CAFS: A Framework for Context-Aware Federated Services

Md. Sadek Ferdous
School of Computing Science University of Glasgow
Glasgow, Scotland G12 8QQ
Email: m.ferdous.1@research.gla.ac.uk

Ron Poet
School of Computing Science University of Glasgow
Glasgow, Scotland G12 8QQ
Email: ron.poet@glasgow.ac.uk

***Abstract*—In this paper we explore two issues: Federated Identity Management and Context-Aware Services. In the last decade or so we have seen these two technologies gaining considerable popularities as they offer a number of benefits to the user and other stakeholders. However, there are a few outstanding security and privacy issues that need to be resolved to harness the full potential of such services. We believe that these problems can be reduced significantly by integrating the federated identity architecture into the context-aware services. With this aim, we have developed a framework for Context-Aware Federated Services based on the Security Assertion Markup Language (SAML) and eXtensible Access Control Markup Language (XACML) standards. We have illustrated the applicability of our approach by showcasing some use-cases, analysed the security, privacy and trust issues involved in the framework and the advantages it offers.**

***Keywords*-Federated Identity Management, Context-Aware Services, SAML, XACML, GeoXACML, Attribute Aggregation.**

I. Introduction

In the last fifteen years or so, we have seen a tremendous expansion of the Internet and web-enabled online services. To allow users to access different online services in a seamless manner while maintaining security and privacy, the concept of Federated Identity Management (FIM, in short) has been introduced. It has gained considerable popularities as it offers a significant number of advantages to different stakeholders [1]. On the other hand, we have also experienced an incredible proliferation of smartphones equipped with an array of sensors in the last decade or so. These devices have been the driving force behind the ever-increasing popularities of context-aware services mainly in the form of location-based services. A study by Oracle predicted $85 billion worth of market share for context-aware services by 2015 [2]. There are numerous works that have explored how context can be used to provide online services. However, there are a few major problems that are yet to be addressed. Firstly, most of these works mainly focus on the authentication or the authorisation of users neglecting other aspects of identity management. In that sense, the issue of Context-Aware Identity Management has not been explored in detail. Secondly, the existing works are mostly based on the SILO Model [3] where the users need to authenticate themselves, if needed, separately to access each service which could be quite inconvenient for the user. The usability of such services could be easily improved by introducing the capability of federated services allowing users to access services only with a single authentication using the Singe Sign On (SSO) feature [1]. Thirdly, the security and privacy issues in the existing context-aware frameworks have not been addressd properly, as such, unresolved security and privacy issues need to be addressed to harness its full potential [4]. Moreover, with the ever increasing popularities of social network accounts, user attributes are scattered over different providers. Allowing users to aggregate attributes from different providers would enable novel service access scenarios. In this work we will explore the possibility of combining the attribute aggregation mechanism with the Context-aware Identity Management to formulate novel use-case scenarios and show how all these problems can be tackled effectively using this. The contributions of this paper are:

- At first, we have provided a concrete definition of Context-Aware Identity Management and used this to develop a framework for Context-Aware Federated Services. The framework is based on a set of requirements that, we believe, can improve the usability, security and privacy of context-aware services.
- We have developed an attribute aggregation mechanism based on the Identity Proxying model as described in [5] and integrated this mechanism into our framework.
- We have illustrated a few use-cases to show the applicability and usefulness of our framework.
- Finally, we have investigated how our framework meets different requirements, discussed the advantages it offers and compared our framework with existing works against the set of requirements.

The rest of the paper is structured as follows. We discuss the existing work related to this paper in Section II. A short background on Federated Identity Management is provided in Section III. The definition of Context-Aware Identity Management is formulated in Section IV and a set of different requirements for a context-aware service is compiled in Section V. Then we discuss our developed framework in Section VI and illustrate a few use-cases in Section VII. A brief discussion of different security, privacy and trust issues regarding our framework, the advantages it offers, its comparison with existing works and the scope for future work can be found in Section VIII. We conclude in Section IX.

II. Related Work

A large number of works can be found in the literature focusing how contexts can be integrated either into an authentication framework to authenticate a user or into an authorisation framework for authorising a user to access a service. There are only a handful of works that analyse the effect of contexts on the whole spectrum of identity management. Similarly, there are also a very few works on attribute aggregation in the setting of FIM. We discuss each of these aspects in the following subsections.

A. Context-Aware Authentication

Hayashi et al. present a probabilistic model for choosing an active authentication factor (No-PIN, PIN and Password) based on several passive factors (location and time of day) in [6], where the active factor is chosen based on a specific location or time. This approach, being only suitable to be used in a mobile phone, cannot be integrated for web-enabled services.

A framework for authenticating a user based on the proximity to external sensors is presented in [7]. The authentication mechanism is activated when a user brings her device near to the sensor. The authentication is done using certificates which are passed between the device and the sensor using a challenge-response protocol. In [8], authors propose a mechanism for mutual authentication of users based on contextual data. The paper does not discuss how their framework can be integrated for online services.

B. Context-Aware Authorisation

One of the earlier works that proposed how contextual information can be incorporated for controlling accesses is called the Generalized Role Based Access Control (or GRBAC) [9]. The authors defined environment roles (the time of day, weather conditions and other spatial and temporal values) as contexts. One major problem of GRBAC is that a large number of environment roles make the system difficult to manage manually. This work also is theoretical in nature and has not been applied in practical scenarios.

Explaining the need to verify the authenticity and validity of contextual information, a context sensitive access control architecture is defined in [10]. The authors achieve these goals by introducing several external components as well as outline a detailed protocol flow using their architecture. The main problem of their proposal is the reliance on several external parties which may be difficult to implement for federated services considering different security domains and trust issues.

Jean-Yves et al. have proposed a model to include contextual information into the authorisation process in highly dynamic environments [11]. Contextual information is provided by observers which are actually different devices. The validity of the contextual information is verified by a trusted third party. A user is authorised to perform an operation on the respective object only if the context is valid.

C. Context-Aware Authentication & Authorisation

In [12], the authors have presented an agent-based framework for authentication and access control in context-aware services using an additional entity called the authentication and access control agent which is a trusted third party. The agent is responsible for collecting contextual data from different servers, obtaining the user's identity and attributes from the IdP, etc. The main problem, again, is the reliance on several external parties which may be difficult to implement for federated services.

An interesting work in which contextual information such as roles and location are used for user authentication can be found in [13]. The paper uses passive cues (contextual information namely roles and the location of the user) and active cues (username/password) for user authentication. The access control system and the policies are based on the XACML architecture and the role and location information are combined to create policies for the XACML system. The role of the user is retrieved from the system once the authentication is done and the location of the user is retrieved from a QRCode (Quick Response Code) that the user scans using a mobile phone. We have used their work as a basis for our proposed deployment. However, the major differences between their work and our deployment are that they did not consider the usage of federated services and different security and privacy issues have not been analysed in detail.

D. Context-Aware Identity Management

The only work to discuss the issue of context-awareness in Identity Management is available in [14] where the authors have investigated which contextual information constitute the identity of a user and how can such information be collected in a standardised way from different sources. The authors also proposed a novel framework consisting of several external components to address these issues. Unfortunately, the inclusion of external components makes this framework impractical to be used with the existing architecture of federated services.

E. Attribute Aggregation

In the traditional identity management, the user can release attributes only from a single IdP to the SP in a single session. The Attribute Aggregation is the mechanism which allows a user to retrieve and combine attributes from multiple IdPs in a single session. There are different models of attribute aggregation in the setting of FIM such as Application Database, SP-Mediated, Linking Service, Identity Federation/Linking, Identity Proxying, Identity Relay and Client-Mediated models. The models have been analysed in greater detail in [5], [15], [16]. Among all these models, only the Linking Service and Identity Federation/Linking models have existing implementations. However, they require a number of complex additional pre-steps (e.g. linking different IdPs by the user) beforehand and have complex trust relationships. Also, the idea of combining the concept of attribute aggregation with the context-aware services has not been considered before.

III. Federated Identity Management

A system that is used for managing the user identity is called an Identity Management System (IMS). Each IMS has several parties involved which are: **Service Provider (SP) or Relying Party (RP)** - an entity that provides services to the users or to the other SPs, **Identity Provider (IdP)** - an entity that provides partial identifiers to the users to enable them to receive services from a SP and **User** - an entity that receives a service from a SP. Among different IMS, the Federated Identity Management has gained much attention and popularities.

The FIM is based on the concept of Identity Federation. An identity federation is a business model in which a group of two or more trusted parties legally bind themselves with a business and technical contract to allow a user to access restricted resources seamlessly and securely from other partners from different Identity Domains [1], [17]. An identity domain is the virtual boundary, context or environment in which a digital identifier is valid [17]. Single Sign On (SSO) is the capability that allows users to log in to a system in one identity domain and then access other related but autonomous systems in other domains without further logins.

A federation can be formed consisting of only one IdP in an identity domain and more than one SP with each SP residing in a separate identity domain. Such a federation can be regarded as the Type 1 Federation. Several Type 1 federations can be combined to form a larger federated identity domain, regarded as the Type 2 Federation. The issue of trust is a fundamental concept in FIM as different autonomous bodies need to trust each other inside the federation. Such parties inside a federation are said to form the so-called Circle of Trust (CoT).

IV. Context-Aware Identity Management

In the literature of Context-Aware Services, what the term *Context* means is highly debated and it has been defined in numerous ways. Schilit and Theimer used the term *context-aware* for the first time in [18] where they described contexts as location, identities of nearby people, objects and changes to those objects. Similarly, Ryan et al. regarded context as the user's location, environment, identity and time [19]. We prefer the definition of context given by Abowd et al. [20] as it considers the application itself as a context along with other situational information. Interestingly, the term *Context* has a specific meaning in Identity Management. A context in identity management is the application domain or namespace under which an entity exists, operates and is identified with a specific identifier. The identity of an entity within that application domain is defined as the partial identity of that entity for that application domain. Combining all these we define a context in terms of Identity Management in the following way:

Context *- Context in Identity Management is any information that represents the user's partial identity (using an identifier), the physical location from where the user tries to access any service, the time and date of the service request and the application domain in which the partial identity of the user is valid (the IdP) or to where the user is requesting to access a service (the SP).*

Such information can also be regarded as the *Contextual Information* or *Dynamic Attributes*. We will be using the term *Context, Contextual Information & Dynamic Attributes* interchangeably throughout the paper. Using this definition of context, we can define the term *Context-Aware Identity Management* in the following way:

Context-Aware Identity Management *- A Context-Aware Identity Management is a specific type of Identity Management that might use contexts for:*
– registering and de-registering the partial identity (containing identifiers) of a user to an IdP.
and must use contexts:
–for highlighting, selecting and/or generating an identifier during a specific service provisioning scenario;
– to aid users to select appropriate attributes that the user wants to release to a specific SP;
– to aid users to select an appropriate subset of contexts to be included as dynamic attributes along with other selected attributes;
– for utilising them as dynamic attributes along with other attributes during the authorisation and the service provisioning phase; and
– for supporting business and security applications to provide innovative service scenarios.

V. Requirement Analysis

A comprehensive set of requirements for an ideal identity management system has been compiled in [21]. Moreover, the nature of a context-aware service imposes some other novel requirements. Collecting a subset of essential requirements from [21] and combining them with the novel requirements, we create the following set of requirements that we want to be met by our framework:

Functional Requirements. The functional requirements outline the conditions that the framework must meet to ensure its desired functionality.

- F1 **Online Service Integration.** The framework should be integratable with online services.
- F2 **Single Sign On Capability.** The framework should provide the SSO capability to allow users to access services from different SPs of the same federation in a seamless manner without any further logins.
- F3 **Indoor/Outdoor Location Tracking.** The framework should have the capability to locate users both indoor and outdoor.
- F4 **Showing Required Attributes.** The framework should inform users regarding the attributes required to access a particular service.
- F5 **Advanced Authorisation Capabilities.** The framework should provide advanced policy-based authorisation capabilities which will allow administrators of the SP

to write advanced yet flexible authorisation policies to enable complex use-case scenarios.

Security Requirements.

S1 **Confidentiality, Integrity, & Authenticity.** Contextual data and other user attributes should be transmitted between different entities maintaining their confidentiality and integrity. The authenticity of contextual data should be verifiable when required.

S2 **Secure User Authentication & Authorisation.** The framework should ensure that the services can be offered only to the securely authenticated and properly authorised users.

S3 **Preventing Relay Attacks.** The framework should be able to prevent any relay attack involving contextual data.

Privacy Requirements.

P1 **Support for Anonymity/Pseudonymity.** The framework should allow the user to provide an anonymous or pseudonymous identifier.

P2 **Selective Disclosure.** The framework should allow the user to choose the attributes that she wants to disclose to the SP.

P3 **Explicit Consent.** The framework should release attributes, including contextual information, to the SP only after the user provides her consent explicitly. The requirements *P2* and *P3* will enable data transparency and user control over their data.

The first condition in the Definition 2 has not been transformed into a requirement since it is mostly dependent on a particular IdP which might deploy various procedures to meet that requirement or even might opt out from deploying it. The above requirements have been compiled in such a way that it meets the other conditions in Definition 2.

VI. CAFS Framework

Designing a context-aware framework that allows federated services would require to base the work on the concept of context-aware identity management and should satisfy all requirements compiled in Section V. Firstly, we will require an IdP with the capability to provide fine-grained contextual information as the tranditional IdP cannot provide such information. A smartphone can be an excellent choice to act as such an IdP since the current generation of smartphones are equipped with sensors and with the fast 3G or 4G network capabilities. We have designed and developed a novel framework to utilise contexts for federated services which we call ***CAFS*** or Context-Aware Federated Services. The basic idea is very similar to the existing federated architecture with the inclusion of a novel type of IdP called *Portable Personal Identity Provider* or *PPIdP* as proposed in [22]. The PPIdP is a special type of IdP that is hosted in a mobile device owned and/or used by the user. It is under the full control of the user. The user decides what attributes should be stored in such an IdP and selects which attributes should be released to which SP. The architecture of PPIdP has two

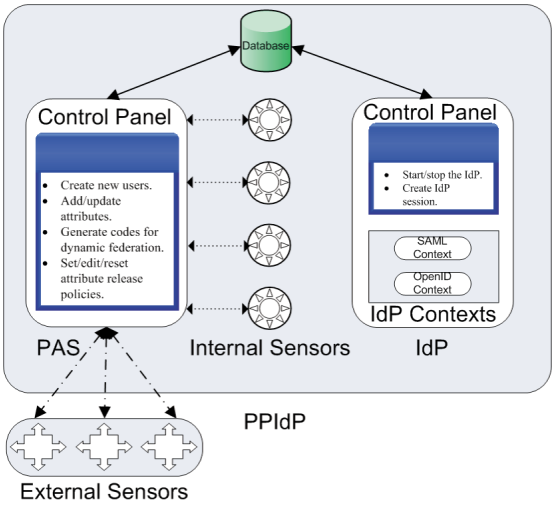

Fig. 1: The architecture of PPIdP.

major components (Figure 1): the Personal Attribute Store (or PAS) and the IdP. The PAS consists of a back-end database to store the user attributes and a Control Panel for users to manipulate their attributes. The current generation of smartphones, equipped with several sensors such as GPS Receiver, Gyroscope, Proximity Sensor, Accelerometer, etc, can be used to sense contextual information. The PAS can sense location information from the GPS receiver and store them as dynamic attributes into the back-end database. In some cases, external sensors can be used to supplement and/or complement other contextual information. The PAS can read such information using the camera and then store them as dynamic attributes in the database. The IdP component is responsible for providing IdP services and consists of two sub-components: the Control Panel and the IdP Context. The control panel is used to start or stop the IdP service and the IdP context consists of several interfaces where each interface is responsible for handling a specific identity management protocol such as Security Assertion Markup Language (SAML) or OpenID, etc.. The IdP component also shares the same back-end database of the PAS to retrieve static or dynamic attributes during the user authentication phase. The PPIdP can be integrated with the SP using the concept of Dynamic Federation to create the federation in a dynamic fashion [23]. Such a federation involving the PPIdP is called the Personal Identity Federation (PIF) [22]. The PIF can be of two types: Type 1 and Type 2, just like the traditional federation. The second type allows the user to federate two IdPs (the PPIdP & a traditional IdP) by a user of both IdPs in such a way that the user can import some dynamic attributes from the PPIdP to the IdP and can subsequently pass those attributes to the SPs. The middle IdP in this setting is known as the *Proxy IdP* and is assumed to be fully trusted by the SP with a mutual trust agreement. On the other hand, the PPIdP will be considered as a semitrusted entity (not fully trusted by another entity, see [23]) by both the proxy IdP and the SP following the condition of the dynamic federation as explained in [23].

The authorisation of users in the CAFS is handled by the eXtensible Access Control Markup Language (XACML). The XACML is an OASIS standard that defines a general-purpose access control and authorisation system [24]. It consists of a policy language based on XML and a processing system that knows how to interpret the policy with respect to the relevant application. The policy language is used to create a policy that enlists the requirements to access a resource in a protected environment. The major components of XACML are: i) Policy Administration Point (PAP) which is responsible for creating and managing all policies, ii) Policy Enforcement Point (PEP) which is responsible for intercepting a user's request and enforcing an XACML decision received from the Policy Decision Point (PDP, see below), iii) Policy Decision Point (PDP) which is responsible for evaluating a user's request based on existing policies and returning an XACML decision to the PEP and iv) Policy Information Point (PIP) which gathers additional user attributes.

The XACML is an example of a request/response language where a user submits a request to the PEP to access a protected resource. The PEP generates an XACML request based on the user attributes, the resource the user wants to access and the action (read/write) the user wants to perform on that resource and forwards that request to the PDP. The PDP evaluates the existing policies to determine if the user can perform that particular action onto that resource using supplied attributes, creates an XACML response and returns it to the PEP. Based on the response, the PEP approves/rejects the user's request.

The existing XACML standard cannot utilise any location data as conditions inside any policy. Geospatial XACML or GeoXACML is an Open Geospatial Consortium (OGC) (*http://www.opengeospatial.org/*) standard that has been introduced as an extension to the XACML Version 2.0 to enforce access restrictions based on geographic information [25]. The geographic information is encoded in Geography Markup Language (GML) [26] and can be used inside a policy as a condition. The GeoXACML architecture has been incorporated with the PIF architecture to complete the full architecture for the CAFS Framework. The full architecture for CAFS using Type 1 PIF is given in Figure 2. This architecture can be easily extended for the Type 2 PIF as well.

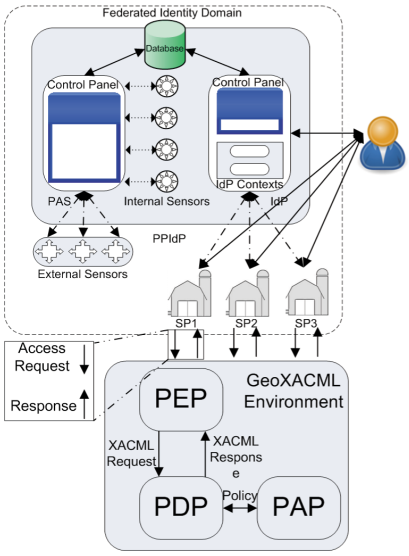

Fig. 2: CAFS Type 1 Architecture.

A. CAFS Development

The two sub-components of the PPIdP (the PAS and the IdP Component) have been developed as Android Apps. The SQLite Database [27] has been used as the back-end database to store user attributes. On top of the database, SQLCipher [28] has been used to store attributes securely. SQLCipher provides fast, secure and automatic 256-bit AES encryption/decryption of SQLite database entries. Users can use the control panel of the PAS to manually insert/update static attributes. For dynamic attributes such as location data, the value (in the format of the Geographic coordinate system where the values are represented with altitude, latitude and longitude) is collected from the Android Location Service that provides location data from the GPS Receiver, cell tower or Wi-Fi signals [29]. The PAS also allows the user to set Attribute Release Policies (ARP) that dictate what attributes should be presented to the user during the user authentication phase. For our current work, we have also updated the previous PAS, as reported in [22], with the capability to retrieve information from *QRCodes* using the camera (see below).

The IdP component consists of two sub-components: the IdP Context and the Control Panel. The IdP Context is a servlet container deployed using the Jetty Web Server [30]. It contains several servlets which are responsible for handling respective protocol. The current implementation of the PPIdP supports SAML based on the Web Browser SSO Profile and HTTP Post binding. The control panel is used to start/stop the IdP. Once the IdP is started, the user can use the preferred web browser of her mobile phone to access federated services.

The existing open source XACML APIs for Java such as SunXACML [31], HERASAF XACML Core [32], enterprise-java-xacml [33] do not provide any support for the GeoXACML. There exists a commercial implementation of the GeoXACML in *http://geoxacml.secure-dimensions.net/*, however, it is not open sourced. After a thorough search, we have been able to locate an incomplete implementation of GeoXACML in the GeoServer SVN Repository [34]. We have exported that project and integrated it with the SunXACML API by going through some major modifications in a number of its classes. The combined API has finally been bundled with the Jetty Web Server that runs on a local web server.

We have been looking for a simpler model of attribute aggregation that does not involve complex pre-steps. After a thorough analysis, the Identity Proxying (IP) model seems to be a viable candidate. In this model, the SP allows the user to aggregate attributes from multiple IdPs using a highly trusted IdP, also known as the Proxy IdP. The trusted IdP is federated with multiple IdPs and the SP. The federations can be either pre-configured at the admin level or can be federated dynamically. In this model, the user is forwarded

to the trusted IdP at first and then the trusted IdP forwards the user to other IdPs. After the user is authenticated separately at each IdP, she returns back to the trusted IdP with different attributes and the trusted IdP combines all these attributes and might supplement the combined set with its own user-attributes and then reasserts all attributes to the SP. We have used the SimpleSAMLphp library [35], a PHP implementation of SAML, for implementing the IP model. Our developed proxy IdP allows attributes to be aggregated from any SAML IdP, however, in this paper we mostly concentrate on aggregating attributes from the PPIdP. Apart from the aggregated attributes, the proxy IdP also adds two pieces of information to each set of attributes from a single IdP: the source of attributes (the entity ID of the IdP where this set of attributes has originated) and a trust metric for each set. We have used the NIST LoA (Level of Assurance or Level of Authentication [36]) level as a trust metric where Level 1 indicates attributes from a semitrusted source and Level 2 value signifies from a highly trusted source. A LoA value of 2 is added to each set for attributes from a highly trusted IdP whereas a LoA value of 1 is added to each set for attributes from semitrusted IdPs (e.g. PPIdP). These two pieces of information are provided to allow the SP to take appropriate authorisation decisions.

VII. Use-cases

Now we will illustrate a few use-cases to show the applicability of our approach.

A. Settings

We have two SPs (denoted as SP1 and SP2 hereafter) deployed with the SimpleSAMLphp. The PEP has been developed in PHP and integrated with each SP. Following the SAML authentication phase when the user returns to the SP with the assertion, the user lands on the PEP. The PEP retrieves the attributes from the assertion, creates an XACML request with those attributes and forwards the request to the PDP. The PDP evaluates the user's request using the existing policies and returns an XACML response to the PEP. Based on the XACML response, the user is granted or rejected by the PEP.

Each SP has a few web pages that have been protected using the PEP. Each page has different requirements of static attributes and contexts (dynamic attributes) that need to be fulfilled to access that page. All these requirements have been used as conditions inside the policies in the GeoXACML environment. Any type of contextual information can be handled by our framework, however, the current implementation only handles the location (indoor and outdoor), date and time, and role as contextual information. Table I specifies the list of attributes (static and dynamic) required for each page in the respective SP. In the table, the value *val* refers to the range of these geographic values: (55.865318, -4.267437), (55.865185, -4.2660857), (55.864722, -4.266203), (55.864806, -4.268220). The *gte* condition signifies that the user attribute should be greater than or equal to the stated value, the *equals* signifies that the user attribute should match the stated value, the *within* signifies that the location of the user should be within the specified value and the *between* signifies that the request should be made during the specified duration. The value *ST or HT* as the value of the *idp* attribute signifies that the IdP has to be semitrusted or highly(or fully) trusted respectively.

TABLE I: Attributes required for each page.

Name	Resource	Attributes	Cond.	Values
SP1	Page1.php	loa	gte	1
		location	within	*val*
	Page2.php	loa	gte	1
		location	within	*val*
		building	equals	1
		floor	equals	3
		room	equals	5
		time	between	14:00 - 14:30
		date	equals	2014-02-04
	Page3.php	idp	equals	*HT/ST*
		location	within	*val*
		building	equals	4
		floor	equals	5
		room	equals	2
		time	between	14:00 - 15:30
		date	equals	2014-02-04
		loa	gte	1
		idp	equals	*HT*
		pos	equals	*student*
		org	equals	*a*
		loa	equals	2
SP2	Page1.php	loa	gte	2
		salarygrade	gte	6
		age	gte	33
	Page2.php	loa	gte	2
		salarygrade	gte	6
		age	gte	33
		position	equals	admin

The location attribute provided by the PPIdP can locate the user externally within a certain geographical location. However, some services or resources (as in our cases of the *Page2.php & Page3.php* of SP1) might require to locate the user internally inside a building (e.g. in a room of a particular floor). Verbally, the requirements for accessing the *Page2.php* can be expressed in this way: "*a user can access Page2.php only if she is within that geographic location and has a loa value greater than or equal to 1 and has submitted the request from Building 1, Floor 3, Room 5 in between 14:00 to 14:30 on 4 February 2014*". There have been a few works to locate a user inside a building, however, the technology is not precise yet. That is why we have adopted an alternative approach using Dynamic QRCodes as introduced in [7], [13]. QRCode is a type of two dimensional bar codes. It can display information in a pictorially encoded format which can be read by a device having a digital camera and a specific piece of software to decode that information [37]. It has gained considerable popularities with the rise of smartphones as most of them are capable to decode information shown via QRCodes. The traditional QRCodes are mostly static in nature meaning that once it is generated its contents cannot be altered. With a computer system running in the background, we can easily generate and display a dynamic QRCode refreshed with new contents after a certain duration. We have adopted this approach. The information encoded into the QRCode is: Building Number, Floor Number, Room Number, Date and Time. In our development, we have emulated this scenario by developing an Android app. The assumed admin provides

the required information (Building no., Floor no., etc.). The information is combined and then encrypted. The secret key is shared between the app and the PEP. The user scans the QRCode using the PAS before entering the specific room from where she has to access the specified resource. The encrypted information retrieved from the QRCode is saved as an attribute to the PAS that the user must release to access that resource. This means that users must know beforehand which attributes they need to release to these SPs. That is why when the user visits the homepage of the respective SP, she is informed which attributes are needed to access any service. For accessing the *Page3.php*, the user, in addition to the attributes similar to *Page2.php*, will need to release the *pos & org* attributes from the proxy IdP to testify that she belongs the organisation (e.g. as a student of the university). This means that the user will need to aggregate attributes from two SAML IdPs: the location, building, floor, room, time and date attributes from the PPIdP and the pos and the org attributes from a highly trusted IdP. Here, we have assumed that the pos and the org attributes are stored at the proxy IdP and thus will have a LoA value of 2.

With these settings now we can discuss a few use-cases. It is assumed that users have already federated the PPIdP with the respective SP and with the trusted(proxy) IdP to create a Type 1 and Type 2 PIF.

B. Use Case: SP1

The flow for accessing each page is given below:

Accessing Page1.php. The user visits the homepage of SP1. She notes that she will need to release the location attribute to access *Page1.php*. The user clicks the *Page 1* button. She is redirected to the WAYF Page of SP1. The user selects the PPIdP. A SAML Authentication Request is sent to the PPIdP. The user is authenticated there with her username and password and is provided with an HTML form, known as the consent form, containing the list of attributes. The user selects the location attributes and clicks the *Submit* button. A SAML assertion with the SAML response is sent back to the PEP. The PEP retrieves the attributes from the SAML response. Since the response is from an untrusted IdP, the PEP assigns a LoA value of 1 for this response. Then an XACML request with the *location* and *loa* attributes are sent to the PDP. The PDP uses the existing policies to evaluate the user's request and an XACML response with *Permit/Deny* value is sent back to the PEP and the user is granted or denied accordingly.

Accessing Page2.php & Page3.php. At this point, if the user tries to access *Page2.php*, she will require additional attributes. For that, she needs to be in that specific room of the floor in the building from where she can access the service. Assuming that the user is in front of that specific room (e.g. Building 1, Floor 3 and Room 5) and there is a Dynamic QRCode displayed in front of the room, the user uses the PAS to scan the QRCode and saves the information as an attribute, called *IntPosition*, into the PAS. Then the user visits the homepage of SP1 and clicks the *Page2.php*, the usual SAML protocol flow takes place. Assuming the user is authenticated at the PPIdP, the user is presented with an HTML form in which she chooses to release the *location & IntPosition* attributes (Figure 3). A SAML assertion with the attributes are sent back to the PEP. Since the assertion contains the *IntPosition* attributes, the PEP decrypts its value and retrieves the *building, floor, room, date & time* attributes. All these attributes are used to create an XACML request to send to the PDP. Then the usual XACML flow takes place. If the user scans the QRCode at the right indoor location, the user can access the *Page2.php*.

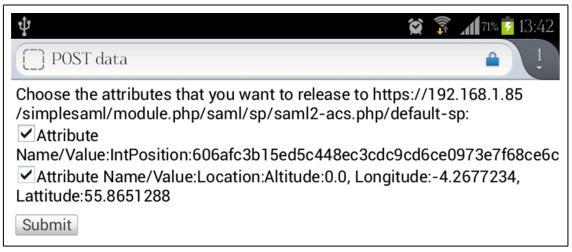

Fig. 3: Releasing attributes at the PPIdP.

To access *Page3.php*, the user needs to release attributes from two IdPs as discussed above. At first, the user is forwarded to the proxy IdP where she is authenticated. Then, the user is presented with a consent form containing the user attributes. The form also contains a button called *Aggregate More Attributes* (Figure 4). If the user clicks the button, a session with the already aggregated attributes is created and the user is forwarded to the IdP selection page at the proxy IdP where other SAML IdPs (including the PPIdP) linked to the proxy IdP are listed. Assuming the user chooses the PPIdP, she is forwarded to the PPIdP with a SAML authentication request. After being authenticated at the PPIdP, the user is presented with the consent form as Figure 3. The user selects attributes and clicks the *Submit* button. A SAML assertion with the attributes are returned to the proxy IdP. The proxy IdP validates the assertion and retrieves the attributes. Then it retrieves the previously aggregated attributes from the session and combines both sets and presents them in the consent form (Figure 5) with the appropriate IdP name and LoA value for each corresponding set of attributes. The IdP name and LoA attributes for each set are uneditable so that they cannot be changed by the user. If the user chooses any single attribute from a set, the IdP and the LoA value for the corresponding set will be added to the released set of attributes. Assuming the user chooses the required attributes for accessing the *Page3.php*, an SAML response with a SAML assertion containing the attributes will be sent back to the PEP and the same XACML protocol flow takes places to check if the user can access the *Page3.php*. Note that, the user can access (*Page1.php & Page3.php*) or (*Page1.php & Page2.php*) when she can access *Page3.php* or *Page2.php* respectively, since the entities in both pair require the same value for the common *location* attribute.

C. Use Case: SP2

Since the pages in the SP2 requires an LoA value of 2, the user cannot use the PPIdP. The user visits the homepage of the

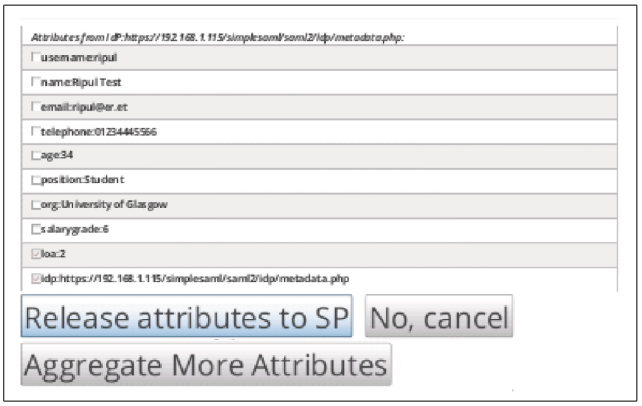

Fig. 4: The consent form at the proxy IdP.

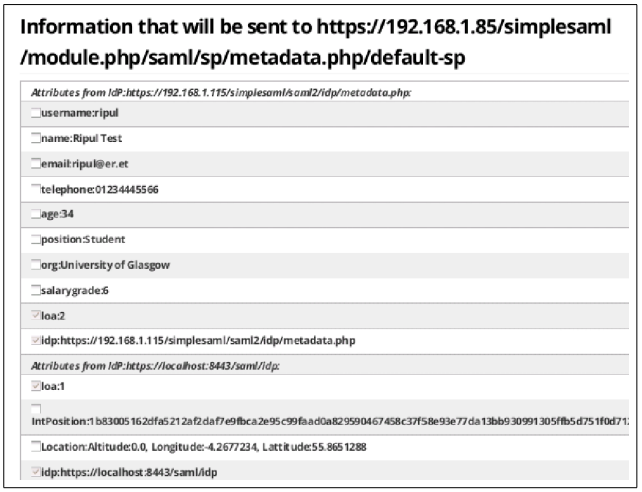

Fig. 5: Aggregated attributes.

SP2. She notes the list of attributes that she will need to release to access *Page1.php & Page2.php* (Table I). Assuming the user clicks the *Page 1* button, she is redirected to the WAYF Page of SP2. The user selects the proxy IdP. A SAML request is sent to the IdP where she is is authenticated and then is provided with the consent form containing the list of attributes. The user selects the attributes and clicks the *Submit* button. A SAML assertion with the SAML response is sent back to the PEP. The PEP retrieves the attributes from the SAML response. Then an XACML request with the attributes is sent to the PDP. The usual XACML flow takes place to check if the user can access the requested page.

VIII. Discussion

The whole framework has been designed and developed with online services in mind and provides a seamless single sign on capability across SPs in the same federation and thereby meets requirements *F1* and *F2*. The framework has the ability to locate users indoor as well as outdoor and thus meets requirement *F3*. To meet *F4*, the SP must have a way to request attributes from the IdP. Unfortunately, the SAML does not have any such mechanism. Therefore, we opted for the option where our deployed SPs let users know what attributes are needed to access a particular service and the user takes note of such attributes. We agree that this might not be a user-friendly option, however, given the current state of SAML, this seems to be the only logical choice. As evident from use-cases, CAFS supports an advanced policy-based authorisation infrastructure XACML and thus meets the requirement *F5* as well.

The framework has been developed using SAML and XACML standards. The communication channel in SAML is fully encrypted. The CAFS also uses the encrypted channel (HTTPS) to communicate with the GeoXACML environment. It guarantees the confidentiality and integrity of information, including contextual information, passed between involved entities. The discussion regarding the authenticity of contextual data, e.g. location, deserves further discussion since location information can be faked in smartphones using available apps. For example, Fake GPS location [38] and Location Spoofer [39] are two popular Android apps for spoofing location data for the Android platform. Therefore it is advisable to provide location-based services in such a way so that users need to be present there physically to avail the service thereby undermining the motive of spoofing location data. The alternative approach is to deploy external sensors that can be used to prove that the user has been near to the vicinity of the sensor at a certain time. We have adopted this approach using dynamic QRCodes as external sensors. To ensure that the information retrieved from the QRCode cannot be spoofed, the information is encrypted at first and then the QRCode is generated using the encrypted data. Here, we have assumed that the dynamic QRCodes are deployed by the SP and thus the SP can decrypt and decode the encrypted QRCode to retrieve the location data passed using the *IntPosition* attribute. This ensures the authenticity of location data and thus meets *S1*. The CAFS also authenticates the user securely over the HTTPS channel and provides the services only to the authroised users which is determined by the respective XACML policy and thus meets *S2*. To prevent relay attacks, the date and time of the day are also embedded inside the information in the QRCode. This ensures that the QRCode cannot be used after that date and time thereby ensuring *S3*.

The PPIdP and the proxy IdP generates a pseudonymous partial identifier unique to each SP during the authentication phase allowing users to maintain a session with a SP, however, several SPs cannot collude to build a profile of the user. The proxy IdP can also provide a unique identifier. This satisfies *P1*. During the authentication process, the user can choose, using the consent form, which attributes she wants to release to that SP which users to select attributes and contextual information, control data flow and to provide her consent while accessing a service and hence satisfies *P2* and *P3*.

Many services might not require any identifying information, providing the location information would be enough to access such services as in the case of *Page1.php & Page2.php*. When the location information as well as an identifier is required from a trusted source to offer services (based on the Role Based Access Control (RBAC) [40] or Attribute Based

Access Control (ABAC) [41]), as in the case of *Page3.php*, we have adopted the attribute aggregation mechanism. The *Page3.php* use-case has only illustrated the RBAC scenarios (e.g. if the role of the user is student). However, it can be easily adopted for scenarios requiring the user's identifier (ABAC).

Discussion of any federated system will be incomplete without analysing the trust issues involved. As we have federated the proxy IdP and the SP in the traditional way, they trust each other whereas the PPIdP is considered as semitrusted to the IdP and the SP. If the authentication is delegated by the proxy IdP to the other IdPs (as in the case of CAFS Type 2), the SP depends on the judgement of the proxy IdP. That is, if the proxy IdP considers other IdPs as trusted then they will be trusted by the SP and if other IdPs are considered as semitrusted then they will be considered as semitrusted as well. As mentioned earlier, the LoA value is used as a trust metric. For the CAFS Type 1 architecture, any attributes released by the PPIdP will be implicitly considered having a LoA value of 1. For the CAFS Type 2 architecture, the proxy IdP holds the responsibility to assign the correct LoA value.

A. Advantages

The CAFS provides several key advantages. A few of them are highlighted below:

– CAFS is the first framework to address the context-aware identity management in a comprehensive way focusing all aspects of identity management and satisfying all requirements as outlined in Section V.

– CAFS is the first framework to demonstrate how location data can be used to provide federated services. This brings benefits to both users and SPs as they can utilise the SSO capability of the IdP thereby reducing the need for further logins for each separate location-based service hosted in different identity domains.

– The CAFS illustrates how geographical location data can be utilised by integrating the GeoXACML API with the SAML and the existing XACML libraries. In addition, our deployment shows how the user can be located precisely inside a building and how the authenticity of the location can be verified using external sensors and the readily available sensing technologies of smartphones.

– In addition, this is the first work to explore the possibility of combining the attribute aggregation mechanism with the context-aware services to formulate innovative and advanced use-case scenarios.

Table II provides a comparative analysis of our framework with existing works on context-aware services against the set of requirements compiled in Section V. The table provides a side-by-side comparison between our work and other existing works. The "✓" symbol has been used in the table to signify that the particular work has considered the respective requirement. The "x " symbol has been used to signify that either the system has failed to meet that respective requirement. As evident from the table, our framework offers a better way to access context-aware services than any existing works considering different functional, security, privacy and trust issues.

B. Future Work

There are a few directions to take from here:

– There are a few XACML Policy Editors such as UMU-XACML-Editor [42], Islandora:XACML Editor 6.x [43], etc., none of them has any support to add geographic information into the policy. The geographic information might contain numerous geographic points which are hard to add manually in the policy. It will be useful to add support for adding such information in those editors.

– The current implementation of attribute aggregation is based on the IP model. We are investigating how other models can be accommodated into our implementation.

IX. Conclusion

As the current trend suggests, the popularity of federated services and context-aware services will be increasing in the upcoming years. An architecture that can provide context-aware federated services can be advantageous for users due to the SSO capability and thus reducing the number of online accounts they need to manage and use to access different services. Such services can also be advantageous for SPs as well since it allow them to offload the burden of managing user information by delegating these tasks to the IdP. In this paper we present a novel framework, based on the concept of Portable Personal Identity Provider and Personal Identity Federation, for context-aware federated services using the SAML, XACML and GeoXACML standards. We have also shown how the attribute aggregation mechanism can play a crucial role to provide innovative use-case scenarios. We believe that our approach has huge potential. However, this is just an introductory step toward this exciting area and we hope that this work will excite further research in enabling users to harness the full potential of context-aware services.

References

[1] D. W. Chadwick, "Federated Identity Management," in *FOSAD 2008/2009*, ser. LNCS, A. Aldini, G. Barthe, and R. Gorrieri, Eds. Berlin: Springer-Verlag, January 2009, no. 5705, p. 96-120. [Online]. Available: http://www.cs.kent.ac.uk/pubs/2009/3030

[2] Marc Boroditsky, "Achieving Context-Aware Security with Integrated Identity Management," 2011, http://www.oracle.com/openworld/lad-en/session-presentations/middleware/14903-enok-1440256.pdf.

[3] A. Jøsang, M. Al, and Z. S. Suriadi, "Usability and privacy in identity management architectures," in *ACSW '07: Proceedings of the fifth Australasian symposium on ACSW frontiers'*, 2007, p. 143-152.

[4] K. Wrona and L. Gomez, "Context-aware security and secure context-awareness in ubiquitous computing environments," in *In the Proceedings of the XXI Autumn Meeting of Polish Information Processing Society*, 2005, p. 255-265. [Online]. Available: http://proceedings2005.imcsit.org/docs/75.pdf

[5] N. Klingenstein, "Attribute Aggregation and Federated Identity," in *Applications and the Internet Workshops, 2007. SAINT Workshops 2007. International Symposium on*, 2007, p. 26-26.

[6] E. Hayashi, S. Das, S. Amini, J. Hong, and I. Oakley, "CASA: Context-aware Scalable Authentication," in *Proceedings of the Ninth Symposium on Usable Privacy and Security*, ser. SOUPS '13. New York, NY, USA: ACM, 2013, p. 3:1-3:10. [Online]. Available: http://doi.acm.org/10.1145/2501604.2501607

TABLE II: Comparison of CAFS with existing works.

Name	FR					SR			FR		
	F1	F2	F3	F4	F5	S1	S2	S3	P1	P2	P3
Hayashi et al. [6]	x	x	✓	x	x	x	✓	x	x	x	x
Jansen et al. [7]	x	x	✓	x	x	✓	✓	x	x	x	x
Malek et al. [8]	x	x	✓	x	x	✓	✓	x	x	x	x
Hulsebosch [10]	✓	x	✓	x	x	✓	x	x	✓	x	x
Jean-Yves et al. [11]	✓	x	✓	x	x	✓	✓	x	x	x	x
Nishiki et al. [12]	✓	✓	✓	x	x	✓	✓	x	x	x	x
Goel et al. [13]	✓	x	✓	x	✓	x	✓	✓	x	x	x
CAFS	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

[7] W. A. Jansen, S. I. Gavrila, and V. Korolev, "Proximity-Based Authentication for Mobile Devices," in *Security and Management*, 2005, p. 398-404.

[8] B. Malek, A. Miri, and A. Karmouch, "A framework for context-aware authentication," in *Intelligent Environments, 2008 IET 4th International Conference on*, 2008, p. 1-8.

[9] M. Moyer and M. Abamad, "Generalized role-based access control," in *Distributed Computing Systems, 2001. 21st International Conference on.*, 2001, p. 391-398.

[10] R. J. Hulsebosch, A. H. Salden, M. S. Bargh, P. W. G. Ebben, and J. Reitsma, "Context sensitive access control," in *Proceedings of the tenth ACM symposium on Access control models and technologies*, ser. SACMAT '05. New York, NY, USA: ACM, 2005, p. 111-119. [Online]. Available: http://doi.acm.org/10.1145/1063979.1064000

[11] J.-Y. Tigli, S. Lavirotte, G. Rey, V. Hourdin, and M. Riveill, "Context-aware Authorization in Highly Dynamic Environments," *CoRR*, vol. abs/1102.5194, 2011.

[12] K. Nishiki and E. Tanaka, "Authentication and Access Control Agent Framework for Context-Aware Services," in *Applications and the Internet Workshops, 2005. Saint Workshops 2005. The 2005 Symposium on*, 2005, p. 200-203.

[13] D. Goel, E. Kher, S. Joag, V. Mujumdar, M. Griss, and A. Dey, "Context-Aware Authentication Framework," in *Mobile Computing, Applications, and Services*, ser. Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, T. Phan, R. Montanari, and P. Zerfos, Eds. Springer Berlin Heidelberg, 2010, vol. 35, p. 26-41. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-12607-9_3

[14] A. S. Q. Arabo and M. Merabti, "A Framework for User-Centred and Context-Aware Identity Management in Mobile Ad Hoc Networks (UCIM) ," *Ubiquitous Computing and Communication Journal*, June 9, 2009. [Online]. Available: http://ssrn.com/abstract=1970333

[15] Bob Hulsebosch, Maarten Wegdam, Bas Zoetekouw, Niels van Dijk, Remco Poortinga - van Wijnen, "Virtual collaboration attribute management," Accessed on 1 May, 2013, 2011, http://www.surfnet.nl/nl/Innovatieprogramma's/gigaport3/Documents/EDS%2011-06%20AttributeManagement%20v1.0.pdf.

[16] D. Chadwick and G. Inman, "Attribute aggregation in federated identity management," *Computer*, vol. 42, no. 5, p. 33-40, 2009.

[17] M. Sadek Ferdous, M. Jabed, M. Chowdhury, M. Moniruzzaman, and F. Chowdhury, "Identity federations: A new perspective for Bangladesh," in *Informatics, Electronics Vision (ICIEV), 2012 International Conference on*, may 2012, p. 219-224.

[18] B. Schilit and M. Theimer, "Disseminating active map information to mobile hosts," *Network, IEEE*, vol. 8, no. 5, p. 22-32, 1994.

[19] R. Nick, J. Pascoe, and D. Morse, "Enhanced reality fieldwork: the context-aware archaeologist assistant," in *Computer Applications & Quantitative Methods in Archaeology*, Exon, Ed., vol. 0. Archaeopress, 1997.

[20] G. D. Abowd, A. K. Dey, P. J. Brown, N. Davies, M. Smith, and P. Steggles, "Towards a Better Understanding of Context and Context-Awareness," in *Proceedings of the 1st international symposium on Handheld and Ubiquitous Computing*, ser. HUC '99. London, UK, UK: Springer-Verlag, 1999, p. 304-307. [Online]. Available: http://dl.acm.org/citation.cfm?id=647985.743843

[21] M. Ferdous and R. Poet, "A comparative analysis of Identity Management Systems," in *High Performance Computing and Simulation (HPCS), 2012 International Conference on*, july 2012, p. 454-461.

[22] M. S. Ferdous and R. Poet, "Portable Personal Identity Provider in Mobile Phones," in *Proceedings of the 2013 12th IEEE International Conference on Trust, Security and Privacy in Computing and Communications*, ser. TRUSTCOM '13, 2013, p. 736-745. [Online]. Available: http://dx.doi.org/10.1109/TrustCom.2013.89

[23] M. Ferdous and R. Poet, "Dynamic Identity Federation Using Security Assertion Markup Language (SAML)," in *Policies and Research in Identity Management*, ser. IFIP Advances in Information and Communication Technology, S. Fischer-Hbner, E. Leeuw, and C. Mitchell, Eds., vol. 396. Springer Berlin Heidelberg, 2013, p. 131-146. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-37282-7_13

[24] "A Brief Introduction to XACML," 14 March, 2003, http://www.oasis-open.org/committees/download.php/2713/Brief_Introduction_to_XACML.html.

[25] O. G. Consortium, "Geospatial eXtensible Access Control Markup Language (GeoXACML), Version 1 Corrigendum," 12 May, 2011, http://www.opengeospatial.org/standards/geoxacml.

[26] ——, "OpenGIS Geography Markup Language (GML) Encoding Standard, Version: 3.2.1," 27 August, 2007, http://www.opengeospatial.org/standards/gml.

[27] "SQLite Database," http://www.sqlite.org/.

[28] "SQLCipher," http://sqlcipher.net/.

[29] "Android Location Strategies," http://developer.android.com/guide/topics/location/strategies.html.

[30] "Jetty WebServer," http://jetty.codehaus.org/jetty/.

[31] "Sun's XACML Implementation," Accessed on 1 March, 2013, http://sunxacml.sourceforge.net/.

[32] "HERASAF XACML Core," Accessed on 1 March, 2013, http://www.herasaf.org/downloads/binaries.html.

[33] "enterprise-java-xacml," Accessed on 1 March, 2013, https://code.google.com/p/enterprise-java-xacml/.

[34] "GeoServer SVN Repository," Accessed on 2 May, 2013, https://github.com/opengeo/geoserver-2.1.x/tree/master/community/geoxacml.

[35] "SimpleSAMLphp," http://simplesamlphp.org/.

[36] NISTWP, "Electronic Authentication Guideline: INFORMATION SECURITY," April 2006, http://csrc.nist.gov/publications/nistpubs/800-63/SP800-63V1_0_2.pdf.

[37] "Wikipedia entry on QRCode," Accessed on 1 April, 2013, http://en.wikipedia.org/wiki/QR_code.

[38] "Fake GPS Location," https://play.google.com/store/apps/details?id=com.lexa.fakegps&hl=en.

[39] "Location Spoofer," https://play.google.com/store/apps/details?id=org.ajeje.fakelocation&hl=en.

[40] R. S. Sandhu, "Role-based access control," *Advances in computers*, vol. 46, p. 237-286, 1998.

[41] E. Yuan and J. Tong, "Attributed based access control (ABAC) for Web services," in *Web Services, 2005. ICWS 2005. Proceedings. 2005 IEEE International Conference on*, July 2005, p. -569.

[42] "UMU-XACML-Editor," http://sourceforge.net/projects/umu-xacmleditor/.

[43] "Islandora:XACML Editor 6.x," https://jira.duraspace.org/browse/ISLANDORA/component/10402.