

Simplified JMS Messaging Support for Liberty Profile

Lohitashwa Thyagaraj
WebSphere Messaging – Product Architect
IBM Master Inventor



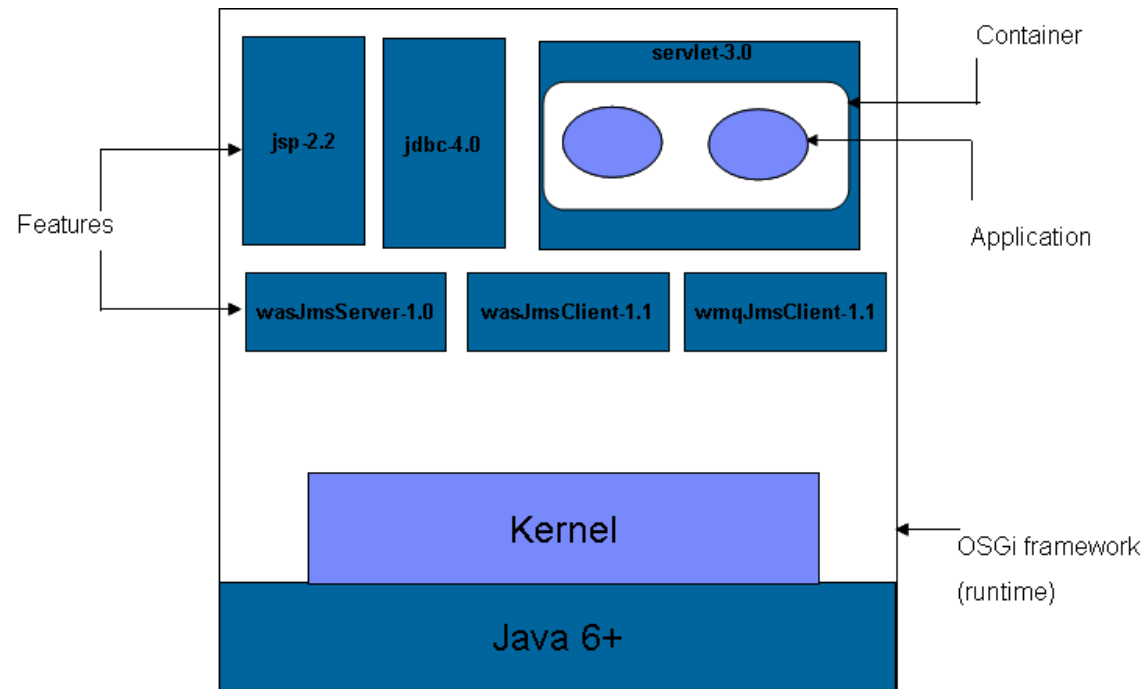
IBM's statements regarding its plans, directions, and intent are subject to change or withdrawal without notice at IBM's sole discretion.

Information regarding potential future products is intended to outline our general product direction and it should not be relied on in making a purchasing decision.

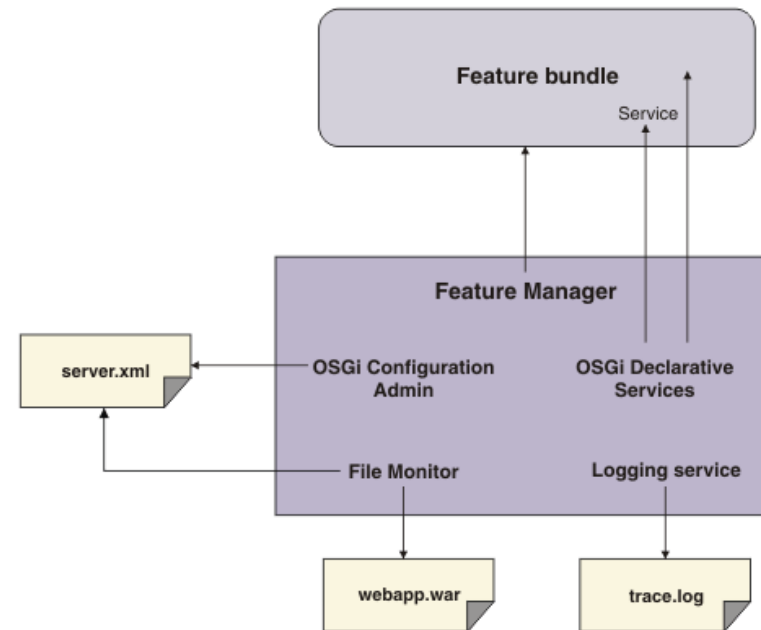
The information mentioned regarding potential future products is not a commitment, promise, or legal obligation to deliver any material, code or functionality. Information about potential future products may not be incorporated into any contract. The development, release, and timing of any future features or functionality described for our products remains at our sole discretion.

Performance is based on measurements and projections using standard IBM benchmarks in a controlled environment. The actual throughput or performance that any user will experience will vary depending upon many factors, including considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve results similar to those stated here.

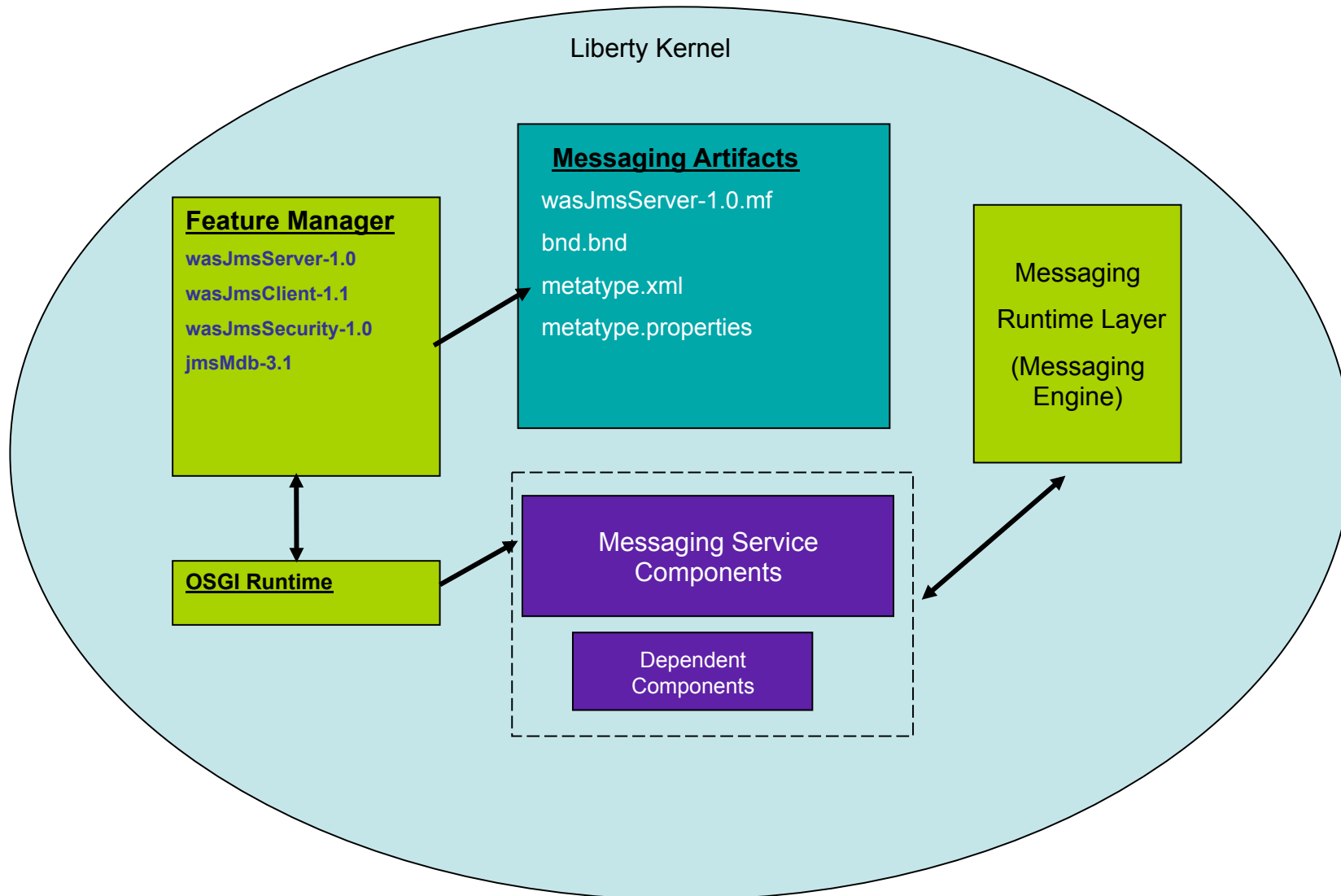
- Provides light weight JMS server
- JMS 1.1 compliant
- Composable, flexible and dynamic messaging service
- Supports WebSphere MQ as the messaging provider
- Supports Message Driven Bean (MDB) deployment
- Interoperability with full profile WebSphere Application Server
- Supports transactions, security and persistence

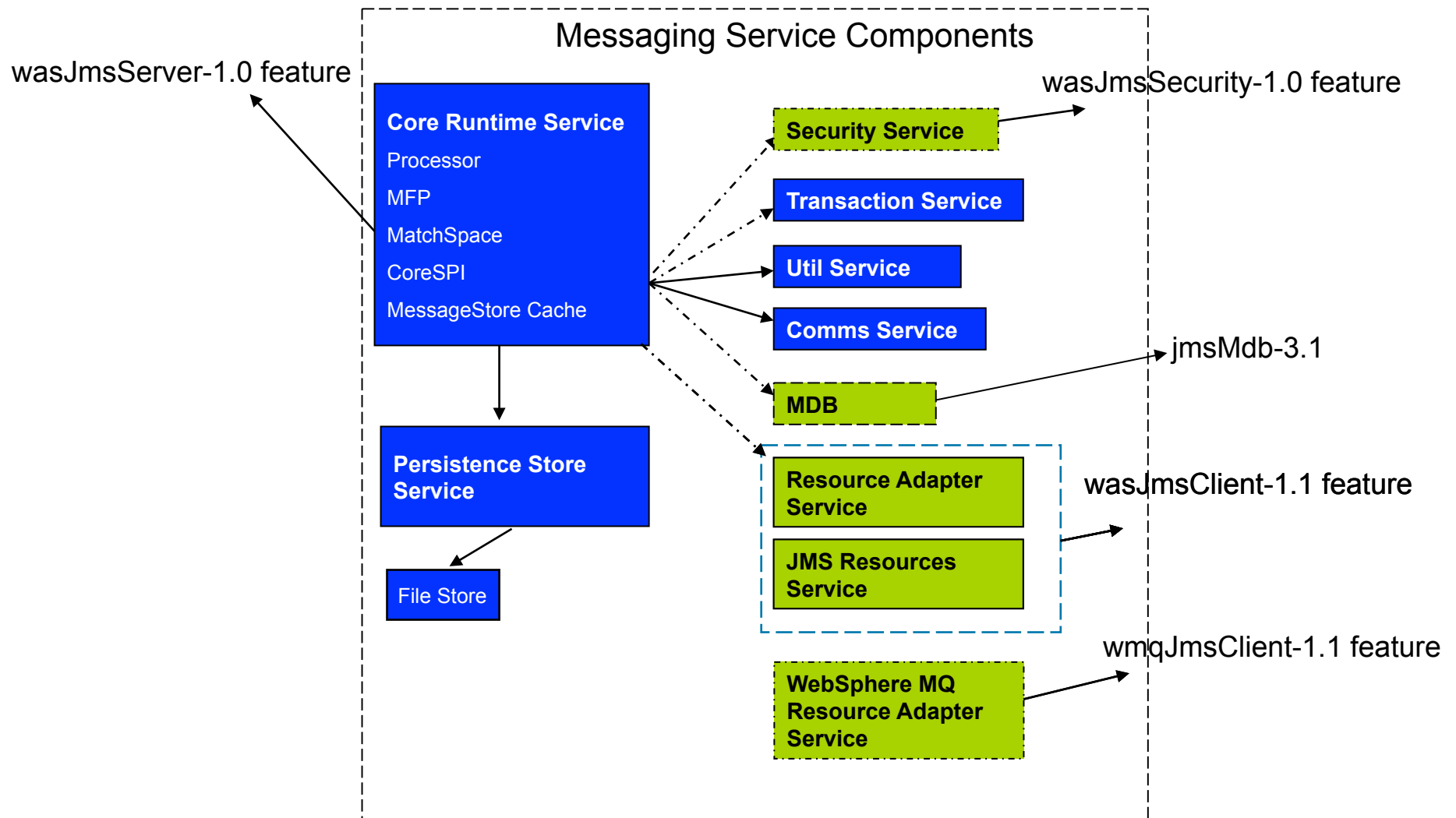


- The Liberty profile is a highly composable and dynamic runtime environment
- OSGi services are used to manage component lifecycles, and the injection of dependencies and configuration
- The server process comprises a single JVM, the Liberty kernel, and any number of optional features.
- The feature code and most of the kernel code runs as OSGi bundles within an OSGi framework.



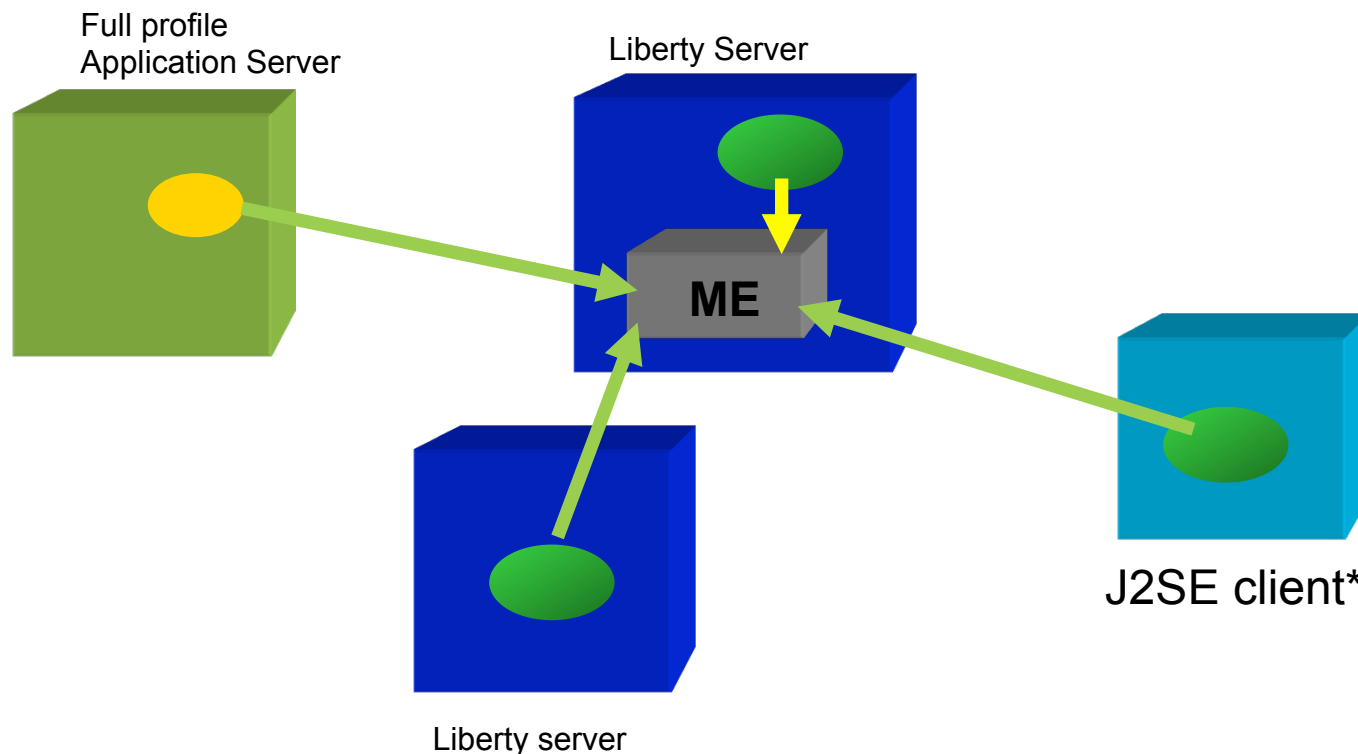
- Features are specified in the system configuration files that are the server.xml file and any other included files
- The server configuration files populate the OSGi Configuration Admin service, which injects the feature configuration into the feature manager service
- The feature manager maps each feature name to a list of bundles that provide the feature. The bundles are installed into the OSGi framework and started
- The feature manager responds to configuration changes by dynamically adding and removing features while the server is running



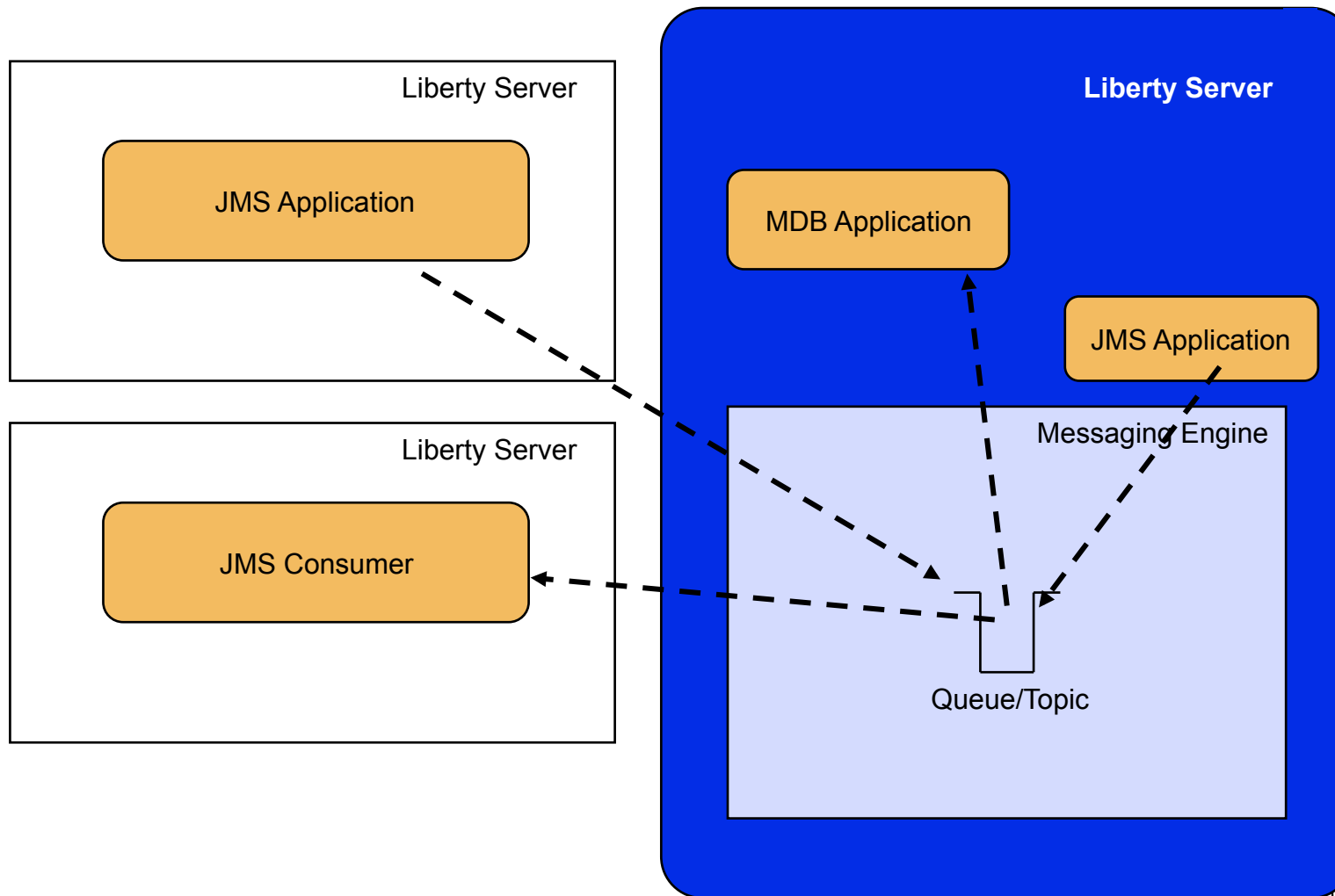


SIBus	Liberty Messaging
Single Jar - com.ibm.ws.sib.server.jar	Multiple Bundles & Service
SIBus jar size – 12.8 MB	Core runtime size < 5MB
Resource Adapter size sibc.jmsra.rar – 3.8 MB WebSphere MQ RA – 8.3 MB	Liberty Messaging Resource Adapter Bundle size – 1.5 MB
Tightly coupled	Loosely coupled

- An application can run:
 - in a liberty profile that is hosting the messaging engine
 - in a liberty profile that is not hosting the messaging engine
 - in a full profile application server
 - in a J2SE client
- An application always connects directly to the messaging engine

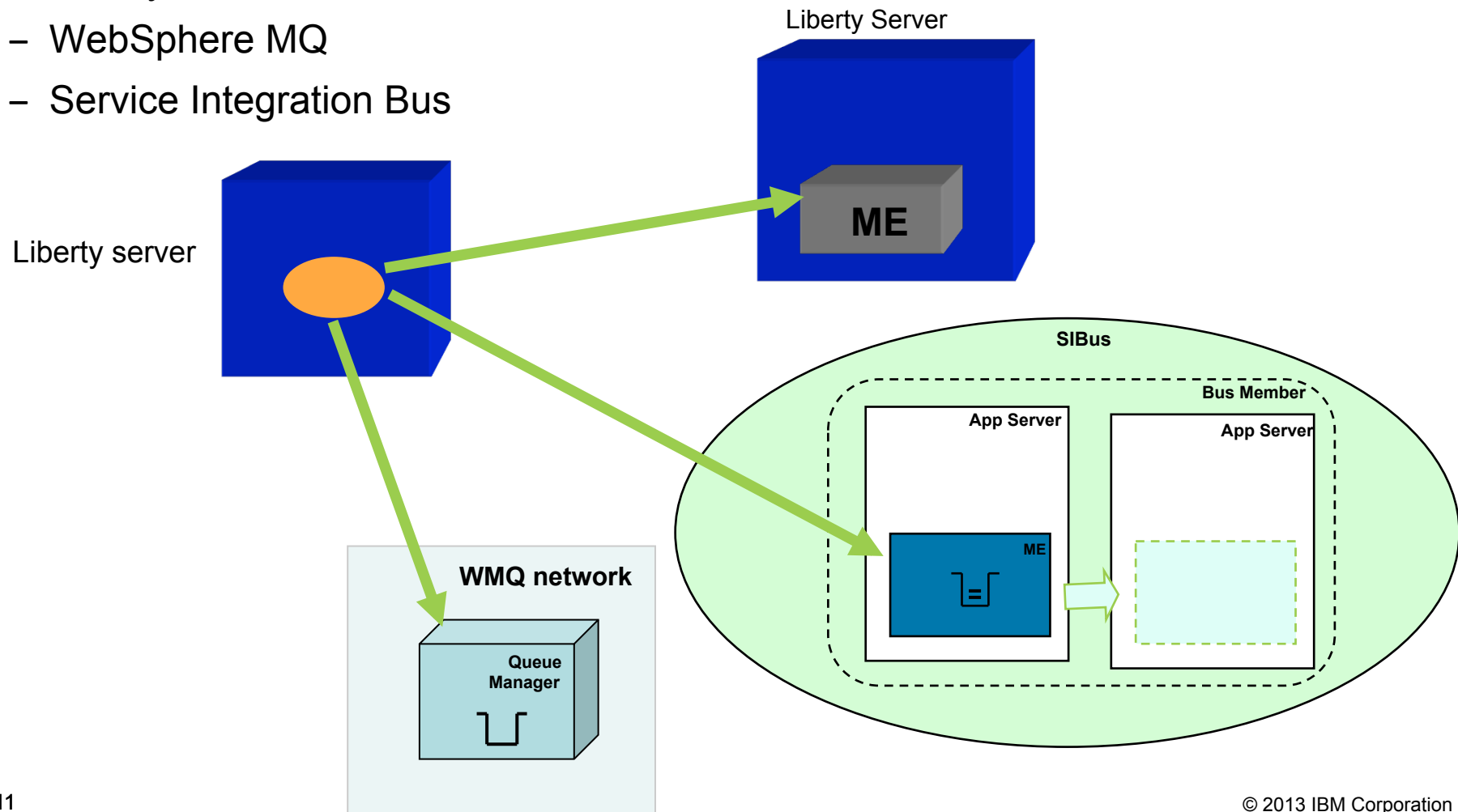


- In liberty messaging, all messages are sent and received from a messaging engine that localizes the destination



•Liberty provides first class support for the following JMS providers

- Liberty embedded JMS server
- WebSphere MQ
- Service Integration Bus



- **wasJmsServer-1.0**
 - Instantiates the messaging engine runtime
- **wasJmsClient-1.1**
 - Provides client libraries and defining the JMS resources
- **wasSecurity-1.0**
 - Enables messaging engine to be secured
- **wmqJmsClient-1.1**
 - Enables connectivity to WebSphere MQ
- **jmsMdb-3.1**
 - Provides capabilities to configure and deploy MDB within Liberty

JMS Server (wasJmsServer-1.0)

- Provides the runtime capabilities for messaging server
 - Manages connections, persistence, transactions, security
- Flexible, Dynamic and Light weight
- Allows creation of destinations and managing authorization for messaging resources

```
<server description="Server with JMS">
```

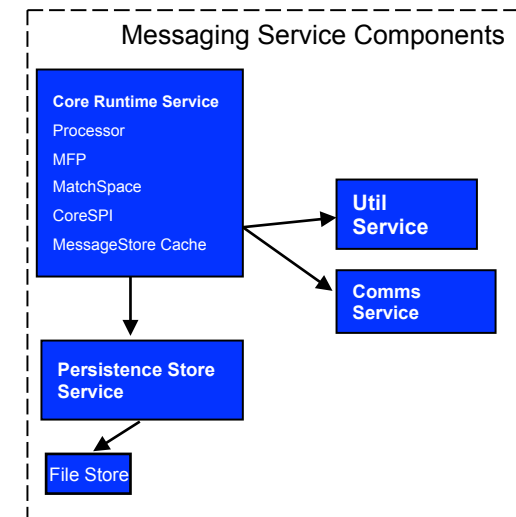
```
<featureManager>
```

```
<feature>servlet-3.0</feature>
```

```
<feature>wasJmsServer-1.0</feature>
```

```
</featureManager>
```

```
</server>
```



```
<server description="Server with JMS">
  <featureManager>
    <feature>servlet-3.0</feature>
    <feature>wasJmsServer-1.0</feature>
  </featureManager>
  <messagingEngine id="defaultME">
    <queue id="CreditQ" receivedAllowed="false"
      maxMessageDepth="1000"/>
    <topicSpace id="Sports.Topic.Space"/>
    <fileStore id="fileStore" logFileSize="20" fileStoreSize="100"/>
  </messagingEngine>
</server>
```

- Provides the definition of the JMS Resources and registers the objects in the repository
 - Connection Factory (`jmsConnectionFactory`)
 - Queue Connection Factory (`jmsQueueConnectionFactory`)
 - Topic Connection Factory (`jmsTopicConnectionFactory`)
 - Activation Spec (`jmsActivationSpec`)
 - JMS Queue (`jmsQueue`)
 - JMS Topic (`jmsTopic`)

<featureManager>

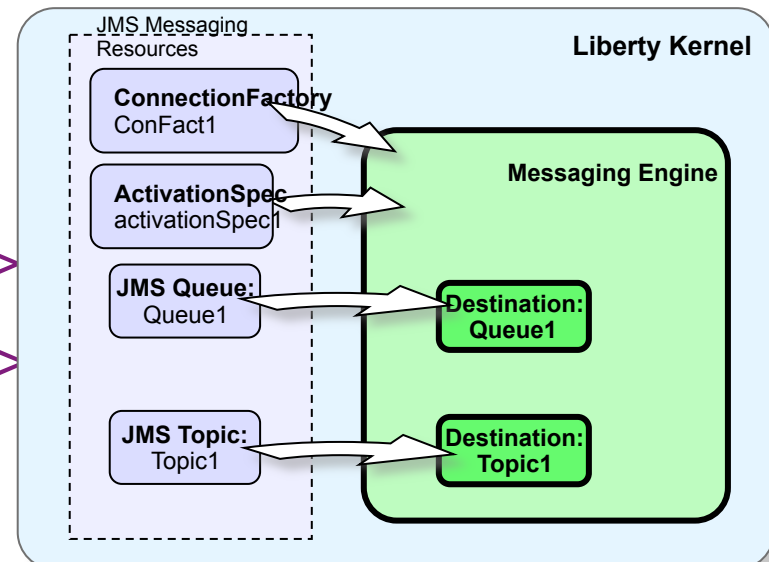
<feature>servlet-3.0</feature>

<feature>wasJmsServer-1.0</feature>

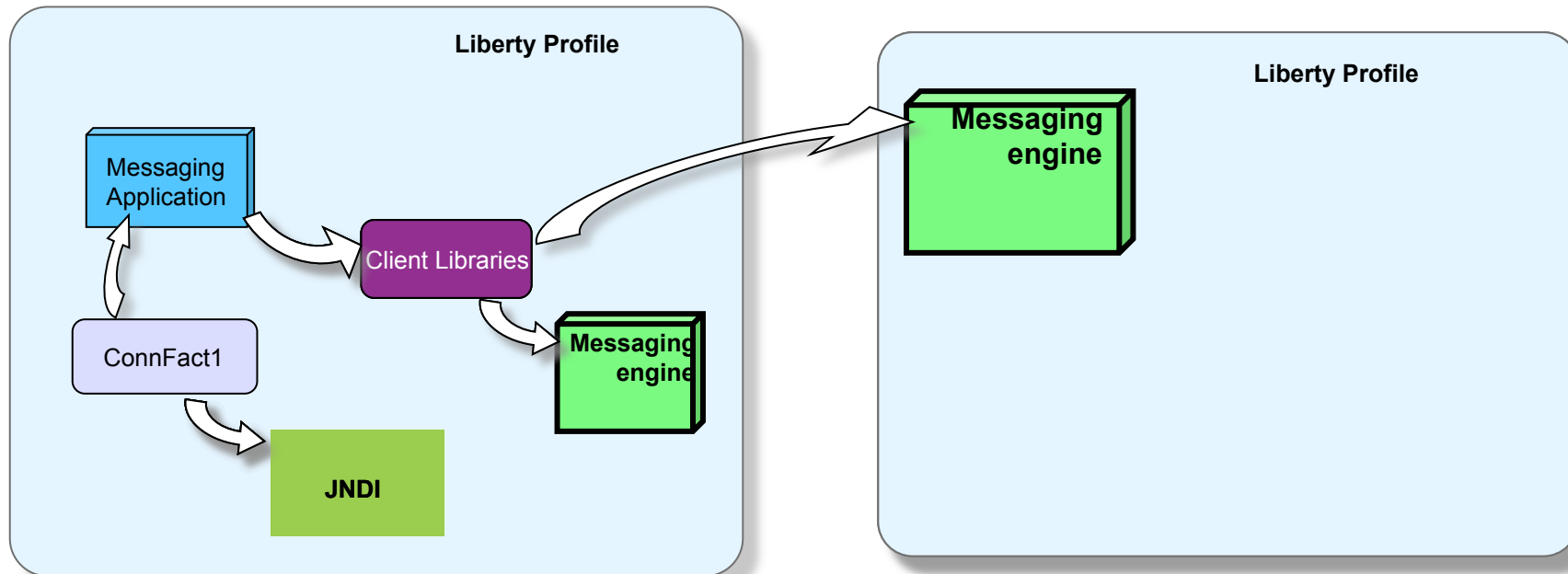
<feature>**wasJmsClient-1.1**</feature>

<feature>**jndi-1.0**</feature>

15 </featureManager>



- Provides the client libraries required for the JMS application to interact with the messaging engine
- Liberty supports two ways of application connectivity
 - IN-MEMORY
 - TCP/IP



- JMS applications can either connect on a secured or non-secured port
- By default JMS server listens for incoming data on 7276 (unsecured) and 7286 (secured). Default ports can be overridden by using the *wasJmsEndPoint* option in server.xml

```
<featureManager>
```

```
    <feature>wasJmsServer-1.0</feature>
```

```
</featureManager>
```

```
<wasJmsEndPoint id="InboundJmsEndpoint" host="*">
```

```
    wasJmsPort="9011"
```

```
    wasJmsSSLPort="9100" />
```

```
<messagingEngine id="defaultME">
```

```
    <queue id="libertyQ"/>
```

```
</messagingEngine>
```

- By default a JMS client application tries to connect to port 7276 (unsecured) and 7286 (secured)
- Use **remoteServerAddress** option in the connection factory to make JMS application connect to a remote messaging engine

```
<jmsQueueConnectionFactory jndiName="myQCF"  
connectionManagerRef="ConMgr4">  
    <properties.wasJms  
        remoteServerAddress="lserver.ibm.com:9011:BootstrapBasicMessaging"/  
    >  
</jmsQueueConnectionFactory>
```

- JMS client connections can be made secured by using custom chains and using SSL
- Use the **wasJmsOutbound** option to specify custom outgoing connection properties

```
<wasJmsOutbound id="jmsSecureCustomChain" useSSL="true">  
    <sslOptions sslRef="CustomSslOptions" />  
</wasJmsOutbound>  
  
<jmsQueueConnectionFactory jndiName="jms/QCF"  
connectionManagerRef="ConMgr6">  
    <properties.wasJms          remoteServerAddress="localhost:  
7286:jmsSecureCustomChain" />  
</jmsQueueConnectionFactory>
```

Server side

```
<ssl id="CustomSslOptions" keyStoreRef="defaultKeyStore"/>
```

```
<keyStore location="serverKey.jks" id="defaultKeyStore"  
password="key123" />
```

```
<wasJmsEndpoint id="InboundJmsEndpoint" host="*"
```

```
    wasJmsPort="7276"
```

```
    wasJmsSSLPort="7286" >
```

```
        <sslOptions sslRef="CustomSslOptions"/>
```

```
</wasJmsEndpoint>
```

Client side

```
<ssl id="CustomSslOptions" keyStoreRef="defaultKeystore"/>
```

```
<keyStore id="defaultKeystore" location="clientKey.jks" type="JKS"  
password="key123"/>
```

```
<wasJmsOutbound id="jmsSecureCustomChain" useSSL="true">
```

```
    <sslOptions sslRef="CustomSslOptions" />
```

```
</wasJmsOutbound>
```

```
<jmsQueueConnectionFactory jndiName="jms/QCF"  
connectionManagerRef="ConMgr6">
```

```
    <properties.wasJms userName="persona99" password="ppersona99"  
remoteServerAddress="localhost:7286:jmsSecureCustomChain" />
```

```
</jmsQueueConnectionFactory>
```

Securing Liberty Messaging (wasJmsSecurity-1.0) Impact2013

- Enables the messaging engine to be secured
- Authentication
 - Validates if user has permission to *connect* to the messaging engine
- Authorization
 - Validates if user has sufficient *roles/permissions* to access the messaging resources (queues/topics)

<featureManager>

<feature>servlet-3.0</feature>

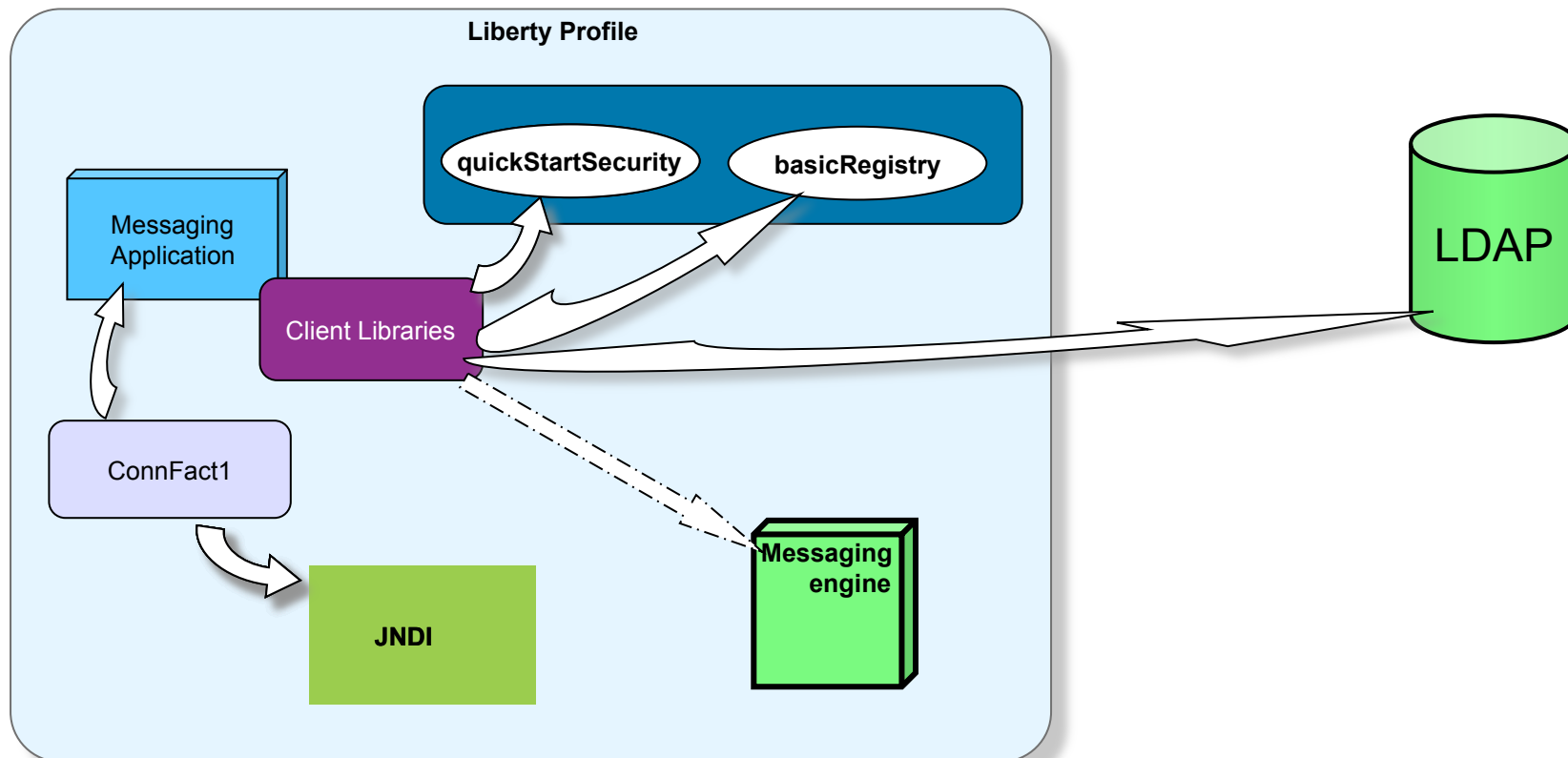
<feature>wasJmsServer-1.0</feature>

<feature>wasJmsClient-1.1</feature>

<feature>**wasJmsSecurity-1.0**</feature>

</featureManager>

- Liberty supports 3 types of user registries
 - quickStartSecurity : Simplest and easiest to configure
 - basicRegistry : Provides groups and member creations
 - ldapRegistry : Connectivity to external ldap registries



- Different way of defining authentication information
 - Application
 - Connection Factory (Component)
 - AuthData (Container)

```
// create connection factory
ConnectionFactory conFac = (ConnectionFactory) context.lookup("
```

server.xml

```
<authData user="libertyUser" password="{xor}Lz4sLCgwLTs=" id="testAuthData"/>

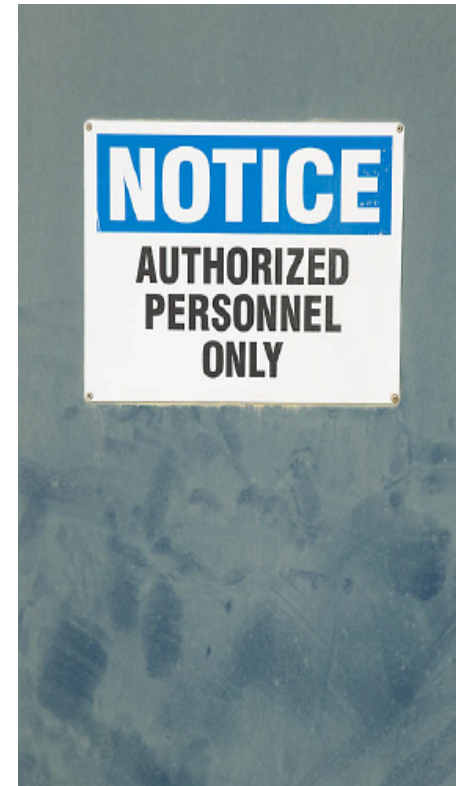
<jmsConnectionFactory jndiName="jms/confFactory"
connectionManagerRef="ConMgr6" containerAuthDataRef="testAuthData">
  <properties.wasJms userName="libertyUser" password="password"/>
</jmsConnectionFactory>
<connectionManager id="ConMgr6" maxPoolSize="5"/>
```

web.xml

```
<resource-ref>
  <res-ref-name>jms/confFactory</res-ref-name>
  <res-type>javax.jms.ConnectionFactory</res-type>
  <res-auth>Container</res-auth>
  <lookup-name> jms/confFactory </lookup-name>
</resource-ref>
```

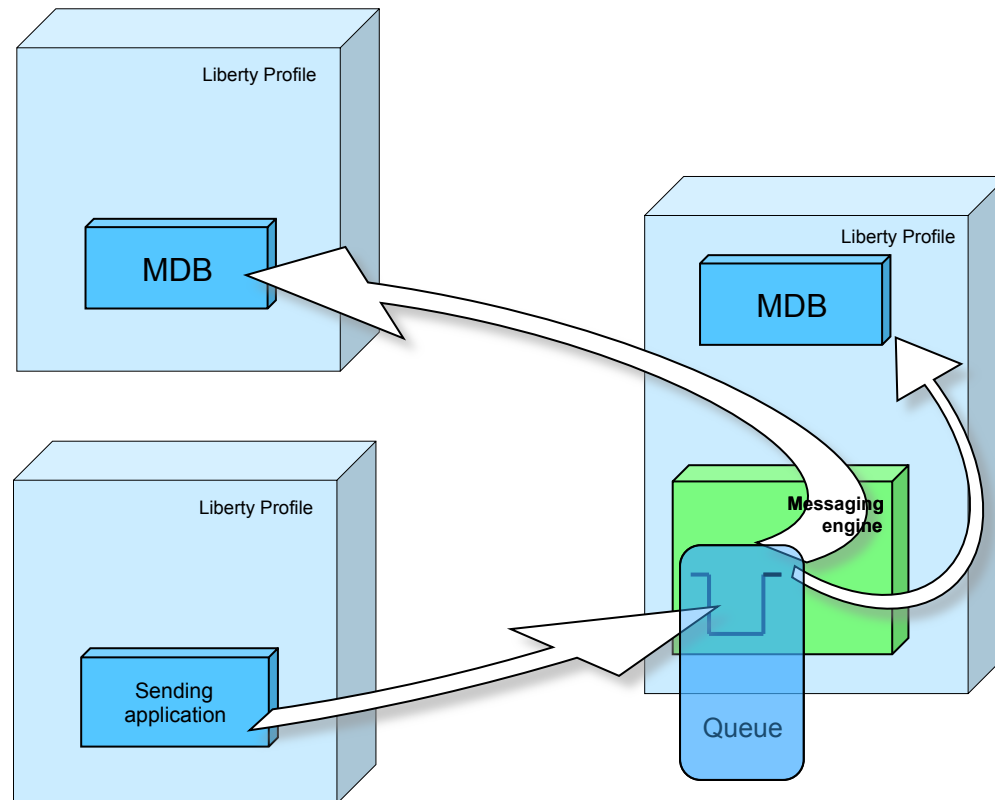

- Access to messaging resources is limited based on roles and credentials
- Only authenticated users will be authorized to access the resources
- Unless specified default access can be “allow all” or “deny all”

```
<messagingSecurity>  
  <role name="developer">  
    <queuePermission name="Q1">  
      <action>ALL</action>  
    </queuePermission>  
  </role>  
  <user name="libertyUser"/>  
</messagingSecurity>
```



Message Driven Beans (jmsMdb-3.1)

- Liberty supports deployment of MDB's to asynchronously consume messages from messaging engine
- Supports both *Point to Point* and *Publish/Subscribe*
- Does not require EJB container to deploy MDB



- MDB can be configured within the web project itself
- MDB utilizes Activation specifications to communicate with the messaging engine

```
<server description="Server with JMS">  
  <featureManager>  
    <feature>wasJmsServer-1.0</feature>  
    <feature>wasJmsClient-1.1</feature>  
    <feature>jmsMdb-3.1</feature>  
  </featureManager>  
</server>
```

```
<jmsActivationSpec id="jms/ActSpec" jndiName="jms/ActSpec" >  
  <properties.wasJms destinationRef="jms/Q"/>  
</jmsActivationSpec>
```

```
<jmsQueue id="jms/Q" jndiName="jms/Q" >  
  <properties.wasJms queueName="Q1"/>  
</jmsQueue>
```

Liberty provides flexibility in the way MDB's can be configured and deployed

- Direct activation specification usage

- Annotation

server.xml

```
<jmsActivationSpec id="JMSSample/JMSSampleMDB">
```

```
@MessageDriven(  
    name = "JMSSampleMDB", activationConfig = {
```

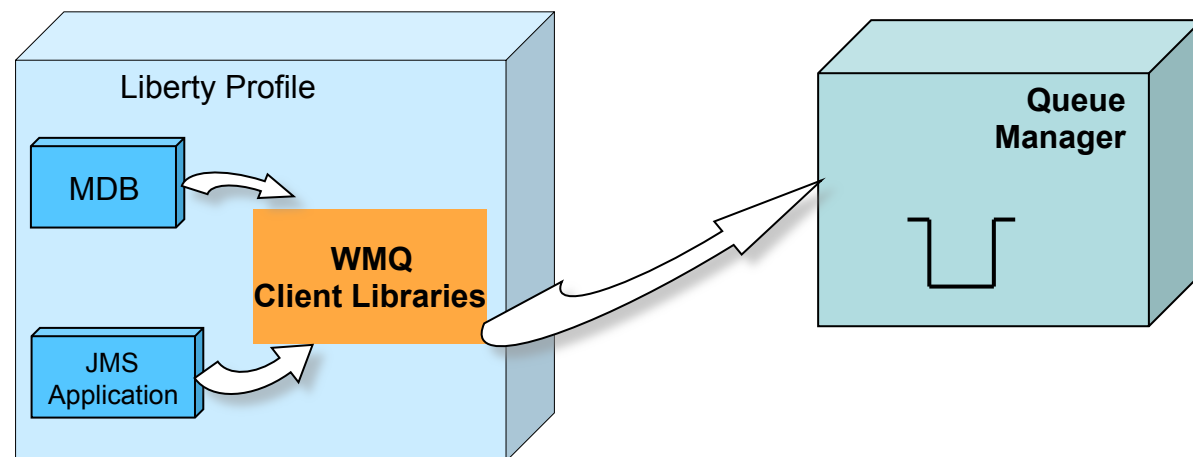
server.xml

```
<jmsActivationSpec id="PriceChangeAS">  
    <properties.wasJms destinationRef="jms/TriggerQ" /> </jmsActivationSpec>  
<jmsQueue id="jms/TriggerQ" jndiName="jms/TriggerQ">  
    <properties.wasJms queueName="Q1"/>  
</jmsQueue>
```

ibm-ejb-jar-bnd.xml

```
<message-driven name="PriceChangeMDBBean">  
    <jca-adapter activation-spec-binding-name="PriceChangeAS" destination-  
        binding-name="jms/TriggerQ" />  
</message-driven>
```

- Liberty supports configuring WebSphere MQ as the external JMS messaging provider
- JMS Applications running within Liberty profile can communicate with WMQ to perform both
 - Point to Point
 - Publish/Subscribe
- Supports both type of connectivity
 - BINDING mode
 - CLIENT mode



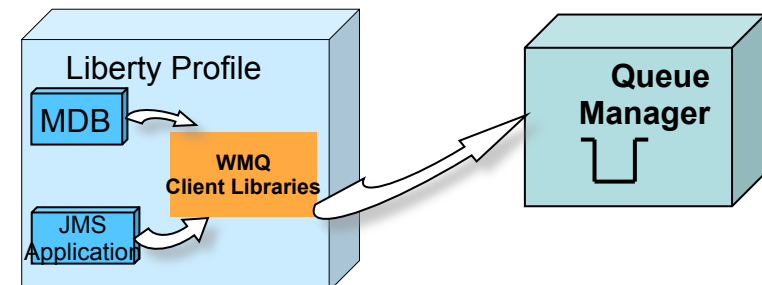
- WMQ client libraries are available to applications using the wmqJmsClient-1.1 feature

```
<featureManager>  
  <feature>wmqJmsClient-1.1</feature>  
</featureManager>
```

- Configuration very similar to liberty messaging configuration

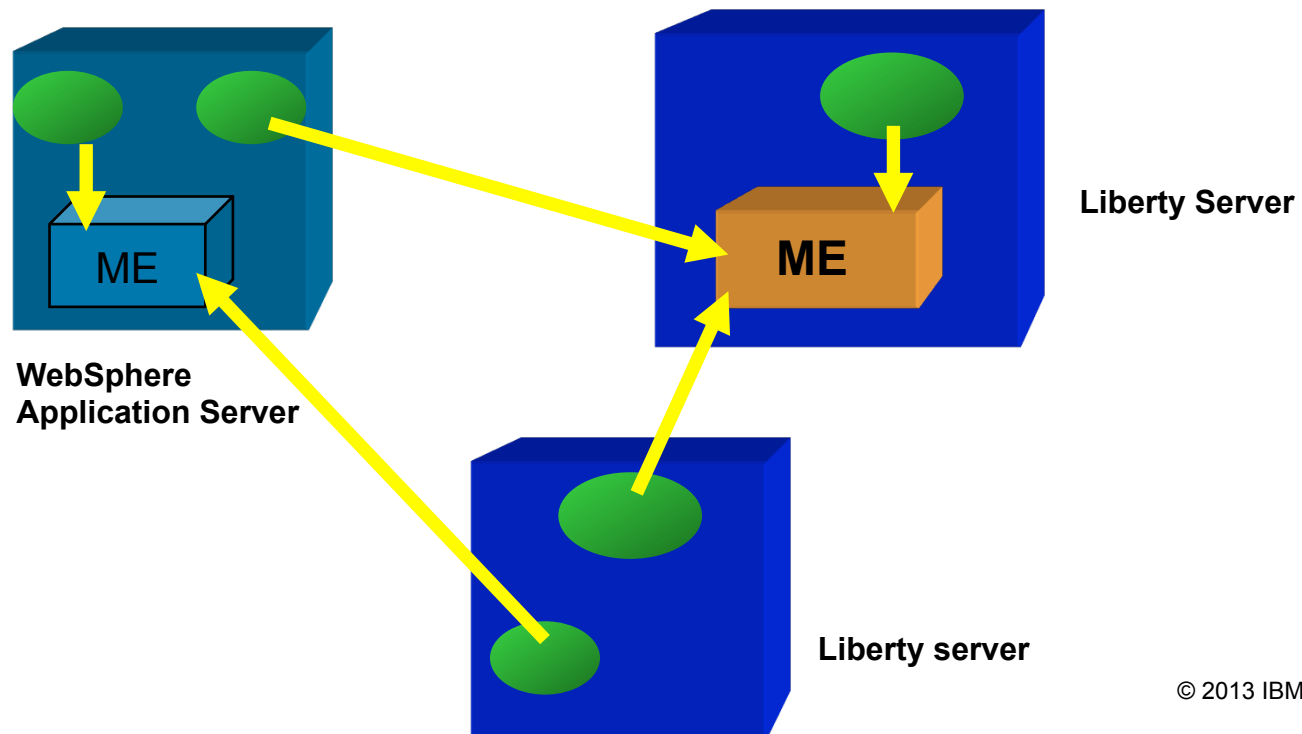
```
<jmsConnectionFactory id="jms/conFac connectionManagerRef="ConMgr1">  
  <properties.wmqJms host="wmqserver.in.ibm.com port="1414"/>  
</jmsConnectionFactory>
```

```
<jmsQueue id="eis/queue1" jndiName="eis/queue1">  
  <properties.wmqJms baseQueueName="Q1"  
    baseQueueManagerName="qm1"/>  
</jmsQueue>
```

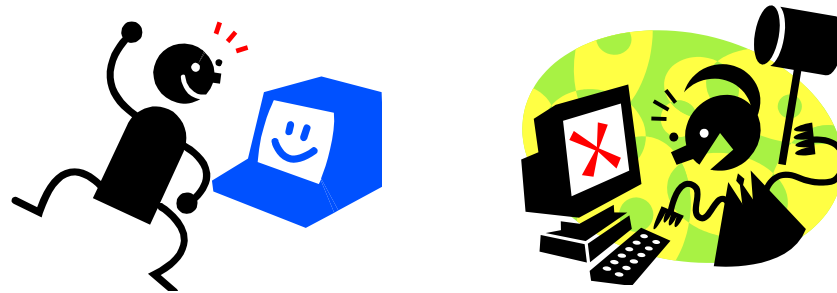


Interoperability between Liberty and Full Profile Application Server

- Application deployed in Liberty profile can connect to SIBus
- Application deployed in full profile WAS can connect to Liberty messaging
- Does not support ME to ME communication between Liberty and SIBus



And now the demo....



- Provides light weight single stand alone JMS server
- Composable, flexible and dynamic messaging service
- Supports WebSphere MQ as the messaging provider
- Supports Message Driven Bean (MDB) deployment
- Interoperability with full profile WebSphere Application Server

Don't forget to submit your Impact session and speaker feedback!

- Your feedback is very important to us – we use it to improve next year's conference
- Go to the Impact 2013 SmartSite (<http://impactsmartsite.com>):
 - Use the session ID number to locate the session
 - Click the “Take Survey” link
 - Submit your feedback

Legal Disclaimer

- © IBM Corporation 2013. All Rights Reserved.
- The information contained in this publication is provided for informational purposes only. While efforts were made to verify the completeness and accuracy of the information contained in this publication, it is provided AS IS without warranty of any kind, express or implied. In addition, this information is based on IBM's current product plans and strategy, which are subject to change by IBM without notice. IBM shall not be responsible for any damages arising out of the use of, or otherwise related to, this publication or any other materials. Nothing contained in this publication is intended to, nor shall have the effect of, creating any warranties or representations from IBM or its suppliers or licensors, or altering the terms and conditions of the applicable license agreement governing the use of IBM software.
- References in this presentation to IBM products, programs, or services do not imply that they will be available in all countries in which IBM operates. Product release dates and/or capabilities referenced in this presentation may change at any time at IBM's sole discretion based on market opportunities or other factors, and are not intended to be a commitment to future product or feature availability in any way. Nothing contained in these materials is intended to, nor shall have the effect of, stating or implying that any activities undertaken by you will result in any specific sales, revenue growth or other results.
- If the text contains performance statistics or references to benchmarks, insert the following language; otherwise delete:
Performance is based on measurements and projections using standard IBM benchmarks in a controlled environment. The actual throughput or performance that any user will experience will vary depending upon many factors, including considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve results similar to those stated here.
- If the text includes any customer examples, please confirm we have prior written approval from such customer and insert the following language; otherwise delete:
All customer examples described are presented as illustrations of how those customers have used IBM products and the results they may have achieved. Actual environmental costs and performance characteristics may vary by customer.
- Please review text for proper trademark attribution of IBM products. At first use, each product name must be the full name and include appropriate trademark symbols (e.g., IBM Lotus® Sametime® Unyte™). Subsequent references can drop "IBM" but should include the proper branding (e.g., Lotus Sametime Gateway, or WebSphere Application Server). Please refer to <http://www.ibm.com/legal/copytrade.shtml> for guidance on which trademarks require the ® or ™ symbol. Do not use abbreviations for IBM product names in your presentation. All product names must be used as adjectives rather than nouns. Please list all of the trademarks that you use in your presentation as follows; delete any not included in your presentation. IBM, the IBM logo, Lotus, Lotus Notes, Notes, Domino, Quickr, Sametime, WebSphere, UC2, PartnerWorld and Lotusphere are trademarks of International Business Machines Corporation in the United States, other countries, or both. Unyte is a trademark of WebDialogs, Inc., in the United States, other countries, or both.
- If you reference Adobe® in the text, please mark the first use and include the following; otherwise delete:
Adobe, the Adobe logo, PostScript, and the PostScript logo are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, and/or other countries.
- If you reference Java™ in the text, please mark the first use and include the following; otherwise delete:
Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.
- If you reference Microsoft® and/or Windows® in the text, please mark the first use and include the following, as applicable; otherwise delete:
Microsoft and Windows are trademarks of Microsoft Corporation in the United States, other countries, or both.
- If you reference Intel® and/or any of the following Intel products in the text, please mark the first use and include those that you use as follows; otherwise delete:
Intel, Intel Centrino, Celeron, Intel Xeon, Intel SpeedStep, Itanium, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.
- If you reference UNIX® in the text, please mark the first use and include the following; otherwise delete:
UNIX is a registered trademark of The Open Group in the United States and other countries.
- If you reference Linux® in your presentation, please mark the first use and include the following; otherwise delete:
Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both. Other company, product, or service names may be trademarks or service marks of others.
- If the text/graphics include screenshots, no actual IBM employee names may be used (even your own), if your screenshots include fictitious company names (e.g., Renovations, Zeta Bank, Acme) please update and insert the following; otherwise delete: All references to [insert fictitious company name] refer to a fictitious company and are used for illustration purposes only.