

# Services, Enablers and Architectures: Definition of a connected Web 2.0 / Telco Service Broker to enable new flexible Service Exposure Models

N. Blum, T. Magedanz, F. Schreiner,

Fraunhofer FOKUS / Technical University of Berlin, Germany

Email:  [{niklas.blum|thomas.magedanz|florian.schreiner}@fokus.fraunhofer.de](mailto:{niklas.blum|thomas.magedanz|florian.schreiner}@fokus.fraunhofer.de)

Fraunhofer Institute for Open Communication Systems (FOKUS)

Kaiserin-Augusta-Allee 31, 10589 Berlin, Germany

Tel.: + 49 30 3463 7172 / Fax: + 49 30 3463 8172

Internet: [www.fokus.fraunhofer.de/ngni](http://www.fokus.fraunhofer.de/ngni) and [www.av.tu-berlin.de](http://www.av.tu-berlin.de)

**Abstract** — Modern telecommunication networks and classical roles of operators are subject to fundamental change. Many network operators are currently seeking for new sources to generate revenue by exposing network capabilities to 3rd party service providers.

At the same time we can observe that applications on the World Wide Web (WWW) are becoming more mature in terms of the definition of APIs that are offered towards other services. The combinations of those services are commonly referred to as Web 2.0 mash-ups.

This report describes our approach to include Next Generation Networks (NGN)-based telecommunications application enabler into Web 2.0 mash-ups by defining a policy-based service broker that mediates between 3<sup>rd</sup> party applications and service enablers and a JavaScript-based service exposure API that allows easy and straight forward integration of telecommunications enablers into such mash-ups.

The orchestration of several service enablers into one complex service for further exposure is anticipated.

**Index Terms** — Web 2.0, Service Exposure, NGN, IP Multimedia Subsystem, Service Delivery Platform, API, mash-up, Service Level Agreements, service composition

## I. INTRODUCTION

Telecommunications is at crossroads, the convergence of fixed and mobile telecommunications, cable networks, as well as the Internet leads into a global all-IP-based Next Generation Network (NGN). Through this ongoing process of the convergence of access networks and the existence of new players in the telecommunications market, traditional operators and carriers are seeking for new business models to increase their revenue.

The reuse of an extensible set of existing service components to create rapidly new market driven applications is a key aspect of telecommunications platforms since many years and gains a new momentum with the definition of dedicated application enablers for NGNs. Real-life examples are British Telecom's BT Web21C SDK [1], the Orange Partner programme [2], and Deutsche Telekom's Open Developer Portal [3] that define APIs to expose telecommunications specific core network functionalities to 3rd party service developers using Web Services.

This paper describes our approach of the realization of a policy-based service broker allowing the definition of request-specific policies serving as Service Level Agreements (SLA) for a service access gateway to applications based on a JavaScript API addressing the specific needs of Web 2.0 developers.

The paper is structured as follows: Section 2 provides a brief state of the art overview of the NGN functionality namely the IP Multimedia Subsystem (IMS) with focus on application enablers, the OMA Service Environment (OSE) and technologies associated to the term Web 2.0. Section 3 describes our concept of the policy-based service broker and enabler access gateway for Web 2.0 mash-ups. Through semantic annotations of service enablers, the dynamic service discovery for composite services is foreseen. The created platform was validated through a use-case at Open SOA Telco Playground at Fraunhofer FOKUS. We end the paper with a conclusion and outlook in section 4.

## II. RELATED STANDARDS AND TECHNOLOGY OVERVIEW

The following subsections describe emerging standards as the IMS, IMS enablers, OSE, and related technologies to the term Web 2.0 like Ajax and the mash-up service architectures.

### A. The IP Multimedia Subsystem

The 3GPP IP Multimedia Subsystem (IMS) provides the interfaces for interaction and underlying communication control infrastructure. The IP Multimedia Subsystem [4] is defined from 3GPP Release 5 specifications on as overlay architecture on top of the 3GPP Packet Switched (PS) Core Network for the provision of real time multi-media services.

Due to the fact that the IMS overlay architecture is widely abstracted from their interfaces, the IMS can be used for any mobile access network technology as well as for fixed line access technology as currently promoted by the European Telecommunications Standards Institute's (ETSI) Telecoms & Internet converged Services & Protocols for Advanced Networks (TISPAN) [5] within the Next Generation Network reference architecture definition.

The central session control protocols are the Session Initiation Protocol (SIP) [6] and Diameter [7]. The SIP Application Server (AS) is the service relevant part in the IMS. How multimedia applications are programmed is out of scope of the standardization committees. But the SIP AS needs to support well defined signaling and administration interfaces (3GPP ISC and Sh-interfaces) to connect to the standardized network architecture. The following figure 1 depicts the simplified IMS architecture.

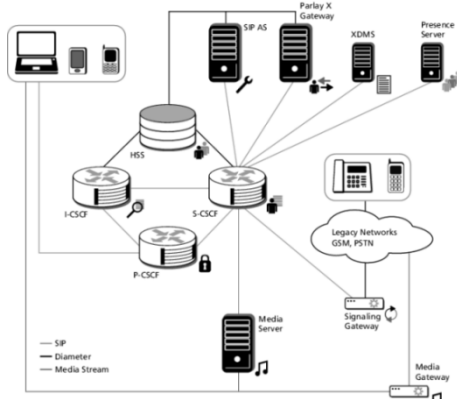


Figure 1: Basic IMS Architecture

The particular techniques and methodologies that are required to gain the advantages of these key functionalities are not completely new, but the IMS provides the first major integration and the interaction of all key functionalities.

### B. IMS Enabler

Similar to service independent building blocks which form part of the conceptual model for Intelligent Networks, the Open Mobile Alliance (OMA) has defined during service enablers for the IP Multimedia Subsystem. The idea was initially born during the specification of a Push-to-Talk over Cellular (PoC) [8] service, a walkie-talkie like communication service between several mobile peers. PoC uses Presence, Group Management and Instant Messaging as enablers to provide information to the users as well as to the PoC service. This lead alongside the standardization of PoC to the definition of Presence SIMPLE [9] for Presence and Instant Messaging and XML Documents Management (XDM) [10] for group and list management.

The definitions of several application service enablers by the OMA and the need for a general access function for 3<sup>rd</sup> party service access led to the specification of the OMA Service Environment (OSE) [11] as a common abstraction layer for IMS-based NGNs.

It defines an enabler layer which incorporates specific enabler components that offer northbound interfaces to services that implement certain application logic. These applications either reside at the operator domain or are hosted at a 3<sup>rd</sup> party domain. An enabler component can

either be part of the OSE or the OSE can act as an application overlay that offers interfaces to other service enabler functions. Figure 2 illustrates the proposed architecture by the OMA.

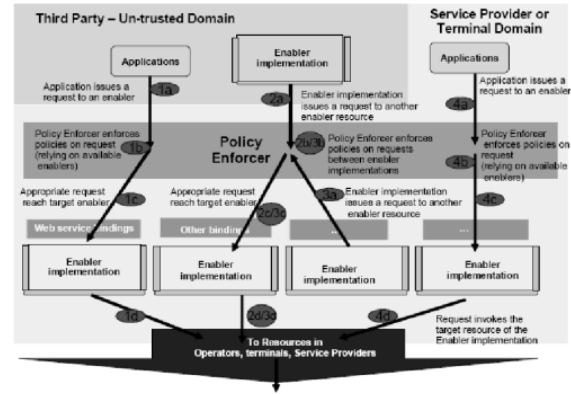


Figure 2: OSE Architecture

Basically, an OSE incorporates Web Services interfaces and translates Web Services requests either directly into enabler logic or to an enabler specific protocol. OMA does at the point of writing not standardize any mapping to a specific middle-ware messaging technology but leaves this open to the implementation of specific service environments. An enabler could as well be a non standardized implementation towards a specific telephony platform or an IN platform. Furthermore an enabler can be implemented towards several protocols to provide a network converging functionality. NGN technologies with legacy networks, e.g. a messaging enabler can be mapped to SIP, short message peer-to-peer protocol (SMPP) [12] to communicate with a SMS-C for sending out SMS and MM-7 [13] to communicate towards a MMS-C.

The Policy Enforcer or Policy Evaluation, Enforcement and Management (PEEM) component as the function has been named officially by the OMA can be used to intercept service requests from a foreign domain as well as from any other service requestor and apply certain rules (policies) on them that are stored at a policy repository. Policies can basically be used for the authorization of requests meaning that service invocation requests that are intercepted by the PEEM are checked for valid authorization and authentication. PEEM may furthermore be used to define enabler capabilities for exposure based on request policies. Depending on the business model different charging rules may also be applied for service requests through specific policies. In this regard the definition of policies may be considered as the expression of Service Level Agreements between a network operator and a service provider.

A PEEM function forms the main integral component of an OMA Service Environment and provides additional functionality based on the definition policy

for the OMA enabler concept. A PEEM may serve as an access gateway authentication function but its capabilities are much greater in regard of the orchestration and manipulation of enabler capabilities. The OMA names two different Policy Expression languages, Common Policy by the Internet Engineering Task Force (IETF) [14] for authorization policies and Business Process Execution Language (for Web Services) WSBPEL 2.0 defined by Advancing Open Standards for the Information Society (OASIS) [15] for the orchestration of enablers.

### C. The WWW and Web 2.0

The WWW is by nature community-driven, not only with regard to content, but also from a technical point of view. Simple protocols like HTTP, description languages as HTML and CSS, and architecture paradigms (e.g. Representational State Transfer - REST) made the Web successful and its simplicity is the decisive factor for the developer community's acceptance of extensions to the Web technology stack. Web 2.0 is less a question of novel technologies in the Web technology stack, but rather a question of how existing technologies are applied to create services tailored to user communities.

In this respect, client-side active scripting and the inherent capability of HTML to integrate content from different sources play a major role. Active scripts are shipped along with the web content to control content presentation and interactivity. The object based programming language ECMAScript [16], better known as JavaScript, is today's mostly used scripting language for Web pages. In addition to operations on the associated document, all noteworthy Web browsers allow active scripts to self-reliantly utilize the HTTP client interface in a pared-down configuration. This feature of active scripts to access their origin server for the exchange of any message is referred to as Asynchronous Java Script (Ajax) [17]. Although the Ajax API introduces with the XMLHttpRequest [18] just one new language construct the amount of available developer tools based on Ajax show its current importance.

The varieties of client-server interaction, given to active scripts through Ajax include Remote Procedure Call (RPC) and Publish-Subscribe. The representatives for RPC over HTTP in favor of the developer community are XML-RPC [19] and JSON-RPC [20]. The major difference between both can be found in the representation of request and response, i.e. marshalling of method calls and objects. While JSON-RPC utilizes a light-weight, non-standard syntax, XML-RPC is based on W3C's XML. However, RPC frameworks for Ajax usually require a respective counterpart on the server-side. In practice, tool support for the selected backend platform (e.g. .NET, J2EE, PHP) is often the decisive criterion for the selection of a RPC framework.

### D. Semantic Service Annotation

SOA based SDPs for NGNs should not only allow for third party access to network functions and service enablers, as previously explained, they should also enable for rapid service discovery, service creation, service composition and deployment.

What is missing is the ability to describe a service or service enabler in a commonly understandable and comprehensive way, so that on the one hand services can easily be discovered and on the other hand services can be composed more and more automatically.

First, a profound, extensive and extensible ontology that describes a Telco / NGN service (like in [21]) is required.

Second, a service registry (such as Universal Description, Discovery and Integration (UDDI) or Electronic Business Extensible Markup Language (ebXML)) for storing and retrieving service information is needed.

Third, the description of each service has to be dynamically fed into and discovered from the service registry. For annotating Web Service Description Language (WSDL) files, there are currently approaches Most importantly the Web Services Semantics (WSDL-S) [22] approach and the Semantic Annotations for WSDL and XML Schema (SAWSDL) [23]. Furthermore, approaches like the Web Service Modeling Ontology (WSMO) [24] and, based on the Web Ontology Language (OWL), the Semantic Markup for Web Services (OWL-S) [25].

## III. WEB 2.0 / TELCO SERVICE BROKER

In this section we describe our design and implementation of a service broker to enable telecommunications specific service access for Web 2.0 mash-ups using policies for SLA definition.

### A. Parlay X Gateway

FOKUS' Open Communication Server for Parlay X (OCS-X) [26] is an implementation of the Parlay X Web Services specification for telecommunication networks. These interfaces provide a network abstraction through a very simple and easy to use API based on Web Services technology, which can be used remotely from 3<sup>rd</sup> party domains and service providers. The OCS-X uses the current Parlay X Version 2.2 [27]. Parlay X defines a set of powerful yet simple, highly abstracted, building blocks of telecom capabilities that developers and the IT community can both quickly comprehend and use to generate new applications.

Each Parlay X building block is abstracted from the set of telecom capabilities exposed by the Parlay X APIs. The capabilities offered by a building block may be homogeneous (e.g. call control only) or heterogeneous (e.g. mobility and presence). A building block will usually not be application-specific.

Figure 3 provides an overview of the Parlay X Gateway as a network abstraction component in a Web 2.0 mash-up service.

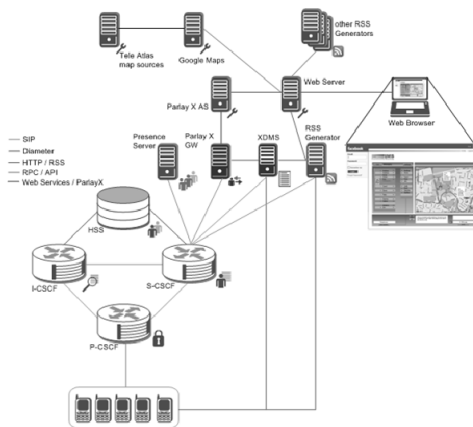


Figure 3: Architecture of a Telco-enabled Web mash-up

### B. Implementation of the Web 2.0 / Telco Gateway

The realization of the above architecture has been transferred to Parlay X Web Services enabling IMS functionality. An overview about this concrete gateway is depicted in Figure 4. The implementation is Java- and JavaScript based and the communication between client and gateway has been realized using JSON-RPC.

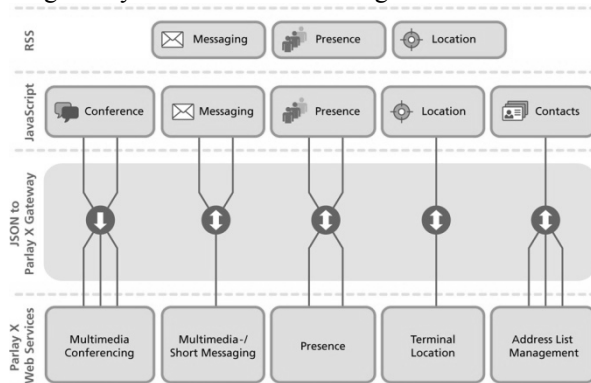


Figure 4: JSON-to-ParlayX Gateway

The open Java to JavaScript Object Broker (JABSORB) [28] is used on the client-side, in order to transparently send JSON requests to remote Java objects. On the server-side, the jabsorb framework is used providing a particular Servlet that makes simple Java objects accessible via JSON-RPC that are automatically called on the accordant request.

### C. Implementation of the Service Broker Component

The Service Broker is implemented as several independent modules that are accessible through OMA compliant interfaces, especially PEM-1 and PEM-2 interfaces to trigger policy evaluation requests for intercepted service requests and for policy retrieval from the policy repository.

In our setup, an OMA XDMS serves as policy repository; the policies are compliant to the IETF's

Common Policy [14] format with service specific extensions that are allowed by the specification. The following figure 5 provides part of a policy for a conferencing interface:

```
<openpe:ruleset
xmlns:openpe="urn:ietf:params:xml:ns:openpe-policy">
<openpe:rule id="1234">
<openpe:conditions>
<openpe:identity />
<openpe:valability>
<openpe:from>11.03.2008</openpe:from>
<openpe:until>11.03.2009</openpe:until>
</openpe:valability>
<openpe:service id="http://localhost:8082/OCS-
X/services/MultimediaConferenceService" />
</openpe:conditions>
<openpe:actions>
<allowAccess>true</allowAccess>
</openpe:actions>
<openpe:transformations>
<openpe:serviceOperation id="createConference">
<openpe:parameter id="conferenceType">
<openpe:noregx>$value!="Message"</openpe:noregx>
</openpe:parameter>
<openpe:parameter id="charging">
<openpe:noregx>$value.amount==0.1</openpe:noregx>
<openpe:noregx>$value.currency="$"</openpe:noregx>
</openpe:parameter>
<openpe:parameter id="conferenceOwner">
<openpe:noregx>$value!="sip:alice@ims.test"</open
pe:noregx>
</openpe:parameter>
<openpe:parameter id="maximumNumber">
<openpe:noregx>$value<6</openpe:noregx>
</openpe:parameter>
</openpe:serviceOperation>
<openpe:serviceOperation id="inviteParticipant" />
</openpe:transformations>
</openpe:rule>
</openpe:ruleset>
```

Figure 5: Policy Example

Policies are applied on three levels of a service request:

1. Service Identifier (e.g. IP address, domain, etc.)
2. Service user (e.g. SIP URI, service identity)
3. Service actions and allowed parameters

Therefore it is possible not only to define service specific policies but to correlate those with user specific service settings (e.g. privacy rules), too. The above policy example in Figure 6 provides a policy for all three levels. It grants general service usage between March 2007-2008 for user sip:alice@ims.test to establish conferences with media type 'anything but messaging' with up to 6 users. Furthermore, a charging event for the services usage may be triggered.

This policy defines a service/user specific behaviour and can be considered a Service Level Agreement between an operator and a 3<sup>rd</sup> party service provider. It is not necessary to define for each service enabler/service relationship all three levels; we consider level 1 and level 3 policies as most important for the definition of effective and flexible policies for service usage.

The following Figure 6 depicts our architecture of the policy-based service broker:

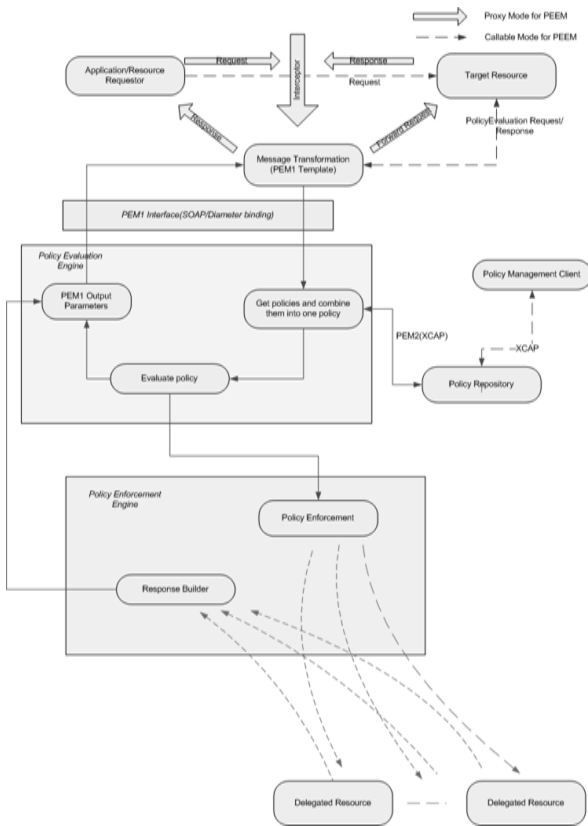


Figure 6: Service Broker Architecture

The most important entities are the service interceptor(s), the policy evaluation component and the policy enforcement engine.

Interceptors may be applied on all critical (Web Services) interfaces within an operator's architecture, either internal or exposed interfaces for necessary policy evaluation upon service requests. Intercepted SOAP messages will be transformed through Extensible Stylesheet Language Transformation (XSLT) into OMA defined PEM-1 interface document format. Requestors are then capable of using PEM-1 callable SOAP interface bindings to request policy evaluation processed by PEEM.

The policy evaluation component retrieved applicable policies from the XDMS and executes a matching process between the request and the retrieved policies. In case that the evaluation process returns simple access policies, the response is initiated directly to the interceptor including either access granted or denied information upon which the interceptor either forwards the request to the corresponding service enabler or replies directly to the request originator with an 'access denied' HTTP 403 Forbidden code.

In case the policy requires delegation to other service enablers fulfil request/policy combination, the policy evaluation process triggers a policy enforcement process. Therefore, the policy enforcer entity has access

to semantically enriched service description files (WSDL-S) that are stored at an UDDI repository.

#### D. Validation

We have validated our service broker implementation using a more complex conferencing scenario. The policy states that conferences for our service are only allowed for participants with IMS presence status online. Therefore, the broker needs to delegate the service request first to the presence enabler to retrieve the presence status of the URI that is part of the parameters for the `inviteParticipant` request to be able to decide whether to allow or deny the transmission for these requests to the conference service enabler. The following figure 7 depicts the simplified signalling flow.

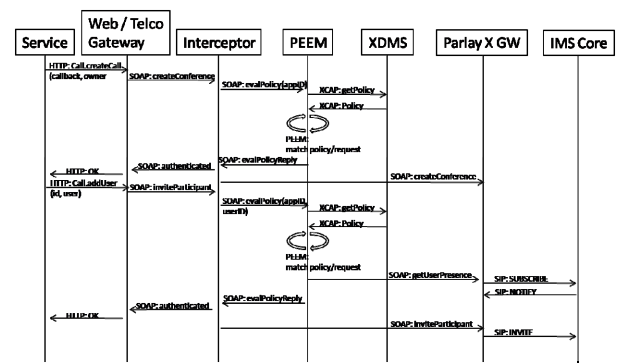


Figure 7: Signalling diagram

Not depicted in the above diagram is the consecutive call for policies in the context of delegation since each delegated request by PEEM needs to be policy-enforced as well.

#### IV. CONCLUSION & OUTLOOK

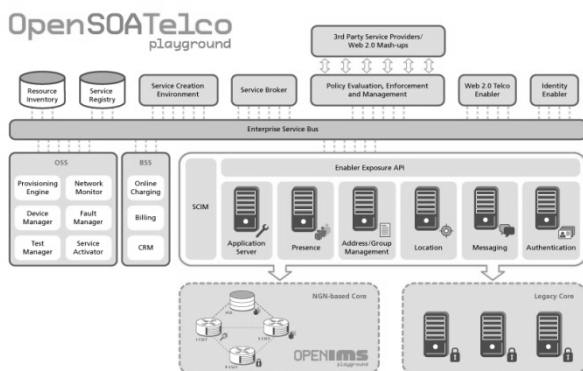
SOA principles have been used inside telecommunications domains for many years, although different terms have been used over the last decades to describe the idea of realizing a programmable network to provide an open market of services. Today, Web Services based APIs including emerging Web 2.0 interfaces represent the state of the art in SOA-based telecommunications, which are going to be integrated with the emerging IMS.

We have depicted in this report our blueprint for service broker component based on SOA principles that takes the latest standards and concepts in telecommunications into account. The major work is based on a policy evaluation and enforcement engine that provides the capability to match service requests to operator defined policies to allow fine granulated service usage policies to which we refer as SLAs between service enablers, 3<sup>rd</sup> party services and users.

Future work will concentrate on the dynamic service discovery and service orchestration based on semantics, to allow orchestration of many (operator-specific)

service enablers automatically for more complex services.

The Open SOA Telco Playground [29], as shown in figure 8 serves as the environment for our NGN SDP activities and is the north bound extension of the FOKUS Open IMS Playground [30] founded in 2004.



**Figure 8: Open SOA Telco Playground by FOKUS**

As the IP Multimedia System (IMS) is considered today as the unifying architectural framework for the provision of seamless IP based services on top of converging networks and the south bound foundation for many Service Delivery Platforms (SDPs), the Open SOA Telco Playground provides the possibility to experience a Service Oriented Architecture (SOA) on top of converging networks.

## V. ACKNOWLEDGMENT

The research described in this paper has been performed within the Fraunhofer FOKUS Next Generation Networks Infrastructures (NGNI) competence center in close cooperation with the chair for next generation networks (AV) within the electrical engineering and computer sciences faculty of the Technische Universität Berlin.

## REFERENCES

- [1] BT. Web21C SDK. <http://web21c.bt.com/>.
- [2] Orange Partner. <http://www.orangepartner.com/>.
- [3] Deutsche Telekom. <http://developer.telekom.de>
- [4] 3GPP. TS 23.228. IP Multimedia Subsystem (IMS). Stage 2 v.7.10.0. 2007.
- [5] ETSI. <http://www.etsi.org/tispan/>.
- [6] H. Schulzrinne, et al.. IETF RFC 3261. SIP: Session Initiation Protocol. 2002.
- [7] P. Calhoun. IETF RFC3588. Diameter Base Protocol. 2003.
- [8] Open Mobile Alliance (OMA). Enabler Release Definition for Push-to-talk over Cellular. Candidate Version 2.0 – 11 Dec 2007. 2007.
- [9] Open Mobile Alliance (OMA). Presence SIMPLE Architecture Document. Approved Version 1.0.1 – 28 Nov 2006. 2006.

- [10] Open Mobile Alliance (OMA). XML Document Management Architecture. Candidate Version 2.0 – 24 Jul 2007. 2007.
- [11] Open Mobile Alliance (OMA). OMA Service Environment. Approved Version 1.0.4 – 01 Feb 2007. 2007.
- [12] Short Message Peer-to-Peer Protocol Specification v5.0, 19 Feb 2003
- [13] 3GPP, TS 23.140 V6.14.0, “Multimedia Messaging Service (MMS); Functional description; Stage 2”, June 2006
- [14] H. Schulzrinne et al., “RFC 4745 Common Policy: A Document Format for Expressing Privacy Preferences”, February 2007
- [15] OASIS Web Services Business Process Execution Language Version 2.0, <http://www.oasis-open.org/committees/wsbpel/>, April 2007
- [16] ECMAScript Language Specification, 3<sup>rd</sup> Edition, 1999, <http://www.ecma-international.org/publications/standards/Ecma-262.htm>
- [17] J. J. Garrett. Ajax: A new Approach to Web Applications. 2005.
- [18] The XMLHttpRequest Object, W3C Working Draft, October, 2007 <http://www.w3.org/TR/XMLHttpRequest/>
- [19] D. Winer; XML-RPC Specification, June, 1999, <http://www.xmlrpc.com/spec>
- [20] JSON-RPC Specification 1.1, Working Draft, August, 2006, <http://json-rpc.org/wd/JSON-RPC-1-1-WD-20060807.html>
- [21] Semantic Interfaces for Mobile Services (SIMS), European 6<sup>th</sup> Framework Program STREP Project, <http://www.ist-sims.org>
- [22] Akkiraju, R., Farrell, J., Miller, J., Nagarajan, M., Schmidt, M.T., Sheth, A., Verma, K.: Web Service Semantics - WSDL-S. W3C Member Submission (November 2005)
- [23] Farrell, J., Lausen, H. (eds.): Semantic Annotations for WSDL and XML Schema. W3C Recommendation (August 2007)
- [24] Bruijn, J.d., Bussler, C., Domingue, J., Fensel, D., Hepp, M., Keller, U., Kifer, M., Konig-Ries, B., Kopecky, J., Lara, R., Lausen, H., Oren, E., Polleres, A., Roman, D., Scicluna, J., Stollberg, M.: Web Service Modeling Ontology (WSMO). W3C Member Submission (June 2005)
- [25] Martin, D., Burstein, M., Hobbs, J., Lassila, O., McDermott, D., McIlraith, S., Narayanan, S., Paolucci, M., Parsia, B., Payne, T., Sirin, E., Srinivasan, N., Sycara, K.: OWL-S: Semantic Markup for Web Services. W3C Member Submission (November 2004)
- [26] FOKUS Open Communication Server, <http://www.open-ims.org/ocs-x>
- [27] Parlay X. <http://www.parlay.org/en/specifications/pxws.asp>, 2008
- [28] Jabsorb Framework. <http://jabsorb.org/>
- [29] FOKUS Open SOA Telco Playground, <http://www.opensoaplayground.org>
- [30] FOKUS Open IMS Playground, <http://www.open-ims.org>