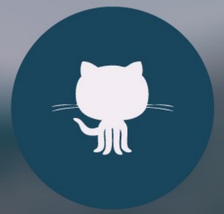
Erwin de Gier

Building reactive microservices with Vert.x 3.3



github.com/erwindeg

@erwindeg

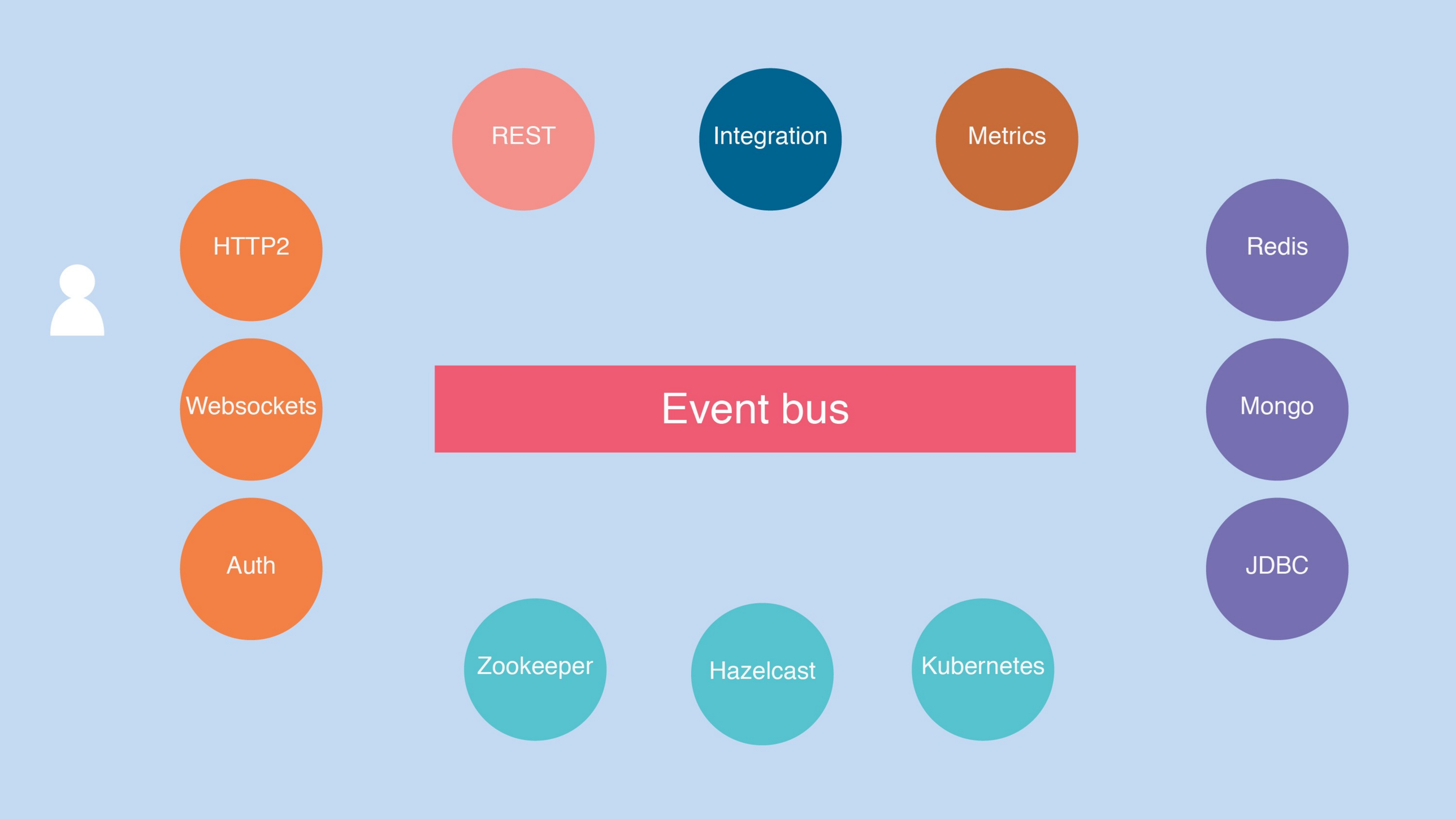
edegier.nl

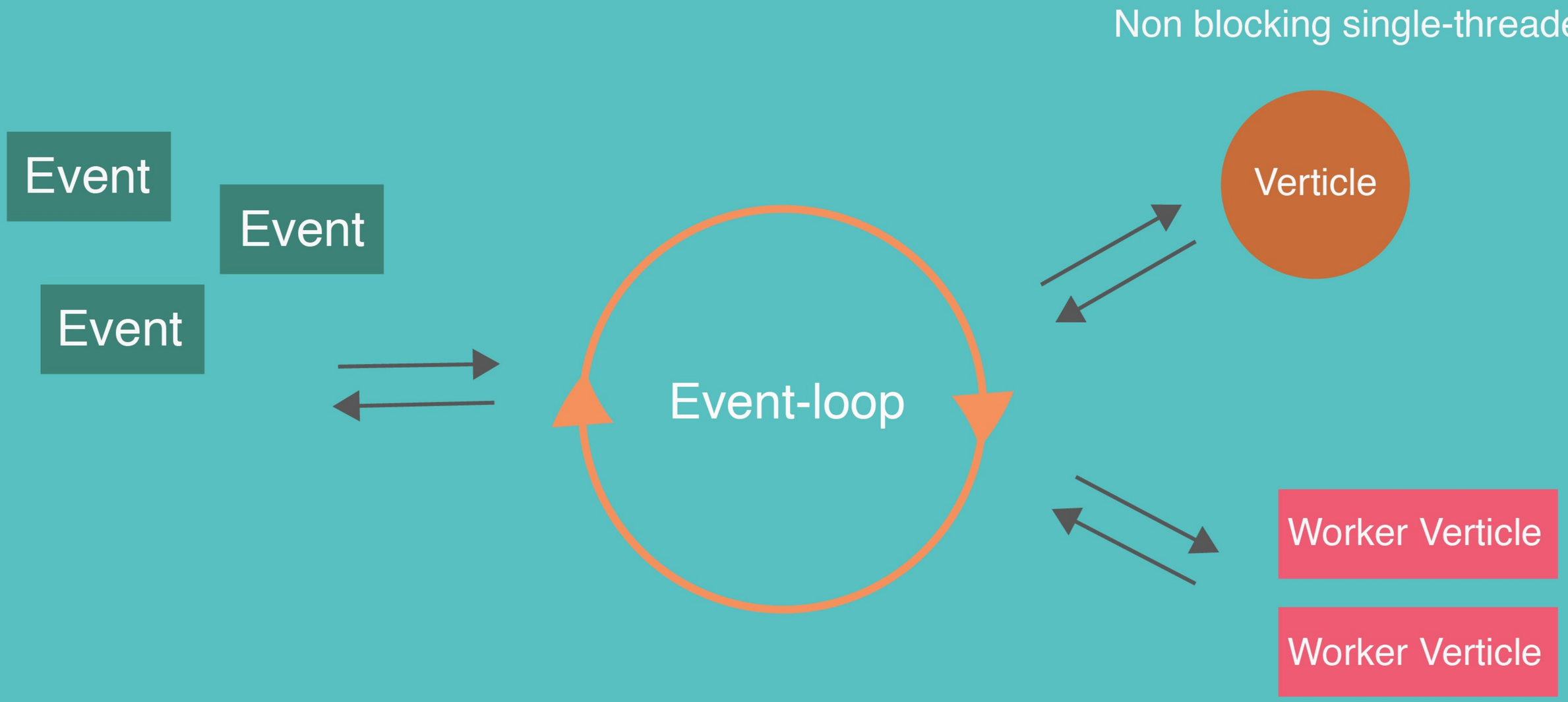
SOGETI

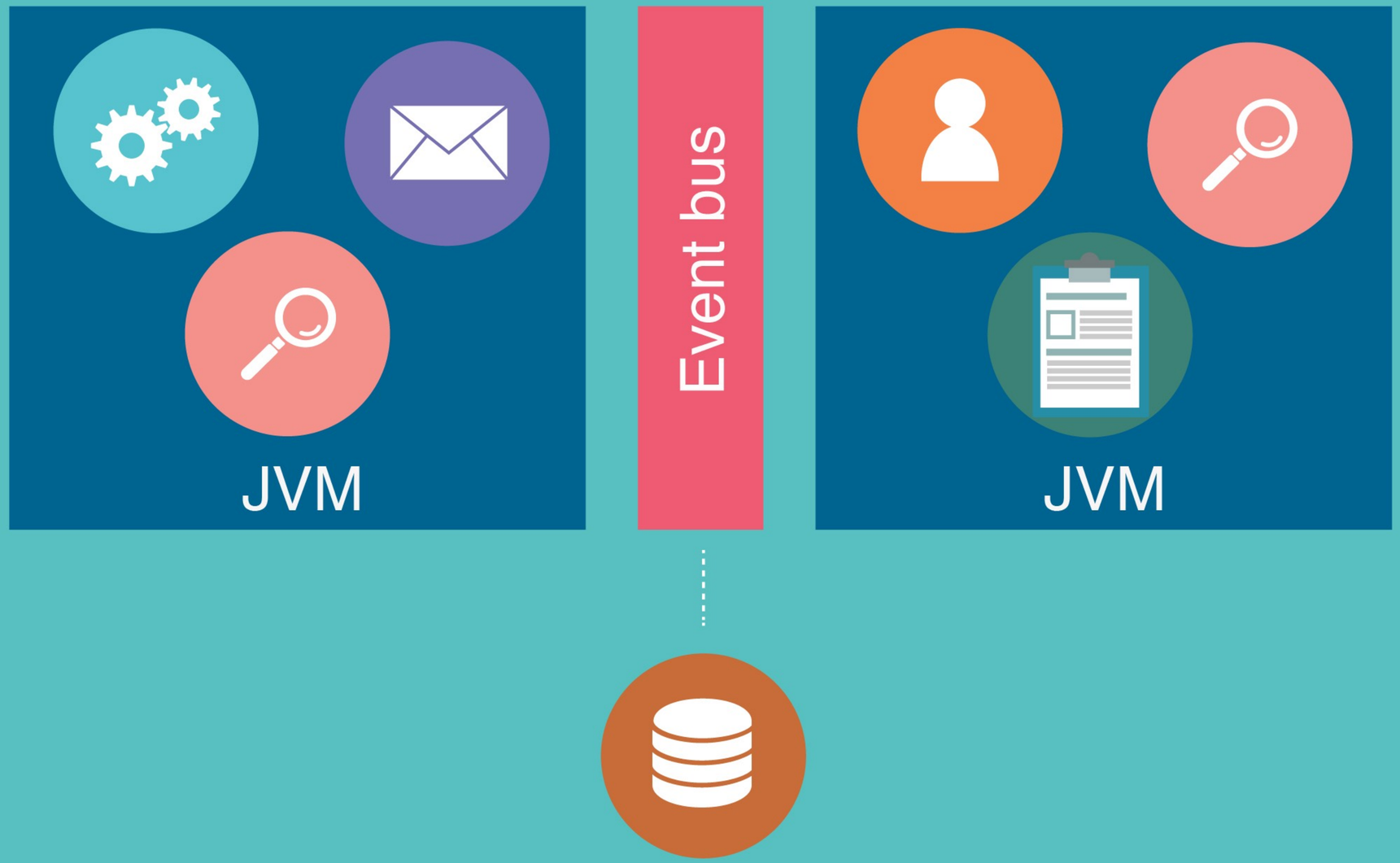
“Vert.x is a toolkit for creating reactive applications on the JVM, supporting multiple development languages simultaneously.”

ck

- Runnable Jar
- Reactive
- Polyglot
- Distributed


REST
Integration
Metrics
HTTP2
Redis
Websockets
Event bus
Mongo
Auth
JDBC
Zookeeper
Hazelcast
Kubernetes


Non blocking single-threaded
Event
Event
Event
Event-loop
Verticle
Worker Verticle
Worker Verticle
Blocking multi-threaded


JVM
Event bus
JVM

“Vert.x is an ideal choice for creating light-weight, high-performance, microservices.”

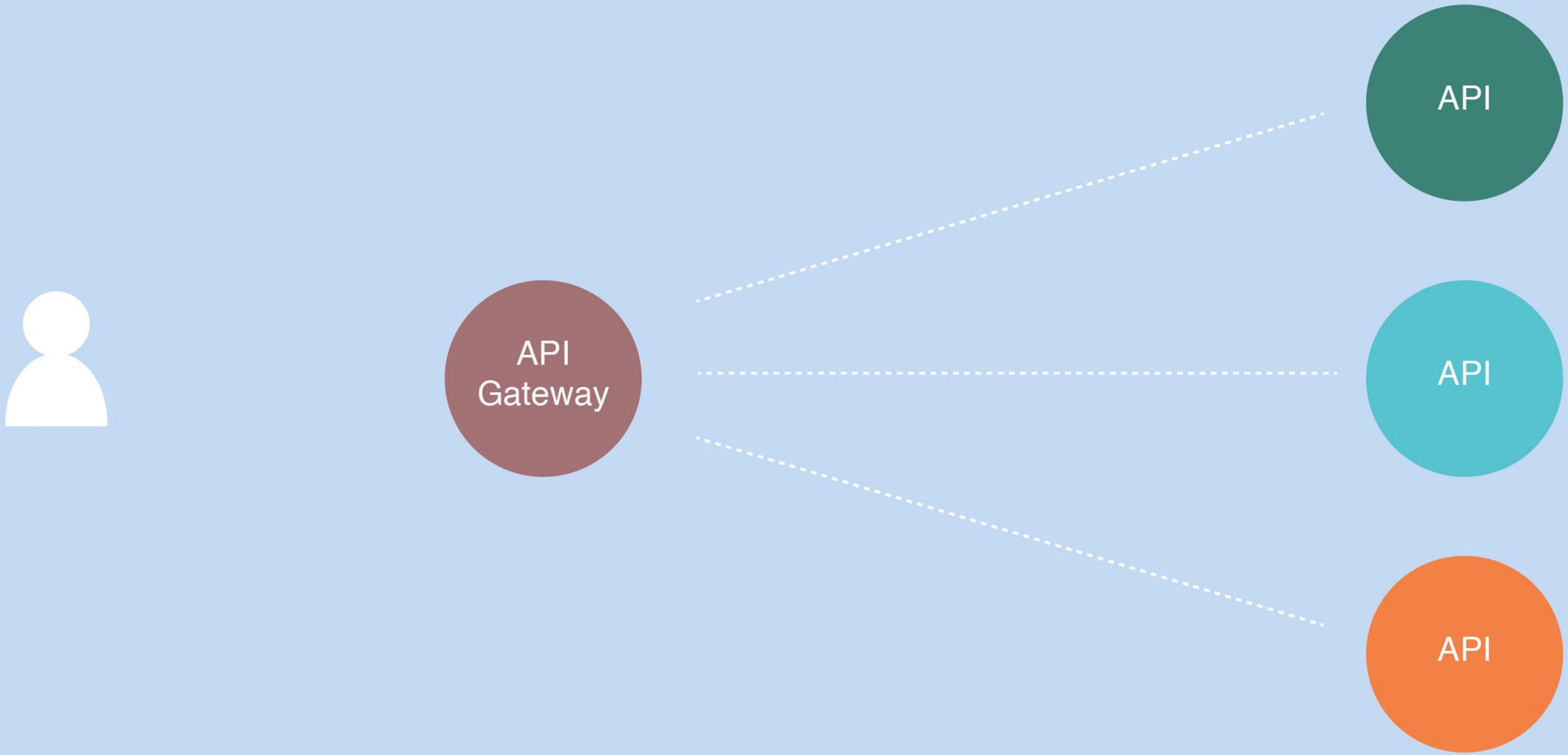
Microservice Features

- HTTP, TCP, UDP servers & clients
- Distributed messaging architecture
- Distributed data structures
- Load balancing / failover
- Shell / metrics / and deployment

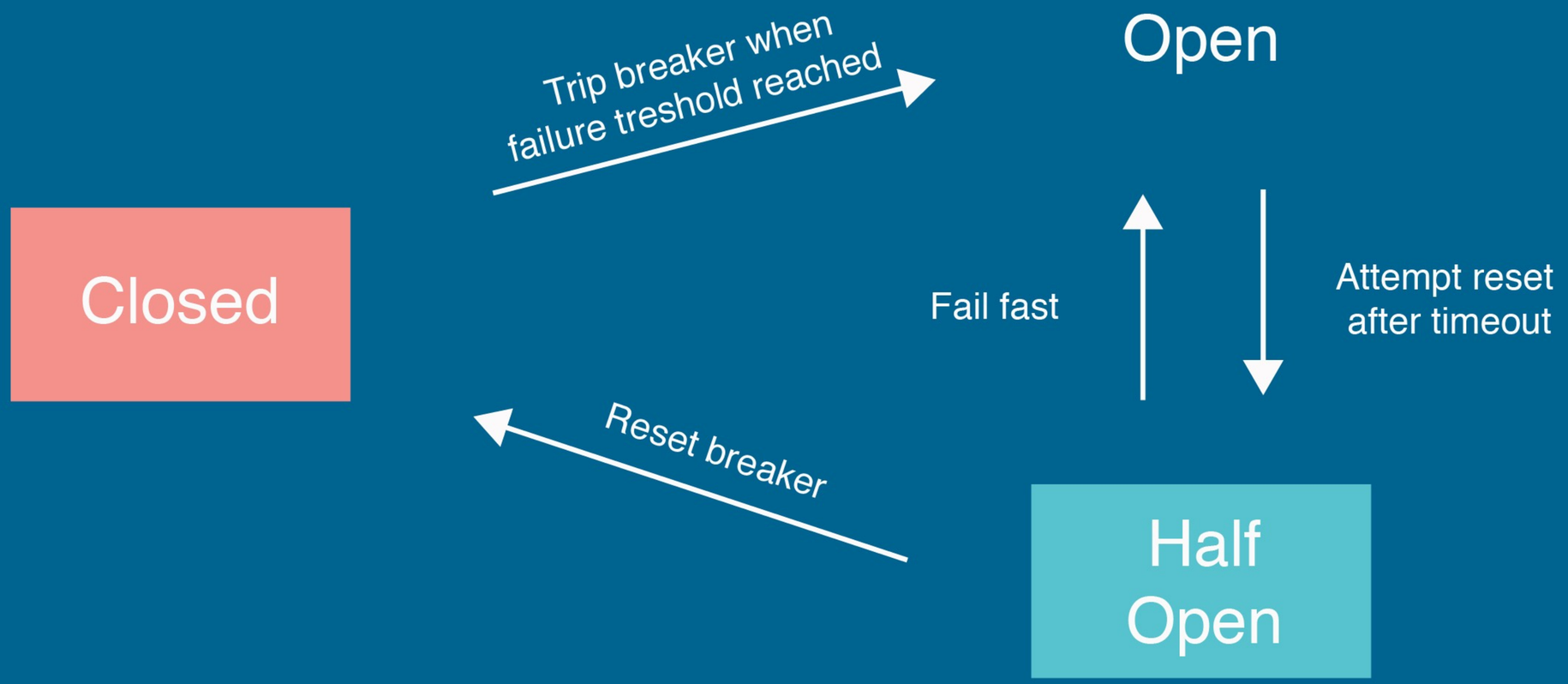
New Microservice Features

- Circuit Breaker
- Service Discovery and repositories
- HTTP2
- Web Client

API Gateway



Circuit breaker



Circuit breaker

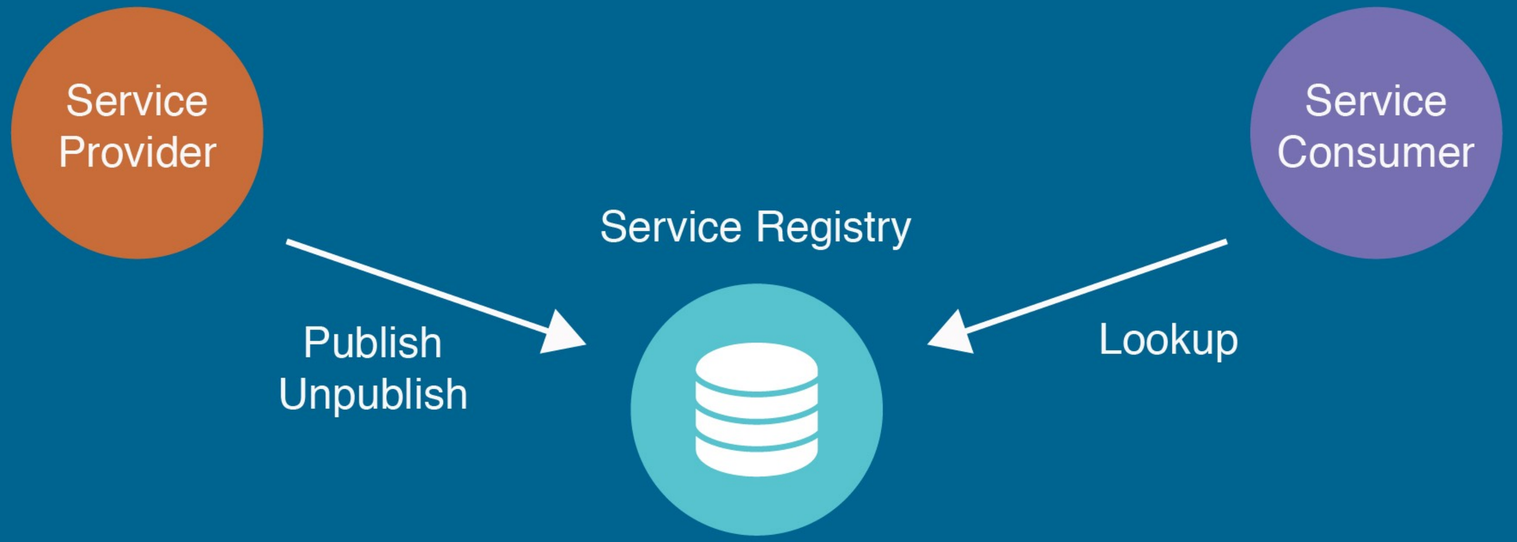
```
CircuitBreaker breaker = CircuitBreaker.create("breaker-name", vertx,
    new CircuitBreakerOptions()
        .setMaxFailures(5) // Failure threshold
        .setTimeout(2000) // Failure timeout
        .setFallbackOnFailure(true) // do we call the fallback on failure
        .setResetTimeout(10000) // Reset timeout
);

breaker.execute(future -> {
  // The call
}).setHandler(ar -> {
  // Process call result
});
```

Circuit breaker

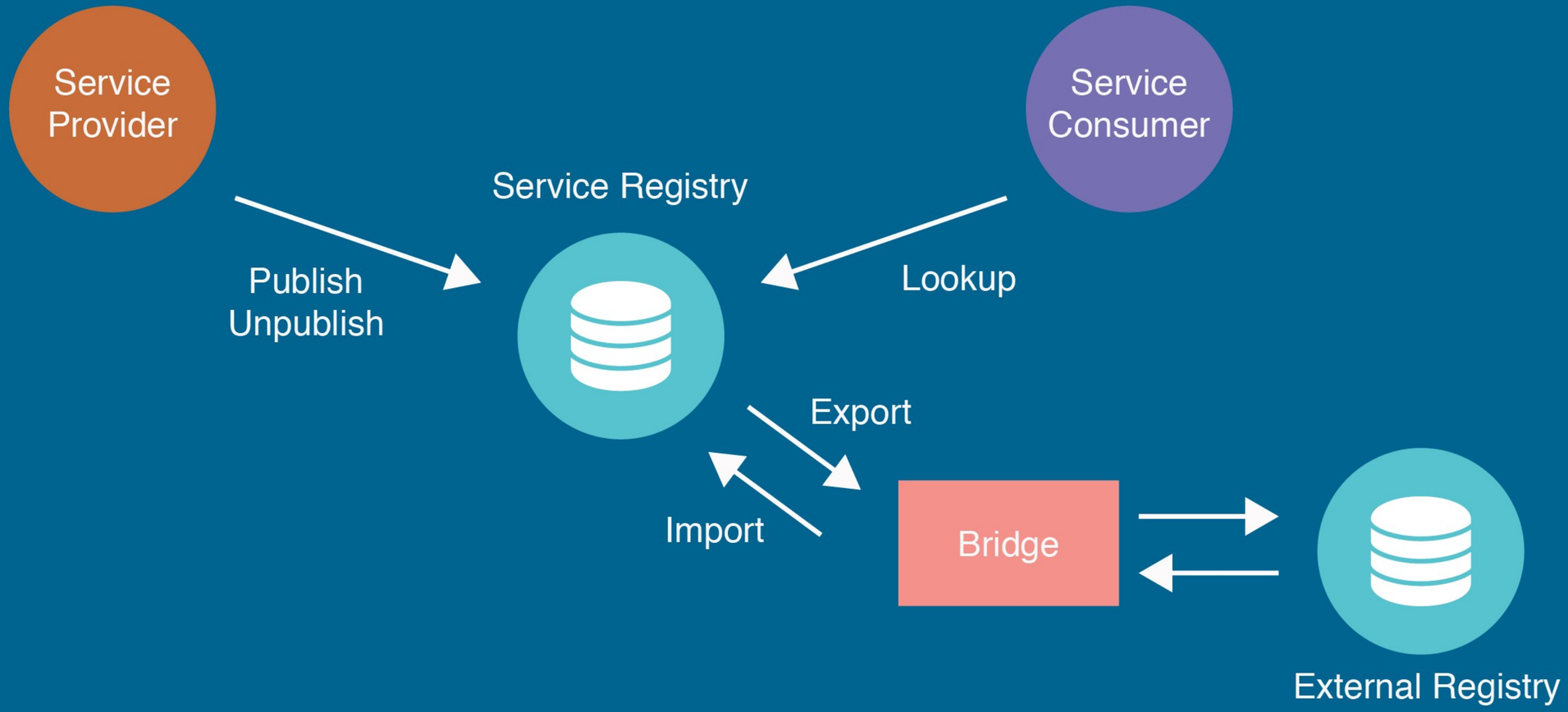
Demo

Service discovery API

Service discovery API



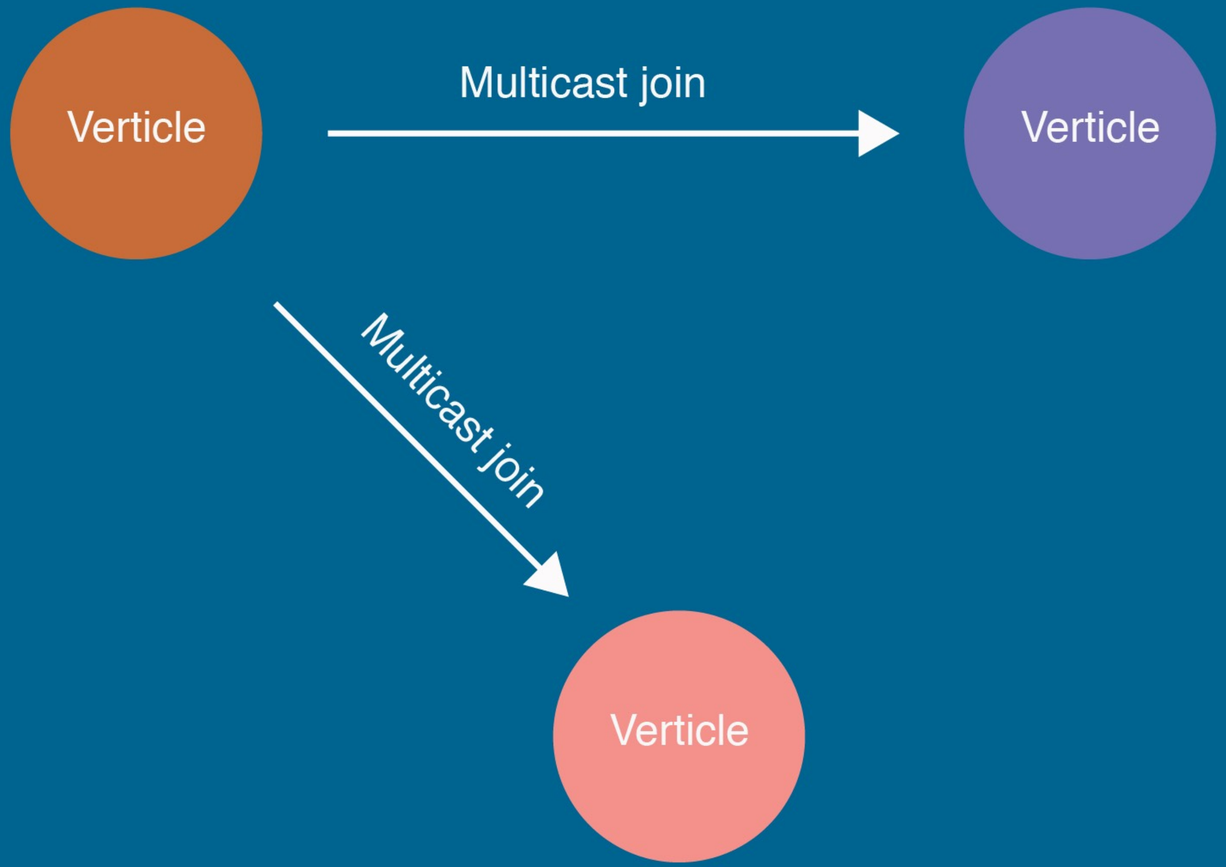
Service discovery API



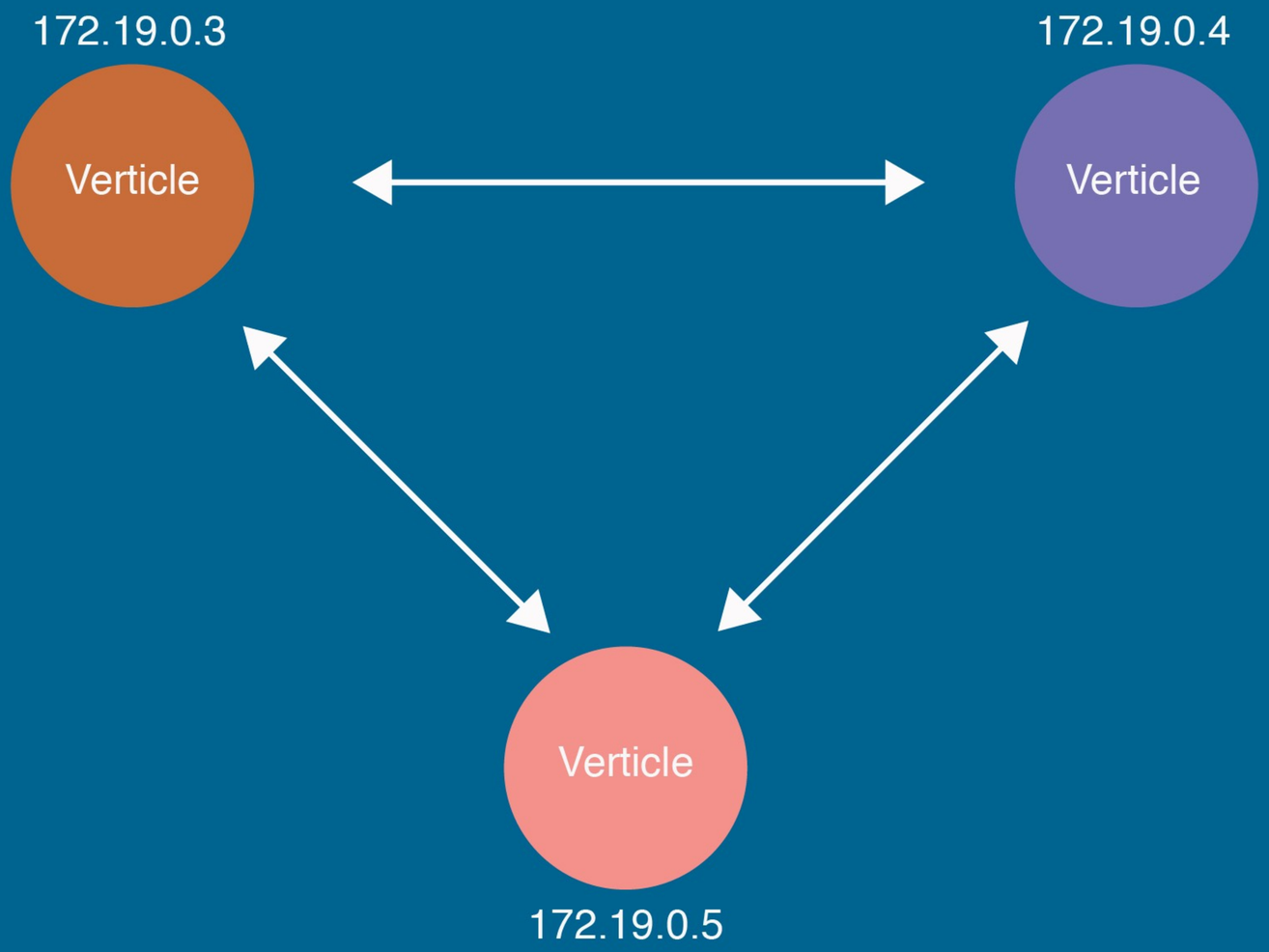
Service disovery API

Demo

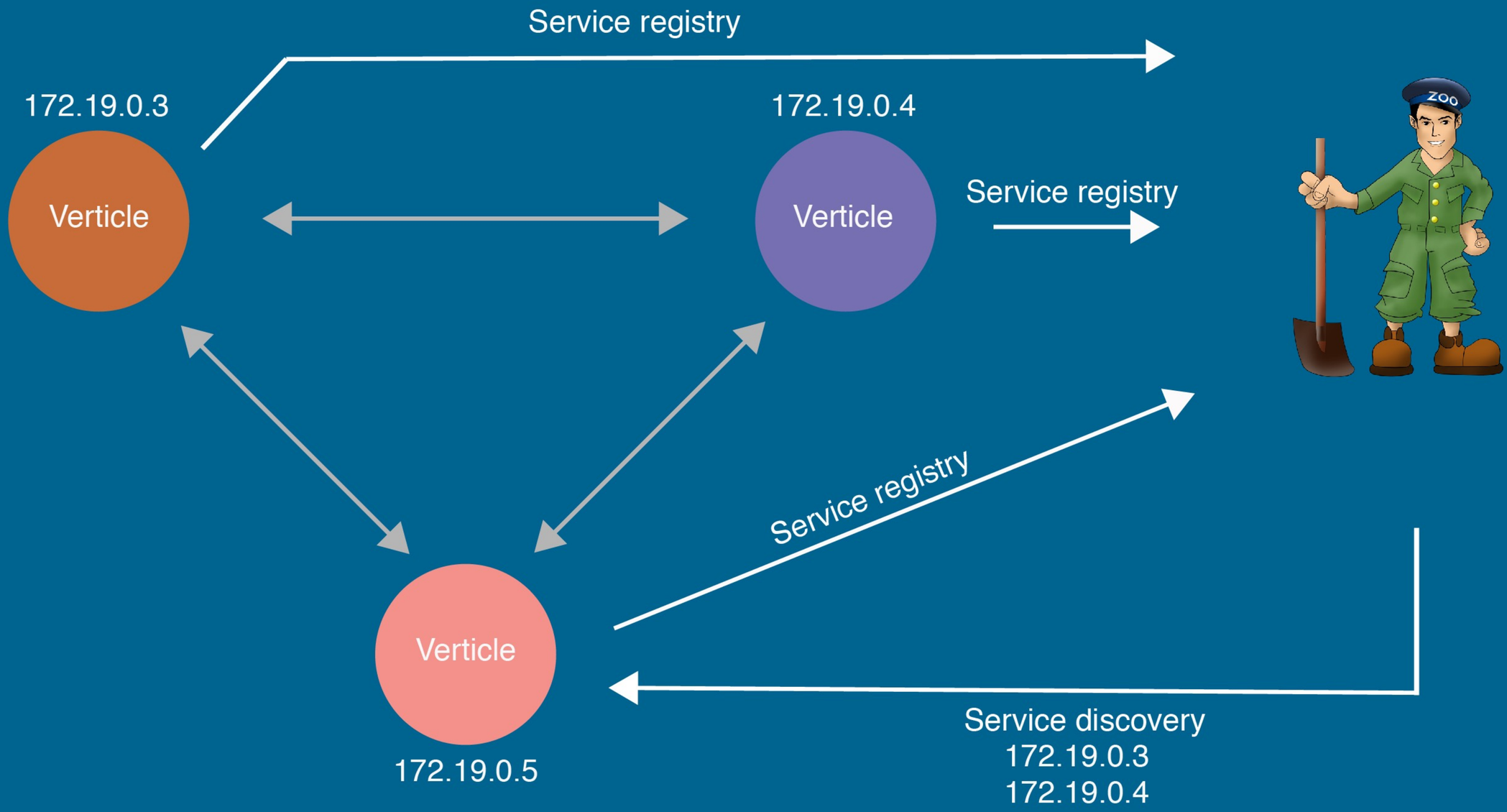
Service discovery bridge



Service discovery bridge



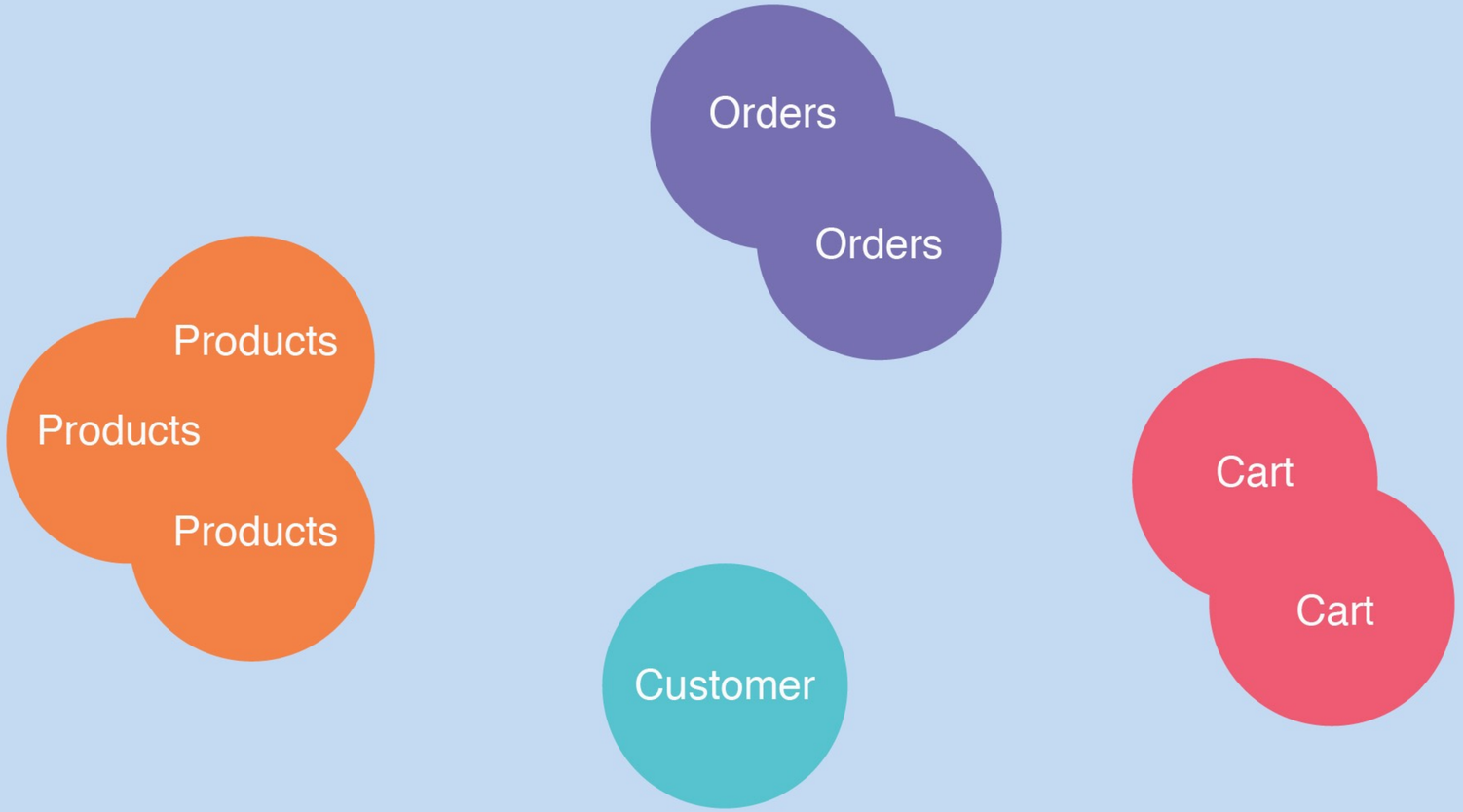
Service discovery Zookeeper



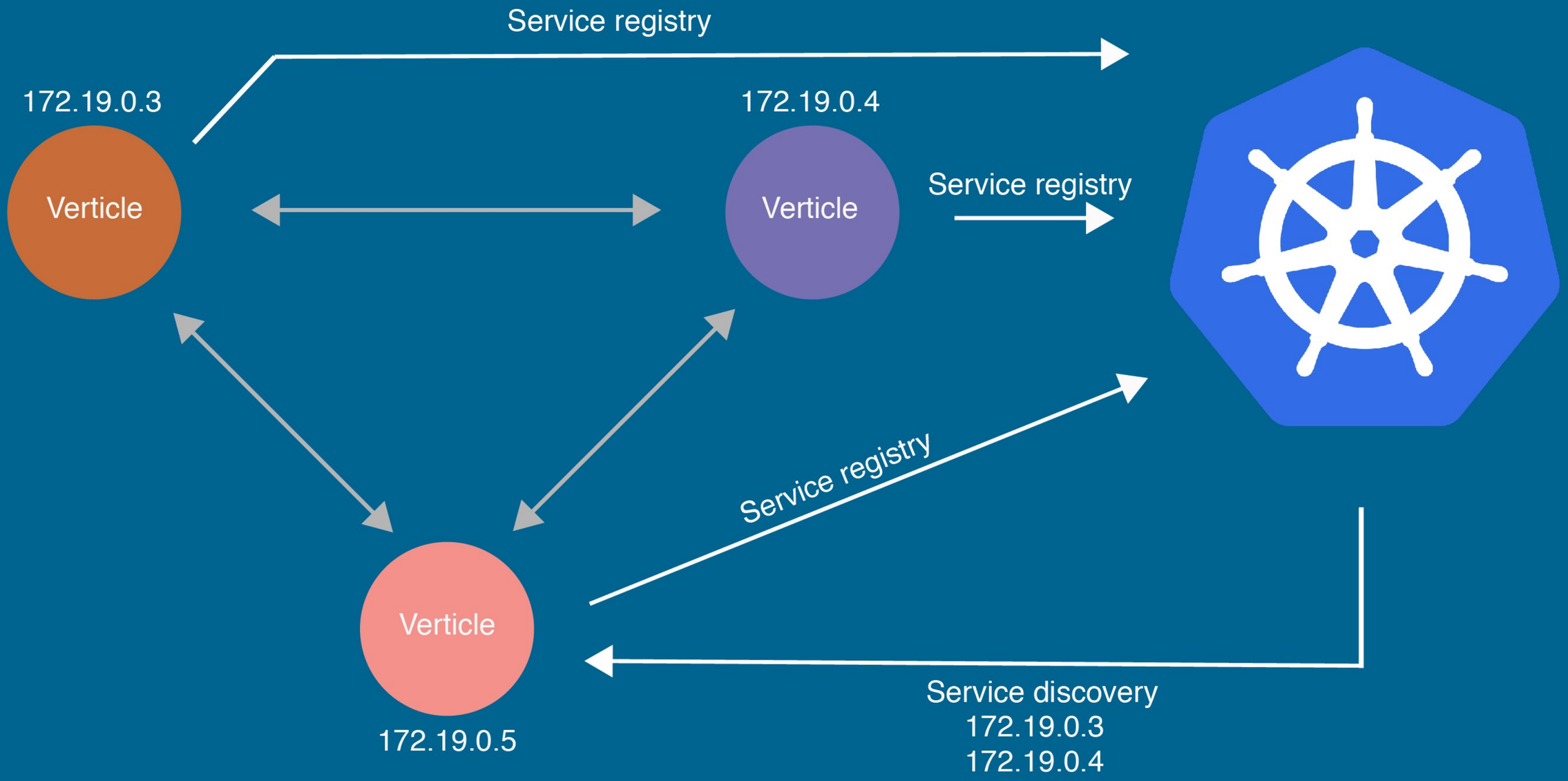
Service discovery Zookeeper

Demo

Microservice deployment



Service discovery Kubernetes



Kubernetes / Openshift

Demo

HTTP/1.1 vs HTTP/2

HTTP/2

- Server push
- Request pipelining
- Multiplexing
- Custom Frames

HTTP/2

Demo

Vert.x 3.4

- Scala & Kotlin
- Web client
- MQTT Server
- Kafka & Consul Clients
- gRPC

Conclusion

- Full stack microservice solution
- Integrations options
- Service discovery
- HTTP2

Event-driven microservices with Vert.x and Kubernetes

Andy Moncsek - Wednesday 3:30 PM

Better performance with HTTP/2

Julien Viet - Thursday 10:00 AM

Vert.x: From Zero to (Micro-) Hero

Julien Viet - Thursday 1:50 PM

“Do one thing, and do it well.”

- unix philosophy

http://edegier.nl/presentations/
erwin@edegier.nl

SOGETI

Web client

- JSON encoding / decoding
- Request parameters
- Unified error handling
- Form submissions
- Request / response pumping

Web client

Demo