

RED HAT
SUMMIT

BOSTON, MA
JUNE 23-26, 2015

JBoss EAP 7

Bilge Ozpeynirci
Sr. Product Manager

Dimitris Andreadis
Sr. Engineering Manager

June 2015

AGENDA

- JBoss EAP 7
 - Release Goals
 - EAP and Community WildFly
 - New Features in EAP7
 - Migration Considerations
- Q & A

Important Notes:

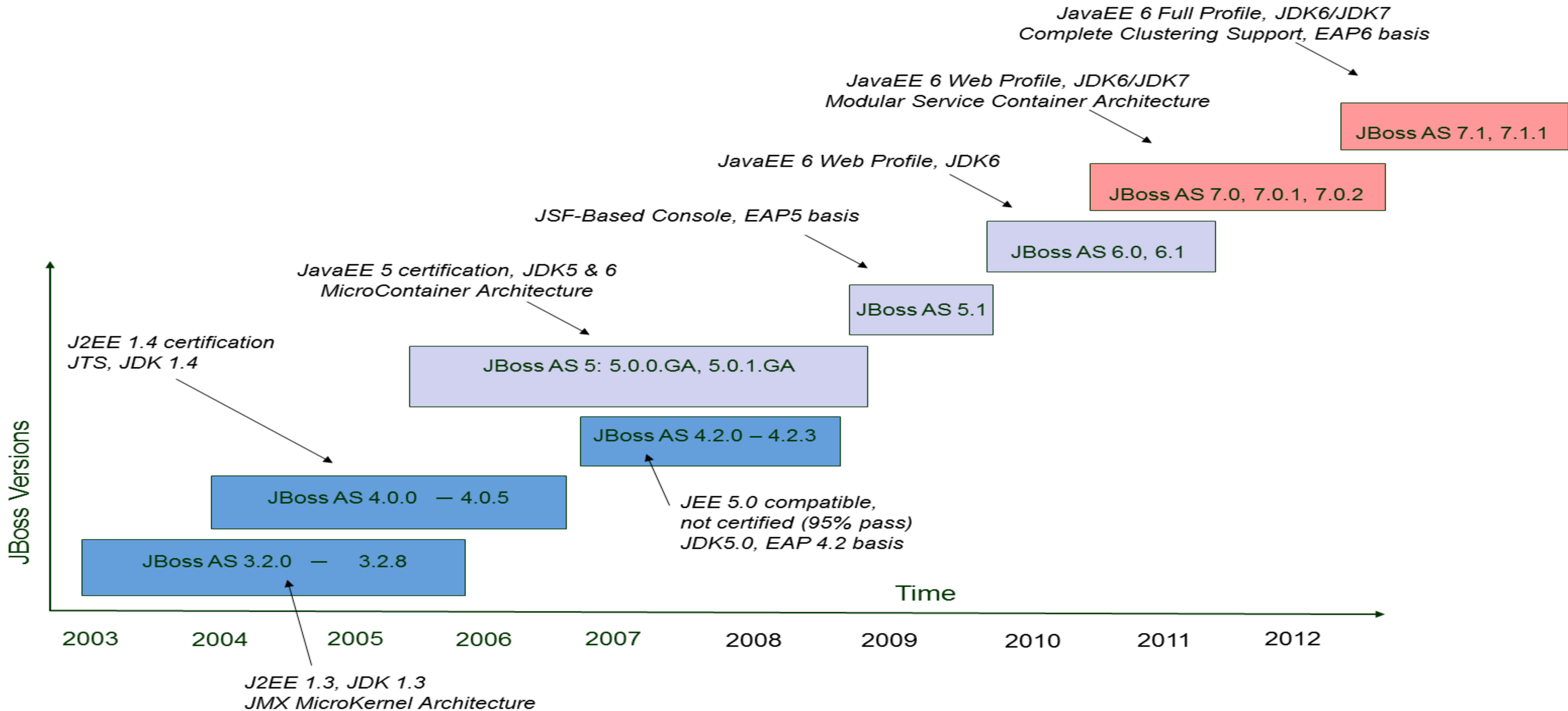
- 1. Usual caveats apply - Features and release dates can and will change**
- 2. Features mentioned in this document do not constitute product commitments**

RELEASE GOALS

- Lightweight Java EE 7 Web Profile and Full Platform
- Compatibility
- Enhanced Management and Security
- Enhanced Modularity
- Enhanced Web Console Interface
- Interfaces for Red Hat Customer Support
- Better Integration with other JBoss Products

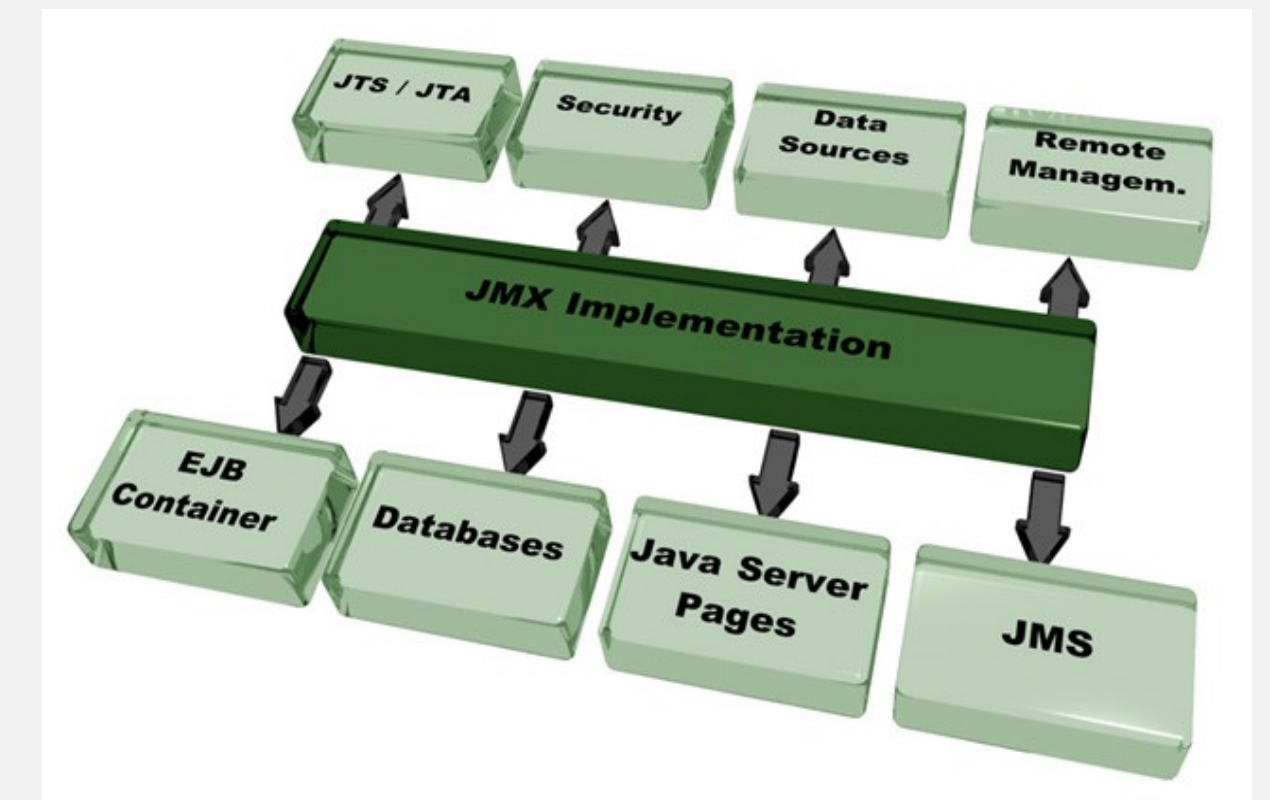
JBoss EAP 7 & Community WildFly

Community JBoss AS Timeline



Kernel Taxonomy

- AS 2.x, 3.x, 4.x / JBoss EAP 4.2 / 4.3
 - JBoss JMX MicroKernel
 -
- AS 5.x, 6.x / JBoss EAP 5.x
 - JBoss MC - MicroContainer
 -
- AS 7.x, WildFly 8.x, 9, 10 / JBoss EAP 6.x, EAP 7
 - JBoss MSC – Modular Service Container



Community Project vs EE spec

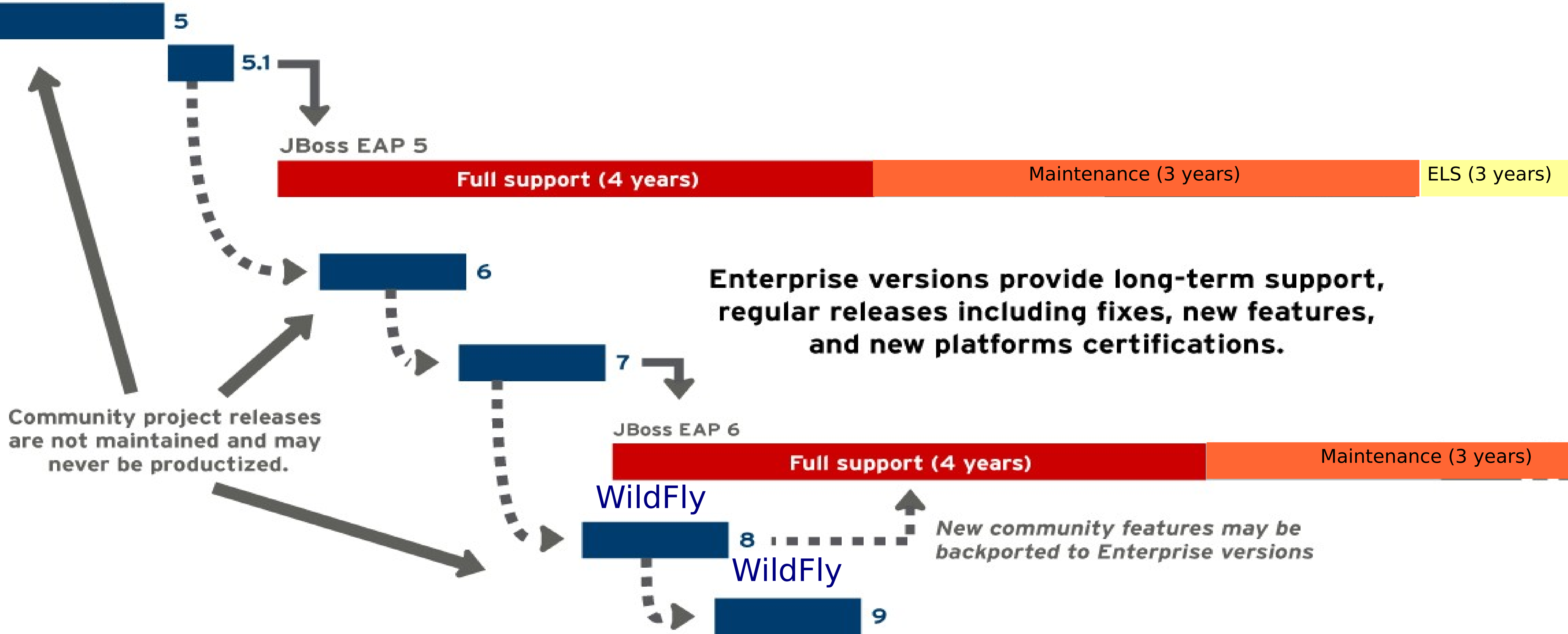
- JBoss AS 2 ==> J2EE 1.2
- JBoss AS 3 ==> J2EE 1.3
- JBoss AS 4 ==> J2EE 1.4
- JBoss AS 5 ==> Java EE 5
- JBoss AS 6, AS7 ==> Java EE 6
- WildFly 8, 9, 10 ==> Java EE 7

Community Project vs EE spec vs Product (JBoss EAP)

• JBoss AS 2	==>	J2EE 1.2		
• JBoss AS 3	==>	J2EE 1.3		
• JBoss AS 4	==>	J2EE 1.4	==>	JBoss EAP 4
• JBoss AS 5	==>	Java EE 5	==>	JBoss EAP 5
• JBoss AS 6, AS7	==>	Java EE 6	==>	JBoss EAP 6
WildFly 8, 9, 10	==>	Java EE 7	==>	<i>JBoss EAP 7</i>

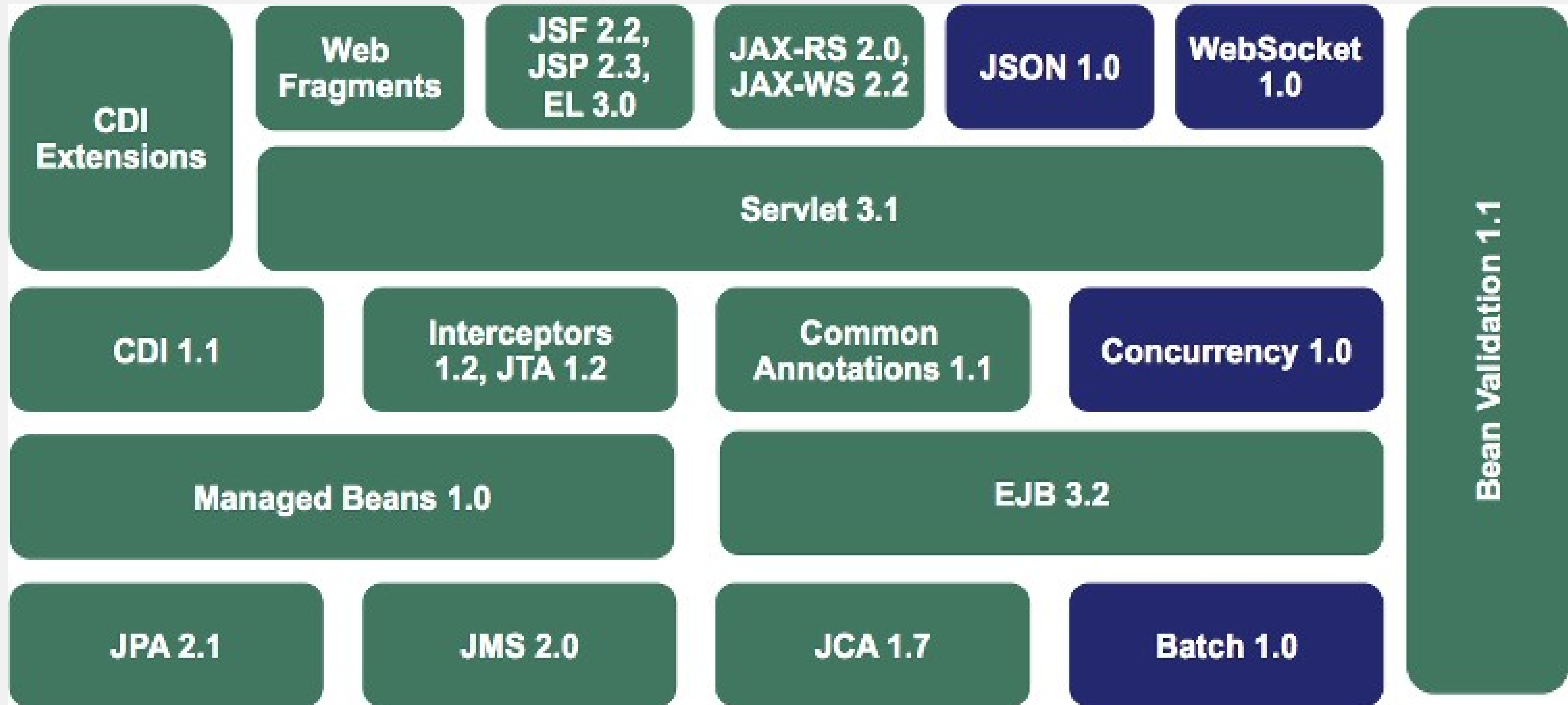
WildFly / JBoss EAP relationship

JBoss Community AS



New Features in EAP7

Java EE7 Highlights



EE7 Highlights – New Technologies

- JSR-352 Batch Applications for the Java Platform
 - Runtime & Artifact API, XML-based Job specification lang.
- JSR-236 Concurrency Utilities for JavaEE
 - Executor, Scheduled Executor, Thread Factory, Context
- JSR-353 Java API for JSON Processing (JSON-P)
 - Parse, transform and query JSON data
- JSR-356 Web Sockets support
 - Annotation driven endpoints and lifecycle callbacks

EE7 Highlights – Specification Updates

- JSR-345 EJB 3.2, plus Interceptors 1.2, Annotations 1.2
 - Misc. improvements
- JSR-340 Servlet 3.1
 - Non-blocking I/O, HTTP upgrade, etc.
- JSR-342 JMS 2.0
 - Shared topic subs, delayed delivery, async send, etc.
- JSR-344 JSF 2.2
 - HTML 5, FaceFlows, Stateless Views, Resource lib contracts
- JSR-322 JCA 1.7
 - Activation name for msg endpoints

EE7 Highlights – Optional Technologies

- EJB 2.1 Entity Beans (CMP)
- JAX-RPC (API for XML-based RPC)
- JAXR (API for XML registries)
- JSR-88 (Deployment API)
 - Re-surfaced in JSR-373 for EE8 (JavaEE Management API 2.0)

BYE
BYE

New WebServer - Undertow

- Blocking / non-blocking NIO based APIs
- Composition / handler based architecture
- Lightweight and fully embeddable
- Support for WebSockets, including JSR-356
- Support for Servlet 3.1
- HTTP Upgrade

Undertow – Embedded Server

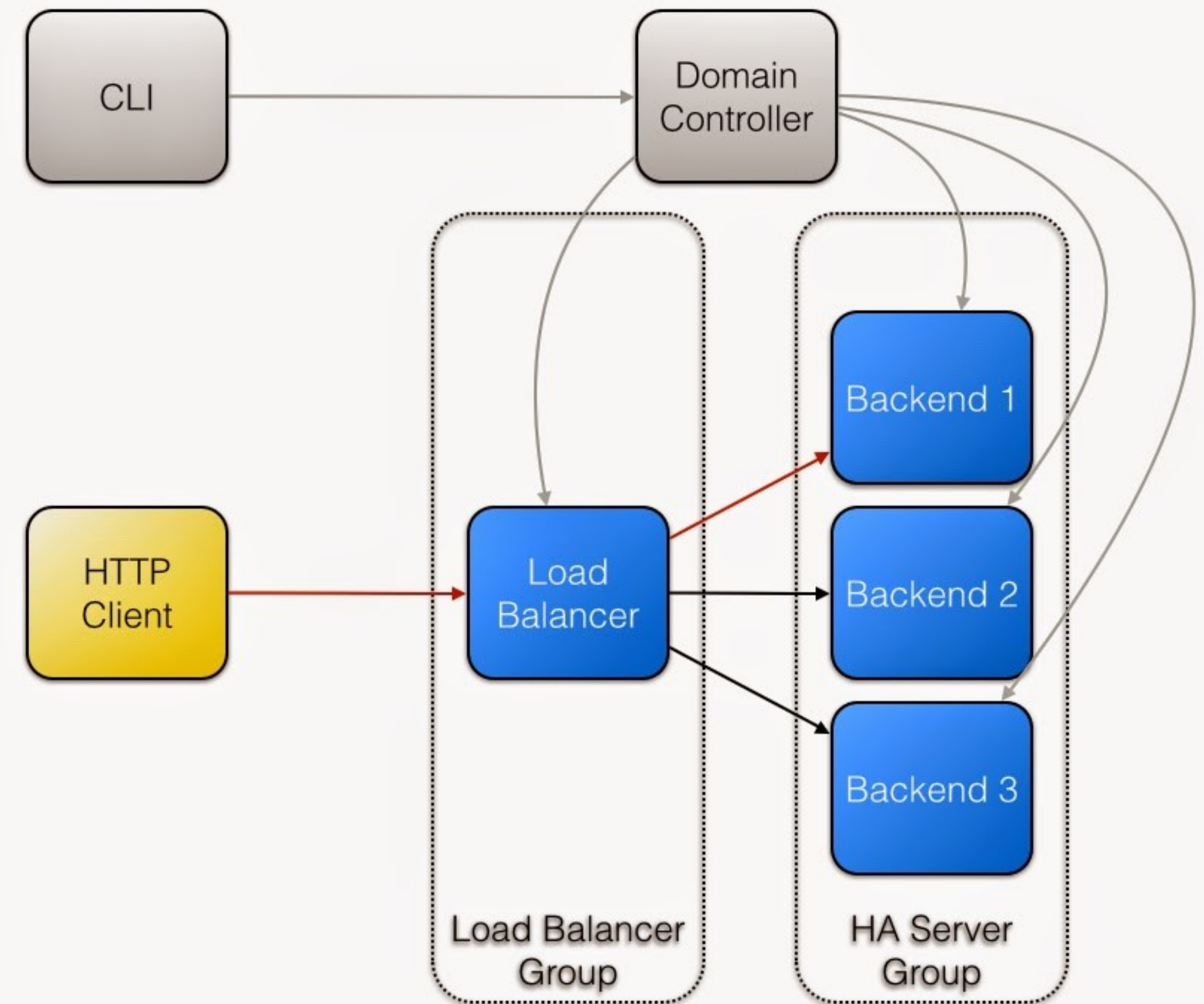
```
public class HelloWorldServer {  
  
    public static void main(final String[] args) {  
        Undertow server = Undertow.builder()  
            .addHttpListener(8080, "localhost")  
            .setHandler(new HttpHandler() {  
                @Override  
                public void handleRequest(final HttpServerExchange exchange) throws Exception {  
                    exchange.getResponseHeaders().put(Headers.CONTENT_TYPE, "text/plain");  
                    exchange.getResponseSender().send("Hello World");  
                }  
            }).build();  
        server.start();  
    }  
}
```

Port Reduction

- HTTP Upgrade allows us to reduce the number of ports in the default installation to just 2:
 - 8080 – for applications with JNDI and EJB multiplexed
 - 9990 – for management, for both HTTP/JSON & native API
- The only overhead is the initial HTTP Upgrade request / response

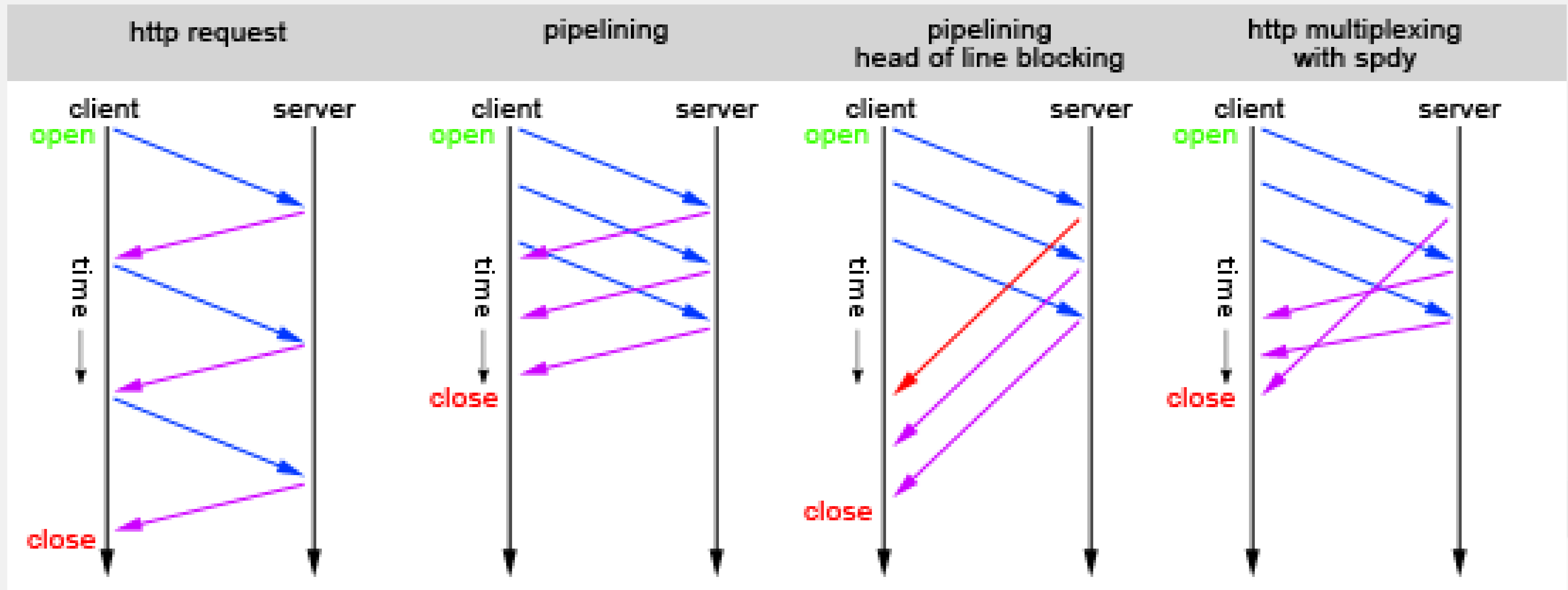
Undertow as Load Balancer

- A full Java solution managed centrally
- Implements mod_cluster
- HTTP, AJP, HTTP/2



HTTP/2 (vs HTTP/v1.x)

- Primary goal of HTTP/2 is to reduce latency and make efficient use of the TCP connection.



Undertow HTTP/2 support

- HTTP/2 characteristics
 - Binary Framing
 - Request/Response multiplexing
 - Header compression (HPACK)
 - Stream prioritization
 - Server push
 - HTTP Upgrade or NPN/ALPN*
(*requires special JDK8 setup)
 -
- Undertow implements HTTP/2 (& SPDY)

JSR-369 Servlet 4.0

- Brings HTTP/2 support to Java EE8
 - Server Push
 - Stream Prioritization
 - Request/Response Multiplexing
 -
- JEP 110 : HTTP/2 Client
 - Basic protocol support in Java SE 9
 - New HTTP client API that implements HTTP/2 and WebSocket and can replace the legacy HttpURLConnection API

Server Push with Undertow

```
package io.undertow.server;

public abstract class ServerConnection ...
{
    /**
     * Attempts to push a resource if this connection supports server push. Otherwise the request is ignored.
     *
     * Note that push is always done on a best effort basis, even if this method returns true it is possible that
     * the remote endpoint will reset the stream
     *
     * @param path The path of the resource
     * @param method The request method
     * @param requestHeaders The request headers
     * @return <code>true</code> if the server attempted the push, false otherwise
     */
    public boolean pushResource(final String path, final HttpString method, final HeaderMap requestHeaders) {
        return false;
    }

    public boolean isPushSupported() {
        return false;
    }
}
```

- Or use the Undertow *Learning Push Handler*
 - Learns which resources are requested per page and pushes them proactively

Graceful Shutdown

- Essentially `suspend(timeout) / resume()`
 - Allow active sessions/requests/in-flight-txs to complete
 - After suspend the server may be restarted
- Action per subsystem may vary
 - Reject new requests (HTTP Error 503 – Service unavailable)
 - Clustering could fail-over sessions
 - `mod_cluster` could notify load balancer

ORB Switch

- Java EE requires deep IIOP/CORBA integration
- JBoss was always based on JacORB
 - Good ORB but inter-ORB interop always challenging
- Switched to the OpenJDK ORB to
 - Pool resources with other vendors
 - Reduce Java EE RI interop issues

Offline CLI

- Configure a server without the server “running”
- Similar to admin-only mode, without any ports bound
- Implemented by embedding the server in the CLI (!)
 - CLI is also embeddable...
 -

```
$ bin/jboss-cli.sh  
[disconnected /] embed-server --std-out=echo  
12:10:15,300 INFO [org.jboss.modules] (main) JBoss Modules version 1.4.1.Final  
...  
(server boots up)
```

Misc. Features

- Log files appear as resources and can be downloaded
- JCA / IronJacamar Tracer module
- JGroups is now a fully managed resource
 - Infinispan Caches can share JGroups Channels using FORK
- Web Console
 - Improved Log Viewer
 - Datasource Templates
 - Flush operation for connection pools
 - Enhanced model browser
 - ...

EAP7 - WIP

Work In Progress

- JMS Graceful Shutdown
- HA Singleton Deployments
- Improved UI for large domains
- Improved Task Flows
- Profile cloning & inclusion
- ...

Messaging

- Messaging Consolidation
 - Same broker technology in JBoss EAP 7 and JBoss A-MQ 7 products
 - HornetQ -> Active MQ Artemis
 - EAP 6 Backward compatibility is considered

Elytron

- Replacement for PicketBox (originating from JbossSX)
- Replace JAAS as the internal SPI
- Stop “sending the password around”
- Multiple security associations per context
- Support for HTTP, SASL+GSSAPI, Kerberos, JASPIC, JACC, TLS, RADIUS, etc.
- Integrate well with PicketLink and Keycloak
- ...

MicroServices: wildfly-swarm (experimental)*

- Turn your .war into a fat .jar with just enough of the AppServer runtime to support your use case
 - Include the wildfly-swarm-plugin in your pom.xml
 - Declare any dependencies on:

Undertow, JAX-RS, Naming/JNDI,
Transactions, Messaging, Datasources,
Logging, Weld/CDI, Security, ...

```
<plugin>
  <groupId>org.wildfly.swarm</groupId>
  <artifactId>wildfly-swarm-plugin</artifactId>
  <version>${version.wildfly-swarm}</version>
  <executions>
    <execution>
      <phase>package</phase>
      <goals>
        <goal>create</goal>
      </goals>
    </execution>
  </executions>
</plugin>
```

MicroServices: wildfly-swarm (experimental)*

- [Write your own main() to override default configuration & reference it in your MANIFEST.MF]

- Voila!

```
java -jar myservice-swarm.jar
```

(boots in < 1sec)

```
public class Main {  
  
    public static void main(String[] args) throws Exception {  
        Container container = new Container();  
  
        container.subsystem(new MessagingFraction()  
                            .server(  
                                new MessagingServer()  
                                    .enableInVmConnector()  
                                    .topic("my-topic")  
                                    .queue("my-queue")  
                                )  
                            );  
  
        // Start the container  
        container.start();  
  
        JaxRsDeployment appDeployment = new JaxRsDeployment();  
        appDeployment.addResource(MyResource.class);  
  
        // Deploy your JAX-RS app  
        container.deploy(appDeployment);  
  
        // Create an MSC deployment  
        ServiceDeployment deployment = new ServiceDeployment();  
        deployment.addService(new MyService("/jms/topic/my-topic"));  
  
        // Deploy the services  
        container.deploy(deployment);  
    }  
}
```

JBOSS EAP 7 MIGRATION CONSIDERATIONS

JBOSS EAP 7 MIGRATION CONSIDERATIONS

CHANGES	CONSIDERATIONS and SUPPORT
Web Container JBoss Web (Deprecated)-> Undertow	Accepts jboss-web.xml Subsystem compatibility Some tc Valve API migration/compatibility
Messaging HornetQ (Deprecated) -> Active MQ Artemis	Active MQ Artemis ~ HornetQ (similar configuration) Interoperability to EAP 6 JMS 2.0
Container Security PicketBox (Deprecated) -> Elytron	Compatibility for user JAAS login modules Compatibility for Security Domain Config
EJB Client Improvements	Same APIs are presented Same wire compatibility Improved configuration (security context)
Remoting Object Request Broker JacORB (Deprecated) -> JDK ORB	Users should not be impacted/unless poked into internals.
SAML Support PicketLink Federation (Deprecated)	Maintained as-is (EAP 6.4) for existing customer applications that uses PicketLink Federation
SAML Support (Post EAP 7.0 GA) PicketLink Federation -> JBoss KeyCloak Client Libraries for EAP 7 / KeyCloak Server	New Keycloak client libraries for EAP (SAML Client SP) and Keycloak server (OOTB Solution for Auth* Server with SAML IdP/OAuth/OpenID Connect)
Web Services Apache CXF Spring for endpoint setup (N/S)	Migration to CDI (standard) based support for endpoints

JBOSS EAP 7 MIGRATION CONSIDERATIONS cont.

CHANGES	CONSIDERATIONS
OSGi (TP in EAP 6) Deprecated	OSGi Enterprise Spec is not complete (No EJB etc.) Talk to your TAM if Karaf (JBoss Fuse / JBoss A-MQ) is right for you. JBoss EAP 7 integrates with A-MQ 6 and 7.
EJB 2 Entity Beans (CMP) Deprecated	Use JPA Instead Java EE 7 does not require
JAX-RPC Deprecated	Use JAX-WS instead (JBoss WS) Java EE 7 does not require
JSR-88 (Deployments) Impl. Deprecated	No or very little Adoption Java EE 7 does not require
JDK 8, JDK 9 Support	JDK 7 will not be supported (EOL by Oracle) JDK 9 community tests are already underway
Port Reduction	Users can still use the old port if they wish
Certified Platforms / Integrations	EAP 7.0 on RHEL for IBM Power Architecture - cease RHEL 5 – cease EAP 7.0 via Generic RA Tibco EMS – cease

RED HAT
SUMMIT

LEARN. NETWORK.
EXPERIENCE OPEN SOURCE.