



SWITCHYARD

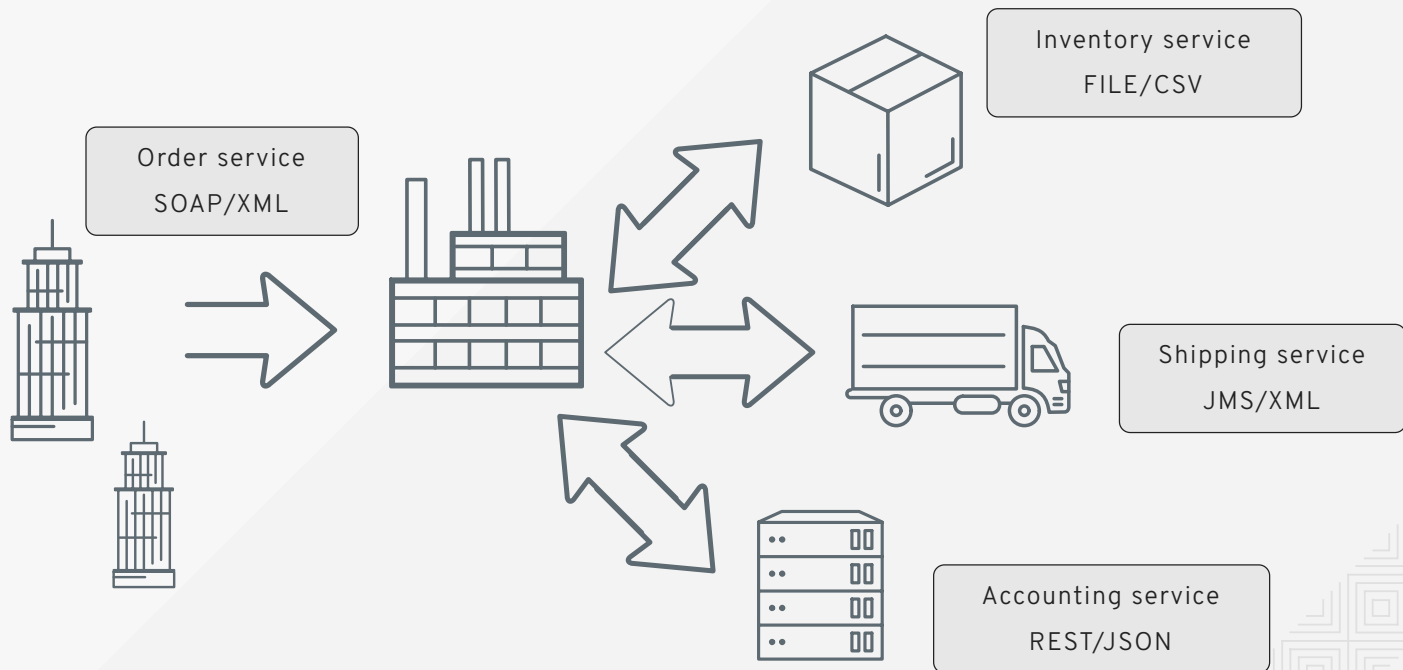
EXERCISE

Tomáš Turek - tturek@redhat.com

Andrej Podhradský - apodhrad@redhat.com

WHAT WE WANT TO ACHIEVE?

USE SWITCHYARD TO COMPOSE 3 APPLICATIONS



ENVIRONMENT

RED HAT JBOSS FUSE + JBOSS
DEVELOPER STUDIO WITH INTEGRATION
STACK

INSTALLATION GUIDE:

[HTTP://WWW.JBOSS.ORG/PRODUCTS/FU
SE/GET-STARTED/](http://www.jboss.org/products/fuse/get-started/)

PREPARATION

Clone git projects:

```
> git clone https://github.com/qa/course-sys-int-systems.git
```

```
> git clone https://github.com/qa/course-sys-int-switchyard-seminar.git
```

- copy file `$WORKSPACE/course-sys-int-switchyard-seminar/src/resources/keystore.jks` to `$FUSE_HOME/bin` folder
- copy file `$WORKSPACE/course-sys-int-switchyard-seminar/activemq.xml` to `$FUSE_HOME/etc` folder
- add a user to `$FUSE_HOME/etc/users.properties`:
`shipuser=shippwd,admin,manager,viewer,Monitor, Operator, Maintainer, Deployer, Auditor, Administrator, SuperUser`
- start/restart JBoss Fuse `$FUSE_HOME/bin/fuse`

run course-sys-int-system application

```
> mvn clean camel:run
```

LAB APPLICATION

project: **course-sys-int-switchyard-seminar**

initial branch: **master**

- Lab01
 - REST binding
- Lab02
 - JMS binding
- Lab03
 - FILE binding

LAB 01

REST

Goals: Integrate Accounting application via REST

Steps:

- Create new reference AccountingService and promote the service
- Set REST binding with
 - `https://localhost:7171`
 - `AccountingServiceReference`
 - `admin/foo/localhost/7171`
- Use the reference in ShopServiceBean

Test: Lab01RESTTest

LAB 02

JMS

Goals: Integrate Shipment application via JMS

Steps:

- Create new reference ShipmentRequestService
- Generate WSDL from ShipmentRequestService.java
- Promote the service and specify the created wsdl file
- Set JMS binding with
 - Queue SHPMNT.REQ
- In ShopOrderServiceBean use ReferenceInvoker
- Create a bean for ShipmentResponseService
- Generate wsdl and promote to the wsdl file
- Set JMS binding
 - Queue SHPMNT.RESP
- Map all properties by setting regex to '.*' (in both bindings)
- Connect ShipmentResponseService to OrderStatusService

LAB 03

FILE

Goals: Integrate Inventory application via File

Steps:

- Create reference InventoryRequestService
- Promote to InventoryInboxService
- Implement Java transformer
- Set File binding with
 - `${sys.base}/target/inbox/inventory`
- In ShopOrderServiceBean use ReferenceInvoker
- Create a bean for InventoryResponseService
- promote to InventoryOutboxService
- Set File binding with
 - `${sys.base}/target/outbox/inventory`
- Use the message composer
- Connect InventoryResponseService to OrderStatusService

HOMework ;-)

DEPLOYMENT

Requires more steps on Karaf

- configure keystore at `$FUSE_HOME/etc/org.ops4j.pax.web.cfg`
- jms connection factory is set via blueprint or spring
- install features listed in `src/main/resources/features.xml`

> `features:addurl mvn:com.redhat.brq.integration/switchyard-seminar/0.0.1-SNAPSHOT/xml/features`

> `features:install switchyard-integration-course`

- probably you will need to fix dependencies according to the current version
- quickstarts are available at `$FUSE_HOME/quickstarts/switchyard`