User-centric Identity Management in Ambient Environments

Hasan Ibne Akram
Fraunhofer Institute for Secure Information Technology
Munich, Germany
hasan.akram@sit.fraunhofer.de

Mario Hoffmann
Fraunhofer Institute for Secure Information Technology
Munich, Germany
mario.hoffmann@sit.fraunhofer.de

Abstract– **Context-aware intelligent systems in ambient environments will have major impact in the near future to the way people will perceive and deal with computer technologies regarding privacy, security and trust. In those environments it will be all about personalized information and digital identities – so the foremost goal we are heading for in our research is: How to avoid *omni-persistence* in a world of *omni-presence*?**

Firstly, we show in this paper how any kind of personalized information, such as identities, preferences and profiles, will fuel those systems to support, serve and simplify people's lives. Secondly, we are convinced that especially privacy and so-called informational self-determination are at stake if protection goals like confidentiality, transparency, and minimal disclose of information are not well balanced and precisely taken into account when realizing such systems.

Existing standards, solutions and technologies in Identity Management are specifically tailored for example for the Internet, company processes or eGovernment. However, for future ambient environments they have to be improved and revised to meet also user-centric requirements. This paper combines certain aspects of existing approaches to introduce a new middleware architecture that supports user-centric Identity Management. We further show that this middleware enables future application developers to meet (almost) all of our postulated ten laws of identity.

Keywords - Identity Management, Ambient Environments, Privacy by Design, Identity Metasystem, Higgins

I. INTRODUCTION

Inhabitants of ambient environments are envisioned to be surrounded by smart devices that are working continuously to make their lives more comfortable. This is achieved by context-aware intelligent systems where virtual networks (converging fixed, wireless and mobile) consist of numerous nodes, smart devices, sensors and actuators. The underlying system is transparent but omni-present to the users and the users are omni-present to the system. The basic research area is known as Pervasive and Ubiquitous Computing (a synonymous term widely used in Europe is Ambient Intelligence).

The basic idea is: The more information about the inhabitants of such environments is fed into the context aware systems working in the background, the better or more personalized it works for them. At the same time, however, it has to be ensured that the inhabitants' privacy is not endangered in such smart environments. The fact that depending on the application area context aware systems in principle are able to store and aggregate whatever information about individuals, groups and communities has to be taken into account seriously. Omni-presence shall not lead to omni-persistence. The most important questions are

- 'What information is stored, aggregated and mined?',
- 'Who is authorized to get access to such information?', and
- 'How long will the information being stored?'.

Thus, privacy and context awareness in smart environments, although being rather contradictory issues, have to be put in practice in a balanced manner. Therefore, in this paper an inherently secured user-centric Identity Management framework is proposed that deals with the *complete life cycle of identities* of users, services, and devices as well as users' awareness in information disclosure and privacy.

This paper elaborats step by step an architecture for an Identity Management Solution for such scenarios. Firstly a typical scenario for ambient environment is shown followed by a brief description of the ten laws of identity for ambient environments which have been previously discussed in [12]. A study on the state of the art technologies and an evaluation based on the ten laws of identity is presented in the following. Finally, based on the evaluation we propose an architecture for Identity Management in ambient environments that is compliant to (almost) all the ten laws of identity.

II. IDENTITY IN AMBIENT SCENARIOS

In the literature many kinds of future application scenarios which may benefit from the support of context-aware smart environments have been introduced already. Examples are intelligent buildings, automotive, and healthcare. In order to illustrate the most typical user-centric requirements we will, therefore, focus on a typical test scenario taken from an EU project for ambient environments called Hydra[1] [6, 7] (the authors are part of the consortium).

[1] Hydra: Networked embedded system middleware for heterogeneous physical devices in a distributed architecture. http://www.hydramiddleware.eu (2007) contract number: IST-2005-034891, duration: 07/2006-06/2010.

In the second section of this chapter we will then summarize the ten laws of identity that we have defined in our previous paper [12]. There you can find a detailed description and analysis with respect to the scenario sketched below. The ten laws of identity then serve as the basis of the architecture discussion and evaluation in Chapter V.

A Scenario Definition

In Hydra, fictitious scenarios have been derived in three domains: Building automation, healthcare, and agriculture, which are likely to be practiced in reality in 2015 [7, 8]. Many of these scenarios are derived from business cases from the perspective of an end-user; i.e., from application level. As a consequence, Identity Management can have a large range of implications to information systems encompassing role-based access control, *Single Sign On (SSO)* in single and cross organizational domains, as well as management of virtual identities, identity life cycles and sessions. However, in case of designing a middleware for Identity Management the perspective of requirements analysis shifts from the end-user to a developer. The question is, thus, which requirements coming from application domains can and should be addressed in a middleware?

With the intention to illustrate the necessity of an Identity Management System in a middleware for developing ambient applications we will take as a basis a detailed technical scenario of a heating system breakdown at “Krøyers Plads” housing complex located in Copenhagen that deploys the “Hydra Building Automation System” (HBAS) [7]. The resident living in a new flat in this building complex is equipped with automated lamps, computers and a wireless network, as well as a Hydra-enabled heating system and many other usual sets of integrated embedded devices. While the resident is at his office, the heating system of the flat breaks down and the water pressure rapidly decreases down to a level that is detected as an emergency situation by the HBAS which is shown as legend 1 in Figure 1. As a result of that HBAS sends out an alert message to the resident (legend 2 in Figure 1).

In order to get the heating system fixed as soon as possible the resident chooses a service provider from a list of providers matching the emergency requirements and his preferences best. The service provider then sends a service agent (e.g., a specialized technician) to the house. The major challenge here is to allow remotely a particularly authorized service provider and his technician to get into the house to fulfill a specific task. Therefore, included in the repair order a dedicated and restricted HBAS authorization ticket guarantees that in this case a service agent can enter the flat and get access to the heating system (legend 3 and 4 in Figure 1). After entering the flat upon successful authentication procedure the service agent gets authorization to access additional context aware information required to perform his job (legend 4 in Figure 1).

This representative scenario can be basically adopted by many kinds of similar scenarios of remote authorization such as large housing areas with housekeeping service, office buildings with restricted access, airports, and hospitals. Thus, with the basic scenario of Hydra being illustrated we can go one step forward in our process of our identity requirements analysis in ambient environments.

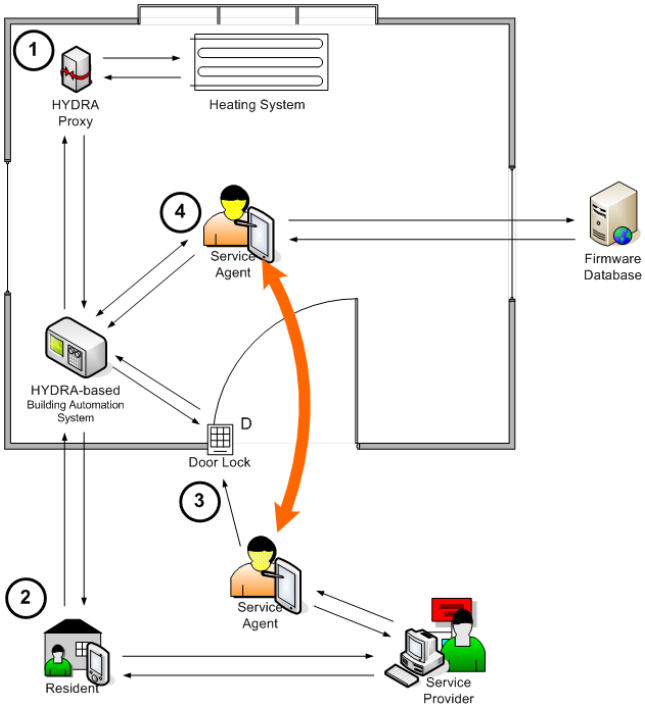

Figure 1: Sequence of steps for the technical scenario [7]

In our previous paper [12] we showed an extended use case analysis of the given scenario, applied the principles of *Federated Identity* in the process of use case analysis and derived ten laws of identity for ambient environments. The details of the use case analysis process can be found in the referenced paper. In the next subsection we will briefly describe the ten laws of identity and their implications in ambient environments.

B Ten Laws of Identity

Identity Management in ambient environments, characterized by pervasive and ubiquitous computing, has been explored by researchers intensively during the last decade. Requirements and principles of Identity Management have been analyzed and derived based on certain needs in certain scenarios. A prominent example is the “Laws of Identity” by Kim Cameron tailored for the Internet [1].

Obviously, these related works have some commonalities and disparities among themselves. This is

simply because all these laws and requirements are based on some variable parameters; namely - perspective, time, computing environment etc. Therefore, there is a need of customized adoption and modification of the existing laws to certain scenarios. In this section we postulate the following ten laws of identity which are meant for ambient scenarios as shown in Section II.A:

1. User Empowerment: Awareness and Control
2. Minimal information disclosure for a constraint use
3. Non-repudiation
4. Support of directional identity topologies
5. Universal Identity Bus
6. Provision of defining strength of identity
7. Decoupling Identity Management layer from application layer
8. Usability issue concerning identity selection and disclosure
9. Consistent experience across contexts
10. Scalability

1) User Empowerment: Awareness and Control

Our first law looks similar to Kim Cameron's first law of identity where it says *"User Control and Consent"* [1]. We do totally agree that user consent and control are necessities in Identity Management but at the same time we believe that the word "consent" does not imply a total *empowerment* of the user. According to the definition of the word "consent" provided in American Heritage Dictionary[2], is - "To give assent, as to the proposal of another; agree." This merely implies an agreement and nothing beyond an agreement; i.e., it does not imply that the user being fully aware of the consequences of the agreement. The following example of one of today's extremely popular Web 2.0 applications examines why a mere agreement of the user is not enough. The "Contact importer" feature of facebook.com has been a very much well-liked feature and it has been very trendy in many other web 2.0 applications.

Figure 2 shows a screen shot of facebook's contact importer feature. Using this feature a user is able to import the user's buddy list from his other email or instant messenger accounts like Google, GMX, MSN, Yahoo, AOL, and many others. What the user has to do here is to provide his username and password credential to facebook and facebook uses that credential to import the buddy list from the corresponding provider. This allows facebook to have access to all the other accounts of the user and even if we consider facebook as a basically trusted party, privacy of the user has been completely compromised.

In this example the username and password have not been stolen without the user's consent, i.e., the user had agreed to giving his username and password and clicked the "Find Friends" button. The question to ask would be if the user is *aware* of the fact that his privacy has been compromised. Therefore, instead of "consent" the first law of identity takes the word "awareness" which subsumes "consent" anyway.

From the perspective of our scenario (Section II.A) the first identity requirement concerns the user in an ambient environment and emphasizes on two key words – "awareness" and "control". In a transaction taking place between two entities in such ubiquitous scenarios, each entity must have full knowledge regarding the information he is about to disclose and to whom he is about to disclose. Besides having full knowledge about the information disclosure the entities must also have full range of control power to decide whether to disclose a particular set of information or not [1] as well as the power to continuously check the authenticity of this information and even change or delete it.

2) Minimal Information Disclosure for a Constrained Use

Whereas the first law has addressed awareness and control, the second law addresses information disclosure. Basically these two laws are complimentary to each other. In a ubiquitous scenario there can be numerous possible ways information can be leaked out without the user being aware of the information disclosure. Therefore, the system must ensure that claims must be satisfied with a minimum set of information required. The support of zero-knowledge-proofs for example is favored over disclosing a credential.

From the perspective of our building automation scenario (Section II.A) the second law of identity means the following: We have already stated that there is a contractual relationship between the resident and the service provider. Therefore, authentication information propagates in a transitive fashion to the service agent; i.e., since the agent is authenticated by the service provider, he is also authenticated by the resident and depending on the security policy all or parts of the smart devices in his apartment. In the process of fixing the heating system, the service agent will need to have access to certain information, e.g., the usage pattern of the heating system. Here the service agent must be provided with a minimal information set that is only relevant for fixing the heating system. The usage pattern of the heating system supplied by the smart devices to the service agent must somehow guarantee that no other information is retrievable from it that goes beyond the necessity of fixing the heating system, e.g., the service agent should not be able to figure out from the usage pattern that during which period of the year the resident is on holiday or remains out of the flat.

[2] Consent. (n.d.). *The American Heritage® Dictionary of the English Language, Fourth Edition*. Retrieved July 03, 2008, from Dictionary.com website: http://dictionary.reference.com/browse/Consent

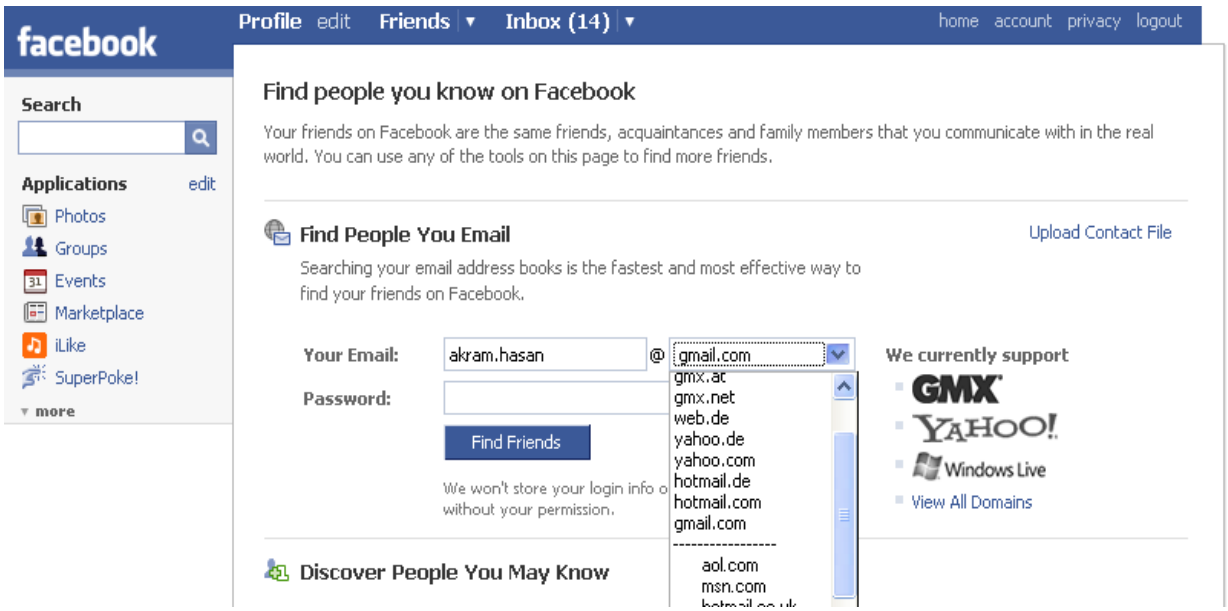

Figure 2: A screenshot of facebook's contact importer feature (screen shot taken on 5th September 2008).

3) Non-repudiation

The term "Non-repudiation" has a traditional legal meaning and at the same time, a different meaning in terms of digital security [19]. We will focus on the latter meaning of "Non-repudiation" and then relate its necessity to our scenario (Section II.A). In a crypto-technical sense transfer of data from one entity to another must guarantee authenticity, integrity, and a time stamp, so that neither of the parties involved can deny that the transfer of the data took place.

Within the scope of the building automation scenario (Section II.A) the issue of authenticity takes place in the following process: The endpoint of the service provider receiving a message from the endpoint of the resident must know whether the message is really transmitted from the resident or if it is under a spoofing or masquerade attack [4]. Therefore, there is a need of mechanism(s) that guarantees identity preservation.

In order to illustrate integrity, we continue with our running scenario example: The service provider receives a message from the resident over HTTP, he must guarantee the integrity of the message content. From a middleware viewpoint, there must be supports that allow the developer ensuring that the messages sent from one node to another is not being modified or misused in an intermediary node or is not under falsification attack [4]. In order to guarantee integrity it is also important that any kind of message manipulation has to be detected.

Another vital point is to ensure that a time stamp is attached to the message. This is required to combat replay attacks. A time stamp attached to the message will make the message valid only for a certain period of time and as a result of that will lower the probability of replay attacks.

Summarizing, unforgeable identity, non-falsifiable message exchange, and provision of a time stamp are required in middleware for such scenario so that the identity of the sender and the integrity of the message cannot subsequently be refuted.

4) Support of directional identity topologies

Kim Cameron identified identity as a vector rather than a scalar in his paper [1], i.e., identity not only has magnitude but also a direction. In his fourth law of identity he expressed the need of omni-directional and unidirectional identities. We have adopted this law in the context of pervasive computing and have modified it according to ambient environments' needs.

In the domain of ubiquitous computing, communication takes place in various topologies and so does *identity federation*. Identity federation in such scenarios can be unidirectional, bi-directional or even omni-directional. An unidirectional federation involves an Identity Provider (IdP) issuing a Security Token for a user when a particular Relying Party (RP), e.g., a service provider, the user wants to get access to, is asking for it. Bi-directional federation takes one step further, where the RP is able to act as an IdP once the user is federated to the RP by an IdP within the circle of trust. This is how authentication information is being propagated node to node [12] in our home automation scenario (Section II.A). Finally, an omni-directional identity refers to a virtual identity emitted to any entity that shows up. An example with respect to our scenario would be, the presence of the service agent is being sensed by the intelligent devices at

the resident's apartment when the service agent transmits his identity in omni-directional manner.

The fourth law of identity states that the following identity federation topologies must be supported in an Identity Management System in ambient environment:

1. Broadcast (omni-directional)
2. Point to point (unidirectional or bi-directional)
3. Multicast (omni-directional and/or bi-directional).

5) Universal Identity Bus

In today's Internet users have multiple virtual personas for one identity and each of these multiple personas has different contexts, purposes and flavours. In the world of *Internet of Things* it can well be imagined that these multiple personas would require being portable from domain to domain, device to device or context to context. No portability of identity will create *Identity Silos* and cross domain interoperability or even inter domain interoperability across platforms or devices will be challenged.

The middleware for an ambient environment Identity Management System inherently requires supporting interoperability between the garden varieties of Identity Management technologies available from different vendors. The fifth law of identity for ambient environments states the necessity of a Universal Identity Bus (UIB) that will provide vendor to vendor interoperability functionalities. In order to achieve this requirement the middleware must support UIB that works as a bridge between different Identity Management technologies.

6) Provision of defining strength of Identity

In order to illustrate why such ambient environments necessitate the provision of the strength of identities two aspects have to be taken into consideration: identity propagation and the dependency of the identity.

Identity Propagation: In a federated environment identity can be lightweight or rather strong depending on policies of the IdP and the RP. Especially, when it comes to bi-directional federation (see law 4) it is important to categorize identity according to its strength. In such federation RP is gaining the power to be the IdP once an entity is authenticated to it by the original IdP and the propagation of identity can continue creating a very long chain in ubiquitous computing which may result in an apocryphal identity. Thus, there is a need of accumulated calculation of its source of identity reliability.

Dependency of the Identity: In a ubiquitous world virtual identities might refer to individuals, devices or services, i.e., more general entities or things. If a device is owned by a person, for example, the identity of the device is somewhat depending on the identity of the person, i.e., the identity of the device is incomplete without relating it to an identity of another entity. In a similar way many use cases may arrive where an identity does not suffice itself without being depending on an identity of another entity. Based on this criteria identity can be categorized to be strong (independent), weak (dependent) or somewhere in the middle. Thus, we can justify the requirement of a provision of having the strength of an identity in the middleware. It is important to note that weak identities and strong identities are not the same as sub-identities, which are basically subsets of identities. Identities or sub-identities both can be rated by their strength depending on their degree of being autonomous.

7) Decoupling Identity Management layer from application layer

This requirement builds up another block on top of the "*Universal Identity Bus*" and separates the application layer from the Identity Management Layer. This is obligatory for the our Identity Manager for two main reasons: 1) organizations are being able to change their identity policies without having an impact on the business layer and 2) the developers have an environment where they can work on the identity layer being transparent of the business layer or vice versa.

8) Usability issue concerning identity selection and disclosure

"A potato peeler is easier to use for peeling potatoes than a knife is, but a lot harder to use for murder." – Ross Anderson [15]

The above quotation figuratively expresses the fact that usability is case specific. High usability of a tool in a certain area can be extremely inconvenient for other purposes. Therefore, appropriate design support of usability for identity selection and discloser is unavoidably important in a middleware.

We have already emphasized the issue of empowerment of the user in case of revealing information in our first identity requirement. Lack of usability will make law 1 (User Empowerment: Awareness and Control) almost impossible to take place. In a user-centric design the user is the ultimate procurer and a methodic requirement specification of usability keeping the procurer in mind is unavoidable [10]. Therefore, our middleware architecture must facilitate the developer with adequate support for implementing usability.

9) Consistent experience across contexts

Context is one of the major concerns in our test scenario (Section II.A) and identity and context are closely related. Therefore, while analyzing requirements of Identity Management in ambient scenarios, the issue of context is considered. In ambient environments an entity and its identity will have an *n* to *m* relationship, i.e. one entity (e.g., a user, device or service) can have multiple identities and one identity can be possessed by several entities. For example, the resident has several identical sets of devices, e.g., temperature or movement sensors,

and he wants to use them with one single device identity. In this example one identity is shared by multiple entities. The example one entity having multiple identities would be, the resident has an identity at his work, a different one for his shopping web sites and another different one for heating system repairing service providers. So identities may change in different contexts based on different roles. In this *n:m* relationship of identities and entities it is very important to have consistence experience for the user depending on contexts.

Along with the consistencies among context, the identities provided in different contexts should also be independent of each other, i.e., the identity the user provides at work should not be related to his identity for his shopping website and vice versa. This is in order to avoid aggregation and concatenation of partial identities following the principle of privacy by design.

10) Scalability

Identity multiplies with time. For the inhabitants of ambient environments a growing number of identities across contexts must be managed properly and at the same time there has to be room for conceiving new identities. Moreover, in an ambient environment the number of nodes joining in and out is dynamic and thus, the capability of an identity to interacting with the identities of the other numerous nodes is necessary. Therefore, scalability of identity refers to an entity that must be able to spawn new identities and a single identity must have the capability to communicate with a growing number of identities.

III. ARCHITECTURAL IMPLICATIONS

Having the laws of identity being illustrated in the previous section we will get back to our home automation scenario (Section II.A) in order to motivate our architectural approach and to analyse the implications. In this section we will see use cases in the home automation scenario (Section II.A) where the propagation of authentication information from entity to entity is based on contractual relationships. We will also observe how the three roles of *Identity Federation – Subject*, IdP and RP – shift from endpoint to endpoint.

A Use case analysis

The first identity federation use case in our scenario is shown in Figure 3. In step 1 the resident is sending a request to the service provider for a service agent to be sent to his flat to fix his heating system. In step 2 the service provider is asking for his credential as a set of claims. Here the resident has an option to choose an IdP that can satisfy the claims from the RP that happens to be the service provider in this case. For simplicity we assume that the resident himself is able to issue an identity token that satisfies the claims and would also be accepted by the RP. So, in step 3 the resident issues himself a token and in step 4 releases it to the service provider. After receiving this token the service provider issues a co-signed token to a service agent (step 5) who is to be sent to the resident's flat for repairing the heating system.

Another use case scenario is shown in Figure 4. Here, the service agent has to authenticate himself at the door lock of the resident's apartment. The roles - *Subject*, RP, and IdP – have been shifted to the service agent, the door lock, and the service provider correspondingly. In step 1 the service agent sends a request to the door lock for accessing the appartment. In step 2 the door lock sends a request for a security token as a set of claims. The service agent requests his IdP (the service provider in this case) for a security token satisfying the claims. The service provider issues a token in step 4 and in step 5 the service agent releases this token to the door lock.

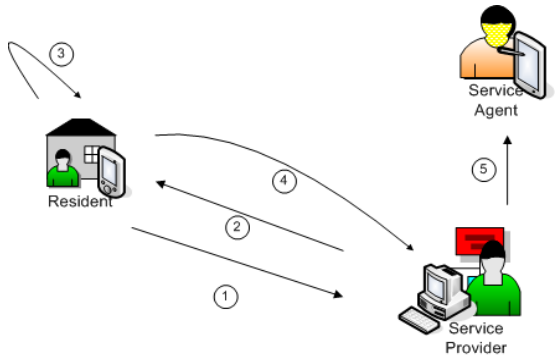

Figure 3: Sequences in the process of the resident authenticating himself to the service provider and the service provider issues a cosigned token to the service agent.

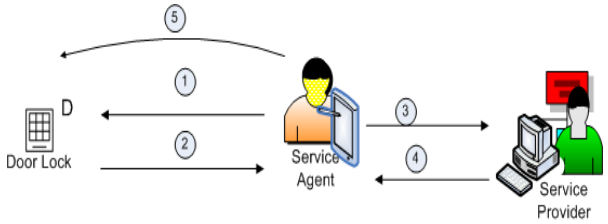

Figure 4: Sequences in the process of the service agent getting authenticated by the door lock of the resident.

Due to transitivity of authentication information flow as a part of the contract between the resident and the service provider, the door lock accepts his request; i.e., the door lock accepts the authentication assertion (in form of a security token) from the resident, the resident sends the token to the service provider and the service provider issues a co-signed token to the service agent, consequently the door lock accepts the authentication information of the service agent. This process is repeated in each identity discovery taking place in the scenario.

B Distributed Identity Provider

The use case analysis clearly motivates us toward the concept of *Distributed Identity Provider*. When bi-directional federation takes place in a way that every

endpoint in the circle of trust[3] (or contractual relationship) can attain the role of IdP and RP the set of the endpoints collectively can be defined as *Distributed Identity Provider*. In Section V we will elaborate in details where the architecture is illustrated. The following section is dedicated to state of the art analysis that evaluates the suitability of the existing IdM technologies to fabricate Identity Management architecture for ambient environments.

IV. STATE OF THE ART

This section will accomplish a state of the art study of standards, frameworks, protocols, and products related to Identity Management. For these purposes we apply our ten laws of identity illustrated in Section II.A in order to find out the closest technology which complies with the corresponding requirements. Therefore, we start with a subsumption of basic enabling standards from the WS-* family and dedicate the following sections to state of the art technologies, namely SAML, OpenID, Windows CardSpace, Higgins, and Liberty Alliance.

A Web Service Related Standards

Identity as a service is a visionary goal of the Service Oriented Architecture (SOA) proponents. The adoption of the spirit of identity as services in futuristic ambient computing is also being pushed by the researchers and scientists. Since Web Services is considered as being a key driving technology for enabling SOA we will briefly highlight some Web Service related standards that are as well relevant for an Identity Management ecosystem.

B WS-Security

The main objective of WS-Security is to secure the Web Service message itself. The SOAP message is secured in order to guarantee authenticity, integrity, and confidentiality of the message. Moreover, it also provides a time stamp for SOAP messages [4].

WS-Security is relevant in our IdM architecture because this standard provides support for our third law of identity (Non-repudiation). Thus, WS-Security is considered as a candidate for being one of the building blocks of the architecture.

C WS-Trust

WS-Trust defines a framework that provides protocol agnostic ways to issue, renew, and validate security tokens. Moreover, it defines ways to establish, assess the presence of and broker trust relationships. The main goal of WS-Trust is enabling applications to construct trusted SOAP message exchanges [11].

The reason WS-Trust is interesting or relevant in our ambient scenario (Section II.A) is, in the home automation scenario, we have seen that the contractual relationship between the service provider and the resident needs to be somehow technologically represented. WS-trust exactly addresses this issue. Moreover, being agnostic to security tokens, the support of WS-Trust in the architecture enables the developer to take advantage of any kind of associated protocol.

D WS-Policy

WS-Policy is a language for representing capabilities and requirements of a Web Service. In other words, it tells the consumer of a Web Service what the requirements are that it must fulfill in order to consume that service. These requirements can also be optional in some cases, which would provide certain advantages to the client if it can fulfill those optional requirements [9]. WS-Policy provides a precise way to write policy expressions for a certain Web Service.

Getting back to law seven (Decoupling Identity Management layer from application layer) we can clearly relate WS-Policy to our requirements. In order to achieve such goal, changes in identity policies should not affect the business policies or vice versa. WS-Policy offers functionalities to facilitate such mechanisms.

E WS-Security Policy

WS-Security Policy language is built on top of the WS–Policy framework and defines a set of policy assertions that can be used in defining individual security requirements or constraints. The motivation for adopting WS-Security Policy in our architecture is the same as for adopting WS-Policy.

F WS-Federation

WS-Security, WS-Trust, and WS-Policy/WS-SecurityPolicy described in the previous sections provide a basic model for federation between IdP and RP. WS-Federation uses these building blocks to define additional federation mechanisms that extend these specifications and leverage other WS-* specifications [21].

WS-Federation allows security realms to broker identities, user attributes and authentication between Web services. This is an essential factor for engineering a *Distributed Identity Provider* architecture.

G WS-MetadataExchange

Web Services use Metadata to describe what other endpoints need to know to interact with them. For example, WS-Policy describes the capabilities, requirements, and general characteristics of Web Services; WSDL describes abstract message operations, concrete network protocols, and endpoint addresses used by Web Services; XML Schema describes the structure

[3] According to the definition stated in OASIS standard: *Web Services Security: SOAP Message Security 1.0* [11] - "Trust is the characteristic that one entity is willing to rely upon a second entity to execute a set of actions and/or to make a set of assertions about a set of subjects and/or scopes."

and contents of XML-based messages received and sent by Web Services [20].

In order to bootstrap communication with a Web Service, this specification defines how an endpoint can request the various types of Metadata it may need to effectively communicate with the Web Service.

H IdM Protocols & Technologies

1) OpenID

OpenID 1.0 was originally developed in 2005 by Brad Fitzpatrick, Chief Architect of Six Apart, Ltd. It is now set up by a wide range of websites, especially which have heavy user-generated contents. OpenID Authentication 2.0 [13] is now turning into an open community-driven platform that permits and motivates federated identity. And the community is on its way for preparing drafts of a fully backward-compatible OpenID Authentication 2.0 specification which is a data transfer protocol to support both push as well as pull use cases. Besides, the community is coming up with extensions to support the exchange of rich profile data and user-to-user messaging [3].

According to an article published in German online computer magazine "Heise Online[4]" on 18th January 2008 there exist already 370 million OpenIDs globally. However, the number of really *active* OpenID users is still unknown. Big companies like Yahoo, AOL offered an OpenID to all their users and as a result, the number of existing OpenID naturally jumped up to such a high number.

There are three key features of OpenID: Single Sign On, decentralized, and light weight identity.

Vulnerabilities of OpenID:

Firstly, OpenID also allows the RP to redirect the client to the IdP for authentication at the IdP site [22, 13]. Therefore, it raises the probability of phishing. The user has no control over choosing his *Identity Provider* and therefore the first law (User Empowerment: Awareness and Control) of identity is violated. The second problem with OpenID is that the URL that is used to identify the *Subject*, is recyclable. Since OpenID permits URL based identification, it brings the issue of privacy. The privacy of the user using an URL as his OpenID would be compromised if somehow lost the possession of that URL.

2) SAML

The most precise and shortest way of defining SAML, presented by Eve Maler, is: *"The Security Assertion Mark-up Language in six words:* ***The universal solvent of identity information."*** SAML comes with the spirit of portable identity.

SAML (Security Assertion Markup Language) is developed by the OASIS Security Services Technical Committee with an objective of conveying security information across cross-organizational boundaries. There are three official versions of SAML – SAML 1.0 was the first official version coming out in November 2002, it was followed by SAML 1.1 in September 2003 and the latest version: SAML 2.0 has come out in March 2005 [24, 25].

Vulnerabilities of SAML:

SAML can be configured in a very lightweight (less secured) identity way and at the same time it can be configured in a much secured manner. In SAML an assertion is a set of security information that is requested by a *RP* about a particular *Subject* or entity. IdPs transport assertions to RPs who allow the requests. In the Google Single Sign On (SSO) implementation, the authentication response did not include the identifier of the authentication request or the identity of the recipient [23]. This may allow malicious RPs to impersonate a user at other RPs.

3) Liberty Alliance Project

Liberty Alliance started its expedition in 2001 with the purpose to be the service provider of the open standards organization for federated Identity Management. Guaranteeing interoperability, supporting privacy, promoting adoption for its specifications, providing guidelines and best practices Liberty Alliance has the objectives to enable users to protect their privacy and identity, to enable SPs to manage their clients lists, to provide an open federated SSO, and to architect a network identity infrastructure that is compatible with all emerging network access devices [17].

Vulnerabilities of Liberty:

Liberty Alliance technology stream is mainly based on SAML 2.0 and therefore inherently it suffers from the similar vulnerabilities as SAML stated in the previous section.

4) Windows CardSpace

Windows CardSpace is a visual metaphor for identity selectors for the end-user. Windows CardSpace provides controlling power to the end-users on the fact which information (about the end-users) should reach to the *Relying Party* and which should not. Windows CardSpace is a production of Microsoft shipped with Windows Vista (or as an add-on in Windows XP); it is not meant to replace the other standards handling digital identity rather to utilize and extend them [2]. Windows CardSpace is token agnostic.

The limitation and criticism of CardSpace is – although it does support virtually any security token format, it is not protocol agnostic. Currently it is only compatible with the WS-* Web Services protocols, which center on WS-Trust. For the reason that it is token

[4] http://www.heise.de/security/Yahoo-will-das-Passwort-Chaos-beenden--/news/meldung/102001

agnostic, but tied to WS-* protocols, we can say that it only partially complies with the fifth law which postulates the need for protocol agnostic as well as token agnostic (Universal Identity Bus).

Vulnerabilities of Windows CardSpace:

On top of its limitations CardSpace has some flaws: Firstly it relies on the users' judgements on the trustworthiness of Relying parties (RPs). A CardSpace user is given the freedom to choose one of the options of high-assurance certificates belonging to the RP, ordinary certificates belonging to the RP or RP with no certificates [14]. In terms of the first law (User Empowerment) this certainly gives a lot of power to the user. At the same time the option of allowing RP with no certificates weakens the compliance with the third law (non-repudiation).

The second vulnerability is, Windows CardSpace relies on a single layer of authentication. The user has to be authenticated to the IdP using traditional authentication mechanisms. If a working session is somehow hijacked or the password is cracked, the security of the whole system is compromised. This has been practically shown by two IT-Security students at Horst Görtz Institute for IT Security (HGI), Bochum, Germany, where they manipulated the DSN server to implement a dynamic pharming attack [18].

5) Higgins

Higgins is a software infrastructure that supports consistence user experience that works with digital identity protocols, e.g., WS-Trust, OpenID, SAML, XDI, LDAP etc. The main objectives of the Higgins project are the management of multiple contexts, interoperability, and the definition of common interfaces for an identity system. Various technologies including LDAP, SAML, WS-*, OpenID etc. can be plugged into the Higgins framework.

The first version, Higgins 1.0 was released in February 2008. The next version, Higgins 1.1 is supposed to be released by June 2009. There are also ideas and concepts in discussion beyond Higgins 1.1.

The architecture of Higgins 1.0 is based on:

- An Identity Attribute Service (IdAS): It provides a virtualized, unified view and a common means of access to identity information from multiple heterogeneous data sources. Simultaneously supports multiple Context Providers to abstract identity information from LDAP, SAML, OpenID, InfoCard, RDF.
- An infocard provider and Security Token Service (STS): It uses IdAS in a way that identity information comes from multiple Identity Providers.
- Multiple forms of Identity Agents: Web-based and client-side card managers are supported as well as browser extensions, and user interfaces (InfoCard selectors).

In Higgins 1.1 it is expected that an enhanced InfoCard with additional features will be supported. Among the enhanced InfoCards two very promising ones are *z-cards* and *r-cards*. The z-card adds functionalities to the managed card (*m-card*). It offers more privacy by caching the security token locally, and it supports subsets of claims. It also supports zero-knowledge proofs, thus enhancing privacy and trust features. An r-card is an enhanced version of managed cards (m-cards) and personal cards (*p-cards*). It sets up a data synchronization relationship between the user and the *Relying Party*. A change at either side updates the other.

Information cards created in CardSpace can be used in Higgins but the z-cards and r-cards created in Higgins are not currently supported in CardSpace. Both systems are in their early stages, and changes in compatibility are expected as this high-level identity architecture catches on.

The ideas and concepts in discussion beyond Higgins 1.1 are targeting the mobile platform which may be named as "Mobile Higgins". The target platforms are Symbian, RIM, Windows, Mobile 6, iPhone, Android etc.

Vulnerabilities of Higgins:

Since Higgins supports various IdM protocols and technologies it inherently takes over the flaws and vulnerabilities of those technologies and protocols. It also does not provide supports for quantitative measure of the identity's strength and lacks, thus, the fulfillment of the sixth law of identity (provision of defining strength of identity). However, the combined approach to provide an umbrella framework for IdM allows Higgins users to choose the best combination of technologies suited to their requirements. Moreover, Higgins architecture is most compliant to the laws of identity (Section II.A) among the state of the art technologies that have been considered in this evaluation. Therefore, in our architecture we have taken some aspects of the Higgins architectural approach and integrated them to our need. In the next section the architecture is illustrated in details.

V. PROPOSED ARCHITECTURE

The Identity Management Module described in this chapter is supposed to be integrated in the Hydra middleware [8] which is a middleware for heterogeneous physical devices in a distributed architecture in ambient environments; the module is named Hydra Identity Manager (HIM).

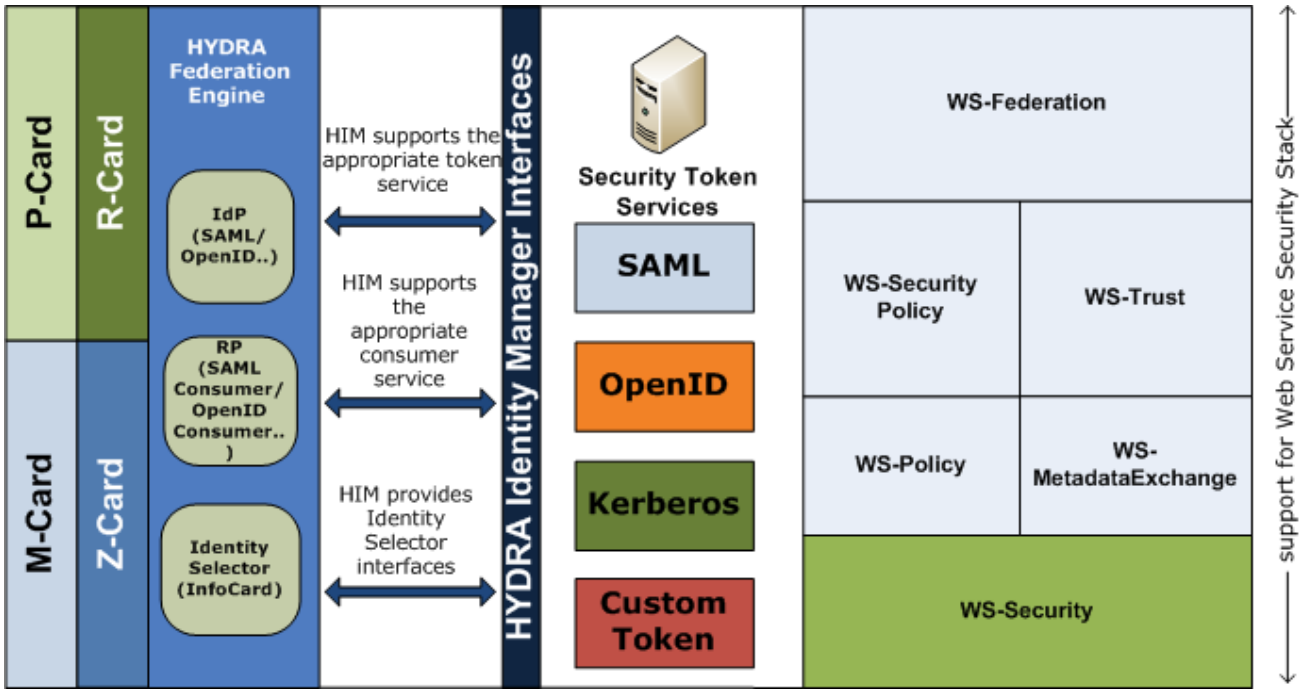

Figure 5: Anatomic view of the proposed architecture of the Hydra Identity Manager.

The architecture of HIM illustrated in our previous paper [16] needed some modifications and improvements due to several reasons. Firstly, the architecture used Windows Communication Foundation (WCF) [5] to take advantage of the out-of-the-box support for WS-Security stack and thus was bound to specific technology. Therefore, the first argument for reengineering the architecture is to redesign it to make it technology agnostic. Secondly, the result of the state of the art evaluation shows that Higgins provides better supports for our law 2 (Minimal Information Disclosure for a Constrained Use) and law 9 (Consistent experience across contexts) with their r-card and z-card concepts. In order to provide the best possible support for the ten laws of identity, there was also the necessity to adopt some further aspects of the Higgins concept into our architecture.

In a service oriented architecture, Hydra's Identity Management System provides support to the developer to implement integrity, confidentiality, and authenticity of such context specific actions, e.g., in work flows, transactions, and processes performed by orchestrated services.

It is important to mention here that the overall architecture of Hydra is designed based on the WS-* family. Because of this technical ground it is necessary for HIM to be compatible to the WS-* family.

Figure 5 shows an anatomic view of the architecture. We propose a hybrid model of existing IdM protocols (SAML, OpenID, InfoCard). This hybrid model enriches the architecture with all round features that are desired by the developer. Moreover, the coexistence of SAML, OpenID, and InfoCard allows to compensate each others' limitations and, thus, to mitigate vulnerabilities. The Hydra Federation Engine supports IdP, RP and Identity Selector of any kind. Moreover, on the client side four different variants of InfoCard are supported. Z-card will bring the user more privacy and r-card will present the user the rich context-aware feature. Since the relationship card will reside on the client machine, the user will have full control over his privacy. Thus, the advantages of intelligent environments and privacy have been put in place in a balanced manner.

The communication viewpoint on the architecture is described in the Figure 6. In this figure the resident of our fictitious scenario is accessing various services in his ubiquitous world. Every node works as IdP and RP thereby realizing the concept of a "Distributed Identity Provider".

"Distributed Identity Provider" means no centralized IdP managing the identity of the user, it is rather the surrounding ubiquitous devices that play the role of IdP and RP back and forth. Authentication to an individual RP has to be realized in 5 basic steps described in the figure. The concept of "Distributed Identity Provider" brings more privacy to the user as no centralized IdP manages the identity of an individual.

In this particular example showed in Figure 6, the subject is being identified by an acceptable IdP (the server) to give access to the laptop. Now once the laptop has the information that the subject has been identified and federated by an acceptable IdP it can take over the

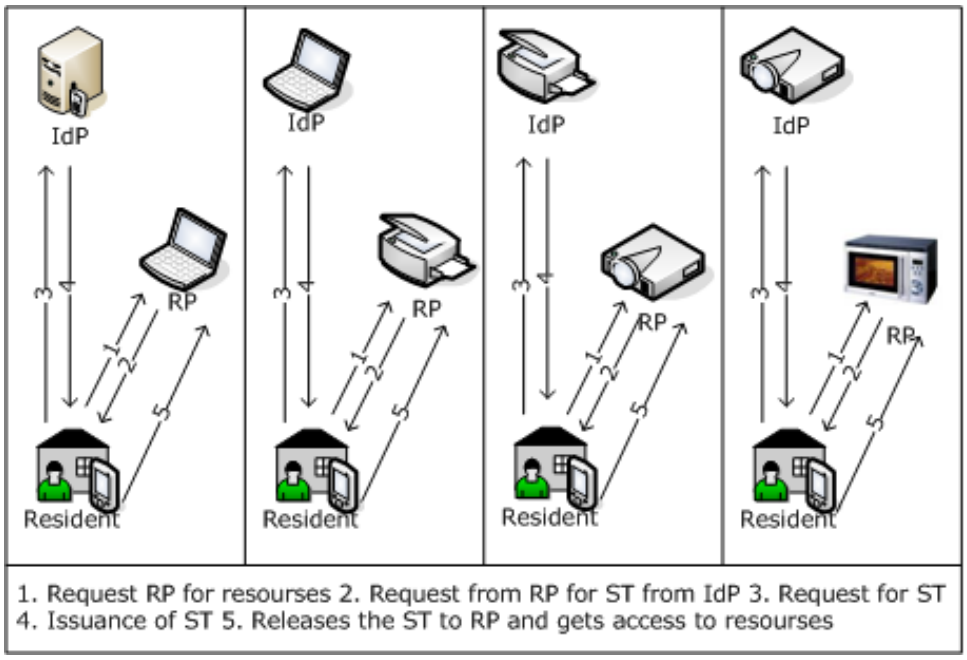

Figure 6: Communication viewpoint on the architecture of Hydra Identity Manager. The resident of the test scenario (Section II.A) accessing various resources using the principle of identity federation with distributed IdP(s).

role of the IdP and federate the subject further to other RPs e.g., the printer. Similarly, the role of IdP propagates through node to node and each node can act as an IdP. A proper transitive chain is maintained. This example elaborates how "Distributed Identity Provider" can possibly work in a pervasive environment.

A Compliance to the Laws of Identity

The proposed architecture takes an attempt to minimize the vulnerabilities in Identity Management in ambient environments by being compliant to the ten laws of identity we have defined. In this section we will summarize how and up to what extend the architecture fulfills the ten laws.

1) Compliance to the First Law (User Empowerment: Awareness and Control)

The best way invented so far bringing user awareness and control in an Identity Management System is InfoCard. Let it be Windows CardSpace, DigitalMe, SeatBelt or any other InfoCard, when one of these cards pops up before a transaction takes place, it certainly raises the awareness of the user regarding the information he is about to disclose. Moreover, InfoCard enables the user to gain control over his data before disclosure. Thus, it complies with the first law.

2) Compliance to the Second Law (Minimal Information Disclosure for a Constrained Use)

The STS supports provided in HIM furnishes the developers with SAML, OpenID, Kerberos or even a custom token type. This is how HIM is designed to be token agnostic and virtually supports any token types. With the virtue of SAML and OpenID it is possible to make a bare assertion that a *Subject* has been authenticated by an IdP without having to disclose any information about the *Subject*. This is one feature of HIM that supports the second law.

Another supporting feature for the second law is the zero-knowledge-proof which is an integral part of HIM. The z-card module in the HIM architecture supports the ability of an identity selector to generate zero knowledge proofs that can be conveyed by the agent to the *Relying Party* without revealing any more than it is absolutely necessary and while maintaining the chain of trust back to the original token issuer. As a result, it brings strong support for the second law of identity.

3) Compliance to the Third Law (Non-repudiation)

The three criteria of "non-repudiation" defined in the third law are: authenticity, integrity, and time stamp. WS-Security specification defines all of these three criteria and thus ensures "non-repudiation". Since WS-Security resides at the very bottom of the HIM architecture stack it makes HIM compliant with the third law.

Laws of Identity	SAML	OpenID	CardSpace	Liberty	Higgins	HIM
1. User Empowerment	-	-	++	+	-	++
2. Minimal Disclosure	+	+	+	+	+	++
3. Non-repudiation	-	-	O	+	O	+
4. Directional Identity	O	++	++	++	+	++
5. Universal Identity Bus	-	-	+	++	++	+
6. Strength of Identity	-	-	-	-	-	+
7. Decoupling Layers	-	O	++	++	O	++
8. Usability	-	O	++	++	O	++
9. Context Consistency	+	++	++	++	++	++
10. Scalability	++	++	++	++	++	+

Table 1: Tabular result of the evaluation of state of the art technologies and the proposed architecture.

4) Compliance to the Fourth Law (Support for directional identity topologies)

Both SAML 2.0 and OpenID support directional identity and, therefore, law 4 is also satisfied by the proposed hybrid architecture. It is possible to configure SAML 2.0 to implement both bi-directional and unidirectional federation. SAML 2.0 as well as OpenID can be exposed to be omni-directional identity.

5) Compliance to the Fifth Law (Universal Identity Bus)

From a developer's viewpoint a UIB is an umbrella platform where he can implement the Identity Management System of his choice and is even able to cross-match different token types, protocols, and information cards. The hybrid architecture of HIM exactly attempts to target such an identity vision for building Identity Management applications.

6) Compliance to the Sixth Law (Provision of defining strength of identity)

The sixth law of identity is facilitated by HIM in the "Hydra.IdentityManager.Identity" namespace. Here, depending on the entity and the identity ownership, the relationship of the entity and the strength of the identity is defined.

7) Compliance to the Seventh Law (Decoupling identity management layer from application layer)

The Hydra Federation Engine (HFE) acts as the orchestrator of the process of the *Identity Metasystem*. The RP, IdP, and the *Subject* roles are defined in the HFE and, thus, federation is facilitated. This notion of *Identity Metasystem* decouples the Identity Management layer from the rest of the application. HFE in the middleware is what the developers can utilize to achieve federation.

8) Compliance to the Eighth Law (Usability issue concerning identity selection and disclosure)

Usability is strongly correlated to the user group and also depends on the nature of the application. Therefore, at a middleware level where the target user group and the nature of the application is not specifically known, it is necessary to facilitate the developer with a wide variety of support to implement usable Identity Management Systems according to his need. HIM gives support for implementing available InfoCards, e.g., CardSpace or DigitalMe. On top of that there is also room for building custom information cards. The developer can then choose the most suited InfoCard in terms of usability.

9) Compliance to the Ninth Law (Consistent experience across contexts)

HIM architecture allows r-card (relationship-card) that manages the users' context experience. These cards can hold different context relevant profiles. An r-card offers a superset of the functionality of an i-card specification by Microsoft. R-cards can be either self-issued, where your identity selector defines and issues the card on your behalf, or issued by a third-party, where an entity other than you defines and issues the r-card. With r-cards, this distinction is less important because in both cases an r-card represents a mutual relationship and agreement to share certain claims/attributes. With the virtue of r-cards the user experiences a context aware smart environment without having to compromise his privacy.

10) Compliance to the Tenth Law (Scalability)

The Hydra Federation Engine is designed in such a way that numerous IdP, RP, and *Subjects* can join in and out and federate identities. At the same time it also supports spawning multiple identities and to manage them in proper ways. This feature enriches the architecture with scalability and, thus, satisfies the tenth law of Hydra identity.

VI. CONCLUSION

The overall comparison of the proposed architecture and the state of the art technologies are presented in Table 1. Since this evaluation is from a middleware viewpoint, it is not justifiable to make a statement that any one of these laws is impossible to realize using one of the existing frameworks. Rather it is more viable to say that some of the frameworks may have strong support to implement one of the laws and on the other hand some of them poorly support that law to be implemented in Identity Management System for ambient environment. That is why we came up with a scale of poor (-) to very good (++) and stated the result in Table 1.

In this paper we have presented an architectural approach to tackle the challenges of Identity Management in ubiquitous computing. The hybrid architecture presented has been adopted from the existing standardize state of the art IdM technologies. The future plan is to implement the architecture and integrate it in the Hydra middleware. The Higgins framework has been chosen for implementation based on the result of the evaluation.

REFERENCES

[1] Cameron, K, Laws of Identity (2005), Microsoft Corporation, last access May 2009.

[2] Mercuri, M. 2007 *Beginning Windows Cardspace: from Novice to Professional*. Apress.

[3] Recordon, D. and Reed, D. 2006. "OpenID 2.0: a platform for user-centric identity management", in *Proceedings of the Second ACM Workshop on Digital Identity Management* (Alexandria, Virginia, USA, November 03 - 03, 2006). DIM '06. ACM, New York, NY, 11-16. DOI= http://doi.acm.org/10.1145/1179529.1179532

[4] Rosenberg, J. and Remy, D. 2004 *Securing Web Services with Ws-Security: Demystifying Ws-Security, Ws-Policy, Saml, XML Signature, and XML Encryption*. Pearson Higher Education.

[5] McMurtry, C., Mercuri, M., Watling, N., and Winkler, M. 2007 *Windows Communication Foundation Unleashed (Wcf) (Unleashed)*. Sams.

[6] OpenID, http://openid.net/

[7] Hydra, Deliverable D2.1a Scenarios for usage of Hydra in Building Automation, 25 January 2007 - version 1.41.

[8] The Hydra Project, http://www.hydramiddleware.eu

[9] Vedamuthu, A. S., Orchard, D., Hondo, M., Boubez, T., Yendluri, P., Web Services Policy 1.5 – Primer, W3C Working Draft 18 October 2006, http://www.w3.org/TR/2006/WD-ws-policy-primer-20061018

[10] Artman, H. 2002. Procurer usability requirements: negotiations in contract development. In *Proceedings of the Second Nordic Conference on Human-Computer interaction* (Aarhus, Denmark, October 19 - 23, 2002). NordiCHI '02, vol. 31. ACM, New York, NY, 61-70. DOI= http://doi.acm.org/10.1145/572020.572029

[11] *WS-Trust 1.3*, OASIS Standard 19 March 2007, http://docs.oasis-open.org/ws-sx/ws-trust/200512/ws-trust-1.3-os.html#_Toc162064937

[12] Akram, H., Hoffmann, M., *Laws of Identity in Ambient Environments: The Hydra Approach*, The Second International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies UBICOMM 2008, September 29 - October 4, Valencia, Spain

[13] Oh, Hyun-Kyung; Jin, Seung-Hun, "The Security Limitations of SSO in OpenID," *Advanced Communication Technology, 2008. ICACT 2008. 10th International Conference on* , vol.3, no., pp.1608-1611, 17-20 Feb. 2008

[14] Alrodhan, W.A.; Mitchell, C.J., "Addressing privacy issues in CardSpace," *Information Assurance and Security, 2007. IAS 2007. Third International Symposium on* , vol., no., pp.285-291, 29-31 Aug. 2007

[15] Anderson, R. J. 2008 *Security Engineering: a Guide to Building Dependable Distributed Systems*. 2nd. John Wiley & Sons, Inc.

[16] Akram, H., Hoffmann, M., *Supports for Identity Management in Ambient Environments: The Hydra Approach*, International Conference on Advances in Human-oriented and Personalized Mechanisms, Technologies, and Services I-CENTRIC 2008 October 26-31, 2008 - Sliema, Malta

[17] Maler. E., SAML, Liberty Alliance, openLiberty, and Concordia, Sun Microsystems, Inc, 2007.

[18] *On the Insecurity of Microsoft's Identity Metasystem CardSpace*, Press release, Bochum, Germany, May 27, 2008, http://demo.nds.rub.de/cardspace/PR-HGI-TR-2008-003-EN.pdf

[19] McCullagh, A., Caelli, W., *Non-Repudiation in the Digital Environment*, First Monday, volume 5, number 8 (August 2000), URL:

http://firstmonday.org/issues/issue5_8/mccullagh/index.html

[20] *Web Services Metadata Exchange (WS-MetadataExchange), Version 1.1*, August 2006, http://download.boulder.ibm.com/ibmdl/pub/software/dw/specs/ws-mex/metadataexchange.pdf

[21] *Web Services Federation Language (WS-Federation),* Version 1.1, December 2006, http://download.boulder.ibm.com/ibmdl/pub/software/dw/specs/ws-fed/WS-Federation-V1-1B.pdf?S_TACT=105AGX04&S_CMP=LP

[22] Hodges, J., *Technical Comparison: OpenID and SAML - Draft* 06, January 17, 2008, http://identitymeme.org/doc/draft-hodges-saml-openidcompare-06.html#tbl-exec-summary

[23] Armando, A., Carbone, R., Compagna, L., Cuellar, J., and Tobarra, L. 2008. Formal analysis of SAML 2.0 web browser single sign-on: breaking the SAML-based single sign-on for google apps. In *Proceedings of the 6th ACM Workshop on Formal Methods in Security Engineering* (Alexandria, Virginia, USA, October 27 - 27, 2008). FMSE '08. ACM, New York, NY, 1-10. DOI= http://doi.acm.org/10.1145/1456396.1456397

[24] E. Maler, SAML basics - A technical introduction to the Security Assertion Markup Language, http://www.itu.int/itudoc/itu-t/com17/tutorial/85573.html

[25] Eve Maler et al. Assertions and Protocol for the OASIS Security Assertion Markup Language (SAML) V1.1 Oasis-Open, September 2003. OASIS Standard, http://www.oasisopen.org/committees/security/