A Survey on FMC Support Technologies for KT OCTAVE

Jae-pil Song, Jin-Su Sohn, *Andrew Jun, *Kong Cheng, *F. Joe Lin

IN Service Development Team, KT Marketing & Technology Laboratory

E-mail:jpsong@kt.co.kr, Tel:02-526-6129, FAX:02-526-6139

*Applied Research, Telcordia Technologies, Inc.

{ajun, kong, fjlin}@research.telcordia.com

삭제됨: *Telcordia Technologies, Inc.

서식 있음: 스웨덴어(스웨덴)

Summary

The KT OCTAVE Initiative is designed to evolve KT's existing best-effort IP network KORNET to a managed broadband network. The initiative covers both core and access networks, network and application layers, and fixed and mobile accesses with an ultimate goal of Fixed Mobile Convergence (FMC).

In order to support FMC services in the Stage 3 of the KT Octave Initiative, the current Internet needs to evolve to a managed broadband network with significant upgrade in both access and core networks. Issues central to this evolution include quality of service, security, convergence service architecture, and mobility. Out of many different aspects of network and service convergence, emphasis of this paper is on the service enabling technologies in support of unified management of contexts, profiles, sessions, and service registries.

We first provide a survey on diverse service enabling technologies from various standards bodies including IETF, 3GPP, OMA, Liberty Alliance, and W3C, for the areas of single sign-on, single context, service mobility, unified profiling, and automated service discovery. Then, a strategy and architecture for integrating those supporting technologies are presented within the context of a future-coming "Ubiquitous Personal Mobility" service environment.

삭제됨: the …M…B…N … [1]

삭제됨: QoS…S…C…S…A…M the … [2]

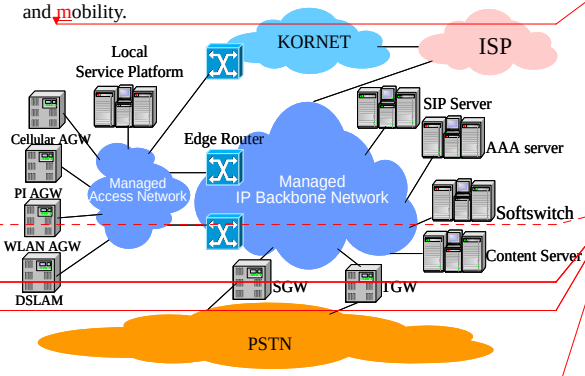
삭제됨: s…s … [3]

I. Introduction

The KT OCTAVE Initiative is designed to evolve KT's existing best-effort IP network KORNET to a managed broadband network that not only guarantees quality and security of the broadband services offered but also extends their access anywhere and anytime via emerging wireless IP technologies such as Portable Internet. The initiative covers both core and access networks, network and application layers, and fixed and mobile accesses with an ultimate goal of Fixed Mobile Convergence (FMC).

The KT OCTAVE architecture has an evolution roadmap based on a three-phase development. The Stage 1 architecture is to enhance the current best-effort network with Quality of Service (QoS) assured edge nodes and Differentiated Services (DiffServ) core routers so that Classes of Service (CoS) can be supported in the network. The goal of the Stage 1 is to offer basic services practically implementable from the current best-effort IP network KORNET. In the Stage 2 architecture, a separate QoS-enabled network based on the IP/MPLS or IP/WDM technology will be constructed. This network will be a "managed IP core" further complemented with a "managed access network". The latter provides a local service platform for the subscribers, performing functions such as policy management, authentication, content cache, billing, and service directory. The goal is to offer end-to-end QoS assured, high-quality premium services. The Stage 3 architecture, as shown in Figure 1, achieves the goal of extending OCTAVE service to include Portable Internet, which naturally leads to the need for FMC services.

In order to support FMC services, the current Internet needs to evolve to a managed broadband network with significant upgrade in both access and core networks. Issues central to this evolution include QoS, security, convergence service architecture, and mobility.



AGW: Acess Gateway TGW: Trunk Gateway

SGW: Signalling Gateway PI: Portable Internet

<Figure 1> Stage 3 Architecture

The main issue on QoS is how to ensure end-to-end QoS across core and access networks at both transport and application levels. As wireless access has unique characteristics of limited bandwidth and error prone transmission, ensuring the end-to-end QoS over wireless access networks poses the most difficult challenge. Like QoS, the main issue on security is how to provide end-to-end security guarantee across core and access networks at both network and application levels. Moreover, as the information transmission over wireless access networks is easily

삭제됨: the …M…B…N … [4]

삭제됨: S…C…S…A…M … [5]

삭제됨: A…-E…N…C…R…A…A…coverage ….These services are thus called (Fixed Mobile Convergence) … [6]

삭제됨: both …to support …IP -level session QoS…IP …-…session QoS…S…also on …both and on how to support security mechanisms … [7]

subject to interception, security guarantee over wireless access networks is the most challenging issue. The main issue on service architecture is how effectively service applications can be delivered to the end customers. To successfully integrate services over the convergent network, an integrated service architecture, either loosely coupled or tightly coupled with centralized or distributed integration logic, needs to be in place to support both convergence service bundling and accounting. Moreover, the role of application session management in the integration of convergence services needs to be defined. The most important job is to integrate multiple control points and accounting management in the convergent networks.

Under the OCTAVE architecture, users should be able to access the OCTAVE services anytime and anywhere through any devices (fixed, mobile, or fixed-mobile convergence devices). To ensure such mobility, the OCTAVE architecture needs to support features such as single sign-on, single identity, and seamless session management for service mobility in fixed mobile convergent networks.

Out of many different aspects of network and service convergence, the emphasis of this paper is on the service enabling technologies in support of unified management of contexts, profiles, sessions, and service registries. In Chapter II, we first provide a survey on diverse service enabling technologies from various standards bodies including IETF, 3GPP, OMA, Liberty Alliance, and W3C, for the areas of single sign-on, single contexts, service mobility, unified profiling, and automated service discovery. Then, in chapter III, a strategy and architecture for integrating those supporting technologies are presented within the context of a future-coming "Ubiquitous Personal Mobility" service environment.

II. Support Technologies for FMC

This Chapter focuses on five support technologies for FMC: (1) single sign-on, (2) service management platform for single context support, (3) session management for service mobility, (4) user policy management via user agent, and (5) automatic discovery of mobile applications.

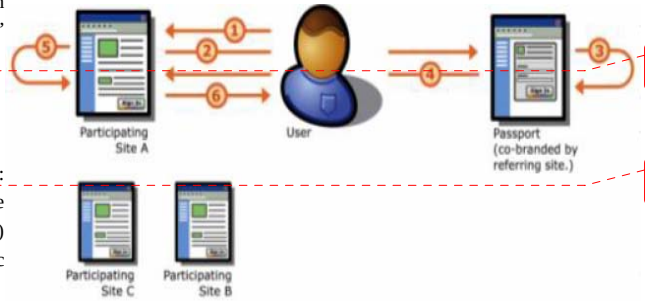
Under single sign-on (SSO), we will give a survey of SSO technologies in industry including the work from Liberty Alliance Project, GAA (Generic Authentication Architecture) from 3GPP, and MSN Passport including a detailed analysis of authentication mechanisms for each method. Under service management platform for single context support, we will discuss what service management platforms are needed to support single context that may potentially include the notions of a single number, a single bill, a single presence, and a single user profile. Under session management for service mobility, we will discuss how to support service mobility across PSTN, VoIP, and mobile networks via session management that includes careful planning of triggering mechanisms in call delivery services. Under user policy management via user agent, technologies for user policy management using user agent will be discussed including intelligent user agent techniques to support single context (e.g., user profile, presence, location, etc.) management. Under automatic discovery of mobile applications, smart, automated mechanisms for service discovery in a mobile environment will be discussed. Such a service discovery should be based on the factors of user policy and service mobility, and should support single sign-on whenever possible.

1. Single Sign On

1.1 Microsoft Passport

Microsoft .NET Passport (http://www.passport.net/) is an online authentication service that makes using the Internet and purchasing online easier and faster. Using Passport accounts, users can seamlessly access all participating Passport Web sites and services without the need to remember or type a specific set of credentials (e.g., identification and password) for each of them. Authentication utilizing Passport is transparent to the end user. Passport also provides an electronic wallet service by storing credit card and shipping information in the central Passport Server repository.

In support of SSI (Single Sign In), Passport supplies a registered user (more precisely user's web browser) a "Ticket-Granting-Cookie" (TGC). This cookie can be used to access other Passport participating sites. As long as the TGC remains valid, the user does not need to re-authenticate in subsequent protocol runs. For confidentiality and integrity, cookies are encrypted in Passport: While cookies with credentials are encrypted with the Passport key, cookies with profile information are encrypted with the participating sites' keys. Figure 2 shows authentication procedure of Passport.



<Figure 2> Passport authentication procedure (excerpt from [1])

Sites can become participating Passport sites by implementing the Passport SSI (Single Sign-In) service. Passport identifications, passwords and associated profile information of users are all stored in a central data repository. Thus, Passport participating businesses do not need to manage user credentials on their own. However, since Passport does not authorize or deny a specific user's access to individual participating sites, Passport participating sites should maintain control over permissions.

Passport uses encryption technologies, such as Secure Sockets Layer (SSL) and the Triple Data Encryption Standard (3DES) algorithm, for data protection, and enforces a privacy policy to participating sites for user privacy protection. Current deployed version of Passport is based on centralized authentication architecture and has revealed security weaknesses

삭제됨: may
삭제됨: F
삭제됨: M
삭제됨: F
삭제됨: M
삭제됨: C
삭제됨: D
삭제됨: ID
삭제됨: s
삭제됨: will
삭제됨: M
삭제됨: N

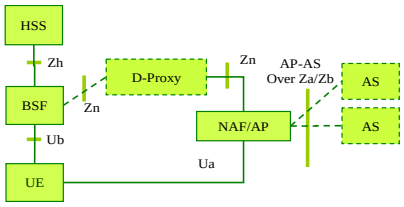
[2], some of which are Cookie problem, Key management, Passport server attack and Hotmail credential assignment.

As a result of the above security weaknesses, Passport seems to be vulnerable for security attacks such as fake merchant attack, active attack, DNS attack, and cookie attack. It is reported that Microsoft is currently upgrading the Passport facilities to provide an XML interface and federation support with Kerberos v5 as the underlying mechanism for securely exchanging credentials.

1.2 3GPP Generic Bootstrapping Architecture

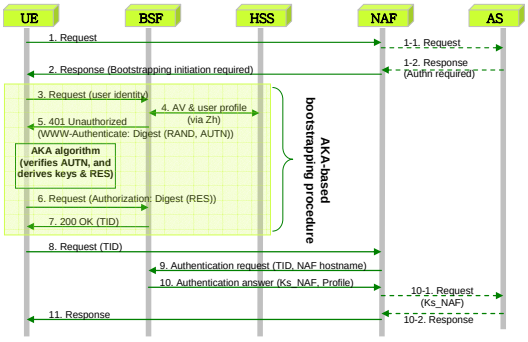
3GPP (3rd Generation Partnership Project: https://www.3gpp.org) defines the concept of GBA (Generic Bootstrapping Architecture) [3] under GAA (Generic Authentication Architecture) [4] for client access to network applications and services. The motivation for GBA comes from the fact that there are many applications and services within 3G mobile systems requiring mutual authentication between a client and an application server. They may also require security associations, either dynamic or static, for secure conversation.

GBA is based on the existing 3GPP network security infrastructures [5], where all users are equipped with security credentials on a smartcard device (USIM) [6], and the corresponding keys are stored in the HSS (Home Subscriber Server) [7] of the home network.



<Figure 3> Generic bootstrapping architecture

GBA is displayed in Figure 3. Essentially, GBA is to provide authentication and key agreement services to the UE (User Equipment) and the NAF (Network Application Function) in support of application-level interactions between them. The NAF is a generic application server that provides applications or services to a mobile device (or the UE). No assumption is made about what protocol should be used between the UE and the NAF although HTTP is a strong candidate.



<Figure 4> Generic bootstrap procedure

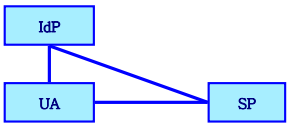
User authentication is performed by the BSF (Bootstrapping Server Function) by means of the HTTP-Digest AKA protocol [9]. The authentication vectors required to run the AKA protocol are fetched from the HSS, which stores information related to the use of the bootstrapping function, such as subscriber profile. The D-Proxy (Diameter Proxy) may convey the communication between the BSF and the NAF in case that the BSF and the NAF belong to different mobile operator domains. Non-GBA applications can be also supported via the AP (Authentication Proxy). In this case, AP plays the role of "reverse proxy" to represent the associated ASs. Precise definition of the AP-AS interface is out of scope of GBA at the moment. Figure 4 shows a typical generic bootstrap procedure

1.3 Liberty Alliance Project: Identity Federation

Liberty Alliance (http://www.projectliberty.org/), a business alliance formed in September 2001, consists of service providers, technology vendors as well as policy and government organizations. The goals of Liberty Alliance are to establish open standards and guidelines for federated identity management spanning various network devices, decentralized authentication and open authorization, and secure management of personal information (user profile).

Identity federation framework [10] of Liberty Alliance is built upon the following protocols and technologies:

- The Security Assertion Markup Language (SAML) standard [11] - specifies an extensible language for securely exchanging user information between security domains.
- The Web services security(WS-Security) specification [12] - defines mechanisms for providing security token based integrity and confidentiality on SOAP messages.
- The Identity Federation Framework (ID-FF) specifications [13] - extend SAML by adding its own profiles for SAML assertions.
- The Identity Web Services Framework (ID-WSF) specifications [14] - are a set of standards that add functionality for discovering and offering identity-related services via the Web services technologies including SOAP.
- The Identity Service Interface Specifications (ID-SIS) [13][15] - provide networked profile services, such as contacts, presence detection, or wallet services, based on the networked identity services provided by ID-FF and ID-WSF.

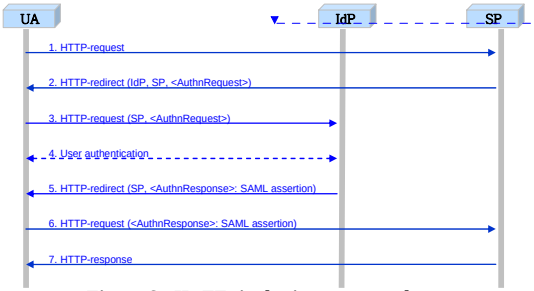


<Figure 5> ID-FF model

High-level ID-FF model is depicted in Figure 5, where there are three distinct roles within a federation: User Agent (UA), Identity Provider (IdP) and Service Provider (SP). For single sign-on and identity federation procedures, ID-FF defines three different mappings/profiles of message types and flow patterns: Liberty artifact profile, Liberty browser POST profile, and Liberty-enabled client and proxy profile. Although details are different from one profile to another, all three profiles share the same abstract-level principles. Figure 6 shows the ID-FF single

삭제됨: 3

sign-on procedure based on the Liberty artifact profile.



<Figure 6> ID-FF single sign-on procedure (HTTP-redirect based)

2. Service Management Platform for Single Context Support

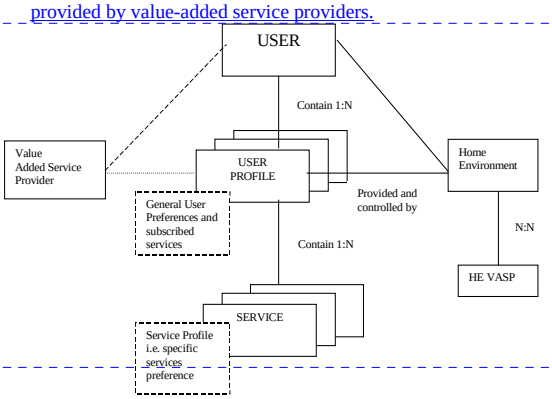
2.1 3GPP Virtual Home Environment

We looks at the Virtual Home Environment (VHE) for single context support (single profile, single number and single bill). The Virtual Home Environment (VHE) is a concept for Personal Service Environment (PSE) portability across network boundaries and between terminals. Through personalized services, users are presented with customized features in accordance with their preferences/profiles, and through service portability, users are consistently presented with the same personalized services in whatever network or terminal.

The VHE concept appeared in the 3G wireless networks as a result of emerging need for portability of personalized service environments in the midst of the increasing capabilities of terminals and networks. The main interest of VHE can be summarized as follows (excerpt from [16]):

- Personalization of Service Environment – PSE describes how the users choose to personalize, modify, access and behave with the services that they have subscribed.
- Adaptation of Service Environment - the customized environment follows the user while he/she is roaming within different networks and using different terminals. Service adaptations should comply with user preferences.
- Service Portability - a user is offered the same service experience in a visited network as in his home network. Users can choose among different profiles, change the profile, activate and deactivate profile services, and restate the user profile in case of lost or malfunction of terminal regardless of the location and terminal in use. The logical VHE role model from the user's point of view is shown in Figure 7.

Individual user's information on how personalized services should be provided and presented is contained in a set of user profiles. Each user's profile consists of user interface related information and subscribed services related information. Because users can have multiple roles for different situations or needs (for instance, being at work or at home), a single user can have multiple profiles. User's profiles are provided and controlled by the home environment (HE). Home environment value-added service providers (HE-VASPs) maintain service agreements with the HE to make personalized services available to users. In addition to home services, a user can utilize other services provided by value-added service providers.



<Figure 7> Logical VHE Role Model (User's View) (excerpt from 17])

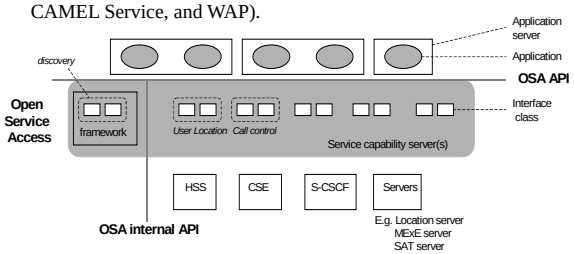
User profile data can be located in one or more entities of HSS (Home Subscriber Server), (U)SIM, ME (mobile equipment), and application-specific databases of HE and HE-VASP. Access to user profiles should be supported in a secure and standardized way.

Functional components of VHE can be realized using standardized service capabilities, including (but not limited to) CAMEL [17], MExE [18][21], USAT, OSA, and GUP. It should be noted that non-3GPP technologies, such as W3C CC/PP, Liberty Identity Federation can complement the aforementioned 3GPP-defined open service technologies in realizing VHE capabilities.

2.2 3GPP Open Service Access[18][19]

The work on the Open Services Access (OSA) was initiated in 1998 as part of VHE, and split from VHE in 2000 as a separate work item. OSA specification is mainly based on the Parlay specification with UMTS specific enhancements. OSA is aligned with the ETSI SPAN and the PARLAY group. The set of OSA specifications includes interface definitions for the following Network Services: Data Session Control, User Interaction, User Status, Terminal Capabilities

OSA provides Application Programming interface (API) for network service capability features (SCFs) to applications providers, allowing the ability to create solutions for network services (such as location, billing, and call control) as well as giving mediated access to service capabilities (such as HLR, CAMEL Service, and WAP).



<Figure 8> Overview of Open Services Access (excerpt from [19])

Figure 8 gives a global overview of OSA. The Framework

삭제됨:

삭제됨: High-level ID-FF model is depicted in Figure 5, where there are three distinct roles within a federation: User Agent (UA), Identity Provider (IdP) and Service Provider (SP). For single sign-on and identity federation procedures, ID-FF defines three different mappings/profiles of message types and flow patterns: Liberty artifact profile, Liberty browser POST profile, and Liberty-enabled client and proxy profile. Although details are different from one profile to another, all three profiles share the same abstract-level principles. Figure 6 shows the ID-FF single sign-on procedure based on the Liberty artifact profile.

… 8

삭제됨: Individual user's information on how personalized services should be provided and presented is contained in a set of user profiles. Each user's profile consists of user interface related information and subscribed services related information. Because users can have multiple roles for different situations or needs (for instance, being at work or at home), a single user can have multiple profiles. User's profiles are provided and controlled by the home environment (HE). Home environment value-added service providers (HE-VASPs) maintain service agreements with the HE to

… 9

provides the functionality required to use the OSA services and the mechanisms to offer these to third parties. The OSA/Parlay APIs are grouped into a set of related functional areas. The current specification details 15 main areas[68]-[82]: (1) Overview, (2) Common data, (3) Framework, (4) Call control, (5) Generic user interaction, (6) Mobility, (7) Terminal capabilities, (8) Data session control, (9) Generic messaging, (10) Connectivity manager, (11) Account management, (12) Charging, (13) Policy management, (14) Presence and availability management (PAM), (15) Multimedia messaging (MM).

Two of the key issues of providing third parties access to core network functionality are security and integrity management of the network. Only authenticated applications should be allowed access to the OSA services. Also the usage of the OSA services should be controlled, as intended or unintended misusage of the services can seriously disturb the integrity of the network. The OSA framework addresses these two issues as follows.

All the security aspects of OSA are the responsibility of the framework. The OSA Authentication interface supports the usage of multiple encryption methods. The authentication process is based on a challenge/response protocol. The challenge mechanism used is in accordance with RFC 1994. Before a client application gets access to a specific OSA service, the application and the framework must also (digitally) sign a service agreement. The exact text of these service agreements should be agreed upon off-line by the parties involved.

Integrity should deal with the performance and availability aspects of the OSA. The integrity management capabilities of OSA consist of the following interfaces:

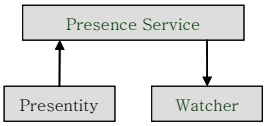
- Heartbeat interfaces, to have the client application and the framework exchange heartbeat messages.
- Load management interfaces, to get load information, both on the client application and the framework.
- Fault management interfaces, to monitor faults that occur at either the framework or the services.

2.3 Presence Services Frameworks

We looks at the Presence Architecture defined in 3GPP for single presence of single context support. The concept of presence has been used in communications applications such as instant messaging and multi-party conferencing. Presence information changes over time, and therefore is differentiated from more static information, such as policies and profile, associated with identities and agents.

The Internet Engineering Task Force (IETF) has proposed a number of specifications for sharing and communicating presence information. The important efforts include: Instant Messaging and Presence Protocol (IMPP), SIP for Instant Messaging and Presence Leveraging Extensions (SIMPLE), and Extensible Messaging and Presence Protocol (XMPP).

In RFC 2778, the presence service has two distinct set of clients: presentity and watcher (see Figure 9). A presentity provides presence information to the presence service for storage and/or distribution. The presence service provides presence information of presentities to interested watchers via either fetching or subscribing mode. While fetchers (i.e., watchers in the fetching mode) simply request the current presence information from the presence service, subscribers (i.e., watchers in the subscribing mode), on the other hand, request notification from the presence service of future change in presence information.



<Figure 9> Overview of presence service

The IETF is developing two protocols, SIMPLE and XMPP, conforming to IMPP specifications (i.e., RFC 2778, RFC 2779, and RFC 3860). In SIMPLE [52]-[55], Session Initiation Protocol (SIP) [48] is used for the distribution of presence information and for establishing IM sessions. All SIP extensions proposed by the SIMPLE working group (http://www.ietf.org/html.charters/simple-charter.html) will be transferred to the SIP working group for consideration as formal SIP extensions. The some SIP extensions have been proposed by the SIMPLE working group (update of presence information can be done through SIP REGISTER method or through some non-SIP means) such as SUBSCRIBE Method, NOTIFY Method, MESSAGE Method.

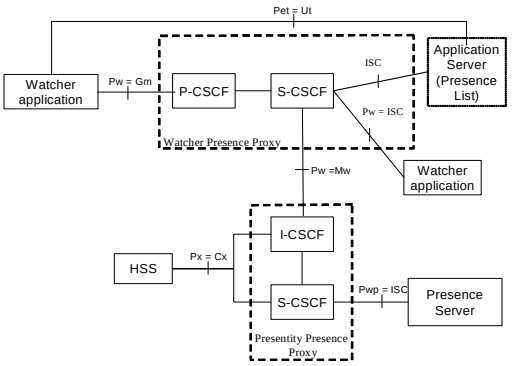
The 3GPP has adopted SIMPLE as the baseline protocol for the definition of 3G presence services. Also, several major companies have chosen SIMPLE for implementing messaging and presence applications. Currently, the SIMPLE working group is putting effort on session mode operation, which uses SIP to establish an IM/presence session and SDP (Session Description Protocol) [56] to actually pass the IM data.

XMPP [57]-[60] is another IMPP-conforming protocol for delivering instant messages and presence information. XMPP is an XML-based client-server protocol running over TCP sockets. XMPP also includes a protocol for server-to-server communication, which, together with XML's extensibility, provides a flexible environment for customizing applications' functionality. Furthermore, security features (such as authentication, privacy, and access control), as mandated in the RFC2779, are incorporated into XMPP. In particular, user authentication and data encryption issues are addressed through the use of Simple Authentication and Security Layer (SASL), Transport Layer Security (TLS). The adoption of XMPP has been advocated by the Jabber open-source community.

In addition to the presence and availability management functionality through OSA, 3GPP has defined presence services in specifications "Presence service; Stage 1" (TS 22.141) [61], "Presence Service; Architecture and functional description" (TS 23.141) [62], and "Presence service using the IP Multimedia (IM) Core Network (CN) subsystem; Stage 3" (TS 24.141) [63]. 3GPP presence service is based on the IETF presence service model as defined in RFC 2778 [49]. 3GPP specifications provides wireless-specific enhancements to IETF presence service through the support of wireless attributes, yet in an interoperable manner

to support both wireless networks and fixed networks.

For alignment with the Internet community, 3GPP makes extensive use of the IETF presence service terminologies. As such, 3GPP presence service entities are also termed as presentities (providers of presence information) and watchers (consumers of presence information). The presence server, which resides in the presentity's home network, manages and distributes the information to interested parties, called watchers. The two sets of entities involved, presentities and watchers, are either internal or external to the home network and access network. Watchers access the server through presentity proxies. Figure 10 is an IMS-specific version of the 3GPP presence service architecture, presenting the mapping of the presence service components to the IMS (IP Multimedia Subsystem) network elements when located within the IMS along with the watcher application.



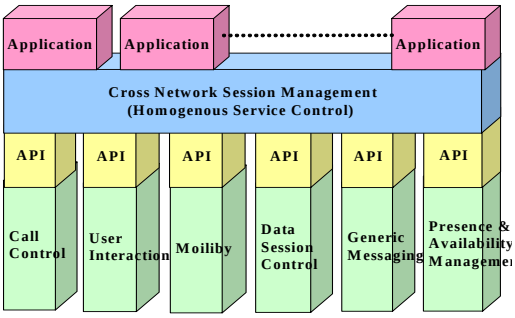
<Figure 10> IMS-specific version of the 3GPP presence (excerpt from [62])

3. Session Management for Service Mobility

3.1 Cross Network Application Session Management

To support session management across PSTN, VoIP, and Mobile Networks for service mobility, the key will be an API-based session model with careful planning of triggering mechanisms in call delivery services. Traditionally, services are provided vertically in independent domains such as PSTN, IP and mobile networks. In a fixed-mobile converged environment, it is highly desirable that the same service can be made available across different heterogeneous networks to avoid replication in service development and management. An example is the use of 3GPP IMS as the core infrastructure to support of both wireless and wireline services through an API-based service architecture [68]-[82].

In an API-based service architecture, service programming and hence control is provided through a collection of network independent APIs. As depicted in Figure 11, the APIs are typically partitioned into different classes based on their functionality such as those specified by Parlay, 3GPP OSA and ETSI SPAN 12 [78]-[82]. From the session management viewpoint, an API-based service architecture facilitates integration and control across different application sessions without the need to deal with lower layer network and protocols.



<Figure 11> API-based Service Architecture

삭제됨: In an API-based service architecture, service programming and hence control is provided through a collection of network independent APIs. As depicted in Figure 11, the APIs are typically partitioned into different classes based on their functionality such as those specified by Parlay, 3GPP OSA and ETSI SPAN 12 [78]-[82]. From the session management viewpoint, an API-based service architecture facilitates integration and control across different application sessions without the need to deal with lower layer network and protocols.

3.2 Call Control Models

Call Control has one of the most complex API realization as it cuts across different network domain and signaling. At the early stages, the approach is to provide a point-to-point signaling interworking across IP and PSTN networks such as ISUP and MGCP, SIP and ISUP. Essentially there are three main approaches to using the call model:

- IN Based Call Model
- Language Based Call Model.
- Model Based Call Model.

3.2.1 IN Based Call Model

In this approach, the IN call model is used as a common call model for all IN and IP based services. Most of the earlier mappings are based on IN call model especially for the first generation softswitches. The attractiveness of this approach is that most existing vendors who have considerable investment on IN can continue to evolve their implementation/product to support IP based services. As the IN call model is very comprehensive, it may not be suitable for providing simple IP based services.

For illustration, Table 1 and Table 2 below depict examples of mapping from various signaling protocols (SIP, H.323 and ISUP) into the originating and terminating IN Call Model, respectively. In this case, we use the ITU-T Q.1204 BCSM for the mapping, and only the Detection Points are listed as the BCSM state. For simplification, error cases are not considered in the mapping.

Table 1. Mapping of ISUP, H.323 and SIP onto the IN Call Model O_BCSM.

SIP	H.323	ISUP	O_BCSM DP (States)
INVITE –>	SETUP –>	IAM –>	1.Orig_Attempt
			3. Orig_Attempt_Auth
			5. Collected_Info
100 Trying <–	Call Proceeding <–		7. Analyzed_Info
			9. Route_Selected
			11. Orig_Auth
180 Ringing <–	Alerting <–	ACM/ CPG <–	14. O_Term_Seized
200 OK <–	Connect <–	ANM <–	16. O_Answer
BYE –>	Release Complete –>	REL –>	19. O_Disconnect

Table 2. Mapping of ISUP, H.323 and SIP onto the IN Call Model T_BCSM.

SIP	H.323	ISUP	T_BCSM DP (States)
			22. Term_Attempt
INVITE <-	SETUP <-	IAM <-	24. Term_Attempt_Auth
			26. Term_Res_Avail
180 Ringing ->	Call Proceeding -> Alerting ->	ACM ->	28. T_Term_Seized
200 OK ->	Connect ->	ANM ->	30. T_Answer
BYE	Release Complete <-	RLC <-	33. T_Disconnect

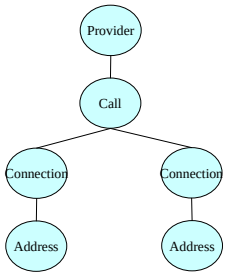
3.2.2 Language Based Call Model

With the extensive use of object-oriented programming languages such as Java, a new class of call model emerges. Instead of a single "object" representing a call model like in IN, the object oriented concepts are used extensively to create a call model that is decomposed into multiple interacting objects that include entities such as terminals that are not explicitly represented in traditional IN Call Model. In addition, the object oriented model also allow incremental addition of call model details and hence making evolution easier to manage than the traditional non object oriented call model approach. A notable example is the JAIN JCC (Java Call Control)[106][107].

The JCC API is structured into two functional areas : (1) Elementary Call Control - Java Call Processing (JCP) package includes the very basic facilities required for initiating and answering calls, (2) Core Call Control - Java Call Control (JCC) package that inherits from JCP includes the facilities required for observing, initiating, answering, processing and manipulating calls, as well as to invoke applications and return results during call processing.

For both JCC and JCP, the API components consist of a related set of interfaces, classes, operations, events, capabilities, and exceptions. The API provides four key objects, which are common to JCP, JCC and more advanced packages. The four key objects are:

- Provider: represents the "window" through which an application views the call processing.
- Call: represents a call and is a dynamic "collection of physical and logical entities" that brings two or more endpoints together.
- Address: represents a logical endpoint (e.g., directory number or IP address).
- Connection: represents the dynamic relationship between a Call and an Address.



<Figure 12> Object model of a two-party call

The relationship among these objects is depicted pictorially in Figure 12 for a two-party call. Multiple parties are represented, with no inherent limitations in the model, simply by additional Connections and Addresses associated with a Call. In traditional telephony parlance, this model is "symmetric" in that there is no fundamental distinction at the highest level between originating or terminating parties of a call, and is a "full" model in the sense that the application, in principle, has a view of all parties of the call.

JAIN [106][107] facilitates mapping from its JCC call model to the underlying signalling protocols through a series of lower level Java based APIs for each of the signalling protocols. Some of the protocols APIs include: JAIN INAP, JAIN MGCP, JAIN TCAP, JAIN MEGACO, JAIN SIP

3.2.3 Model Based Call Model

In the model based approach, a call model is defined in a programming language independent manner using high level models such as UML. A notable effort in this area is the Parlay group. Parlay is the first to address open API to public networks and has since maintained its leadership role and influence in the API arena. The Parlay Group is the forerunner in this area with the first demonstrable implementation. The Parlay's industry leadership position is further strengthened when it was adopted by 3GPP as it OSA API.

The Parlay Call Control working group is responsible for defining Parlay Call Control APIs [71] to enable services for the real-time multimedia (audio, video, and/or data) applications independent of platforms and networks. There are four different call control APIs supported by Parlay API.

- The Generic Call Control (GCC) API was traditionally developed for two party calls.
- The Multi-party Call control (MPCC) API was developed to manipulate two or more legs to a call to enable advanced services.
- The Multi-media Call Control (MMCC) API enhances the MPCC functionality with multi-media capabilities.
- The Conference Call Control (CCC) API, as implied by the name is to provide conferencing control. It is used in conjunction with MPCC and MMPC interfaces

The joint work between 3GPP CN5, ETSI TISPAN and the Parlay Call Control Working group with collaboration from JAIN has decided that the MPCC is to be considered as the future base Call Control family and the technical work will not be continued on GCC. Hence our discussion here is based on the Parlay MPCC model.

The call model used for the Parlay Call Control has the following objects:

- Call object. A call is a relation between a number of parties. The call object relates to the entire call view from the application. Note that different applications can have different views on the same physical call. The API currently does not specify any feature interaction mechanisms.
- Call leg object. The leg object represents a logical association between a call and an address. The relationship includes at least the signaling relation with the party. The

relation with the address is only made when the leg is routed.

- Address. The address logically represents a party in the call.
- Terminal. A terminal is the end-point of the signaling and/or media for a party. This object type is currently not addressed.

The call object is used to establish a relation between a number of parties by creating a leg for each party within the call. Associated with the signaling relationship represented by the call leg, there may also be a bearer connection (e.g., in the traditional voice only networks) or a number (zero or more) of media channels (in multi-media networks). A leg can be attached to the call or detached from the call. When the leg is attached, this means that media or bearer channels related to the legs are connected to the media or bearer channels of the other legs that are attached to the same call; that is, only legs that are attached can 'speak' to each other. A leg can have a number of states, depending on the signaling received from or sent to the party associated with the leg.

Some networks distinguish between controlling and passive legs. By definition the call will be released when the controlling leg is released. All other legs are called passive legs. There can be at most one controlling leg per call. However, there is currently no way the application can influence whether a Leg is controlling or not.

3.2.4 Cross Network Call Model Approach and Evolution

Current industry trend indicates that the most promising and dominant approach is the Parlay Call Model. JAIN, while has contributed significantly to the realization of Call Model implementation has not been successful in attracting vendors in developing commercial products. On other hand, Parlay has quite a number of vendors providing commercial products. There are essentially two classes of vendors providing Parlay OSA based product. One is from the traditional IN vendors where the IN based products are enhanced to support Parlay Call Model. The by product is that interworking with PSTN is implicit. The other class of vendors are “green field vendors” that has the opportunity to create Parlay product from scratch, providing implementations that directly reflect the Parlay call model. However, additional effort is needed to provide interworking with the IN/PSTN domain such as through existing IN products. In either case, the products typically support service across traditional IN/PSTN and IP services.

Parlay OSA should be the focus as it is the current industry leader in standardization and future evolution of the call control model. However, the implementation towards Parlay can be driven from existing products such as softswitch implementations that may be based on IN Call Model. In practice, the IN based call model implementation and the pure Parlay call implementation will coexist, as long as both provide the same set of Parlay APIs from the application viewpoint.

4. User Policy/Profile Management

4.1 User Agent Platforms

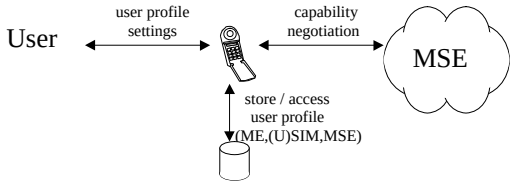
This subsection discusses technologies for user policy management using user agent. A user agent here is defined to be an application on the user terminal. A user agent platform is thus the operating environment on the user terminal. Two such environments are discussed below: MExE (Mobile Execution Environment) and SAT/USAT (SIM/USIM Application Toolkit) from 3GPP.

4.1.1 3GPP Mobile Execution Environment

MExE (Mobile Execution Environment) [18][21] is a 3GPP standard providing a standardized approach for the delivery of services, applications, and contents to smart mobile phones. MEXE’s aim is to provide a standardized environment on mobile phones for executing operator or service provider specific applications. MExE specifies a set of requirements on a standard execution environment for applications in Mobile Stations (MSs) and requirements of an application in MSs to interact with service providers. Through MExE, MSs can negotiate supported capabilities with the MExE service provider, which allows applications to be developed independently of any MS platform.

In MExE specifications, terminals are categorized by different MExE classmarks. Classmark 1 corresponds to terminals with WAP capabilities; classmarks 2 to terminals with Personal Java environment; classmark 3 to terminals with J2ME environment; and classmark 4 to terminals with CLI (Common Language Infrastructure) environment including web services capabilities.

Depicted in Figure 13 is the MExE model of user profile and capability relationship. MExE specification defines user profile as a set of interface preferences and service management preferences, governing the MExE device behavior. The user profile is accessed by a set of defined APIs based on the Composite Capability/Preference Profile (CC/PP) specification from W3C. MExE extends CC/PP by allowing not only the “direct referencing mechanism” to store the data in the MS, as supported by CC/PP, but also the “indirect referencing mechanism” to store the data in the network. The user profile information is delivered to the MExE service environment through the MExE capability negotiation process. For user preferences and device capabilities related to UI features, UAProf is also supported in MExE.



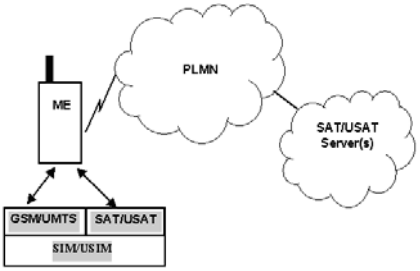
<Figure 13> Model of user profile and capability relationship (excerpt from [21]).

4.1.2 3GPP SIM/USIM Application Toolkit

Just like MExE, SAT (SIM Application Toolkit) supports VHE. Instead of providing execution environments in the terminal, SAT or USAT (USIM Application Toolkit) provides execution environments in SIM (Subscriber Identity Module) or USIM (Universal Subscriber Identity Module). The SAT/USAT specifies the interface between SIM/USIM and Mobile Equipment (ME). A high-level SAT/USAT architecture is

depicted in Figure 14.

SIM was defined as a security module for authentication of the user in GSM phase 1. The communication could only go from ME to SIM, but not the other way [83]. In GSM Phase 2, a special facility of running user applications in SIM was standardized as SAT [84]. SAT is a technology allowing SIM card to issue commands to ME, such as displaying menus, getting user input, and sending/receiving SMS messages. SAT allows remote loading of applications to SIM cards. Since only the network operator has access to this capability, no third party applications can be provided to GSM terminals through SAT.



<Figure 14> Scope of SAT/USAT technical specification (excerpt from [85]).

In 3GPP, Universal SAT (USAT) [85]-[89] has been defined as an extension to GSM SAT. USAT provides a standardized execution environment for applications in SIM/USIM card. USAT allows the flexibility to update SIM/USIM via download of new services over the air. USAT provides mechanisms which allow applications in USIM/SIM to interact and operate with any ME. One added feature of USAT is a bytecode interpreter [88] that allows SIM cards to download, display, and interact with the subscriber. Also, USAT facilitates push-style as well as web-based communications in SIM/USIM, such that applications can be downloaded and run in SIM/USIM and then be removed after execution. The USAT server is an entity that is typically in the home environment. As for the MExE server, the functionality of a USAT server could be realized in many different ways.

A significant aspect of USAT is the highly secure environment provided by SIM/USIM card. This facilitates end-to-end security between a service provider and the SIM/USIM, which enables certain features, such as call control or charging, to be implemented with a degree of freedom that would not be acceptable in a “non-trusted” environment. USAT is a very useful tool, especially for electronic commerce, to access all the security/identity information about the end user contained in SIM/USIM.

4.2 Policy/Profile Management Frameworks

Three policy/profile management frameworks are surveyed in this subsection including Composite Capabilities/Preferences Profiles from W3C, User Agent Profile from OMA/WAPF, and Generic User Profile from 3GPP.

4.2.1 W3C Composite Capability/Preference Profiles

The W3C CC/PP working group (http://www.w3.org/Mobile/CCPP/) has specified a structure for profiles, called Composite Capabilities/Preferences Profiles (CC/PP) [91]. The goal of the CC/PP framework is to define vocabularies for describing device capabilities and user preferences for devices. The CC/PP profile is used for adapting content for end-user devices. The CC/PP profile can be stored in a device, and can be transmitted to the content provider for content customization purpose.

CC/PP is based on the Resource Description Framework (RDF), a general purpose metadata description language. This profile structure is vocabulary independent and extensible via XML namespaces and interoperability. The vocabulary neutral nature of CC/PP allows additional vocabularies to be developed and implemented by application developers, device manufacturers and browser developers. Also, as an extension to the base XML format, RDF allows the concept of metadata (data about data) to be added to an XML document: the author of the document, for instance.

A CC/PP profile is broadly constructed as a 2-level hierarchy (see Figure 15):

- A profile having at least one or more components;
- Each component having at least one or more attributes.

```
[example:MyProfile]
 |
 +--ccpp:component-->[example:TerminalHardware]
 +--ccpp:component-->[example:TerminalSoftware]
 +--ccpp:component-->[example:TerminalBrowser]
```

<Figure 15> CC/PP profile components (excerpt from [91]).

The CC/PP framework allows for the dynamic composition of a profile from fragments of information distributed among multiple repositories. Also supported is capabilities asserted by intermediate proxies/gateways. It should be noted that the CC/PP framework defines profile vocabularies, but neither the protocol for exchanging profiles nor the way the profile should be used for content adaptation. Implementation of CC/PP is guided by two associated recommendations: “Privacy and Protocols” [92] (using P3P [94]), and “Harmonization with Existing Vocabularies and Content Transformation Heuristics” [93].

4.2.2 OMA/WAPF User Agent Profile

The WAP Forum (http://www.wapforum.org/), which now has been consolidated to OMA (http://www.openmobilealliance.org/), defined the User Agent Profile (UAProf) [95] as an implementation of CC/PP for WAP (Wireless Application Protocol) [96] enabled mobile terminals. The goal is to capture classes of device capabilities and preference information into RDP-based vocabulary, and provide end-to-end flow of the profile information through the device, the intermediate network points, and the origin.

The profile classes include hardware characteristics , software characteristics , and user preferences for devices . Also included in the profile vocabulary are assertions regarding the hardware and software characteristics and WAP specific features of the mobile terminal and associated transport network capabilities. The capability and preference profile information is

collected on the client device, optionally combined with other information available over the network, and made available to the origin server to ensure that the user receives content tailored for the environment. It should be noted that a "user agent profile" is distinct from a "user preference profile." A user preference profile would contain application-specific information about the user for content selection purposes (i.e., user's interest for particular content), which is beyond the scope of the UAProf specification.

Over the Internet, the UAProf specification assumes the use of CC/PP, HTTP, and optionally the CC/PP Exchange Protocol over HTTP (CCPPex) [97] together with the HTTP Extension Framework (HTTPext) [98] or the WAP wireless profiled HTTP (W-HTTP) [99]. WAP 1.2.1 [96] clients can connect to an origin server using the CCPPex. Conformant origin servers are expected to honor the HTTP 1.1 protocol with the HTTPext. However, non-conformant HTTP 1.0 and HTTP 1.1 origin servers could process HTTP requests containing CC/PP profiles by specifically parsing CCPPex headers. Due to the lack of implementations of HTTPex, however, WAP 2.0 introduced W-HTTP with custom headers to substitute HTTPex in support of UAProf. W-HTTP clients can connect to an origin server directly, or using a feature or performance enhancing proxy. UAProf – CC/PP framework has been adopted to 3GPP MExE.

4.2.3 3GPP Generic User Profile

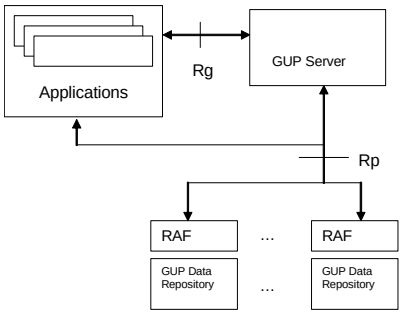
Generic User Profile (GUP) [43][44] is a 3GPP solution to user profiling. 3GPP introduces the GUP concept as a solution to the problem of defining a flexible and extendible user profile, which all mobile network applications can refer to as a single point of access. The objective of GUP is to provide a conceptual description to enable harmonized usage of the user-related information located in different entities. The GUP and associated concepts are applicable to the VHE but are not limited to it.

The information that is available on this concept is very preliminary, and is not complete. Here are 3GPP standards documents for GUP (Generic User Profile): (1) Requirements in TS 22.240 (3GPP SA1), (2) Architecture specification in TS 23.240 (3GPP SA2), (3) Data Description Framework in TS 23.241 (3GPP T2), (4) Common Objects in TS 24.241 (3GPP T2). (5) 3GPP TS 24.241, 3GPP generic user profile requirements; Stage 3; Access; Common objects.

GUP is the collection of user-related data which affects the way a community of entities/applications shares this data, and the way an individual user experiences services. GUP can be stored in the home network environment and additionally storage can be extended to the UE and/or Value Added Service Provider (VASP) equipment. GUP will be accessed and managed by different stakeholders such as the user, subscriber, etc.

GUP proposes a structure according to which data have to be organized, but leave great flexibility on the content of the data themselves. For example, GUP may store data like authorized and subscribed services, general user information, user privacy control data, information about specific services and billing information whereas historical/statistical and Runtime data are not included in GUP. The most important aspect of GUP is that it could be adapted to every system and context, providing the flexibility to network applications.

The GUP reference architecture is illustrated in Figure 16. For each user one User Profile exists, which may consist of several 'components'. These components may be distributed in the home network, the User's Equipment and value added service provider's environment. Within the home network, the components may be distributed in various network nodes. The GUP reference architecture consists of GUP Server, Repository Access Function (RAF), GUP Data Repositories, Rg and Rp reference points, and Applications.



<Figure 16> GUP reference architecture (excerpt from [44]).

GUP allows intra-network usage (i.e. data exchange between applications within a mobile operator's network) and inter-network usage (between mobile operator's network and value added service providers). The Generic User Profile provides a generic mechanism to access and manipulate user related data for suppliers and consumers. Using this mechanism the data can be retrieved and managed in a uniform way. However the data contents are not described within the Generic User Profile, but only the data model and a schema shall be defined. The suppliers and consumers of the data can be divided into the following groups of applications: terminal applications in the UE, applications in the home network, 3rd-party applications, and OAM/subscription management applications [46][47].

GUP provides a means of supporting access to data for ranges of services and functions (e.g. MMS, Presence). The support of users' services and personalization data may result in manipulating data in a structured manner and a standardized way of describing and accessing these data structures, utilizing a Data Description Method based on XML Schema . An important role in the provision of personalized and context aware services have also been the activities and the specifications related to the 3GPP Location Services (LCS) and Presence.

5. Automatic Discovery of Mobile Applications

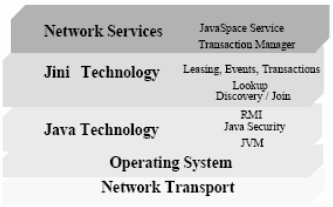
This section discusses automatic discovery of mobile applications. We introduce four potential technologies: JINI, Universal Plug and Play (UPnP), SALUTION, and Service Location Protocol. Note that service discovery should be based on the factors of user policy and service mobility, and should support single sign-on whenever possible.

5.1 JINI

The purpose of the Jini architecture is to federate groups of devices and software components into a single, dynamic distributed system [29]. Jini systems provide mechanisms for service construction, lookup, communication, and use in a distributed system.

The heart of the Jini system is a trio of protocols called discovery, join, and lookup [29]. The following steps show what interactions are needed among a client, a service provider, and a lookup service for a service to be used by the client in a Jini community [28][29].

1) Service provider locates a lookup service by multicasting a request on the local network or a remote lookup service known to it in priori.
2) A service provider registers a service object and its service attributes with the lookup service. This service object contains Java programming language interface for the service, including the methods that users and applications will invoke to execute the service, along with any other descriptive attributes.
3) A client requests a service by Java type and, perhaps, other service attributes. A copy of the service object is moved to the client and used by the client to talk to the service.
4) Then, client interacts directly with the service provider via the service object.



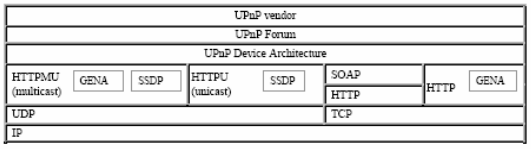
<Figure 17> Architecture of Jini Connection Technology

Based on Java technology as shown in Fig.17 [28][29], Jini technology uses Java Remote Method Invocation protocols to move code around the network. Network services run on top of the Jini software architecture.

The lookup service can be viewed as a directory service, in that services are found and resolved through it. In a Jini community, services register their proxy objects with a lookup service through discovery and join process, and clients query the lookup service to find out the services they want. Jini uses three related discovery protocols, useful in different situation [30][31]. Multicast Request Protocol is used when an application or service first becomes active, and needs to find lookup services in the vicinity. Multicast Announcement Protocol is used by lookup services to announce their presence to the services that may have interest in the community. Unicast Discovery Protocol is used to establish communications with a specific lookup service known to it in priori over a wide-area network. Access to services in the Jini system is granted on lease basis: A service is requested for a time period and, then, granted for negotiated period between the service user and provider. This lease must be renewed before its expiration. Otherwise, the resources associated with the services are released. Besides the basic service discovery/join and lookup mechanism, Jini supports remote events and transactions that help programmers write distributed programs in reliable and scalable fashion. Remote event enables an object to be notified when desired change occurs in the system. These events can be triggered by newly-published services or some state changes of services.

5.2 Universal Plug and Play [32][33][108]

Universal Plug and Play (UPnP) is architecture for pervasive peer-to-peer network connectivity of intelligent appliances, wireless devices, and PCs of all form factors. Although it's introduced as an extension to the plug and play peripheral model, UPnP is more than a simple extension to it. In UPnP, a device can dynamically join a network, obtain an IP address, convey its capabilities upon request, and learn about the presence and capabilities of other devices. Finally, a device can leave a network smoothly and automatically without leaving any unwanted state behind [32]. Universal Plug and Play leverages TCP/IP and the Web technologies, including IP, TCP, UDP, HTTP and XML, to enable seamless proximity networking in addition to control and data transfer among networked devices in the home and office.



<Figure 18> UPnP protocol stack

For communication between devices, UPnP uses the protocol stack shown in Fig.18 [32]. According to the latest specification [32], UPnP features can be epitomized as the following five steps.

1) Discovery: The UPnP discovery protocol is based on SSDP. When a device added to the network, the device advertises its services to the control points on the network. Similarly, when a control point is added to the network, the UPnP allows that control point to search for devices of interest on the network.
2) Description: After a control point has discovered a device, the control point still knows very little about the device. For the control point to learn more about the device and its capabilities, or to interact with the device, the control point must retrieve the device's description from the URL provided by the device in the discovery message.
3) Control: After a control point has retrieved a description of the device, the control point can send actions to a device's service. To do this, a control point sends a suitable control message to the control URL for the service.
4) Eventing: An UPnP description for a service includes a list of actions the service responds to and a list of variables that model the state of the service at run time. The service publishes updates when these variables change, and a control point may subscribe to receive this

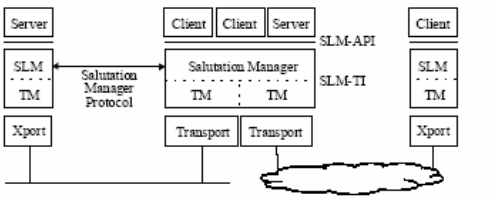
information. The service publishes updates by sending event messages.

5) Presentation: If a device has a URL for presentation, then the control point can retrieve a page from this URL, load the page into a browser, and depending on the capabilities of the page, and allow a user to control the device and/or view device status.

Another important feature of UPnP is automatic configuration of IP addresses being plugged in. When a device is connected to the network, it tries to acquire an IP address from a DHCP server on the network. But in the absence of a DHCP server, an IP address is claimed automatically from a reserved range for the local network use.

5.3 SALUTATION [32][33][108]

Salutation is another major cooperation architecture, which is being developed by the Salutation Consortium, to solve the problems of service discovery and utilization among a broad set of appliances and equipment and in an environment of widespread connectivity and mobility. Given the diverse nature of target devices, it is processor, operating system and communication protocol independent. The architecture provides a standard method for applications, services and devices to describe and to advertise their capabilities to other applications, services and devices. The architecture also enables application, services and devices to search other applications, services or devices for a particular capability, and to request and establish interoperable sessions with them to utilize their capabilities [34].



<Figure19> Model of the Salutation Manager

As shown in Fig.19 [34], the Salutation architecture is composed of two major components: Salutation Manager and Transport Manager. The Salutation Manager is the core of the architecture, similar to the lookup service in Jini. It is defined as a service broker. A service provider registers its capability with a Salutation Manager. When a client asks its local Salutation Manager for a service search, the search is performed by coordination among Salutation Managers. Then, the client can use the returned service. A Salutation Manager sits on the Transport Managers that provide reliable communication channels, regardless of what the underlying network transports are. The Salutation Manager provides a transport-independent interface to Server and Client applications. This interface (SLM-API) includes service registration, service discovery, and service access function.

The main tasks provided by the Salutation Manager can be summarized as follows.

1) Service Registry: The Salutation Manager contains a registry to keep information about Services. This is correspondent to the lookup service in Jini.
2) Service Discovery: The Salutation Manager discovers other Salutation Managers and services registered there. Remote services are discovered by matching type(s) and set of attributes specified by local Salutation Manager. This cooperation among Salutation Managers forms the conceptually same lookup service, but distributed over the network, as Jini does.
3) Service Availability: A client application can ask the local Salutation Manager to periodically check the availability of services. This is a narrow version of Jini's Remote Event concept.
4) Service Session Management: A service session is established when a client wants to use a service discovered through Service Discovery. The service session is operated in one of 3 different modes: native mode, emulated mode, and salutation mode. The Salutation Manager may or may not be involved in message exchanges in the service session, depending on the modes.

5.4 Service Location Protocol [32]-[38]

Service Location Discovery (SLP) [34] is an IETF version of service discovery protocol but it has unique backgrounds, target areas, and features, as other service discovery protocols do. SLP is a decentralized, lightweight, scale and extensible protocol for service discovery within a site [33]. SLP defines Service URL which defines service type and address for the service. For example, "service:printer:lpr://hostname" s the Service URL for line printer service available at hostname. Based this Service URL, a user browses services available in its site and makes use of selected services to meet the user's need. For example, a user (application) uses SLP to find out any color printer on the same floor.

There are three agents in SLP: User Agent (UA), Service Agent (SA), and Directory Agent (DA). UA is a software entity that sends service discovery requests on behalf of a user application. SA is an entity that advertises service on behalf of a service. As a centralized service information repository, DA caches advertisements from SAs and, afterwards, responds to requests from UAs. An SA advertises itself by registering with a DA. This registration message contains the URL for the advertised service, lifetime for the service, and a set of descriptive attributes for the service. The SA should periodically refresh the registration with DA before its expiration. This lifetime is meant to prevent the network from being left in transient state and similar concept is found at other service discovery protocols such as Jini and UpnP. A DA caches the registration and sends an acknowledge message to the SA. A UA send a service request message to the DA to request the location of a service. Then, the DA responds with a service reply message including the URLs of the services matched against the UA needs. Now, the UA can access the service pointed by the returned URL. In SLP, DA is optional. There may be no DA in small networks. In this case the UA's service request message is directly sent to SAs.

The UA can request query operators such as AND, OR, comparators (=, <, <=, >, >=), and substring matching. This is more powerful than others. For example, in Jini, service attribute matching can be done only against equality. Finally, SLP is said to be a solution to the intranet service discovery needs but it scales well to larger network. The scalability is supported by various features such as the minimal use of multicast messages, scope concept, and multiple DAs.

III. Example Application of FMC

In this chapter, we introduce an example "Ubiquitous Personal (U-Personal) Mobility Service" of a future FMC service and discuss functional features and requirements for U-Personal Mobility Service. We will also propose a generic model for U-Personal Mobility Service.

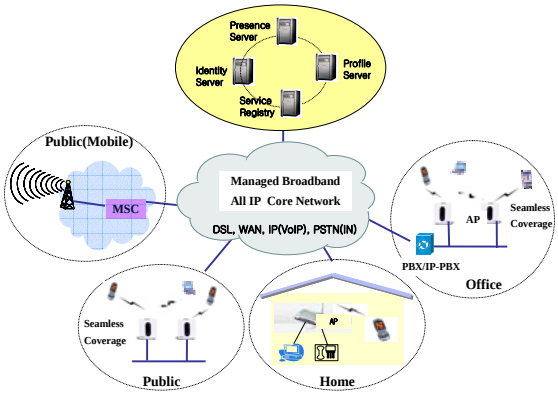
Focus of discussions in this chapter is on the issue of how the support technologies surveyed in the previous section can be projected to the FMC services as unified service enablers. Our focus is on the issues of single sign-on, single identity and single user profile.

1 Ubiquitous Personal Mobility Service

1.1 Introduction to U-Personal Mobility Service

U-Personal Mobility Service is an aggregate service environment for advanced personalized convergence services with mobility support. It consists of both circuit-switched and packet-switched networking technologies and service infrastructures.

U-Personal Mobility Environment provides a convergent networking/computing services environment with the joint characteristics of ubiquitous networking, mobile networking, context-aware computing, service-oriented computing, and personalized services. U-Personal Mobility Environment should include advanced wireline/wireless convergent multimedia services based on emerging networking/development environments of BcN and open APIs (e.g., Parlay/OSA).



<Figure 20> U-Personal Mobility Networking Environment

Figure 20 shows networking environment to host the U-Personal Mobility Environment, It consists of various networks (e.g., PSTN, UMTS/CDMA, AIN/CAMEL, IP-based WAN, etc), networking technologies (e.g., LAN, DSL, Bluetooth, WiFi, WiBro, PBX/IP-PBX, MPLS, etc), and service/control subsystems (e.g., IMS, VoIP, etc). Note that different combinations of access networking technologies are available in different service access zones. Mobile network access is assumed everywhere. Regardless of the access technologies available, both voice/phone and data/Internet services should be provided to users in all the zones. Seamless handling of location registration, authentication, roaming, and hand-over should be included as part of the U-Personal Mobility Environment features.

1.2 Requirements for U-Personal Mobility Service

Requirements for U-Personal Mobile Service are as follows:

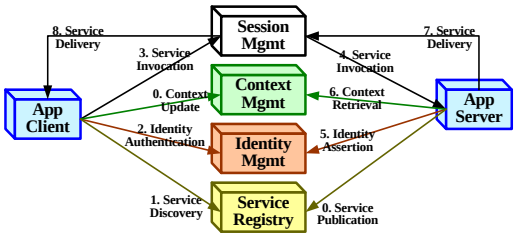
- ~~Automatic registration and authentication – single sign-on;~~
- Single identity – embedded in user device (potentially mapped with RFID);
- Single billing – voice call (PSTN, Mobile, VoIP), Internet access, contents;
- Single presence – Office Zone, Home Zone, Public Zone, or out of Zone, etc;
- Single user profile - single identity based single user profile;
- Triggering mechanism for call delivery service – call delivery and service portability/mobility among PSTN(IN), CDMA(CAMEL/WIN), and VoIP(IMS/softswitch);
- Managing user agent under user policy – time (on/off duty, day of the week, etc), location (office Zone, Home Zone, Public Zone, out of Zone, etc), and available applications (phone, messenger, e-mail, etc);
- Discovering the mobile application services – (RFID based) automatic service discovery.

삭제됨: n

2 Convergence Architecture for U-Personal Mobility Service

2.1 Generic Model

In attempting to develop U-Personal Mobility architecture, we first introduce a generic model for unified management of services and applications. The model is illustrated in Figure 21, where four service management functions are shown: session management, context management, identity management, and service registry.



<Figure 21> Generic model for unified service management environment.

Before any service invocation and consumption, it is assumed that information for the service should be published by

the application server at the Service Registry, and that context information regarding the client should be updated at the Context Management function. Then, the client can discovery the service and invoke the service with its identity information. Upon receiving the service request, the application server consults with the Identity Management to confirm the identity of the client for authentication and authorization purposes. Once authenticated, context information for the client can be retrieved from the Context Management function to best serve the client for the request. Note that the service management procedure shown in the figure serves only for the purpose of high-level illustration. Actual procedure(s) would vary depending upon technologies employed.

2.2 Integration of FMC support Technologies

In this subsection, we will discuss on the issues of how the support technologies surveyed in chapter II can be projected to the convergence architecture for U-Personal Mobility Service. We will focus on the issue of automated service discovery, automated authentication (or single sign-on), and single context management.

We propose convergence architecture for unified context, identity, and service registry management, where technology and service components are distributed across three different networking environments: terminal, network, and application. The terminal environment is implemented in user devices such as PCs, PDAs, and cellular phones. Application clients and user/device agents reside within the terminal environment. The server-side counterparts of applications and services exist within the application environment. Typically, server functions are deployed in PCs and workstations. All the rest of networking and service enabling functions go into the network environment, which in general is provided by network operators. It is important to note that, while application servers are located within the application environment, servers for supporting functions reside with the network environment.

Among many different groups of data networking and services technologies, IMS and associated technologies within and atop provide most rich and complete set of functionalities. Based on SIP and SIMPLE, IMS provides control plane functionality for network-level session management and messaging capabilities. Started as the 3GPP solution for UMTS mobile networks, IMS is extending its presence through wireline networks as the technology of choice toward all-IP networking. Thus, it is recommended that IMS or something equivalent serve as the basis of the U-Personal Mobility Environment. The combination of SIP/SIMPLE, IP (potentially with Mobile IP), and underlying access networking technologies like WiFi and WiBro can provide basic ubiquitous and mobile networking capabilities for the U-Personal Mobility Environment.

By adding capabilities of unified management of contexts, identities, and service registries, the U-Personal Mobility Environment can be further enhanced for context awareness, service personalization, and service-oriented computing. For service discovery functionality, we expect that many technologies, protocols, and systems will coexist as the functionality is highly dependent upon the associated service delivery and execution platforms.

For user profile management (or context management in general), GUP seems most proper choice out of technologies surveyed before. Although GUP was initiated with 3GPP, the scope of GUP is beyond the mobile networking environment. The interface and profile information definitions of GUP are generic enough to serve various tasks of profiling user related information, ranging from device capabilities and user preferences through user presence and location information. Furthermore, GUP's capability to support distributed storing of profile information enables smooth interworking with other context management systems and technologies, such as SIMPLE-based presence service and UAProf-based profile service. As both GUP and ID-SIS are based on ID-WSF (over Web services framework), they can evolve together with close relationship. WAP capabilities.

Due to relatively heavy processing and memory requirements and poor performance of SOAP/XML-based Web services engines, GUP client functionality is recommended to be included only in fixed terminals (e.g, PCs and laptops), but not in mobile terminals. Alternatively, a special-purpose (GUP-optimized), light-weight version of Web services engine can be developed for mobile terminals. Without GUP, UAProf can cover the profile management functionality in mobile terminals through MExE or WAP capabilities.

For user identity management, we would recommend the use of ID-FF. This is because ID-FF is the common denominator of capabilities of both fixed and mobile terminals. However, as GBA and ID-FF are optimized solutions for their own operating environments, we actually recommend the use of the both technologies.

IV. Conclusion

In this paper, we first provide a survey on diverse service enabling technologies from various standards bodies including IETF, 3GPP, OMA, Liberty Alliance, and W3C, for the areas of single sign-on, single contexts, service mobility, unified profiling, and automated service discovery. Then, after the supporting technology survey, an example service "U-Personal Mobility Service" of FMC was investigated.

Along the same line of the technology trend in the telecommunications industry, our viewpoint was IP centric. Moving forward from the current mixed-network status to the future all-IP environment, the following three steps will be needed to evolve the paths of developing a ubiquitous and personalized convergence service environment:

1) Interworking of heterogeneous voice networking environments
2) Interworking of voice and data networking environments
3) Interworking of heterogeneous data networking environments

After many years of heavy research and development in the

삭제됨: **Convergence Architecture**

삭제됨: In this subsection, we propose convergence architecture for integrating the support technologies surveyed in chapter II for U-Personal Mobility Service. In next subsection, we will focus on the integrating issues of automated service discovery, automated authentication (or single sign-on), and single context management.

삭제됨: **<#>Integrating of FMC Support Technologies** In this subsection, we will discuss on the issues of how the support technologies surveyed in chapter II can be projected to the convergence architecture for U-Personal Mobility Service. We will focus on the issue of automated service discovery, automated authentication (or single sign-on), and single context management.

삭제됨: ere

삭제됨: ere

삭제됨: into

industry, it is believed that good practices and solutions already exist for the first two topics. On the other hand, the last topic is new and technologies in the space are still emerging.

Providing solutions to meet the requirements for U-Personal Mobility Service is a challenging task, especially when the network environment consists of a mixture of fixed and mobile, and voice and data technologies. Although technologies exist today to support the single identity and other features, they only fulfill the requirements piecewise, and are applicable only to certain network technologies and conditions in their current forms.

However, if we have a robust identity management solution for single identity, solutions for the rest of features such as single billing, single presence, and single user profile can be built upon the identity management solution.

[References]

[1] "Microsoft .NET Passport: review guide," January 2004

[2] David P. Kormann and Aviel D. Rubin, "Risks of the Passport Single Signon Protocol," Computer Networks, Elsevier Science Press, volume 33, pages 51-58, 2000 .

[3] 3GPP TS 33.220: "Generic Authentication Architecture (GAA); Generic Bootstrapping Architecture."

[4] 3GPP TR 33.919: "Generic Authentication Architecture (GAA); System description."

[5] 3GPP TS 33.210: "Network Domain Security; IP network layer security."

[6] 3GPP TS 31.102: "Characteristics of the USIM application."

[7] 3GPP TS 23.228: "IP Multimedia Subsystem (IMS)."

[8] 3GPP TS 33.102: "3G security; Security architecture."

[9] A. Niemi, *et al*, "Hypertext Transfer Protocol (HTTP) Digest Authentication Using Authentication and Key Agreement (AKA)," IETF RFC 3310, September 2002.

[10] Liberty Alliance specifications at http://www.projectliberty.org/resources/specifications.php

[11] OASIS Security Services (SAML) TC at http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=security

[12] OASIS Web Services Security (WSS) TC at http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wss

[13] Liberty Alliance Project, "Liberty ID-FF Architecture Overview, Version: 1.2," available at http://www.projectliberty.org/specs/liberty-idff-arch-overview-v1.2.pdf

[14] Liberty Alliance Project, "Liberty ID-WSF Security and Privacy Overview, Version: 1.0," available at http://www.projectliberty.org/specs/liberty-idwsf-security-privacy-overview-v1.0.pdf

[15] Liberty Alliance Project, "Liberty ID-SIS Employee Profile Service Specification, Version: 1.0," available at http://www.projectliberty.org/specs/liberty-idsis-ep-v1.0.pdf

[16] J. Sasak and T. Marenic, "Establishing Virtual Home Environment Across Terminals with Diverse Capabilities," 7th International Conference on Telecommunications, Zagreb, Croatia, June 2003.

[17] 3GPP TS 22.078: "Customised Applications for Mobile network Enhanced Logic (CAMEL); Service definition - Stage 1".

[18] "Service Requirement for the Open Services Access (OSA)," 3GPP TS 22.127, 2005-01.

[19] "Virtual Home Environment (VHE) / Open Service Access (OSA)," 3GPP TS 23.127, 2004-06.

[20] 3GPP TS 22.057: "Mobile Station Application Execution Environment (MExE); Stage 1 (Release 5)", V6.0.0 (2005-01).

[21] 3GPP TS 23.057 "Mobile Execution Environment (MExE); Functional description," V6.2.0 (2003-09).

[22] "The Virtual Home Environment," 3GPP TR 22.121, 2002-06.

[23] "External Functionality Interface Framework," Open Mobile Alliance, OMA-WAP-EFI-V1_1-20040609-C, June 2004.

[24] "Multimedia Broadcast/Multicast Service," 3GPP TS 22.146, 2004-09.

[25] "Open Services Architecture; Application Programming Interface," 3GPP TR 29.998, 2000-12.

[26] Maarten Wegdam, Dirk-Jaap Plas, and Musa Unmehopa, "Validation of the Open Service Access API for UMTS Application Provisioning," PROMS 2001, LNCS 2213, pp. 210-221, 2001.

[27] Thomas Magedanz, "Getting Started with Parlay/OSA: What, Why, How," Parlay Education Track Seminar, San Diego, May 19th, 2003.

[28] Sun Microsystems, Jini Connection Technology at http://www.sun.com/jini

[29] Sun Microsystems, Jini Community Resources: Jini Technology Architectural Overview at http://www.sun.com/jini /whitepapers/architecture.html

[30] Sun Microsystems, Jini Community Resources: Jini Specification v1 at http://www.sun.com/jini/specs

[31] W. Keith Edwards, "Core JINI," The Sun Microsystems Press Java Series, Prentice Hall, 1999.

[32] Microsoft Corporation, Universal Plug and Play: Background at http://www.upnp.org/resources/UpnPbkgnd.htm

[33] Microsoft Corporation, Universal Plug and Play Device Architecture at http://www.upnp.org/

[34] Salutation Consortium, "Salutation Architecture Specification," The Salutation Consortium (available at http://www.salutation.org)

[35] Bob Pascoe, "Salutation-Lite: Find-and-Bind Technologies for Mobile Devices," Salutation Consortium (available at http://www.salutation.org/whitepaper/Sal-Lite.PDF)

[36] Bob Pascoe, "Salutation Architectures and the newly defined service discovery protocols from Microsoft and Sun," Salutation Consortium (available at http://www.salutation.org/whitepaper/Jini-UPnP.PDF)

[37] Brent Miller, "Mapping Salutation Architecture APIs to Bluetooth Service Discovery Layer," Bluetooth Consortium

[38] IETF SVRLOC Working Group, "Service Location Protocol Home Page," available at http://www.srvloc.org

[39] E. Guttman, C. Perkins, J. Veizades, and M. Day, "Service Location Protocol, Version 2," IETF RFC 2608, June 1999 (available at http://www.ietf.org/rfc/rfc2608.txt).

삭제됨: f

삭제됨: based on identity management

삭제됨: developed

삭제됨: based

삭제됨: single

삭제됨: f

[40] Marco Liebsch and Ajoy Singh (editors), “Candidate Access Router Discovery”, Internet Draft draft-ietf-seamoby-card-protocol-06.txt, Internet Engineering Task Force, June 2004.
[41] INS: Intentional Naming System, available at http://wind.lcs.mit.edu/projects/ins
[42] The Berkeley Service Discovery Service, available at http://www.cs.berkeley.edu/~caerwin/sds-project.html
[43] 3GPP TS 22.240: “Service requirements for the 3GPP Generic User Profile (GUP); Stage 1”.
[44] 3GPP TS 23.240: “3GPP Generic User Profile (GUP) requirements; Architecture (Stage 2)”.
[45] N. Blefari-Melazzi, G. Ceneri, S. Salsano, “Ubiquitous Personalized Environment,” Progress Report for TASK 6.1 – QoS and mobility management in multi-access systems, PRIMO Project.
[46] 3GPP TS 32.140, “Telecommunication management; Subscription Management (SuM) requirements.”
[47] 3GPP TS 32.141, “Telecommunication management; Subscription Management (SuM) architecture.”
[48] J. Rosenberg, *et al*, “SIP: Session Initiation Protocol,” IETF RFC 3261, June 2002.
[49] M. Day, et al, “A Model for Presence and Instant Messaging,” IETF RFC 2778, February 2000.
[50] M. Day, *et al*, “Instant Messaging / Presence Protocol Requirements,” IETF RFC 2779, February 2000.
[51] J. Peterson, *et al*, “Common Profile for Instant Messaging (CPIM),” IETF RFC 3860, August 2004.
[52] B. Campell, J. Rosenberg, H. Schulzrinne, D. Willis, R. Sparks, C. Huitema, D. Gurle, “*Session Initiation Protocol (SIP) Extension for Instant Messaging”*, IETF RFC 3428, December 2002.
[53] J. Rosenberg, D. Willis, R. Sparks, B. Campell, H. Schulzrinne, J. Lennox, B. Aboba, C. Huitema, D. Gurle, D. Oran, “*SIP Extensions for Presence”*, Internet Draft, March 2001.
[54] J. Rosenberg, “A Presence Event Package for the Session Initiation Protocol (SIP),” IETF RFC 3856, August 2004.
[55] J. Rosenberg, “A Watcher Information Event Template-Package for the Session Initiation protocol (SIP),” IETF RFC 3857, August 2004.
[56] M. Handley and V. Jacobson, "SDP: Session Description Protocol", IETF RFC 2327, April 1998.
[57] P. Saint-Andre (Editor), “Extensible Messaging and Presence Protocol (XMPP): Core,” IETF RFC 3920, October 2004.
[58] P. Saint-Andre (Editor), “Extensible Messaging and Presence Protocol (XMPP): Instant Messaging and Presence,” IETF RFC 3921, October 2004.
[59] P. Saint-Andre, “Mapping the Extensible Messaging and Presence Protocol (XMPP) to Common Presence and Instant Messaging (CPIM),” IETF RFC 3922, October 2004.
[60] P. Saint-Andre, “End-to-End Signing and Object Encryption for the Extensible Messaging and Presence Protocol (XMPP),” IETF RFC 3923, October 2004.
[61] 3GPP TS 22.141: “Presence service; Stage 1,” V6.3.0 (2005-01).
[62] 3GPP TS 23.141: “Presence Service; Architecture and functional description,” V6.6.0 (2004-06).
[63] 3GPP TS 24.141, “Presence service using the IP Multimedia (IM) Core Network (CN) subsystem; Stage 3,” V6.2.0 (2005-01).
[64] 3GPP TS 23.228, “IP Multimedia Subsystem (IMS),” V6.8.0 (2004-12).
[65] 3GPP TS 23.218, “IP Multimedia (IM) session handling; IM call model,” V6.2.0 (2004-12).
[66] OMA, “WV-040 System Architecture Model,” Candidate Version 1.2 – 22, OMA-IMPS-WV-Arch-V1_2-20040522-C, May 2004.
[67] OMA, “Stage 2 - Presence using SIMPLE,” Draft Version 1.0 – 02 August 2004, OMA-PAG-SIMPLE-AD-V1_1_0-20040802-D.
[68] 3GPP TS 29.198-1,”Open Service Access (OSA); Application Programming Interface (API); Part 1: Overview,” V4.3.5 (2004-06).
[69] 3GPP TS 29.198-2,”Open Service Access (OSA); Application Programming Interface (API); Part 2: Common data,” V4.7.0 (2003-12).
[70] 3GPP TS 29.198-3,”Open Service Access (OSA); Application Programming Interface (API); Part 3: Framework,” V6.0.1 (2004-12).
[71] 3GPP TS 29.198-4,”Open Service Access (OSA); Application Programming Interface (API); Part 4: Call Control,” V2.0.0 (2001-06).
[72] 3GPP TS 29.198-5,”Open Service Access (OSA); Application Programming Interface (API); Part 5: Generic User Interaction,” V5.7.0 (2004-08).
[73] 3GPP TS 29.198-6,”Open Service Access (OSA); Application Programming Interface (API); Part 6: Mobility,” V4.5.0 (2003-03).
[74] 3GPP TS 29.198-7,”Open Service Access (OSA); Application Programming Interface (API); Part 7: Terminal Capabilities,” V5.6.0 (2004-08).
[75] 3GPP TS 29.198-8,”Open Service Access (OSA); Application Programming Interface (API); Part 8: Data Session Control,” V5.6.0 (2004-08).
[76] 3GPP TS 29.198-9,”Open Service Access (OSA); Application Programming Interface (API); Part 9: Generic Messaging SCF.”
[77] 3GPP TS 29.198-10,”Open Service Access (OSA); Application Programming Interface (API); Part 10: Connectivity manager SCF,” SPECIFICATION WITHDRAWN.
[78] 3GPP TS 29.198-11,”Open Service Access (OSA); Application Programming Interface (API); Part 11: Account management Service Capability Feature (SCF),” V6.3.1 (2004-12).
[79] 3GPP TS 29.198-12,”Open Service Access (OSA); Application Programming Interface (API); Part 12: Charging Service Capability Feature (SCF),” V6.3.1 (2004-12).
[80] 3GPP TS 29.198-13,”Open Service Access (OSA); Application Programming Interface (API); Part 13: Policy management Service Capability Feature (SCF),” V6.2.2 (2004-12).
[81] 3GPP TS 29.198-14, “Open Service Access (OSA); Application Programming Interface (API); Part 14: Presence and Availability Management (PAM) Service Capability Feature (SCF),” V6.2.2 (2004-12).

[82] 3GPP TS 29.198-15,"Open Service Access (OSA); Application Programming Interface (API); Part 15: Multi-media Messaging (MM) Service Capability Feature (SCF)," V6.1.1 (2004-12).
[83] 3GPP TS 11.11: "Specification of the Subscriber Identity Module - Mobile Equipment (SIM - ME) interface."
[84] 3GPP TS 11.14, "Specification of the SIM Application Toolkit for the Subscriber Identity Module - Mobile Equipment (SIM-ME)"
[85] 3GPP TS 22.038: "USIM/SIM Application Toolkit (USAT/SAT); Service description," V4.1.0 (2001-01).
[86] 3GPP TS 31.111: "USIM Application Toolkit (USAT)," V4.11.0 (2003-09).
[87] 3GPP TS 31.112: "USIM Application Toolkit (USAT) interpreter architecture description," V5.2.0 (2002-06).
[88] 3GPP TS 31.113: "USAT Interpreter Byte Codes," V6.2.0 (2003-09).
[89] 3GPP TS 31.114: "USAT Interpreter Protocol and Administration," V5.3.0 (2003-03).
[90] B. Moore (Editor), "Policy Core Information Model (PCIM) Extensions," IETF RFC 3460, January 2003.
[91] W3C, "Composite Capability/Preference Profiles (CC/PP): Structure and Vocabularies 1.0," W3C Recommendation, 15 January 2004 (available at http://www.w3.org/TR/2004/REC-CCPP-struct-vocab-20040115/)
[92] W3C, "CC/PP Implementors Guide: Privacy and Protocols," W3C Working Draft, 20 December 2001 (available at http://www.w3.org/TR/2001/WD-CCPP-trust-20011220/).
[93] W3C, "CC/PP Implementors Guide: Harmonization with Existing Vocabularies and Content Transformation Heuristics," W3C Note, 20 December 2001.
[94] W3C, "The Platform for Privacy Preferences 1.1 (P3P1.1) Specification," W3C Working Draft, 20 July 20004 (available at http://www.w3.org/TR/2004/WD-P3P11-20040720/).
[95] OMA, "User Agent Profile 1.1," Candidate Version 12-December-2002, OMA-WAP-UAProf-v1_1-20021212-c (available at http://www.openmobilealliance.org/release_program/docs/UAProf/OMA-WAP-UAProf-V1_1-20021212-C.pdf)..
[96] OMA, "WAP Architecture," Version 12-July-2001, WAP-210-WAPArch-20010712 (available at http://www.openmobilealliance.org/tech/affiliates/wap/wap-210-waparch-20010712-a.pdf).
[97] Hidetaka Ohto and Johan Hjelm, "CC/PP exchange protocol based on HTTP Extension Framework," W3C Note 24 June 1999 (available at http://www.w3.org/1999/06/NOTE-CCPPexchange-19990624).
[98] H. Nielsen, *et al*, "An HTTP Extension Framework," IETF RFC 2774, February 2000.
[99] OMA, "Wireless Profiled HTTP," WAP-229-HTTP-20010329-a, Version 29-Mar-2001 (available at http://www.openmobilealliance.org/tech/affiliates/wap/wap-229-http-20010329-a.pdf).
[100] OMA, "Wireless Application Protocol Wireless Session Protocol Specification," WAP-230-WSP, Approved Version 5-July-2001 (available at http://www.openmobilealliance.org/tech/affiliates/wap/wap-230-wsp-20010705-a.pdf).
[101] Tim Bray, et al, "Extensible Markup Language (XML) 1.1," W3C Recommendation 04 February 2004 (available at http://www.w3.org/TR/2004/REC-xml11-20040204/).
[102] J. Franks, et al, "HTTP Authentication: Basic and Digest Access Authentication," IETF RFC 2617, June 1999.
[103] J. Rosenberg, et al, "SIP: Session Initiation Protocol," IETF RFC 3261, June 2002.
[104] Telcordia Technologies, *AINGR: SWITCHING SYSTEMS*, GR-1298-CORE, Nov. 1999.
[105] Telcordia Technologies, *ISCP Generic Data Interface Specification for TCP/IP Using GR-246-CORE ANSI TCAP Encoding*, SR-3389, May 1999.
[106] JAIN Website at http://java.sun.com/products/jain
[107] JAIN, "Java Call Control (JCC) API", Version 1.0, January 2000.
[108] Yaron Y. Goland, Ting Cai, Paul Leach, Ye Gu, and Shivaun Albright, "Simple Service Discovery Protocol," IETF Draft draft-cai-ssdp-v1-03.txt, October 28, 1999 (available at http://www.ietf.org/internetdrafts/draft-cai-ssdp-v1-03.txt)

페이지 1: [1] 삭제됨	admin	2005-04-21 오후 8시 34분
the		
페이지 1: [1] 삭제됨	admin	2005-04-21 오후 8시 36분
M		
페이지 1: [1] 삭제됨	admin	2005-04-21 오후 8시 36분
B		
페이지 1: [1] 삭제됨	admin	2005-04-21 오후 8시 36분
N		
페이지 1: [2] 삭제됨	admin	2005-04-21 오후 8시 49분
QoS		
페이지 1: [2] 삭제됨	admin	2005-04-21 오후 8시 49분
S		
페이지 1: [2] 삭제됨	admin	2005-04-21 오후 8시 49분
C		
페이지 1: [2] 삭제됨	admin	2005-04-21 오후 8시 49분
S		
페이지 1: [2] 삭제됨	admin	2005-04-21 오후 8시 49분
A		
페이지 1: [2] 삭제됨	admin	2005-04-21 오후 8시 49분
M		
페이지 1: [2] 삭제됨	admin	2005-04-21 오후 8시 55분
the		
페이지 1: [3] 삭제됨	admin	2005-04-21 오후 8시 55분
s		
페이지 1: [3] 삭제됨	admin	2005-04-21 오후 8시 58분
s		
페이지 1: [4] 삭제됨	admin	2005-04-21 오후 10시 44분
the		

페이지 1: [4] 삭제됨	admin	2005-04-21 오후 10시 44분
M		
페이지 1: [4] 삭제됨	admin	2005-04-21 오후 10시 44분
B		
페이지 1: [4] 삭제됨	admin	2005-04-21 오후 10시 44분
N		
페이지 1: [5] 삭제됨	admin	2005-04-21 오후 11시 17분
S		
페이지 1: [5] 삭제됨	admin	2005-04-21 오후 11시 17분
C		
페이지 1: [5] 삭제됨	admin	2005-04-21 오후 11시 17분
S		
페이지 1: [5] 삭제됨	admin	2005-04-21 오후 11시 17분
A		
페이지 1: [5] 삭제됨	admin	2005-04-21 오후 11시 17분
M		
페이지 1: [6] 삭제됨	admin	2005-04-21 오후 11시 3분
A		
페이지 1: [6] 삭제됨	admin	2005-04-21 오후 11시 16분
-		
페이지 1: [6] 삭제됨	admin	2005-04-21 오후 11시 2분
E		
페이지 1: [6] 삭제됨	admin	2005-04-21 오후 11시 3분
N		
페이지 1: [6] 삭제됨	admin	2005-04-21 오후 11시 3분
C		
페이지 1: [6] 삭제됨	admin	2005-04-21 오후 11시 3분
R		

페이지 1: [6] 삭제됨	admin	2005-04-21 오후 11시 4분

A

페이지 1: [6] 삭제됨	admin	2005-04-21 오후 11시 7분

A

페이지 1: [6] 삭제됨	admin	2005-04-21 오후 11시 8분

coverage

페이지 1: [6] 삭제됨	admin	2005-04-21 오후 11시 14분

.

페이지 1: [6] 삭제됨	admin	2005-04-21 오후 11시 13분

These services are thus called

페이지 1: [6] 삭제됨	admin	2005-04-21 오후 11시 10분

(Fixed Mobile Convergence)

페이지 1: [7] 삭제됨	admin	2005-04-21 오후 11시 50분

both

페이지 1: [7] 삭제됨	admin	2005-04-21 오후 11시 51분

to support

페이지 1: [7] 삭제됨	admin	2005-04-21 오후 11시 51분

IP

페이지 1: [7] 삭제됨	admin	2005-04-21 오후 11시 51분

-level session QoS

페이지 1: [7] 삭제됨	admin	2005-04-21 오후 11시 52분

IP

페이지 1: [7] 삭제됨	admin	2005-04-21 오후 11시 53분

-

페이지 1: [7] 삭제됨	admin	2005-04-21 오후 11시 53분

session QoS

페이지 1: [7] 삭제됨	admin	2005-04-21 오후 11시 37분

S

페이지 1: [7] 삭제됨	admin	2005-04-21 오후 11시 46분

also on

페이지 1: [7] 삭제됨	admin	2005-04-21 오후 11시 49분

both

페이지 1: [7] 삭제됨	admin	2005-04-21 오후 11시 50분

and on how to support security mechanisms

페이지 4: [8] 삭제됨	user	2005-04-22 오후 5시 22분

High-level ID-FF model is depicted in Figure 5, where there are three distinct roles within a federation: User Agent (UA), Identity Provider (IdP) and Service Provider (SP). For single sign-on and identity federation procedures, ID-FF defines three different mappings/profiles of message types and flow patterns: Liberty artifact profile, Liberty browser POST profile, and Liberty-enabled client and proxy profile. Although details are different from one profile to another, all three profiles share the same abstract-level principles. Figure 6 shows the ID-FF single sign-on procedure based on the Liberty artifact profile.

페이지 4: [9] 삭제됨	user	2005-04-22 오후 5시 22분

Individual user’s information on how personalized services should be provided and presented is contained in a set of user profiles. Each user’s profile consists of user interface related information and subscribed services related information. Because users can have multiple roles for different situations or needs (for instance, being at work or at home), a single user can have multiple profiles. User’s profiles are provided and controlled by the home environment (HE). Home environment value-added service providers (HE-VASPs) maintain service agreements with the HE to make personalized services available to users. In addition to home services, a user can utilize other services provided by value-added service providers.