Preserving Privacy with Fine-grained Authorization in an Identity Management System

Gerson Luiz Camillo[*] Carla Merkle Westphall[†], Jorge Werner[‡], Carlos Becker Westphall[§]

Post-graduate Program in Computer Science (PPGCC)
Networks and Management Laboratory (LRG-UFSC)
Federal University of Santa Catarina - UFSC
P.O. Box 476, 88040-970, Florianópolis, SC, Brazil
Email: [*]gerson.camillo@posgrad.ufsc.br [†]carla.merkle.westphall@ufsc.br
[‡]j.werner@posgrad.ufsc.br [§]westphall@lrg.ufsc.br

Abstract—**In policy-based management, service providers want to enforce fine-grained policies for their resources and services. Besides the assurance of digital identity, service providers usually need personal data for evaluation of access control policies. The disclosure of personal data, also known as Personally Identifiable Information (PII), could represent a privacy breach. This paper proposes an architecture that allows an individual to obtain services without the need of releasing all personal attributes. The architecture achieves that outcome evaluating the targeted policy in the domain of the identity provider, that is, policies are sent from service providers to identity providers to be evaluated, without the need of releasing some PIIs to the service provider side. We also present an implementation of a prototype using XACML 3.0 for fine-grained authorization and OpenID Connect for identity management. The prototype was evaluated through an use case representing an hypothetical scenario of a bookstore. The project demonstrated that for certain situations an user can restrict the release of PII data and still gain access to services.**

Keywords–Privacy; Identity Management; OpenID Connect; Fine-grained Authorization; XACML.

I. Introduction

The data that identifies and distinguishes the users have acquired invaluable importance in the digital society, to the extent that any online transaction usually requires some information to be disclosed. These data are known as Personally Identifiable Information (PII) and they are represented by attributes.

The risks to personal information in service providers (SP) are totally related to the amount of collected attributes of individuals [1]. In addition, the personal identification data can be collected by the service providers to identify users and create profiles for business. Many Internet companies grew up selling personally identifiable information and behavioral data. These two situations represent attacks on the privacy of individuals and the risks increase with the amount of personal attributes in the SP.

Thus, privacy aims to minimize the release of personal information and/or prevent that attributes are linked to the user [2][3][4]. Privacy can be achieved by law, techniques, and mechanisms, aiming to empower the individuals in controlling their personal information. This work presents a technique that aims to minimize the disclosure of PII data.

The access control is a central security point and the authorization systems evolved from identity-based to attribute-based. The attribute is an assertion describing a quality, state, appearance, and characteristic of some entity in the context of authorization. There can be attributes of the subject, resource, action, and environment. The Attribute-Based Access Control (ABAC) model [5] is a formalization of the requirements for an attribute-based authorization. The ABAC evaluates rules and policies against the attributes of the entities (subjects, resources, actions, and environment). The model is characterized as policy-based authorization, because the logic of access control is represented by rules that compose policies.

The architecture of ABAC is constituted by functional points, which were already defined in [6]. The Organization for the Advancement of Structured Information Standards (OASIS) specified the *eXtensible Access Control Markup Language* (XACML) [7] as an implementation for the ABAC model and for the authorization framework [6]. The XACML is a policy language for fine-grained authorization which provides a request-response protocol and a reference architecture. The functionality of the model starts with the request that arrives at the Policy Enforcement Point (PEP), which acts protecting the resource. The PEP receives user's request and asks the Policy Decision Point (PDP) for an access control decision. The PDP evaluates the policy that matches the request and returns a decision to PEP for enforcement. Also, there is the Policy Administration Point (PAP), which manages the repository of policies and the Policy Information Point (PIP), which searches for the attributes that are not present in the request.

Service providers (SP) need user's attributes to enforce fine-grained policies and to perform appropriate authorization decisions. One solution for the SP is to use the authentication token to get attributes from an identity management system (IdM). An IdM is the process and technology that enables the creation, management, use, and removal of digital identities. Digital identities are electronic representations of the real identities and can be characterized by a subset of values of attributes [8]. Thus, IdM systems were created in order to maintain PII data in an identity provider (IdP) and to securely transport attributes and identity assertions among different parties.

In this paper, we will present the scenario of a bookstore to explain the problem to be solved. The Web service of the bookstore sells materials online but the company is seeking to include a competitive edge in the field of user privacy. The bookstore has included the possibility to view books online, but some of them with restricted access. Firstly, users have to login in the IdP and then the SP will evaluate XACML policies against user's attributes to generate an access decision. Consequently, the SP needs to obtain the personal data from the user maintained by the IdP. However, this situation creates a privacy risk to the individual because the bookstore could increase the amount of collected personal attributes. This

ISBN: 978-1-61208-546-3

problem led us to propose an architecture to preserve user's privacy.

The proposal of this paper is an approach that maintains the SP needs for fine-grained authorization while protecting user's privacy. To ensure privacy of users, the architecture will evaluate the service policy in the domain of the IdP. Thus, the complete set of personal attributes are not conveyed from the IdP to the SP to evaluate the policies. The trust relationship that enables the SP to rely on assertions from IdP is used by the architecture to obtain an access control decision from the same IdP. The development of our architecture is based on recent protocols and specifications: OAuth 2.0 [9], OpenID Connect (OIDC) [10], and RESTful Web services. In addition, the SP applies fine-grained authorization using XACML architecture and policies.

One of the main contribution of this work is the introduction of an architecture that evaluates attribute-based access control policies in the IdP side, returning to the SP only the result of the evaluation, aiming to prevent the service provider from obtaining private user data. The other contribution is the enforcement of fine-grained access control policies using XACML by the SP while keeping user's privacy regarding PII. The proposal and development of a prototype to test a use case scenario can also be considered a contribution of this work.

The remaining of this paper is arranged as follows: Section II presents the related work; the problem statement is in Section III; in Section IV, the proposed architecture is presented; the Section V describes the implementation of a prototype and the results from the test case; Section VI discusses the findings; and Section VII sums up the text.

II. Related Work

Different works and Privacy Enhancing Technologies (PETs) have the purpose of increasing or establishing privacy in the relationship between users and service providers in IdM environment. This section restricts the descriptions of the works that aim to provide privacy of personal data, as defined in EU Directive 95/46/EC [11]. The directive defines personal data as a piece of information that identifies directly or indirectly a natural person.

The Privacy-preserving Attribute-based Credentials (Privacy-ABC) is an approach to authentication with private credentials in IdM scenarios, which provides user privacy. The Privacy-ABC are technologies which enable users to obtain credentials and derive unlinkable tokens that reveal only a subset of attributes. They are based on cryptographic primitives, and two examples of them are the schemes of Brand [12] and Camenisch-Lysyanskaya [13].

The Privacy-ABC were developed in the European projects PRIME [14] and PRIMELife [15]. IBM Identity Mixer (Idemix) [16][17] and the Microsoft U-Prove [18] are commercial deployments based on Privacy-ABC. Those technologies do not have a widespread use, owing to the fact that the areas of user interface, policy languages, and infrastructure need further research [19][20]. In addition, the Privacy-ABC have difficult understanding and use [21]. The ABC4Trust [22] project was created to overcome some of those technical issues.

The User-Managed Access (UMA) [23] is a profile of OAuth 2.0 and its principal aim is to enable users to manage the policies of their protected resources (personal data, content, and services). The users are central in UMA, however, they may be confronted with complex policies in Web scenarios, that require complicated authorization choices and could negatively affect the user's privacy decisions.

Chadwick and Fatema [24] proposed an architecture that aims to provide authorization services in cloud infrastructure. Privacy is addressed by the use of *sticky* policies that consist of privacy policies that are attached to the data. The premise was that the SPs in the cloud are reliable in such a way that they will honor the privacy policies defined in *sticky* policies.

Architectures for policy decomposition [25] and policy federation [26][27] aimed to provide confidentiality and privacy when enforcing access control policies in distributed environments. The proposed works are supported by the XACML architecture because the entities of XACML are specified to be easily distributed. The entities use SOAP/SAML protocols to convey the request/response messages and the policies, all defined in XACML specification [7]. Despite the privacy achieved in some scenarios, the models do not explicitly include user authentication through identity management.

The Shibboleth 2.0 [28] is a well-known example of implementation of the Security Assertion Markup Language (SAML) protocol [29] for IdM. However, some characteristics of Shibboleth 2.0 limit its use in our architecture: the set of attributes are predefined between the SP and IdP and there is no consent mechanism (this was included natively in IdPv3). Thus, the SAML/Shibboleth 2.0 has a difficult integration with RESTful Web API and mobile applications. The OpenID Connect (OIDC) [10] is a recent specification for IdM and was developed on the top of OAuth 2.0. The main advantages are the use of RESTful Web APIs and the transport of data through Javascript Object Notation (JSON) format. OIDC is a natural choice for identity management in Web 2.0 environments.

Werner and Westphall [30] defined a model for an IdM with privacy in cloud infrastructure. Even though the authors have presented a model that tries to help users to make decisions about their privacy, the architecture still depends on SP to enforce the privacy policy.

Ma and Sartipi [31] proposed an infrastructure that integrates the OIDC to XACML for sharing diagnostic images in cloud deployments. Their solution transfers to the end user the management of policies, which could be administrative burden when users have data in different types of services.

III. Problem Statement

The main problem that this work aims to solve is the amount of PII data released in IdM scenarios. The proposed solution transfers the policy from the SP to the IdP. For this work, the words Service Provider (SP), Relying Party (RP), and Client have the same functional definition.

The externalization and distribution of policy evaluation have been studied before [26][27]. Those references adopted the XACML for the architecture and for the policy language. The XACML and the ABAC were defined for distributed environments [5], but considered in a single domain of security. This work proposes the inclusion of a PDP in an IdP domain to evaluate policies that need end-users attributes. However, the approach is unusual when considering the IdM scenarios. The proposed architecture uses the trust agreement created to support a federated identity management to federalize the authorization concerning PII data.

The architecture proposed can be defined as a PET solution. The taxonomy of PETs [3] defined the aspect of privacy that is targeted by PET, and that can be the *identity*, the *content*, or the *behavior*. The proposal of this paper aims to protect the data that represents the identity of the user stored in an IdP. The architecture does not include mechanisms to protect the content of data that are created, stored, and manipulated during service interaction. The aspect of behavior is related to access pattern and it is obtained by correlating actions with identities. The proposed architecture only can guarantee such aspect if the underlying IdM provides transient pseudonyms identifiers or anonymity.

The following set of trust relationships were assumed for this work: the IdP is trustworthy for management of end user attributes; the RP is untrustworthy, which follows the protocol but wants more information than is really necessary; and, the RP relies on the IdP to provide the identity claims about the end user.

The previous definition leads to the configuration of the architecture in security domains. There is the domain of the SP and the domain of the IdP. The classification is regarding to the protection of the PII data. The IdM technologies adopted the concept of minimization of data releasing only the attributes required for the purpose of the service. The trust relationship between IdP and RP includes agreement on what attributes of IdP are needed to what services of SP during transactions related to identity and authorization management. The trust agreement can be static or dynamic. Static agreements are used by IdMs based on SAML. In that type of IdM, the user has little or no control about the personal data released to the SP.

On the other hand, recent specifications of architectures and protocols for authorization and identity are more dynamic. They define the user as the central entity for controlling data access and the main mechanism is the consent management [11][1]. Examples of systems that include user's consent: OAuth 2.0, OIDC, UMA [32], Shibboleth IdPv3 [33]. This demonstrates that consent is relevant in IdM scenarios and it is why this proposal can be considered for increasing the privacy of personal data.

IV. Proposed Architecture for Privacy Preserving User Attributes

The architecture proposed here includes elements and flows in a network-based IdM. This type of IdM provides the functionality of Web authentication, known as Web Single Sign-on (SSO). The proposed architecture is depicted in Fig. 1. The elements of ABAC model are included in the RP and in the IdP. The ABAC functional points provide the following features: evaluation of fine-grained policies; request/response authorizations; and, distribution of the functional points. Those characteristics enable the creation of a loosely-coupled architecture for authorization and a means to convey the policies to the IdP.

Fig. 1 shows that there are the domain of RP and the domain of IdP. The end user trusts the IdP to be the provider of personal data (attributes). The RP trusts the IdP for end-user authentication. This trust relationship can be statically agreed upon or dynamically created. The dynamic mechanism occurs through some form of discovery and metadata exchange for registration. The elements included in the architecture are: PDP and PAP in RP domain; and, PDP in IdP domain. The PEP in RP must enforce the result of PDP evaluation of policies managed by the PAP. The PAP stores the authorization policies for the RP. This scheme defines an externalized architecture of authorization.

The inclusion of the PDP and PAP points for authorization purposes are common scenarios in ABAC models. However, inclusion of a PDP point in IdP is a novel proposal. This creates another point of policy evaluation in the domain of IdP. In ABAC model, the same policy that is evaluated by the PDP in RP domain can be evaluated in the domain of IdP, because of its distributed architecture. If the service policy requires user attributes, then the policy can be conveyed to the IdP domain for evaluation. This approach eliminates the release of personal data from IdP to RP domain.

The flows highlighted in Fig. 1 will be now described. The steps 1 to 4 are related to the process of authentication (Web SSO). An end user through user agent (Web browser) requests services from RP (step 1). The RP redirects the end user to IdP via Web browser (step 2). The end user is authenticated by IdP (step 3) and the IdP generates a token that is redirected to RP (step 4). The token is an authentication assertion and the token corresponds to the user credentials.

The next phase (step 5) only occurs when end users have decided to release personal attributes to the RP through the consent dialog. The RP uses the token obtained in the phase of authentication to get those user released attributes. Those attributes can be used by the RP to enrich the user experience on the Web and create authorization with fine-grained controls. However, this consent phase can also increase the risks to the privacy of the user because the risk is directly related to the amount of personal data transferred to the RP.

The next steps are concerned with the description of the contribution of this proposal. The RP implements the ABAC model to protect resources using fine-grained policies. The end user's demand is captured by the PEP which generates a XACML request with the available attributes. The PEP sends the XACML request to the PDP for an access decision (step 6). The PDP chooses the applicable policy based on the attributes of different categories: subject, object, action, and environment. If the end user have denied access to the subject attributes, the PDP cannot evaluate the applicable policy to the XACML request and thus the PDP returns the "Indeterminate" response. Besides, the PDP includes the status of missing attributes in response. With that outcome, the PEP could deny the user's request to resources. However, in the proposed architecture, this lack of necessary attributes starts the phase of the evaluation of the RP policy in the IdP domain.

The RP discovers that the IdP can evaluate XACML policies when the IdP announces the PDP endpoints through

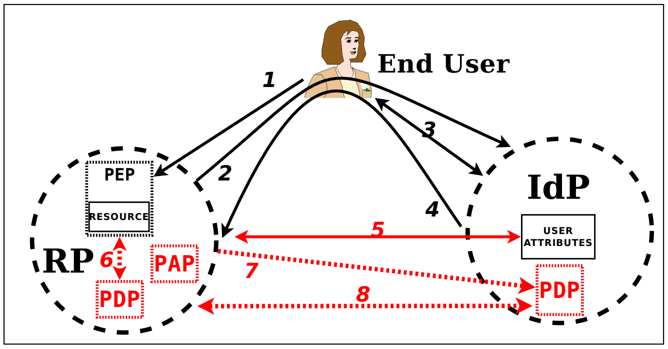


Figure 1. Proposed architecture. Our contribution is highlighted and comprises PDP/PAP points and steps 6, 7 and 8 for conveying policies.

metadata information. The implementation of the XACML standard is not specified because the policy-based XACML language does not rely on the technical engine that will run it. In consequence, the architecture is ready to evaluate the authorization policy in IdP domain.

The response from PDP also contains the identification of the target policy. The PEP uses that information to retrieve the policy from the PAP and then sends it to the endpoint of the PDP in the IdP (step 7). The PEP sends the same XACML request to the PDP at IdP for evaluation (step 8). The PDP starts the evaluation of the policy against the attributes that come from the following sources: XACML request for the resource, action, and environment attributes; and, the subject attributes from the IdP. The subject attributes refer to the attributes about end users at IdP. As the PDP figures out an authorization decision, it is returned to the PEP in the domain of RP for enforcement (step 8).

V. Prototype Results

The prototype consists of an IdM and the elements of the ABAC authorization model. Fig. 2 shows the entities grouped in security domains along with the principal steps. The steps may represent one or more flows of interaction among the entities. The diagram also depicts which flows are related to OIDC, XACML, and those provided by the contribution of this work.

The domain of RP was built with the following elements: PEPClientApp, PDP-rp, and PAP. The PEPClientApp is the application that owns and protects the access to resources or services through the PEP. It intercepts the end-user request and generates a XACML request for the PDP-rp to obtain an access control decision. The PEPClientApp was constructed using base code of a sample application which is included in the MITREid project. It uses the Spring Framework to provide the security elements for protection of the services. The PDP-rp evaluates the XACML request against the policies, which protect the services and resources. The PAP manages the access control policies for the RP.

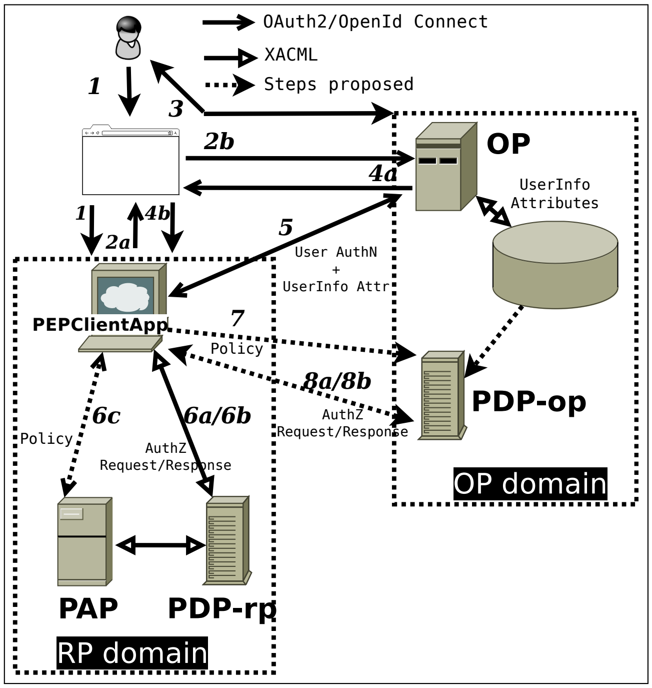


Figure 2. Prototype of the proposed architecture.

In the security domain of OP, there were the following elements: OP, PDP-op, and *UserInfo*. The OP is the IdP provider, which will authenticate the end user and will provide claims about the user to the RP. The MITREid Connect [34] was defined for the IdP because it is an implementation of the OIDC standard. The PDP-op is the PDP point of the XACML architecture that evaluates the RP policies that are sent by the PEPClientApp. The *UserInfo* is the repository of end-user attributes, which are stored in a database, that enables both the OP and the PDP-op to retrieve attribute information.

The OpenAZ [35] is a reference implementation of the XACML 3.0 standard and it was chosen because it supports REST interfaces and JSON messages for communication among PDP, PAP, and PEP. This REST support makes it more easy to distribute the XACML points as RESTful Web services. It also enabled the integration of the XACML with the OIDC. The OpenAZ, MITREid Connect, and PEPClientApp are all open-source software based on the Java language, and they performed on the Tomcat Web server.

A. *Test Case*

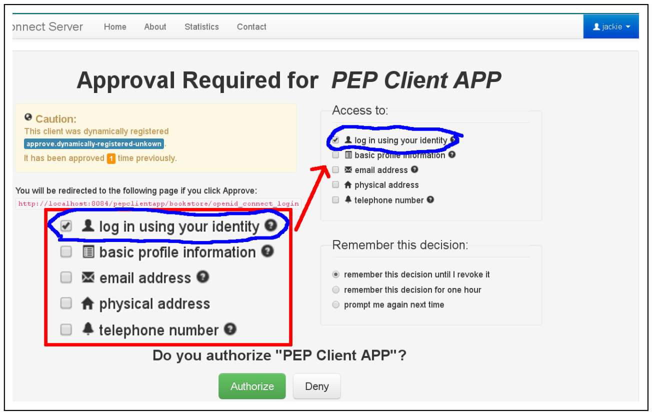
The scenario used for this test case was based on an online bookstore that sells books and offers some other services. One service allows users to view and read books online from their catalog. However, there are titles that need different types of authorization because they are restricted material.

The bookstore adheres to an IdM and obtains the authentication result from an identity provider (IdP). The authorization system needs the following characteristics: policy-based, fine-grained controls, and dynamic management of access controls. In addition, the outsourced authentication and authorization need to adopt principles of RESTful architecture style. These requirements are complied by XACML 3.0 standard for fine-grained authorization and OpenID Connect for identity management.

The following fine-grained policy was defined to assess the use case and that was identified as P1: users authenticated by an IdP and whose residence is in either of Japan, China, or South Korea can view online books restricted by locality. Besides, the books are only available between December 1st and December 31, 2016. The policy is expressed in the XACML 3.0 language, which is deployed in PAP. It will protect the resources at RP domain besides other policies.

The steps in the test are described below. First, it was assumed that the end user “Jackie” was registered in the OP. And similarly, the RP identified as PEPClientApp was already registered as a client application in the same OP. The steps 1 to 4 are related to the phase of authentication of the end user to the OP. After that, a page of consent was presented to him.

The consent phase is shown in Fig. 3. It sets up the user’s decision about his privacy. It is where the user has the power of choice on the release of personal attributes to the RP domain. In our example, the end user “Jackie” authorized the PEPClientApp to access resources on his behalf. However, he does not want to share his personal information with the RP. Thus, “Jackie” just chose to release the *sub* claim to RP, clicking in the option “login using your identity”. The claim *sub* identifies the end user at the issuer (OP) and it is included in the assertion that OP sends to the RP.



Approval Required for *PEP Client APP*

Caution:
This client was dynamically registered approve.dynamically-registered-unkown.
It has been approved 1 time previously.

You will be redirected to the following page if you click Approve:
http://localhost:8084/pepclientapp/bookstore/openid_connect_login

☑ log in using your identity
□ basic profile information
□ email address
□ physical address
□ telephone number

Access to:
☑ log in using your identity
□ basic profile information
□ email address
□ physical address
□ telephone number

Remember this decision:
● remember this decision until I revoke it
○ remember this decision for one hour
○ prompt me again next time

Do you authorize "PEP Client APP"?

Figure 3. Screen of consent at OP.

The PEPClientApp generated a XACML request with the attributes available at RP. The request contains the attributes from resource, action, environment, and subject. The attributes of the subject contained only the identifier of OP and the identifier of subject (“subject-id”). The PEPClientApp sent the XACML request to the PDP-rp for an access control decision (step 6a). The PDP-rp evaluated the request against the applicable P1 XACML policy to renders an authorization decision. The main rule of P1 policy contains the element *Condition* that has two set of functions. The parameters are the attributes *country* and *current-dateTime*. However, the value of the attribute *country* was missing and was not available for evaluation by the PDP-rp. Thus, the PDP-rp returned the response “Indeterminate” with the status “missing-attribute” (step 6b). That resulted in a new authorization step.

The next phase is where the proposal of preserving privacy is included in the architecture. The XACML response included the identification of the policy and in sequence the PEPClientApp obtained it from PAP (step 6c). Considering that the targeted policy was sent to the cache of PDP-op (step 7), the PEPClientApp performed the same request to the RESTful endpoint of PDP-op (step 8a).

The PDP-op evaluated the request against the policy and arrived in the XACML *Condition*, which contained a function that needed the attribute *country*. Thus, the PDP-op consulted the PIP through Context Handler for the missing attribute. The PIP obtained a value for the attribute *country* making a query using the end user identified by *sub* in the *UserInfo* database. Then, the PDP-op evaluated the rule and arrived a decision. The response containing the decision “Permit” was returned to the PEPClientApp for enforcement (step 8b).

VI. Discussion

The results from the test case (Section V-A) showed that the architecture allowed an end user to access the protected resources without releasing personal attributes to the RP. In the test case, the end user “Jackie” has not released the attribute *country* to the RP. However, the PDP-op arrived at decision “Permit” because it obtained the value of attribute *country* from the OP domain. Summarizing, an architecture was proposed in this paper that transferred the authorization service from RP to the OP and that achieved the outcome of avoiding to release personal information to RP.

The prototype presented low complexity to implement the proposed architecture. Besides, there was no need to change flows and specifications of the OIDC and XACML. In contrast, the works and systems [14]-[20] that use private credentials have to deal with the complexity of the Public Key Infrastructure (PKI) and with questions related to integration, data formats, and user interfaces.

Organizations that collect and store PII data should establish security controls to provide the confidentiality of this information [36][37]. This represents an administrative and operational costs for those companies. However, when an organization can provide service without the need for personal attributes, it can minimize these costs. This is a benefit that can be achieved with the use of the proposed architecture.

The test case also demonstrated the usability of the prototype, because the end users did not need to establish privacy policies to manipulate their personal data. The users only needed to deny the release of attributes to protect their PII data. The outcome is that the SPs can modify their authorization logic without updating them in the agreement with the IdP. The proposals [23][24][30][31], which depend on the user’s ability to define policies, may create risks to privacy of PII due to an increase in management complexity.

The aspects of confidentially, integrity, threats and security risks are directly related to the measures adopted when implementing the IdM infrastructure. If the architecture uses the OpenID Connect for IdM, as in the prototype, the security measures are those specified in the [10] and in the [38]. The [38] presents the threat model and security considerations when implementing systems and protocols that use the underlying protocol OAuth 2.0.

There is a potential limitation in the confidentiality concerning service provider policies in our work. There is a need of security mechanisms to protect the policy when it leaves the domain of RP, because it may contain sensitive information about the service provider. This problem can be minimized considering the trust relationship established between the OP and RP.

There is another issue that needs to be considered. The privacy feature of anonymity depends on the IdM system used in the architecture. Pseudonymity and anonymity can be obtained in OIDC by the use of Pairwise Pseudonymous Identifier (PPID) [10] for the value of *sub* claim. PPID is an identifier that identifies the end user to an RP that cannot be correlated with the end-user PPID at another RP. The PPID can be used in OIDC without any problem in the proposed architecture.

There is a performance limitation regarding the authorization actions. As the architecture included steps to evaluate the policy in the domain of OP, the decision time increases. Moreover, there are concerns regarding the runtime of the XACML policies. However, there are works [39][40] that aim to optimize PDP performance. In addition, the mechanism of caching can be used for the PDP and PAP points of the architecture. The question of performance can be considered a valid trade-off between user privacy and the performance penalty to get an authorization decision. The end user can assume the performance impact considering that the request can be denied in the absence of the proposed architecture.

VII. Conclusion and Future Work

The proposed architecture presents a new way of obtaining privacy to users, when dealing with fine-grained resource permissions. The SP policies are carried out and assessed in the domain of the IdP to avoid the release of personal attributes to SP domain. This approach allows users to deny the release of personal data to SP while getting a decision for accessing resources or services. The outcome of the architecture is the minimization of use, collection, and retention of personal data (PII), that attends the principle of collection limitation from OECD privacy guideline. Future work might go towards research on the inclusion of the decomposition of policies to protect the confidentiality of some elements of the policy.

References

[1] K. Cameron. The laws of identity. [retrieved: March, 2017]. [Online]. Available: http://myinstantid.com/laws.pdf

[2] S. Gürses, C. Troncoso, and C. Diaz. (2011) Engineering privacy by design. [retrieved: March, 2017]. [Online]. Available: http://www.cosic.esat.kuleuven.be/publications/article-1542.pdf

[3] J. Heurix, P. Zimmermann, T. Neubauer, and S. Fenz, "A taxonomy for privacy enhancing technologies," *Computers & Security*, vol. 53, pp. 1–17, 2015.

[4] C. Landwehr *et al.*, "Privacy and cybersecurity: The next 100 years," *Proceedings of the IEEE*, vol. 100, no. Special Centennial Issue, pp. 1659–1673, May 2012.

[5] V. C. Hu *et al.*, "Guide to Attribute Based Access Control (ABAC) Definition and Considerations," *NIST SP 800-162*, 2014.

[6] J. Vollbrecht *et al.*, "AAA Authorization Framework," RFC 2904, Tech. Rep., August 2000.

[7] E. Rissanen, "eXtensible Access Control Markup Language (XACML) version 3.0 OASIS standard," January 2013.

[8] A. Pfitzmann and M. Hansen, "A terminology for talking about privacy by data minimization: Anonymity, unlinkability, undetectability, unobservability, pseudonymity, and identity management," 2010.

[9] D. Hardt, "The OAuth 2.0 Authorization Framework (RFC 6749)," RFC 6749, Internet Engineering Task Force, Proposed Standard, October 2012, [retrieved: March, 2017]. [Online]. Available: https://tools.ietf.org/html/rfc5849

[10] N. Sakimura, J. Bradley, M. B. Jones, B. d. de Medeiros, and C. Mortimore. (2014) OpenID Connect Core 1.0. [retrieved: March, 2017]. [Online]. Available: http://openid.net/specs/openid-connect-core-1_0.html

[11] EU, *Directive 95/46/EC of the European Parliament and of the Council*, 1995.

[12] S. A. Brands, *Rethinking public key infrastructures and digital certificates: building in privacy*. MIT Press, 2000.

[13] J. Camenisch and A. Lysyanskaya, "An efficient system for non-transferable anonymous credentials with optional anonymity revocation," in *Advances in Cryptology – EUROCRYPT 2001*, ser. Lecture Notes in Computer Science. Springer, 2001, vol. 2045, pp. 93–118.

[14] (2016) PRIME. The PRIME Consortium. [retrieved: December, 2016]. [Online]. Available: https://www.prime-project.eu/

[15] (2016) PrimeLife. PrimeLife Project Consortium. [retrieved: March, 2017]. [Online]. Available: http://primelife.ercim.eu/

[16] J. Camenisch and E. Van Herreweghen, "Design and Implementation of the Idemix Anonymous Credential System," in *Proceedings of the 9th ACM Conference on Computer and Communications Security*, ser. CCS '02. New York, NY, USA: ACM, 2002, pp. 21–30. [Online]. Available: http://doi.acm.org/10.1145/586110.586114

[17] Identity Mixer. IBM Research. [retrieved: March, 2017]. [Online]. Available: http://www.research.ibm.com/labs/zurich/idemix/

[18] U-Prove. Microsoft Research. [retrieved: March, 2017]. [Online]. Available: https://www.microsoft.com/en-us/research/project/u-prove/

[19] C. A. Ardagna *et al.*, "Enabling Privacy-preserving Credential-based Access Control with XACML and SAML," in *10th IEEE International Conference on Computer and Information Technology*. IEEE, 2010, pp. 1090–1095.

[20] P. Bichsel *et al.*, "H2.2 - ABC4Trust Architecture for Developers," *ABC4Trust*, 2013.

[21] J. Camenisch *et al.*, "Concepts and languages for privacy-preserving attribute-based authentication," *Journal of Information Security and Applications*, vol. 19, no. 1, pp. 25–44, 2014.

[22] ABC4Trust - Project description. ABC4Trust EU Project. [retrieved: March, 2017]. [Online]. Available: https://abc4trust.eu/download/ABC4Trust-Project-Description.pdf

[23] T. Hardjono, E. Maler, M. Machulak, and D. Catalano, "User-Managed Access (UMA) Profile of OAuth 2.0," Internet Engineering Task Force, Internet-Draft, January 2016, work in Progress.

[24] D. W. Chadwick and K. Fatema, "A privacy preserving authorisation system for the cloud," *Journal of Computer and System Sciences*, vol. 78, no. 5, pp. 1359–1373, 2012.

[25] D. Lin, P. Rao, E. Bertino, N. Li, and J. Lobo, "Policy decomposition for collaborative access control," in *Proceedings of the 13th ACM symposium on Access control models and technologies*. ACM, 2008, pp. 103–112.

[26] M. Decat, B. Lagaisse, and W. Joosen, "Toward Efficient and Confidentiality-aware Federation of Access Control Policies," in *Proceedings of the 7th Workshop on Middleware for Next Generation Internet Computing*, ser. MW4NG '12. New York, NY, USA: ACM, 2012, pp. 4:1–4:6. [Online]. Available: {http://doi.acm.org/10.1145/2405178.2405182}

[27] ——, "Middleware for efficient and confidentiality-aware federation of access control policies," *Journal of Internet Services and Applications*, vol. 5, no. 1, pp. 1–15, 2014. [Online]. Available: http://dx.doi.org/10.1186/1869-0238-5-1

[28] M. Erdos and S. Cantor, "Shibboleth architecture draft v05," *Internet2/MACE, May*, vol. 2, 2002.

[29] N. Ragouzis *et al.*, "Security Assertion Markup Language (SAML) v2.0 Technical Overview," 2008.

[30] J. Werner and C. Westphall, "A Model for Identity Management with Privacy in the Cloud," in *2016 IEEE Symposium on Computers and Communication (ISCC)*, Messina, Italy, June 2016, pp. 463–468.

[31] W. Ma and K. Sartipi, "Cloud-based Identity and Access Control for Diagnostic Imaging Systems," in *Proceedings of the International Conference on Security and Management (SAM)*. Athens: The Steering Committee of The World Congress in Computer Science, Computer Engineering and Applied Computing (WorldComp), 2015, pp. 320–325.

[32] E. Maler, "Extending the Power of Consent with User-Managed Access," in *2015 IEEE Security and Privacy Workshops*. IEEE, May 2015, pp. 175–179.

[33] (2016) Shibboleth IdPv3 Consent Configuration. Shibboleth Consortium. [retrieved: March, 2017]. [Online]. Available: https://wiki.shibboleth.net/confluence/display/IDP30/ConsentConfiguration

[34] (2016) MITREid Connect. MIT Consortium for Kerberos and Internet Trust (MIT-KT). [retrieved: March, 2017]. [Online]. Available: http://kit.mit.edu/projects/mitreid-connect

[35] (2016) OpenAZ. [retrieved: March, 2017]. [Online]. Available: https://github.com/apache/incubator-openaz

[36] OECD, *Guidelines on the Protection of Privacy and Transborder Flows of Personal Data*. Organisation for Economic Co-operation and Development, 1981.

[37] E. McCallister, T. Grance, and K. Scarfone, "Guide to Protecting the Confidentiality of Personally Identifiable Information (PII)," *NIST SP 800-122*, 2010.

[38] T. L. (ed.), M. McGloin, and P. Hunt, "OAuth 2.0 Threat Model and Security Considerations (RFC 6819)," RFC 6819, Internet Engineering Task Force, Informational, January 2013, [retrieved: March, 2017]. [Online]. Available: {https://tools.ietf.org/html/rfc6819}

[39] S. Marouf, M. Shehab, A. Squicciarini, and S. Sundareswaran, "Adaptive reordering and clustering-based framework for efficient XACML policy evaluation," *IEEE Transactions on Services Computing*, vol. 4, no. 4, pp. 300–313, 2011.

[40] A. Mourad and H. Jebbaoui, "Towards efficient evaluation of XACML policies," in *Privacy, Security and Trust (PST), 2014 Twelfth Annual International Conference on*. IEEE, 2014, pp. 164–171.