

---

# Integration with ESB

---

Filip Nguyen  
Jiří Kolář

---

# Task - Open Project and build it

---

- `cp -r /home/xnguyen/maven-repo/* ~/.m2/repository`
  - `cp -r /home/xnguyen/pa165/pa165-esb /tmp/jboss4`
-

# Task prepare for running ESB server

---

- Open two terminals.
  - In **first** terminal run
    - `rm -rf /tmp/jboss3 && mkdir /tmp/jboss3 && chmod -R 777 /tmp/jboss3`
    - `unzip /var/tmp/jboss/jbossesb-server-4.11.zip -d /tmp/jboss3`
  - In **both** terminals run
    - `export JBOSS_HOME=/tmp/jboss3/jbossesb-server-4.11`
    - `module add ant maven-3.0.4 jdk-1.7.0_03`
-

# Example: Sport Equipment Store

---

- Sport Equipment retail store receives orders as CSV files
  - Few years later they start accepting the orders through Web Service
-

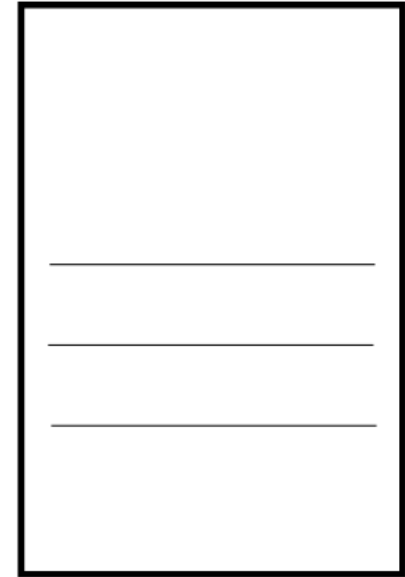
# Sport Equipment Store

---

30 Tennis shorts  
Golf club

Putting files via SSH or FTP

Company server



## THE NEW WAY

```
<order>  
<item>  
  <quantity> 30 </quantity>  
  <product> Tennis shorts </product>  
.....
```

SOAP Requests

# What we need for Sports Store Integration?

---

- Create Web Service?
  - Create File/FTP folder watching daemon?
  - Create common data format for the Order?  
(transformations)
  - E-mail notifications
  - Clustering
  - Robustness
-

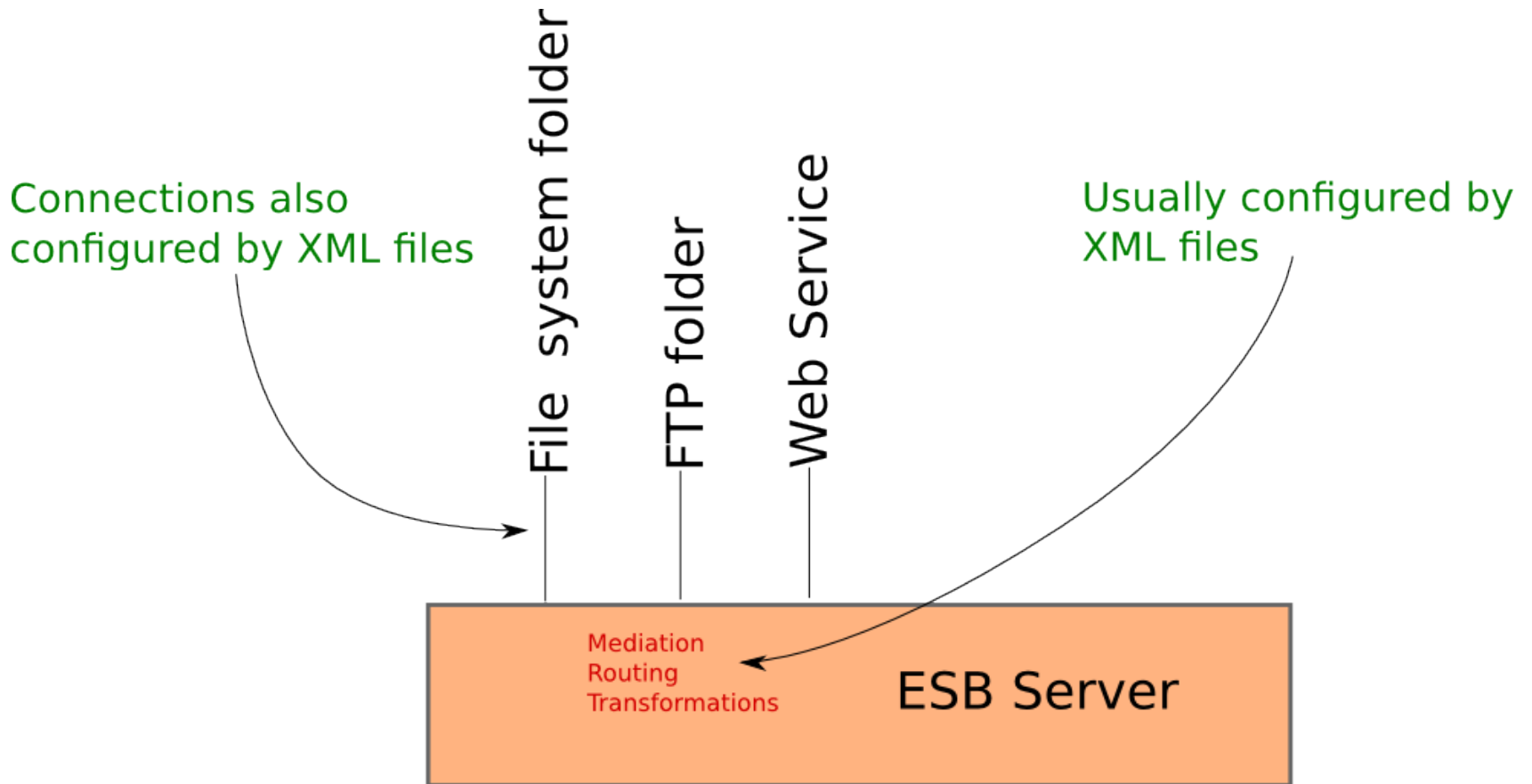
# Possible problems with handmade solution

---

- Simple answer is "too low level" and...
  - We need simple things
    - take file from FTP
    - take file from disk
    - send an email
    - apply XSLT transformation
  - Why bother with low level APIs?
  - We need also very complicated things
    - Robustness - what if the file is corrupted, if one connected EIS is down?
    - What if we want to scale out horizontally?
-

# How to do it in ESB way?

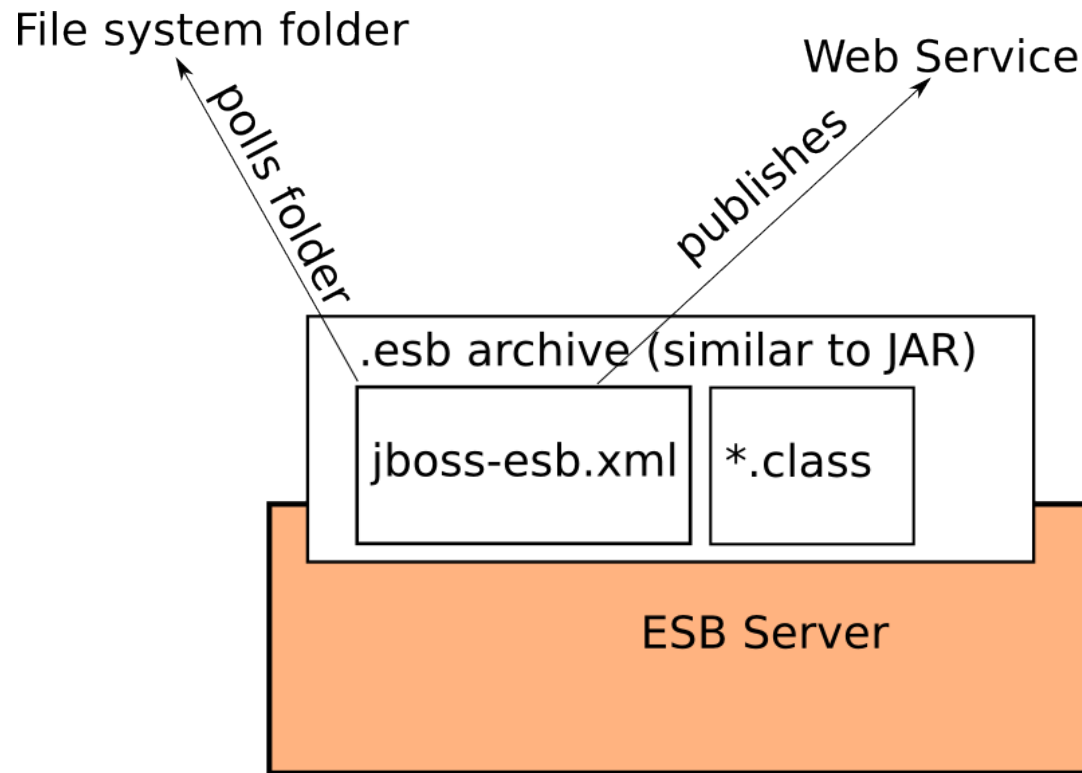
---





# Details of JBoss ESB solution

---



# JBoss ESB basics

---

- Written in Java
  - XML centered configuration
-

# Task start the server

---

- Start the JBoss ESB server: In first terminal run: `$JBOSS_HOME/bin/run.sh`
- In the second run
  - `cd $JBOSS_HOME/samples/quickstarts/helloworld`
  - `ant deploy && sleep 5 && ant runtest`

Server:

10:53:15,682 INFO [STDOUT] Body: Hello World

---

# JBoss ESB links

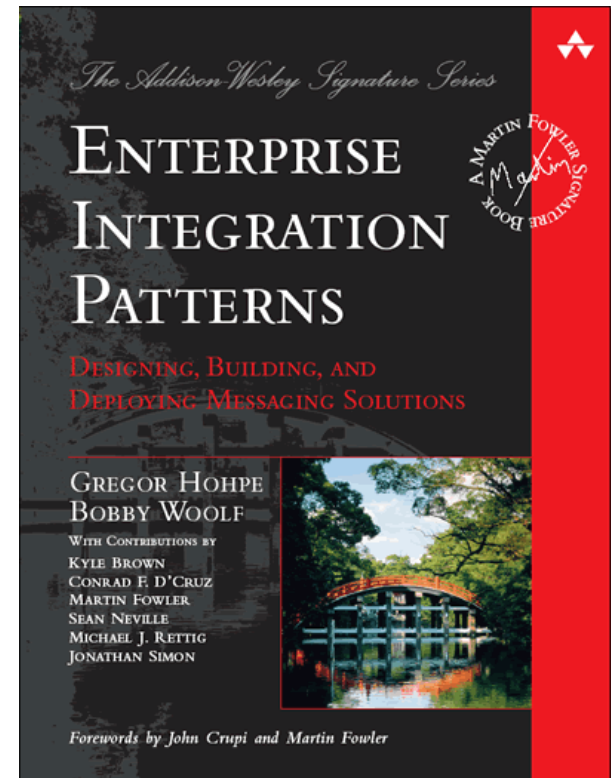
---

- `wget http://download.jboss.org/jbossesb/4.11/binary/jbossesb-server-4.11.zip`
  - Docs + JavaDoc: <http://www.jboss.org/jbossesb/docs/index>
-

# Sport Store Enterprise Integration

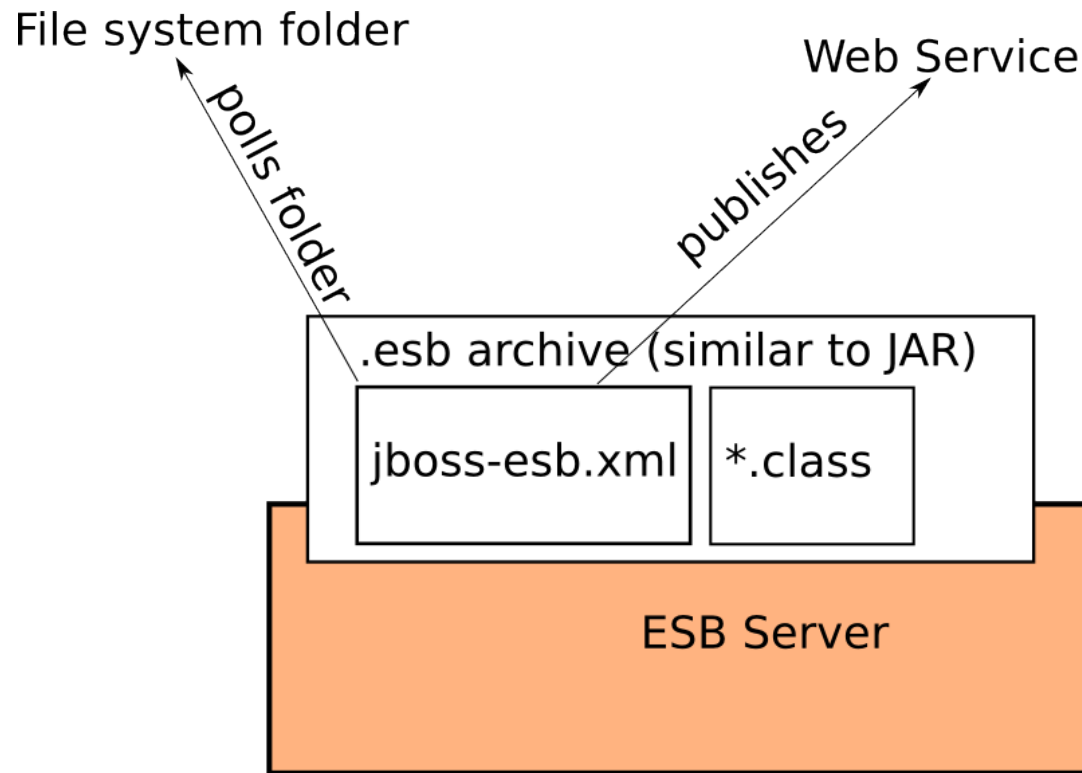
---

- We create a .esb archive and put it on ESB Server
- The .esb contains
  - Connections of data sources to the ESB
  - Routing information
  - Transformation
  - Class files
  - All the integration specific resource



# Details of JBoss ESB solution

---



# Build the project

---

- open the project `/tmp/jboss4/pa165-esb` in your IDE
  - cd into `pa165-esb` and run `"mvn package"`
-

# \*.esb content: META-INF/jboss-esb.xml

---

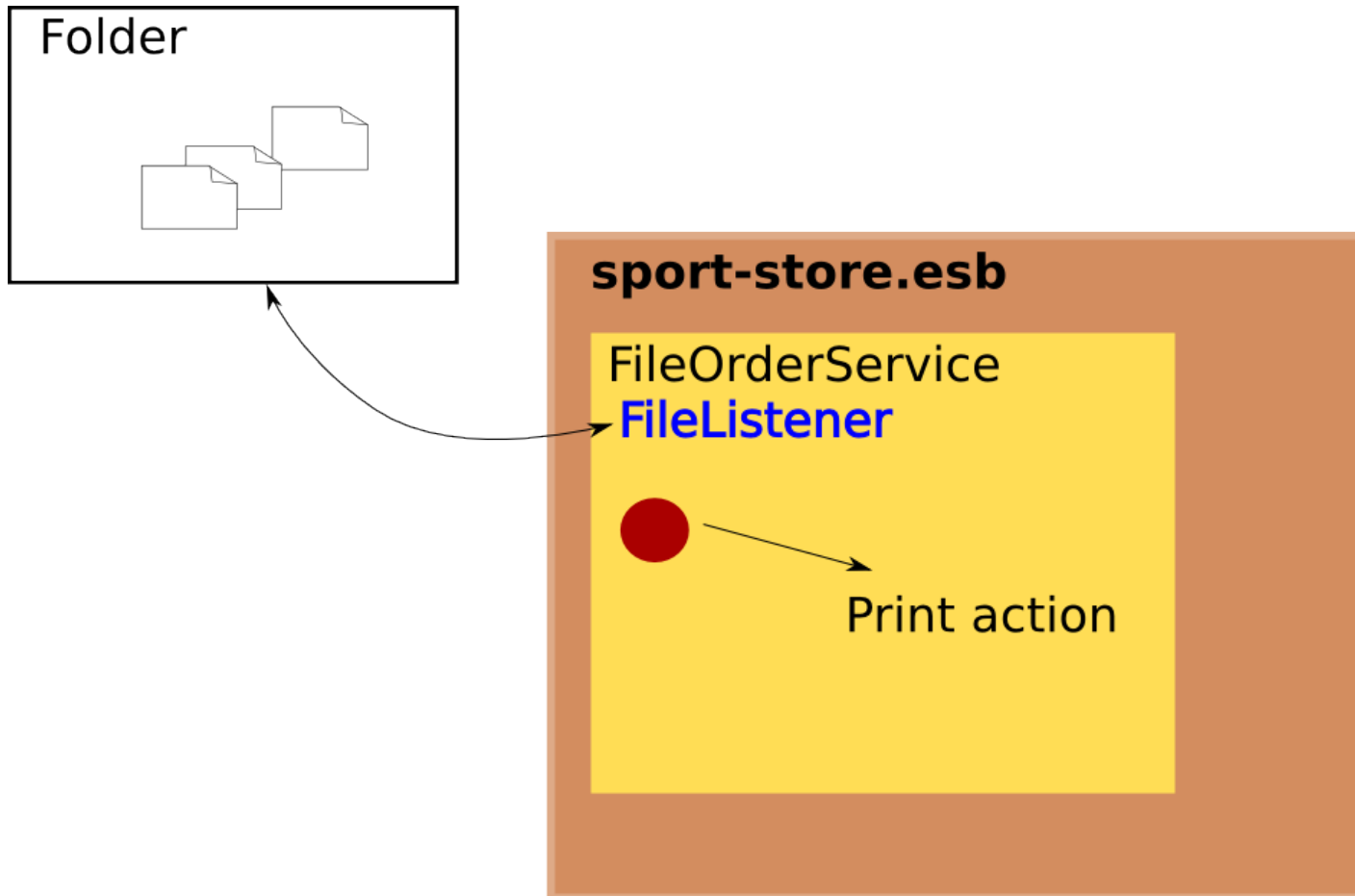
```
<providers> .... </providers>
<services>
  <service name="FileOrderService" ... >
    <listeners> ... </listeners>
    <actions> .... </actions>
  </service>
</services>
```

---



# \*.esb contains Services

---



# Submit order by file

---

- use **mvn package** to deploy to running server
  - **copy**  
SportStore/sport-store/order-folder/order1.order  
**into**  
SportStore/sport-store/order-folder/order-input
  - Watch console
-

# Add custom code into pa165.

## FileOrderProcessor

---

- Use action to manipulate message (in this case file content)
  - **Your task is:** Print out the content of the message to console. Comment built-in print action in the jboss-esb.xml
  - What else you can do in this custom action?
    - Modification of the message
    - Doing custom stuff with the message
    - Invoking other services in the ESB via ServiceInvoker
-

# Publish Web Service

---

- JBoss ESB can automatically publish a `<service>` as `WebService`. It will automatically generate WSDL
    - `http://localhost:8080/jbossws/services`
  - One WS (needed by Sport Store) is already running in your ESB
  - **Task:** Inspect `jboss-esb.xml` and find out how is it defined
    - Its very cryptic :-)
    - How is the contract for the service defined?
    - What is possible SOAP request you can make to service - dont look into WSDL ;)
-

# The Answer to WS questions

---

- inXsd="/request.xsd"
  - WSDL found at: <http://localhost:8080/jbossws/services>
-

# Task: Submit order by WS client

---

- Modify Main class OrdersWSClient in project sport-store-ws-client and run it with mvn exec:java
  - Watch console output
-

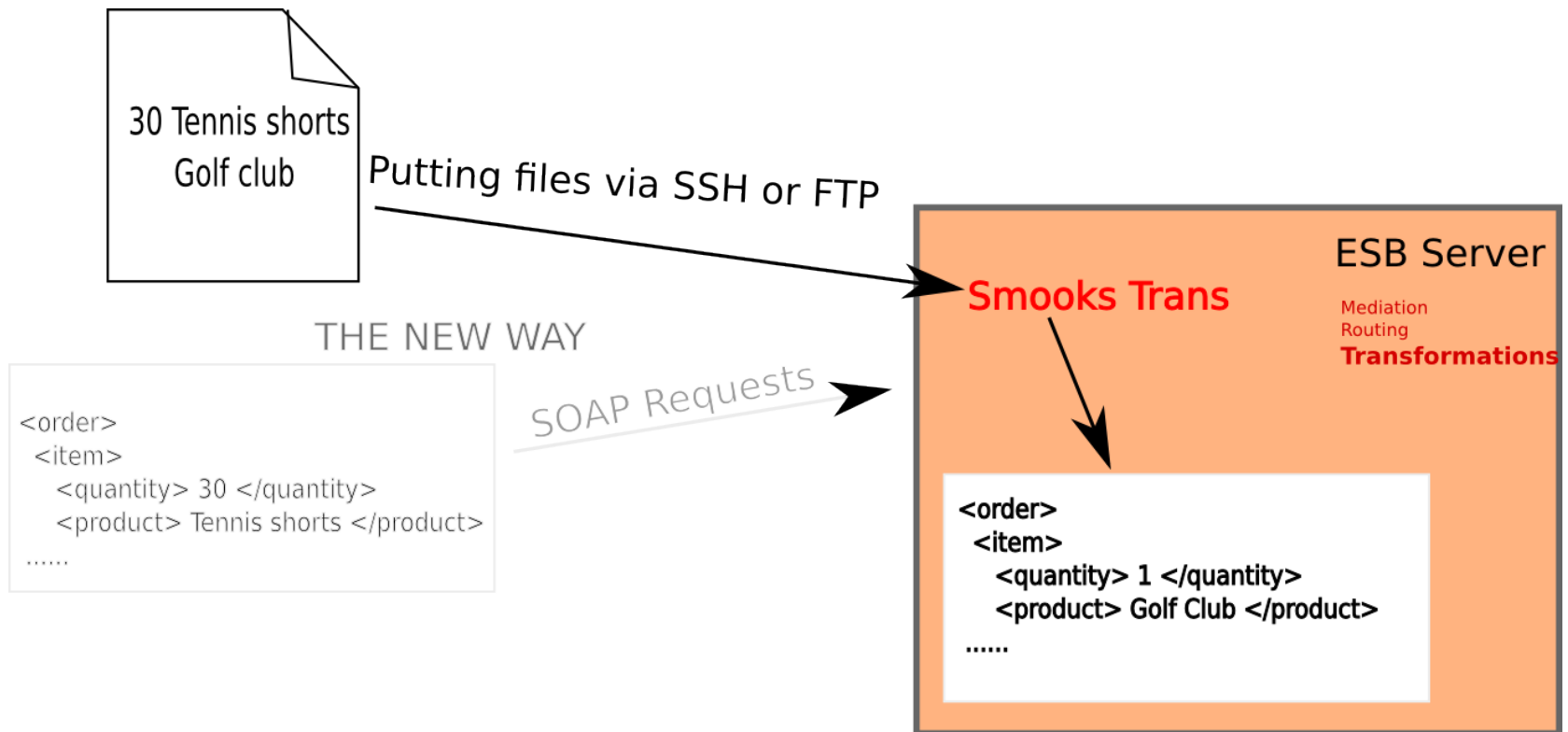
# Common Format for the Order

---

- File order comes in CSV format
  - What about formatting it as XML?
    - Custom action with JAXP?
  - Transformations via Smooks!
  -
-

# Transformation

---





# TASK

---

Inspect your `jboss-esb.xml` and find configuration for smooks. Uncomment it. It should use XSLT transformation! Find out where is it

---

# Static Router

---

- Our goal is to route messages from our two services (WS, File) to one common service
  - Inspect jboss-esb.xml. WS <service> already does this
  - **Task:** reconfigure File <service> to do it also
  - Why do you think its called Static routing?
-

# Because there is also ContentBasedRouter

---

- Based on message content it routes to various <service>
  - Inspect quickstart wiretap in `$JBOSS_HOME/samples/quickstarts`.
    - **Task:** What is the content based router doing?
-

# Homework: Reconfigure the .esb

---

- META-INF/jboss-esb.xml
  - Add notification to e-mail about any order
    - Custom Action Perhaps?
    - There is a better way! Go through JBossESB programmers guide. 4.1.3. Notifiers
    - Use Notifiers (commented in your <service name="OrderService"
-

# Importance of messaging

---

- Most decoupled way of communication
    - Please read about it on blogs :-D
    - try JMS (download ActiveMQ and connect to it using Spring JMS centered classes)
  - JBoss ESB makes it easy to define JMS queues
    - inspect jbm-queue-service.xml
    - **Homework Task:** add new queue "my\_queue" and run "mvn package" if you don't do any mistake your sport-store.esb should deploy without errors in console. You should see the output of the console to inform you about "my\_queue" deployed
-

# Other commands

---

**Task:** mvn install:install-file -  
Dfile=\$JBOSS\_HOME/server/default/deploy/jbos  
seseb.sar/lib/jbosseseb-rosetta.jar -  
DgroupId=org.jboss -DartifactId=jbosseseb-  
rosetta -Dversion=4.11 -Dpackaging=jar

cd into /tmp/jboss and clone repo:  
git clone git://github.com/nguyenfilip/pa165-esb.git

---