

# Solving Integration Problems With JBoss Fuse

LEARN. NETWORK. EXPERIENCE OPEN SOURCE

redhat.com/summit

Scott England Sullivan - Principle Integration Architect Peter Blinstrubas - Middleware Solutions Architect



# Agenda

- Customer's Problem
- Fuse In 0 to 60 MPH
- Architecture
- Steps to Multitenacy
- What's Next?
- Conclusion



# Customer Quote

# "Camel is the right way to do integration!"

# The Story

- Healthcare Company
  - Modernize SOA
  - Gain Agility
  - Reduce Cost
  - Monolith to Multitenancy





# **Customer's Problem**

# Problem

Migrate from a COTS product while maintaining operational parity and the capability to evolve into multitenancy.

# Fuse in 0 to 60 MPH

### **JBoss Fuse Architecture**



Fuse Cartridge

**OPEN**SHIFT<sup>®</sup> by Red Hat<sup>®</sup>

A-MQ Cartridge



### **Middleware Integration**



### JBoss A-MQ



- Small-footprint messaging system
- Built for interoperability:
- persistence, clustering, failover, security,etc

JBoss F	use
---------	-----

	DEVELOPMENT WITH JBDS	
	Services (Apache CXF)	
	Integration (Apache Camel)	
	Messaging (JBoss A-MQ)	
JBOSS FUSE		
MANAGEMENT WITH JON & FUSE MANAGEMENT CONSOLE		

- Provides Core Integration functionality
- EIP as a first-order concept
- Simplifies Integration, Transformation, Mediation

# **Apache Camel**



### **50+ Enterprise Integration Patterns**

÷



Designing, Building, and Deploying Messaging Solutions

GREGOR HOHPE BOBBY WOOLF

WITH CONTRIBUTIONS BY KYLE BROWN CONRAD F. D'CRUZ MARTIN FOWLER SEAN NEVILLE MICHAEL J. RETTIG JONATHAN SIMON



Forewords by John Crupi and Martin Fowler



Claus Ibsen Jonathan Anstey Forewords by Gregor Hober and James Strad

http://camel.apache.org/eip

# A-MQ



- High performance, reliable message broker
   O Clustering and Fault Tolerance
- Myriad of connectivity options
   Native Java, C/C++, and .NET
   AMQP, MQTT, STOMP, and OpenWire
- Embedded and standalone deployment options



- ActiveMQ in Action
- Community website
  - O <u>http://activemq.apache.org/</u>



# Apache CXF

### **Standards Support**

### JSR Support

- JAX-WS Java API for XML-Based Web Services (JAX-WS) 2.0
   JSR-224
- Web Services Metadata for the Java Platform JSR-181
- JAX-RS The Java API for RESTful Web Services JSR-311
- SAAJ SOAP with Attachments API for Java (SAAJ) JSR-67

### **WS-\* & Related Specifications Support**

- Basic support: WS-I Basic Profile 1.1
- Quality of Service: WS-Reliable Messaging
- Metadata: WS-Policy, WSDL 1.1 Web Service Definition Language
- Communication Security: WS-Security, WS-SecurityPolicy, WS-SecureConversation, WS-Trust (partial support)
- Messaging Support: WS-Addressing, SOAP 1.1, SOAP 1.2, Message Transmission Optimization Mechanism (MTOM)



Apache CXF Web Service Development

Develop and deploy SOAP and RESTful Web Services

Naveen Balani Rajeev Hathi





# Managing Fuse with Fabric & Profiles







# Architecture





14

## Discovery



- Auditable and Reproducible
- Scalability and Isolatation
  - Grow to Multitenancy
  - Scale containers and services independently
- Standard VM environment.
- New to JBoss Fuse



### Key Recommendations

### - Fuse:

- Validated as the best path to multitenancy
- Encourages modular development practice
- Fabric delivers fine to coarse grained service definitions

### - Simplification:

• Simplify initial physical architecture w.r.t services/Domains

### - Mututable:

 Sets foundation for the level of scalability and isolation required by the customer in their environment

#### **JBOSS FUSE GATEWAY AND SERVICES DEPLOYMENT**



### JBOSS FUSE RECOMMENDED DEPLOYMENT TOPOLOGY





 $\blacksquare$  ....  $\blacksquare$  = Zookeeper Connections (Dashed Line)

VM

VM

# 8

# Fuse/Fabric – Simplified Architecture





# Steps to Multitenancy

# STEPS TO MULTITENANCY



- First deploy monolith application
- Identify candidates for partitioning
   Use metrics or reliability history to identify outliers
   Handle them on a case-by-case basis
- Consider isolating services for various forces, including:
  - Performance characteristics/contention
  - Business domain functionality
  - Deployment churn/update frequency
  - Criticality
  - Security (data at rest, in motion, etc)
  - SLAs
- Establish a CI/CD process

# Fuse/Fabric – Architecture with "n" Partitions





# Fuse/Fabric - Architecture "n" Partitions







# What's Next?



### Growth Plan

- Process for scaling to multiple organizations
  - shared Fuse platform
  - balance central automation with autonomous teams
  - scale up to 1000s of fabric profiles
- Migration between environments
- Works within existing recommended Architecture



### Next Steps

- Today
  - We have the foundation for multitenancy with modular services
  - Established a flexible, scalable, auditable, and repeatable architecture
- "Tomorrow"
  - Build out multitenancy environment
  - Establish a Dev/Ops tool chain
    - JBoss Fuse is a first class citizen in a CI/CD tool chain.

# Conclusion

# Conclusion



- Migration plan established
- Foundation for multitenancy
- Highly Scalable and Available
- Establish standard Dev/Ops Processes

### Thank You Any questions?



DEVIATION

JOIN THE OPEN SOURCE NATION

# AN OPEN SOURCE CONFERENCE, BY AND FOR DEVELOPERS ACROSS THE GLOBE June 23 - 26, 2015 | Boston, MA | USA



#### Agenda

- Customer's Problem
- Fuse In 0 to 60 MPH
- Architecture
- Steps to Multitenacy
- What's Next?
- Conclusion



**Customer Quote** 

"Camel is the right way to do integration!"







Migrate from a cots product while maintaining operational parity and the capability to evolve into multitenenancy

Establish an Integration Architecture that is scalable and provides isolation for fine grained and coarse grained services.





Middleware Integration			
JBoss A-MQ	JBoss Fuse		
DEVELOPMENT WITH JBDS WITH FUSE IDE	DEVELOPMENT WITH JBDS		
Apache Active MQ	Services (Apache CXF)		
	Integration (Apache Camel)		
Apache Karaf +Fuse Fabric	Messaging (JBoss A-MQ)		
JBOSS A-MQ	JBOSS FUSE		
MANAGEMENT WITH JON & JBOSS MANAGEMENT CONSOLE	MANAGEMENT WITH JON & FUSE MANAGEMENT CONSOLE		
Small-footprint messaging system     Provides Core Integration functionality			
• Built for interoperability:	EIP as a first-order concept		
• persistence, clustering, failover, security,etc	<ul> <li>Simplifies Integration, Transformation, Mediation</li> </ul>		





Supports publish/subscribe, point to point, message groups, out of band messaging and streaming, distributed transactions, ...

STOMP protocol enables Ruby, JS, Perl, Python, PHP, ActionScript, ...

Pre-integrated with open source integration and application frameworksDeep integration with Spring Framework, OSGi, and Java EE

### Apache CXF

#### **Standards Support**

#### JSR Support

- JAX-WS Java API for XML-Based Web Services (JAX-WS) 2.0 - JSR-224
- Web Services Metadata for the Java Platform JSR-181
- JAX-RS The Java API for RESTful Web Services JSR-311
- SAAJ SOAP with Attachments API for Java (SAAJ) JSR-67

#### WS-\* & Related Specifications Support

- Basic support: WS-I Basic Profile 1.1
- Quality of Service: WS-Reliable Messaging
- Metadata: WS-Policy, WSDL 1.1 Web Service Definition Language
- Communication Security: WS-Security, WS-SecurityPolicy, WS-SecureConversation, WS-Trust (partial support)
- Messaging Support: WS-Addressing, SOAP 1.1, SOAP 1.2, Message Transmission Optimization Mechanism (MTOM)

**Apache CXF** Web Service Development

Soap and REST stack



PACKT





#### Discovery



- Auditable and Reproducible
- Scalability and Isolatation
  - Grow to Multitenancy
  - Scale containers and services independently
- Standard VM environment.
- New to JBoss Fuse

#### Key Recommendations

#### - Fuse:

- Validated as the best path to multitenancy
- Encourages modular development practice
- Fabric delivers fine to coarse grained service definitions

#### - Simplification:

- Simplify initial physical architecture w.r.t services/Domains
- Mututable:
  - Sets foundation for the level of scalability and isolation required by the customer in their environment







Simplified and efficient single JVM per VM

- In the absence of hard data, always start here
- Used throughout industry verticals with success

Triple Active at both GW/Services and Message Broker Layers

 Eliminates single point of failure condition if one instance is removed for maintenance

Fabric Separation of concerns

- Isolates message flows from management
- Creates independently scalable partitions



#### STEPS TO MULTITENANCY



- First deploy monolith application
- Identify candidates for partitioning
   Use metrics or reliability history to identify outliers
   Handle them on a case-by-case basis
- Consider isolating services for various forces, including:
  - Performance characteristics/contention
  - Business domain functionality
  - Deployment churn/update frequency
  - Criticality
  - Security (data at rest, in motion, etc)
  - SLAs
- Establish a CI/CD process





Key take away is Jboss fuse provides the flexibility to architect a solution that meets the needs of your organization. Not everyone else.



#### Growth Plan

S

- Process for scaling to multiple organizations
  - shared Fuse platform
  - balance central automation with autonomous teams
  - scale up to 1000s of fabric profiles
- Migration between environments
- Works within existing recommended Architecture



Jboss Fuse is a first class citizen in a CI/CD tool chain.

- Take advantage of Fuse's native CI/CD capabilities
- CI/CD process includes automated tests/integration/acceptance, etc) to enact change to a production system.
- Increases visibility into the auditable and reproducible environment.
- Establish operational metrics feedback loop.



#### Conclusion



- Migration plan established
- Foundation for multitenancy
- Highly Scalable and Available
- Establish standard Dev/Ops Processes



