RED HAT SUMMIT

LEARN. NETWORK. EXPERIENCE OPEN SOURCE.

June 11-14, 2013
Boston, MA

redhat

JBoss SOA Platform 6 Quick Start

Kevin Conner
SOA Platform Architect, Red Hat
Keith Babo
SwitchYard Project Lead, Red Hat

June 13th, 2013

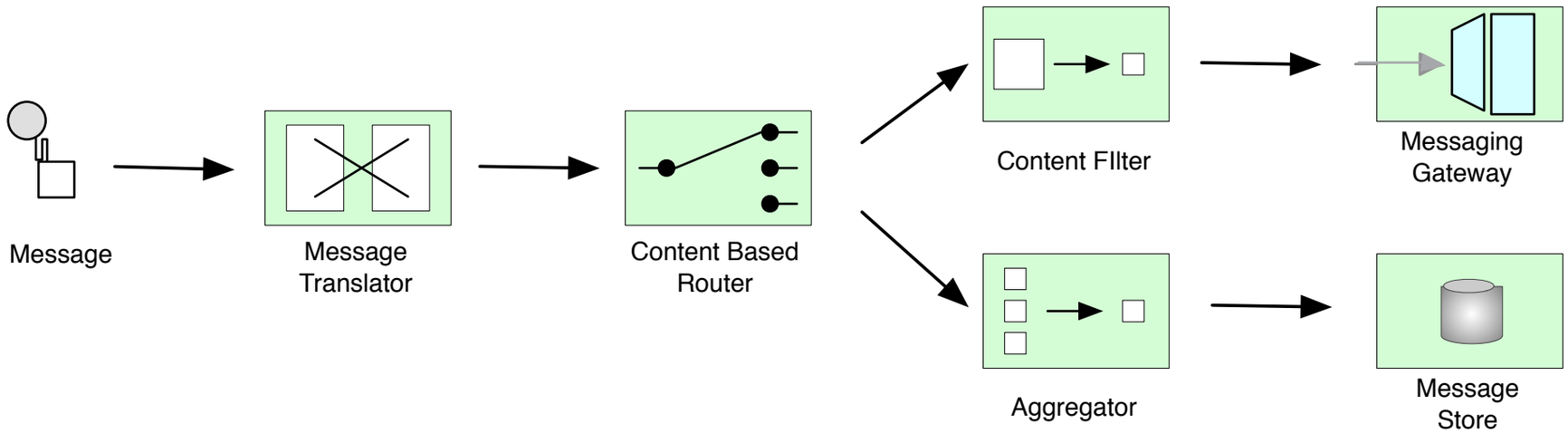
There is Still Time To Leave

- Technical cruise through SOA Platform 6
- Important high-level details
- Technical meat
- Live demonstration

SOA 6 Platform Themes

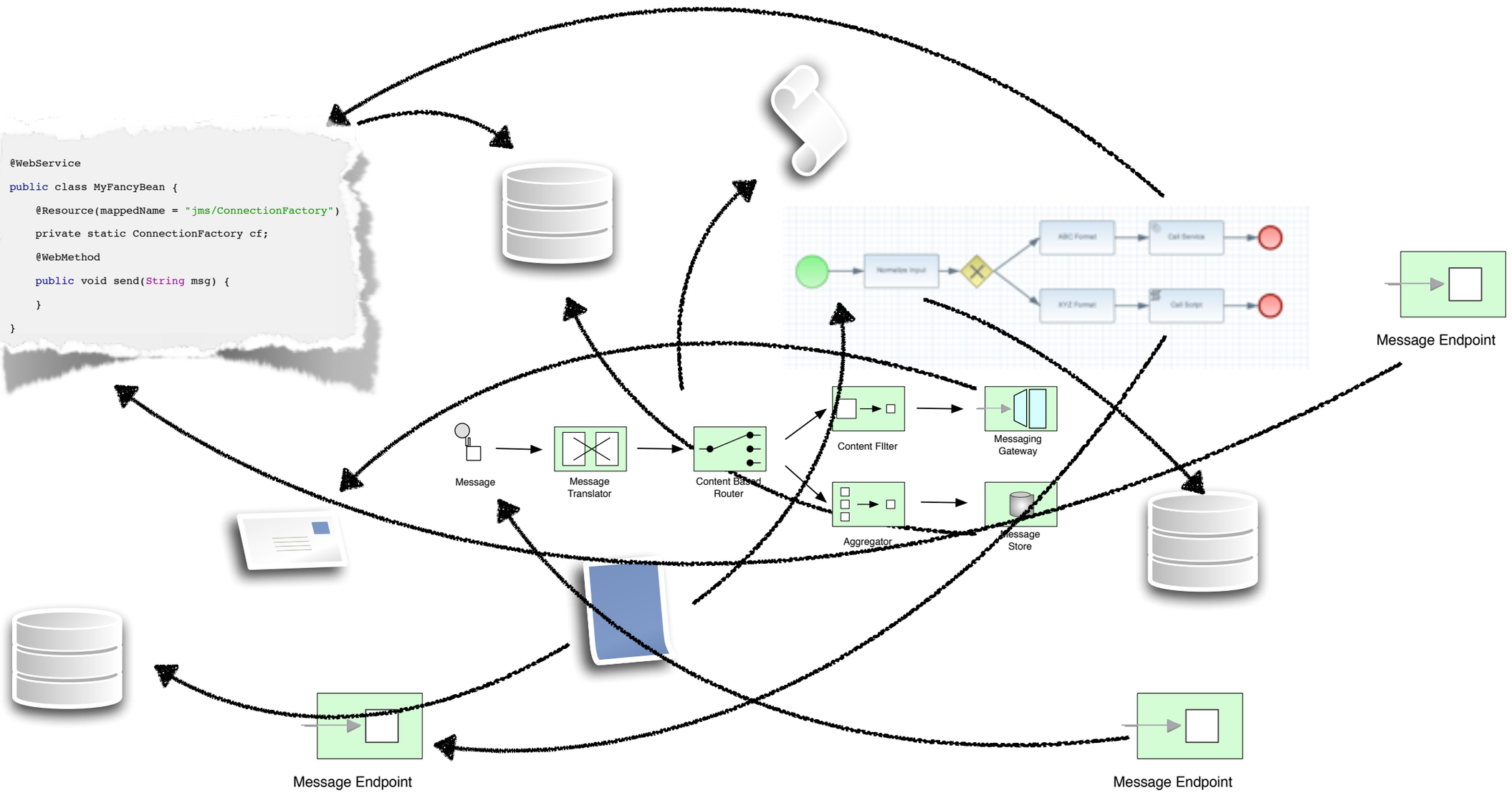
- Structured, service-oriented development model
- Ease of use
- Technology refresh
- Governance
- Functional equivalence with SOA 5

With Great Power

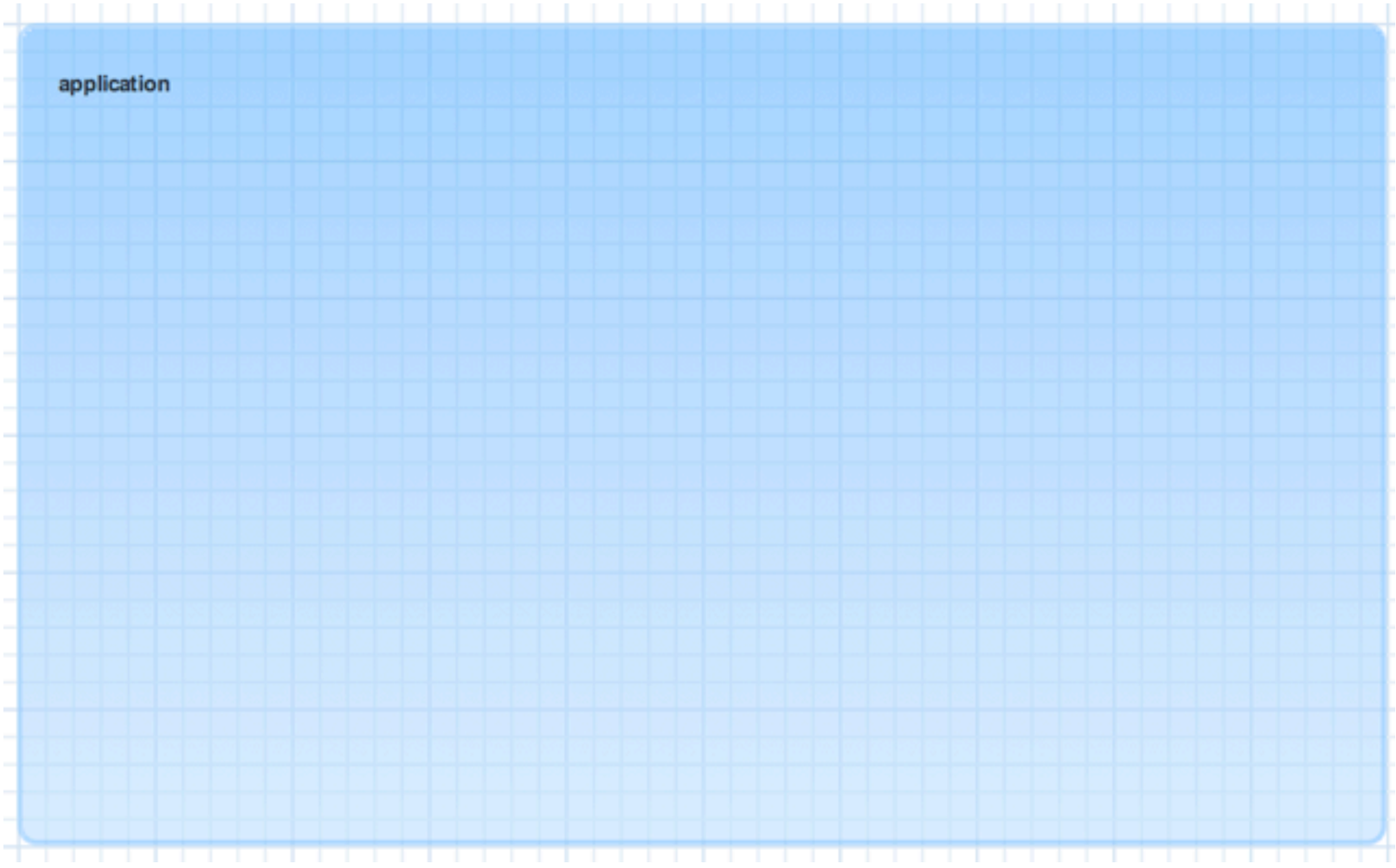


Comes Great Responsibility

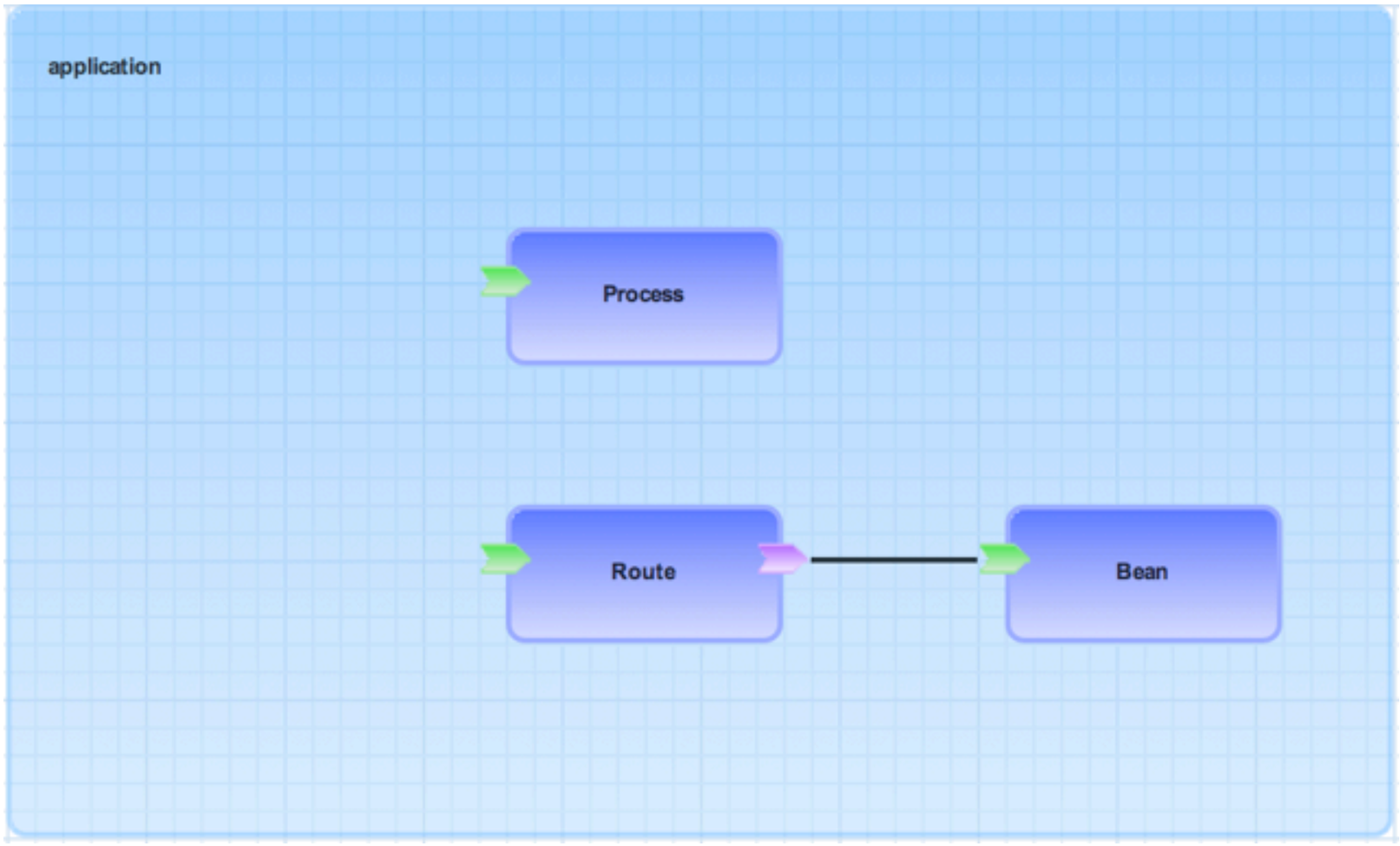
```
@WebService
public class MyFancyBean {
    @Resource(mappedName = "jms/ConnectionFactory")
    private static ConnectionFactory cf;
    @WebMethod
    public void send(String msg) {
    }
}
```



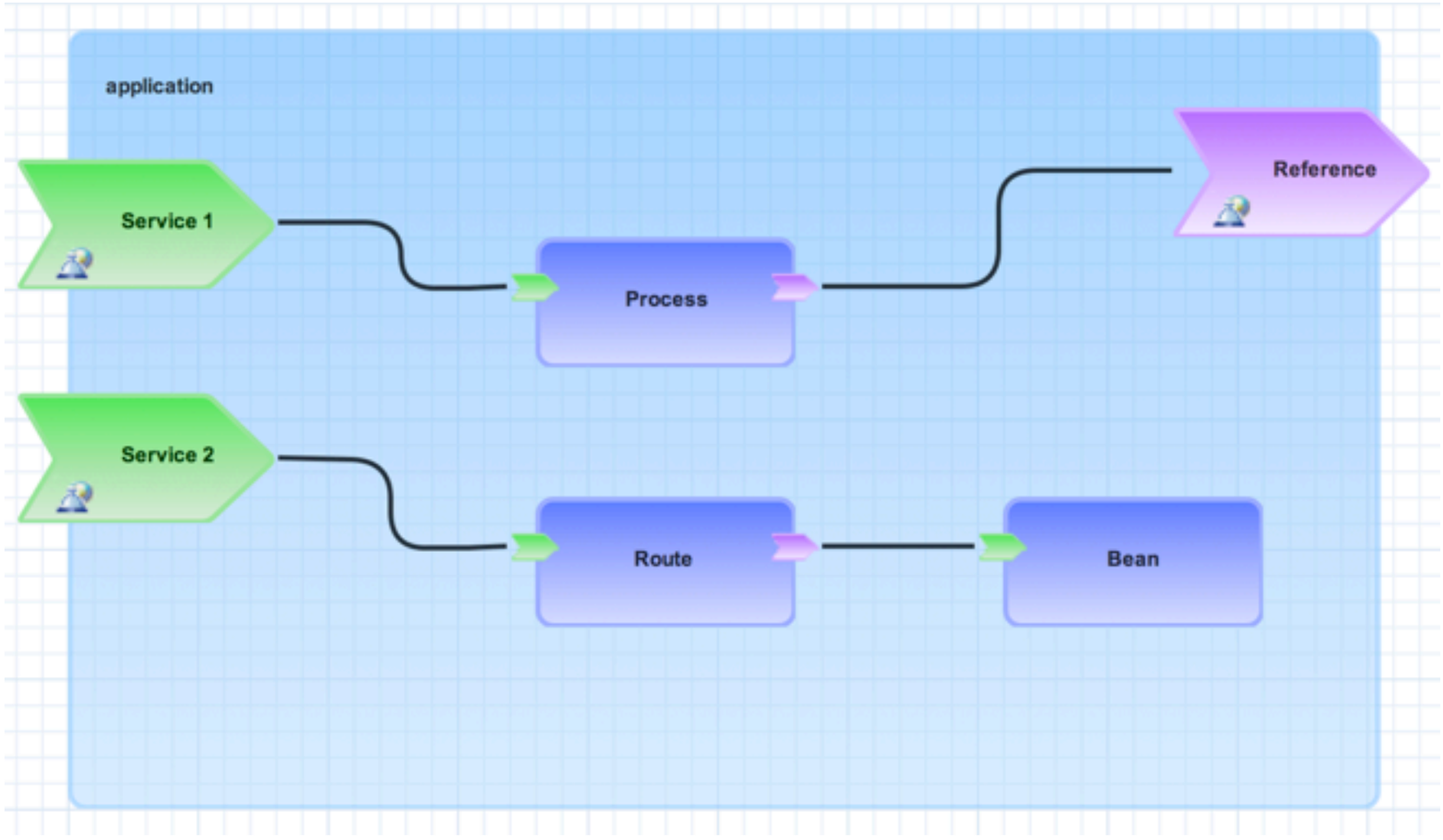
Structured Development



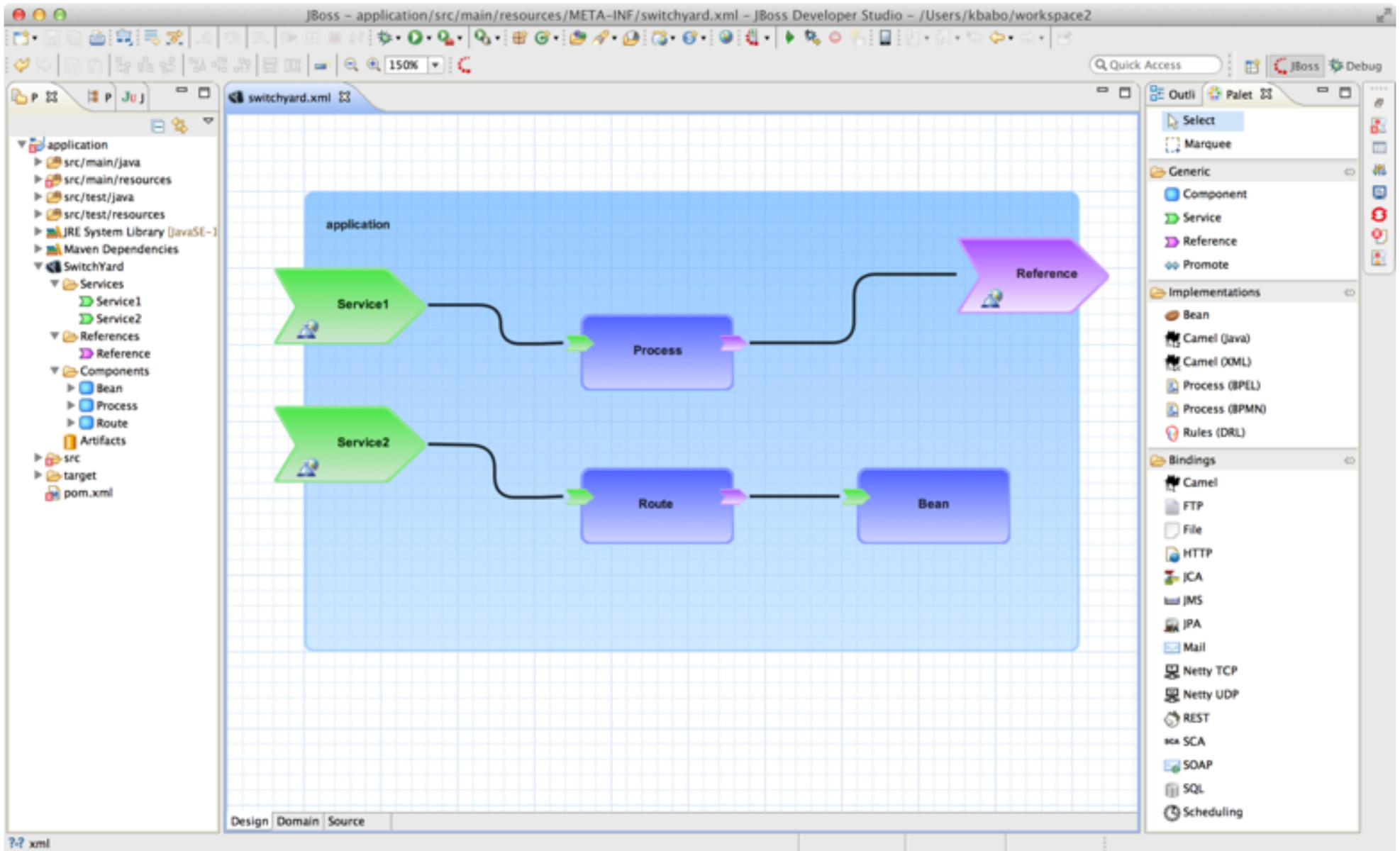
Structured Development



Structured Development



Ease of Use


JBoss - application/src/main/resources/META-INF/switchyard.xml - JBoss Developer Studio - /Users/kbabo/workspace2
Quick Access
JBoss
Debug
150%
switchyard.xml
application
src/main/java
src/main/resources
src/test/java
src/test/resources
JRE System Library [JavaSE-1
Maven Dependencies
SwitchYard
Services
Service1
Service2
References
Reference
Components
Bean
Process
Route
Artifacts
src
target
pom.xml
application
Service1
Service2
Process
Route
Reference
Bean
Design
Domain
Source
?-? xml
Outli
Palet
Select
Marquee
Generic
Component
Service
Reference
Promote
Implementations
Bean
Camel (Java)
Camel (XML)
Process (BPEL)
Process (BPMN)
Rules (DRL)
Bindings
Camel
FTP
File
HTTP
JCA
JMS
JPA
Mail
Netty TCP
Netty UDP
REST
SCA
SOAP
SQL
Scheduling

Technology Refresh

- EAP 6
- Java EE 6 / CDI
- Camel
- Maven
- OASIS (SCA, S-RAMP)

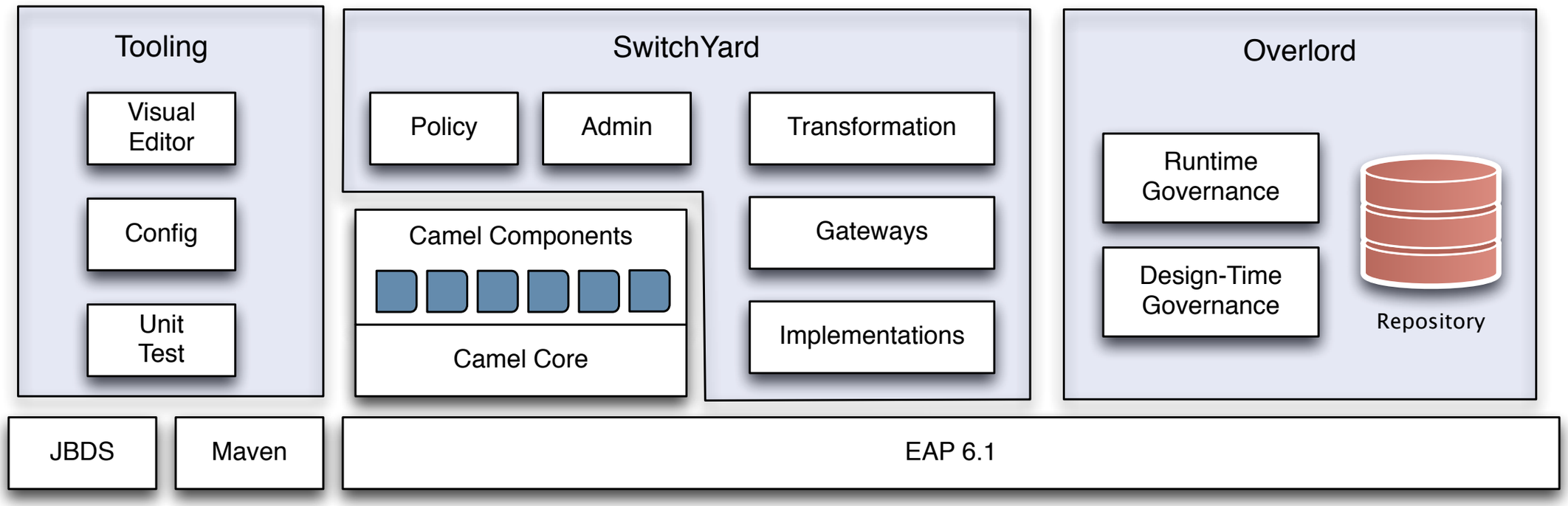
Governance

- SOA Repository
- Design Time Governance
- Runtime Governance

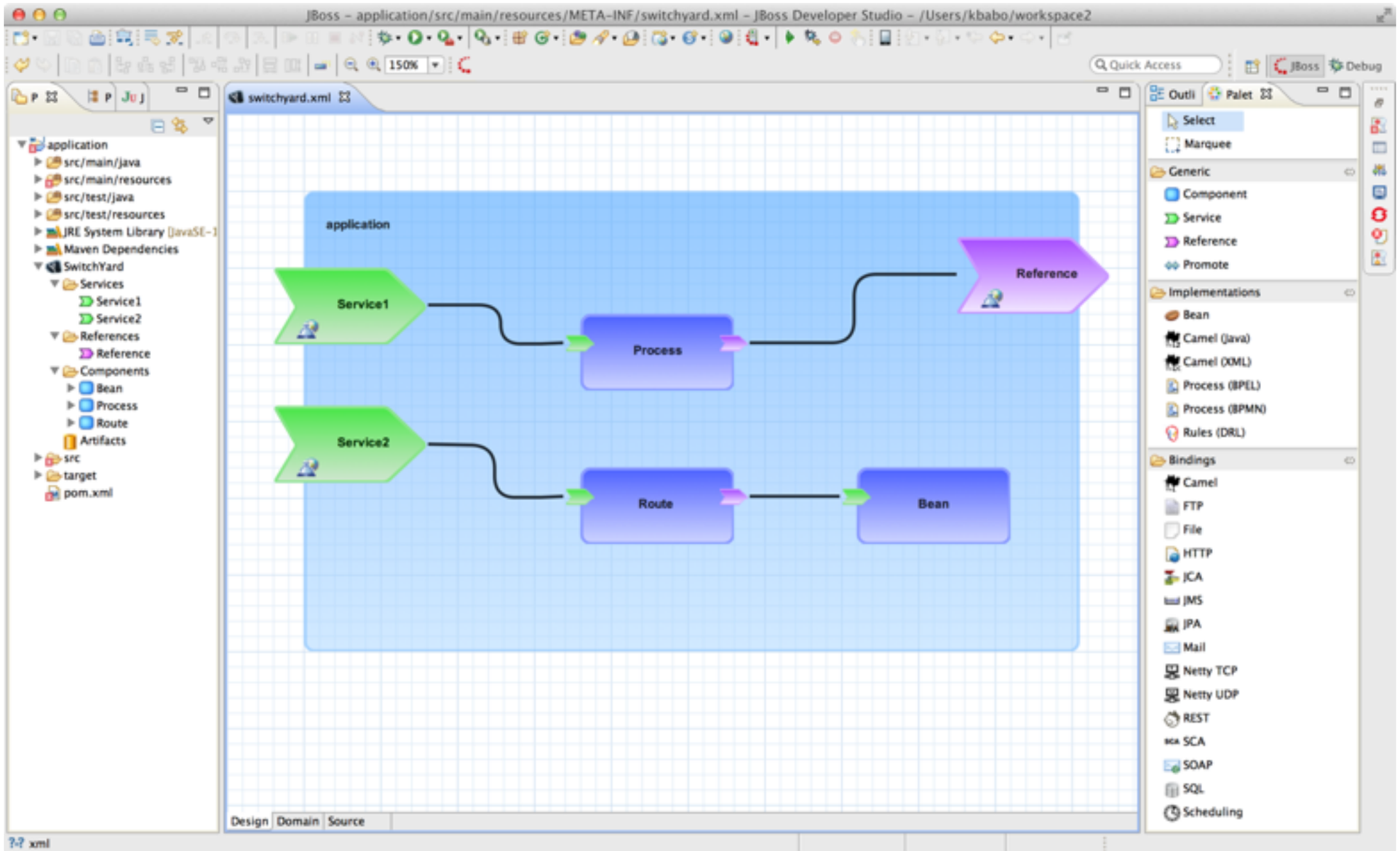
Relationship to SOA 5

- Functional equivalence to capabilities provided in SOA 5
- Conceptually similar in many ways
- Interoperability for side-by-side deployment
- Migration assistance

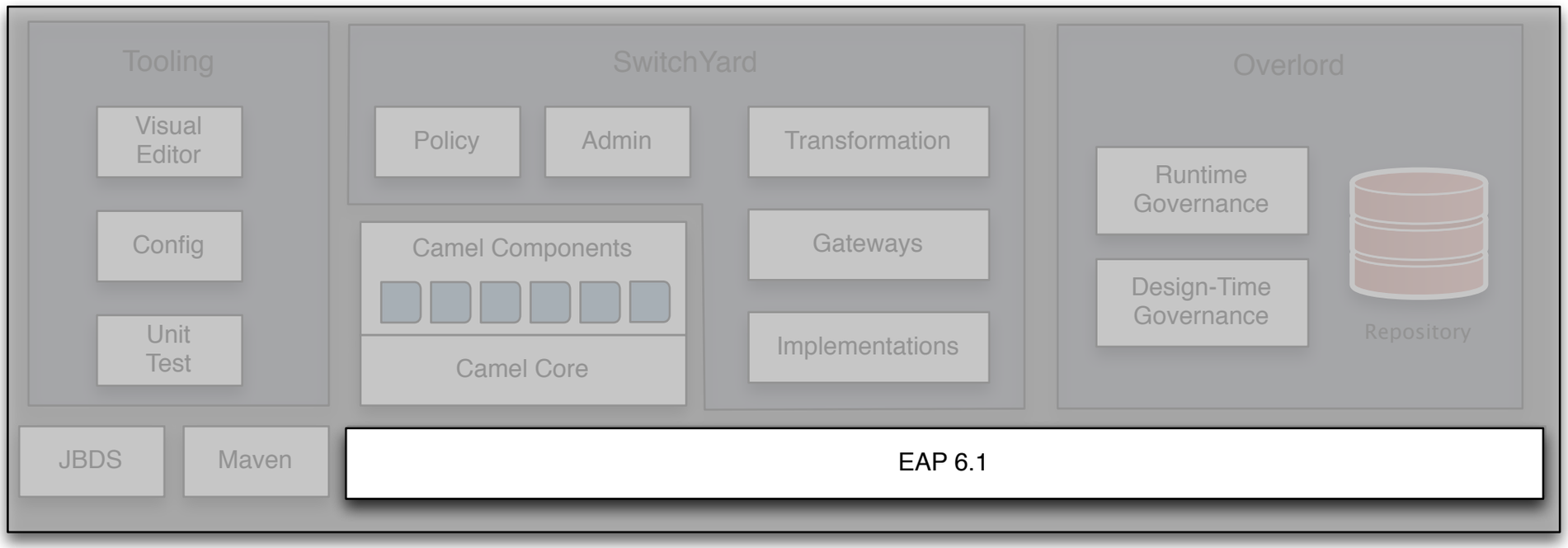
I Hate This Diagram



More My Speed



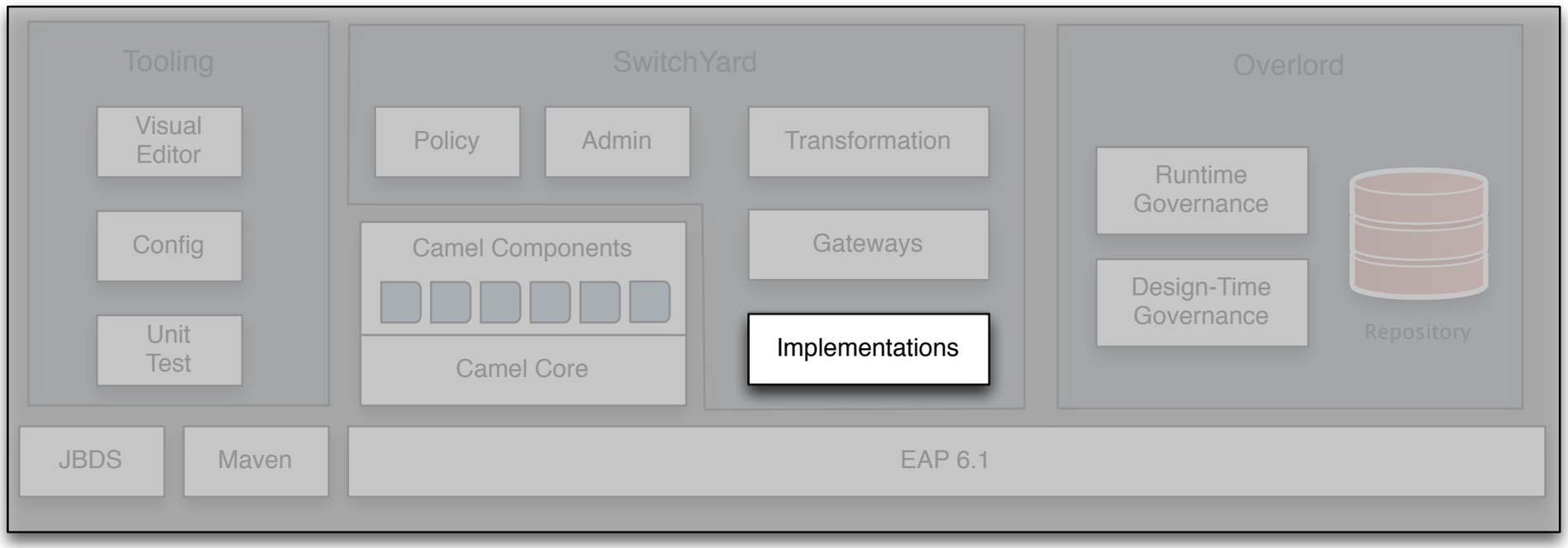
Runtime


Tooling
Visual Editor
Config
Unit Test
SwitchYard
Policy
Admin
Transformation
Camel Components
Camel Core
Gateways
Implementations
Overlord
Runtime Governance
Design-Time Governance
Repository
JBDS
Maven
EAP 6.1

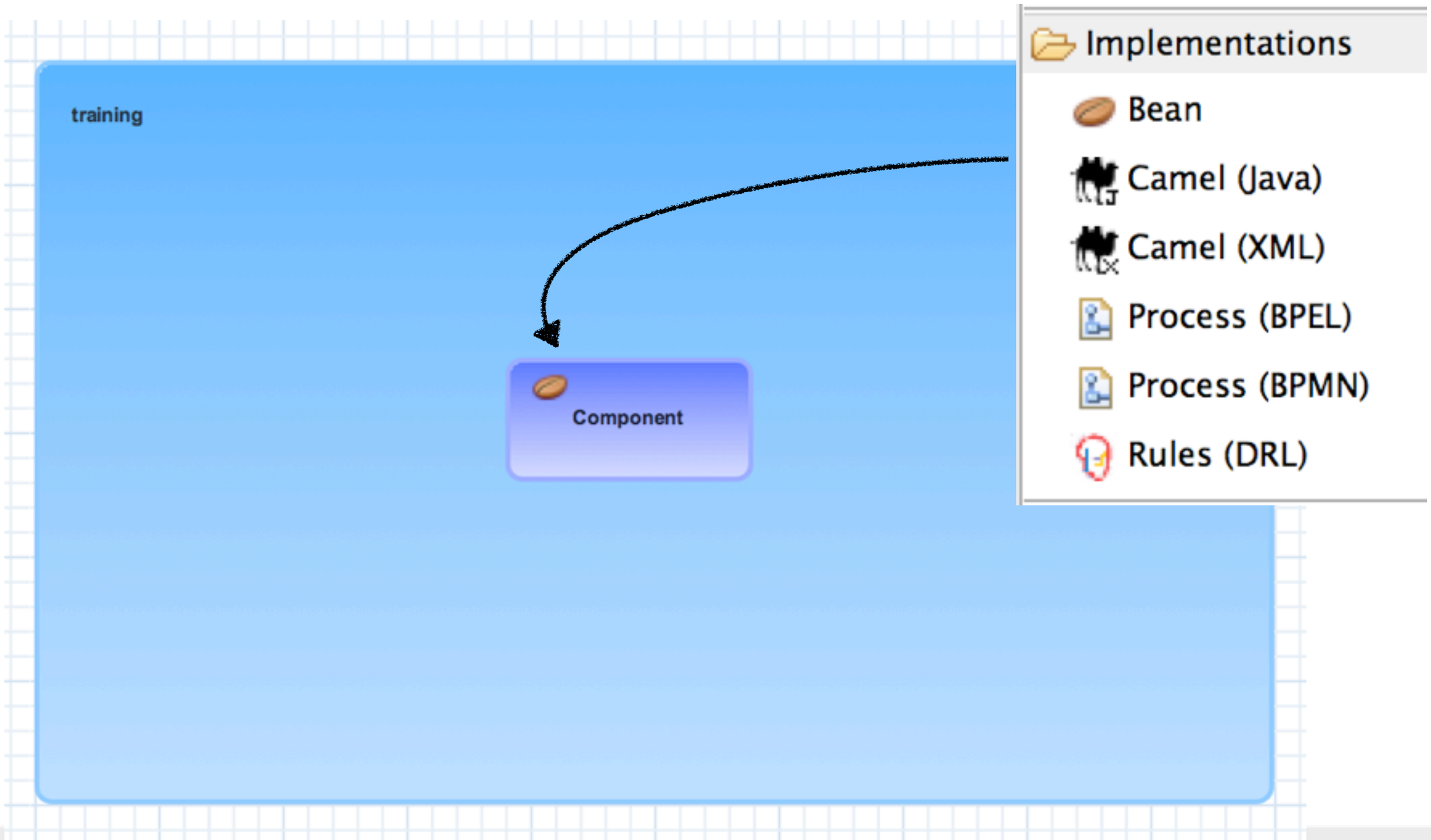
Runtime

- Layered Platform on EAP 6.1
- Standalone and Domain mode
- Subsystem and modules
- Deep integration
  - Administration
  - Security
  - Container services

Implementations



Implementations



Bean Implementations

- POJO = Service ... 'nuff said
- Easy to use
  - Annotation-driven
  - No wrapper/routing logic
  - Service auto-registered
- Standard programming model (Java EE / JSR 299)
- Straightforward integration into the web tier

Bean Service

```
public class OrderServiceBean implements OrderService {

    private InventoryService inventory;

    public OrderAck submitOrder(Order order) {
       // Check the inventory
       Item orderItem = inventory.lookupItem(
       order.getItemId());
    }
}
```

Bean Service

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

    private InventoryService inventory;

    public OrderAck submitOrder(Order order) {
       // Check the inventory
       Item orderItem = inventory.lookupItem(
       order.getItemId());
    }
}
```

Bean Service

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

    @Inject @Reference
    private InventoryService inventory;

    public OrderAck submitOrder(Order order) {
       // Check the inventory
       Item orderItem = inventory.lookupItem(
       order.getItemId());
    }
}
```

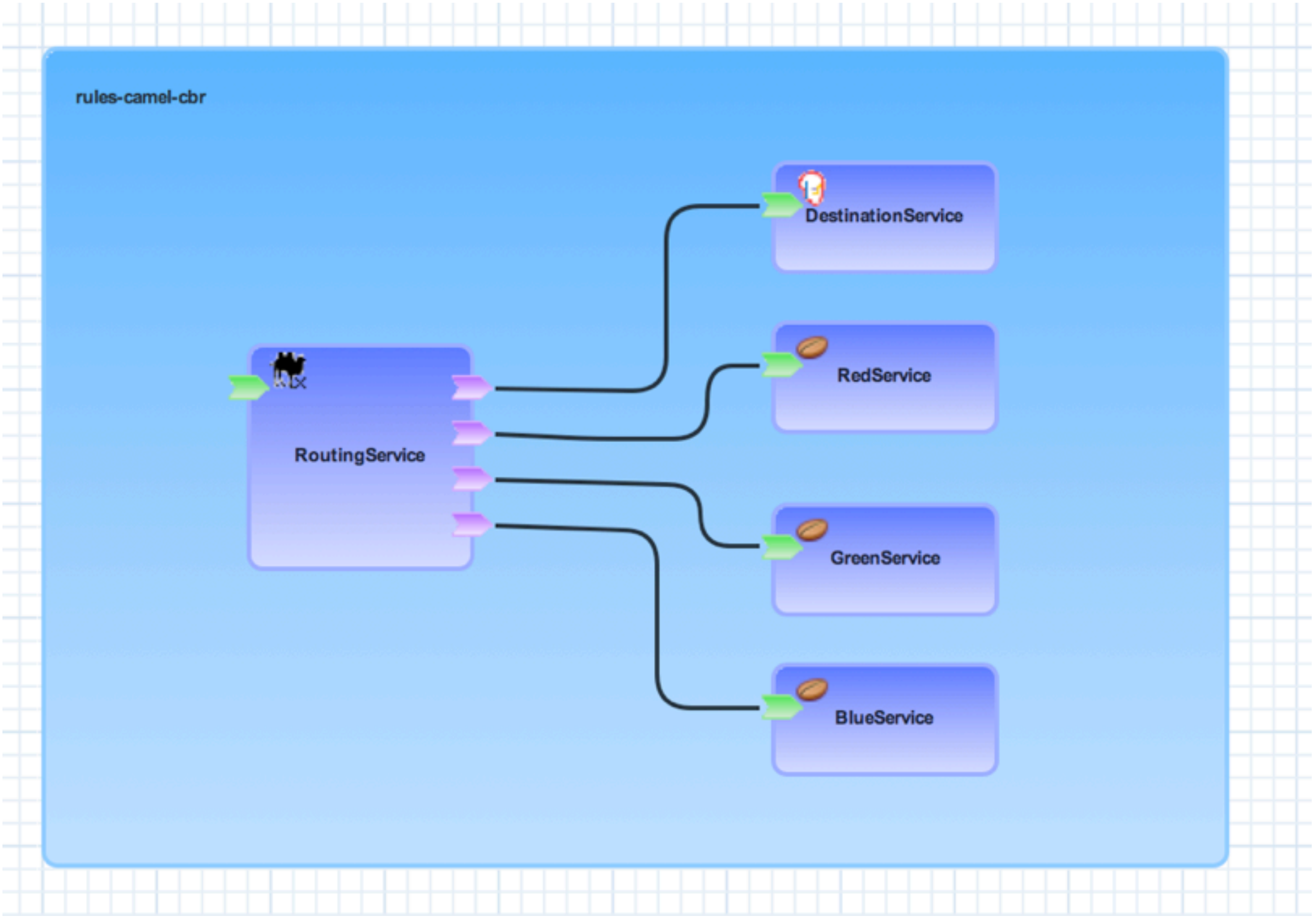
Camel Implementations

- Camel route as a service implementation
- Replaces action processing pipeline in JBoss ESB
- Use it for
  - Routing
  - Enterprise Integration Patterns
  - Scripting Languages
  - Data Formats

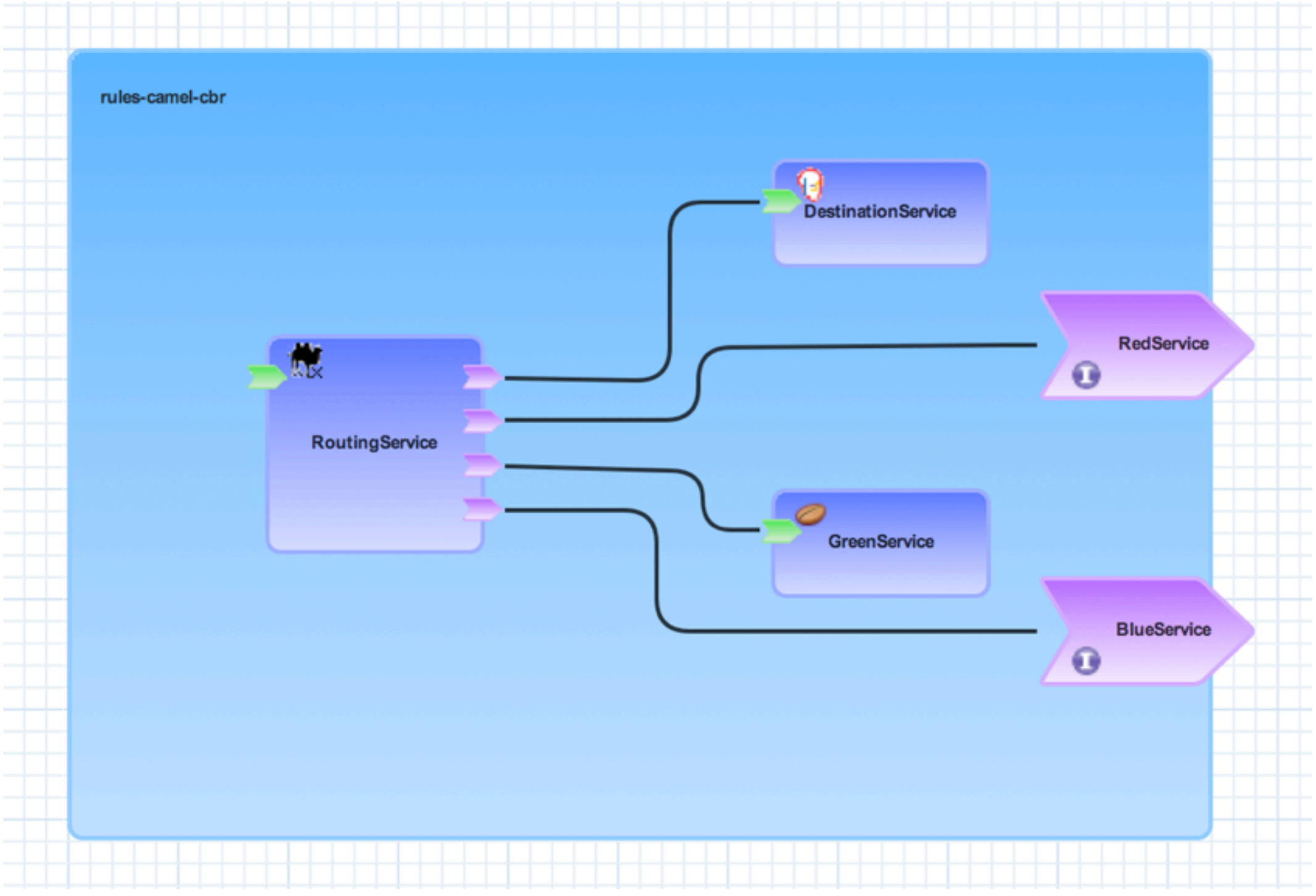
Camel Implementation

```
<route xmlns="http://camel.apache.org/schema/spring">
    <from uri="switchyard://RoutingService"/>
    <to uri="switchyard://DestinationService"/>
    <choice>
        <when>
            <simple>${body.destination} == 'Red'</simple>
            <to uri="switchyard://RedService"/>
        </when>
        <when>
            <simple>${body.destination} == 'Green'</simple>
            <to uri="switchyard://GreenService"/>
        </when>
        <when>
            <simple>${body.destination} == 'Blue'</simple>
            <to uri="switchyard://BlueService"/>
        </when>
        <otherwise>
            <log message="Unknown Destination!"/>
        </otherwise>
    </choice>
</route>
```

Local Reference



Remote Reference



Beans In Camel

- Allows Java objects to be called inside a route
- Very useful for fine-grained integration tasks
  - EIP configuration - route, split, etc.
  - Metadata access
  - Bolt-on logic
- Bean registry is pluggable

CDI Bean

```
import javax.enterprise.context.ApplicationScoped;
import javax.inject.Named;

@Named("MyBean")
@ApplicationScoped
public class SomeBean {

   public String doSomething(String content) {
      return "I did something with " + content;
   }
}
```

CDI Bean in Camel

```
import org.apache.camel.builder.RouteBuilder;

public class CamelServiceRoute extends RouteBuilder {

    public void configure() {
        from("switchyard://InventoryService")
            .split(body(String.class).tokenize("\n"))
            .filter(body(String.class).startsWith("item:"))
                .bean("bean:MyBean");
    }
}
```

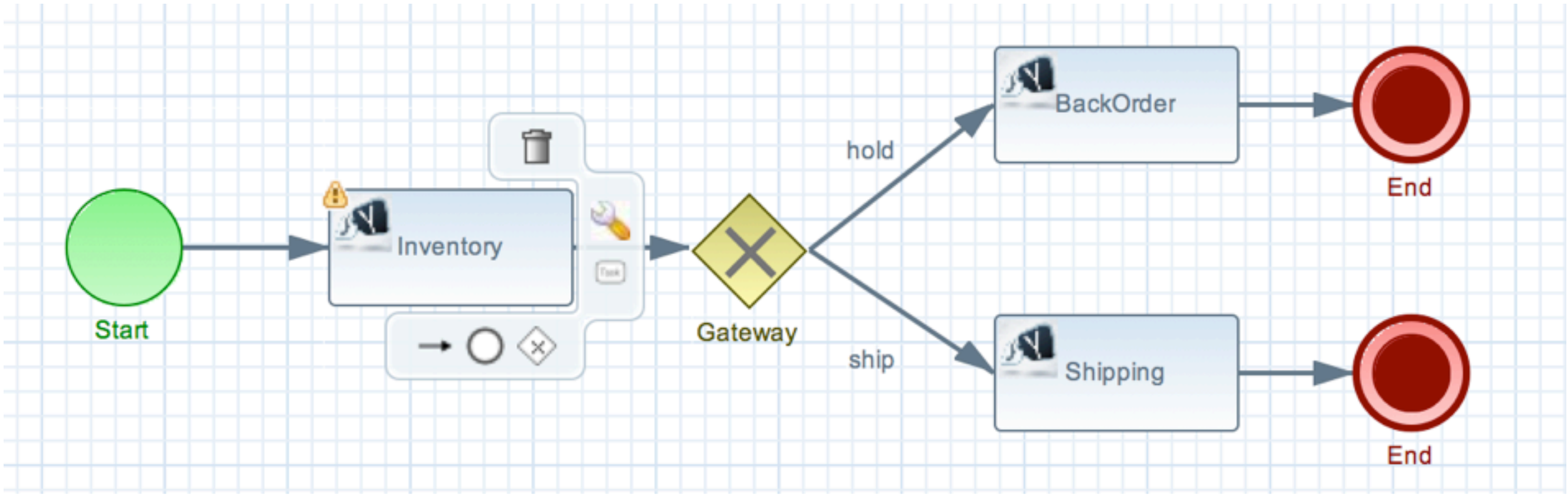
Rules Implementations

- Based on Drools 6
- Stateful and stateless sessions
- Complex Event Processing (with BRMS)
- Local and remote knowledge definitions
- Provide and consume services

BPMN Implementations

- Based on jBPM 6, integrates with BPM Platform
- Start and interact with process instances
- Provide and consume services
  - Standard BPMN 2 Service Task
  - Dynamic Service Task
- Fault handling
- Human task integration

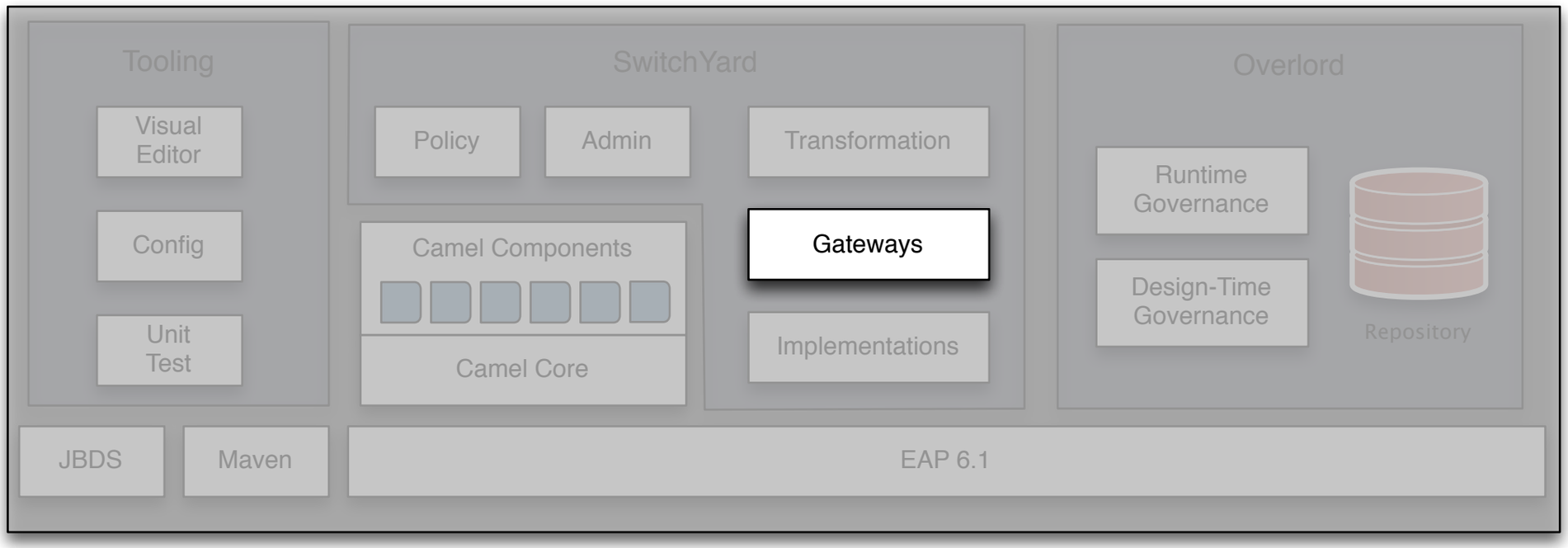
Service Orchestration with BPMN 2



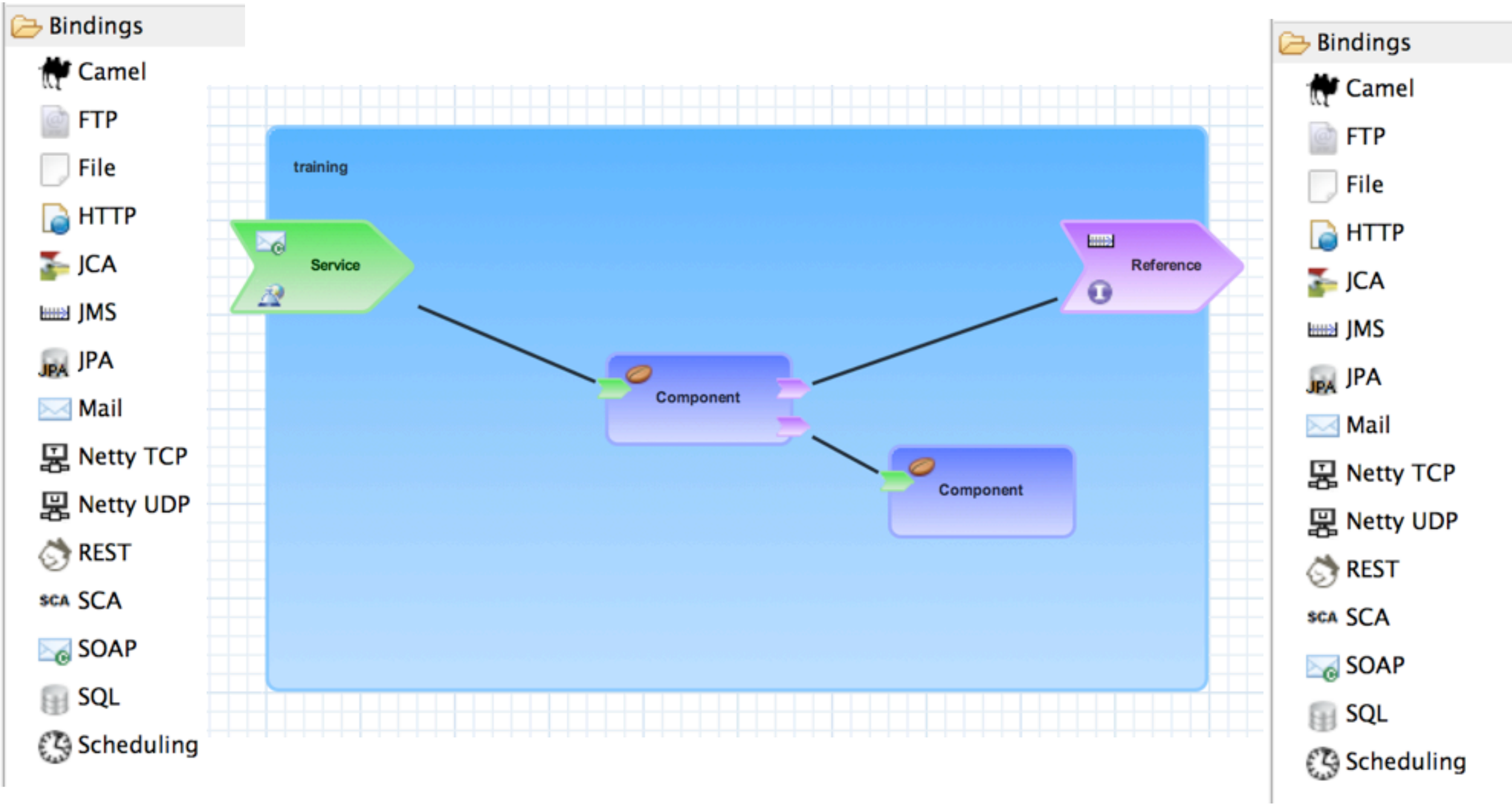
BPEL Implementations

- WS-BPEL 2.0 courtesy of Riftsaw
- Provide and consume from BPEL
- Combine with any gateway binding
- Native support for WSDL contracts

Gateways


Tooling
Visual Editor
Config
Unit Test
SwitchYard
Policy
Admin
Transformation
Camel Components
Camel Core
Gateways
Implementations
Overlord
Runtime Governance
Design-Time Governance
Repository
JBDS
Maven
EAP 6.1

Gateways


Bindings
Camel
FTP
File
HTTP
JCA
JMS
JPA
Mail
Netty TCP
Netty UDP
REST
SCA
SOAP
SQL
Scheduling
training
Service
Component
Component
Reference
Bindings
Camel
FTP
File
HTTP
JCA
JMS
JPA
Mail
Netty TCP
Netty UDP
REST
SCA
SOAP
SQL
Scheduling

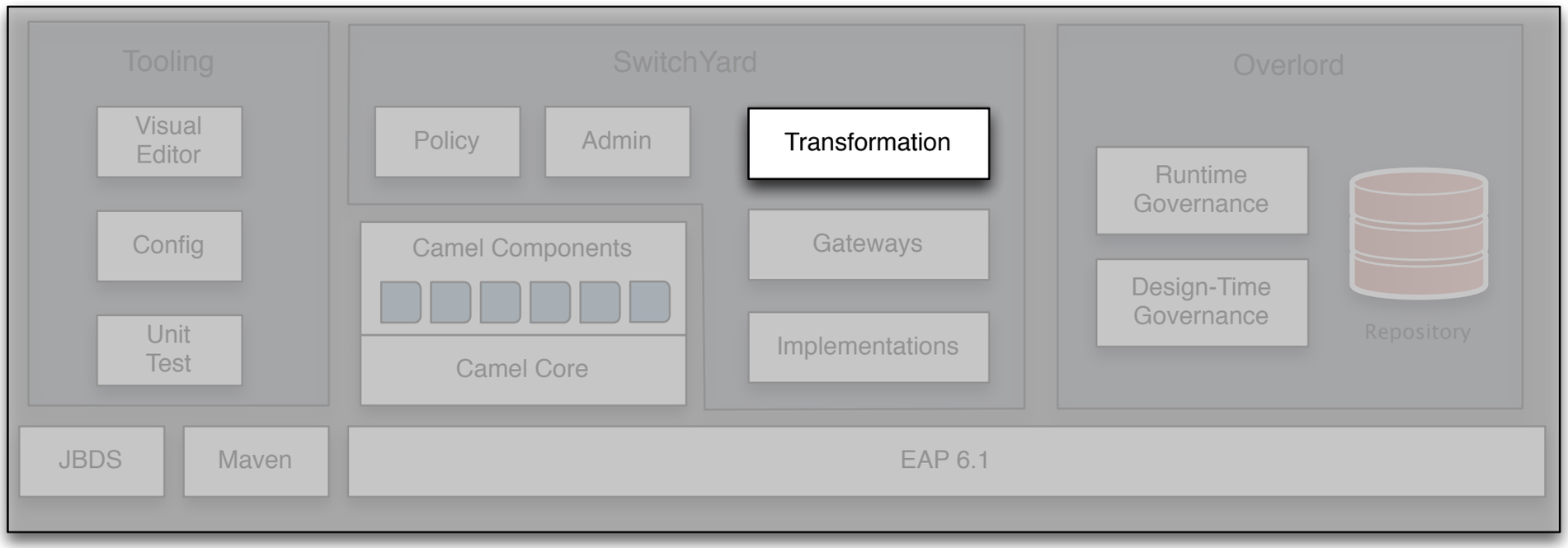
Gateways

- Enable communication outside the application through bindings
- Decoupled from service implementations
- Configuration defined by XML Schema
- Extensions supported through Camel

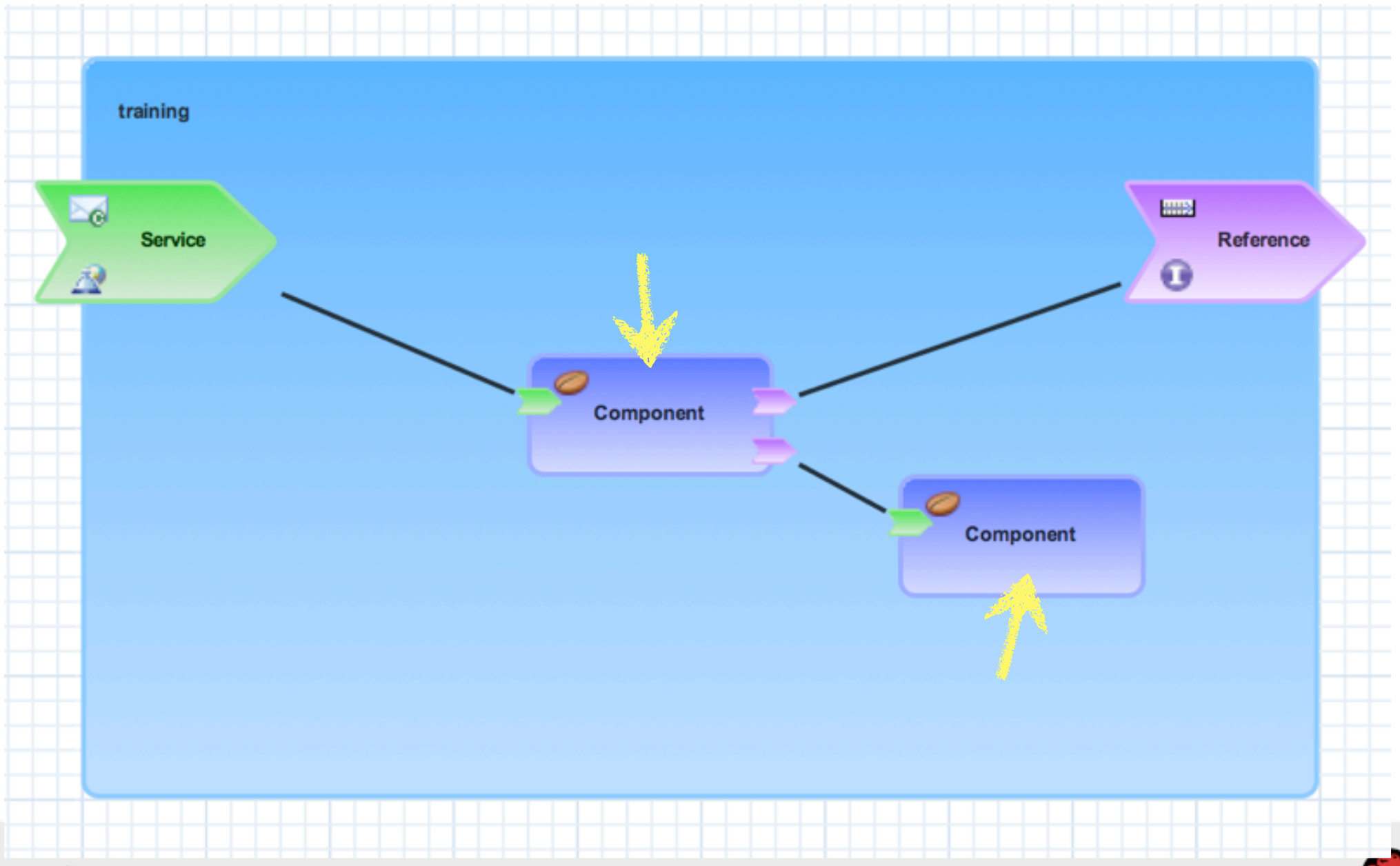
Extensions

- Camel components not included directly in SwitchYard
  - Fuse
  - Custom
- Using extensions
  - Create module(s)
  - Add to runtime
  - Configure via Camel URI gateway

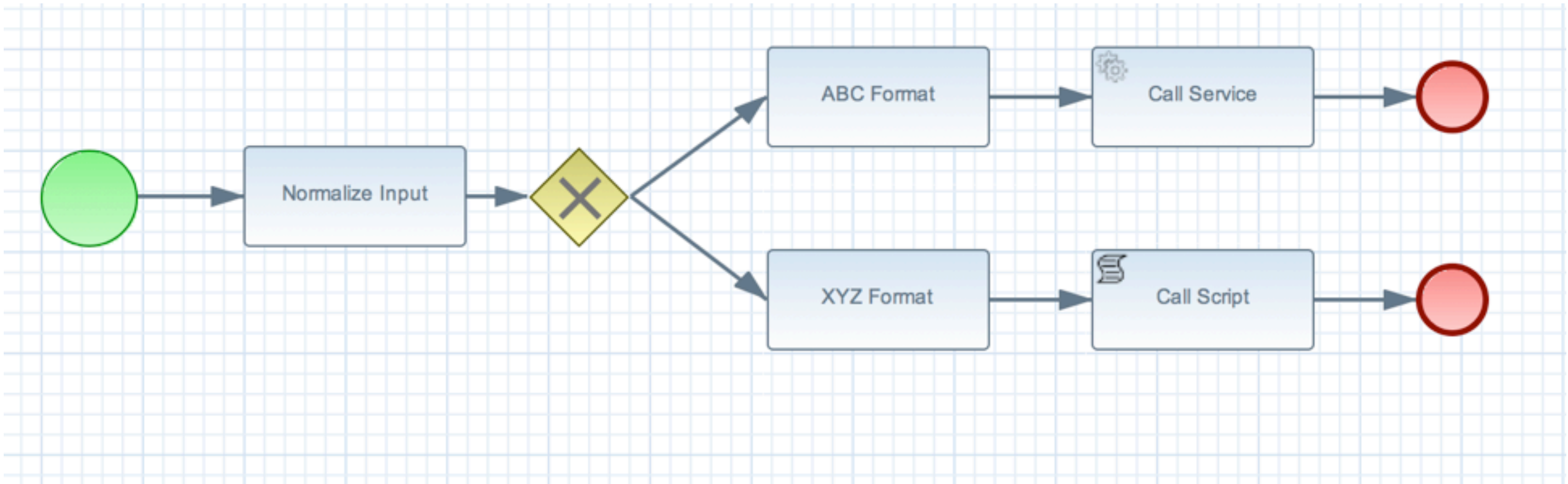
Transformation



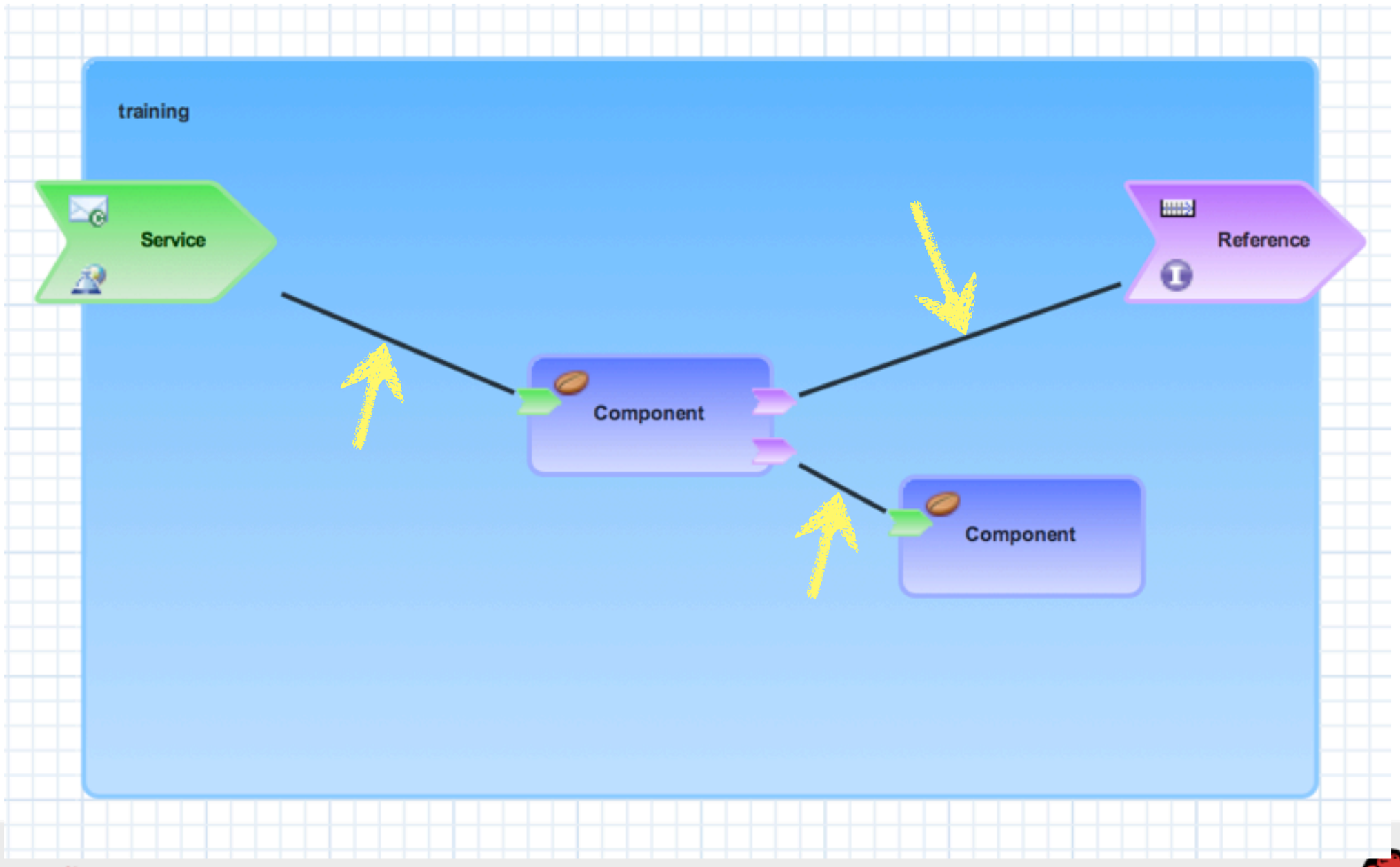
Procedural



Procedural in BPM

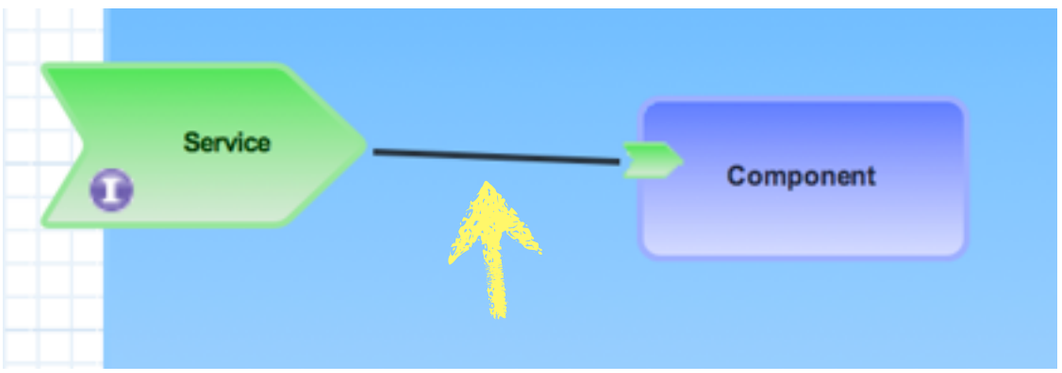


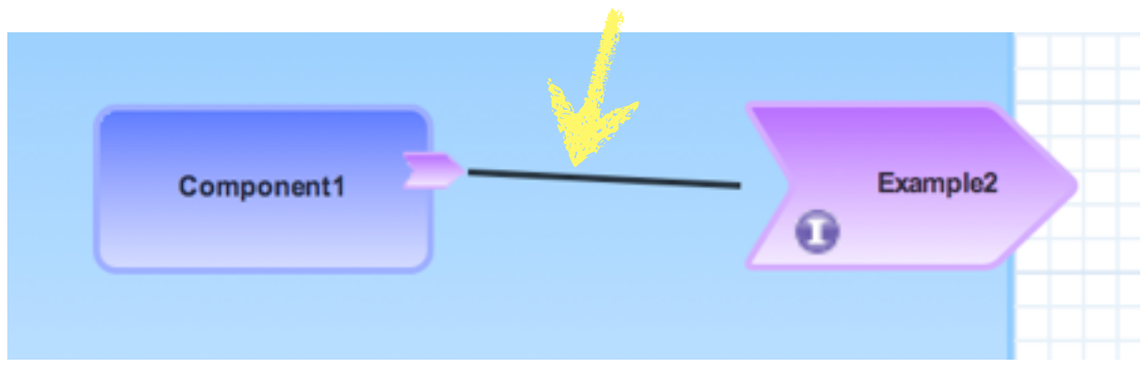
Declarative



Transformers

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

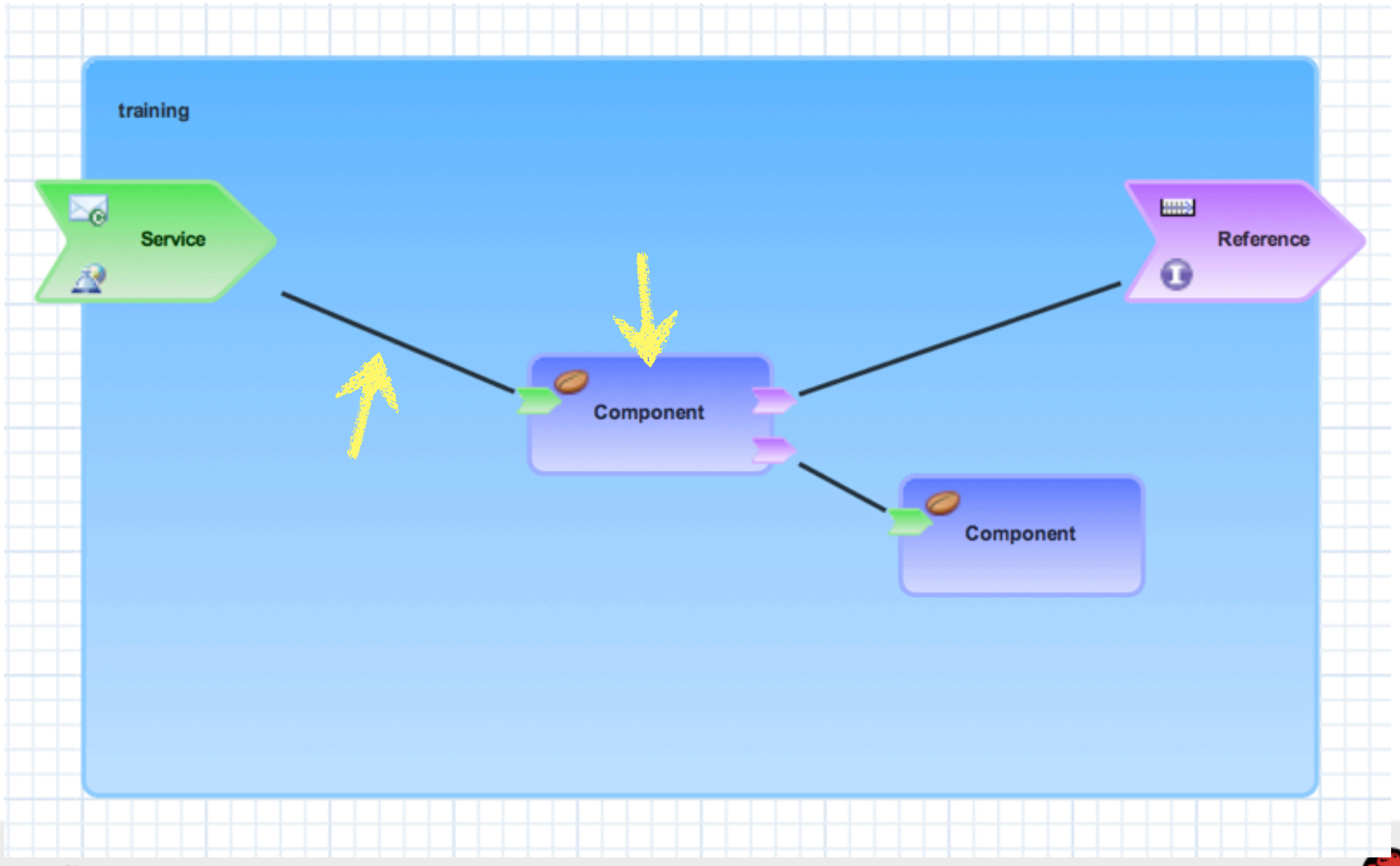




Java Transformer

```
@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”)));
}
```

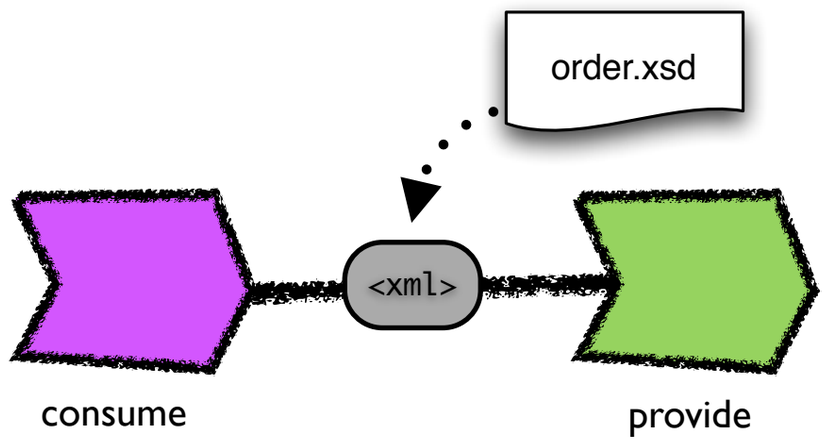
Hybrid



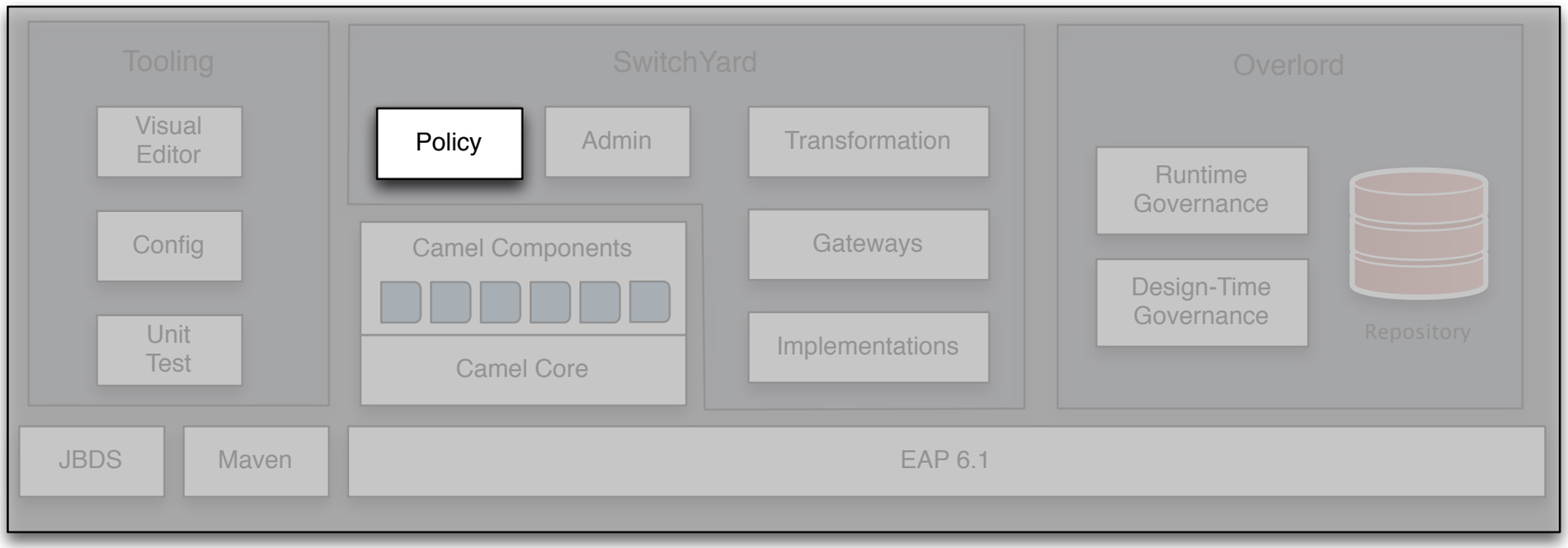
Validators

- 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"/>
```



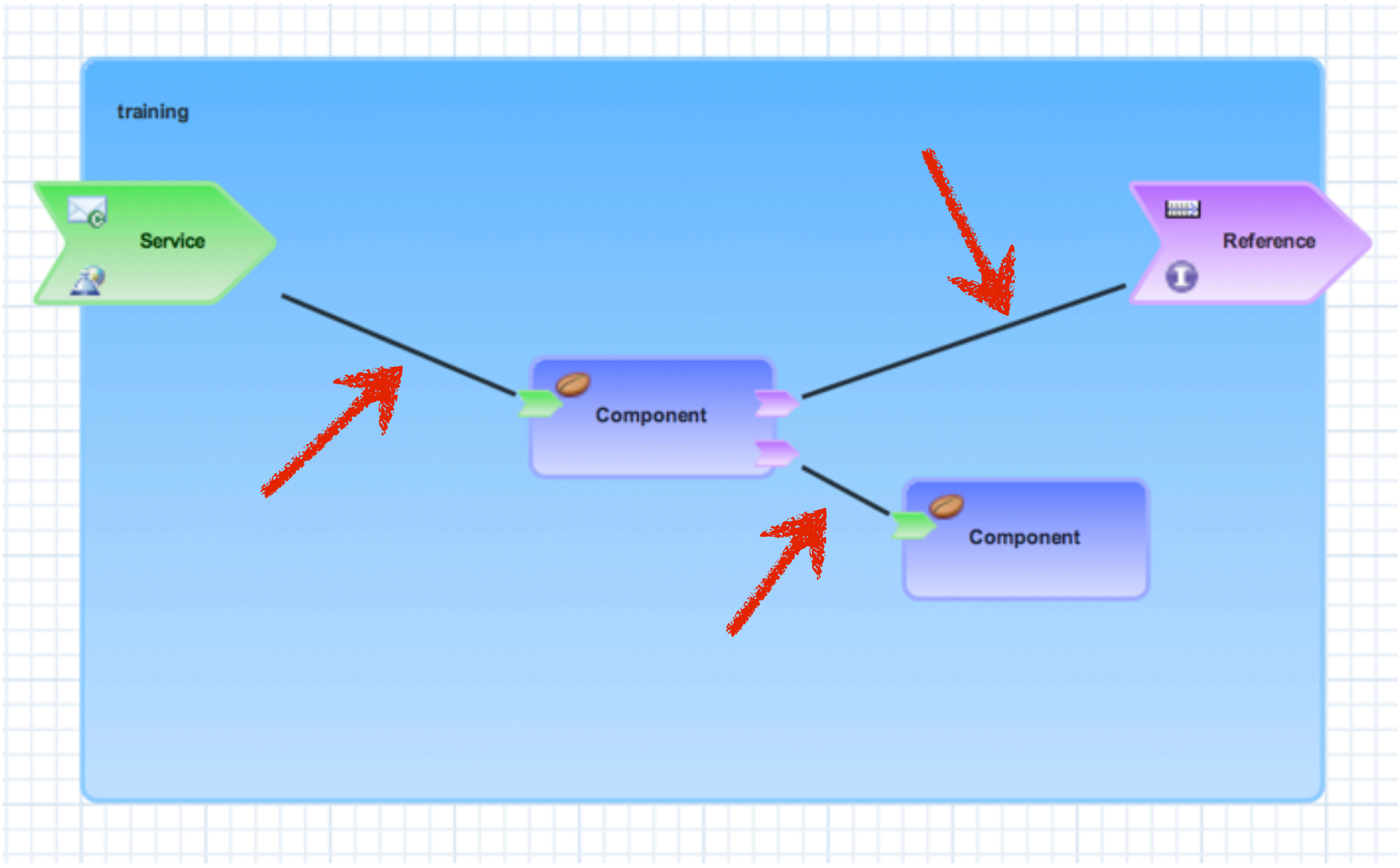
Policy


Tooling
Visual Editor
Config
Unit Test
SwitchYard
Policy
Admin
Transformation
Camel Components
Camel Core
Gateways
Implementations
Overlord
Runtime Governance
Design-Time Governance
Repository
JBDS
Maven
EAP 6.1

Declarative Policy

- Declare the ‘what’, defer the ‘how’
- Requirements attached to
  - Component service
  - Component reference
  - Implementation
- Runtime injects policy enforcement point

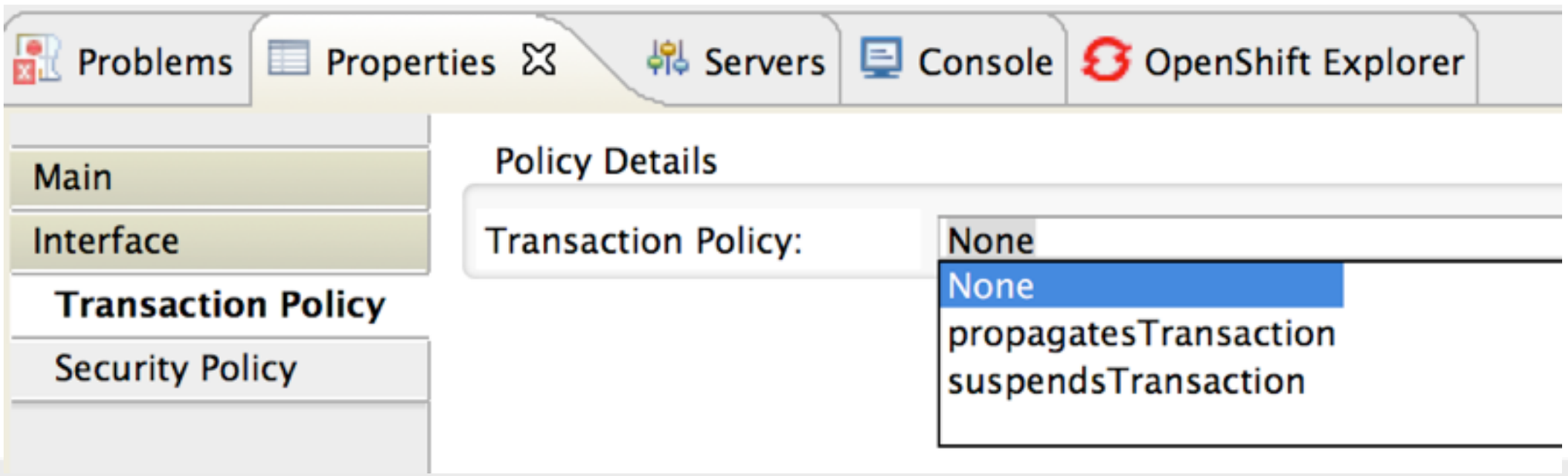
Policy Enforcement Points



Transaction Policy

- Interaction Policy
  - Suspend transaction
  - Require transaction
- Implementation Policy
  - Global transaction
  - Local transaction
  - No transaction

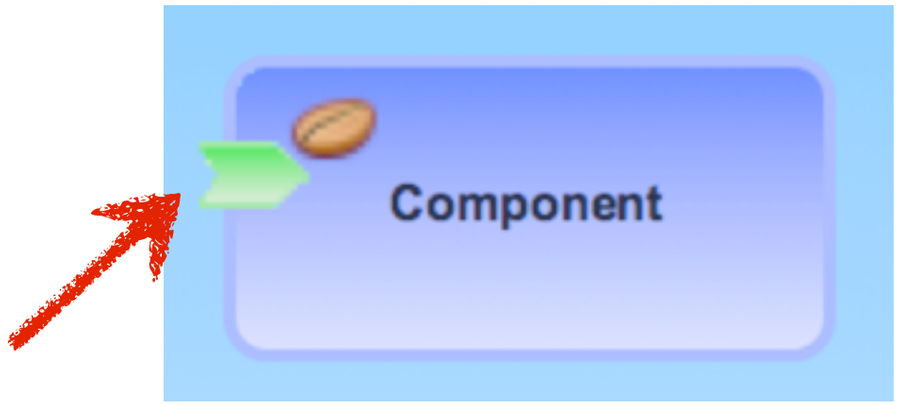
Interaction Policy

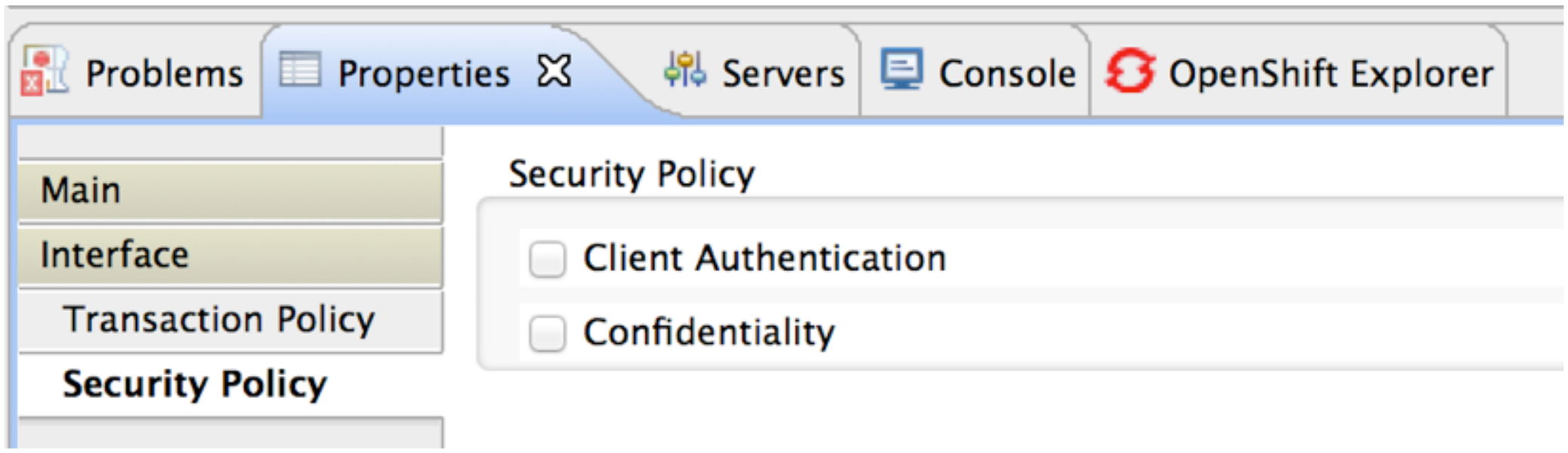


Security Policy

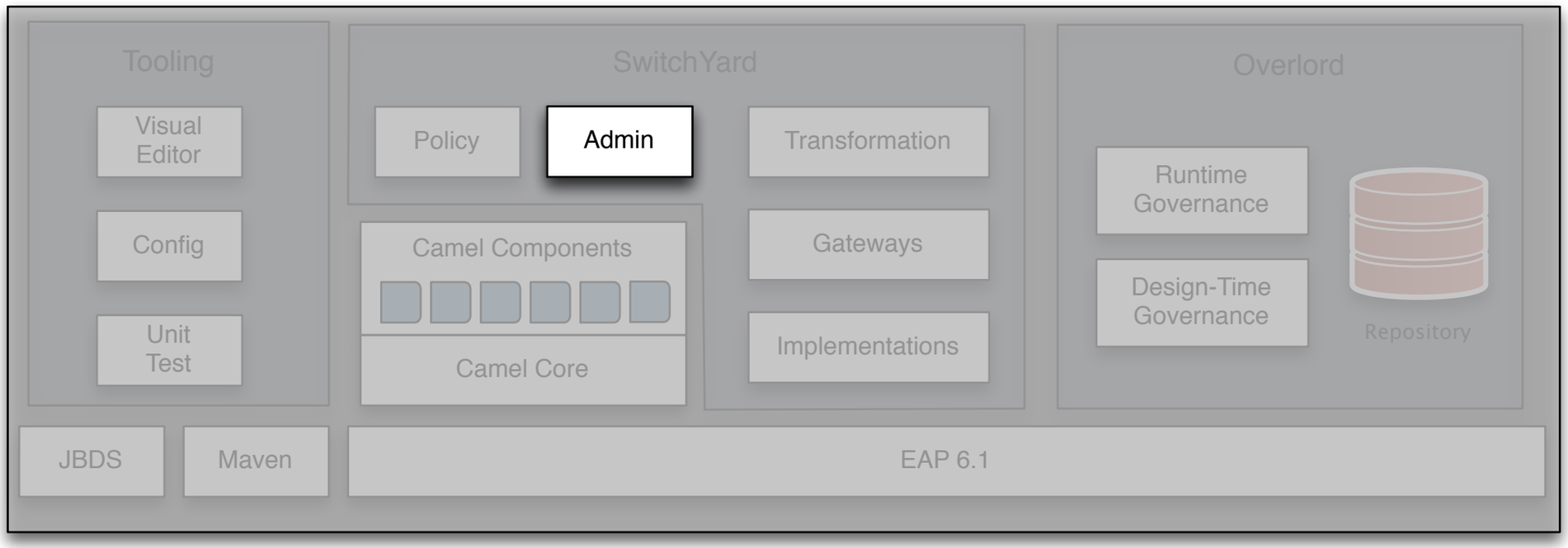
- Interaction policy only
  - Authorization
  - Client Authentication
  - Confidentiality
- Integrated with container
  - JAAS
  - PicketLink
  - WS-Security

Service Security

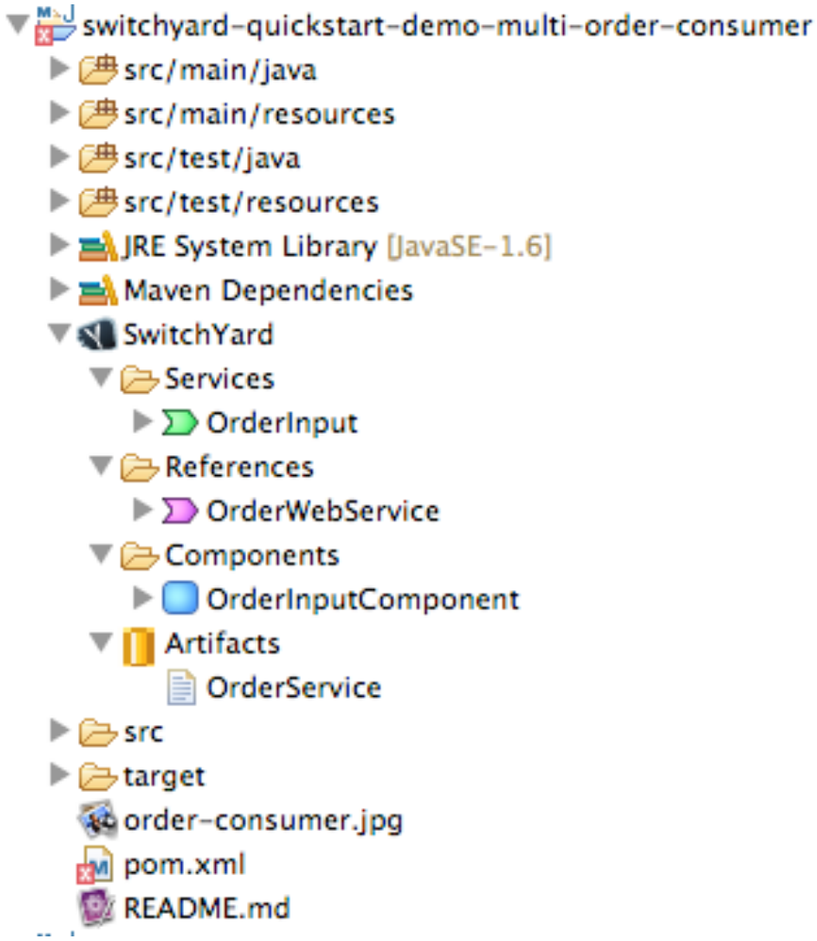


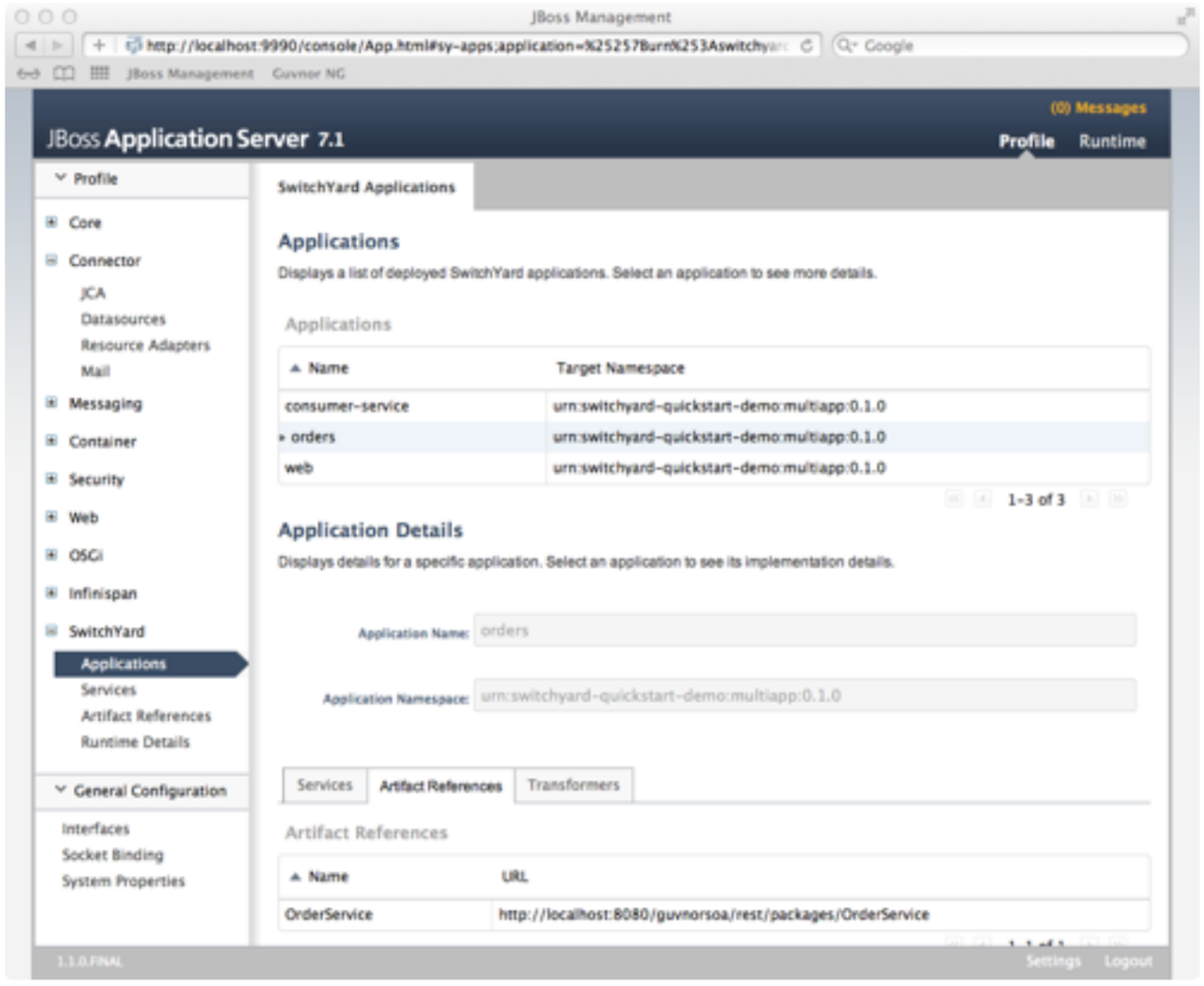


Administration

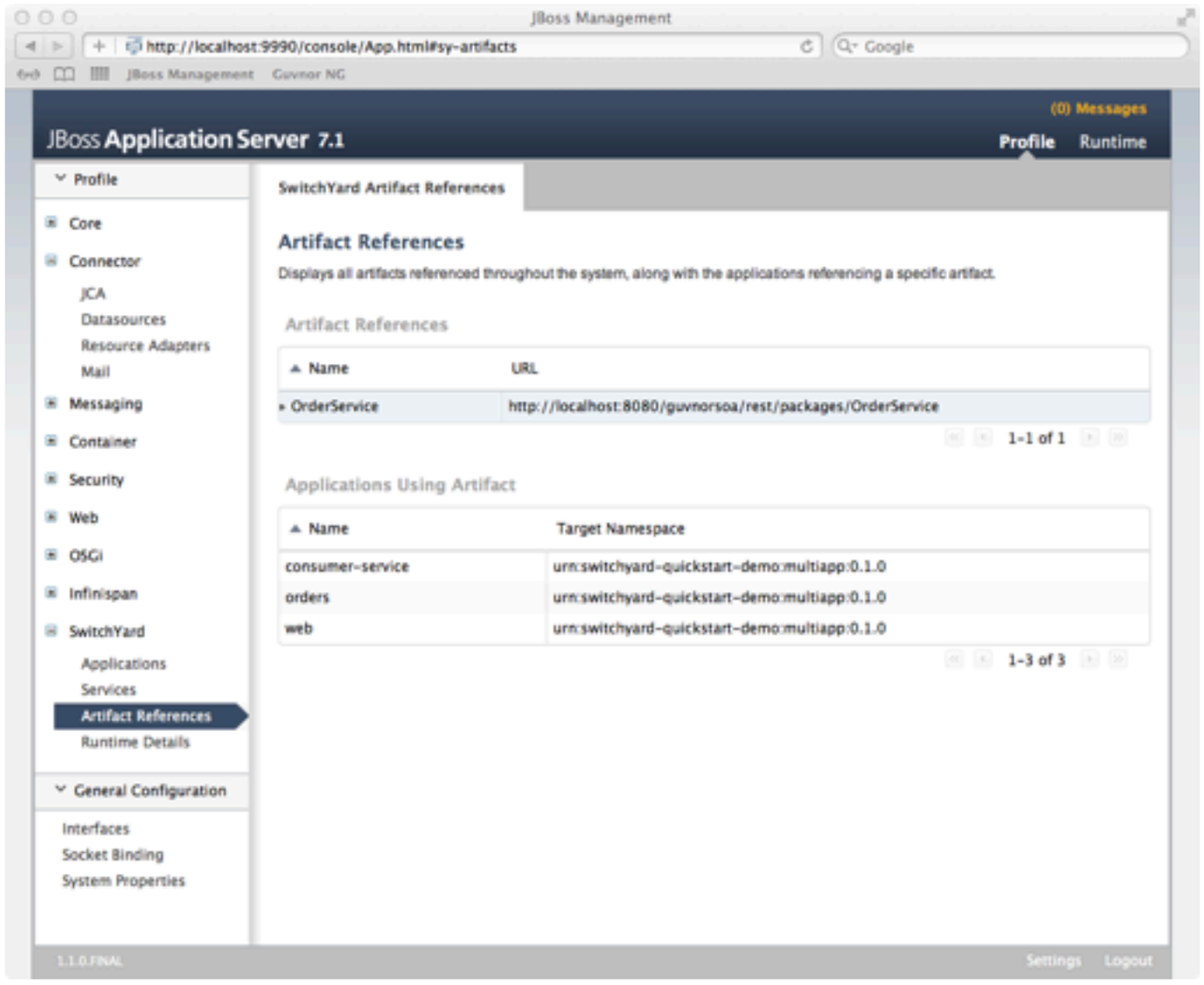


Administration Console

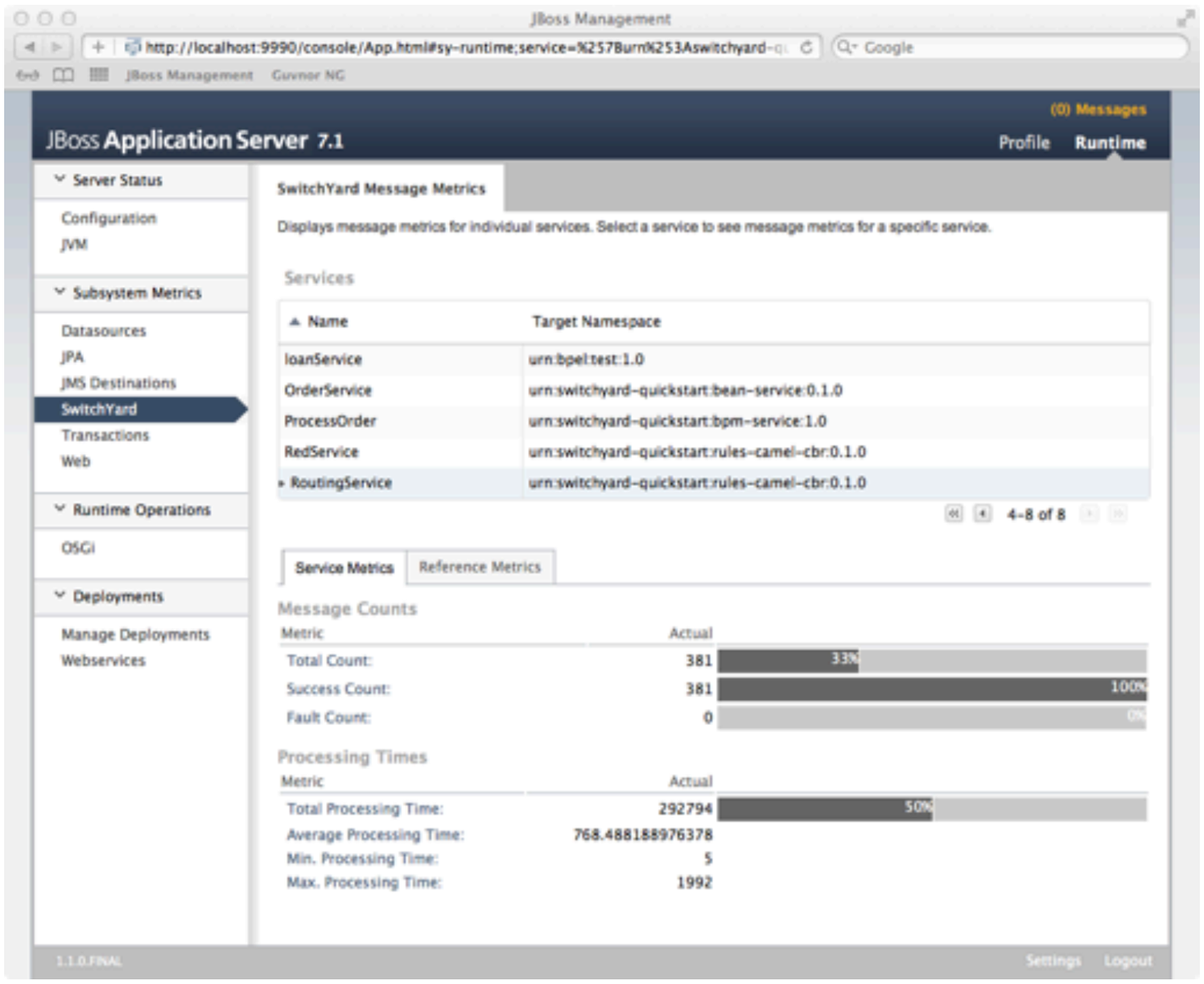




Administration Console



Service Metrics



Reference Metrics

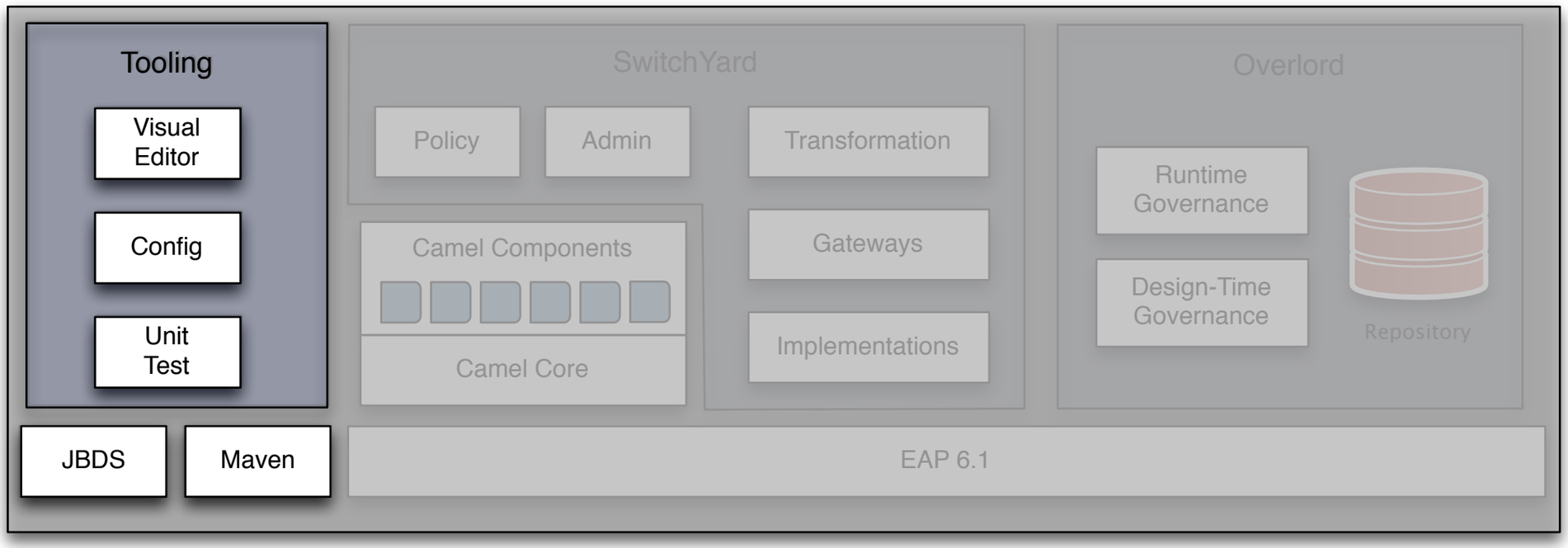
Service Metrics | **Reference Metrics**

Referenced Service Metrics

Name	Message Count	Average Time	Time %	Fault %
BlueService	0	0	0%	0%
DestinationService	382	0.15183246073298 43	0%	0%
GreenService	191	49.74869109947644	3%	0%
RedService	190	1483.9315789473685	96%	0%

1-4 of 4

Tooling


Tooling
Visual Editor
Config
Unit Test
JBDS
Maven
SwitchYard
Policy
Admin
Transformation
Camel Components
Camel Core
Gateways
Implementations
Overlord
Runtime Governance
Design-Time Governance
Repository
EAP 6.1

RED HAT SUMMIT

Demo

redhat.

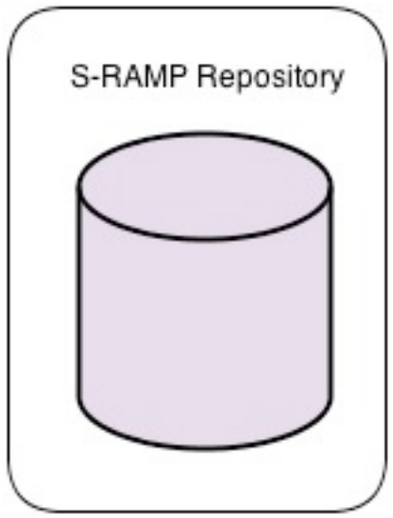
SOA Governance

- Processes to manage delivery of Service Oriented Architecture (SOA)
  - Compliance (e.g. Sarbanes-Oxley)
  - Artifact Lifecycle
    - Inception through to Retirement
  - Policy enforcement
  - Monitoring
  - Auditing

Overlord

- Umbrella project for SOA Governance projects
  - http://www.jboss.org/overlord
  - https://github.com/governance
- Primary projects are
  - S-RAMP (SOA Repository)
  - DTGov (Design Time Governance)
  - RTGov (Runtime Governance)

S-RAMP



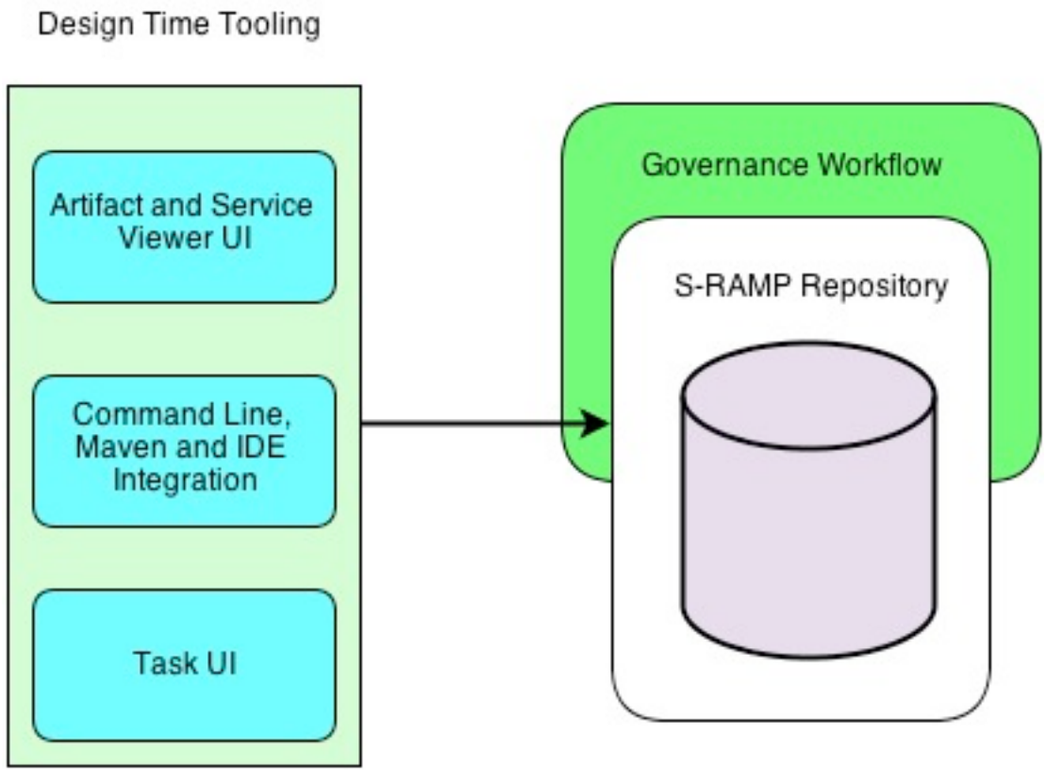
S-RAMP

- SOA Repository Artifact Model and Protocol
  - OASIS Technical Committee
- Common Data Model for SOA
- Interaction protocol
  - ATOM REST
- Enables
  - Common Tooling
  - Data Sharing

S-RAMP

- Central Artifact Repository
- Extract Metadata
  - Classifiers
  - Relationships
- Event Notification
- ModeShape (JCR) + Infinispan

Design Time Governance



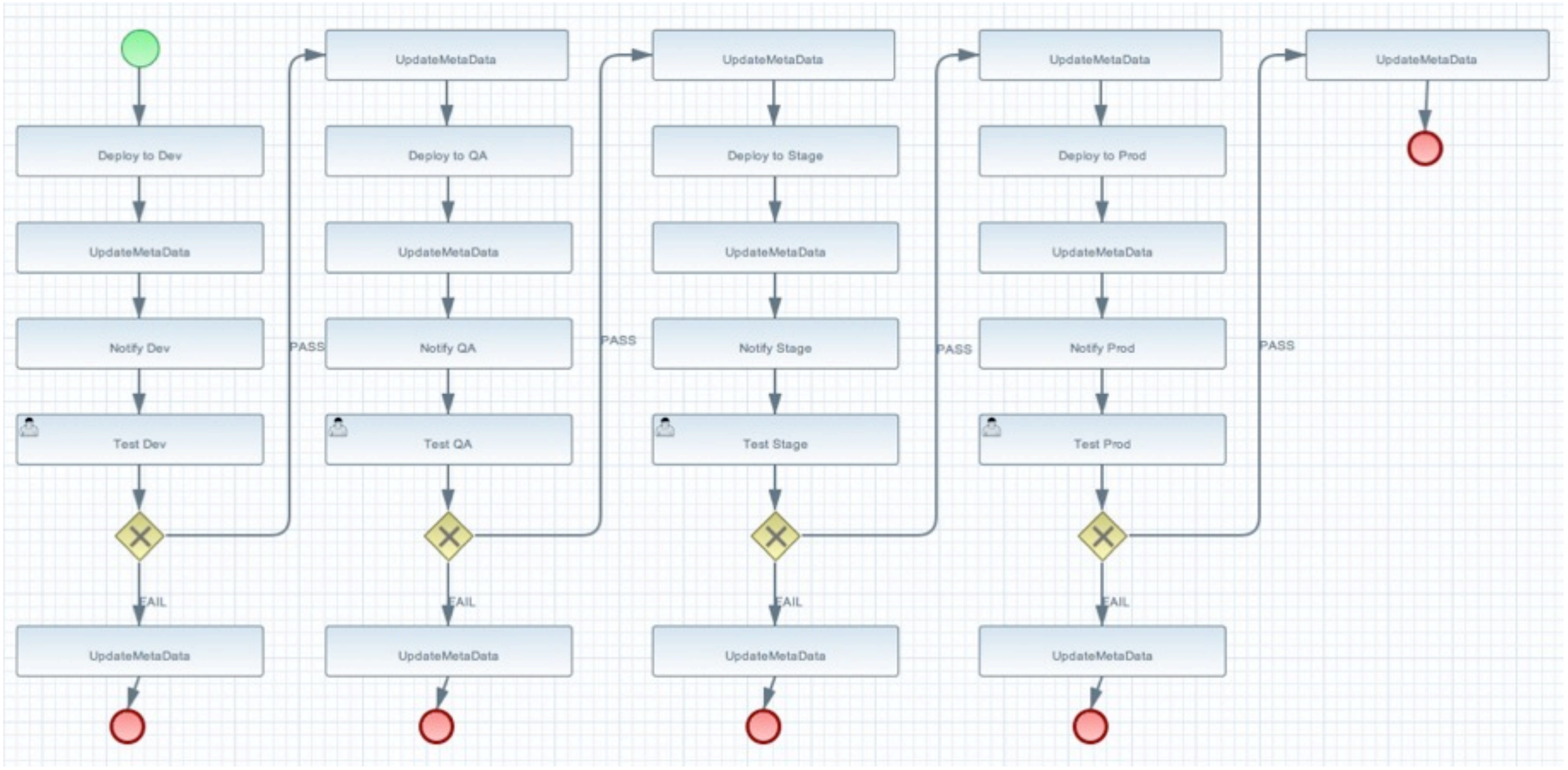
Design Time Governance

- Artifact Discovery
- Artifact Lifecycle
  - Services
  - Policies
- Relationship Visualisation
- Change Notification
- Impact Analysis

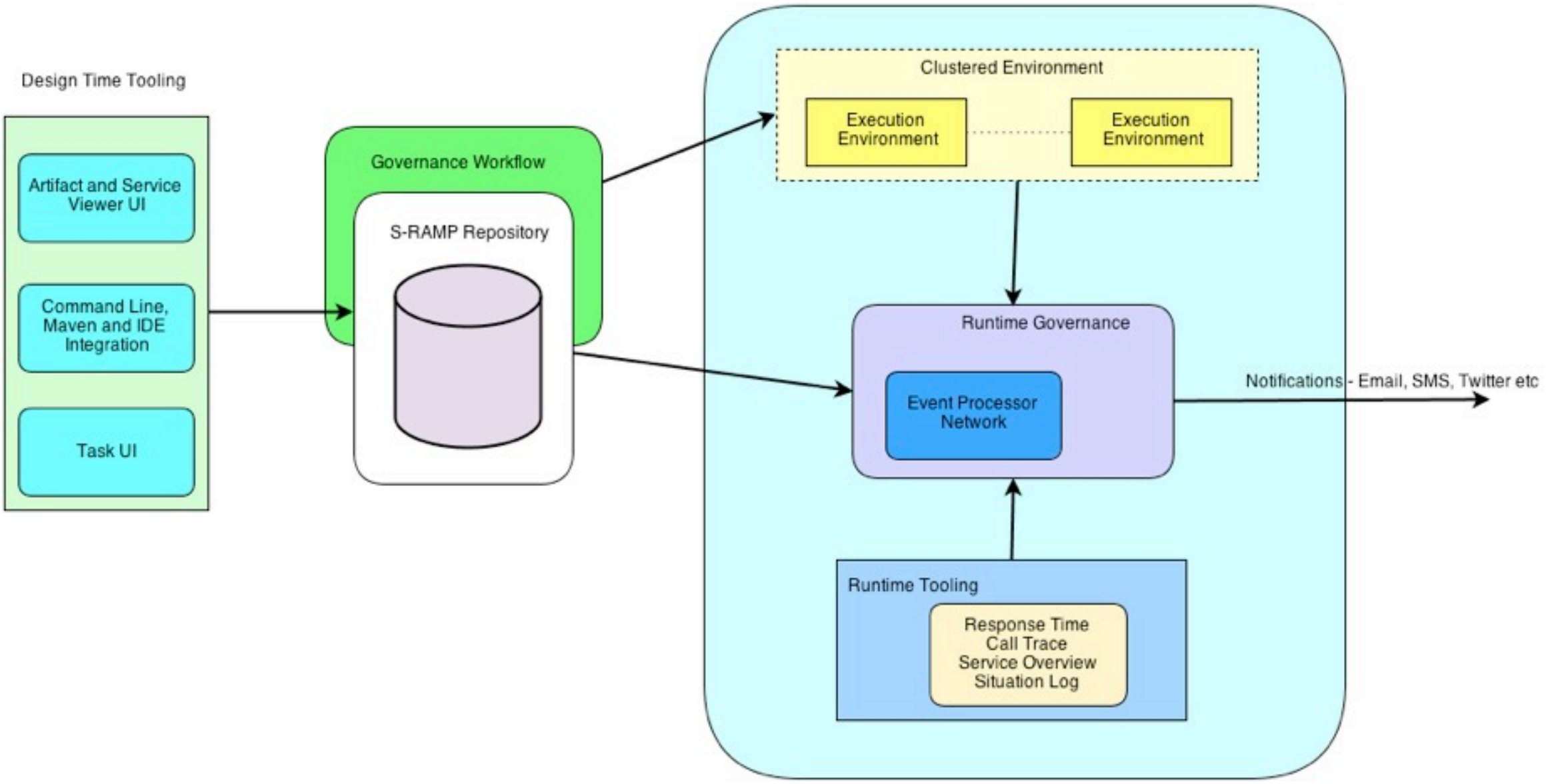
Design Time Governance

- Workflow Integration (BPMN2)
  - Triggered by S-RAMP
  - Manage Artifact Lifecycle
  - Deploy to Execution Environment (direct, JON)
- Task UI

Design Time Governance



Runtime Governance



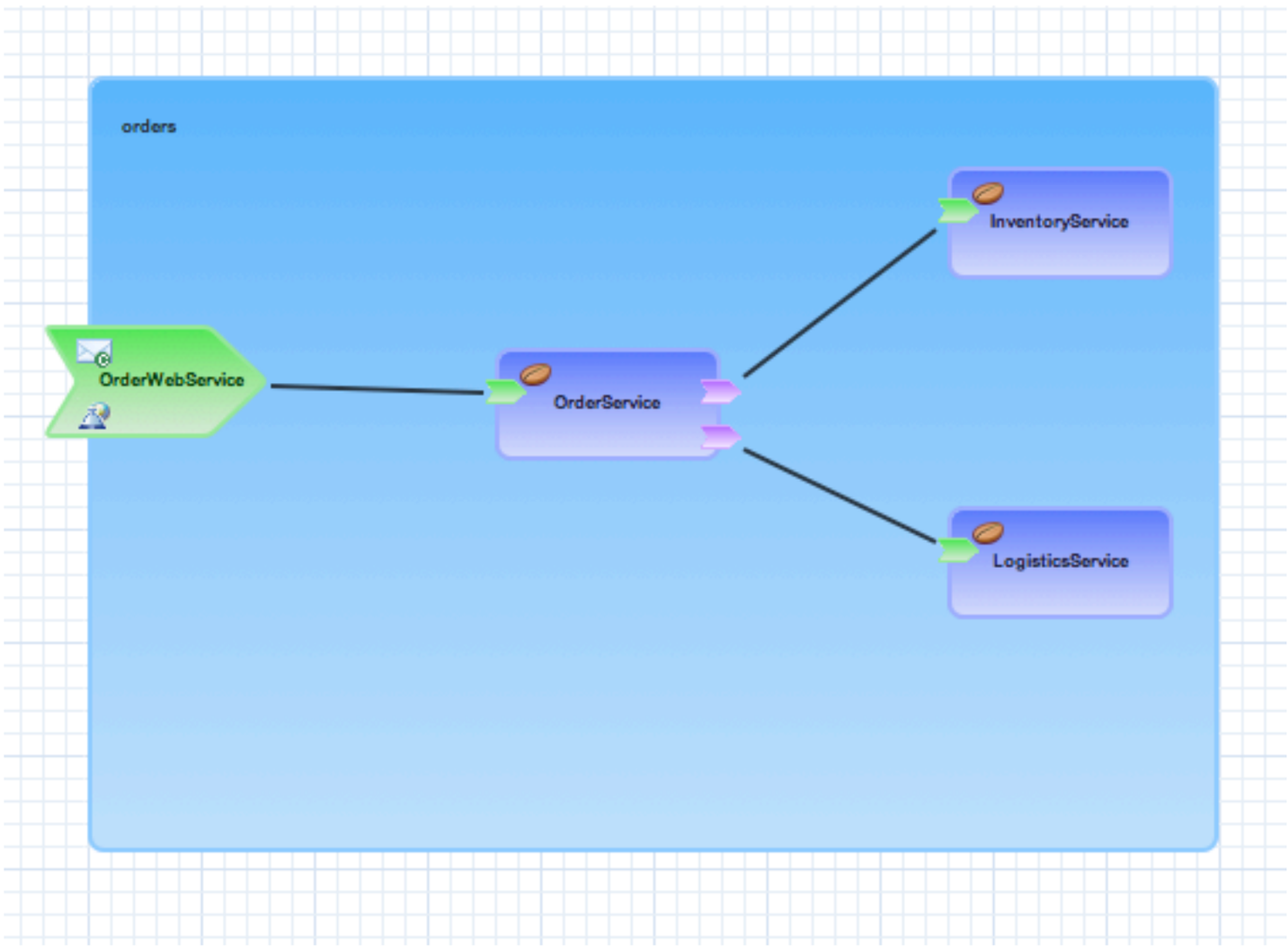
Runtime Governance

- Real Time Analysis of Business Transactions
  - Customisable Analysis and Presentation
- Policy Enforcement
- Historical Analysis
  - Business Reporting

Event Processors

- Synchronous or Asynchronous
- Monitor SLAs
- Trace Execution
- Enforce Business Policies
  - Block Transaction
- Extract Correlation Information
- Implement using Java, Drools and MVEL

Demo Application



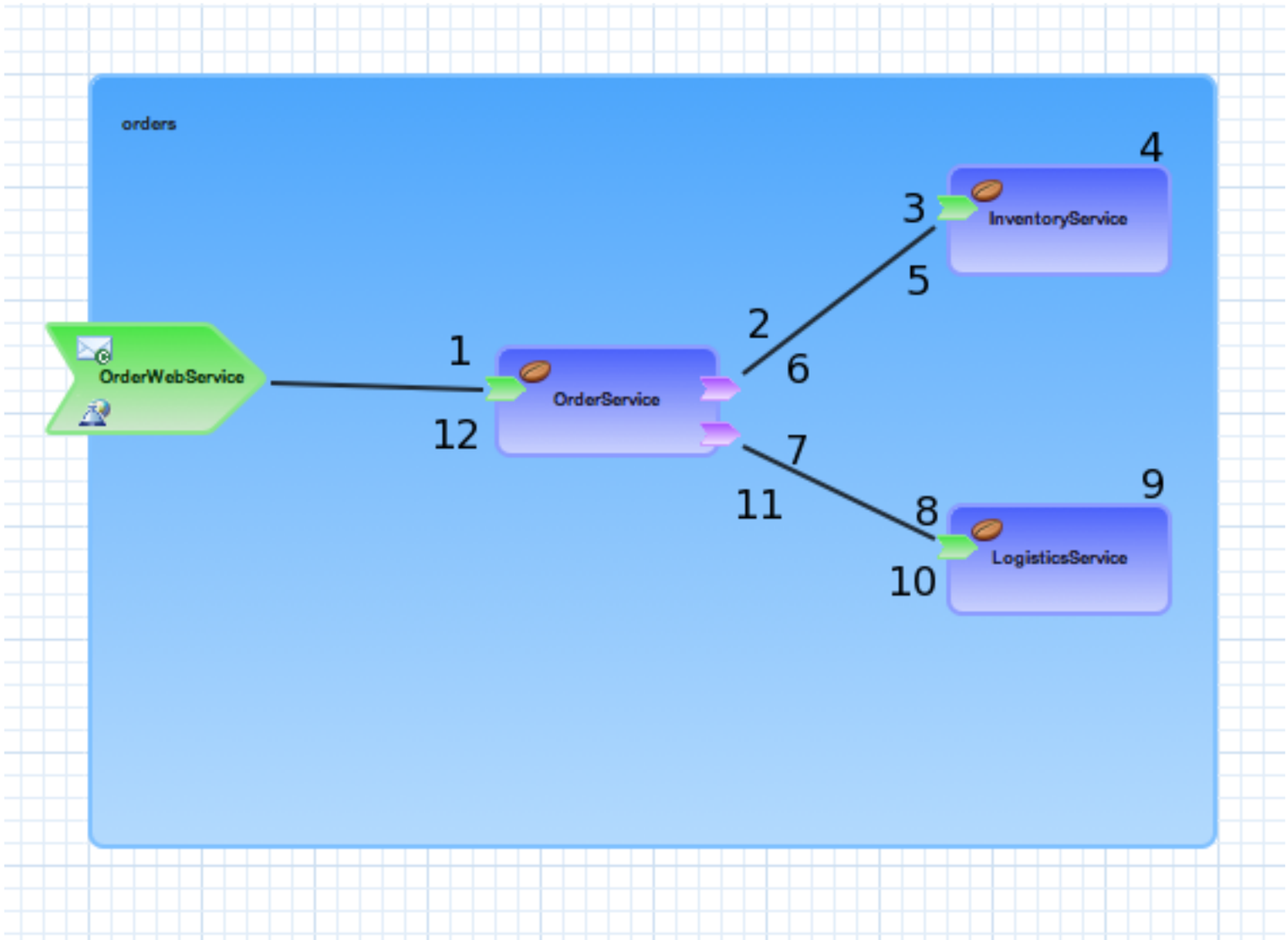
Services

- OrderService
  - submitOrder
  - makePayment
- InventoryService
  - lookupItem
- LogisticsService
  - deliver

Policies

- Credit Assessment
  - Suspend account if exposure > $150
- Monitor SLA
  - Critical - 400 ms,
  - High - 320 ms
  - Medium - 260 ms
  - Low - 200ms
- Limit Invocations
  - One every two seconds

Events



- RequestReceived
  - 1, 3, 8
- RequestSent
  - 2, 7
- ResponseSent
  - 5, 10, 12
- ResponseReceived
  - 6, 11
- LogMessage
  - 4, 9

RED HAT SUMMIT

Demo

redhat.

More SOA 6 @ Summit

- Sessions
  - Loosely Coupled, Reusable & Interoperable : Integration Overview & Roadmap*

- Labs
  - Getting Started with Red Hat JBoss SOA Platform 6*
  - Migrating Applications from Red Hat JBoss SOA Platform 5 to 6
  - Choose Your Own Adventure : Getting Started with Red Hat JBoss Middleware

* catch the recordings or rent a time machine

RED HAT SUMMIT

Questions?

redhat.