# Creating New Communication Services Efficiently

By Anett SCHÜLKE,* Ernö KOVACS,* Heinrich STÜTTGEN,* Amjad AKKAWI,* Marcus KUHNEN,* Arabella RIU* and Florian WINKLER*

**ABSTRACT** Mobile Network Operators have invested massively into the roll-out of their UMTS or 3rd Generation Mobile Networks. Now they are changing their focus from infrastructure roll-out to the deployment of new attractive data and multimedia services in 3G networks. Within the Open Mobile Alliance (OMA) requirements for a future Mobile Service Architecture have been defined. We describe the Service Creation Environment developed at NEC's European Network Laboratories (NL-E) in collaboration with NEC's Mission Critical Business Operations Unit, meeting the requirements as set forth by the OMA. Key aspects of our SCE middleware platform are policy control and security technologies, helping application developers to tie OMA service enablers efficiently into new user friendly mobile services. With this technology, service providers can develop new communication services faster and open their network for new services developed by 3rd party service providers.

**KEYWORDS** **Service creation environment, Service delivery platform, UMTS (Universal Mobile Telecommunications System), 3rd generation mobile network, Policy control, Service security, Service enablers, OMA (Open Mobile Alliance)**

## 1. INTRODUCTION

Rapid introduction of new services into telecommunication networks has been an unsolved problem for several decades. On the other hand, the Internet has shown that fast creation and worldwide provisioning of new services is possible. Within 3rd Generation Mobile Networks, more and more Internet concepts are being introduced into the telecommunication environment.

At the same time the industry is standardizing services and service enablers within the Open Mobile Alliance (OMA). OMA has defined a reference architecture that defines how basic services can be combined to form new and advanced services. This high-level architecture abstracts from concrete technologies so that we need to map the architectural concepts to specific technologies. Software Engineering is now providing mature object-oriented development technologies, e.g. Java Enterprise Edition (J2EE), CORBA, and Web Services, that can be used to build large, carrier-grade distributed systems. Even more important, these techniques are known to a large and growing number of developers, thus providing the opportunity to move the magic of "service creation" from the hand of a few skilled experts into the reach of software engineers educated with these OO technologies and applying good engineering discipline.

Thirdly, the basic OMA service enablers, i.e. Presence, Group Management, Push-to-Talk over Cellular, and Instant Messaging can be built as re-usable building blocks. NEC is already offering several of these building blocks as products.

What is missing on the way to even more attractive new services is the glue that ties network and service enablers together and makes service creation easy and efficient. This glue is provided by NL-E's "Service Creation Environment" (SCE). The SCE is a development and runtime environment that provides to engineers (a) concrete interfaces (bindings) to service enablers (as open APIs), (b) a programming model that fits the asynchronous nature of telecommunication services, (c) an implementation model allowing to compose more complex services from simpler ones, (d) a fine-grained authentication and access control, and (e) specialized application servers providing a rich and flexible development environment. In addition the SCE will offer graphical or very high-level language tools that support specification, testing, and deployment of new services.

In the following, we first look at the Open Mobile Alliance and particularly the OMA Service Environment (OSE). Based on this, we introduce our Service

*NEC Network Laboratories Heidelberg

Integration Framework, an architecture for a complete service creation environment. We describe how network services and basic service enablers can be accessed, how the service integration framework ties together existing services and new services, and how the policy management allows fine-grained control over the service usage. At the end we show through an example how new services can be created elegantly and give an outlook on open issues.

## 2. OPEN MOBILE ALLIANCE

The OMA[1] is the leading standardization organization for mobile service related technologies. Founded in 2002 it has incorporated several existing fora like WAP Forum, Location Interoperability Forum (LIF), Wireless Village Initiative, Mobile Games Interoperability Forum (MGIF), Mobile Wireless Internet Forum (iMWIF) and many other.

### 2.1 OMA Service Environment

The OSE is a conceptual environment that includes service enablers, interfaces to applications that make use of these enablers, interfaces to Service Providers' Execution Environment (e.g. software life cycle management) and the interfaces to invoke and use underlying capabilities and resources for enabler implementations. The IP Multimedia Subsystem (IMS) (as defined by 3GPP) is a Session Initiation Protocol (SIP) based IP multimedia infrastructure that provides a complete platform for globally interoperable IP multimedia services — especially for the mobile environment. The IMS Service Control (ISC) interface allows applications, i.e. commercial services, to access IMS capabilities. IMS provides service-enabling functions and IP transport, which are relevant for the OSE as defined by OMA. Such applications may either use OMA enablers or may also use directly any IMS functions. Applications only using the IMS functions directly are not in the scope of the OSE. In addition, service enabler implementations may also make use of IMS capabilities, e.g. charging, authentication, service management, etc. IMS related applications/enablers could use OSE capabilities in addition to IMS capabilities.

The main requirements defined for an OSE can be summarized as

- Mechanisms for authentication, authorization, federated identity, subscription management,
- Single sign-on/log-out,
- Accounting and charging handling,
- Provisioning of services, service enablers, and user parameters,
- Service registration, discovery mechanisms, policy management,
- O&M support (including service monitoring).

The conceptual architecture of the OSE is shown in **Fig. 1**. In OMA a service performs useful work for

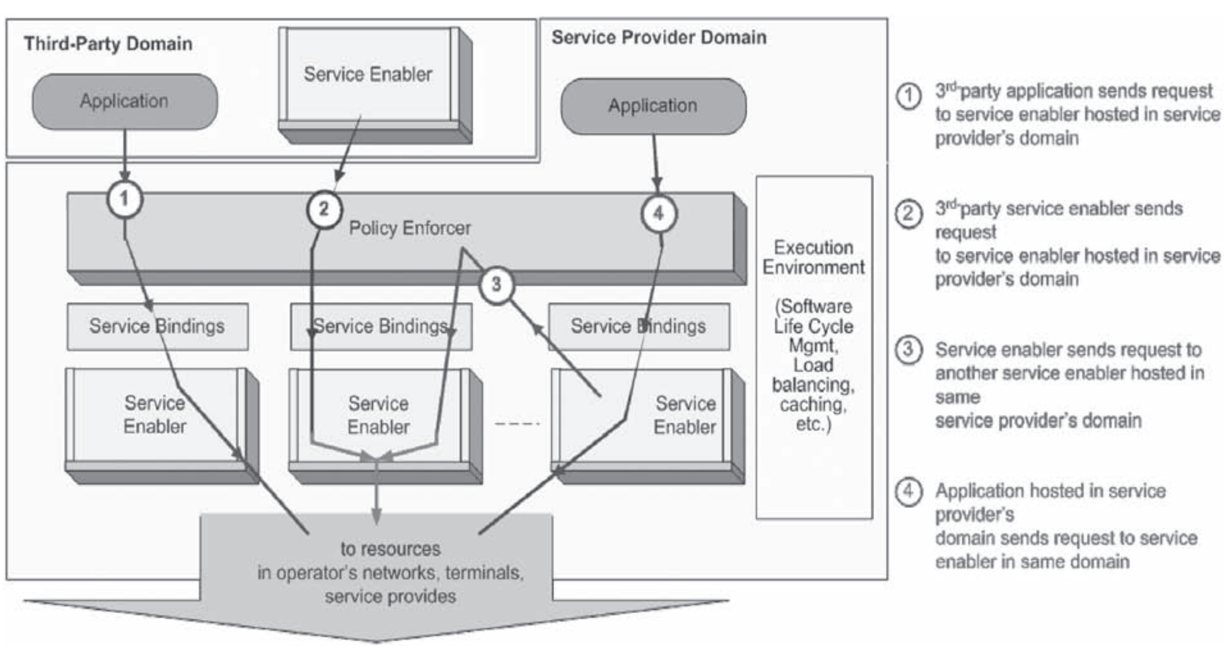


**Fig. 1 Description of OSE and request flow.**

users or service providers. Applications are defined as software or hardware implementation of a related set of functions. In this sense, applications are the way to access one or more services using service enablers. Applications may be deployed within or outside a service provider's domain or in a terminal. Service enablers provide standard access to network and terminal resources, and to other service enablers using service bindings.

The OSE deals with service requests from applications (located inside or outside the service provider's domain), as well as requests from other service enablers. Enablers shall be re-usable in the same or in different service provider domains. Enablers can be exposed by various binding mechanisms (e.g. Web services, standardized APIs, CORBA, etc.)

To secure existing networks and their operations, the OSE may protect network resources by a policy management allowing fine grain control of access and system behavior. The policy management is also used when application access service enablers or when service enablers interact with each other. Policy mechanisms are used

- to enforce access control by dynamic policy evaluation.
- to manage the use of network resources e.g. through appropriate charging, logging and enforcement of user privacy or preferences.
- to allow extensibility by offering service platform controlled delegation between enablers.

Figure 1 also presents a simplified view of the OSE request flow from various entities towards the network.

### 2.2 Existing Service Enablers and Bindings

Today's NEC's product portfolio contains a variety of service enablers. In the context of IMS the enablers for Presence, Group Management, Text Messaging, Voice over IP (VoIP) and Push-to-Talk over Cellular (PoC) are of specific importance.

Presence introduces a new communication style as well a new communication service. It provides knowledge about a user's status and availability and his/her willingness for communication. This concept can also be applied to event and object status control. One of the communication styles leading to increased service revenues will be group communication. It is a basic part of various services like PoC or conferencing. For that, a standardized solution is currently being completed by the OMA consortium. Besides e-mail and SMS service, instant messaging and text chat as real-time services are becoming more and more popular. Text chat over IMS provides the possibility for interoperability across different implementations. SIP-based VoIP services are available over IMS as well as proprietary SIP solution. Push-to-Talk over Cellular (PoC) is an extension of a basic VoIP service. The major difference in user experience is the introduction of the floor control (right-to-speak) concept over a half-duplex mode for PoC service. PoC can be viewed as a very basic voice conferencing service.

The above enabler descriptions are not exhaustive. Many other enablers in the areas of video calls, content-based services, location-based services and others are under development. However, in the context of this paper we have focused on IMS related enablers. Any extension to service enablers beyond IMS is possible.

## 3. DESIGN OF THE SERVICE CREATION ENVIRONMENT

### 3.1 Functional Concept

NL-E has designed a "Service Integration Framework (SIF)" as specific solution for a Service Creation Environment allowing the creation of commercial end-user services. The SIF is a solution for

- Service delivery integrating service enablers by providing access control and controlled service usage scenarios,
- Dynamic service orchestration for value-added service creation by means of dynamically integrating single or composed services into application logics via policy handling and mediation control, and
- Application provisioning by supporting subscription handling, profiling, advertisements, etc.

The SIF supports voice, data and content services in a terminal and network independent manner. This framework additionally needs to offer features like service exposure, service provisioning, security or service management. **Figure 2** introduces the functional view of the SIF, focusing on the server-side.

Applications are deployed into the Application Execution Platform and executed there. The execution environment can be a server as well as terminal-based. The platform contains the interfaces to the built-in functions as illustrated in Fig. 2. When an application implements a service, it can utilize established service expose mechanisms like push, pull, streaming or session-based interactions to be accessible from the end-user terminal. Provisioning supports the delivery of these new commercial services to end-users. It provides advertisement and discovery mechanisms. It manages the subscription to the service, provides needed resources for service usage (e.g. the terminal software), and personalizes the advertisement as well as the usage of existing user preferences. Applications can use internal interfaces in order to compose services into advanced applications. The composition mechanism is controlled via the transaction management functions. The functions for transaction handling provide the controlled access and delegation mechanism using a "policy/resource" model. In this model every request to a resource is governed by a set of policies.

### 3.2 Architecture

The current SIF implementation emphasizes the compliance with the requirements specified in the OSE standardization. The mapping of the functional architecture into the logical view which is directly comparable with the OSE architecture view is shown in **Fig. 3**.

The SIF includes the Service Provisioning, a core unit called Service Enabling Framework and the supporting Development Tools.

The Service Provisioning performs mainly the tasks for the application subscription process, user and service profile management, application advertisement, discovery and download facilities. For this, the SIF can integrate the Service Provisioning Platform MUSE (also developed at NL-E ) [4] or any other suitable provisioning platform.

The Service Enabling Framework (SEF) is the core within the SIF and combines the functions for service exposure, AAA, system and transaction management and capability control. It is divided in three modules: Application Control, Capability Control and Management functions.

(1) Framework Capability Control

Service enablers are made accessible to applications in the form of binding objects. These binding objects are created in an implementation layer called the Framework Capability Control (FCC). The FCC is a complex unit representing the networks abstraction layer towards the higher layers. It contains specific binding modules to offer SIP-based and non-SIP-based applications access to the IMS by initiating an IMS agent and respective media manager. Depending on the scenario used in a specific application, a combination of various service binding agents in single or

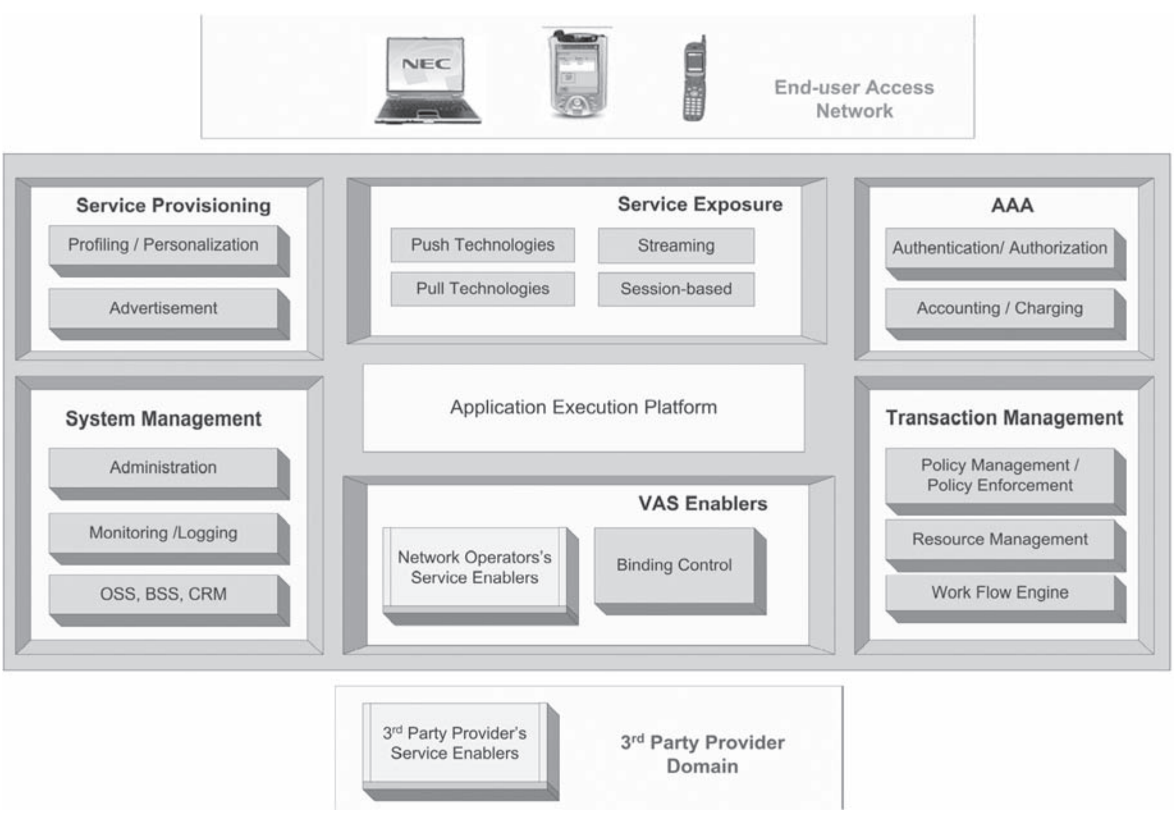


**Fig. 2. Functional view of the Service Integration Framework (server-side).**

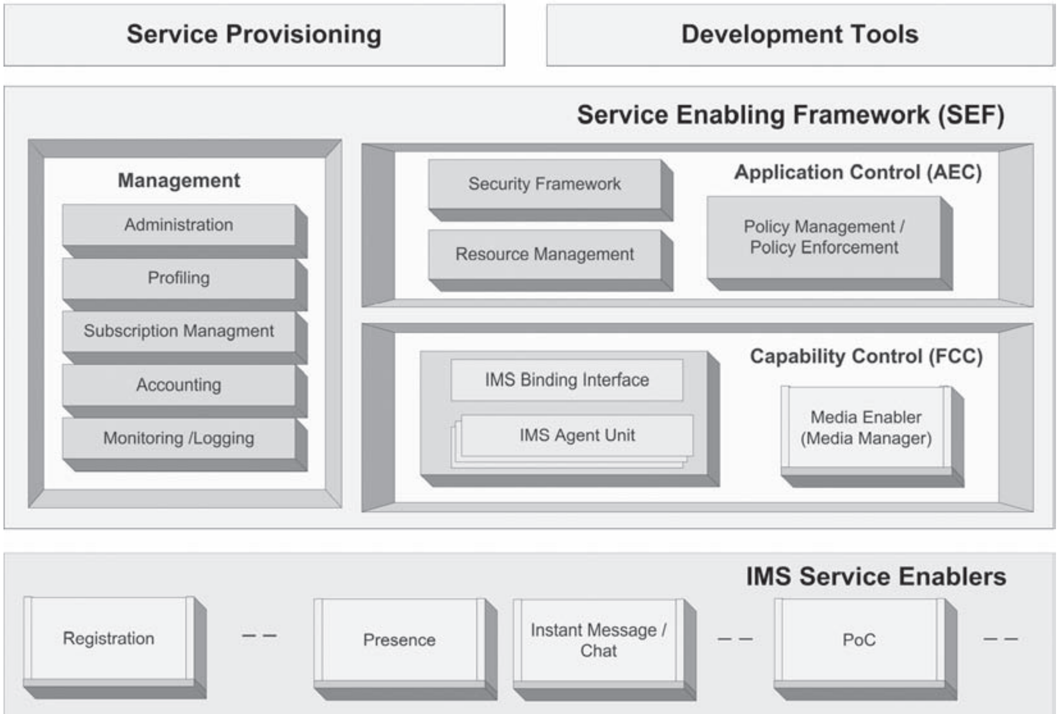


**Fig. 3 SIF logical view in line with OMA OSE.**

multi mode can be initiated. In multi mode, an application (having an Application Id, AID) allocates multiple bindings to the IMS system, e.g. to announce the presence of multiple users or to manage different groups. Using so called Item IDs (IID), the multi-mode application can reference to the binding object for the respective network resources. **Figure 4** illustrates the concept for an application which has to utilize several different IMS applications like Presence or PoC as well as the location service and related group management functions.

Using the binding objects it provides an effective shielding of the service enablers from the application execution environment. The binding can be realized by Web Services, JAIN APIs, HTTP or SIP messages. The NEC implementation of the OMA enablers uses the SIP SIMPLE protocol messages for communication. NL-E has implemented the client side of the protocols and the JAIN API for the binding object interfaces.

(2) Policy Management and Enforcement (PEEM)

The PEEM provides a fine grain transaction management for requests from applications or third party service enablers towards the underlying service enablers and resources. It is the basis for enforcing

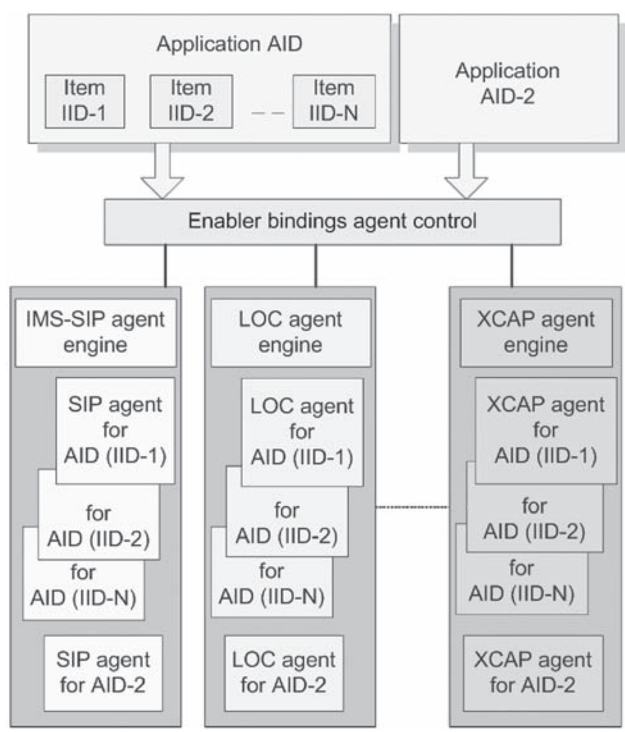


**Fig. 4 Service binding agent handling for multi-mode applications.**

security policies and allows for an easy upgrade and extension of the Service Enabling Framework.

The PEEM follows a modular and distributable design. It consists of: (a) Policy management: allows to create and to maintain policies for registered resources based on request/response types, requestors/responders roles and access rights. (b) Resource management: allows dynamic registering of resources to the policy framework so that access to them can be controlled. A resource is an abstraction for any kind of service-providing entity. (c) Policy Decision Module: assigns the semantically correct subset of policies to the incoming requests/responses. (d) Policy Enforcement: enforces the applicable policies during request processing. (e) Event Management: provides the mechanisms to handle events issued by service enablers and forwarded to the applications.

The usage of the PEEM can be described by two distinct usage patterns shown in **Fig. 5**. Using the Proxy Usage Pattern (A) allows placing the Policy Enforcer between the Requestor and the Target Resource. In this way, policies can be applied to requests. As shown in the picture, a request can be policed and also re-directed to other resources in order to execute additional processing steps (e.g. charging, load control, or special processing for premium customer). The callable usage pattern (B) is used for access control by the security framework. The incoming requests are sent to the security module which authorizes its actions using the defined policies.

The described Policy Management greatly enhances the extensibility and flexibility of the Service Enabling Framework. It provides a way of transparently integrating security, charging, load balancing and other strategies into the system. Using the dynamic request processing, an upgrade of existing functions as well as the addition of new functions is easily possible. Using these modern object-oriented technologies, the SCE is prepared for future changes to the execution environment.

### 3.3 Implementation

The current SEF realization is based on J2EE application servers. The communication between the external application and the SEF is realized over RMI/JMS. The SEF intrinsic communication uses JMS, RTP/RTCP and JAIN for handling the request chain from the application down to the resource binding. The IMS service enabler binding is based on SIP/SIMPLE for the control plane and the RTP/RTCP for

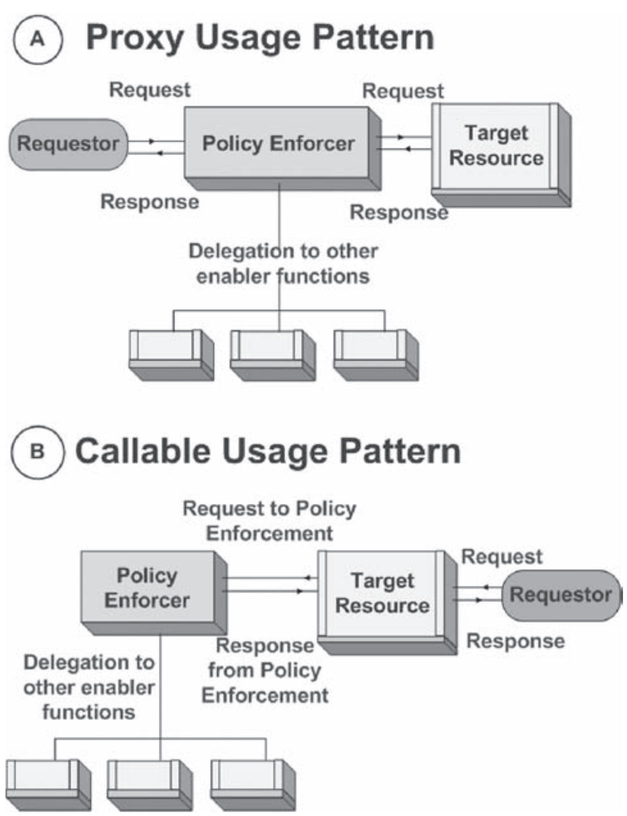


**Fig. 5 PEEM usage patterns.**

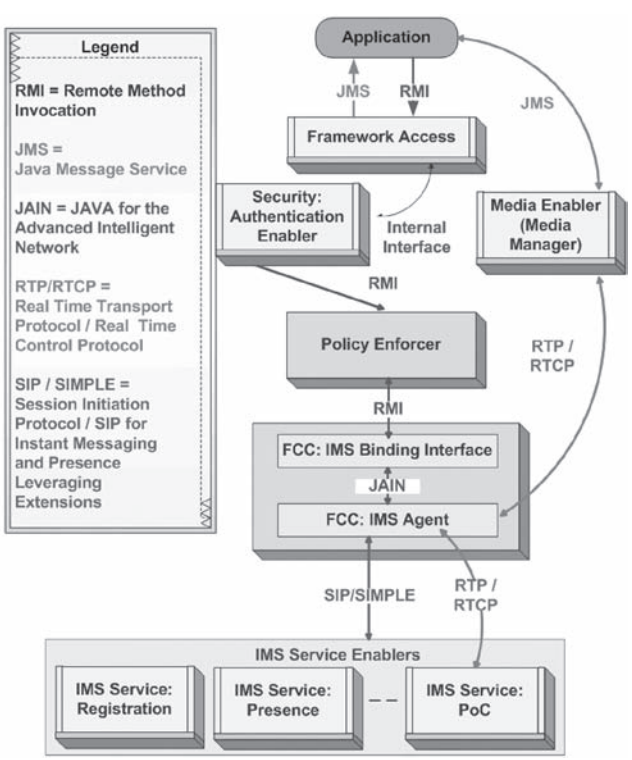


**Fig. 6 Protocol utilization for the SEF.**

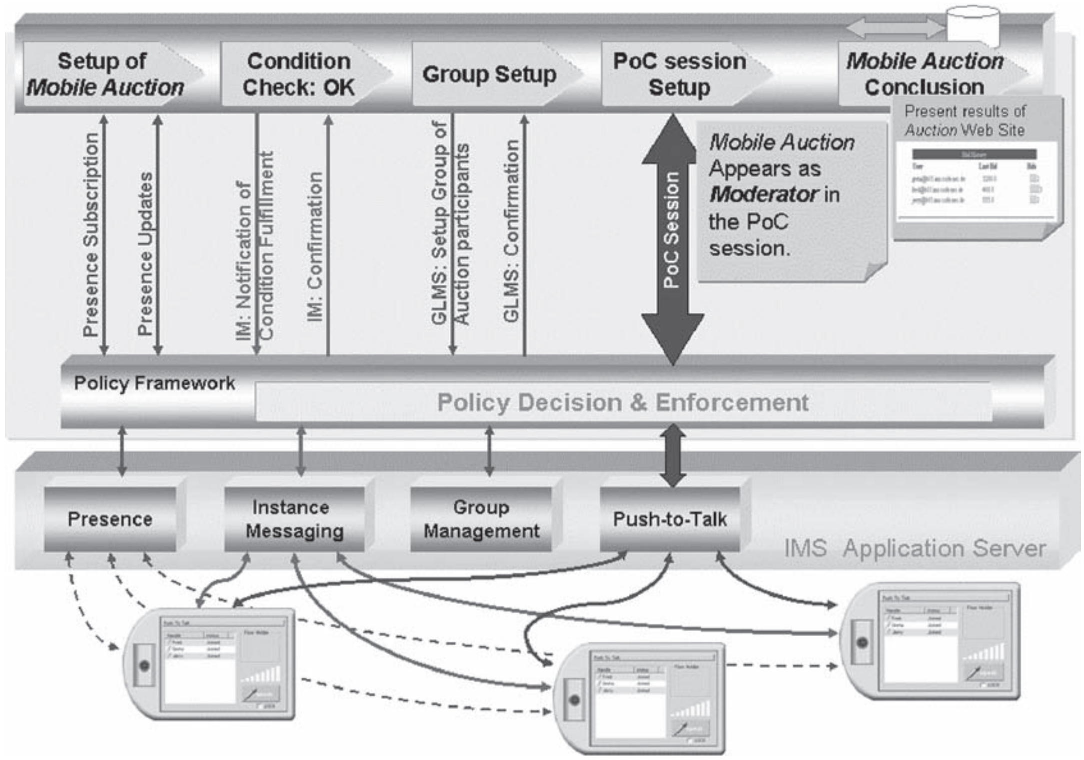


**Fig. 7 Mobile auction scenario.**

the user plane (media plane). **Figure 6** depicts the protocol utilization as given in the single-mode application scenario.

### 3.4 Mobile Auction: an Advanced Service Example

In the following paragraph we explain how one specific new service, a Mobile Auction can be created using the SCE. In a real world auction ("live" auction) the auction process continues until the highest bid is found. Internet auctions differ from that model by having a fixed time deadline on which the highest bid has to be stated. That prevents to find a "real" price for the object and also lacks the excitement of bidding in a "live" auction.

The "Mobile Auction" starts like any Internet auction. The moderator sets a condition when the Internet auction switches to the "live" auction mode. The condition includes different context variables like how many bidders are online (using the presence enabler), how high the current bid is and also a time when the live auction mode shall start earliest. For example, the earliest start can be set to a special time in the afternoon, provided that at least 2 bidders are online, and a minimum bid of 20 Euro has been made (**Fig. 7**).

The Mobile Auction is realized as an application using the SIF for IMS service access. When the online conditions are met, the Mobile Auction initiates a PoC session with all bidders currently online. During the auction, the moderator (seller) is repeating the current offer using the voice communication and text using IM over the auctions' Internet portal. Bidders can use both media for bidding. When no more higher offer are arriving, the auction is over. To satisfy legal requirements, the auction can be recorded in order to have a proof of who has bought the item.

This example shows how we can develop an application that combines different OMA service enablers, e.g. Presence, IM, GM, and PoC, to provide an advanced and more attractive service.

## 4. OPEN CHALLENGES

The SCE provides the ability to create new services out of existing building blocks in a fast, reliable and secure manner. The system is currently at the prototype level and there are still some open challenges.

### 4.1 Tool Support

In the current version, the service logic needs to be implemented by a software engineer. In the future, services shall be created by business developers, not by software engineers. It is therefore useful to extend the SCE with tools for creating new services, e.g. based on UML, BPEL or a Model-Driven Architecture.

Furthermore, the SCE can be extended with tools for testing services offline or in controlled environments.

### 4.2 New Service Enablers

We expect that new service enabler will be specified and implemented. A promising new horizontal service enabler is a context service enabler that allows to access and process of context information.

### 4.3 Service Interworking

The SCE provides the means to allow third party service provider access to the service enablers of a network. In order to make this reliable and secure, we need to examine security mechanism for service interworking and for managing the network between the service execution environment and the third party network (e.g. security, QoS, etc.)

## 5. CONCLUSION

The world of mobile network operators and service providers in Europe is different from the situation in Japan in several aspects. Many large and small providers are competing with each other. Their main challenge is to be ahead of the competition regarding the roll-out of new and attractive services. In order to improve service development speed and to contain development costs, mobile operators want to build as much as possible on standardized architectures and solutions. OMA is the undisputed leading mobile services standards body. The SCE developed at NEC's Network Laboratories in Heidelberg is fully compliant with the OMA model and therefore satisfies the market demand for flexible and standard compliant service creation technology.

Additionally, the provider of a service creation platform is also a natural choice for a mobile network operator as service integrator, thus the SCE opens up a large set of new business opportunity for NEC.

## REFERENCES

[1] Open Mobile Alliance TM (OMA) Standardization, http://www.openmobilealliance.org
[2] OMA-Service_Environment-V1_0-20040907-A, Open Mobile Alliance TM, http://www.openmobilealliance.org
[3] JSR-32 JAIN Sip Specification, http://www.jcp.org/aboutJava/communityprocess/final/jsr032/
[4] A. Schuelke and C. Kuecherer, "Mobile User Service Environment — A Service Mediation Platform with Intelligent Composition Mechanism," WTC, Sept. 2002.

*Received March 17, 2005*

* * * * * * * * * * * * * * *

Anett SCHÜLKE received her Ph.D in physics in 1995 from Dresden University of Technology. From 1996 to 1998 she was awarded a DAAD fellowship to continue her research at the LBNL, California/USA and worked until 1999 as a Postdoctoral Fellow. Since July 2000 she has worked as Senior Research Staff Member at NEC Network Laboratories Heidelberg. She leads the Service Creation Environment project and was recently elected as Vice Chair of the OMA PoC Working Group.

Ernö KOVACS received his Informatik Diploma from Kaiserslautern University and a Ph.D in Computer Science from Stuttgart University. Since 1997 he worked for Sony as group leader and chief engineer for mobile multimedia middleware. In 2005 he joined NEC's "Networking Laboratories" as Internet Services group manager.

Heinrich STÜTTGEN holds a Master of Science degree (1979) from the State University of New York at Buffalo (USA). In 1985 he obtained a Doctor of Science degree in computer science from Dortmund University. From 1985 to 1997 he worked at IBM Germany. In 1997 he joined NEC Europe Ltd., where he is now General Manager of NEC's Network Laboratories at Heidelberg.

Dr. Stuttgen is currently a Member of the Board of Governors of the IEEE Communications Society.

Amjad AKKAWI holds a Master in Computer Science from the Technical University of Braunschweig (2004). In April 2004 he joined the Internet Service Group at NEC Europe Ltd. as Software Engineer, and has been working on IMS service technologies and SIP communications.

Marcus KUHNEN received his Computer Science Diploma in 2003 from the Technical University Berlin and worked on UMTS NodeB development for Siemens. In April 2004 he joined the Internet Service Group of NEC Europe Ltd. and has been working on IMS technologies and web service.

Arabella RIU received her Master degree in Computer Science in 2001 from the Polytechnic University of Turin, Italy. In April 2003 she joined NL-E's Internet Services Group. She is currently Research Staff Member and working on the IMS Service Creation Environment as well as on IMS System Engineering.

Florian WINKLER received his Computer Science and Engineering Diploma from the University of Mannheim in 2003. He worked at IBM's European Networking Center on the integration of EHS (a powerline-based network protocol for home appliance control) into an OSGi compliant framework. In April 2004 he joined the Internet Service Group of NEC Europe Ltd. in Heidelberg as a Software Engineer working on Policy-Management.

* * * * * * * * * * * * * * *