Exploiting Access Control Information in User Profiles to Reconfigure User Equipment

Giovanni Bartolomeo, Stefano Salsano, Nicola Blefari-Melazzi
Dipartimento di Ingegneria Elettronica, University of Rome "Tor Vergata", Rome, Italy
stefano.salsano@uniroma2.it, giovanni.bartolomeo@uniroma2.it, , blefari@uniroma2.it

Abstract

Reconfigurable radio systems allow user terminals to access different communication technologies, without duplication of hardware. To this end, terminals need appropriate software (SW) modules, which, in advanced scenarios, can be downloaded on demand from the network. The network element that provides these modules to the terminals may need to know information about the terminal/user, in order to better adapt the SW modules to user needs and terminal capabilities and to grant only requests coming from authorized users. For this reason, such network element needs to access user profile information. In this paper, we propose a distributed approach to retrieve users profile information that does not require the mandatory presence of a public network operator. The goal of this mechanism is to allow a suitably appointed entity to retrieve all the needed information, related to the user/terminal, from a suitable location/server in the network, given some information provided by the terminal itself. Our approach is based on the "Dataweb" paradigm, under standardization in the Oasis consortium.

1. Introduction

"Reconfigurable radio" or "software radio" technology enables mobile devices to dynamically reconfigure the wireless interfaces up to the physical layers. Without duplication of hardware, this technology will make it possible for a terminal to use a wide range of communications systems, ranging from cellular, wireless local area networks, wireless personal area networks, radio/tv broadcast networks.

In the more advanced scenario, the reconfiguration should be possible "on the fly" by downloading new SW modules on a wireless channel. In this scenario there is the need of a communication channel between the base station that drives the reconfiguration process and the terminal to be reconfigured. This communication channel may be used: 1) to discover which communications systems are available; 2) to select which system should be downloaded according to the user needs; 3) to perform authentication/authorization procedures so that a given module can be downloaded only by authorized terminals/users; 4) to download the modules themselves.

The E2R II project co-founded by the European Union under the IST framework program deals with the whole set of issues related to reconfigurable radio systems. Among these issues, E2R II is addressing the definition of a communication channel between terminals and base stations named "Cognitive Pilot Channel" (CPC). More specifically, the CPC is a channel between the terminal and an entity that controls the radio access networks. This channel can include the functionality listed above for the control of the reconfiguration process; however, the CPC functionality is not limited to software radio reconfiguration of devices. As of the current status of the work in the E2R II project, the physical definition of the CPC channel is not completed, yet. The CPC channel could be a dedicated physical channel or it could be transmitted over one or more different access technologies. A logical view of the CPC channel is given in Figure 1, where an entity called Access Network Controller ANC communicates with the terminal over the CPC, in a scenario comprising several Radio Access Technologies (RAT)

The reconfiguration process can be logically divided in three phases. In the first phase the discovery, negotiation, authentication and authorization functions are executed. In this phase the terminal can select the desired system and/or the network can choose which system is more appropriate for the terminal/user; the ANC can check if the user is entitled to download a given module; the ANC could configure the module to

Work performed in the framework of the EU funded project E2R II

be downloaded according to user/terminal characteristics and requirements. In the second phase, the module is downloaded into the terminal, while in the third phase the radio device is reconfigured.

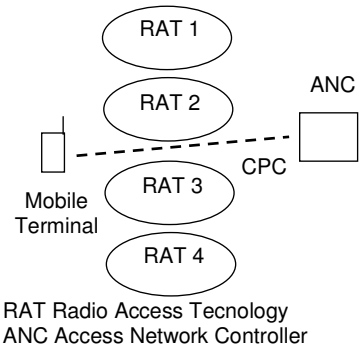

Figure 1: Logical view of the CPC channel

During the first phase, the ANC need to access a set of information related to the user and to his terminal. The information that characterizes a user/terminal can be collectively referred to as "user profile". Part of this information can be stored in the terminal itself so that it can be presented to the base station during the negotiation phase; more in general, it is likely that most of this information is kept in some "network side" repository. Just to give an example, in cellular networks the terminal just provides a user identifier (the IMSI) and then the user profile information is retrieved from the HLR.

The example of the cellular network well describes the concept of "network side" storage of user profile information. On the other hand, it represents a case of "structured" approach: the relevant standard bodies defined the IMSI and the related procedures (login, authentication ecc.). All GSM/3G cellular network providers and vendors comply to thess standards and the interoperability is ensured. The terminal/users need to have a valid contract with a public network operator in order to gain access to the network.

We argue that this structured approach should be complemented by a looser and more distributed approach, which still makes it possible to retrieve users profile information but do not require the mandatory presence of a public network operator. For example in an enterprise scenario or a campus scenario, the "provider" is not a public network operator, and the users/terminals that needs to be reconfigured are not necessarily tied to a contract with a network operator.

Therefore, we think that a more general mechanism for user identification and for retrieving user profile information is desirable, for the initial phase of the reconfiguration process. Obviously, this mechanism should also be able to support (with the same level of security) the procedures and mechanisms of a "structured approach", like the one adopted in current cellular networks.

The goal of this mechanism is to allow an ANC entity to retrieve all the needed information, related to the user/terminal, from a suitable location/server in the network, given some information provided by the terminal itself. The ANC entity should be recognized as an entity which is authorized to access the user profile information. In this paper, we propose a new approach to address these issues, based on the "Dataweb" paradigm under standardization in the Oasis consortium.

In section II we discuss the "classical" approach to user profile definition and management. In section III some important security/privacy aspects related to user profile are discussed. In section IV the Oasis Dataweb paradigm is shortly introduced. In section V a generic architecture to handle profile information is proposed.

2. Profile Definition

The concept of user profile as understood in this paper has a very wide meaning. It is the repository of all the information related to the user, his devices, the services he uses, the networks and so on. It also includes information and data under the user control and information and data under control of the users' applications and services.

A complete solution for handling user profiles basically includes two different aspects. The first one is the representation of the profile content (the profile schema). The second one includes: i) the definition of the architectural entities that will deal with the profile information; ii) the definition of the interactions among these entities. Both aspects should be subject of standardization, since a solution to handle user profile will be successful if it is adopted by a large number of users/ systems. Ideally, a profile handling solution should be universally applicable to different networks/terminals/services.

An attempt to provide such a universal solution is the GUP (Generic User Profile) [1] defined by 3GPP, which has been developed taking into account the requirements of a future cellular network. According to 3GPP documents, the GUP is the collection of user related data which affects the way in which an individual user experiences services and which may be accessed in a standardised manner. The objective of specifying the GUP is to provide means to enable

harmonised usage of the user-related information originating from different entities. The specification of the GUP is also flexible enough to meet future developments. The 3GPP GUP specification provides a data description mechanism and an architecture with interfaces and mechanisms to handle the data. Note that the GUP specification does not provide a concrete schema for a user profile, rather it provides a framework to define such a schema.

Starting from the work carried out in 3GPP on the GUP, in the context of the Simplicity project [2] [3], we have defined a concrete schema for a user profile. This profile is named Simplicity User Profile (SUP); it was meant to be as general as possible and can be enriched with new and more advanced features. The proposed SUP includes five components (see Figure 2): user profile, device profile, network profile, service profile and PID (Personal Identification Device) profile. The PID is a device that can assist the user in presenting his identity and transferring his profile information to the networks, the terminal devices, the services. The most known example of PID is the Subscribe Identity Module (SIM), anyway other solutions (including java cards, memory cards or even mobile phones) are possible. Note that having a "physical" PID is not strictly needed for the scenarios described in this paper, as the PID can be replaced by other forms of user identification, such as typing a login/password sequence.

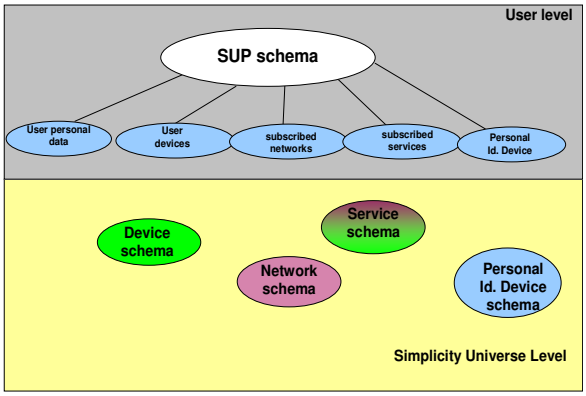

Figure 2: User Profile – Abstract view (xml schemas)

The SUP is a "User Level" representation of the user himself and of his surrounding "information and communication technology" world. The SUP provides a logically unified representation of the information related to the user and of the "ICT" context in which the user is "embedded": the devices that he is using and that he owns, the services he has subscribed to, the network he is accessing or he could access. Figure 2 shows also that there is another level of representation, in addition to the "User Level", the "Simplicity Universe Level", which contains the description of all existing devices, access networks, services and simplicity devices. Suitable XML schemas have been defined in order to describe each component [4]. For each component we investigated existing proposals and standards and integrated them in our proposal whenever possible.

The "user personal data" component includes information such as identity, biographical information, language, user's interests, hobbies and so on. This component also includes a free area that can be personalized and handled by external applications that want to store and retrieve personalization/configuration information related to the user. The part of user profile component that holds personal information data is defined taking as reference the Liberty Alliance Project Personal Profile (PP) [5].

The "user device" component provides information on the devices owned or leased by the user (device type, device capabilities). The device profile is based on a UAProf Schema provided by the WAP Forum [6], which is an RDF schema document describing the different hardware elements (e.g., mobile devices, presentation devices, terminals, etc.). In order to port the UAProf into a SUP component, the RDF schema has been translated into an XML Schema.

The "subscribed networks" component contains connection preferences, policies, network parameters and accounts for a specific network. Within this component, we can find information about user's accounts and the way the user has personalized his connection accounts.

The "subscribed services" component, like the "subscribed networks", is not currently defined in any standard, at least not in a compact and comprehensive way. The "subscribed services" component is divided in three sub-components: *ServiceList*, *SessionList* and *PrefsPolicies*. *ServiceList* contains the list of subscribed services. *SessionList* contains the list of suspended sessions, in order to be able to resume them.

The PID component includes the information about the type of the PID owned by the user and its hardware/software capabilities.

As for the architecture to access profile information, we propose a more distributed architecture which respect to the 3GPP GUP architecture, as it will be described in section 5 hereafter.

To conclude this section, we note that the limitation of this approach proposed by 3GPP (and adopted by

Simplicity) is that it foresees a standardization agreement or the prevalence of a "de-facto" standard for the definition of a complete user profile schema.

3. Security, trust and privacy aspects

The aspects related to protection of personal data are of fundamental importance when dealing with user profiles. In recent years, in fact, the traditional concept of security has been sided by a relatively new aspect, namely privacy, which has gained much more importance in parallel with the growing attention the national governments began to pay to electronic data handling. In this section we report a summary of our search in this area together with consideration on suitability and feasibility.

The GUP component called "common properties" could be employed also to describe a mechanism for data access control. But, unfortunately, and probably because of the abstract nature of the GUP, the existing specifications do not give any indication on how the control mechanism works and how the access right are stored in the user profile. To be more precies, in an earlier GUP specification, an xml schema for common properties was expected to be available at the URI http://www.3gpp.org/gup-ns/common i.e. the namespace bound to the GUP common properties). In a later specification, 3GPP stated that the whole GUP will use the work performed in the Liberty Alliance Project. In fact, inside the Liberty framework there exist some specifications for supporting privacy policy and preferences. Rather than defining a semantic for this purpose, the Liberty proposal focuses on defining a mean of agreement between a user (agent) ("Principal") and a Service Provider (SP). It assume that a way to describe generic policies for privacy handling has been chosen. The agreement is based on a comparison between the "level of privacy" offered by the SP and the one desired by the Principal. A set of five levels of privacy has been proposed, ranging from "strict" (highest level) to "casual" (lowest level). The agreement is reached if the SP offers an equal or lower level that the one requested by the Principal, otherwise the transaction is aborted.

But what about the semantics? The Liberty's proposal, named PPEL [7], is an abstract way of defining privacy rules, and, according to the specifications, may use as a concrete syntax the W3C Platform for Privacy Preferences (P3P) [8]. P3P is an ongoing W3C standard for Service Provider to describe in xml format the privacy practices a Service Provider conforms to. One or more policies can be associated to any resource (e.g. a web form asking user's data or retrieving a cookie) pointed by an URI (in latest version the specifications provide a new binding mechanism for increasing granularity beyond the URI level and allowing policies to apply to content inside a resource pointed by an URI) which the user agent is going to access; Each policy describes which kind of data the resource will access, the purpose of the data collection, who will make use of these data, how long data will be kept and what happens in case the service, while using the data, will not comply with the declared policies and which organization is the responsible for resolving disputes. P3P also defines a taxonomy for user profile data, by enumerating some very common data descriptors (e.g. name, credit card number, address, etc.) in a hierarchical way (e.g. "business.contact-info.postal.street"), grouping them (Physical Contact Information, Unique Identifiers, Demographic and Socioeconomic Data, etc.) and defining a mechanism to allow extensibility. Furthermore, P3P is complemented by another W3C standard, "A P3P Preference Exchange Language" (APPEL) [9], a xml based language allowing a user express her preferences about privacy. Preferences are expressed in terms of a set of rules ("ruleset") the user can define. Basing on matching between the conditions expressed in the rule and a group of policies associated to the resource the user agent is going to request, a rule engine decides whether to "fire the rule" or not. If the rule is fired, the user agent follows a behaviour specified in the rule (basically, accessing the resource or blocking the access); optionally, the user agent may ask the user to learn more about her will.

Use of P3P/APPEL is intended mainly for interaction between user agents and web servers, and a number of products compliant with P3P specification both on client (Netscape, Internet Explorer to say but a few) and server side (IBM Tivoli) have already been provided. Also a some public Internet sites already support P3P, a list of them can be found here (http://www.w3.org/P3P/compliant_sites). In addition, there exist already organization like TRUSTe (http://www.truste.org) or BBB Online http://www.bbbonline.org) which check the compliance of online services with the privacy policies they declares. "Their Privacy Is Your Business", is the TRUSTe's motto.

However, the aforementioned approach opens two issues: the first one is the difficulty the user might experiment in formulating her preferences for a number of different real cases using the very detailed APPEL language. Of course, it is reasonable that the user might have a set of (customizable) predefined rules provided by a trusted 3rd party organization. An extreme solution

could be the aforementioned possibility proposed by the Liberty's approach, based on a fixed set of privacy practice, classified on different degree of restrictiveness.

The second issue is about the time it takes to access a resource following this interaction schema; though performances could be greatly improved by encoding P3P policies in a compact format (a sort of mnemonic codes replacing the verbose xml statements), the overhead due to verifying the compliance of policies to user's preferences may lead to bad performances, especially when an high granularity is expected, i.e. when there are a lot of small resources to be accessed with a single request having associated different policies to each of them.

Cobricks [10] a research work from the Technical University of Munich, solves this problem by defining a so called Access Ticket (AT), which is the result of the negotiation performed between the user agent and the Service Provider in order to agree on accessing user's data [11] .The AT is cached as a persistent data; It is digitally signed and contains information on the ticket issuer, the expiration date, the ticket owner (who can access data) and the access modes (read, write, etc.). User profile data are then accessed according the instruction provided in it.

The AT is a proprietary solution tailored for user profile data access, but it is very similar to other XML based access control approaches such as Oasis' XML Access Control Markup Language (XACML, [12]) which now includes also the former XACL (XML Access Control Language). Currently, XACML defines both a policy language and an access control decision request/response language. A semantic for concepts like "resource", "subject" (the actor allowed or not to access a resource), "action" (read only/ read & write…) and environments (e.g. the time the resource can be accessed) has been defined in XACML which appears therefore suitable also for controlling accesses to user profile data.

4. OASIS' Dataweb, XRI and XDI

Oasis' "eXtensible Resource Identifier" (XRI) and "XRI data Interchange" (XDI) aim to define standard mechanisms is to allow any electronically represented data be shared independent of the application or domain from which they have been originated. According to [13], this approach should pave the way toward the so called "Dataweb". The novelty of XDI, compared to similar approaches (like the aforementioned 3GPP GUP), is that this technology, in order to solve the problem of interoperability between different data formats and between different domains, does not mandate a particular data format to conform with. Each organization ("authority" in XDI terminology) may still keep its own data format; the interoperability is guaranteed by two complementary mechanisms: first, data (and metadata or data "types") are bound to one or more resource unique identifier (XRI) [14], which is an extension of the current Internationalized Resource Identifier IRI defined in RFC3987 [15]. This solution allows to associate a semantics to the legacy data model, without relaying on ontology models like OWL [16] (which, in turn, would represent just another data model!); second, XDI provides a distribute, secure and ubiquitous available mechanism for sharing data, very similar to what the Web actually does for presenting data to the navigators: the data-links. A data-link is similar to the well known hyperlinks; anyway it is intended to be used by software applications (instead of humans) for accessing data. Data-links introduces also access control mechanism, which is exactly what we need for our scenario. In the last section, we will explain how it is possible to manage access control to profile data using XDI data-links.

5. Architectural aspects

In this section, we present the proposed architecture for profile handling , shown in Figure 3. This proposal is derived from the one presented in [17]. The entities that manages profile information are called Profile Managers (PM). A PM is able to retrieve (and store) profile information about a user, on request of another entity which is generically denoted as Profile Requester Entities (PRE). PMs may reside both in the terminal (central Profile Manager, cPM) as well as in the network nodes (remote Profile Managers, cPMs).

In this architecture, we can identify two relevant interfaces: the PM-PM interface and the PRE-PM interface. Though these interfaces are defined using WSDL (Web Services Description Language), the implementation of the communication can rely on different solutions (e.g. SOAP, middleware like CORBA or JXTA, agent communication platforms like JADE). We are currently implementing the solution using the JXTA platform [18].

To access the profile information, the Profile Managers exploits the Data Access Manager (DAM). This element translates the requests coming from the PM into a form suitable for the native communication mechanism used by the data repository (e.g. the PID, a relational database, a distributed storage system).

If we look at the main architectural entities which play a role in managing, handling and storing the User

Profile we can notice that the components specified in the GUP architecture have a counterpart in our approach. In fact, the role played by the GUP server corresponds to the one of the Profile Managers. Furthermore, the GUP's RAFs are very similar to the Data Access Managers, as they both provide a common interface which hides the implementation details of different data repositories.

The difference between the two models lies in the way user's data are handled: unlike the GUP architecture, which use a single central entity in the network to manage user profile data, we use trusted distributed peer entities to accomplish the same task. This approach allows to reach a totally distributed handling and management of user profile information.

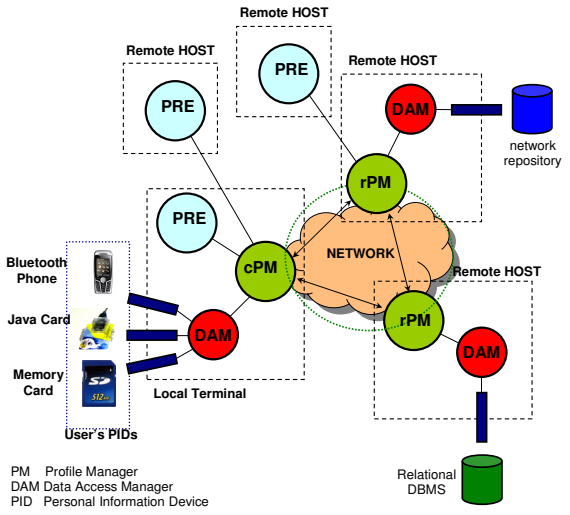

Figure 3: Proposed architecture for handling profile information.

6. Exploiting XDI links for integrating security and privacy features in user profiles

In order to illustrate a possible use of XDI, XRI, and XDI link contracts in user profile handling, let us consider the following example, which describe a handover scenario on a reconfigurable terminal.

Sam Hamilton is an employee of a large company, Acme co.; after a working day, he leaves straight by train for another city. As soon as he embarks on the train, he continues working with his terminal using his company groupware software to access the company intranet. Sam's device uses WLAN connectivity offered in the train station area. When the WLAN connectivity breaks as the train moves out of the station, Sam's reconfigurable terminal is able to discover an alternative radio access technology (RAT), let's say UMTS. This process is possible through the use of a Cognitive Pilot Channel (CPC) which transmits information on available RATs. This allows to free or at least relax the need for spectrum scanning in the user equipment, and can support the advertising of RATs that are previously unknown to the terminal. It also provides a way to allow resource reservation for the chosen RAT, according to the user preferences and policies defined both at network level (e.g. operator-defined) as well as at terminal level (user-defined or manufacturer-defined). If software radio functionality are realized, the CPC may also be used to download new radio modules enabling the user equipment to access new RATs. When Sam was under WLAN coverage, he was experiencing an excellent Quality of Service. However, after a seamless handover to UMTS, service adaptation took place to compensate for the QoS degradation…

In the aforementioned story, many aspects are related to user's identity, user profile, and data access control. We will try to analyze how these could be implemented using the XDI technology. First, it is important to provide a way to identify the user, as conventional mechanism used by operators (IMSI) are not suitable for non-cellular networks, and different RATs may use different identifier. The user therefore may be identified by an XRI, like

@Acme*SamHamilton

This tells that Sam Hamilton has a temporary data web account with the Acme corporation (similar identifiers may be defined also for private people). Note that this identifier is independent from a particular network access provider; However, this identifier may be linked to any other identifier used by any network access provider; for example a network operator which traditionally uses IMSI to identify its subscribers might maintain backward compatibility using a so called XRI cross reference [14], expressed like this:

@Acme*SamHamilton*(@MobileOp/+IMSI)

Second, Sam has his own user profile, which could be made of a number of components; as discussed in the previous section, Sam's profile may be for example an instance of 3GPP GUP and thus containing Sam's preferences and policies for reconfiguring his device when more than one RAT is detected; each instance of them can be addressed by one XRI, like:

@Acme*SamHamilton/+preference/pref3487

Physically, these data may reside on the terminal, on the Sam's Personal Identification Device or somewhere in a network repository. Again, the distributed nature of

XDI and the use of XDI links helps in maintain a logically unitary representation of Sam's profile information despite the native distributed nature of profile information. Note that this approach also stresses the idea that the user profile belongs to Sam and not to one single operator, as it is correct in an environment with heterogeneous networks available. A sample XDI/XRI graph modeling the relationships between user identity, network operator profile and service profile through XDI data links and XRI cross references is shown in Figure 4.

When the terminal connects to the UMTS RAT, the operator database is updated to reflect the change. This could be represented in a very simple way using XDI, by establishing a XDI link between the user identifier and the identifier of the base station which the user's terminal connects with. Furthermore, since the user identifier may be linked, in turn, to his profile data, this representation allows to answer a number of queries just by "navigating" the established links. Possible queries include: "How many users are actually using this base station?", "Which RAT/operator my reconfigurable device is connected to?", "What QoS is delivered to the user?", and so on.

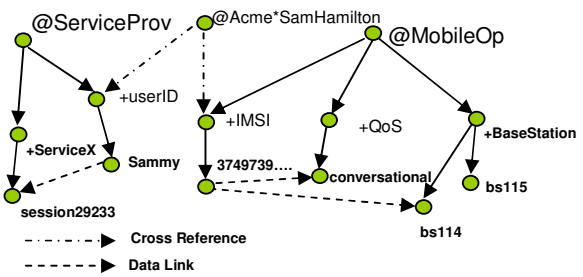

Figure 4: A sample XDI/XRI graph.

The aforementioned questions might be originated by three different motivations. It is likely that the first one ("How many users are actually using this base station?") is of interest for the network access provider; The second question "Which RAT/operator my reconfigurable device is connected to?" may be typically asked by the user or by some user agent acting on the terminal; finally, the question about QoS may be of interest for both the aforementioned subjects, as well as for third party, like value added service providers. Complementary, it is important to prevent that unauthorized subjects access these data. In order to provide a way to control data sharing through links, XDI defines a mechanism based on so called link contracts. Inside a link contract one could find two main items: a "Data Share Agreement" (DSA) and a list of resources which are available under the DSA. The DSA is a set of terms and conditions under which the resources described in the contracts could be accessed. Since the DSA doesn't mandate a particular format for them, any possibility discussed in section 3 could be potentially valid. The only requirement is that they could be pointed by an XRI. To be valid, a contract should be signed; a signed contract keeps also further information, like the involved parties, the signature, the expiration date and so on. The signature takes place using a couple of private/public key. Public key can be revoked and, since XDI allows to maintain a list of older version of the resource it is describing, a revocation list is implicitly and automatically built by the system. In XDI, cryptographic operations are performed by "I-Brokers", which in our specialized architecture for profile handling (section 5) are implemented by Profile Managers.

To go on with the example, let us suppose that a value added service provider is willing to know which QoS Sam Hamilton is actually experimented with his terminal, e.g. in order to perform seamless service adaptation. The service provider assumes the role of a PRE (Profile Requester Entity). Both the service provider and the network operator have established a trusted relationship with their respective profile managers (it is possible that they share the same profile manager, but more in general they will refer to different profile managers). The operator has defined a contract with a DSA to allow a PRE to access information about its subscribers under some conditions. The service provider identifies Sam through Sam's nickname, which is bound to Sam's XRI through a cross reference; therefore it is potentially able to access the wanted information just "navigating" the established data links, e.g. making a query to its profile manager. The distributed community of profile managers exchange messages between themselves to solve the links in the query and find the requested data. Anyway, at a certain step in the procedure, the DSA defined by the operator is found. Therefore, a warning message with the condition defined by the operator's DSA is returned to the service provider which is asked to sign a contract with the operator. After the provider signs the contract, the profile manager validates the signature and finally the wanted data are returned.

An initial XML schema proposed for XDI link contracts can be found in [19]

Acknowledgment

This work has been performed in the framework of the EU funded project E2RII. The authors would like to

acknowledge the contributions of their colleagues from the E2RII project.

7. References

[1] 3rd Generation Partnership Project. 3GPP Generic User Profile (GUP) 3GPP TS 29.240.

[2] http://www.ist-simplicity.org

[3] N. Blefari Melazzi, S. Salsano, G. Bartolomeo, F. Martire, E.Fischer, C. Meyer, C. Niedermeier, R. Seidl, E. Rukzio, E. Koutsoloukas, J. Papanis, I. S. Venieris: "The Simplicity System Architecture", 14th IST Mobile Summit, 19-23 June 2005, Dresden, Germany

[4] XML schemas for Simplicity User Profile: http://netgroup.uniroma2.it/SUP

[5] Project Liberty Alliance Web site : http://projectliberty.org

[6] "UAProf" URL: http://www.openmobilealliance.org/

[7] Lorrie Cranor, Rigo Wenning (ed.), et al., "The Platform for Privacy Preferences 1.1 (P3P1.1) Specification", W3C Working Draft 4 January 2005, http://www.w3.org/TR/P3P11/

[8] Platform for Privacy Preferences (P3P) Project, http://www.w3.org/P3P/

[9] Lorrie Cranor, Marc Langheinrich (ed.), et al., "A P3P Preference Exchange Language 1.0 (APPEL1.0)", W3C Working Draft 15 April 2002, http://www.w3.org/TR/P3P-preferences

[10] Cobricks Home page, http://www.cobricks.de/

[11] Michael Koch, Wolfgang Wörndl, "Community Support and Identity Management", Proc. European Conf. on Computer Supported Cooperative Work (ECSCW 2001), Bonn, Germany, Sept. 2001

[12] XACML home page, http://www.oasis-open.org/committees/xacml/

[13] Drummond Reed, Geoffrey Strongin, "The Dataweb: An Introduction to XDI, A White Paper for the OASIS XDI Technical Committee - v2", April 12, 2004

[14] OASIS consortium, "Extensible Resource Identifier (XRI) Syntax V2.0", Committee Specification, 14 November 2005

[15] M. Duerst, M. Suignard, "Internationalized Resource Identifiers (IRIs)", IETF RFC 3987

[16] OWL Web Ontology Language, W3C Recommendation, http://www.w3.org/TR/owl-features/

[17] G. Bartolomeo, N. Blefari Melazzi, F. Martire , S. Salsano, "Defining and Using Profiles to Personalize and Manage Reconfigurable Services", 15th IST Mobile&Wireless Communications Summit 2006, June 4-8 2006, Myconos, Greece

[18] The JXTA home site, www.jxta.org

[19] http://www.oasis-open.org/committees/download.php/15292/Link Contracts.doc