# JBoss Community

SWITCHYARD

# Enterprise Service Development

JBoss Community

# Agenda

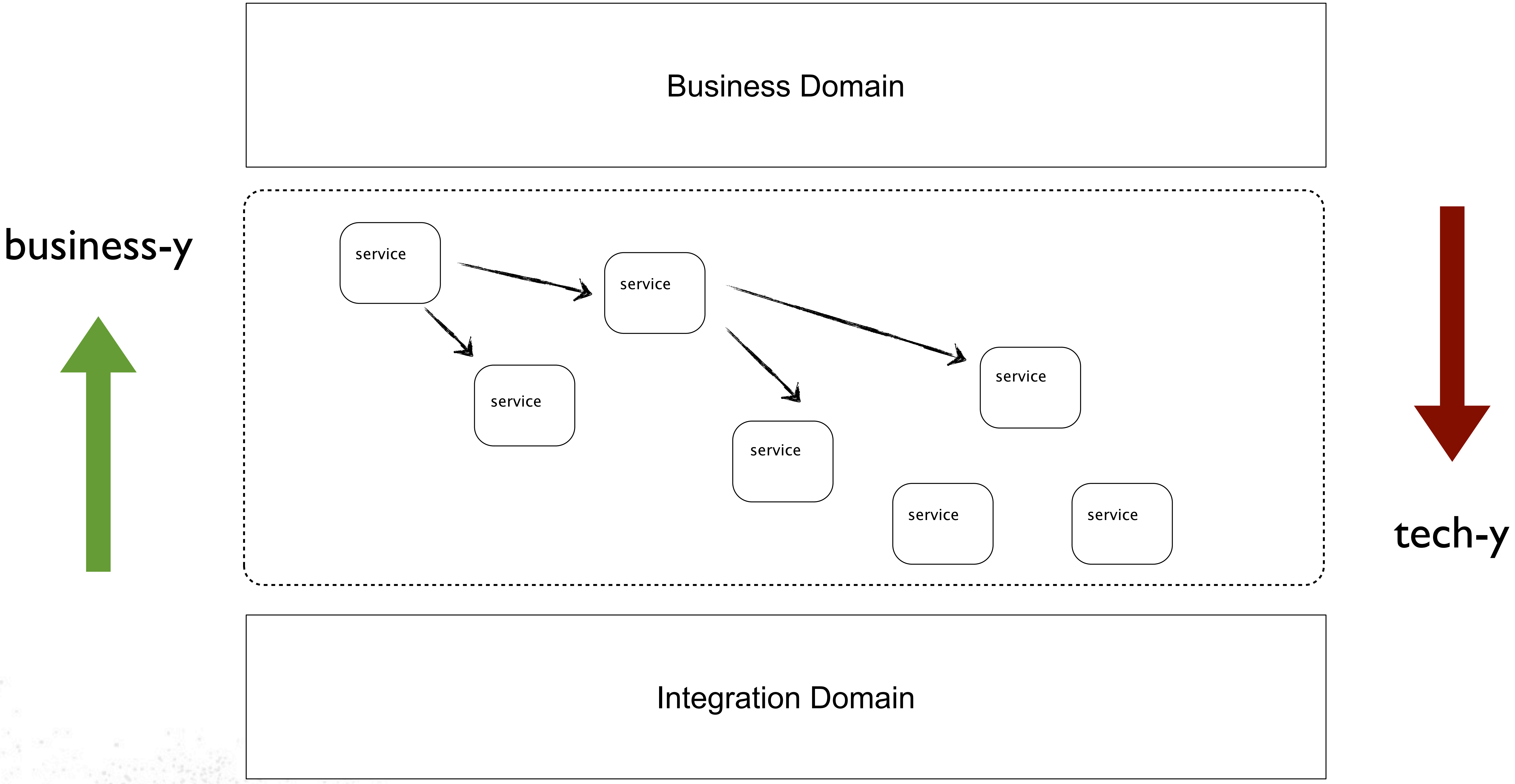
- Enterprise Services
- Decision Services
- BPM Services
- Administration
- Q & A

JBoss Community

# Enterprise Services

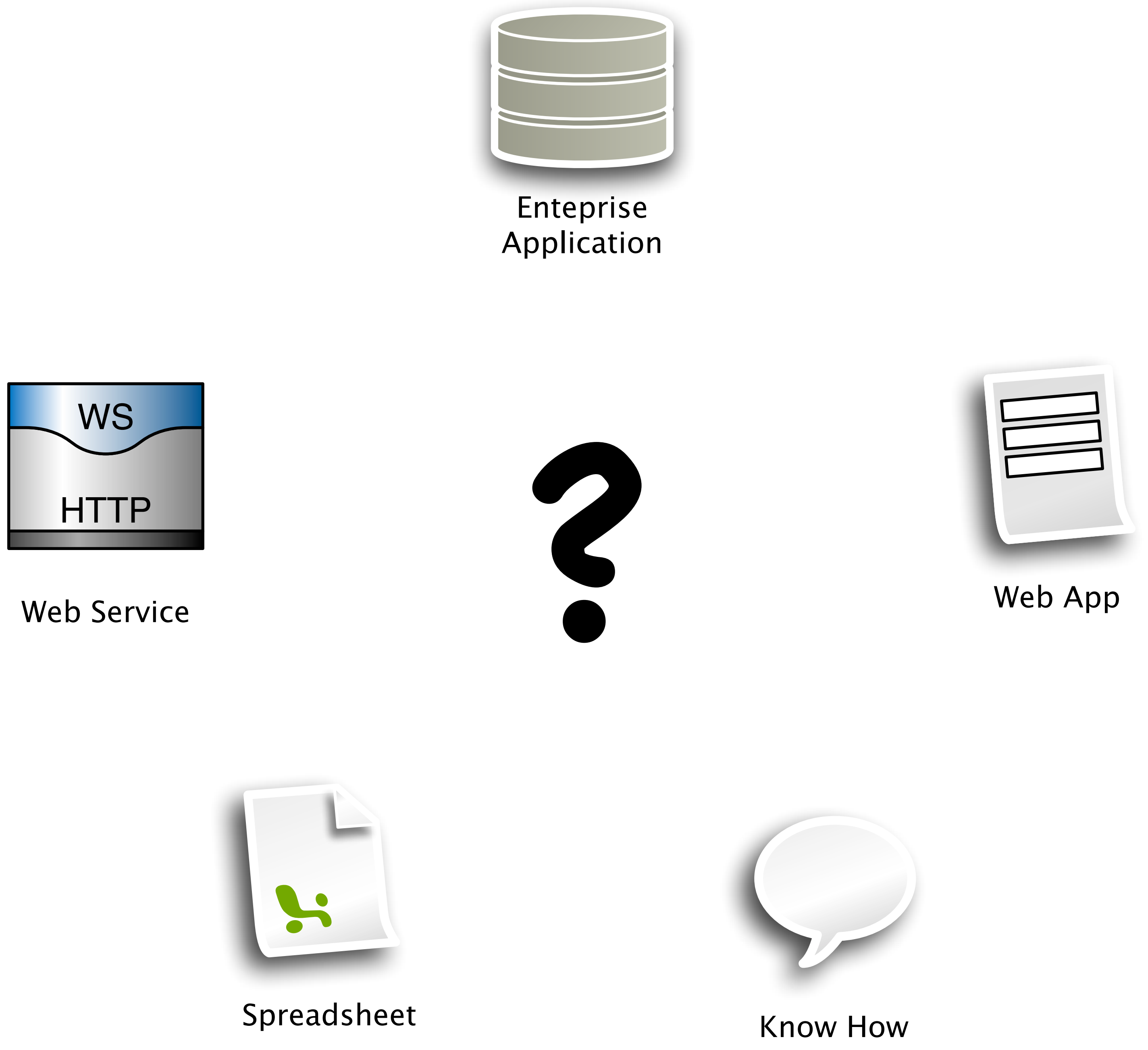
- Business-y
- Composed and Composable
- Loosely Coupled
- Scaleable, manageable, monitorable, versionable, interoperable, unbreakable, unstoppable ... oh, just shoot me now

# This Is What I’m Talking About



# Decision Services

# Decisions are Everywhere



*"When remainder < .01 move remainder to Swiss bank account"*

Enteprise Application

*"Orders > $1M get priority processing"*

Web Service

Web App

*"Users in role reviewer can see submissions"*

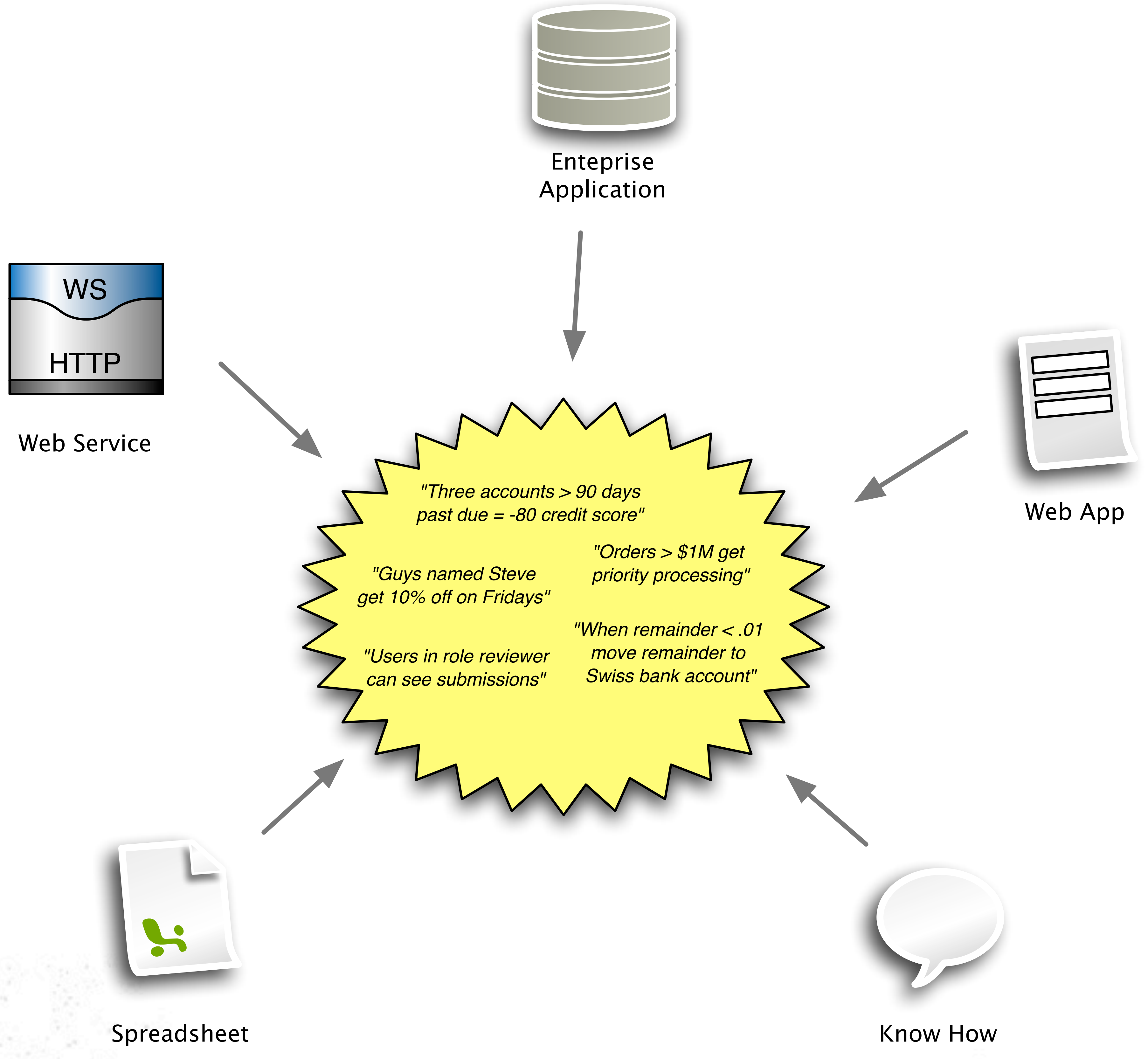
Spreadsheet

Know How

*"Three accounts > 90 days past due = -80 credit score"*

*"Guys named Steve get 10% off on Fridays"*

# All Your RuleBase Are Belong In One Place



Enteprise Application

Web Service

Web App

Spreadsheet

Know How

"Three accounts > 90 days past due = -80 credit score"

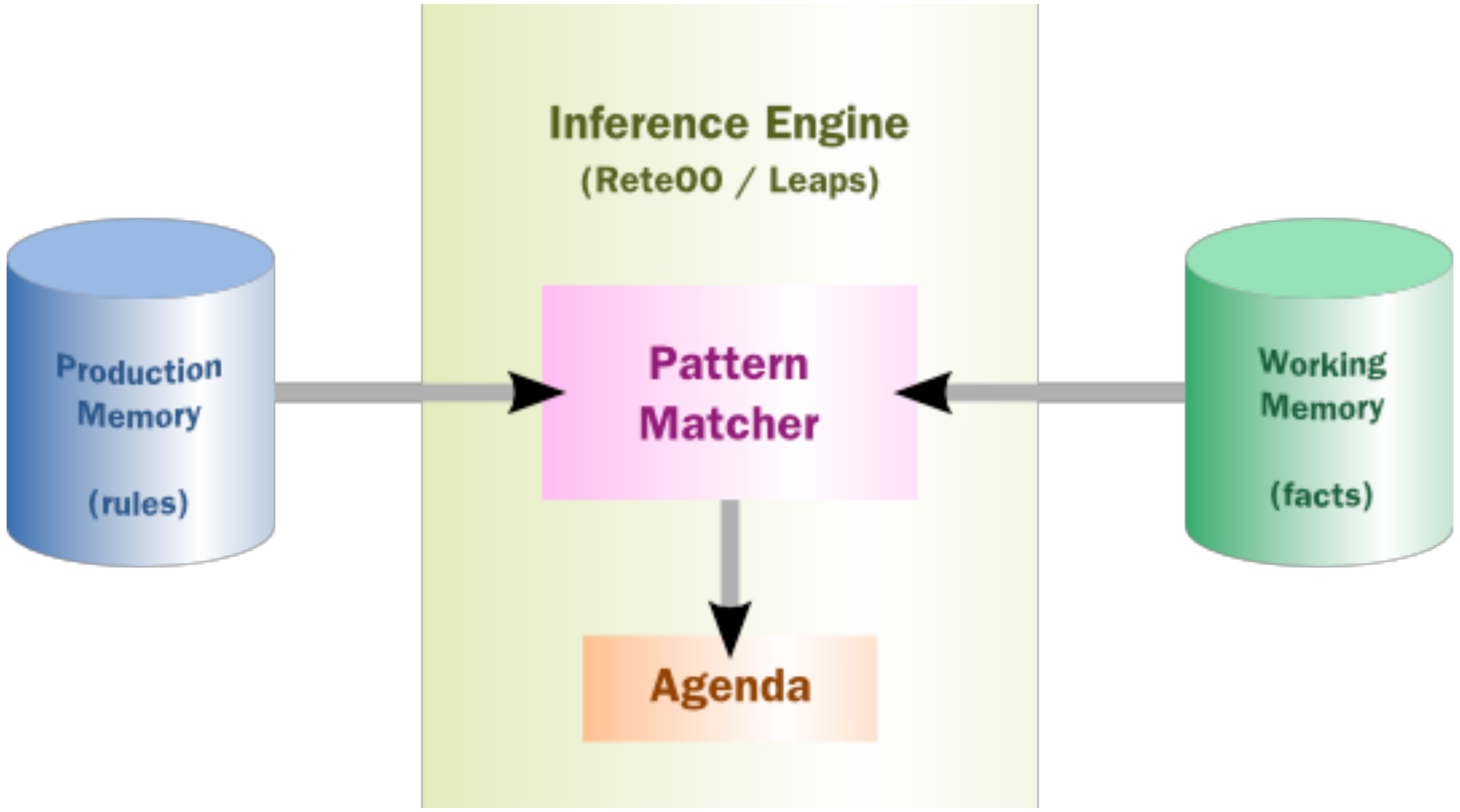
"Orders > $1M get priority processing"

"Guys named Steve get 10% off on Fridays"

"When remainder < .01 move remainder to Swiss bank account"

"Users in role reviewer can see submissions"

# Drools Expert



JBoss Community

# Facts

```
public class Applicant {

    private String name;
    private int age;
    private boolean valid;

    // getter and setter methods here
}
```

or ...

```
declare Applicant
    name  : String
    age   : int
    valid : boolean
end
```

# Rules

```
package com.company.license

rule "Is of valid age"
when
    $a : Applicant( age < 18 )
then
    $a.setValid( false );
end
```

# Runtime

```
KnowledgeBuilder kbuilder = KnowledgeBuilderFactory.newKnowledgeBuilder();

kbuilder.add( ResourceFactory.newClassPathResource(
         "licenseApplication.drl", getClass() ), ResourceType.DRL );

if ( kbuilder.hasErrors() ) {
    System.err.println( builder.getErrors().toString() );
}

kbase.addKnowledgePackages( kbuilder.getKnowledgePackages() );
StatelessKnowledgeSession ksession = kbase.newStatelessKnowledgeSession();

Applicant applicant = new Applicant( "Mr John Smith", 16 );
assertTrue( applicant.isValid() );
ksession.execute( applicant );
assertFalse( applicant.isValid() );
```

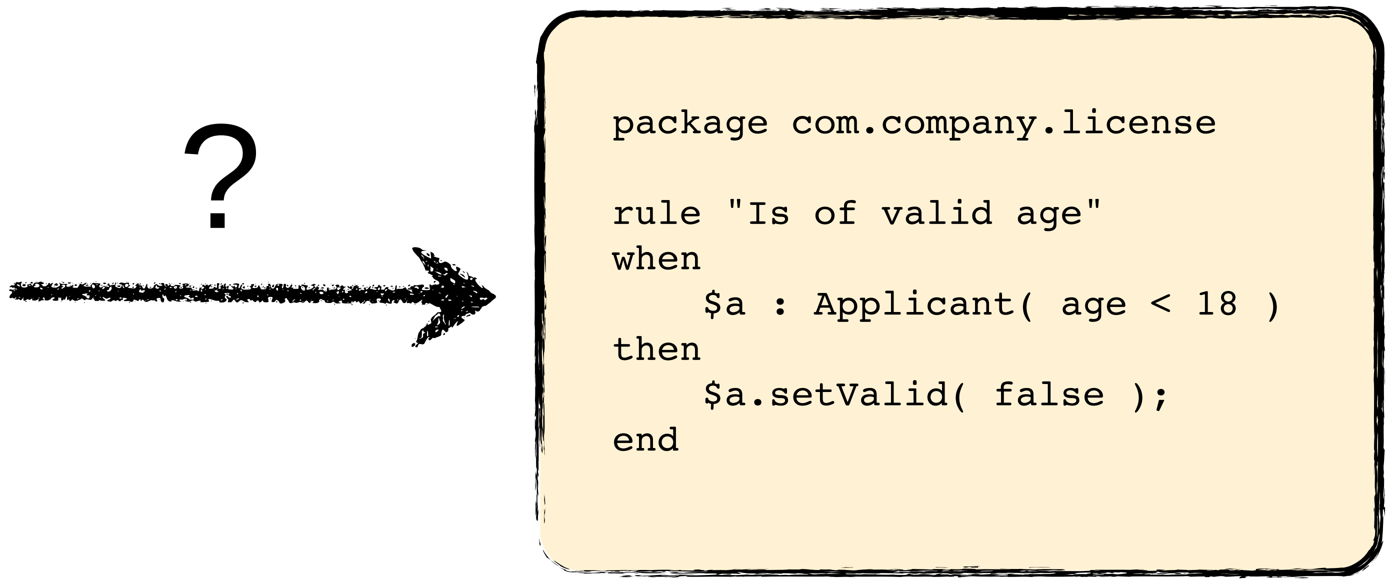
# Rules Component

- Allows Business Rules to be exposed as Decision Services within SwitchYard
- Based on Drools
- Provides
  - Bootstrap of Knowledge Runtime and Session
  - Explicit contract for decision service
  - Binding agnostic fact insertion
  - Data format isolation

# Bootstrap

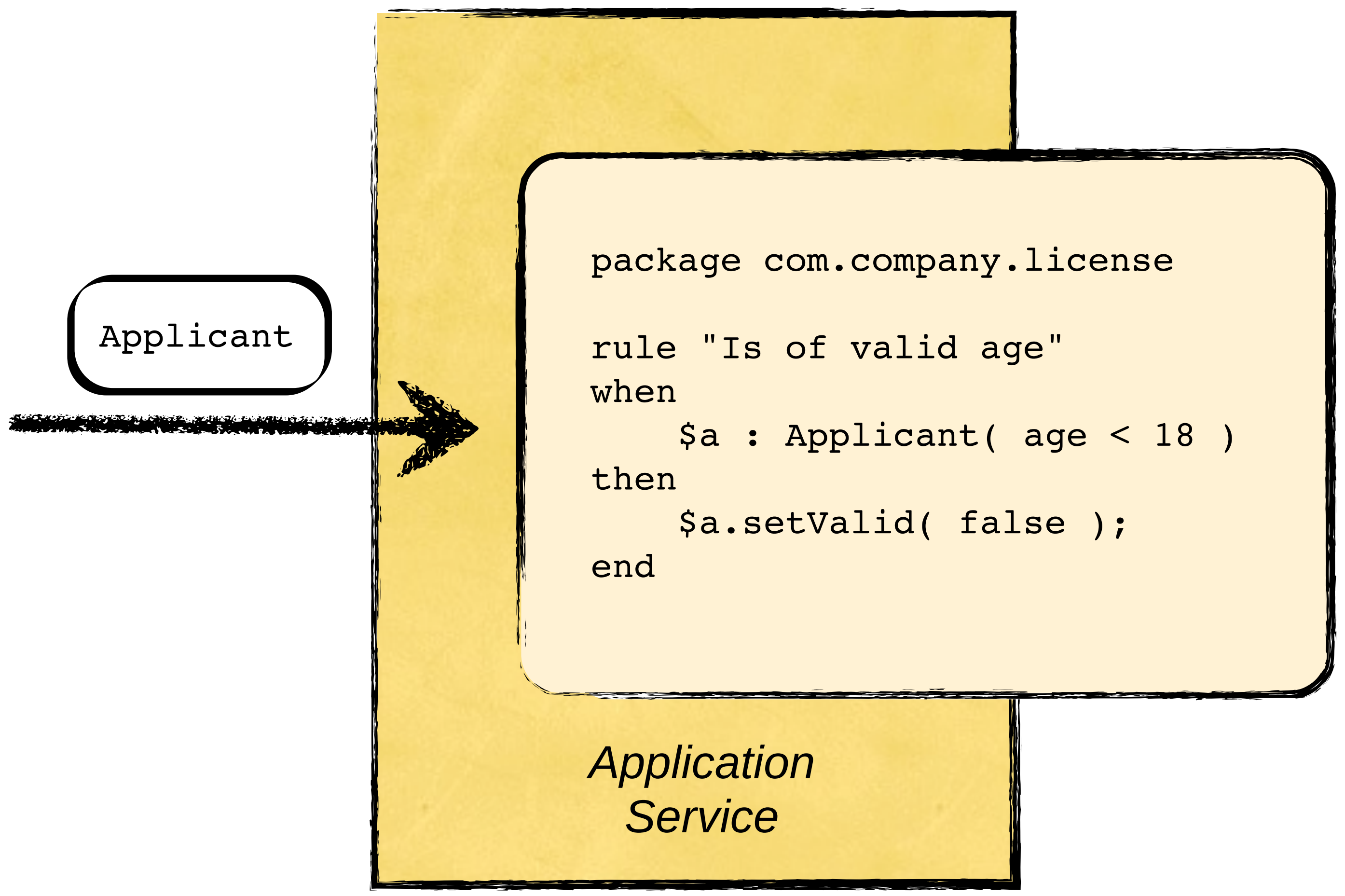
```
KnowledgeBuilder kbuilder = KnowledgeBuilderFactory.newKnowledgeBuilder();

kbuilder.add( ResourceFactory.newClassPathResource(
         "licenseApplication.drl", getClass() ), ResourceType.DRL );

if ( kbuilder.hasErrors() ) {
    System.err.println( builder.getErrors().toString() );
}

kbase.addKnowledgePackages( kbuilder.getKnowledgePackages() );
StatelessKnowledgeSession ksession = kbase.newStatelessKnowledgeSession();

Applicant applicant = new Applicant( "Mr John Smith", 16 );
assertTrue( applicant.isValid() );
ksession.execute( applicant );
assertFalse( applicant.isValid() );
```

# Service Contract



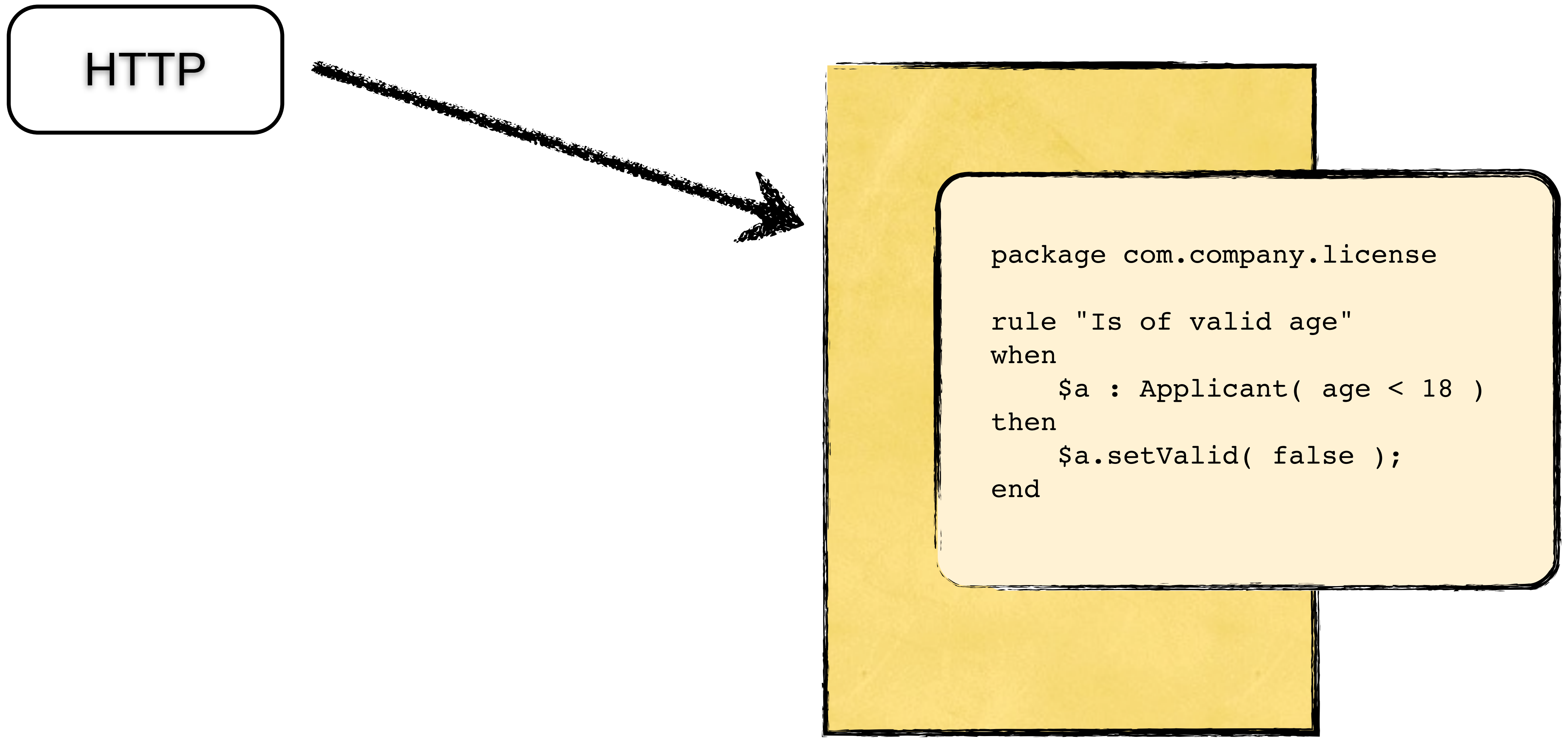
```
package com.company.license

rule "Is of valid age"
when
    $a : Applicant( age < 18 )
then
    $a.setValid( false );
end
```

# Service Contract



```
package com.company.license

rule "Is of valid age"
when
    $a : Applicant( age < 18 )
then
    $a.setValid( false );
end
```

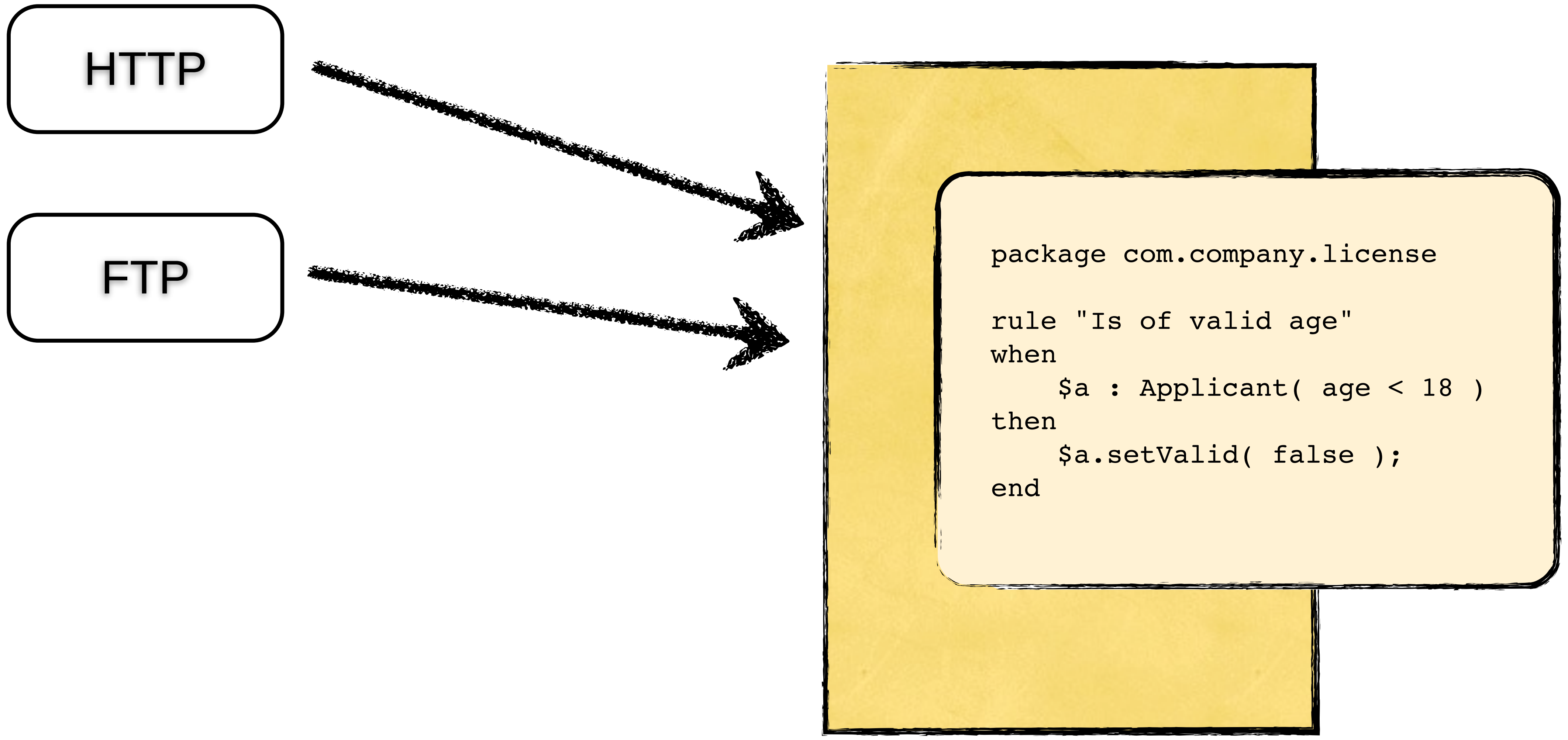
# Binding Agnostic



HTTP

```
package com.company.license

rule "Is of valid age"
when
    $a : Applicant( age < 18 )
then
    $a.setValid( false );
end
```

# Binding Agnostic



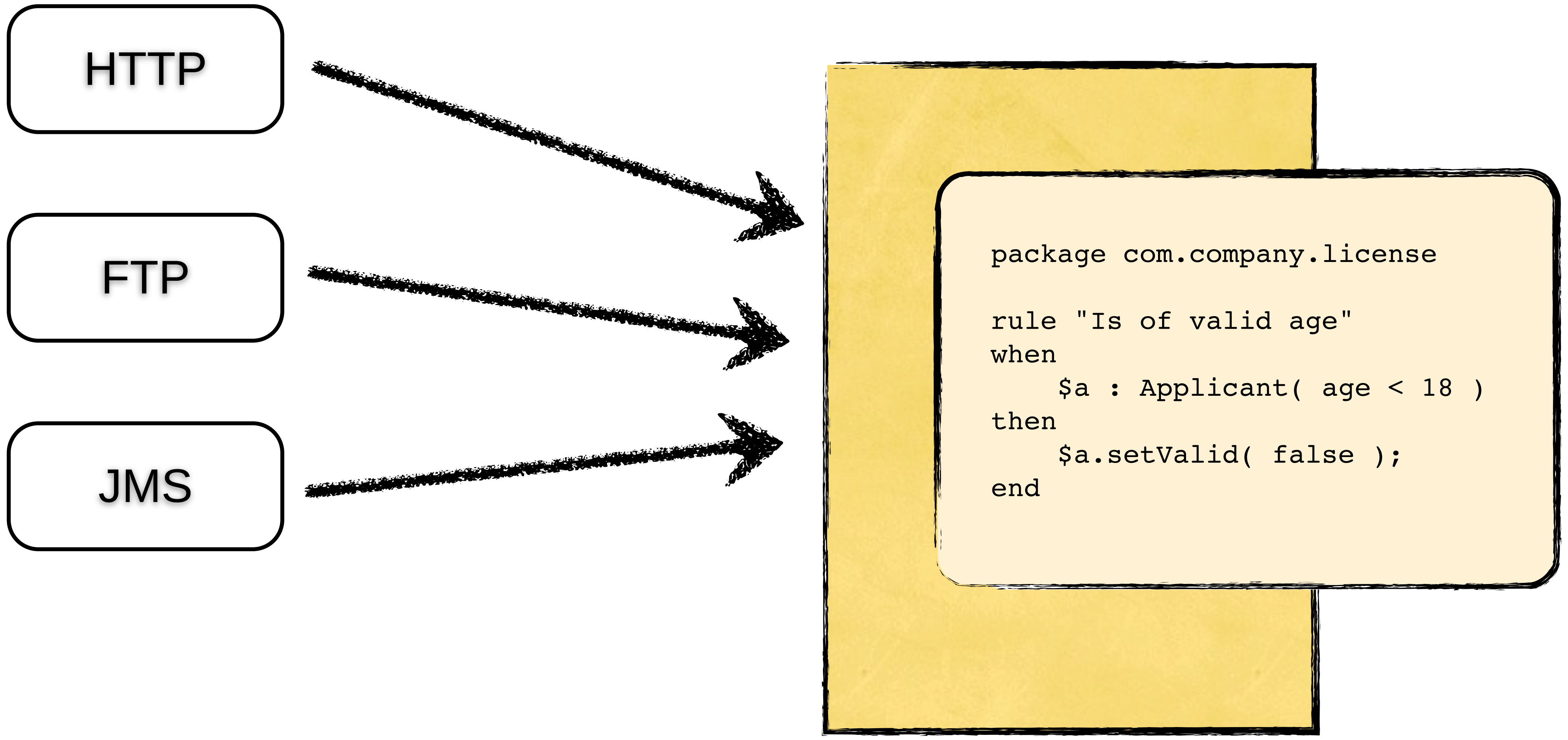
HTTP

FTP

```
package com.company.license

rule "Is of valid age"
when
    $a : Applicant( age < 18 )
then
    $a.setValid( false );
end
```

JBoss Community

# Binding Agnostic



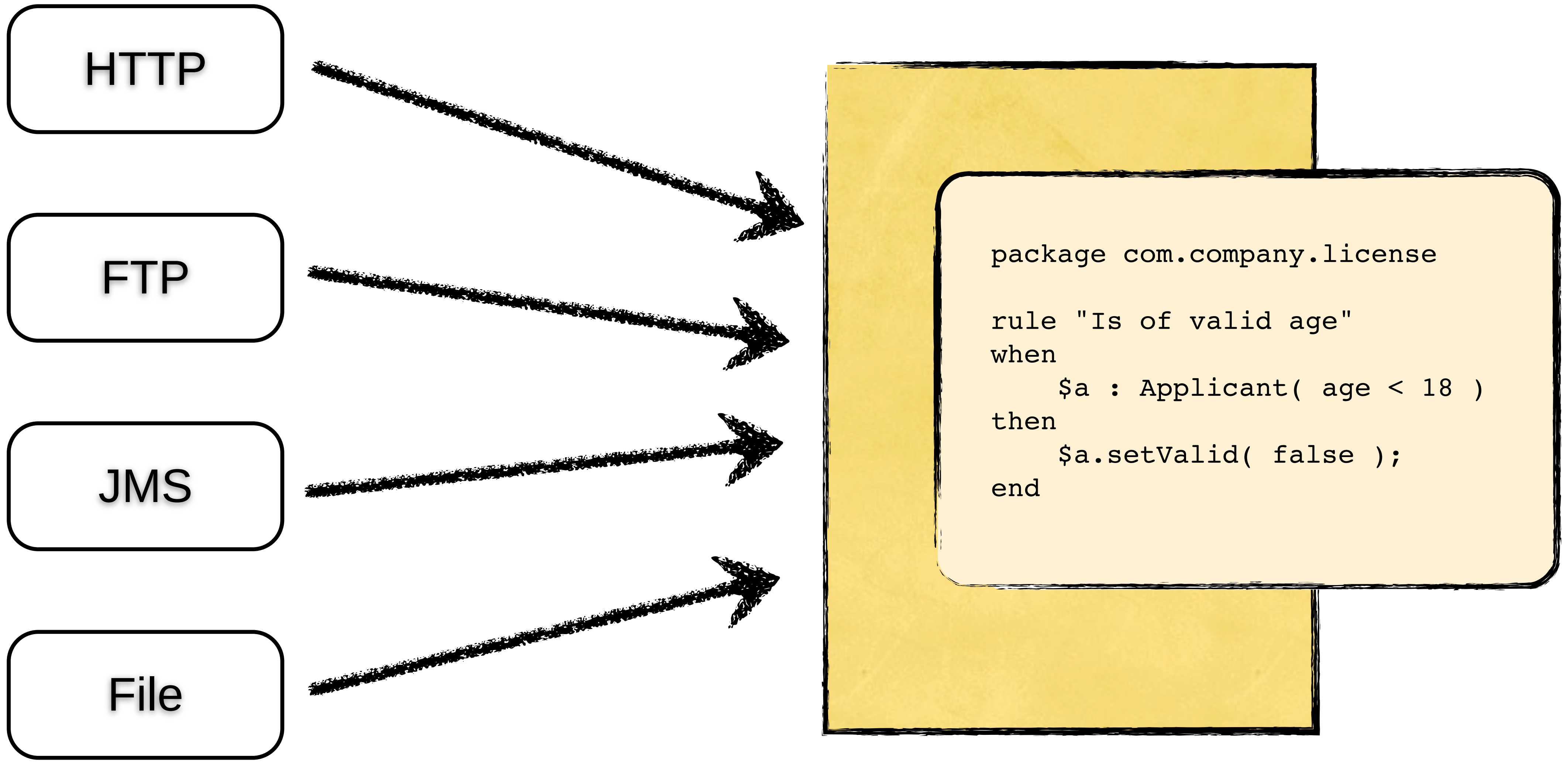
HTTP

FTP

JMS

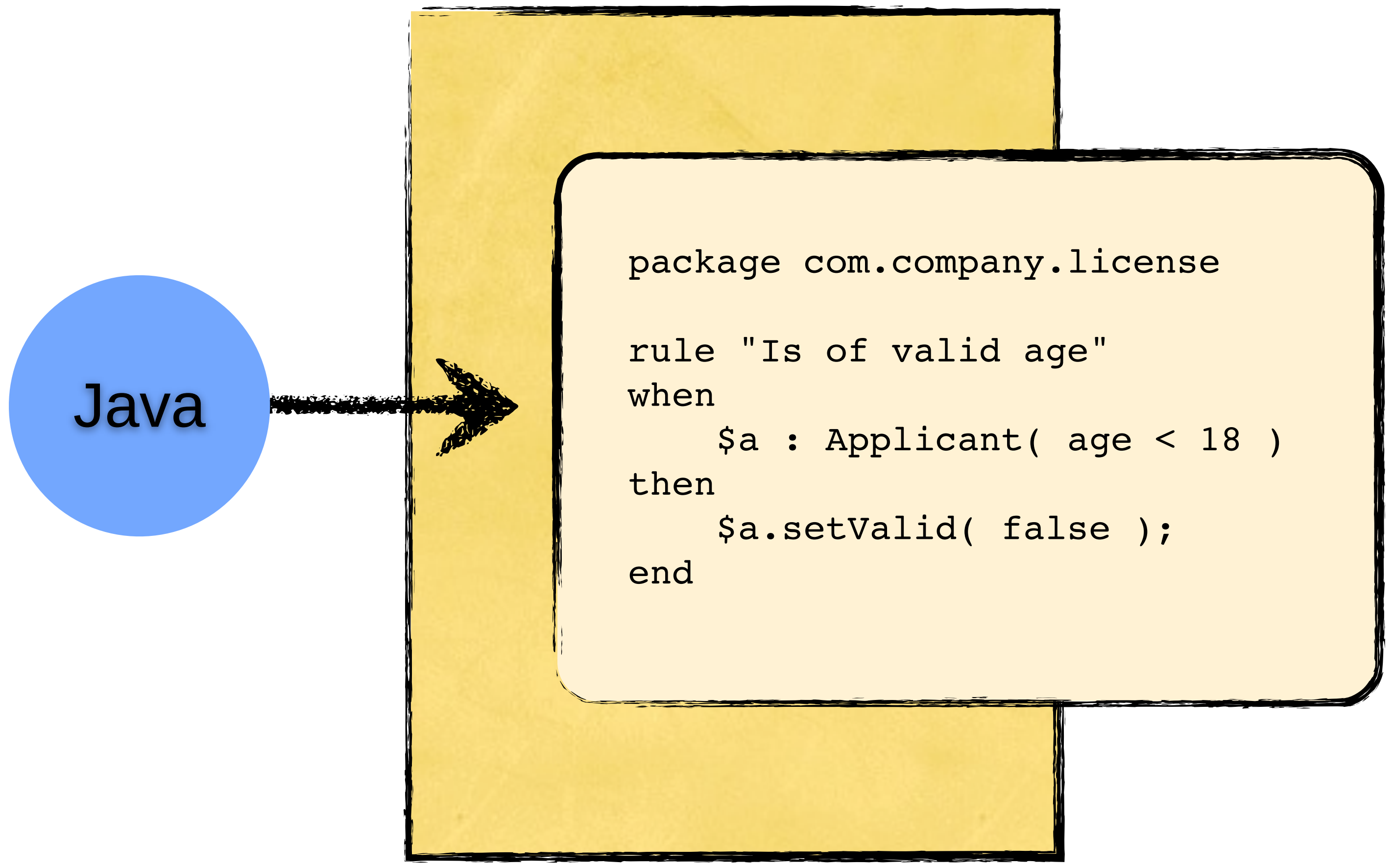
```
package com.company.license

rule "Is of valid age"
when
    $a : Applicant( age < 18 )
then
    $a.setValid( false );
end
```

# Binding Agnostic



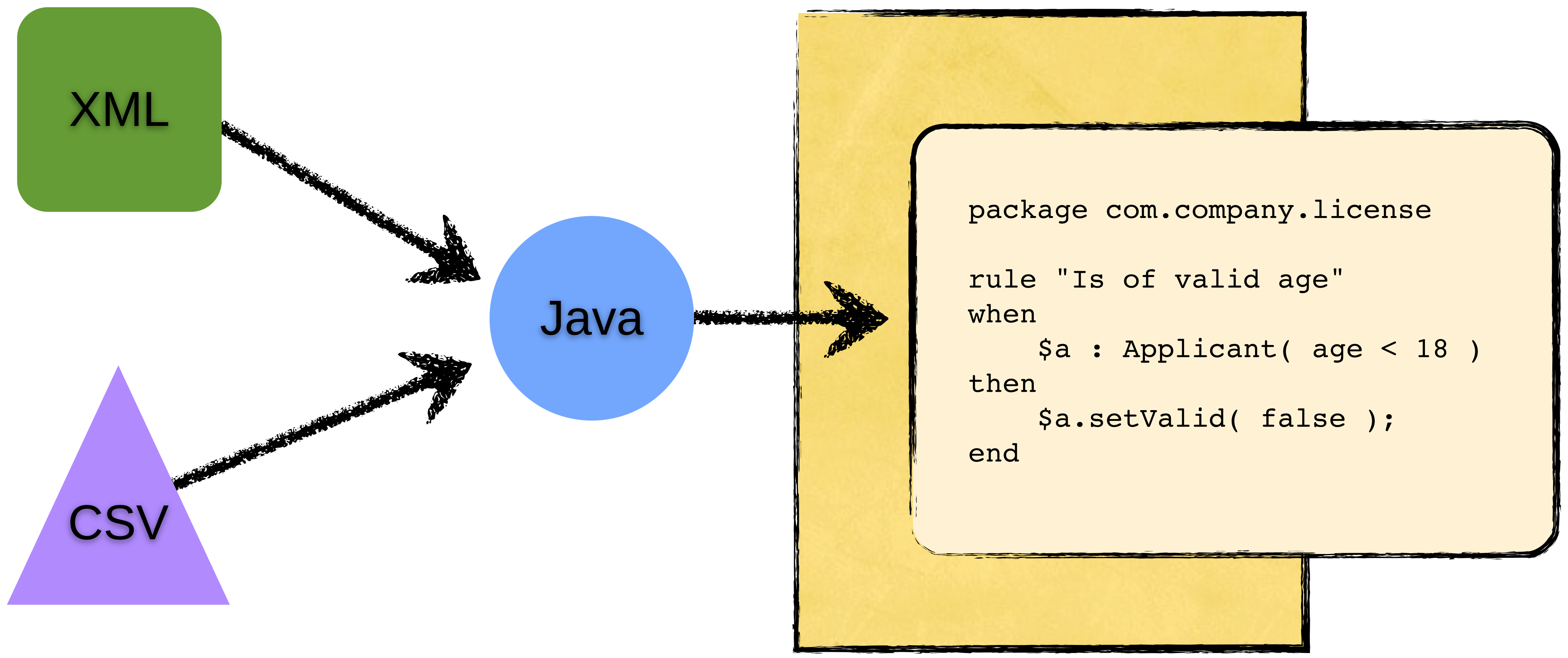
```
package com.company.license

rule "Is of valid age"
when
    $a : Applicant( age < 18 )
then
    $a.setValid( false );
end
```

# Data Format Isolation



```
package com.company.license

rule "Is of valid age"
when
    $a : Applicant( age < 18 )
then
    $a.setValid( false );
end
```

# Data Format Isolation



```
package com.company.license

rule "Is of valid age"
when
    $a : Applicant( age < 18 )
then
    $a.setValid( false );
end
```

# An Example

interview.drl

```
package org.switchyard.quickstarts.rules.interview

rule "Is of valid age"
    when
        $a : Applicant( age > 17 )
    then
        $a.setValid( true );
end

rule "Is not of valid age"
    when
        $a : Applicant( age < 18 )
    then
        $a.setValid( false );
end
```

Interview.java

```
public interface Interview {
    public void verify(Applicant applicant);
}
```

switchyard.xml

```
<implementation.rules stateful="false">
   <rulesAction name="verify" type="EXECUTE_RULES"/>
   <resource
      location="/org/switchyard/quickstarts/rules/interview/Interview.drl"
      type="DRL"/>
</implementation.rules>
```

# Message vs. Content

- Message and content are inserted into Knowledge Session
- Disadvantages of using Message
  - Rules are coupled to SwitchYard API
- Advantages of using Message
  - Easily convert types using `Message.getContent(Class<T>)`
  - Access to attachments

JBoss Community

# Drools Support

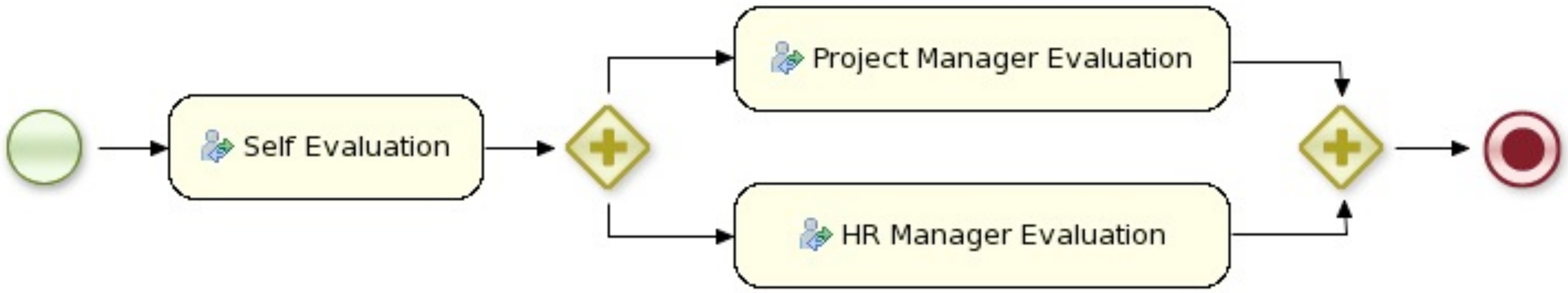
- Current
  - Drools 5.2.0
  - Stateless execution
  - Stateful execution with continue and dispose
  - Audit Logging
  - Forge tooling
- Planned
  - Drools Fusion for CEP
  - Changeset
  - Agent / Repository interaction

# BPM Services

JBoss Community

# BPM

## Business Process Management



A ***business process*** is a process that describes the order in which a series of steps need to be executed, using a flow chart.

# Compositional Models

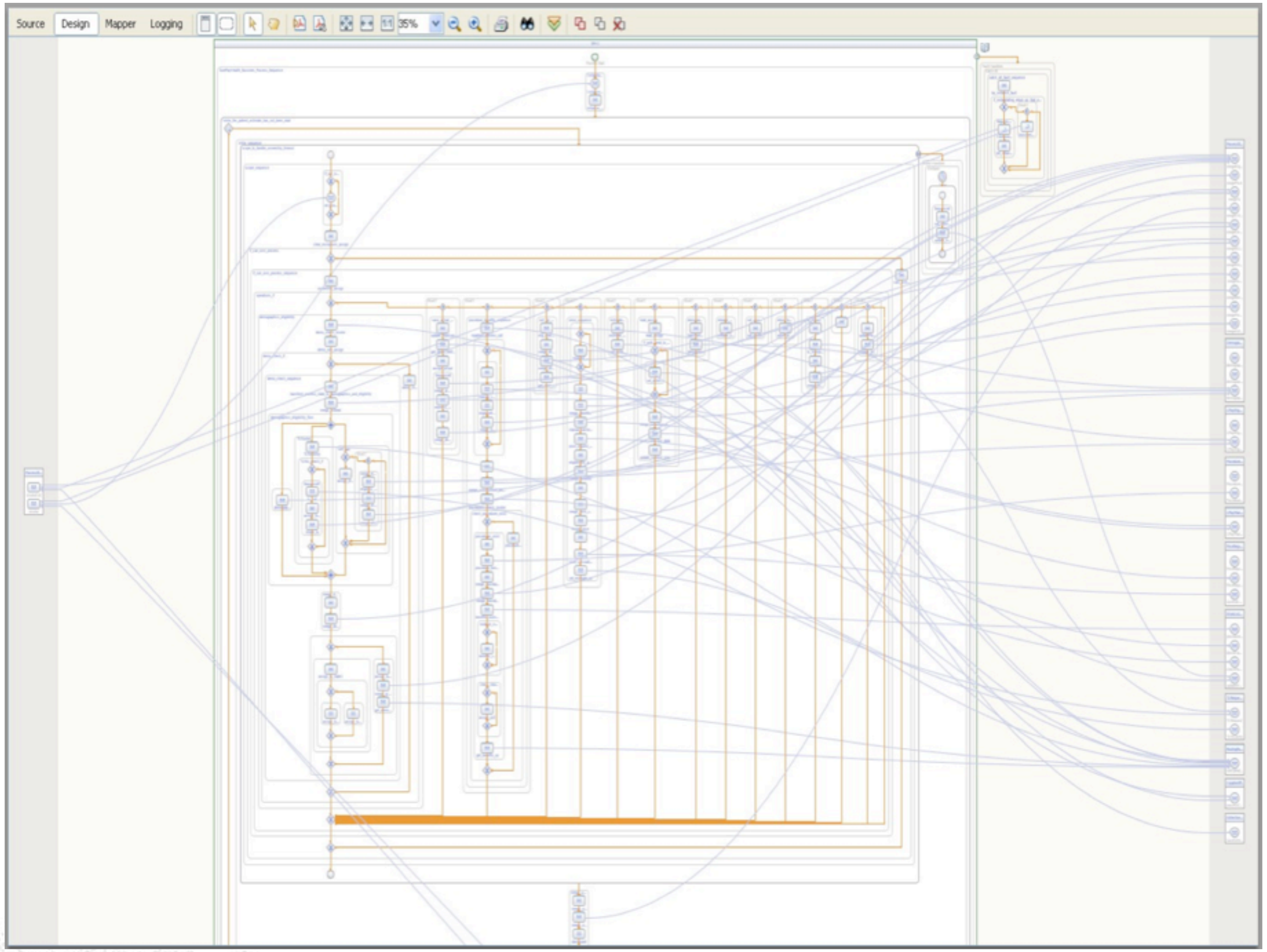
- Direct Reference
- Pipeline Execution
- Orchestration

# Compositional Models

- Direct Reference
- Pipeline Execution
- Orchestration

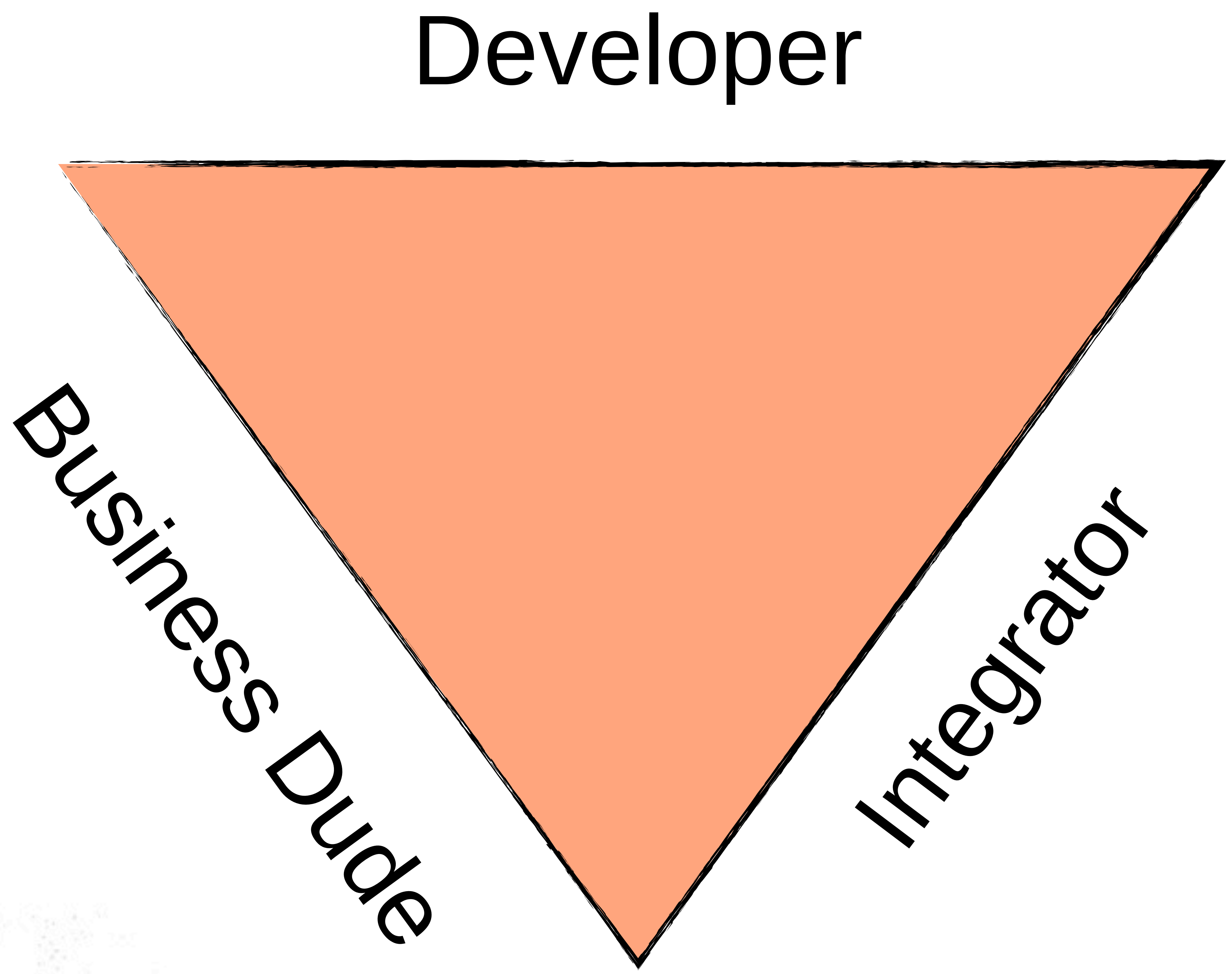
this

# When Good Composition Goes Bad



# What Composition and When?

- Who’s asking?



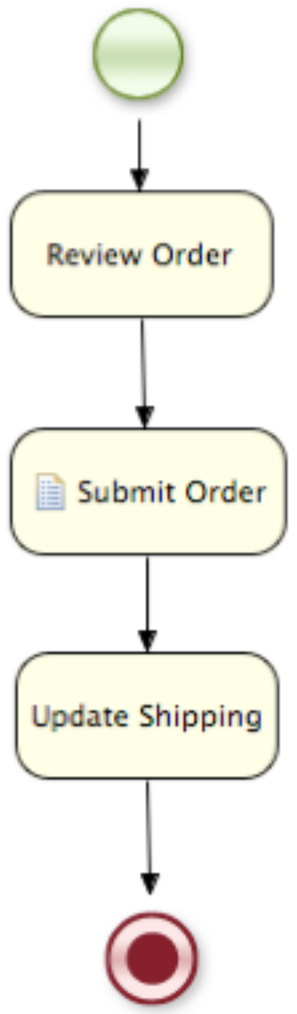
# What Composition and When?

- What are you trying to do?
  - Make business analyst’s head explode
  - Parallel activities
  - Long-lived transactions
  - Compensation
  - Human workflow
  - Activity monitoring

# BPM Component

- Provides business process and human workflow support
- Based on jBPM 5
- Same advantages as Drools
  - Bootstrap of runtime
  - Explicit contract for business processes
  - Binding agnostic interaction
  - Data format isolation
  - ++

# Example Workflow



# Workflow as a Service

- Drive workflow state based on service invocation
- Service contract defines state mappings
  - Start process
  - Signal process
  - Terminate process

JBoss Community

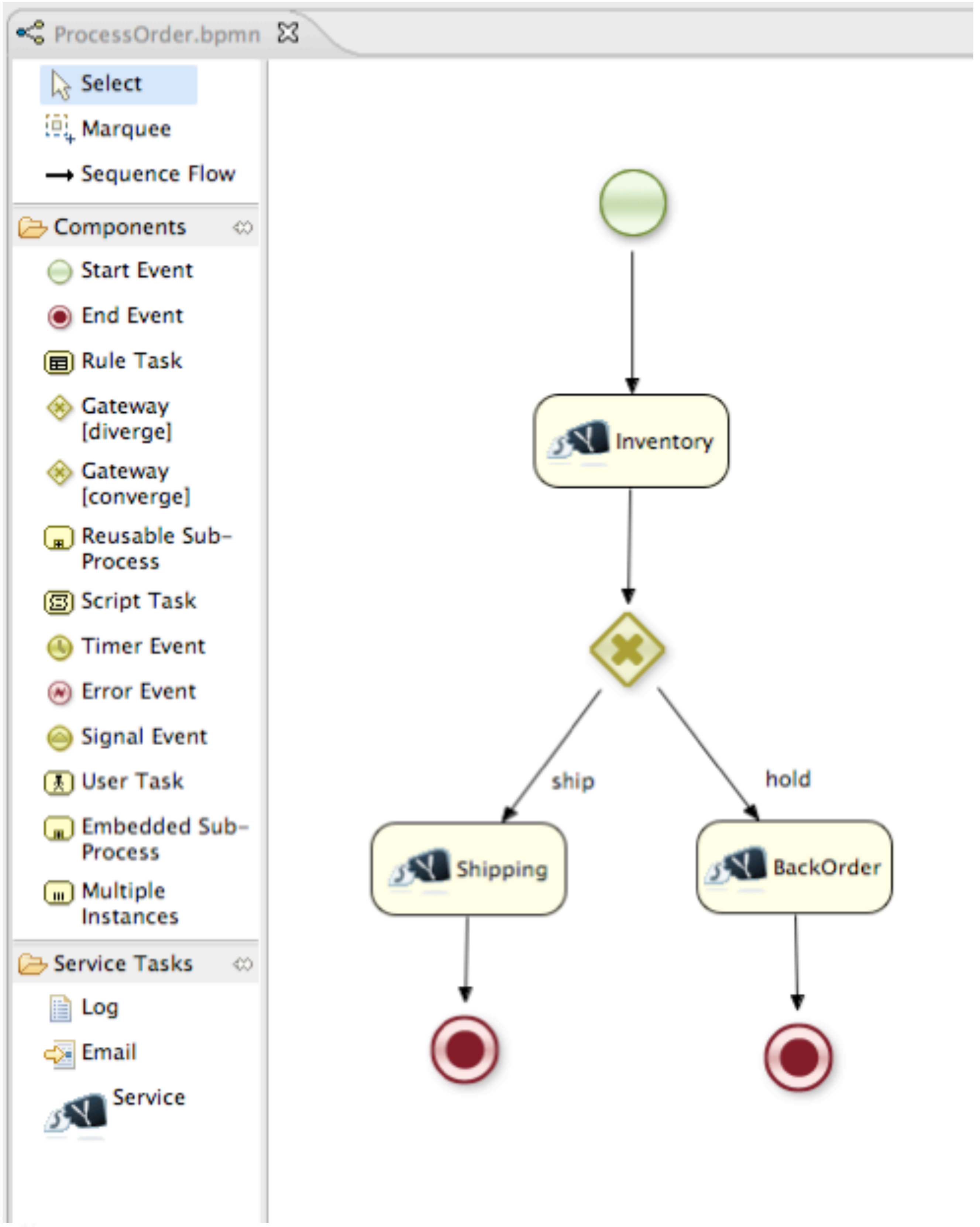
# Invocation Mapping

```
@Process(OrderService.class)
public interface OrderProcess extends OrderService {

    @StartProcess
    void start(Order order);

    @SignalEvent("shipped")
    public void orderShipped(ShipNotice shipping);

    @AbortProcessInstance
    public void cancelOrder(String reason);
}
```

# Invoking a Service

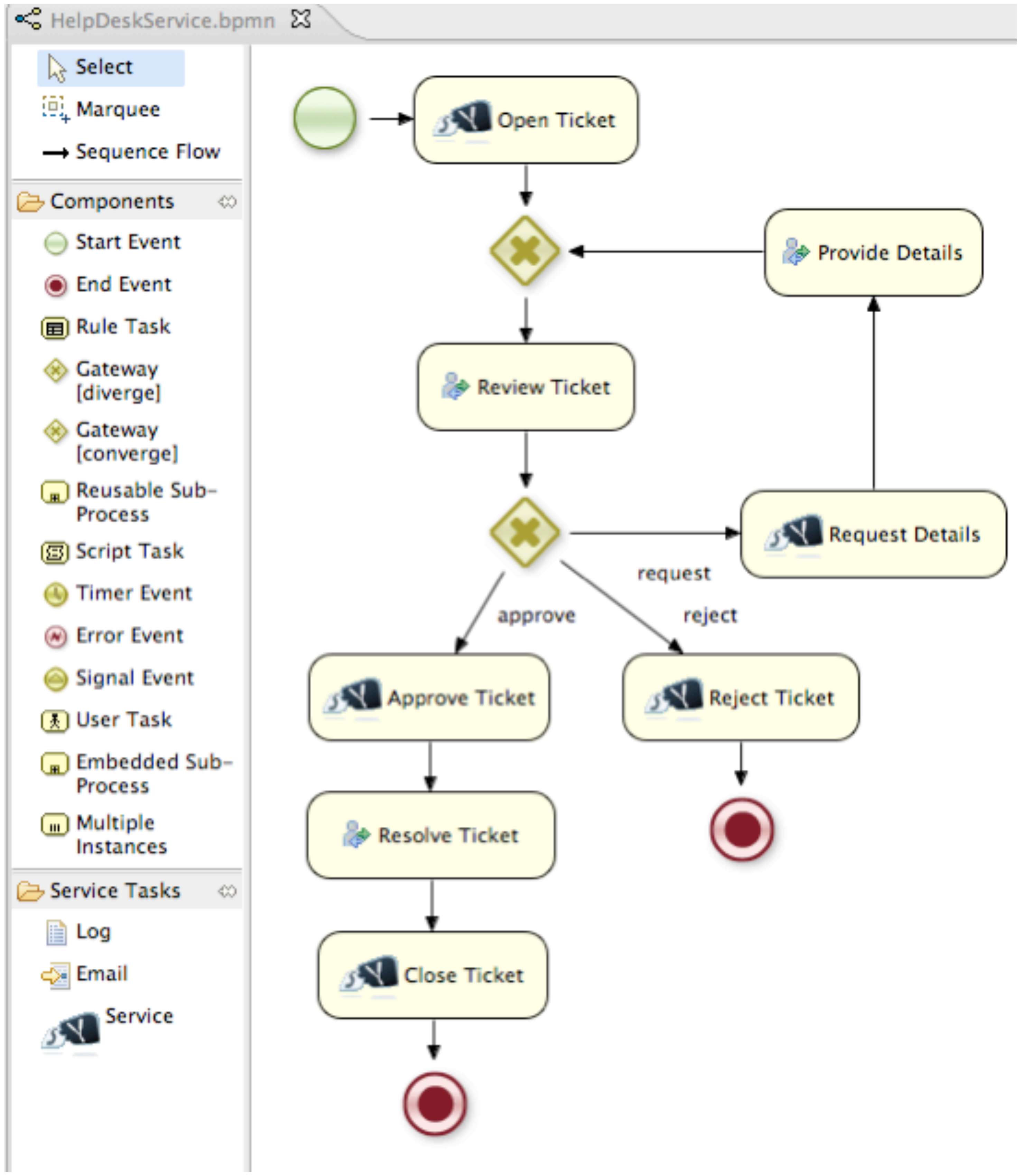
- Native integration in BPMN2 modeler
- Based on service name and contract
  - No location/binding details
- Flexible mapping between process variables and service messages
- Data format issues covered by declarative transformation

# Service Orchestration



JBoss Community

# Integrated Workflow



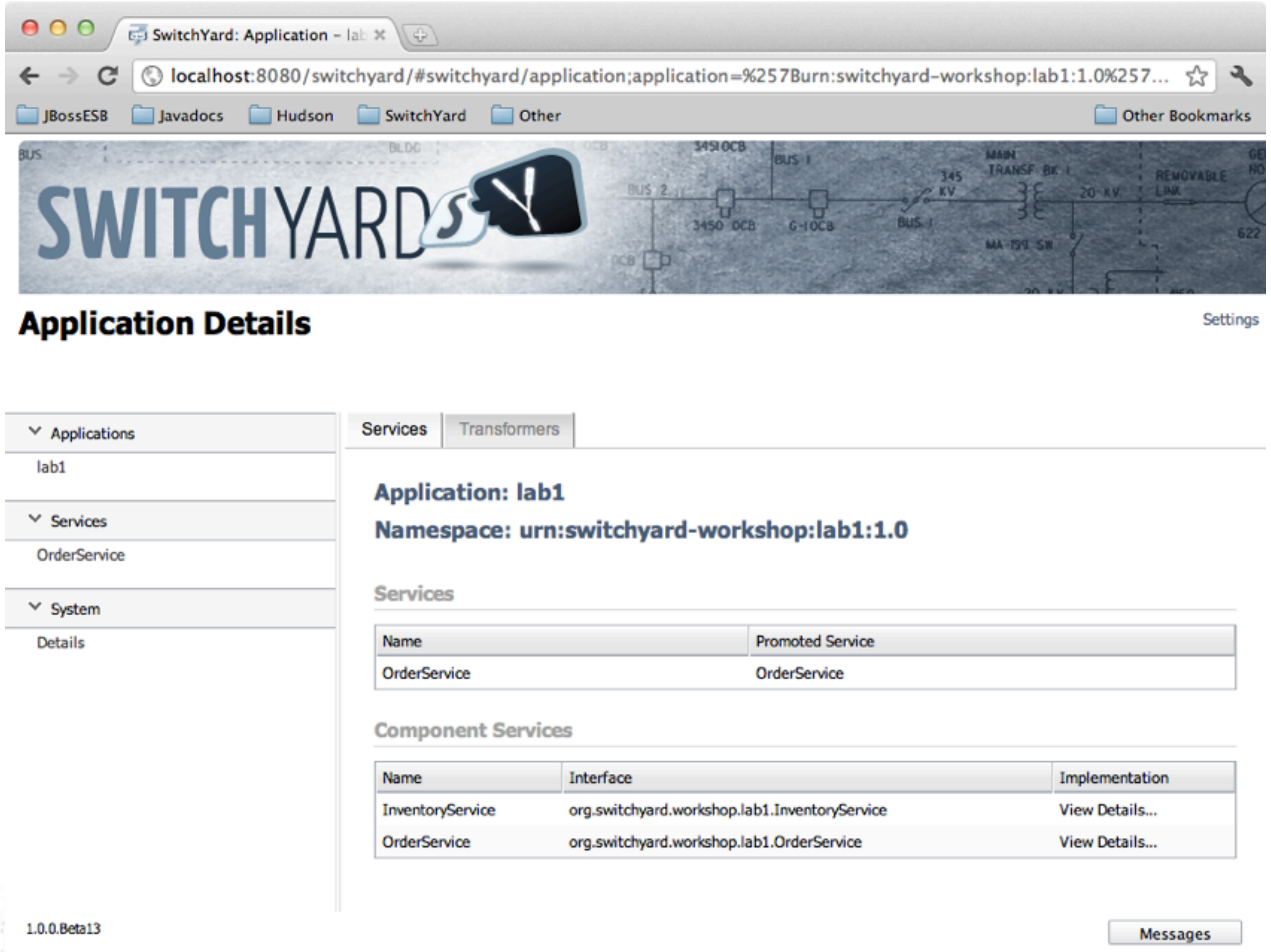
JBoss Community

# Administration

# Admin Console

- Preliminary version in 0.2
- View application details
  - Service implementations
  - Promoted services
  - Bindings
  - Declared transformations
  - References
- Much bigger plans here
  - Monitoring
  - Repository integration

# Application View



## Application Details

### Application: lab1

### Namespace: urn:switchyard-workshop:lab1:1.0

Services

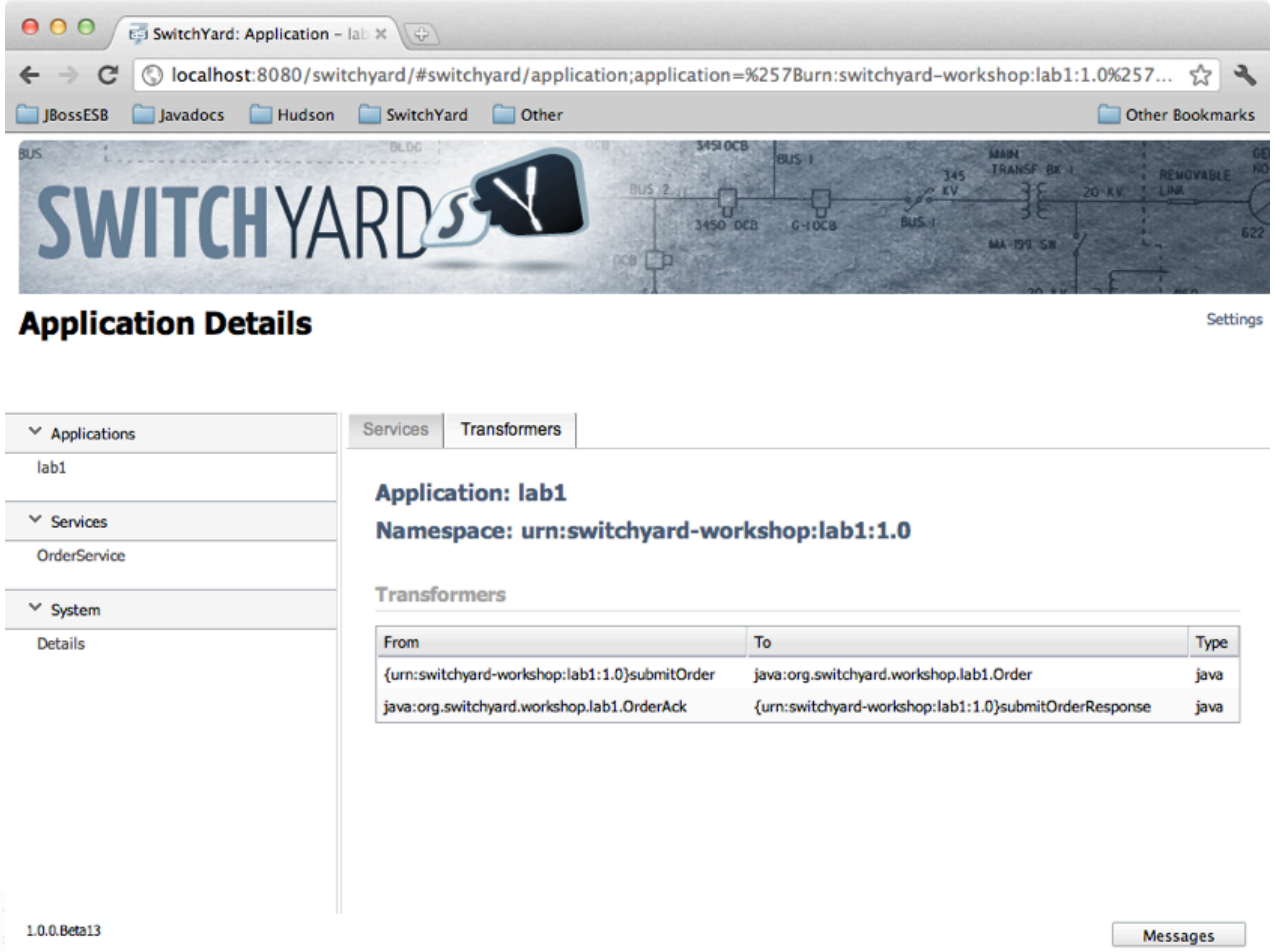
| Name | Promoted Service |
|---|---|
| OrderService | OrderService |

Component Services

| Name | Interface | Implementation |
|---|---|---|
| InventoryService | org.switchyard.workshop.lab1.InventoryService | View Details... |
| OrderService | org.switchyard.workshop.lab1.OrderService | View Details... |

1.0.0.Beta13

# Transformer View



## Application Details

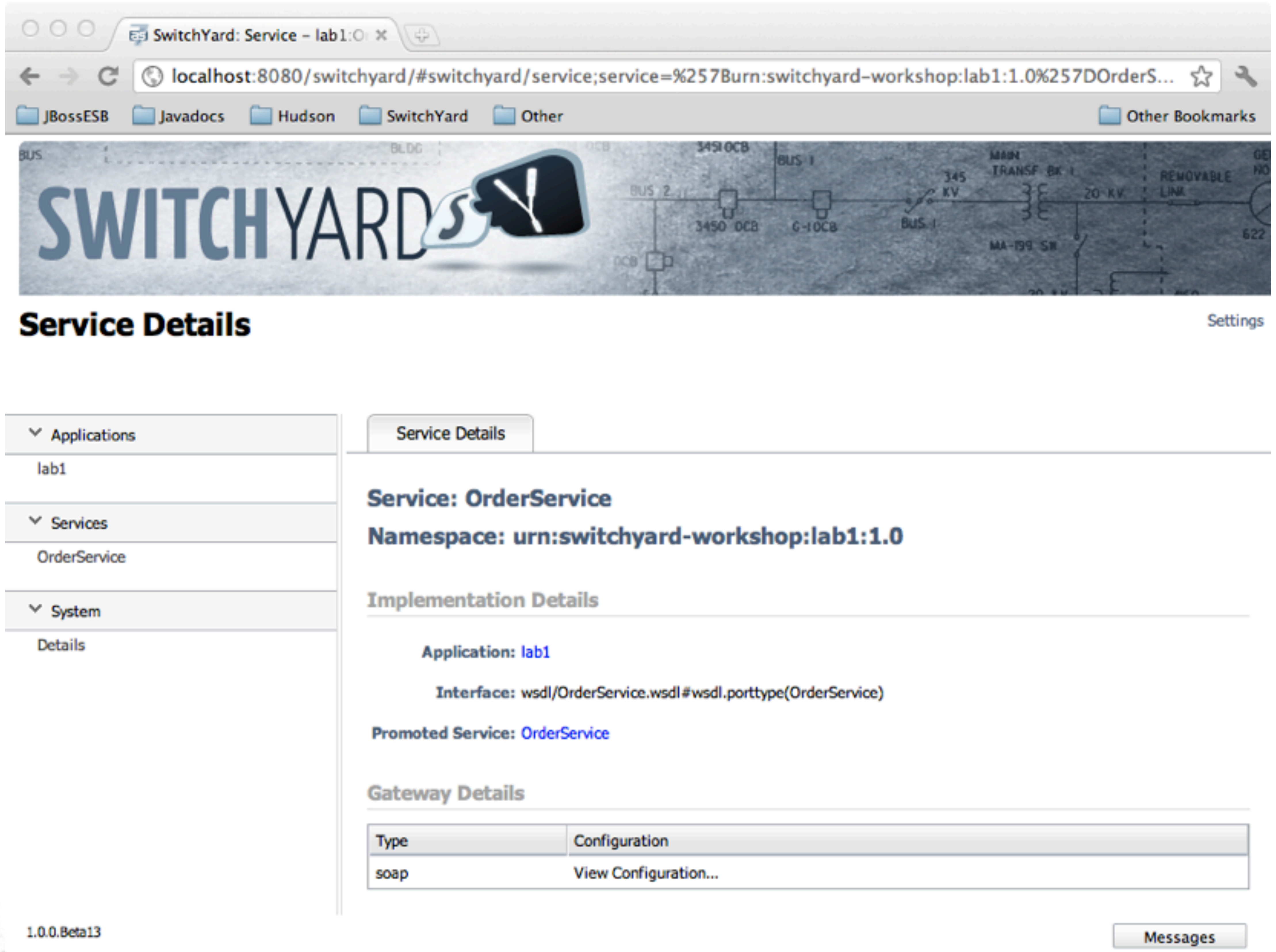
**Application: lab1**

**Namespace: urn:switchyard-workshop:lab1:1.0**

Transformers

| From | To | Type |
|---|---|---|
| {urn:switchyard-workshop:lab1:1.0}submitOrder | java:org.switchyard.workshop.lab1.Order | java |
| java:org.switchyard.workshop.lab1.OrderAck | {urn:switchyard-workshop:lab1:1.0}submitOrderResponse | java |

# Service View



## Service Details

**Service: OrderService**

**Namespace: urn:switchyard-workshop:lab1:1.0**

### Implementation Details

**Application:** lab1

**Interface:** wsdl/OrderService.wsdl#wsdl.porttype(OrderService)

**Promoted Service:** OrderService

### Gateway Details

| Type | Configuration |
|---|---|
| soap | View Configuration... |

JBoss Community

# Questions?

JBoss Community

JBoss
Community