17th FFV Workshop

Protecting Multi-Interfaced Mobile Web Services using Agreements

Fahad Aijaz

- Research Engineer -

Communication Networks

RWTH Aachen University

March 12, 2010

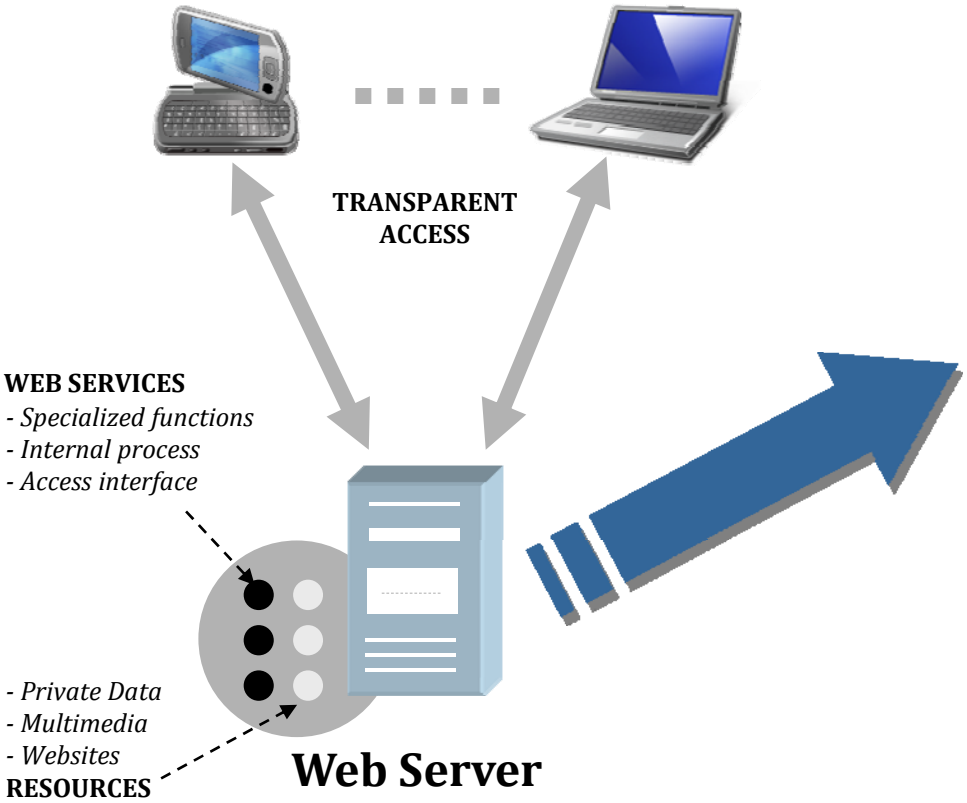
Outline of Talk

- **Research Scope**
  - Architectural Evolution of the Mobile Web Server
- **Multi-Interfaced Mobile Web Services**
  - SOAP Services on Mobile Devices
  - Service as a Resource (SaaR)
- **Service Level Agreements Framework**
  - Phases and Life Cycles
- **Performance Evaluation**
- **Conclusion**

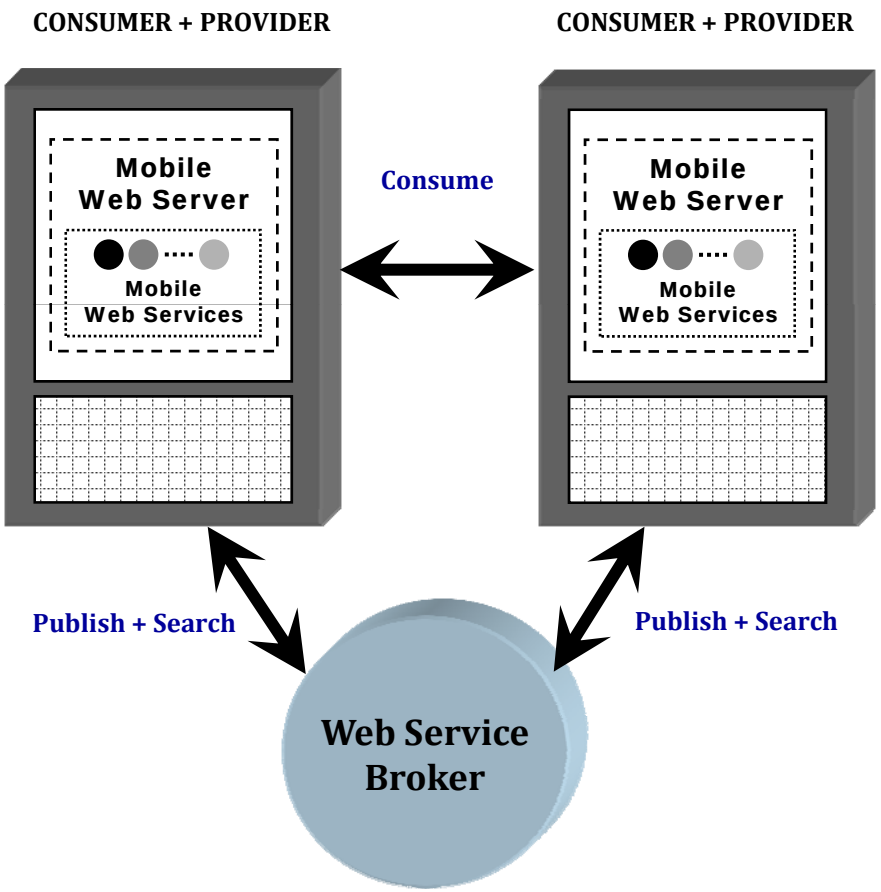
Research Scope
(Introduction)

GENERAL CONCEPT OF TODAY'S WEB

- → High-tech Web Servers.
- → Hosts Web Service and Resources.
- → Transparent Access to the Clients.
- → Neutral towards diverse clients.



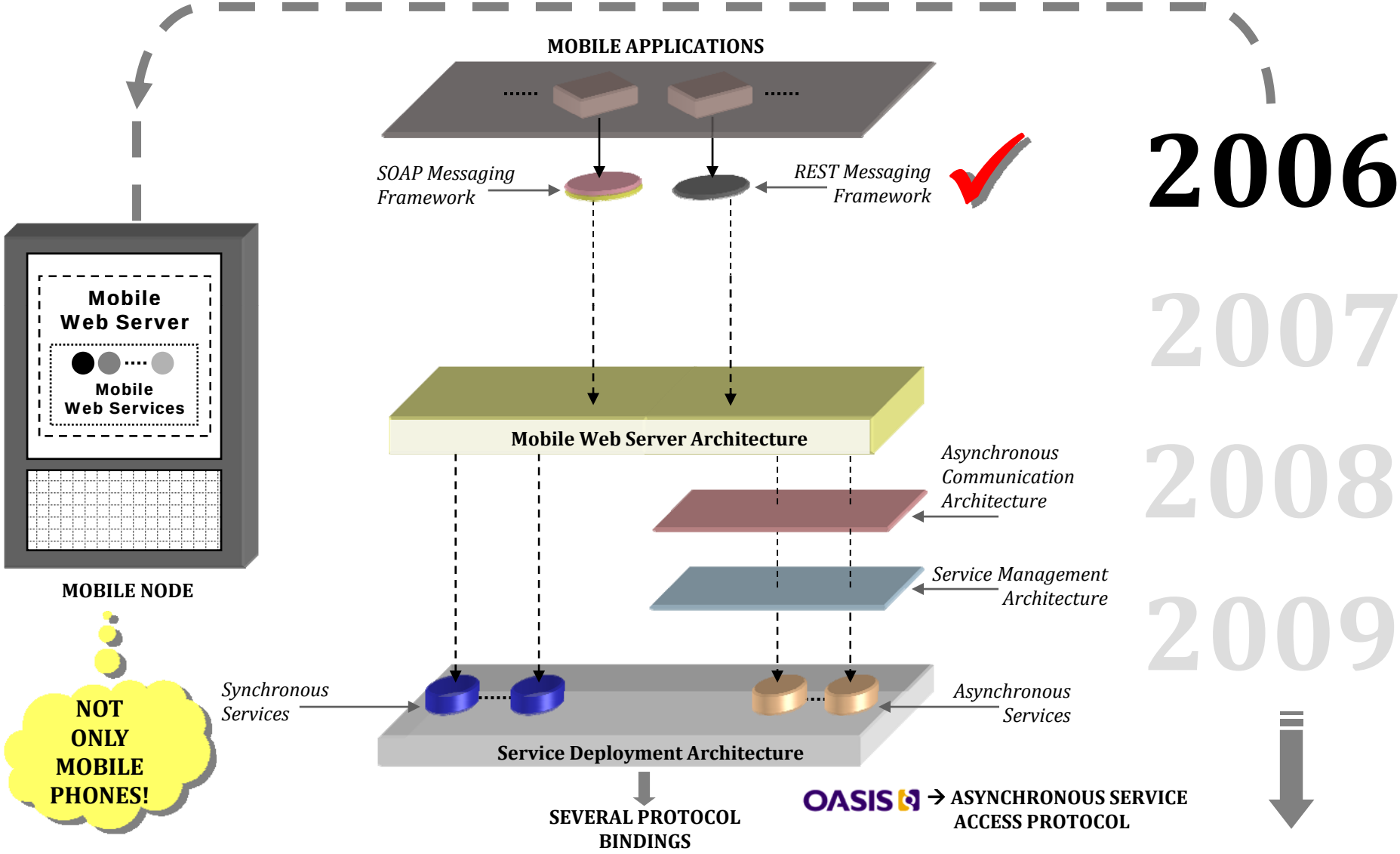
MOBILE WEB SERVER
IP Access to Mobile Nodes



P2P Mobile Web Services

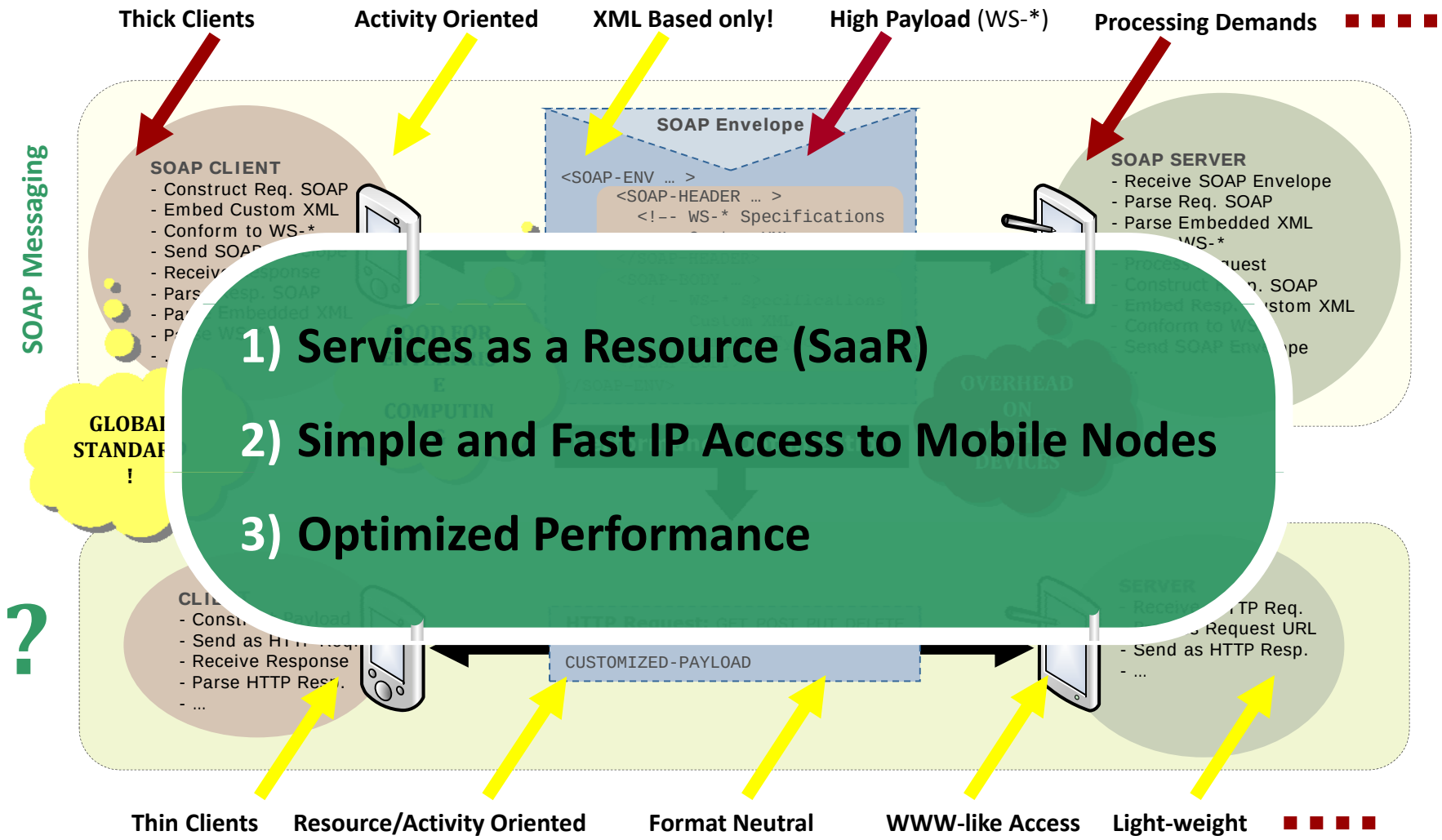
Research Scope

(Evolution of the Mobile Web Server Architecture)



Multi-Interfaced Mobile Web Services

(SOAP Services on Mobile Devices)



1) Services as a Resource (SaaR)

2) Simple and Fast IP Access to Mobile Nodes

3) Optimized Performance

Multi-Interfaced Mobile Web Services

(Service as a Resource → SaaR)

FUNDAMENTAL PRINCIPLE

Identify the network resources using URLs!

↓

Map the request action to HTTP Methods!

Not a Standard…
…but it uses standards!

REpresentational

Network resources
Example: HTML, XML, JPG, GIF…

State

Every resource changes the client's state

Transfer

Resources are transferred using HTTP

Architecture style…
…analogous to Client-Server

World Wide Web is RESTful!

State

http://comnets.rwth-aachen.de/fah.html

Transfer *Representation*

Services as a Resource (SaaR)

(Defining Synchronous & Asynchronous Access)

SYNCHRONOUS ACCESS

rest.comnets.de *LocationService*

HTTP://[IP]:[PORT]/[SERVICE]/[RESOURCE]

9091 *Coordinates*

SERVICE IDENTIFIES THE REQUESTED ACTION…
…BY USING HTTP METHODS

GET | POST | PUT | DELETE

→ FETCH → UPDATE → INSERT → REMOVE

ASYNCHRONOUS ACCESS

rest.comnets.de *aLocationService* *Factory*

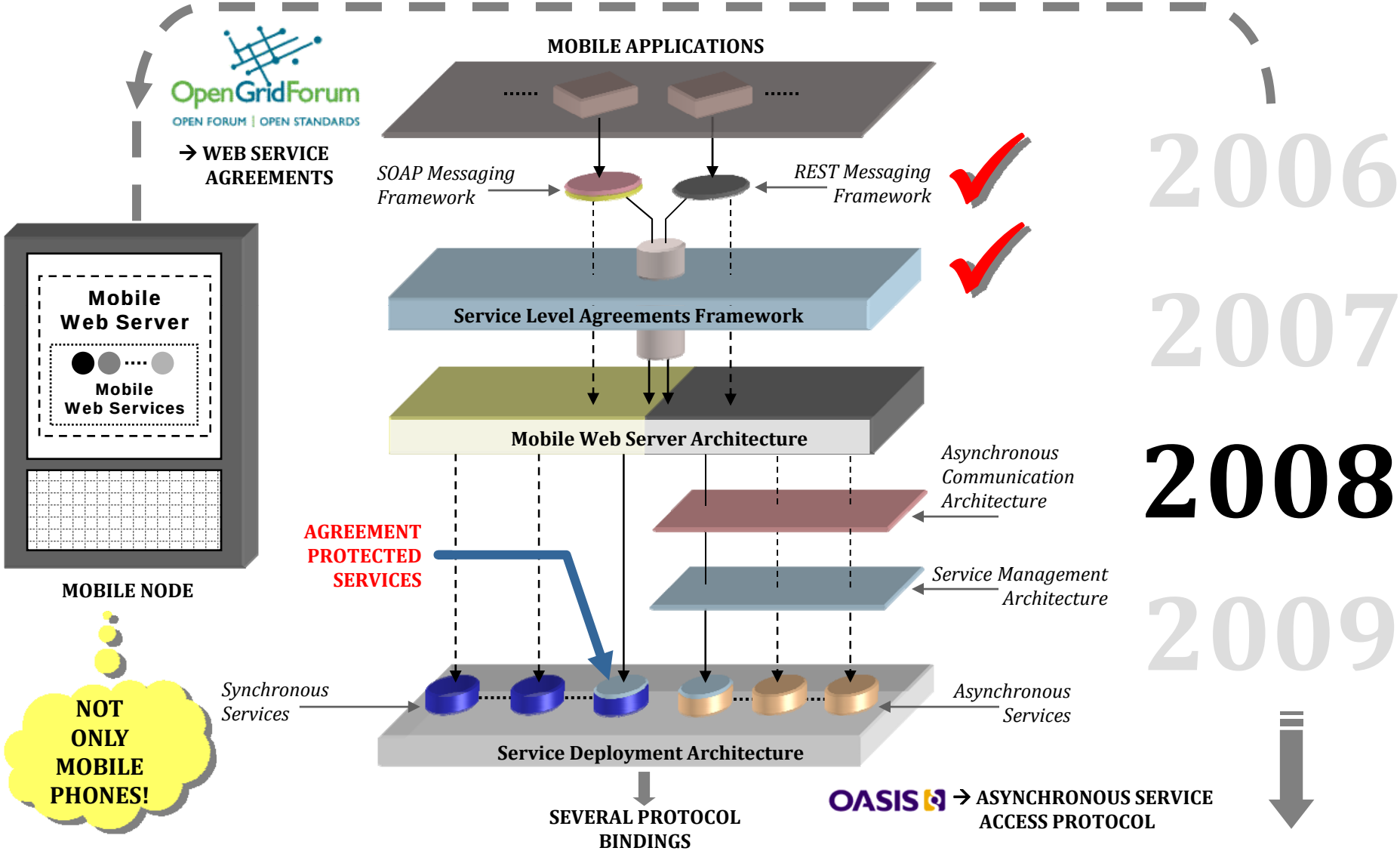
HTTP://[IP]:[PORT]/[SERVICE]/[TYPE]/[RESOURCE]/[OPERATION]

9091 *Asynchronous* *CreateInstanceRq*

THE SERVICE DEVELOPER DEFINES…
…WHICH METHOD MEANS WHAT ACTION

Research Scope → Quick Recap

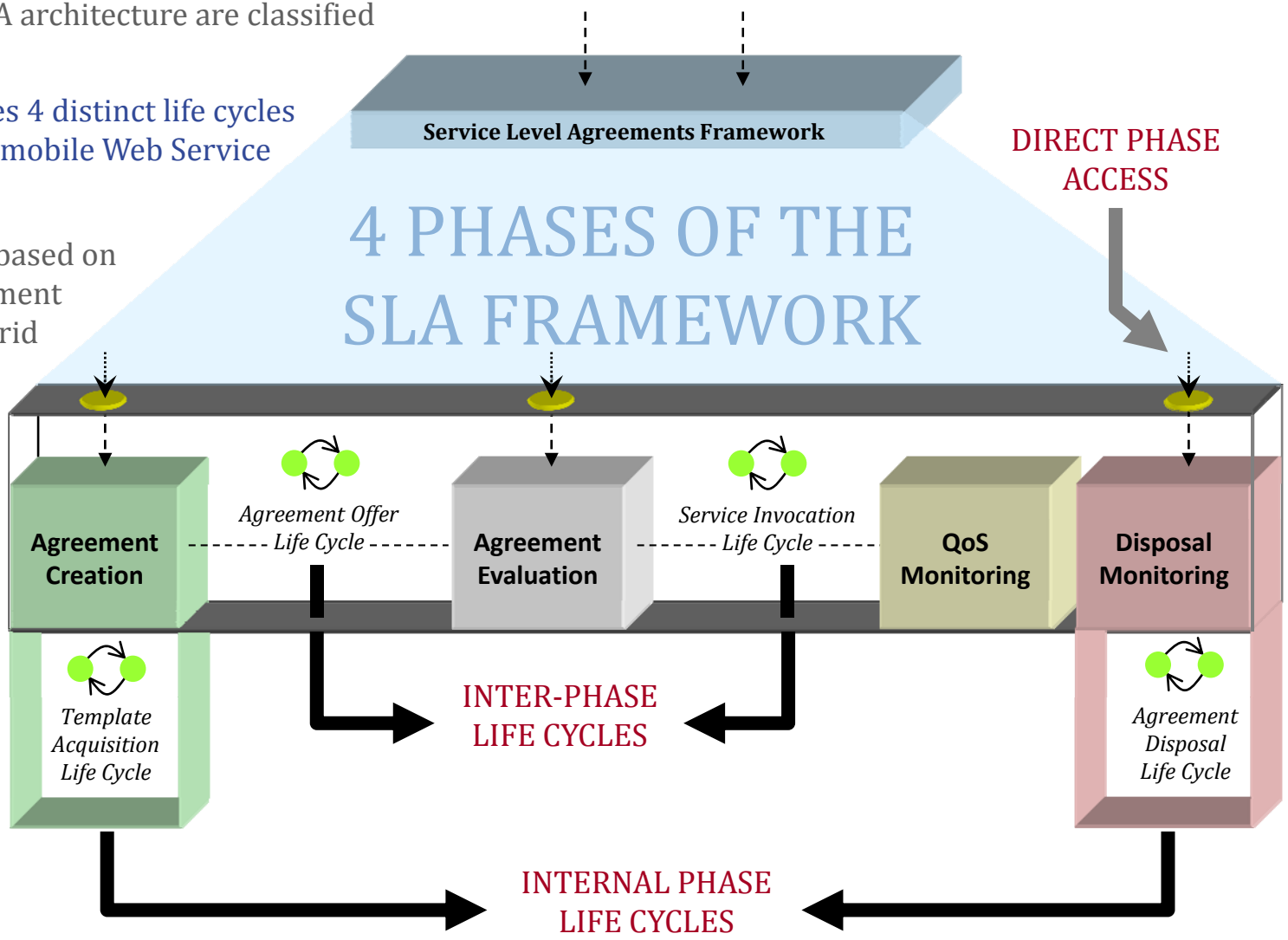
(Evolution of the Mobile Web Server Architecture)



Service Level Agreements (SLA) Framework

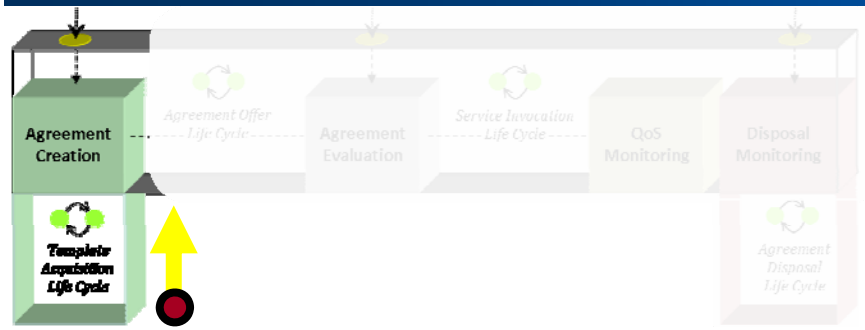
(Phases and Life Cycles)

- → The functions of the SLA architecture are classified into 4 distinct phases
- → The SLA phases executes 4 distinct life cycles based on the incoming mobile Web Service requests
- → The SLA negotiation is based on the Web Service Agreement standard of the Open Grid Forum
- → The standard is Optimized to define the SLA messaging for mobile nodes
- → The SLA framework is compatible with the REST and SOAP access interfaces
- → Based on the synchronous and asynchronous server architecture



Agreement Creation Phase

(Template Acquisition Life Cycle)



A) TEMPLATE WITH VALIDITY
B) MUST BE USED BEFORE EXPIRY
C) AUTOMATED DELETION
(DISPOSAL MONITORING)

MAY RESIDE IN THE CLOUD OR THE MOBILE NODE

A) READS & MANIPULATES THE TEMPLATE
B) GENERATES A UUID FOR THE CLIENT
C) SAVES A COPY AGAINST THE UUID
D) DISPATCHES THE TEMPLATE

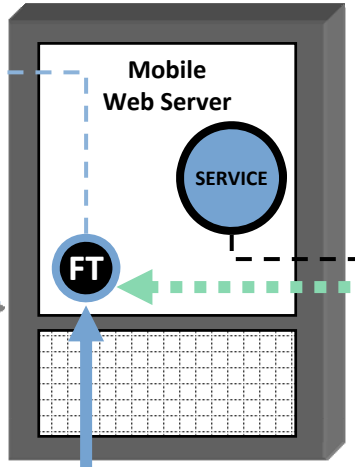
MOBILE WEB SERVICE CLIENTS

EXAMPLE REST REQUEST
http://mobile.comnets.de/FetchTemplate

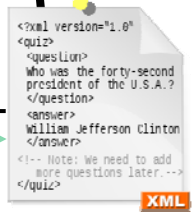
1 FETCH TEMPLATE

AGREEMENT TEMPLATE 2

UUID + AGREEMENT TEMPLATE



FETCH TEMPLATE SYNCHRONOUS MOBILE WEB SERVICE

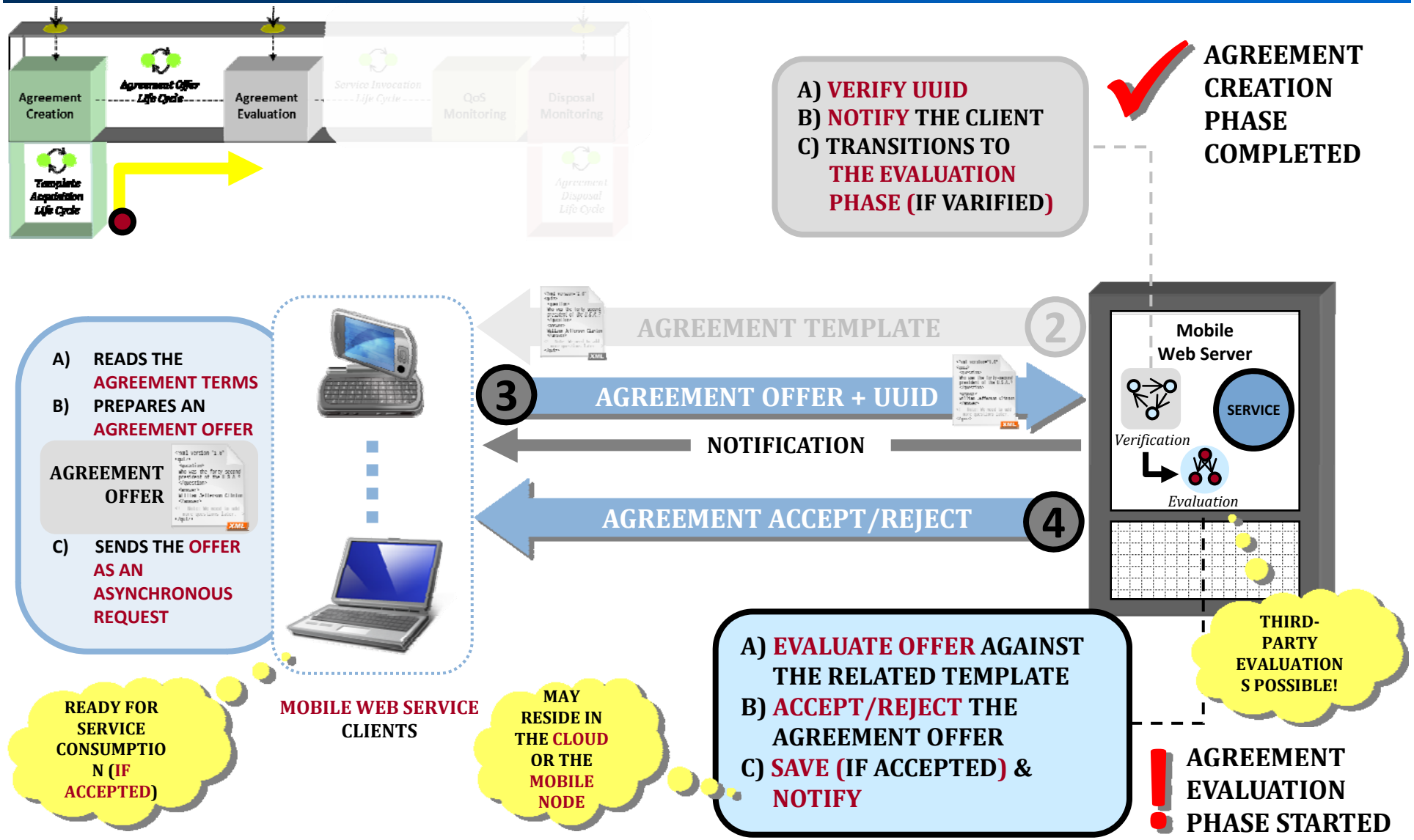


SERVICE PROVIDER'S AGREEMENT TEMPLATE

DERIVED FROM THE WEB SERVICE AGREEMENT STANDARD

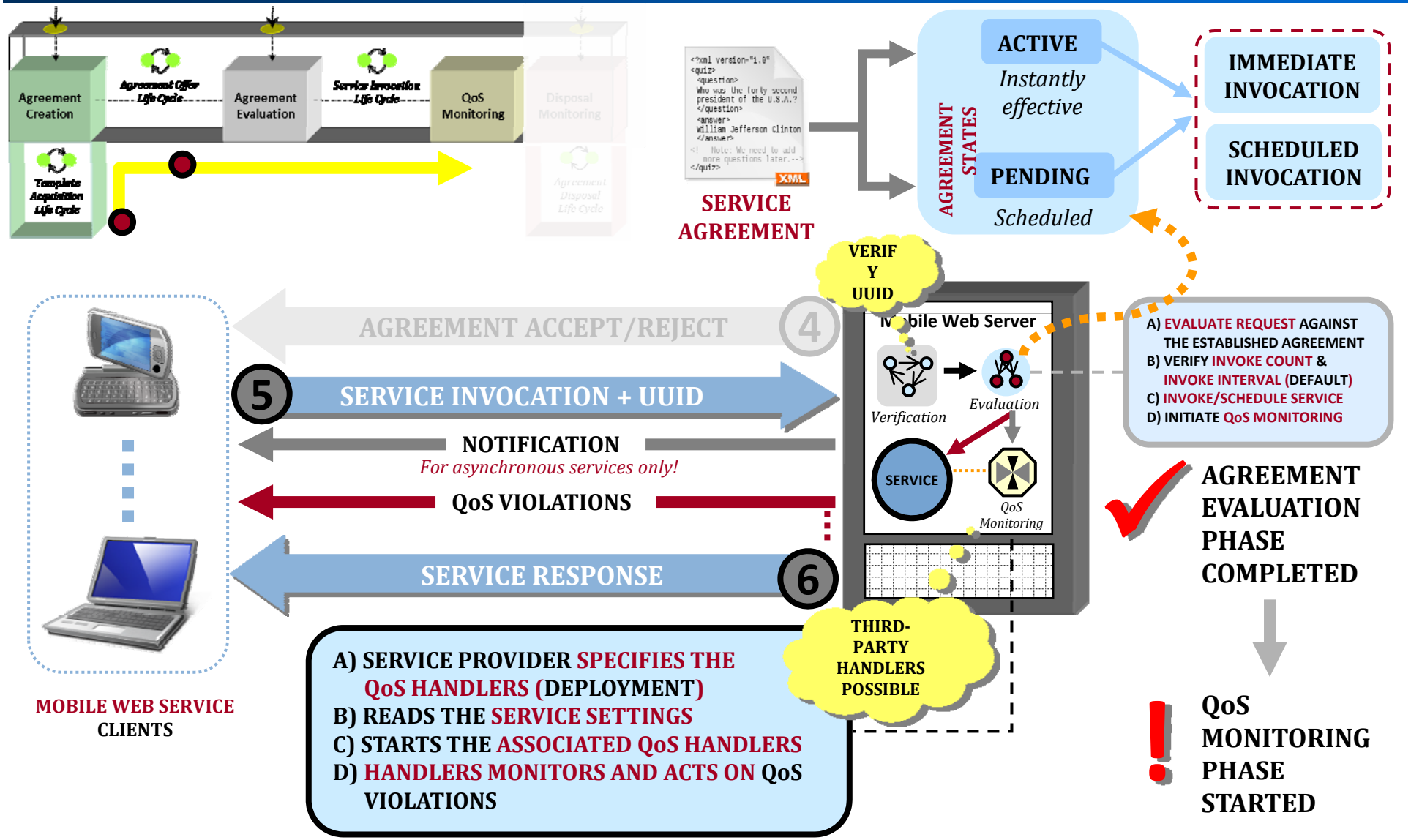
Agreement Creation and Evaluation Phase

(Agreement Offer Life Cycle)



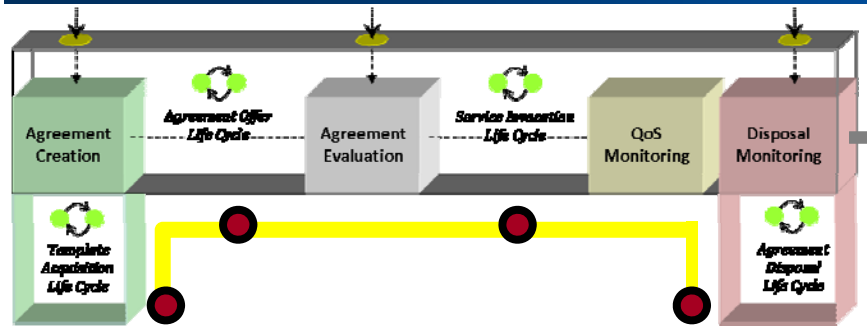
Agreement Evaluation and QoS Monitoring Phase

(Service Invocation Life Cycle)



Disposal Monitoring Phase

(Agreement Disposal Life Cycle)



CLIENT – CONTROLLED — SPECIFIED BY THE SERVICE PROVIDER DURING THE SERVICE DEPLOYMENT

AUTOMATIC DISPOSAL — DEFAULT

VERIFY UUID

FOR CLIENT – CONTROLLED ONLY

MOBILE WEB SERVICE CLIENTS

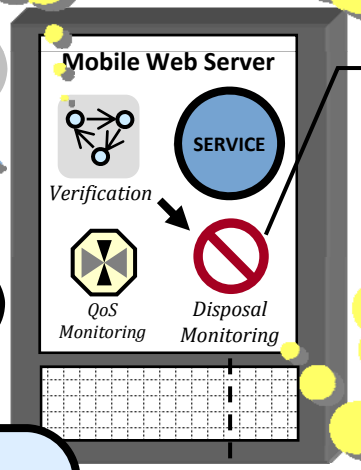
2 AGREEMENT TEMPLATE

5 AGREEMENT DISPOSAL + UUID

Client – controlled only!

6 DISPOSAL RESPONSE

Client – controlled only!



PRECONDITIONS

CLIENT IS ALLOWED | SERVICE NOT IN EXECUTION

TO AVOID QoS & AGREEMENT VIOLATION RISKS

A) PERIODICAL CLEANUP CYCLES IN AUTOMATIC DISPOSAL
B) LOOKS FOR EXPIRED AGREEMENTS & TEMPLATES
C) END DATE OF AN AGREEMENT AND INVOKE COUNT IS MONITORED AS EXPIRATION CRITERIA
D) DISPOSES TEMPLATES, AGREEMENTS & CLIENT RECORDS (UUID)
E) ONE PROCESS FOR ALL AGREEMENTS

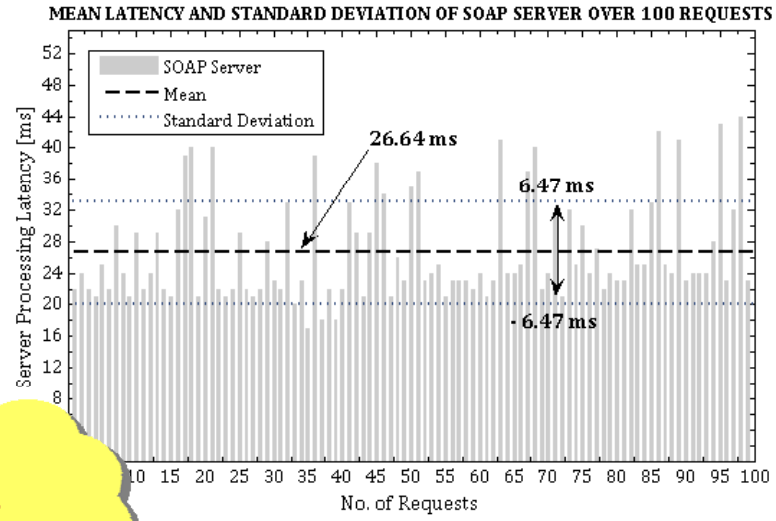
SERVICE PROVIDER MAY SPECIFY THE CLEANUP INTERVAL FOR AUTOMATIC PROCESS

Performance Evaluation - 1/3

(SYNCHRONOUS → Mean Server Processing Latency)

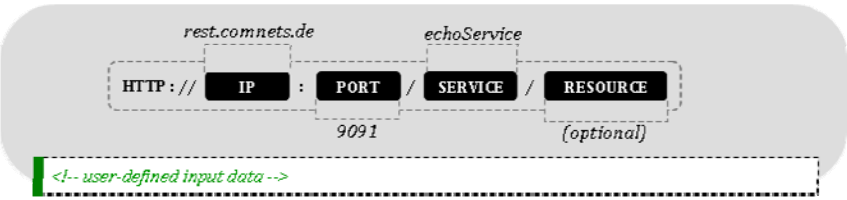
SOAP REQUEST

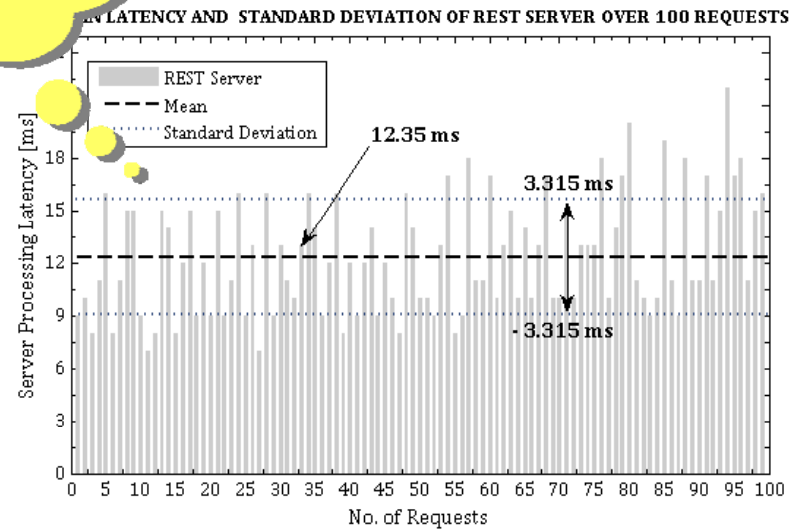
```
<SOAP-ENV:Envelope
  xmlns:xsi=http://www.w3.org/2001/XMLSchema-instance
  xmlns:xsd=http://www.w3.org/2001/XMLSchema
  xmlns:SOAP-ENC=http://schemas.xmlsoap.org/soap/encoding/
  xmlns:SOAP-ENV=http://schemas.xmlsoap.org/soap/envelope/>
  <SOAP-ENV:Body SOAP-ENV:encodingStyle=http://schemas.xmlsoap.org/soap/encoding/>

    <echoService xmlns="urn:Services" id="o0" SOAP-ENC:root="1">
      <ServiceType xmlns="" xsi:type="xsd:string">Synchronous</ServiceType>

      <ContextData xmlns="" xsi:type="xsd:ContextData">
          <!-- user-defined input XML elements -->
      </ContextData>

    </echoService>

  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```



~ 2 TIMES FASTER!

REST REQUEST





Performance Evaluation - 2/3

(ASYNCHRONOUS → Mean Server Processing Latency)

SOAP REQUEST

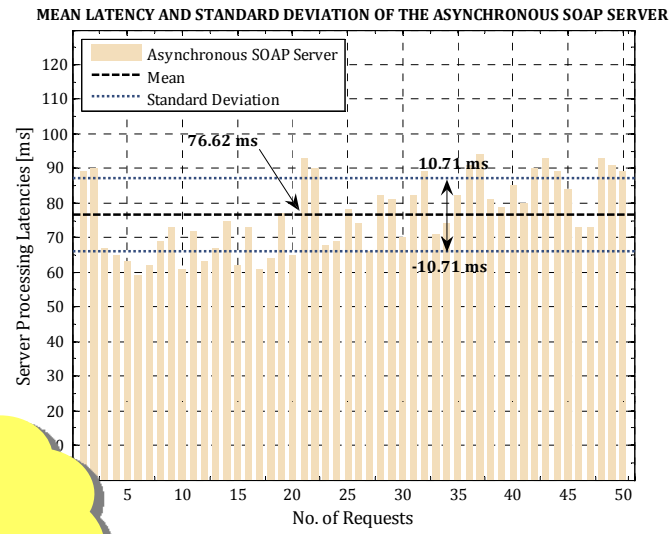
```
<SOAP-ENV:Envelope
  xmlns:xsi=http://www.w3.org/2001/XMLSchema-instance
  xmlns:xsd=http://www.w3.org/2001/XMLSchema
  xmlns:SOAP-ENC=http://schemas.xmlsoap.org/soap/encoding/
  xmlns:SOAP-ENV=http://schemas.xmlsoap.org/soap/envelope/
  xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing">
  <SOAP-ENV:Header>
    <wsa:To>http://soap.comnets.de:9090/soaprpc</wsa:To>
    <wsa:Action> urn:Services#aEchoService</wsa:Action>
    <wsa:MessageId>uuid:ffb471ec-194-1cb37664-125c53c2</wsa:MessageId>
    <wsa:ReplyTo>
      <wsa:Address>http://soap.comnets.de:40985/soaprpc</wsa:Address>
    </wsa:ReplyTo>
  </SOAP-ENV:Header>
  <SOAP-ENV:Body SOAP-ENV:encodingStyle=http://schemas.xmlsoap.org/soap/encoding/>

    <aEchoService xmlns="urn:Services" id="o0" SOAP-ENC:root="1">
      <ServiceType xmlns="" xsi:type="xsd:string">Asynchronous</ServiceType>
      <MethodType xmlns="" xsi:type="xsd:string">Factory</MethodType>
      <createInstanceRq xmlns="" xsi:type=":createInstanceRq">
            <StartImmediatelyxsi:type="xsd:boolean">true</StartImmediately>
            <ObserverEPR xsi:type="xsd:string">
                http://soap.comnets.de:40985/soaprpc
            </ObserverEPR>
            <Name xsi:type="xsd:string">aEchoService</Name>
            <Subject xsi:type="xsd:string">Echo Service</Subject>
            <Description xsi:type="xsd:string">aEchoService Description</Description>

            <ContextData xsi:type="xsd:ContextData">
                    <!– user-defined input XML elements –>
            </ContextData>

      </createInstanceRq>
    </aEchoService>

  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

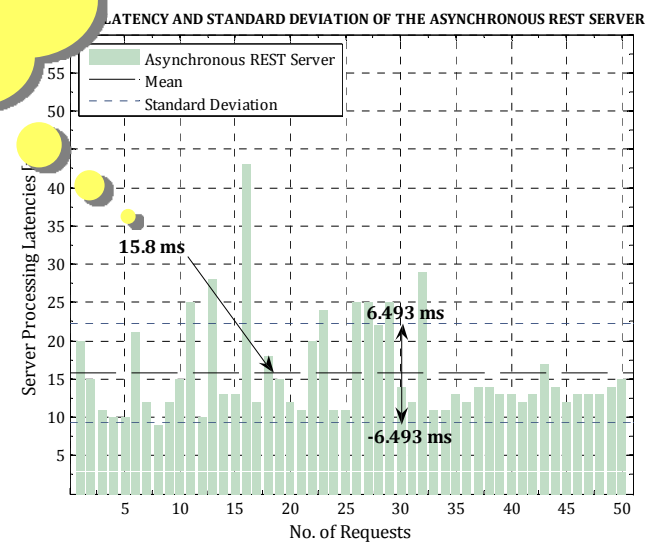
REST REQUEST

	rest.comnets.de		aEchoService		Factory		
HTTP : //	IP	: PORT	/ SERVICE	/ TYPE	/ RESOURCE	/ OPERATION	
		9091		Asynchronous		CreateInstanceRq	

```
<createInstanceRq xmlns="" xsi:type=":createInstanceRq">
    <StartImmediatelyxsi:type="xsd:boolean">true</StartImmediately>
      <ObserverEPR xsi:type="xsd:string">
          http://rest.comnets.de:9090/aEchoService/Asynchronous/Observer/createInstanceRs
      </ObserverEPR>
      <Name xsi:type="xsd:string">aEchoService</Name>
      <Subject xsi:type="xsd:string">Echo Service</Subject>
      <Description xsi:type="xsd:string">aEchoService Description</Description>

      <ContextData xsi:type="xsd:ContextData">
            <!– user-defined input XML elements –>
      </ContextData>

</createInstanceRq>
```

~ 5 TIMES FASTER

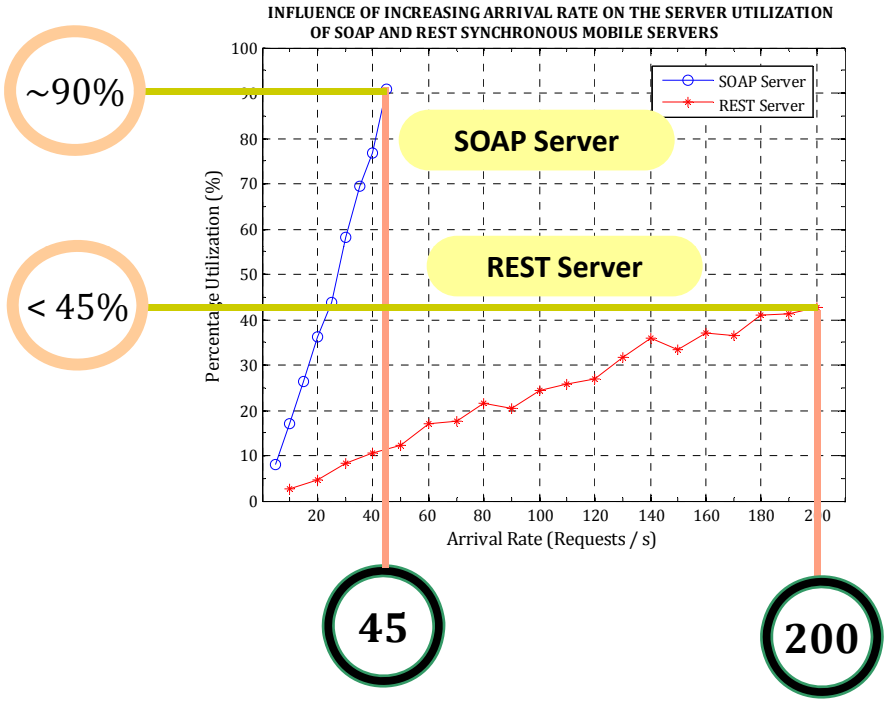




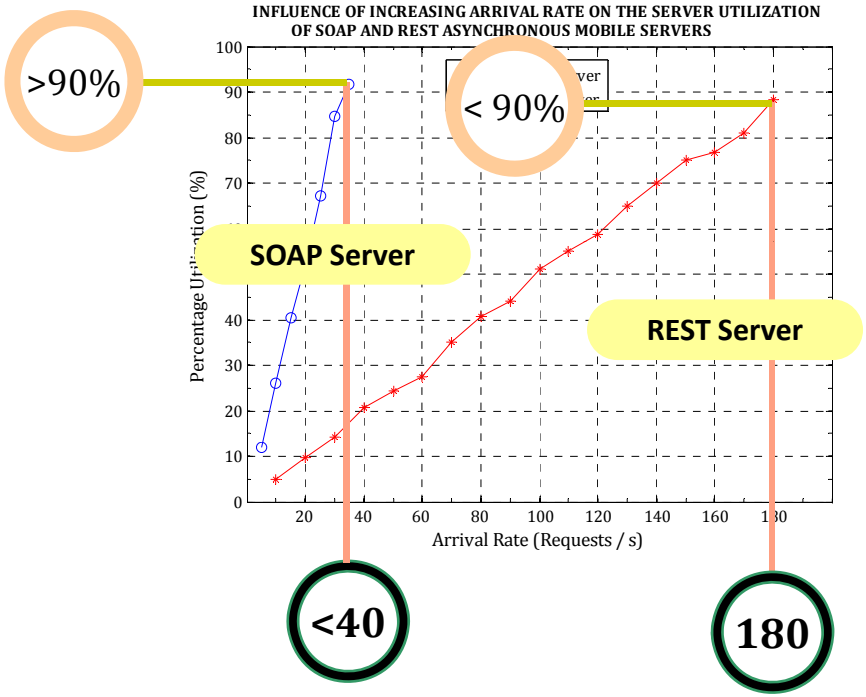
Performance Evaluation - 3/3

(Server Utilization Analysis)

SYNCHRONOUS
MOBILE WEB SERVER



ASYNCHRONOUS
MOBILE WEB SERVER



CAPACITY INCREASE → approx. > 8 TIMES

CAPACITY INCREASE → ~ 5 TIMES

> 400 Reqs./s possible! — REST SERVER — **> 180 Reqs./s possible!**

Conclusion

- SLA framework provides a layer of protection for mobile Web Services
- WS Agreement standard is optimized for SLA negotiations in mobile computing
- A comprehensive SLA architecture is developed on top of synchronous and asynchronous mobile Web Services
  - Classifies its function into 4 distinct life cycles
  - Supports the creation of active and pending agreements
  - Allows client-controlled and automatic disposal functions
- Supporting architecture for the QoS monitoring and agreement evaluation is provided
- Integration of the third-party QoS handlers and agreement evaluations is possible
- The multi-interfaced SLA negotiations are supported through SOAP and REST access interfaces
- Performance of the SLA framework is independent of REST and SOAP
- REST architecture of the Mobile Web Server shows promising performance improvements
  - SYNCHRONOUS SERVER: ~ 2 times faster, approx. > 8 times capacity increase
  - ASYNCHRONOUS SERVER: ~5 times faster, ~ 5 times capacity increase

Thank you for your attention !

RWTH AACHEN ComNets
RHEINISCH–WESTFÄLISCHE TECHNISCHE HOCHSCHULE AACHEN

RWTH Aachen University
Communication Networks
Kopernikusstr. 5
52074 Aachen
Germany

MSc.
Fahad Aijaz
Research Engineer

Tel +49 (0) 241 80 23923
Fax +49 (0) 241 80 22242
email fah@comnets.rwth-aachen.de
WWW http://www.comnets.rwth-aachen.de/fah.html

Questions are welcome!