



JAKARTA EE

# Microservice Patterns

## Implemented by Eclipse MicroProfile

Ivar Grimstad 

*Jakarta EE Developer Advocate, Eclipse Foundation*





JAKARTA EE



@ivar\_grimstad



<https://github.com/ivargrimstad>



<https://www.linkedin.com/in/ivargrimstad>



 **Java™**  
Community Process  
Executive Committee  
Member



JAKARTA EE  
MEMBER

 **ECLIPSE**  
FOUNDATION

 THE **APACHE**  
SOFTWARE FOUNDATION

**JAVA**  
**FORUM**  
MALMØ



MICROPROFILE™  
OPTIMIZING ENTERPRISE JAVA



JAKARTA EE

# Eclipse MicroProfile



JAKARTA EE

Home / Projects / Eclipse Technology / Eclipse MicroProfile

# Eclipse MicroProfile

Overview

Downloads

Who's Involved

Developer Resources

Governance

Contact Us

The Eclipse MicroProfile project is aimed at optimizing Enterprise Java for the microservices architecture.

Many innovative "microservice" Enterprise Java environments and frameworks already exist in the Java ecosystem. These projects are creating new features and capabilities to address microservice architectures -- leveraging both Java EE and non-Java EE technologies.

The goal of the Eclipse MicroProfile project is to iterate and innovate in short cycles to propose new common APIs and functionality, get community approval, release, and repeat. Eventually, the outputs of this project could be submitted to the Eclipse Jakarta EE, JCP, OpenJDK or any relevant standards body.

## Licenses:

[Apache License, Version 2.0](#)



**ECLIPSE**  
INCUBATION

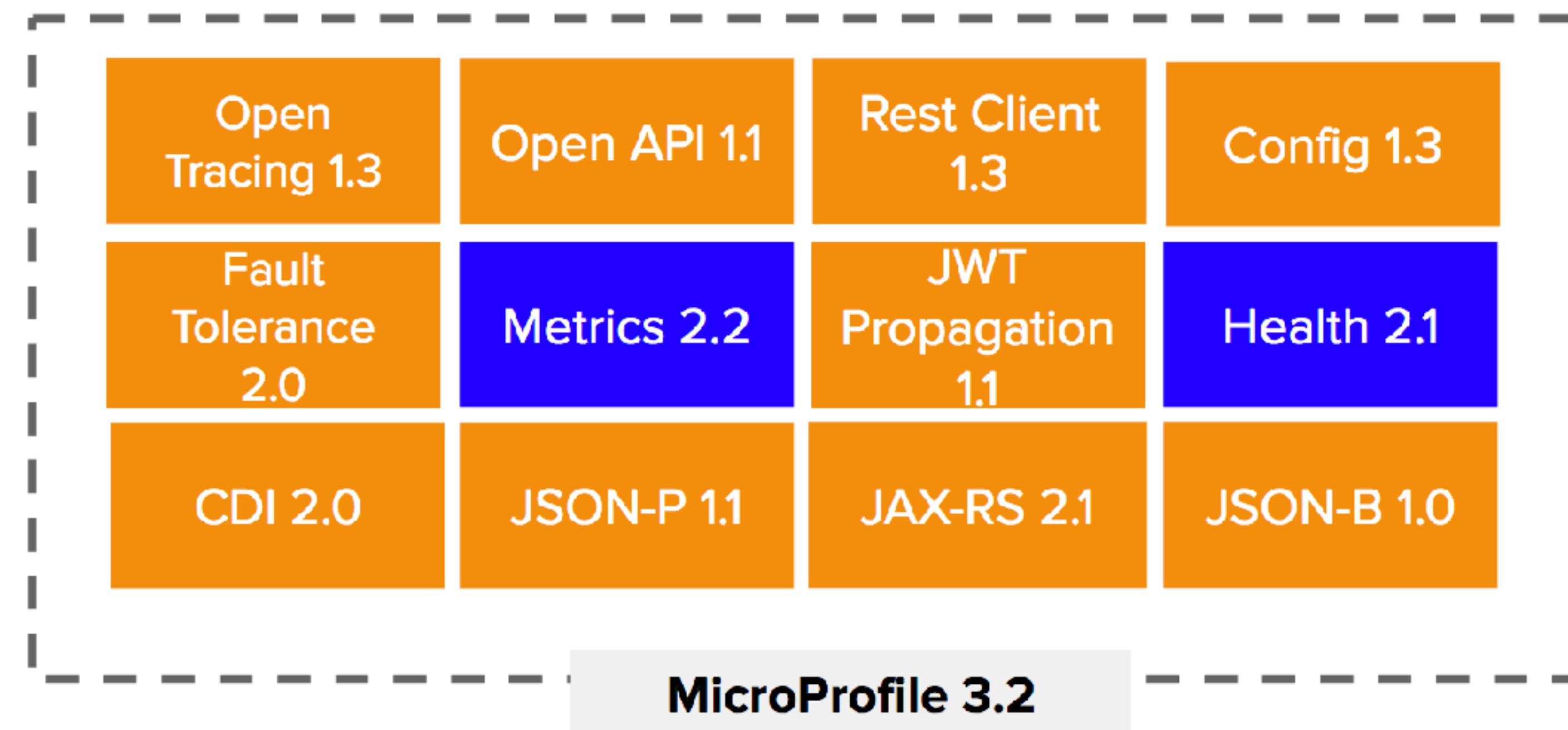




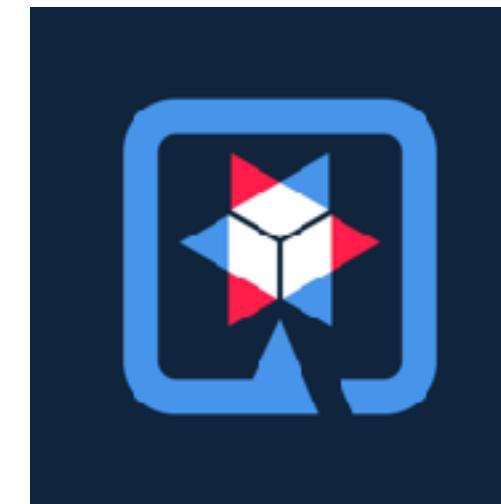
JAKARTA EE

# MicroProfile 3.2

November 1, 2019



- = New
- = Updated
- = No change from last release (MicroProfile 3.1)



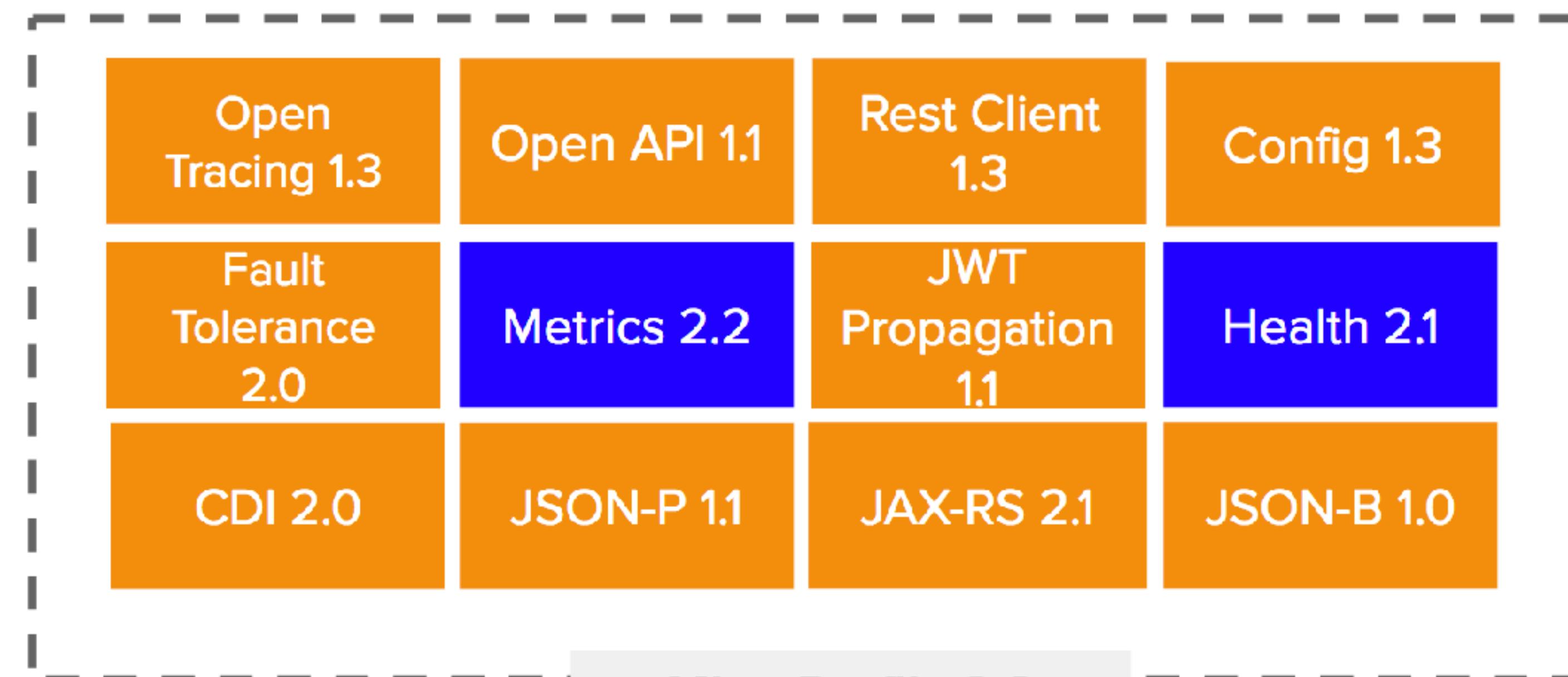


JAKARTA EE

**java -jar dukes.jar**



JAKARTA EE



= New

= Updated

= No change from last release (MicroProfile 3.1)



JAKARTA EE

# Getting Started



JAKARTA EE

The screenshot shows the Eclipse MicroProfile website on a browser. The top navigation bar includes links for Projects, Resources, Starter "Beta" (which is circled in white), Blog, Contributors, Join The Discussion, and See The Code. The main content features the Eclipse MicroProfile logo and the text: "Eclipse MicroProfile Optimizing Enterprise Java for a Microservices Architecture". It also mentions "Now Available Eclipse MicroProfile 3.2" and a "More Information" button. A large image in the background shows a laptop displaying code, a coffee cup, and a hand holding a smartphone. Overlaid on this image is a graphic of the Java mascot, Duke, wearing a sash that reads "ORACLE" and a badge that says "Duke's Choice Award 2018 Winner". The Java logo is also present at the bottom of this graphic.



Give Feedback

TA EE

# MicroProfile Starter "Beta"

Generate MicroProfile Maven Project with Examples

groupId \*

com.example

artifactId \*

demo

MicroProfile Version

Java SE Version

## Project Options

MicroProfile Runtime \*

Examples for specifications

DOWNLOAD



JAKARTA EE

# Microservice Patterns



JAKARTA EE

- 1. Health Check API**
- 2. Application Metrics**
- 3. Externalised Configuration**
- 4. Circuit Breaker**
- 5. Access Token**
- 6. Distributed Tracing**
- 7. API Description**



JAKARTA EE

# Health Check API



JAKARTA EE

# How to detect that a running service instance is unable to handle requests?



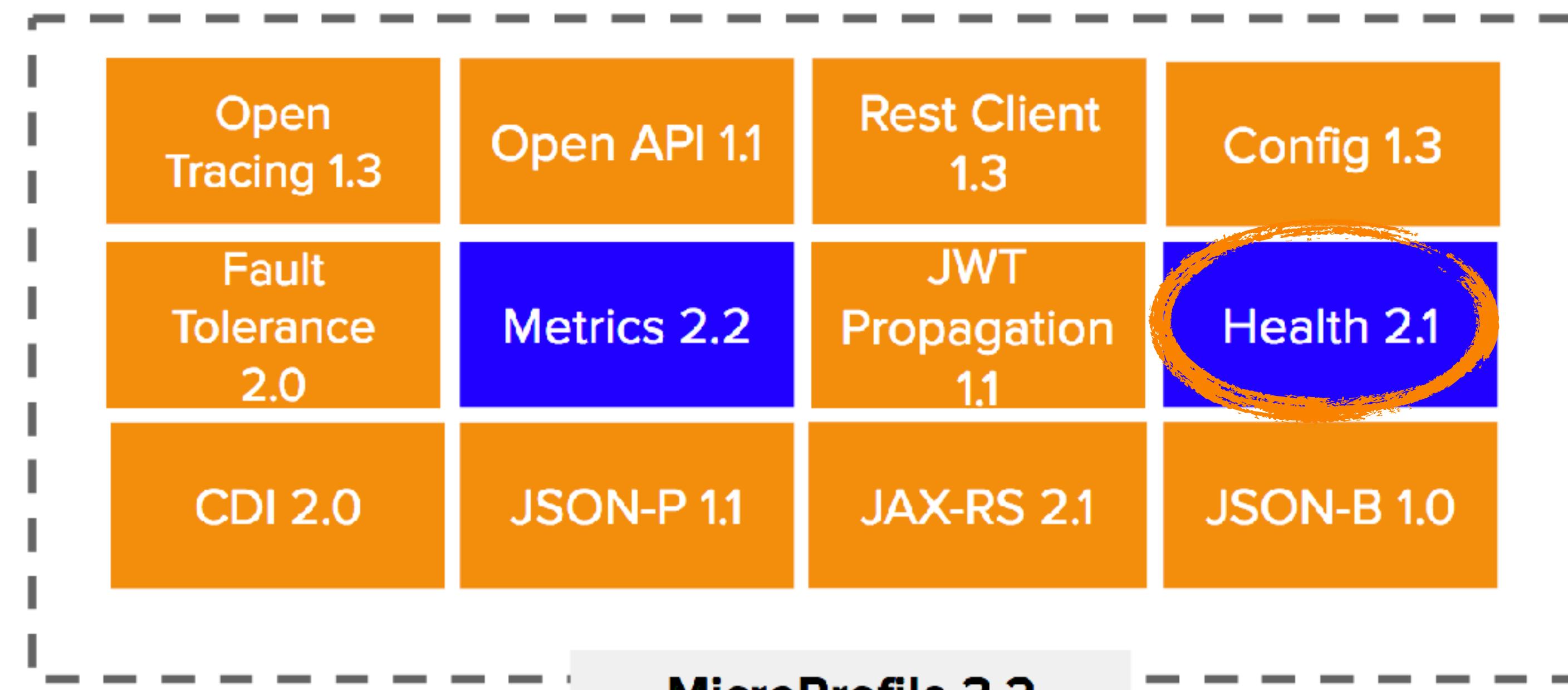
JAKARTA EE



# MicroProfile Health API



JAKARTA EE





JAKARTA EE

# DEMO



# Application Metrics



# How to understand the behavior of an application and troubleshoot problems?

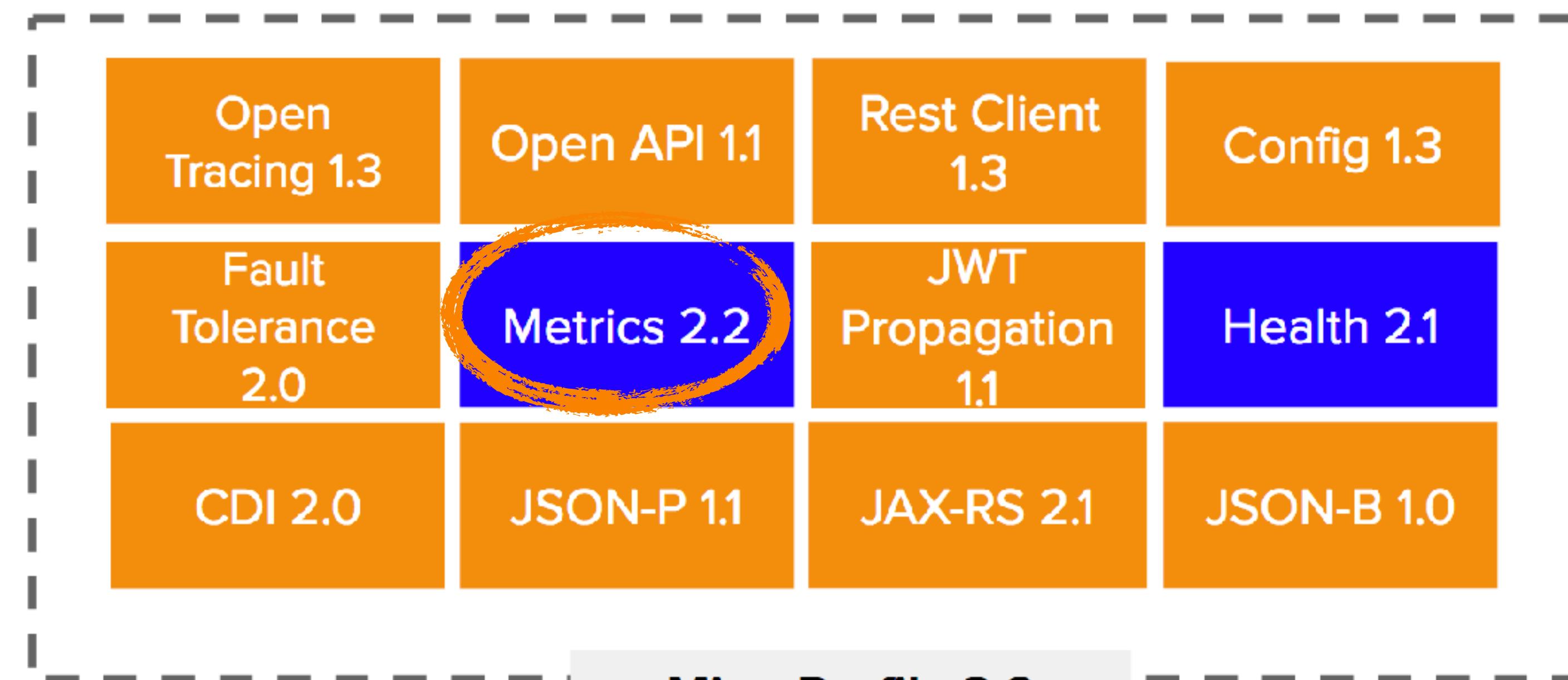


JAKARTA EE

# MicroProfile Metrics



JAKARTA EE



- = New
- = Updated
- = No change from last release (MicroProfile 3.1)



JAKARTA EE

# DEMO



JAKARTA EE

# Externalised Configuration



# How to enable a service to run in multiple environments without modification?



# Configuration for MicroProfile



JAKARTA EE

# System Properties

`System.getProperties()`

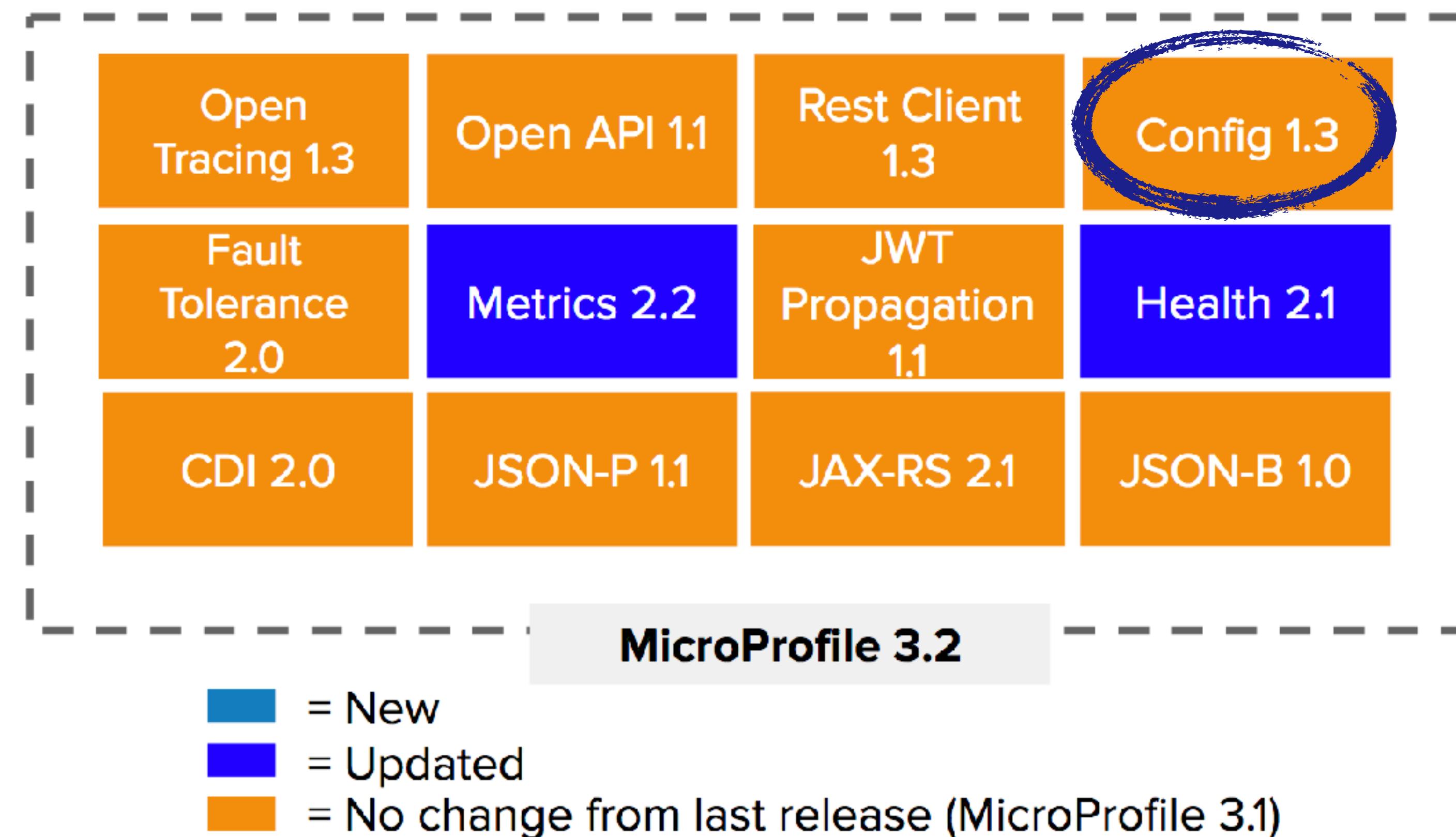
# Environment Variables

`System.getenv()`

# Application Configuration

`META-INF/micropattern-config.properties`





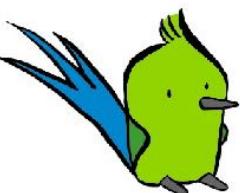


JAKARTA EE

# DEMO

**http://localhost:8484/data/location**

location



**THORNTAIL**

**Response**

in Sofia



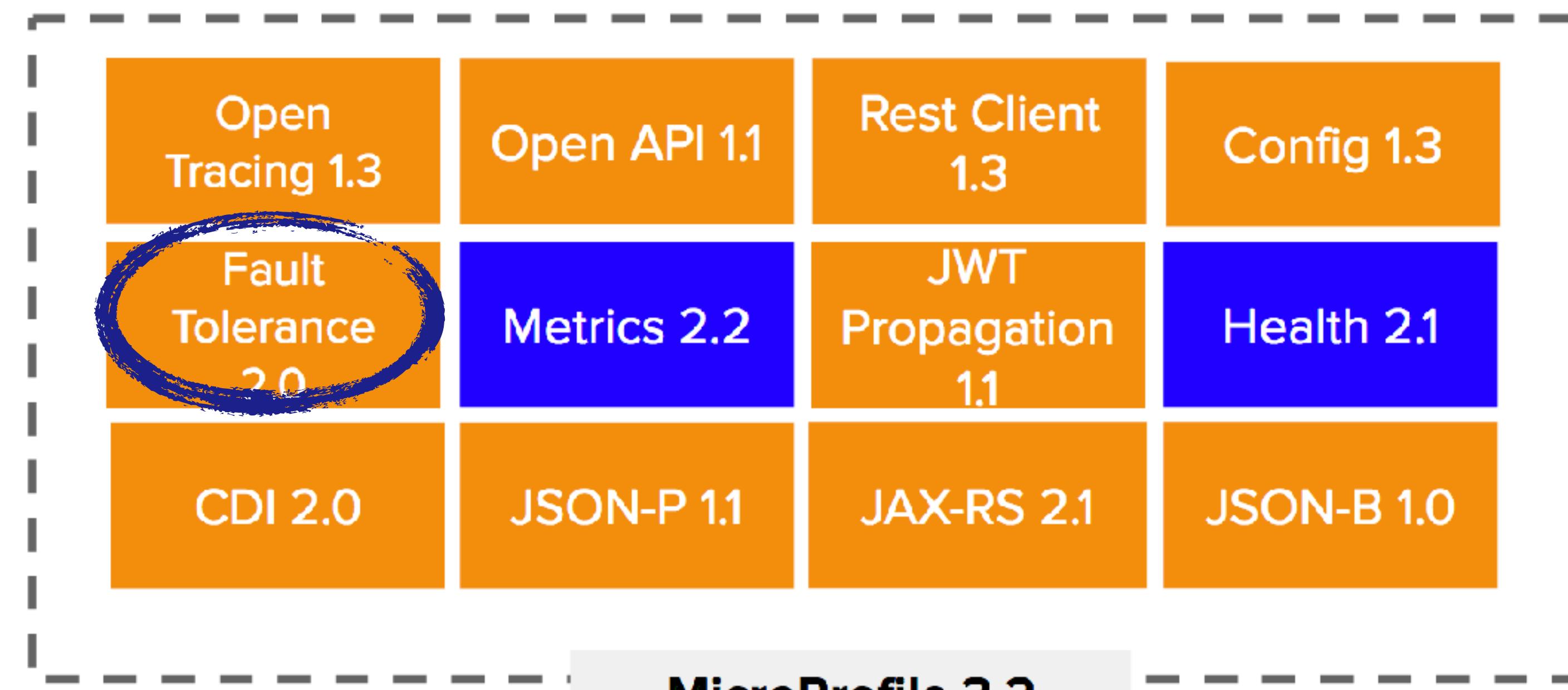
# Circuit Breaker

# How to prevent a network or service failure from cascading to other services?



JAKARTA EE





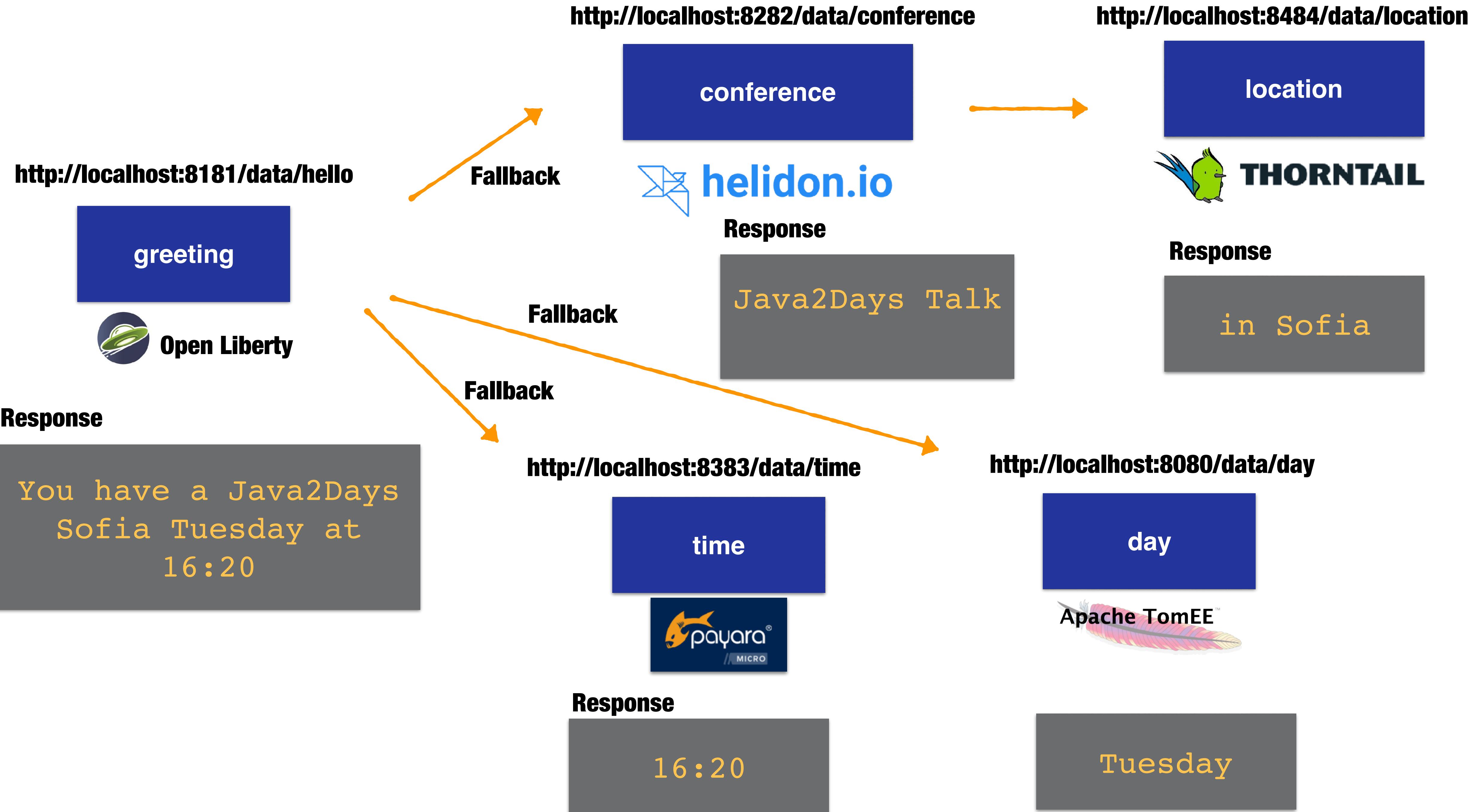
### MicroProfile 3.2

- = New
- = Updated
- = No change from last release (MicroProfile 3.1)



JAKARTA EE

# DEMO





JAKARTA EE

# Access Token

# How to communicate the identity of the requestor to the services that handle the request?



JAKARTA EE

# Basic Authentication



JAKARTA EE

POST /hello HTTP/1.1

Host: localhost:8080

**Authorization: Basic c25vb3850nBhc3M=**

User-Agent: HTTPie/0.9.9

Accept: \*/\*

Content-Type: application/json

Content-Length: 26

{ "message": "Hello, Duke!" }



JAKARTA EE

# OAuth 2.0



JAKARTA EE

POST /hello HTTP/1.1

Host: localhost:8080

**Authorization: Bearer c250b2829e3b4c20sidMwpAA**

User-Agent: HTTPie/0.9.9

Accept: \*/\*

Content-Type: application/json

Content-Length: 26

```
{"message": "Hello, Duke!"}
```



JAKARTA EE

# JSON Web Token (JWT)



JAKARTA EE

POST /hello HTTP/1.1  
Host: localhost:8080  
**Authorization: Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.eyJ0b2tlb190eXB1IjoiQmVhcmVyIiwiYXVkJioic2ltcGx1YXBwIiwic3ViIjoiYm9iIiwidXBuIjoiYm9iIiwiY3VzdG9tQ2xhaw0iOijdxN0b21WYWx1ZSIsImdyb3VwcyI6WyJhZG1pbisInVzZXIxSwiaXNzIjoiaHR0cDovL29wZW5saWJlcnR5LmlviIwiZXhwIjoxNTI1MzM5MjU5LCJpYXQiOjE1MjUyNTI4NTl9.nDDpCR2wMWsENaDGORQb1ErOCJ6ii1R-N6iSzVsAz2RoUlvm2PrST2qT6kje-Nz9vcptbSpOS1F1OuJydvP8Tg3puJKI5vGyL4RMowgJHwdvv7hMZvrK1loNrPfbyv727L9UwJSZMdzL\_sibiXDuj-bIBIr6yGjVF86aQnh-PWBV8HHM1iW811\_y3511p2CAPkfjKwLU91iIT1jo1QCwItF8Dv-zAzmhd\_rOIF58eB809\_5rm0MMGIuAxdgUiU2bdeJQCKTWJKNNaSu5oddzz1KaCX4hIgBdrVV6447DM0i31YRhLpYHfbzh\_NlRrRPjGPNgzoIN1RInwPC8jdg\_g**

User-Agent: HTTPie/0.9.9

Accept: \*/\*

Content-Type: application/json

Content-Length: 26

{ "message": "Hello, Duke!" }



JAKARTA EE

```
POST /hello HTTP/1.1
Host: localhost:8080
Authorization: Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.eyJ0b2tlb190eXB1IjoiQmVhcmVyIiwiYXVkJoic2ltcGx1YXBwIiwic3ViIjoiYm9iIiwidXBuIjoiYm9iIiwiY3VzdG9tQ2xhaw0iOijdxN0b21WYWx1ZSIsImdyb3VwcyI6WyJhZG1pbisInVZXIiXSwiaXNzIjoiaHR0cDovL29wZW5saWJlcnR5LmlvIiwiZXhwIjoxNTI1MzM5MjU5LCJpYXQiOjE1MjUyNTI4NTl9.nDDpCR2wMWsENaDGORQb1ErOCJ6ii1R-N6iSzVsAz2RoUlvm2PrST2qT6kje-Nz9vcptbSpOS1F1OuJydvP8Tg3puJKI5vGyL4RMowgJHwdvv7hMZvrK1loNrPfbyv727L9UwJSZMdzL_sibiXDuj-bIBIr6yGjVF86aQnh-PWBV8HHM1iW811_y3511p2CAPkfjKwLU91iIT1jo1QCwItF8Dv-zAzmhd_rOIF58eB809_5rm0MMGIuAxdgUiU2bdeJQCKTWJKNNaSu5oddzz1KaCX4hIgBdrVV6447DM0i31YRhLpYHfbzh_NlRrRPjGPNgzoIN1RInwPC8jdg_g
User-Agent: HTTPie/0.9.9
Accept: */*
Content-Type: application/json
Content-Length: 26

{ "message": "Hello, Duke!" }
```



JAKARTA EE

## ♦**Header**

- ♦ **JSON, Base64 URL Encoded**
- ♦ **Algorithm, Type**

## ♦**Payload**

- ♦ **JSON, Base64 URL Encoded**
- ♦ **Standard + Custom entries**

## ♦**Signature**

- ♦ **Binary, Base64 URL Encoded**
- ♦ **The actual signature**



JAKARTA EE

```
POST /hello HTTP/1.1
Host: localhost:8080
Authorization: Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.eyJ0b2tlb190eXB1IjoiQmVhcmVyIiwiYXVkJoic2ltcGx1YXBwIiwic3ViIjoiYm9iIiwidXBuIjoiYm9iIiwiY3VzdG9tQ2xhaw0iOijdxN0b21WYWx1ZSIsImdyb3VwcyI6WyJhZG1pbisInVZXIiXSwiaXNzIjoiaHR0cDovL29wZW5saWJlcnR5LmlvIiwiZXhwIjoxNTI1MzM5MjU5LCJpYXQiOjE1MjUyNTI4NTl9.nDDpCR2wMWsENaDGORQb1ErOCJ6ii1R-N6iSzVsAz2RoUlvm2PrST2qT6kje-Nz9vcptbSpOS1F1OuJydvP8Tg3puJKI5vGyL4RMowgJHwdvv7hMZvrK1loNrPfbyv727L9UwJSZMdzL_sibiXDuj-bIBIr6yGjVF86aQnh-PWBV8HHM1iW811_y3511p2CAPkfjKwLU91iIT1jo1QCwItF8Dv-zAzmhd_rOIF58eB809_5rm0MMGIuAxdgUiU2bdeJQCKTWJKNNaSu5oddzz1KaCX4hIgBdrVV6447DM0i31YRhLpYHfbzh_NlRrRPjGPNgzoIN1RInwPC8jdg_g
User-Agent: HTTPie/0.9.9
Accept: */*
Content-Type: application/json
Content-Length: 26

{ "message": "Hello, Duke!" }
```



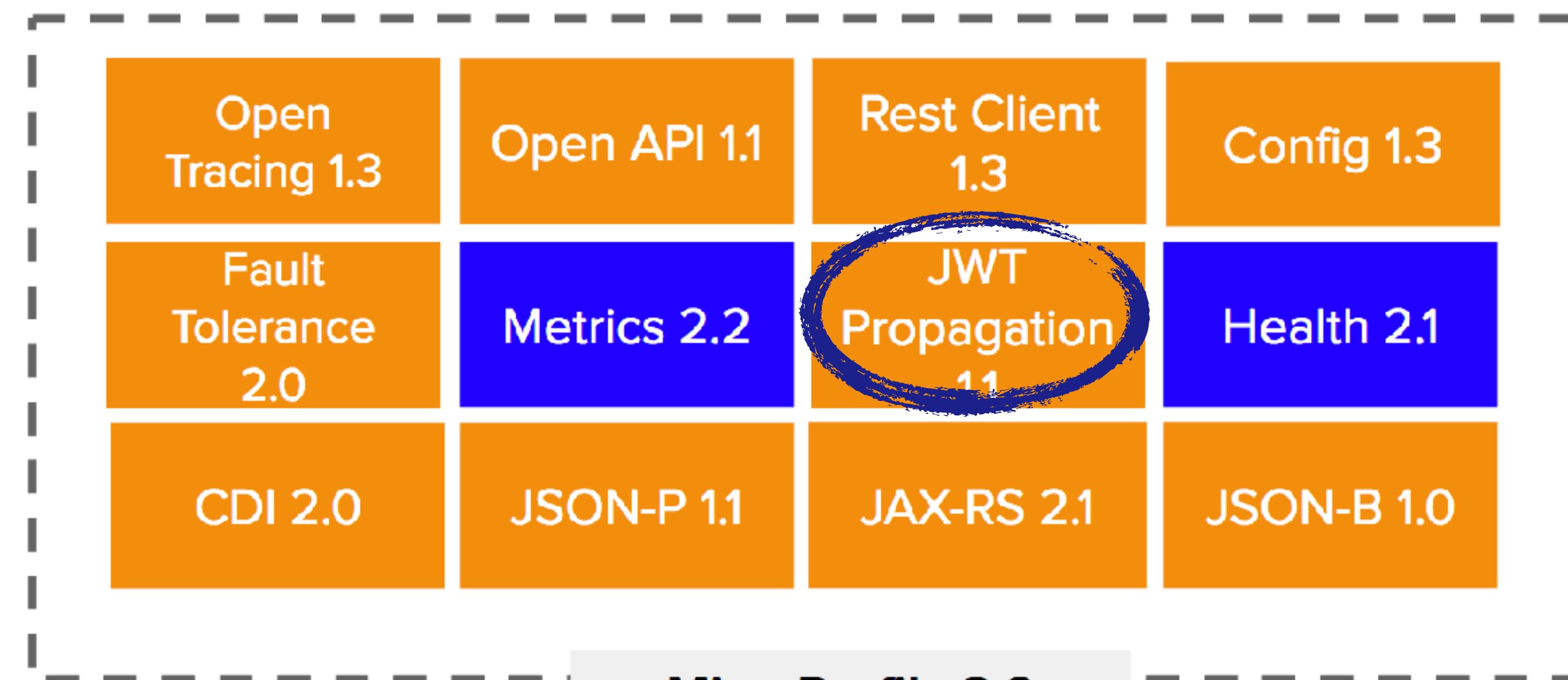
JAKARTA EE

```
{ "alg": "RS256", "typ": "JWT"}  
{  
  "token-type": "access-token",  
  "username": "duke",  
  "scopes": [  
    "java": "everywhere"  
  ],  
  "exp": 1474280963,  
  "iat": 1474279163  
  "jti": "66881b068b249ad9"  
}
```

nDDpCR2wMWsENaDGORQblErOCJ6ii1R-N6iSzVsAz2RoUlvM2PrST2qT6kje-Nz  
9vcptbSpOSlF1OuJydvP8Tg3puJKI5vGyL4RMowg-JHwdvv7hMZvrK1loNrPfby  
v727L9UwJSZMdzL\_sibiXDuj-bIBIr6yGjVF86aQnh-PWBV8HHM1iW811\_y351l  
p2CAPkfjKwLU91iT1j01QCwItF8Dv-zAzmhd\_rOIF58eB809\_5rm0MMGIuAXdg  
UiU2bdEJQCKTWJKNNaSu5oddZzlKaCX4hIgBdrVV6447DM0i31YRhLpYHfbzh\_N  
lRrRPjGPNGzOIN1RInwPC8jdg\_g



JAKARTA EE

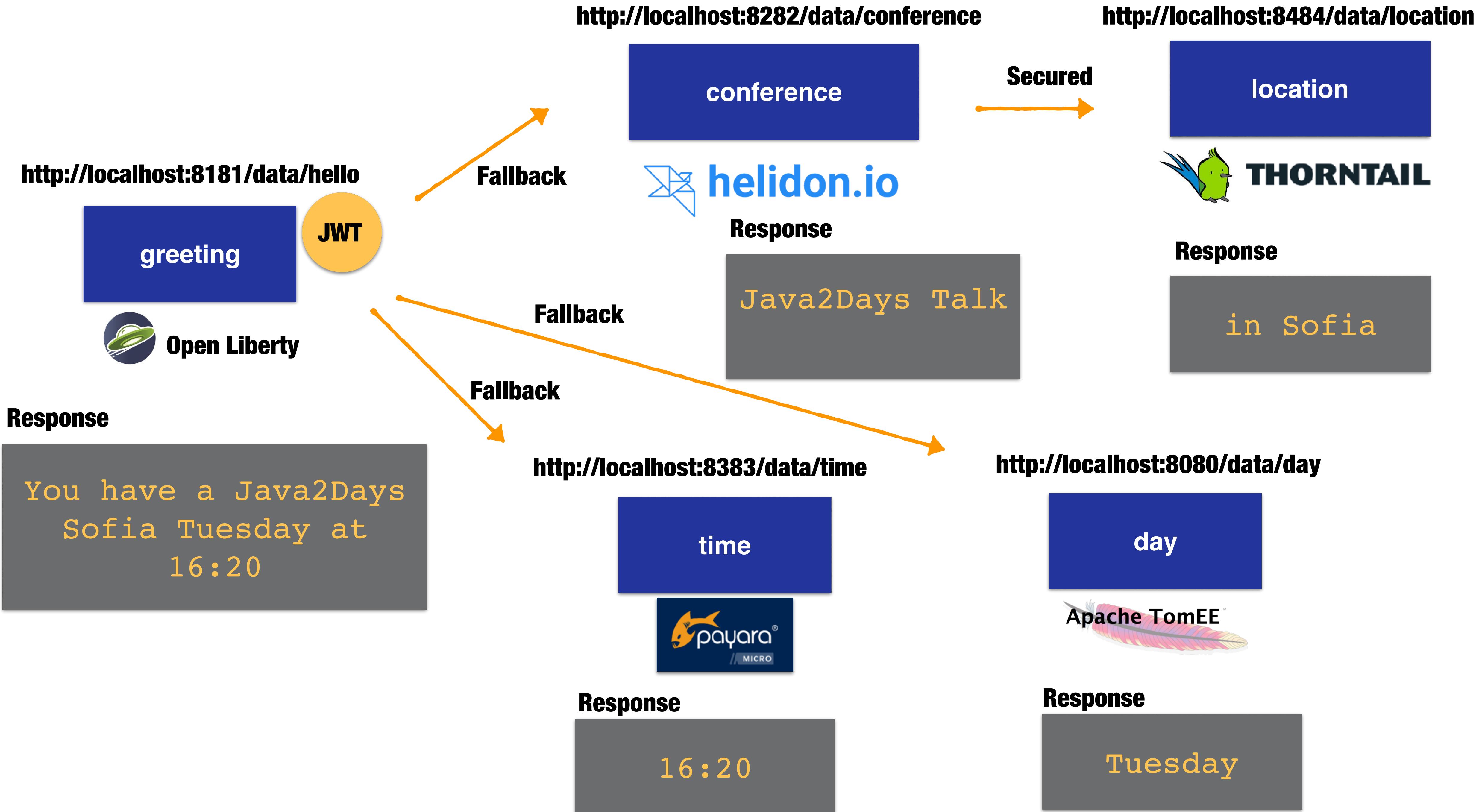


- = New
- = Updated
- = No change from last release (MicroProfile 3.1)



JAKARTA EE

# DEMO





JAKARTA EE

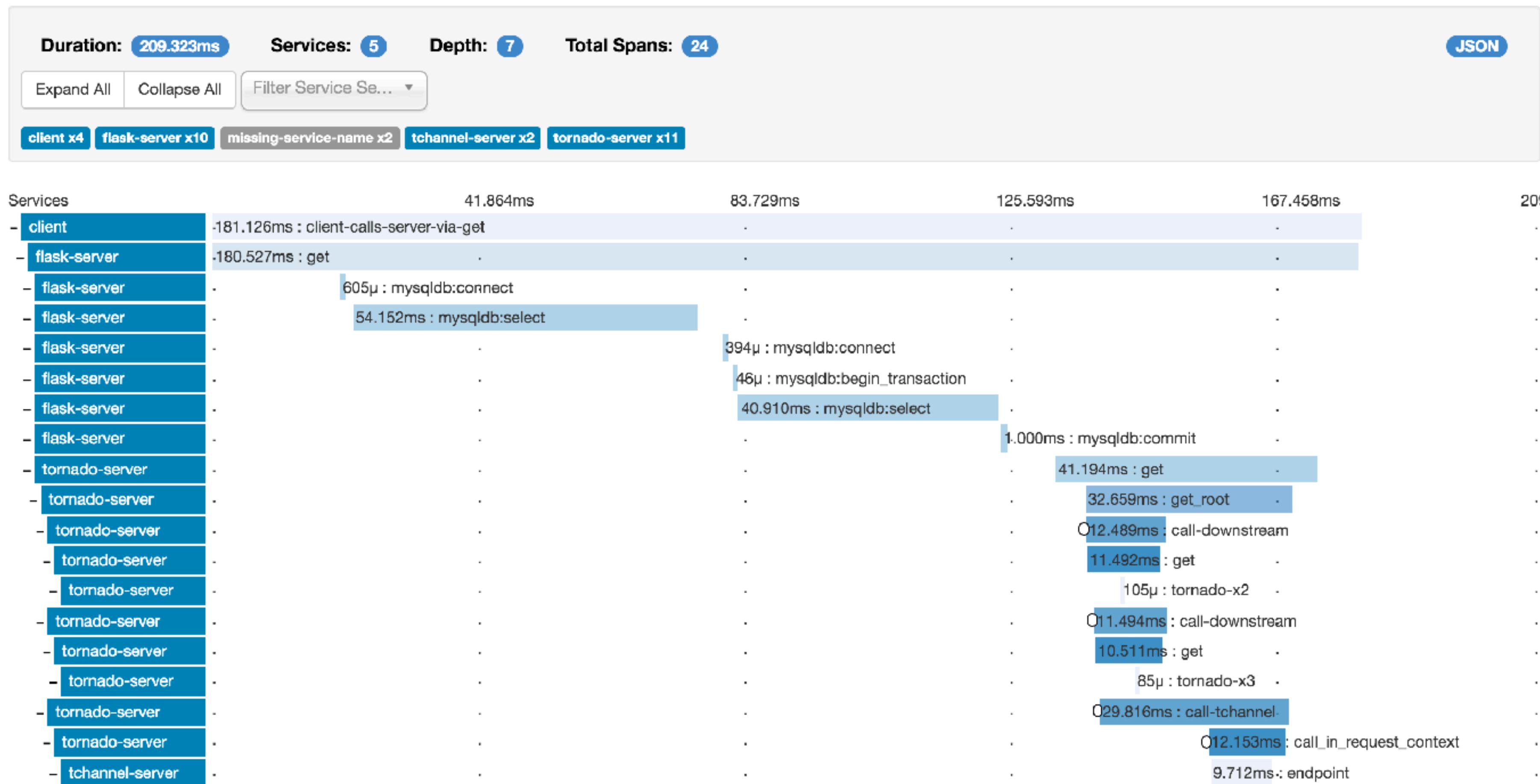
# Distributed Tracing



# How to understand the behavior of an application and troubleshoot problems?

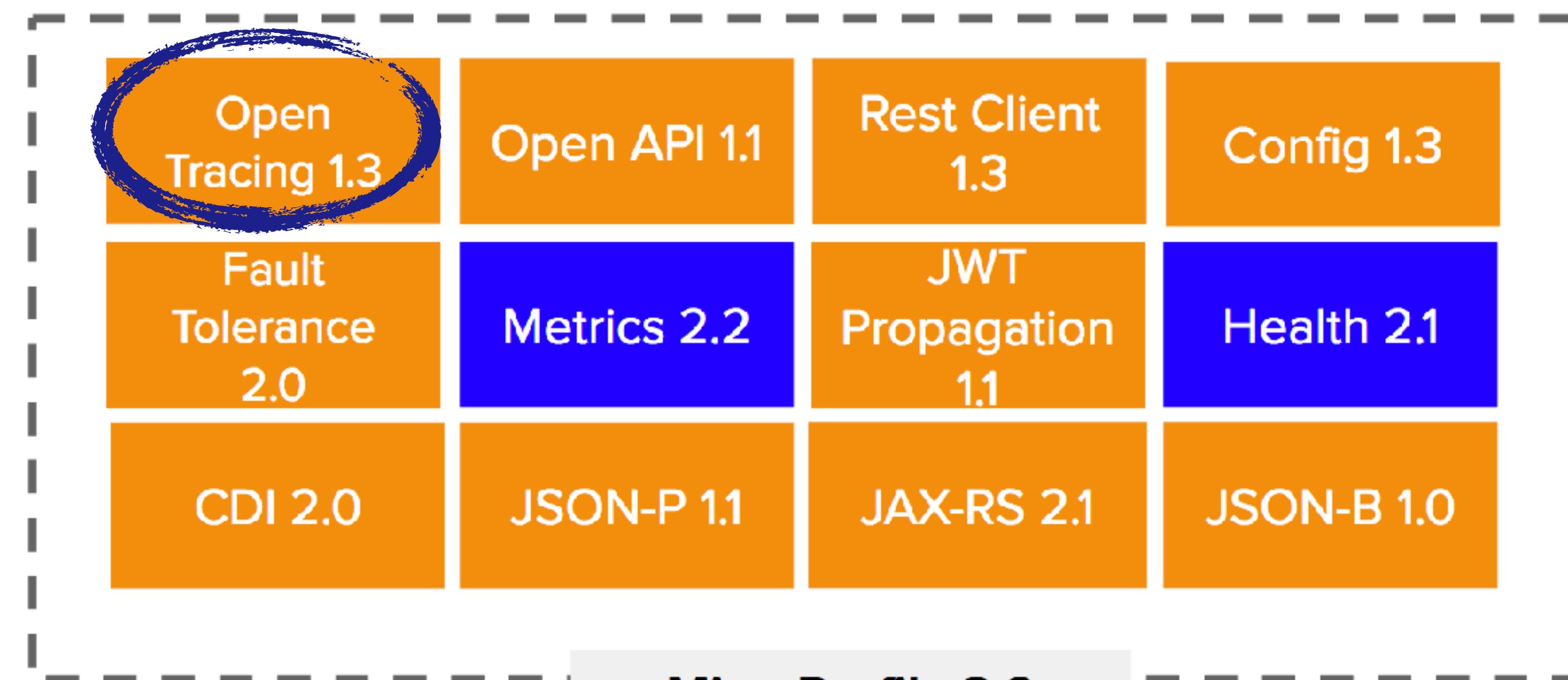


JAKARTA EE





JAKARTA EE



- = New
- = Updated
- = No change from last release (MicroProfile 3.1)



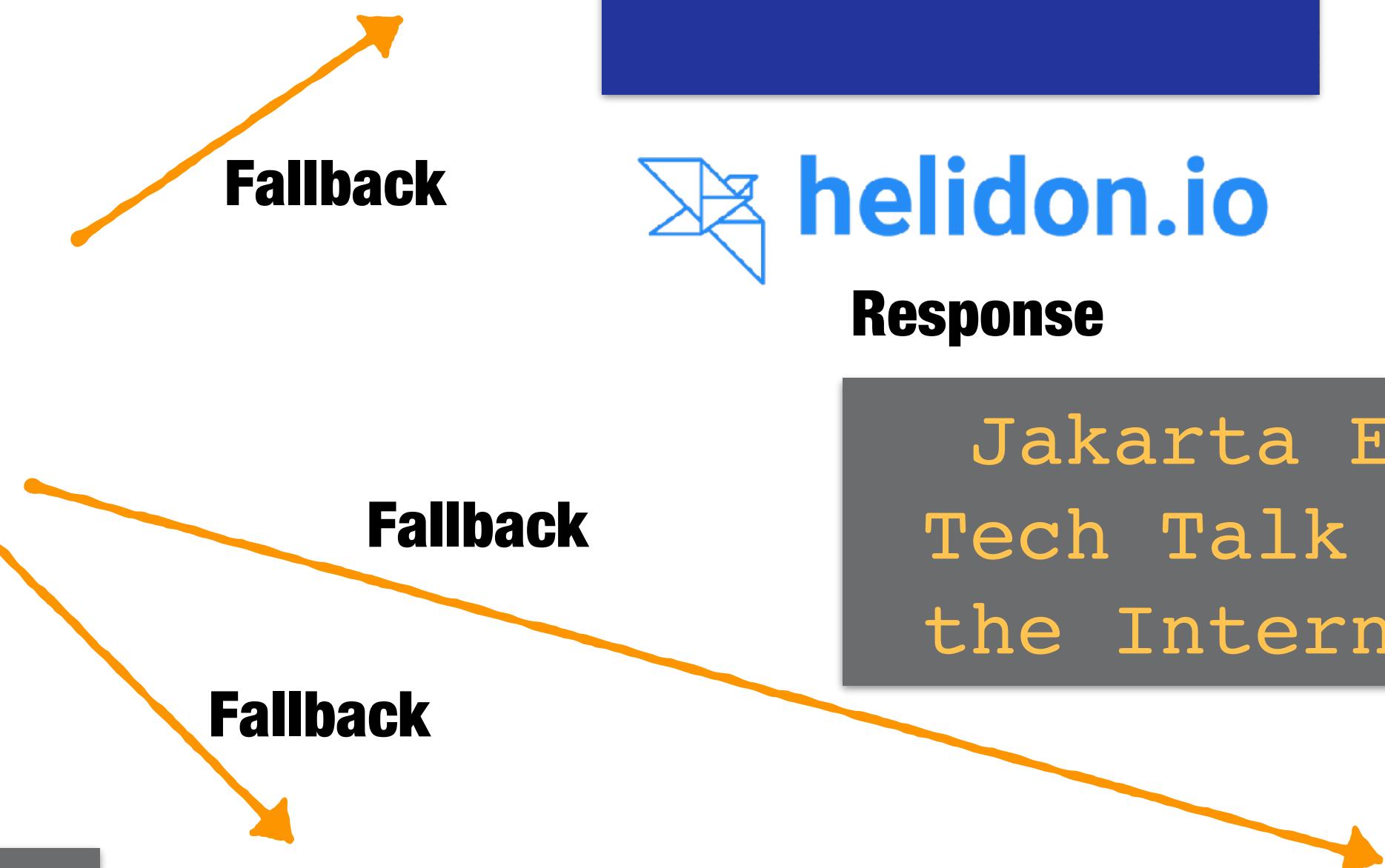
JAKARTA EE

# DEMO



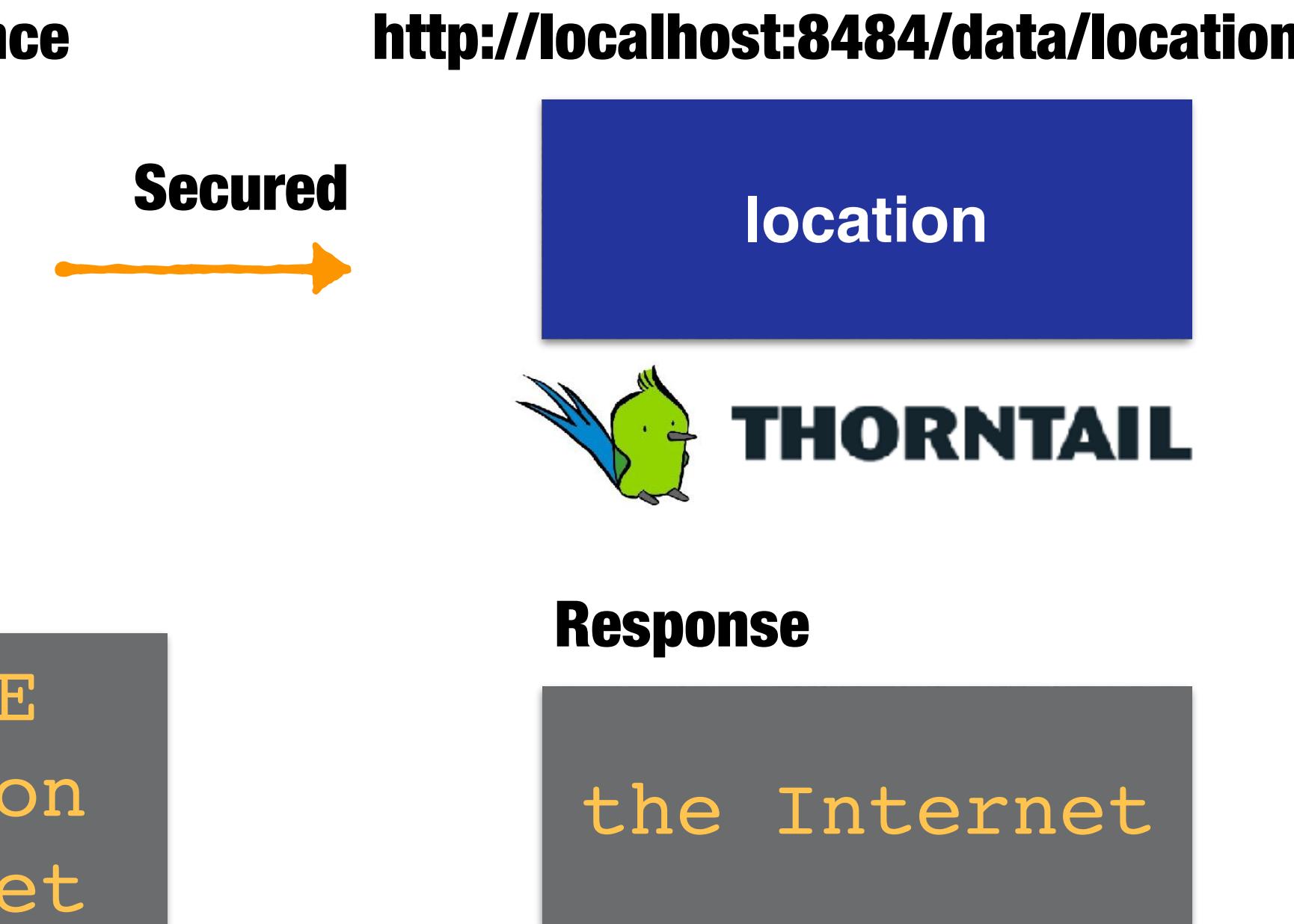
**Response**

You have a Jakarta EE  
Tech Talk on the  
Internet Wednesday at  
17:00



**Response**

17:00



**http://localhost:8080/data/day**

**day**

**Response**

Wednesday



JAKARTA EE

# API Description

# Which knowledge should be shared between an API provider and its clients? How should this knowledge be documented?



JAKARTA EE



- = New
- = Updated
- = No change from last release (MicroProfile 3.1)



JAKARTA EE

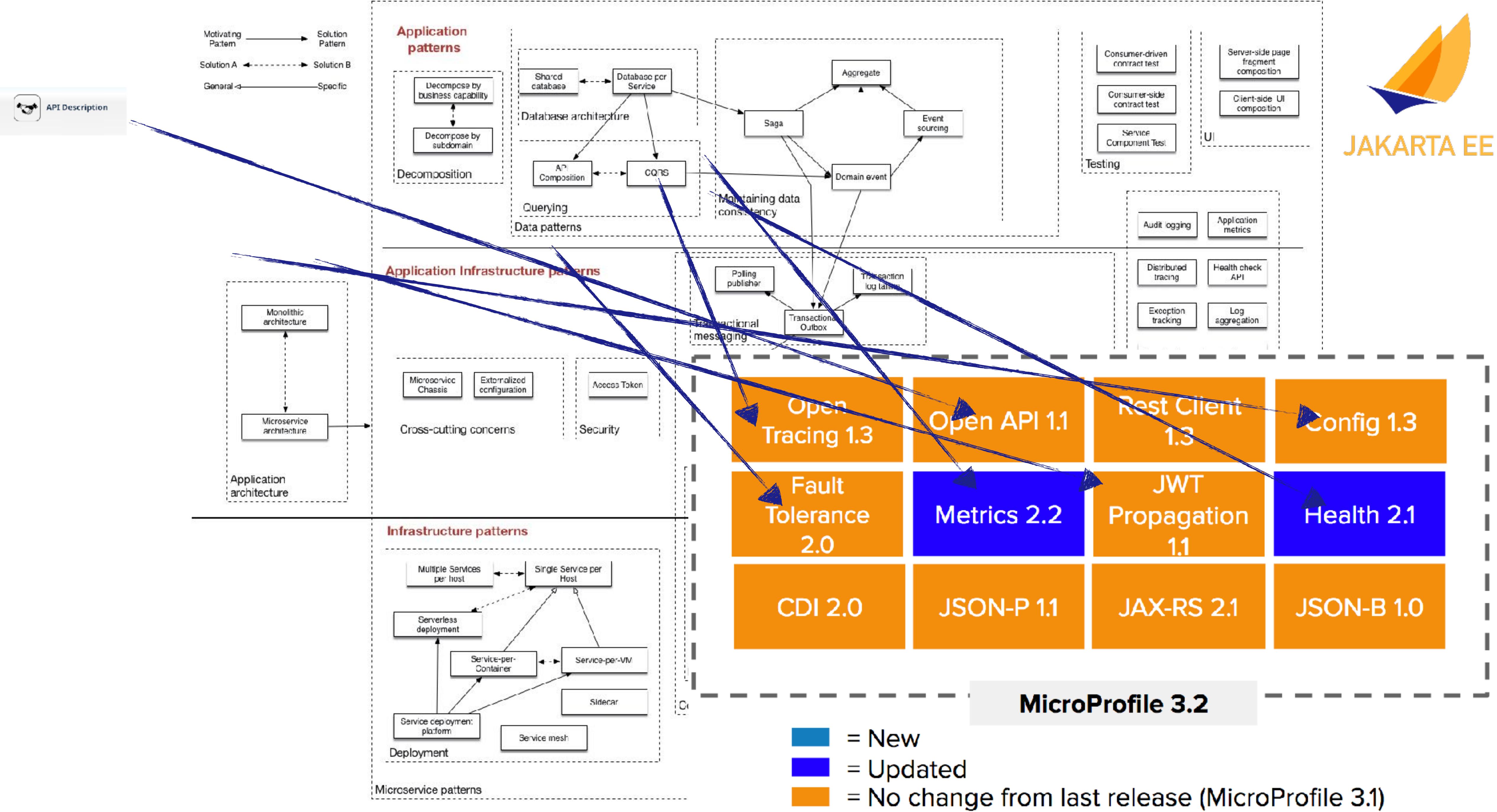
# DEMO



# Summary



JAKARTA EE





JAKARTA EE

- 1. Health Check API**
- 2. Application Metrics**
- 3. Externalised Configuration**
- 4. Circuit Breaker**
- 5. Access Token**
- 6. Distributed Tracing**
- 7. API Description**

# Microservice Patterns

<http://microservices.io>

<https://microservice-api-patterns.org/>



## Samples

<https://github.com/ivargrimstad>

## Microprofile

<http://microprofile.io>

## Jakarta EE

<https://jakarta.ee>



**Duke's Choice Award  
Winner**