

SUMMIT

**JBoss
WORLD**

PRESENTED BY RED HAT

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

www.theredhatsummit.com

ESB RELOADED PROJECT SWITCHYARD

Keith Babo
SwitchYard Project Lead
JBoss by Red Hat
May 4th, 2011

SUMMIT

JBoss
WORLD

PRESENTED BY RED HAT



Agenda

- Project Background
- What is an ESB?
 - Just kidding
 - Well, kind of
- How We Roll
- Demo
- Futures

SUMMIT

**JBoss
WORLD**

PRESENTED BY RED HAT



Me First!

- Keith Babo
 - JBoss Core Developer at Red Hat
 - Project Lead for SwitchYard
 - Spent the 00s at Sun Microsystems developing EAI/B2B/ESB products
 - Experience in the field and the dungeon

SUMMIT

**JBoss
WORLD**

PRESENTED BY RED HAT



Introducing SwitchYard



- New JBoss community project with the goal of creating our next generation Enterprise Service Bus
- What happened to JBoss ESB?
 - Same team
 - Active development continues in support of SOA-P
- Taking the next evolutionary step
 - Focus on consistent, intuitive user experience
 - Refactor core to eliminate known pain points
 - Leverage standards and complimentary technologies

SUMMIT

JBoss
WORLD

PRESENTED BY RED HAT



Introducing SwitchYard

- Why a separate project?
 - Isolate disruptive changes
 - Focus community
 - Implement faster, get feedback sooner
- Project goals aligned
 - End deliverable is a better SOA Platform

SUMMIT

**JBoss
WORLD**

PRESENTED BY RED HAT



Activity

- SwitchYard
 - Milestone 1 in February, 2011
 - 0.1 release end of May
 - 0.*n* releases every 8-10 weeks until 1.0
- JBoss ESB
 - 4.9 in August 2010
 - SOA Platform 5.1 in March 2011
 - 4.10 in progress

SUMMIT

JBoss
WORLD

PRESENTED BY RED HAT



The Birth of the ESB

Business Domain

Integration Domain

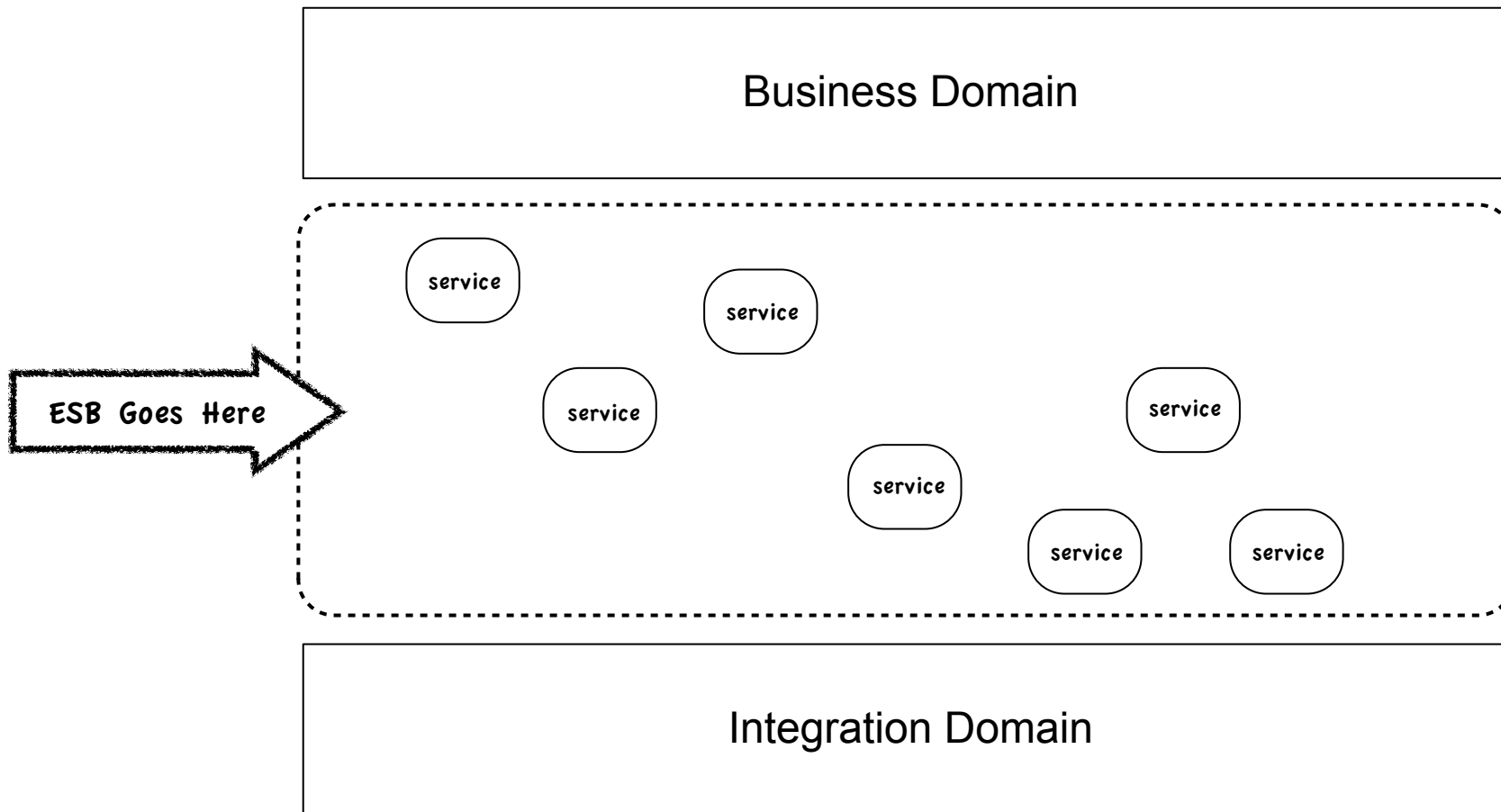
SUMMIT

**JBoss
WORLD**

PRESENTED BY RED HAT



The Birth of the ESB



SUMMIT

**JBoss
WORLD**

PRESENTED BY RED HAT



A Superficial Guide to Service-Oriented Principles

Service



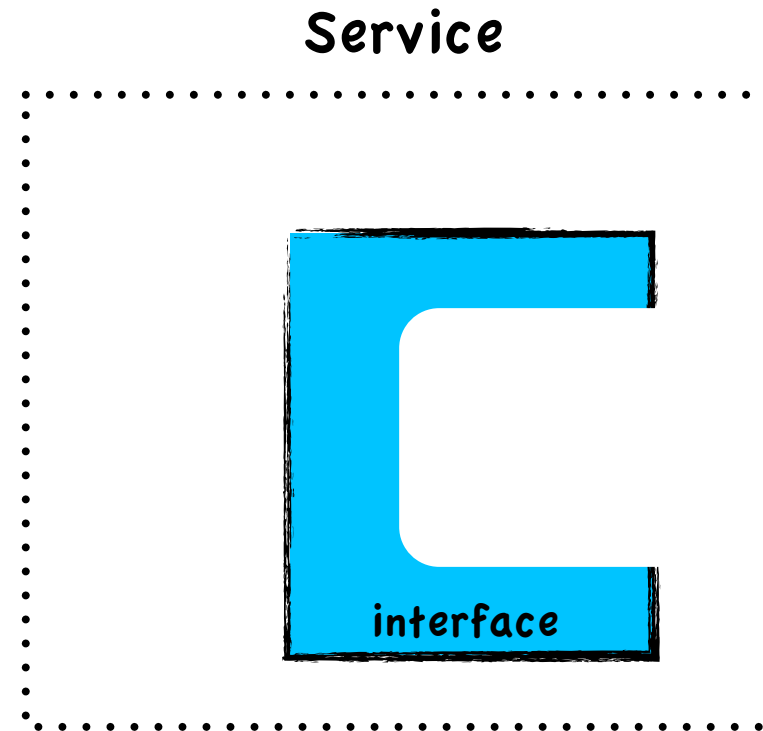
SUMMIT

**JBoss
WORLD**

PRESENTED BY RED HAT



A Superficial Guide to Service-Oriented Principles



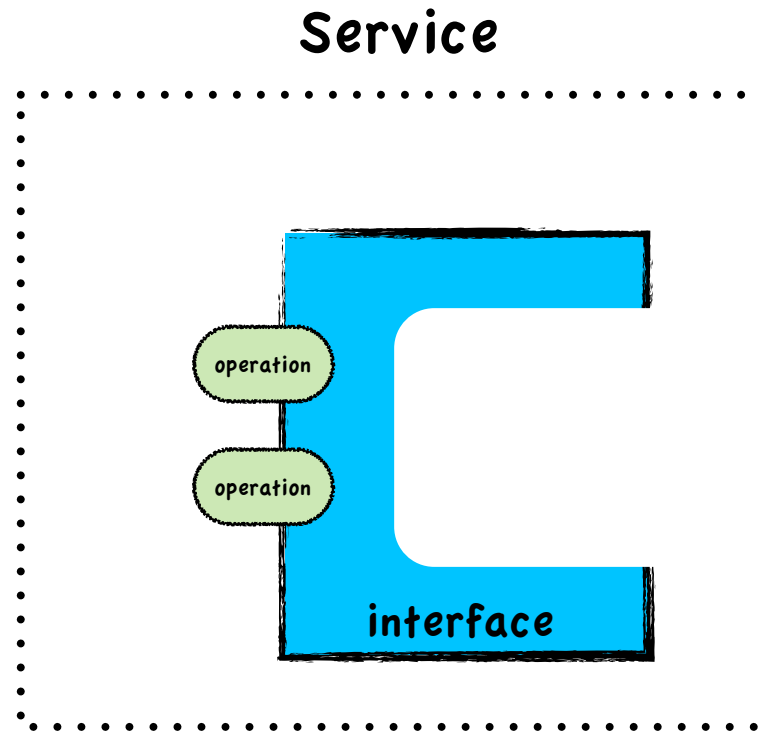
SUMMIT

**JBoss
WORLD**

PRESENTED BY RED HAT



A Superficial Guide to Service-Oriented Principles



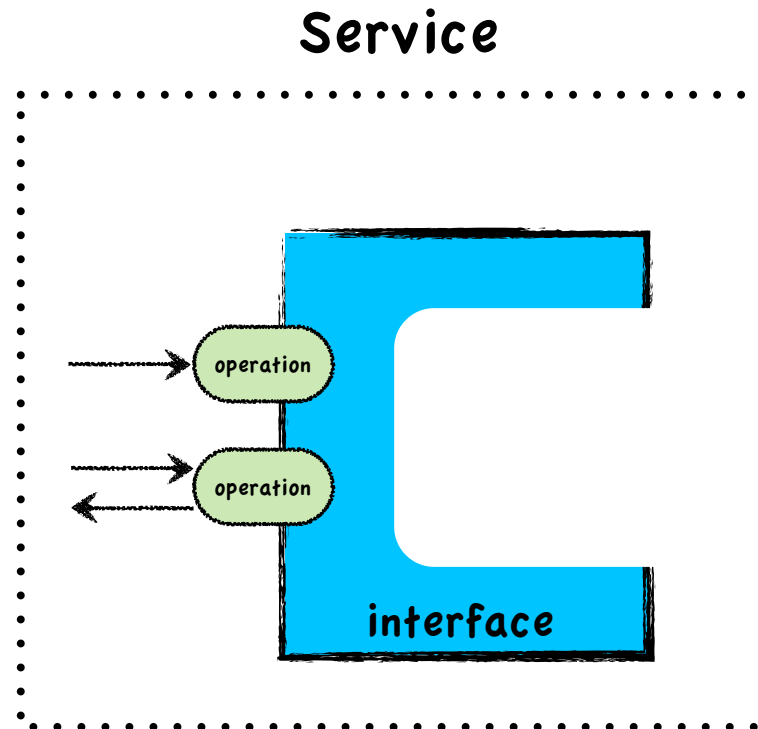
SUMMIT

**JBoss
WORLD**

PRESENTED BY RED HAT



A Superficial Guide to Service-Oriented Principles



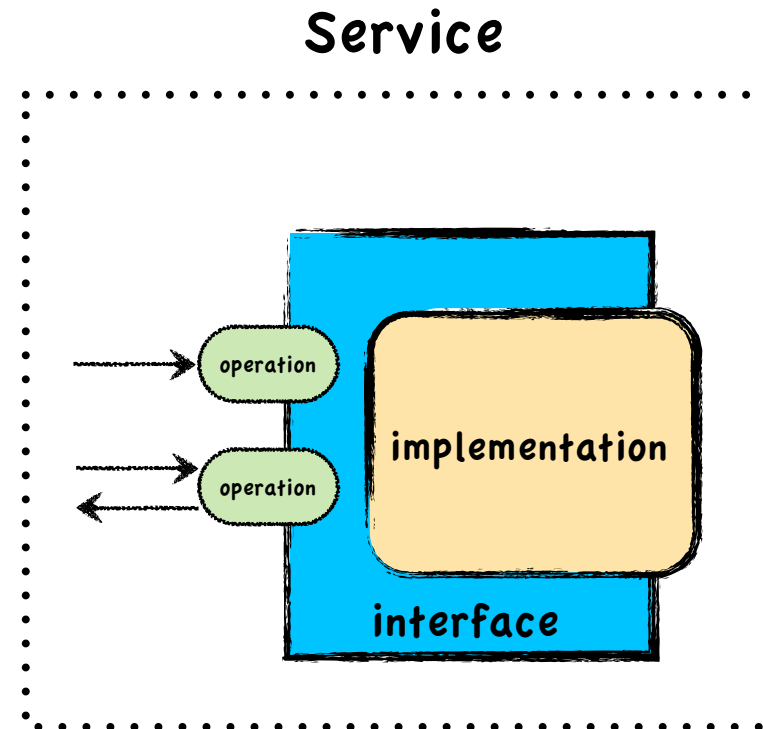
SUMMIT

JBoss
WORLD

PRESENTED BY RED HAT



A Superficial Guide to Service-Oriented Principles



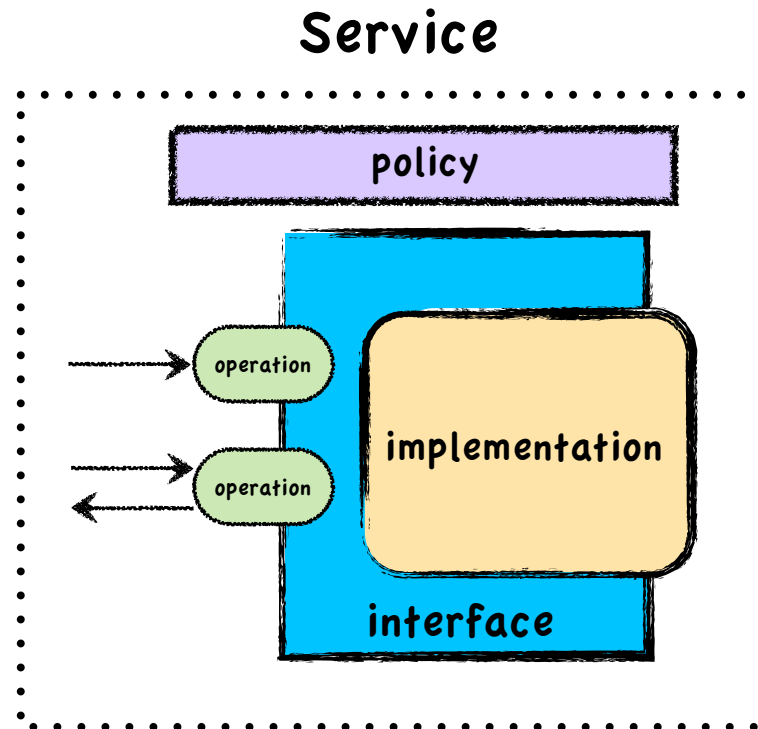
SUMMIT

**JBoss
WORLD**

PRESENTED BY RED HAT



A Superficial Guide to Service-Oriented Principles



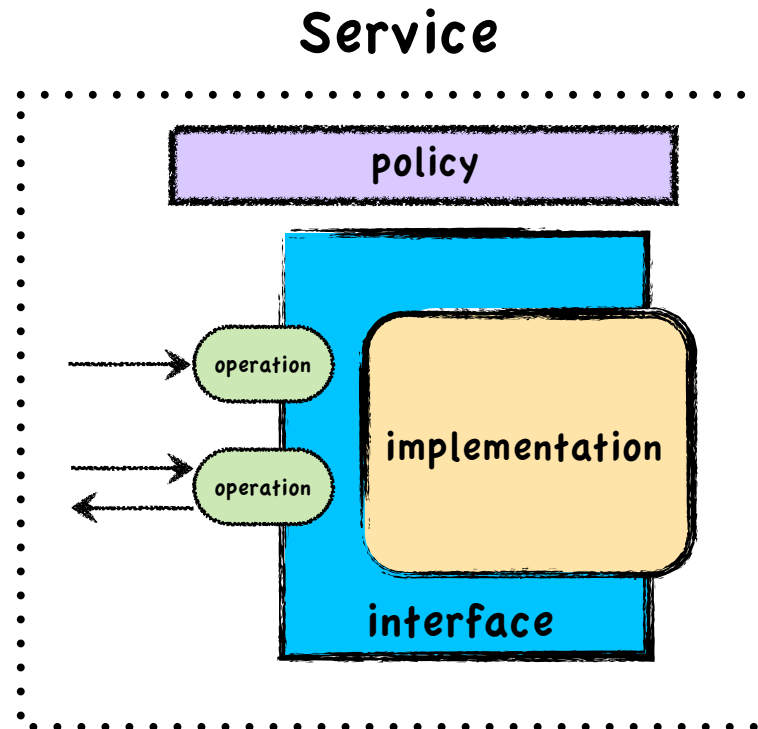
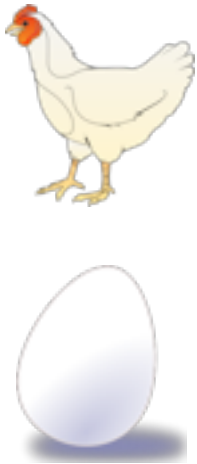
SUMMIT

JBoss
WORLD

PRESENTED BY RED HAT



A Superficial Guide to Service-Oriented Principles



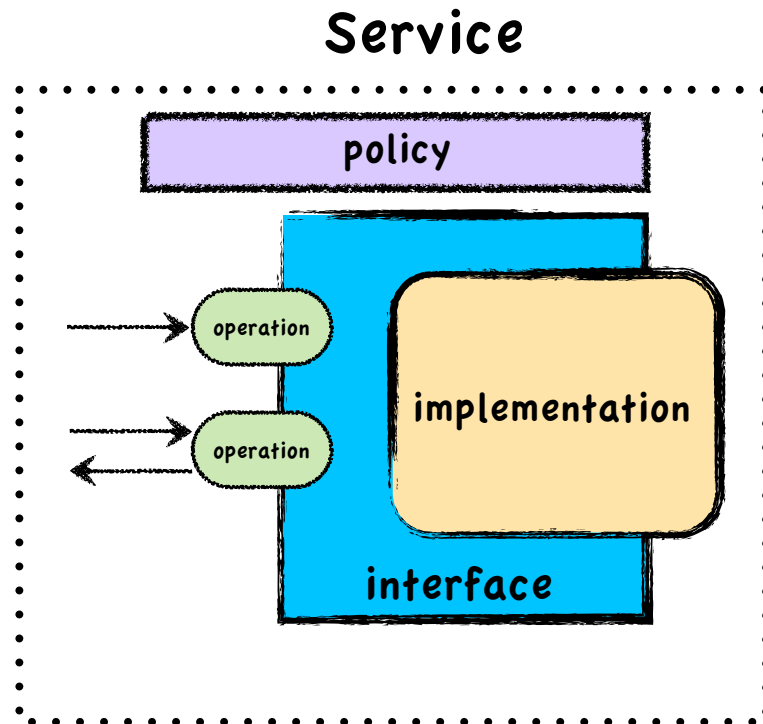
SUMMIT

**JBoss
WORLD**

PRESENTED BY RED HAT



A Superficial Guide to Service-Oriented Principles



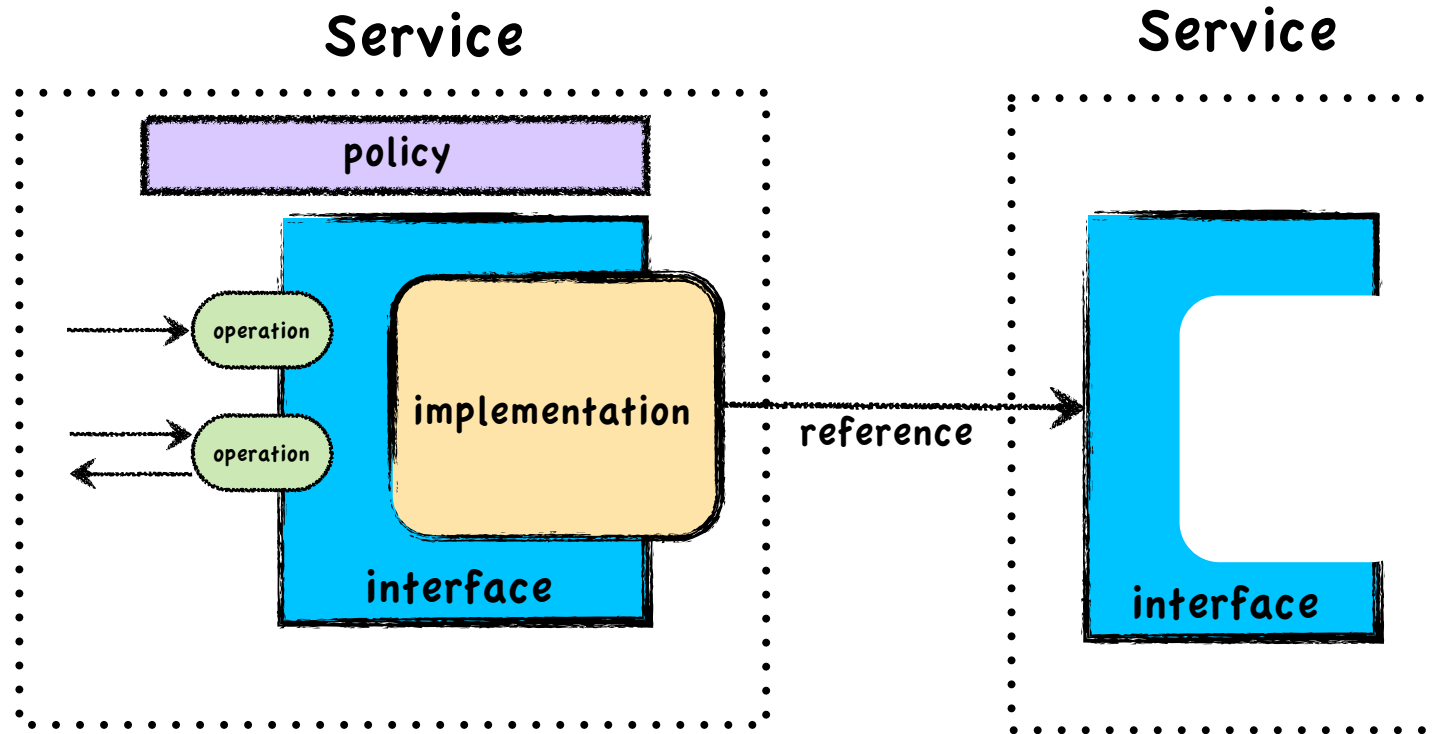
SUMMIT

JBoss
WORLD

PRESENTED BY RED HAT



A Superficial Guide to Service-Oriented Principles



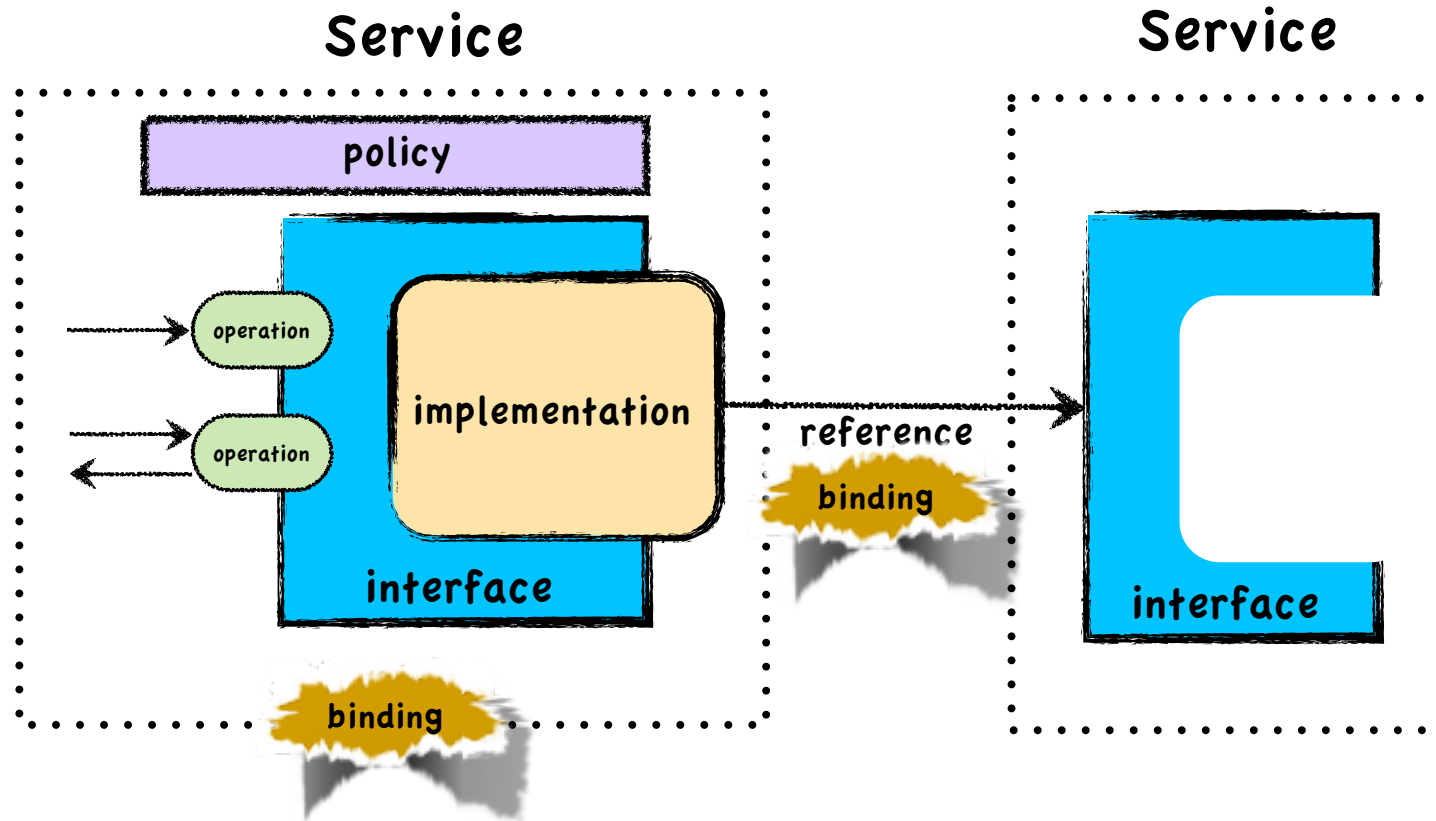
SUMMIT

JBoss
WORLD

PRESENTED BY RED HAT



A Superficial Guide to Service-Oriented Principles



SUMMIT

JBoss
WORLD

PRESENTED BY RED HAT





SUMMIT

**JBoss
WORLD**

PRESENTED BY RED HAT



Service Interface

```
public interface OrderService {  
    OrderAck submitOrder(Order order);  
}
```

SUMMIT

JBoss
WORLD

PRESENTED BY RED HAT



Service Implementation

```
public class OrderServiceBean implements OrderService {
    @Override
    public OrderAck submitOrder(Order order) {
        // Create an order ack
        return new OrderAck()
            .setOrderId(order.getOrderId())
            .setAccepted(true)
            .setStatus("Processing Order");
    }
}
```

SUMMIT

JBoss
WORLD

PRESENTED BY RED HAT



Service Implementation

```
@Service(OrderService.class)
```

```
public class OrderServiceBean implements OrderService {  
    @Override  
    public OrderAck submitOrder(Order order) {  
        // Create an order ack  
        return new OrderAck()  
            .setOrderId(order.getOrderId())  
            .setAccepted(true)  
            .setStatus("Processing Order");  
    }  
}
```

SUMMIT

JBoss
WORLD

PRESENTED BY RED HAT



Service Reference

```
@Service(OrderService.class)
public class OrderServiceBean implements OrderService {

    @Inject @Reference
    private InventoryService _inventory;

    @Override
    public OrderAck submitOrder(Order order) {
        // Check the inventory
        Item orderItem = _inventory.lookupItem(order.getItemId());

        // Create an order ack
        ...
    }
}
```

SUMMIT

JBoss
WORLD

PRESENTED BY RED HAT



Service Binding

```
$ soap-binding bind-service  
--serviceName OrderService  
--wsdl wsdl/feedbackService.wsdl
```

SUMMIT

JBoss
WORLD

PRESENTED BY RED HAT



Message Transformation

```
public class OrderAckTransform
    extends BaseTransformer<OrderAck, Element> {

    public OrderAckTransform() {
        super(new QName("urn:examples:order:1.0", "submitOrderResponse"),
            new QName("java:org.examples.order.Order"));
    }

    @Override
    public Element transform(OrderAck orderAck) {
        // Transformation code goes here
    }
}
```

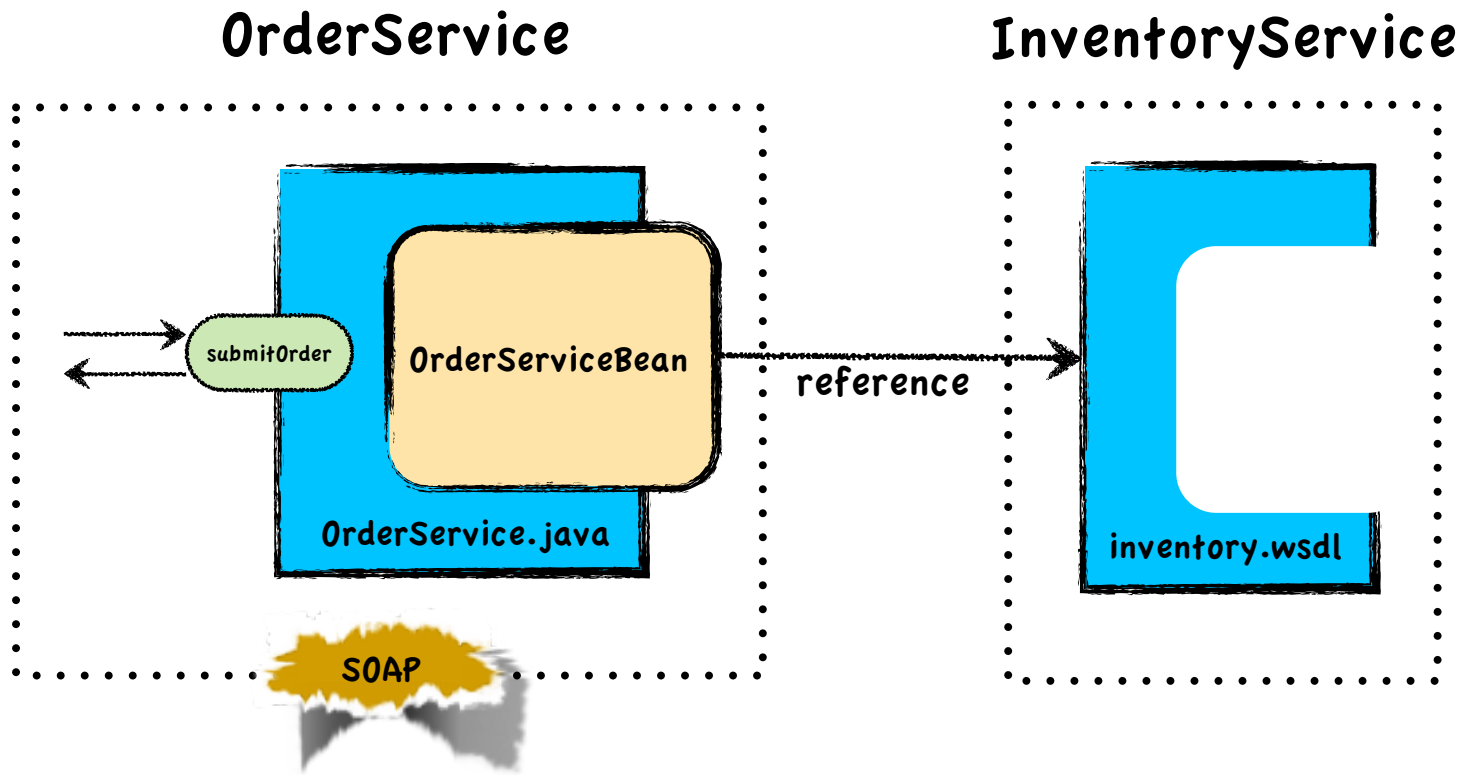
SUMMIT

**JBoss
WORLD**

PRESENTED BY RED HAT



Voilà!



SUMMIT

JBoss
WORLD

PRESENTED BY RED HAT





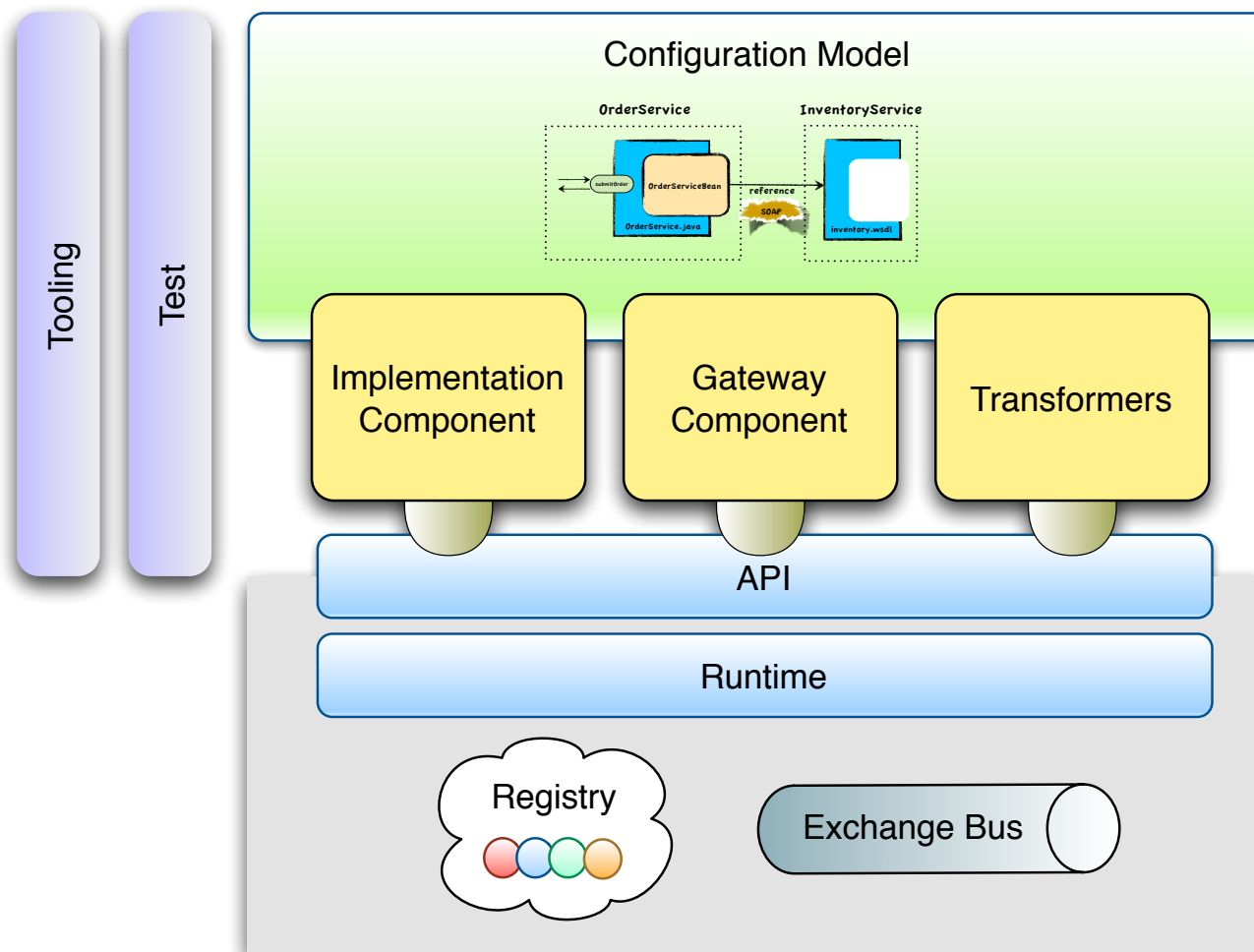
SUMMIT

**JBoss
WORLD**

PRESENTED BY RED HAT



What's In SwitchYard



SUMMIT

JBoss
WORLD

PRESENTED BY RED HAT



SwitchYard Core

- API
 - Service registration
 - Service metadata
 - Message exchange
 - Transformers
 - Service Domain
- Runtime
 - Local exchange bus
 - Local registry
 - Deploy anywhere (AS6, AS7, embedded)

SUMMIT

**JBoss
WORLD**

PRESENTED BY RED HAT



Transformation

- Ubiquitous challenge in application integration and SOA
- Three flavors of transformation
 - Change in data representation
 - Conversion
 - Change in data format
 - Translation
 - Change in data itself
 - Transformation as a service

SUMMIT

**JBoss
WORLD**

PRESENTED BY RED HAT



Transformation of Representation

- Change in representation
- Representation = Java type
- Transformation is simply a type conversion
- No semantic knowledge required

java.lang.String

```
<order>
  <item>XYZ123</item>
  <quantity>5</quantity>
</order>
```

=

org.w3c.dom.Node

```
<order>
  <item>XYZ123</item>
  <quantity>5</quantity>
</order>
```

=

java.io.InputStream

```
<order>
  <item>XYZ123</item>
  <quantity>5</quantity>
</order>
```

SUMMIT

**JBoss
WORLD**

PRESENTED BY RED HAT



Transformation of Data Format

- Requires semantic understanding of data types
- Machines cannot do this on their own
 - yet ...?

```
HDR*1*0*59.97*64.92*4.95*Wed Nov 15 13:45:28 EST 2006
CUS*user1*Harry*Fletcher*SD
ORD*1*1*364*The 40-Year-Old Virgin*29.98
ORD*2*1*299*Pulp Fiction*29.99
```



```
public class Order {
    private Header header;
    private List<OrderItem> orderItems;
}

public class Header {
    private String orderId;
    private long orderStatus;
    private BigDecimal netAmount;
    private BigDecimal totalAmount;
    private BigDecimal tax;
    private Date date;
    private Customer customer;
}

public class OrderItem {
    private int quantity;
    private String productId;
    private BigDecimal price;
    private String title;
}
```

SUMMIT

**JBoss
WORLD**

PRESENTED BY RED HAT



Transformers

- Transformation is wired into SwitchYard core
 - Types declared via service contract
 - Transformer resolved dynamically at runtime
- Bring on the canonical data models
- Current Transformers
 - Java, Smooks
- In the works
 - XSLT, JSON, GPB

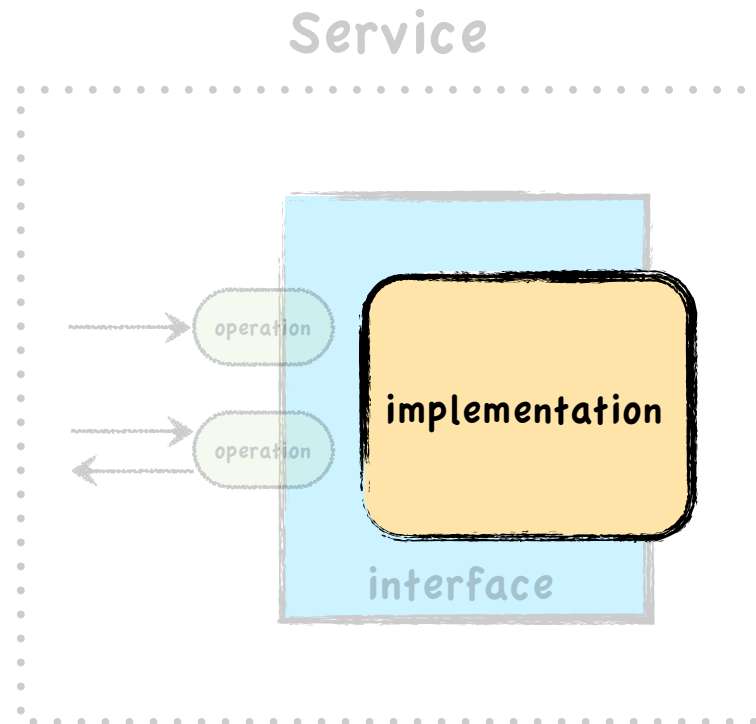
SUMMIT

**JBoss
WORLD**

PRESENTED BY RED HAT



Implementation Components



SUMMIT

**JBoss
WORLD**

PRESENTED BY RED HAT



Bean Component

- POJO = Service ... 'nuff said
- Easy to use
 - Annotation-based
 - Config auto-generated
 - Service auto-registered
- Consistent with core principles
 - Services declare a service interface
 - References injected based on service interface
 - Dependencies are explicit

SUMMIT

JBoss
WORLD

PRESENTED BY RED HAT



Bean Component

- Implemented as a CDI Extension
 - Standard programming model (JSR 299)
 - "The *theme* of CDI is *loose-coupling with strong typing*."
- Weld provides the implementation framework
 - Less work for us
 - More features for users
- Straightforward integration into the web tier

SUMMIT

JBoss
WORLD

PRESENTED BY RED HAT



Camel Component

- Integrates Apache Camel with SwitchYard
- Camel provides
 - Routing engine and language(s)
 - Loads of EIP
 - Cornucopia of components
- Camel as a service
 - Routes provide pipeline orchestration
 - Service interface
 - Service references resolved independent of binding

SUMMIT

JBoss
WORLD

PRESENTED BY RED HAT



An Example Camel Route

```
<route>
  <from uri="file://orders/in"/>
  <log message="Order Received : ${body}"/>
  <to uri="OrderValidator"/>
  <filter>
    <xpath>/order[@priority='high']</xpath>
    <to uri="file://shipping/in"/>
  </filter>
</route>
```

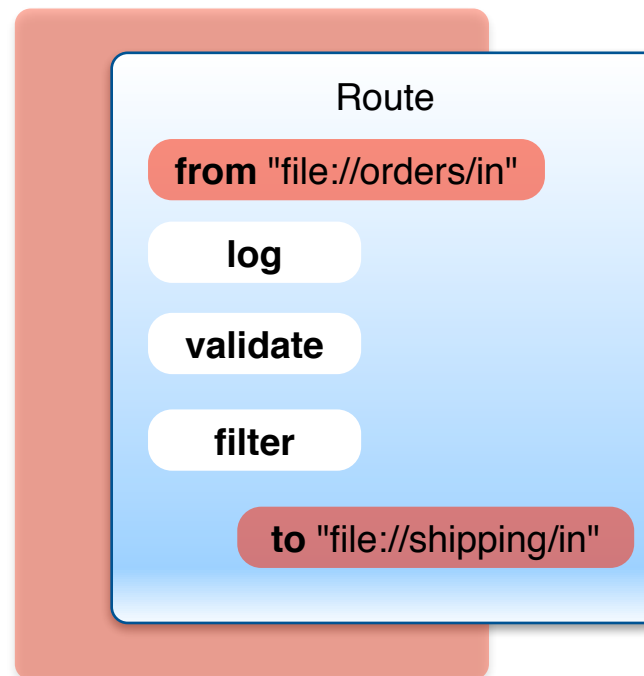
SUMMIT

JBoss
WORLD

PRESENTED BY RED HAT



An Example Camel Route



SUMMIT

**JBoss
WORLD**

PRESENTED BY RED HAT



Camel In SwitchYard

```
<sca:component name="CamelComponent">
```

```
  <sca:service name="OrderService" >  
    <sca:interface.java interface="org.example.OrderService"/>  
  </sca:service>
```

```
  <sca:reference name="ShippingService">  
    <sca:interface.java interface="org.example.ShippingService"/>  
  </sca:reference>
```

```
  <implementation.camel>  
    <route>  
      <log message="Order Received : ${body}"/>  
      <to uri="OrderValidator"/>  
      <filter>  
        <xpath>/order[@priority='high']</xpath>  
        <to uri="switchyard://ShippingService?operationName=shipOrder"/>  
      </filter>  
    </route>  
  </implementation.camel>
```

```
</sca:component>
```

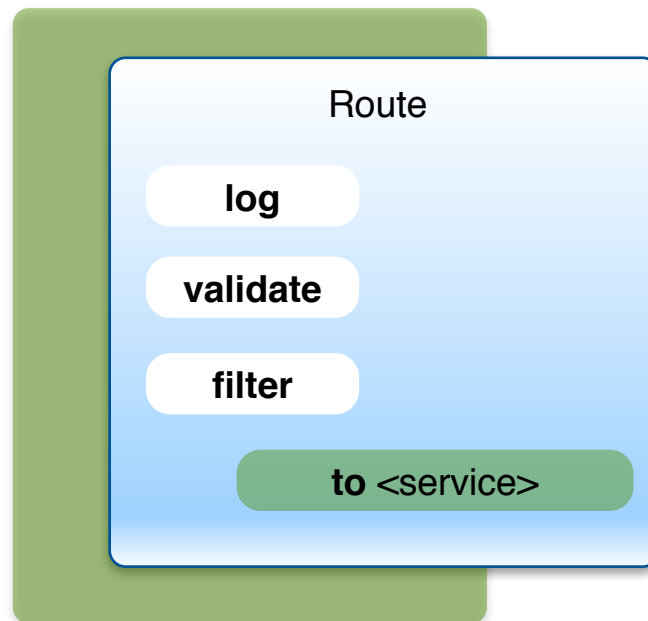
SUMMIT

JBoss
WORLD

PRESENTED BY RED HAT



Camel In SwitchYard



SUMMIT

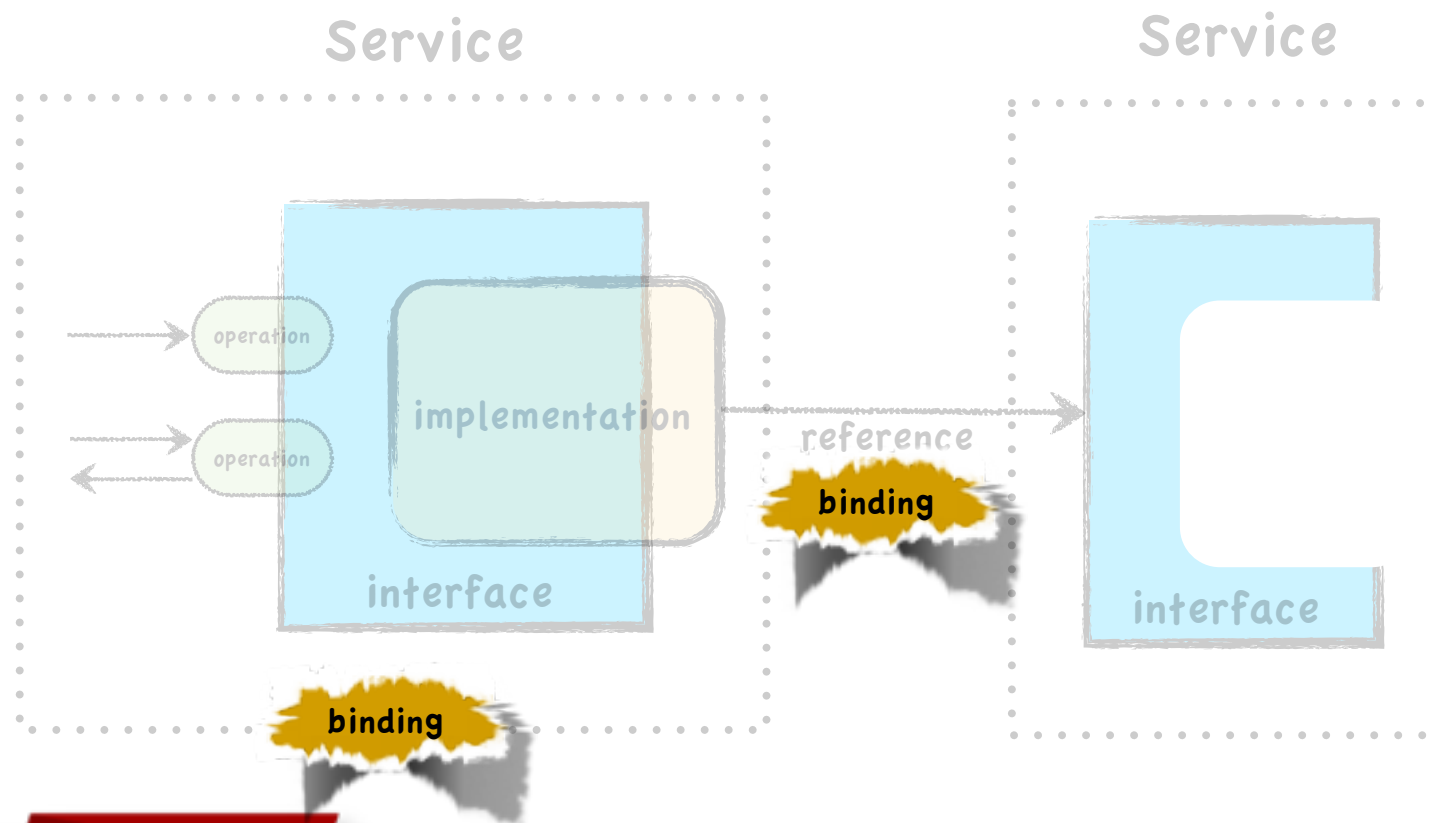
**JBoss
WORLD**

PRESENTED BY RED HAT



Gateway Components

- Provide protocol binding for services and references
- Gateway components are bi-directional



SUMMIT

**JBoss
WORLD**

PRESENTED BY RED HAT



Gateway Components

- Two fundamental rules of gateways
 - Never have enough
 - The ones you have don't do enough
- Our approach
 - Focus on key gateways for platform
 - SOAP first
 - Incorporate adapters from other communities
 - Camel components
 - Straightforward pluggability for rolling your own
 - Tooling, configuration, deployment

SUMMIT

**JBoss
WORLD**

PRESENTED BY RED HAT



Camel Gateway

- Allows Camel components to be used as gateways
- Flexible schema allows for XML or URI endpoint configuration

XML-Based

```
<camel:binding.file>
  <camel:operationSelector operationName="print" />
  <camel:consume>
    <camel:inputDir>/tmp/in</camel:inputDir>
    <camel:autoCreate>true</camel:autoCreate>
    <camel:initialDelay>10</camel:initialDelay>
    <camel:delete>true</camel:delete>
  </camel:consume>
</camel:binding.file>
```

URI-Based

```
<camel:binding.camel
  configURI="file:///tmp/in?autoCreate=true&initialDelay=10&delete=true">
  <camel:operationSelector operationName="print"/>
</camel:binding.camel>
```

SUMMIT

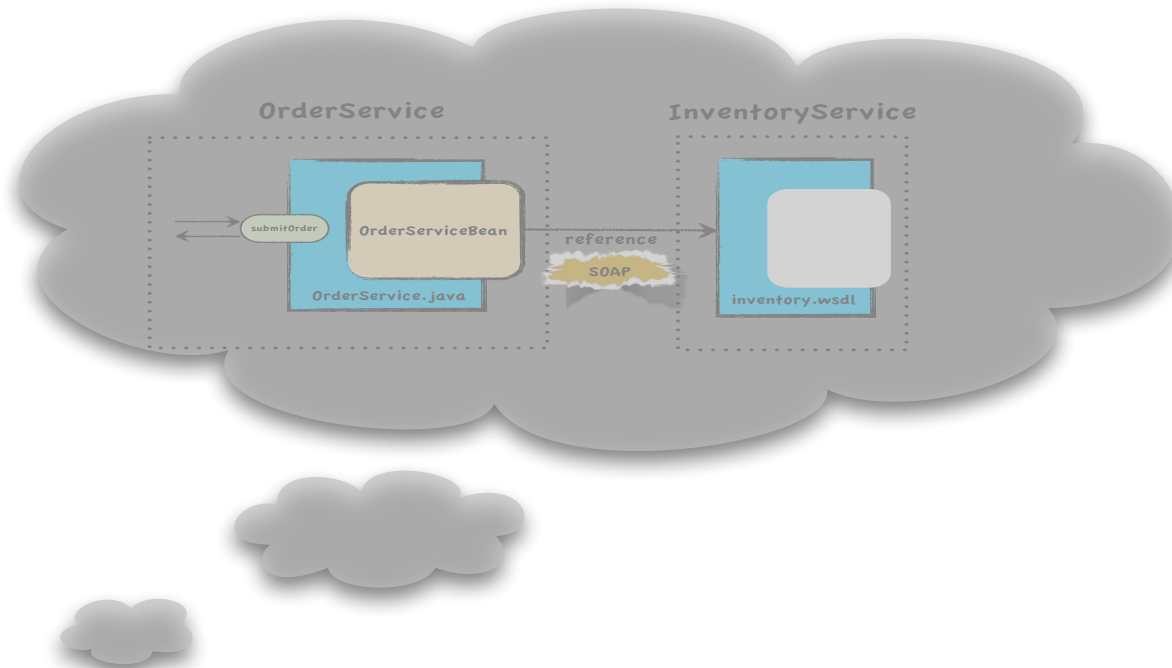
JBoss
WORLD

PRESENTED BY RED HAT



Configuration

- We need a way to represent this



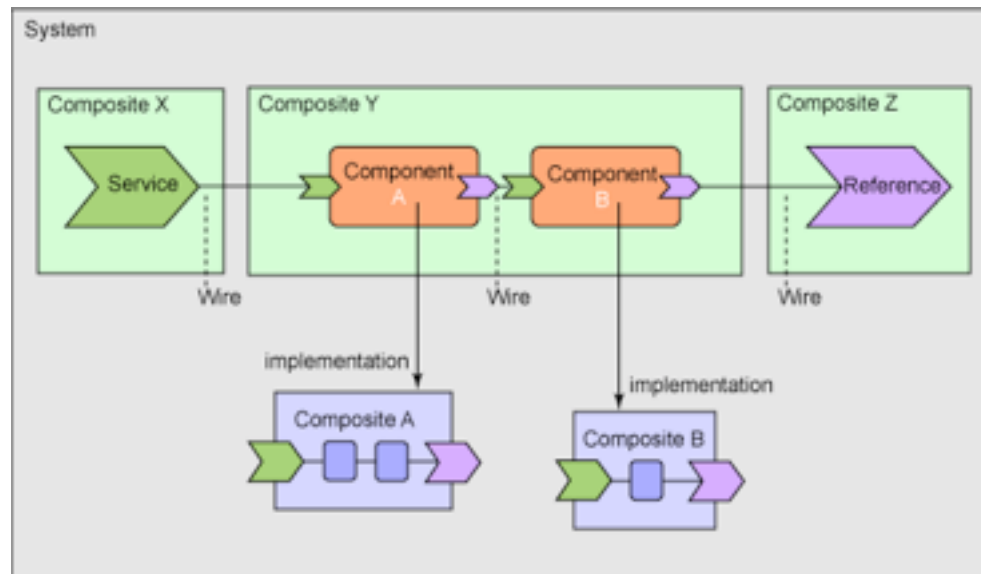
SUMMIT

**JBoss
WORLD**

PRESENTED BY RED HAT



Configuration



SUMMIT

**JBoss
WORLD**

PRESENTED BY RED HAT



SCA - Service Component Architecture

- Set of specifications for building applications in a manner consistent with SOA principles
- Assembly spec is none too shabby
 - Service definition language
 - Encapsulation model
- Better than defining our own configuration format?
 - Skills portability is nice
 - Runtime portability much less certain

SUMMIT

JBoss
WORLD

PRESENTED BY RED HAT



Testing

- Big Bang testing of SOA applications must stop!
- Develop and test your services iteratively
 - Service, transformation, binding, etc.
- SwitchYardTestCase
 - Bootstraps runtime, components, and application
- MixIns
 - Enriches test case via composition vs. extension
 - CDI, HTTP, Smooks
- Arquillian coming soon

SUMMIT

**JBoss
WORLD**

PRESENTED BY RED HAT



Maven Support

- Started with Maven, staying with Maven
- Modularity
 - Project structure
 - Artifacts
- Extends to SwitchYard applications
 - Archetype
 - Plugins

SUMMIT

**JBoss
WORLD**

PRESENTED BY RED HAT



Development Tooling

- Focus on intuitive user experience with quick onramp
- Seam Forge
 - Rapid application development tool
 - Ease of a wizard, power of a shell
 - More coming in the demo ...
- IDEs
 - Maven support provides baseline functionality across IDEs
 - Graphical tooling in Eclipse/JBDS

SUMMIT

JBoss
WORLD

PRESENTED BY RED HAT



Demo



SUMMIT

**JBoss
WORLD**

PRESENTED BY RED HAT



Looking Forward

- 0.1 -- 05/31
 - Resolve ankle-biter issues
 - Documentation
 - Release drudgery
- 0.2 -- 08/05
 - Policy
 - BPM
- Further release details available in JIRA

SUMMIT

**JBoss
WORLD**

PRESENTED BY RED HAT



Resistance is Futile - Join Us!

- Project Site
 - <http://www.jboss.org/switchyard>
- Community and Forums
 - <http://community.jboss.org/en/switchyard>
- IRC
 - freenode : #switchyard
- JIRA
 - <https://issues.jboss.org/browse/SWITCHYARD>
- GitHub
 - <https://github.com/jboss-switchyard>

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

