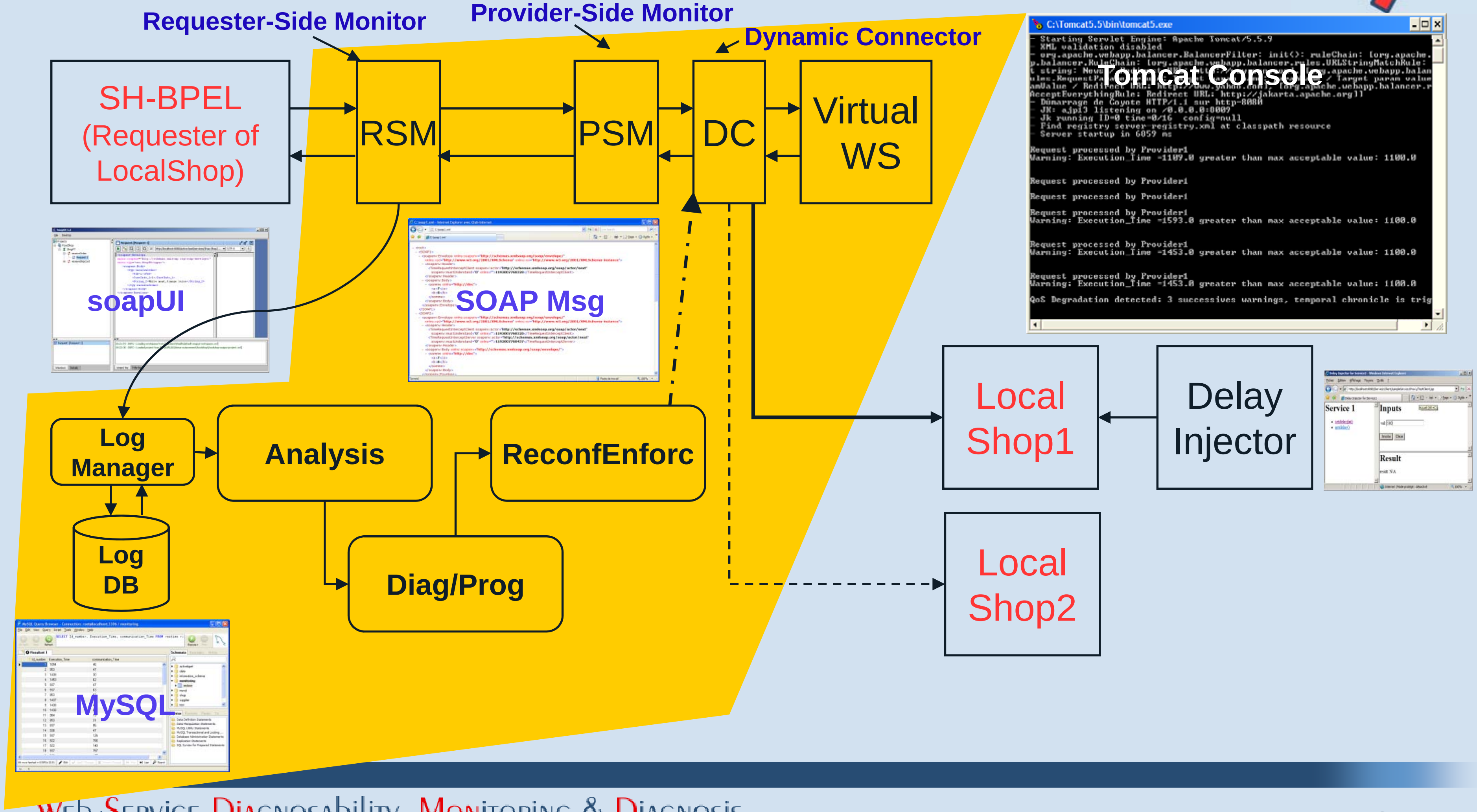
Demonstration: Scenario description

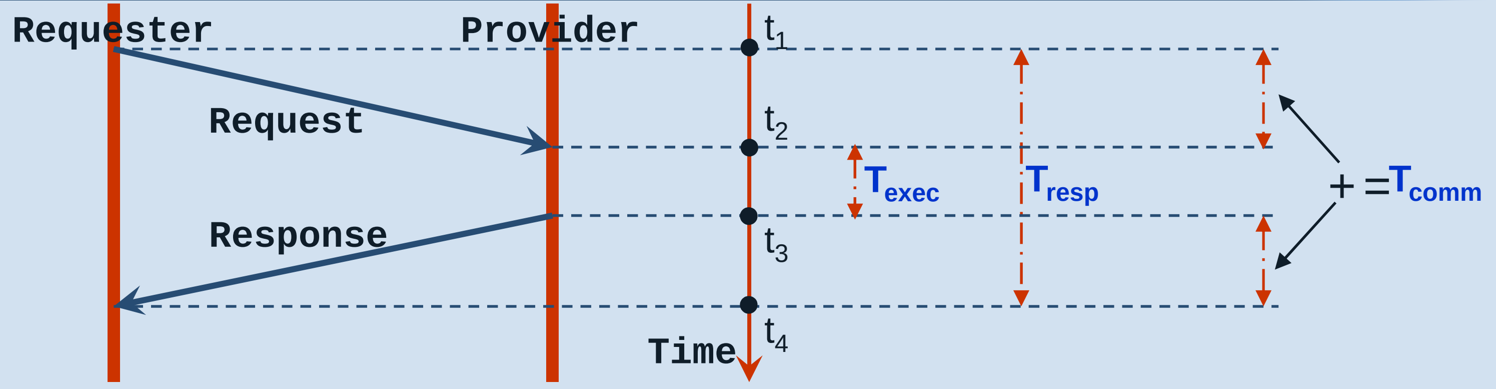
LAAS-CNRS, France
Riadh BEN HALIMA & Khalil DRIRA
ReDCAD, Tunisia
Mohamed JMAIEL

Following the scenario….



```
C:\Tomcat5.5\bin\tomcat5.exe
- Starting Servlet Engine: Apache Tomcat/5.5.9
- XML validation disabled
- org.apache.webapp.balancer.BalancerFilter: init(): ruleChain: [org.apache.
p.balancer.RuleChain: [org.apache.webapp.balancer.rules.URLStringMatchRule:
t string: News [illegible] [org.apache.webapp.balan
ules.RequestPa[illegible] / Target param value
amValue / Redirect URL: http://www.yahoo.com], [org.apache.webapp.balancer.r
AcceptEverythingRule: Redirect URL: http://jakarta.apache.org]]
- Dùmarrage de Coyote HTTP/1.1 sur http-8080
- JK: ajp13 listening on /0.0.0.0:8009
- Jk running ID=0 time=0/16  config=null
- Find registry server-registry.xml at classpath resource
- Server startup in 6059 ms

Request processed by Provider1
Warning: Execution_Time -1107.0 greater than max acceptable value: 1100.0


Request processed by Provider1

Request processed by Provider1

Request processed by Provider1
Warning: Execution_Time -1593.0 greater than max acceptable value: 1100.0


Request processed by Provider1
Warning: Execution_Time -1453.0 greater than max acceptable value: 1100.0


Request processed by Provider1
Warning: Execution_Time -1453.0 greater than max acceptable value: 1100.0

QoS Degradation detected: 3 successives warnings, temporal chronicle is trig
```

Considered QoS parameters

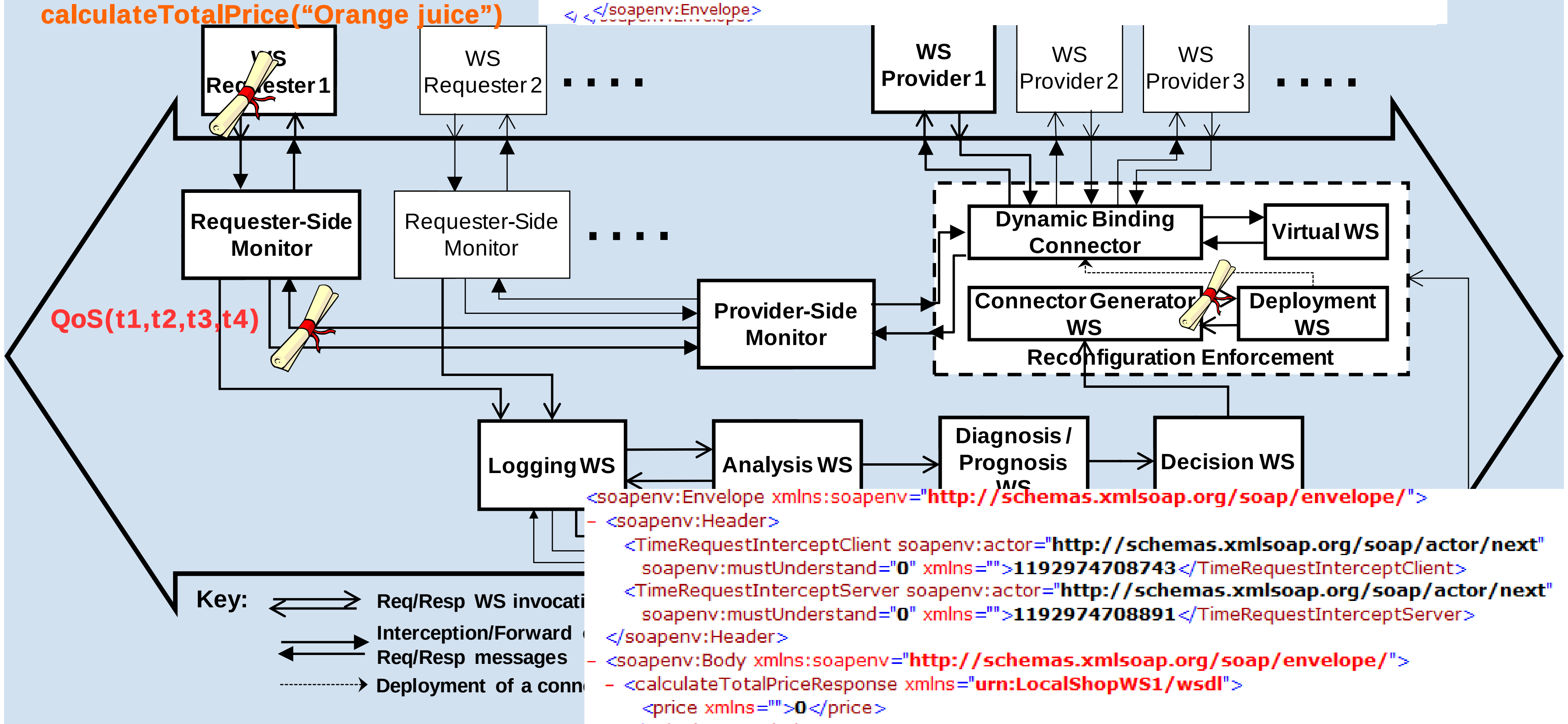


- **Execution Time**: The time that the provider needs to achieve the processing of the request:
  - ***Texecution = t3 – t2*** *(The considered parameter for the FoodShop demo)*
- **Response Time** : The time between sending a request and receiving the response:
  - ***Tresponse = t4 – t1*** *(Has been considered for other scenarios)*
- **Communication Time**: The time that the SOAP message needs to reach its destination:
  - ***Tcommunication = Tresponse – Texecution*** *(Has been considered for other scenarios)*

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xml
- <soapenv:Header>
    <TimeRequestInterceptClient soapenv:actor="htt
      soapenv:mustUnderstand="0" xmlns="">11929
  </soapenv:Header>
- <soapenv:Body xmlns:soapenv="http://schemas.
  - <ns1:calculateTotalPrice soapenv:encodingStyle=
      xmlns:ns1="urn:LocalShopWS1/wsdl">
      <itemList xsi:type="xsd:string" xmlns:xsi="http
        instance">Orange juice</itemList>
    </ns1:calculateTotalPrice>
  </soapenv:Body>
</soapenv:Envelope>
```

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
- <soapenv:Header>
    <TimeRequestInterceptClient soapenv:actor="http://schemas.xmlsoap.org/soap/actor/next"
      soapenv:mustUnderstand="0" xmlns="">1192974708743</TimeRequestInterceptClient>
    <TimeRequestInterceptServer soapenv:actor="http://schemas.xmlsoap.org/soap/actor/next"
```

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Header />
- <soapenv:Body xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  - <calculateTotalPriceResponse xmlns="urn:LocalShopWS1/wsdl">
      <price xmlns="">5</price>
    </calculateTotalPriceResponse>
  </soapenv:Body>
</soapenv:Envelope>
    </soapenv:Body>
  </soapenv:Envelope>
```



```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
- <soapenv:Header>
    <TimeRequestInterceptClient soapenv:actor="http://schemas.xmlsoap.org/soap/actor/next"
      soapenv:mustUnderstand="0" xmlns="">1192974708743</TimeRequestInterceptClient>
    <TimeRequestInterceptServer soapenv:actor="http://schemas.xmlsoap.org/soap/actor/next"
      soapenv:mustUnderstand="0" xmlns="">1192974708891</TimeRequestInterceptServer>
  </soapenv:Header>
- <soapenv:Body xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  - <calculateTotalPriceResponse xmlns="urn:LocalShopWS1/wsdl">
      <price xmlns="">0</price>
    </calculateTotalPriceResponse>
  </soapenv:Body>
</soapenv:Envelope>
  </soapenv:Body>
```

Demonstration Steps

- Already started: DBMS, Tomcat server (HTTP server, WS Container, ActiveBpel, all FoodShop WS and Self-Healing WS,…), soapUI, Delay injectors
- Step1: Correct state with LocalShop1 as provider
- Step2: Inject delay in LocalShop1’s processing time, and detect QoS degradation
- Step3: Reconfigure by dynamically rebinding the requester to LocalShop2 instead of LocalShop1
- Step4: Back to “correct state” with LocalShop2 as provider, and check reconfiguration is still valid for future requesters

Step1: Correct state

For each request:

- The SOAP interceptors intercept SOAP messages and extend them with QoS metadata and associated values
- The extended SOAP messages are written in the file “soap.xml” in order to be shown in the demo
- The content of the log is updated with the three QoS parameter values (t2,t3,texec)
- The name of the currently used concrete Web service is printed in the Tomcat console

Step2: Inject delay and detect QoS degradation

1. Using the delay injector, we inject delay of 500ms to LocalShop1
2. Warnings are observed after each request
3. After 3 consecutive “Texec” greater than a max acceptable value (1100ms), the analysis service notifies a QoS degradation towards the Diagnosis service. (successive increasing of QoS)
4. The diagnosis service will conclude obviously that LocalShop1 is faulty
5. It will propose rebinding as a repair action based on architectural reconfiguration

Step3: Reconfiguration by rebinding

1. "LocalShop1" will be replaced by "LocalShop2" as follows:
   1. Generating the Java code of the new dynamic connector bound to LocalShop2
   2. Compiling on-line the generated code
   3. Redeploying the new dynamic connector as a handler within the Virtual WS
2. Visually, this will be shown:
   1. In the directory where these files are generated
   2. In the Tomcat console, by a message of the Tomcat manager

Step4: back to correct state

1. Invoke service
2. Observe:
    1. Request processed by LocalShop2
    2. Acceptable response time (logged in database)
3. The substitution is class-level: all future requesters will be bound to LocalShop2.

Demo: configuration

- Web services of application scenarios and prototype:
  - Web service container: Axis 1.4
  - Web server: Tomcat 5.5.17
  - Programming language: Java 1.5
  - Monitors & Connectors: Axis Handlers
  - Communication level: SOAP
- Logging
  - MySQL DBMS

Thank you