

Liberty Quarterly
Update

Liberty Quarterly Update 19.0.0.7-19.0.0.9

Alasdair Nottingham STSM

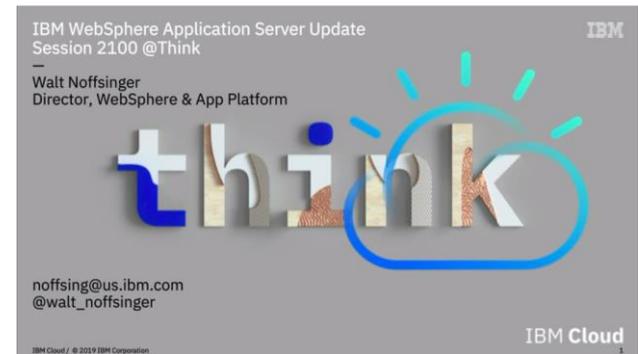
 @nottycode



Agenda

- Part 1: Liberty value and evolution
- Part 2: Technical deep-dive: what's new this quarter
- Part 2a: Jakarta EE & Eclipse MicroProfile – Kevin Sutter
- Part 2c: Liberty Performance Update – Vijay Sundaresan
- Part 5: Q&A

- For complete WAS Update presentation, see <http://ibm.biz/WebSphereUpdate>



Liberty Quarterly Update

Liberty Quarterly
Update

Liberty value and evolution

Why would I want it?



Why are customers choosing Liberty?

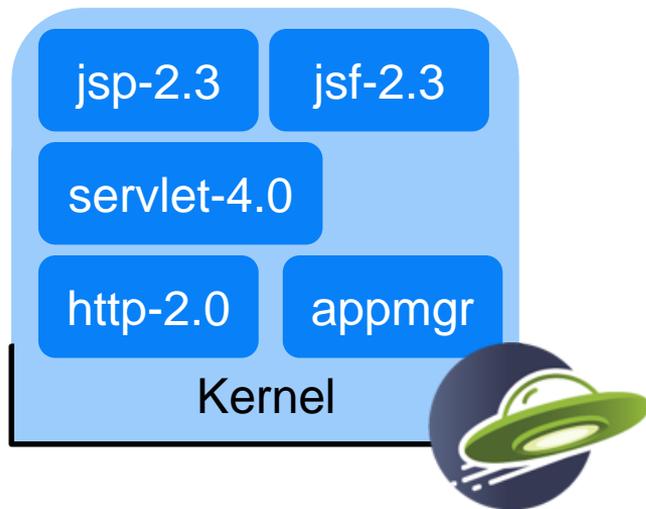
- “We want to reduce costs, simplify, and become cloud ready”
- Development efficiencies
- Flexible license options
 - even with collective management
- Rapid provisioning
 - small disk size
 - zip packages
 - remove the Ops team from the app update flow
- Lower running costs
 - dynamic config
 - zero migration
 - self-tuning thread pools

Fit-for-purpose server

- You control which features are loaded into each server instance



```
<feature>jsf-2.3</feature>
```



Liberty in Docker


app and config

+



liberty and java

+

os config

+



=


portable container

Hosted
Cloud

Private
Cloud

IBM Cloud Kubernetes Service
Azure Kubernetes Service
Google Kubernetes Engine
Amazon Elastic Kubernetes Service
Jelastic

Red Hat Open Shift Container Platform
Pivotal Kubernetes Service
Pivotal Cloud Foundry

FROM open-liberty
ADD myapp.war /config/dropins/myapp.war

Liberty Zero Migration

- Zero config migration
 - Write once, run forever
- Zero migration for apps
 - No behavior changes in existing features
 - New behaviors in new features
- Choose your Java
 - Java 12, 11, 8
 - AdoptOpenJDK
 - IBM
 - OpenJDK
 - Oracle



How to get Support

WebSphere Family



z/OS
Network
Deployment
Base
Core

IBM Cloud Pack for Applications



**WebSphere
Application Server**

Network Deployment
Base
Core



Red Hat OpenShift



Kabanero Enterprise

**IBM Integrated Application
Runtimes**

Java:

- Open Liberty
- MicroProfile
- Jakarta EE
- Spring
- OpenJ9
- HotSpot

Node.js:

- Loopback
- Express



Integrated DevOps

Focus areas

Security

APIs

Foundation

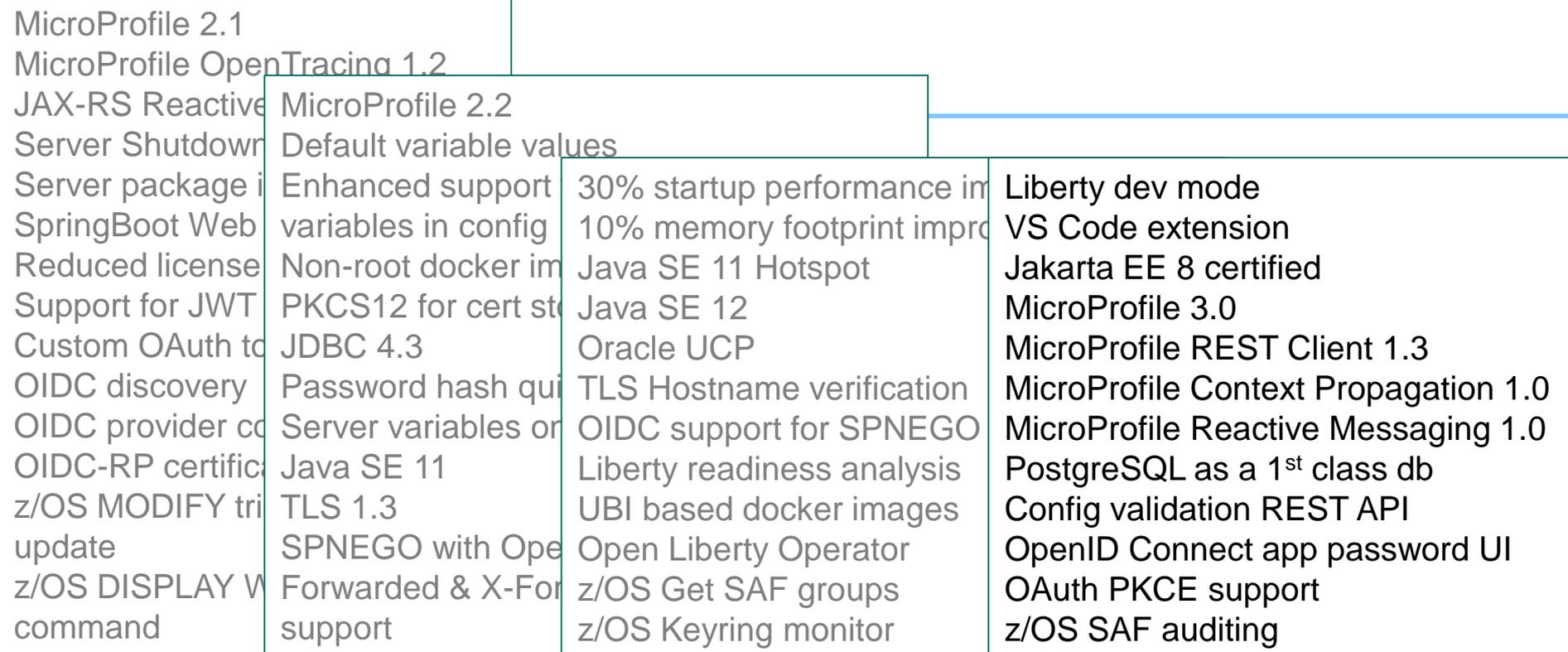
Orchestration

Developer Experience

Recent updates

What have we delivered since the last vPOT?





18.0.0.4
4Q18

19.0.0.1-3
1Q19

19.0.0.4-6
2Q19

19.0.0.7-9
3Q19

Liberty Quarterly Update

Periodic Table of Liberty (19.0.0.9)

zOS

ND

Base

Core

Open Liberty

New in 3Q19

New in 2Q19

New in 1Q19

New in 4Q18

batchSMFLogging-1.0		zosLocalAdapters-1.0		zosTransaction-1.0		zosSecurity-1.0	
collectiveController-1.0		zosRequestLogging-1.0		zosWlm-1.0			
clusterMember-1.0		healthAnalyzer-1.0		scalingController-1.0			
dynamicRouting-1.0		healthManager-1.0		scalingMember-1.0		Security	
cloudant-1.0		batchManagement-1.0		Operations		passwordUtilities-1.0	
javaee-7.0		wsAtomicTransaction-1.2				wsSecurity-1.1	
javaee-8.0						wsSecuritySaml-1.0	
sipServlet-1.0							
bells-1.0		mpContextPropagation-1.0		adminCenter-1.0		constrainedDelegation-1.0	
concurrent-1.0		mpReactiveMessaging-1.0		collectiveMember-1.0		federatedRepository-1.0	
javaMail-1.6		mpReactiveStreams-1.0		distributedMap-1.0		jwt-1.0	
jaxb-2.2		opentracing-1.3		eventLogging-1.0		jwtSso-1.0	
jdbc-4.3		osgiConsole-1.0		logstashCollector-1.0		sessionDatabase-1.0	
jpaContainer-2.2		springBoot-2.0		monitor-1.0		webCache-1.0	
jsfContainer-2.3		webProfile-7.0		openapi-3.1			
json-1.0		webProfile-8.0		requestTiming-1.0			
jsonbContainer-1.0				usageMetering-1.0			
jsonpContainer-1.1				restConnector-2.0			
microProfile-3.0				sessionCache-1.0			
						audit-1.0	
						ldapRegistry-3.0	
						oauth-2.0	
						openid-2.0	
						openidConnectClient-1.0	
						openidConnectServer-1.0	
						samlWeb-2.0	
						scim-1.0	
						socialLogin-1.0	
						spnego-1.0	
						transportSecurity-1.0	

APIs

Periodic Table of Liberty (19.0.0.9)

zOS

ND

Base

Core

Open Liberty

New in 3Q19

New in 2Q19

New in 1Q19

New in 4Q18

batchSMFLogging-1.0				appClientSupport-1.0	ejbHome-3.2	jacc-1.5	managedBeans-1.0
collectiveController-1.0				appSecurityClient-1.0	ejbPersistentTimer-3.2	jaxb-2.2	mdb-3.2
clusterMember-1.0				batch-1.0	ejbRemote-3.2	jaxws-2.2	wasJmsClient-2.0
dynamicRouting-1.0				concurrent-1.0	j2eeManagement-1.1	jca-1.7	webProfile-8.0
cloudant-1.0				ejb-3.2	javaMail-1.6	jms-2.0	wmqJmsClient-2.0
javaee-7.0							
javaee-8.0							
sipServlet-1.0							
bells-1.0		mpContextPropagation-1.0		appSecurity-3.0	jaxrs-2.1	jsonp-1.1	websocket-1.1
concurrent-1.0		mpReactiveMessaging-1.0		beanValidation-2.0	jaxrsClient-2.1	jsf-2.3	
javaMail-1.6		mpReactiveStreams-1.0		cdi-2.0	jdbc-4.2	jsp-2.3	
jaxb-2.2		opentracing-1.3		ejbLite-3.2	jndi-1.0	managedBeans-1.0	
jdbc-4.3		osgiConsole-1.0		el-3.0	jpa-2.2	servlet-4.0	
jpaContainer-2.2		springBoot-2.0		jaspic-1.1	jsonb-1.0	ssl-1.0	
jsfContainer-2.3		webProfile-7.0					
json-1.0		webProfile-8.0		cdi-2.0	jsonb-1.0	mpMetrics-2.0	mpOpenTracing-1.3
jsonbContainer-1.0				jaxrs-2.1	mpConfig-1.3	mpJwt-1.1	mpRestClient-1.3
jsonpContainer-1.1				jsonp-1.1	mpFaultTolerance-2.0	mpOpenAPI-1.1	mpHealth-2.0
microProfile-3.0							

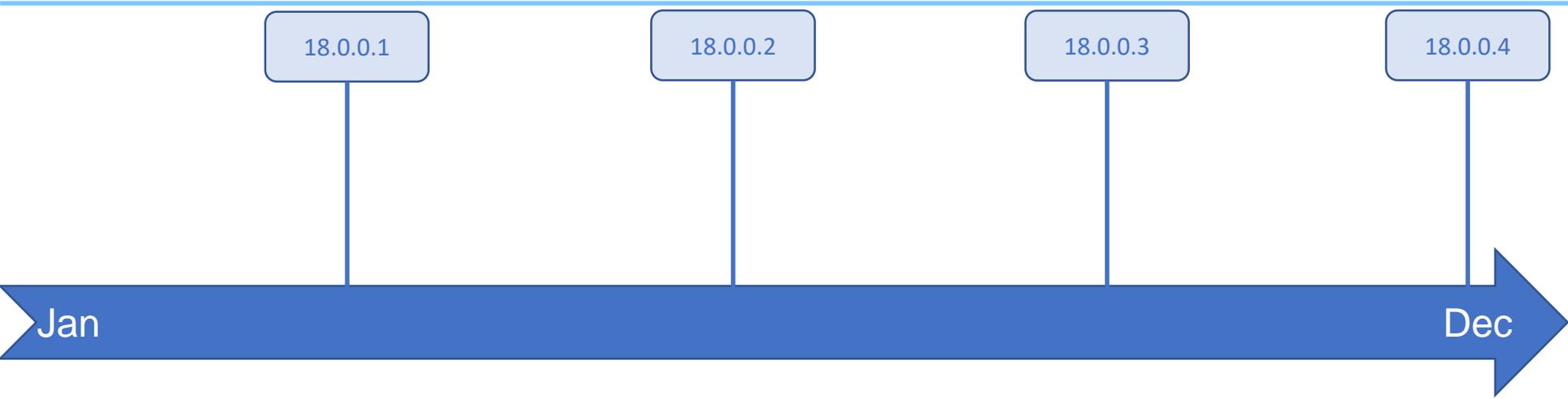
APIs

Liberty Quarterly
Update

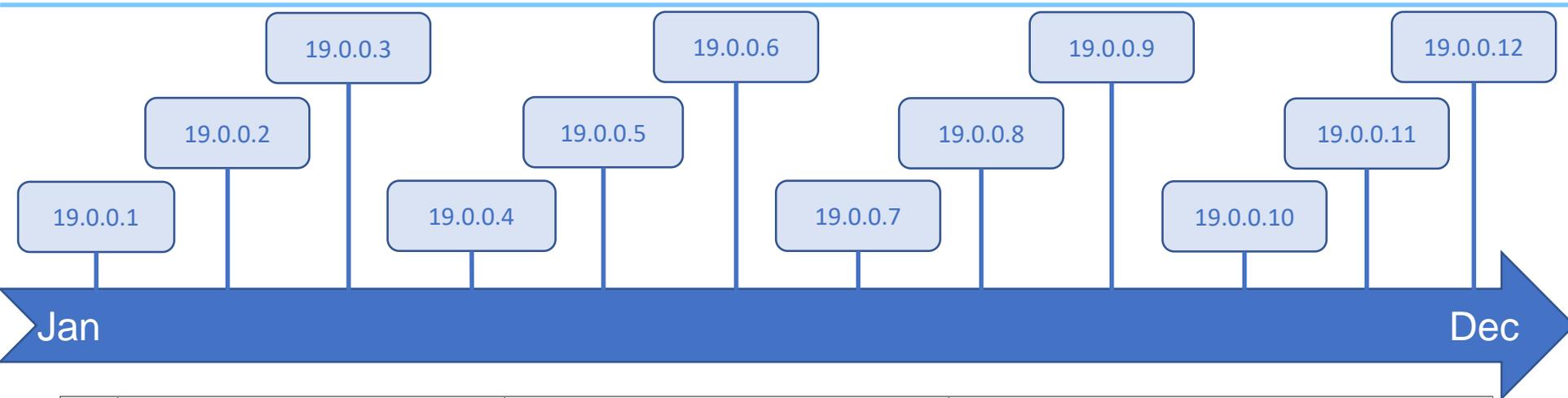
4-weekly release update

IBM

2018 - Quarterly Releases

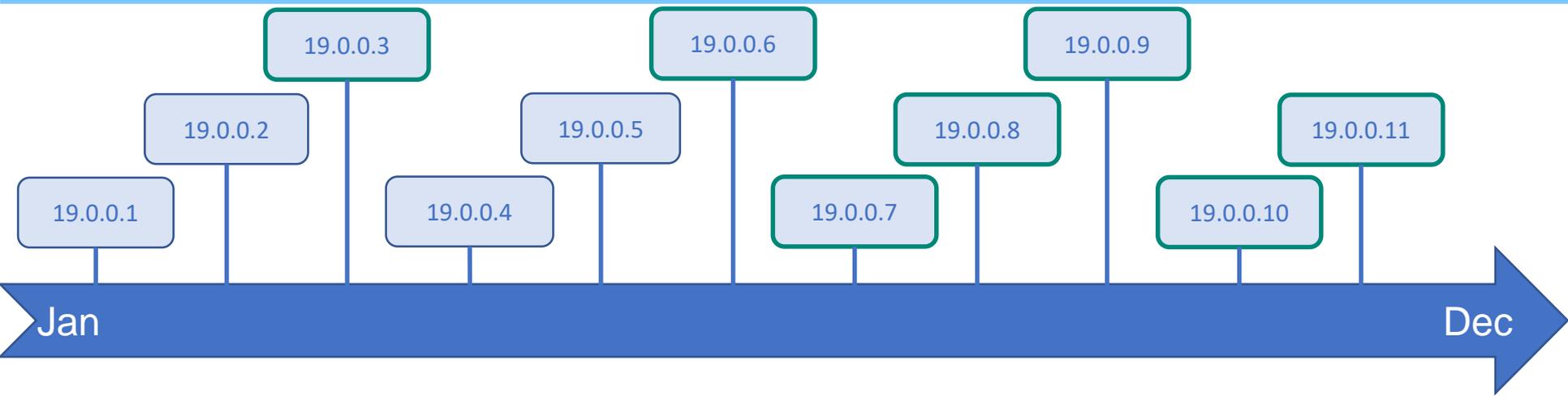


2019 - 4-weekly Releases



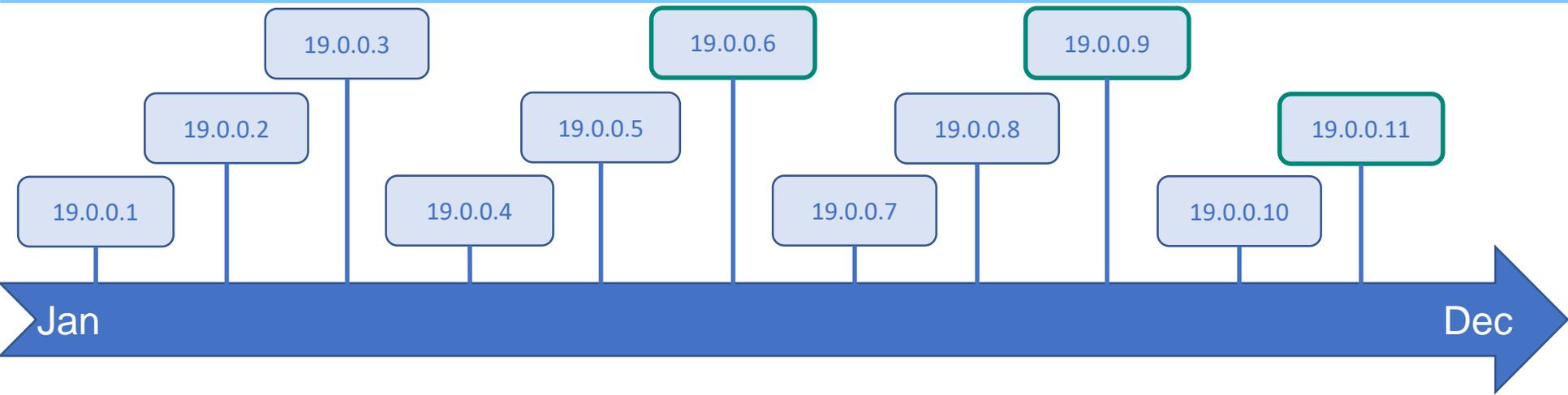
→	Fix Pack 19.0.0.7	25 July 2019	This is an estimated future release date.
→	Fix Pack 19.0.0.6	28 June 2019	
→	Fix Pack 19.0.0.5	31 May 2019	
→	Fix Pack 19.0.0.4	3 May 2019	
→	Fix Pack 19.0.0.3	5 April 2019	
→	Fix Pack 19.0.0.2	8 March 2019	
→	Fix Pack 19.0.0.1	8 February 2019	
→	Fix Pack 18.0.0.4	14 December 2018	
→	Fix Pack 18.0.0.3	21 September 2018	
→	Fix Pack 18.0.0.2	29 June 2018	
→	Fix Pack 18.0.0.1	16 March 2018	

2019 - iFixes



- Fixpacks released prior to 2019 still get 2 years of iFix coverage
- Fixpacks released 2019 and later get at least 6 months of iFix coverage
- Fixpacks ending in .3, .6, .9 or .12 get 2 years of iFix coverage

2019 - Proactive security fixes



- We provide security patches automatically for the last two fixpacks ending in one of .3, .6, .9, .12
- We also provide security patches automatically for the last fixpack

Liberty Quarterly Update

API

The IBM logo, consisting of the letters "IBM" in a bold, sans-serif font, with each letter formed by eight horizontal stripes of varying lengths, creating a striped effect.

An aerial photograph of a white boat on a body of water, viewed from above. The boat is moving towards the bottom left, leaving a white wake. The water is dark with some ripples. The entire image is overlaid with a semi-transparent yellow filter. The text 'Jakarta EE 8' and 'Open Liberty 19.0.0.6' is written in white, bold, sans-serif font on the left side of the image. A thin yellow horizontal line is positioned below the text.

Jakarta EE 8

Open Liberty 19.0.0.6

<https://jakartaone.org/>

Java EE -> Jakarta EE

The rapid success and momentum of MicroProfile leads to a rethink at Oracle.

In 2017 Oracle embrace the open community approach and contribute their assets to a new Eclipse Foundation Jakarta EE Project.

Future releases of Java EE technology will be branded **Jakarta EE**.

Eclipse MicroProfile remains the focus for cloud native innovation on top of Jakarta EE.



Final release Java EE 8



First release Jakarta EE 8 (binary and source compatible)

BUT: Oracle is aggressively protecting Java™ trademarks, references and acronyms. The community is working through the consequences.

Jakarta EE progressing outside the JCP



Specification First



Code First



Led by Specification Lead



Collaborative



Documents and TCKs are **closed source**



Documents and TCKs are **open source**



One normative **“Reference Implementation”**



One or more **“Compatible Implementations”**



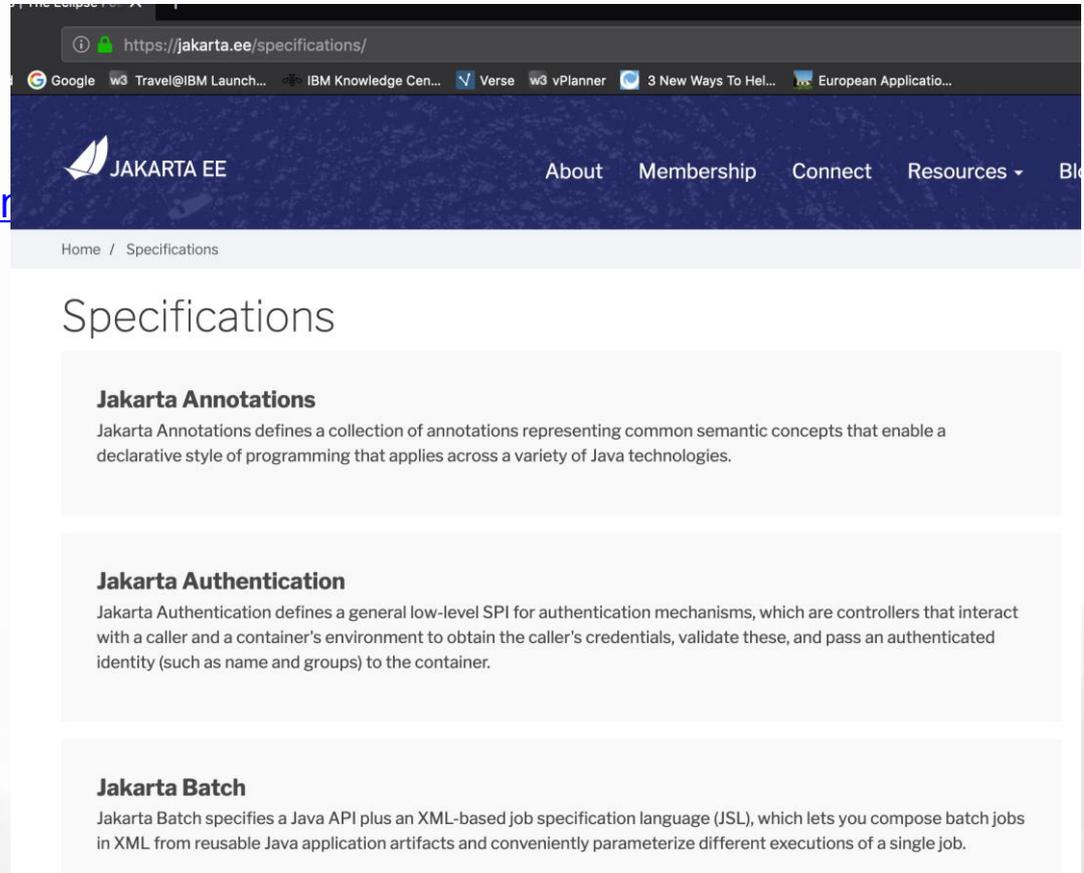
Oracle certification process



Self certification

Specifications

<https://jakarta.ee/specifications/>

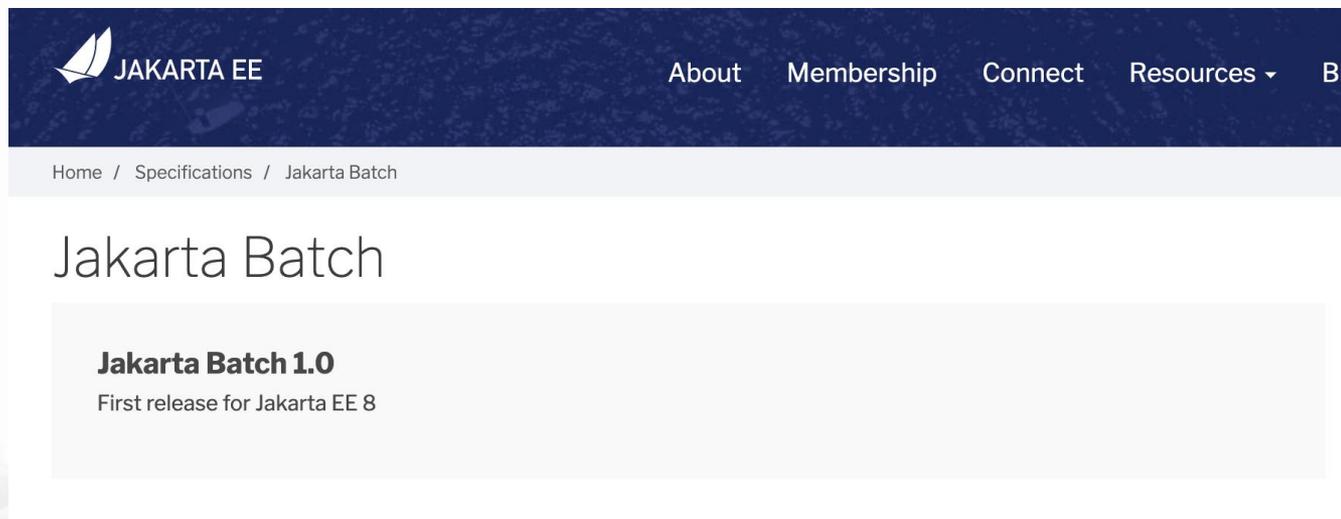


The screenshot shows a web browser displaying the Jakarta EE Specifications page. The browser's address bar shows the URL <https://jakarta.ee/specifications/>. The page features a dark blue header with the Jakarta EE logo and navigation links for About, Membership, Connect, Resources, and Blog. Below the header, a breadcrumb trail reads "Home / Specifications". The main content area is titled "Specifications" and lists three specifications:

- Jakarta Annotations**: Jakarta Annotations defines a collection of annotations representing common semantic concepts that enable a declarative style of programming that applies across a variety of Java technologies.
- Jakarta Authentication**: Jakarta Authentication defines a general low-level SPI for authentication mechanisms, which are controllers that interact with a caller and a container's environment to obtain the caller's credentials, validate these, and pass an authenticated identity (such as name and groups) to the container.
- Jakarta Batch**: Jakarta Batch specifies a Java API plus an XML-based job specification language (JSL), which lets you compose batch jobs in XML from reusable Java application artifacts and conveniently parameterize different executions of a single job.

Specifications – Jakarta Batch (example)

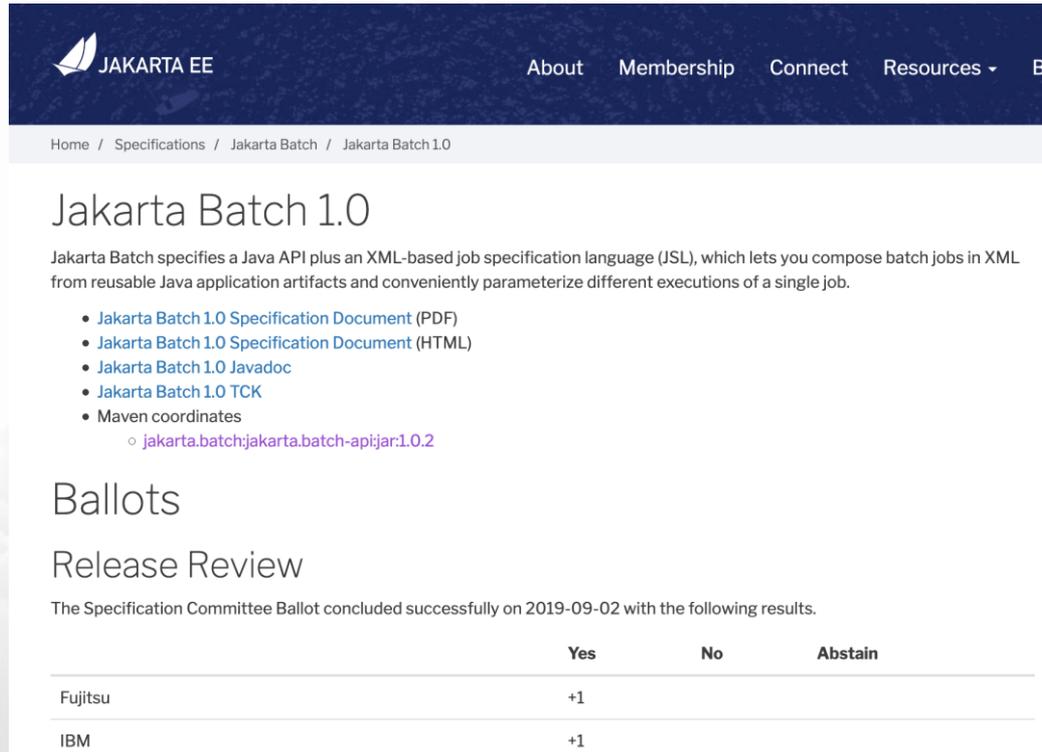
<https://jakarta.ee/specifications/batch/>



The screenshot shows the Jakarta EE website header with the logo and navigation links: About, Membership, Connect, Resources, and Blog. Below the header is a breadcrumb trail: Home / Specifications / Jakarta Batch. The main content area features the title "Jakarta Batch" and a section for "Jakarta Batch 1.0" with the subtitle "First release for Jakarta EE 8".

Specifications – Jakarta Batch 1.0 (example)

<https://jakarta.ee/specifications/batch/1.0/>



JAKARTA EE

About Membership Connect Resources ▾ Blo

Home / Specifications / Jakarta Batch / Jakarta Batch 1.0

Jakarta Batch 1.0

Jakarta Batch specifies a Java API plus an XML-based job specification language (JSL), which lets you compose batch jobs in XML from reusable Java application artifacts and conveniently parameterize different executions of a single job.

- [Jakarta Batch 1.0 Specification Document \(PDF\)](#)
- [Jakarta Batch 1.0 Specification Document \(HTML\)](#)
- [Jakarta Batch 1.0 Javadoc](#)
- [Jakarta Batch 1.0 TCK](#)
- [Maven coordinates](#)
 - [jakarta.batch:jakarta.batch-apjjar:1.0.2](#)

Ballots

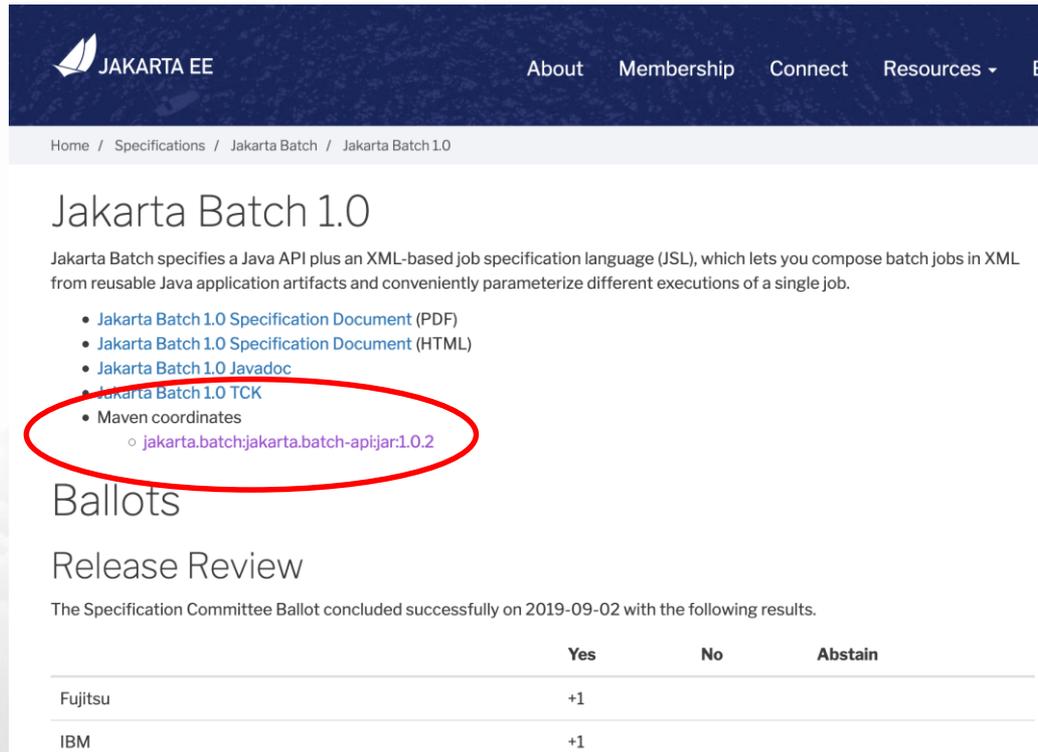
Release Review

The Specification Committee Ballot concluded successfully on 2019-09-02 with the following results.

	Yes	No	Abstain
Fujitsu	+1		
IBM	+1		

APIs – each Specification page...

<https://jakarta.ee/specifications/batch/1.0/>



JAKARTA EE

About Membership Connect Resources ▾ Blo

Home / Specifications / Jakarta Batch / Jakarta Batch 1.0

Jakarta Batch 1.0

Jakarta Batch specifies a Java API plus an XML-based job specification language (JSL), which lets you compose batch jobs in XML from reusable Java application artifacts and conveniently parameterize different executions of a single job.

- [Jakarta Batch 1.0 Specification Document \(PDF\)](#)
- [Jakarta Batch 1.0 Specification Document \(HTML\)](#)
- [Jakarta Batch 1.0 Javadoc](#)
- [Jakarta Batch 1.0 TCK](#)
- Maven coordinates
 - [jakarta.batch:jakarta.batch-apjjar:1.0.2](#)

Ballots

Release Review

The Specification Committee Ballot concluded successfully on 2019-09-02 with the following results.

	Yes	No	Abstain
Fujitsu	+1		
IBM	+1		

Jakarta EE TCKs

- Everyone is Welcome!
- <https://download.eclipse.org/jakartaee>

Directory Contents

- ..
- eclipse-annotations-tck-1.3.0.zip
- eclipse-annotations-tck-1.3.0.zip.sha256
- eclipse-annotations-tck-1.3.0.zip.sig
- jakarta-annotations-tck-1.3.0.zip
- jakarta-annotations-tck-1.3.0.zip.sha256
- jakarta-annotations-tck-1.3.0.zip.sig

Eclipse Downloads

This page is not designed to be web-accessible, as the project has not created a web page here. You can try the following:

- Contact the project members on their user [forum](#).
- Browse the directory contents to find the file you're looking for.
- Go to the [main Downloads page](#).

Directory Contents

- ..
- annotations
- authentication
- authorization
- batch
- bean-validation
- cdi
- concurrency
- connectors
- debugging
- dependency-injection
- expression-language
- jsonb
- jsonp
- mail
- messaging
- pages
- persistence
- restful-ws
- servlet
- tags
- transactions
- websocket
- xml-registries
- xml-rpc

Note: There are [separate branding requirements](#), but the use of the TCKs is for everyone!

Jakarta EE Compatible Implementations

- <https://jakarta.ee/compatibility>



The screenshot shows the 'Compatible Products' page on the Jakarta EE website. The page has a dark blue header with the Jakarta EE logo and navigation links: 'About', 'Membership', 'Connect', 'Resources', and 'Compatible Products'. The main heading is 'Compatible Products' with a 'Get Listed' button below it. The page content includes a breadcrumb 'Home / Compatible Products', a sub-heading 'Compatible Products', and a paragraph explaining the success of Jakarta EE technology. Below this are two bullet points: 'Jakarta EE 8 Full Platform Compatible Products' and 'Jakarta EE 8 Web Profile Compatible Products'. A section titled 'Jakarta EE 8 Full Platform Compatible Products' lists three implementations: Eclipse Glassfish (Eclipse Foundation, Version: 5.1.0, Full Profile), Open Liberty (IBM Corporation, Version: 19.0.0.6), and WildFly (Red Hat, Version: 17.0.1.Final). Each implementation has a 'Download' button.



Eclipse MicroProfile:

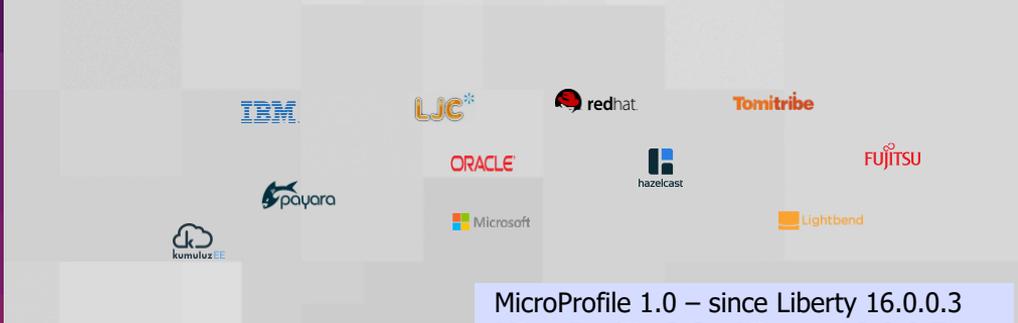
▶ <https://microprofile.io>

Jakarta EE:

▶ <https://www.jakarta.ee/>

2019

Both Eclipse MicroProfile and Jakarta EE continue to evolve and mature



MicroProfile 1.0 – since Liberty 16.0.0.3
MicroProfile 3.0 - Liberty 19.0.0.7

MICROPROFILE 3.0

NEW ENTERPRISE CAPABILITIES FOR MICROSERVICES

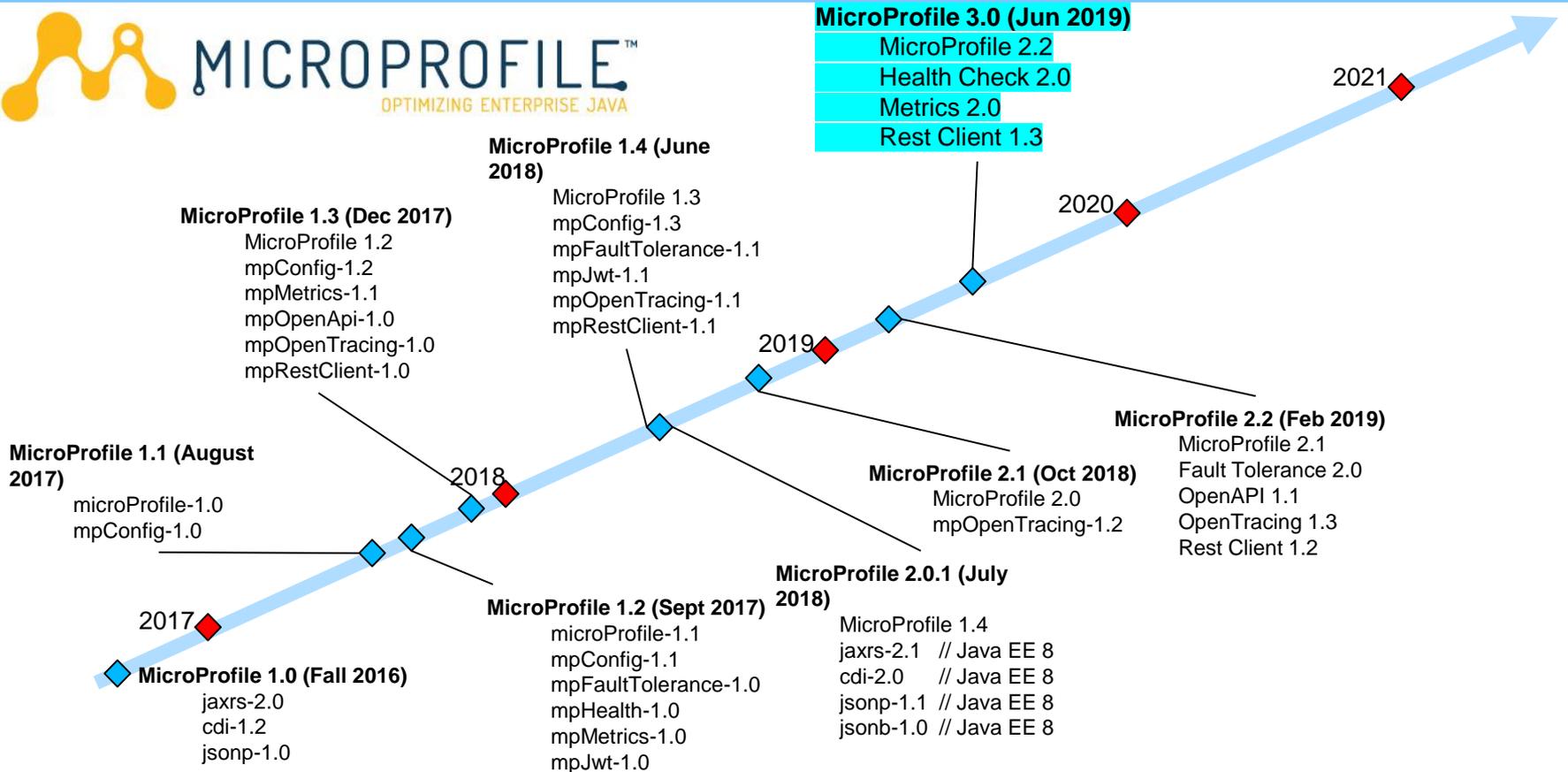
Open Tracing 1.3	Open API 1.1	Rest Client 1.3	Config 1.3
Fault Tolerance 2.0	Metrics 2.0	JWT 1.1	Health Check 2.0
CDI 2.0	JSON-P 1.1	JAX-RS 2.1	JSONB 1.0

BUILDS ON ... JAVA EE 8.0 → JAKARTA EE 8.0

Customer Overview



MICROPROFILE™
OPTIMIZING ENTERPRISE JAVA



Eclipse MicroProfile 3.0 (June 2019)

**Based on
Java EE 8**

Open Tracing 1.3	Open API 1.1	Rest Client 1.3	Config 1.3
Fault Tolerance 2.0	Metrics 2.0	JWT Propagation 1.1	Health Check 2.0
CDI 2.0	JSON-P 1.1	JAX-RS 2.1	JSON-B 1.0

MicroProfile 3.0

-  = New
-  = Updated
-  = No change from last release (MicroProfile 2.2)

MicroProfile Metrics 2.0

New in mpMetrics-2.0

- `@Counted` annotation has been simplified. Counters only count up now (1, 2, 3, ...).
- New `@ConcurrentGauge` annotation used to count concurrent requests, including high/low watermarks, of annotated methods
- Multiple labeled time series can be tracked for a single metric name using tags

```
@Counted(name="countConcurrent")
void somethingToCountInParallel() {
    // method logic here...
}
```

```
@Counted(name="countSequential", monotonic=true)
void somethingToCount() {
    // method logic here..
}
```

```
@ConcurrentGauge(name="countConcurrent ")
void somethingToCountInParallel() {
    // method logic here...
}
```

```
@Counted(name="countSequential")
void somethingToCount() {
    // method logic here..
}
```

mpMetrics-1.x



mpMetrics-2.0

```
@Counted(name="servletCountGet", tags={"app=webshop"} )
void doGet() {
    // method logic here...
}

@Counted(name="servletCountPost", tags={"app=webshop"})
void doPost() {
    // method logic here..
}
```

```
@Counted(name="servletCount", tags={"app=webshop", "req=get"}
)
void doGet() {
    // method logic here...
}

@Counted(name="servletCount", tags={"app=webshop",
"req=post"})
void doPost() {
    // method logic here..
}
```

MicroProfile Health 2.0

- **mpHealth-2.0** - Provide your own health check procedures, to be invoked by Liberty, to verify the health of your microservice.
- Code using new convenient annotations (**@Liveness** and **@Readiness**) and standard Java APIs.
 - **@Health** annotation from MicroProfile Health Check 1.0 is now deprecated.
- View the health statuses of your microservices in simple UP/DOWN states in a JSON encoded payload response from new HTTP endpoints (**/health/ready**, **/health/live**, and **/health**).
- Message body of Health Check Response was modified from 1.0, the attributes outcome and state was replaced by status.

```
@Liveness
@ApplicationScoped
public class CheckMemoryUsage implements HealthCheck {

    @Override
    public HealthCheckResponse call() {

        if (getMemUsage() < 0.9) {
            return HealthCheckResponse.named("memUsage")
                .up()
                .build();
        }
        return HealthCheckResponse.named("memUsage")
            .down()
            .build();
    }
}
```

microservice

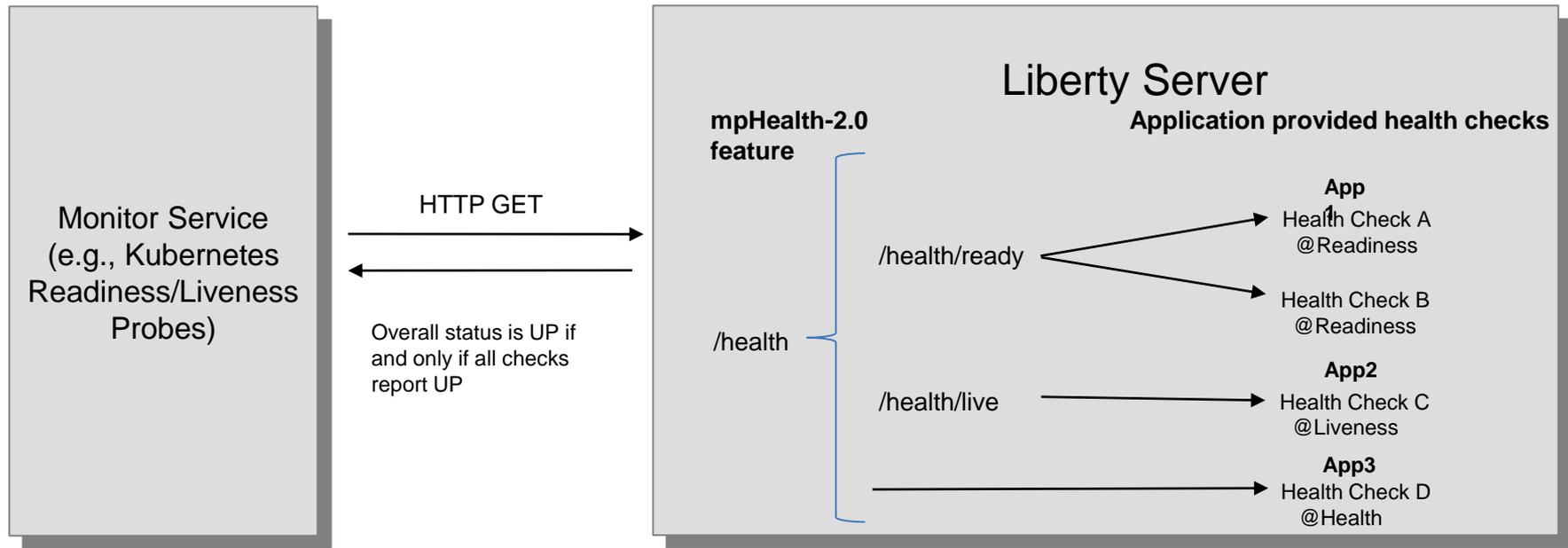


GET /health/live

```
{
  "status": "DOWN",
  "checks": [
    {
      "name": "memUsage",
      "status": "DOWN",
    },
    { "name": "threadCount",
      "status": "UP"
    }
  ]
}
```

Liberty Quarterly Update

MicroProfile Health 2.0



- `/health/ready` – Response provides aggregated data of all the @Readiness health check procedures and an overall UP/DOWN status indicator.
- `/health/live` – Response provides aggregated data of all the @Liveness health check procedures and an overall UP/DOWN status indicator.
- `/health` – Response provides all the aggregated data of all the @Liveness, @Readiness, and @Health health check procedures and an overall UP/DOWN status indicator.

MicroProfile Rest Client 1.3

What's new in MicroProfile Rest Client 1.3?

- Spec-defined SSL support
- Config Keys
- Default MIME type (application/json)
- Close/dispose methods

myKey-uri=http://myhost:9080/

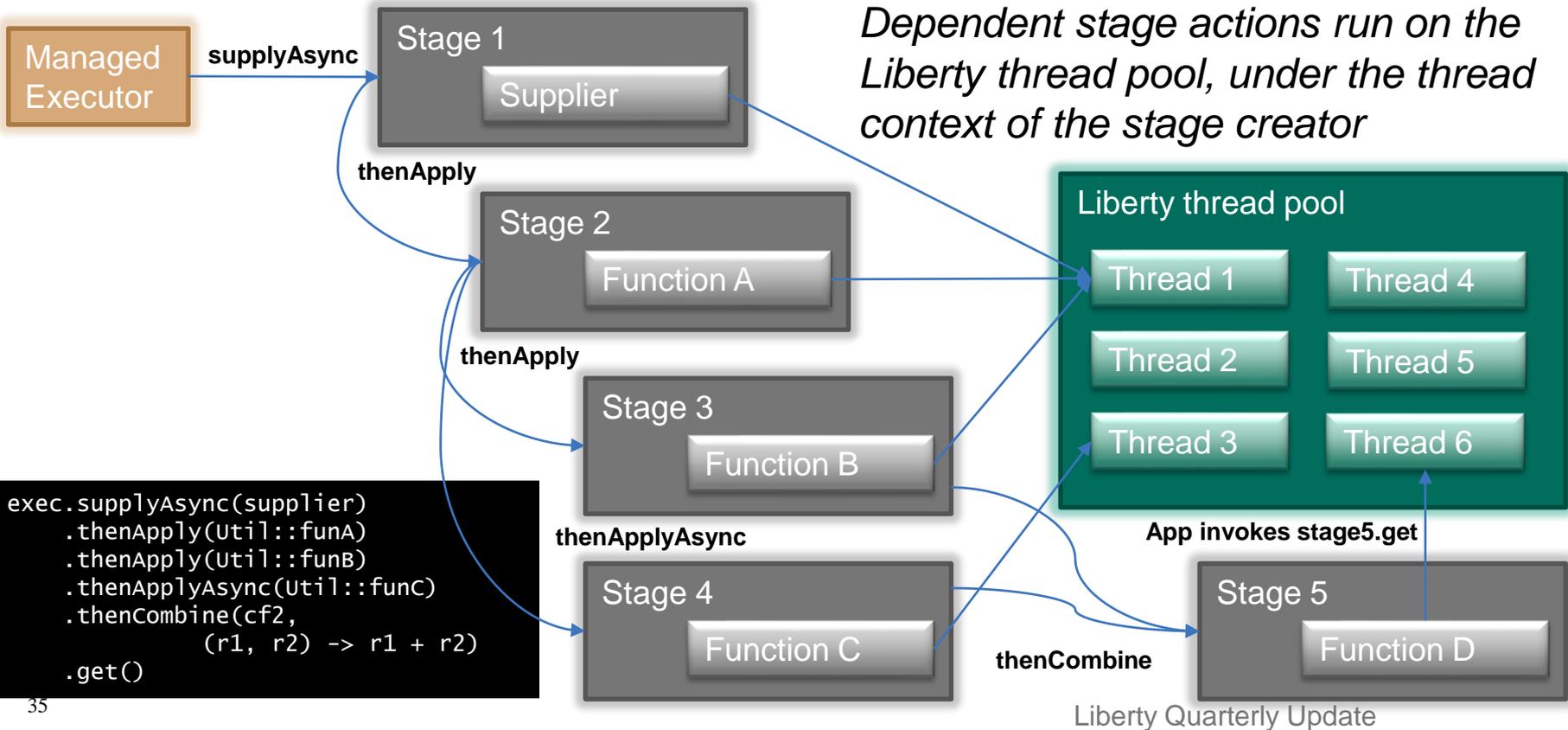
@RegisterRestClient(configKey="myKey")
public interface MyClient1 { ...

@RegisterRestClient(configKey="myKey")
public interface MyClient2 { ...

@RegisterRestClient(configKey="myKey")
public interface MyClient3 { ...

MicroProfile Context Propagation 1.0

Completion Stages backed by managed executors



MicroProfile Reactive Messaging 1.0

- Reactive Message
- Simple Kafka integration
- Extensible to other 'messaging' providers (e.g. Database, JMS)
- Configured through MP Config

```
@Incoming("greet-in")
@Outgoing("greet-out")
@Acknowledgement(MANUAL)
public Message<String>
    toUpper(Message<String> greeting) {
    return Message.of(
        greeting.getPayload().toUpper(),
        ()->greeting.ack());
}
```



```
mp.messaging.incoming.greet-in.connector=liberty-kafka
mp.messaging.incoming.greet-in.bootstrap.servers=kafkabrokerhost:9092
mp.messaging.outgoing.greet-out.connector=liberty-kafka
mp.messaging.outgoing.greet-out.bootstrap.servers=kafkabrokerhost:9092
```

Liberty Quarterly
Update

Open Liberty “Instant-on” Startup

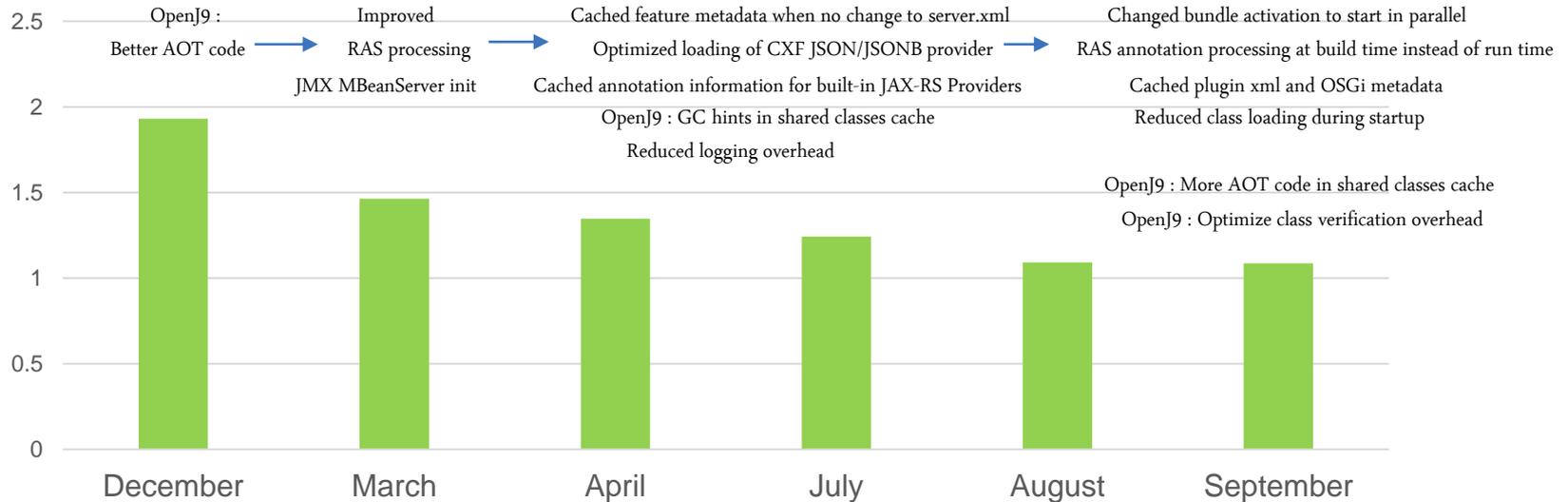


OpenLiberty “instant-on” startup

- Goal : get to “sub-second” startup on common systems/tasks
- De-construct Liberty startup steps
 - Per-task : bundle starting, feature manager, CDI container/equinox construction
 - Per-thread : look at degree of parallelism, which threads are doing which tasks
- Optimization
 - Look at parallelizing computation that cannot be eliminated
 - Look at the algorithmic complexity and improve
 - Avoid expensive code paths via persisting data from run-to-run

OpenLiberty+OpenJ9 : the road to one second startup time

2019 Progression of OpenLiberty+OpenJ9 PingPerf startup time (in seconds)

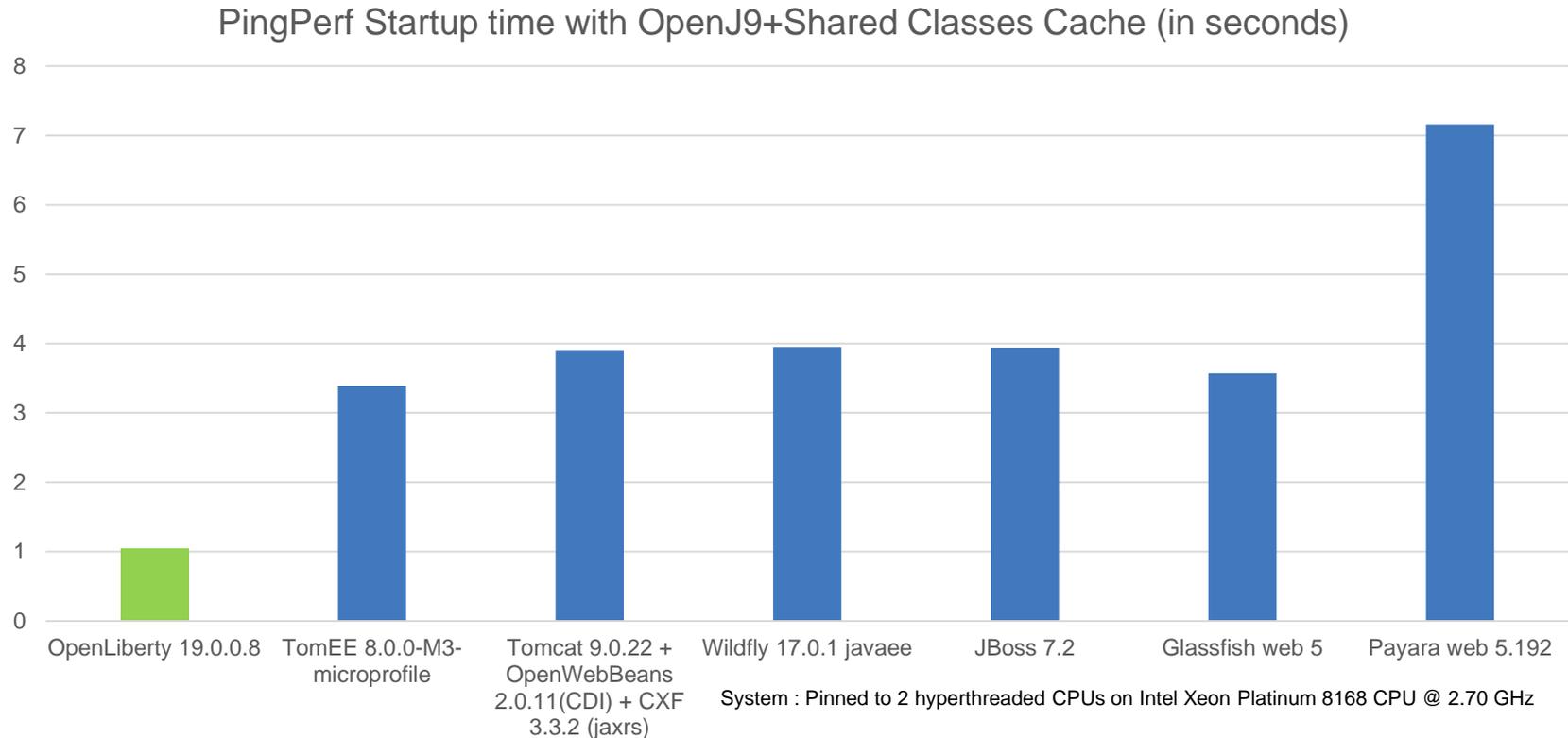


System : Pinned to 2 hyperthreaded CPUs on Intel Xeon Platinum 8168 CPU @ 2.70 GHz

Halved startup time due to app server and JVM changes done consistently throughout 2019
OpenLiberty and OpenJ9 changes improve startup time to be almost at **one second** now

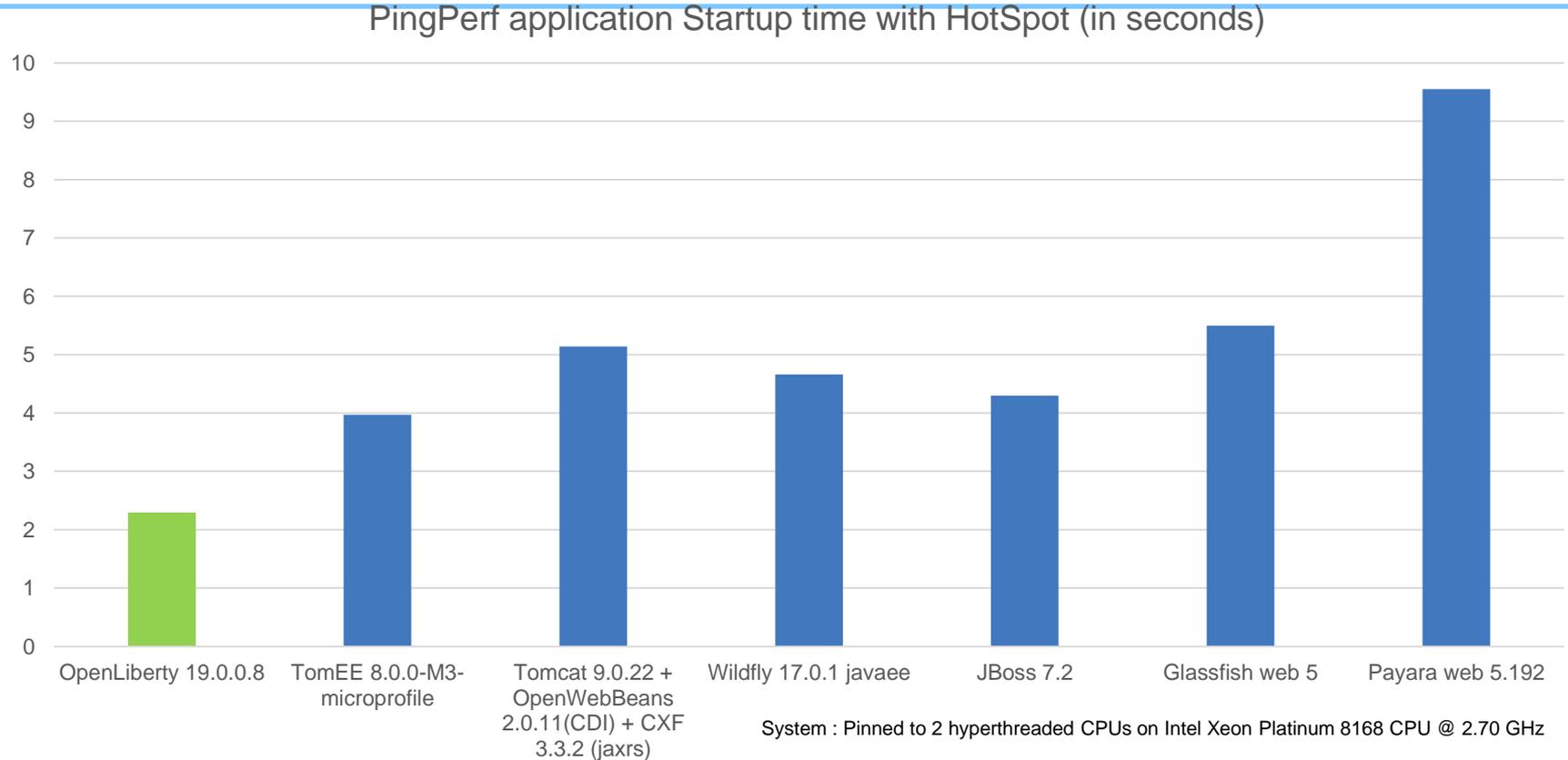
Liberty Quarterly Update

OpenLiberty startup time comparison with OpenJ9



- OpenLiberty startup time is significantly shorter than other application servers

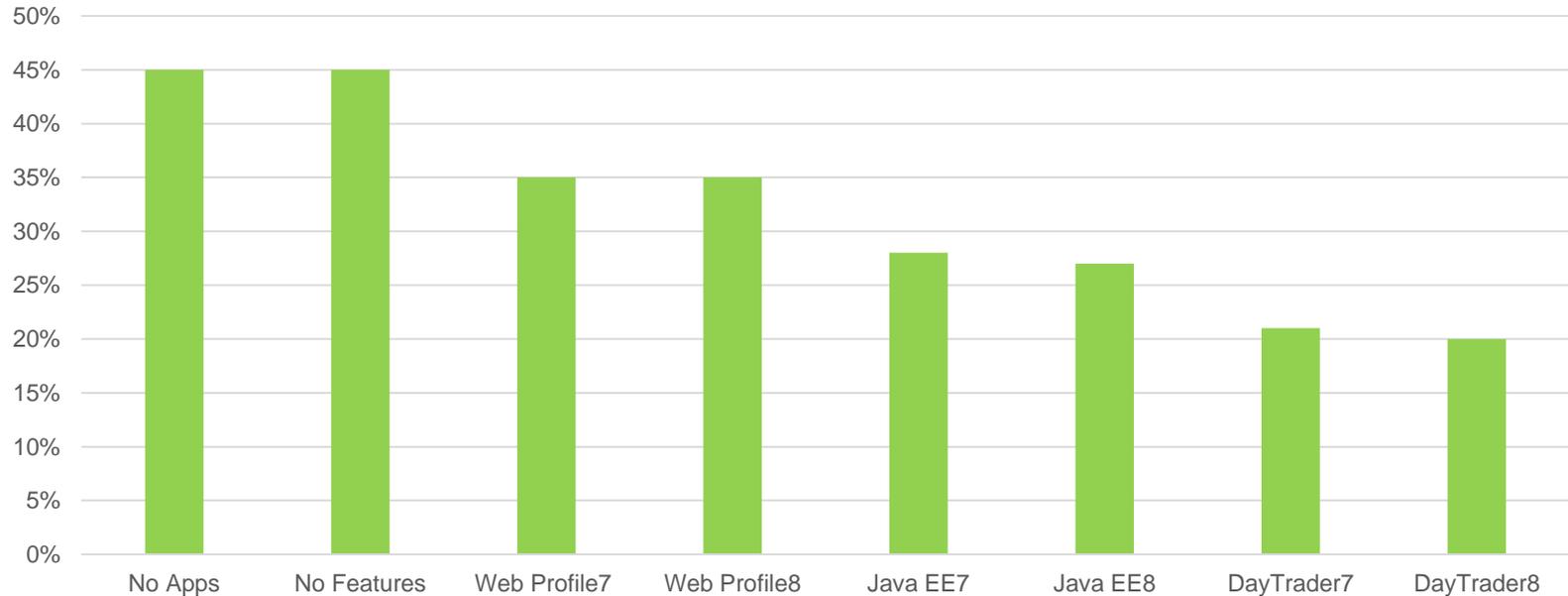
OpenLiberty startup time comparison with HotSpot



- OpenLiberty startup time is significantly shorter than other application servers

OpenLiberty+OpenJ9 startup time improvements

OpenLiberty+OpenJ9 19.0.0.9 vs 18.0.0.4
Startup Time Improvement Percentage



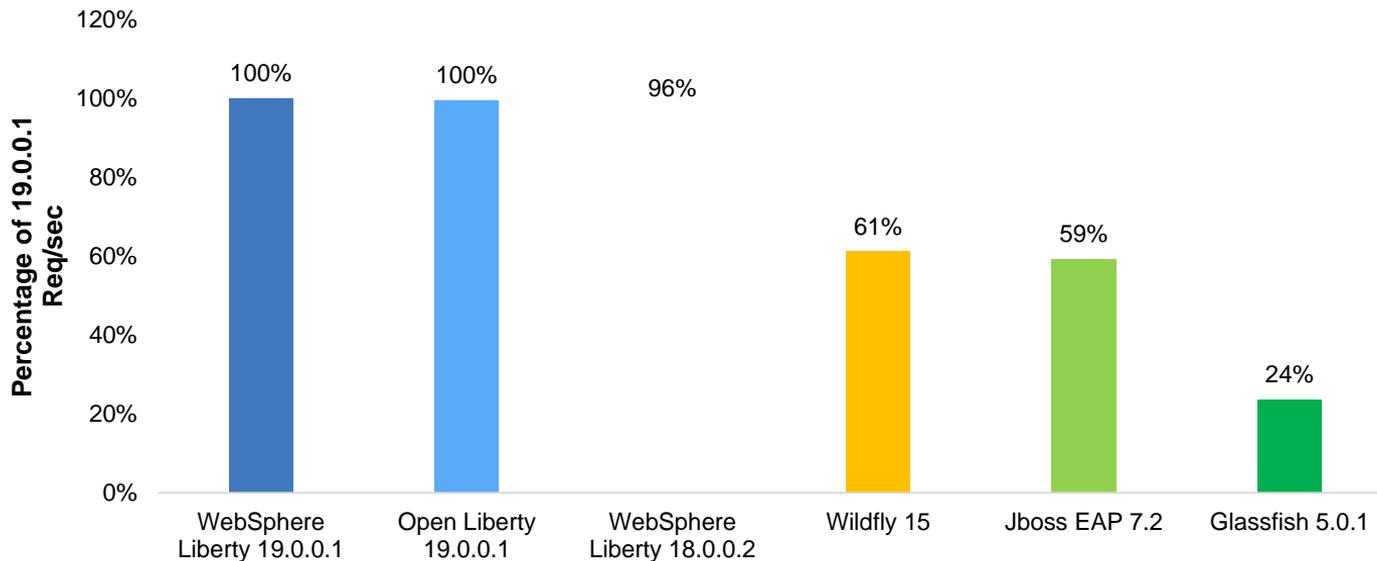
- OpenLiberty+OpenJ9 changes in 2019 improved startup time across the board

OpenLiberty+OpenJ9 startup optimizations : What's next ?

- Liberty changes
 - More bundles enabled for parallel activation with a goal of all of them eventually
 - New Annotation service engine with caching function
 - **Some CDI heavy applications expected to start up in less than half the time after first startup and data is cached**
 - Additional config caching
- OpenJ9 changes
 - Class verification overhead reduction when classes are in the shared class cache
 - More AOT optimizations
 - Make shared classes cache more consumable in Docker by having multi layered cache

EE8 Throughput (Daytrader8)

WebSphere Liberty ee8-Daytrader8 comparison (higher is better)



- WAS 19.0.0.1 Liberty Server EE8 throughput is at least 39% better than other Lightweight App Servers

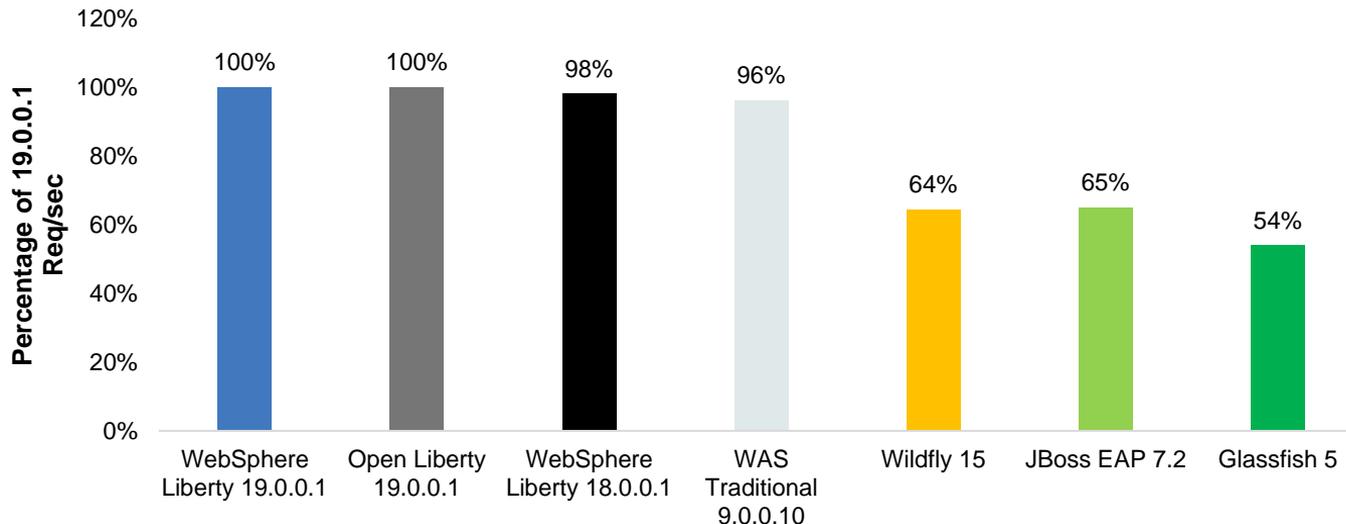
System Configuration:

- **SUT:** LinTel – SLES 11.4, Intel(R) Xeon(R) Platinum 8180 CPU @ 2.50GHz, 4 physical cores, 64GB RAM.
- Oracle JDK 8 u191 is used for non-IBM app servers & IBM JDK 8 SR5 FP30 is used for WebSphere Liberty 19.0.0.1 and IBM JDK SR5 FP17 for 18.0.0.2

EE7 Throughput (Daytrader7)

- WAS 19.0.0.1 Liberty Server EE7 throughput is at least 36% better than other Lightweight App Servers

**WebSphere Liberty ee7-Daytrader7 comparison
(higher is better)**



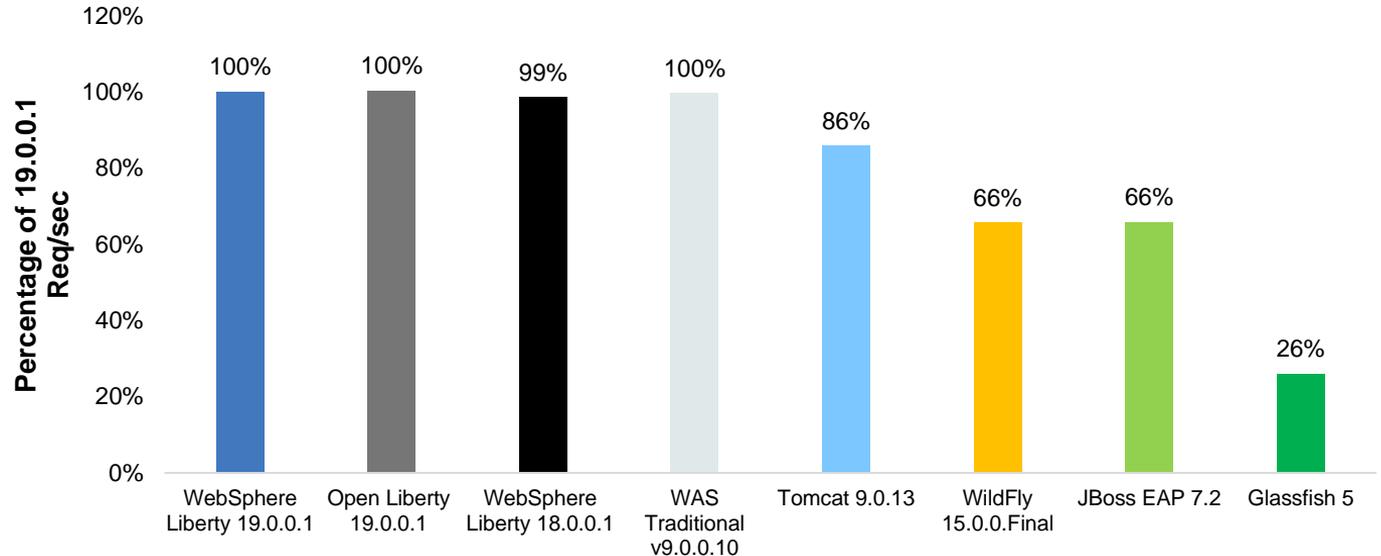
System Configuration:

- **SUT:** LinTel – SLES 11.4, Intel(R) Xeon(R) Platinum 8180 CPU @ 2.50GHz, 4 physical cores, 64GB RAM.
- Oracle JDK 8 u191 is used for non-IBM app servers & IBM JDK 8 SR5 FP30 is used for WebSphere Liberty 19.0.0.1 and IBM JDK SR5 FP17 for 18.0.0.1. IBM JDK 8 SR5 FP25 for WAS Traditional 9.0.0.10

Simple EE7 Throughput (Tradelite7)

- WAS 19.0.0.1 Liberty Server EE7 throughput is at least 14% better than other Lightweight App Servers

Throughput Comparison of Lightweight Servers - tradelite7 app
(higher is better)



System Configuration:

• **SUT:** LinTel – SLES 11.4, Intel(R) Xeon(R) Platinum 8180 CPU @ 2.50GHz, 4 physical cores, 64GB RAM.

• Oracle JDK 8 u191 is used for non-IBM app servers & IBM JDK 8 SR5 FP30 is used for WebSphere Liberty 19.0.0.1 and IBM JDK SR5 FP17 for 18.0.0.1. IBM JDK 8 SR5 FP25 for WAS Traditional 9.0.0.10

Liberty Quarterly
Update

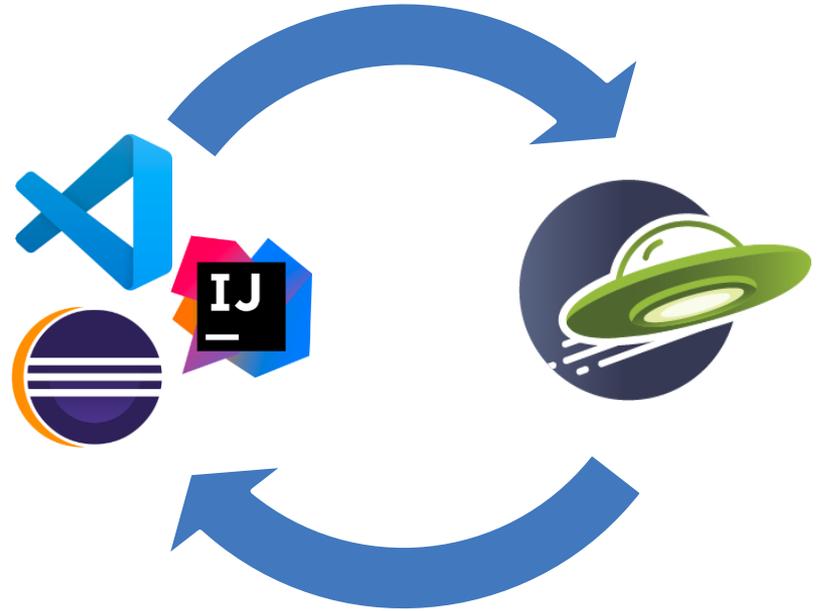
Developer Experience



Developer Experience: dev mode

- Boosts developer productivity
- Immediate feedback for code and config changes
- No re-build necessary

mvn liberty:dev



MicroShed Testing

- Integration tests that are easy to setup, write, and run
- Test your apps the same way they run in production...in Containers
- Can run multiple containers on same network (e.g. test DB integration)
- <https://microshed.github.io/>

```
@MicroShedTest
public class MyTest {

    // Search for Dockerfile.
    // Start app in Container.
    // Wait for Container before running tests.
    @Container
    public static MicroProfileApplication app
        = new MicroProfileApplication()
            .withAppContextRoot("/myservice");

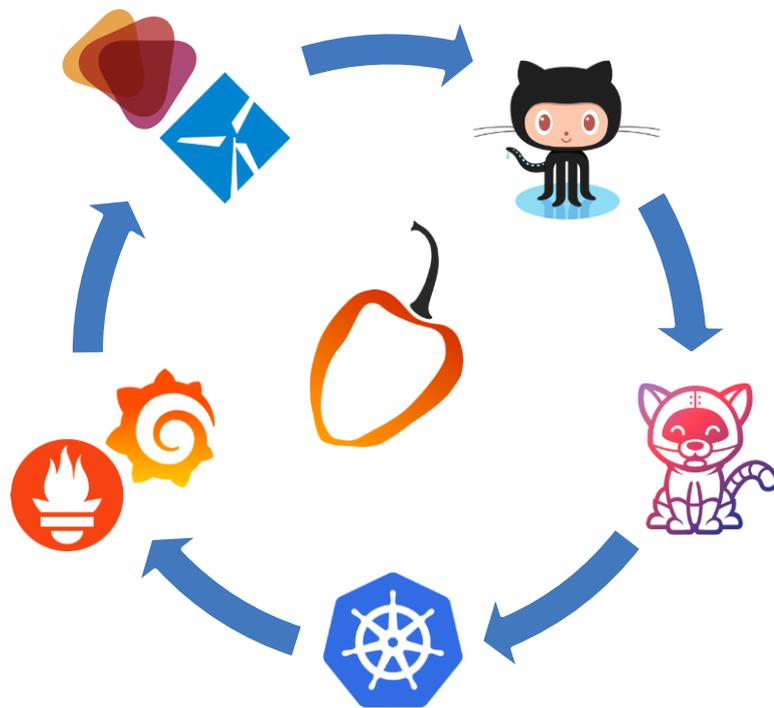
    // Inject JAX-RS REST Client
    @Inject
    public static MyService mySvc;

    // A test method like any other
    @Test
    public void testMyService() {
        ...
    }
}
```

Developer Experience: Kabanero

- Cloud-native container-based development
- Targeting cloud-native programming models and runtimes
- With an end-to-end cloud-native developer experience
- Targets include MicroProfile on Open Liberty

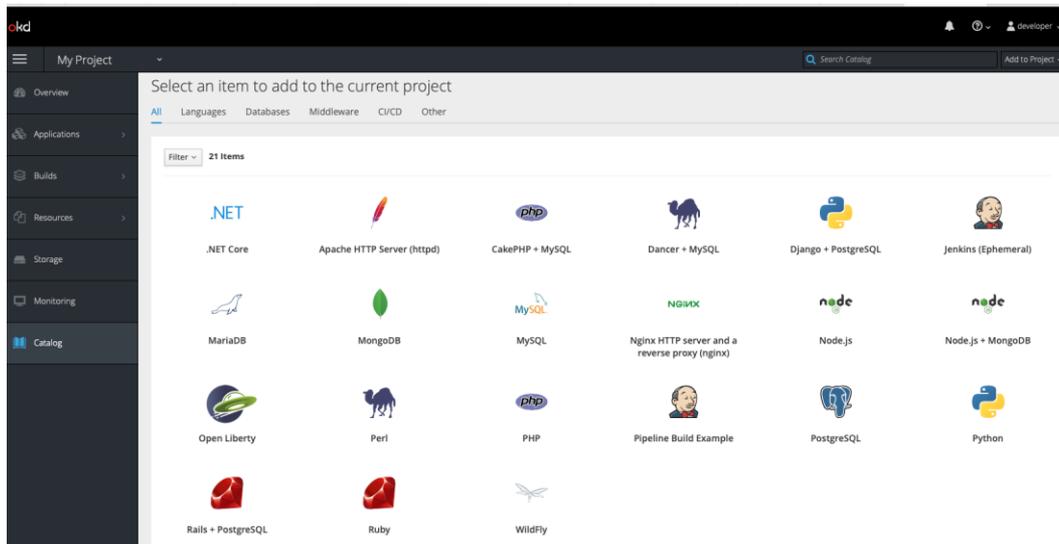
appsody init incubator/java-microprofile



Liberty Quarterly Update

Open Liberty source 2 image

- Builds a maven project to an Open Liberty docker image
- CI pipeline to build from github repository to Open Liberty container in Open Shift
- Use s2i command line for local experience

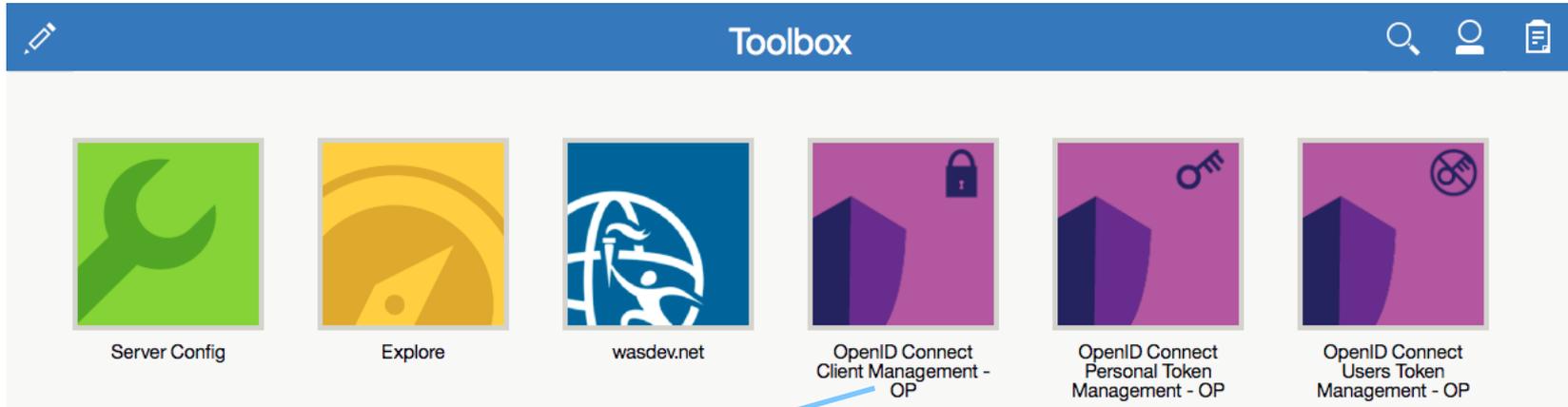


Liberty Quarterly
Update

Security

IBM

OpenID Connect App Password UI



Manage OAuth Clients

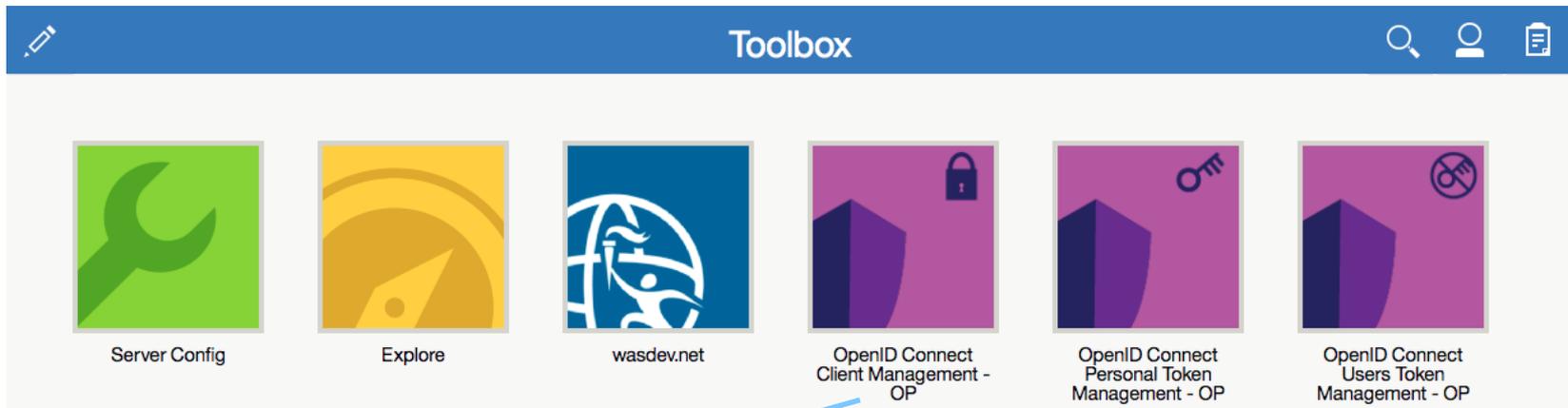
Use this tool to add and edit clients and regenerate client secrets.

[Add New](#)

Client name	Client ID		
DevOps Tool CLI	f3472cb43bd14d2fbd6fe272fdabc72a	Edit	Delete
DB CLI	125b2afdcf9f4fd5817e89fe985983e0	Edit	Delete
Custom Tool CLI	f8eeba02ced42e9b05c72767815c75d	Edit	Delete

1 of 1 pages < 1 >

OpenID Connect App Password UI



Manage OAuth Clients

Use this tool to add and edit clients and regenerate client secrets.

Client name	Client ID
DevOps Tool CLI	f3472
DB CLI	125b
Custom Tool CLI	f8eeb

Manage Personal Tokens

Create, delete and regenerate app-passwords and app-tokens.

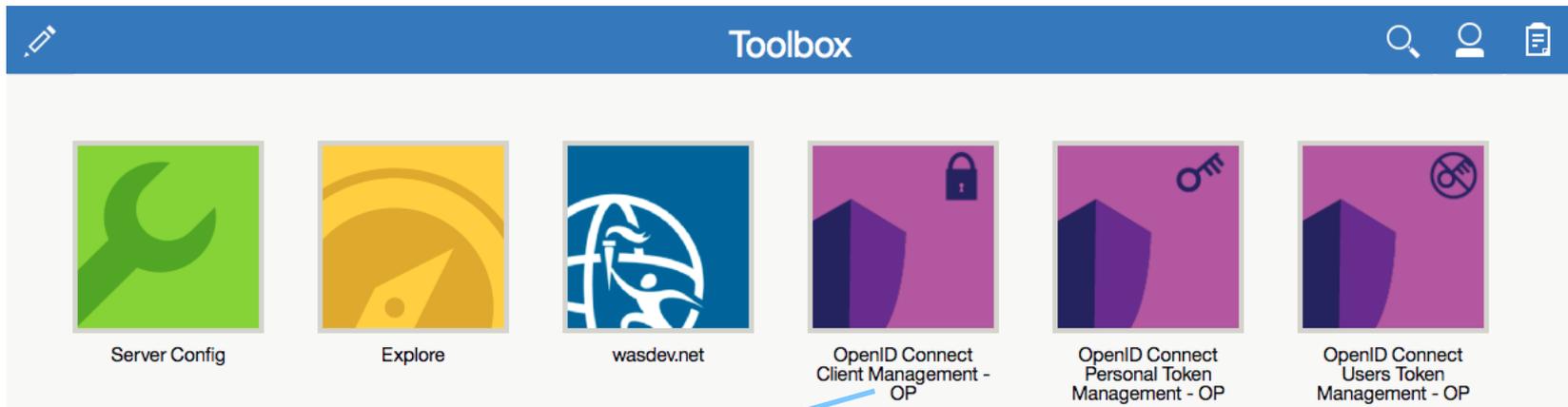
Add New +

Name	Type	Issued On	Expires On		
DB CLI	app-password	August 23, 2019	November 21, 2019	Regenerate	Delete
Custom Tool CLI	app-token	August 23, 2019	November 21, 2019	Regenerate	Delete
DevOps Tool CLI	app-token	August 23, 2019	November 21, 2019	Regenerate	Delete

1 of 1 pages < 1 >

1 of 1 pages < 1 >

OpenID Connect App Password UI



Manage OAuth Clients

Use this tool to add and edit clients and regenerate client secrets.

Client name	Client ID
DevOps Tool CLI	f3472
DB CLI	125b
Custom Tool CLI	f8eeb

Manage Personal Tokens

Create, delete and regenerate app-passwords and app-tokens.

Name	Type	Issued On
DB CLI	app-password	August 23, 2019
Custom Tool CLI	app-token	August 23, 2019
DevOps Tool CLI	app-token	August 23, 2019

1 of 1 pages < 1

Delete Tokens

Delete app-passwords and app-tokens for a specified user.

Q user1 x

0 Items selected Delete

<input type="checkbox"/>	Name	Client Name	Type	Issued On	Expires On	Delete
<input type="checkbox"/>	Custom Tool CLI	*	app-token	August 23, 2019	November 21, 2019	Delete
<input type="checkbox"/>	DevOps Tool CLI	*	app-token	August 23, 2019	November 21, 2019	Delete
<input type="checkbox"/>	DB CLI	*	app-password	August 23, 2019	November 21, 2019	Delete

1 of 1 pages < 1 >

Liberty Quarterly
Update

Foundation



PostgreSQL

```
<server>

  <dataSource jndiName="jdbc/postgres">
    <jdbcDriver libraryRef="PostgresLib" />
    <properties.postgresql serverName="db.mycompany.com"
      portNumber="5432" databaseName="sampledb"
      user="user1" password="pwd1"/>
  </dataSource>

  <library id="PostgresLib">
    <file name="${server.config.dir}/jdbc/postgresql-42.2.8.jar"/>
  </library>
</server>
```

Config REST API (read only)

- Access server config via REST API

```
/ibm/api/config/<elementName>/<uid>
```

```
<server>
```

```
<library id="derby">
```

```
<file name="${server.config.dir}/derby/derby.jar"/>
```

```
</library>
```

```
<dataSource id="DefaultDataSource">
```

```
<jdbcDriver libraryRef="derby"/>
```

```
<properties.derby.embedded
```

```
  databaseName="memory:defaultdb"
```

```
  createDatabase="create" user="dbuser"
```

```
  password="dbpass"/>
```

```
</dataSource>
```

```
</server>
```

```
{
  "configElementName": "dataSource",
  "uid": "DefaultDataSource",
  "id": "DefaultDataSource",
  "beginTranForResultSetScrollingAPIs": true,
  "beginTranForVendorAPIs": true,
  "connectionSharing": "MatchOriginalRequest",
  "enableConnectionCasting": false,
  "jdbcDriverRef": {
    {
      "configElementName": "jdbcDriver",
      "uid": "dataSource[DefaultDataSource]/jdbcDriver[default-0]",
      "libraryRef": {
        {
          "configElementName": "library",
          "uid": "derby",
          "id": "derby",
          "apiTypeVisibility": "spec,ibm-api,stable",
          "fileRef": {
            {
              "configElementName": "file",
              "uid": "library[derby]/file[default-0]",
              "name": "/home/mmittles/git/liberty/open-liberty/dev/build.image/wlp/usr/servers/blog/derby/derby.jar"
            }
          }
        }
      }
    }
  },
  "statementCacheSize": 10,
  "syncQueryTimeoutWithTransactionTimeout": false,
  "transactional": true,
  "properties.derby.embedded": {
    {
      "createDatabase": "create",
      "databaseName": "memory:defaultdb",
      "password": "*****",
      "user": "dbuser"
    }
  },
  "api": [
    "/ibm/api/validation/dataSource/DefaultDataSource"
  ]
}
```

Config Validation REST API

- Validate configuration for JDBC, JMS JCA, and Cloudant

```
/ibm/api/validation/<elementName>/<uid>
```

```
<server>

  <library id="derby">
    <file name="{server.config.dir}/derby/derby.</file>
  </library>

  <dataSource id="DefaultDataSource">
    <jdbcDriver libraryRef="derby"/>
    <properties.derby.embedded
      databaseName="memory:defaultdb"
      createDatabase="create" user="dbuser"
      password="dbpass"/>
  </dataSource>
</server>
```

```
{
  "uid": "DefaultDataSource",
  "id": "DefaultDataSource",
  "successful": true,
  "info": {
    "databaseProductName": "Apache Derby",
    "databaseProductVersion": "10.11.1.1 - (1616546)",
    "jdbcDriverName": "Apache Derby Embedded JDBC Driver",
    "jdbcDriverVersion": "10.11.1.1 - (1616546)",
    "schema": "DBUSER",
    "user": "dbuser"
  }
}
```

Liberty Quarterly
Update

Labs, Q&A

IBM

WebSphere Liberty Virtual POT

- Download the operating system specific content zip file from either <https://ibm.box.com/WASLibertyVPoT> (fast - about 10 minute download) or <https://public.dhe.ibm.com/ibmdl/export/pub/software/websphere/wasdev/pot/> (slower but firewall friendly – about 1 hour download)



VirtualPOT_19.0.0.7-9.final.pdf



LibertyPoT_19.0.0.9_WIN.zip V2



LibertyPoT_19.0.0.9_LINUX.zip V2



LibertyPoT_19.0.0.9_MAC.zip V2



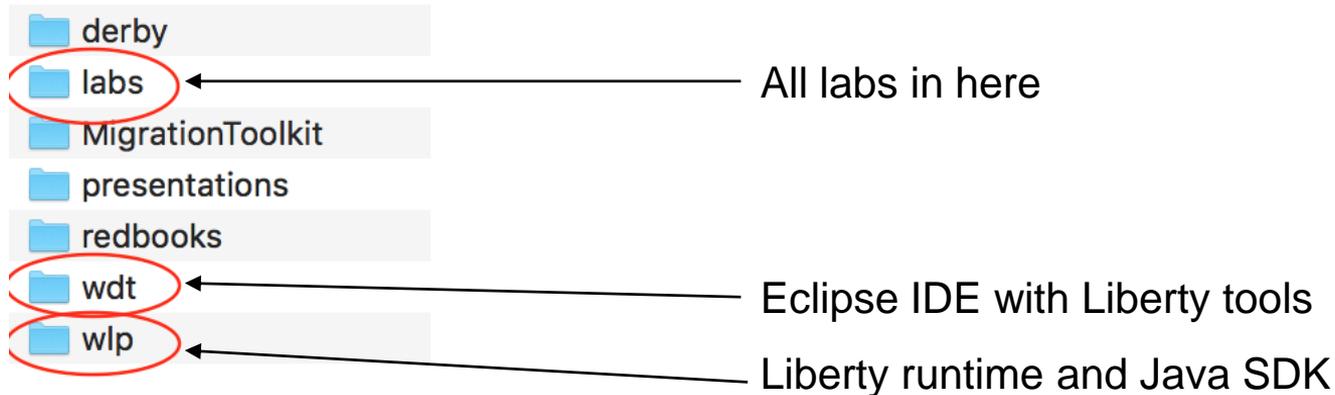
labs_n_presentations_only.zip V9

charts only

charts & lab
instructions only

WebSphere Liberty Virtual POT

- Unzip to C:\wlp_pot
 - Note: You can unzip to anywhere you wish but the lab instructions assume the unzip location is C:\wlp_pot



- Follow [labs\gettingStarted\0_setup_20180105\setup.pdf](#)
- Then choose any labs you want to do

Resources

- Why Liberty Profile is the best Java runtime for the Cloud <https://developer.ibm.com/wasdev/docs/liberty-profile-best-java-runtime-cloud/>
- WebSphere traditional or WebSphere Liberty: how to choose? <https://developer.ibm.com/wasdev/docs/was-classic-or-was-liberty-how-to-choose/>
- The WebSphere Contrarian: If liberty means freedom of choice, am I free to choose Liberty?
http://www.ibm.com/developerworks/websphere/techjournal/1506_webcon/1506_webcon-trs.html
- OpenLiberty – An IBM Open Source Project <https://www.openliberty.io/>
- MicroProfile: <http://microprofile.io/>
- IBM has always been a huge supporter of open source - Top 20 companies according to the number of Apache Committers
www.linkedin.com/hp/update/6222828697750167552
- WebSphere Application Server Liberty Profile Guide for Developers (Redbook) <http://www.redbooks.ibm.com/abstracts/sg248076.html?Open>
- Configuring and Deploying Open Source with WebSphere Application Server Liberty Profile (Redbook)
<http://www.redbooks.ibm.com/abstracts/sg248194.html?Open>
- Not just any private cloud. This is IBM Cloud Private <https://www.ibm.com/cloud-computing/products/ibm-cloud-private/>
- WebSphere Application Server V8.5 Administration and Configuration Guide for Liberty Profile (Redbook)
<http://www.redbooks.ibm.com/abstracts/sg248170.html?Open>
- IBM Media Center - WebSphere and Liberty channel <https://mediacenter.ibm.com/channel/t/33964822>
- WASdev YouTube channel <https://www.youtube.com/c/wasdevnet>
- Liberty Documents on WASdev.net <https://developer.ibm.com/wasdev/docs/>
- Ask question about Liberty on StackOverflow: <http://stackoverflow.com/questions/tagged/websphere-liberty>

Resources (2)

- Opportunity to modernize your app infrastructure - choose your environment <http://whichwas.mybluemix.net>
- Increase ROI by 122 percent by migrating from open source <https://www.ibm.com/blogs/systems/increase-roi-122-percent-migrating-open-source/>
- Future strategy will use IBM Java dates <http://www.ibm.com/developerworks/java/jdk/lifecycle>
- WebSphere Application Server Migration Workshops <http://ibm.biz/webspheremigration>
- Single Stream Continuous Delivery <https://www-01.ibm.com/support/docview.wss?uid=ibm10869798>
- WebSphere Migration Knowledge Collection: Planning and Resources <https://www-01.ibm.com/support/docview.wss?uid=swg27008724>

Next Quarterly Update Planned Dates

- **Liberty 19.0.0.7-9 Update**

Session#1: ~~October 2, 2019 from 1-3pm ET - <http://ibm.biz/Liberty-Oct02>~~

Session#2: October 9, 2019 from 9-11am ET - <http://ibm.biz/Liberty-Oct09>

- **Liberty 19.0.0.10-12 Update**

Session#1: January 15, 2020 from 1-3pm ET - <http://ibm.biz/Liberty-Jan15>

Session#2: January 22, 2020 from 9-11am ET - <http://ibm.biz/Liberty-Jan22>

WebSphere & IBM Cloud Pak for Applications Customer Advisory Board

all customers welcome

<http://ibm.biz/WebSphereAdvisoryBoard>

email: claudiab@us.ibm.com

OPEN invitation

Join 140 other companies

Be part of customer round tables and deep dive meetings

New engagements tracks added.

Reach out to find out more!

Choose your engagement level:

1. **Fly on the wall – NEW****
2. **Stay ahead of the curve:** more time commitment
3. **Close the gap:** quarterly involvement
4. **At your own pace:** impact longer term goals



IBM meets you wherever you are - get involved and be successful.

Liberty Quarterly Update

Questions?

<http://stackoverflow.com/questions/tagged/websphere-liberty>

alasdair@us.ibm.com



Liberty Quarterly
Update

Thank You

Your Feedback is Important!

