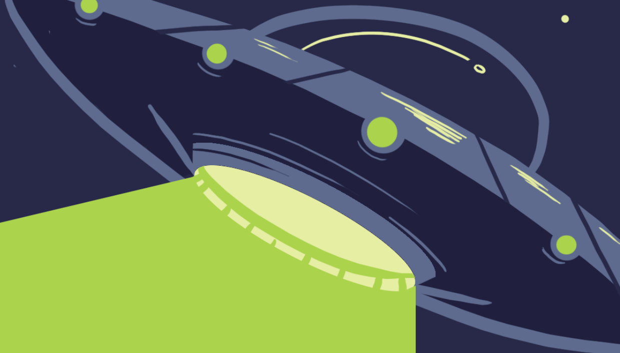
Open Liberty



Liberty Quarterly Update 22.0.0.12-22.0.0.3

Alasdair Nottingham – Liberty Lead Architect

🐦 @nottycode

Agenda

Part 1: 30 Minute Liberty overview

Part 2: What is new this quarter

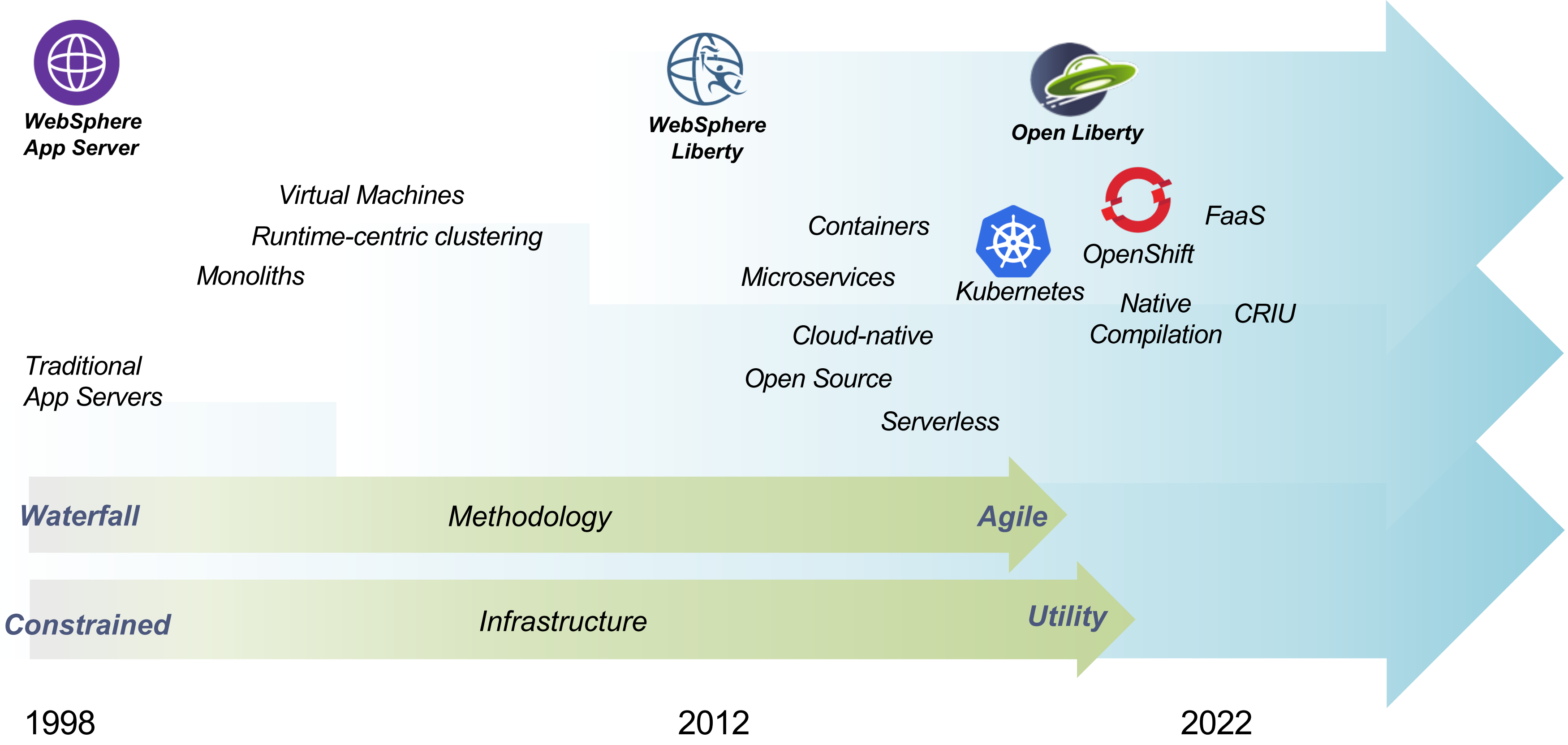
Part 3: Q&A

30 minute overview

Open Liberty

WebSphere Evolution

and the influence of cloud



WebSphere App Server

Virtual Machines

Runtime-centric clustering

Monoliths

Traditional App Servers

WebSphere Liberty

Containers

Microservices

Cloud-native

Open Source

Serverless

Open Liberty

Kubernetes

OpenShift

FaaS

Native Compilation

CRIU

Waterfall *Methodology* ***Agile***

Constrained *Infrastructure* ***Utility***

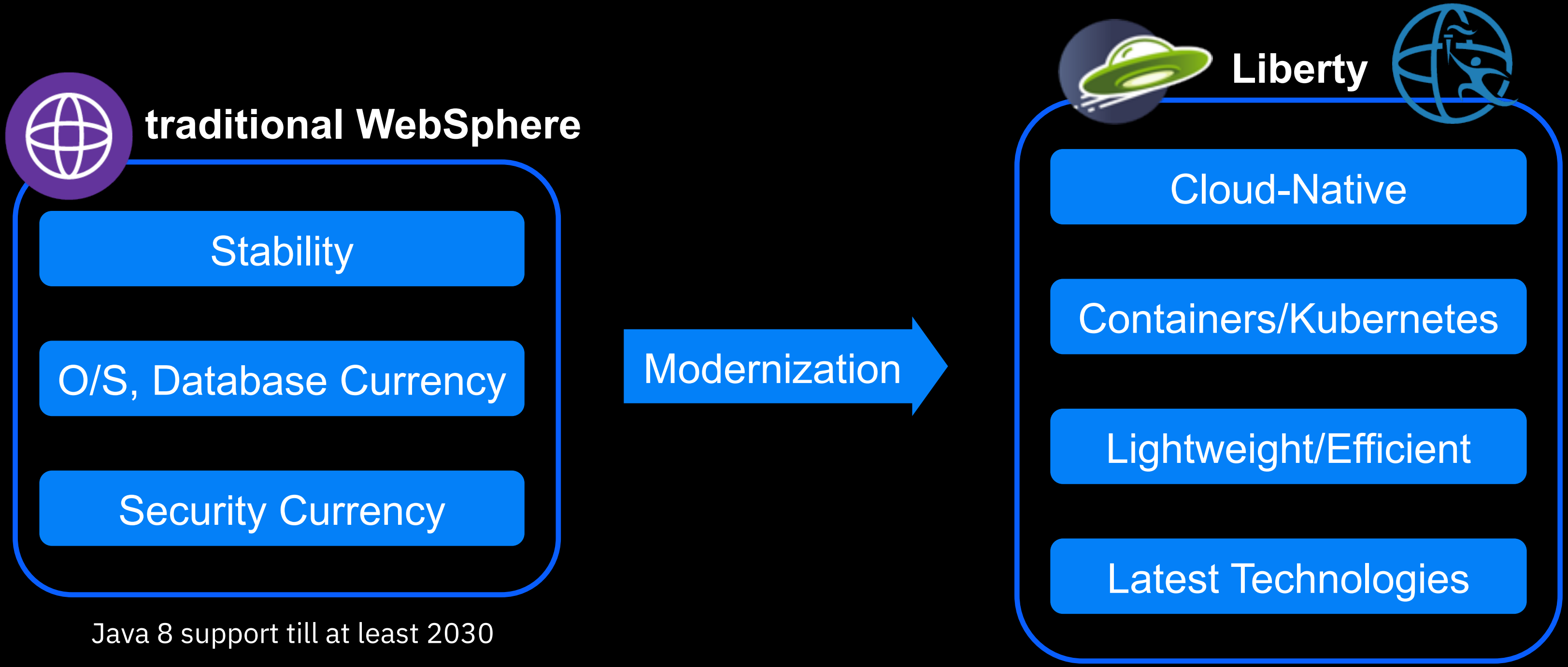
1998

2012

2022

WebSphere Runtimes

Support current and future runtime needs



traditional WebSphere

Stability

O/S, Database Currency

Security Currency

Modernization

Liberty

Cloud-Native

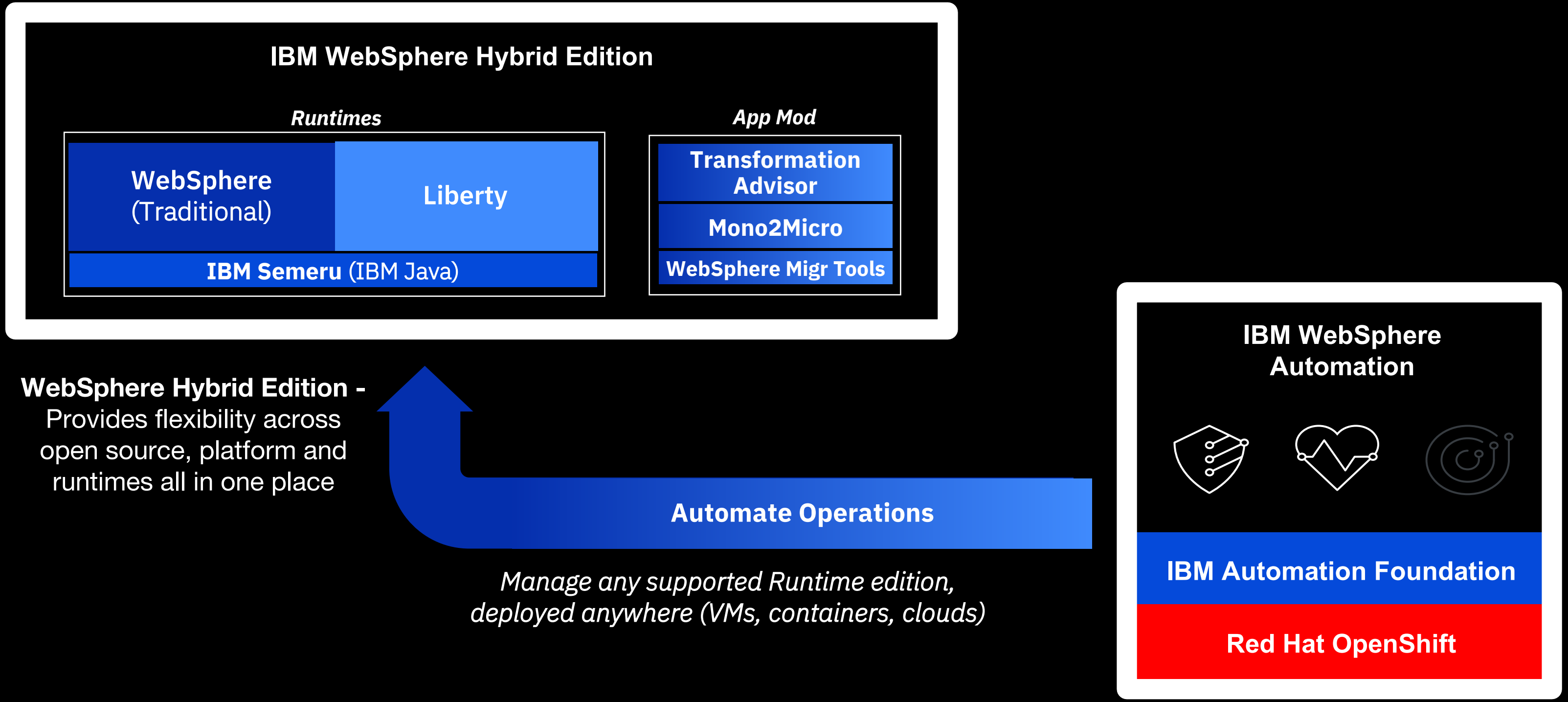
Containers/Kubernetes

Lightweight/Efficient

Latest Technologies

Java 8 support till at least 2030

Liberty, WebSphere and Java in 2022



WebSphere Hybrid Edition - Provides flexibility across open source, platform and runtimes all in one place

Manage any supported Runtime edition, deployed anywhere (VMs, containers, clouds)

AI-infused automation for WebSphere
Frees operations teams from routine "care and feeding" of WebSphere environments

IBM WebSphere Liberty

1 50% increase in developer productivity

2 40% increase in IT admin productivity

3 195% ROI Payback 8 months

Cloud-Native Development

- Simple rapid inner-loop developer experience in any IDE
- Optimized for Containers and Kubernetes
- Optimized for Continuous Integration, Continuous Delivery

Operational Optimization

- Reduce costs with world-leading performance for microservices and monoliths
- Auto-tuning for continuous optimal performance in any environment
- Simple operator-based management in Kubernetes

Application Modernization

- API & configuration compatibility for reduced effort and risk
- HA clustering in Containers and VMs
- Transformation Advisor & Mono2Micro tools help plan and execute move to container and microservices

https://ibm.biz/WSHE-TEI

Demo: Rapid, iterative development and deployment

1 50% increase in developer productivity

2 40% increase in IT admin productivity

1. Start a new microservice project
2. Develop the code
3. Build a container
4. Deploy to Azure Kubernetes Service

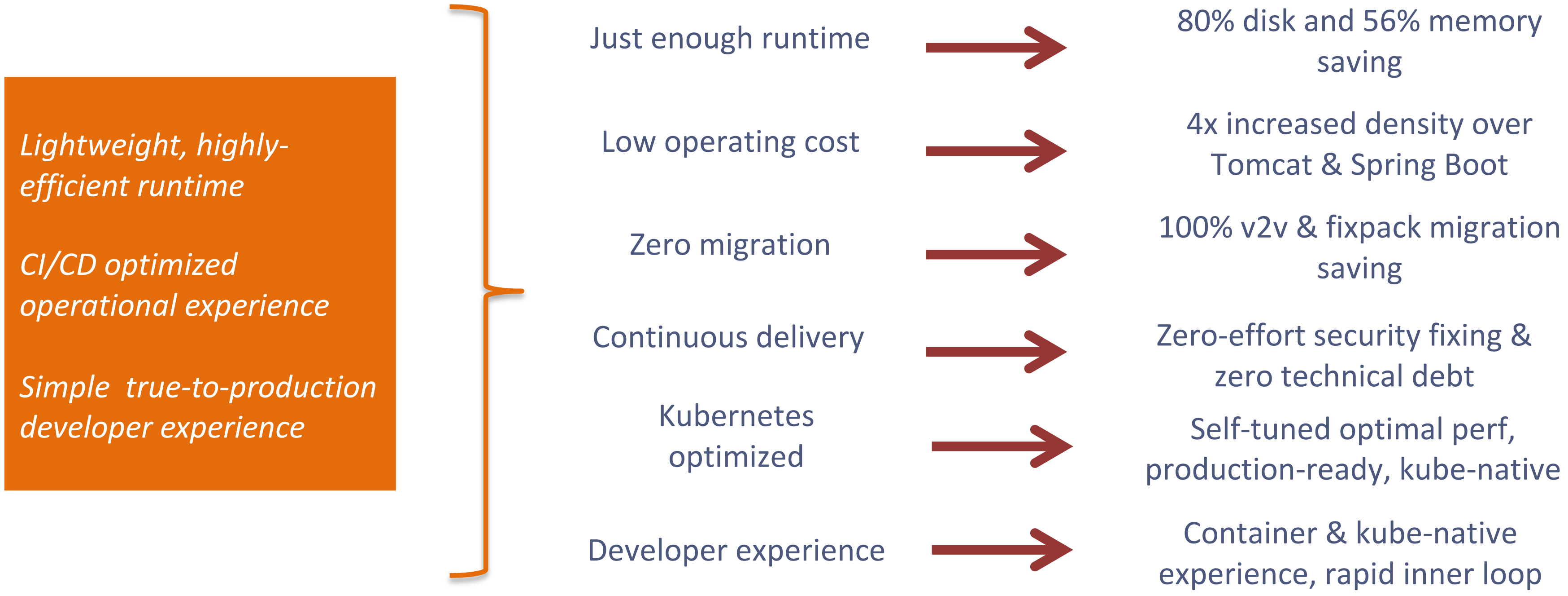
Cloud-Native Development

- Simple rapid inner-loop developer experience in any IDE
- Optimized for Containers and Kubernetes
- Optimized for Continuous Integration, Continuous Delivery

Operational Optimization

- Reduce costs with world-leading performance for microservices and monoliths
- Auto-tuning for continuous optimal performance in any environment
- Simple operator-based management in Kubernetes

6 reasons why Liberty



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

Zero migration → 100% v2v & fixpack migration saving

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

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

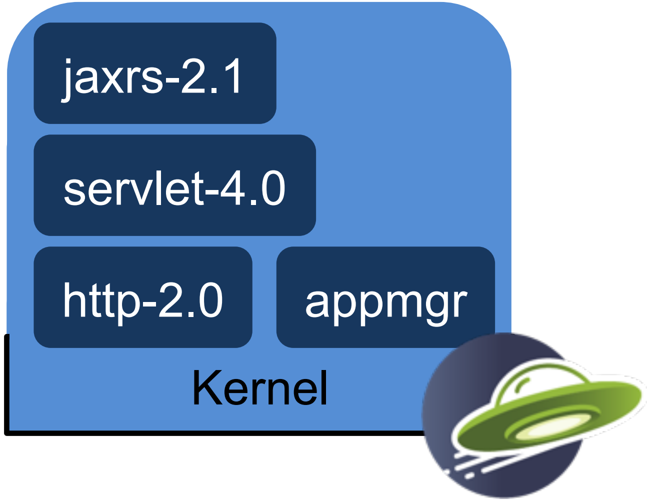
Developer experience → Container & kube-native experience, rapid inner loop

Just Enough Application Server

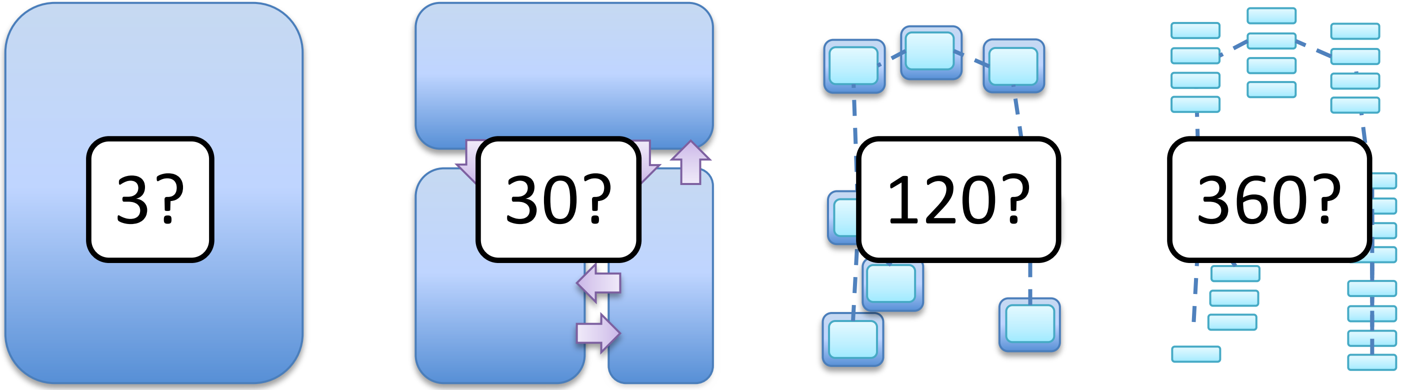
Open Liberty

You control which features are loaded into each server instance

```
<feature>jaxrs-2.1</feature>
```



Granularity cost implications



3?

30?

120?

360?

Disk?
Memory?
$$$?

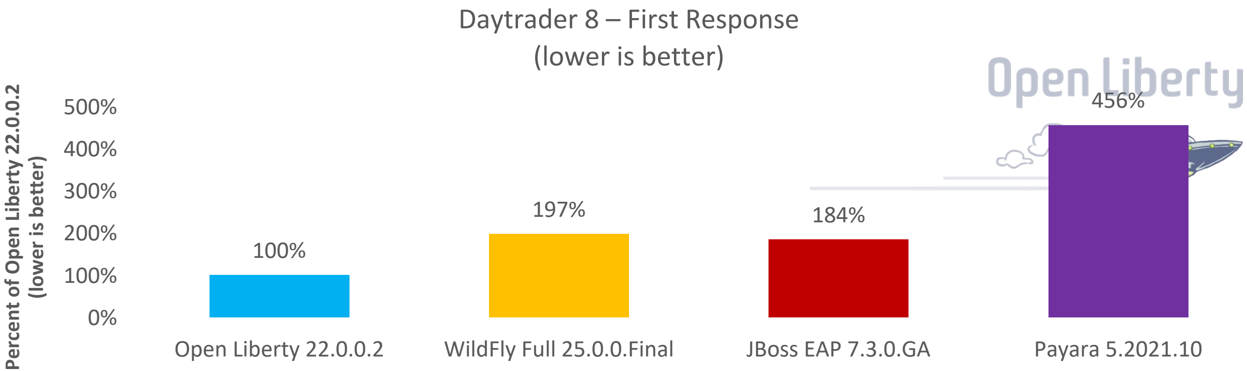
X	X	X	X
200MB	200MB	200MB	200MB
=	=	=	=
600MB	6GB	24GB	72GB

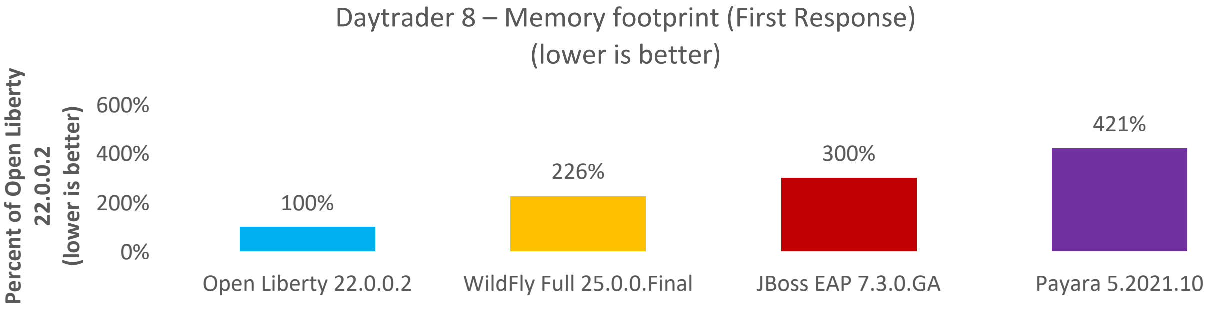
EE8 Performance (Daytrader8)

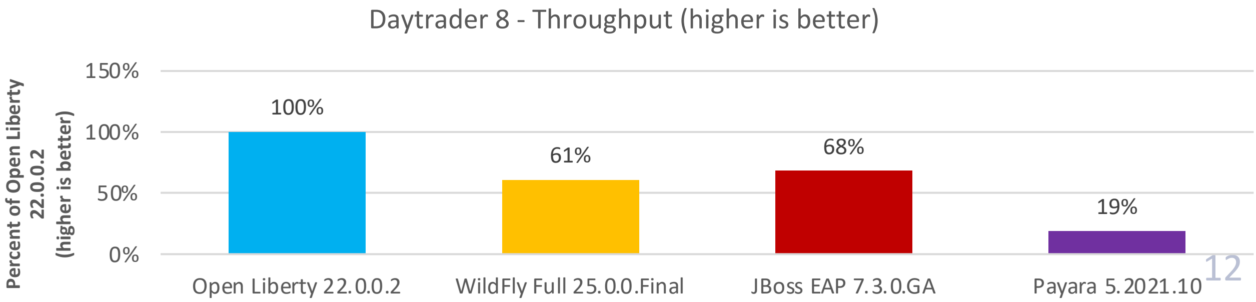
- Liberty outperforms others on all metrics for EE8 performance (startup time and memory footprint is almost half, throughput is 32% better)
- Comparisons used each application server’s Docker image

System Configuration:

***SUT**:* LinTel – SLES 12.4, Intel(R) Xeon(R) Platinum 8180 CPU @ 2.50GHz, 4 cpus, 4GB RAM.
JDK version distributed with the docker images used for each server instance.


Daytrader 8 – First Response (lower is better)


Daytrader 8 – Memory footprint (First Response) (lower is better)


Daytrader 8 - Throughput (higher is better)

Low Operating Cost

Modernization led to optimized resource usage by **75%**

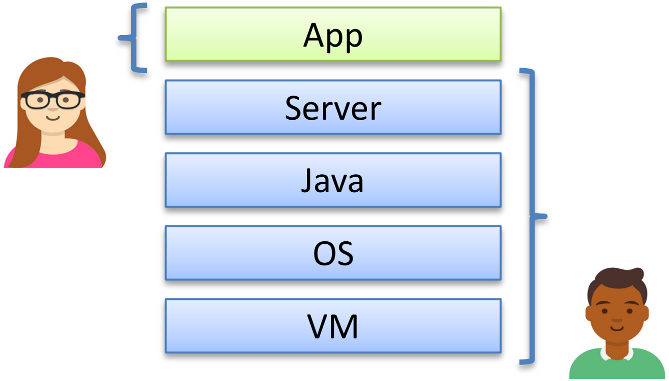
and reduced infrastructure footprint by **50%**

Major US healthcare provider

Cloud platforms shift responsibilities

Traditional Deployment

- I develop the app
- I give the Ops team a WAR file
- I *occasionally* update app dependencies

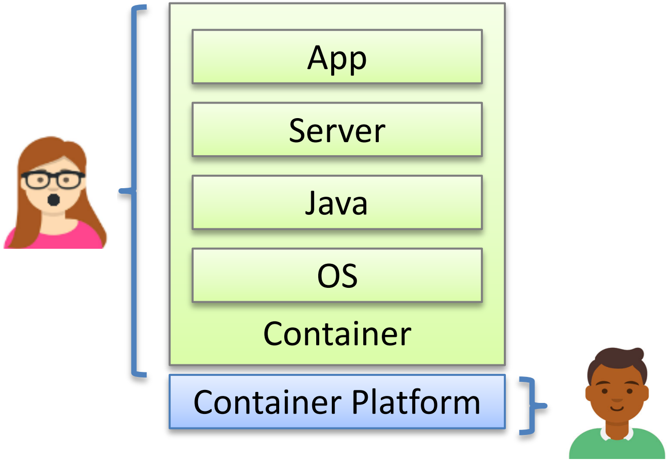


- I deploy the app
- I handle automation
- I monitor the app
- I maintain the infrastructure
- I apply important security fixes
- I plan and execute migrations

Cloud platforms shift responsibilities

Cloud-native Platform Deployment

- I develop the app
- I *occasionally* update app dependencies
- I deploy the app
- I handle automation
- I monitor the app
- I maintain the infrastructure
- I apply important security fixes
- I plan and execute migrations

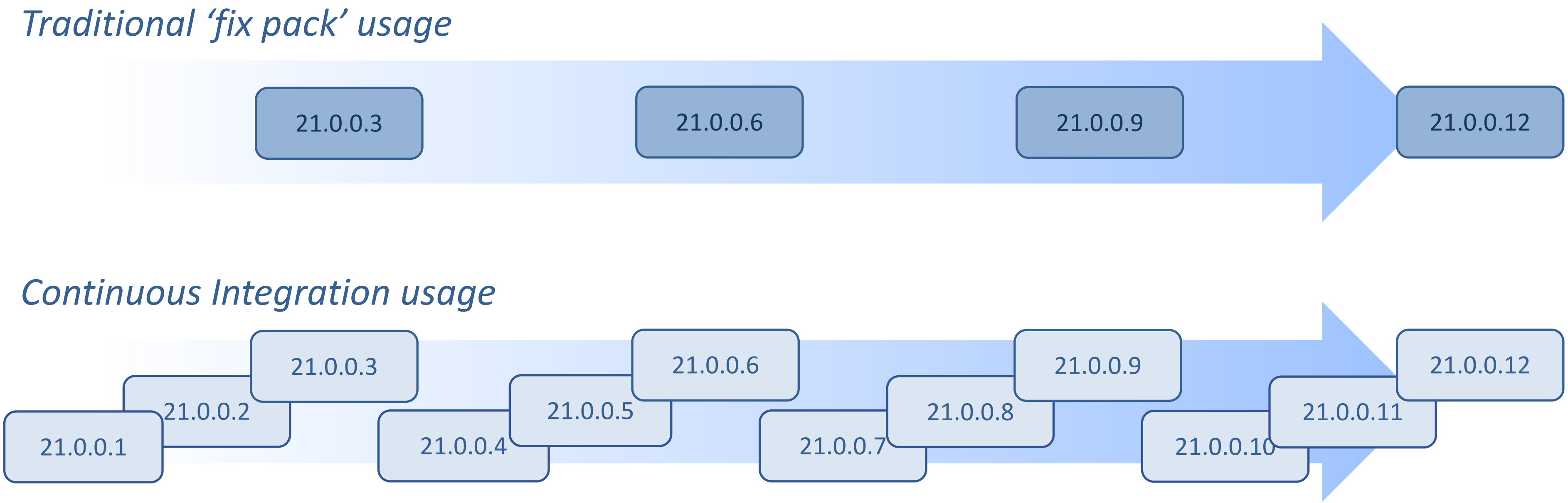


- I manage a cloud platform
- I provide services to the application teams

Liberty Release Cadence

Liberty's 'zero migration' architecture makes picking up a new release simple

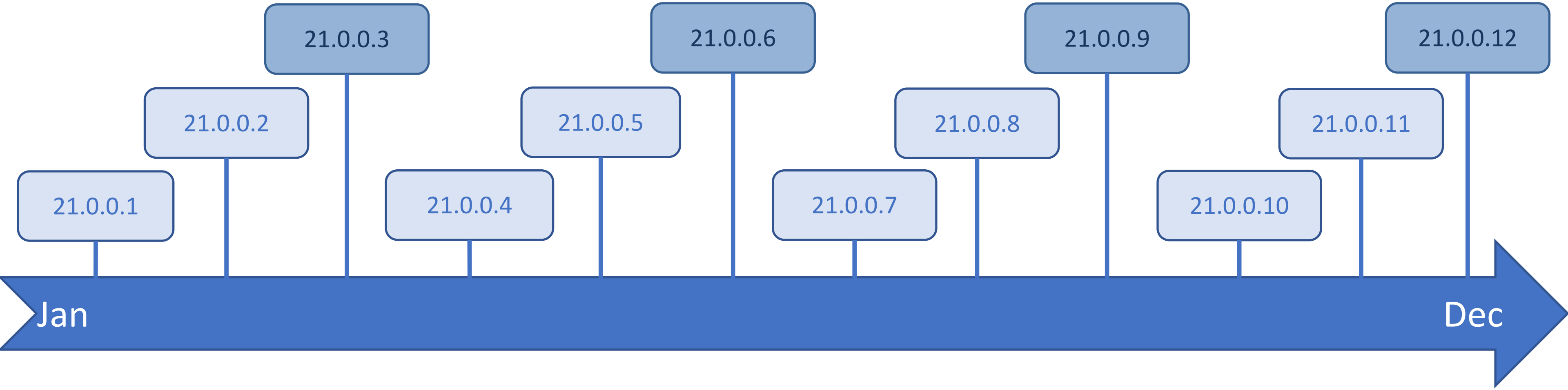
Skipping a release does not introduce migration work



Traditional 'fix pack' usage

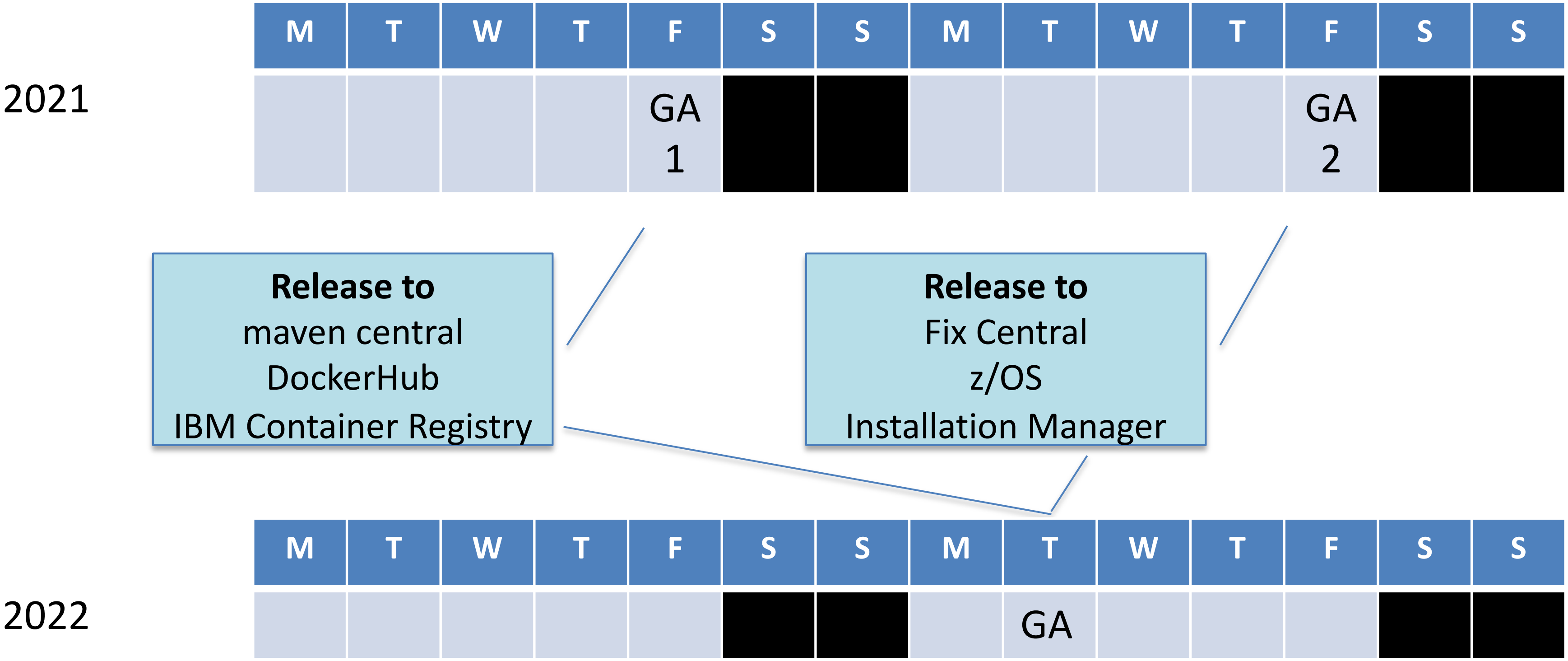
Continuous Integration usage

Liberty Release Cadence



	All CD releases	CD releases ending .3 .6 .9 .12
Support Provided	5 years	5 years
iFixes	24 weeks	2 years
Proactive Security iFixes	Most recent	Most recent 2

Update to release day 2022



	M	T	W	T	F	S	S	M	T	W	T	F	S	S
2021					GA 1	■	■					GA 2	■	■

Release to
maven central
DockerHub
IBM Container Registry

Release to
Fix Central
z/OS
Installation Manager

	M	T	W	T	F	S	S	M	T	W	T	F	S	S
2022						■	■		GA				■	■

Zero Migration

Open Liberty

- ✓No configuration behavior changes
- ✓No runtime feature behavior changes
- ✓No removals

Stay current with a rebuild
(no app or config changes necessary)

Skipping a release does not introduce additional migration work

Never apply an ifix again

Zero Migration

Today we migrated all our Liberty servers config from EE7 to EE8. This process normally take 18 months in traditional WebSphere, cannot say how many manhours exactly.

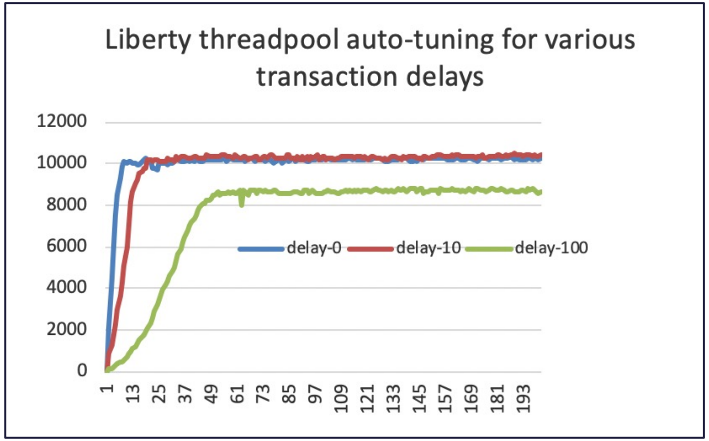
Today it took 18 minutes, with Liberty's continuous delivery stream.

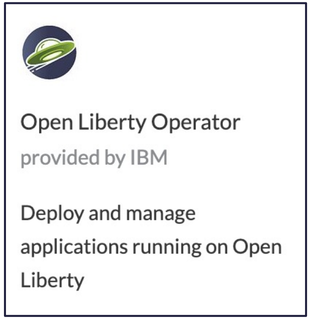
In July all apps starting to use EE8 framework.

Henrik Lundström, WAS Systems Administrator, Handelsbanken (Sweden)

43,800x improvement

Kubernetes optimized




Open Liberty Operator

provided by IBM

Deploy and manage applications running on Open Liberty

docker hub

MICROPROFILE™
OPTIMIZING ENTERPRISE JAVA

- **Deliver faster** without costly tuning exercises
- Get **optimal performance** even as the environment changes

- **Simple Operator-based deploy** and day-2 operations experience
- Supported **production-ready images**
- **APIs** for Kubernetes integration
- Container-based **usage tracking**

Kubernetes Optimized

“You don't have to tune thread pools. Liberty does an outstanding job”

WebSphere Technology Owner
Large health provider

Developer experience

Open Liberty

IDEs

Dev Mode

Repositories

The Central Repository

docker hub

Build

Maven™

Gradle

APIs

MICROPROFILE™

Java EE™

JAKARTA EE

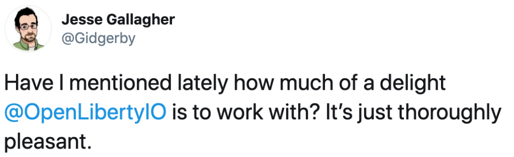
Spring Boot®

Testing

microshed testing

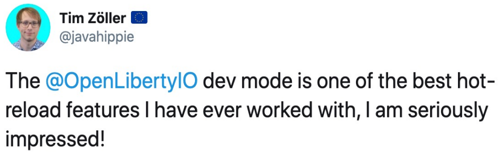
JUnit

arquillian



Jesse Gallagher
@Gidgerby

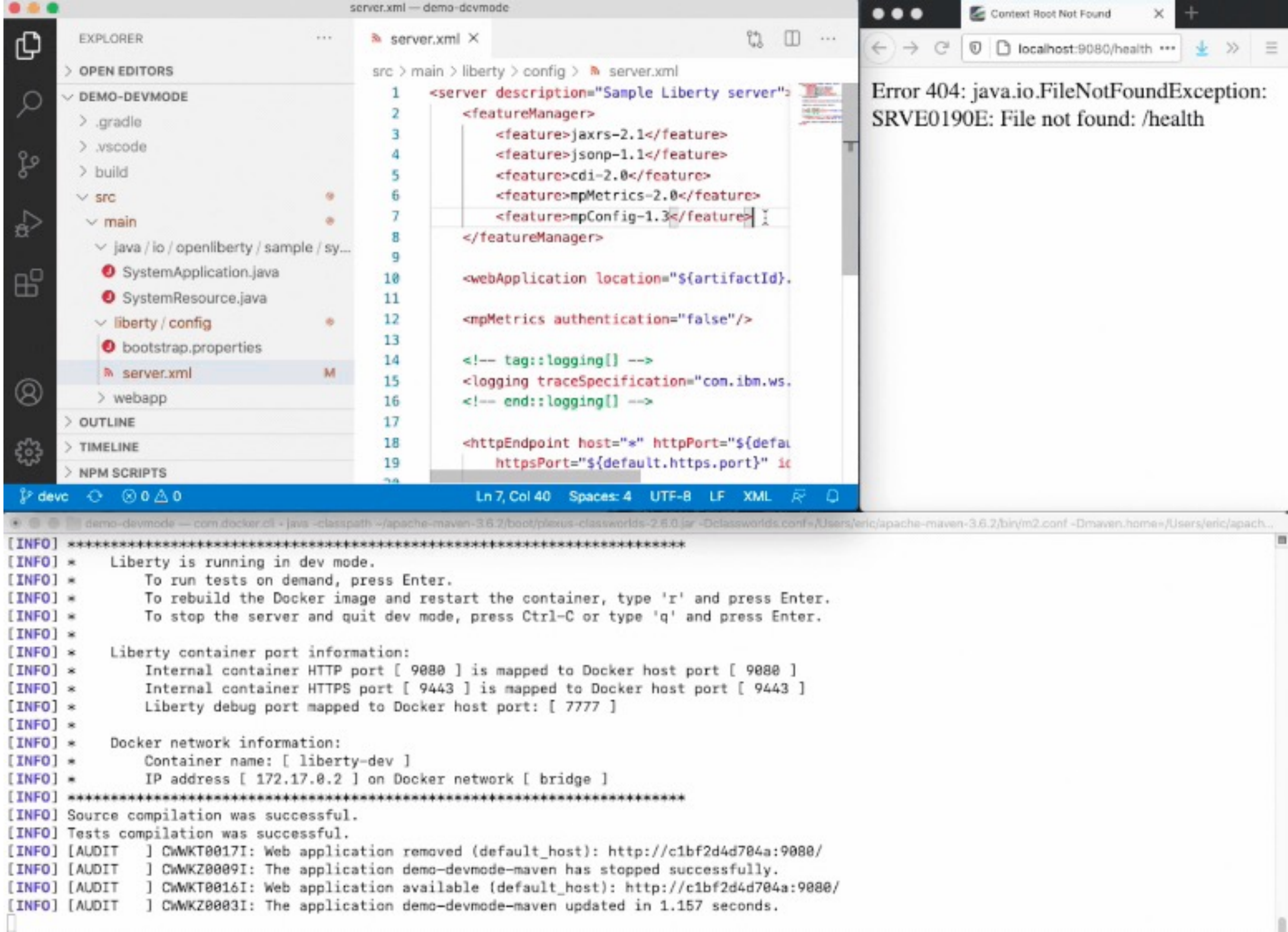
Have I mentioned lately how much of a delight @OpenLibertyIO is to work with? It's just thoroughly pleasant.



Tim Zöller 🇪🇺
@javahippie

The @OpenLibertyIO dev mode is one of the best hot-reload features I have ever worked with, I am seriously impressed!

Dev mode in action



```
<server description="Sample Liberty server">
    <featureManager>
        <feature>jaxrs-2.1</feature>
        <feature>jsonp-1.1</feature>
        <feature>cdi-2.0</feature>
        <feature>mpMetrics-2.0</feature>
        <feature>mpConfig-1.3</feature>
    </featureManager>

    <webApplication location="${artifactId}.

    <mpMetrics authentication="false"/>

    <!-- tag::logging[] -->
    <logging traceSpecification="com.ibm.ws.
    <!-- end::logging[] -->

    <httpEndpoint host="*" httpPort="${defau
        httpsPort="${default.https.port}" id
```

Error 404: java.io.FileNotFoundException: SRVE0190E: File not found: /health

```
demo-devmode — com.docker.cli ‹ java -classpath ~/apache-maven-3.6.2/boot/plexus-classworlds-2.6.0.jar -Dclassworlds.conf=/Users/eric/apache-maven-3.6.2/bin/m2.conf -Dmaven.home=/Users/eric/apach...
[INFO] ************************************************************************
[INFO] *    Liberty is running in dev mode.
[INFO] *        To run tests on demand, press Enter.
[INFO] *        To rebuild the Docker image and restart the container, type 'r' and press Enter.
[INFO] *        To stop the server and quit dev mode, press Ctrl-C or type 'q' and press Enter.
[INFO] *
[INFO] *    Liberty container port information:
[INFO] *        Internal container HTTP port [ 9080 ] is mapped to Docker host port [ 9080 ]
[INFO] *        Internal container HTTPS port [ 9443 ] is mapped to Docker host port [ 9443 ]
[INFO] *        Liberty debug port mapped to Docker host port: [ 7777 ]
[INFO] *
[INFO] *    Docker network information:
[INFO] *        Container name: [ liberty-dev ]
[INFO] *        IP address [ 172.17.0.2 ] on Docker network [ bridge ]
[INFO] ************************************************************************
[INFO] Source compilation was successful.
[INFO] Tests compilation was successful.
[INFO] [AUDIT   ] CWWKT0017I: Web application removed (default_host): http://c1bf2d4d704a:9080/
[INFO] [AUDIT   ] CWWKZ0009I: The application demo-devmode-maven has stopped successfully.
[INFO] [AUDIT   ] CWWKT0016I: Web application available (default_host): http://c1bf2d4d704a:9080/
[INFO] [AUDIT   ] CWWKZ0003I: The application demo-devmode-maven updated in 1.157 seconds.
```

How to get Support

Open Liberty

WebSphere

z/OS

ND

Base

Core

WebSphere Hybrid Edition

IBM Integrated Application Runtimes

Java:

- WebSphere
- Liberty
- MicroProfile
- Jakarta EE
- OpenJ9

Cloud Foundry Migration Runtime

Transformation Advisor

Mono2micro

Red Hat OpenShift

Recent Updates

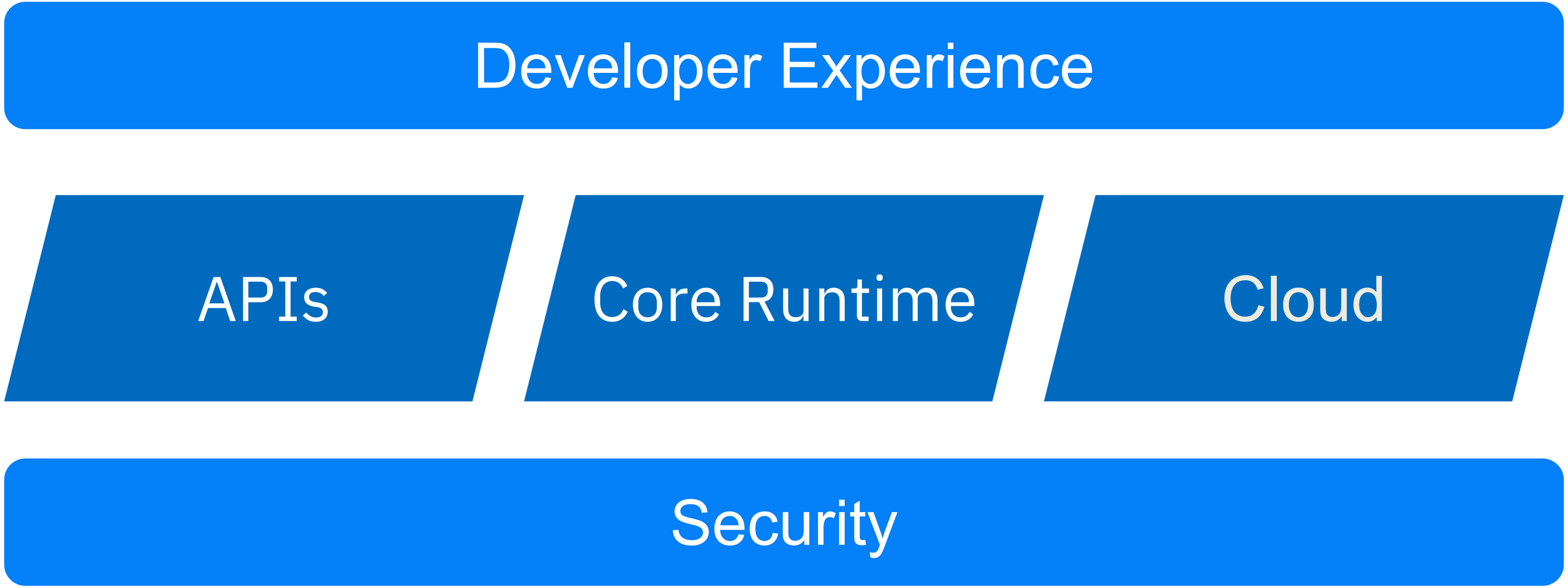
Periodic Table of Liberty (22.0.0.3)

- zOS
- ND
- Base
- Core
- Open ● Liberty
- 🟥 New in 4Q21
- 🟪 New in 3Q21
- 🟨 New in 2Q21
- 🟩 New in 1Q22

	batchSMFLogging-1.0	zosLocalAdapters-1.0	zosTransaction-1.0	
		zosRequestLogging-1.0	zosWlm-1.0	zosSecurity-1.0
collectiveController-1.0	dynamicRouting-1.0	healthManager-1.0	scalingController-1.0	Security
	clusterMember-1.0	healthAnalyzer-1.0	scalingMember-1.0	
cloudant-1.0 ●	jakartaee-8.0 ●	batchManagement-1.0 ●	Operations	acmeCA-1.0 ●
javaee-8.0 ●	🟩 heritageAPIs-1.1	wsAtomicTransaction-1.2 ●		passwordUtilities-1.0 ●
javaee-7.0 ●	sipServlet-1.1 ●			wsSecurity-1.1 ●
🟥 jakartaee-9.1 ●				wsSecuritySaml-1.0 ●
bells-1.0 ●	mpGraphQL-1.0 ●	adminCenter-1.0 ●	audit-1.0 ●	ldapRegistry-3.0 ●
🟥 concurrent-2.0 ●	mpReactiveMessaging-1.0 ●	collectiveMember-1.0	constrainedDelegation-1.0 ●	oauth-2.0 ●
🟥 facesContainer-3.0 ●	mpReactiveStreams-1.0 ●	distributedMap-1.0 ●	federatedRepository-1.0 ●	openid-2.0 ●
grpc-1.0 ●	opentracing-1.3 ●	eventLogging-1.0 ●	jwt-1.0 ●	openidConnectClient-1.0 ●
jdbc-4.3 ●	osgiConsole-1.0 ●	logstashCollector-1.0 ●	jwtSso-1.0 ●	openidConnectServer-1.0 ●
json-1.0 ●	🟥 persistenceContainer-3.0 ●	monitor-1.0 ●	sessionDatabase-1.0 ●	samlWeb-2.0 ●
🟥 jsonbContainer-2.0 ●	springBoot-2.0 ●	openapi-3.1 ●	webCache-1.0 ●	scim-1.0
🟥 jsonpContainer-2.0 ●	🟥 webProfile-9.1 ●	requestTiming-1.0 ●		socialLogin-1.0 ●
🟥 mail-2.0 ●	webProfile-8.0 ●	usageMetering-1.0		spnego-1.0 ●
🟩 microProfile-5.0 ●	webProfile-7.0 ●	restConnector-2.0 ●		transportSecurity-1.0 ●
🟩 mpContextPropagation-1.3 ●	🟥 xmlBinding-3.0 ●	sessionCache-1.0 ●		

APIs

Focus areas



Developer Experience

APIs

Core Runtime

Cloud

Security

Liberty Last Quarter Review

Security

- JWE OIDC Support
- Sign Liberty Binaries

Dev Exp

- JAX-RPC Conversion tool

Core Runtime

- Ignore Affinity Requests plugin configuration
- Variable expansion in server.env
- Set working dir
- mpOpenAPI supports multiple applications
- JDBC configure exceptions

API

- MicroProfile 5
- CommonJ WorkManager

Cloud

Variable Expansion in server.env

- Variable expansion in server.env was not allowed so in:

```
hostname=a.b.com
url=http://${hostName}:9080
```

- url resolved to

```
http://${hostName}:9080
```

- instead of

```
http://a.b.com:9080
```

Variable Expansion in server.env

- Variable expansion in server.env was not allowed so in:

```
#enable_variable_expansion
hostname=a.b.com
url=http://${hostName}:9080
```

- url resolves to

```
http://a.b.com:9080
```

- This is unix specific so not cross platform

Variable Expansion in server.env

- Variable expansion in server.env on windows allows:

```
#enable_variable_expansion
hostname=a.b.com
url=http://!hostname!:9080
```

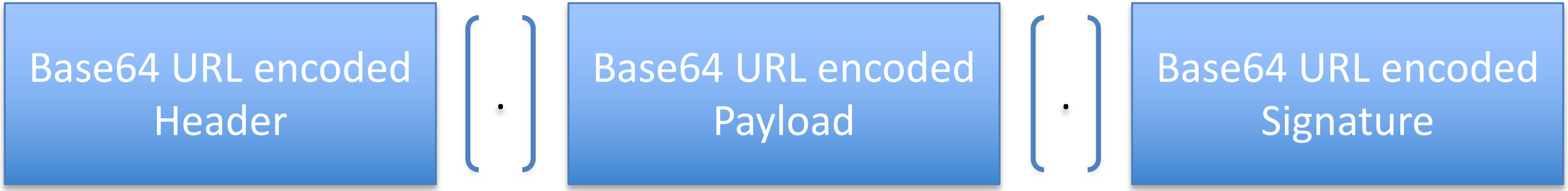
- url resolves to

```
http://a.b.com:9080
```

- This is windows specific so not cross platform

JSON Web Token

JSON Web Signature (JWS)


Base64 URL encoded Header
.
Base64 URL encoded Payload
.
Base64 URL encoded Signature

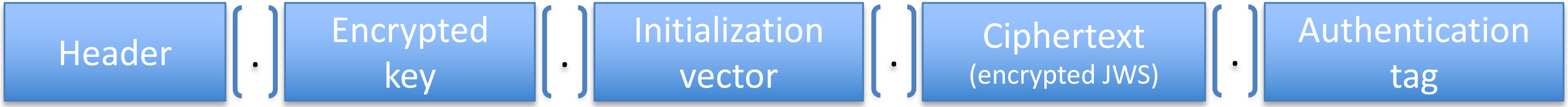
```
{
  "typ":"JWT",
  "alg":"RS256"
}
```

```
{
  "aud":"server",
  "iss":"https://ibm.com/oidc/endpoint/OP",
  "iat":1605792600,
  "exp":1605799800,
  "sub":"jdoe",
  "email":"jdoe@ibm.com"
}
```

```
eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.eyJhdWQiOiJzZXJ...jb20ifQ==.{Signature of JWT}
```

Encrypted JSON Web Token

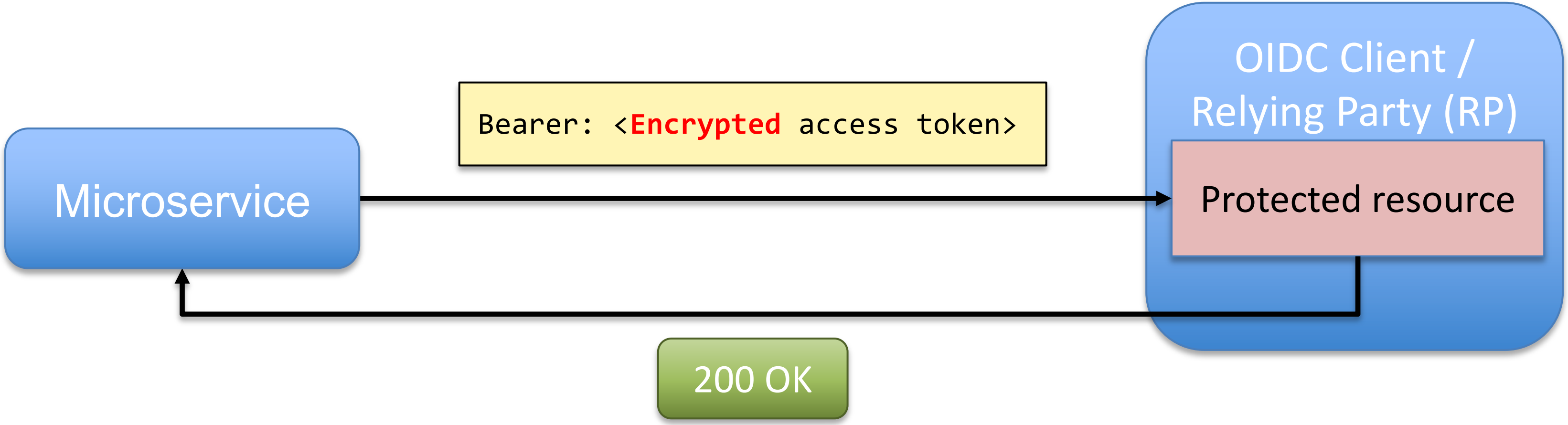
JSON Web Encryption (JWE)



Header . Encrypted key . Initialization vector . Ciphertext (encrypted JWS) . Authentication tag

1. Issuer builds header based on how the token will be built
2. Issuer creates JWS and random key (Content Encryption Key)
3. JWS is encrypted with CEK
   - Produces: Initialization vector, ciphertext, and authentication tag
4. Issuer uses receiver's public key to encrypt CEK
   - Produces: Encrypted key
5. Issuer joins each segment with a '.' to produce the JWE string

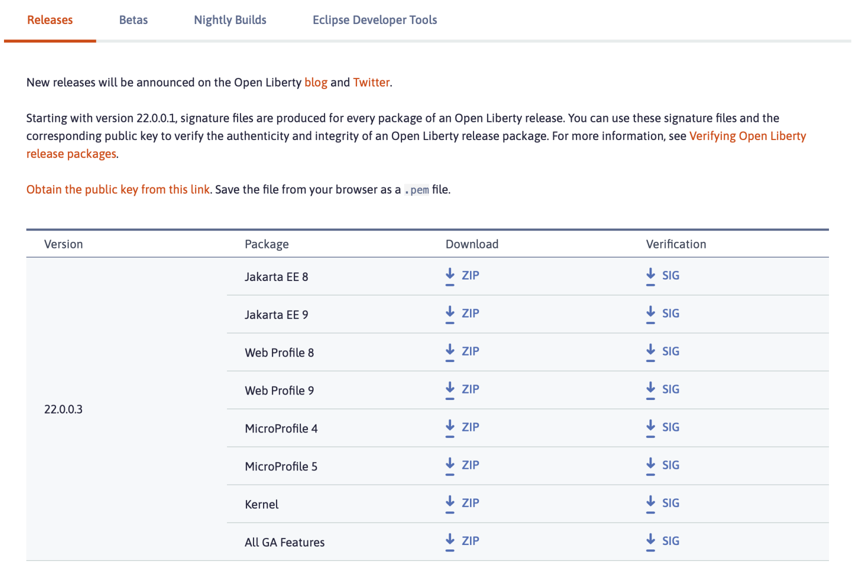
JWE Support for OIDC


Microservice
Bearer: <Encrypted access token>
OIDC Client / Relying Party (RP)
Protected resource
200 OK

Signing Open Liberty

- Open Liberty downloads from Open Liberty.io are now signed
- Use OpenSSL to verify signature

```
openssl dgst -sha256 -verify signer.pem
   -signature
   openliberty-javaee8-22.0.0.1.zip.sig
   openliberty-javaee8-22.0.0.1.zip
```



New releases will be announced on the Open Liberty blog and Twitter.

Starting with version 22.0.0.1, signature files are produced for every package of an Open Liberty release. You can use these signature files and the corresponding public key to verify the authenticity and integrity of an Open Liberty release package. For more information, see Verifying Open Liberty release packages.

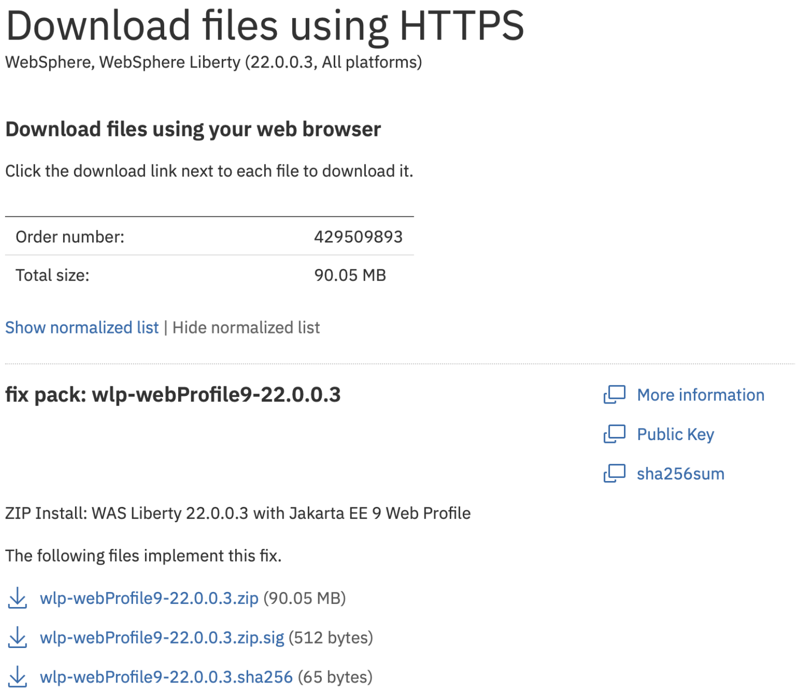
Obtain the public key from this link. Save the file from your browser as a `.pem` file.

Version	Package	Download	Verification
22.0.0.3	Jakarta EE 8	ZIP	SIG
	Jakarta EE 9	ZIP	SIG
	Web Profile 8	ZIP	SIG
	Web Profile 9	ZIP	SIG
	MicroProfile 4	ZIP	SIG
	MicroProfile 5	ZIP	SIG
	Kernel	ZIP	SIG
	All GA Features	ZIP	SIG

Signing WebSphere Liberty

- WebSphere Liberty downloads from IBM Fix Central are now signed
- Use OpenSSL to verify signature

```
openssl dgst -sha256 -verify signer.pem
   -signature
   wlp-webProfile9-22.0.0.3.zip.sig
   wlp-webProfile9-22.0.0.3.zip
```



Download files using HTTPS

WebSphere, WebSphere Liberty (22.0.0.3, All platforms)

Download files using your web browser

Click the download link next to each file to download it.

Order number:	429509893
Total size:	90.05 MB

Show normalized list | Hide normalized list

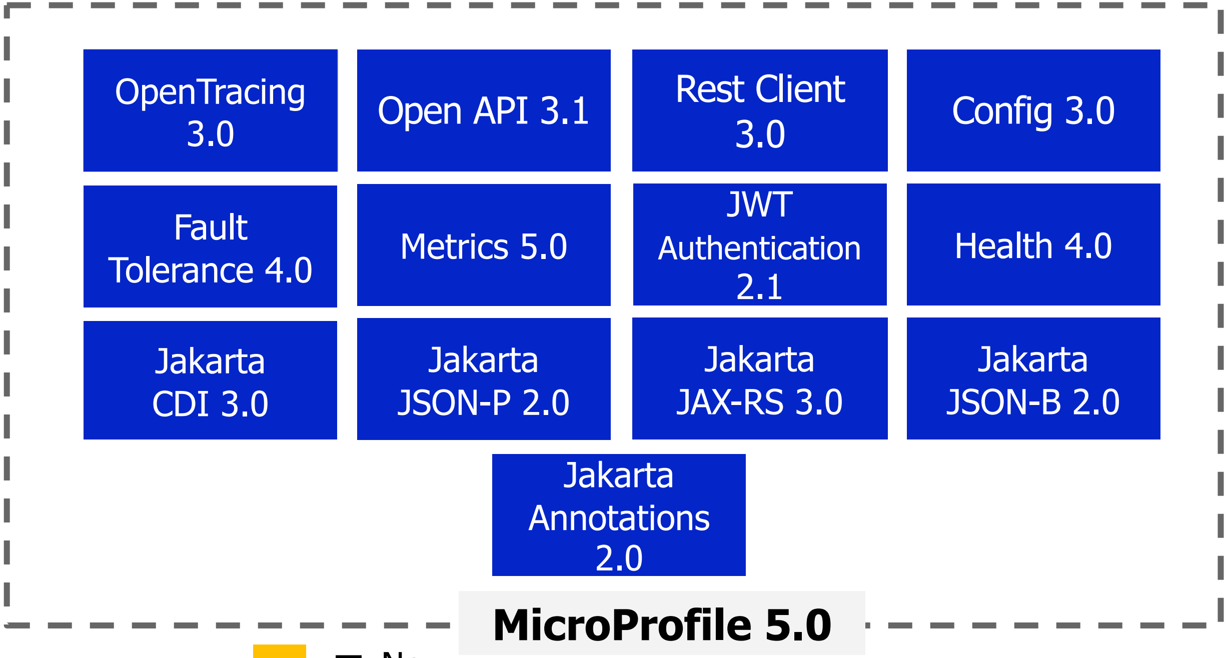
fix pack: wlp-webProfile9-22.0.0.3

ZIP Install: WAS Liberty 22.0.0.3 with Jakarta EE 9 Web Profile

The following files implement this fix.

- wlp-webProfile9-22.0.0.3.zip (90.05 MB)
- wlp-webProfile9-22.0.0.3.zip.sig (512 bytes)
- wlp-webProfile9-22.0.0.3.sha256 (65 bytes)

MicroProfile 5



OpenTracing 3.0	Open API 3.1	Rest Client 3.0	Config 3.0
Fault Tolerance 4.0	Metrics 5.0	JWT Authentication 2.1	Health 4.0
Jakarta CDI 3.0	Jakarta JSON-P 2.0	Jakarta JAX-RS 3.0	Jakarta JSON-B 2.0
	Jakarta Annotations 2.0		

MicroProfile 5.0

🟨 = New
🟦 = Updated
▨ = No change from last release

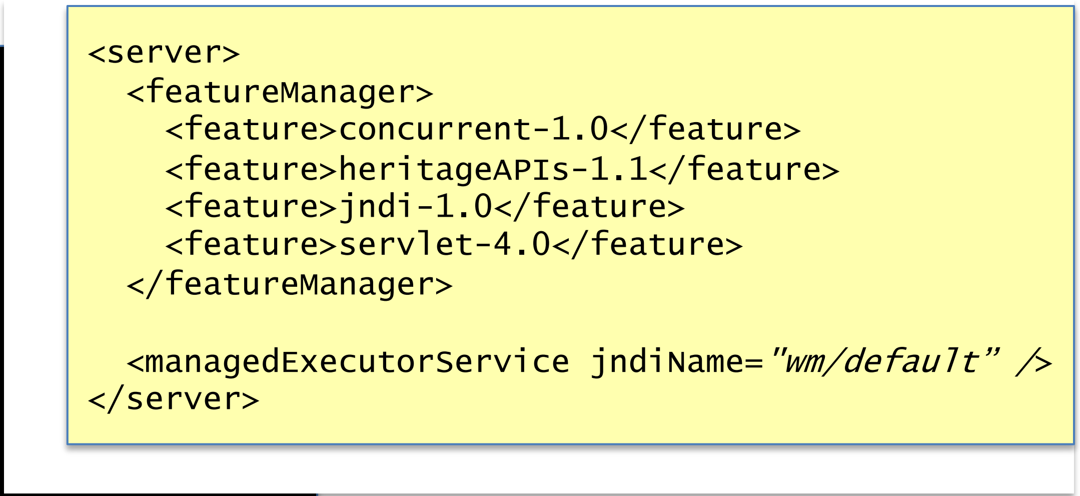
- Updated to work with Jakarta EE 9

CommonJ WorkManager & Timer

- CommonJ WorkManager and Timer is an old API for concurrent Programming in Java EE
- New Applications should use Concurrency Utilities for Jakarta EE
- Quartz Scheduler (an older API) integrates using CommonJ
- Liberty now supports it

CommonJ WorkManager

```
public class MyServlet
       extends HttpServlet {
   @Resource(lookup = "wm/default")
   WorkManager wm;

   public void doStuff() {
     Work work1 = new Work() {
       public void run() {
        // can perform lookups in
        // servlet’s name space
     };

     WorkItem item1 = wm.schedule(work1);
     WorkItem item2 = wm.schedule(work2);
     if (wm.waitForAll(
                 Arrays.asList(item1, item2),
                 TimeUnit.SECONDS.toMillis(30)))
```



```
<server>
  <featureManager>
    <feature>concurrent-1.0</feature>
    <feature>heritageAPIs-1.1</feature>
    <feature>jndi-1.0</feature>
    <feature>servlet-4.0</feature>
  </featureManager>

  <managedExecutorService jndiName="wm/default” />
</server>
```

CommonJ Timer

```
<server>
  <featureManager>
    <feature>concurrent-1.0</feature>
    <feature>heritageAPIs-1.1</feature>
    <feature>jndi-1.0</feature>
    <feature>servlet-4.0</feature>
  </featureManager>

   <commonjTimerManager jndiName="timer/tm1"
                        maxConcurrency="5” />
</server>
```

```
public class MyServlet
       extends HttpServlet {
   @Resource(lookup="timer/tm1")
   TimerManager tm;

   public void doStuff() {

     TimerListener tl = new TimerListener() {
       public void timerExpired(Timer timer) {
         // can perform lookups in
         // servlet/s name space
     };

     Timer timer1 = tm.scheduleAtFixedRate(tl,
                         startDate,
                         TimeUnit.HOURS.toMillis(8));
```

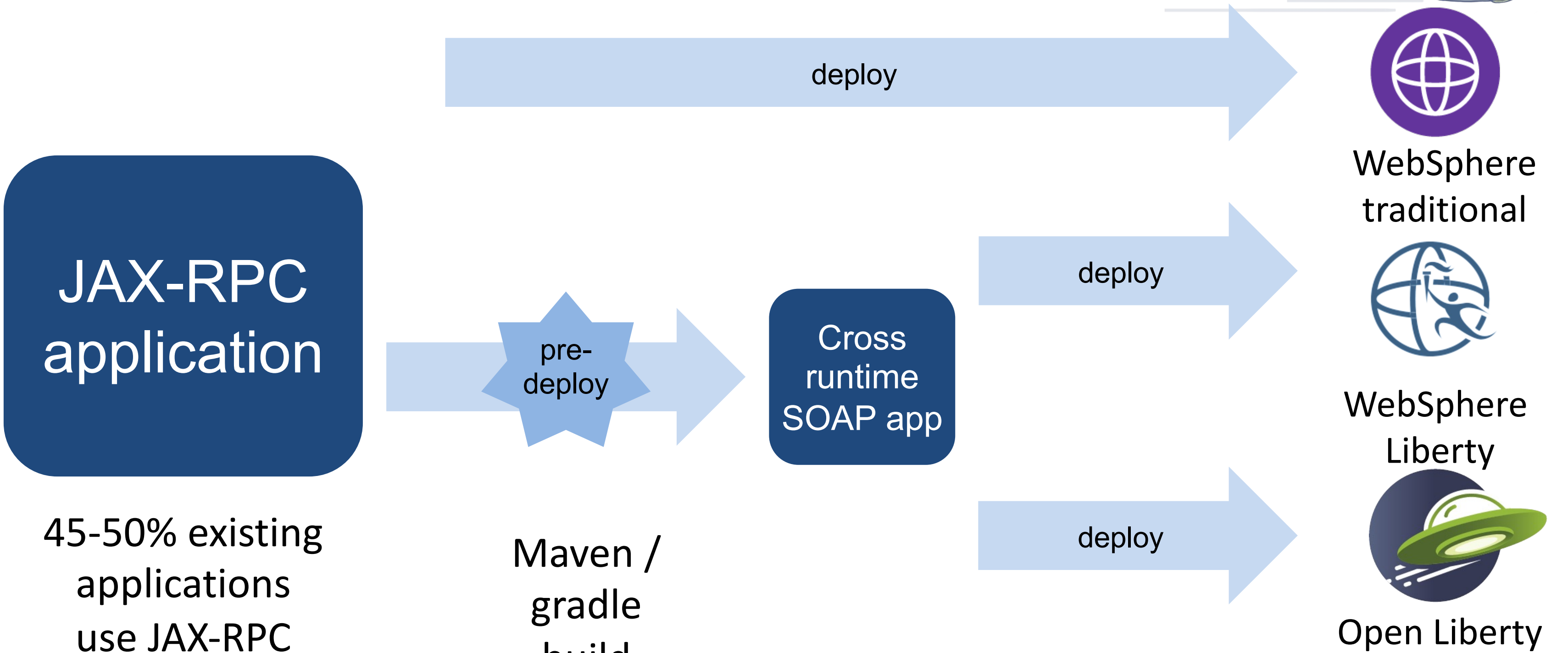
Quartz Timer

```
public class MyJob
      implements Job {
   DataSource ds;
   public MyJob() {
     ds = doLookup("java:comp/env/myDS");
   }

   public void execute() {
     try (Connection con = ds.getConnection()) {
```

```
<server>
  <featureManager>
    <feature>concurrent-1.0</feature>
    <feature>heritageAPIs-1.1</feature>
    <feature>jndi-1.0</feature>
    <feature>servlet-4.0</feature>
  </featureManager>

  <managedExecutorService
                jndiName="wm/default” />
</server>
```

```
org.quartz.scheduler.instanceName=MyQuartzScheduler
org.quartz.threadExecutor.class=org.quartz.commonj.WorkManagerThreadExecutor
org.quartz.threadExecutor.workManagerName=wm/default
org.quartz.threadPool.threadCount=4
```

JAX-RPC Pre-Deployment Tool



JAX-RPC application

pre-deploy

Cross runtime SOAP app

deploy

deploy

deploy

WebSphere traditional

WebSphere Liberty

Open Liberty

45-50% existing applications use JAX-RPC

Maven / gradle build process

```
<featureManager>
    <feature>jaxws-2.2</feature>
</featureManager>
```

How to use

1. Download from Fix Central: https://www.ibm.com/support/pages/node/6538940

2. Install into maven repository

```
mvn org.apache.maven.plugins:maven-install-plugin:2.5.2:install-file
    -Dfile=./jaxrpc-tools-maven/liberty-jaxrpc-maven-plugin-1.0.jar
    -DpomFile=./jaxrpc-tools-maven/pom.xml
```

How to use

- Add the following to project maven pom

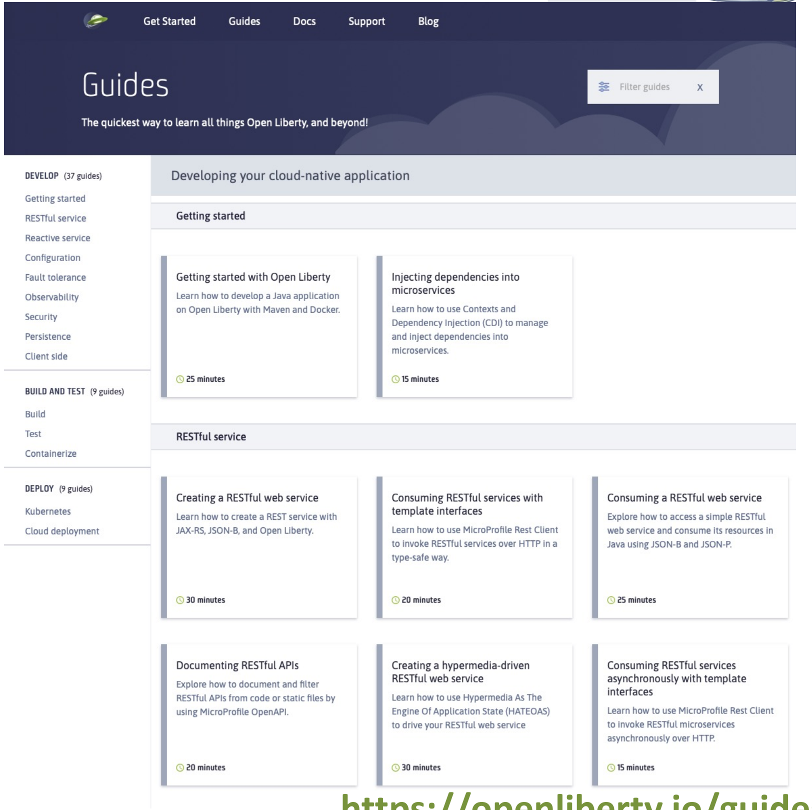
```
<plugin>
   <groupId>com.ibm.websphere.appserver.tools</groupId>
   <artifactId>liberty-jaxrpc-maven-plugin</artifactId>
   <version>1.2</version>
   <executions>
       <execution>
           <id>replace-jaxrpc</id>
           <phase>package</phase>
           <goals>
               <goal>replace-jaxrpc</goal>
           </goals>
       </execution>
   </executions>
</plugin>
```

Labs, Questions

Open Liberty Guides

Open Liberty

- Hands-on learning in ~20 minutes
- 55 guides
  - MicroProfile & Jakarta EE
  - Open Shift, Docker, Kubernetes Istio
- Latest Guide
  - *Containerizing microservices with Podman*



Guides

The quickest way to learn all things Open Liberty, and beyond!

DEVELOP (37 guides)
- Getting started
- RESTful service
- Reactive service
- Configuration
- Fault tolerance
- Observability
- Security
- Persistence
- Client side

BUILD AND TEST (9 guides)
- Build
- Test
- Containerize

DEPLOY (9 guides)
- Kubernetes
- Cloud deployment

Developing your cloud-native application

Getting started

Getting started with Open Liberty
Learn how to develop a Java application on Open Liberty with Maven and Docker.
25 minutes

Injecting dependencies into microservices
Learn how to use Contexts and Dependency Injection (CDI) to manage and inject dependencies into microservices.
15 minutes

RESTful service

Creating a RESTful web service
Learn how to create a REST service with JAX-RS, JSON-B, and Open Liberty.
30 minutes

Consuming RESTful services with template interfaces
Learn how to use MicroProfile Rest Client to invoke RESTful services over HTTP in a type-safe way.
20 minutes

Consuming a RESTful web service
Explore how to access a simple RESTful web service and consume its resources in Java using JSON-B and JSON-P.
25 minutes

Documenting RESTful APIs
Explore how to document and filter RESTful APIs from code or static files by using MicroProfile OpenAPI.
20 minutes

Creating a hypermedia-driven RESTful web service
Learn how to use Hypermedia As The Engine Of Application State (HATEOAS) to drive your RESTful web service
30 minutes

Consuming RESTful services asynchronously with template interfaces
Learn how to use MicroProfile Rest Client to invoke RESTful microservices asynchronously over HTTP.
15 minutes

https://openliberty.io/guides

Resources

Useful Liberty Links

- Why choose Liberty for Microservices: https://ibm.biz/6ReasonsWhyLiberty
- Choosing the right Java runtime: https://ibm.biz/ChooseJavaRuntime
- How to approach application modernization: https://ibm.biz/ModernizeJavaApps
- Open Liberty: https://www.openliberty.io
- Open Liberty Guides: https://www.openliberty.io/guides

Programming API Links

- Eclipse MicroProfile: https://microprofile.io
- Jakarta EE: https://jakarta.ee

Support Links

- Java support dates: http://www.ibm.com/developerworks/java/jdk/lifecycle
- Single Stream Continuous Delivery: https://www-01.ibm.com/support/docview.wss?uid=ibm10869798
- Container Support Policy: https://www.ibm.com/support/pages/container-deployment-support-policy-websphere-liberty
- Enhancement Requests: https://cloud-platform.ideas.ibm.com

Migration Tools

- IBM Transformation Advisor http://ibm.biz/cloudta
- WebSphere Binary Migration Toolkit: http://ibm.biz/WAMT4AppBinaries

Resources

Red Hat UBI images

- https://icr/r/ibmcom/websphere-liberty
- https://hub.docker.com/r/openliberty/open-liberty

Ubuntu images

- https://hub.docker.com/_/websphere-liberty
- https://hub.docker.com/_/open-liberty

IBM Container Registry images

- https://cloud.ibm.com/docs/Registry?topic=RegistryImages-ibmliberty

Configuration/build files in github

- https://github.com/WASdev/ci.docker
- https://github.com/OpenLiberty/ci.docker

Next Quarterly Update

Liberty 22.0.0.4-6 Update

Session#1: Jun 23, 2021 from 9-10am ET - https://ibm.biz/Liberty-Jun23

Session#2: Jun 30, 2021 from 1-2pm ET - http://ibm.biz/Liberty-Jun30

Broadcast on Expert TV: http://ibm.biz/IBMExpertTV-LetsCode

Annual WebSphere Survey

Help shape the future of WebSphere by taking our annual survey

https://ibm.biz/2022survey

Two lucky participants will win a $100 gift card!

Each gift card amount will total $100 USD and the recipient will receive equivalent converted amounts in their local currency.

Join the WebSphere & Liberty CAB

The Customer Advisory Board for WebSphere and Liberty customers and partners

Let’s continue our conversation.
Anyone can join.

- ✓ Join weekly (optional) sessions
- ✓ Connect directly with experts
- ✓ Share pain points and best practices
- ✓ Provide feedback
- ✓ Be the first to review roadmaps
- ✓ Get insights from peers
- ✓ Access to opportunities

Register → http://ibm.biz/WAS&LibertyCAB

Contact: claudiab@us.ibm.com

Questions?

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

alasdair@ibm.com

Thank You

Your Feedback is Important

Open Liberty