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
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*

CONSUMER + PROVIDER

CONSUMER + PROVIDER

TRANSPARENT ACCESS

WEB SERVICES
- Specialized functions
- Internal process
- Access interface

Web Server

P2P Mobile Web Services

Fahad Aijaz, ComNets, RWTH Aachen University
Research Scope
(Evolution of the Mobile Web Server Architecture)

Mobile Web Server Architecture

MOBILE APPLICATIONS

SOAP Messaging Framework

REST Messaging Framework

Mobile Web Services

Asynchronous Communication Architecture

Service Management Architecture

Asynchronous Services

Synchronous Services

SEVERAL PROTOCOL BINDINGS

ASYNCHRONOUS SERVICE ACCESS PROTOCOL

Mobile Phones!

MOBILE NODE

Fahad Aijaz, ComNets, RWTH Aachen University
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**!

**RE** (presentational)

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

**S** (state)

Every resource changes the client’s state

**T** (transfer)

Resources are transferred using HTTP

World Wide Web is **RESTful**!

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

Architecture style...
...analogous to Client-Server

Not a Standard...
...but it uses standards!
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

Æ FETCH Æ UPDATE Æ INSERT Æ 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

Fahad Aijaz, ComNets, RWTH Aachen University
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 execute 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)

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

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

MOBILE WEB SERVICE
CLIENTS

MOBILE WEB SERVICE
SERVICE PROVIDER'S
AGREEMENT TEMPLATE
DERIVED FROM
THE WEB SERVICE
AGREEMENT STANDARD

Fahad Aijaz, ComNets, RWTH Aachen University
**Agreement Creation and Evaluation Phase**

(Agreement Offer Life Cycle)

- **A)** Verify UUID
- **B)** Notify the client
- **C)** Transitions to the Evaluation Phase (if verified)

**Agreement Creation**

- **A)** Reads the Agreement Terms
- **B)** Prepares an Agreement Offer
- **C)** Sends the offer as an asynchronous request

**Agreement Offer + UUID**

**Agreement Accept/Reject**

- **A)** Evaluate offer against the related template
- **B)** Accept/reject the Agreement Offer
- **C)** Save (if accepted) & Notify

**Mobile Web Service Clients**

- Ready for service consumption (if accepted)

**MAY RESIDE IN THE CLOUD OR THE MOBILE NODE**

**Third-Party Evaluation Possible!**

**Agreement Evaluation Phase Started**

**Agreement Creation Phase Completed**
Agreement Evaluation and QoS Monitoring Phase
(Service Invocation Life Cycle)

A) SERVICE PROVIDER SPECIFIES THE QoS HANDLERS (DEPLOYMENT)
B) READS THE SERVICE SETTINGS
C) STARTS THE ASSOCIATED QoS HANDLERS
D) HANDLERS MONITORS AND ACTS ON QoS VIOLATIONS
Disposal Monitoring Phase
(Agreement Disposal Life Cycle)

CLIENT – CONTROLLED
AUTOMATIC DISPOSAL

SPECIFIED BY THE SERVICE PROVIDER DURING THE SERVICE DEPLOYMENT
DEFAULT

FOR CLIENT – CONTROLLED ONLY

PRECONDITIONS
CLIENT IS ALLOWED
SERVICE NOT IN EXECUTION

TO AVOID QoS & AGREEMENT VIOLATION RISKS

SERVICE PROVIDER MAY SPECIFY THE CLEANUP INTERVAL FOR AUTOMATIC PROCESS

Disposal Monitoring + UUID

Client – controlled only!

Disposal Response

Client – controlled only!

Agreement Template

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
Performance Evaluation - 1/3
(SYNCHRONOUS → Mean Server Processing Latency)

SOAP REQUEST

```xml
<SOAP-ENV:Envelope
    xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
    soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
>
    <soapenv:Body>
        <env:EchoRequest>
            <env:ServiceType>Syncronous</env:ServiceType>
        </env:EchoRequest>
    </soapenv:Body>
</SOAP-ENV:Envelope>
```

REST REQUEST

```text
rest.comnets.de
<table>
<thead>
<tr>
<th>HTTP</th>
<th>IP</th>
<th>PORT</th>
<th>SERVICE</th>
<th>RESOURCE</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>IP</td>
<td>9911</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
```

~ 2 TIMES FASTER!

MEAN LATENCY AND STANDARD DEVIATION OF SOAP SERVER OVER 100 REQUESTS

- SOAP Server
- Mean
- Standard Deviation

- 26.64 ms
- 6.47 ms

MEAN LATENCY AND STANDARD DEVIATION OF REST SERVER OVER 100 REQUESTS

- REST Server
- Mean
- Standard Deviation

- 12.35 ms
- 3.315 ms
Performance Evaluation - 2/3

(ASYMETHRONOUS → Mean Server Processing Latency)

~ 5 TIMES FASTER
Performance Evaluation - 3/3
(Server Utilization Analysis)

SYNONYMOUS
MOBILE WEB SERVER

INFLUENCE OF INCREASING ARRIVAL RATE ON THE SERVER UTILIZATION
OF SOAP AND REST SYNCHRONOUS MOBILE SERVERS

~90%
< 45%

~90%
>90%

< 90%

45
200

> 400 Reqs./s possible!

<40
180

CAPACITY INCREASE \(\rightarrow\) approx. > 8 TIMES

ASYNCHRONOUS
MOBILE WEB SERVER

INFLUENCE OF INCREASING ARRIVAL RATE ON THE SERVER UTILIZATION
OF SOAP AND REST ASYNCHRONOUS MOBILE SERVERS

REST SERVER

> 180 Reqs./s possible!

< 45%

CAPACITY INCREASE \(\rightarrow\) ~ 5 TIMES

Fahad Aijaz, ComNets, RWTH Aachen University
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!

Questions are welcome!

MSc.
Fahad Aijaz
Research Engineer

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

RWTH Aachen University
Communication Networks
Kopernikusstr. · 5
52074 Aachen
Germany