

SUMMIT

**JBoss
WORLD**

PRESENTED BY RED HAT

**LEARN. NETWORK.
EXPERIENCE OPEN SOURCE.**



SOA at Scale with SwitchYard

Keith Babo
SwitchYard Project Lead
Red Hat
06.28.12

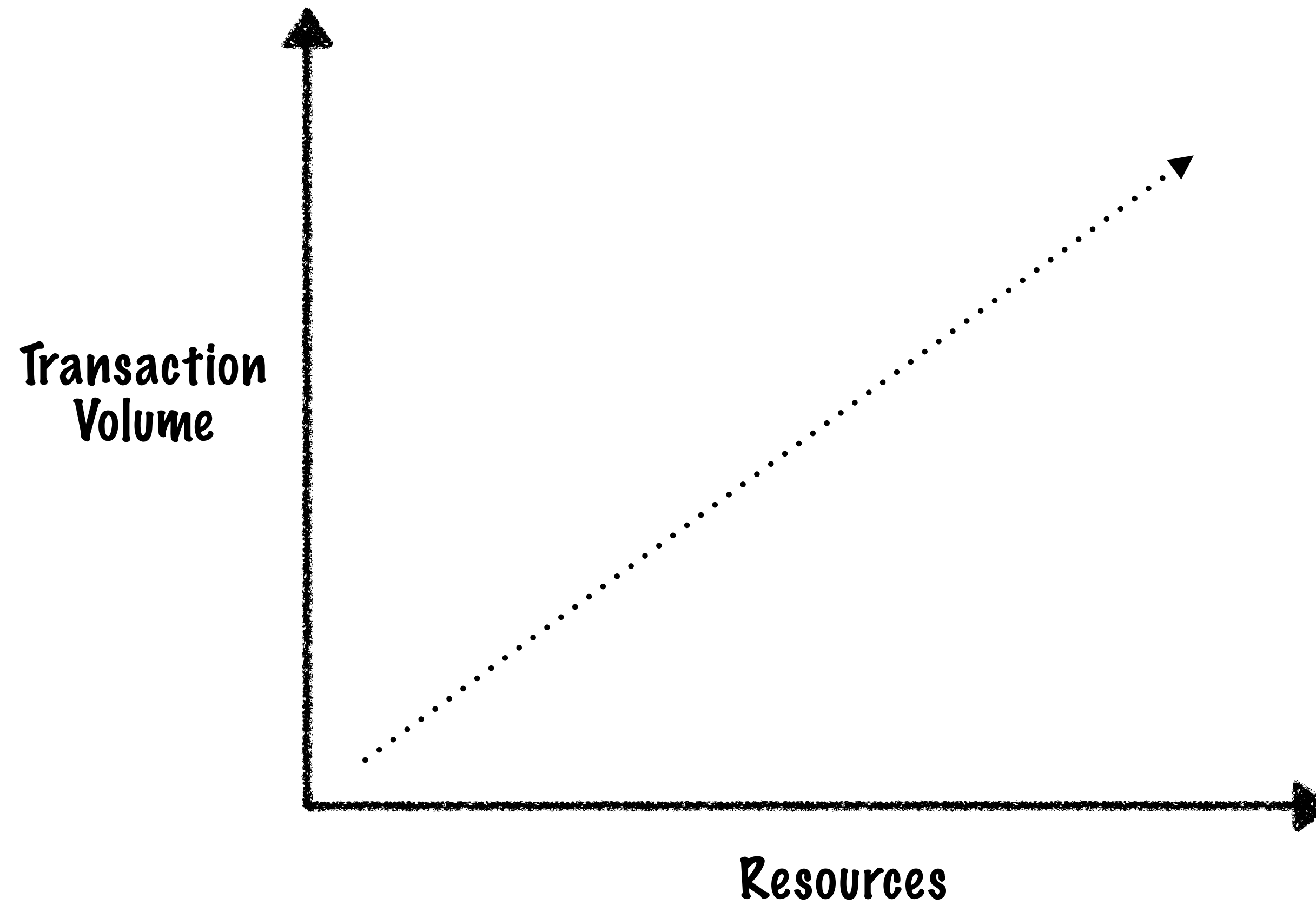
SUMMIT

**JBoss
WORLD**

PRESENTED BY RED HAT



Scalability



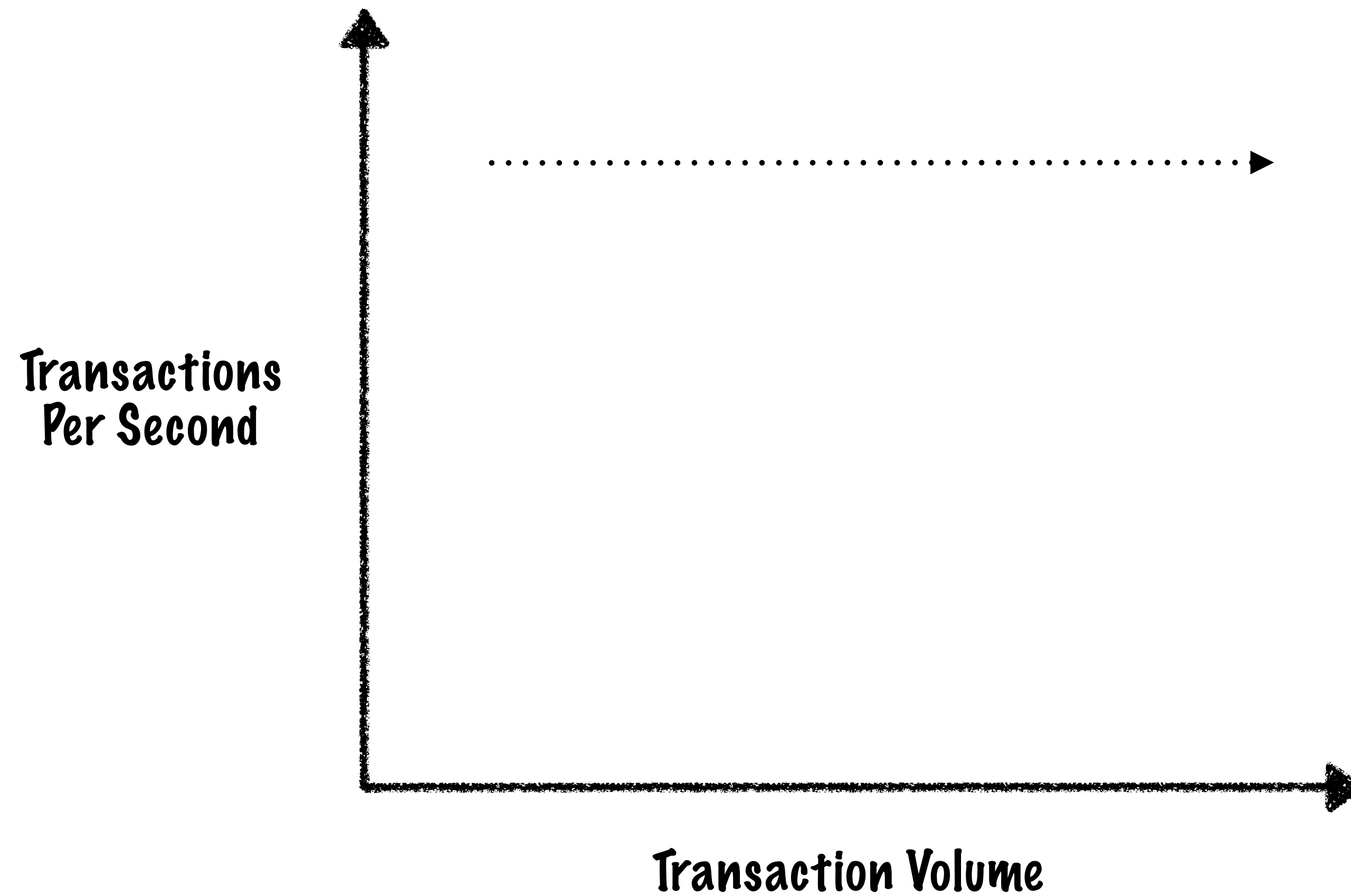
SUMMIT

**JBoss
WORLD**

PRESENTED BY RED HAT



Scalability



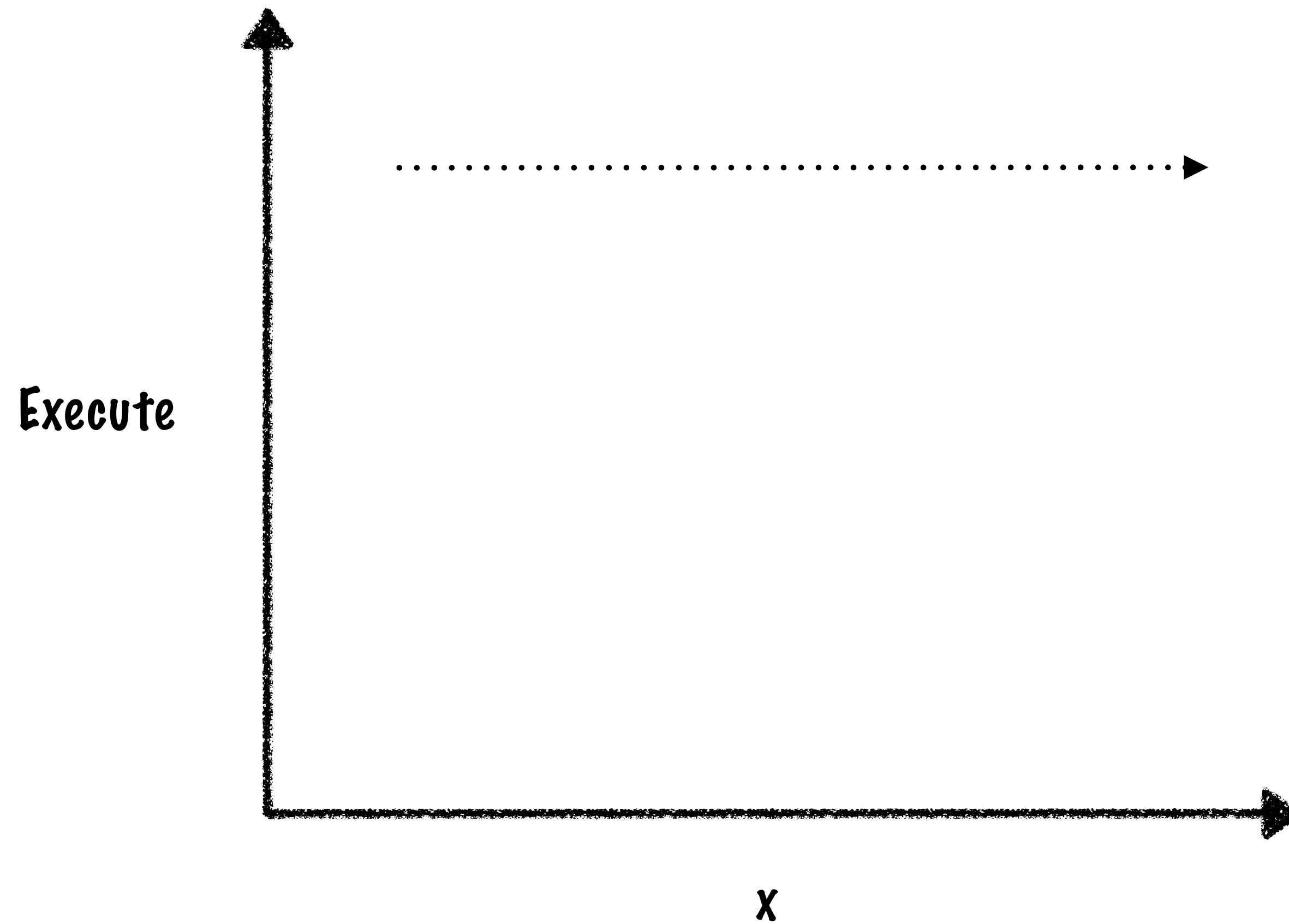
SUMMIT

**JBoss
WORLD**

PRESENTED BY RED HAT



Scalability



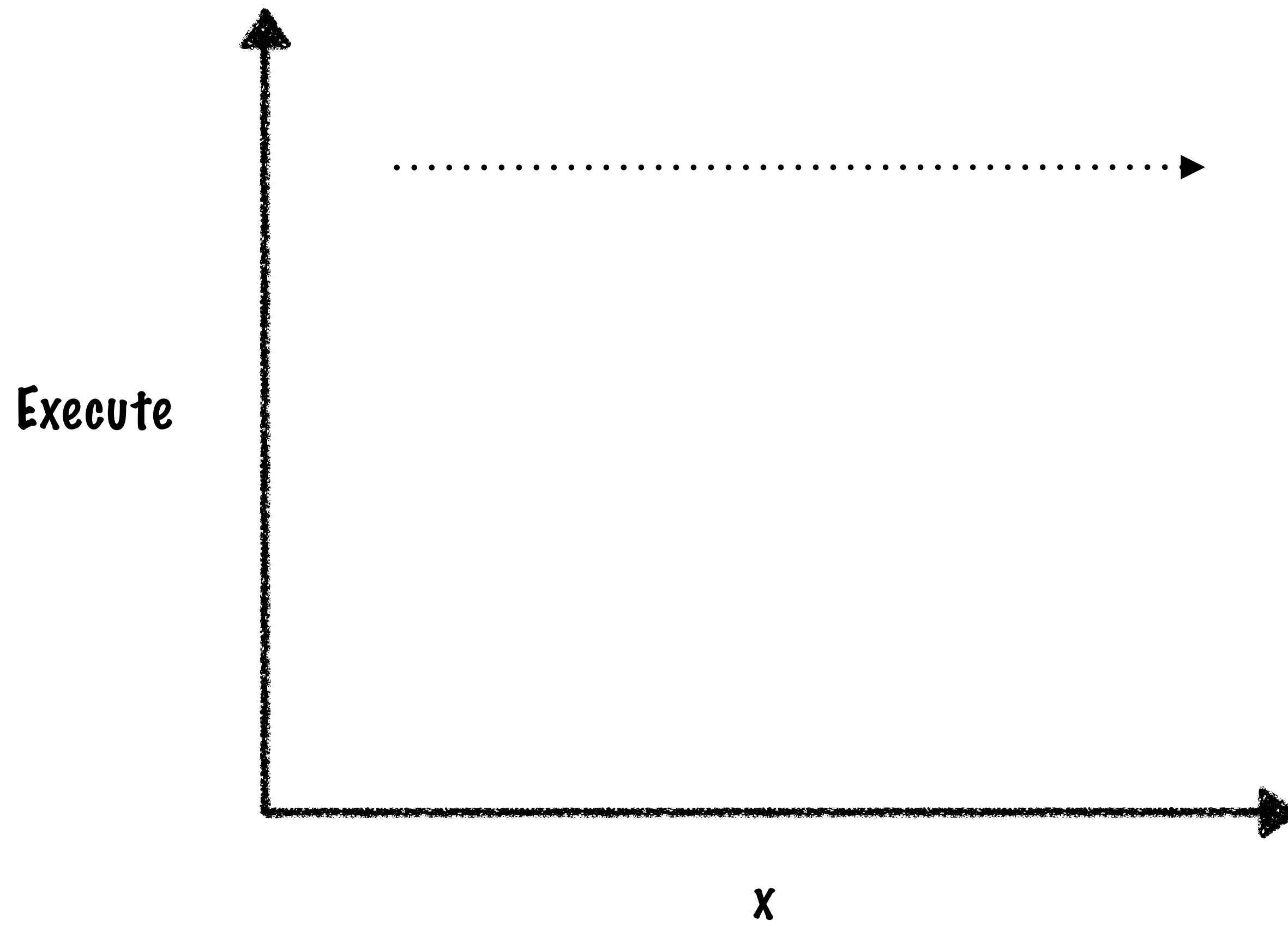
SUMMIT

**JBoss
WORLD**

PRESENTED BY RED HAT



Scalability



Values for 'x'

- ▶ Applications
- ▶ Technologies
- ▶ Dependencies
- ▶ Environments
- ▶ Developers
- ▶ Complexity

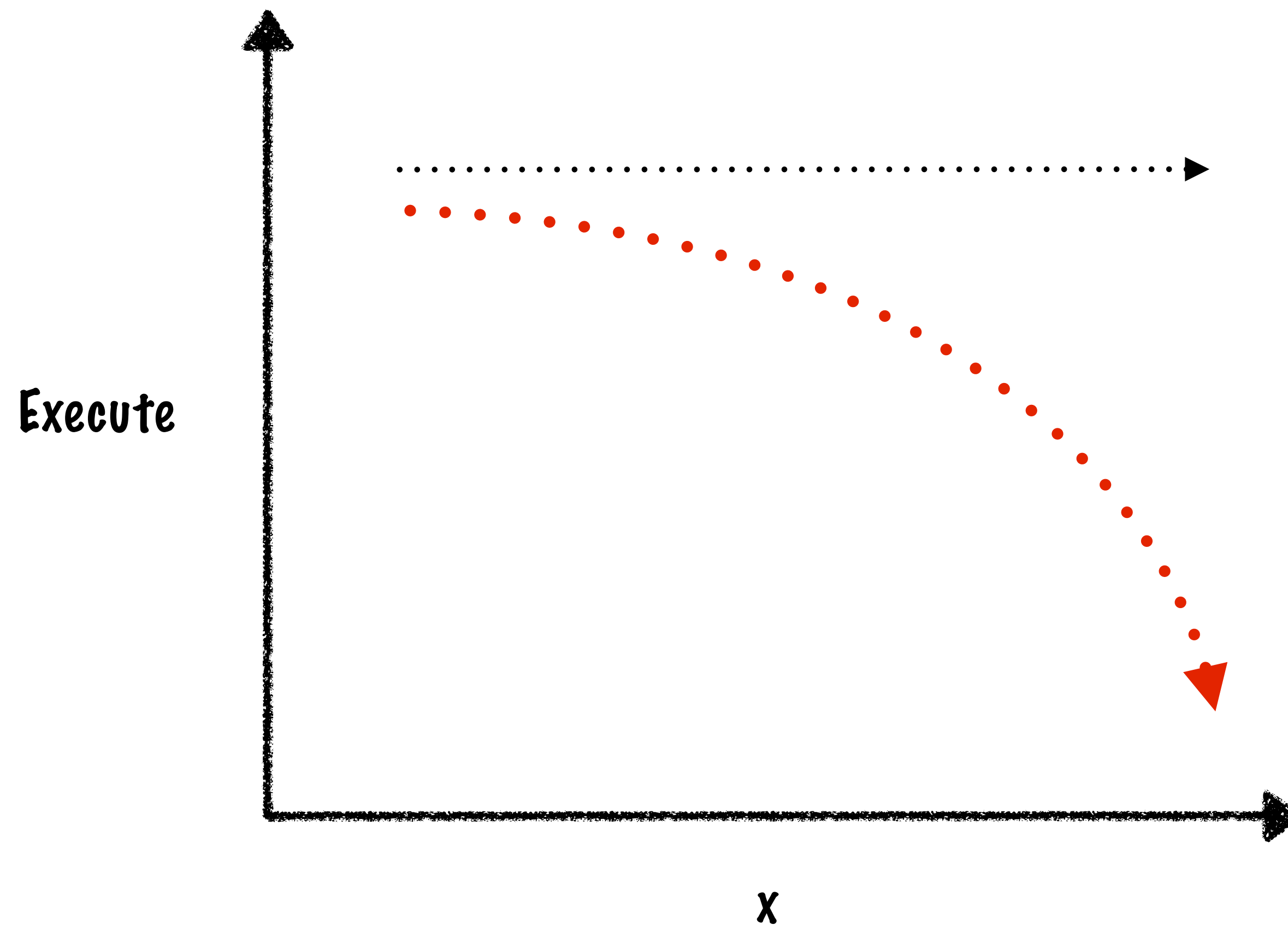
SUMMIT

**JBoss
WORLD**

PRESENTED BY RED HAT



Scalability



Values for 'x'

- ▶ Applications
- ▶ Technologies
- ▶ Dependencies
- ▶ Environments
- ▶ Developers
- ▶ Complexity

SUMMIT

**JBoss
WORLD**

PRESENTED BY RED HAT



Introducing SwitchYard

- Next generation Enterprise Service Bus
- What happened to JBoss ESB?
- Taking the next evolutionary step



SUMMIT

**JBoss
WORLD**

PRESENTED BY RED HAT



Focal Points

- Uniform Application Model
- Explicit Service Contracts
- Dependency Management
- Binding Abstraction
- Declarative Data Handling
- Declarative Policy
- Governance

SUMMIT

**JBoss
WORLD**

PRESENTED BY RED HAT



Uniform Application Model

- What's in your application?
 - Services provided
 - Services consumed
 - Service contracts
 - Policies

SUMMIT

**JBoss
WORLD**

PRESENTED BY RED HAT



Java

```
@WebService
public class MyFancyBean {
    @Resource(mappedName = "jms/ConnectionFactory")
    private static ConnectionFactory cf;

    @WebMethod
    public void send(String msg) {
        ...
    }
}
```

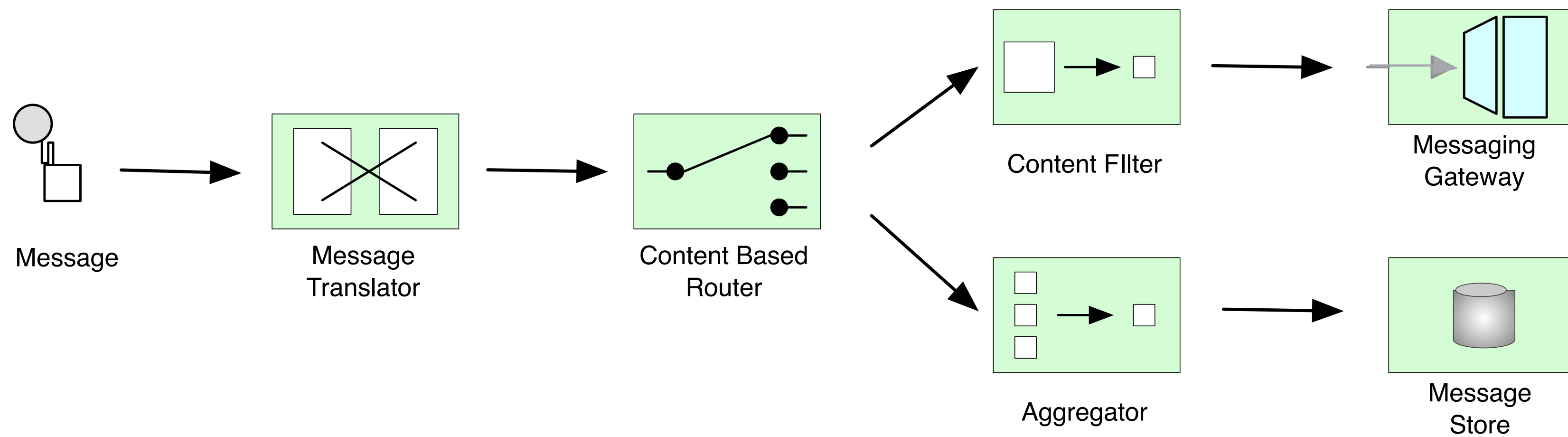
SUMMIT

**JBoss
WORLD**

PRESENTED BY RED HAT



Integration Flow



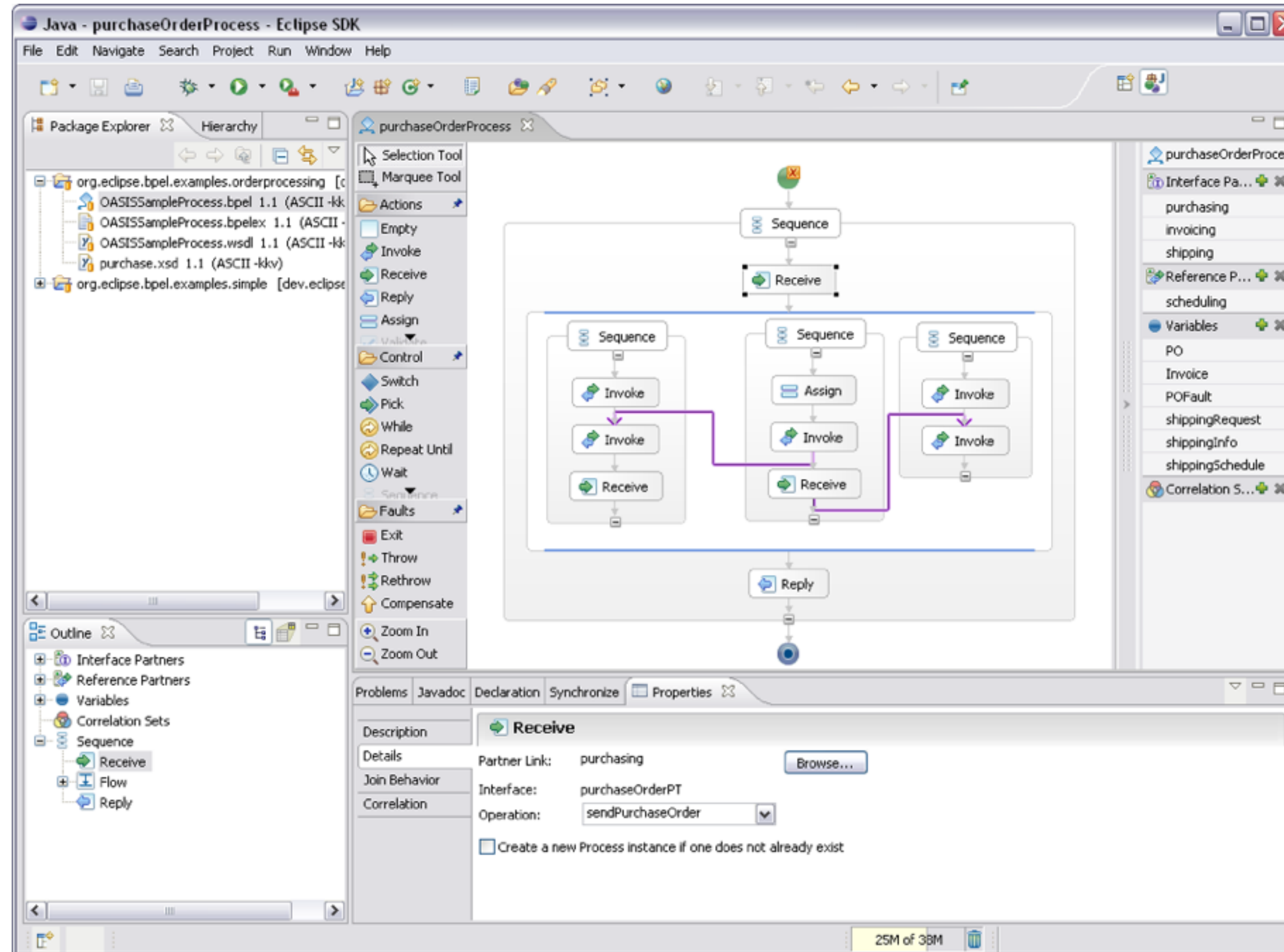
SUMMIT

**JBoss
WORLD**

PRESENTED BY RED HAT



Orchestration



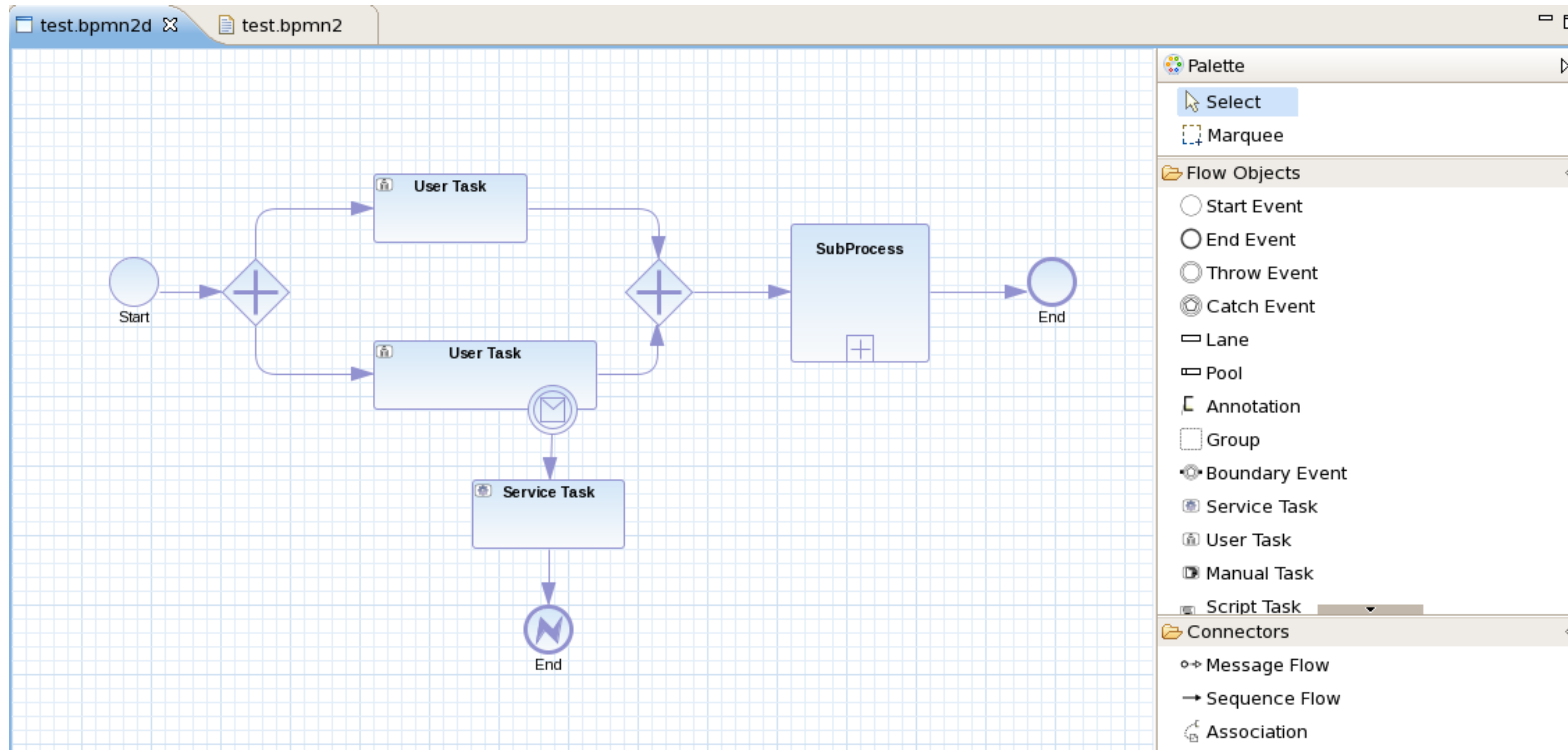
SUMMIT

**JBoss
WORLD**

PRESENTED BY RED HAT



Workflow



SUMMIT

**JBoss
WORLD**

PRESENTED BY RED HAT



Event Processing

```
declare StockTick
  @role( event )
  @timestamp (datetime)

  datetime : java.util.Date
  symbol   : String
  price    : double
end
```

```
declare MonkeyDart
  @role( event )
  @timestamp (datetime)

  datetime : java.util.Date
  dart     : org.dartboard.Position
end
```

```
rule "Buy Low Sell High"
when
  YearLow( $low : low )
  $st : StockTick( price <= $low )
  $md : MonkeyDart( this after[0,5m] $st )
then
  // buy all the stock you can get!
  OrderService.buy(st.getTicker(), Integer.MAX_VALUE)
end
```

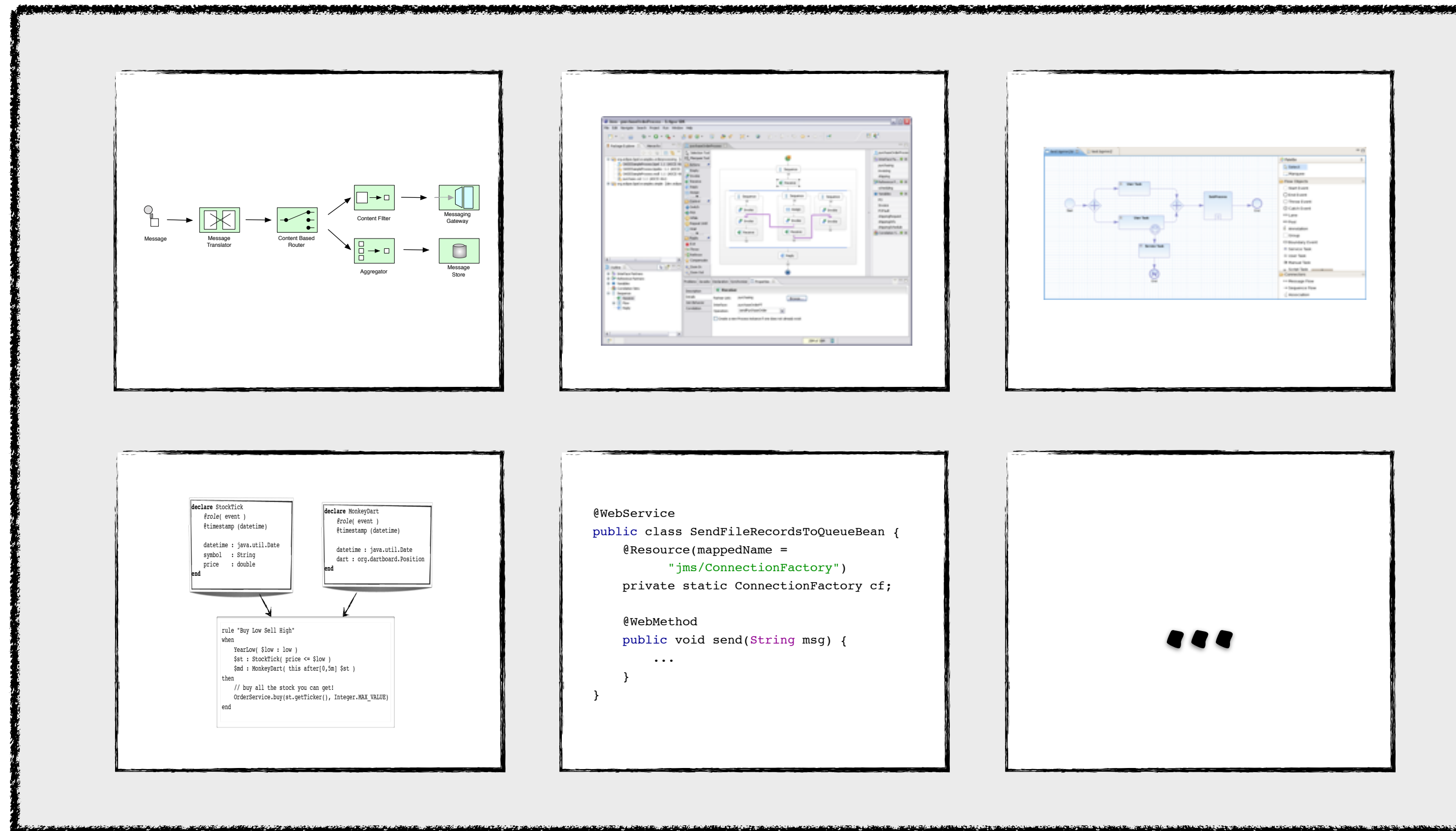
SUMMIT

JBoss
WORLD

PRESENTED BY RED HAT



Big Picture



SUMMIT

JBoss
WORLD

PRESENTED BY RED HAT



SCA

- Service Component Architecture
- OASIS standard for creating service-oriented applications
 - Describe
 - Assemble
 - Build

SUMMIT

**JBoss
WORLD**

PRESENTED BY RED HAT



Application Descriptor

```
<composite xmlns="http://docs.oasis-open.org/ns/opencsa/sca/200912"
  xmlns:bean="urn:switchyard-component-bean:config:1.0"
  xmlns:camel="urn:switchyard-component-camel:config:1.0">
  name="orders" targetNamespace="urn:example:orders">
  <service name="OrderService" promote="OrderService">
    <interface.wSDL interface="wSDL/OrderService.wSDL#wSDL.porttype(OrderService)"/>
    <binding.soap xmlns="urn:switchyard-component-soap:config:1.0">
      <wSDL>wSDL/OrderService.wSDL</wSDL>
      <socketAddr>:18001</socketAddr>
      <contextPath>demo-orders</contextPath>
    </binding.soap>
  </service>
  <component name="InventoryService">
    <bean:implementation.bean class="org.example.orders.InventoryServiceBean"/>
    <service name="InventoryService">
      <interface.java interface="org.example.orders.InventoryService"/>
    </service>
  </component>
  <component name="Order">
    <bean:implementation.bean class="org.example.orders.Order"/>
    <reference name="OrderService">
      <interface.java interface="org.example.orders.OrderService"/>
    </reference>
  </component>
  <component name="OrderService">
    <camel:implementation.camel>
      <camel:java class="org.example.orders.OrdersRoute"/>
    </camel:implementation.camel>
    <service name="OrderService">
      <interface.java interface="org.example.orders.OrderService"/>
    </service>
    <reference name="InventoryService">
      <interface.java interface="org.example.orders.InventoryService"/>
    </reference>
  </component>
</composite>
```

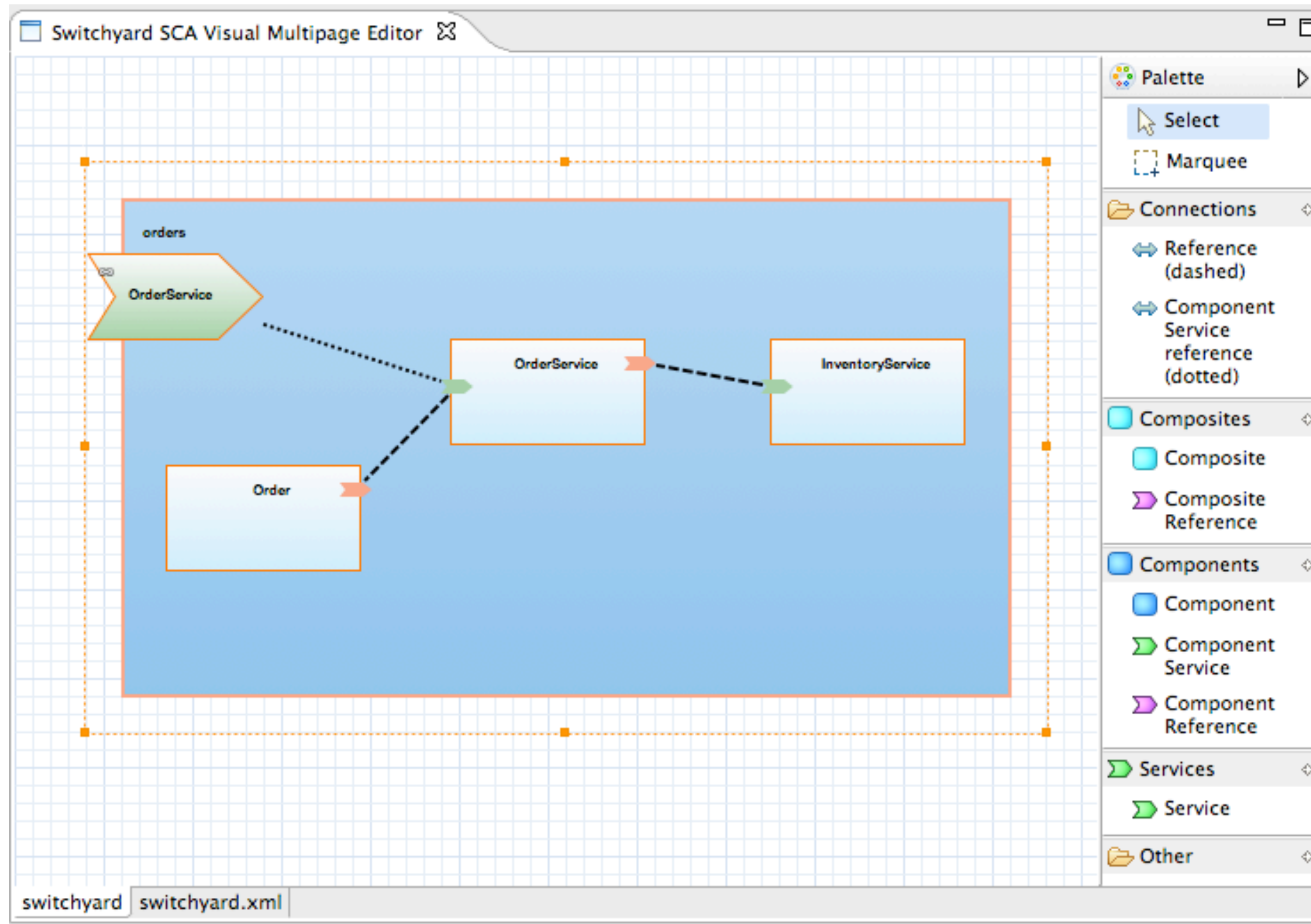
SUMMIT

**JBoss
WORLD**

PRESENTED BY RED HAT



Application Model



SUMMIT

**JBoss
WORLD**

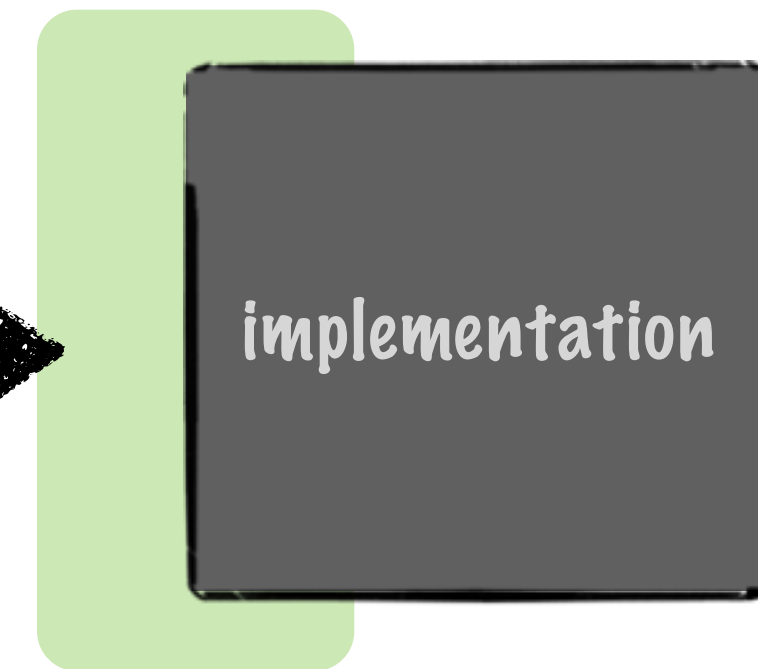
PRESENTED BY RED HAT



Service Contracts

- Agreement between consumer and provider
 - Messages
 - Policy
 - SLA
- Contract-based development
 - Service Interface
 - Implementation Hiding

contract



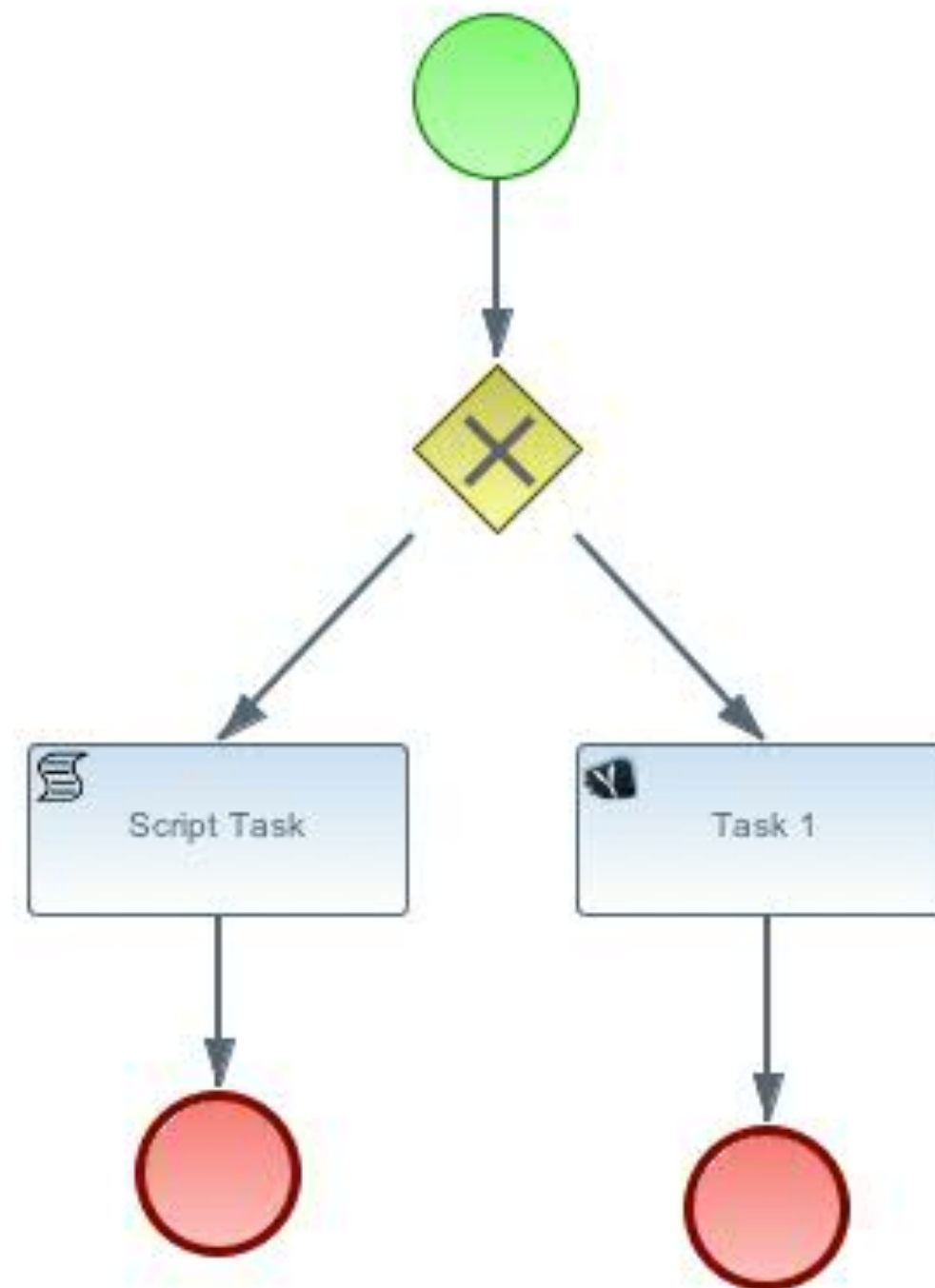
SUMMIT

**JBoss
WORLD**

PRESENTED BY RED HAT



What's Expected Here?



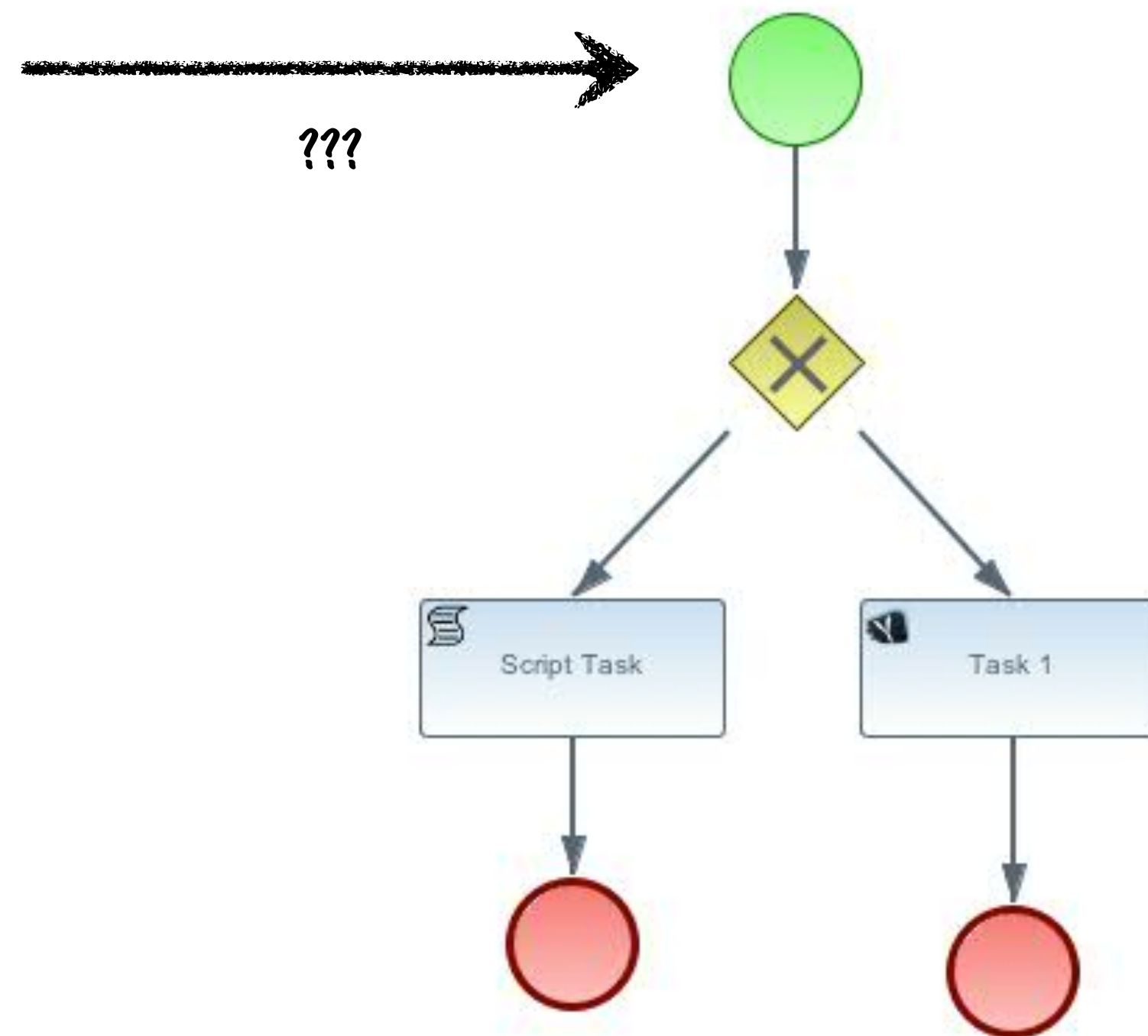
SUMMIT

**JBoss
WORLD**

PRESENTED BY RED HAT



What's Expected Here?



SUMMIT

**JBoss
WORLD**

PRESENTED BY RED HAT



Service Contract

```
<sca:component name="ProcessPayments">
  <bpm:implementation.bpm
    processDefinition="PaymentsProcess.bpmn"
    processId="Payments"/>
</bpm:implementation.bpm>
<sca:service name="Payments">
  <sca:interface.java interface="org.example.Payments"/>
</sca:service>
<sca:reference name="AccountsPaid">
  <sca:interface.java
    interface="com.example.switchyard.demo2.AccountsPaid"/>
</sca:reference>
</sca:component>
```



SUMMIT

**JBoss
WORLD**

PRESENTED BY RED HAT



Dependency Management

- Relationship between services
- Inside and outside the application boundary
- SOA = dependencies
 - Service consumer
 - Composition (service pipeline, orchestration, etc.)

SUMMIT

**JBoss
WORLD**

PRESENTED BY RED HAT



Dependency Analysis is Easy

```
from("switchyard://OrderService")
  .split().tokenizeXML("order")
  .choice()
    .when(xpath("/order/region = 'NA'"))
      .to("direct:DirectSales")
    .when(xpath("/order/region = 'LAM'"))
      .to("bean:LAMOrderProcessor")
    .when(xpath("/order/region = 'EMEA'"))
      .to("vm:eu.orders.inbound")
    .when(xpath("/order/region = 'APAC'"))
      .to("jms:queue:ordersQueue");
```

SUMMIT

**JBoss
WORLD**

PRESENTED BY RED HAT



Dependency Analysis is Easy

Internal

```
from("switchyard://OrderService")
  .split().tokenizeXML("order")
  .choice()
    .when(xpath("/order/region = 'NA'"))
      .to("direct:DirectSales")
    .when(xpath("/order/region = 'LAM'"))
      .to("bean:LAMOrderProcessor")
    .when(xpath("/order/region = 'EMEA'"))
      .to("vm:eu.orders.inbound")
    .when(xpath("/order/region = 'APAC'"))
      .to("jms:queue:ordersQueue");
```

SUMMIT

**JBoss
WORLD**

PRESENTED BY RED HAT



Dependency Analysis is Easy

Internal

```
from("switchyard://OrderService")
  .split().tokenizeXML("order")
  .choice()
    .when(xpath("/order/region = 'NA'"))
      .to("direct:DirectSales")
    .when(xpath("/order/region = 'LAM'"))
      .to("bean:LAMOrderProcessor")
    .when(xpath("/order/region = 'EMEA'"))
      .to("vm:eu.orders.inbound")
    .when(xpath("/order/region = 'APAC'"))
      .to("jms:queue:ordersQueue");
```

External

SUMMIT

**JBoss
WORLD**

PRESENTED BY RED HAT



Well, Sort Of ...

```
from("switchyard://OrderService")
  .split().tokenizeXML("order")
  .choice()
    .when(xpath("/order/region = 'NA'"))
      .to("direct:DirectSales")
    .when(xpath("/order/region = 'LAM'"))
      .to("bean:LAMOrderProcessor")
    .when(xpath("/order/region = 'EMEA'"))
      .to("vm:eu.orders.inbound")
    .when(xpath("/order/region = 'APAC'"))
      .to("jms:queue:ordersQueue");
```

SUMMIT

JBoss
WORLD

PRESENTED BY RED HAT



Well, Sort Of ...

```
from("direct://DirectSales")  
.to("cxf:bean:OrderWebService")
```

```
public class LAMOrderProcessor {  
    @Produce(uri = "activemq:lam.orders")  
    ProducerTemplate producer;  
  
    public void send(String body) {  
        producer.sendBody(body);  
    }  
}
```

```
from("switchyard://OrderService")  
.split().tokenizeXML("order")  
.choice()  
    .when(xpath("/order/region = 'NA'"))  
        .to("direct:DirectSales")  
    .when(xpath("/order/region = 'LAM'"))  
        .to("bean:LAMOrderProcessor")  
    .when(xpath("/order/region = 'EMEA'"))  
        .to("vm:eu.orders.inbound")  
    .when(xpath("/order/region = 'APAC'"))  
        .to("jms:queue:ordersQueue");
```

SUMMIT

**JBoss
WORLD**

PRESENTED BY RED HAT



Well, Sort Of ...

```
from("direct://DirectSales")  
.to("cxf:bean:OrderWebService")
```

```
public class LAMOrderProcessor {  
    @Produce(uri = "activemq:lam.orders")  
    ProducerTemplate producer;  
  
    public void send(String body) {  
        producer.sendBody(body);  
    }  
}
```

```
from("switchyard://OrderService")  
.split().tokenizeXML("order")  
.choice()  
    .when(xpath("/order/region = 'NA'"))  
        .to("direct:DirectSales")  
    .when(xpath("/order/region = 'LAM'"))  
        .to("bean:LAMOrderProcessor")  
    .when(xpath("/order/region = 'EMEA'"))  
        .to("vm:eu.orders.inbound")  
    .when(xpath("/order/region = 'APAC'"))  
        .to("jms:queue:ordersQueue");
```

External



SUMMIT

JBoss
WORLD

PRESENTED BY RED HAT



Maybe Not

```
from("switchyard://OrderService")
  .split().tokenizeXML("order")
  .choice()
    .when(xpath("/order/region = 'NA'"))
      .to("direct:DirectSales")
    .when(xpath("/order/region = 'LAM'"))
      .to("bean:LAMOrderProcessor")
    .when(xpath("/order/region = 'EMEA'"))
      .to("vm:eu.orders.inbound")
    .when(xpath("/order/region = 'APAC'"))
      .to("jms:queue:ordersQueue");
```

SUMMIT

**JBoss
WORLD**

PRESENTED BY RED HAT



Maybe Not

```
from("switchyard://OrderService")
  .split().tokenizeXML("order")
  .choice()
    .when(xpath("/order/region = 'NA'"))
      .to("direct:DirectSales")
    .when(xpath("/order/region = 'LAM'"))
      .to("bean:LAMOrderProcessor")
    .when(xpath("/order/region = 'EMEA'"))
      .to("vm:eu.orders.inbound")
    .when(xpath("/order/region = 'APAC'"))
      .to("jms:queue:ordersQueue");
```

```
from("jms://queue:ordersQueue")
  .to("bean:APACOrderProcessor")
```

```
from("vm://eu.orders.inbound")
  .to("direct:ThisIsRidiculous")
```

SUMMIT

**JBoss
WORLD**

PRESENTED BY RED HAT



Maybe Not

Internal

```
from("switchyard://OrderService")
  .split().tokenizeXML("order")
  .choice()
    .when(xpath("/order/region = 'NA'"))
      .to("direct:DirectSales")
    .when(xpath("/order/region = 'LAM'"))
      .to("bean:LAMOrderProcessor")
    .when(xpath("/order/region = 'EMEA'"))
      .to("vm:eu.orders.inbound")
    .when(xpath("/order/region = 'APAC'"))
      .to("jms:queue:ordersQueue");
```

```
from("jms://queue:ordersQueue")
  .to("bean:APACOrderProcessor")
```

```
from("vm://eu.orders.inbound")
  .to("direct:ThisIsRidiculous")
```

SUMMIT

JBoss
WORLD

PRESENTED BY RED HAT



Explicit Dependencies

```
from("switchyard://OrderService")
  .split().tokenizeXML("order")
  .choice()
    .when(xpath("/order/region = 'NA'"))
      .to("switchyard:NAOrders")
    .when(xpath("/order/region = 'LAM'"))
      .to("switchyard:LAMOrders")
    .when(xpath("/order/region = 'EMEA'"))
      .to("switchyard:EMEAOrders")
    .when(xpath("/order/region = 'APAC'"))
      .to("switchyard:APACOrders");
```

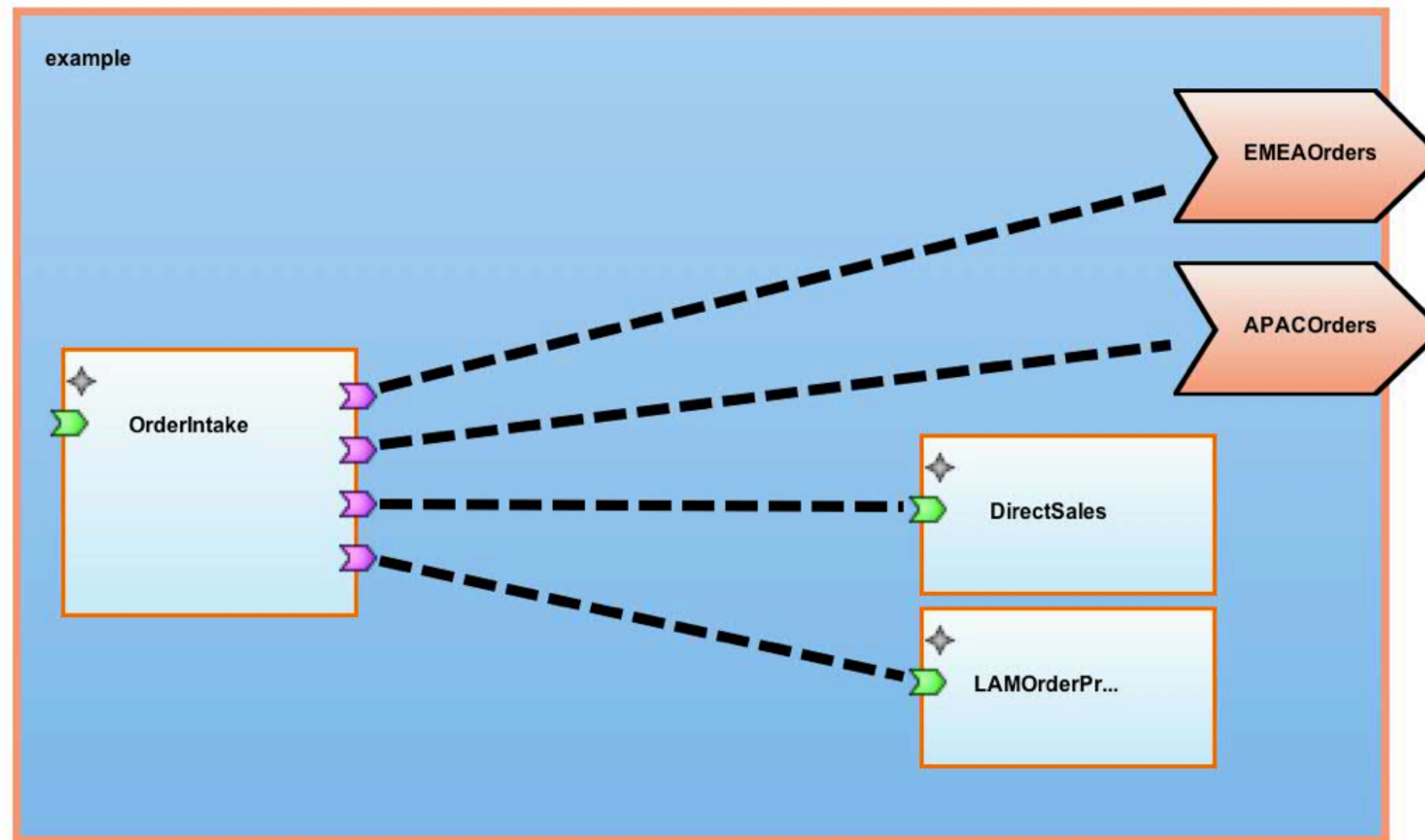
SUMMIT

**JBoss
WORLD**

PRESENTED BY RED HAT



Explicit Dependencies



SUMMIT

**JBoss
WORLD**

PRESENTED BY RED HAT



Demo Time!



SUMMIT

**JBoss
WORLD**

PRESENTED BY RED HAT



Binding Abstraction

- Separating abstract and concrete
 - Service interface
 - Service endpoint
- Flexibility
- Reuse

SUMMIT

**JBoss
WORLD**

PRESENTED BY RED HAT



Java

```
@WebService
public class MyFancyBean {
    @Resource(mappedName = "jms/ConnectionFactory")
    private static ConnectionFactory cf;

    @WebMethod
    public void send(String msg) {
        ...
    }
}
```

SUMMIT

JBoss
WORLD

PRESENTED BY RED HAT



Java

```
@Service(FancyService.class)
public class MyFancyBean implements FancyService {

    @Inject @Reference
    private HelloService hello;

    public void send(String msg) {
        ...
    }
}
```

SUMMIT


**JBoss
WORLD**

PRESENTED BY RED HAT




Configuration

```
<sca:service name="FancyServicePortType" promote="MyFancyBean/FancyService">  
  <sca:interface.wSDL interface="FancyService.wSDL#wsdl.porttype(FancyServicePortType)"/>  
  <soap:binding soap>  
    <soap:wSDL>FancyService.wSDL</soap:wSDL>  
  </soap:binding soap>  
</sca:service>
```



```
<sca:component name="MyFancyBean">  
  <bean:implementation bean class="com.example.switchyard.soa1.MyFancyBean"/>  
  <sca:service name="FancyService">  
    <sca:interface.java interface="com.example.switchyard.soa1.FancyService"/>  
  </sca:service>  
  <sca:reference name="HelloWorld">  
    <sca:interface.java interface="com.example.switchyard.soa1>HelloWorld"/>  
  </sca:reference>  
</sca:component>
```



```
<sca:reference name="HelloWorld" multiplicity="0..1" promote="MyComponent/HelloWorld">  
  <camel:binding.jms>  
    <camel:queue>queue/helloworld</camel:queue>  
    <camel:connectionFactory>#ConnectionFactory</camel:connectionFactory>  
  </camel:binding.jms>  
</sca:reference>
```

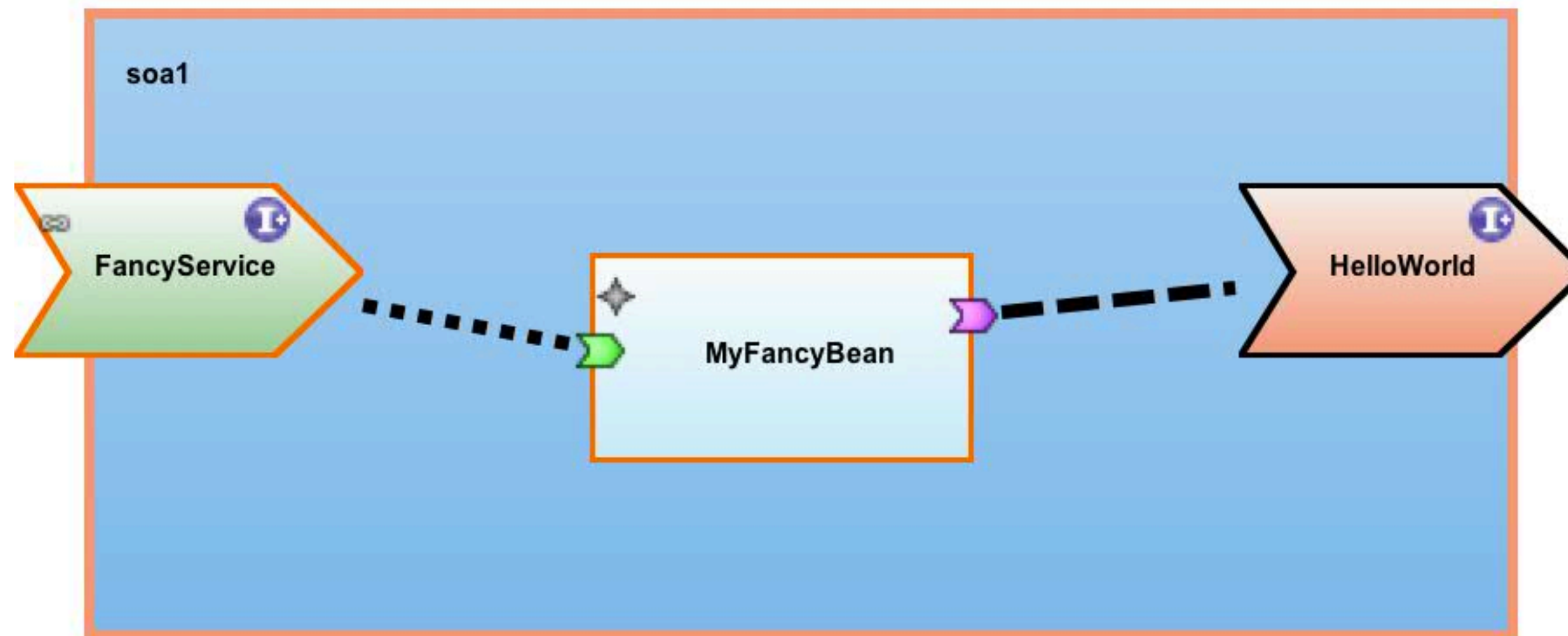
SUMMIT

**JBoss
WORLD**

PRESENTED BY RED HAT



Visual



SUMMIT

**JBoss
WORLD**

PRESENTED BY RED HAT



Declarative Data Handling

- Data handling as a cross cutting concern
- Transformation
- Validation
- Insidious coupling

SUMMIT

**JBoss
WORLD**

PRESENTED BY RED HAT



Transformation

- Ubiquitous challenge in application integration and SOA
- Change in data representation
- Change in data format
- Change in data itself

SUMMIT

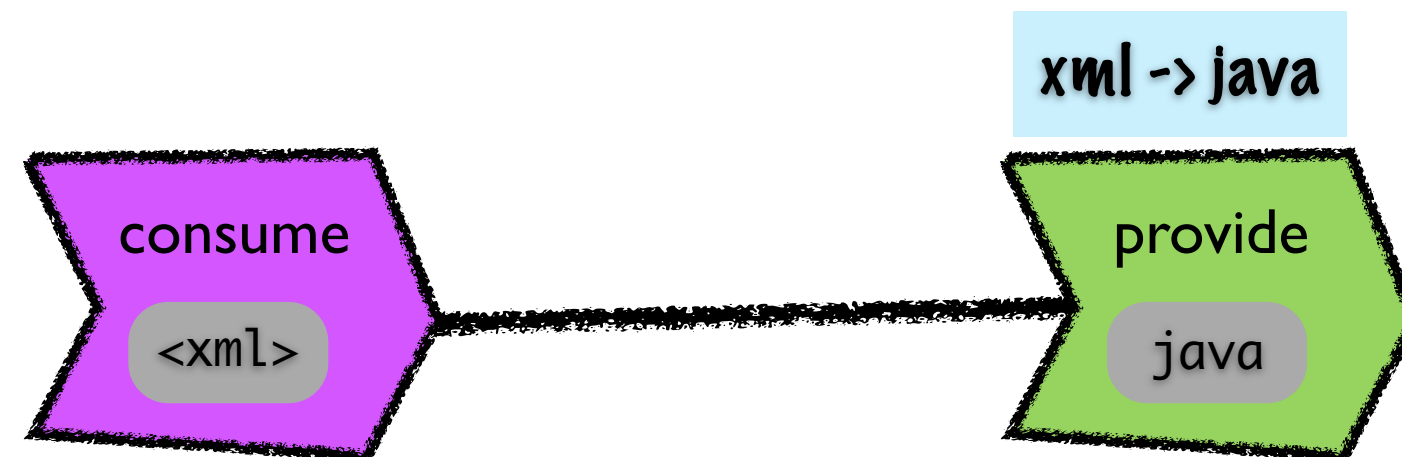
**JBoss
WORLD**

PRESENTED BY RED HAT

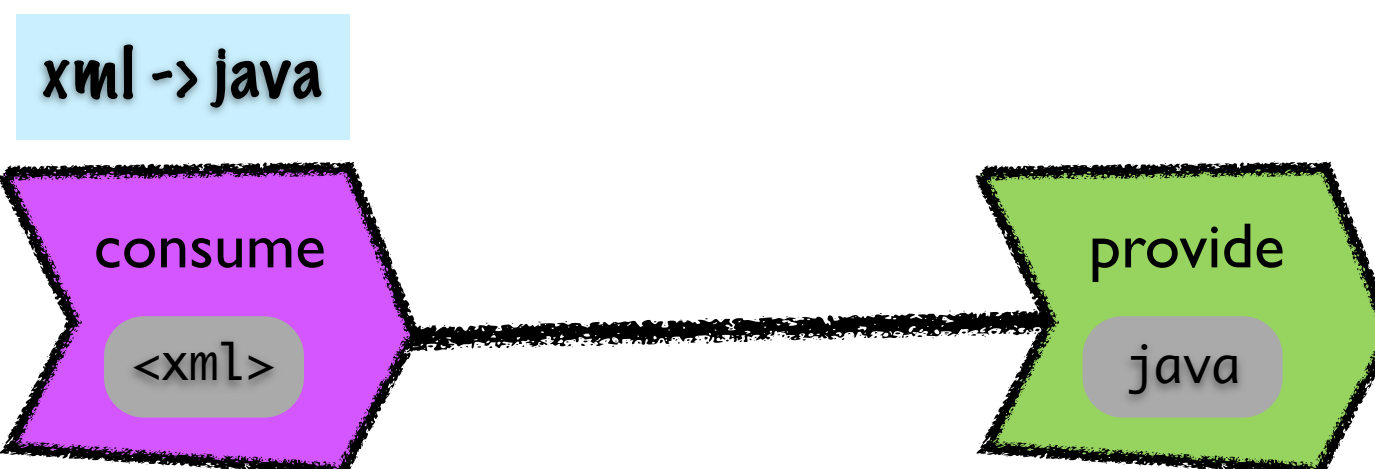


Where To Transform

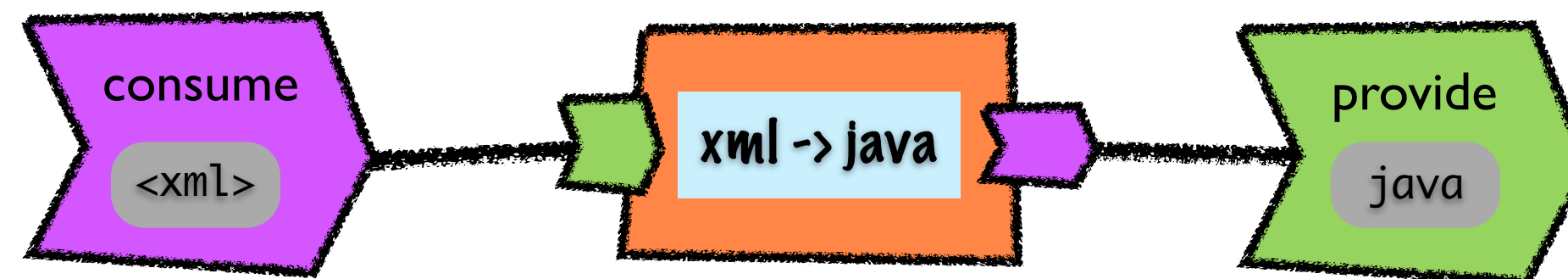
- In the provider?



- In the consumer?



- Add a routing service?



SUMMIT

**JBoss
WORLD**

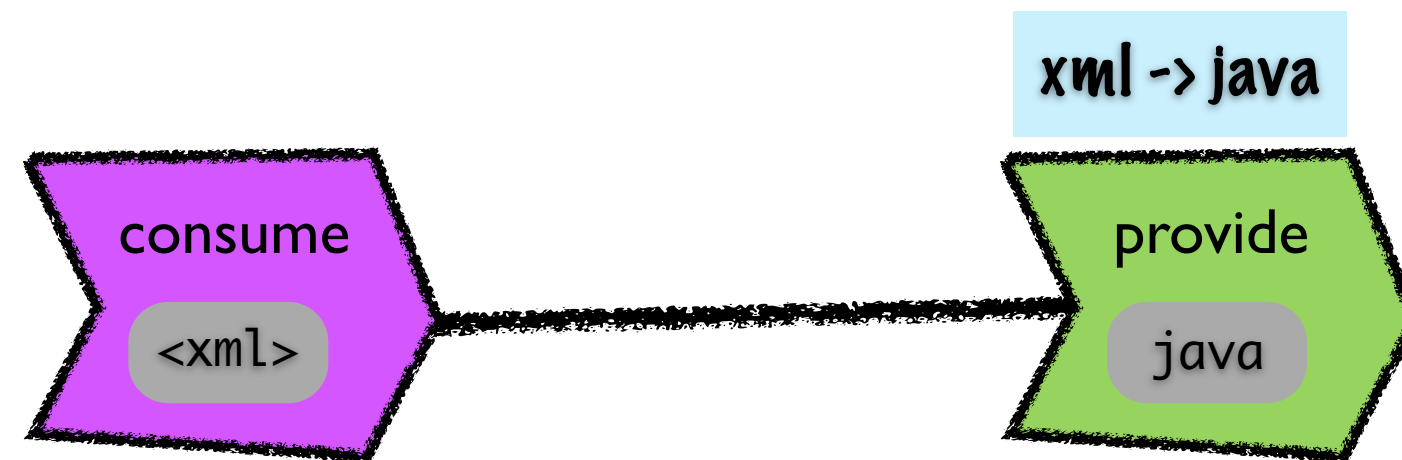
PRESENTED BY RED HAT



Where To Transform

- In the provider?

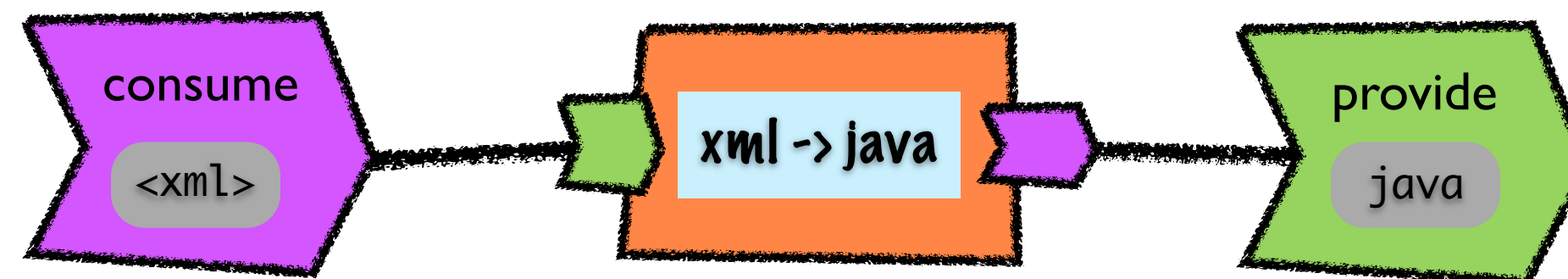
NO!



- In the consumer?



- Add a routing service?



SUMMIT

**JBoss
WORLD**

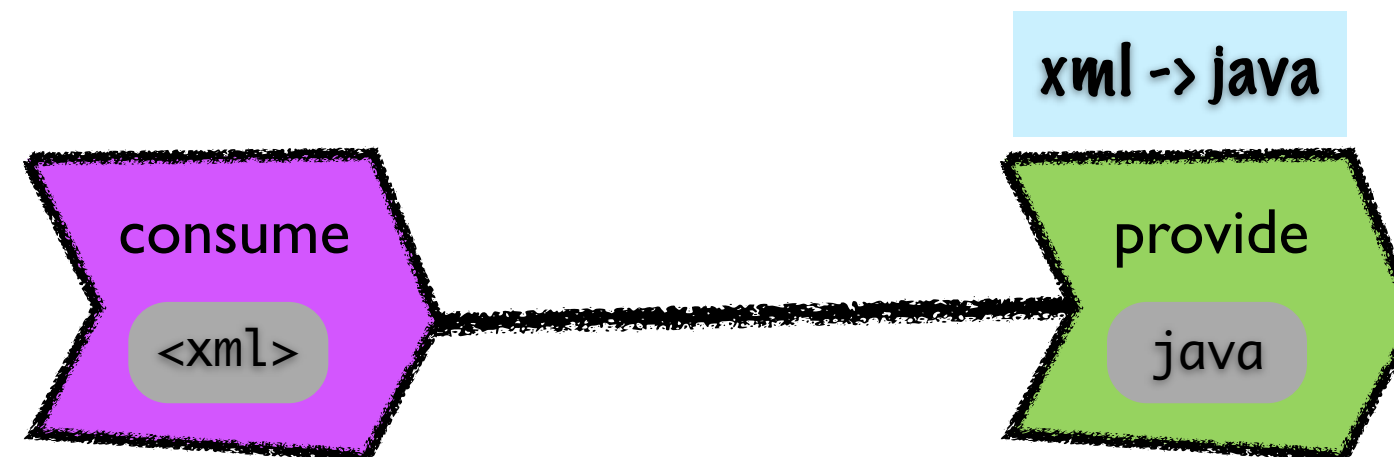
PRESENTED BY RED HAT



Where To Transform

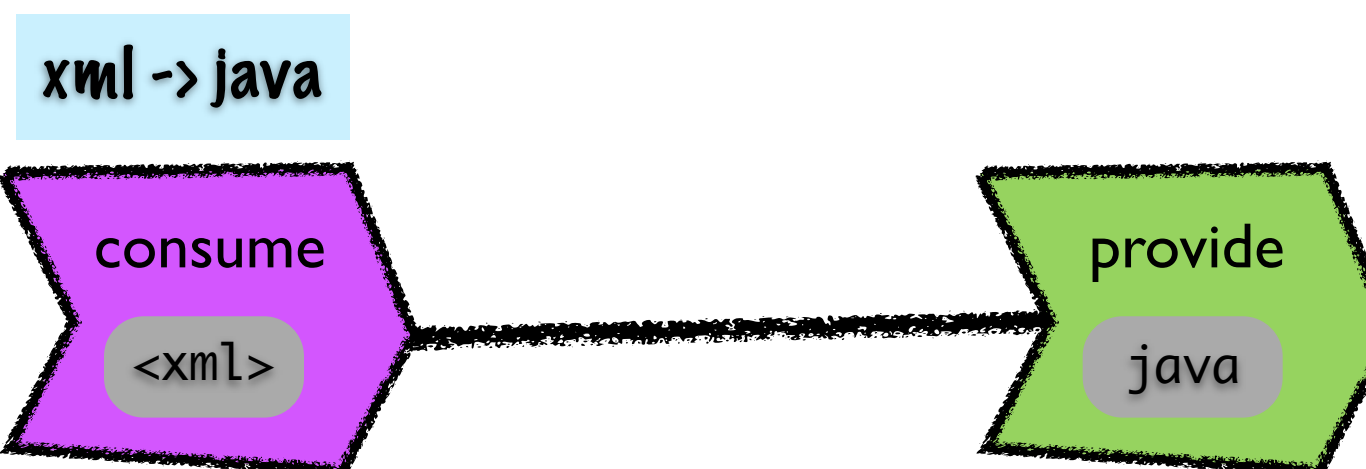
- In the provider?

NO!

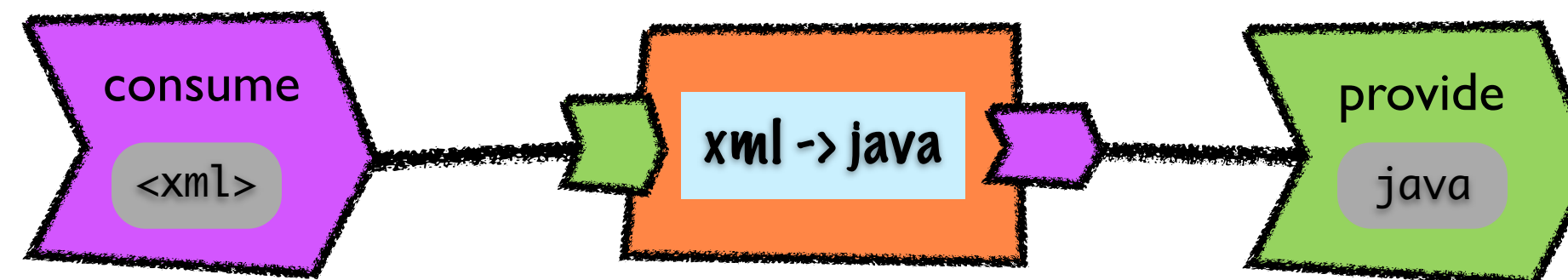


- In the consumer?

NO!



- Add a routing service?



SUMMIT

**JBoss
WORLD**

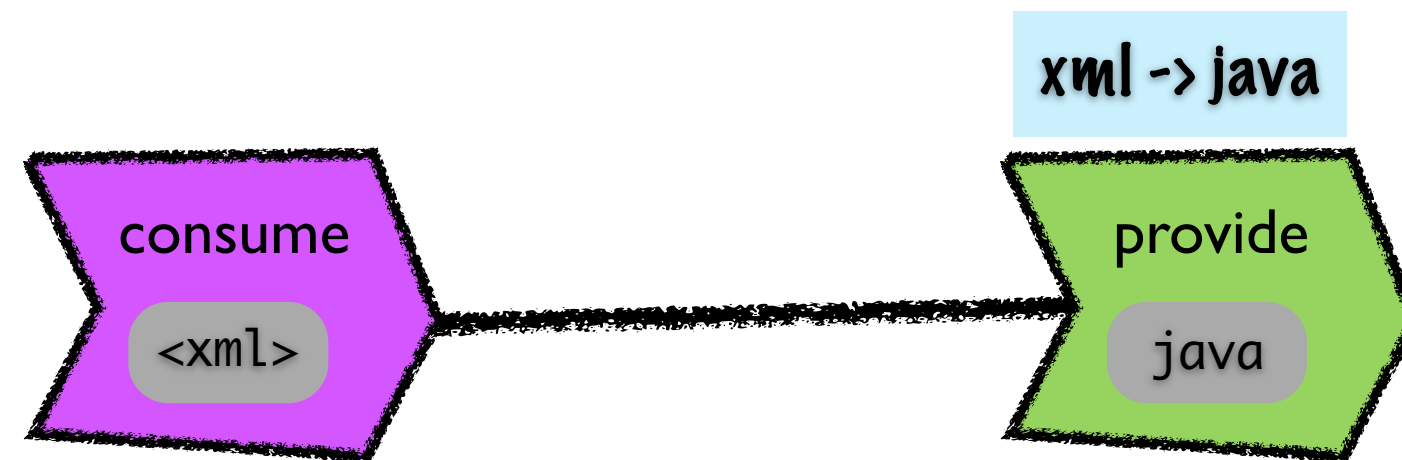
PRESENTED BY RED HAT



Where To Transform

- In the provider?

NO!



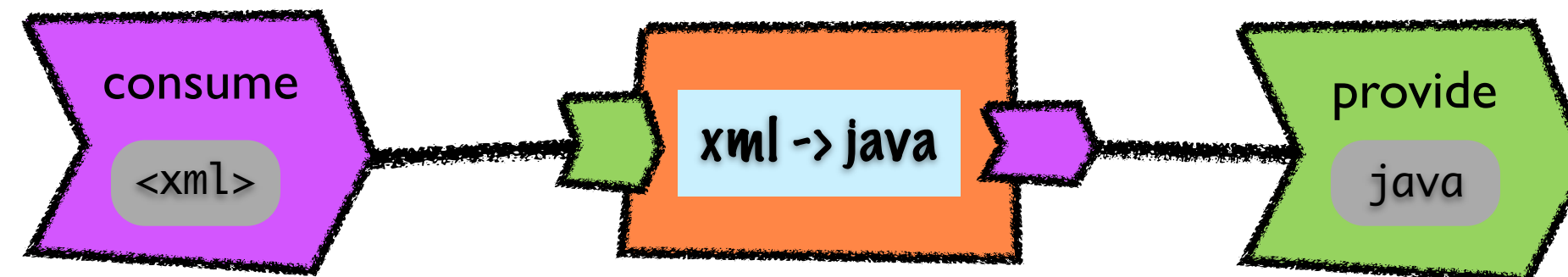
- In the consumer?

NO!



- Add a routing service?

MAYBE



SUMMIT

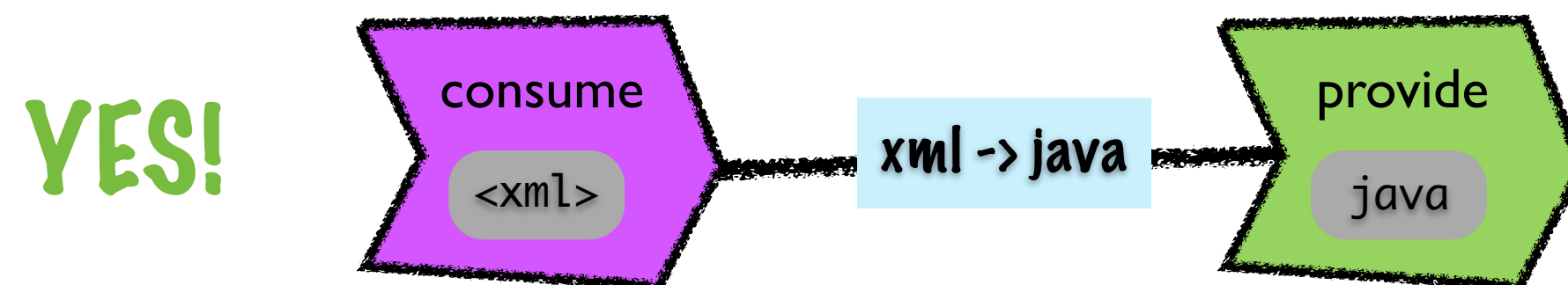
**JBoss
WORLD**

PRESENTED BY RED HAT



Transformers

- Transformation is wired into SwitchYard core
 - Types declared via service contract
 - Transformer resolved dynamically at runtime
- Declarative, not procedural



- Java, JAXB, XSLT, JSON, and Smooks

SUMMIT

JBoss
WORLD

PRESENTED BY RED HAT



Java Transformer

```
@Transformer(from = "{urn:switchyard-example:orders:1.0}submitOrder")
public Order transform(Element from) {
    return new Order()
        .setOrderId(getElementValue(from, "orderId"))
        .setItemId(getElementValue(from, "itemId"))
        .setQuantity(Integer.valueOf(getElementValue(from, "quantity")));
}
```

SUMMIT

JBoss
WORLD

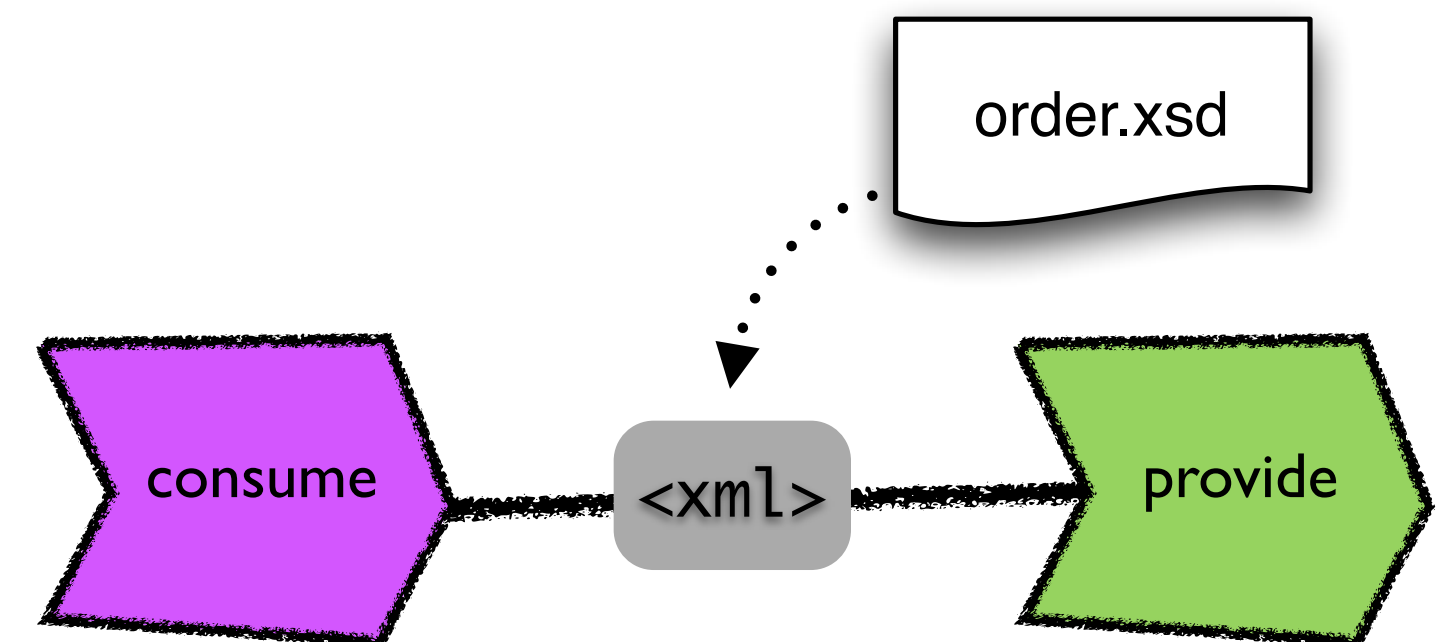
PRESENTED BY RED HAT



Validators

- Declarative validation
- Supports XML Schema and Java validation
- Executes pre and post transformation

```
<validate.xml  
  schemaType="XMLSCHEMA"  
  name="{urn:example:purchasing}order"  
  schemaFile="xsd/order.xsd" />
```



SUMMIT

**JBoss
WORLD**

PRESENTED BY RED HAT



Demo Time!



SUMMIT

**JBoss
WORLD**

PRESENTED BY RED HAT



Declarative Policy

- Declare the 'what', defer the 'how'
- Requirements attached to service definition
- Runtime injects policy enforcement point

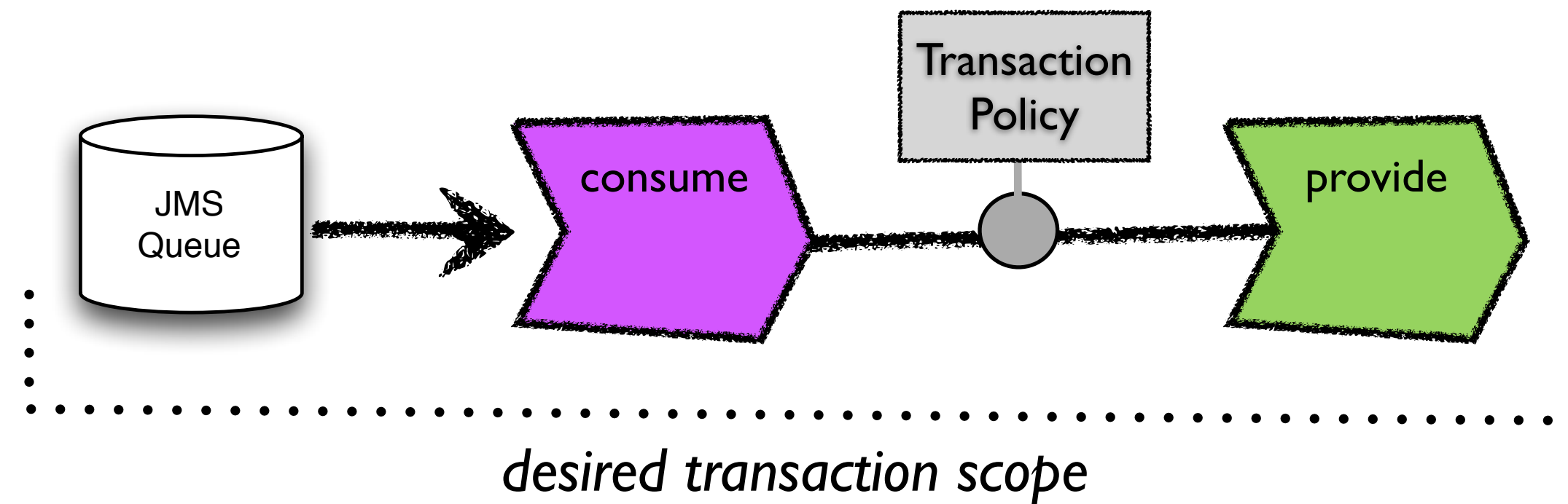
SUMMIT

**JBoss
WORLD**

PRESENTED BY RED HAT



Transaction Policy



```
<service name="OrderService requires="propagatesTransaction">  
  ...  
</service>
```

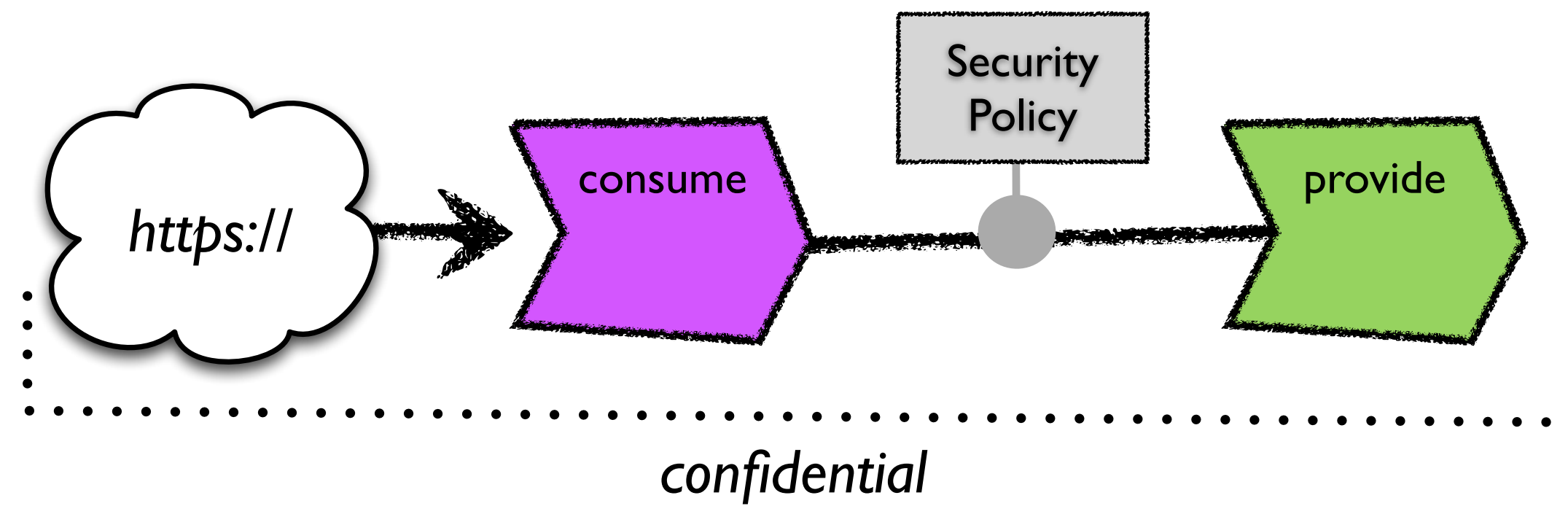
SUMMIT

**JBoss
WORLD**

PRESENTED BY RED HAT



Security Policy



```
<service name="OrderService requires="confidentiality">  
  ...  
</service>
```

SUMMIT

**JBoss
WORLD**

PRESENTED BY RED HAT



Governance

- Design-time and runtime
- Basics
 - Service Repository Integration
 - Service Monitoring
- Advanced
 - Overlord
 - Savara

SUMMIT

**JBoss
WORLD**

PRESENTED BY RED HAT



Service Repository

- There are artifacts related to your application services that need to be:
 - Published
 - Discovered
 - Related
 - Managed
 - Versioned

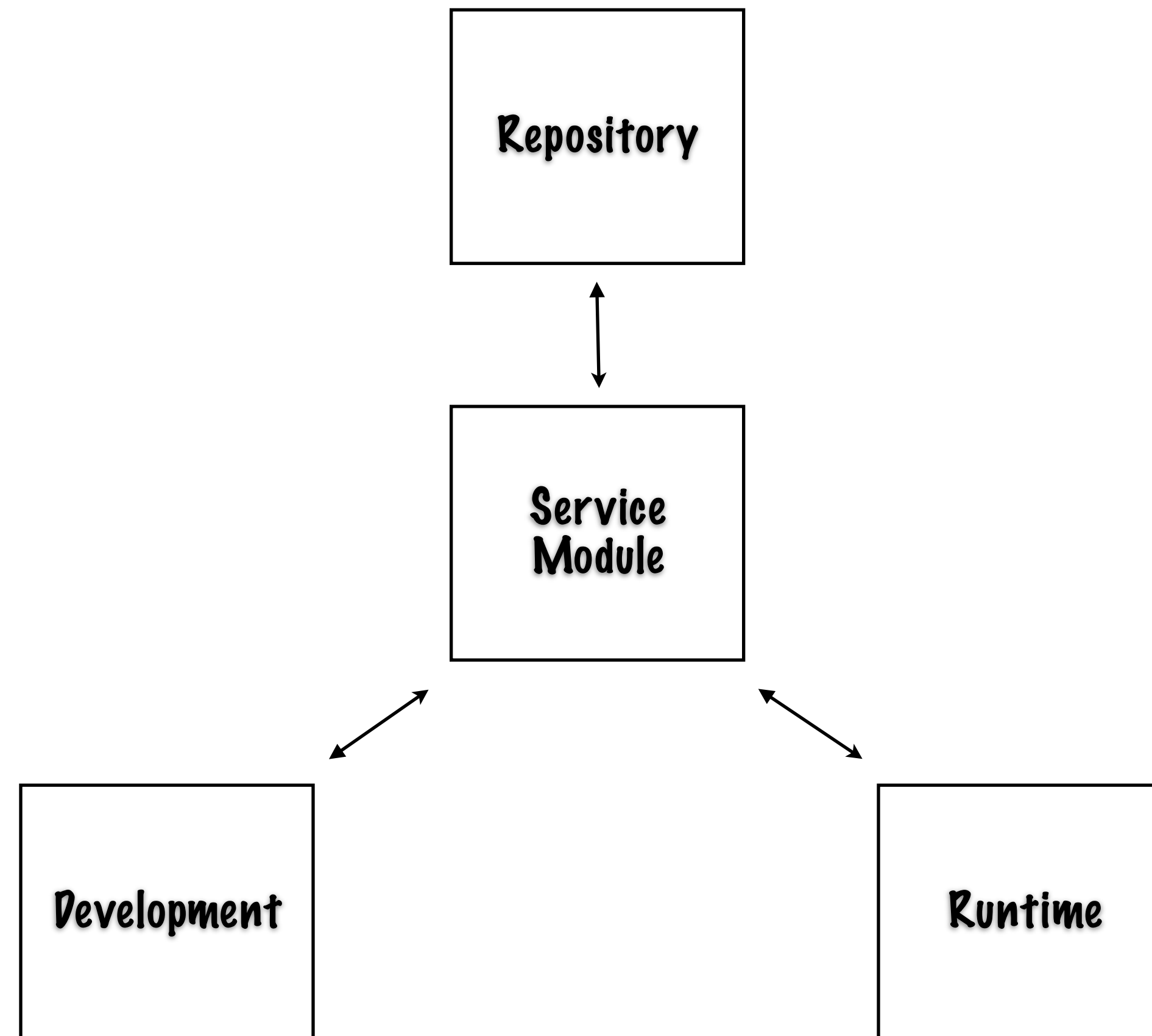
SUMMIT

**JBoss
WORLD**

PRESENTED BY RED HAT



Repository Integration



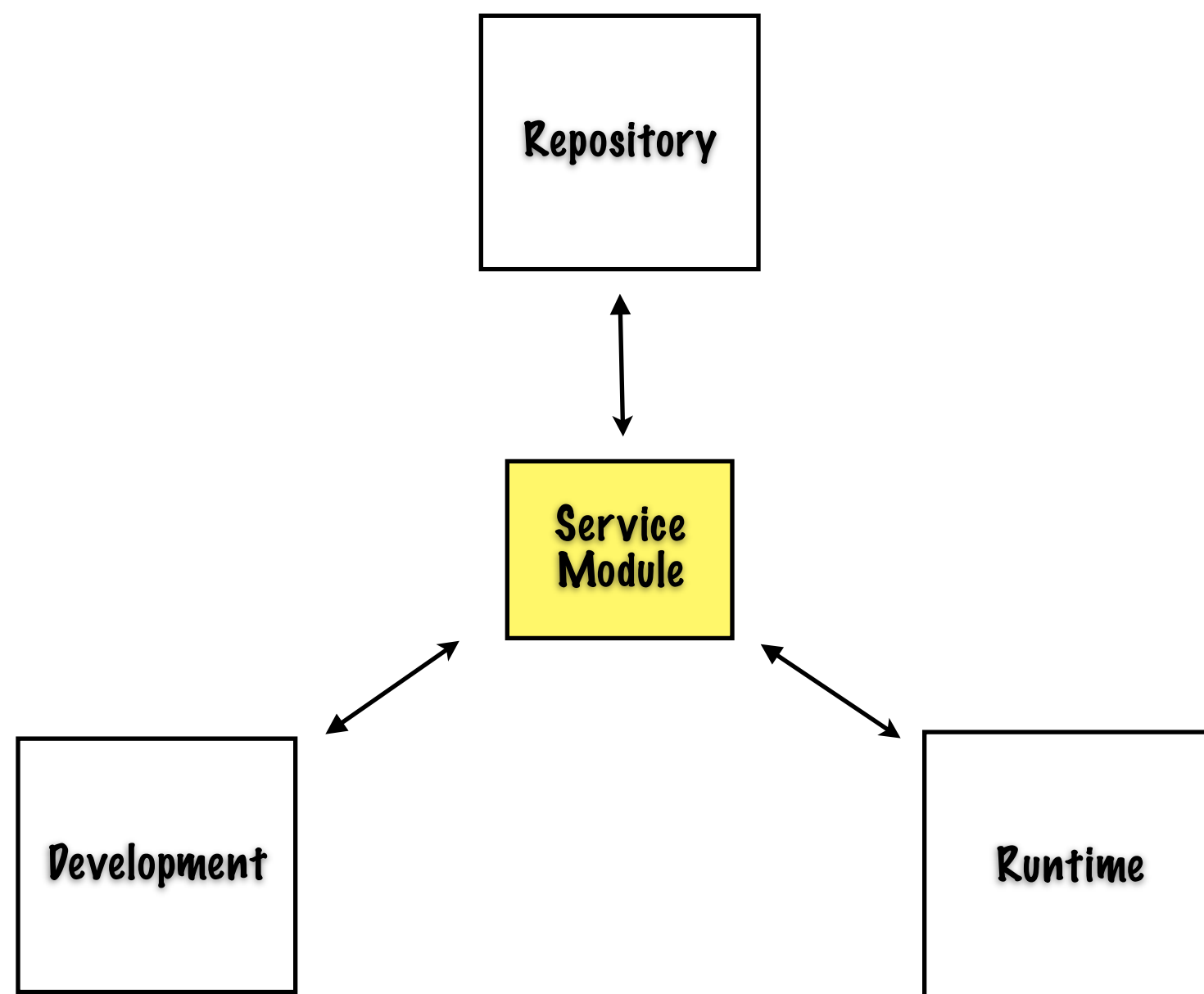
SUMMIT

**JBoss
WORLD**

PRESENTED BY RED HAT



Service Modules



```
<project xmlns="http://maven.apache.org/POM/4.0.0">
  <modelVersion>4.0.0</modelVersion>
  <groupId>org.example</groupId>
  <artifactId>MyServiceArtifacts</artifactId>
  <version>3.2.0</version>
  <name>My Service Artifacts</name>
</project>
```

```
src/main/java:
  MyService.java
```

```
src/main/resources:
  MyService.wsdl
  MyTypes.xsd
```

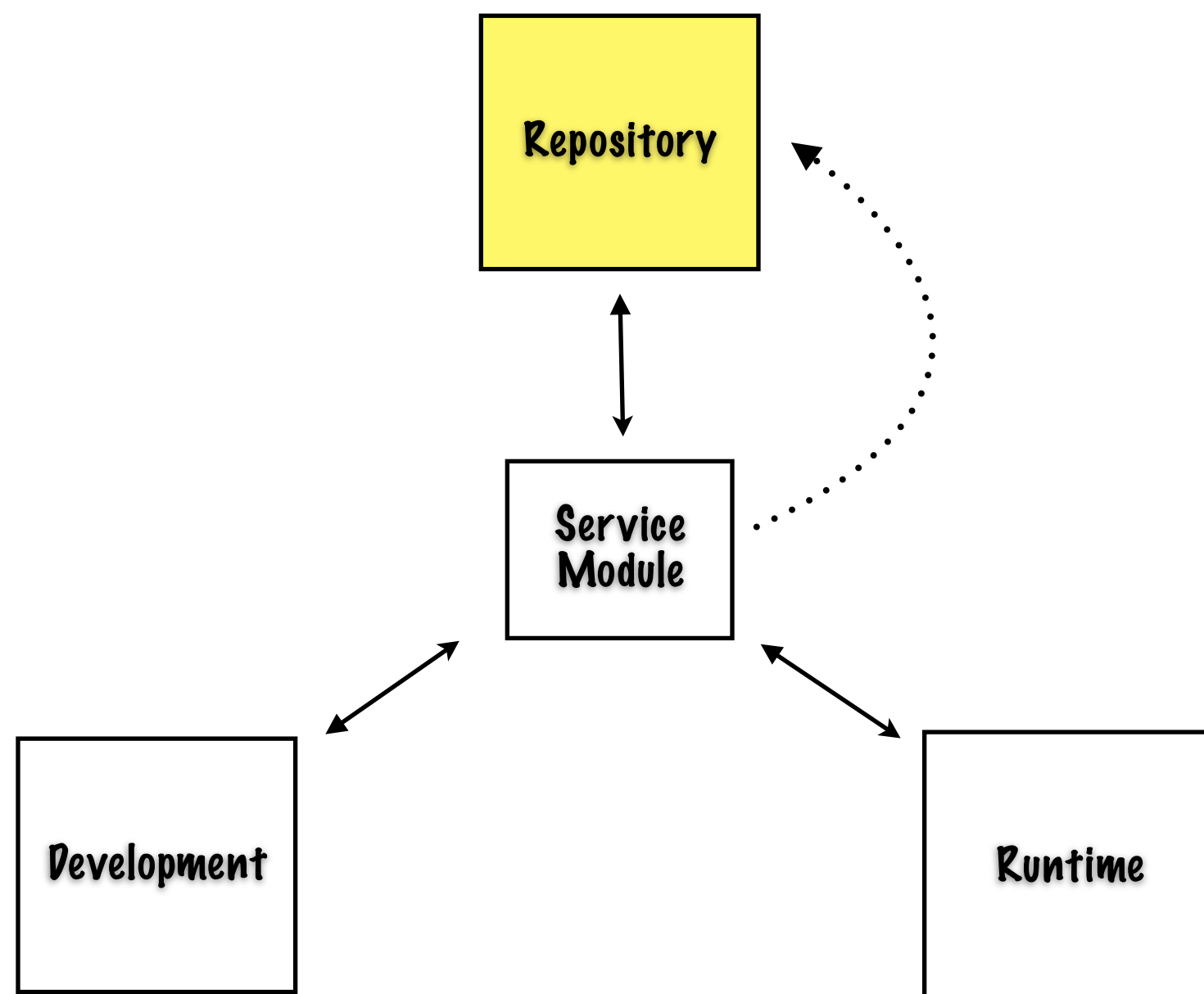
SUMMIT

**JBoss
WORLD**

PRESENTED BY RED HAT



Repository



- Maven
- Guvnor
- S-RAMP

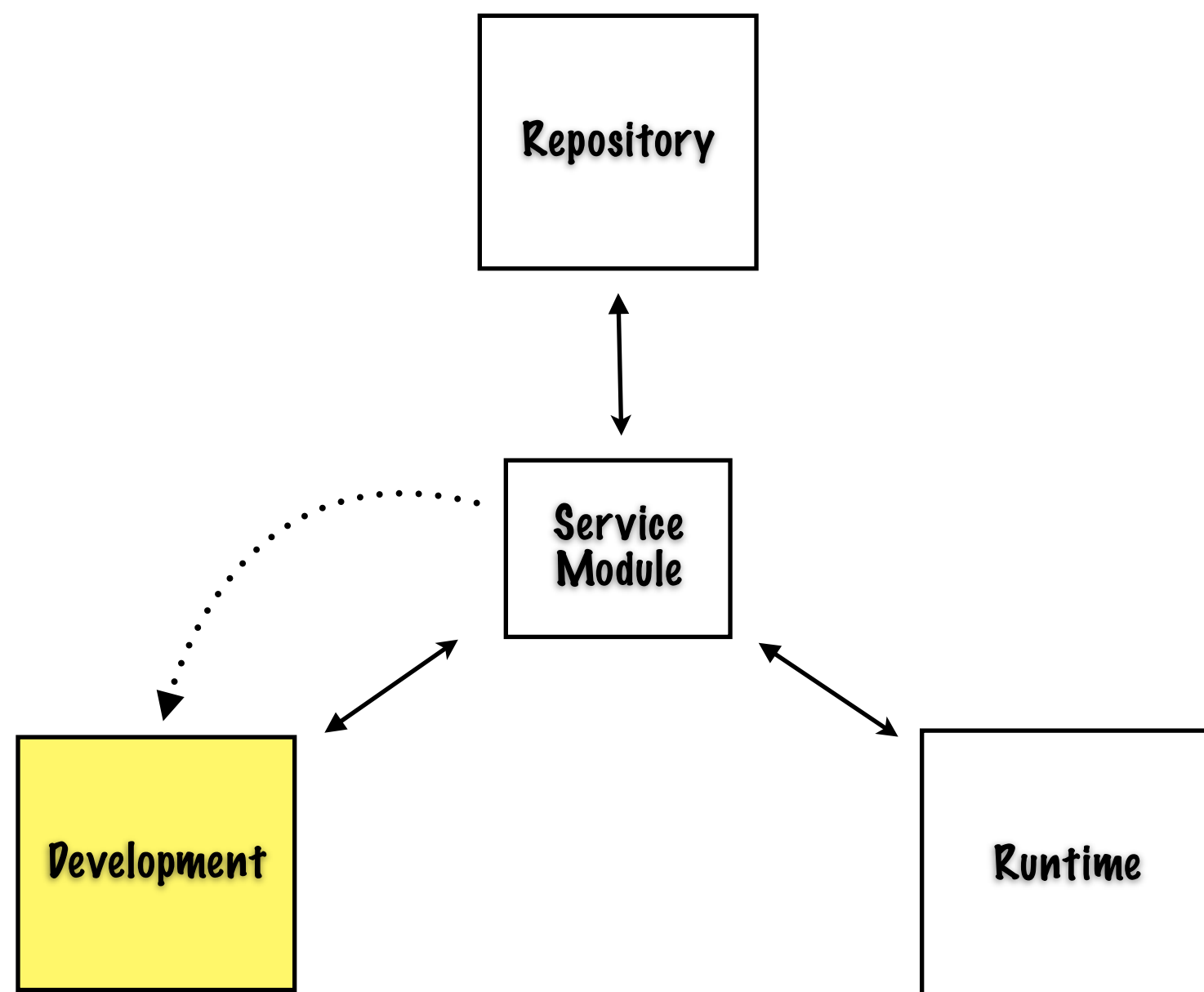
SUMMIT

**JBoss
WORLD**

PRESENTED BY RED HAT



Development



```
<dependency>  
  <groupId>org.example</groupId>  
  <artifactId>MyServiceArtifacts</artifactId>  
  <version>3.2.0</version>  
</dependency>
```

```
<artifacts>  
  <artifact  
    name="MyServiceArtifacts" url="[repo-url]"/>  
</artifacts>
```

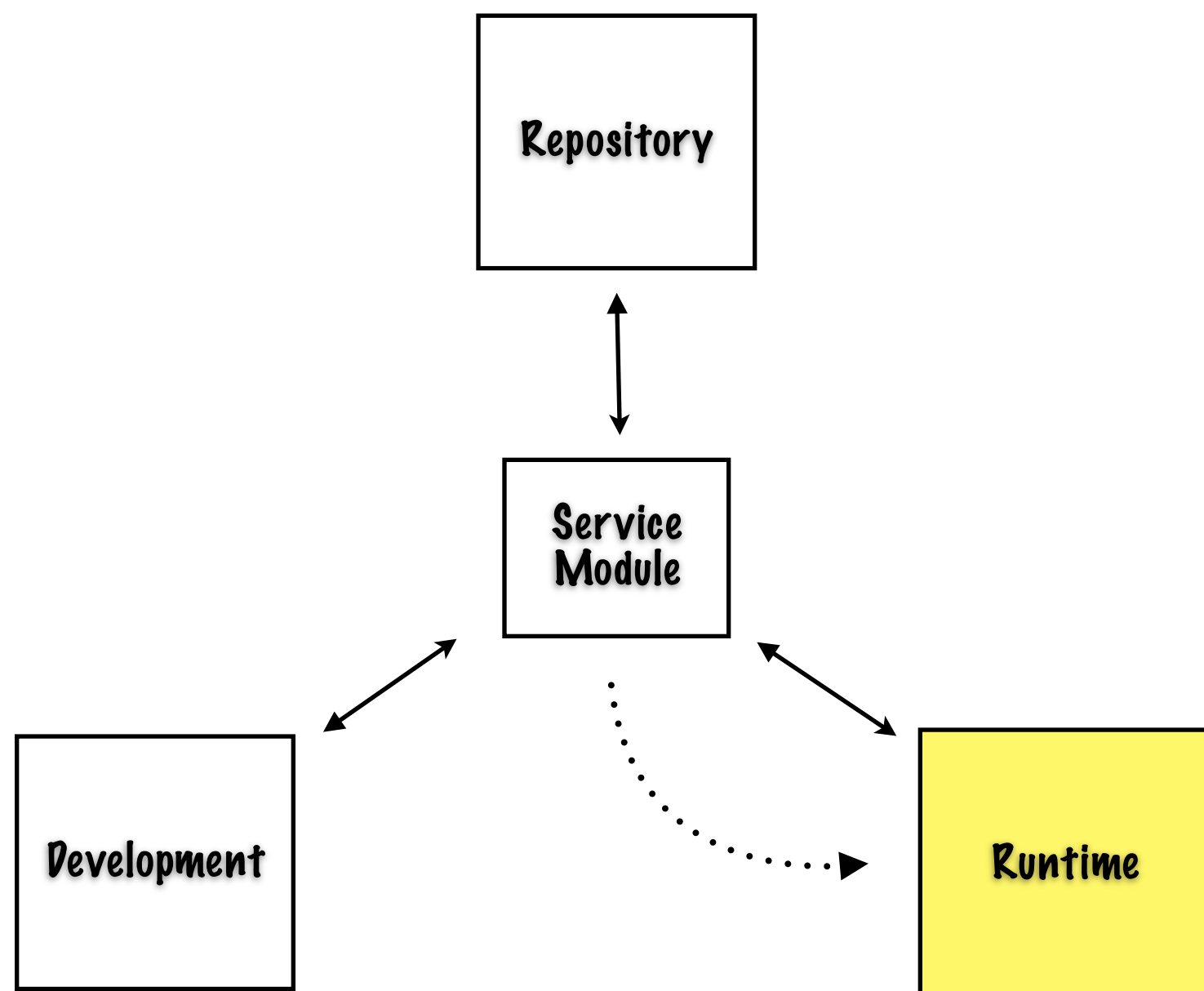
SUMMIT

**JBoss
WORLD**

PRESENTED BY RED HAT



Runtime



```
<plugin>
  <groupId>org.apache.maven.plugins</groupId>
  <artifactId>maven-jar-plugin</artifactId>
  <configuration>
    <archive>
      <manifestEntries>
        <Dependencies>deployment.MyServiceArtifacts.jar</Dependencies>
      </manifestEntries>
    </archive>
  </configuration>
</plugin>
```

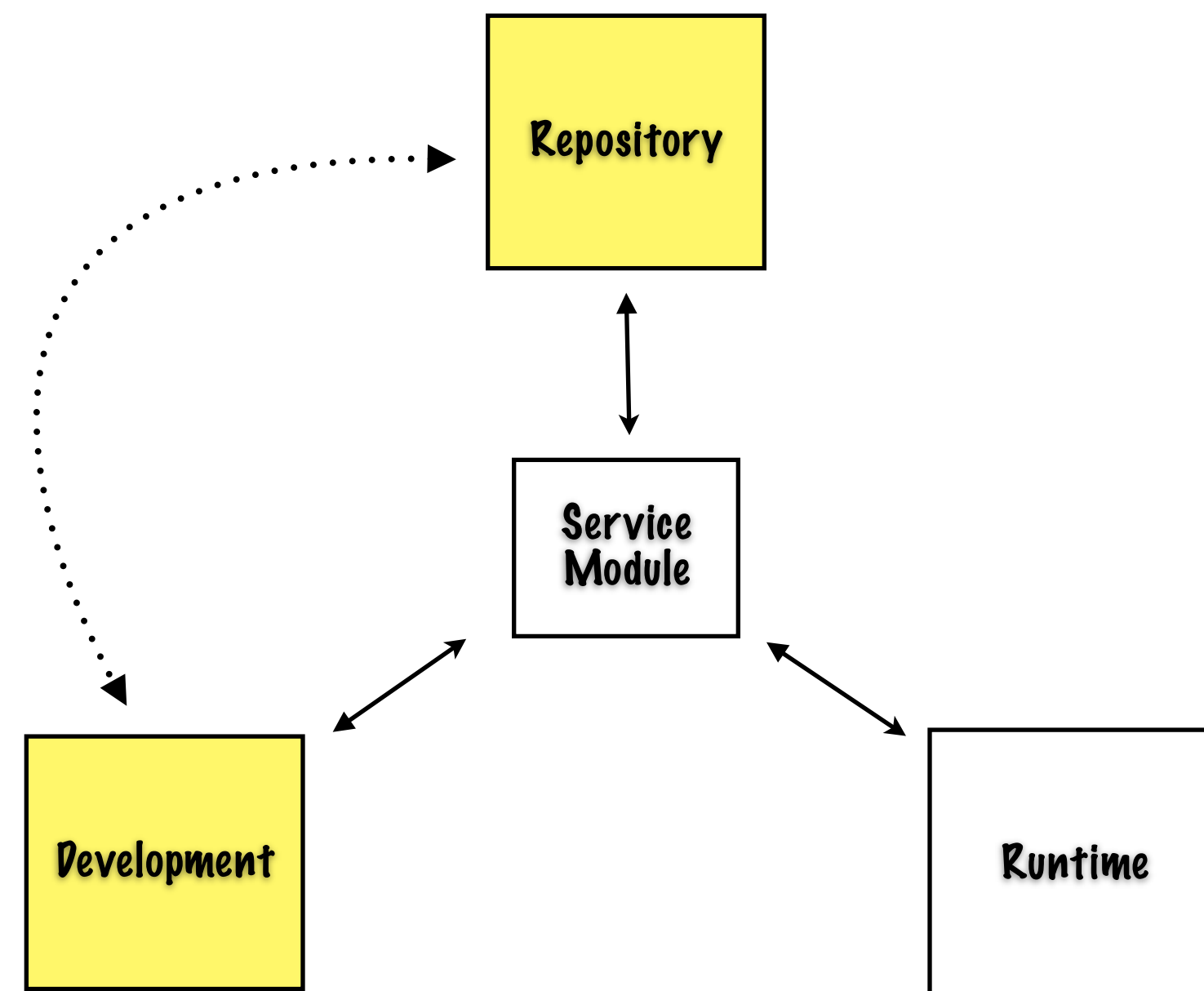
SUMMIT

**JBoss
WORLD**

PRESENTED BY RED HAT



Design-Time Collaboration



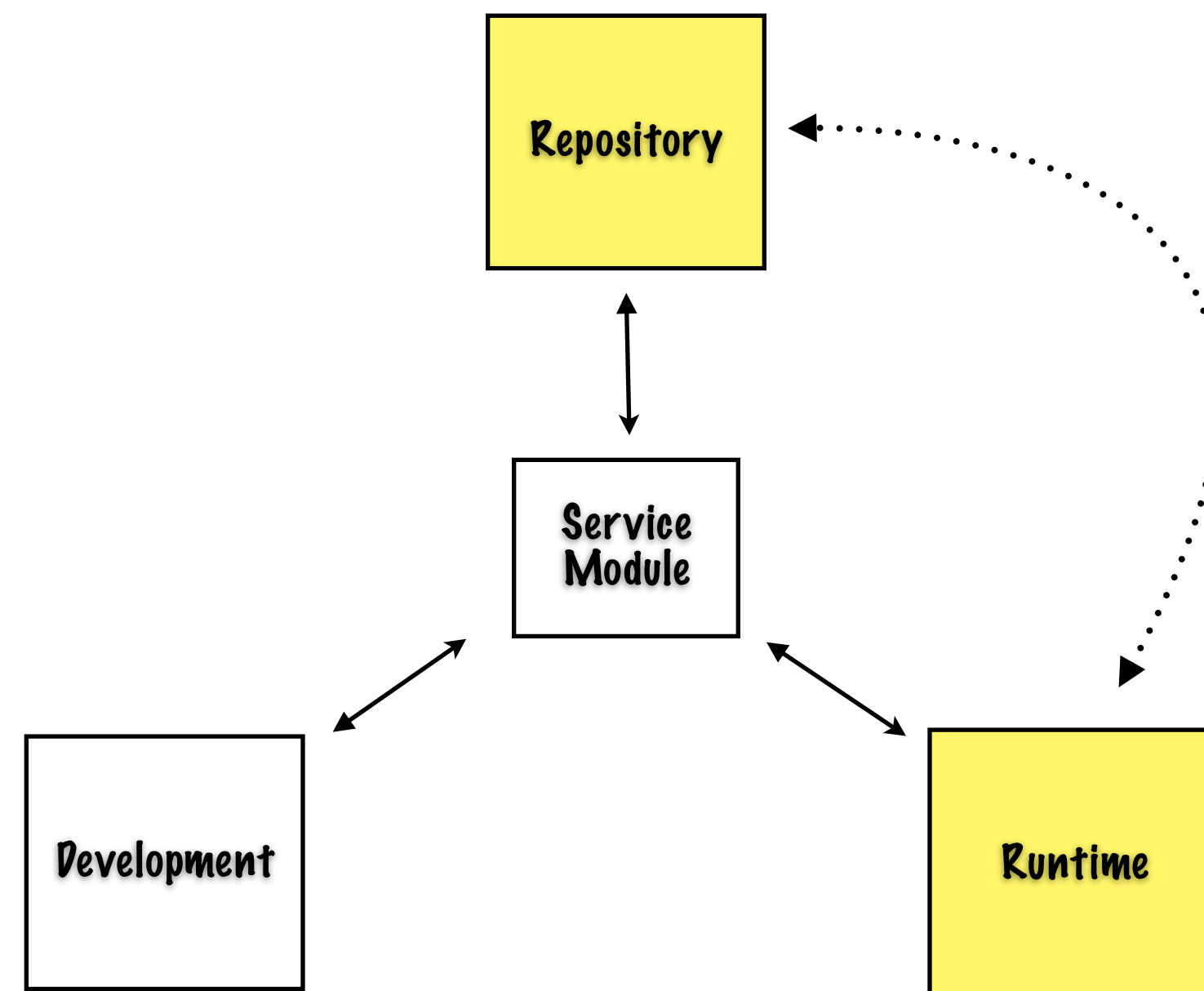
SUMMIT

**JBoss
WORLD**

PRESENTED BY RED HAT



Impact Analysis



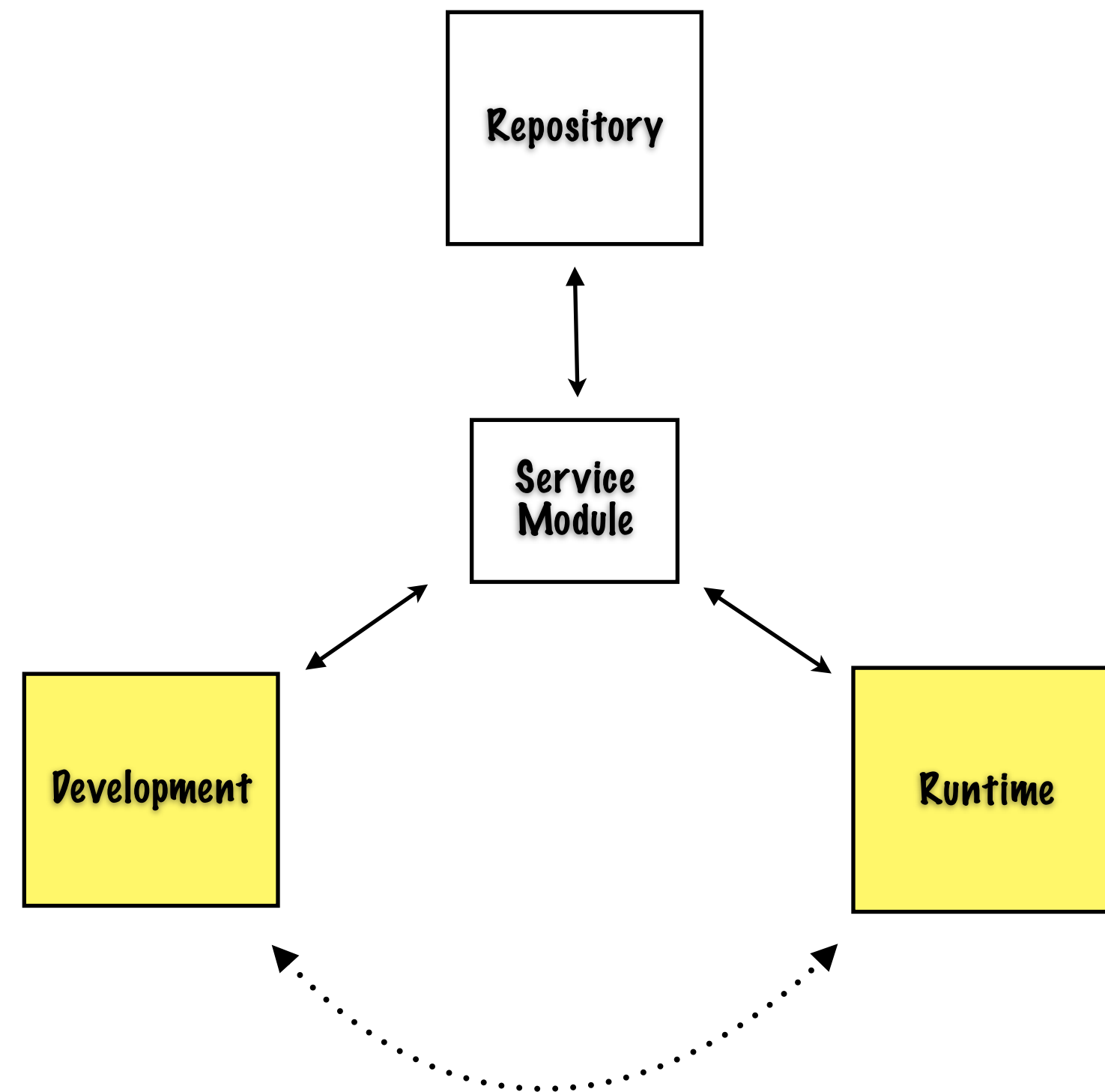
SUMMIT

**JBoss
WORLD**

PRESENTED BY RED HAT



Consistency



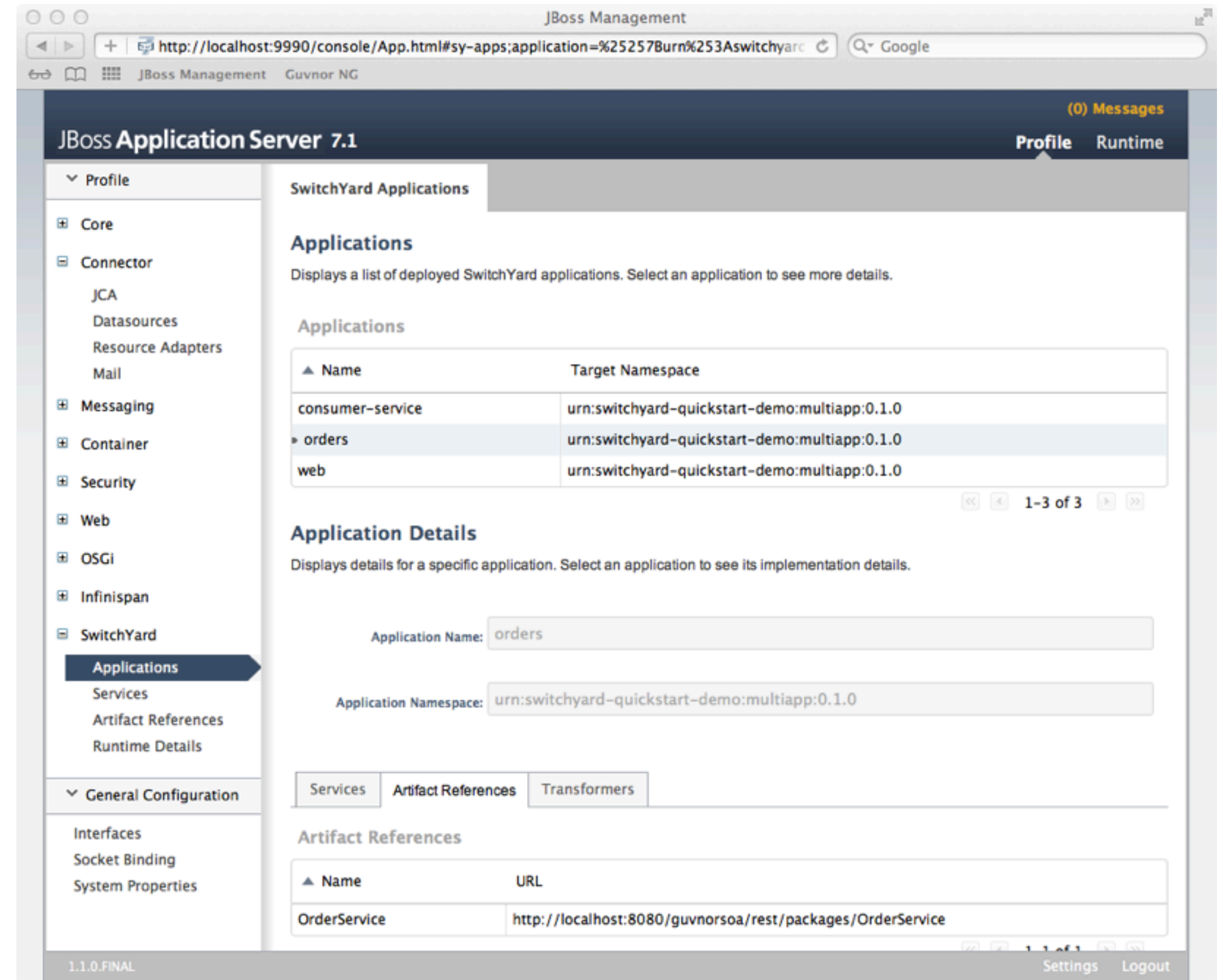
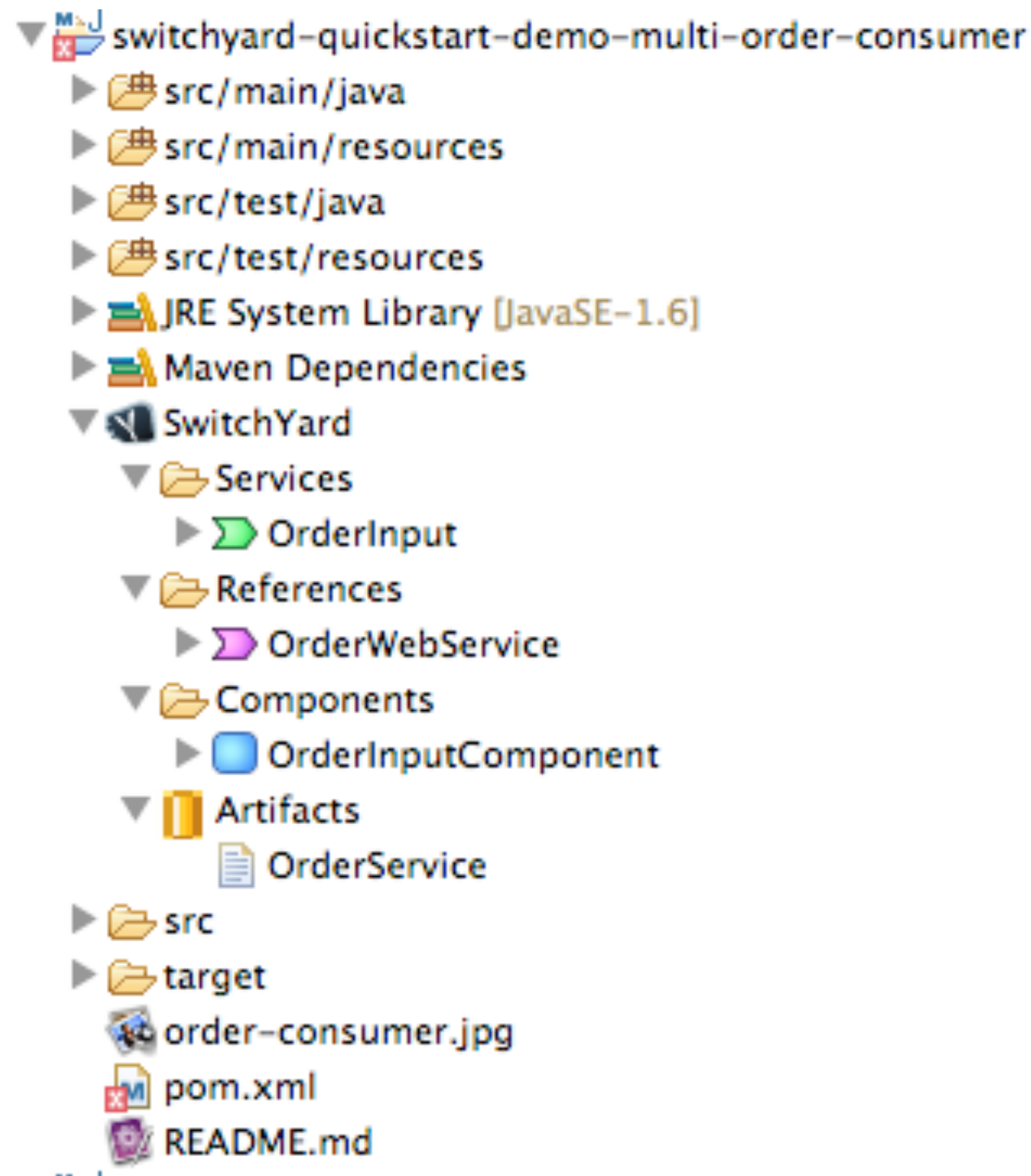
SUMMIT

**JBoss
WORLD**

PRESENTED BY RED HAT



Administration Console



SUMMIT

JBoss
WORLD

PRESENTED BY RED HAT



Administration Console

The screenshot shows the JBoss Management Administration Console interface. The browser address bar displays `http://localhost:9990/console/App.html#sy-artifacts`. The page title is "JBoss Application Server 7.1". The left sidebar contains a navigation menu with categories like Profile, Core, Connector, Messaging, Container, Security, Web, OSGI, Infinispan, SwitchYard, and General Configuration. The "Artifact References" option under SwitchYard is selected. The main content area is titled "SwitchYard Artifact References" and "Artifact References". It includes a description: "Displays all artifacts referenced throughout the system, along with the applications referencing a specific artifact." Below this, there are two tables. The first table, "Artifact References", has columns "Name" and "URL" and contains one entry: "OrderService" with URL "http://localhost:8080/guvnorsoa/rest/packages/OrderService". The second table, "Applications Using Artifact", has columns "Name" and "Target Namespace" and contains three entries: "consumer-service", "orders", and "web", all with target namespace "urn:switchyard-quickstart-demo:multiapp:0.1.0". The bottom of the page shows the version "1.1.0.FINAL" and links for "Settings" and "Logout".

SUMMIT

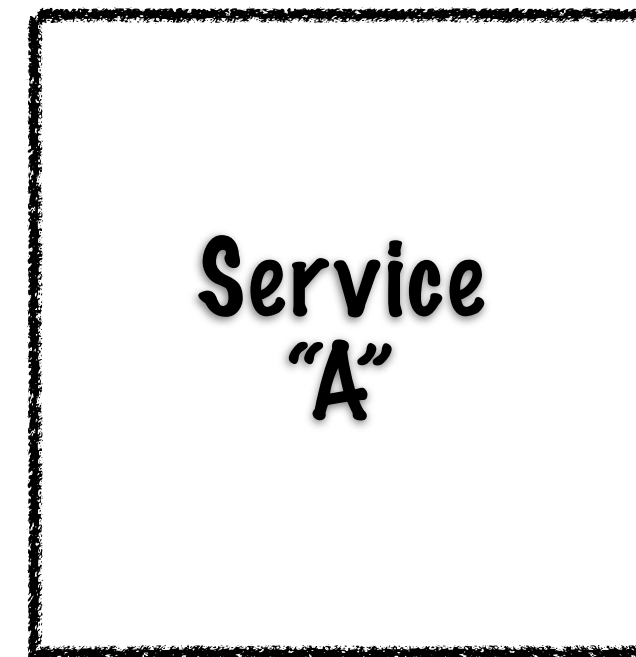
JBoss
WORLD

PRESENTED BY RED HAT



Runtime Governance

Service Metrics



SUMMIT

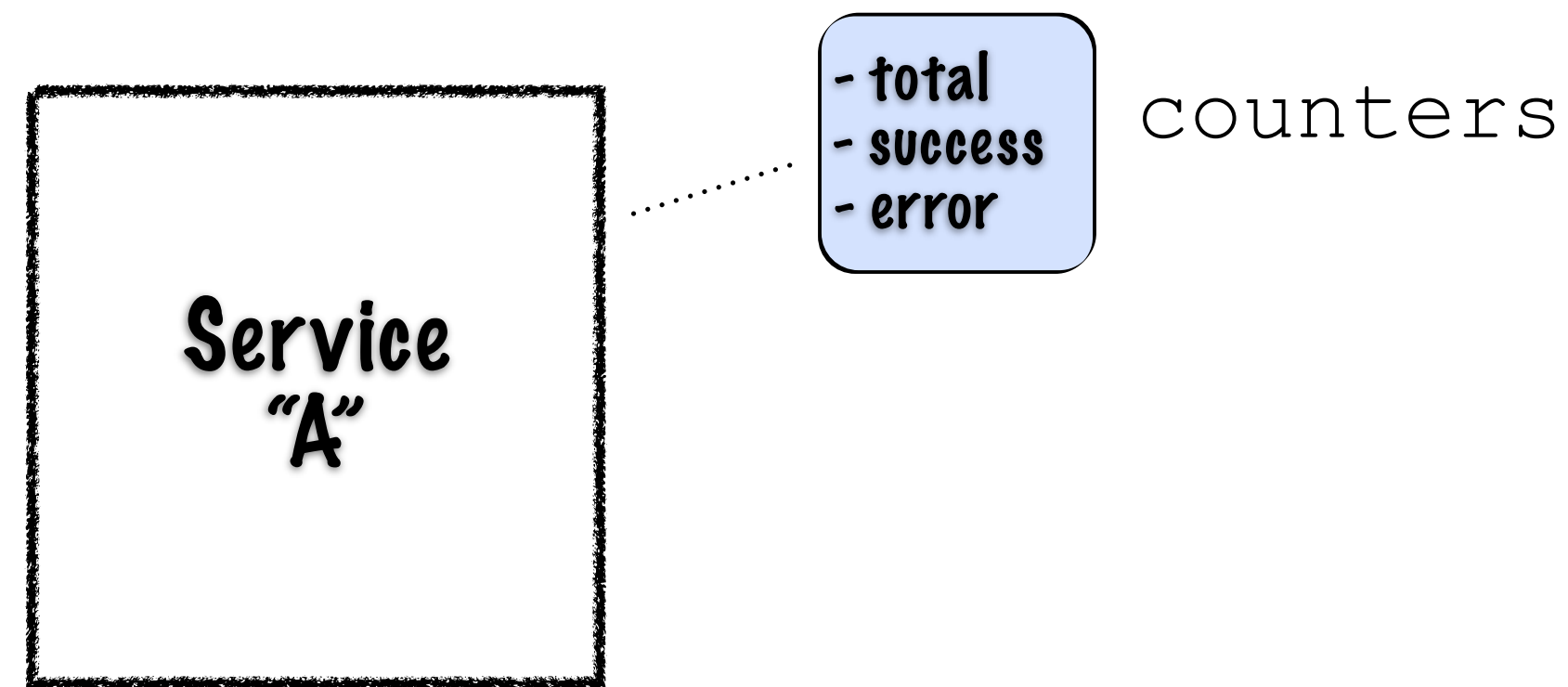
**JBoss
WORLD**

PRESENTED BY RED HAT



Runtime Governance

Service Metrics



SUMMIT

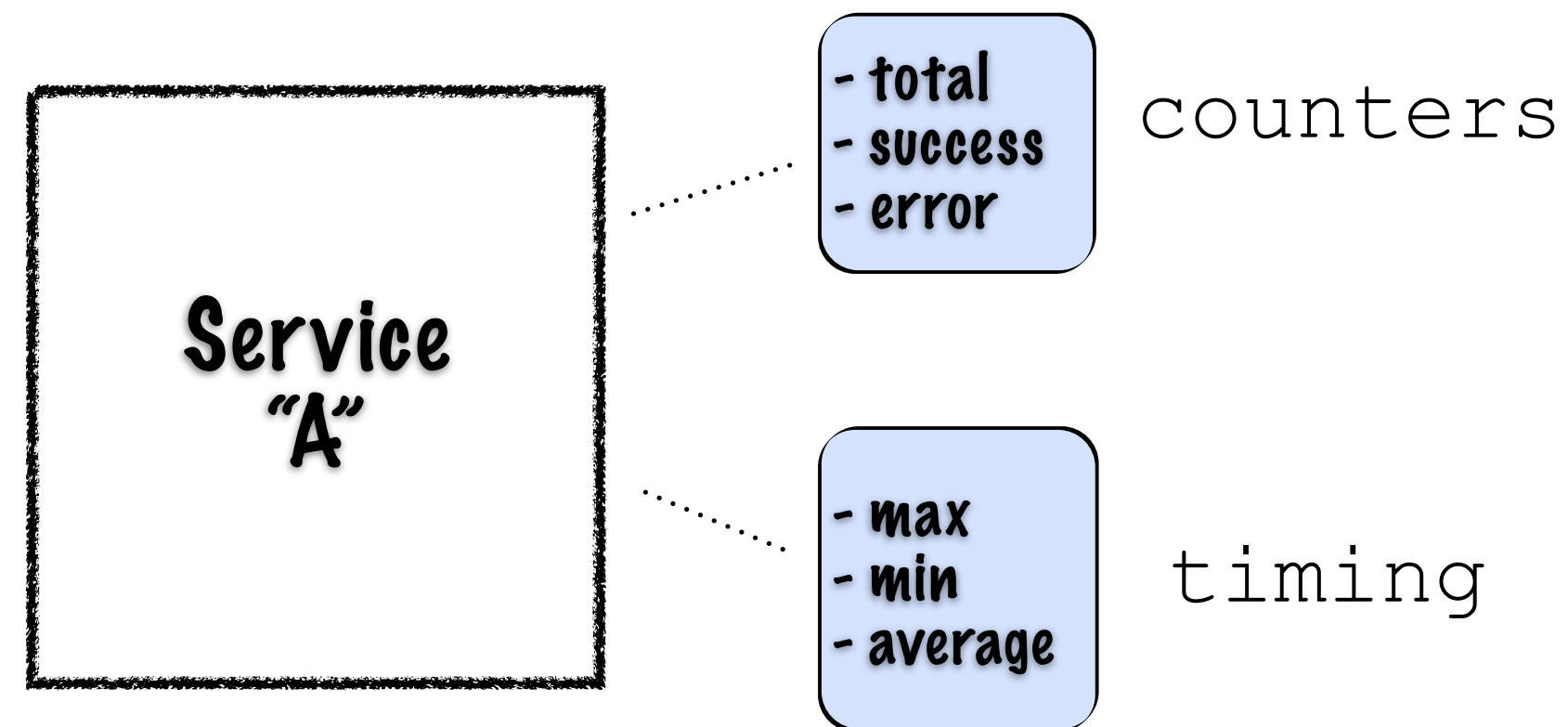
**JBoss
WORLD**

PRESENTED BY RED HAT



Runtime Governance

Service Metrics



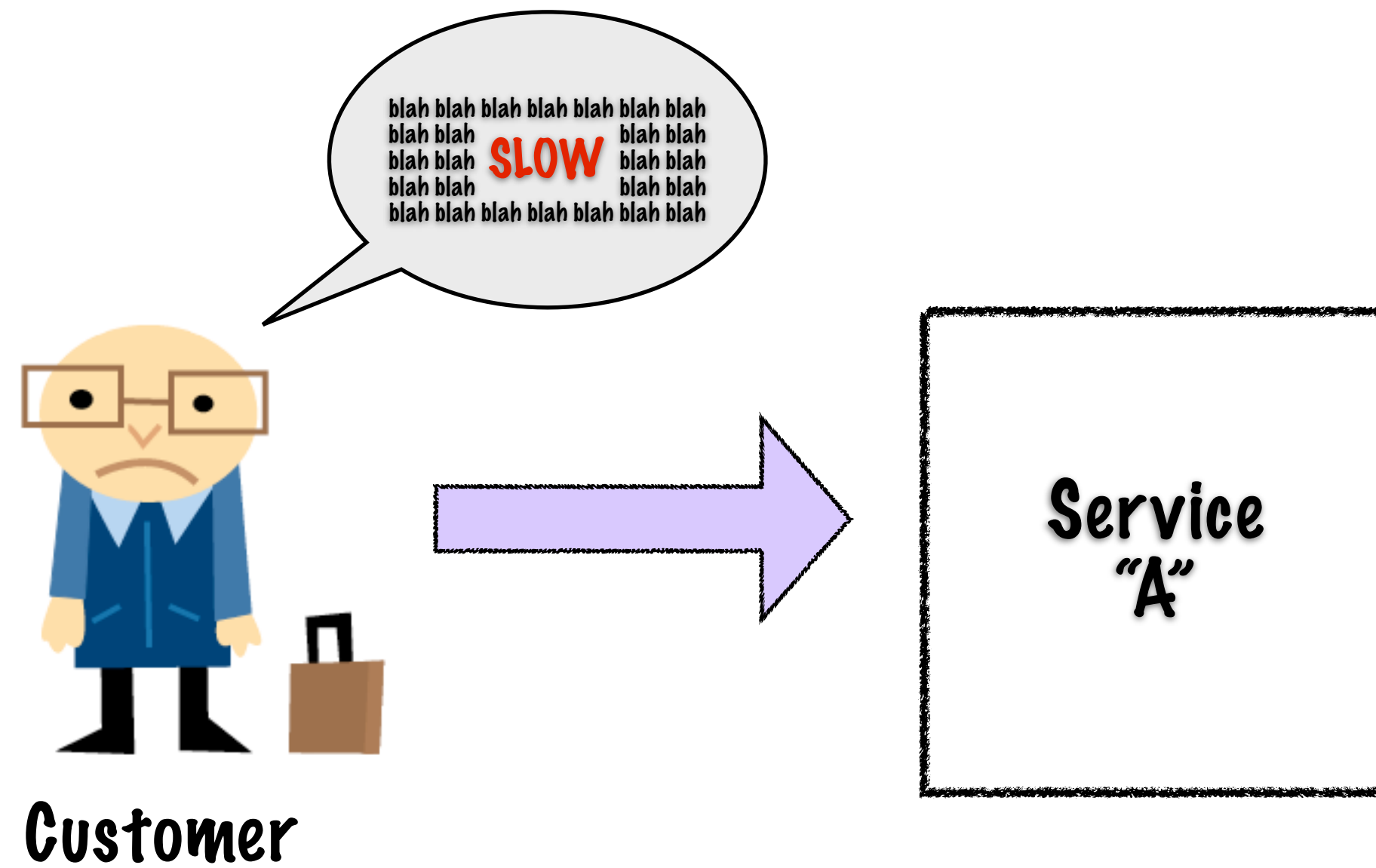
SUMMIT

JBoss
WORLD

PRESENTED BY RED HAT



Metrics In Action



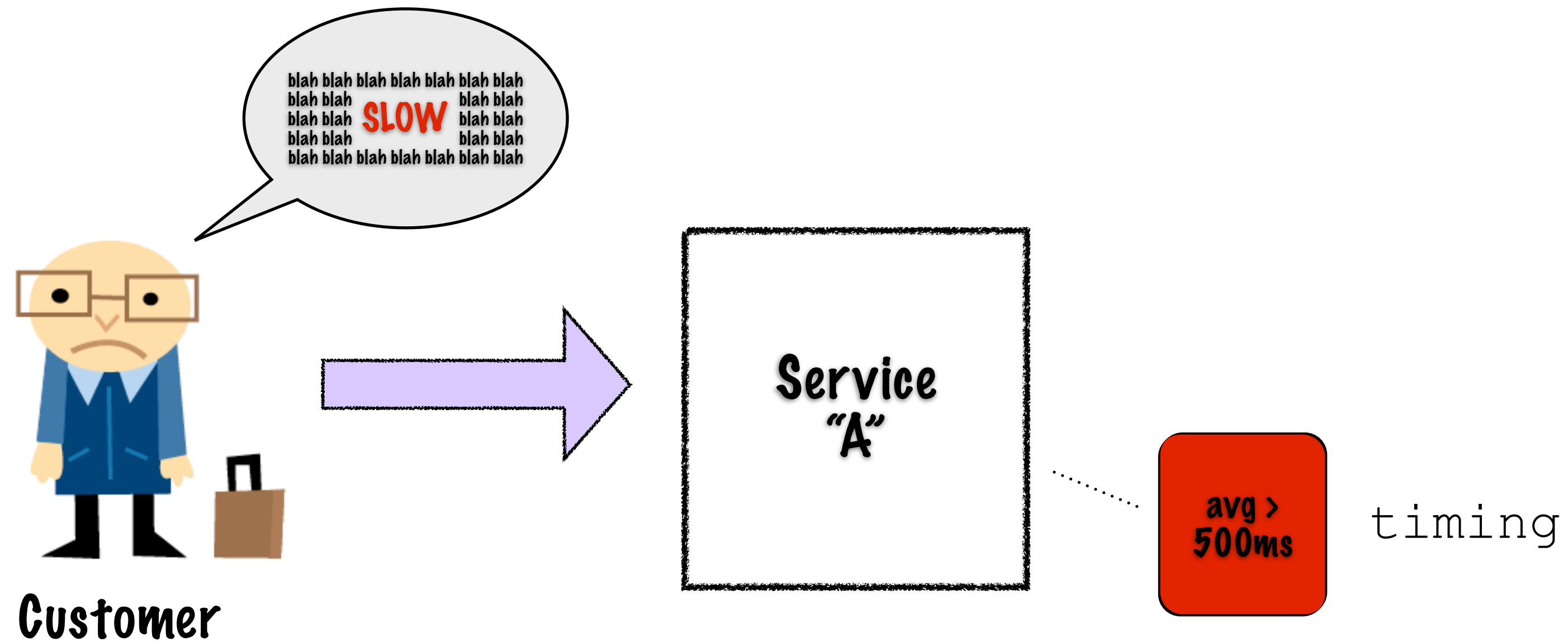
SUMMIT

JBoss
WORLD

PRESENTED BY RED HAT



Metrics In Action



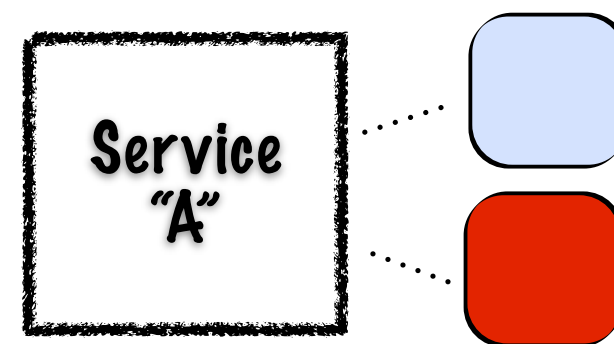
SUMMIT

JBoss
WORLD

PRESENTED BY RED HAT



Metrics at Scale



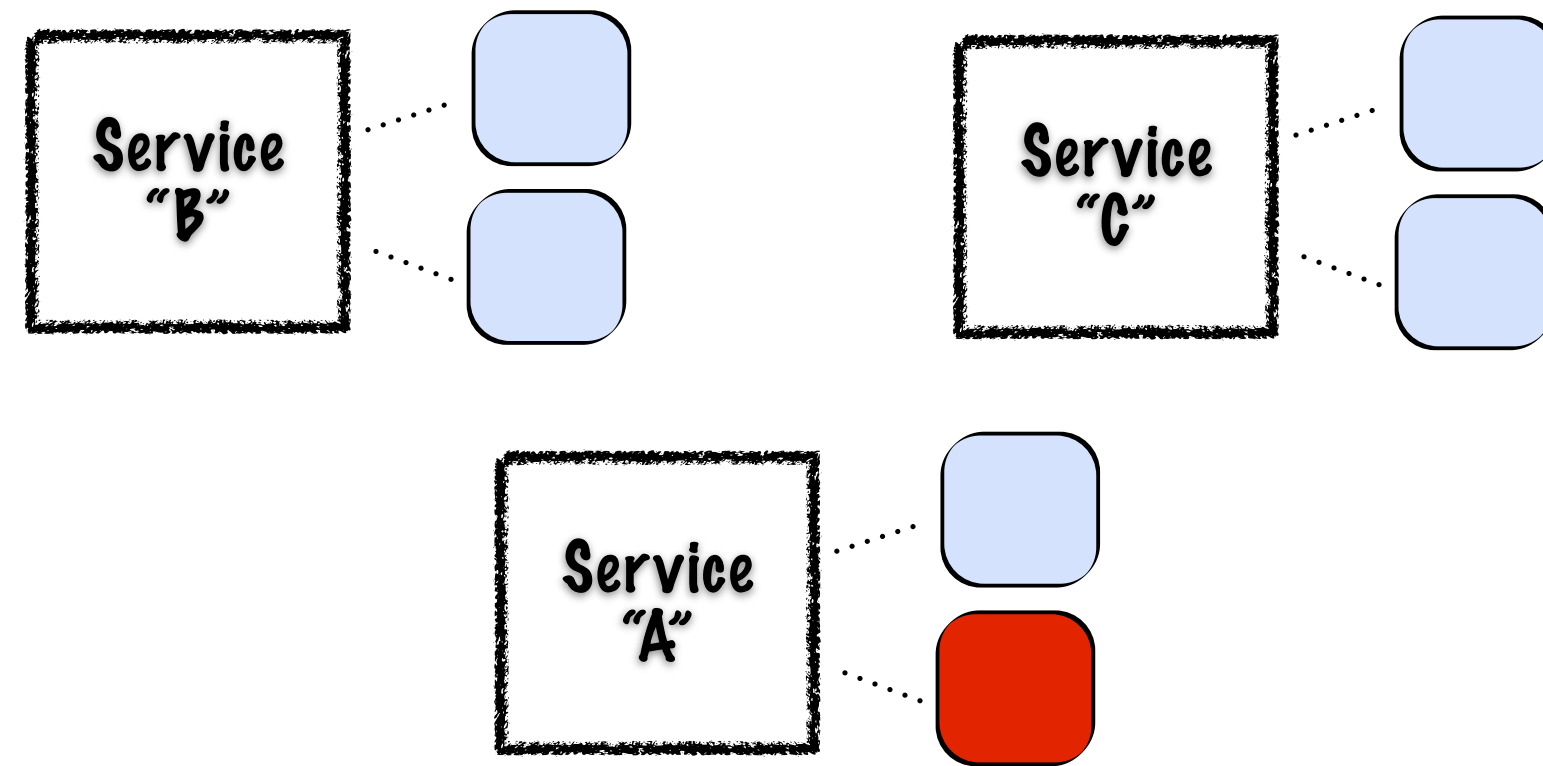
SUMMIT

**JBoss
WORLD**

PRESENTED BY RED HAT



Metrics at Scale



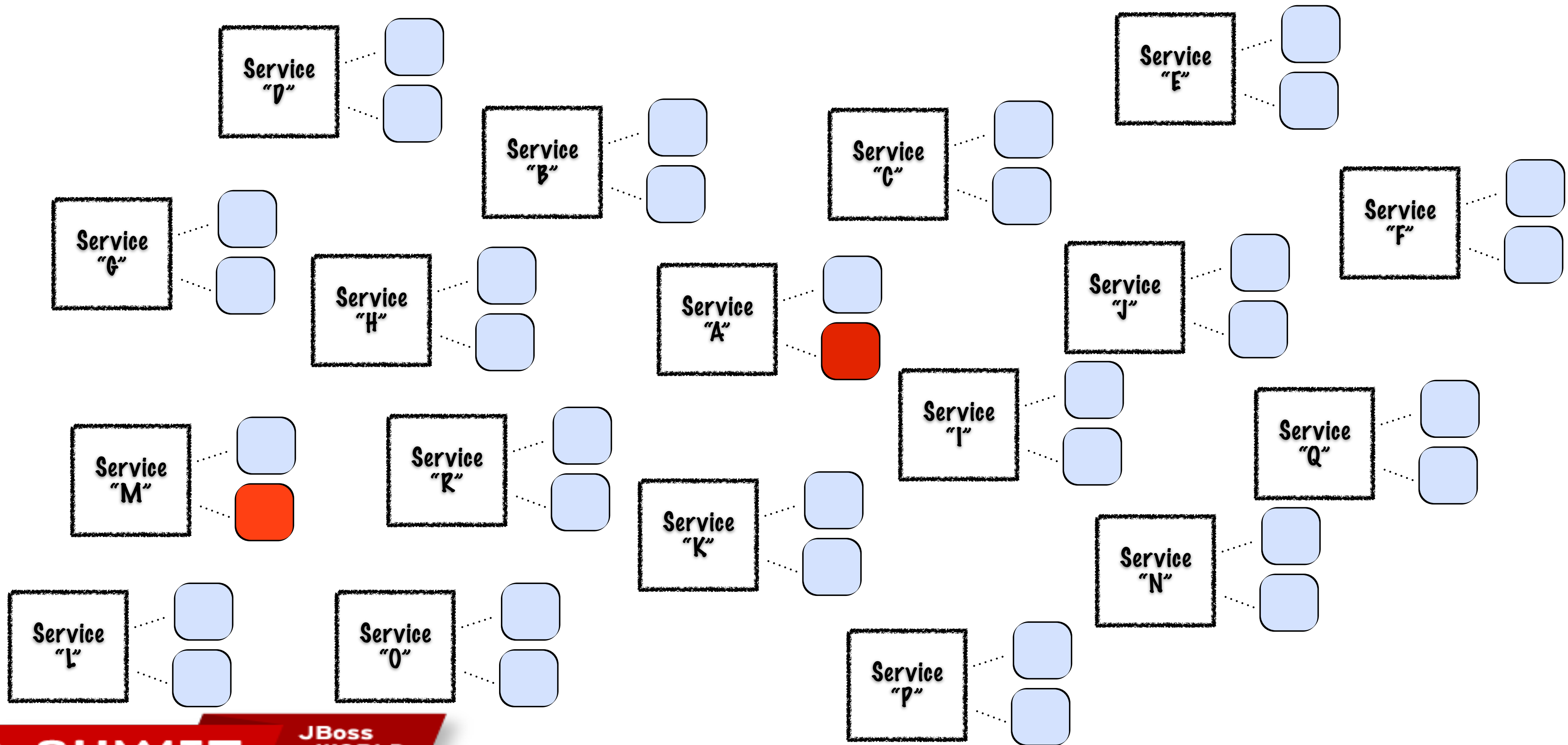
SUMMIT

**JBoss
WORLD**

PRESENTED BY RED HAT



Metrics at Scale



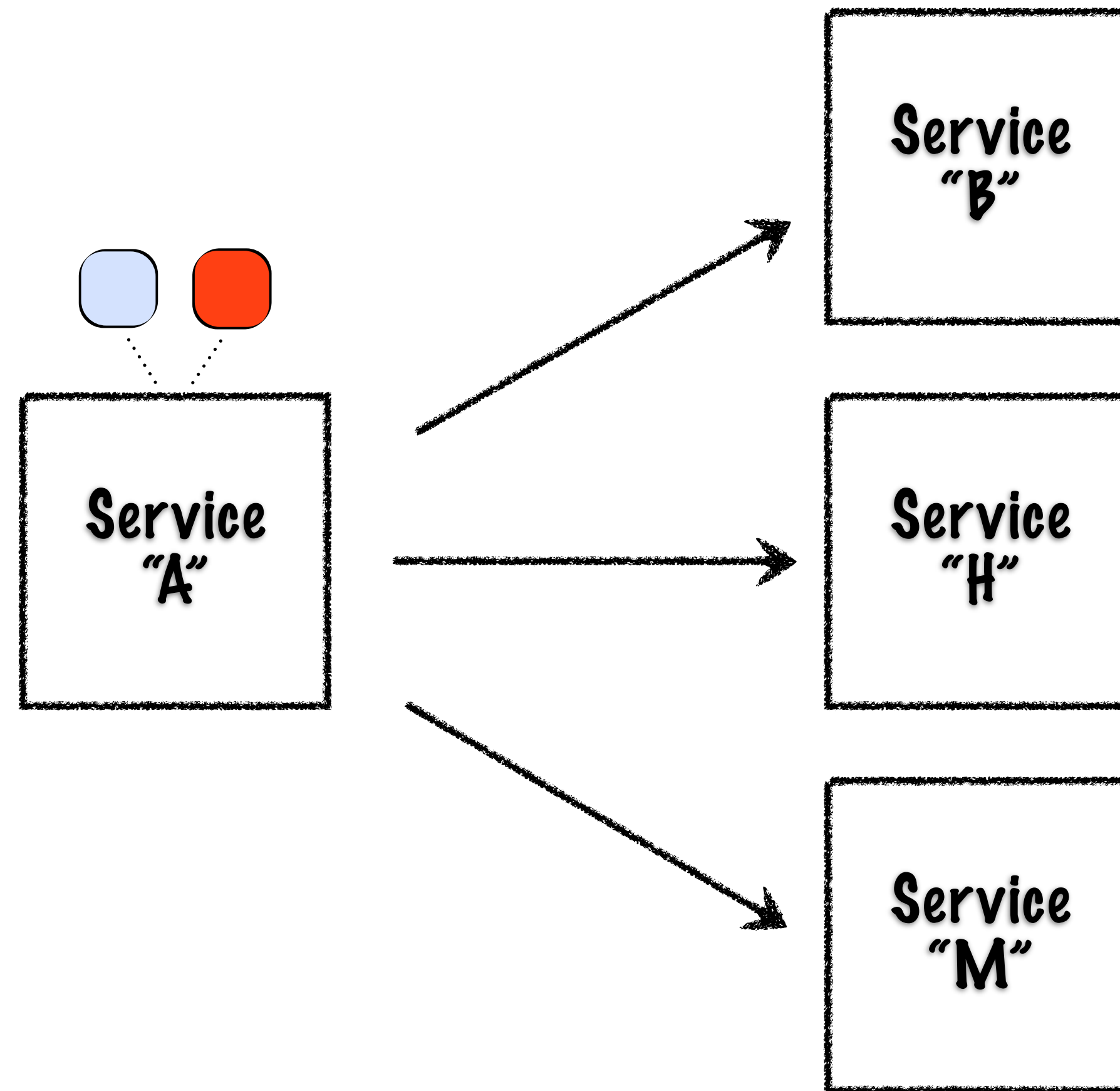
SUMMIT

**JBoss
WORLD**

PRESENTED BY RED HAT



Service References



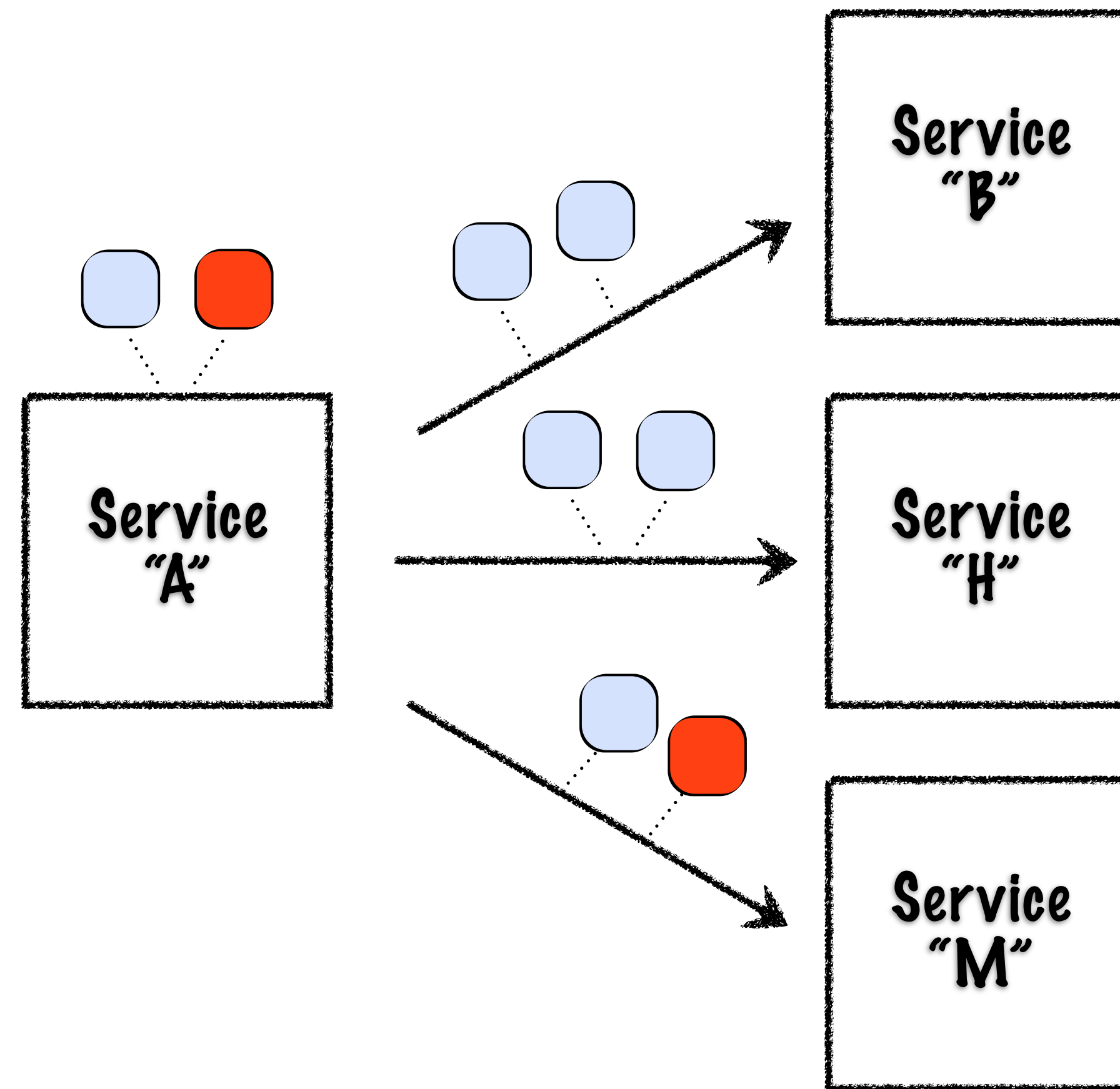
SUMMIT

**JBoss
WORLD**

PRESENTED BY RED HAT



Service References



SUMMIT

**JBoss
WORLD**

PRESENTED BY RED HAT



Service Metrics

The screenshot shows the JBoss Management console interface. The browser address bar displays the URL: `http://localhost:9990/console/App.html#sy-runtime;service=%257Burn%253Aswitchyard-ql`. The page title is "JBoss Application Server 7.1". The left sidebar contains a navigation menu with categories: Server Status, Subsystem Metrics, Runtime Operations, and Deployments. The "SwitchYard" subsystem is selected. The main content area is titled "SwitchYard Message Metrics" and includes a description: "Displays message metrics for individual services. Select a service to see message metrics for a specific service." Below this is a table of services:

Name	Target Namespace
loanService	urn:bpel:test:1.0
OrderService	urn:switchyard-quickstart:bean-service:0.1.0
ProcessOrder	urn:switchyard-quickstart:bpm-service:1.0
RedService	urn:switchyard-quickstart:rules-camel-cbr:0.1.0
RoutingService	urn:switchyard-quickstart:rules-camel-cbr:0.1.0

Below the services table are two tabs: "Service Metrics" (selected) and "Reference Metrics". The "Service Metrics" section contains two sub-sections: "Message Counts" and "Processing Times".

Message Counts

Metric	Actual	Percentage
Total Count:	381	33%
Success Count:	381	100%
Fault Count:	0	0%

Processing Times

Metric	Actual	Percentage
Total Processing Time:	292794	50%
Average Processing Time:	768.488188976378	
Min. Processing Time:	5	
Max. Processing Time:	1992	

The footer of the console shows "1.1.0.FINAL" on the left and "Settings Logout" on the right.

SUMMIT

JBoss
WORLD

PRESENTED BY RED HAT



Reference Metrics

The screenshot shows the JBoss Management console interface. The browser address bar indicates the URL: `http://localhost:9990/console/App.html#sy-runtime;service=%257Burn%253Aswitchyard-ql`. The page title is "JBoss Application Server 7.1" and the current view is "Runtime".

The left sidebar contains a navigation menu with the following sections:

- Server Status
 - Configuration
 - JVM
- Subsystem Metrics
 - Datasources
 - JPA
 - JMS Destinations
 - SwitchYard (selected)
 - Transactions
 - Web
- Runtime Operations
 - OSGi
- Deployments
 - Manage Deployments
 - Webservices

The main content area is titled "SwitchYard Message Metrics" and includes a description: "Displays message metrics for individual services. Select a service to see message metrics for a specific service." Below this is a table of services:

Name	Target Namespace
loanService	urn:bpel:test:1.0
OrderService	urn:switchyard-quickstart:bean-service:0.1.0
ProcessOrder	urn:switchyard-quickstart:bpm-service:1.0
RedService	urn:switchyard-quickstart:rules-camel-cbr:0.1.0
RoutingService	urn:switchyard-quickstart:rules-camel-cbr:0.1.0

Below the services table are two tabs: "Service Metrics" and "Reference Metrics". The "Reference Metrics" tab is active, showing a table of "Referenced Service Metrics":

Name	Message Count	Average Time	Time %	Fault %
BlueService	0	0	0%	0%
DestinationService	382	0.1518324607329843	0%	0%
GreenService	191	49.74869109947644	3%	0%
RedService	190	1483.9315789473685	96%	0%

The footer of the console shows "1.1.0.FINAL" on the left and "Settings Logout" on the right.

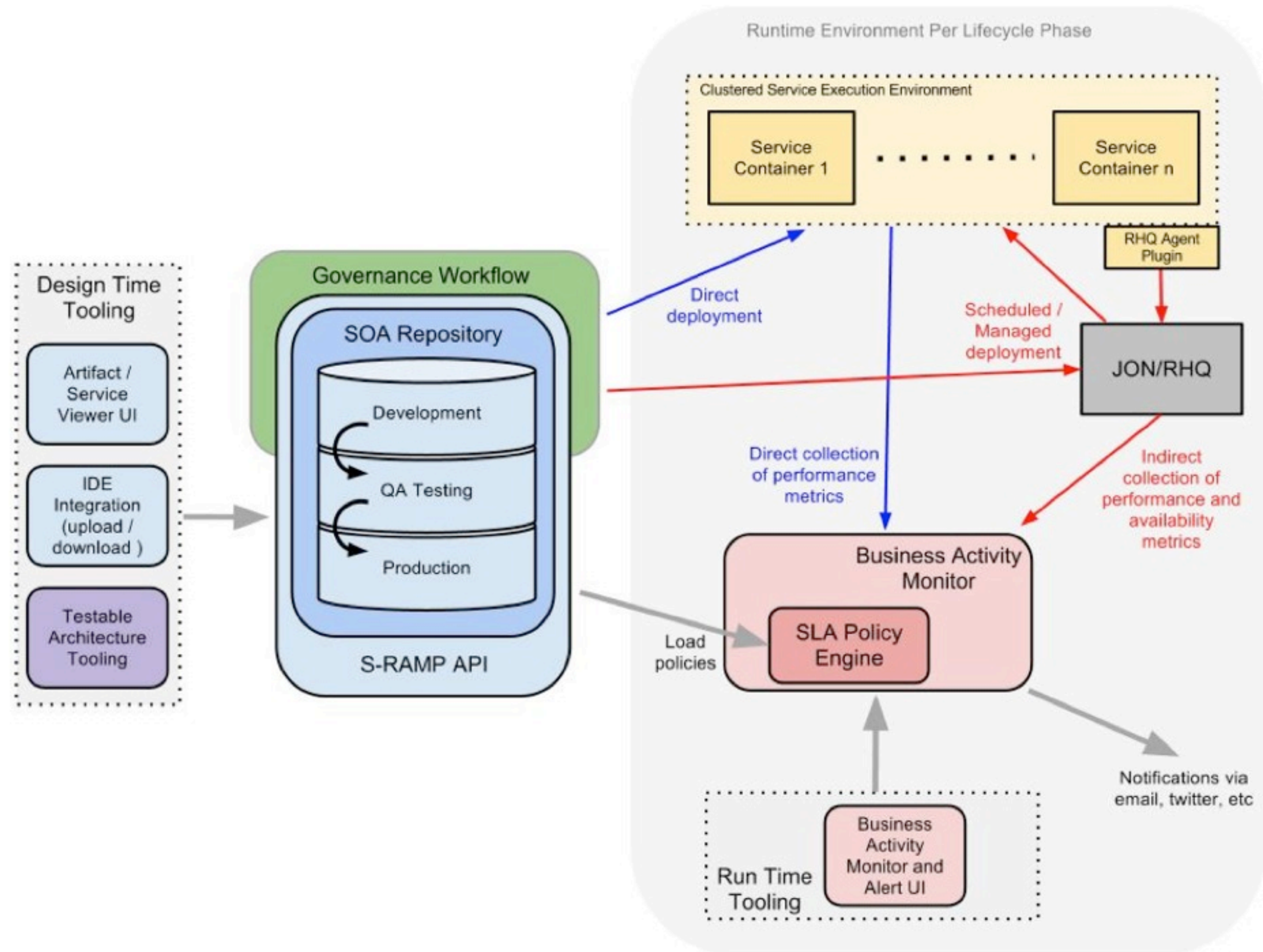
SUMMIT

JBoss
WORLD

PRESENTED BY RED HAT



Governance Architecture



<http://www.jboss.org/overlord>

SUMMIT

JBoss
WORLD

PRESENTED BY RED HAT



SOA Repository

- S-RAMP (SOA Repository and Model Protocol)
 - Common data model for service artifacts and metadata
 - Standardized CRUD access method (ATOM binding)
- Built-in Governance Workflows
- Maven integration

<https://github.com/governance>

SUMMIT

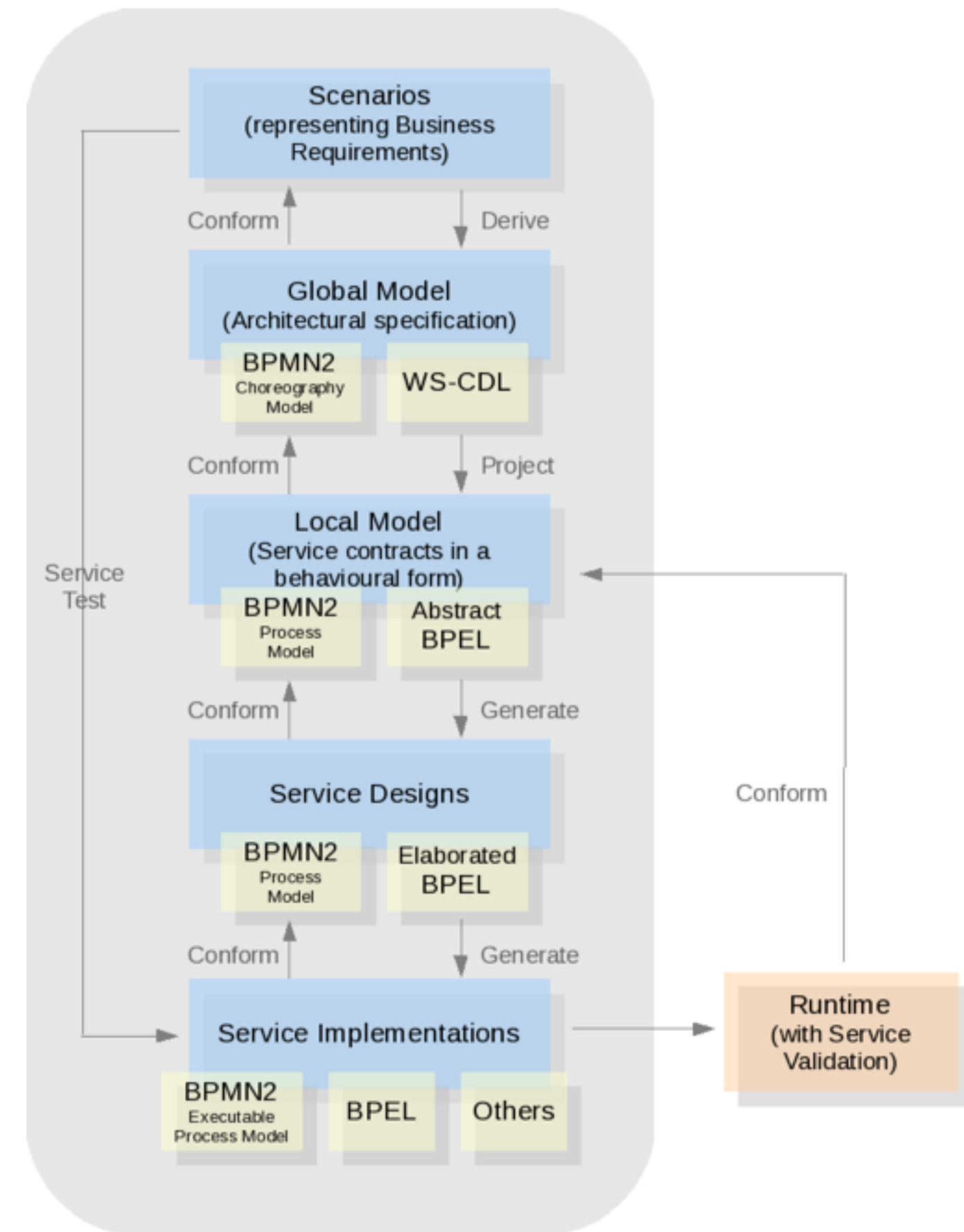
**JBoss
WORLD**

PRESENTED BY RED HAT



Savara

- Testable architecture
- Start with use cases
- Generate
 - Choreography, Services, Implementations
- Validate



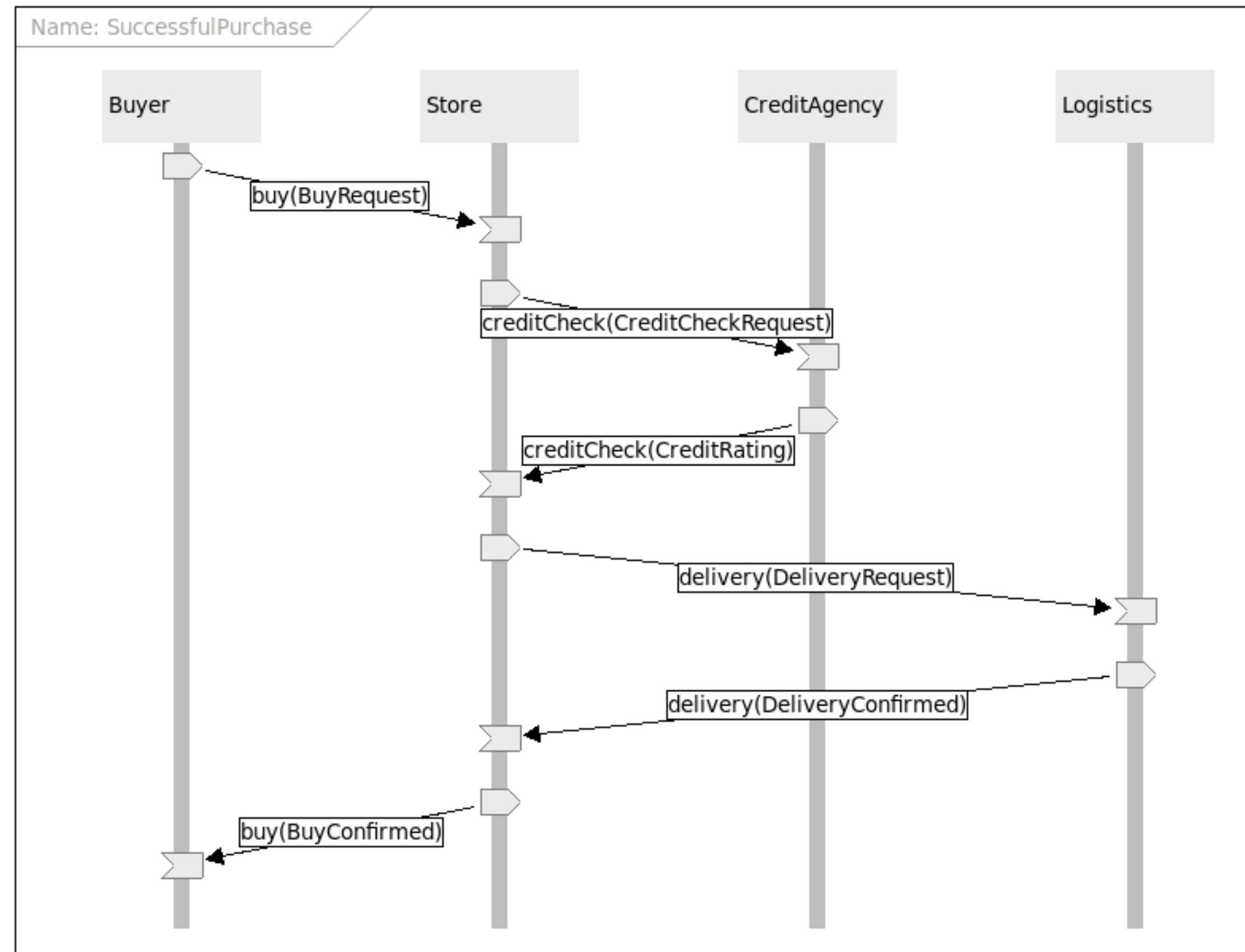
SUMMIT

**JBoss
WORLD**

PRESENTED BY RED HAT



Sequence Diagram



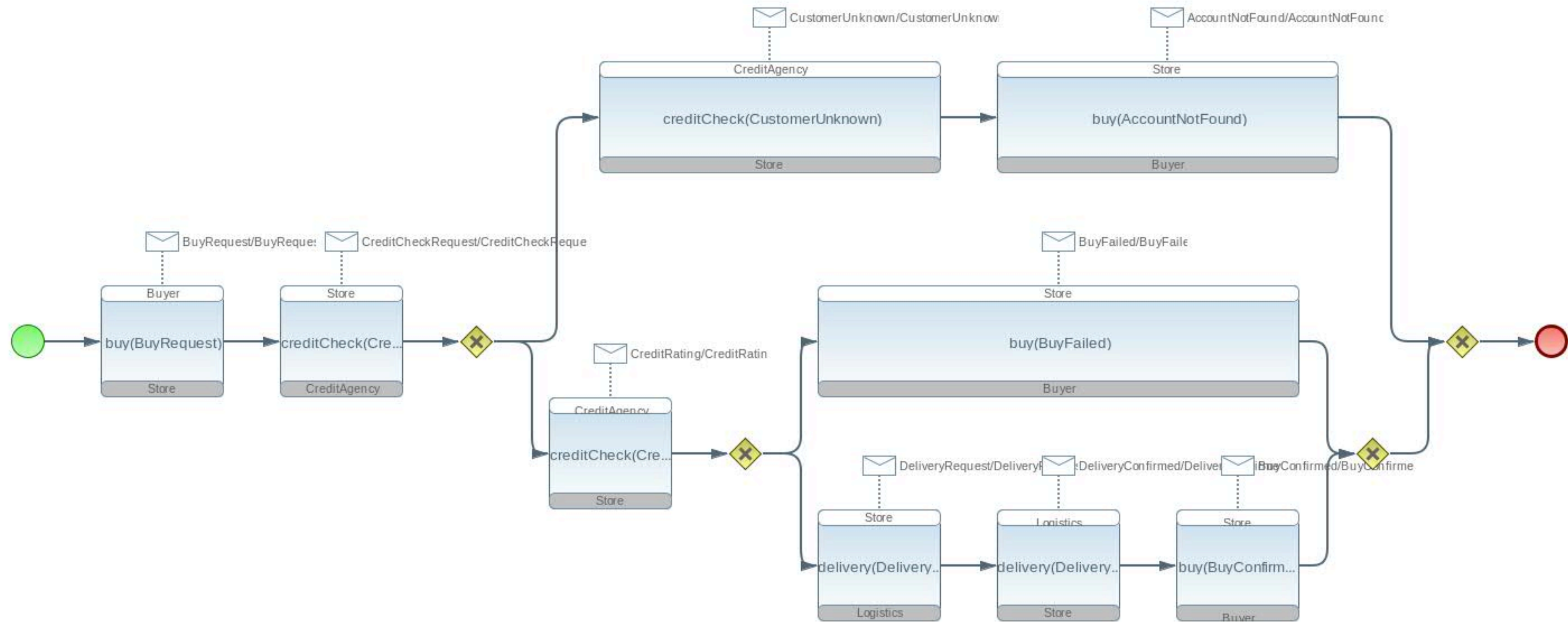
SUMMIT

**JBoss
WORLD**

PRESENTED BY RED HAT



Sequence Diagram



SUMMIT

JBoss
WORLD

PRESENTED BY RED HAT



Join Us!

- **Learn**

- <http://www.jboss.org/switchyard>

- **Play**

- 0.5 Beta is out!
- <http://github.com/jboss-switchyard/quickstarts>

- **Chat**

- [#switchyard](http://chat.freenode.net)

- **Fork**

- <http://github.com/jboss-switchyard>



SUMMIT

**JBoss
WORLD**

PRESENTED BY RED HAT



Questions?



SUMMIT

**JBoss
WORLD**

PRESENTED BY RED HAT



LIKE US ON FACEBOOK

www.facebook.com/redhatinc

FOLLOW US ON TWITTER

www.twitter.com/redhatsummit

TWEET ABOUT IT

#redhat

READ THE BLOG

summitblog.redhat.com

GIVE US FEEDBACK

www.redhat.com/summit/survey

SUMMIT

**JBoss
WORLD**

PRESENTED BY RED HAT

