JAVA USER GROUP CH

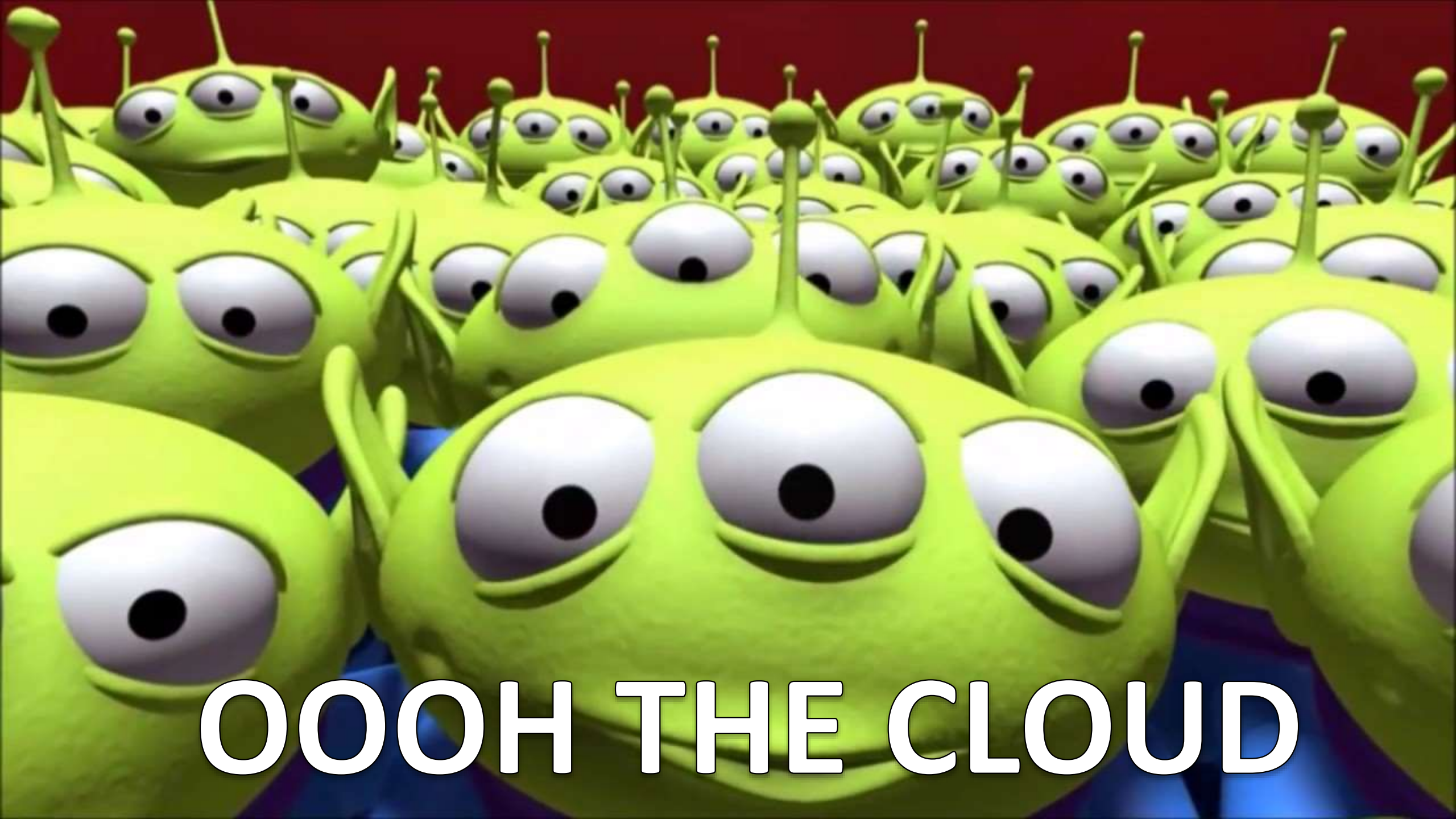
Thriving in the Cloud:

Going Beyond the 12 Factors

GRACE, JANSEN

DEVELOPER ADVOCATE, IBM

@gracejansen27


OOOH THE CLOUD

What does the Cloud offer?

Co$t

Speed

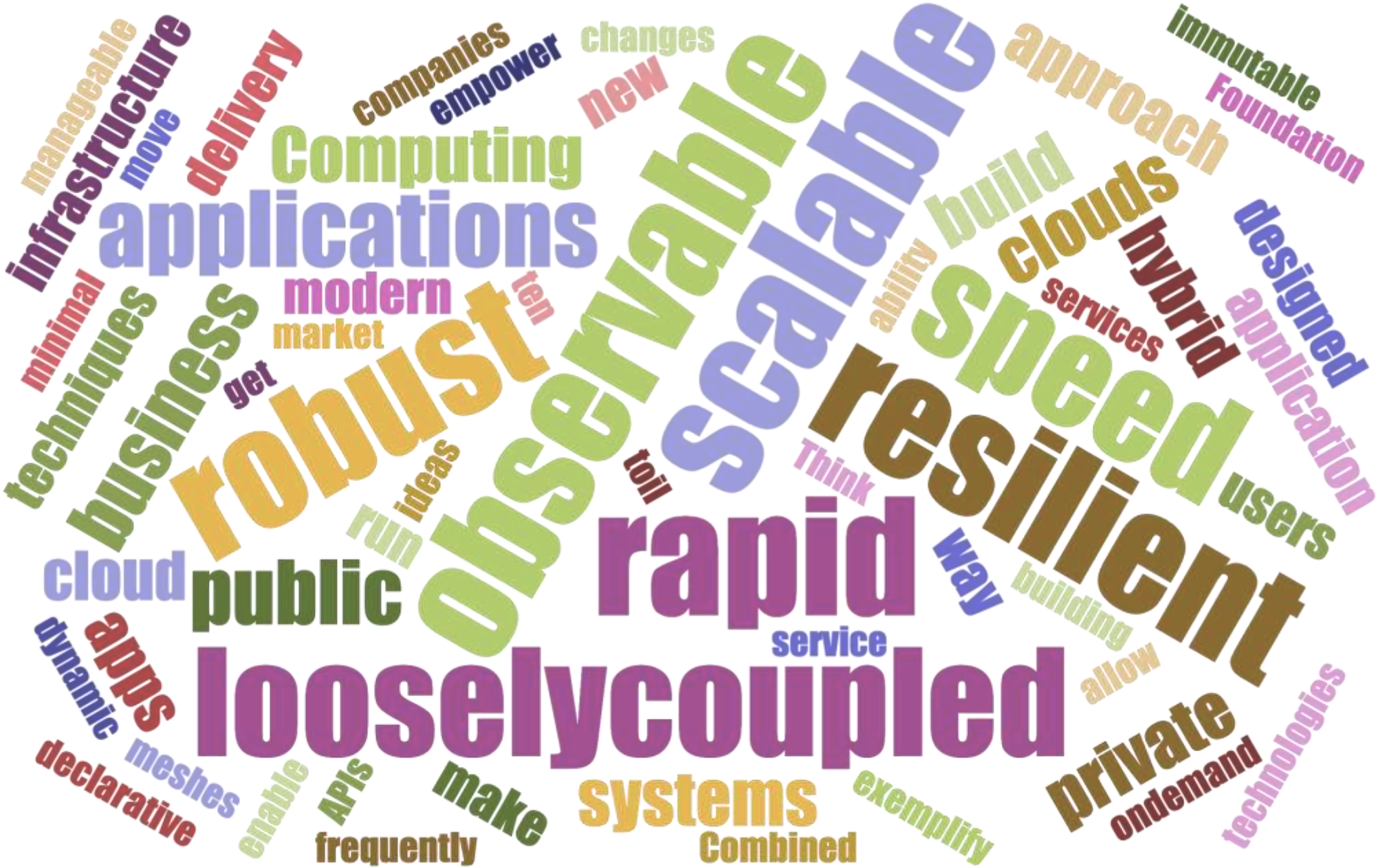
Scalability

Resiliency

Fashion

Flexibility

Innovation



@gracejansen27

12 Factor Apps

12 Factor App Methodology

The Twelve-Factor App

Introduction

In the modern era, software is commonly delivered as a service: called *web apps*, or *software-as-a-service*. The twelve-factor app is a methodology for building software-as-a-service apps that:

- Use **declarative** formats for setup automation, to minimize time and cost for new developers joining the project;
- Have a **clean contract** with the underlying operating system, offering **maximum portability** between execution environments;
- Are suitable for **deployment** on modern **cloud platforms**, obviating the need for servers and systems administration;
- **Minimize divergence** between development and production, enabling **continuous deployment** for maximum agility;
- And can **scale up** without significant changes to tooling, architecture, or development practices.

The twelve-factor methodology can be applied to apps written in any programming language, and which use any combination of backing services (database, queue, memory cache, etc).

Background

The contributors to this document have been directly involved in the development and deployment of hundreds of apps, and indirectly witnessed the development, operation, and scaling of hundreds of thousands of apps via our work on the Heroku platform.

This document synthesizes all of our experience and observations on a wide variety of software-as-a-service apps in the wild. It is a triangulation on ideal practices for app development, paying particular attention to the dynamics of the organic growth of an app over time, the dynamics of collaboration between developers working on the app's codebase, and avoiding the cost of software erosion.

Our motivation is to raise awareness of some systemic problems we've seen in modern application development, to provide a shared vocabulary for discussing those problems, and to offer a set of broad conceptual solutions to those problems with accompanying terminology. The format is inspired by Martin Fowler's books *Patterns of Enterprise Application Architecture* and *Refactoring*.

https://12factor.net/

The original 12 factors:

1. Codebase
2. Dependencies
3. Configuration
4. Backing Services
5. Build, release, run
6. Processes
7. Port Binding
8. Concurrency
9. Disposability
10. Dev/prod parity
11. Logs
12. Admin Processes

Revised 15 factors

1. One Codebase, one application
2. API first
3. Dependency management
4. Design, build, release, and run
5. Configuration, credentials and code
6. Logs
7. Disposability
8. Backing services
9. Environment parity
10. Administrative processes
11. Port binding
12. Stateless processes
13. Concurrency
14. Telemetry
15. Authentication and authorization

https://www.oreilly.com/library/view/beyond-the-twelve-factor/9781492042631/

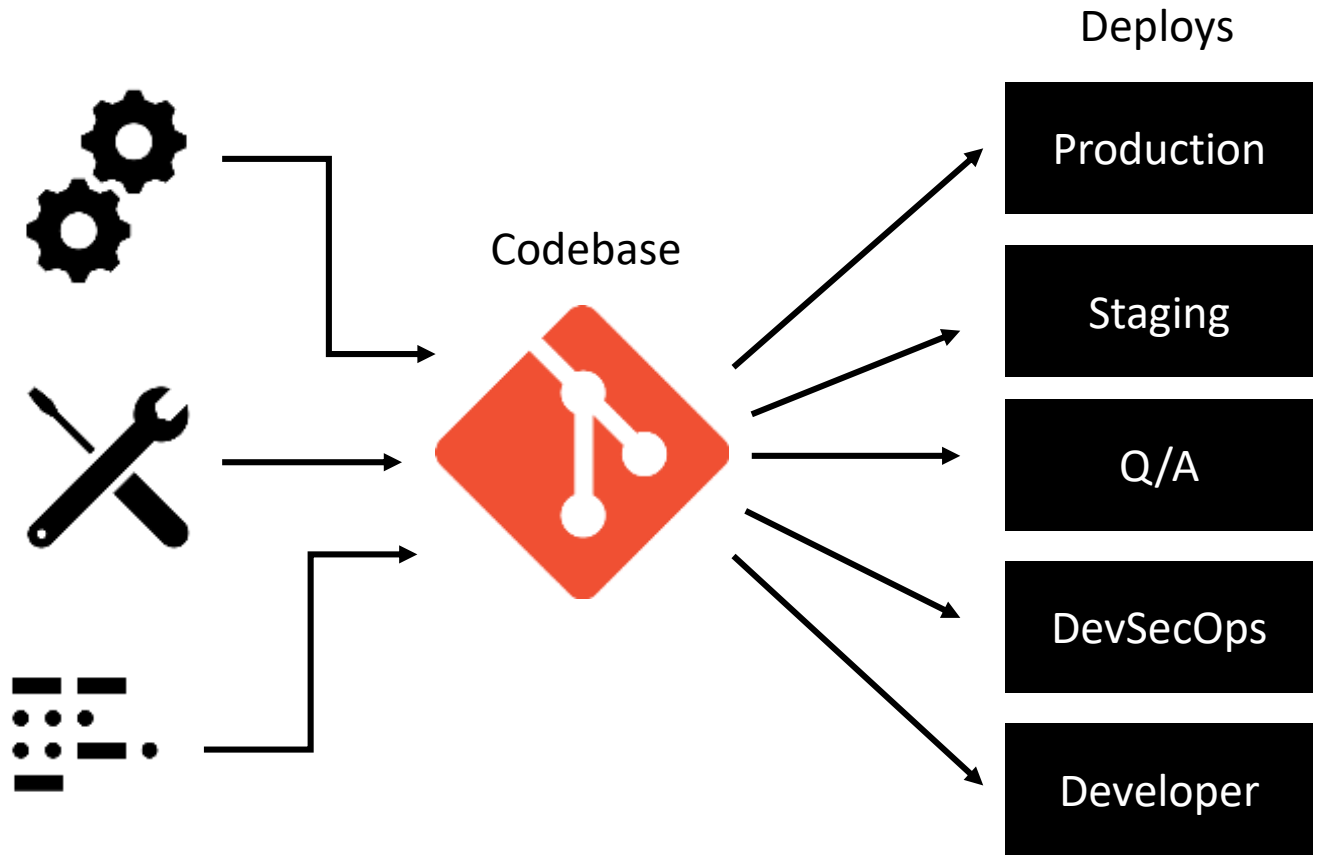
Revised 15 factors

1. One Codebase, one application
2. API first
3. Dependency management
4. Design, build, release, and run
5. Configuration, credentials and code
6. Logs
7. Disposability
8. Backing services
9. Environment parity
10. Administrative processes
11. Port binding
12. Stateless processes
13. Concurrency
14. Telemetry
15. Authentication and authorization

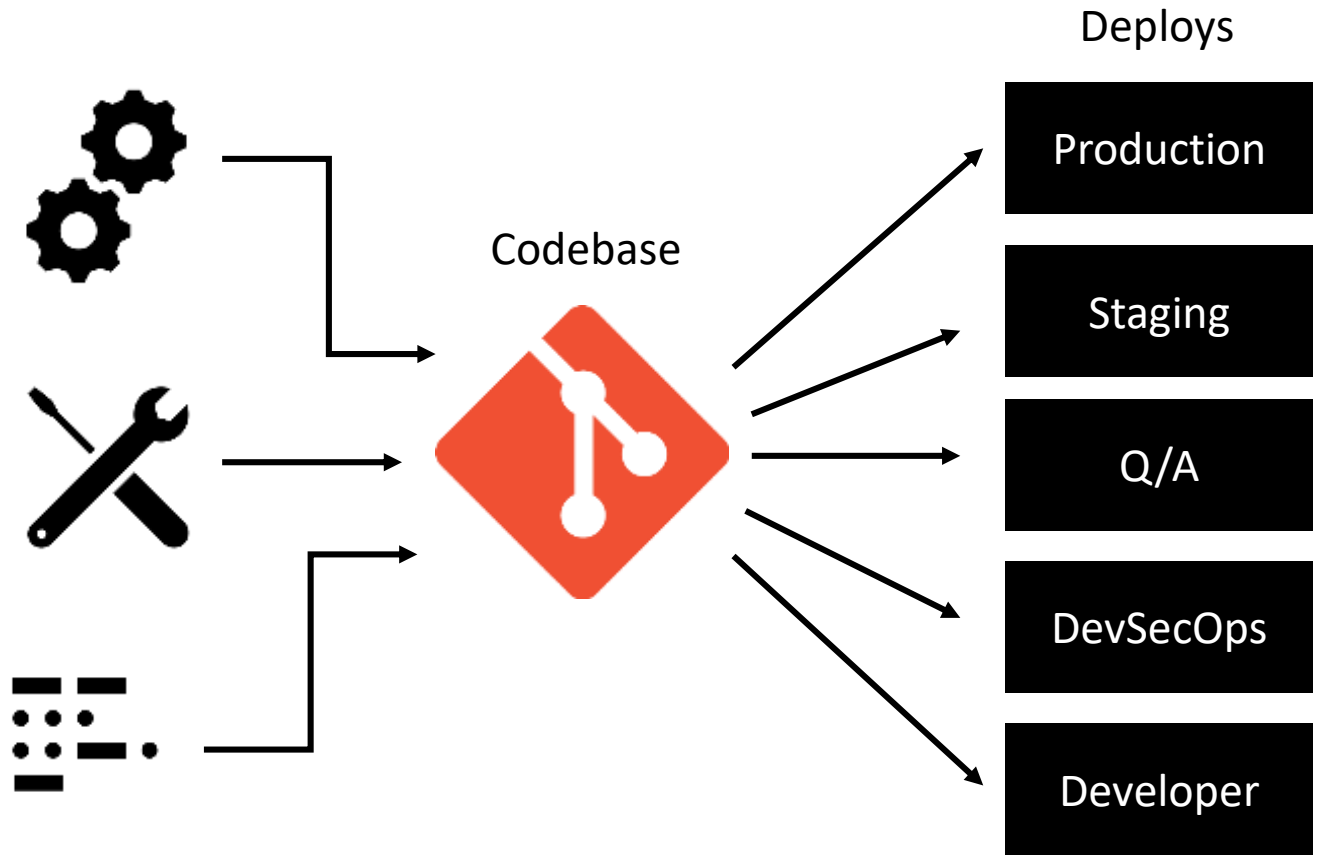
https://www.oreilly.com/library/view/beyond-the-twelve-factor/9781492042631/

The Original 12 Factors

1. One codebase, one application



1. One codebase, one application



GitHub

GitHub Enterprise

GitLab

3. Dependency management

Application Source Code

Makes use of…

```
<dependencies>
    <!-- Provided dependencies -->
    <dependency>
        <groupId>jakarta.platform</groupId>
        <artifactId>jakarta.jakartaee-api</artifactId>
        <version>8.0.0</version>
        <scope>provided</scope>
    </dependency>
    <dependency>
        <groupId>org.eclipse.microprofile</groupId>
        <artifactId>microprofile</artifactId>
        <version>4.0.1</version>
        <type>pom</type>
        <scope>provided</scope>
    </dependency>
    <!-- For tests -->
    <dependency>
        <groupId>org.junit.jupiter</groupId>
        <artifactId>junit-jupiter</artifactId>
        <version>5.7.1</version>
        <scope>test</scope>
    </dependency>
```

3. Dependency management

Application Source Code

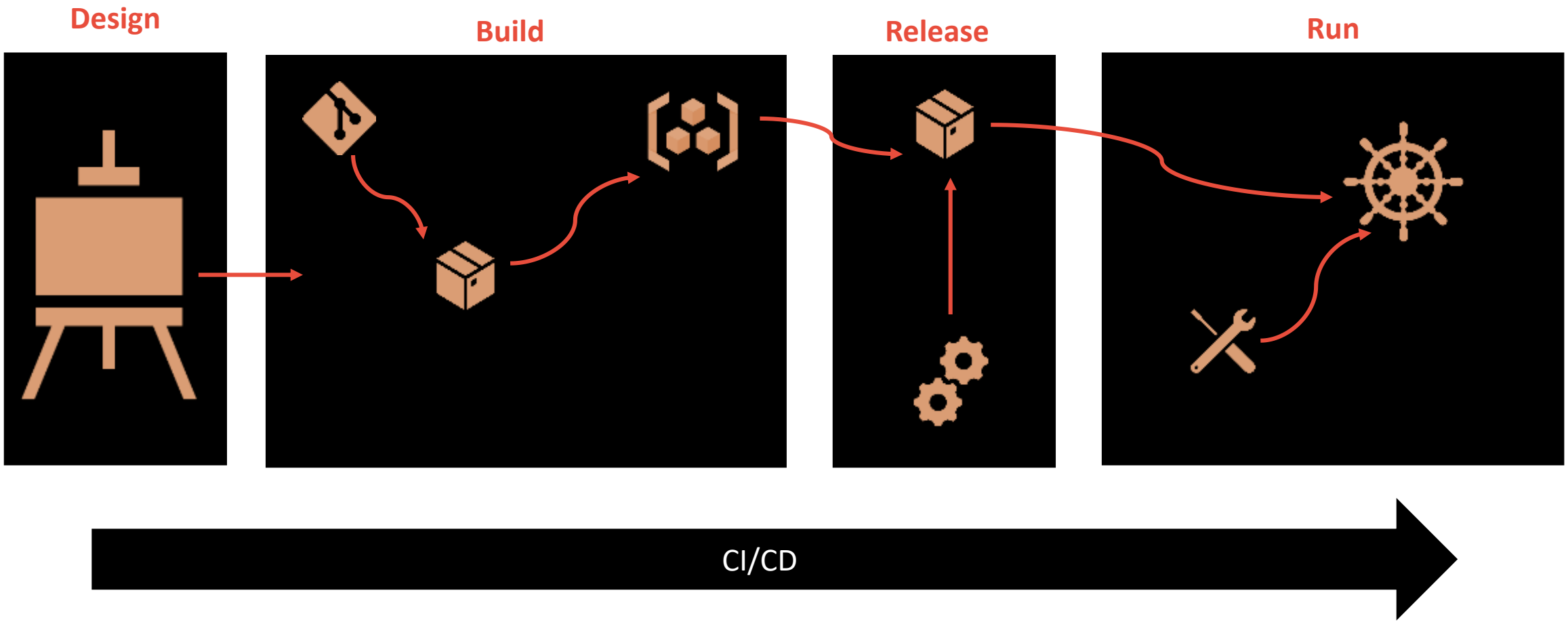
Makes use of…

```
<dependencies>
    <!-- Provided dependencies -->
    <dependency>
        <groupId>jakarta.platform</groupId>
        <artifactId>jakarta.jakartaee-api</artifactId>
        <version>8.0.0</version>
        <scope>provided</scope>
    </dependency>
    <dependency>
        <groupId>org.eclipse.microprofile</groupId>
        <artifactId>microprofile</artifactId>
        <version>4.0.1</version>
        <type>pom</type>
        <scope>provided</scope>
    </dependency>
    <!-- For tests -->
    <dependency>
        <groupId>org.junit.jupiter</groupId>
        <artifactId>junit-jupiter</artifactId>
        <version>5.7.1</version>
        <scope>test</scope>
    </dependency>
```

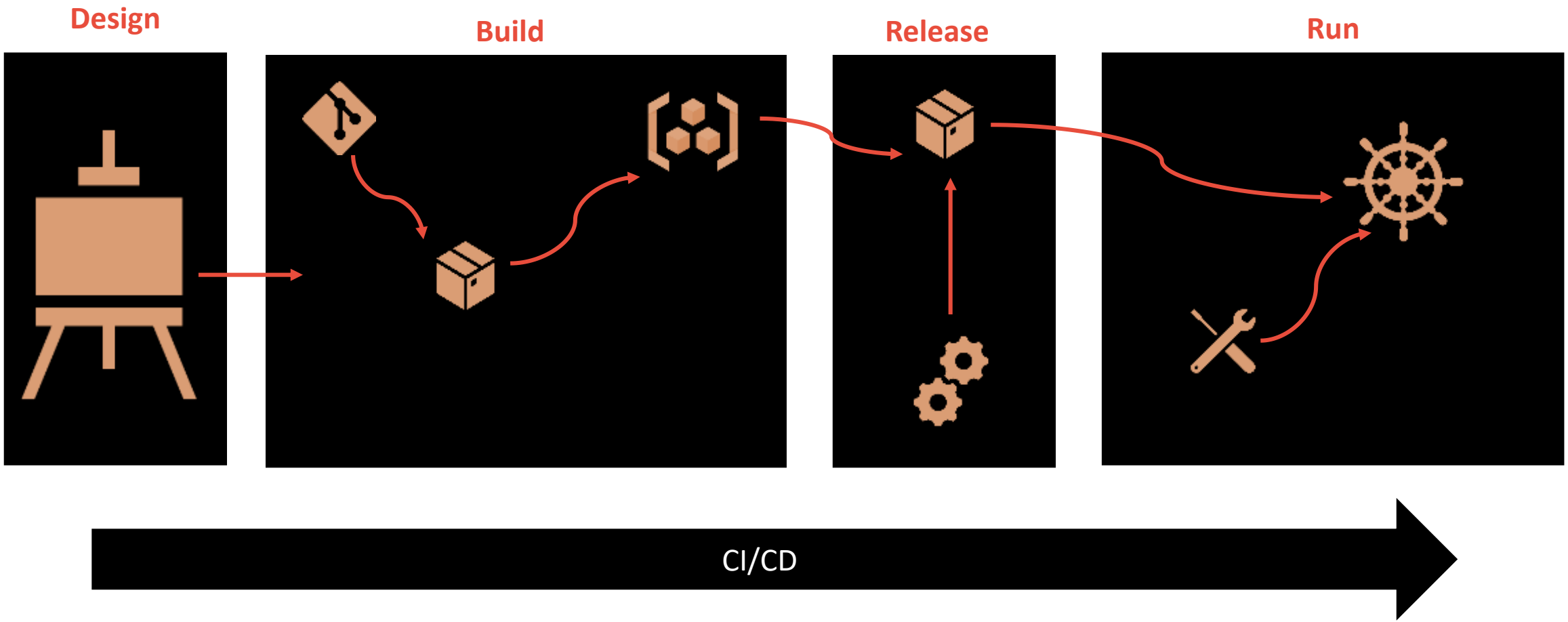
Maven™

Gradle

4. Design, build, release, run

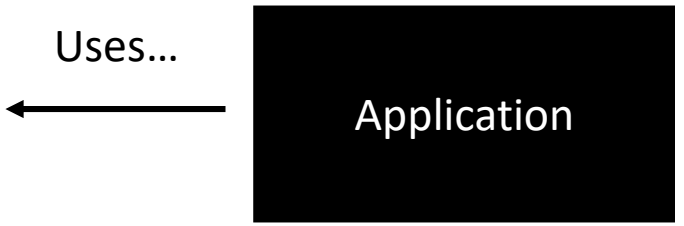


4. Design, build, release, run



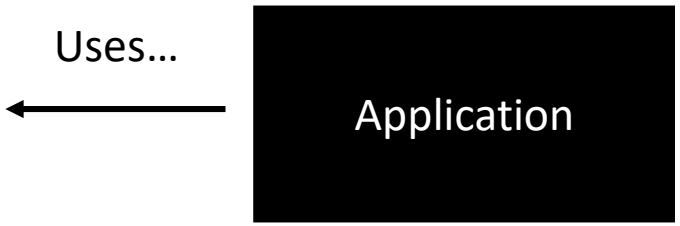
5. Configuration, credentials and code

```
15    <variable name="default.http.port" defaultValue="9080"/>
16    <variable name="default.https.port" defaultValue="9443"/>
17
18    <webApplication location="guide-getting-started.war" contextRoot="/dev" />
19
20    <mpMetrics authentication="false"/>
21
22    <!-- tag::logging[] -->
23    <logging traceSpecification="com.ibm.ws.microprofile.health.*=all" />
24    <!-- end::logging[] -->
25
26    <httpEndpoint host="*" httpPort="${default.http.port}"
27        httpsPort="${default.https.port}" id="defaultHttpEndpoint"/>
```



5. Configuration, credentials and code

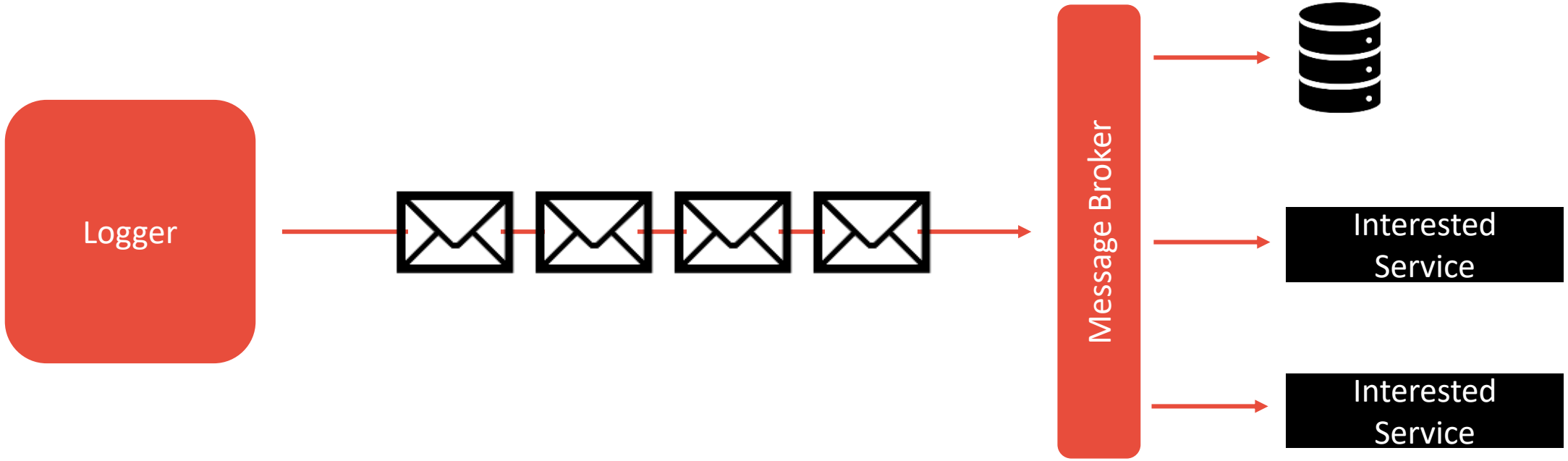
```
15    <variable name="default.http.port" defaultValue="9080"/>
16    <variable name="default.https.port" defaultValue="9443"/>
17
18    <webApplication location="guide-getting-started.war" contextRoot="/dev" />
19
20    <mpMetrics authentication="false"/>
21
22    <!-- tag::logging[] -->
23    <logging traceSpecification="com.ibm.ws.microprofile.health.*=all" />
24    <!-- end::logging[] -->
25
26    <httpEndpoint host="*" httpPort="${default.http.port}"
27        httpsPort="${default.https.port}" id="defaultHttpEndpoint"/>
```



MICROPROFILE™

https://openliberty.io/guides/microprofile-config.html

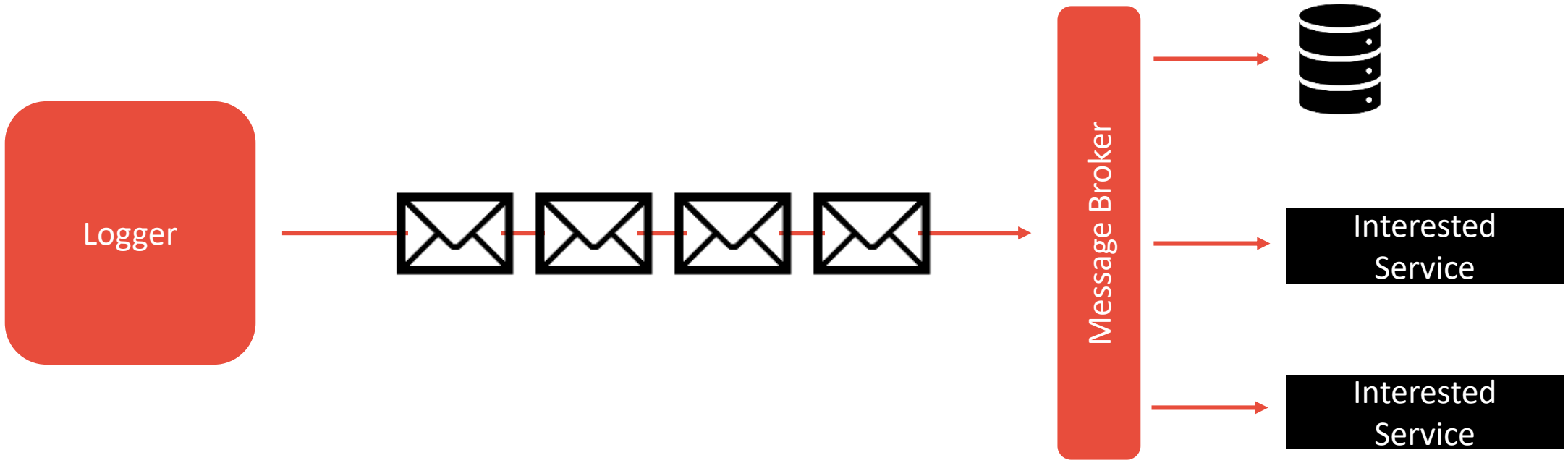
6. Logs


Logger
Message Broker
Interested Service
Interested Service

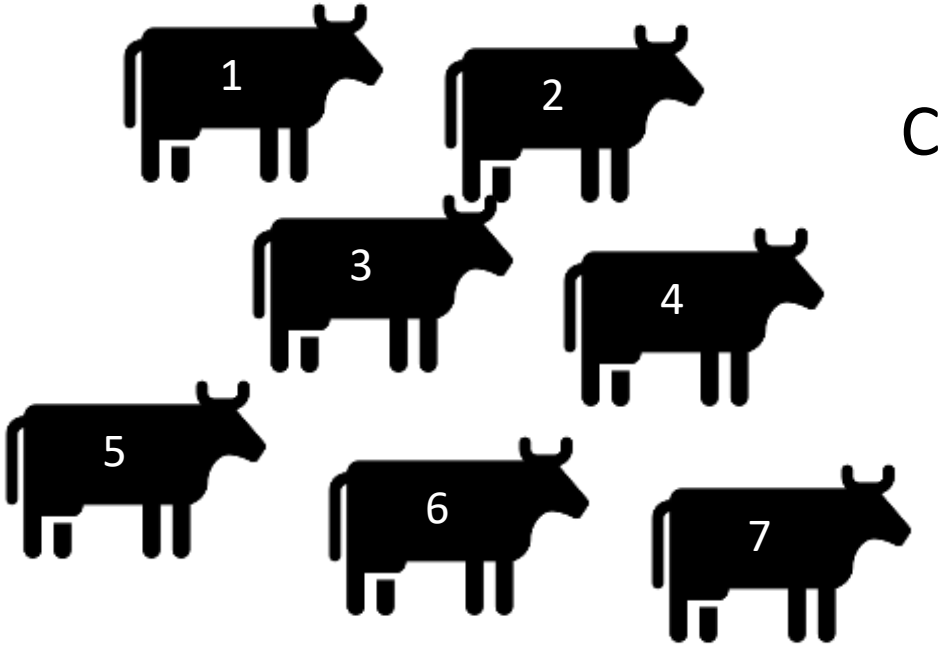
6. Logs

elasticsearch + logstash + kibana

fluentd



7. Disposability



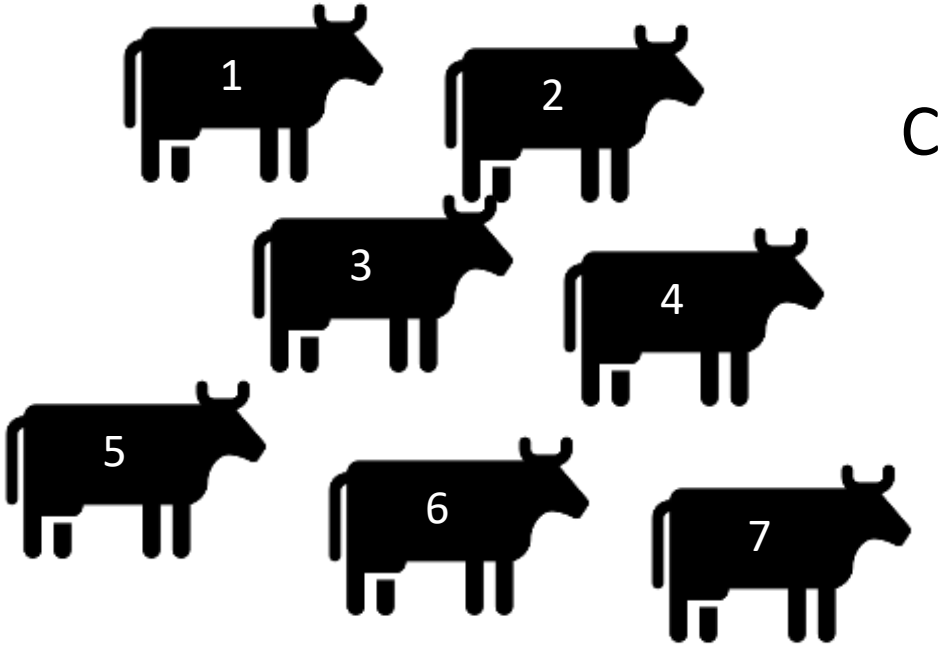
Cattle

VS

Pet

Carl the Cat

7. Disposability



Cattle

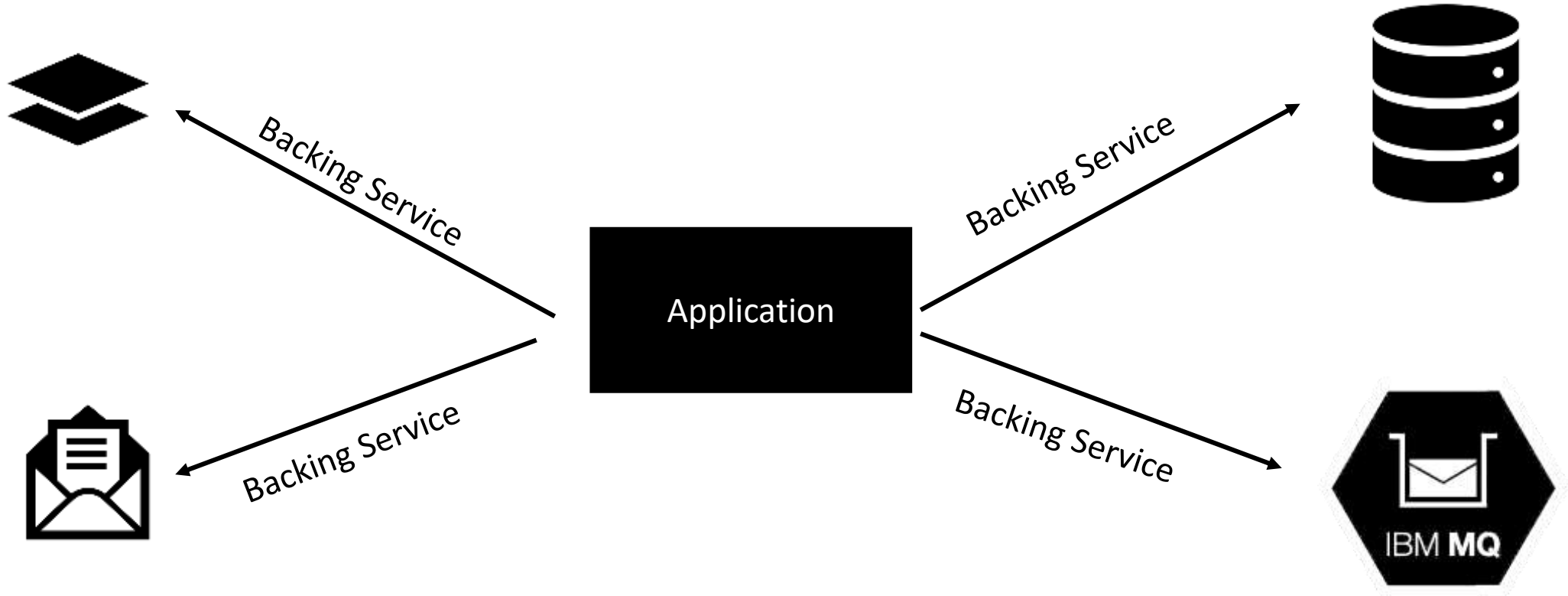
VS

Pet

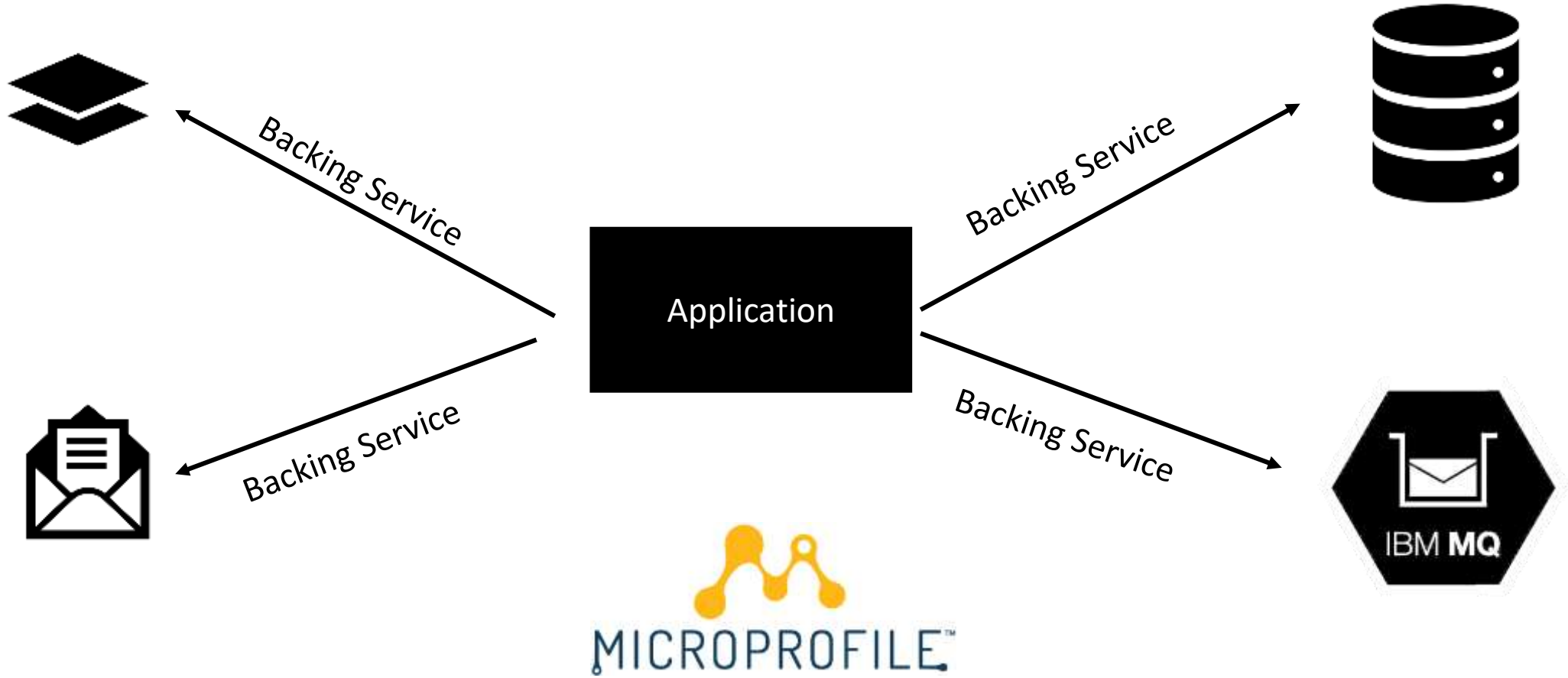
Carl the Cat

https://openliberty.io/guides/microprofile-fallback.html

8. Backing Services

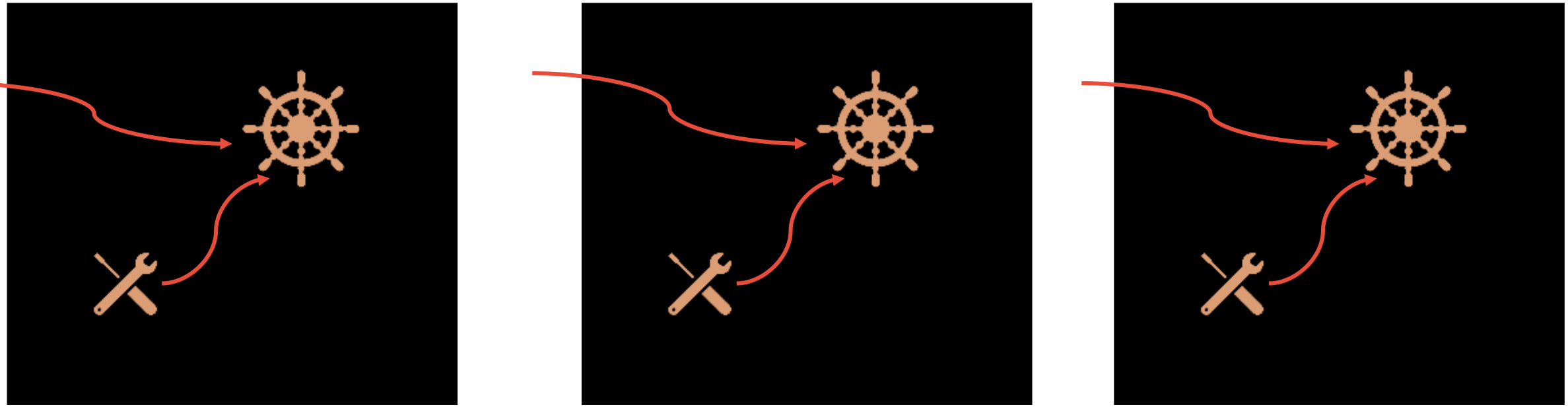


8. Backing Services



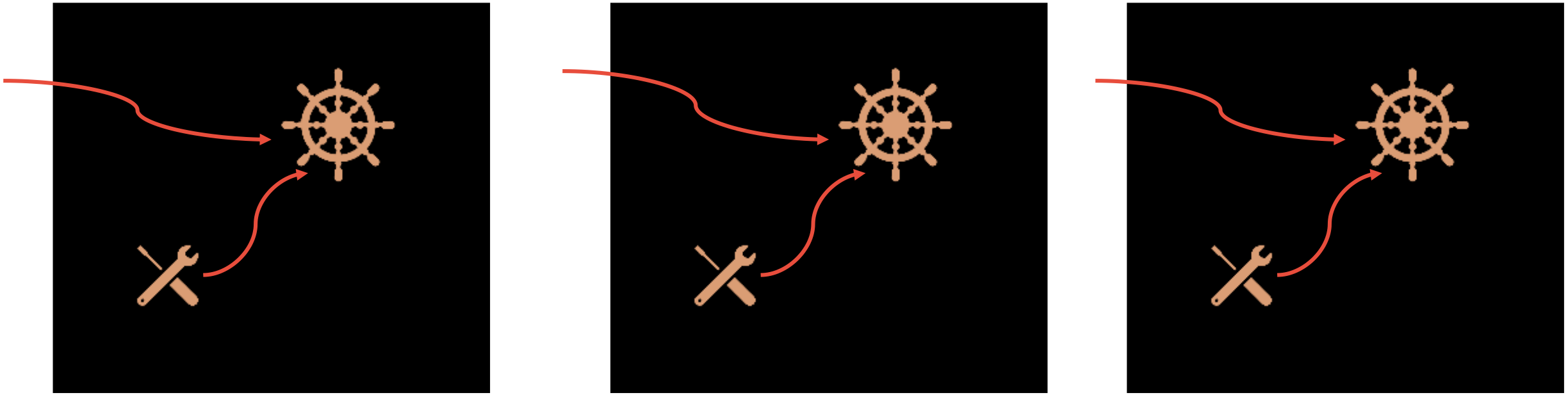
9. Environmental parity

Development = Q/A = Production



9. Environmental parity

Development = **Q/A** = **Production**



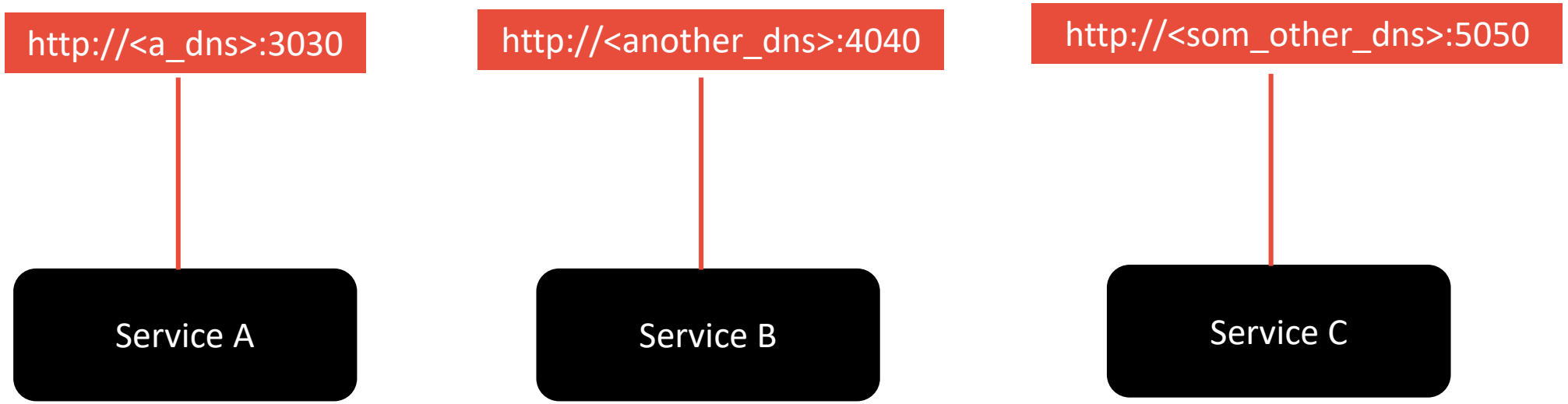
https://openliberty.io/guides/microshed-testing.html

10. Administrative processes

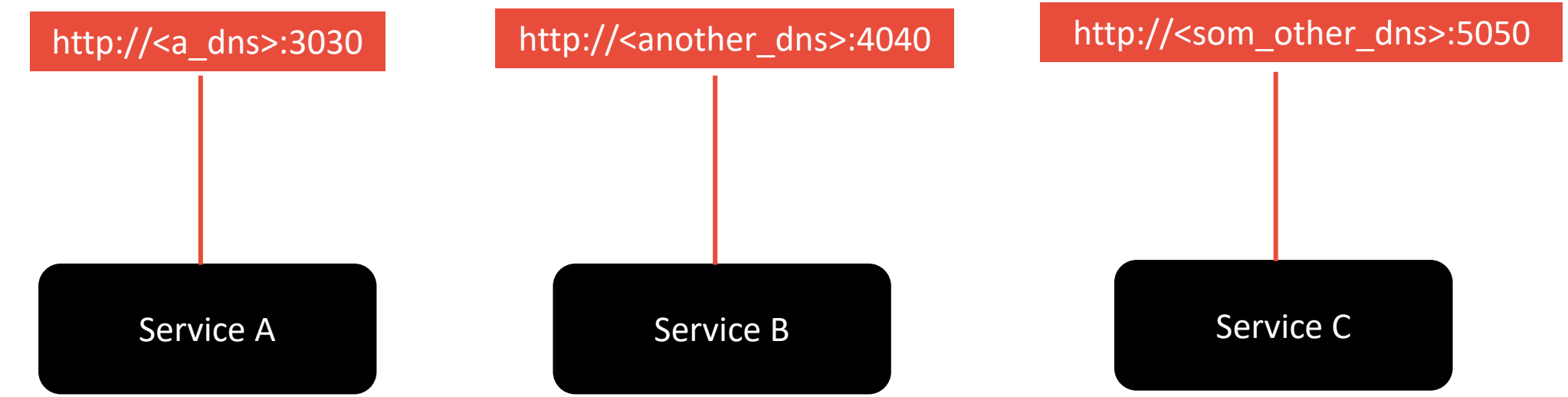
kubernetes

https://openliberty.io/guides/kubernetes-intro.html

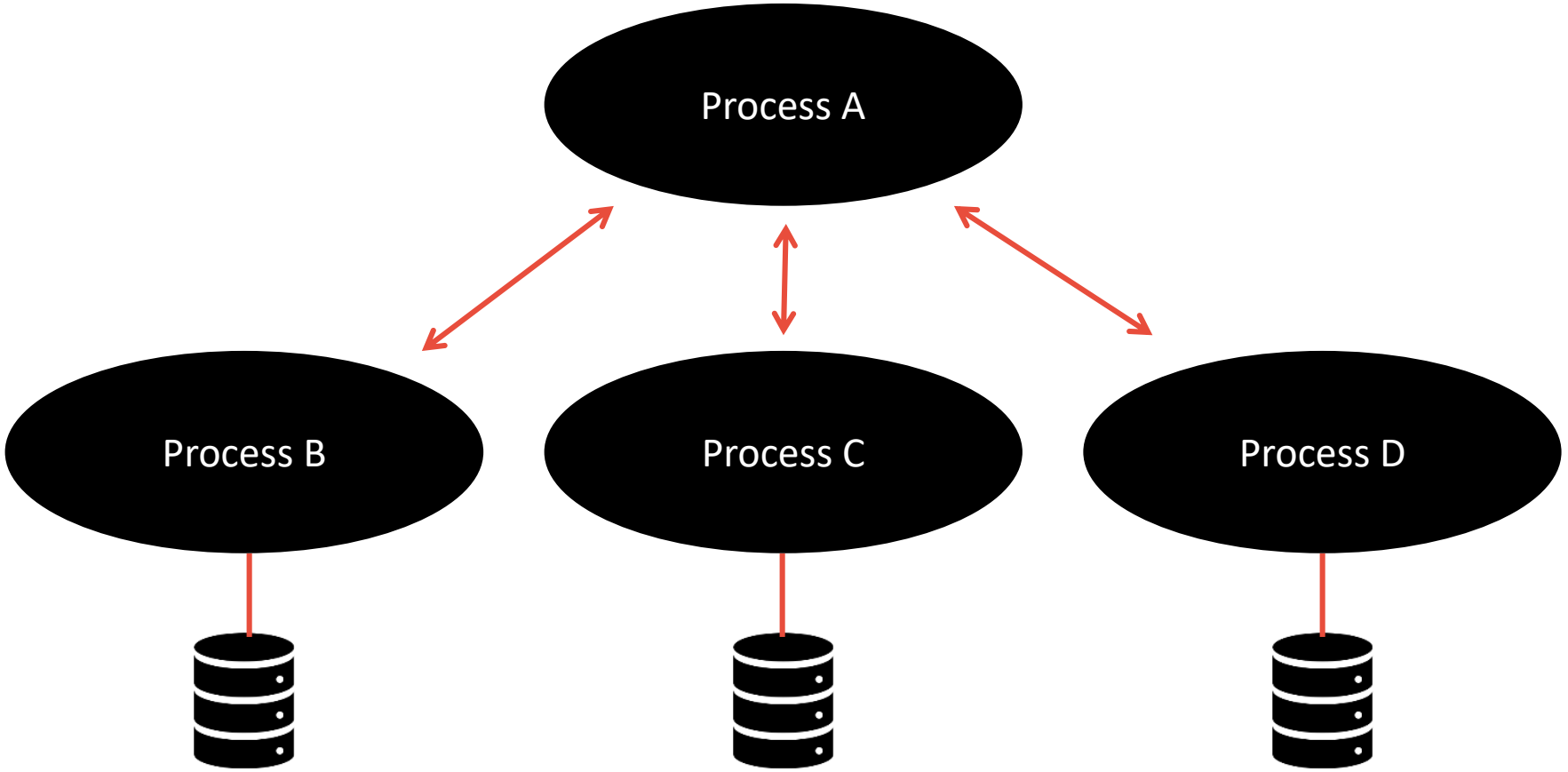
11. Port Binding



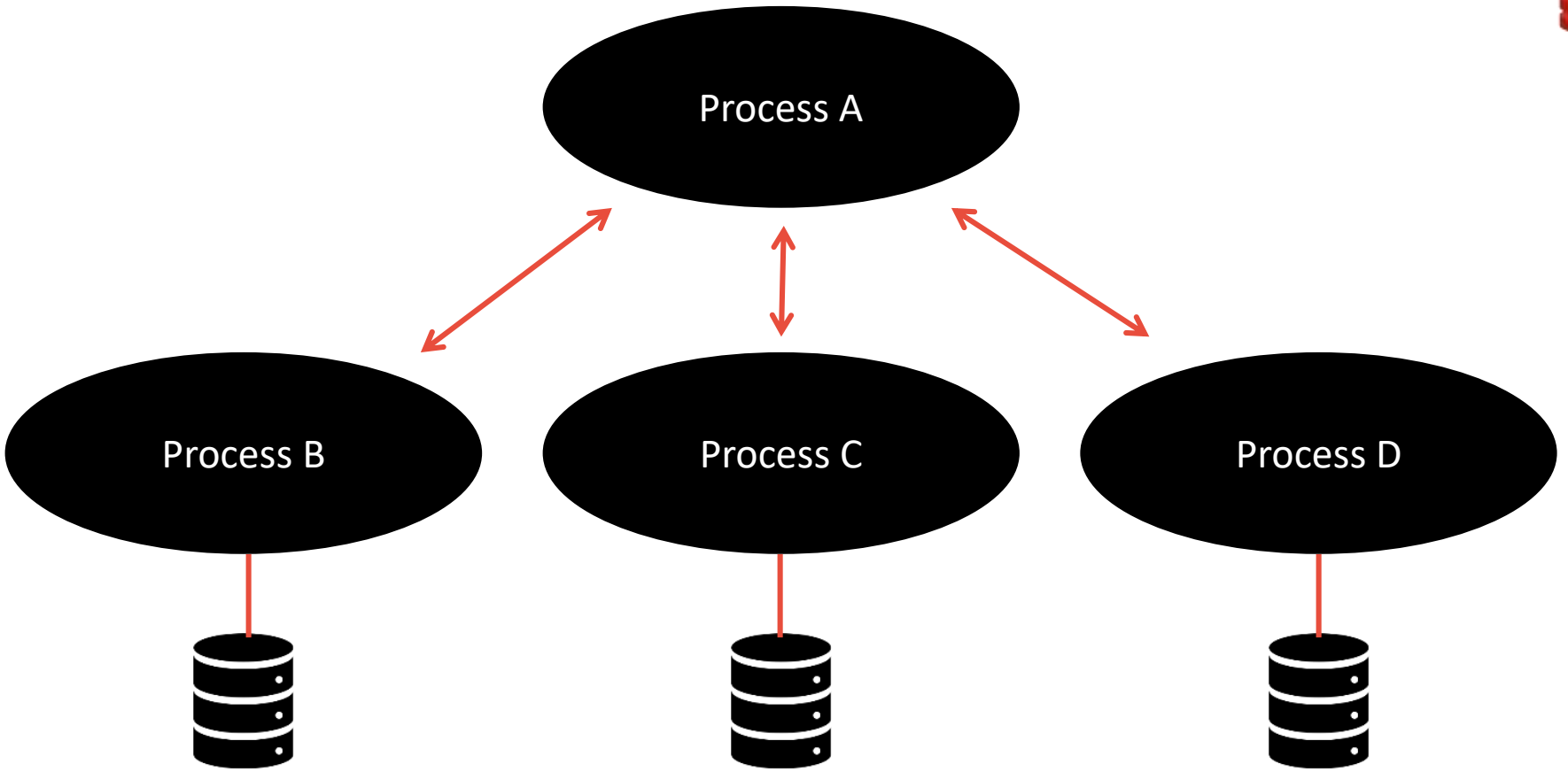
11. Port Binding



12. Stateless processes

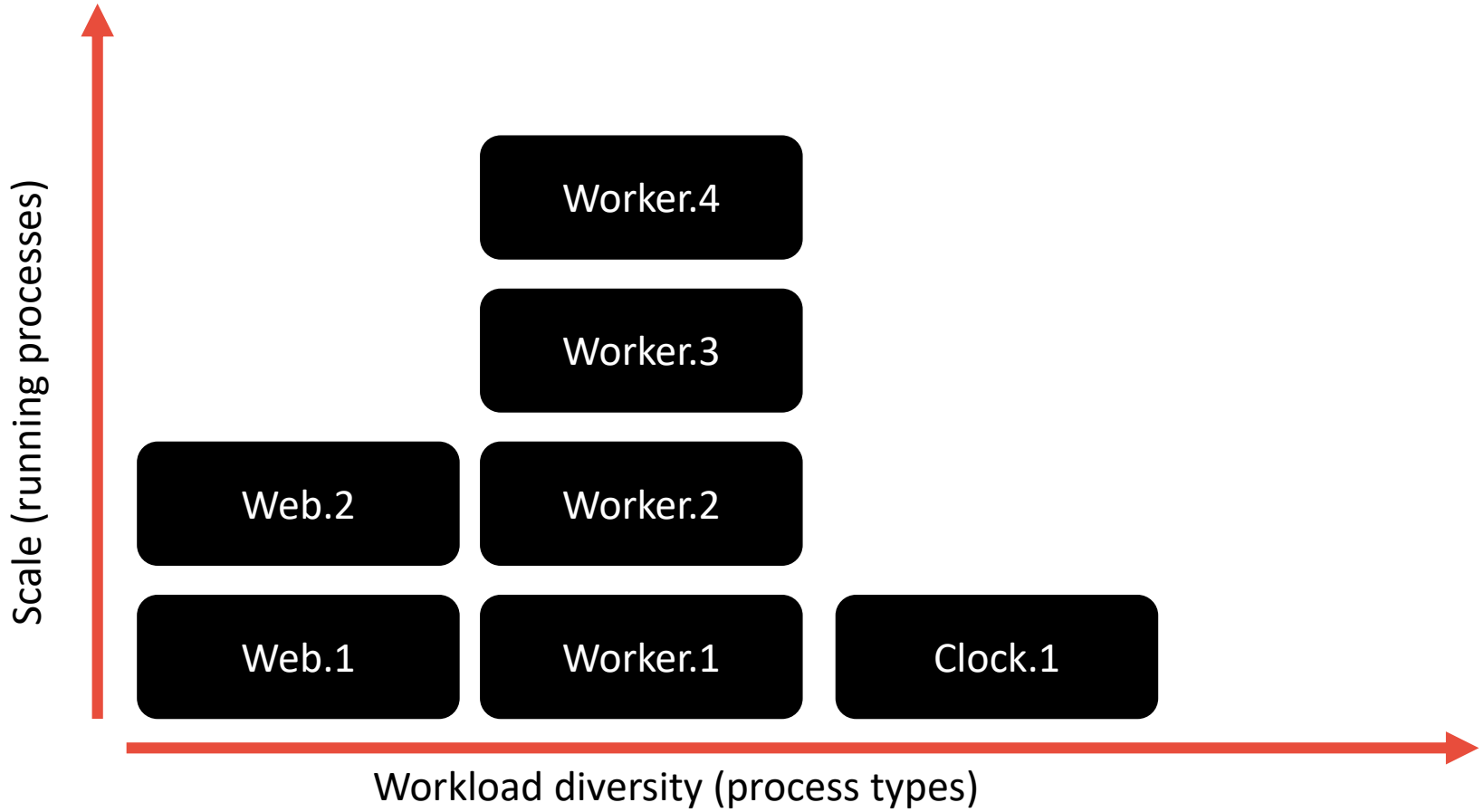


12. Stateless processes

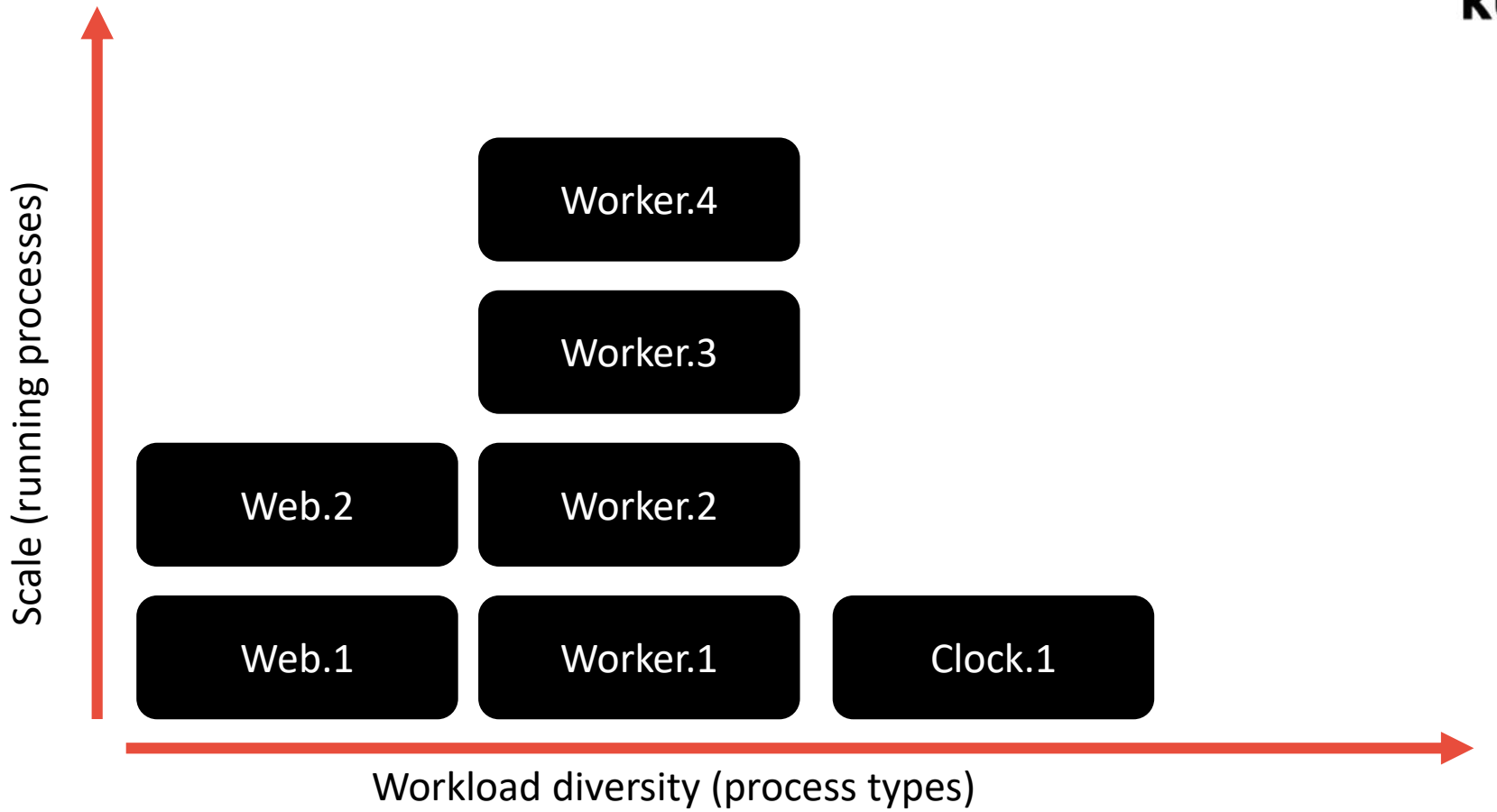


https://openliberty.io/guides/rest-intro.html

13. Concurrency

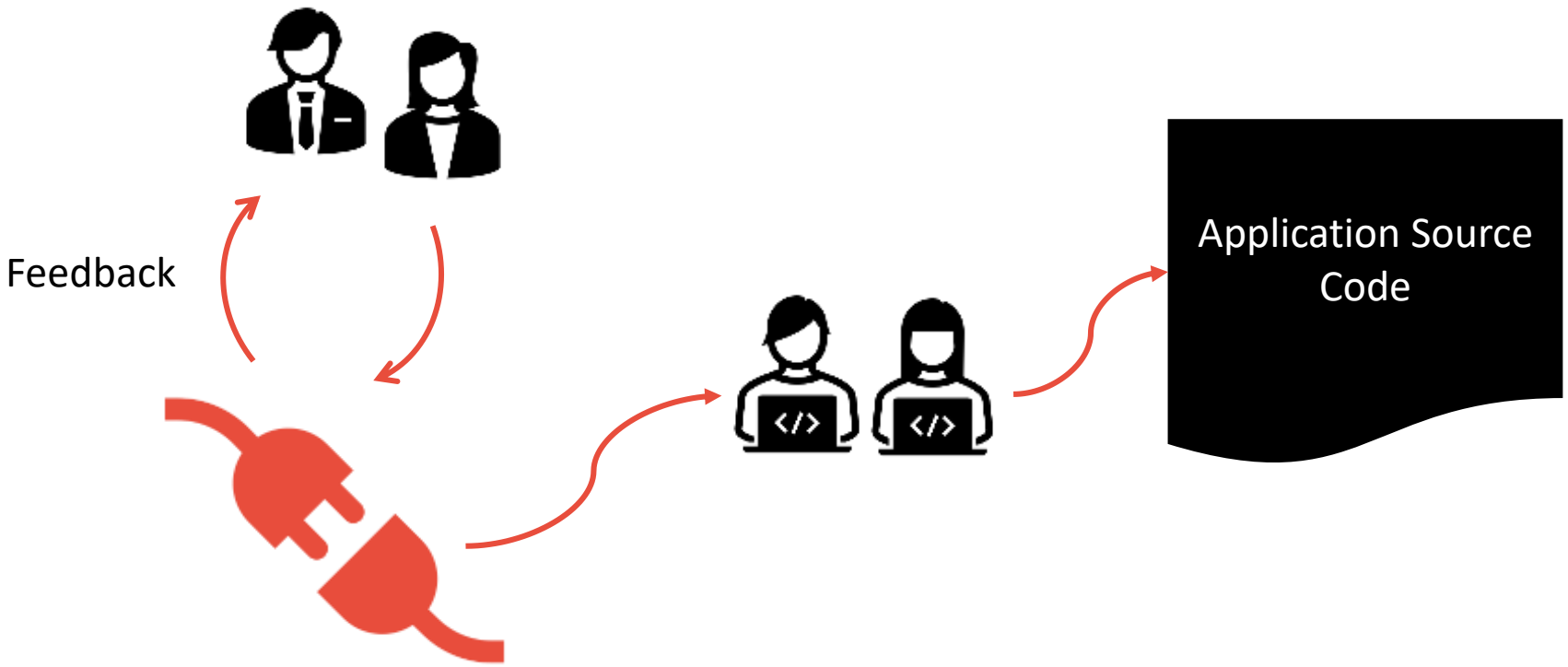


13. Concurrency



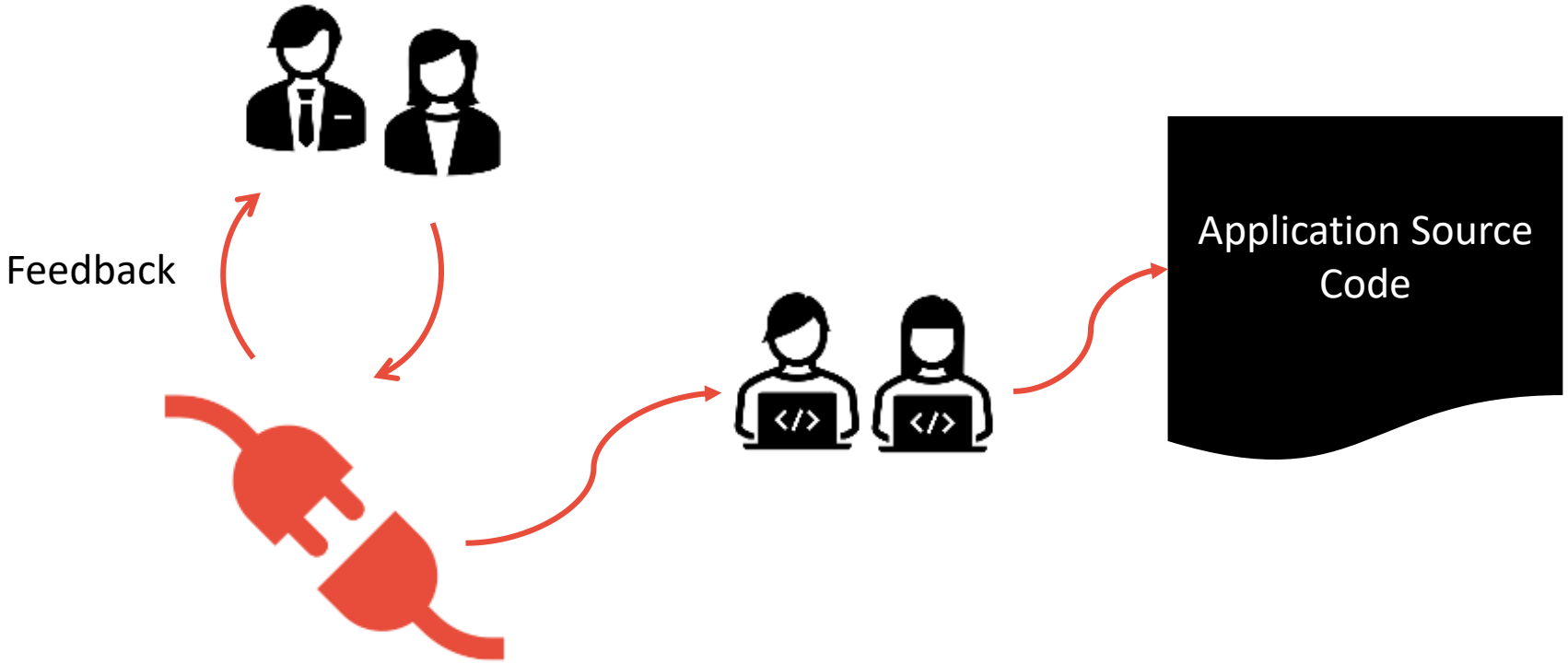
Thriving in the cloud through the revised 15 factors

2. API first



2. API first

https://openliberty.io/guides/microprofile-openapi.html

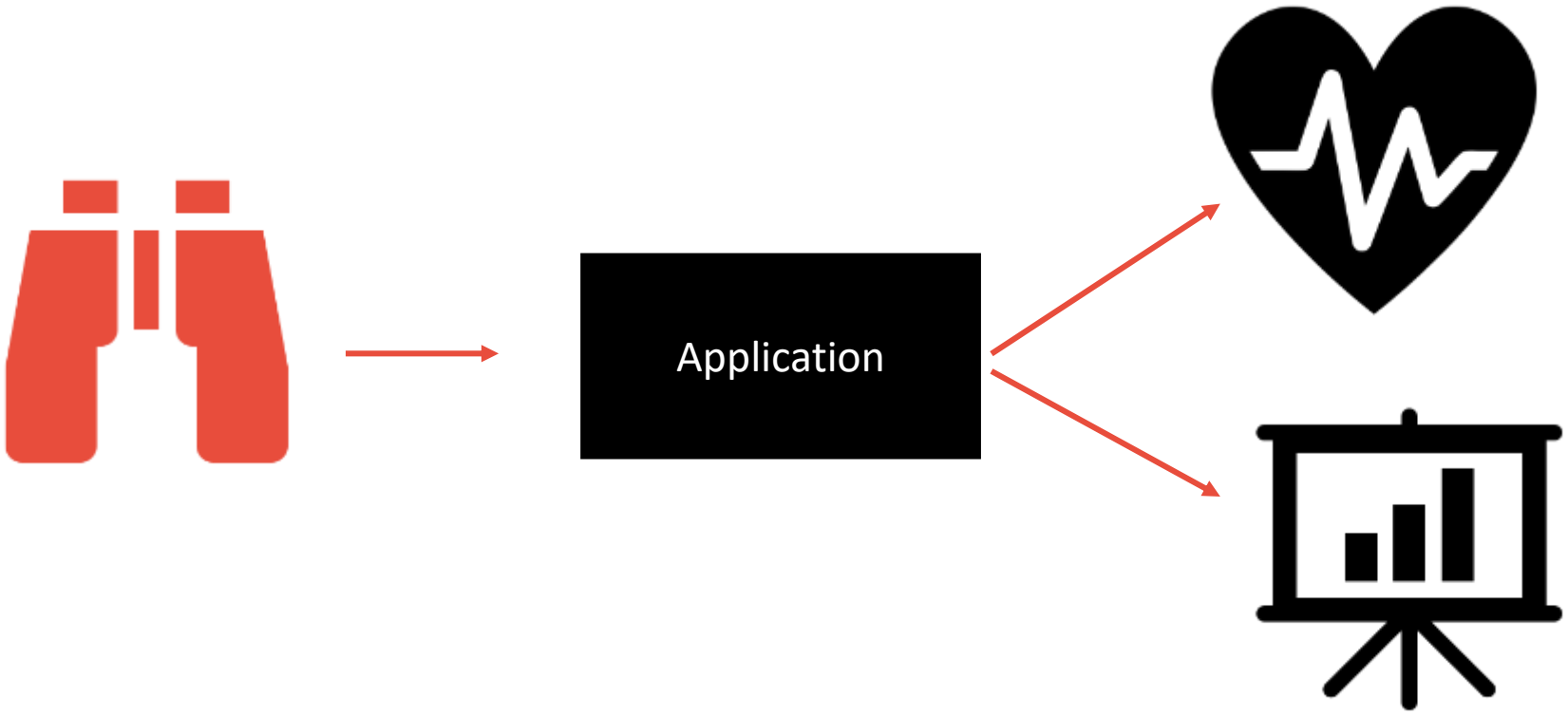


OPENAPI INITIATIVE

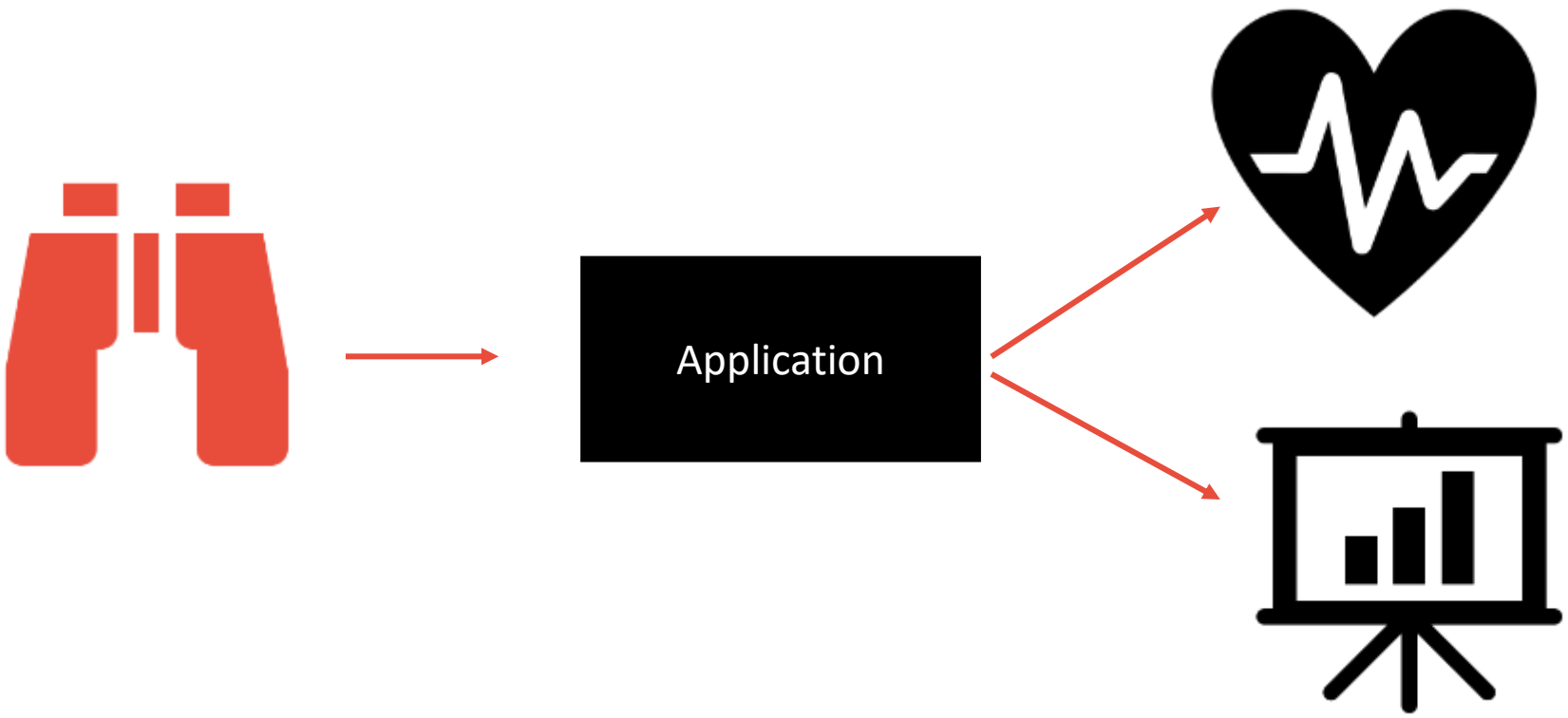
api blueprint

MICROPROFILE

14. Telemetry



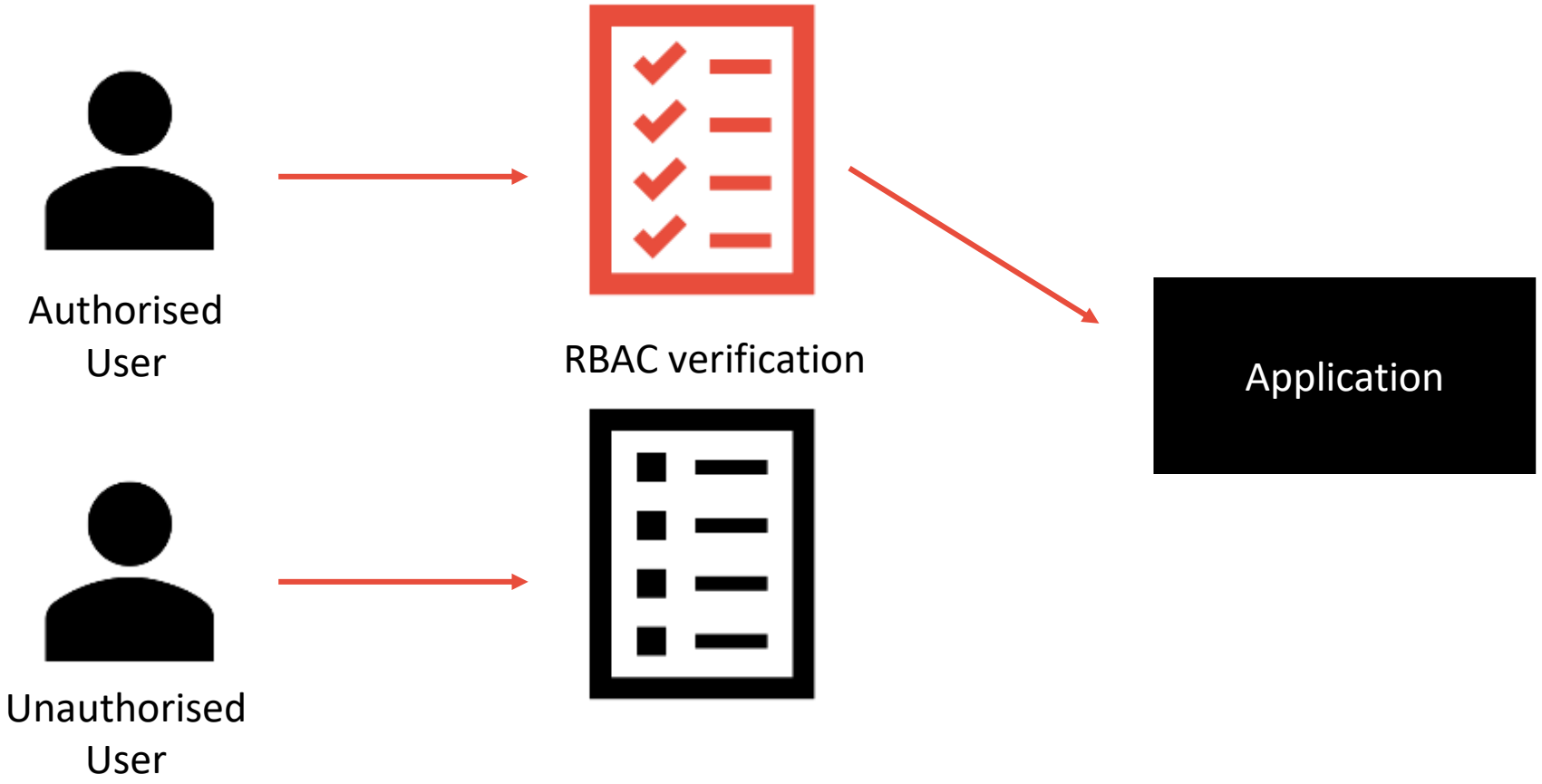
14. Telemetry



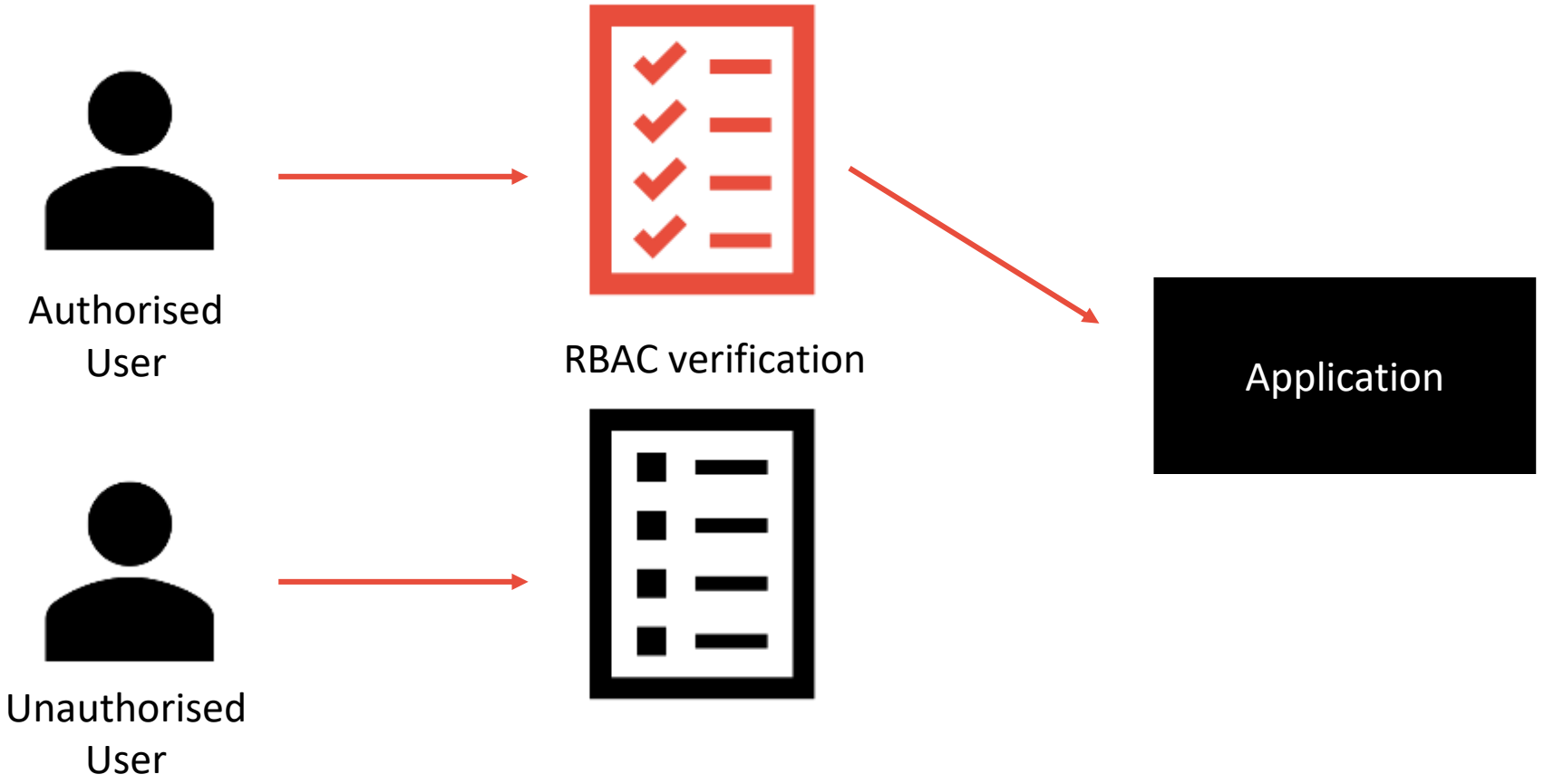
https://openliberty.io/guides/microprofile-metrics.html

https://openliberty.io/guides/microprofile-health.html

15. Authentication and authorization

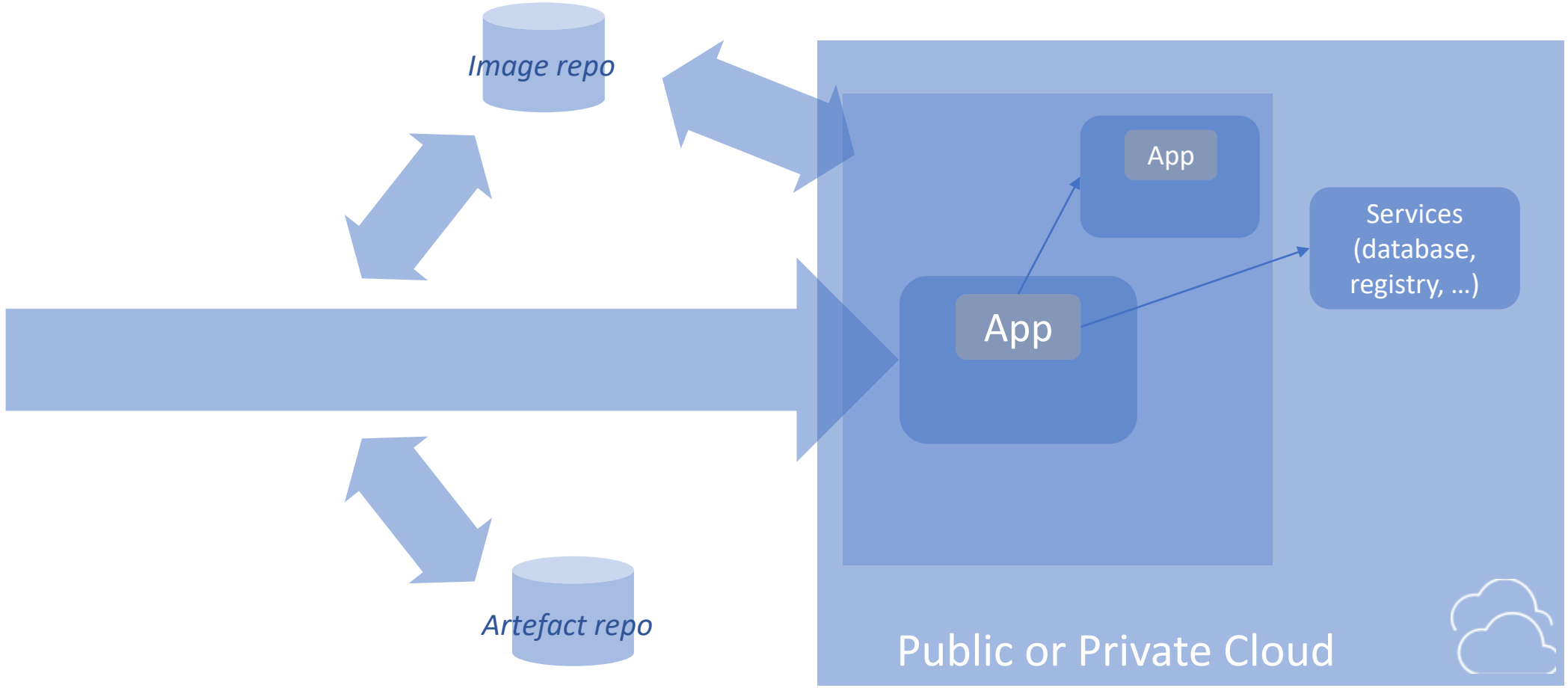


15. Authentication and authorization

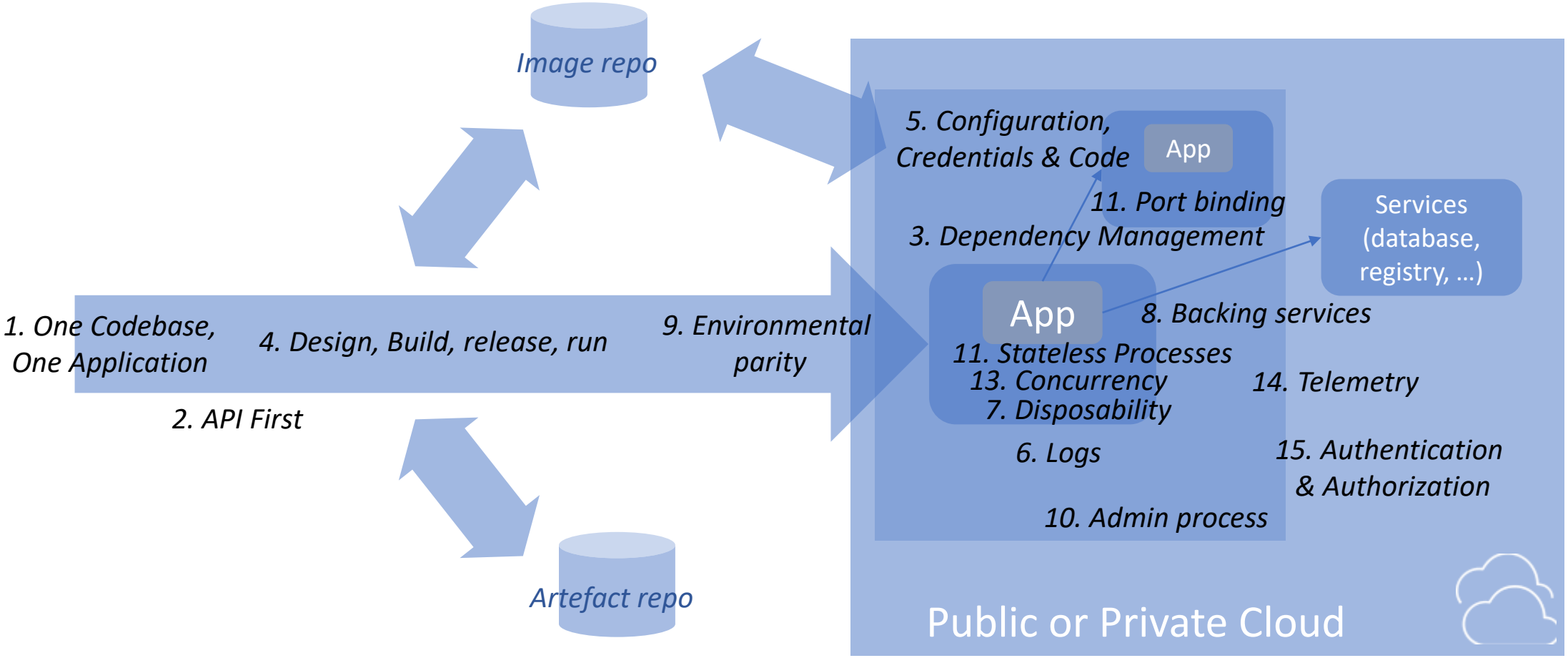


https://openliberty.io/guides/microprofile-jwt.html

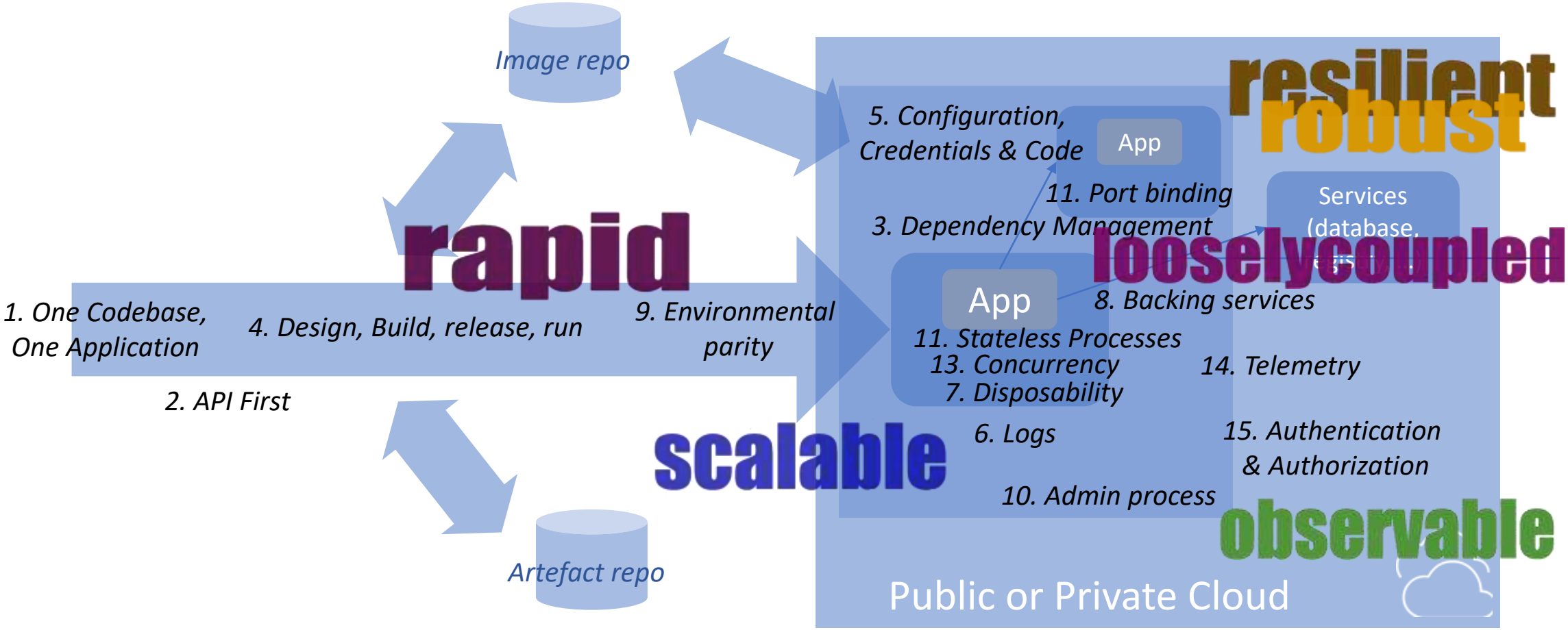
15-Factor App



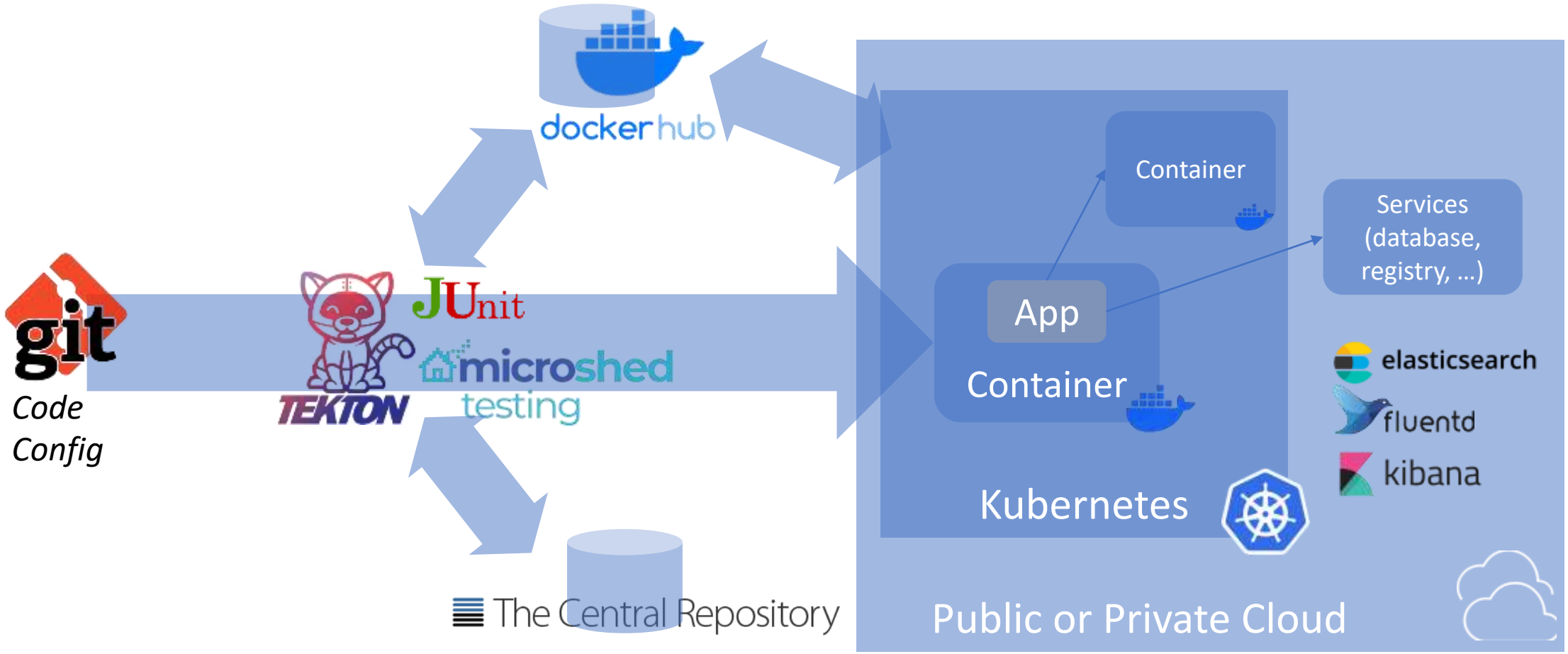
15-Factor App



15-Factor App

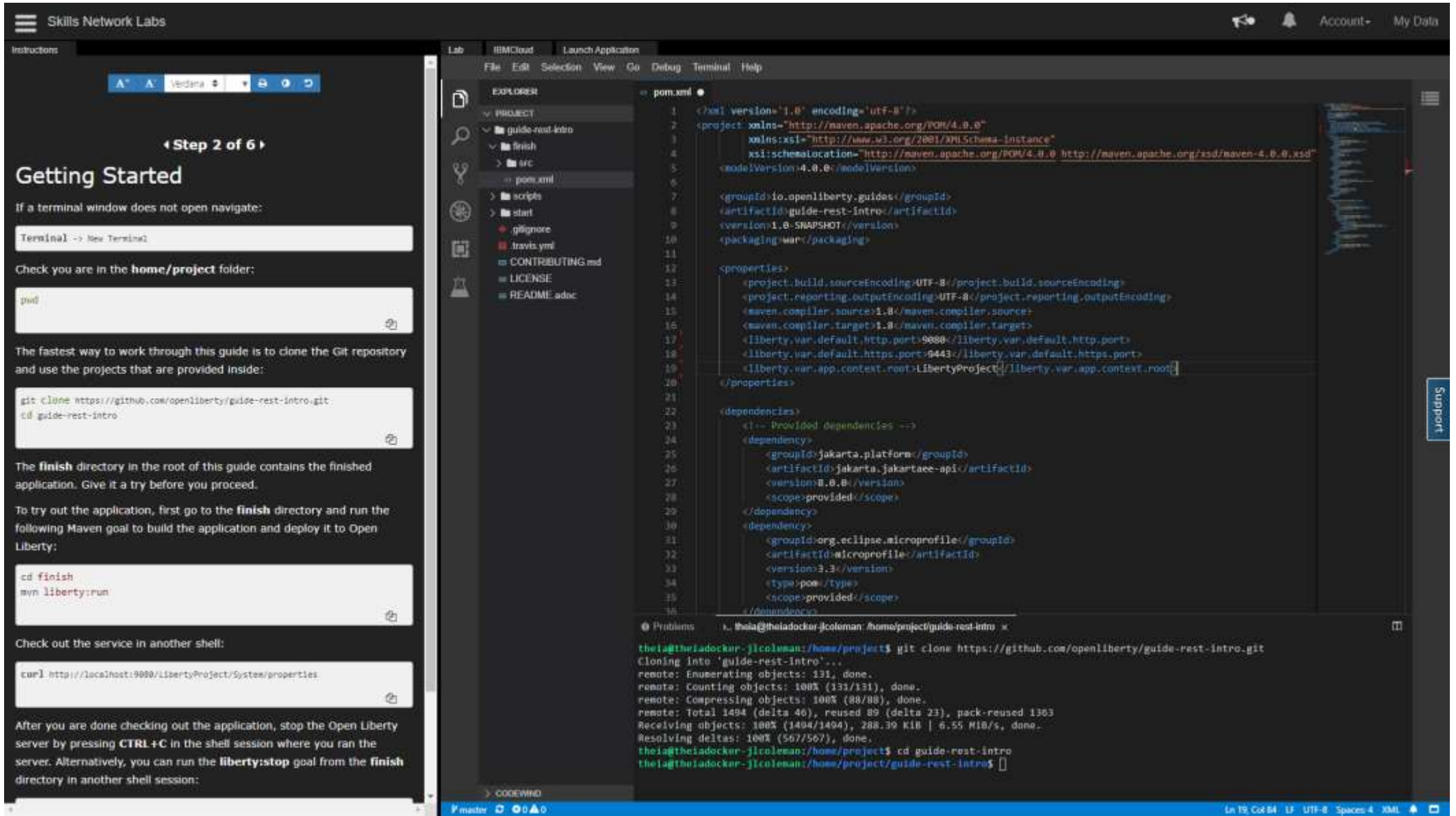


A Cloud-native instantiation



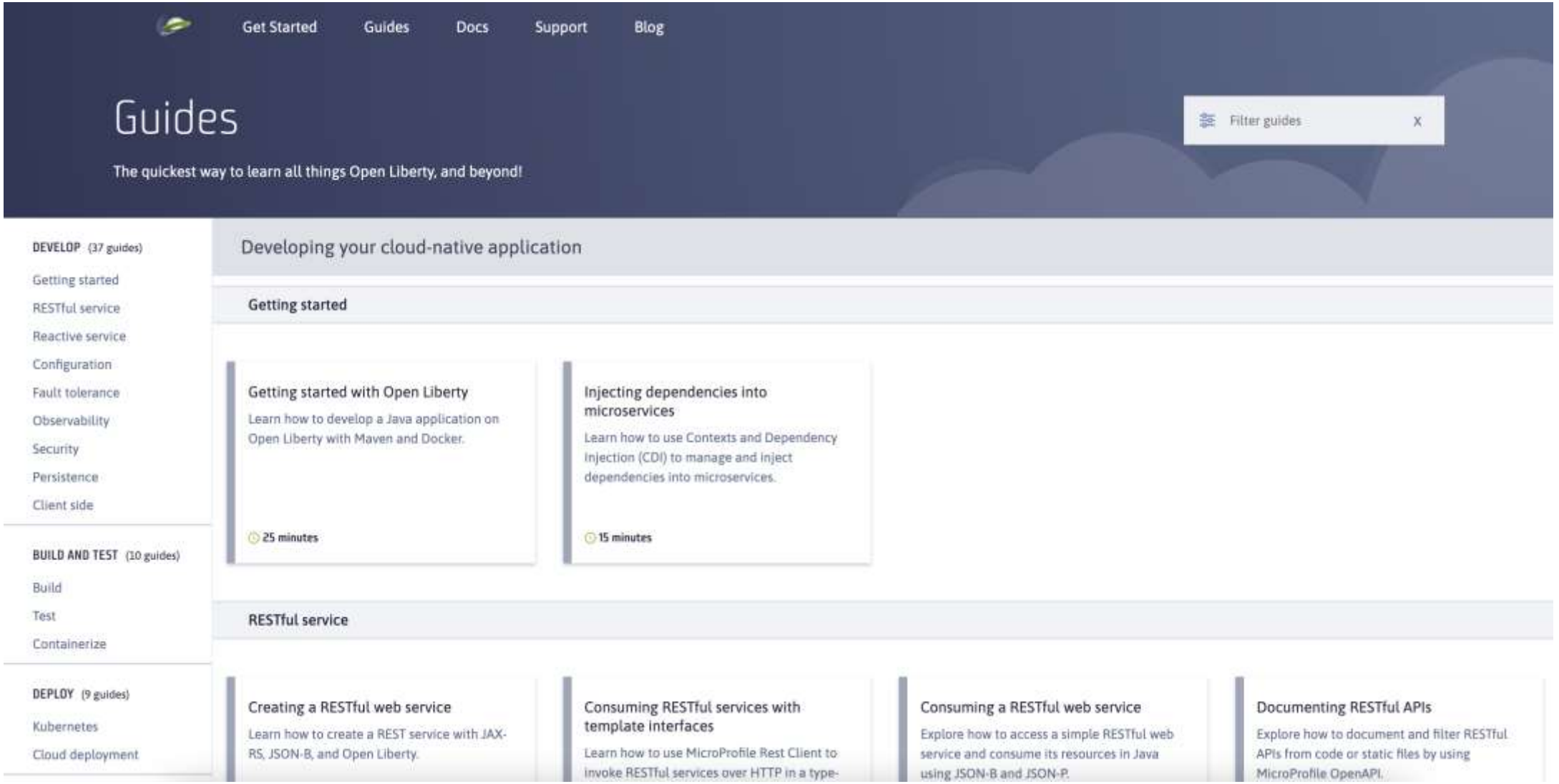
Hands on experience

Interactive cloud-native labs



Open Liberty Guides

https://openliberty.io/guides/



Summary

Summary:

- Twelve-factor applications = great start
- But… to thrive in the cloud, we need to look beyond the 12 factors
- No excuses!
  - Lots of open-source tools and technologies available to help
- Action: Evaluate your own applications against these 15 factors and consider what you could do to enable them to truly thrive in the cloud

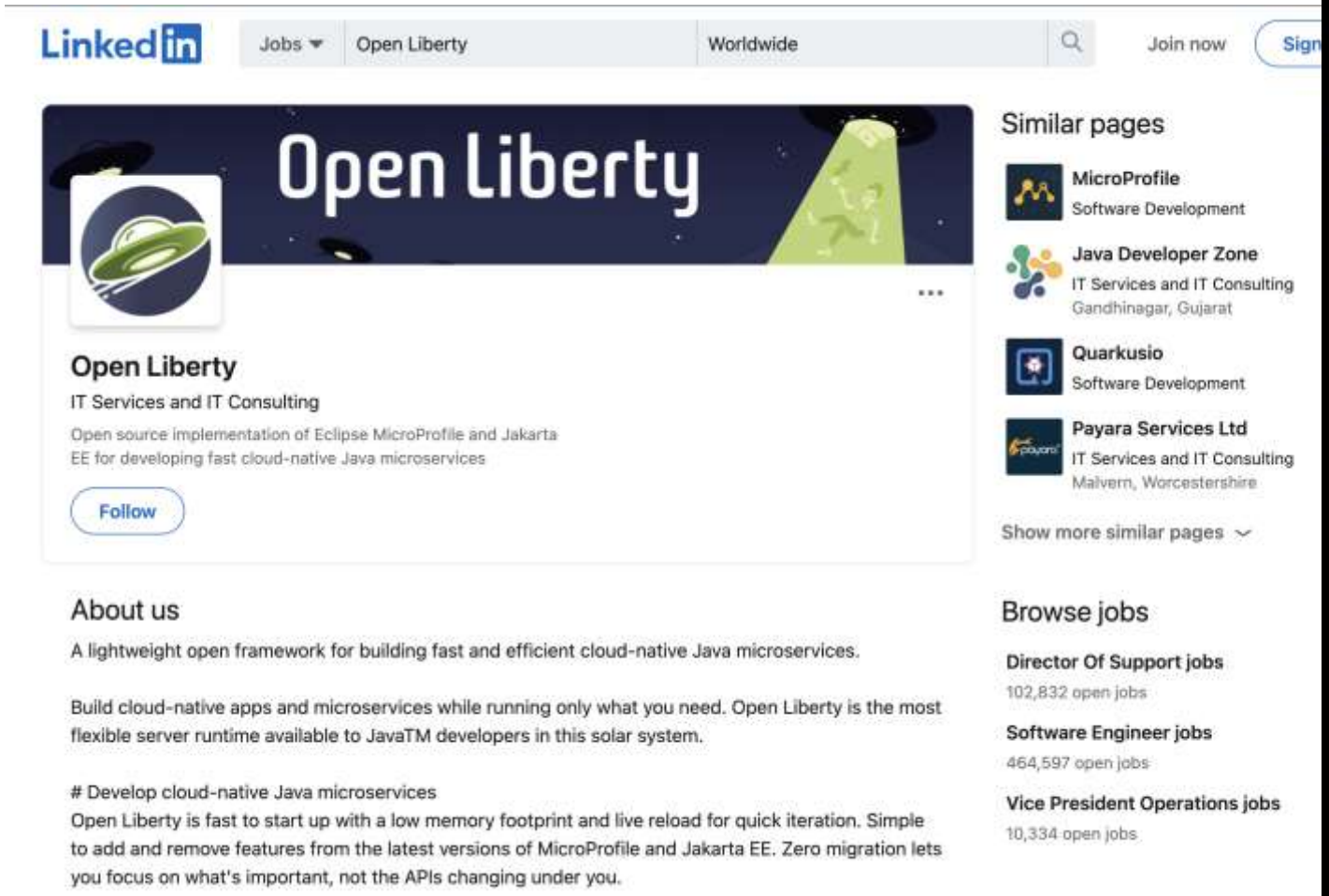
Useful Resources

- General:
  - https://openliberty.io/blog/2019/09/05/12-factor-microprofile-kubernetes.html
  - https://www.cdta.org/sites/default/files/awards/beyond_the_12-factor_app_pivotal.pdf
  - https://developer.ibm.com/articles/creating-a-12-factor-application-with-open-liberty/
- Design, build, release, run:
  - https://developer.ibm.com/devpractices/devops/patterns/make-continuous-delivery-easier-with-tekton-dashboards/
  - https://dzone.com/articles/deploying-microprofile-microservices-with-tekton
- Logging:
  - https://developer.ibm.com/videos/use-json-logging-in-open-liberty/
  - https://developer.ibm.com/videos/send-open-liberty-logs-to-elastic-stack/
  - https://openliberty.io/blog/2021/02/10/ocp-log-forwarding.html
  - https://community.ibm.com/community/user/communities/community-home/librarydocuments/viewdocument?DocumentKey=65596910-8d01-48d2-a99e-d94794d022af

Useful Resources

- Stateless Processes:
  - https://openliberty.io/guides/sessions.html
- Concurrency:
  - https://developer.ibm.com/technologies/containers/tutorials/autoscale-application-on-kubernetes-cluster/
- Authentication and Authorisation:
  - https://openliberty.io/blog/2021/03/26/MP-JWT-1.2.html
  - https://openliberty.io/blog/2020/09/04/securing-open-liberty-azure.html
- Open Liberty Tools:
  - https://openliberty.io/blog/2021/04/21/admin-center-21004.html

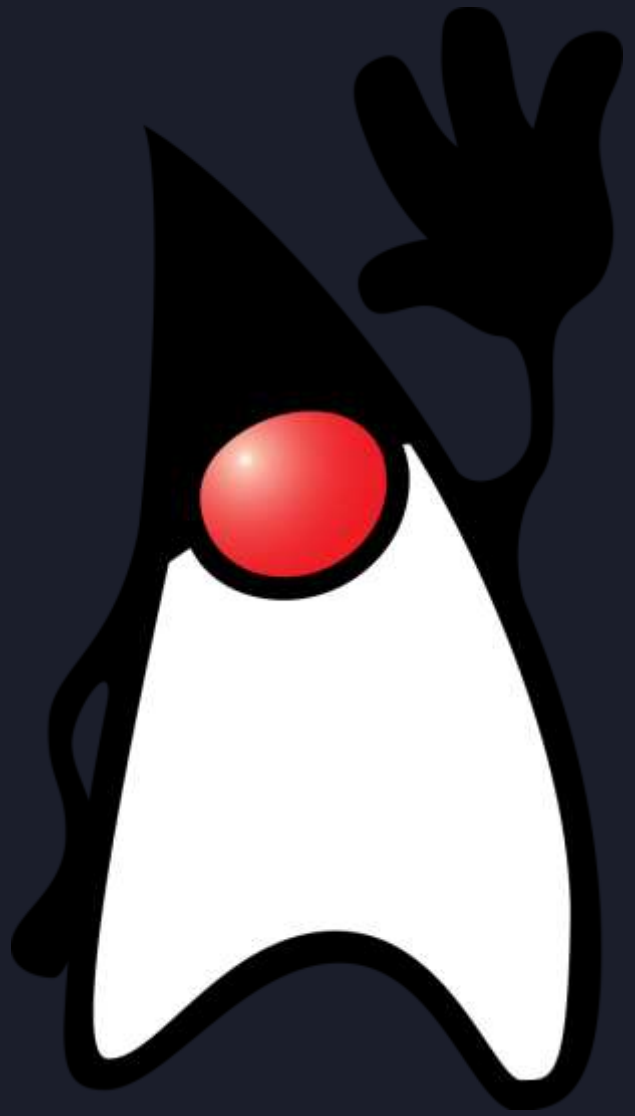
Connect with us



https://www.linkedin.com/company/openlibertyio/

https://twitter.com/OpenLibertyIO

THANK YOU



@gracejansen27