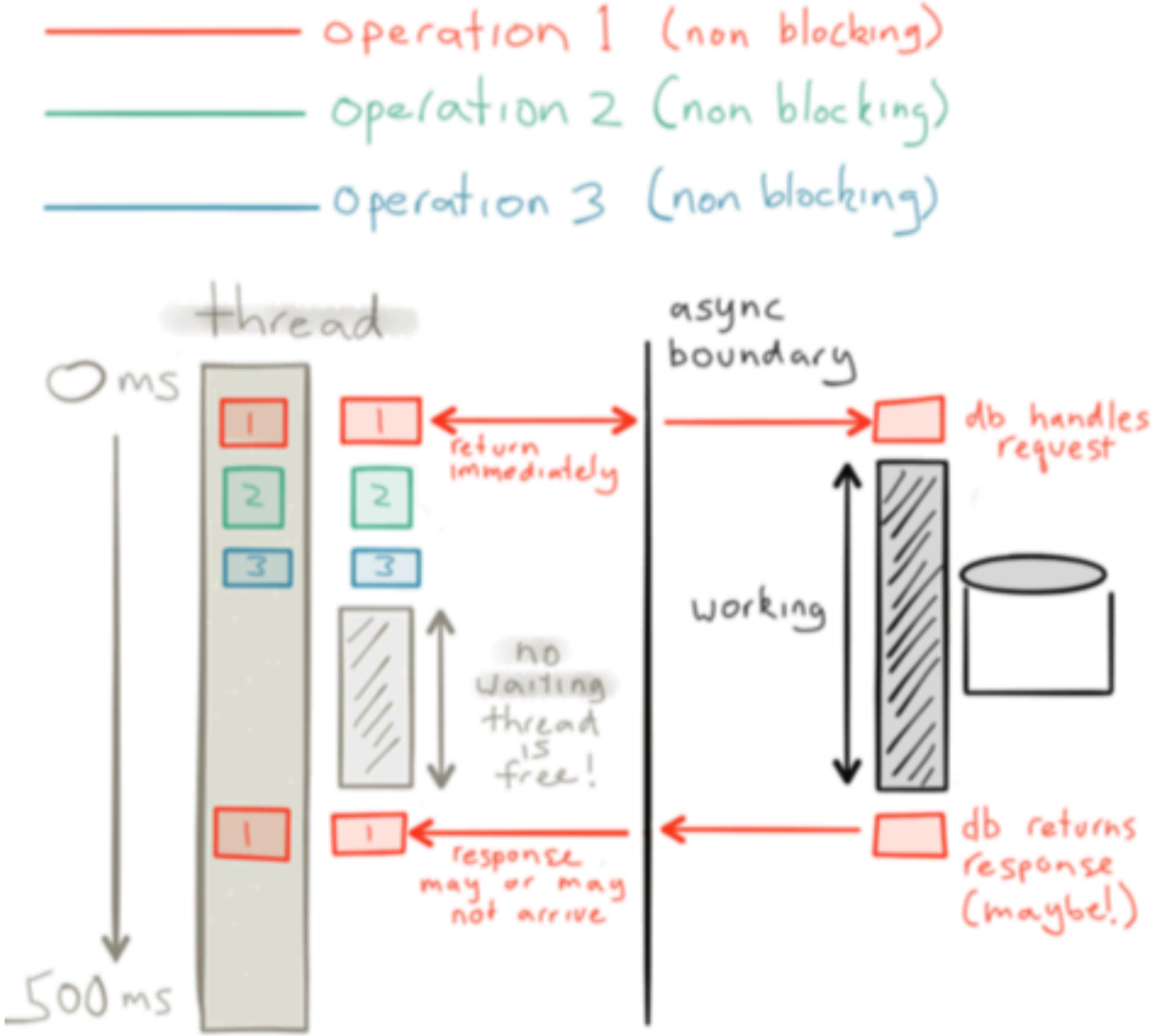
@itrjwyss

Oracle Groundbreaker Ambassador



operation 1 (blocking)
operation 2 (non blocking)
operation 3 (non blocking)
thread
async boundary
0 ms
500 ms
1
1
1
2
2
3
3
waiting
working

@itrjwyss

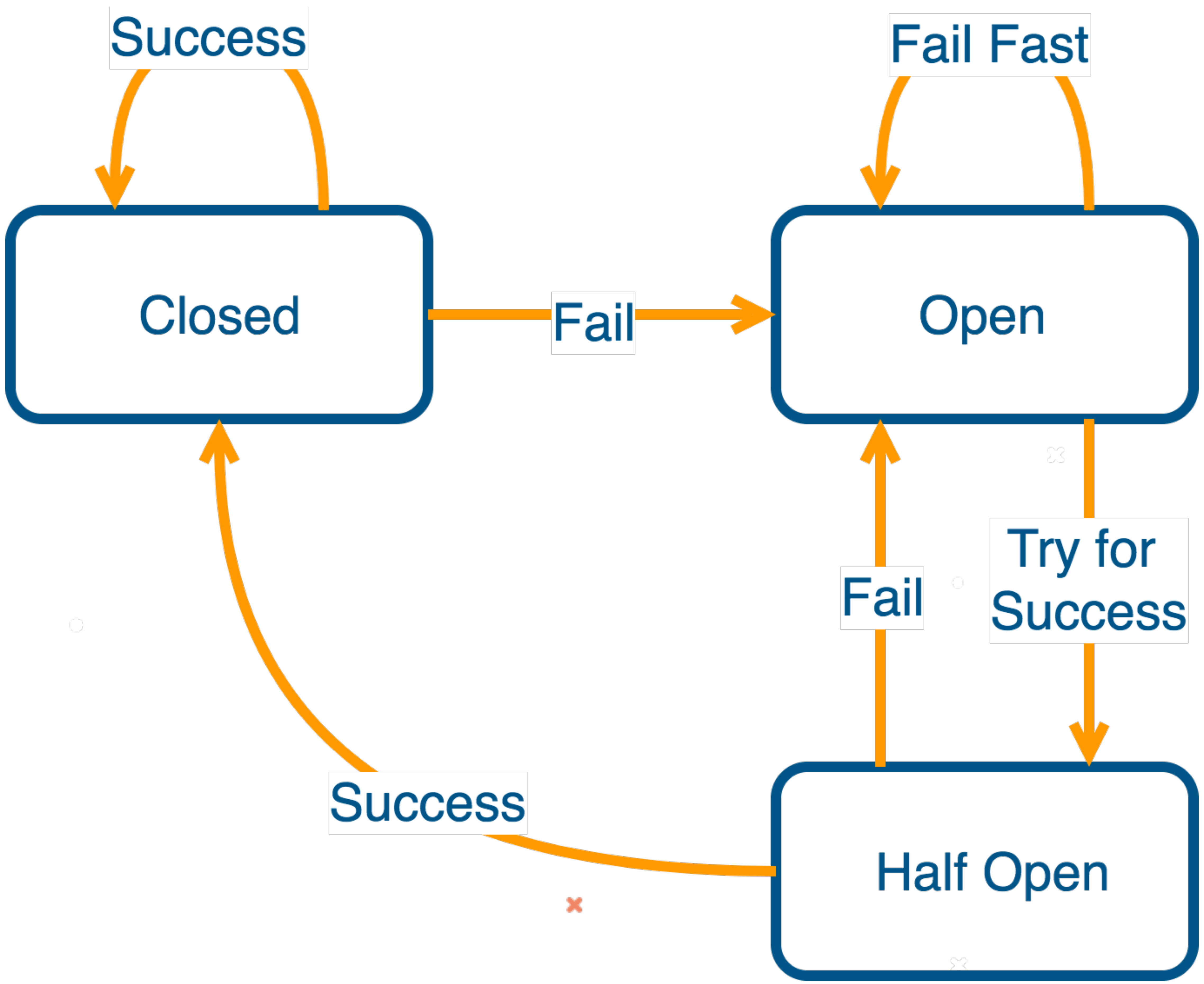

operation 1 (non blocking)
operation 2 (non blocking)
operation 3 (non blocking)
thread
async boundary
0 ms
1
2
3
1
2
3
return immediately
db handles request
working
no waiting thread is free!
1
1
response may or may not arrive
db returns response (maybe!)
500 ms

# Reactive Systems

- Definition of the Reactive Manifesto.
- message-passing (concurrency, distribution, resilience and elasticity. Full isolation between components)
- Isolation is a prerequisite for resilience and elasticity.

@itrjwyss

# About Resilience



- Patterns (Bulkheads and Circuit Breaker)
- Libraries (Netflix's Hystrix, resilience4j)
- Remove failures from the call chain, freeing the client and handling on the server.

@itrjwyss

Oracle Groundbreaker Ambassador

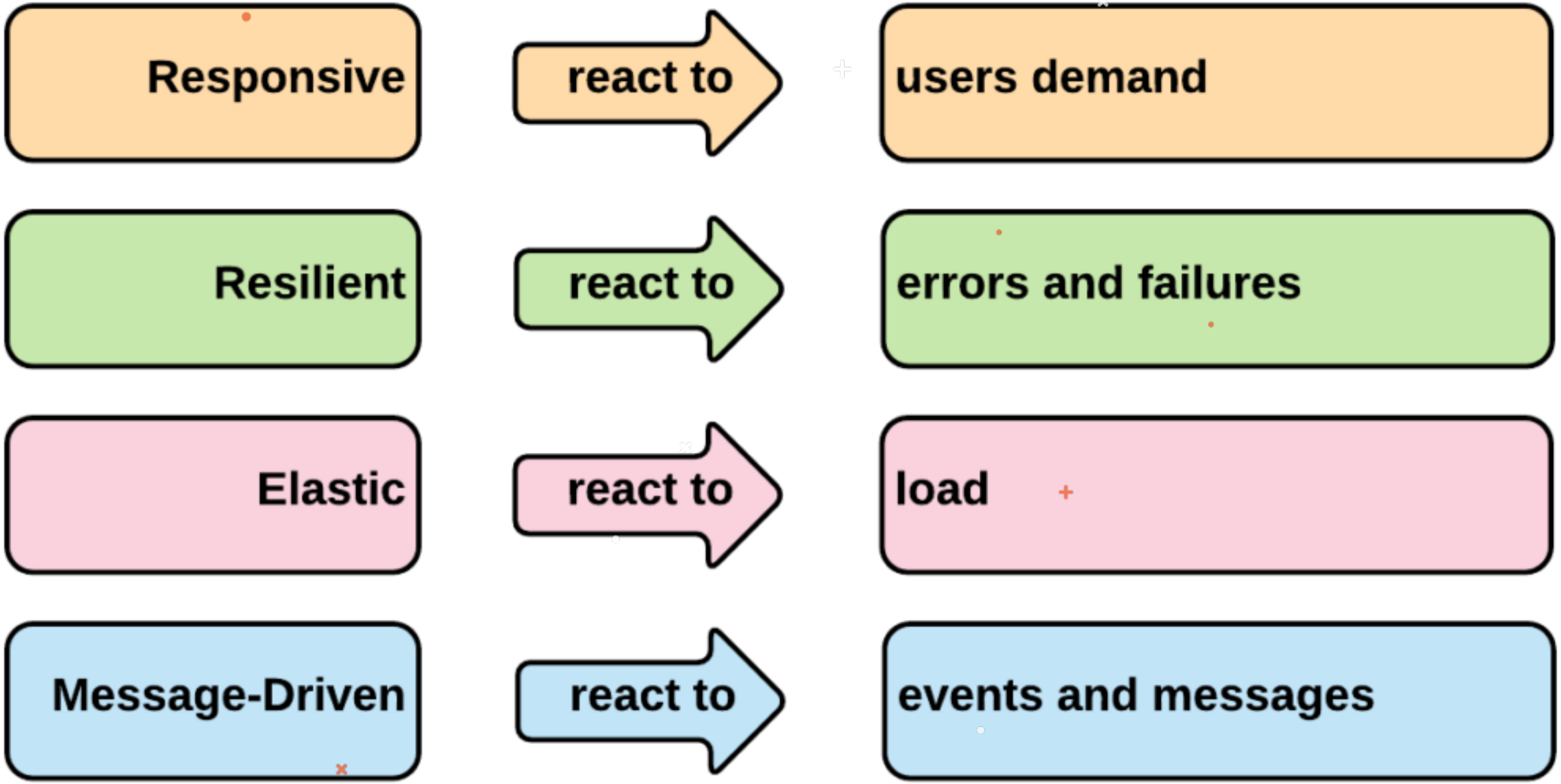
# About Elasticity

- Responsiveness under load (resource efficient, cost-efficient, environment-friendly and pay-per-use).

- Need to be adaptive (auto-scaling, replication of state and behavior, load-balancing, failover, and upgrades)

# Produactivity on Reactive Systems

- Most productive systems architecture (multicore, cloud and mobile)
  - ✓ Isolation (Resilience)
  - ✓ Supervisor hierarchies
  - ✓ Message-passing and location transparency
  - ✓ Replication (data loss, information storage and recovery)
  - ✓ Elasticity (resources, operational costs, load)



Responsive → react to → users demand

Resilient → react to → errors and failures

Elastic → react to → load

Message-Driven → react to → events and messages

# Reactive Relate to Microservices

- Microservices is an Architecture

✓ ***Reactive programming***: to implement the service-internal logic and dataflow management.

✓ **Reactive Systems**: between microservices.

@itrjwyss

Oracle Groundbreaker Ambassador

# Monolith

@itrjwyss

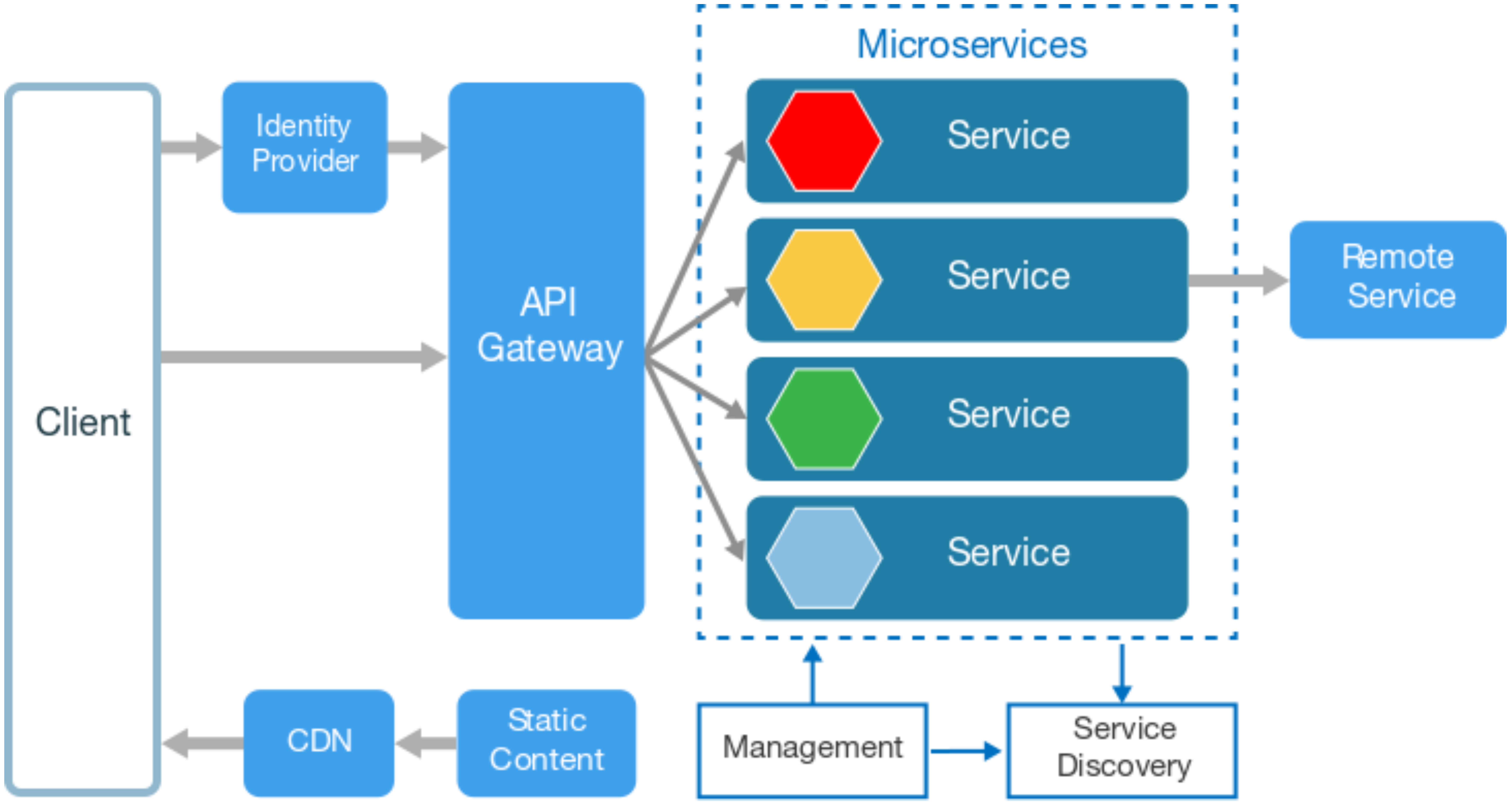
AMBASSADOR

# Microservices

# Reactive Microservices

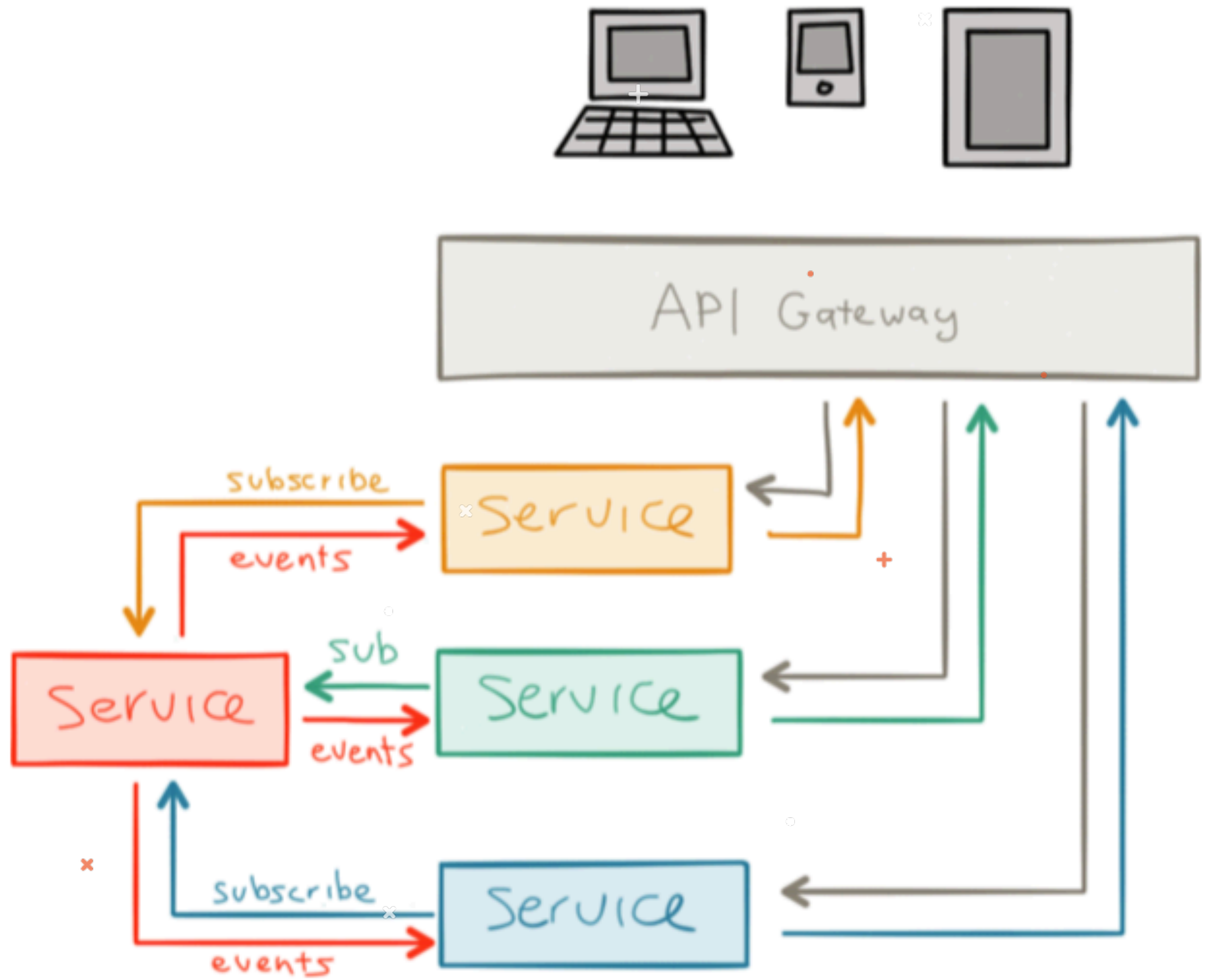
@itrjwyss

AMBASSADOR

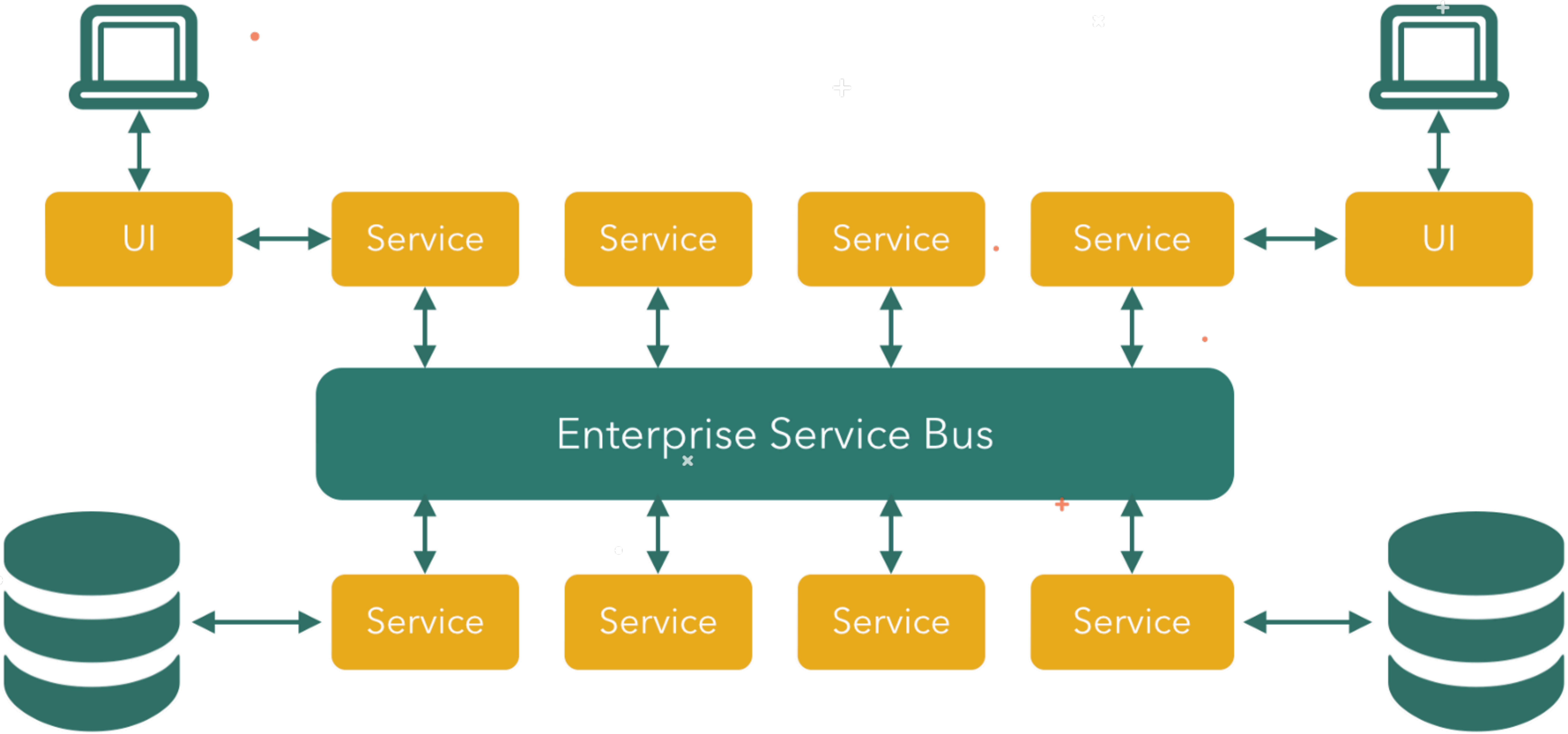


@itrjwyss

Oracle Groundbreaker Ambassador


API Gateway
subscribe
Service
events
sub
Service
Service
events
subscribe
Service
events

@itrjwyss



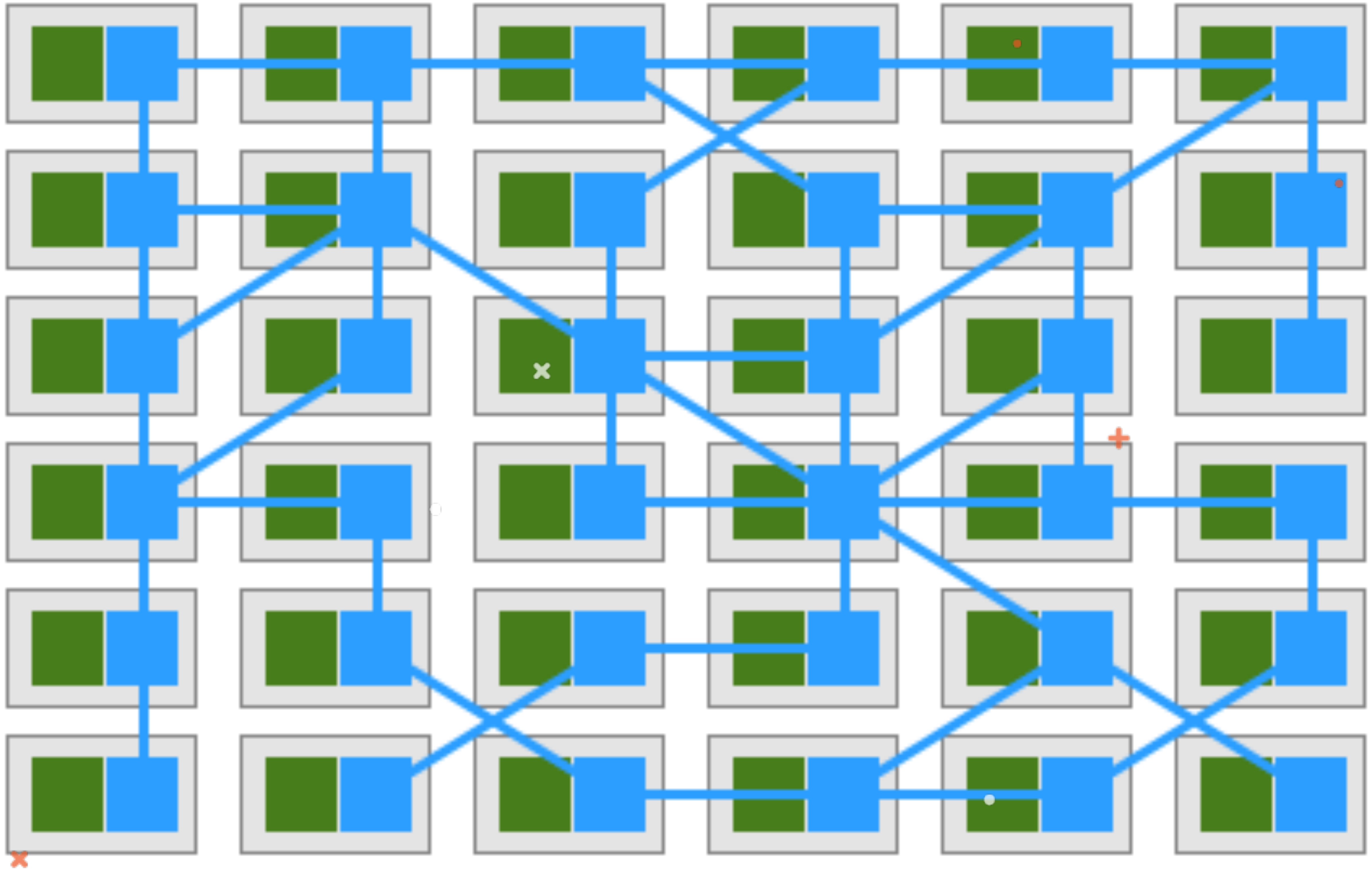
@itrjwyss

Oracle Groundbreaker Ambassador

# Integration Core Principles

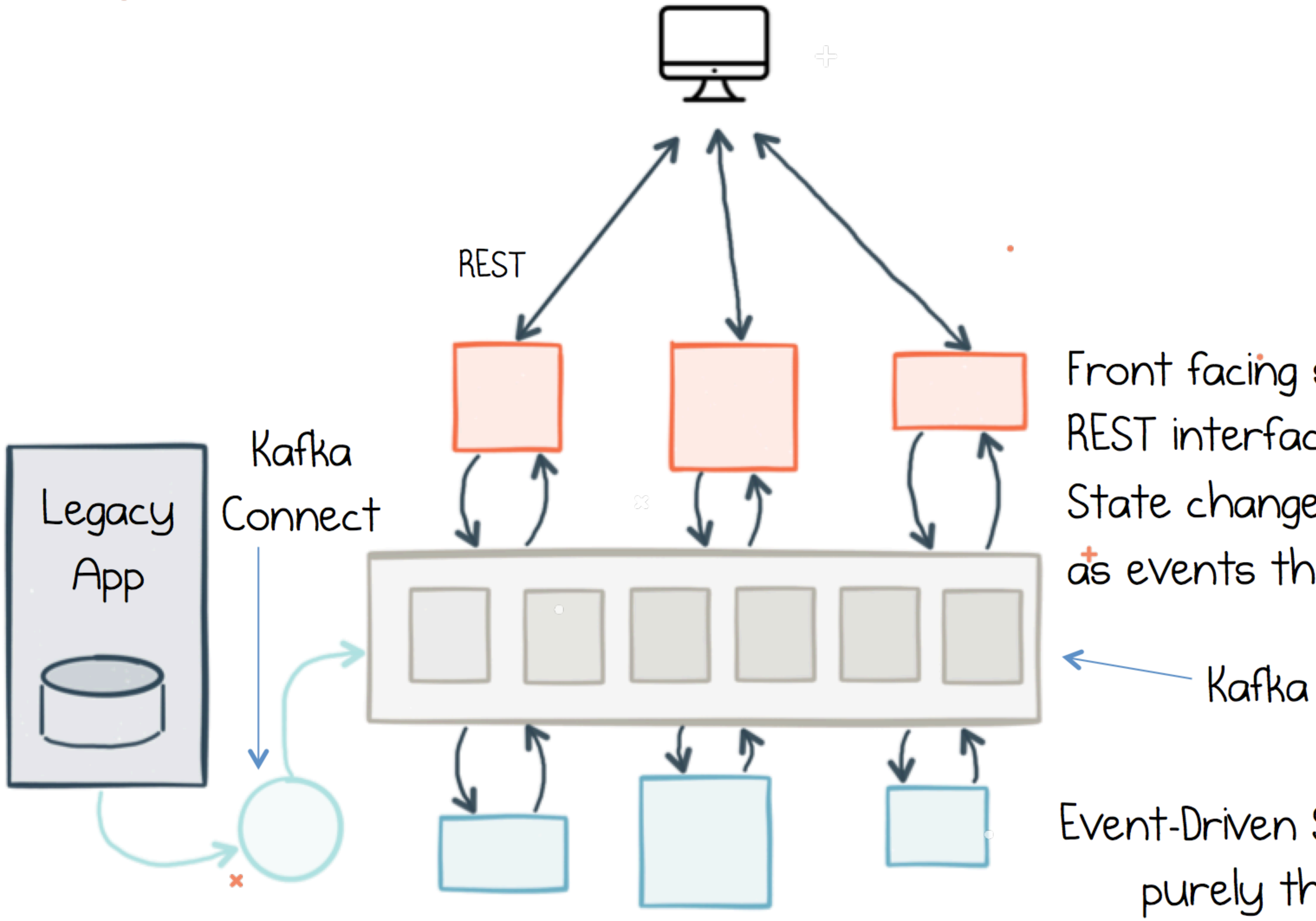
- Orchestration
- Transformation
- Transportation (HTTP, JMS, JDBC)
- Mediation (supporting multiple versions, multiple channels)
- Non-functional consistency

# Service Mesh



@itrjwyss

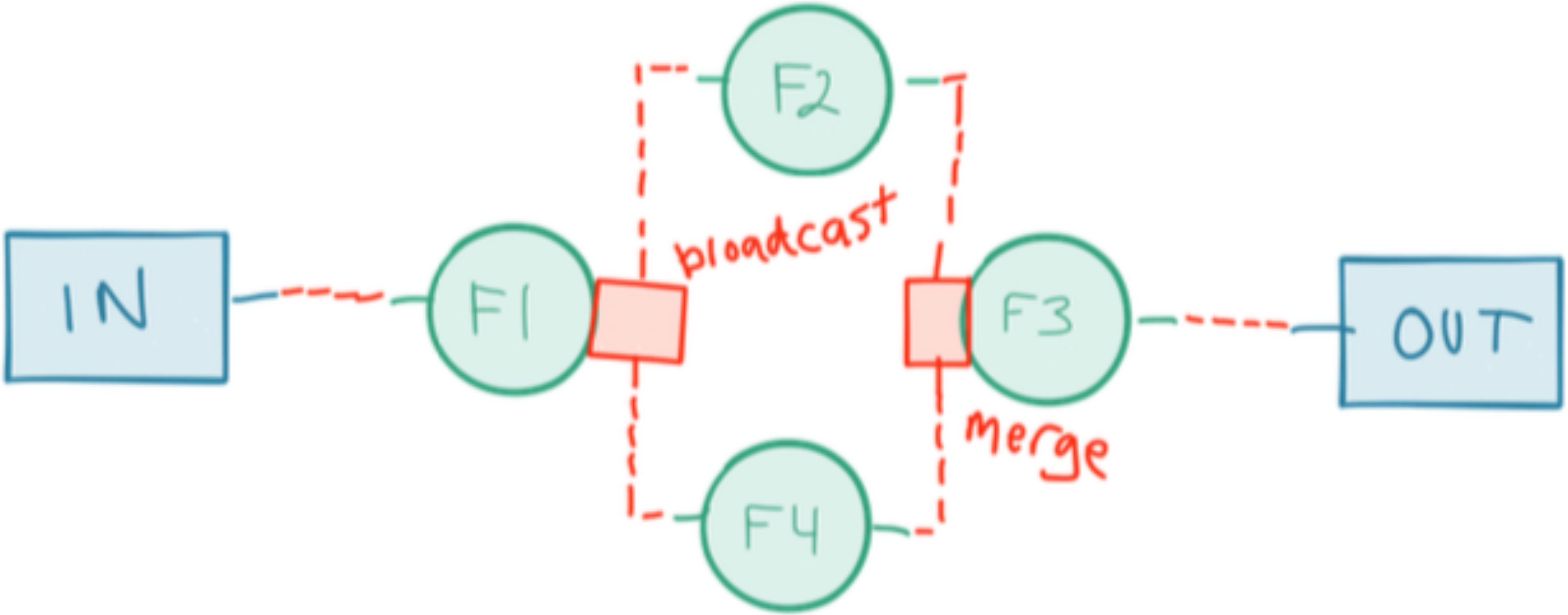
Oracle Groundbreaker Ambassador



Front facing services provide REST interfaces to the UI.

State changes are propagated as events through Kafka.

Event-Driven Services interact purely through Kafka



@itrjwyss

Oracle Groundbreaker Ambassador

# Security

- Authentication and Authorization
- TLS Client Certificates
- HTTPS Basic Authentication
- Asymmetric Request Signing
- Hash Message Authentication Code (HMAC)

https://github.com/itrjwyss/ReactiveMicroservices

https://www.facebook.com/itrjwyss

**@itrjwyss**