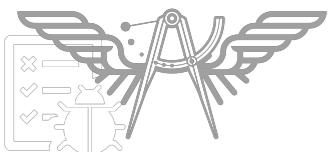


Develop Microservices





Somkiat Puisungnoen

Somkiat Puisungnoen

Update Info 1

View Activity Log 10+

...
Timeline About Friends 3,138 Photos More

When did you work at Opendream?
... 22 Pending Items

Post Photo/Video Live Video Life Event

What's on your mind?

Public Post

Intro
Software Craftsmanship

Software Practitioner at สยามชัมนาภิกิจ พ.ศ. 2556

Agile Practitioner and Technical at SPRINT3r

Somkiat Puisungnoen 15 mins · Bangkok · ⚙️

Java and Bigdata



Page

Messages

Notifications 3

Insights

Publishing Tools

Settings

Help ▾



somkiat.cc

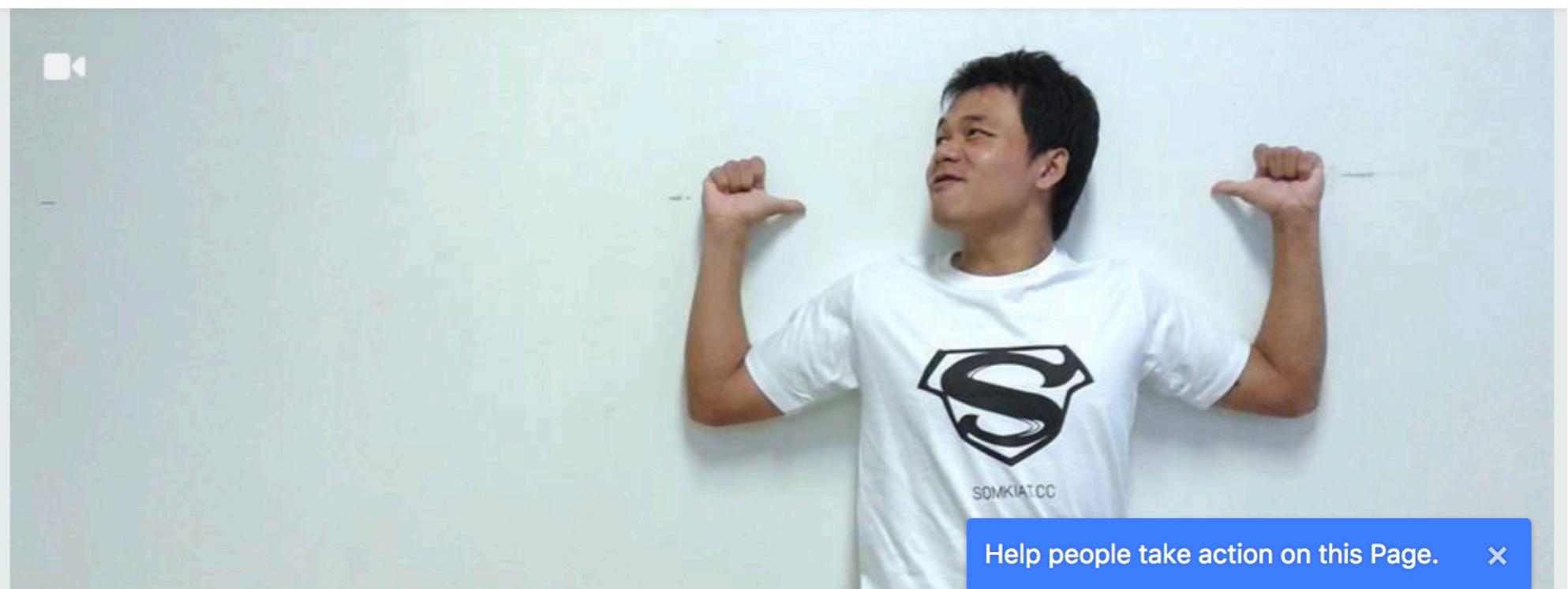
@somkiat.cc

Home

Posts

Videos

Photos



Help people take action on this Page. 

+ Add a Button



Agenda

Recap Microservice

Properties of Microservice

Microservice 1.0 - 4.0

How to develop Microservice ?

How to test Microservice ?

12-factors app

Workshop



Recap Microservice



Microservices ?

More/Low splitting
More network interaction
Data storing and sharing
Compatibility issues
Testing
Operation & Monitoring

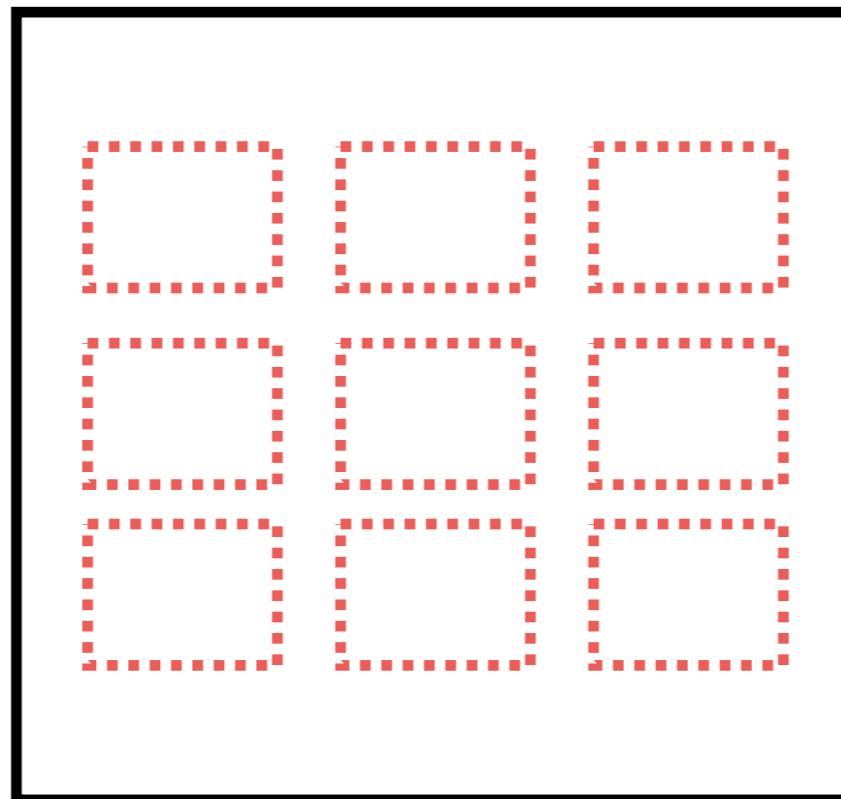


Monolith

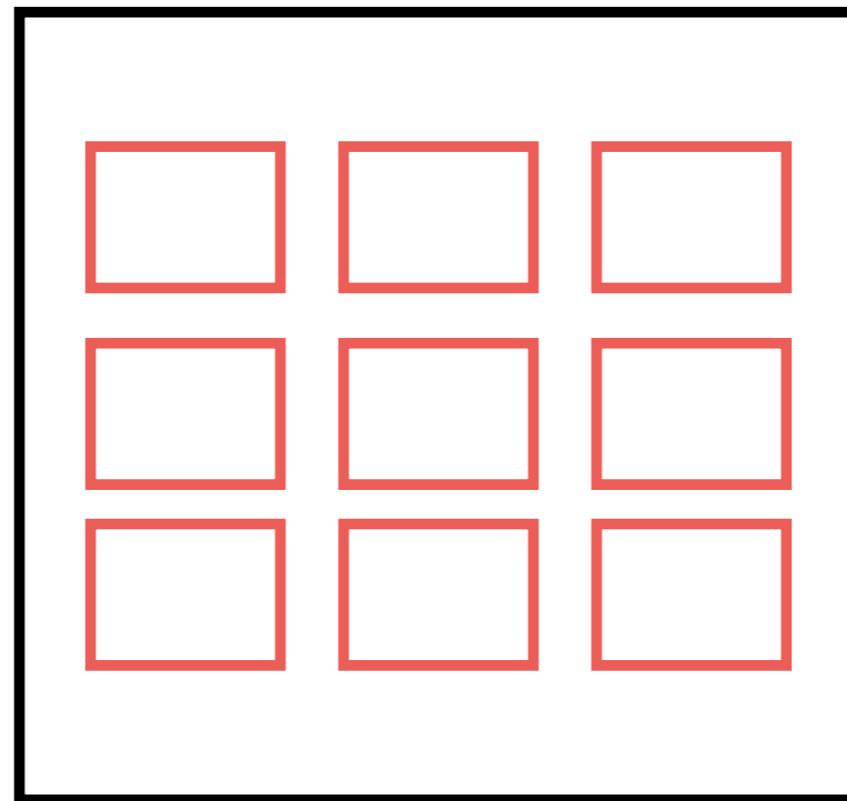
App



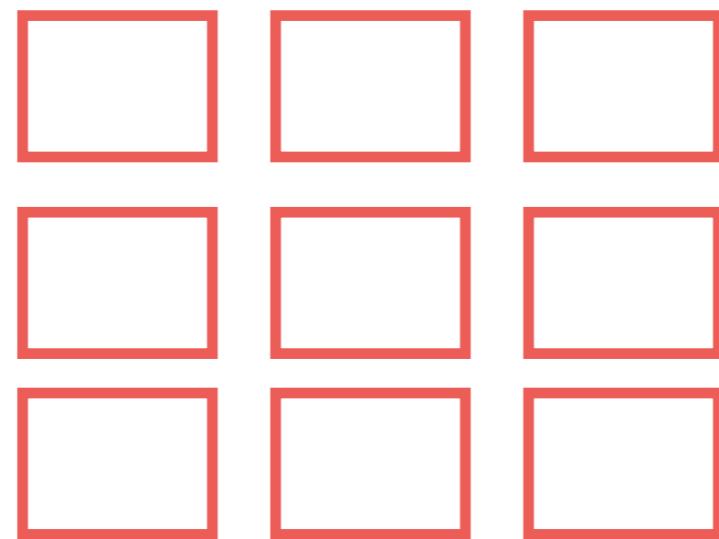
Modules



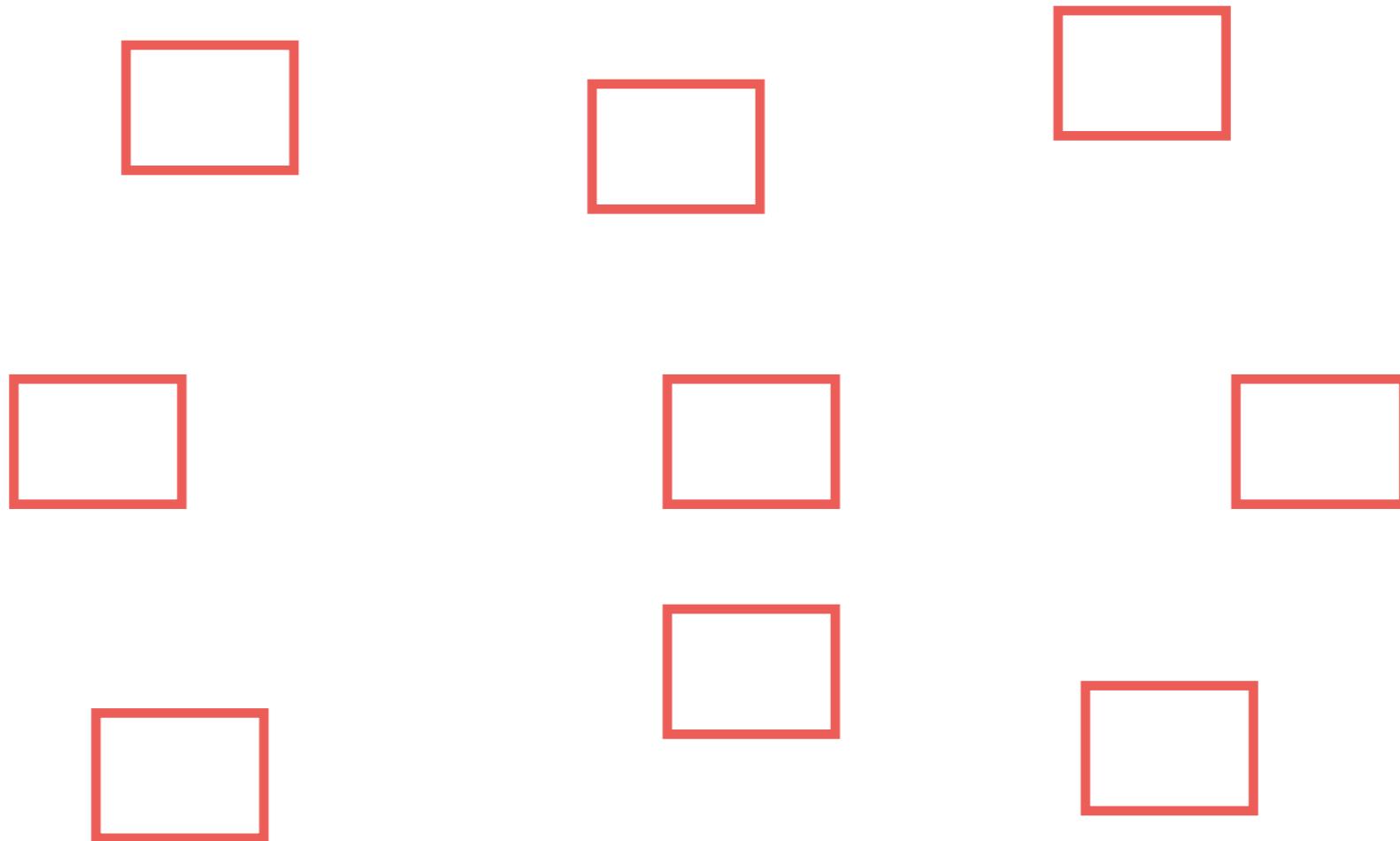
Microservices



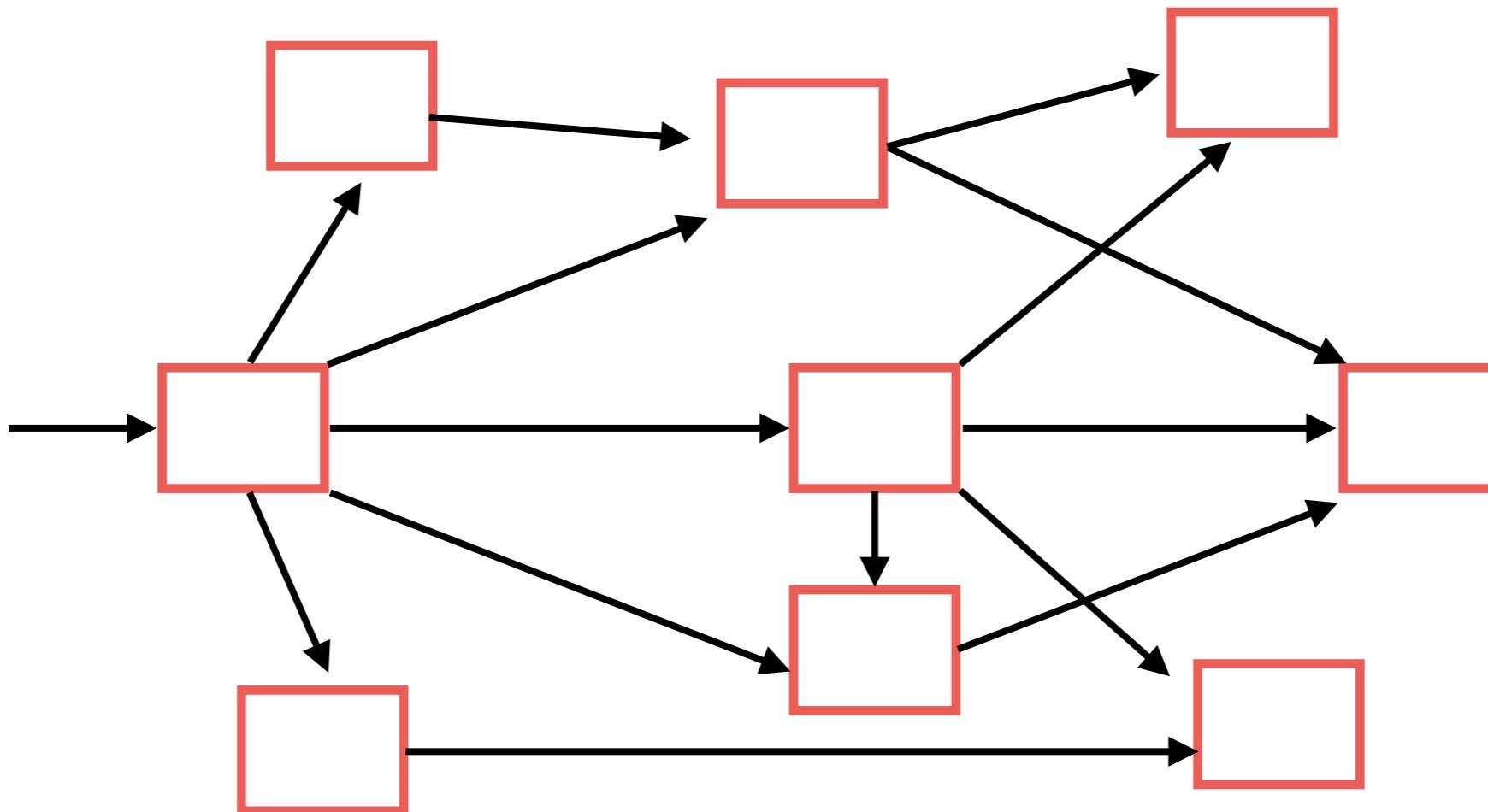
Microservices



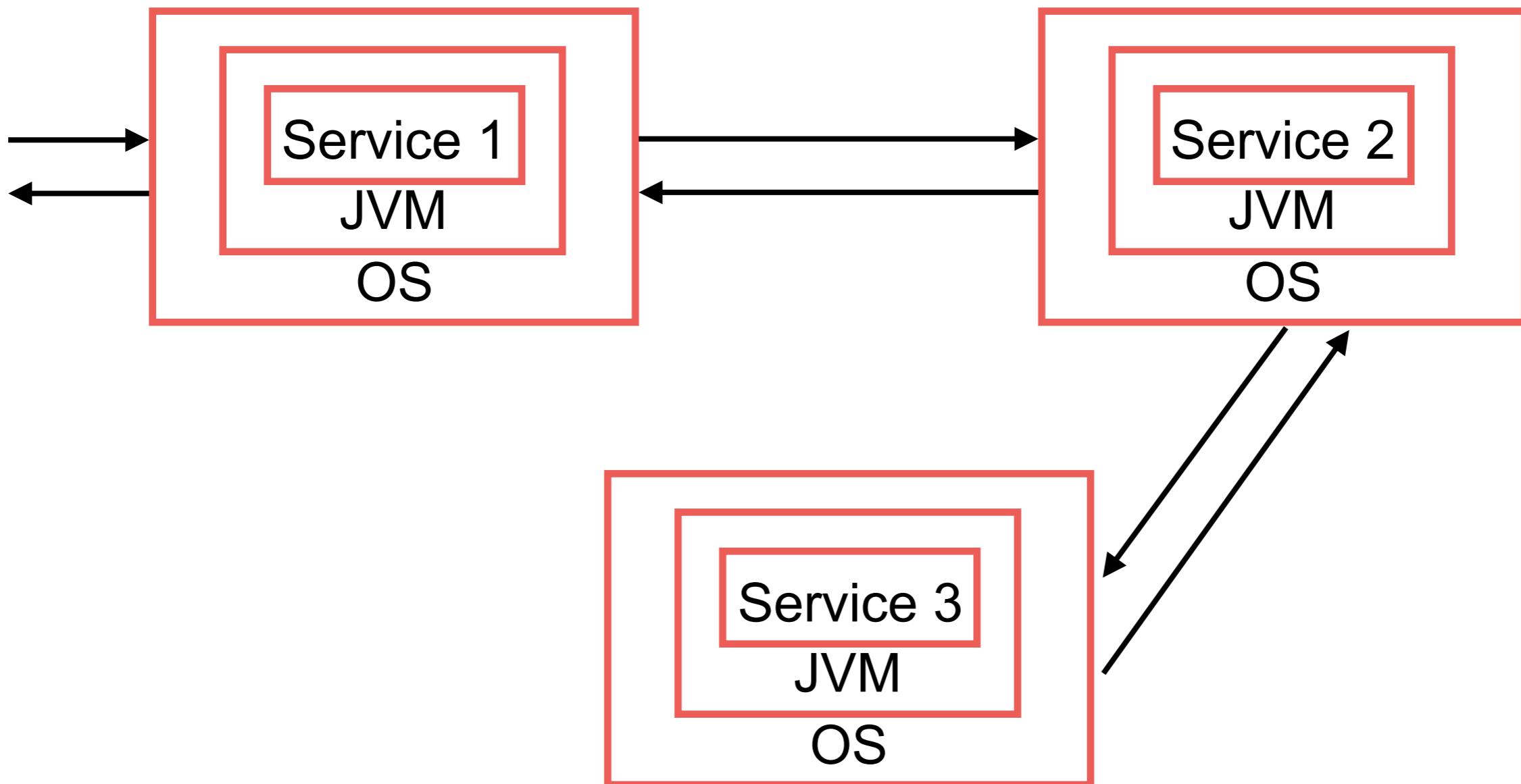
Microservices



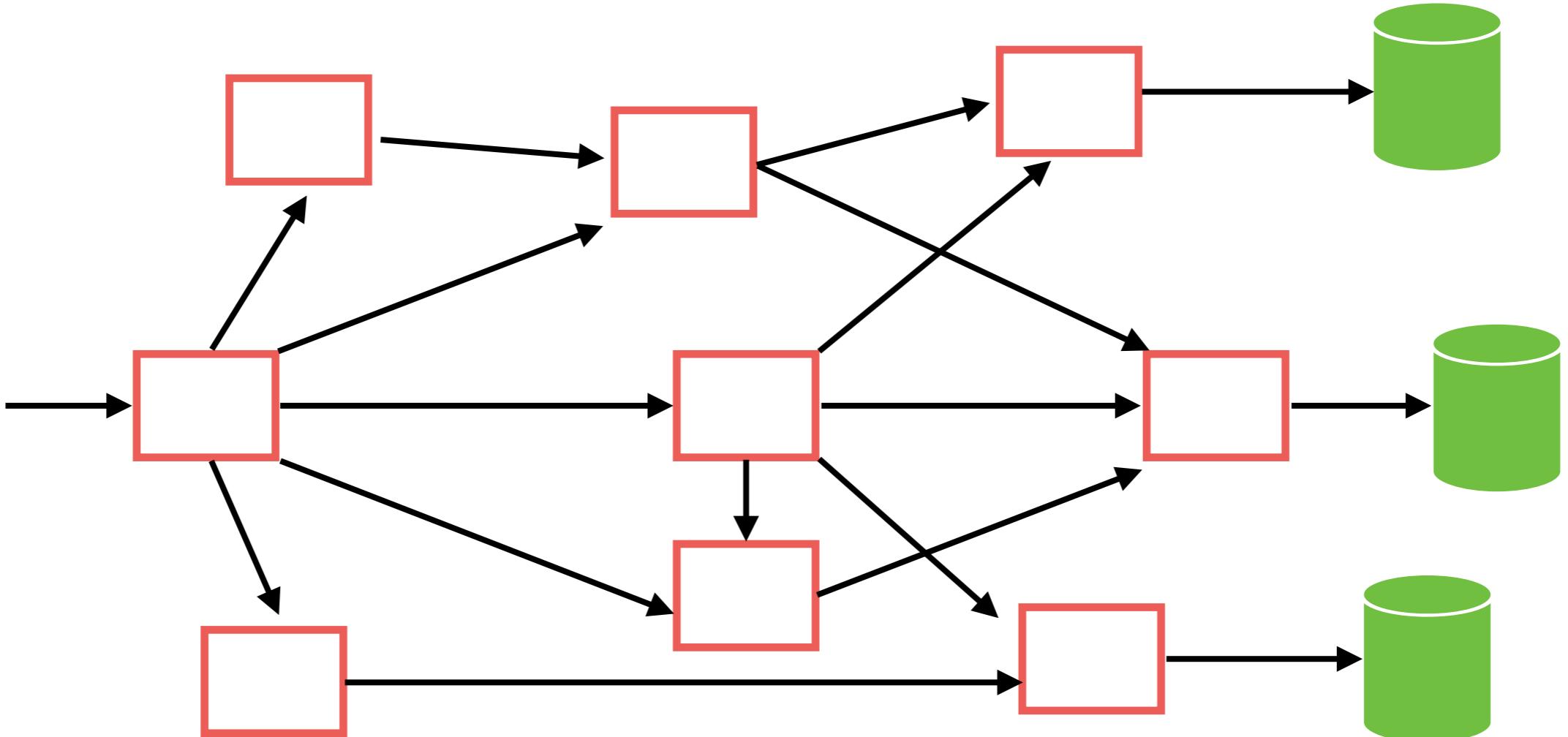
Microservices == Distributed



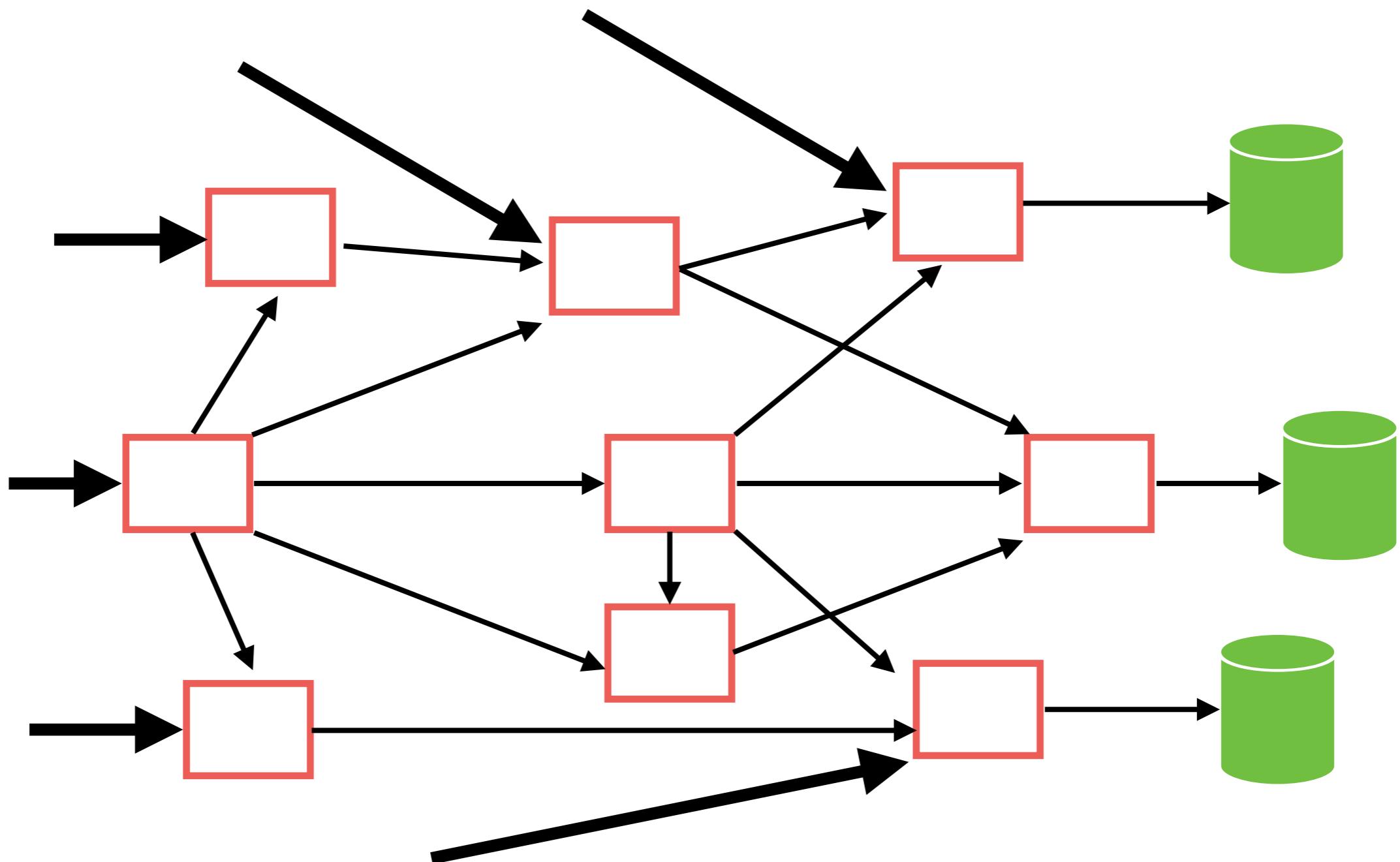
Network of services



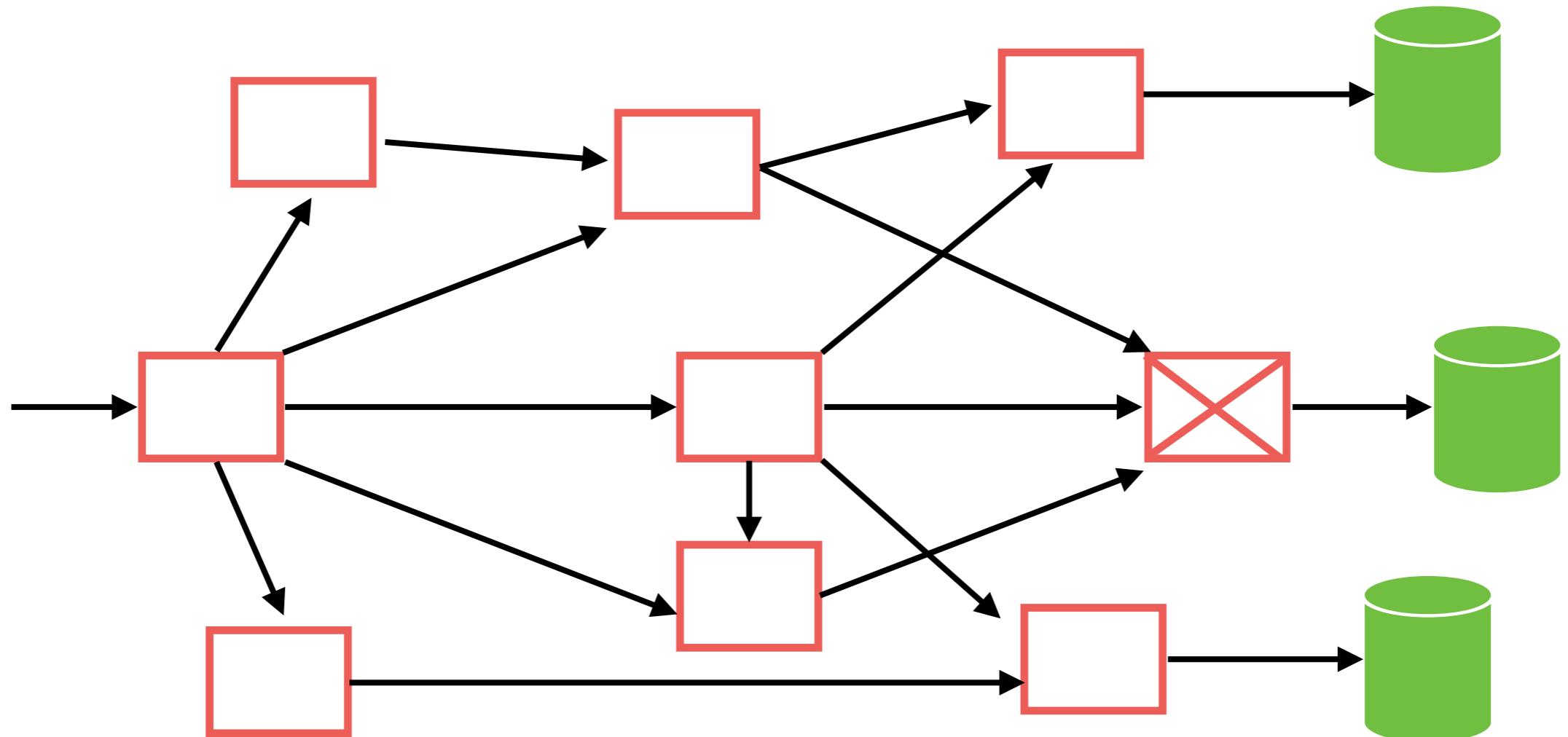
Own data



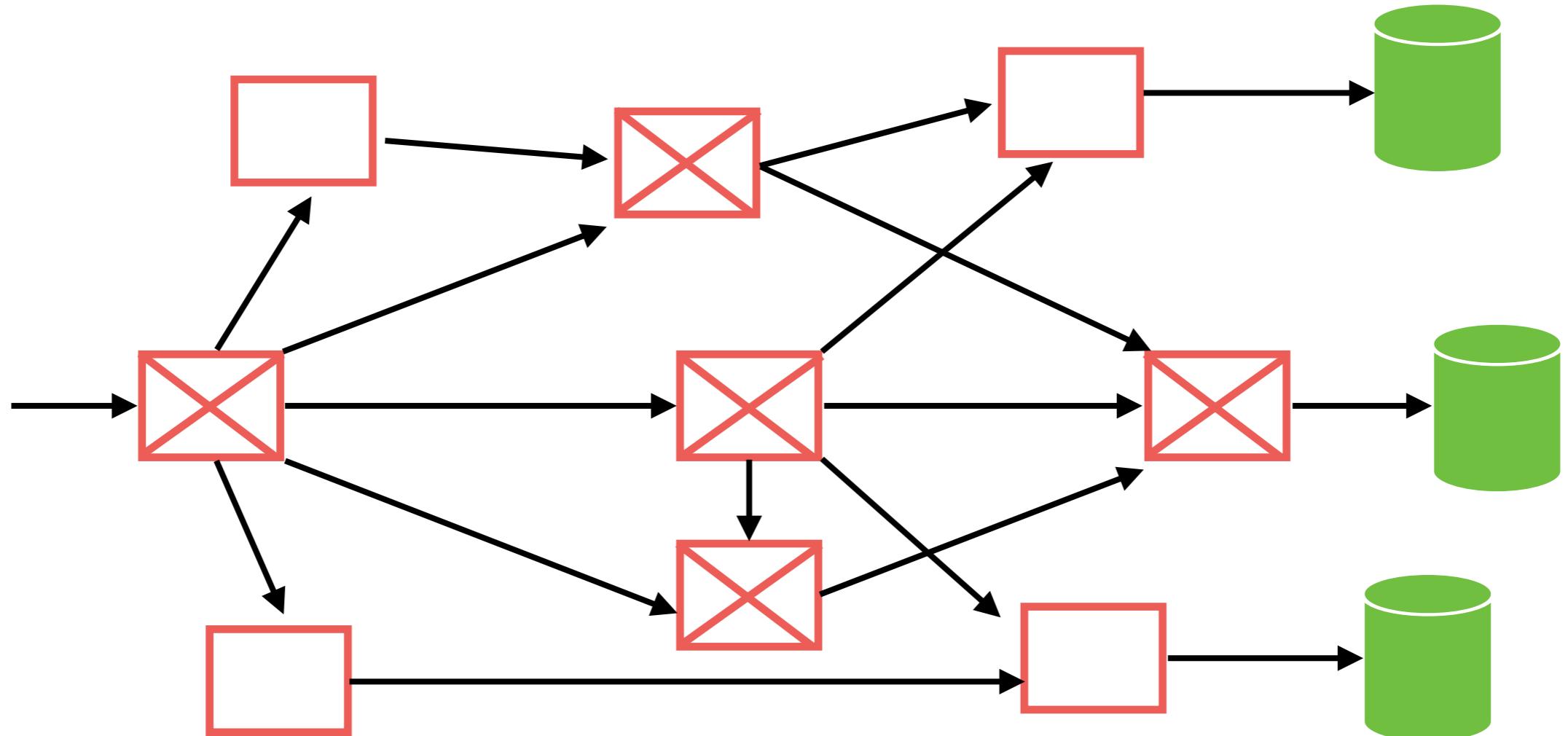
Multiple entry points



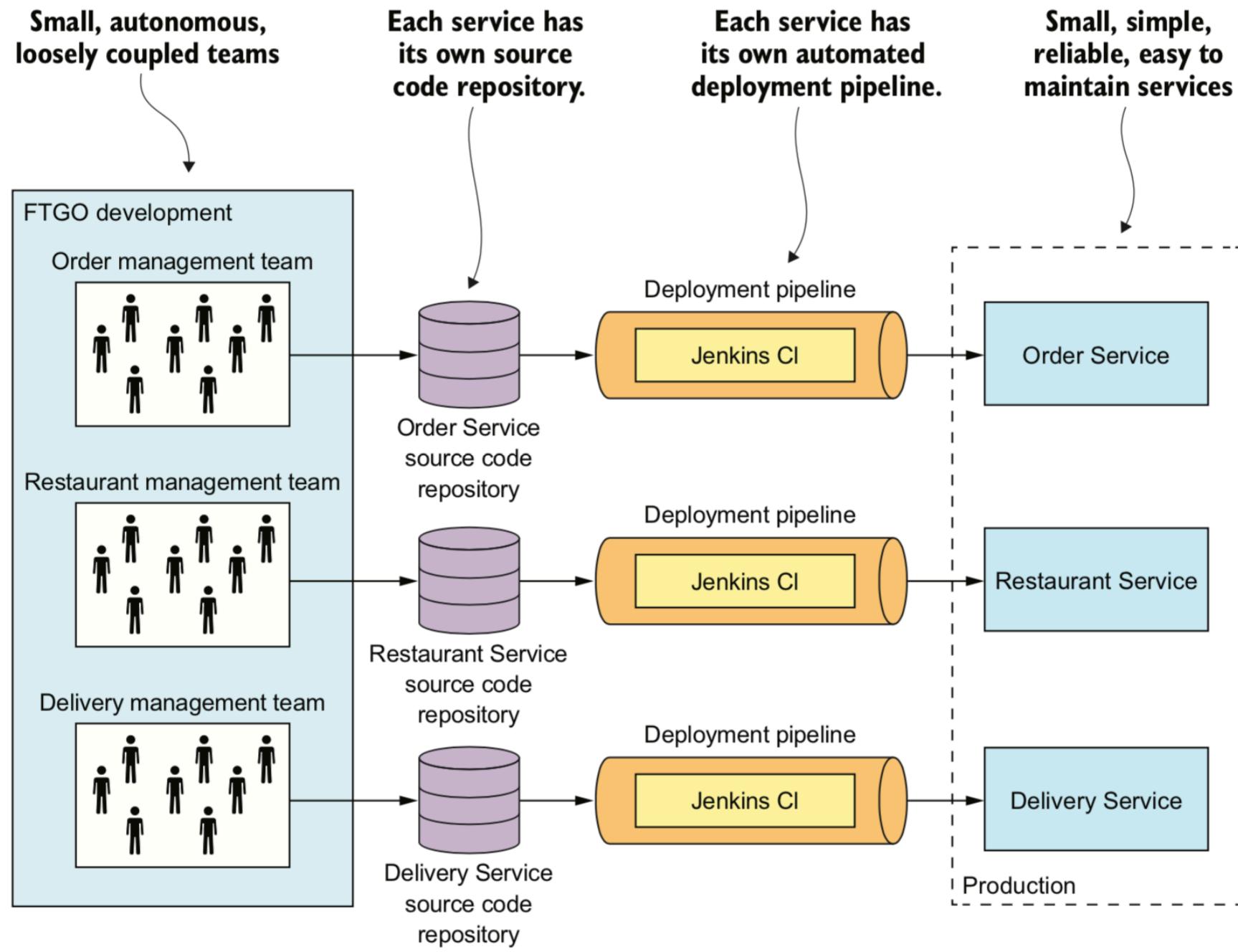
Failure !!



Cascading Failure !!



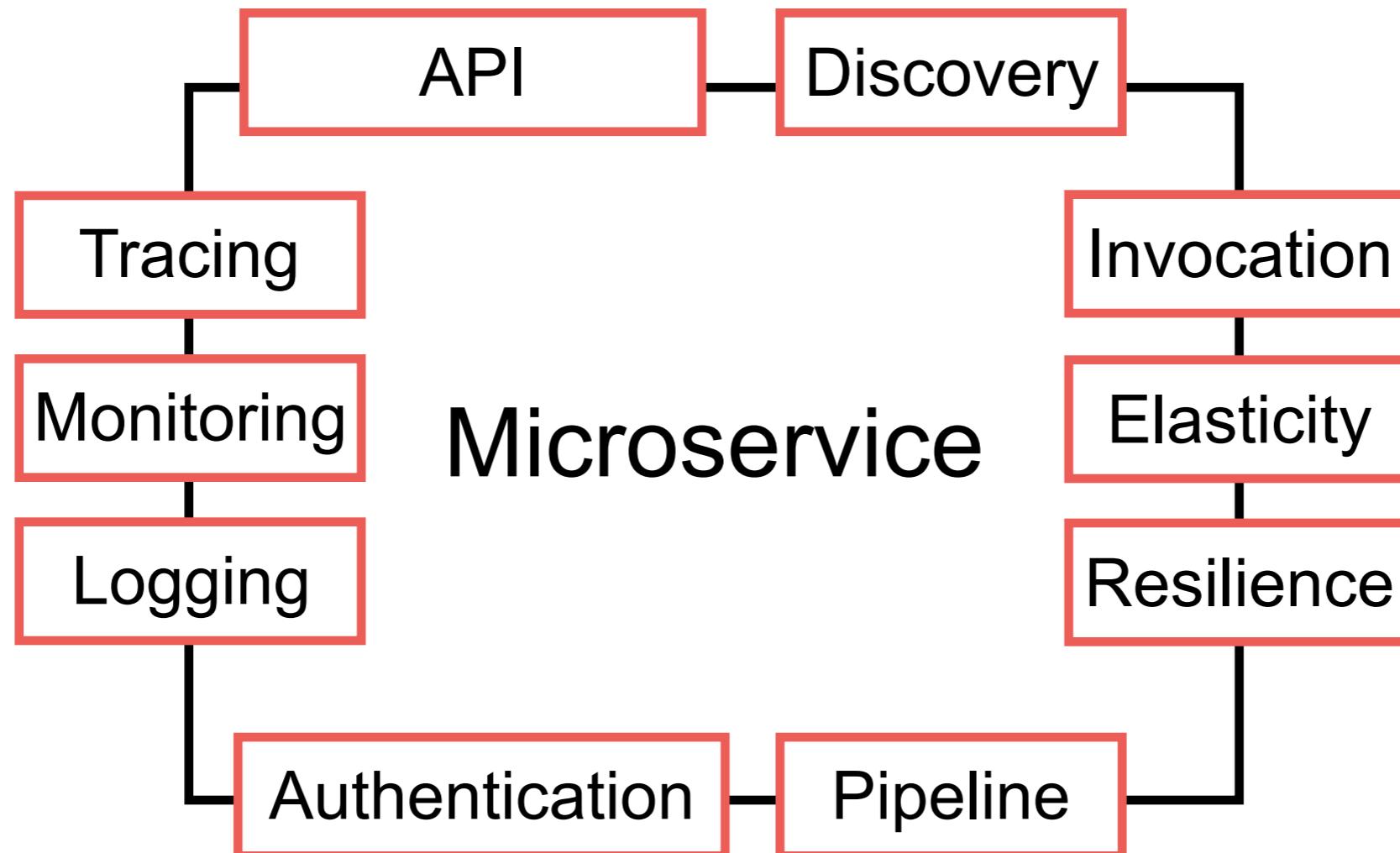
Each team develop, test, and deploy their services independently



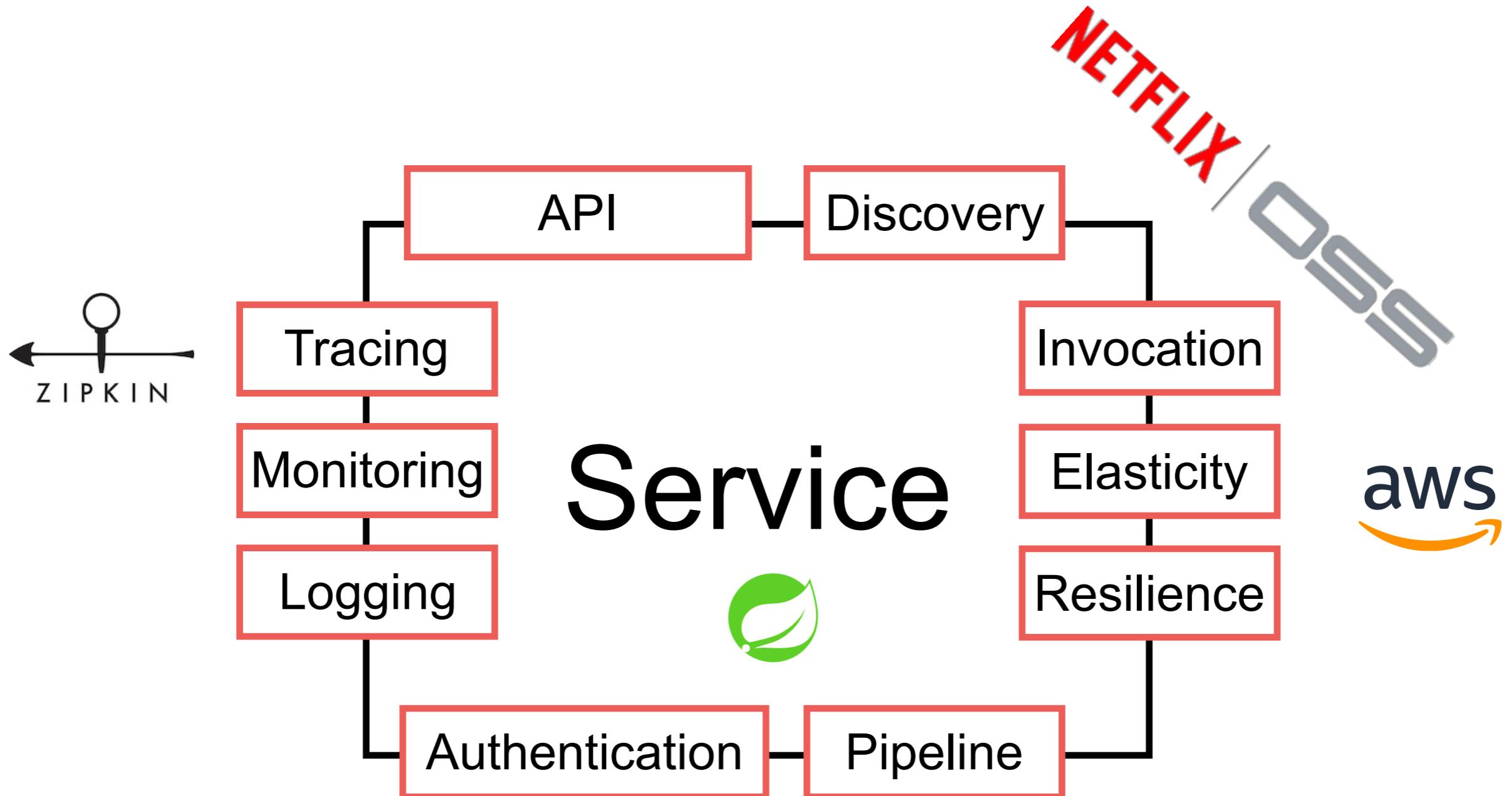
Develop Microservice



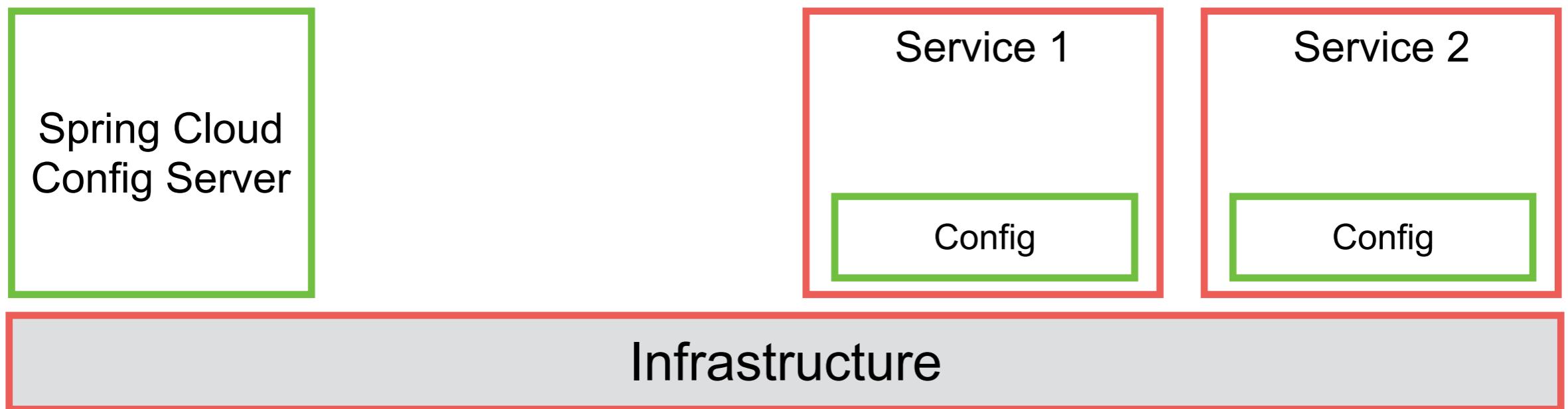
Properties of Microservice



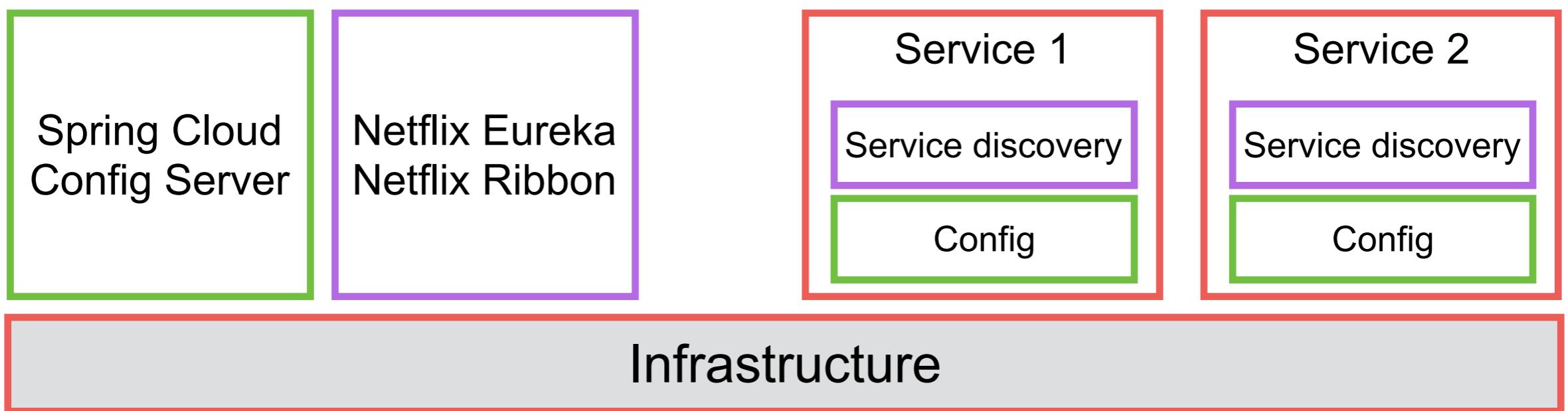
Microservice 1.0



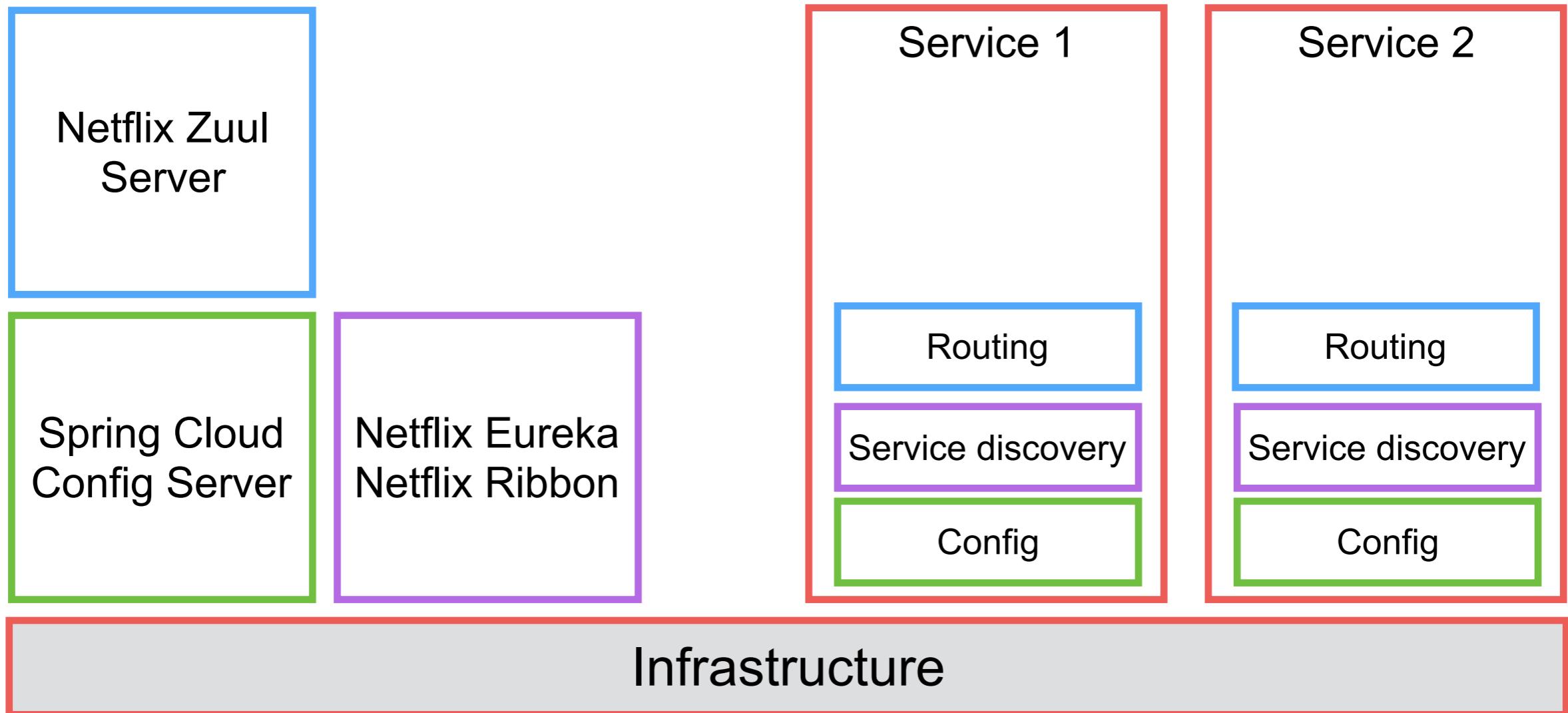
Configuration



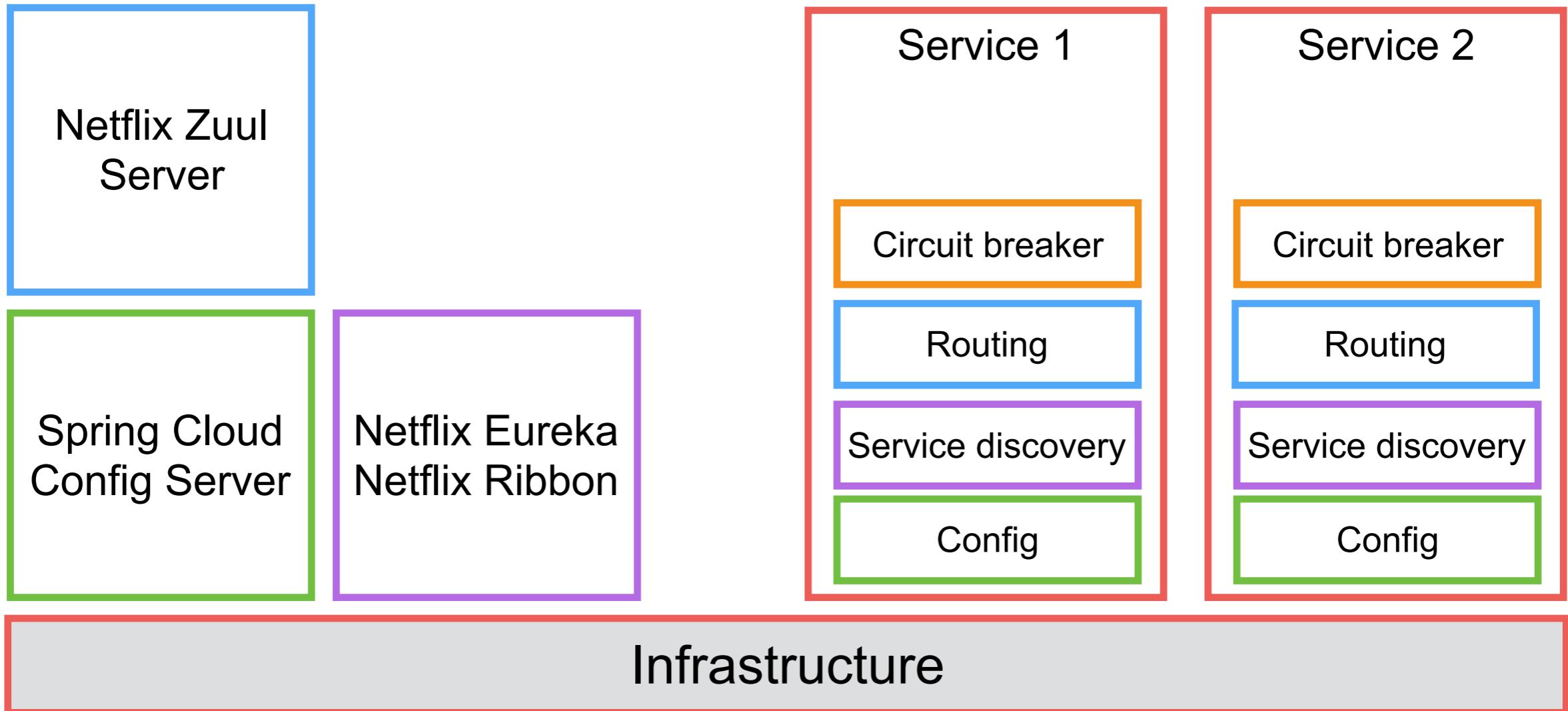
Service Discovery



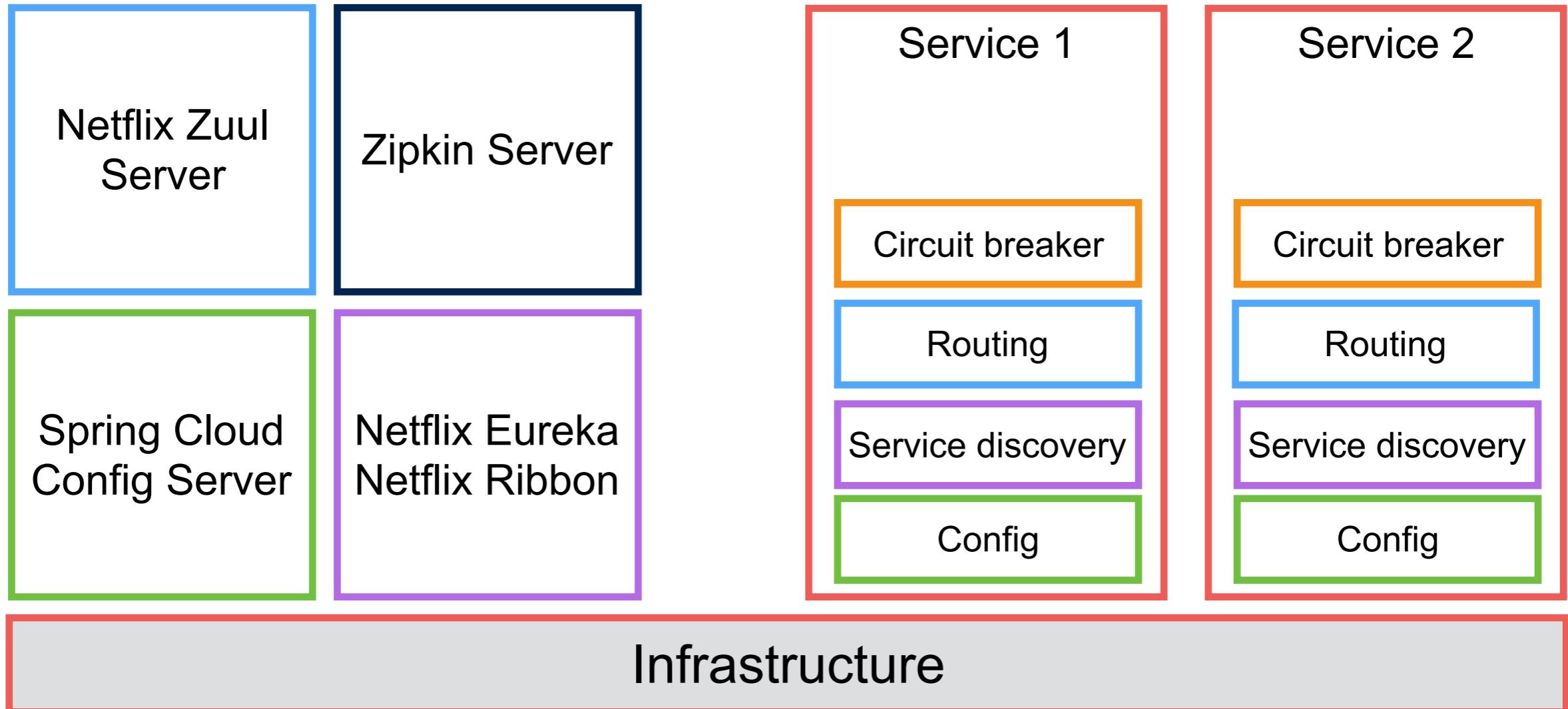
Dynamic Routing



Fault Tolerance



Tracing and Visibility



Microservice 1.0

JVM only

Add libraries to your code/service

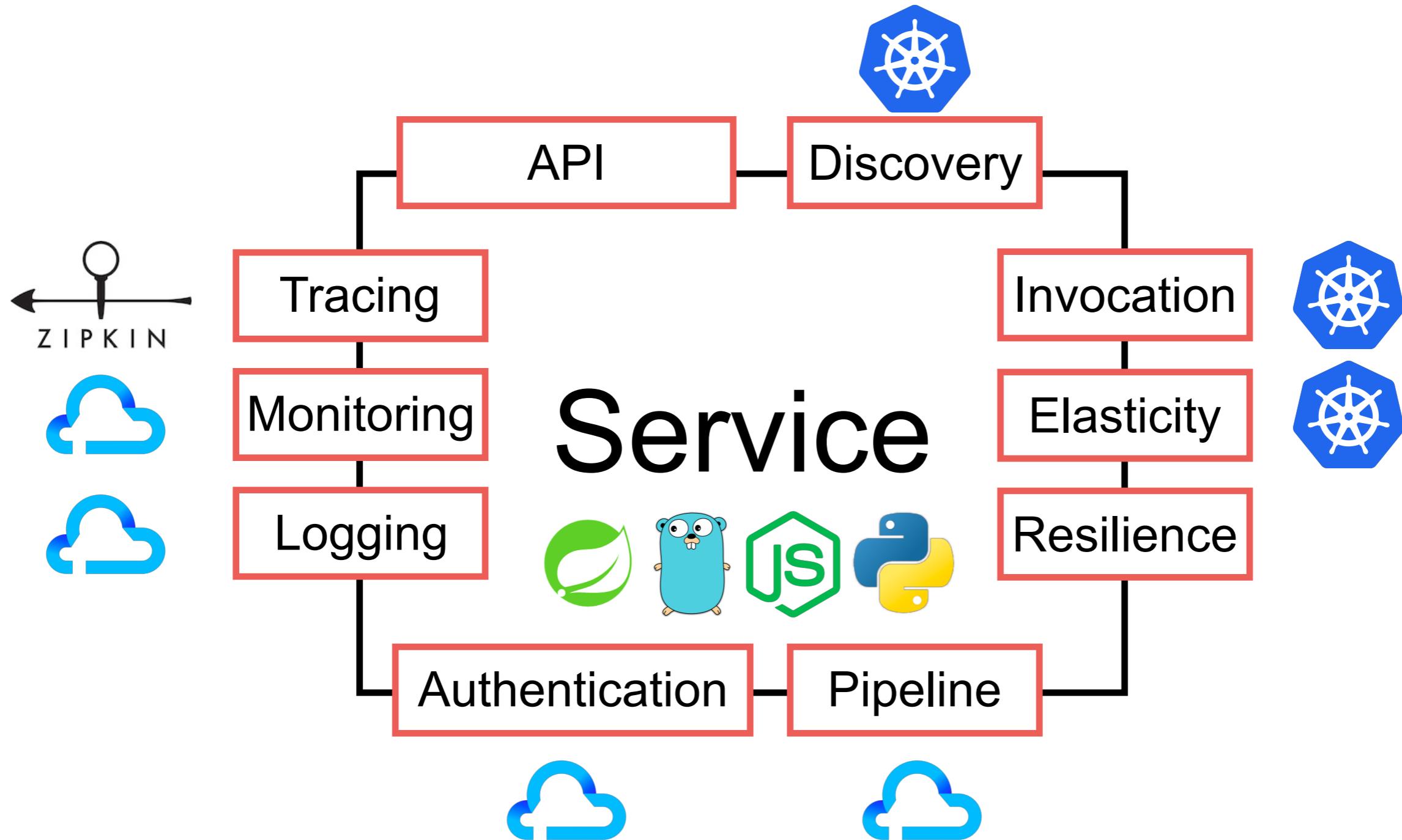


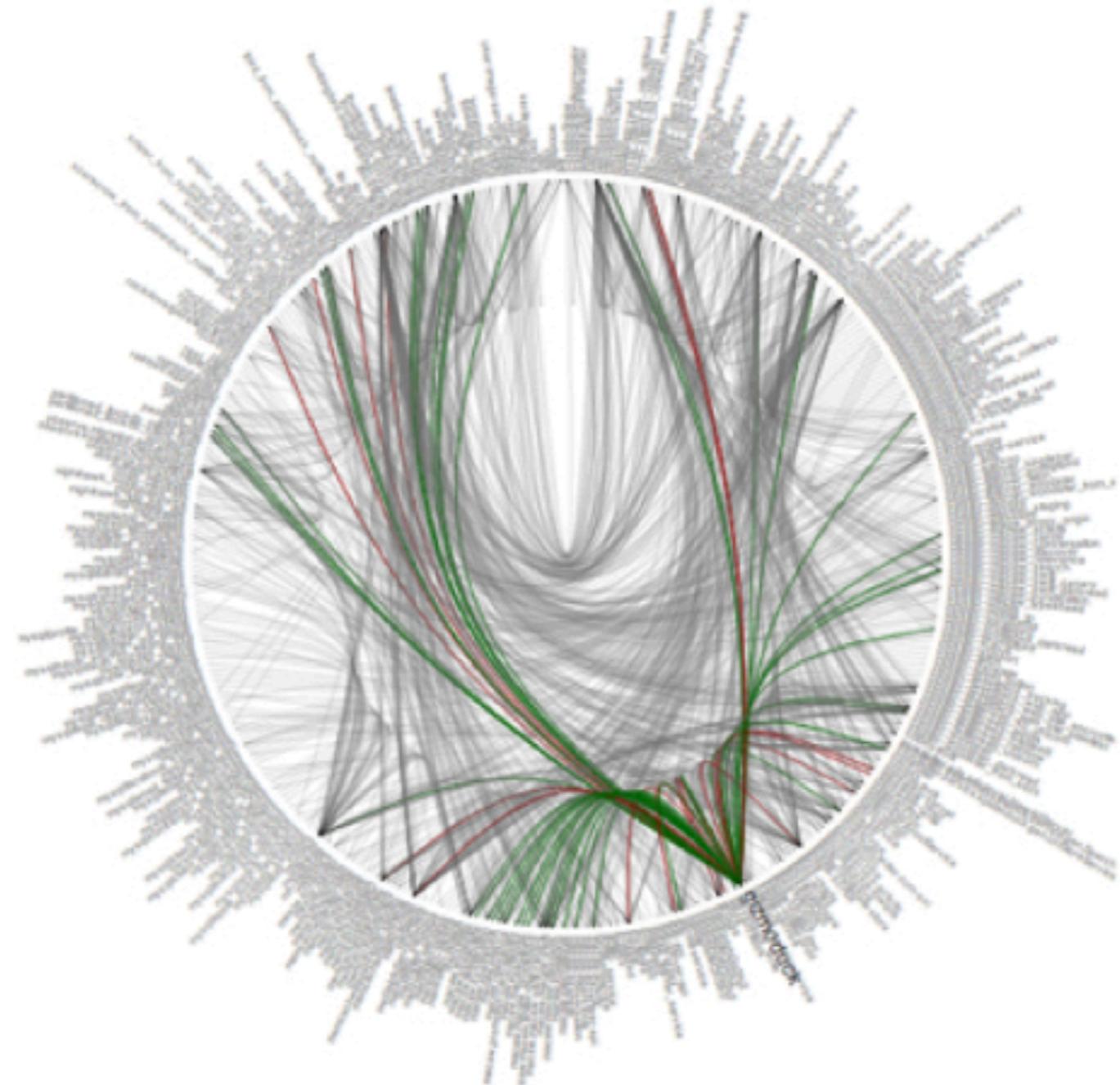


kubernetes



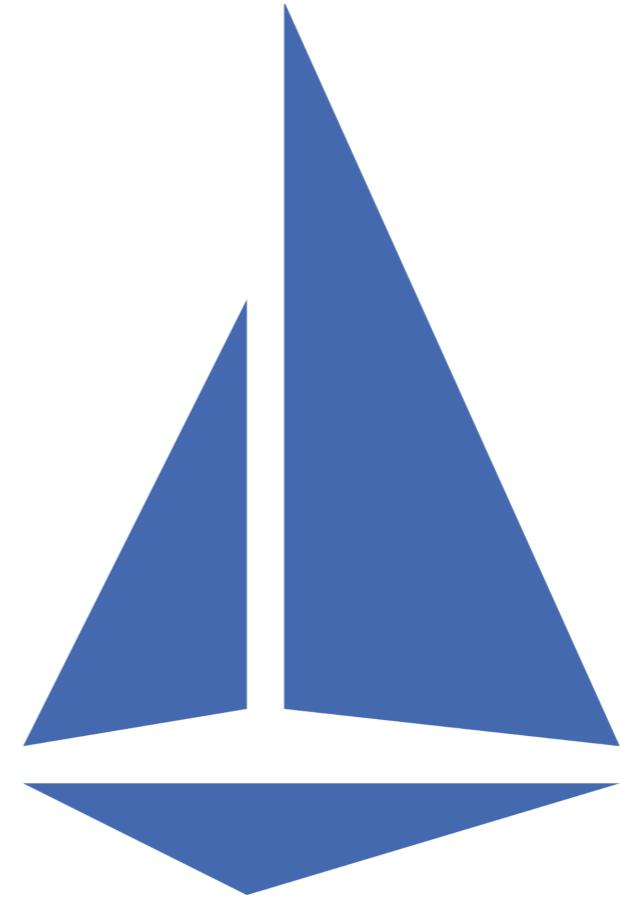
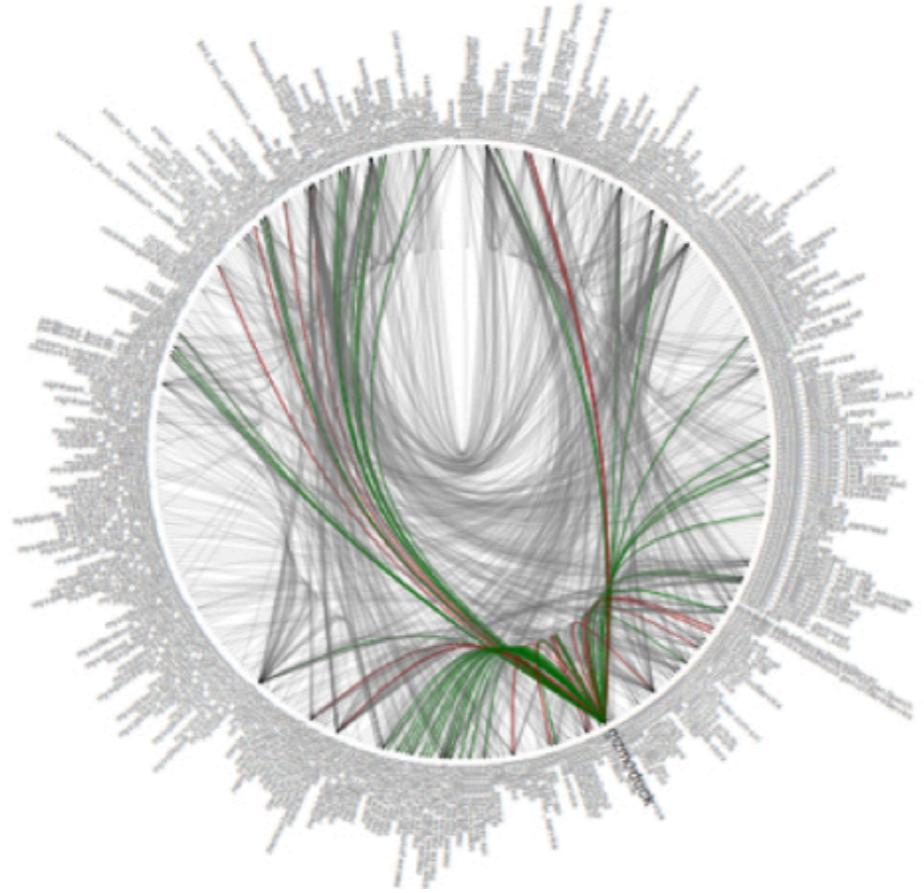
Microservice 2.0





Service Mesh

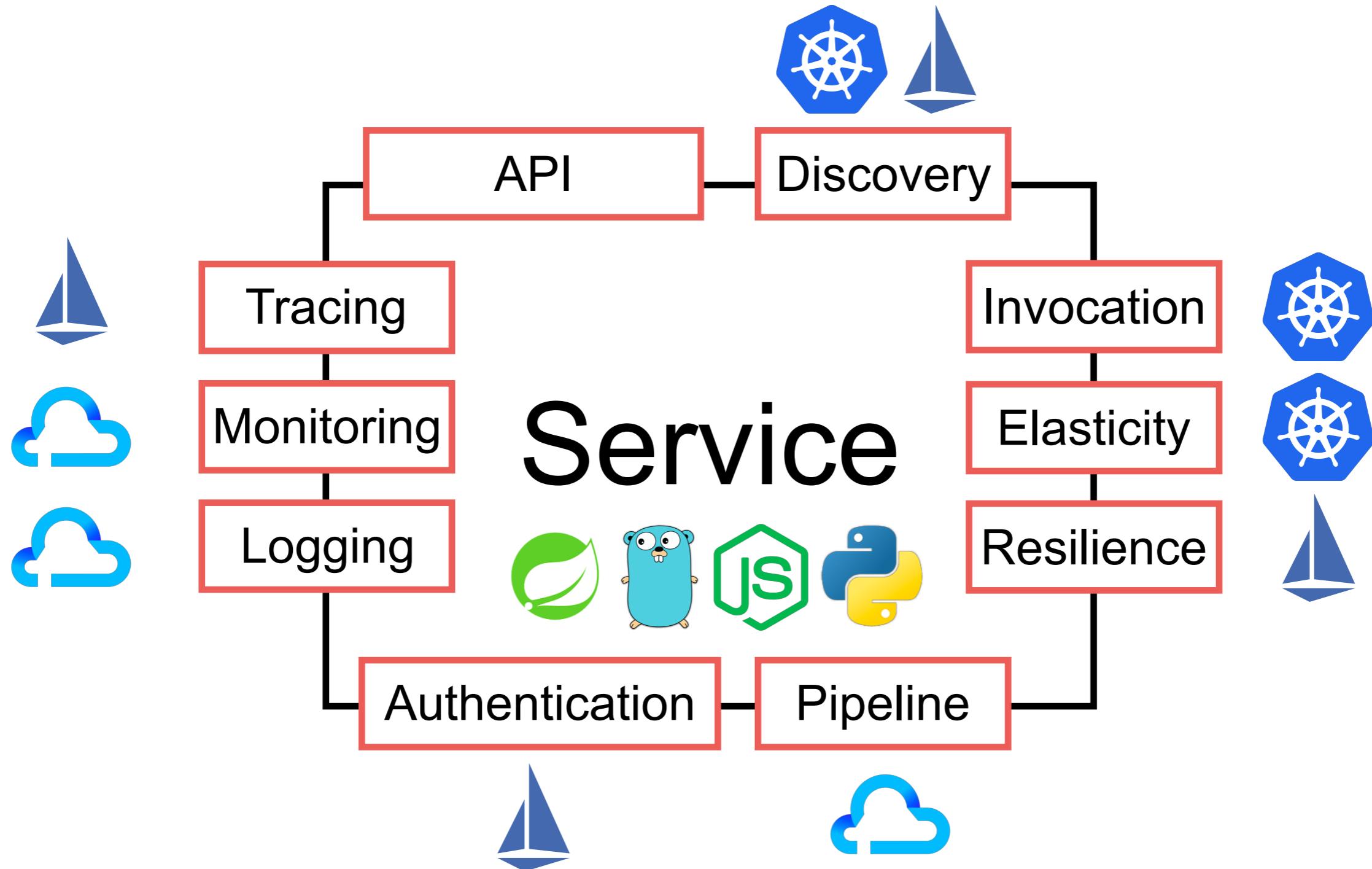




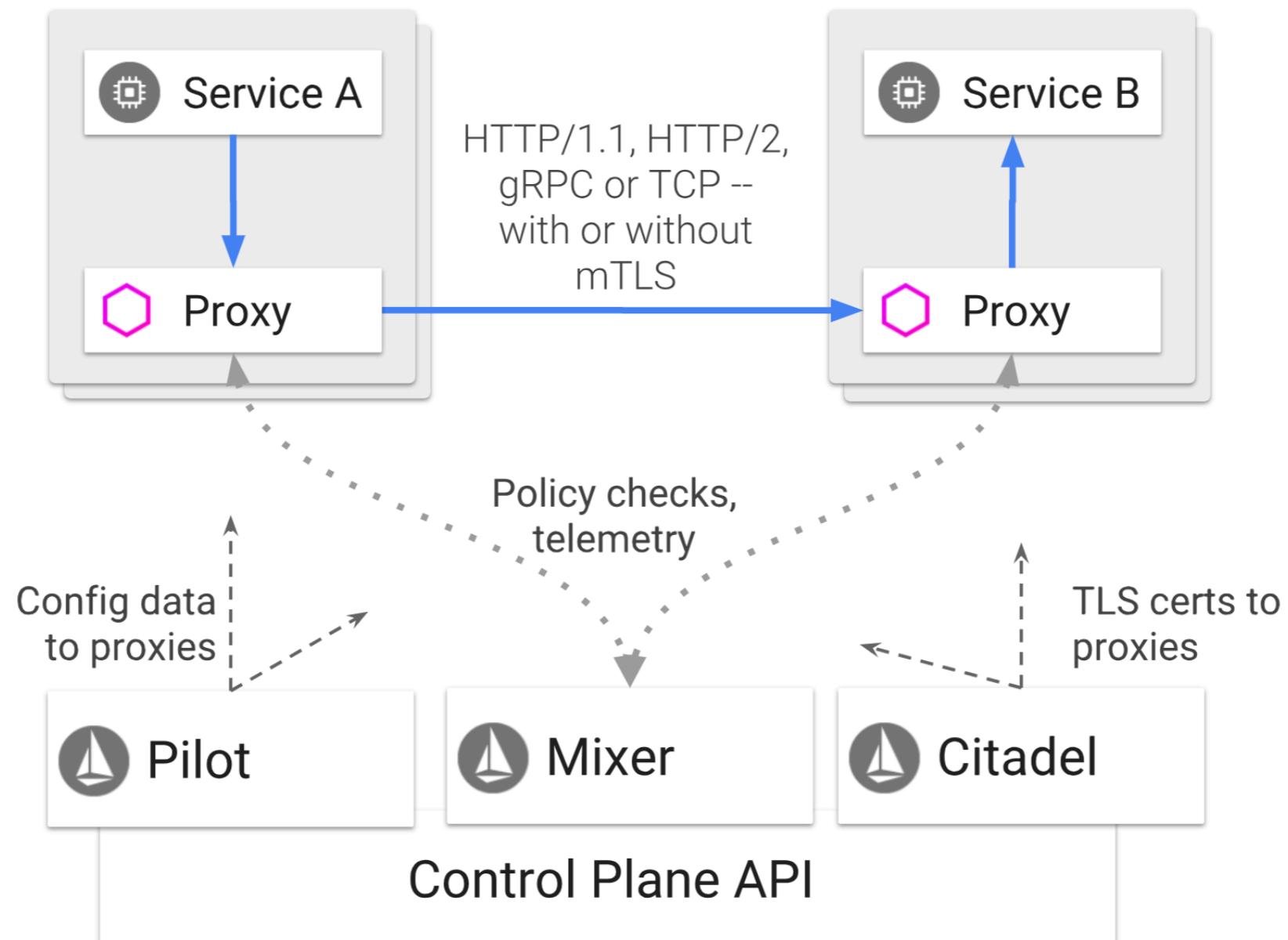
Service Mesh with Istio



Microservice 3.0



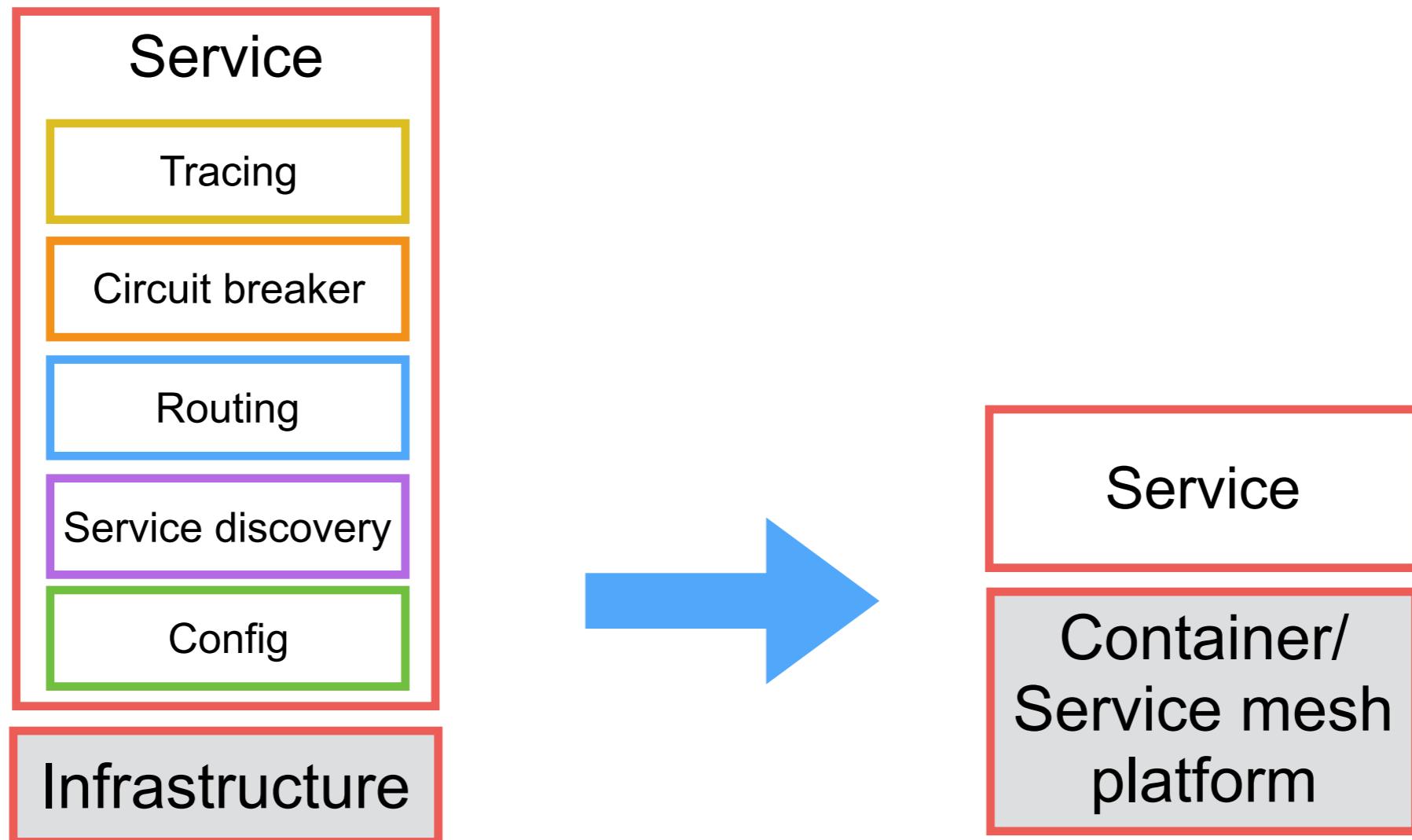
Istio



<https://istio.io/docs/concepts/what-is-istio/>



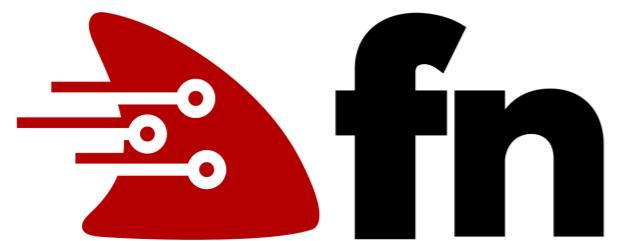
Microservice Evolution



Function-as-a-Service (FaaS)



Microservice 4.0 == FaaS



OPENFAAS



APACHE
OpenWhisk™



Microservice 1.5

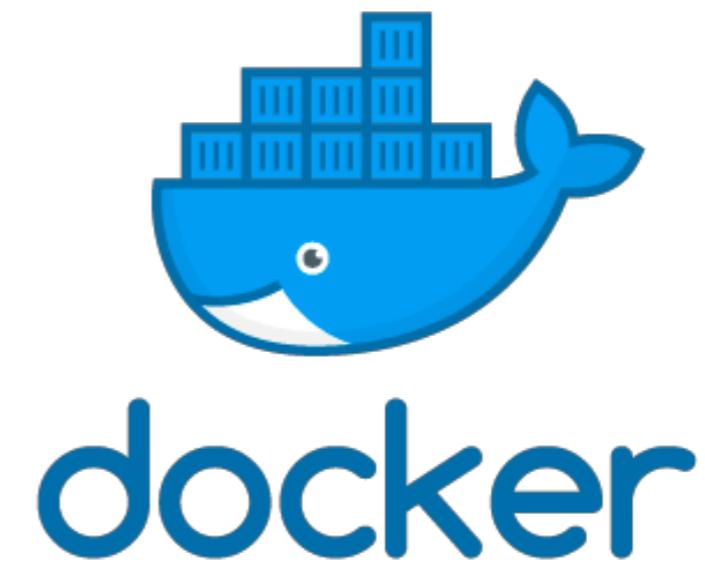


Workshop

Microservice 1.5



+



Twelve-Factor App



Twelve-Factor App

<https://12factor.net/>

Initial to build app for Heroku

Principles to cloud and container native app



1. Codebase

Codebase must be tracked in version control
and will have many deploy



2. Dependencies

Dependencies are explicitly declared and isolated

Use dependency management tools to shared
libraries



3. Configuration

Store configuration in the environment

Add configuration in environment variables or config files



4. Backing services

Treat backing services as an attached resource

Should easy to deploy and change



5. Build, Release, Run

Always have a build and deploy strategy

Build strategies for repeated builds, versioning of running system and rollback



6. Processes

Execute the application as a stateless process



7. Port Binding

Expose services via port bindings



8. Concurrency

Scale out with the process model



9. Disposability

Quick application startup and shutdown times
Graceful shutdown



10. Dev/prod parity

Application is treated the same way in dev, staging and production

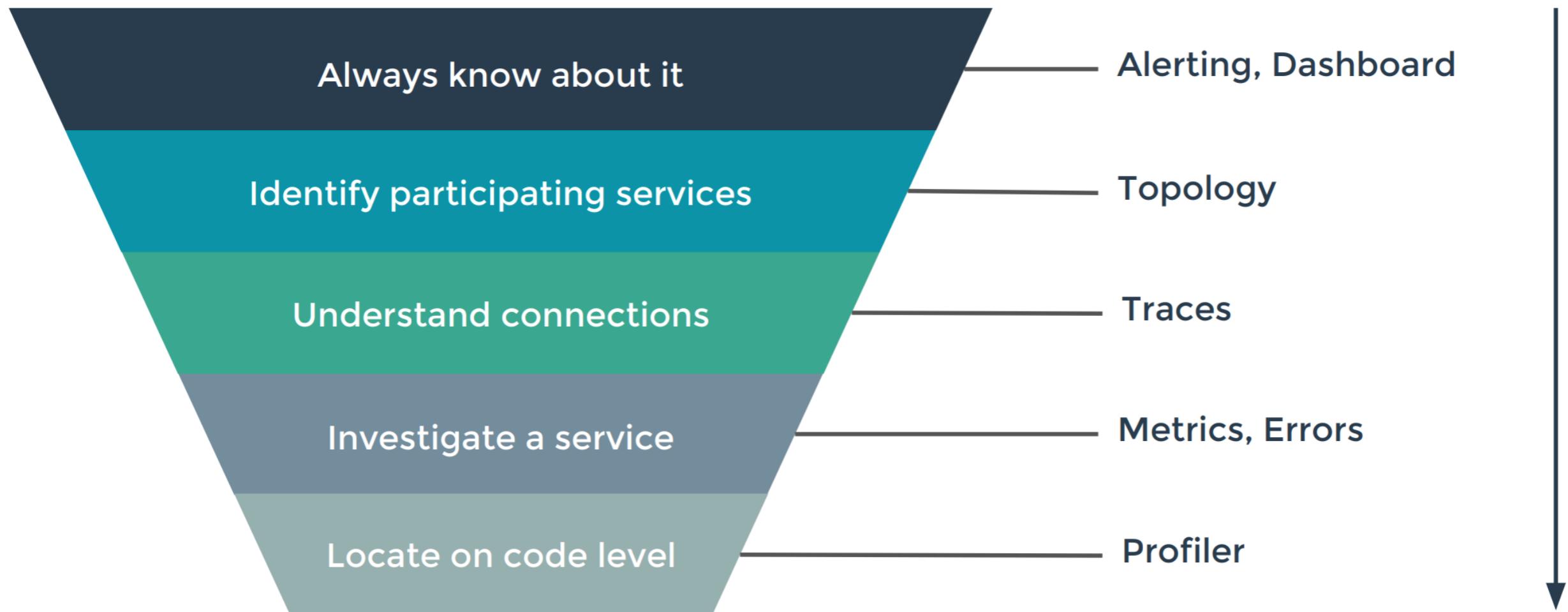


11. Log management

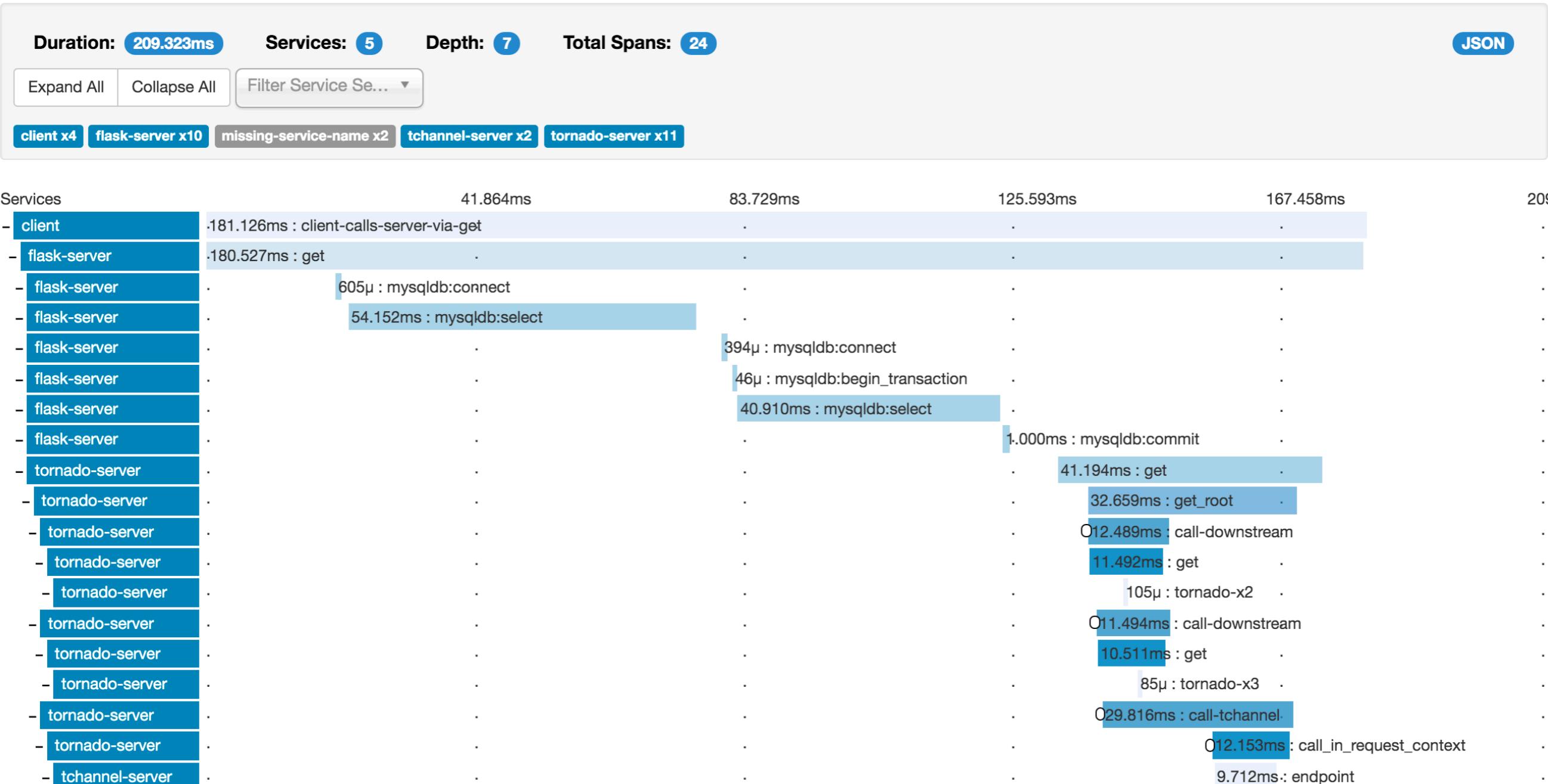
Treated as an event stream



How to find an issue ?



Logging/Tracing



<https://zipkin.io/>



Microservices

© 2017 - 2018 Siam Chamnankit Company Limited. All rights reserved.

12. Admin tasks

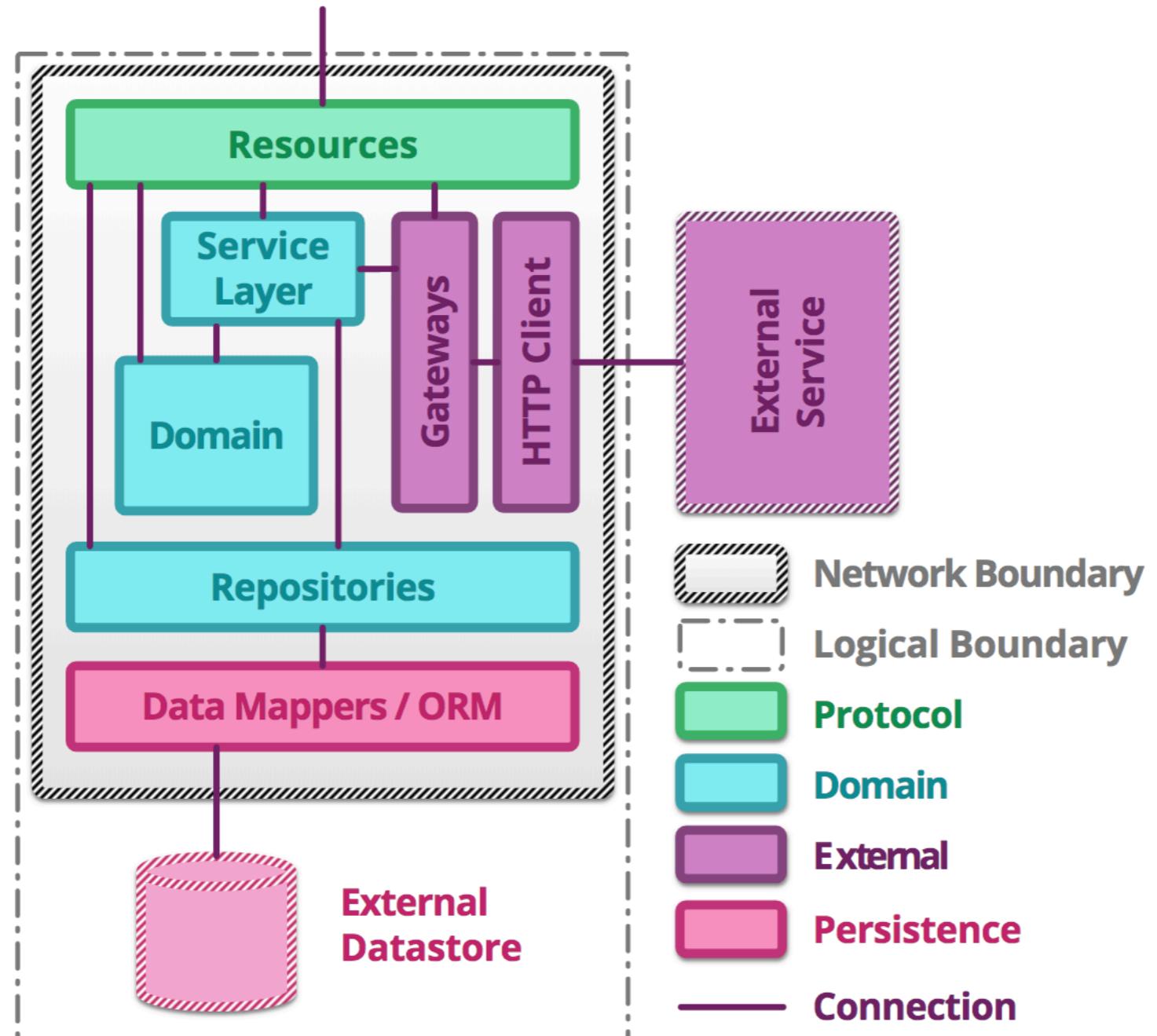
Treated the same way like the rest of the application



Microservice Testing



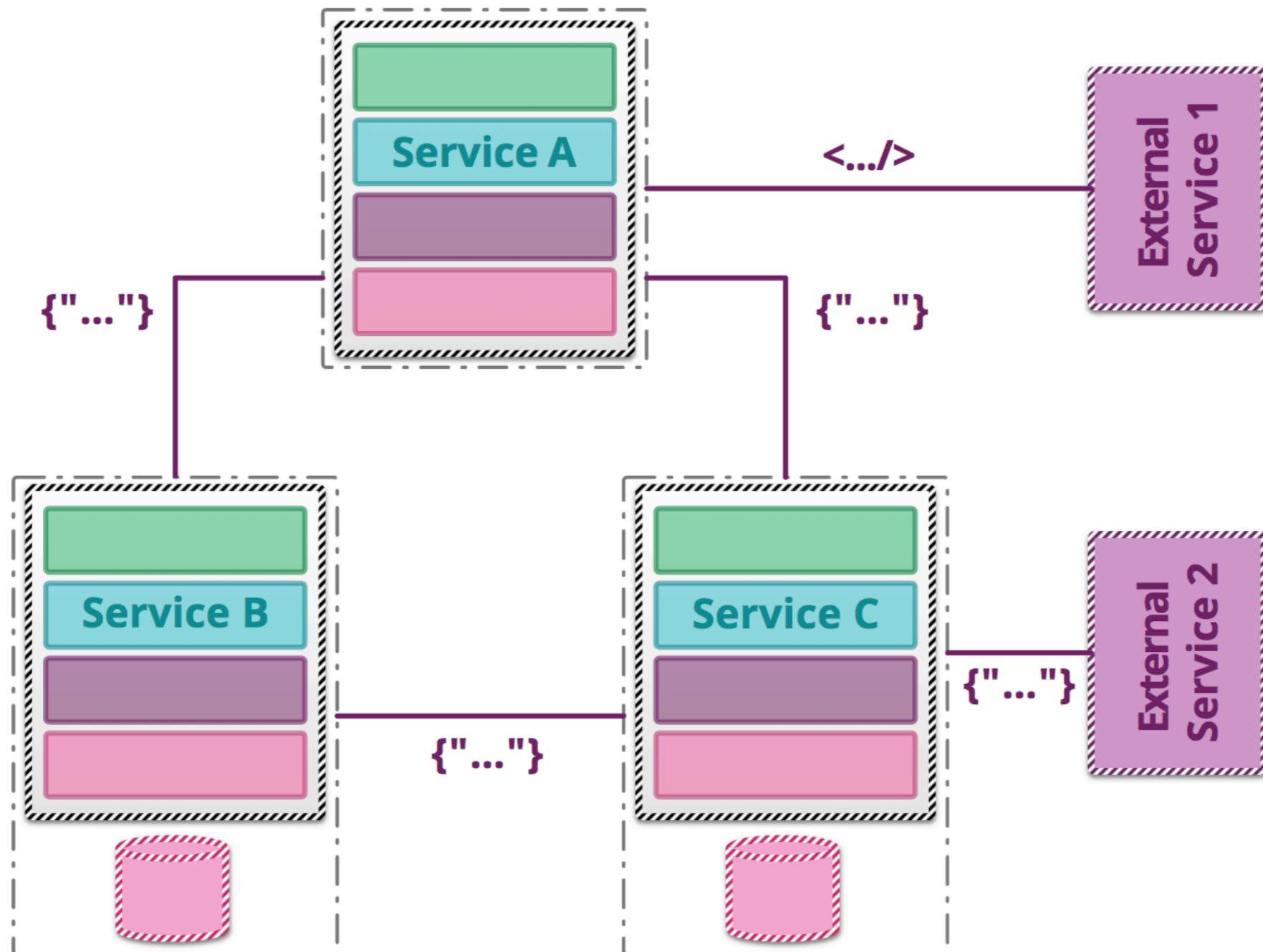
Service structure



<https://martinfowler.com/articles/microservice-testing>



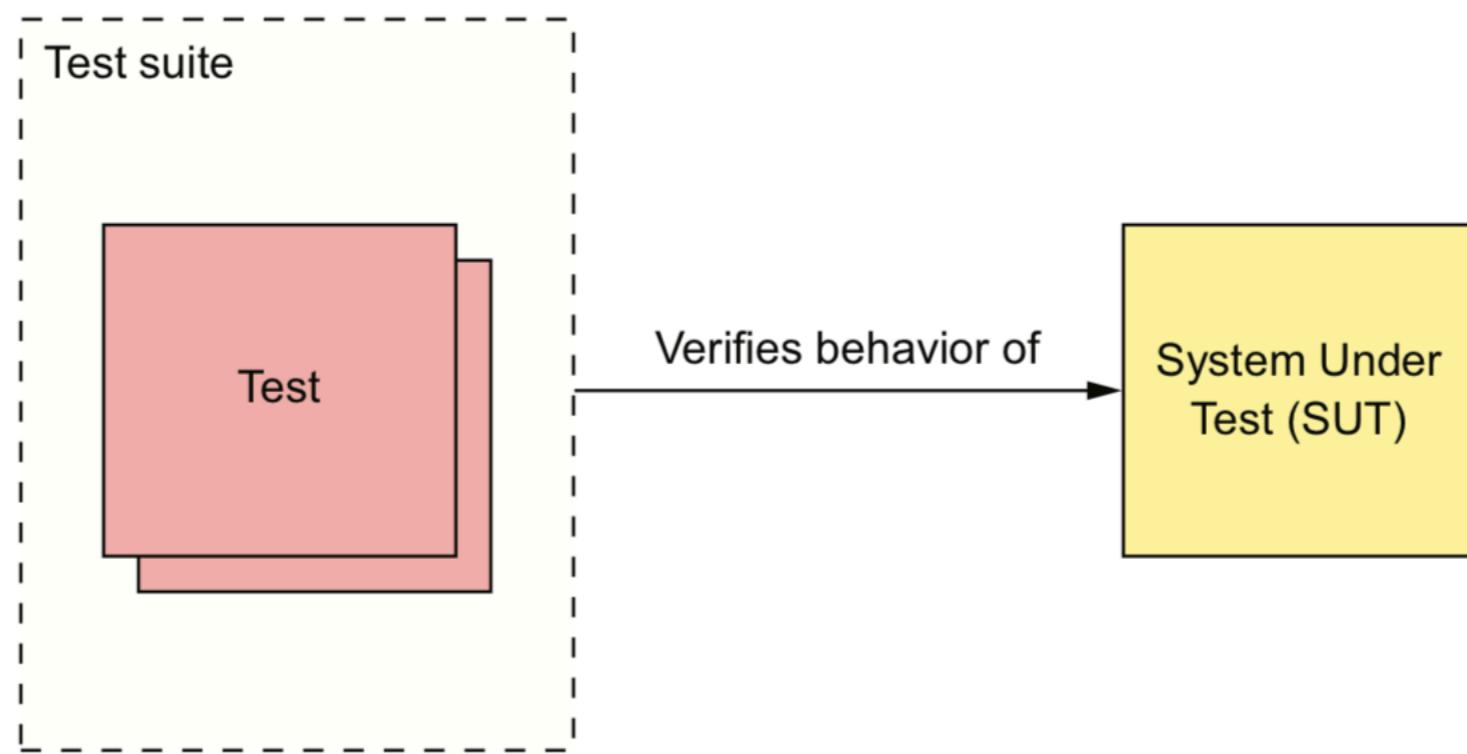
Multiple services



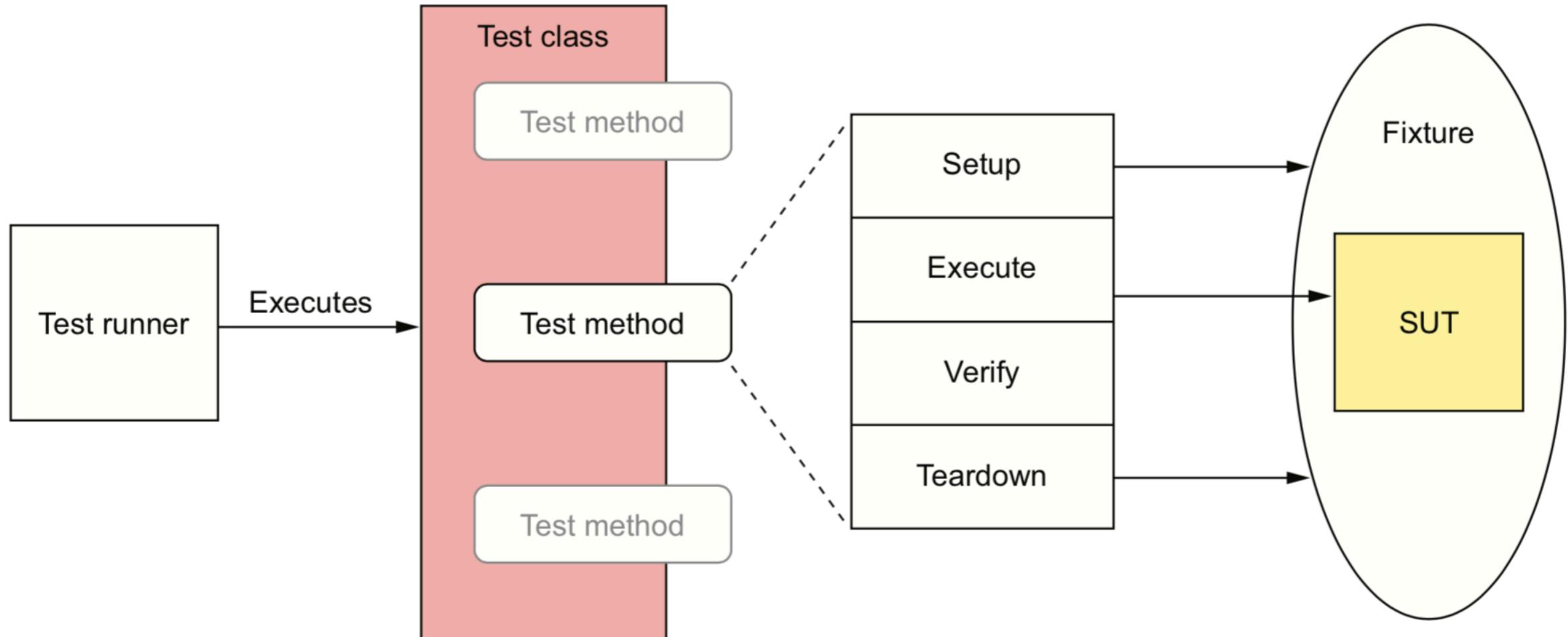
<https://martinfowler.com/articles/microservice-testing>



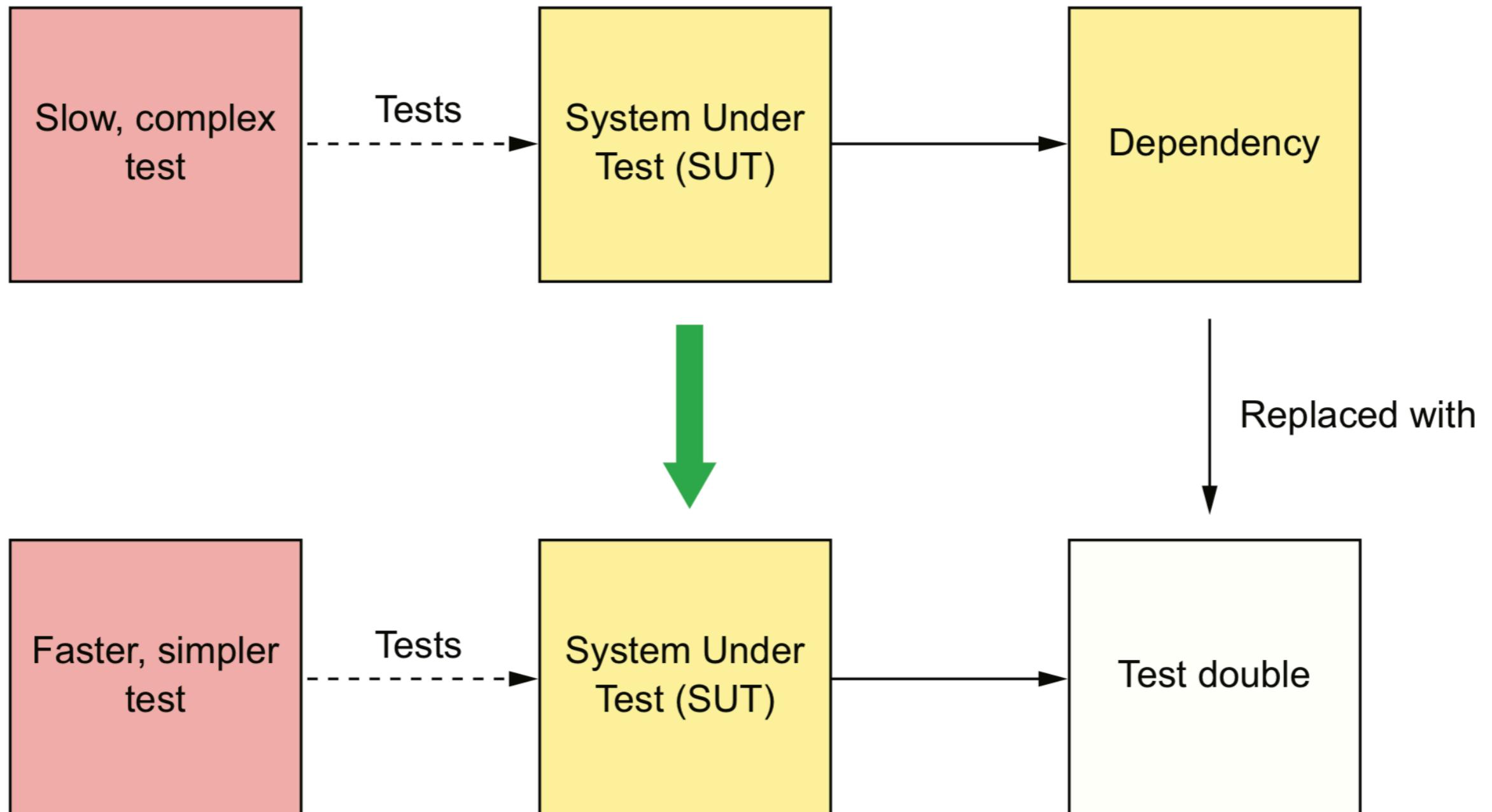
Testing

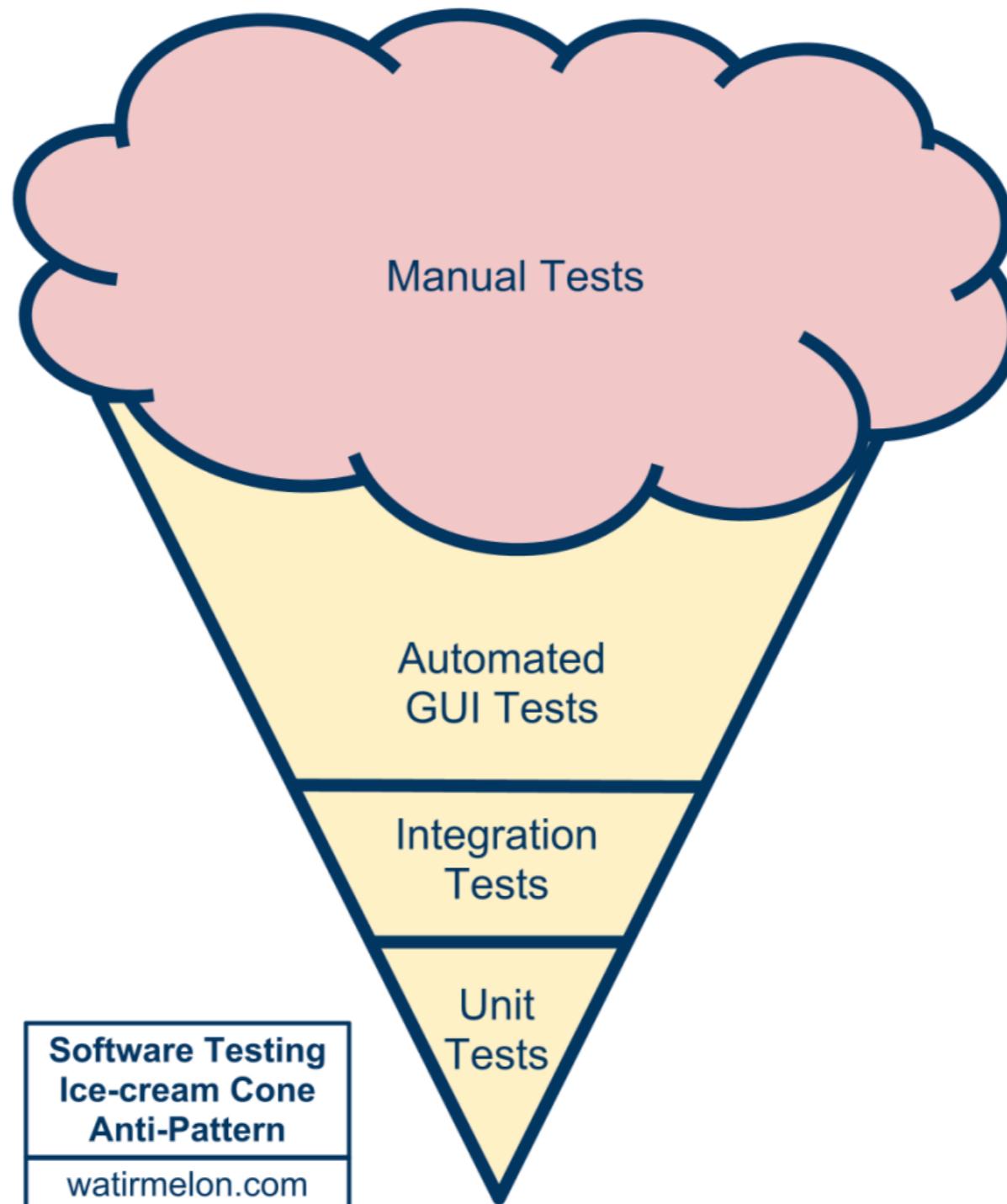


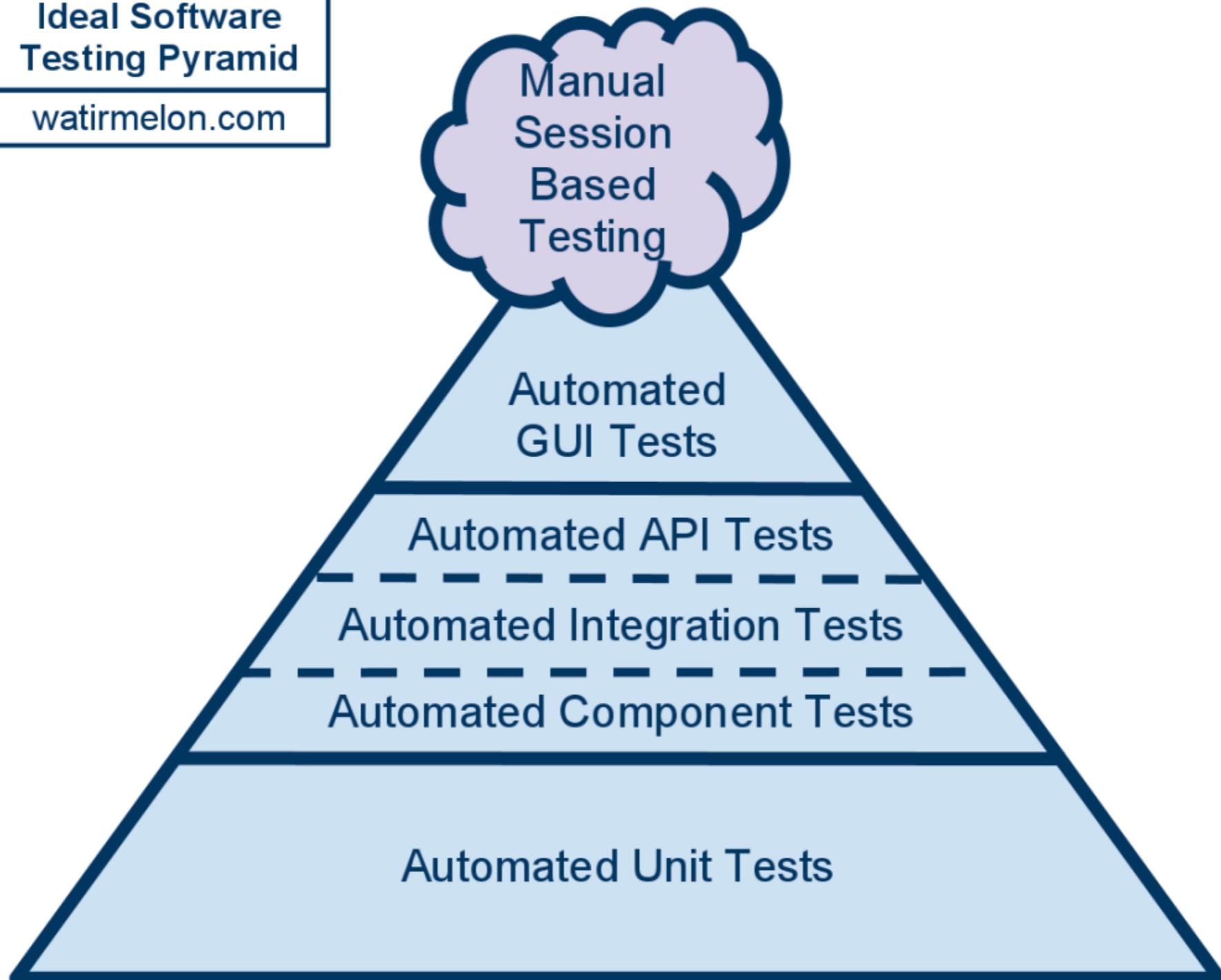
Structure of Automated tests

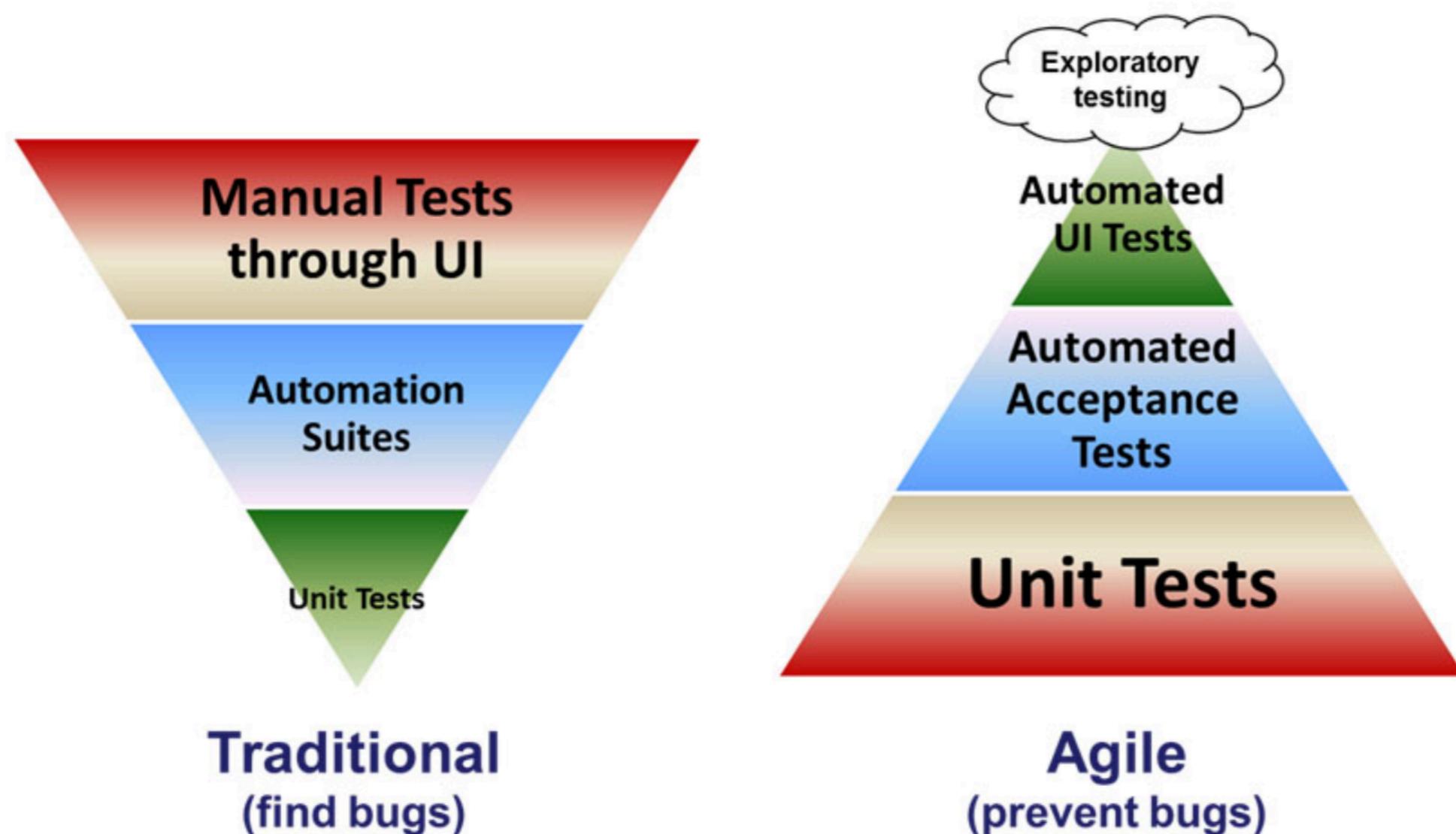


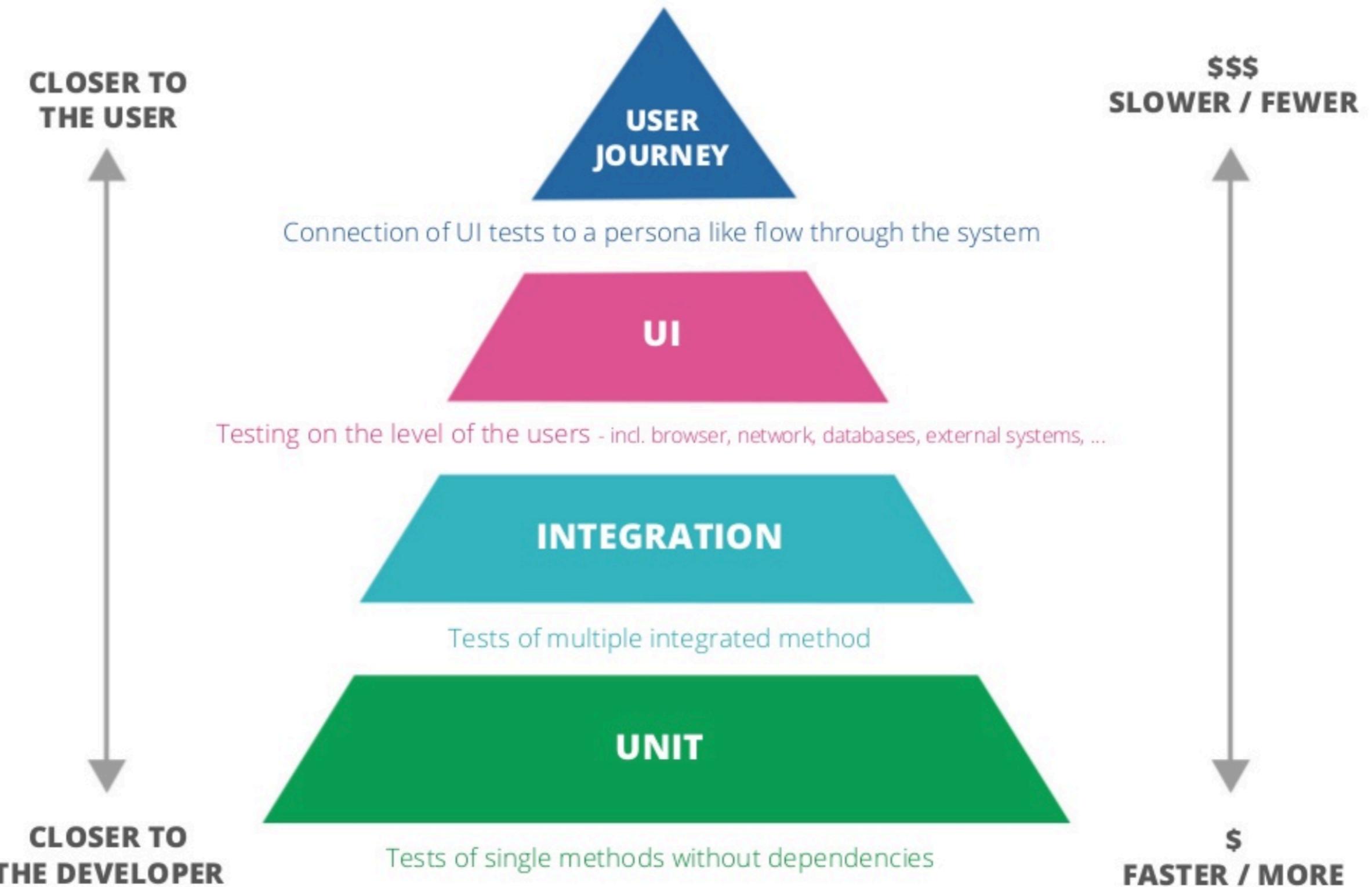
Working with Test double



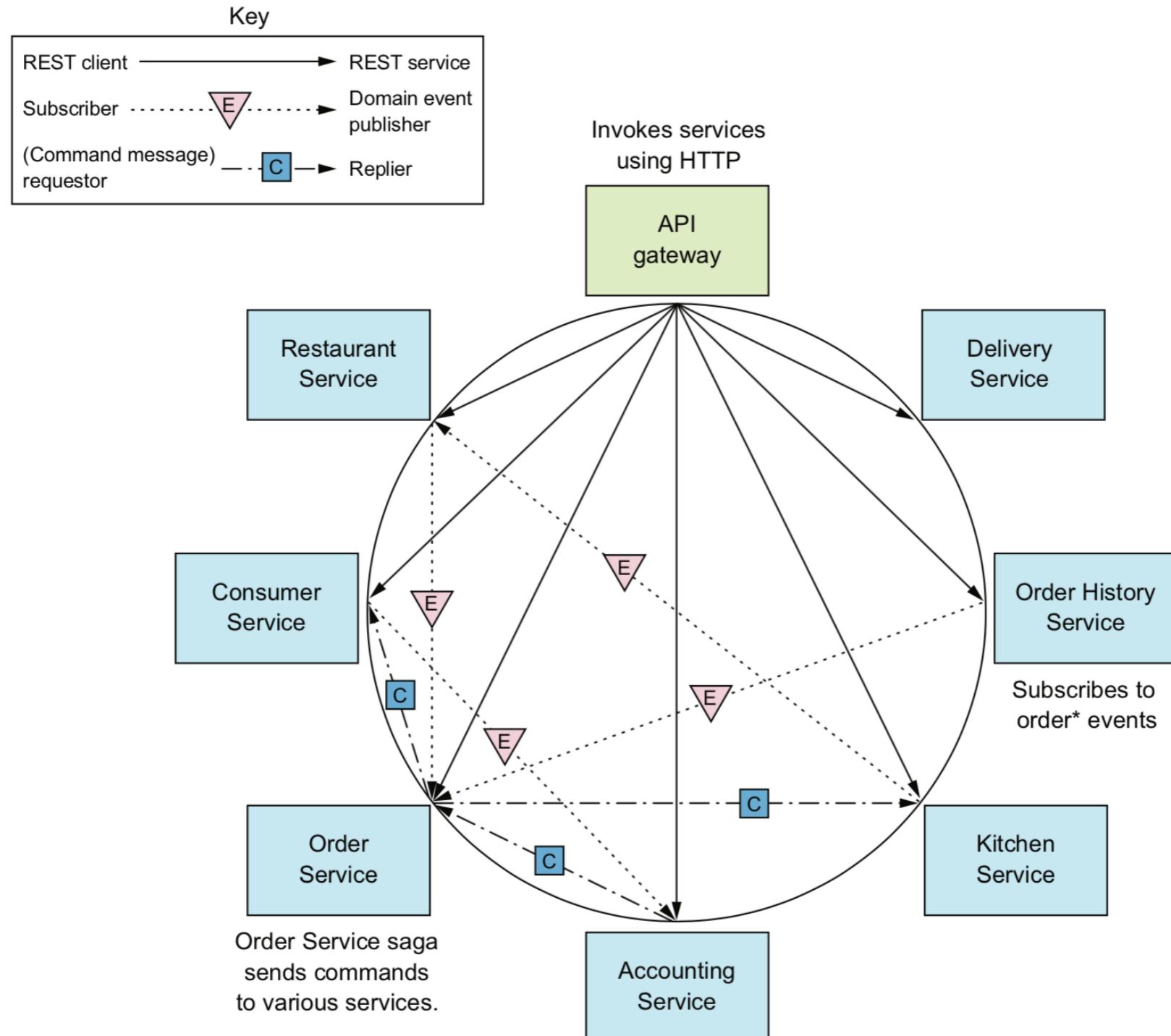


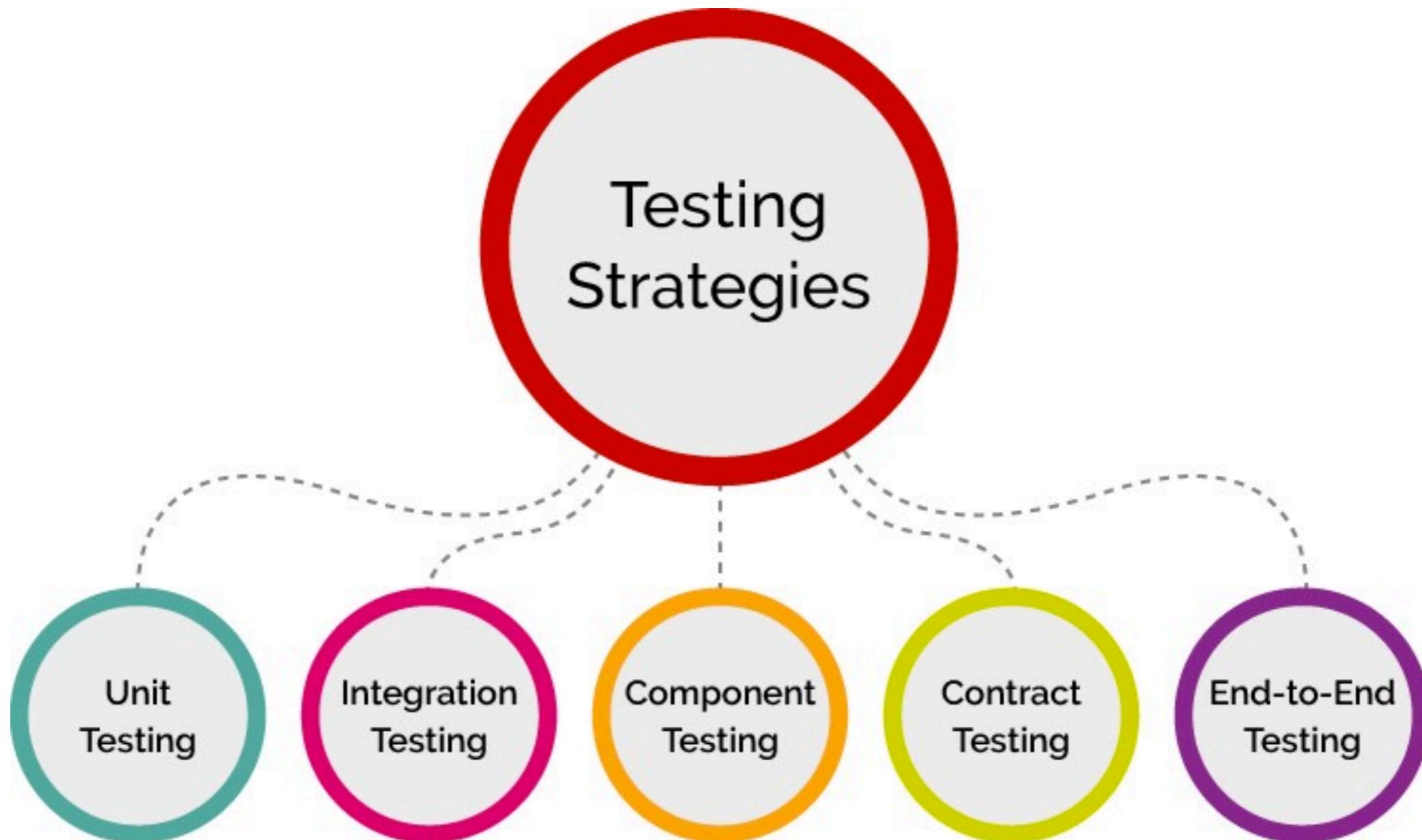




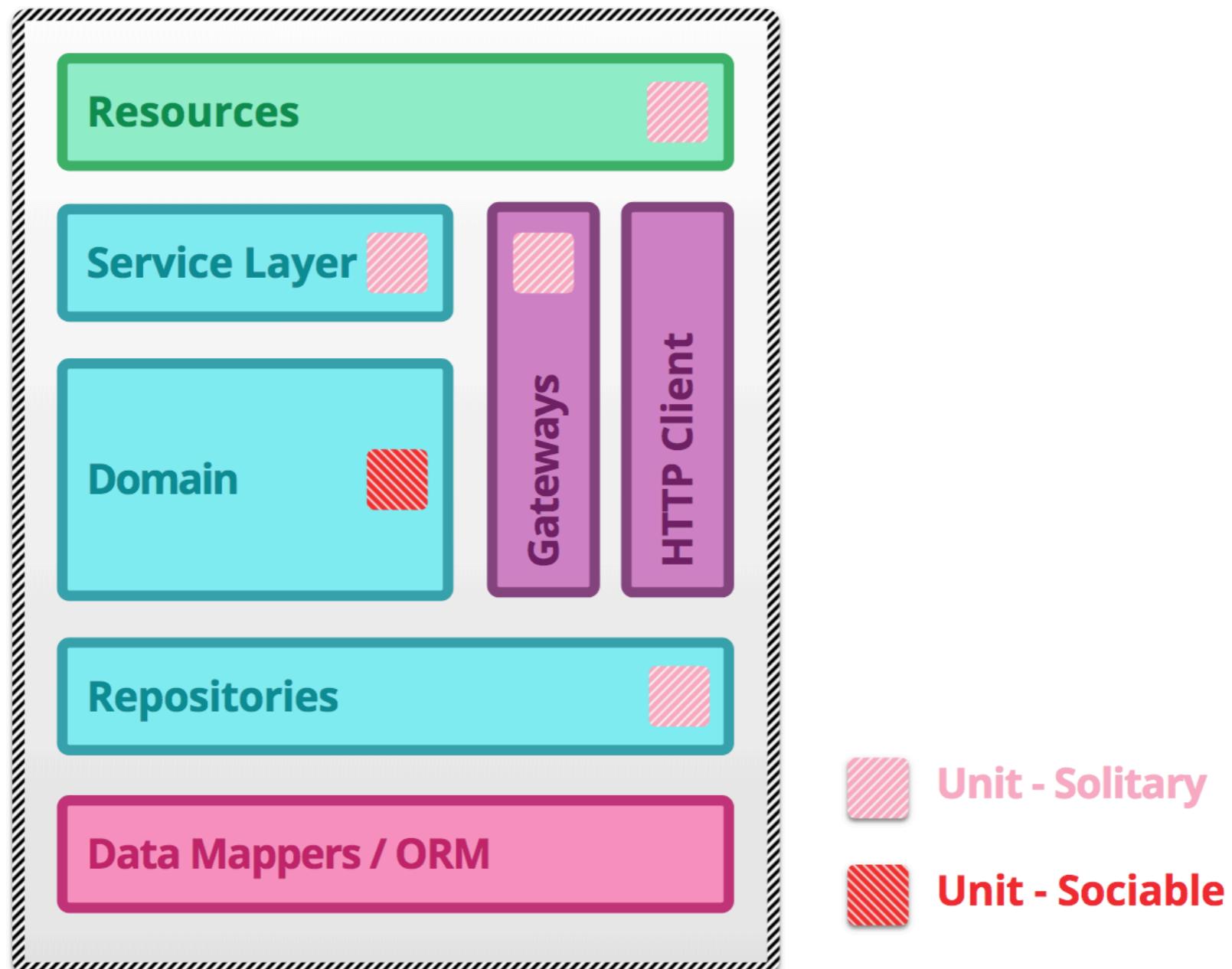


Testing Microservices ?

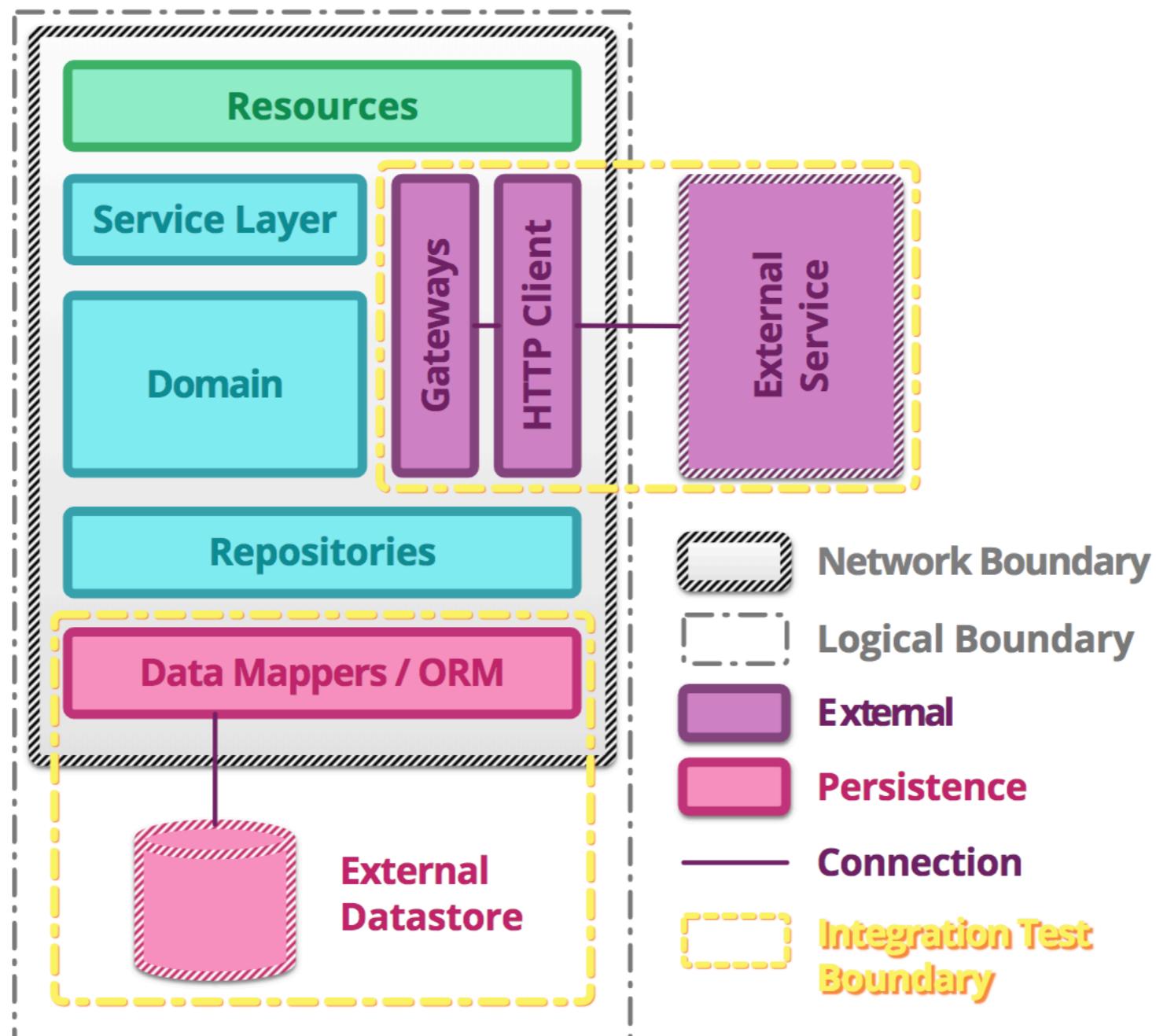




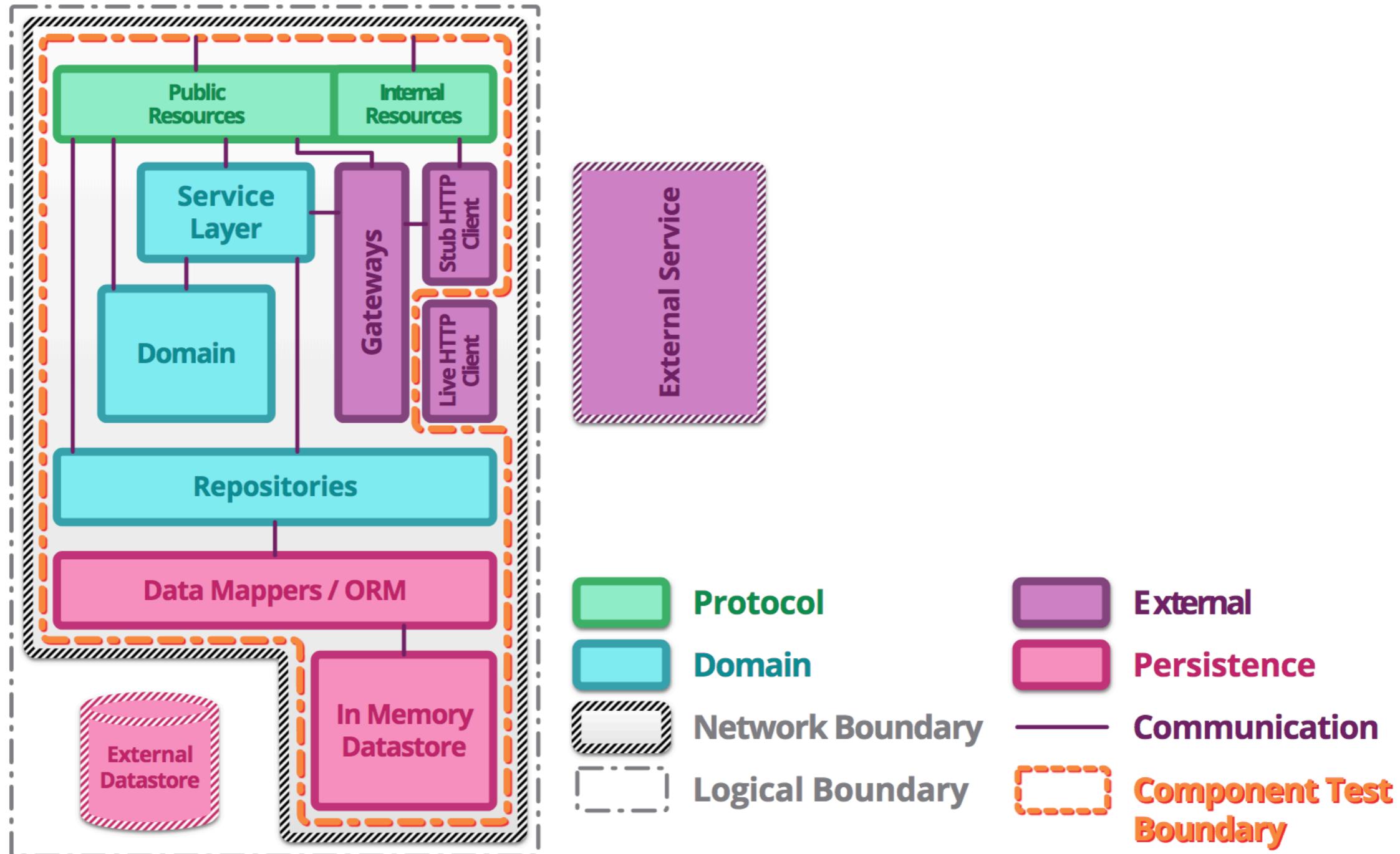
Unit testing



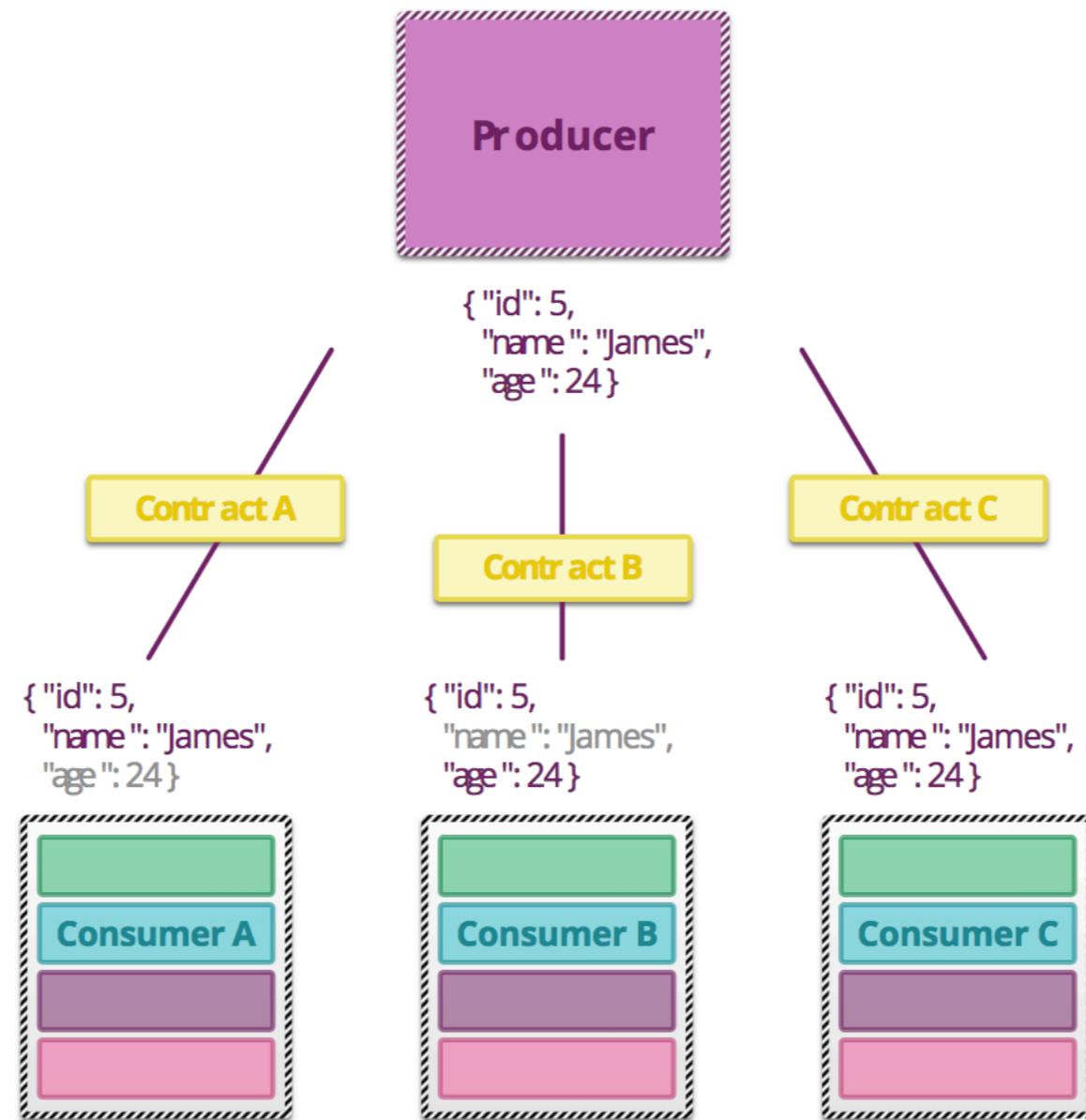
Integration testing



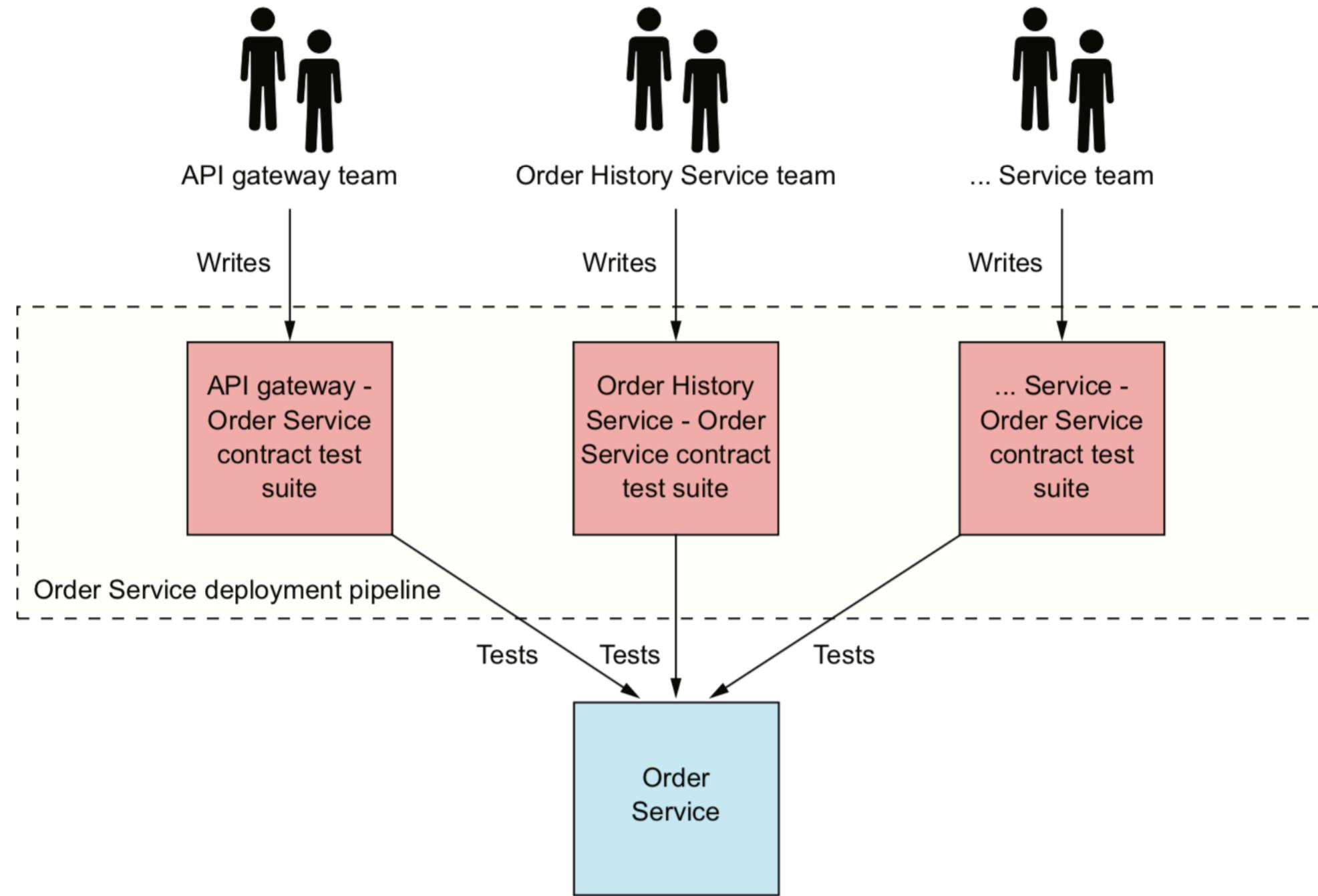
Component testing



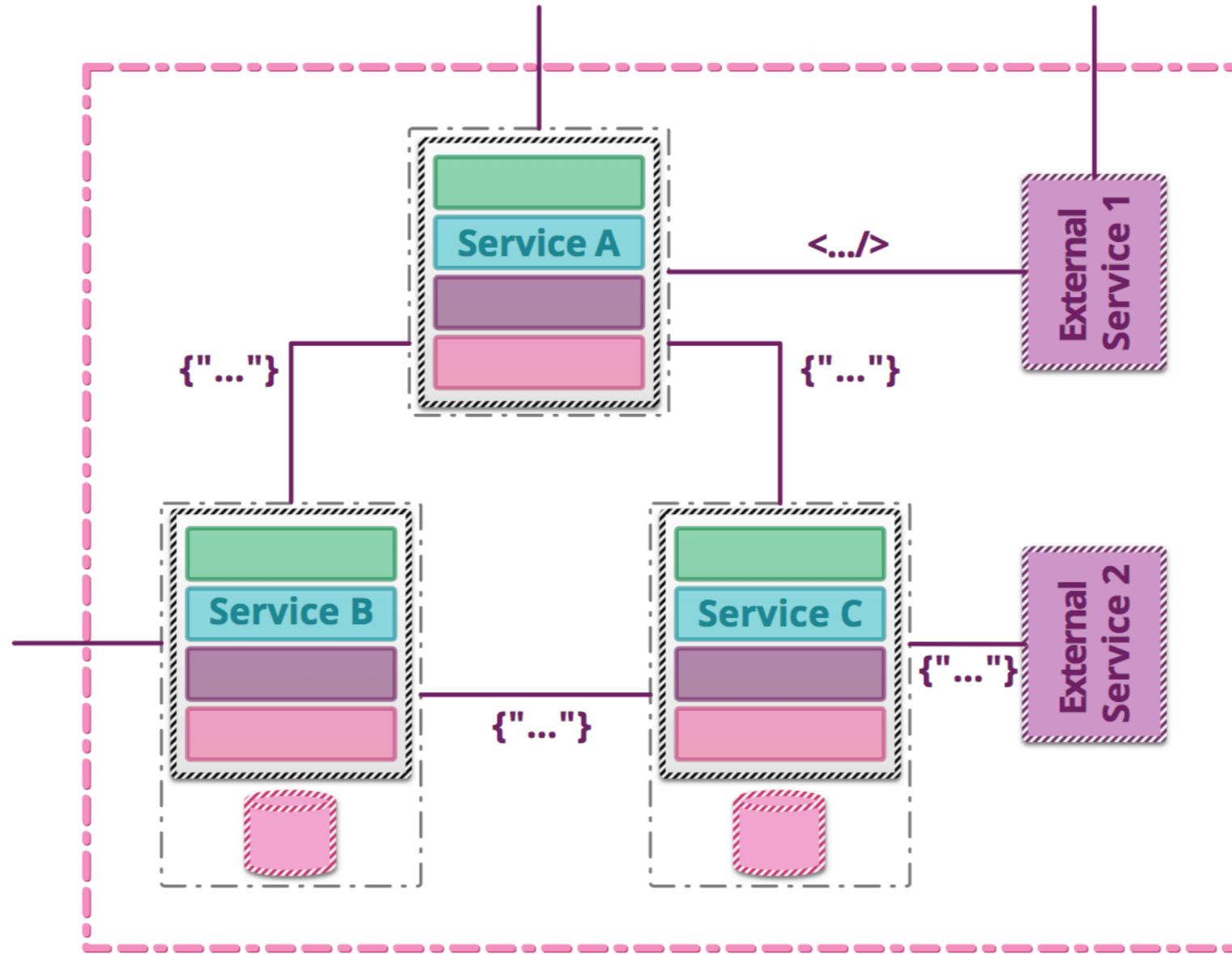
Contract testing



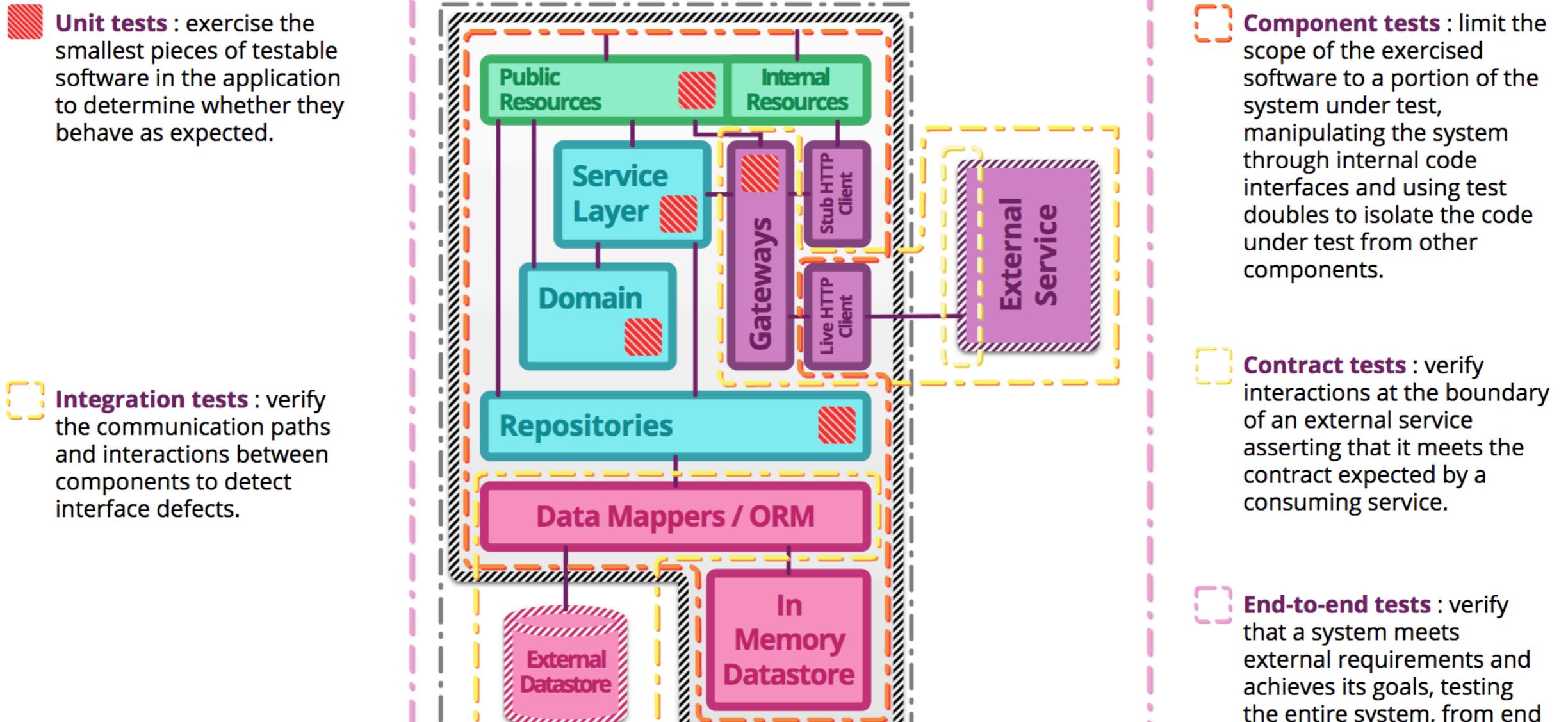
Consumer Contract testing



End-to-End testing



Summary



Workshop



Performance testing ?

Security testing ?

