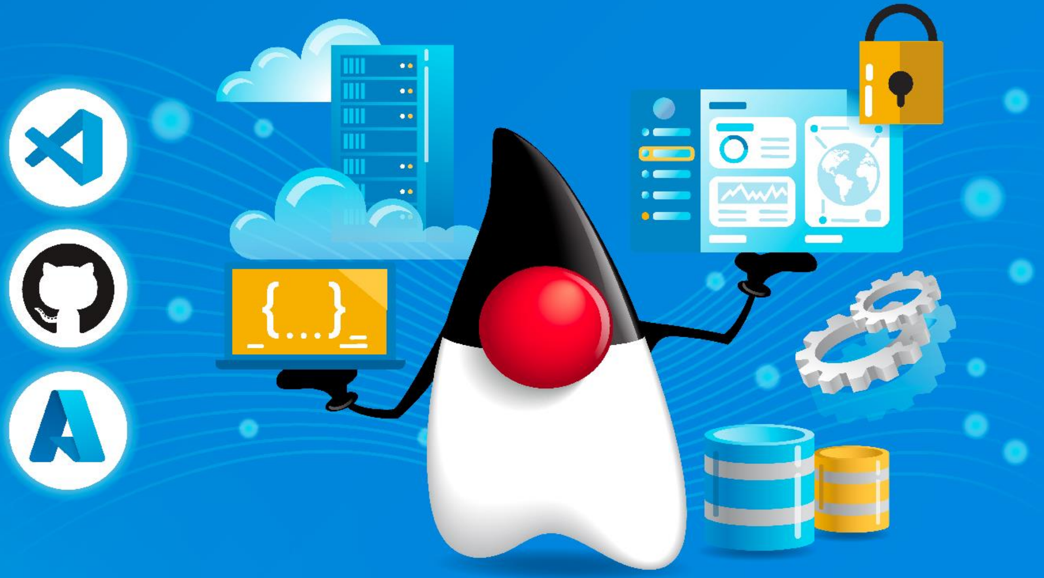
Microsoft

Hybrid Cloud Application Built with Pure Openness

Code. Cloud. Community.

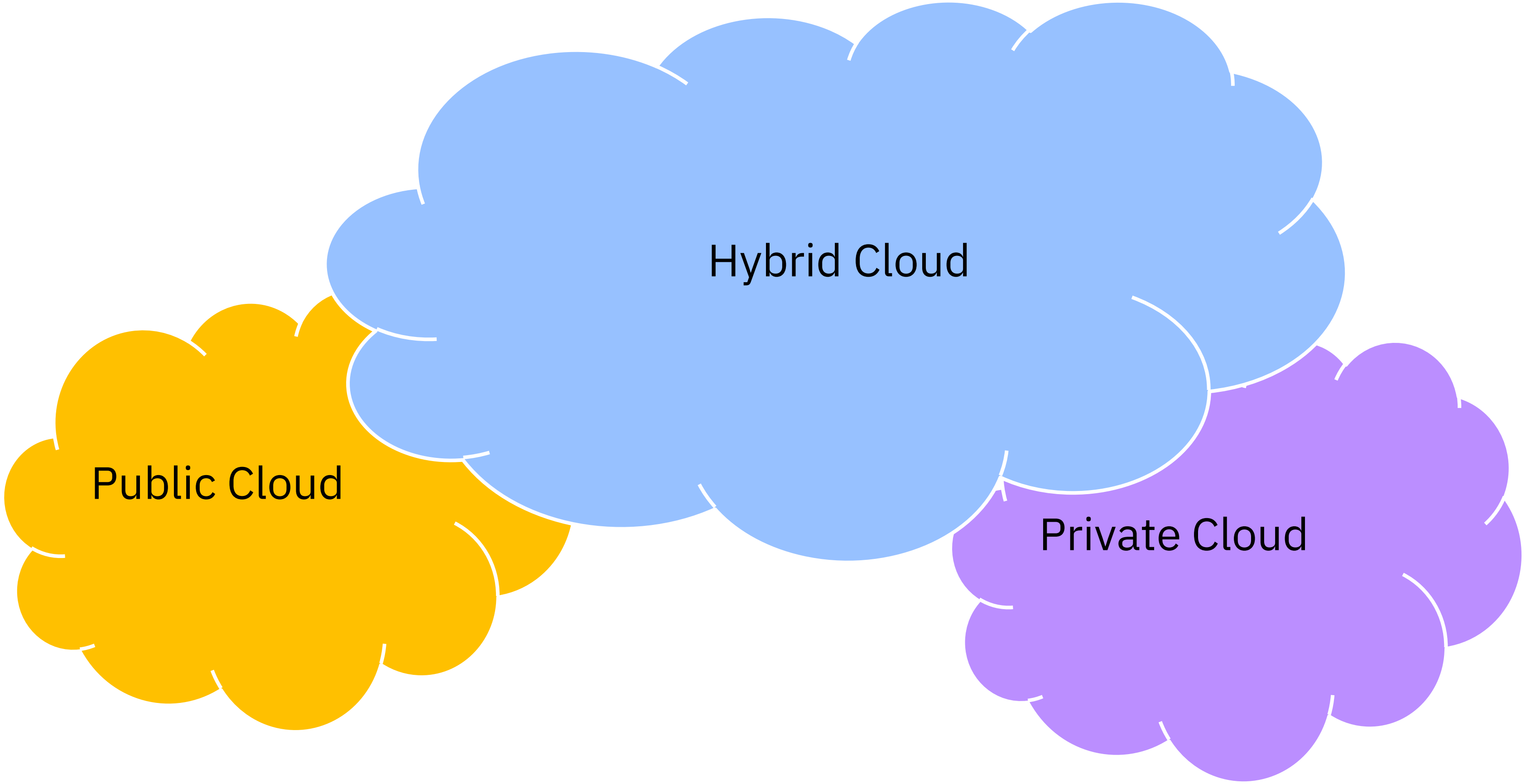
Emily Jiang, Java Champion
Cloud Native Architect and Advocate @IBM



Agenda

- Hybrid Cloud Overview
- Open Standards: Jakarta EE and MicroProfile
- Meet a modern open-source runtime: Open Liberty
  A runtime for Hybrid Cloud Applications to Serverless
- Demo

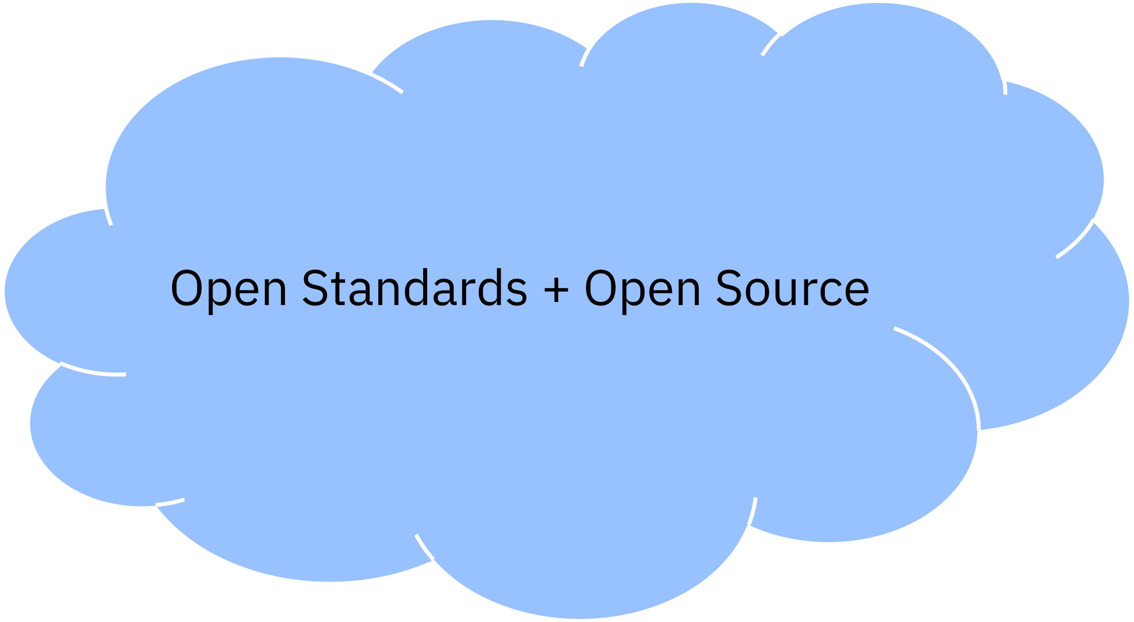
Hybrid Cloud



Hybrid Cloud Applications

Applications to provide services
(e.g. microservices)

Hybrid Cloud Applications with Openness



Open Standards + Open Source

Industry Standard Java APIs

ECLIPSE™
FOUNDATION

Build modern portable enterprise apps
Protect your investments in Java EE

JAKARTA™ EE

Optimizing Enterprise Java
for a Microservices Architecture
and Kubernetes

MICROPROFILE™

Jakarta EE
=
The future of Java EE

Opportunities for Improvement

Better support for cloud native architectures

A faster pace of innovation

Native integration with Kubernetes

Tools and specifications needed

Backed by Java industry leaders

Strategic Members

FUJITSU

IBM

ORACLE

payara

Tomitribe join the tribe

Enterprise Members

Microsoft

PRIMETON · 普元

INFORS 中创中间件®

redhat

Participating Members

BES 宝兰德 BESSYSTEM

THUNISOFT 华宇

東方通 TongTech

vsettan 华胜信泰

AsiaInfo 亚信科技

iJUG Verbund

jadeva GmbH a small step for your team – a big step for your company!

Apusic 金蝶天燕

LJC

Orchestrating a brighter world NEC

OMNI FISH

OpenElements

平安科技 PING AN TECHNOLOGY

TmaxSoft We Speak in Technology

Guest Members

SOUJava sociedade de usuários java

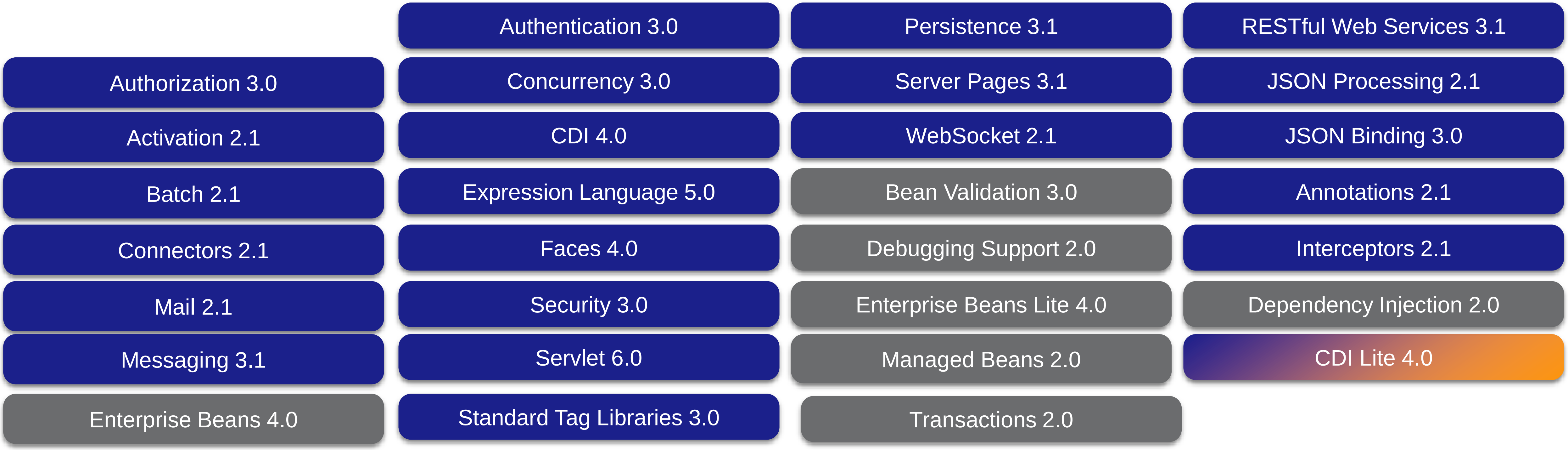
GARDEN STATE JAVA USER GROUP NEW JERSEY

ISTANBUL Java User Group

THE APACHE SOFTWARE FOUNDATION

JAKARTA EE

Jakarta EE 10 Platform



Authorization 3.0

Activation 2.1

Batch 2.1

Connectors 2.1

Mail 2.1

Messaging 3.1

Enterprise Beans 4.0

Authentication 3.0

Concurrency 3.0

CDI 4.0

Expression Language 5.0

Faces 4.0

Security 3.0

Servlet 6.0

Standard Tag Libraries 3.0

Persistence 3.1

Server Pages 3.1

WebSocket 2.1

Bean Validation 3.0

Debugging Support 2.0

Enterprise Beans Lite 4.0

Managed Beans 2.0

Transactions 2.0

RESTful Web Services 3.1

JSON Processing 2.1

JSON Binding 3.0

Annotations 2.1

Interceptors 2.1

Dependency Injection 2.0

CDI Lite 4.0

Updated

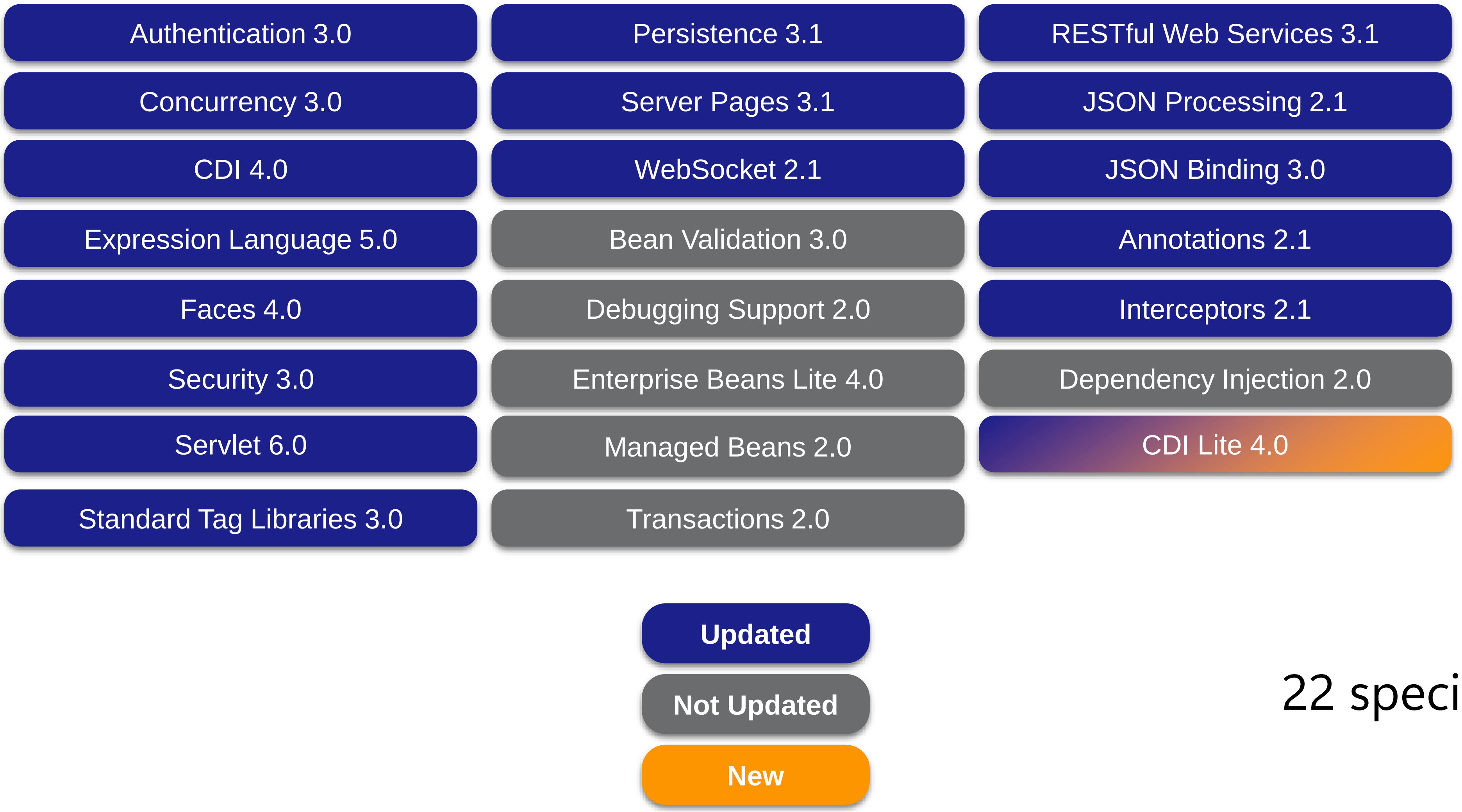
Not Updated

New

Released on 22nd September 2022

30 specifications

Jakarta EE 10 Web Profile



Authentication 3.0

Concurrency 3.0

CDI 4.0

Expression Language 5.0

Faces 4.0

Security 3.0

Servlet 6.0

Standard Tag Libraries 3.0

Persistence 3.1

Server Pages 3.1

WebSocket 2.1

Bean Validation 3.0

Debugging Support 2.0

Enterprise Beans Lite 4.0

Managed Beans 2.0

Transactions 2.0

RESTful Web Services 3.1

JSON Processing 2.1

JSON Binding 3.0

Annotations 2.1

Interceptors 2.1

Dependency Injection 2.0

CDI Lite 4.0

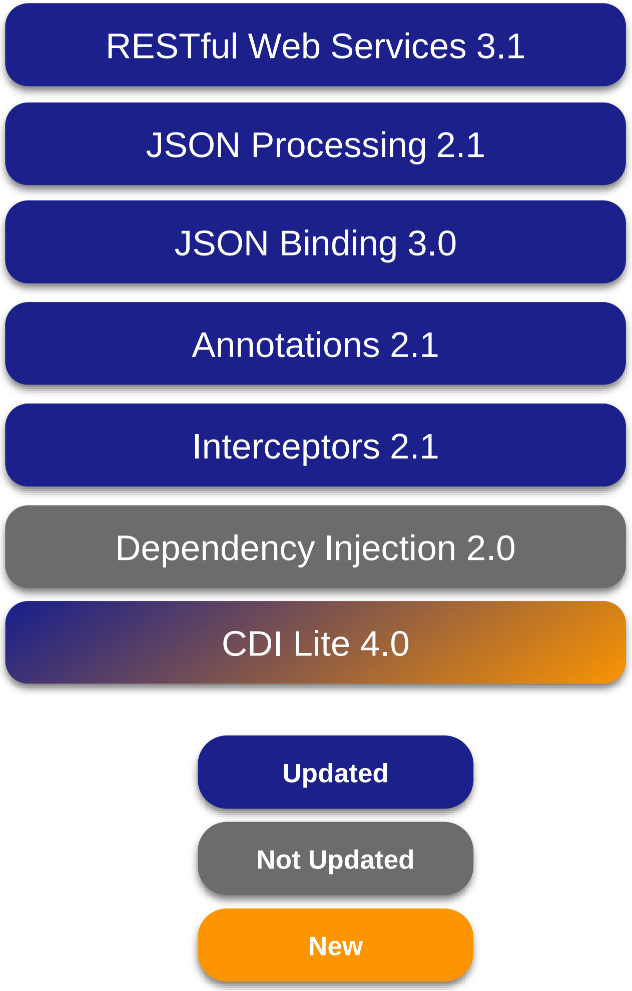
Updated

Not Updated

New

22 specifications

Jakarta EE 10 Core Profile



RESTful Web Services 3.1

JSON Processing 2.1

JSON Binding 3.0

Annotations 2.1

Interceptors 2.1

Dependency Injection 2.0

CDI Lite 4.0

Updated

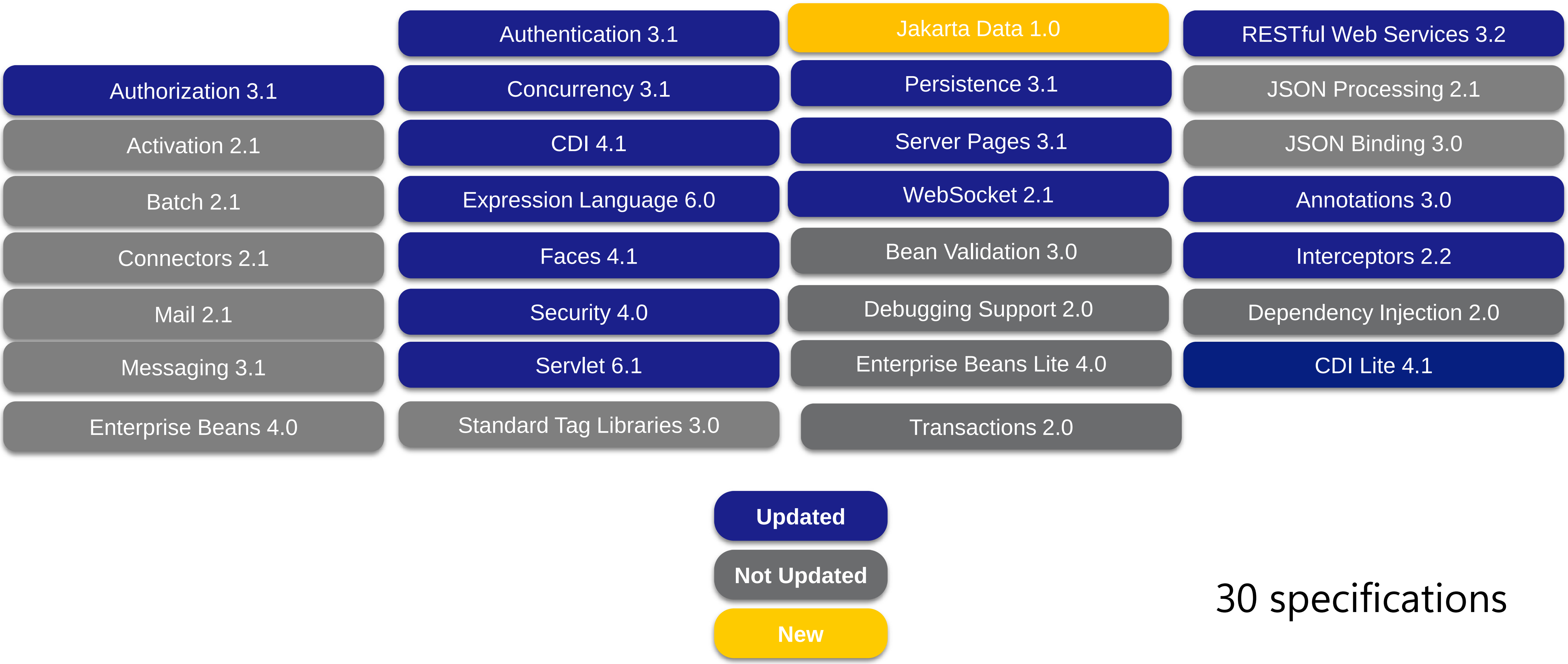
Not Updated

New

7 specifications

Jakarta EE 11 to be released in 2024

Jakarta EE 11 Platform



Authorization 3.1
Activation 2.1
Batch 2.1
Connectors 2.1
Mail 2.1
Messaging 3.1
Enterprise Beans 4.0

Authentication 3.1
Concurrency 3.1
CDI 4.1
Expression Language 6.0
Faces 4.1
Security 4.0
Servlet 6.1
Standard Tag Libraries 3.0

Jakarta Data 1.0
Persistence 3.1
Server Pages 3.1
WebSocket 2.1
Bean Validation 3.0
Debugging Support 2.0
Enterprise Beans Lite 4.0
Transactions 2.0

RESTful Web Services 3.2
JSON Processing 2.1
JSON Binding 3.0
Annotations 3.0
Interceptors 2.2
Dependency Injection 2.0
CDI Lite 4.1

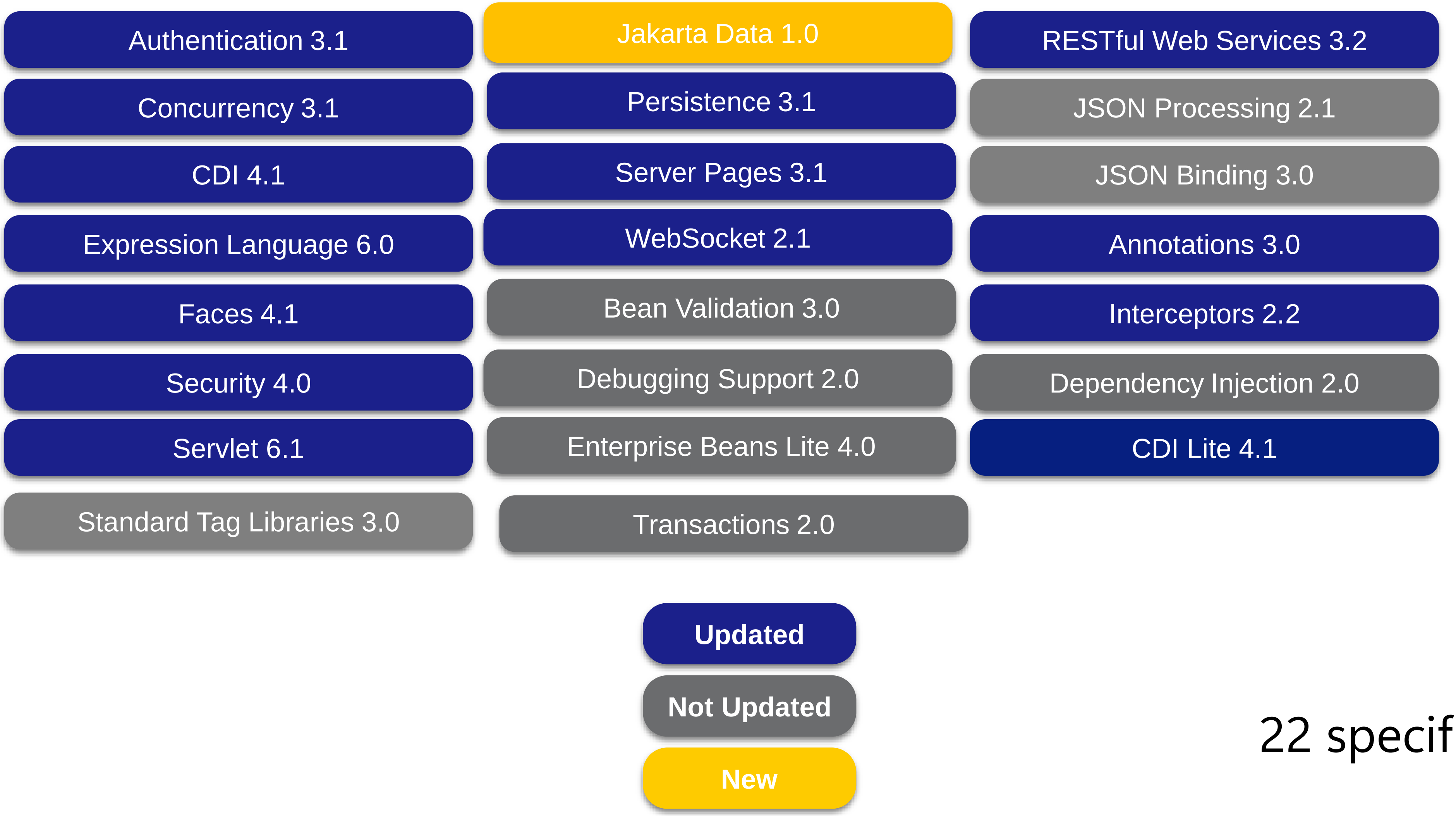
Updated

Not Updated

New

30 specifications

Jakarta EE 11 Web Profile



Authentication 3.1

Concurrency 3.1

CDI 4.1

Expression Language 6.0

Faces 4.1

Security 4.0

Servlet 6.1

Standard Tag Libraries 3.0

Jakarta Data 1.0

Persistence 3.1

Server Pages 3.1

WebSocket 2.1

Bean Validation 3.0

Debugging Support 2.0

Enterprise Beans Lite 4.0

Transactions 2.0

RESTful Web Services 3.2

JSON Processing 2.1

JSON Binding 3.0

Annotations 3.0

Interceptors 2.2

Dependency Injection 2.0

CDI Lite 4.1

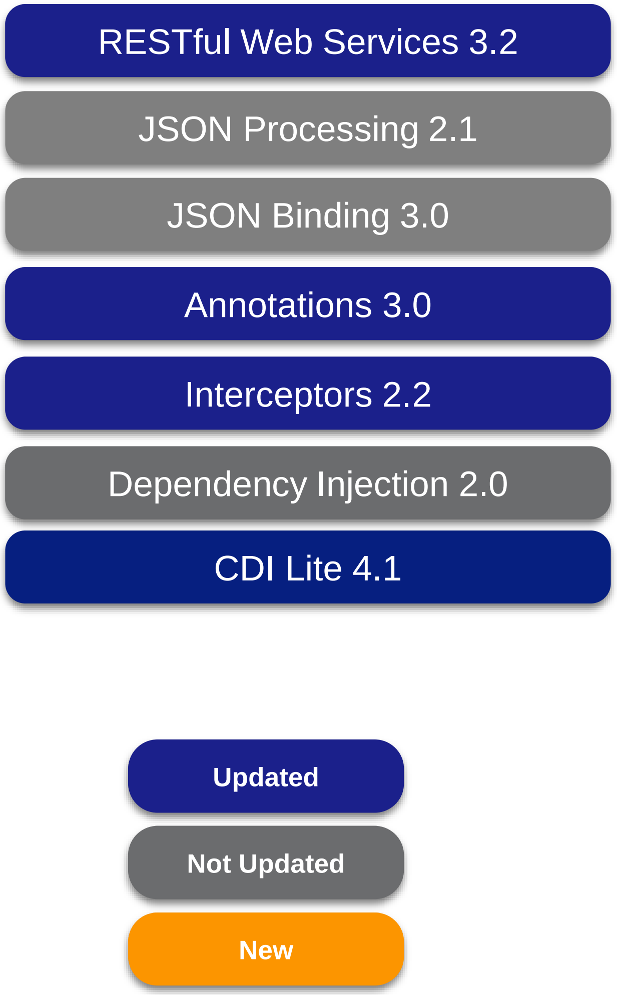
Updated

Not Updated

New

22 specifications

Jakarta EE 11 Core Profile



RESTful Web Services 3.2

JSON Processing 2.1

JSON Binding 3.0

Annotations 3.0

Interceptors 2.2

Dependency Injection 2.0

CDI Lite 4.1

Updated

Not Updated

New

7 specifications

MicroProfile

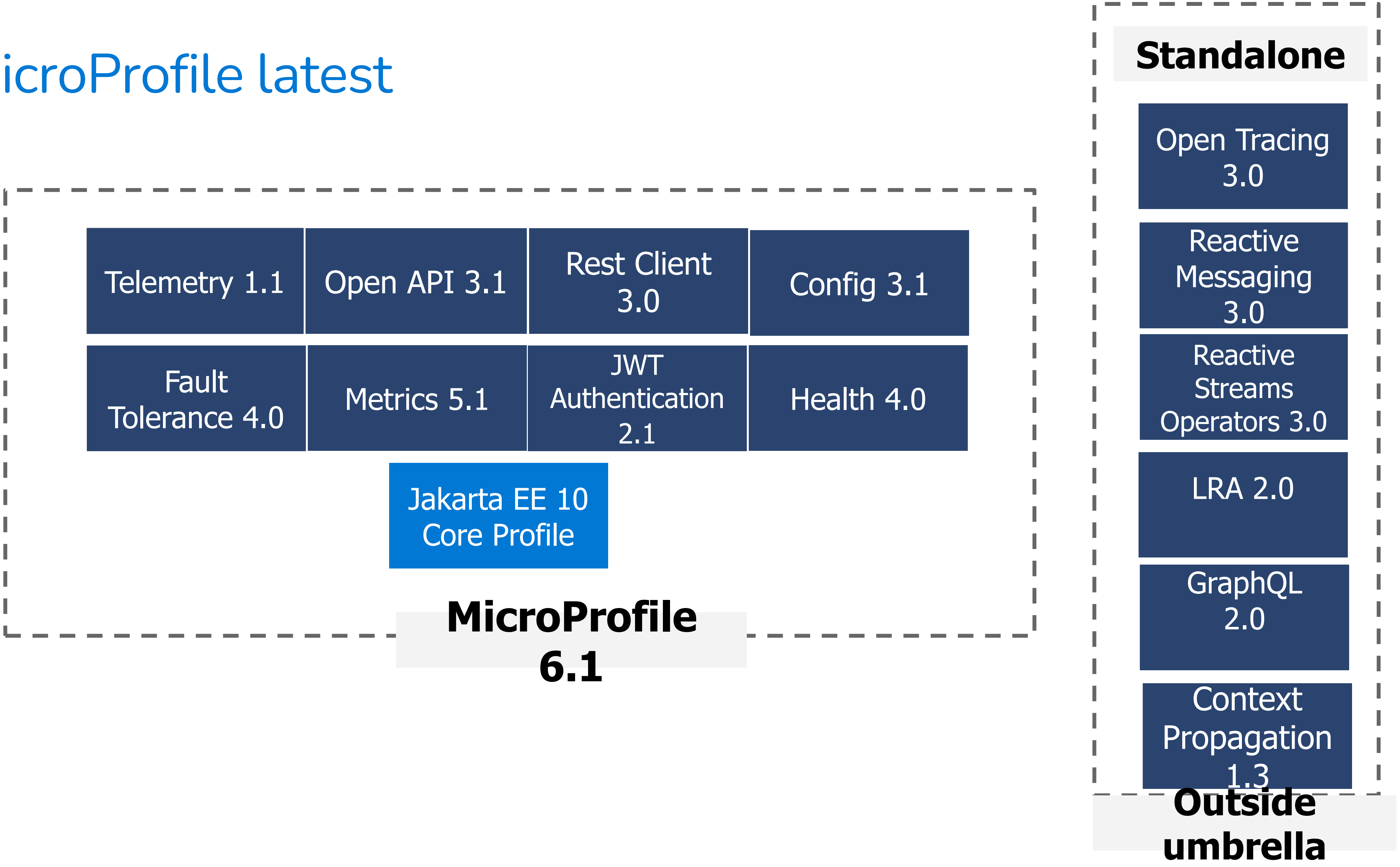
MicroProfile – Necessary Disruption

Java EE progressed very slowly, and Oracle almost stopped investing in Java EE. In the meanwhile, microservice architecture became more and more popular.

In response to the market need, a few vendors including IBM, Red Hat, Tomitribe, Payara, and others came together to set up MicroProfile in June 2016.

In Jan 2017, it was moved under Eclipse Foundation.
MicroProfile directly provoked the contribution of Java EE to Eclipse Foundation

MicroProfile latest



Telemetry 1.1

Open API 3.1

Rest Client 3.0

Config 3.1

Fault Tolerance 4.0

Metrics 5.1

JWT Authentication 2.1

Health 4.0

Jakarta EE 10 Core Profile

MicroProfile 6.1

Standalone

Open Tracing 3.0

Reactive Messaging 3.0

Reactive Streams Operators 3.0

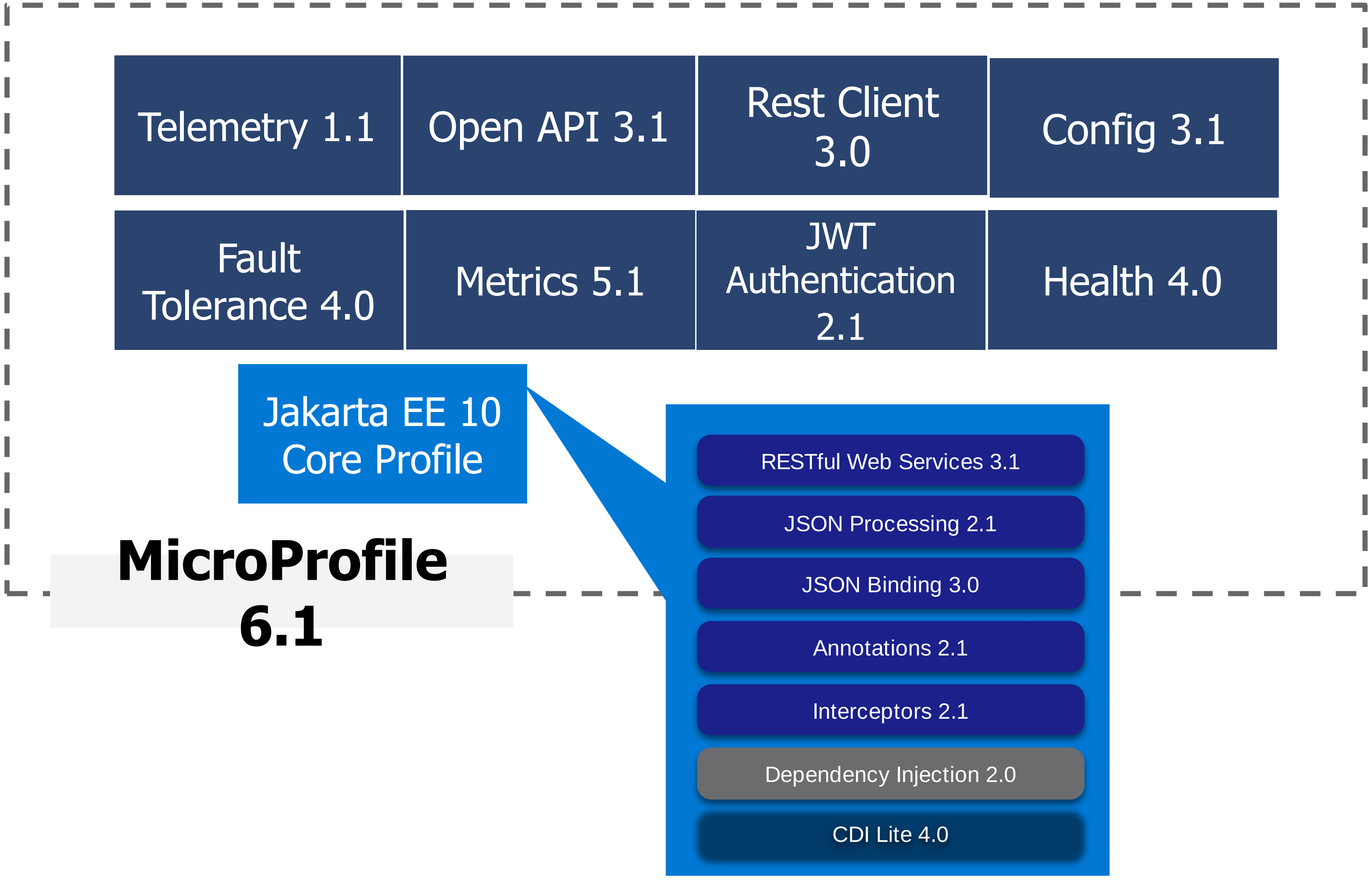
LRA 2.0

GraphQL 2.0

Context Propagation 1.3

Outside umbrella

MicroProfile latest



Telemetry 1.1

Open API 3.1

Rest Client 3.0

Config 3.1

Fault Tolerance 4.0

Metrics 5.1

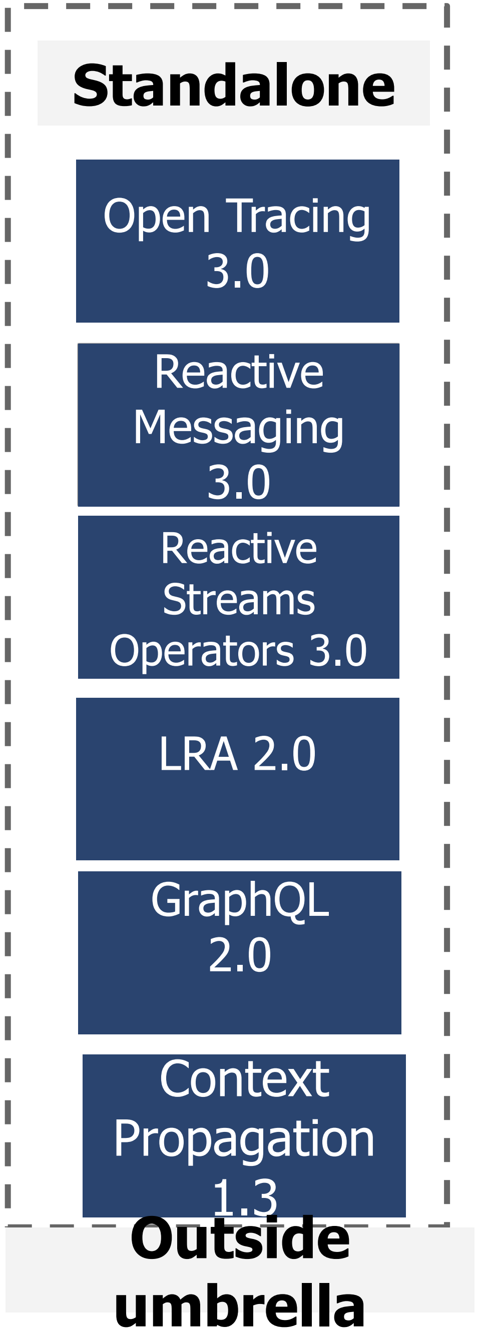
JWT Authentication 2.1

Health 4.0

Jakarta EE 10 Core Profile

- RESTful Web Services 3.1
- JSON Processing 2.1
- JSON Binding 3.0
- Annotations 2.1
- Interceptors 2.1
- Dependency Injection 2.0
- CDI Lite 4.0

MicroProfile 6.1



Standalone

Open Tracing 3.0

Reactive Messaging 3.0

Reactive Streams Operators 3.0

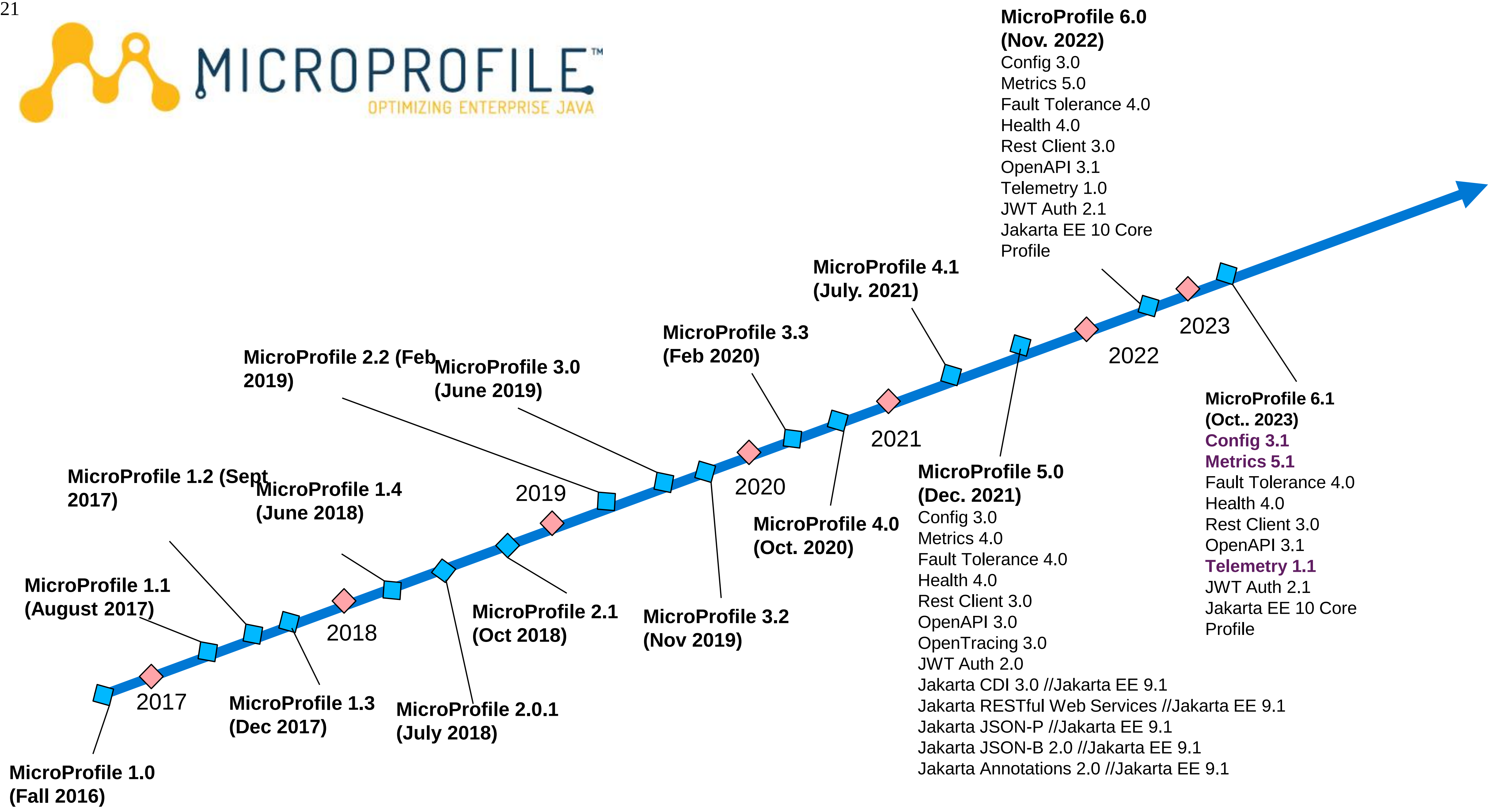
LRA 2.0

GraphQL 2.0

Context Propagation 1.3

Outside umbrella

MICROPROFILE™
OPTIMIZING ENTERPRISE JAVA



MicroProfile 1.0 (Fall 2016)

MicroProfile 1.1 (August 2017)

MicroProfile 1.2 (Sept 2017)

MicroProfile 1.3 (Dec 2017)

MicroProfile 1.4 (June 2018)

MicroProfile 2.0.1 (July 2018)

MicroProfile 2.1 (Oct 2018)

MicroProfile 2.2 (Feb 2019)

MicroProfile 3.0 (June 2019)

MicroProfile 3.2 (Nov 2019)

MicroProfile 3.3 (Feb 2020)

MicroProfile 4.0 (Oct. 2020)

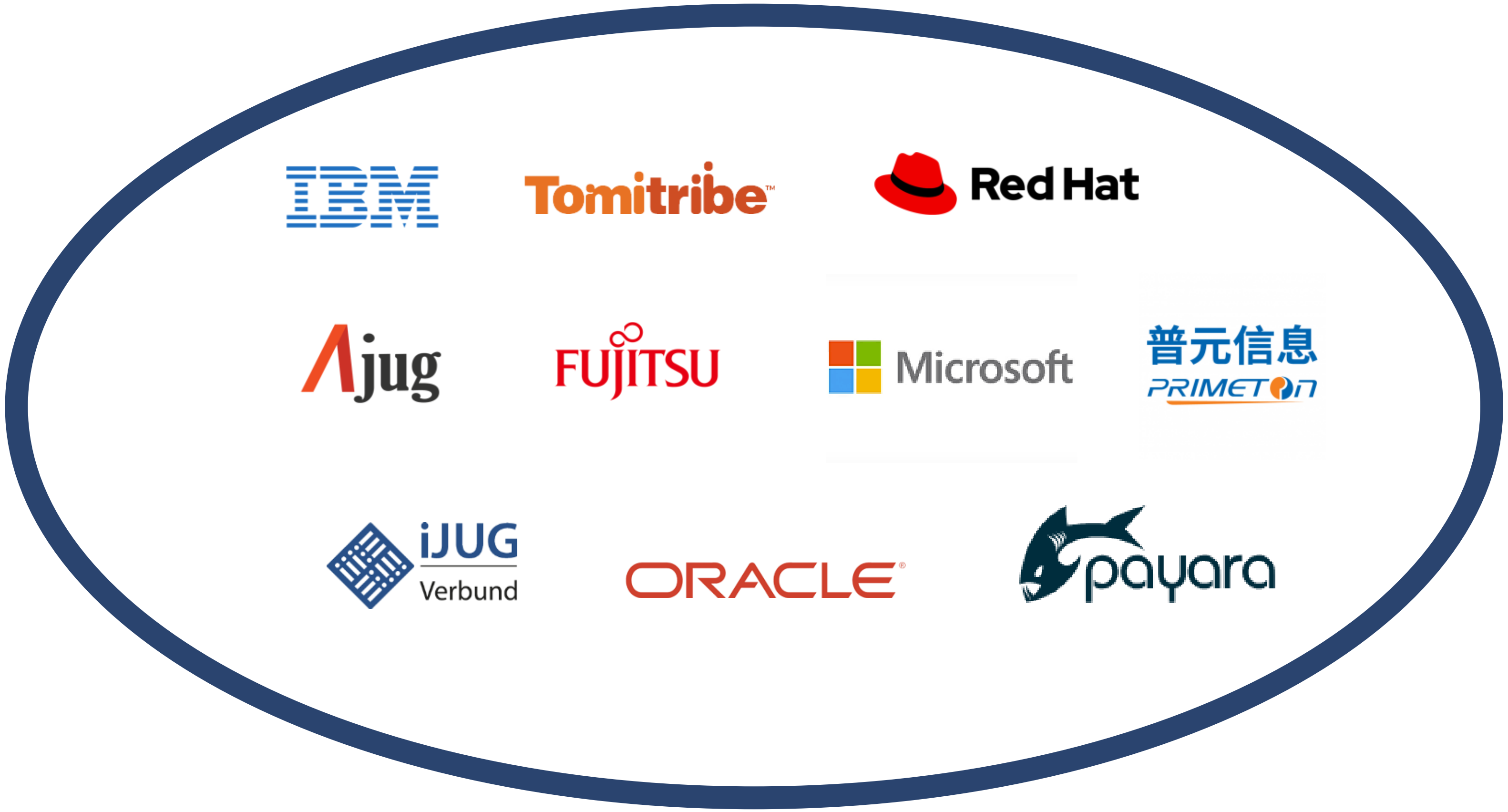
MicroProfile 4.1 (July. 2021)

MicroProfile 5.0 (Dec. 2021)
Config 3.0
Metrics 4.0
Fault Tolerance 4.0
Health 4.0
Rest Client 3.0
OpenAPI 3.0
OpenTracing 3.0
JWT Auth 2.0
Jakarta CDI 3.0 //Jakarta EE 9.1
Jakarta RESTful Web Services //Jakarta EE 9.1
Jakarta JSON-P //Jakarta EE 9.1
Jakarta JSON-B 2.0 //Jakarta EE 9.1
Jakarta Annotations 2.0 //Jakarta EE 9.1

MicroProfile 6.0 (Nov. 2022)
Config 3.0
Metrics 5.0
Fault Tolerance 4.0
Health 4.0
Rest Client 3.0
OpenAPI 3.1
Telemetry 1.0
JWT Auth 2.1
Jakarta EE 10 Core Profile

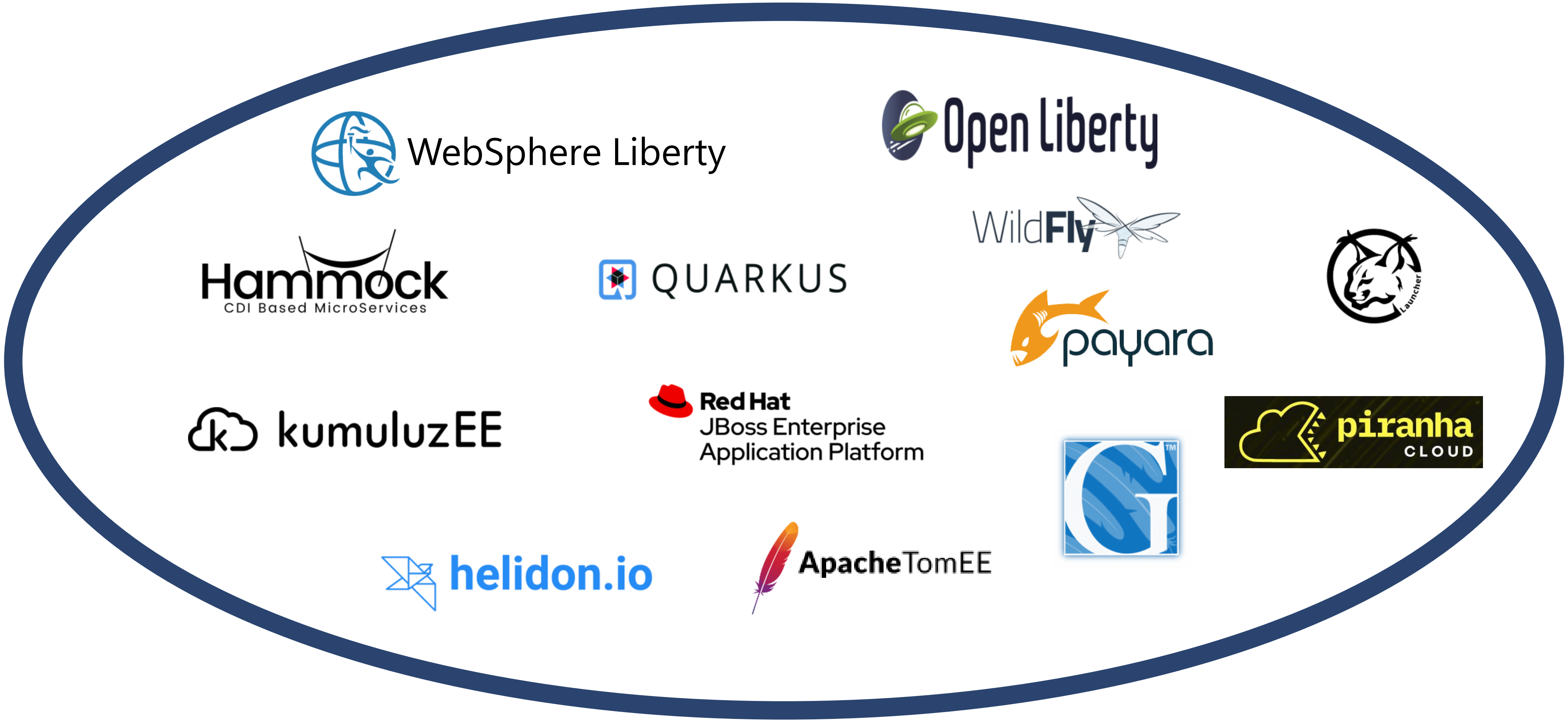
MicroProfile 6.1 (Oct.. 2023)
Config 3.1
Metrics 5.1
Fault Tolerance 4.0
Health 4.0
Rest Client 3.0
OpenAPI 3.1
Telemetry 1.1
JWT Auth 2.1
Jakarta EE 10 Core Profile

Working Group Members

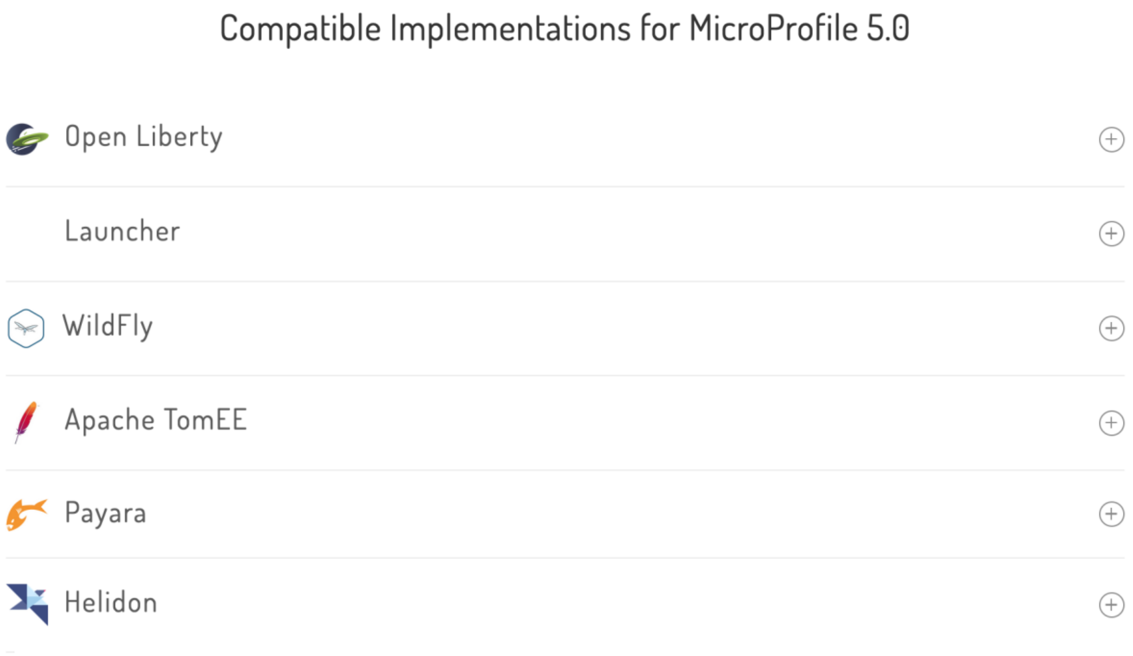


Implementations

MICROPROFILE®

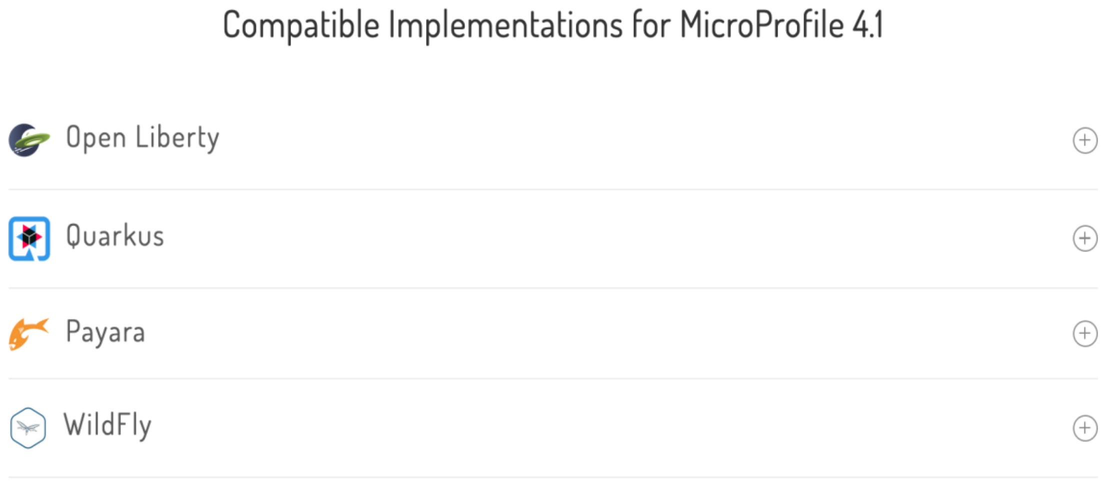


Compatible Implementations



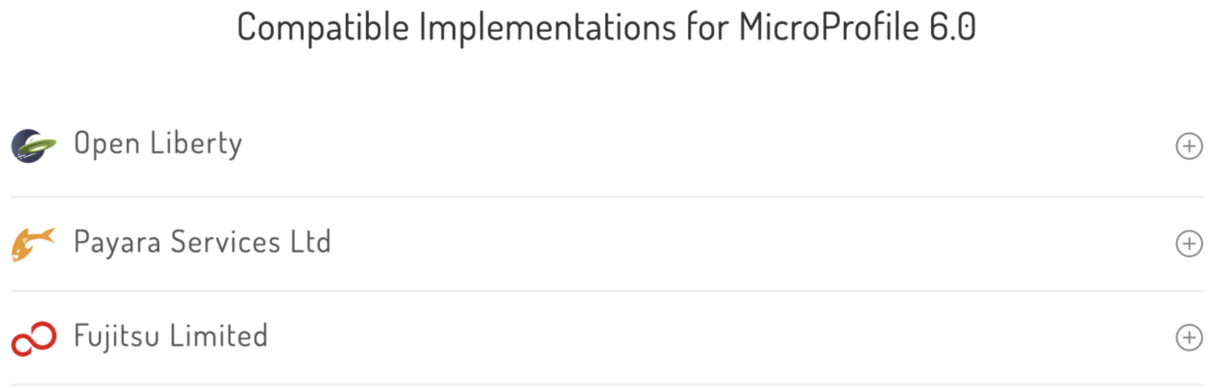
Compatible Implementations for MicroProfile 5.0

- Open Liberty ⊕
- Launcher ⊕
- WildFly ⊕
- Apache TomEE ⊕
- Payara ⊕
- Helidon ⊕



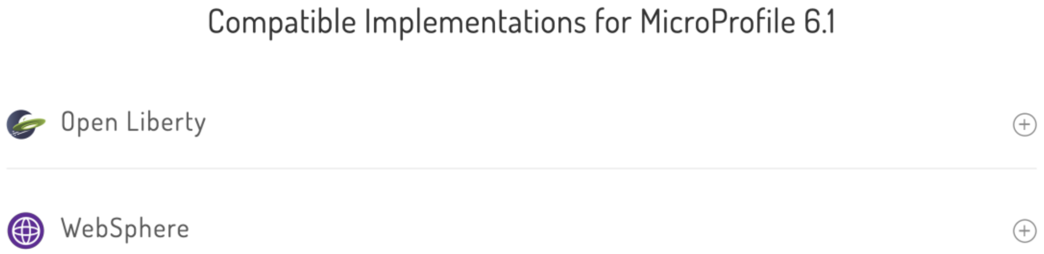
Compatible Implementations for MicroProfile 4.1

- Open Liberty ⊕
- Quarkus ⊕
- Payara ⊕
- WildFly ⊕



Compatible Implementations for MicroProfile 6.0

- Open Liberty ⊕
- Payara Services Ltd ⊕
- Fujitsu Limited ⊕



Compatible Implementations for MicroProfile 6.1

- Open Liberty ⊕
- WebSphere ⊕

Open Liberty used as the compatible implementation to release MicroProfile 4.1, 5.0 and 6.0

Open Liberty

Open Liberty

Designed with both **developers** and **application business owners** in mind.

- Liberty delivers the latest Java APIs and integrates with the most popular Developer and Build tools.
- Liberty has built-in innovation to reduce application runtime costs and delivery effort.

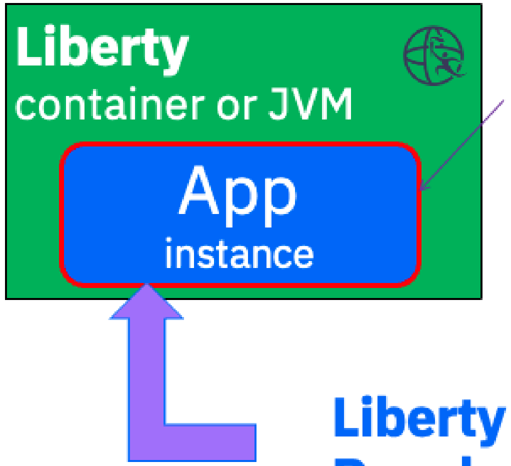
Built on open source

Open Liberty

https://openliberty.io/

JAKARTA EE

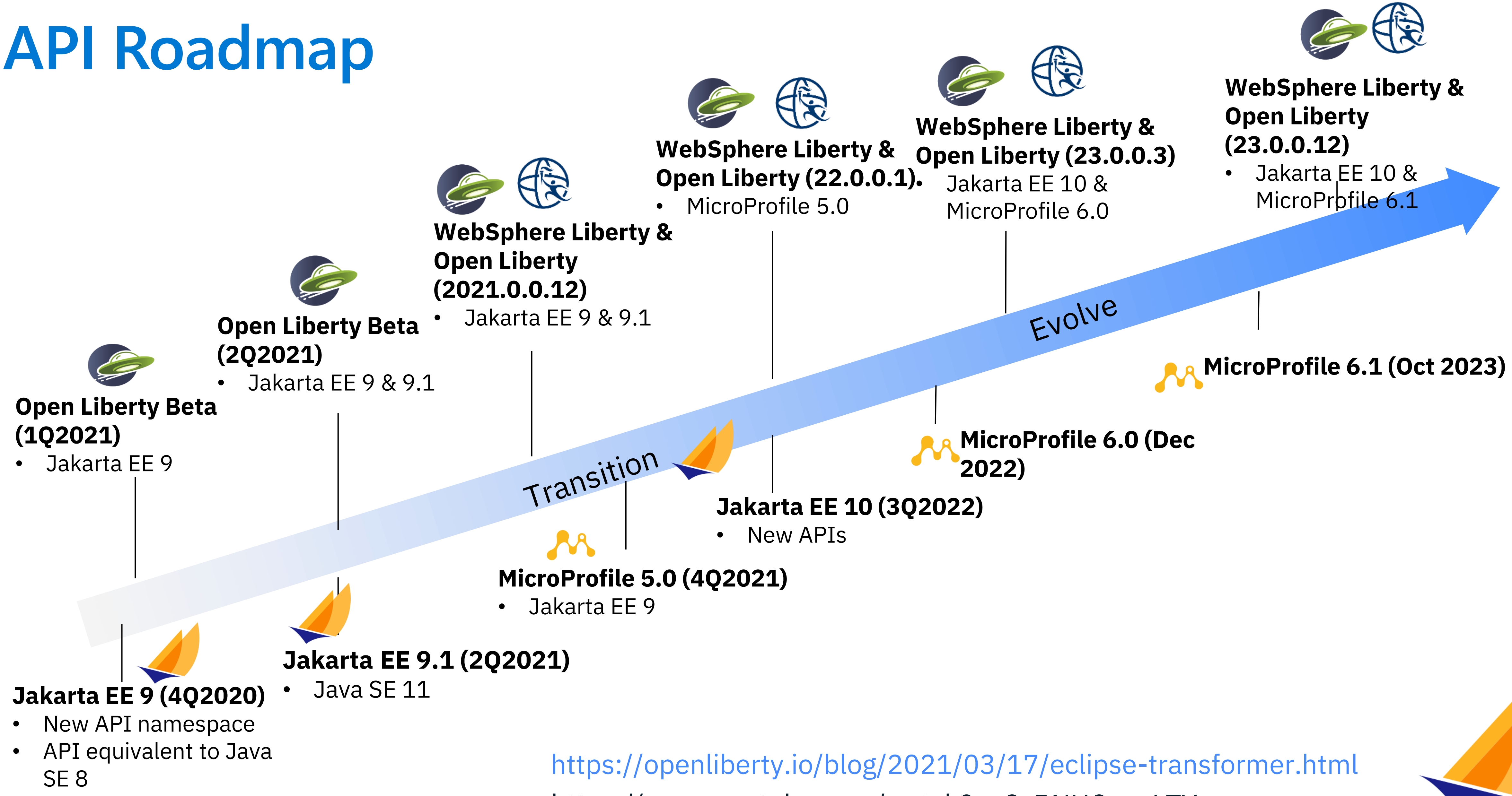
MICROPROFILE



MicroProfile and Jakarta EE APIs for Applications

Liberty Tools With All Popular Developer Environments (IDEs)

API Roadmap



Jakarta EE 9 (4Q2020)

- New API namespace
- API equivalent to Java SE 8

Open Liberty Beta (1Q2021)

- Jakarta EE 9

Open Liberty Beta (2Q2021)

- Jakarta EE 9 & 9.1

Jakarta EE 9.1 (2Q2021)

- Java SE 11

WebSphere Liberty & Open Liberty (2021.0.0.12)

- Jakarta EE 9 & 9.1

MicroProfile 5.0 (4Q2021)

- Jakarta EE 9

WebSphere Liberty & Open Liberty (22.0.0.1).

- MicroProfile 5.0

Jakarta EE 10 (3Q2022)

- New APIs

MicroProfile 6.0 (Dec 2022)

WebSphere Liberty & Open Liberty (23.0.0.3)

Jakarta EE 10 & MicroProfile 6.0

MicroProfile 6.1 (Oct 2023)

WebSphere Liberty & Open Liberty (23.0.0.12)

- Jakarta EE 10 & MicroProfile 6.1

https://openliberty.io/blog/2021/03/17/eclipse-transformer.html

https://www.youtube.com/watch?v=CyRNHCwmLTY

JAKARTA™ EE

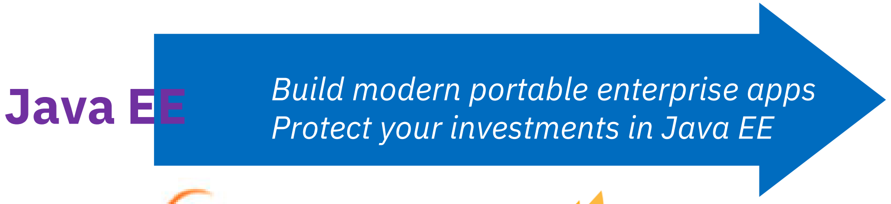
Vendor Neutral APIs

Microservice and Enterprise cloud-native APIs free from vendor lock-in

https://microprofile.io/compatible/
https://jakarta.ee/compatibility/

- **Build new** open cloud-native microservices with MicroProfile, leveraging existing Java EE/Jakarta EE skills and assets
- **Modernize existing** Java EE applications to cloud-native through Jakarta EE and MicroProfile

Optimizing Enterprise Java
for a Microservices Architecture


Java EE
Build modern portable enterprise apps
Protect your investments in Java EE

2022: First MicroProfile 5.0 and 6.0 Compatible runtimes
2023: First Jakarta EE 10 + MicroProfile 6.0 + Java 20 compatible implementation

Liberty Blogs: Jakarta EE 10, MicroProfile 6, and Java SE 20 support in Open Liberty 23.0.0.3

Liberty Overview

Lightweight, highly-efficient runtime

CI/CD-optimized operational experience

Simple true-to-production developer experience

Just enough runtime → 80% disk and 56% memory saving

Low operating cost → 4x increased density over Tomcat & Spring Boot

Continuous delivery → Zero-effort security fixing & zero technical debt

Zero migration → 100% v2v & fixpack migration saving

Kubernetes optimized → Self-tuned optimal perf, production-ready, kube-native

Developer experience → Container & kube-native experience, rapid inner loop

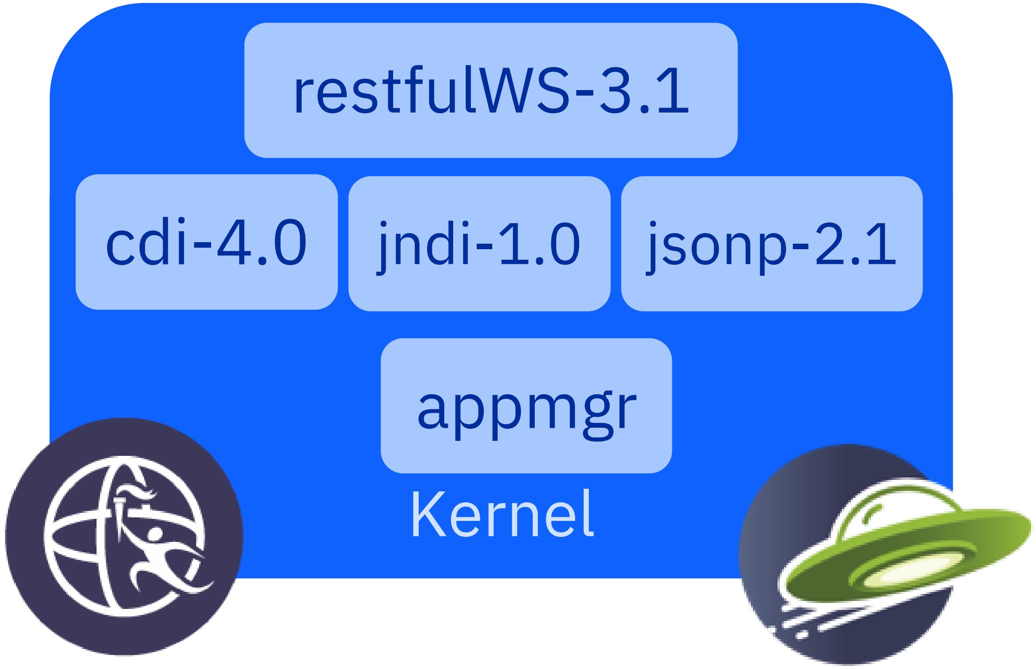
https://developer.ibm.com/articles/6-reasons-why-open-liberty-is-an-ideal-choice-for-developing-and-deploying-microservices/

Right-size deployments

With a Traditional App Server, the Full API stack as well as administration and operations features are loaded in each server instance

With Liberty, you control which features are loaded into each server instance

```
<feature>restfulWS-3.1</feature>
```

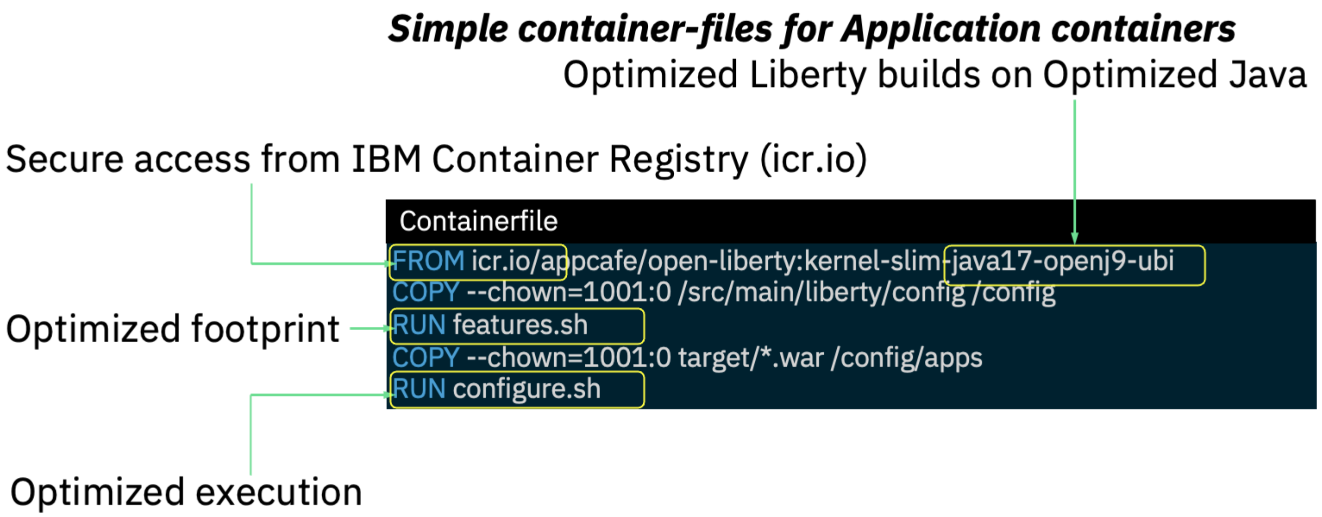


Simple right-size build

Friction-free, right-size application and container build

- **Application Build**
- Maven and Gradle Plugins
- All Liberty artefacts released to maven central

- **Container Build**
- Leading container build approaches – Dockerfile, Cloud Native Buildpack, Source-2-Image
- Certified Liberty images released to IBM Container Registry



Simple container-files for Application containers

Optimized Liberty builds on Optimized Java

Secure access from IBM Container Registry (icr.io)

Containerfile

```
FROM icr.io/appcafe/open-liberty:kernel-slim-java17-openj9-ubi
COPY --chown=1001:0 /src/main/liberty/config /config
RUN features.sh
COPY --chown=1001:0 target/*.war /config/apps
RUN configure.sh
```

Optimized footprint

Optimized execution

CI/CD Optimized

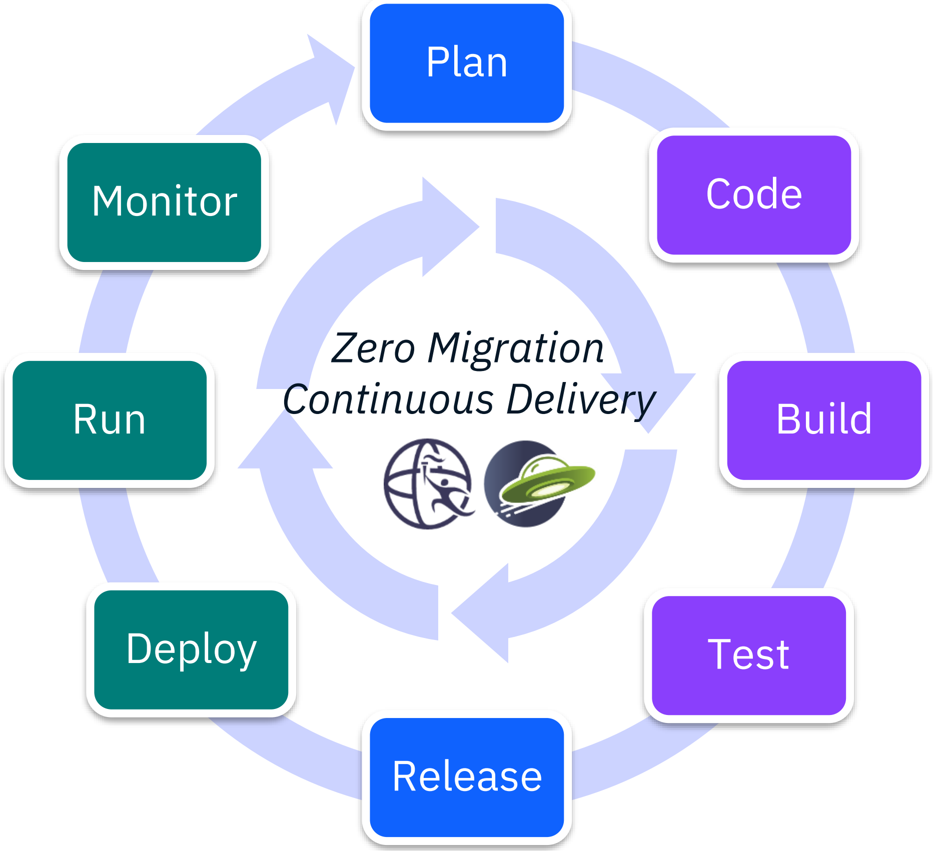
Seamless currency to eliminate technical debt and stay secure

Zero Migration makes staying current easy

- No configuration or runtime behavior changes

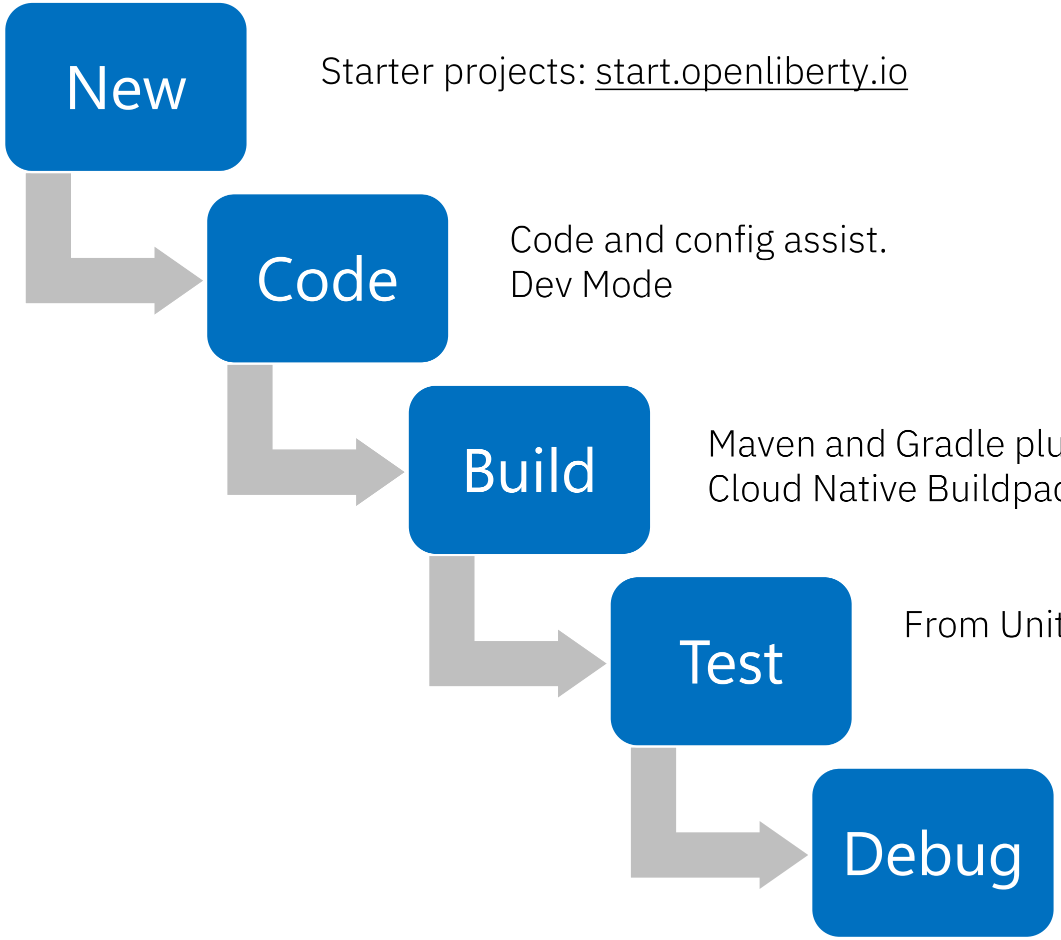
Continuous Delivery gives frequent and reliable access to the latest features and fixes

- Full releases every 4 weeks
- Quarterly LTS releases



Developer Efficiency

Rapid ***iterative development in your IDE of choice***



New

Starter projects: start.openliberty.io

Code

Code and config assist.
Dev Mode

Build

Maven and Gradle plugins
Cloud Native Buildpack, Source-2-image, Dockerfile

Test

From Unit Testing to Integration Testing in Containers

Debug

Hot code replace

Liberty Tools

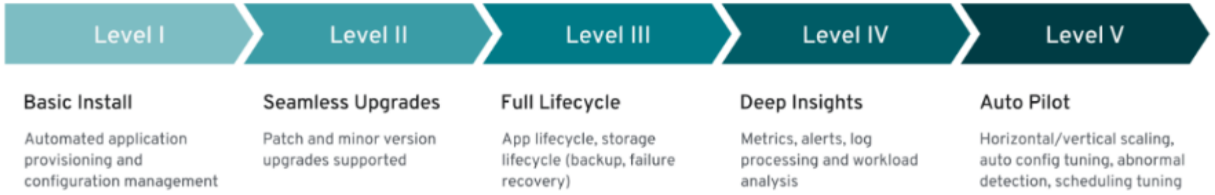
New

https://openliberty.io/blog/2023/06/27/23.0.0.6.html#devTools

Liberty Operators

Deploy

Addressing the Kubernetes skills gap



Level I

Basic Install

Automated application provisioning and configuration management

Level II

Seamless Upgrades

Patch and minor version upgrades supported

Level III

Full Lifecycle

App lifecycle, storage lifecycle (backup, failure recovery)

Level IV

Deep Insights

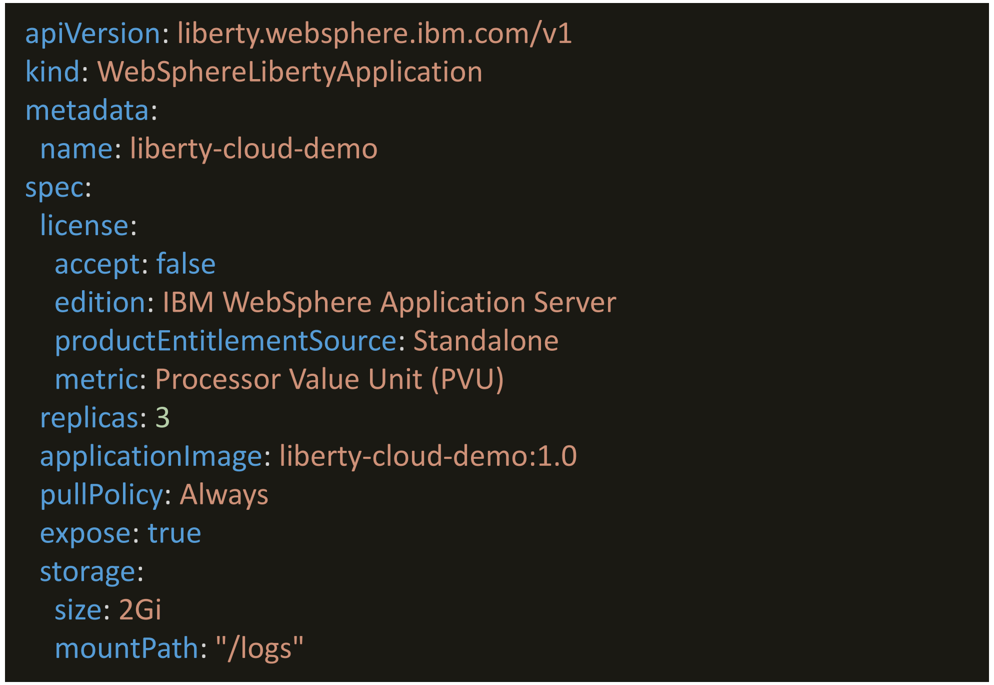
Metrics, alerts, log processing and workload analysis

Level V

Auto Pilot

Horizontal/vertical scaling, auto config tuning, abnormal detection, scheduling tuning

- Insulate from Kubernetes complexities
- Automate common task: deploy, scale, upgrade, dump gather
- Security capabilities out-of-the-box
- Reduce configuration by up to 80%



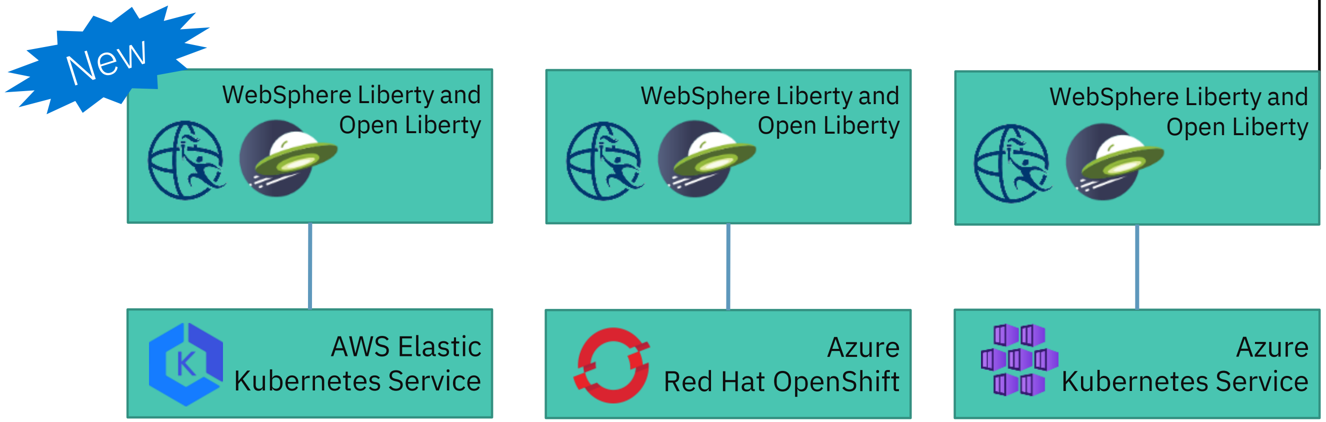
```
apiVersion: liberty.websphere.ibm.com/v1
kind: WebSphereLibertyApplication
metadata:
 name: liberty-cloud-demo
spec:
 license:
   accept: false
   edition: IBM WebSphere Application Server
   productEntitlementSource: Standalone
   metric: Processor Value Unit (PVU)
 replicas: 3
 applicationImage: liberty-cloud-demo:1.0
 pullPolicy: Always
 expose: true
 storage:
   size: 2Gi
   mountPath: "/logs"
```

Cloud Deployments

Deploy

- **Support on leading clouds**
- Liberty supported on all leading Cloud Virtual Machine, Kubernetes and OpenShift Environments
- Azure, AWS, IBM Cloud, Google Cloud, … (Bring-Your-Own-License)
- **New:** AWS ECS Fargate CaaS

- **Simplified setup**
- Marketplace and Partner Solution options simplify and accelerate setup in AWS and Azure at no extra cost
  - Provision or re-use Cluster
  - Provision or re-use container registry
  - Networking and load-balancing
  - Operator install and configuration
  - Application deployment


New
WebSphere Liberty and Open Liberty
AWS Elastic Kubernetes Service
WebSphere Liberty and Open Liberty
Azure Red Hat OpenShift
WebSphere Liberty and Open Liberty
Azure Kubernetes Service

https://docs.microsoft.com/en-us/azure/developer/java/ee/websphere-family

Observe/Day-2

Monitor

The right information in the right place at the right time

Observability

- All four categories of observability enabled through Liberty runtime, developer instrumentation and leading observability tools

Day-2 Operations

- Day-2 problem determination enabled through Liberty Operators – dumps, traces

Logging

- JSON logging (logs, trace, ffdc, access logs, audit logs)
- Integration with java logging API (JUL)
- LogRecordContext (add custom fields to log records)
- LogstashCollector feature, Kibana Dashboards
- Works with common log aggregators (Splunk, Humio, LogDNA, Elastic Stack, etc)

Metrics

- Built-in JVM and Liberty metrics
- App metrics using MicroProfile
  Metrics aggregation and dashboarding with Prometheus and Grafana

Tracing

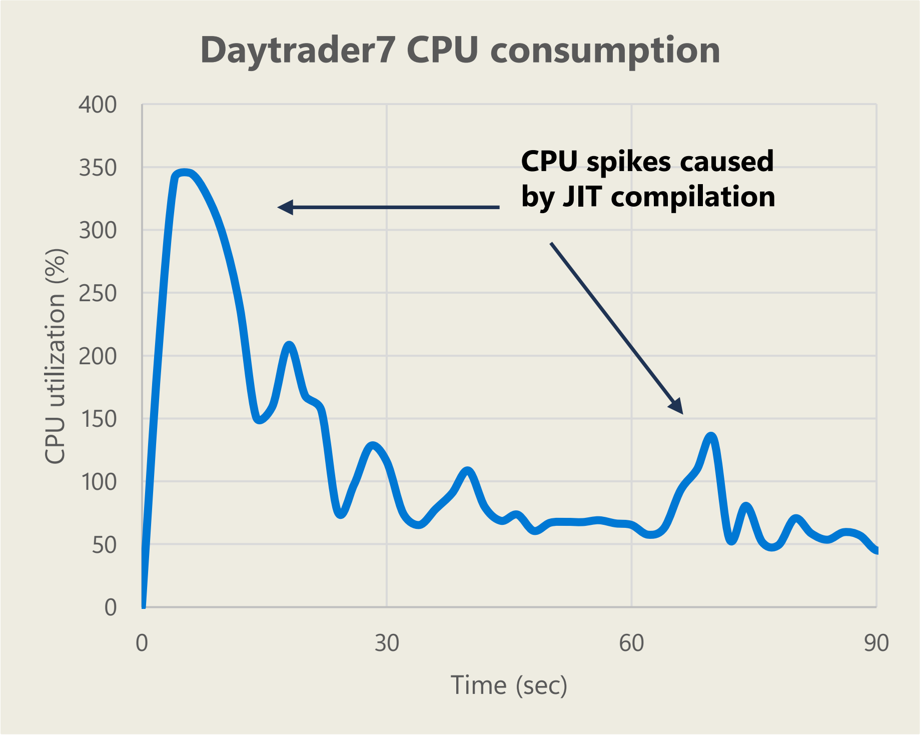
- Distributed tracing using MicroProfile Telemetry
- Built-in Jakarta REST instrumentation
- App instrumentation using MicroProfile Telemetry
- Zipkin, Jaeger to aggregate/visualize
- OpenTelemetry adoption

Health

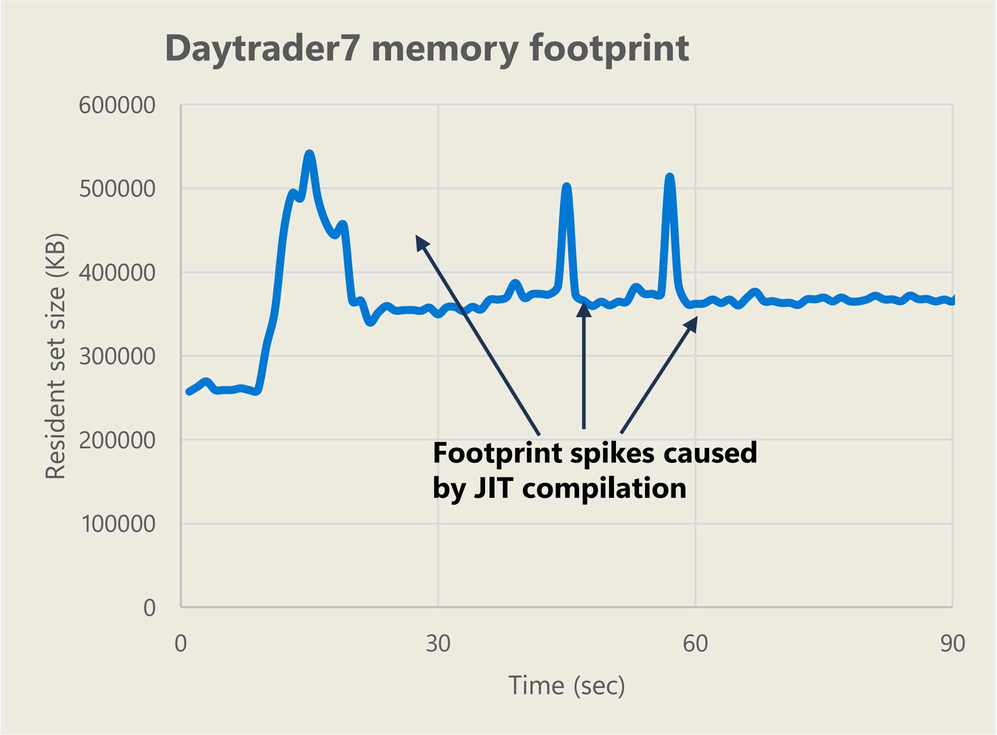
- Kube Health endpoint using MicroProfile
- Startup, Liveness, Readiness for different lifecycle states
- App health checks using MicroProfile

Cloud-Optimized Performance

- Initial execution run is “interpreted”, which is relatively slow
- “Hot” methods compiled by JIT can create CPU and memory spikes
- CPU spikes cause lower QoS
- Memory spikes cause OOM issues, including crashes
- Slow start-up time
- Slow ramp-up time



CPU spikes caused by JIT compilation

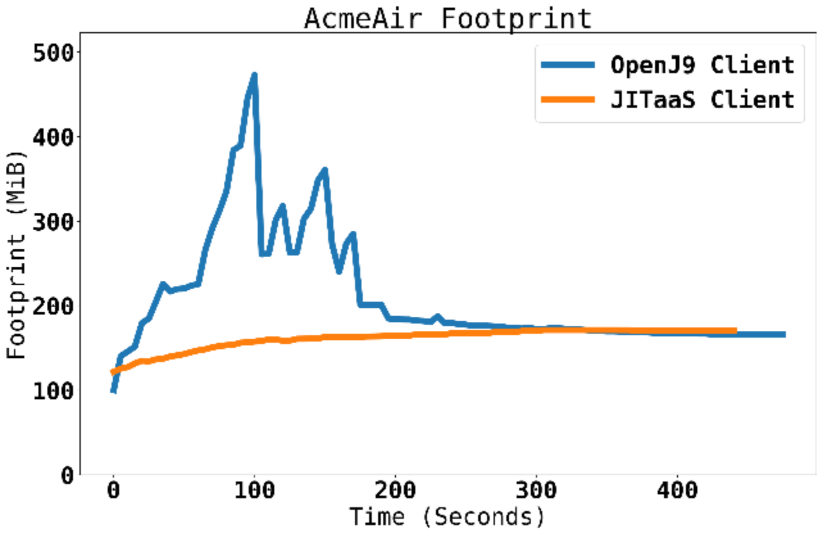


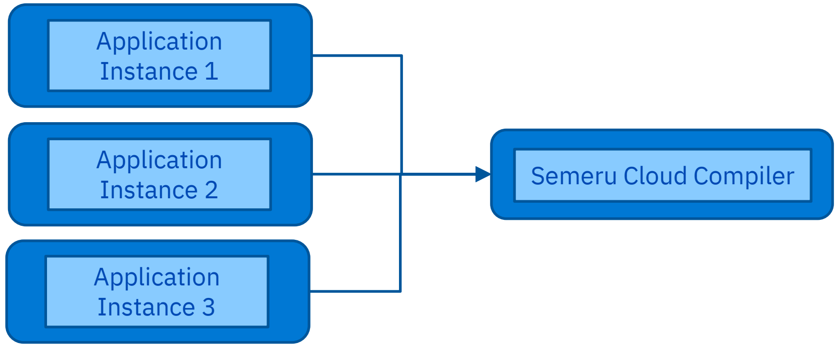
Footprint spikes caused by JIT compilation

Cloud Optimization - SCC

Optimizing memory for Kubernetes deployments

- Offloads costly JIT compilation to separate server
- Dramatically reduces peak memory usage
- New: Simple Kubernetes enablement through Liberty Operator

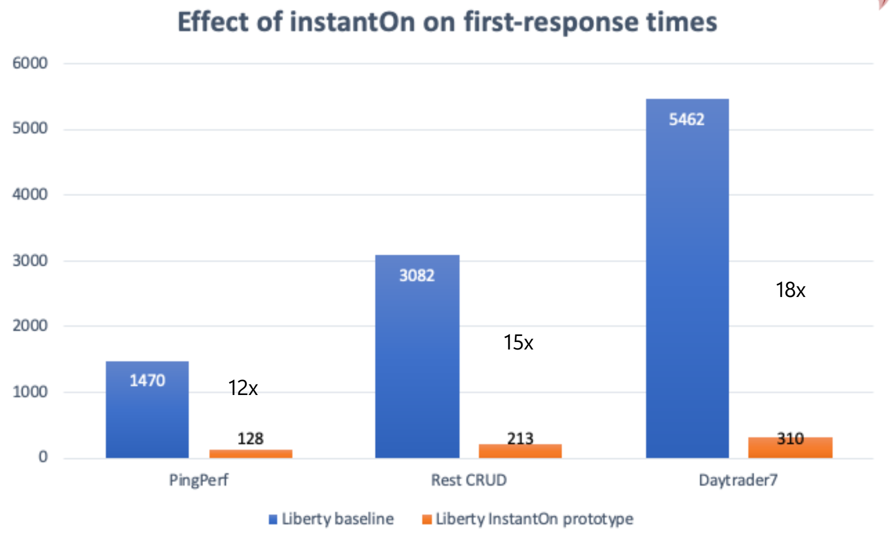


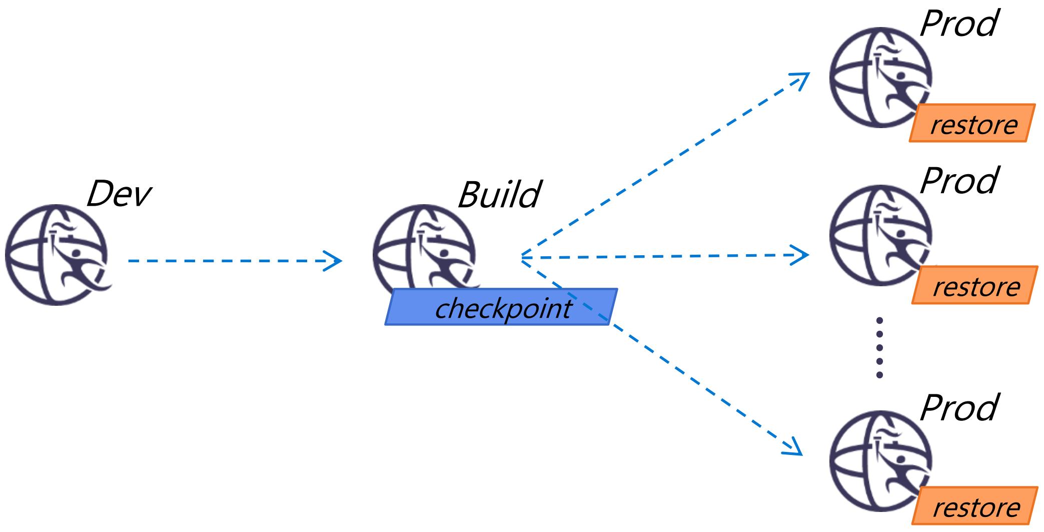


Service	Memory limit w/o SCC Server	Memory limit with SCC Server	Saving
Auth	1,050 MB	750 MB	300MB
Booking	3,300 MB	2,400 MB	900MB
Customer	1,650 MB	1,050 MB	600MB
Flight	2,250 MB	1,250 MB	1,000 MB
Main	600 MB	450 MB	150MB
Total	8,850 MB	5,900 MB	2,950 MB

InstantOn without compromise

- Use Linux CRIU Technology
- Start applications in milliseconds
- Ideal for serverless
- Up to 18x faster
- With all the benefits of the JVM and none of the compromises of Native Image





Characteristics	Semeru InstantOn	Semeru JVM	Graal Native
Full Java support	Yes	Yes	No
‘Instant on’	Yes	No	Yes
High throughput	Yes	Yes	No
Low memory (under load)	Yes	Yes	No
Dev-prod parity	Yes	Yes	No

Knative to mange scaling to zero

Knative to Serverless

Knative

A Kubernetes-based serverless framework

Build

Utilize and extend the existing Kubernetes’ primitives

Enables the source code from the dependencies and repository

Building container images and registering them

Serving

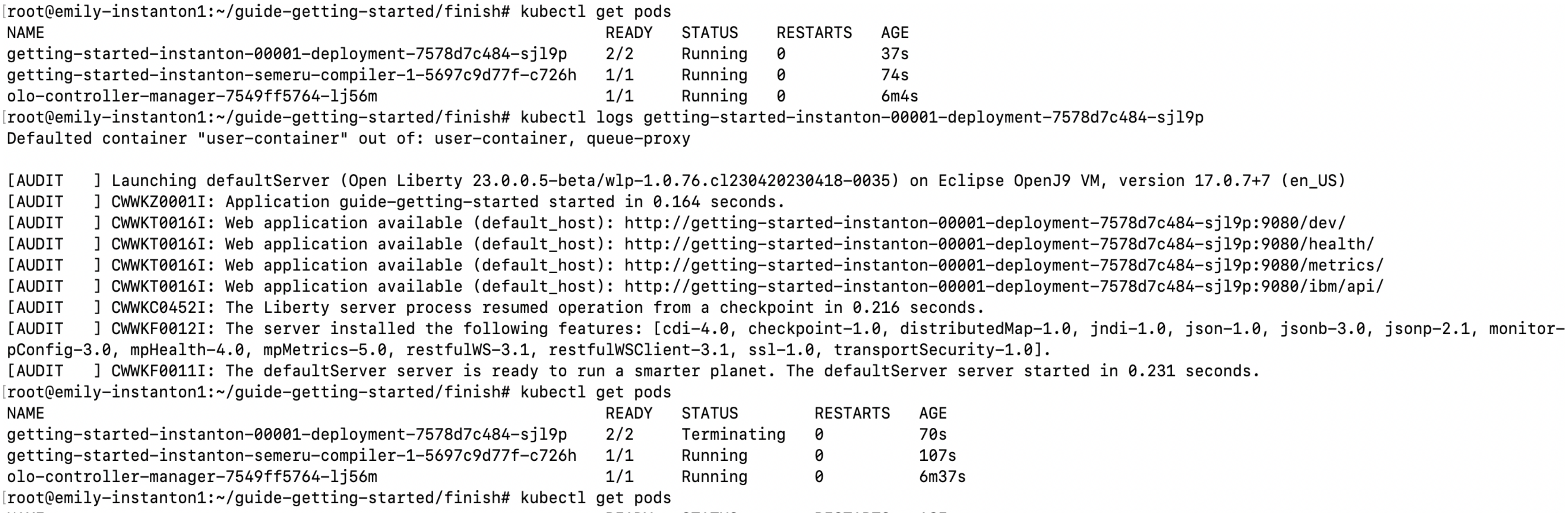
Enables rapid development of serverless containers
Automatic scaling for Istio components

Eventing

Event-driven architecture

SCC, InstantOn, Knative

SCC, Knative, InstantOn all together



```
[root@emily-instanton1:~/guide-getting-started/finish# kubectl get pods
NAME                                                   READY   STATUS    RESTARTS   AGE
getting-started-instanton-00001-deployment-7578d7c484-sjl9p   2/2     Running   0          37s
getting-started-instanton-semeru-compiler-1-5697c9d77f-c726h   1/1     Running   0          74s
olo-controller-manager-7549ff5764-lj56m                1/1     Running   0          6m4s
[root@emily-instanton1:~/guide-getting-started/finish# kubectl logs getting-started-instanton-00001-deployment-7578d7c484-sjl9p
Defaulted container "user-container" out of: user-container, queue-proxy

[AUDIT   ] Launching defaultServer (Open Liberty 23.0.0.5-beta/wlp-1.0.76.cl230420230418-0035) on Eclipse OpenJ9 VM, version 17.0.7+7 (en_US)
[AUDIT   ] CWWKZ0001I: Application guide-getting-started started in 0.164 seconds.
[AUDIT   ] CWWKT0016I: Web application available (default_host): http://getting-started-instanton-00001-deployment-7578d7c484-sjl9p:9080/dev/
[AUDIT   ] CWWKT0016I: Web application available (default_host): http://getting-started-instanton-00001-deployment-7578d7c484-sjl9p:9080/health/
[AUDIT   ] CWWKT0016I: Web application available (default_host): http://getting-started-instanton-00001-deployment-7578d7c484-sjl9p:9080/metrics/
[AUDIT   ] CWWKT0016I: Web application available (default_host): http://getting-started-instanton-00001-deployment-7578d7c484-sjl9p:9080/ibm/api/
[AUDIT   ] CWWKC0452I: The Liberty server process resumed operation from a checkpoint in 0.216 seconds.
[AUDIT   ] CWWKF0012I: The server installed the following features: [cdi-4.0, checkpoint-1.0, distributedMap-1.0, jndi-1.0, json-1.0, jsonb-3.0, jsonp-2.1, monitor-
pConfig-3.0, mpHealth-4.0, mpMetrics-5.0, restfulWS-3.1, restfulWSClient-3.1, ssl-1.0, transportSecurity-1.0].
[AUDIT   ] CWWKF0011I: The defaultServer server is ready to run a smarter planet. The defaultServer server started in 0.231 seconds.
[root@emily-instanton1:~/guide-getting-started/finish# kubectl get pods
NAME                                                   READY   STATUS        RESTARTS   AGE
getting-started-instanton-00001-deployment-7578d7c484-sjl9p   2/2     Terminating   0          70s
getting-started-instanton-semeru-compiler-1-5697c9d77f-c726h   1/1     Running       0          107s
olo-controller-manager-7549ff5764-lj56m                1/1     Running       0          6m37s
[root@emily-instanton1:~/guide-getting-started/finish# kubectl get pods
```

Demo

Find out more

Read

Watch

Try

Why Liberty

ibm.biz/6ReasonsWhyLiberty

Explore the latest on WebSphere and Liberty
ibm.biz/LibertyTV

Learn Liberty
https://openliberty.io/guides/

Try Liberty InstantOn
ibm.biz/InstantOn_HowToBlog

Resources

1. https://start.liberty.io
2. https://openliberty.io/guides/
3. https://microprofile.io/
4. https://jakarta.ee/
5. https://start.microprofile.io/
6. https://blog.openj9.org/2021/10/20/save-money-with-jitserver-on-the-cloud-an-aws-experiment/
7. https://openliberty.io/docs/latest/instanton.html

Microsoft

Thank you

Emily Jiang
IBM, Cloud Native Architect and Advocate
emijiang@uk.ibm.com
X/LinkedIn: @emilyfhjiang