

Reasons to love WebSphere Application Server.....



Secure

IBM Support and
Training



Scalable



Well-tooled



Very
Fast!



Robust

... but you asked us for something more.....

- Lightweight
- Composable
- Simple
- Flexible
- Dynamic
- Open & Extensible
- Cloud-ready



WebSphere Application Server
Liberty Profile

WAS 8.5, June 2012

.... but as production-ready as the original.....
.... with even higher management scale!

- ▶ Must I prepare to move from WAS Classic? Absolutely not
 - Full profile remains fully supported and strategic. If it fits your needs, stay

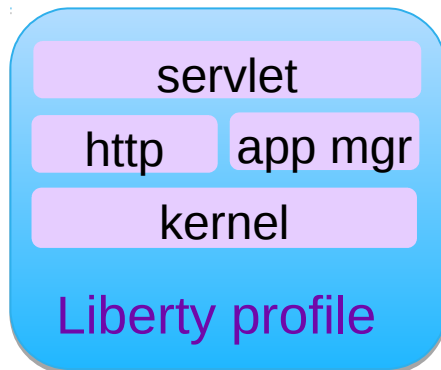
Simple to get started locally

1. download liberty archive
2. `java -jar <archive>`
3. `bin/server create myServer`
4. `cp myapp.war usr/servers/myServer/dropins`
5. `bin/server run myServer`
6. `http://localhost:9080/myapp`

Composable server instances

- You control which features are loaded into each server instance

`<feature>servlet-3.1</feature>`



`<feature>jsf-2.2</feature>`

`<feature>appSecurity-2.0</feature>`

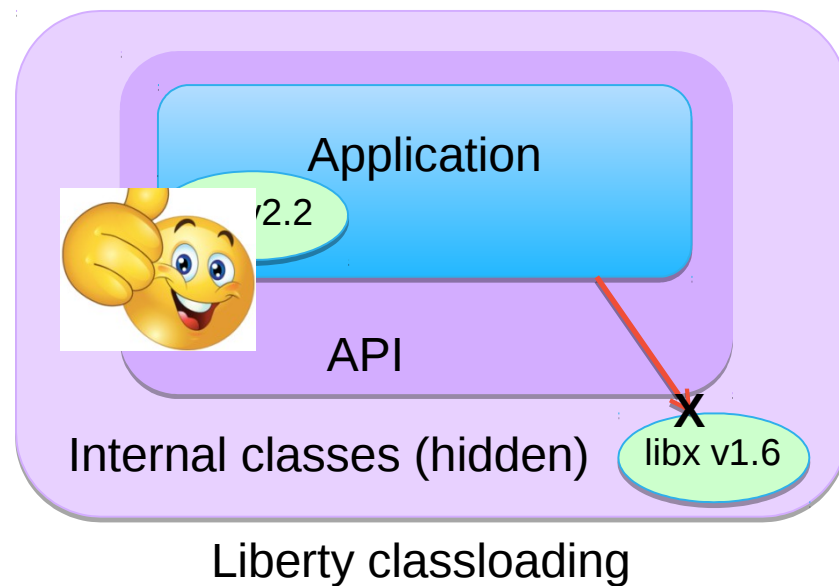
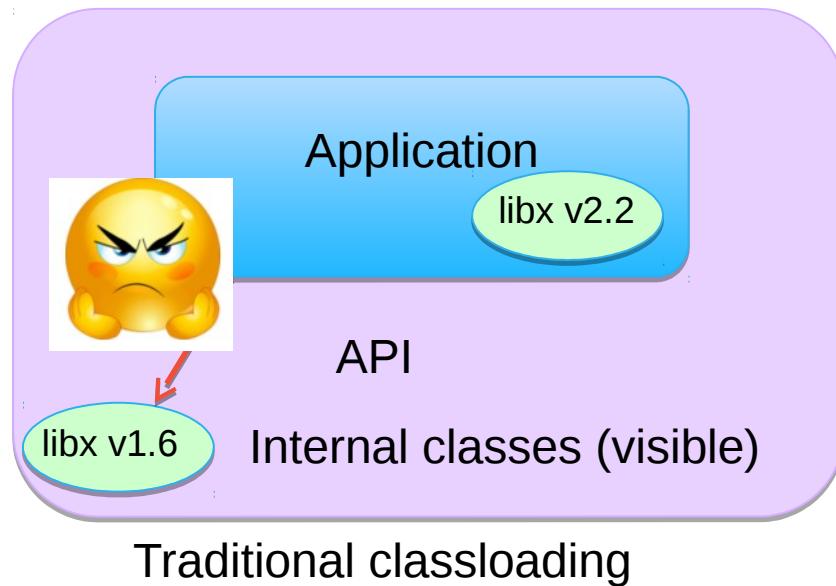


Internal feature dependencies are managed for you, for example:

- jsf feature includes jsp
- jsp feature include servlet
- jdbc feature includes transactions

Simple to use third-party libs

- Hidden internals keep classloaders clean



```
<classloader  
xmi:id="ClassLoader_1347543866613"  
mode="PARENT_LAST"/>
```

Even simpler in the cloud

Developer desktop



app is packaged
with runtime by
liberty buildpack

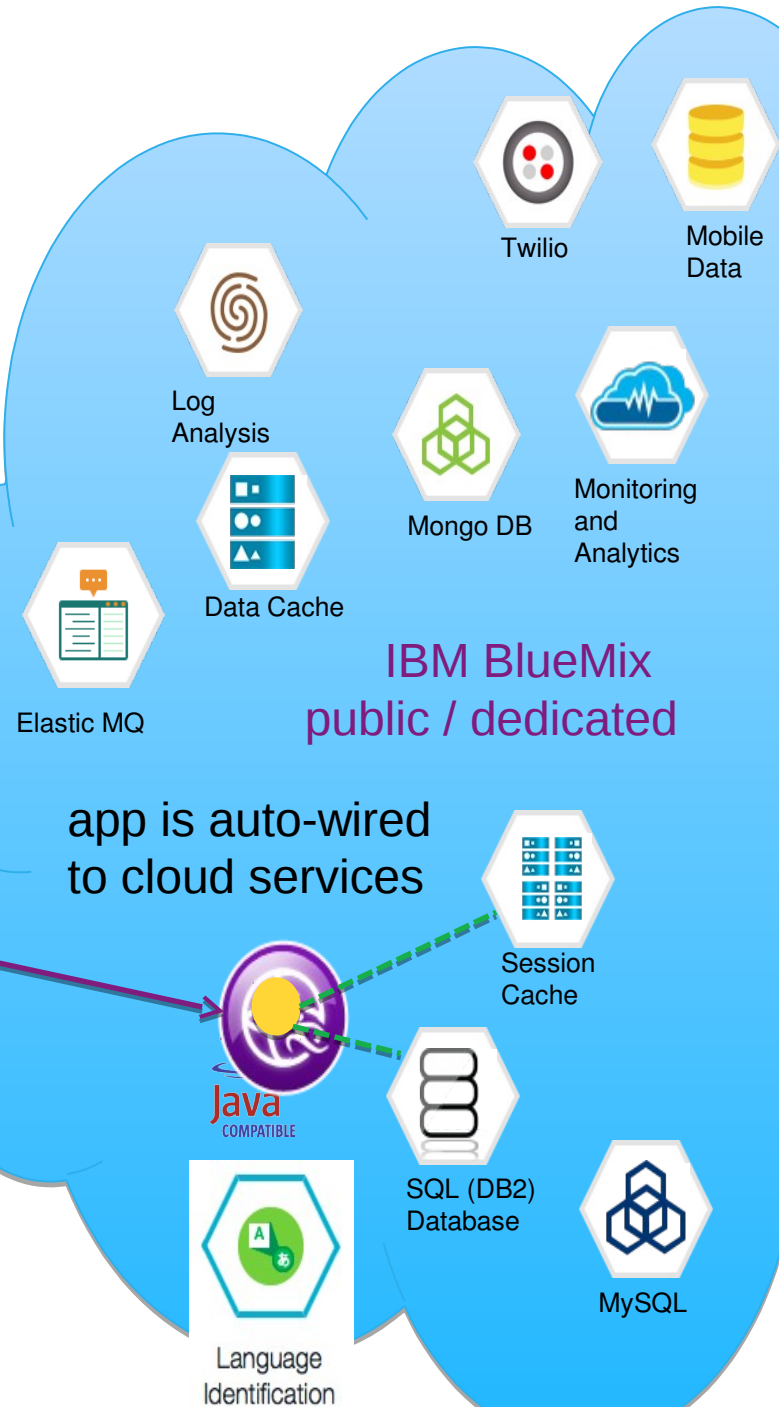


cf push
app.war

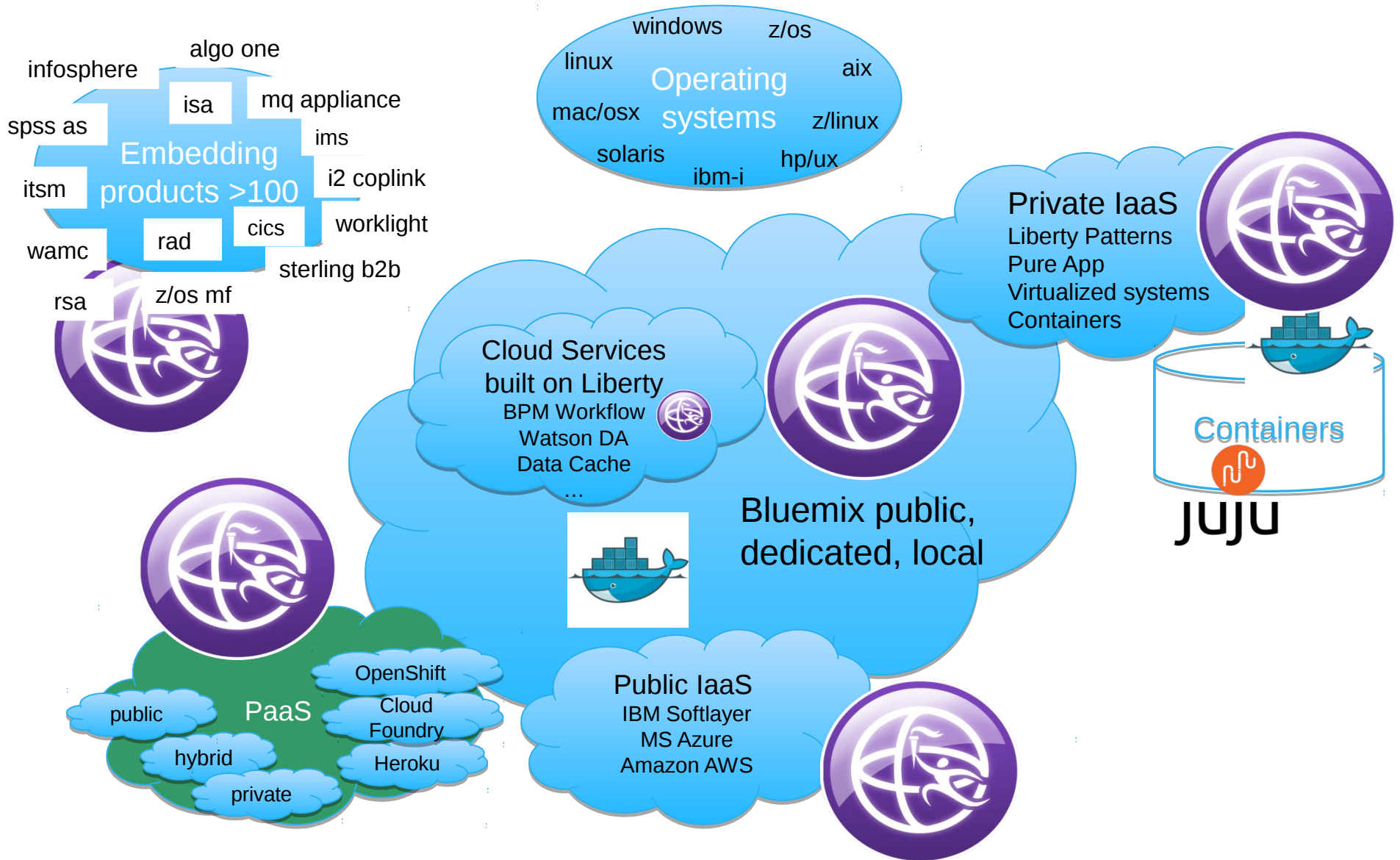


IBM BlueMix public / dedicated

app is auto-wired
to cloud services



Flexible to Run Everywhere





Docker and WebSphere Application Server



- Support for WebSphere Application Server Liberty Profile and Full Profile running under Docker
- WAS Liberty images on Docker Hub for Development use
 - Latest WAS V8.5.5. Liberty driver
 - WAS Liberty V9 Beta with Java EE 7
- Dockerfiles on WASdev GitHub to:
 - Upgrade the Docker Hub image with Liberty Core, Base or ND commercial license
 - Build your own Docker image for Liberty or Full Profile



websphere-liberty 

3 days ago

Official IBM WebSphere Application Server for Developers Liberty Profile image.

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

Simple to configure

- ONE required configuration file
 - `${wlp.install.dir}/usr/servers/server_name/server.xml`
 - User configuration for kernel and features
- Optional files in same server directory
 - `server.env`
 - read by server script
 - useful to override locations
 - eg. `JAVA_HOME`, `WLP_USER_DIR`, and `WLP_OUTPUT_DIR`
 - `jvm.options`
 - provided to JVM at startup
 - JVM settings, system properties
 - `bootstrap.properties`
 - to initialize core runtime (before XML config is read)
 - predefined properties
 - any custom property, variable for XML config etc

Simple to configure

- *server create* command generates simple working server.xml



```
<server description="new server">

    <!-- Enable features -->
    <featureManager>
        <feature>jsp-2.2</feature>
    </featureManager>

    <httpEndpoint id="defaultHttpEndpoint"
        host="localhost"
        httpPort="9080"
        httpsPort="9443" />

</server>
```

- Runs servlets and JSPs
- Default endpoint shown to make port overrides easy

Simple to share config files

- Include files from local filesystem or network

```
<server description="new server">  
  <featureManager>  
    <feature>jsp-2.2</feature>  
  </featureManager>  
  <include location="${shared.config.dir}\common-config.xml" />  
</server>
```

`${wlp.install.dir}/usr/servers/server_name/server.xml`

- Use variables to maximise sharing

```
<fileset id="DerbyFileset"  
  dir="${derbypath}"  
  includes="derby-10.10.1.1.jar" />
```

`${wlp.install.dir}/usr/servers/shared/config/common-config.xml`

`derbypath=/usr/derby`

```
<variable name="derbypath"  
  value="/usr/derby" />
```

`export derbypath=/usr/derby`

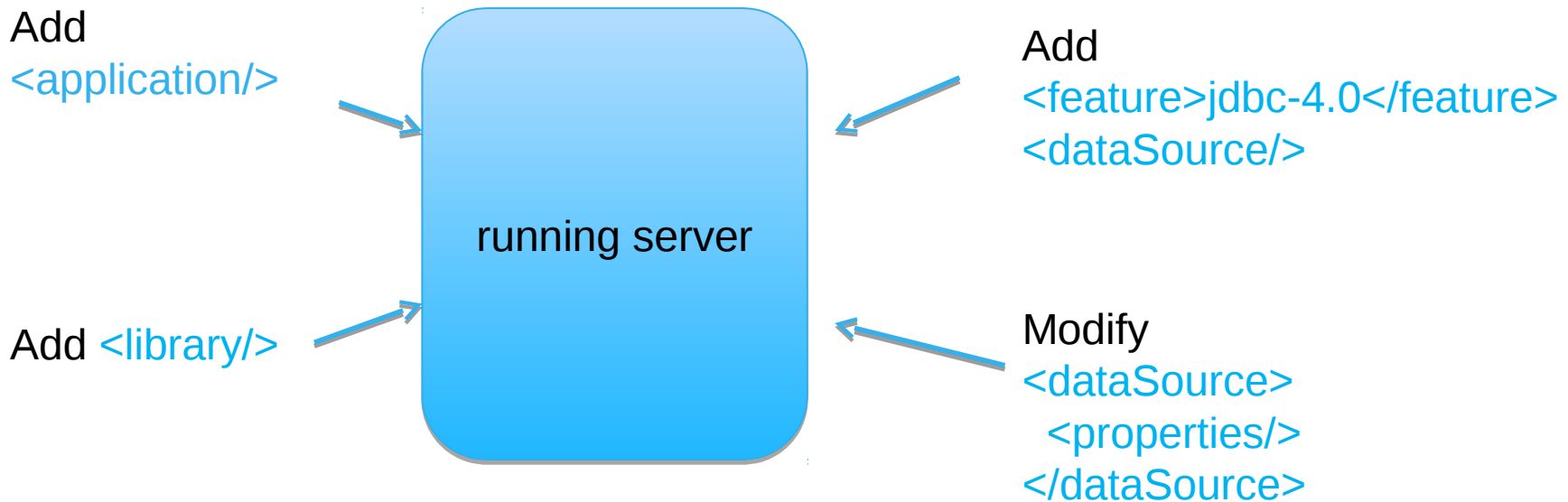
bootstrap.properties file

server.xml or included xml file

environment variable

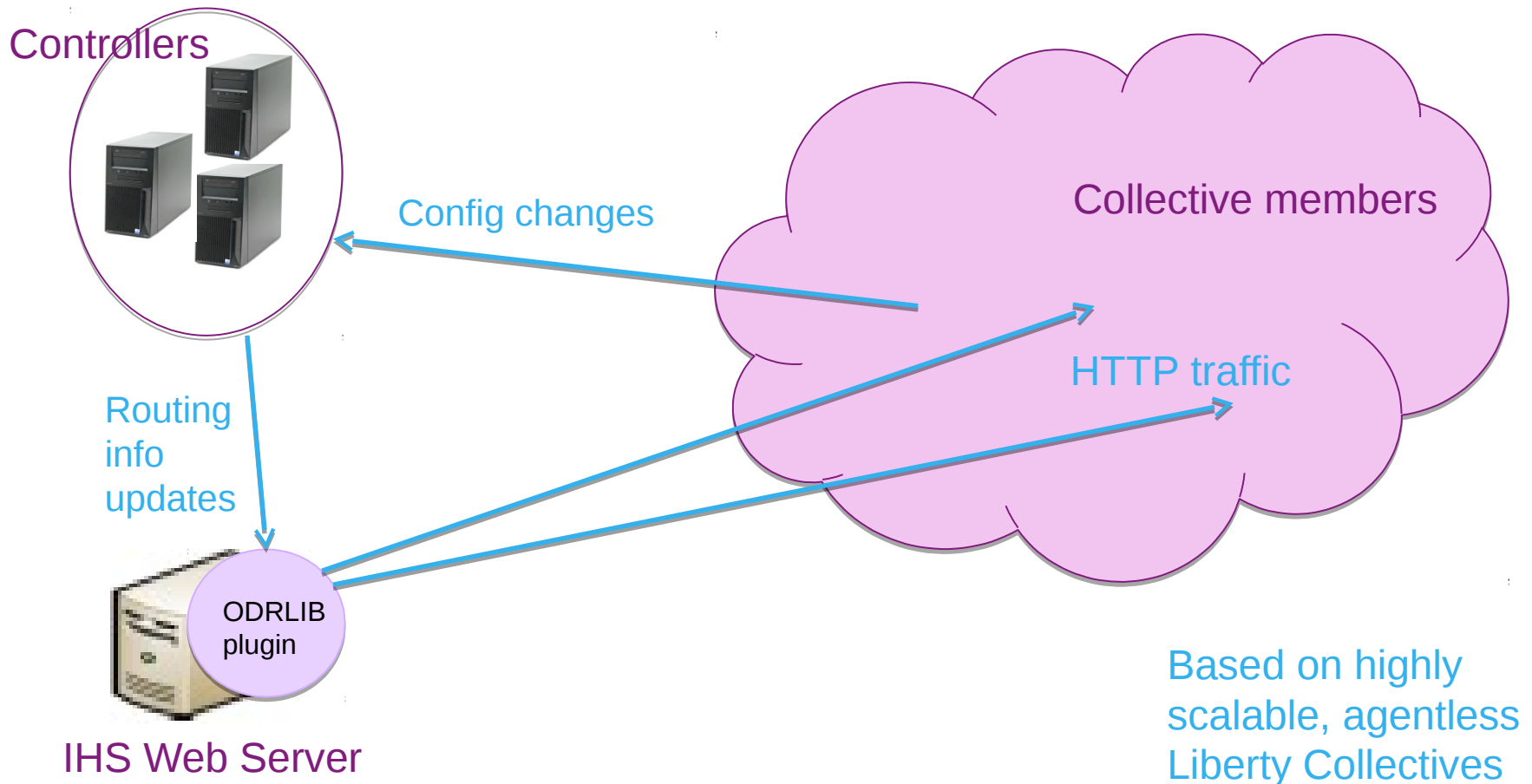
Dynamic updates to the runtime

- Dropin application install and update
- Configuration files also monitored for updates
 - All configuration changes are dynamic



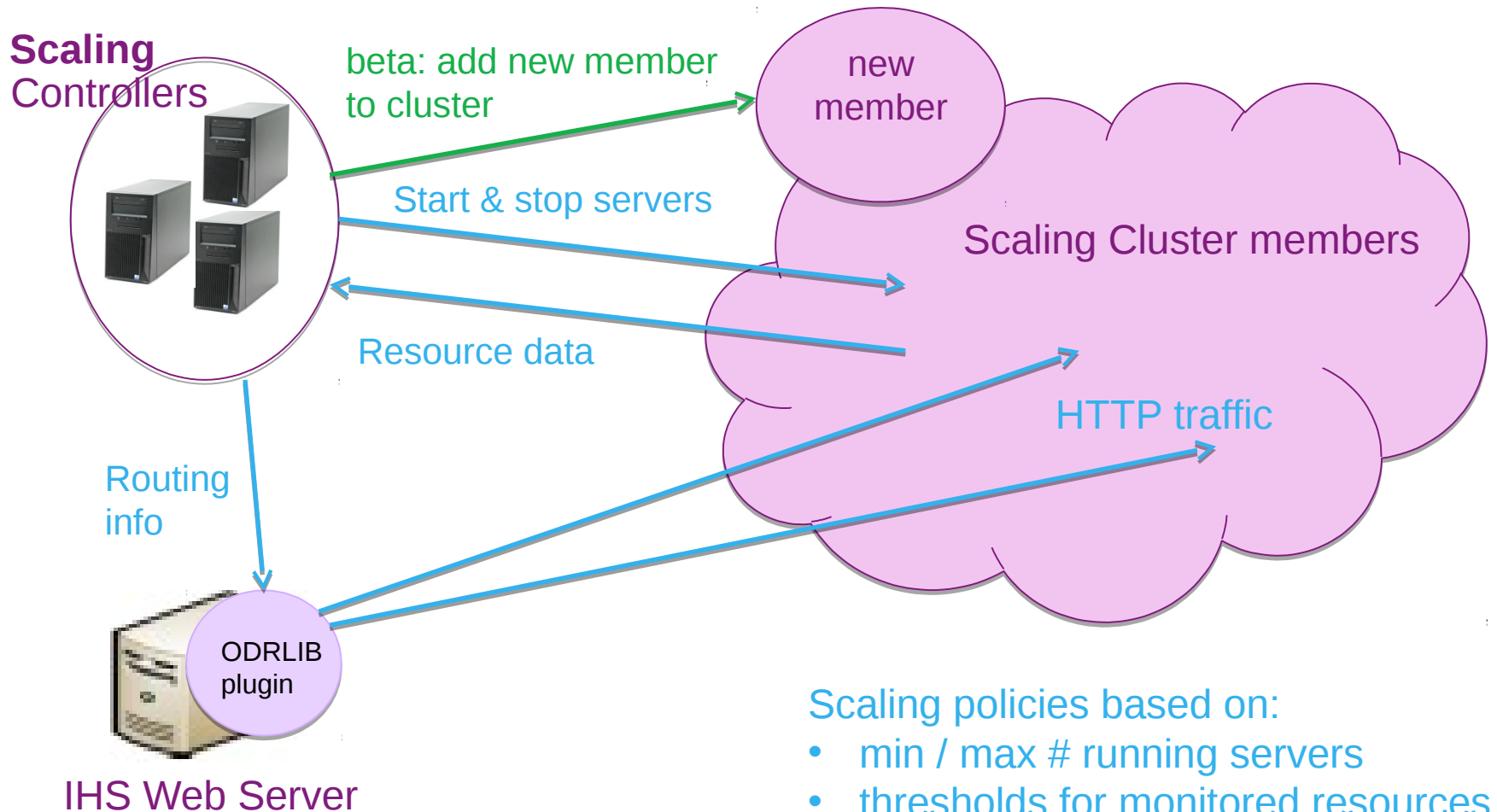
Dynamic routing

- Routing information in web server plugin can be dynamically updated as application endpoints change



Dynamic scaling

- Servers can be started and stopped per scaling policy



WAS Liberty workshop agenda 19/11/2015

Liberty Technical Overview

Discover the Liberty Profile - environment setup, application deployment and basic administration tasks

Management labs

- Managing Liberty collective with Admin Center
- Liberty Profile Administration using Jython Scripts
- Monitoring Liberty Profile
- Auto-Scaling Lab for Liberty Profile
- Using Apache httpd with SSL to a Liberty server

Development labs

- Liberty JSP and application security
- Liberty Migration Toolkit
- JDBC and Liberty Profile
- JMS, MDB and Liberty Profile
- Liberty extension and custom user registry
- WebCache and Liberty Profile
- WebSockets and Liberty Profile
- JAXWS Web Service and Liberty Profile
- Java Batch lab and Liberty Profile