Open liberty

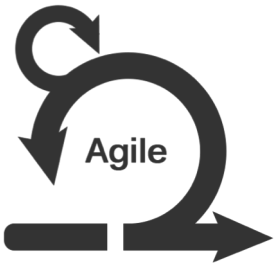
Practical Cloud-native Java with Eclipse MicroProfile

Alasdair Nottingham
@NottyCode

Background to Cloud-native

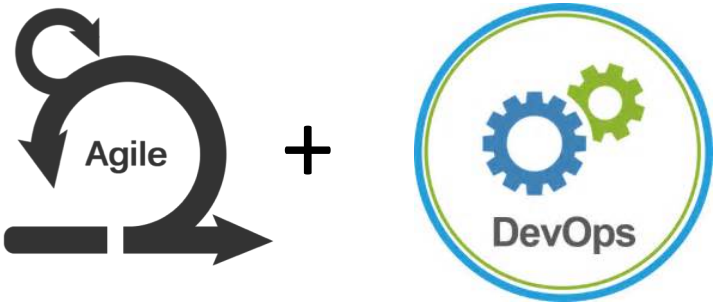
Agile

Open liberty



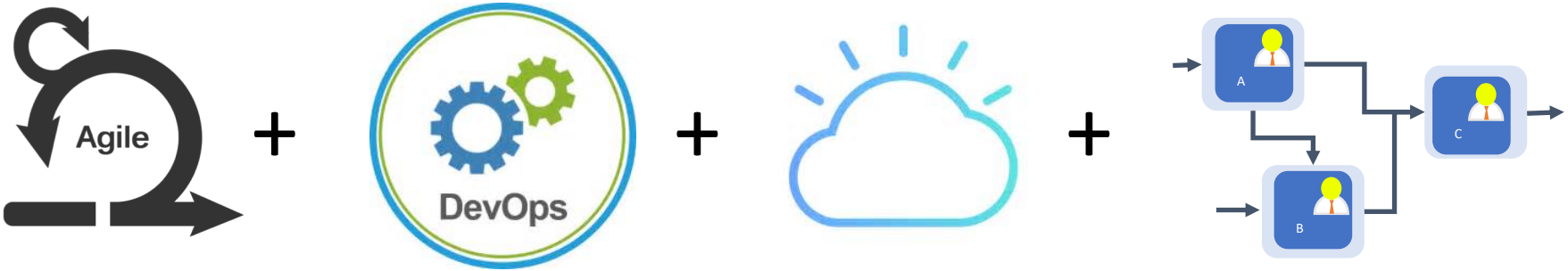
Agile & DevOps

Open liberty



Agile & DevOps & Cloud

Agile & DevOps & Cloud & Microservices



Good characteristics of a cloud-native environment

Open Liberty

Provides APIs for distributed computing

Open liberty

Starts fast and shuts down clean

Has a (proportionately) small footprint

Facilitates dev/prod parity, including through externalized config

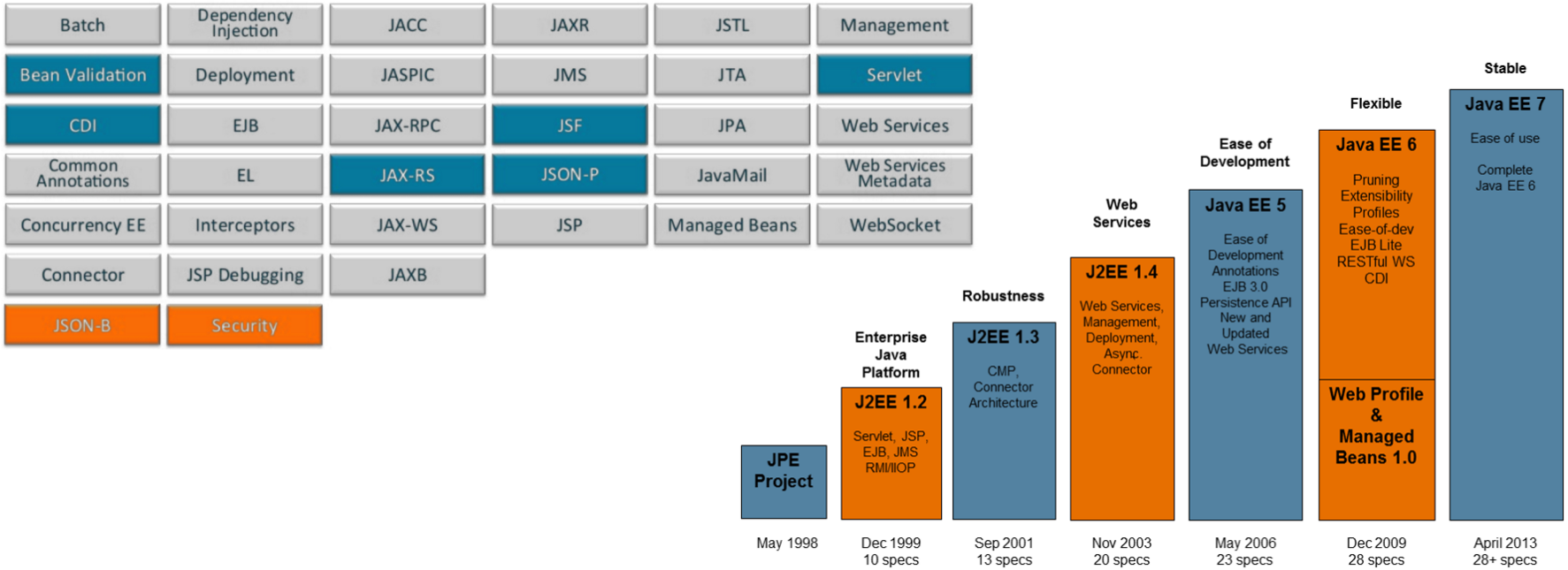
Can be easily containerized

Being fast, small and open

Open Liberty

Middle-aged spread

Open Liberty



https://www.slideshare.net/delabassee/java-ee-8-february-2017-update
https://medium.com/@alextheedom/java-ee-past-present-future-8bf25df7b6a3

Liberty Features

server

```
<featureManager>
 <feature>jaxrs-2.0</feature>
 <feature>openapi-3.0</feature>
</featureManager>
```

build

```
<packaging.type>minify,runnable</packaging.type>
```

~~Wildfly Swarm~~ Thorntail Fractions

build

```
<dependency>
  <groupId>org.wildfly.swarm</groupId>
  <artifactId>jaxrs</artifactId>
</dependency>
<dependency>
  <groupId>org.wildfly.swarm</groupId>
  <artifactId>swagger</artifactId>
</dependency>
```

What does it mean to provide microservice technologies?

Open Liberty

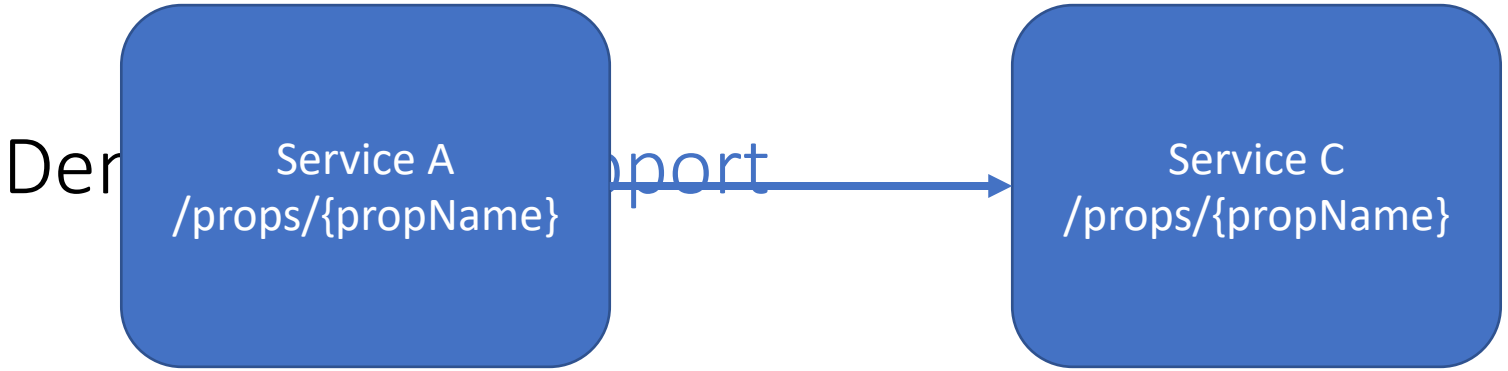

MICROPROFILE
OPTIMIZING ENTERPRISE JAVA
PROJECTS
PRESENTATIONS
FAQ
BLOG
CONTRIBUTORS
JOIN THE DISCUSSION
SEE THE CODE
Eclipse MicroProfile
Optimizing Enterprise Java for a microservices architecture
IBM
LJC
redhat
Tomitribe
payara
SOUJava
hazelcast
FUJITSU
kumuluzEE
ORACLE
Hammock
Lightbend
Microsoft

There’s a good chance you’ll use REST APIs.

Eclipse MicroProfile

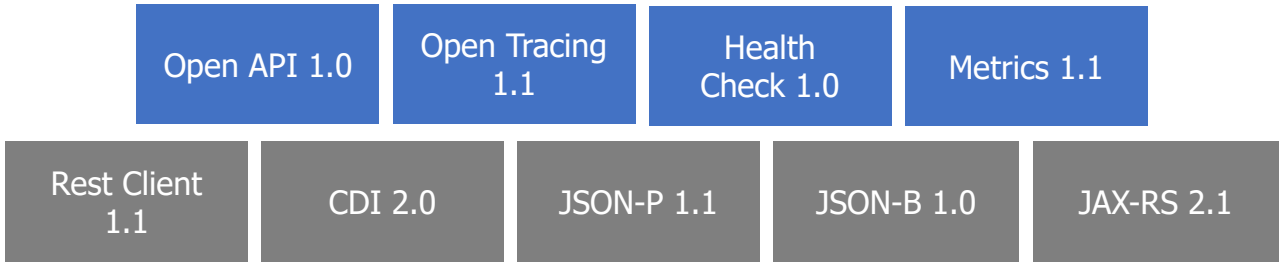
Rest Client 1.1	CDI 2.0	JSON-P 1.1	JSON-B 1.0	JAX-RS 2.1

Demo of REST support


Service A
/props/{propName}
Service C
/props/{propName}

Handling “100s” of collaborating services requires a strong operations focus

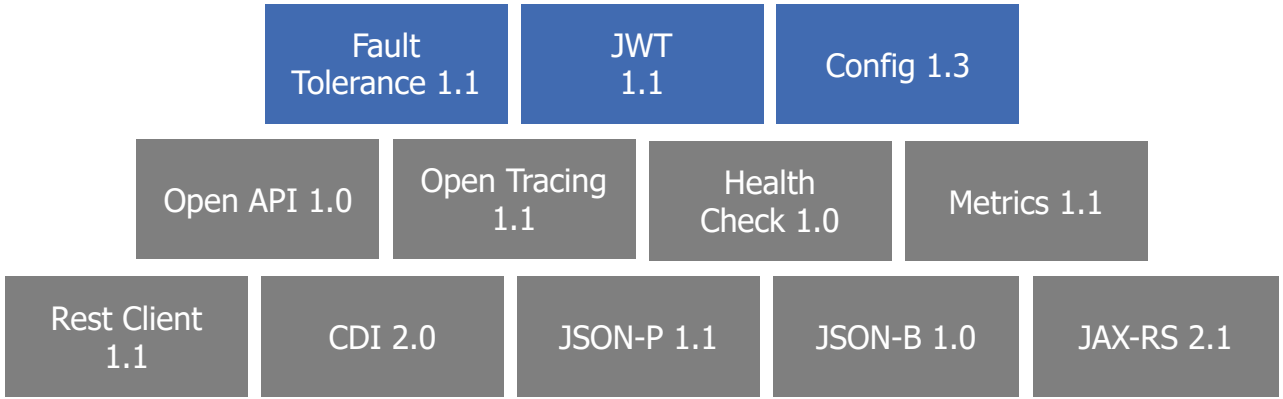
Eclipse MicroProfile



Demo openapi, health & metrics

Handling ”100s” of collaborating and frequently evolving services requires new APIs

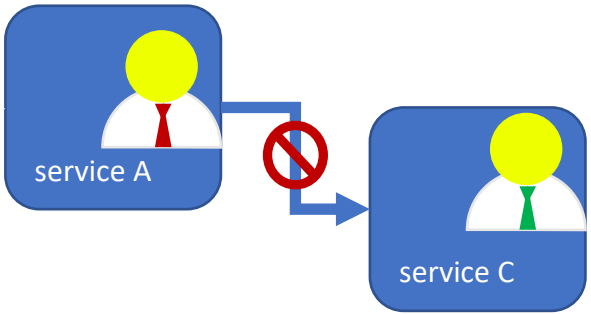
Eclipse MicroProfile



Demo of config and Fault Tolerance

28

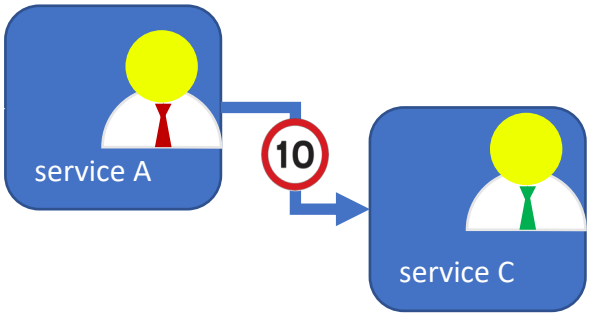
Fault Tolerance in microservices



Circuit breaker

```
@CircuitBreaker(
failOn=IOException.class,
   delay = 500)
public void callServiceC() {
    // call the service
}
```

Fault Tolerance in microservices

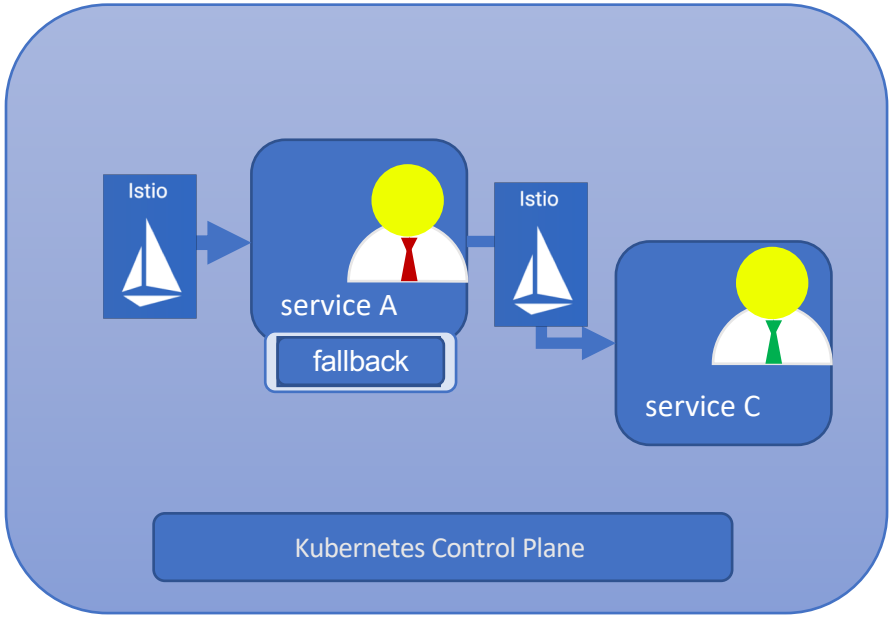


Bulkhead

```
@Asynchronous
@Bulkhead(value = 10,
   waitingTaskQueue = 15)
public void callServiceC() {
    // call the service
}
```

Can’t I do all this with a service mesh?

Fault Tolerance in microservices



App Container can defer to Cloud Platform

```
@Retry
@Timeout
@CircuitBreaker
@Bulkhead
```

Application still provides:

```
@Fallback
```

- What is next?

Open Liberty

33

- Reactive
- Data access
- Istio integration
- Updates to existing specs

34

Packaging for deployment

Open Liberty

Making the most of Docker

Summary

- Consider both organizational and technological changes to increase likelihood of success
- Leverage MicroProfile to solve cloud-native challenges
- Choose appropriate packaging for your cloud
- Reduce overheads and cost with a right-sizeable runtime
- In Docker, strive for a thin application layer
- Learn more with Open Liberty guides https://ibm.biz/mpGuides

Open liberty

Thank You