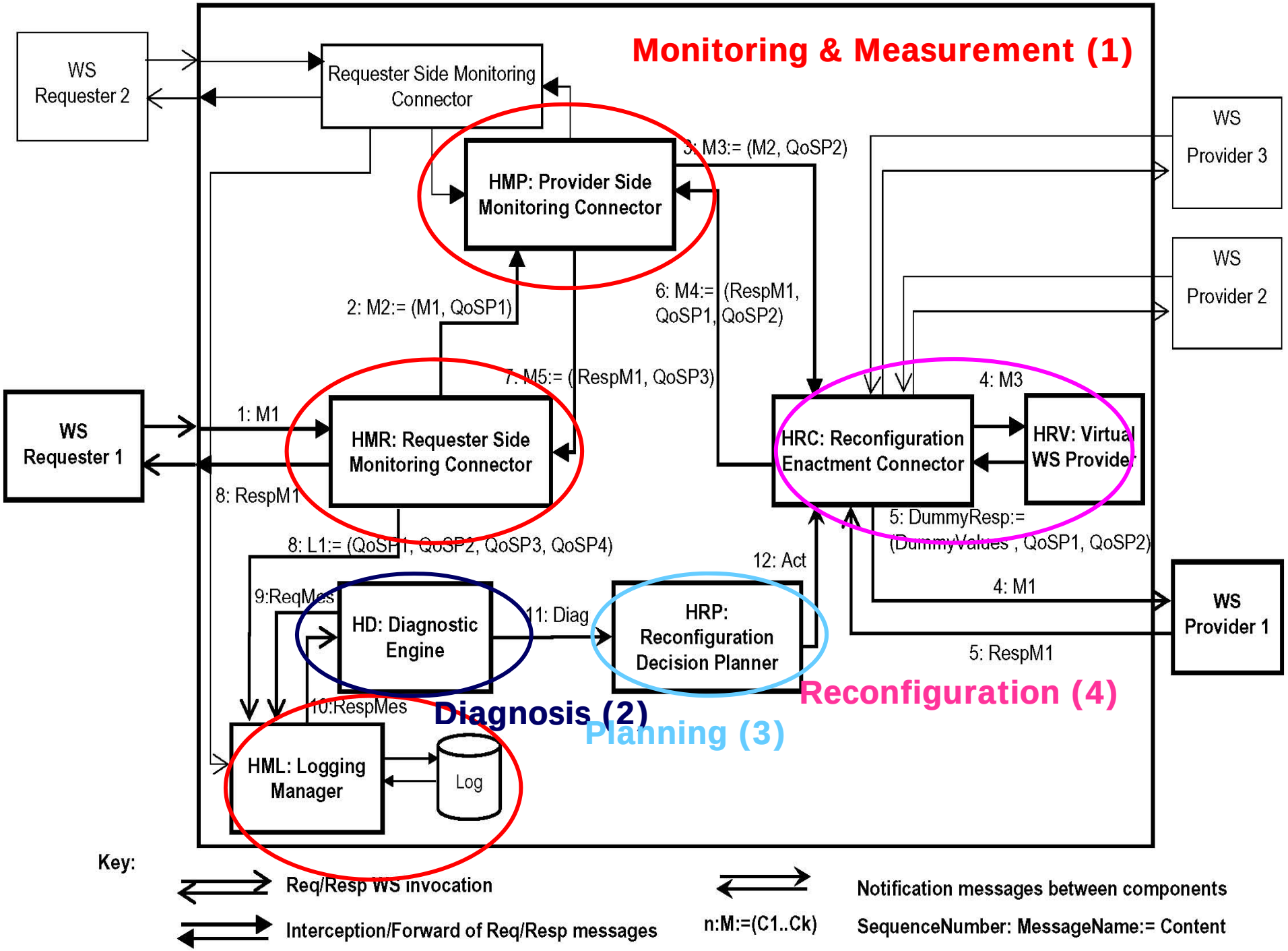
WS-DIAMOND meeting in Paris, August 29

Riadh BEN HALIMA & Khalil DRIRA

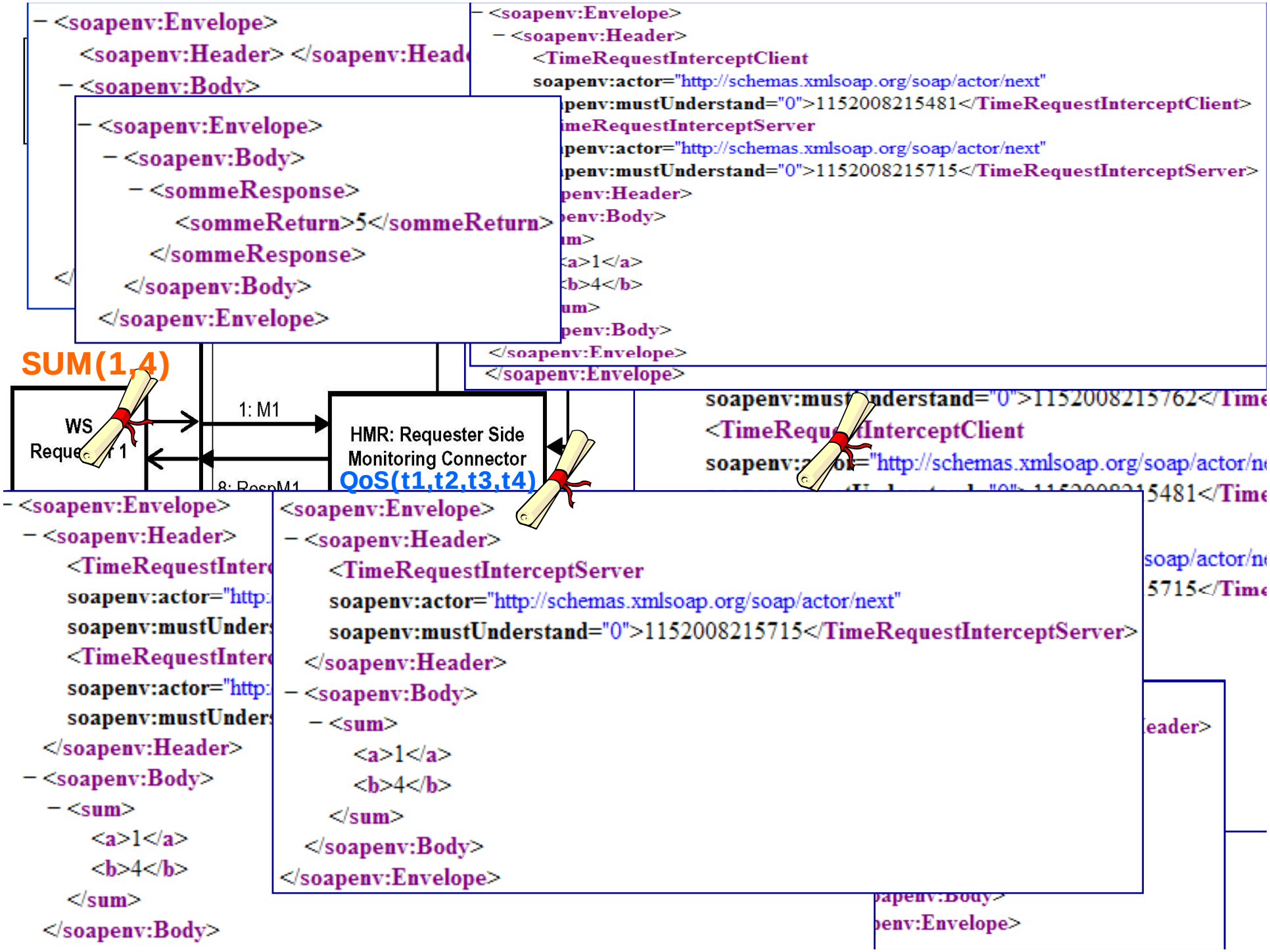
LAAS-CNRS

Outline

- Implemented modules for QoS prototype
  - Monitoring
  - Diagnosis
  - Planning
  - Repair


Monitoring & Measurement (1)
WS Requester 2
Requester Side Monitoring Connector
HMP: Provider Side Monitoring Connector
3: M3:= (M2, QoSP2)
WS Provider 3
WS Provider 2
2: M2:= (M1, QoSP1)
6: M4:= (RespM1, QoSP1, QoSP2)
7: M5:= (RespM1, QoSP3)
4: M3
WS Requester 1
1: M1
HMR: Requester Side Monitoring Connector
HRC: Reconfiguration Enactment Connector
HRV: Virtual WS Provider
8: RespM1
5: DummyResp:= (DummyValues , QoSP1, QoSP2)
8: L1:= (QoSP1, QoSP2, QoSP3, QoSP4)
12: Act
4: M1
9:ReqMes
HD: Diagnostic Engine
11: Diag
HRP: Reconfiguration Decision Planner
WS Provider 1
5: RespM1
Reconfiguration (4)
10:RespMes
Diagnosis (2)
Planning (3)
HML: Logging Manager
Log
Key:
Req/Resp WS invocation
Interception/Forward of Req/Resp messages
Notification messages between components
n:M:=(C1..Ck)
SequenceNumber: MessageName:= Content

Monitoring


SUM(1,4)
WS
1: M1
HMR: Requester Side
Monitoring Connector
QoS(t1,t2,t3,t4)
<soapenv:Envelope>
<soapenv:Header>
<TimeRequestInterceptClient
soapenv:actor="http://schemas.xmlsoap.org/soap/actor/next"
soapenv:mustUnderstand="0">1152008215481</TimeRequestInterceptClient>
<TimeRequestInterceptServer
soapenv:actor="http://schemas.xmlsoap.org/soap/actor/next"
soapenv:mustUnderstand="0">1152008215715</TimeRequestInterceptServer>
</soapenv:Header>
<soapenv:Body>
<sum>
<a>1
4
</sum>
</soapenv:Body>
</soapenv:Envelope>
<soapenv:Envelope>
<soapenv:Body>
<sommeResponse>
<sommeReturn>5</sommeReturn>
</sommeResponse>
</soapenv:Body>
</soapenv:Envelope>

Considered QoS parameters

- **Response Time** : The time taken to send a request and to receive the response:
  - ***Tresponse = t4 – t1***
- **Execution Time**: represents the time that the provider needs to achieve the processing of the request:
  - ***Texecution = t3 – t2***
- **Communication Time**: The time that the SOAP message needs to reach its destination:
  - ***Tcommunication = Tresponse – Texecution***
- **Throughput**: The number of Web service requests for an operation that can be processed by a service within a given period:
  - ***Throughput= # requests/time period***
- **Availability**: The availability of a service is the probability that the service is accessible:
  - ***Availability = Number of successful executions/Total number of invocations***

Monitoring: Techniques & Policies

- Implementation:
  - Handlers for Connectors
  - WS for the Logging Manager
- Policies
  - Send notification after each storage request
  - Send notification periodically (each 5sec, each 100 requests, ...)
  - ...

Diagnosis

Diagnosis: Techniques & Policies

- Implementation:
  - WS for the Diagnosis Engine
- Policies
  - 3 successive times :
    - Texec>(AVG_{Texec}+lapse) [lapse= “standard deviation”]
  - 3 successive increasing of Texec:
    - Texec3>Texec2>Texec1
  - Texec increases abruptly:
    - Texec2 >> Texec1
  - Chronics

Planning

Planning: Techniques & Policies

- Implementation:
  - WS for the Decision Planner
- Policies
  - Pre-prepared lists of equivalent services
    - Simple service ✸ Simple service
    - Simple service ✸ Composed service
  - Ontology + UDDI= Search for equivalent services

Reconfiguration

Reconfiguration: Techniques & Policies

- Automated generation of redirection connectors:
  - Java Reflection, Compile java, Compile WSDL
- Redeployment of the new connector
- Send “wait()” for current client requests until deployment achievement
- Release blocked client requests and continue request processing

Development environment

- Web service-based application:
  - Web services engine: Axis 1.4
  - Container: Apache Tomcat 5.5.17
  - Programming language: Java 1.5
  - Connector: Axis Handlers
- Logging
  - MySQL DBMS, through a Web service (Logging manager)

Demo

Thank you