Federated Access Control in Service Oriented Architecture

Abdramane Bah, Pascal André, Christian Attiogbé, Jacqueline Konaté

Federated Access Control in Service Oriented Architecture

Abdramane BAH
University of Nantes
LINA UMR CNRS 6241

Pascal ANDRE
University of Nantes
LS2N UMR CNRS 6241

Christian ATTIOGBE
University of Nantes
LS2N UMR CNRS 6241

Jacqueline KONATE
FST-USTTB
University of Science and Technology of Bamako

***Abstract*—Service-oriented architectures implemented by web services technologies provide standardized protocols for communicating and sharing information across organizational boundaries. The composition or federation of the services of independent organizations allows them to work together to quickly achieve their common goals. The access control of the shared services becomes an essential requirement for a secure federation of services. The identity federation provides part of the response by allowing users to authenticate once in an organization and to access the services of others with his authorization information or attributes. However, in a federation, the organizations may have different access control models that use authorization attributes with different, or even incompatible semantics. Interoperability between the access control models becomes crucial to the federation of services. Existing federated access control solutions are based on the single sign-on with common authorization attributes or the identity mapping that is not scalable in a service-oriented environment. In this paper, we propose a cross-organizational access control method for the federation of services protected by heterogeneous access control models. Our method is based on a new federation architecture that responds to the heterogeneity of authorization attributes via independent attributes introduced at the federation level.**

***Keywords*-Access control, Attribute mapping, Trust, Federated single sign-on, Web service composition**

I. Introduction

Service-oriented architecture (SOA) implemented through web services technologies provides standardized protocols for sharing information across organizational boundaries. With the globalization of business, the federation of services appears as a solution allowing independent organizations to collaborate together in order to reach quickly their common objectives. A *federation of service* is the composition of the services of different service providers across organizational boundaries [1][2].

However, to ensure the security of the services, each service provider has its own access control mechanism in place to identify service consumers and to ensure that services are accessible only to those who are authorized. The service providers and consumers can have different access control models using different or even incompatible authorization methods or attributes. We call the a*uthorization attributes*, the access control information assigned to an entity (user, service) by a trusted party. We can cite for example the user's role of *Role-Based Access Control* (RBAC) model, the user's clearance of *Mandatory Access Control* (MAC) model. The heterogeneity of the authorization attributes of access control models creates a major challenge for a secure federation of services. The service consumers must be able to compose the services of different service providers using their original authorization attributes. The access control mechanism of service providers must be able to support the authorization attributes specific to each service consumer.

The access control mechanisms do not natively support these service federation requirements. Although the identity federation provides some of the response by delegating the authentication to service consumers, the federation of services requires a common understanding of authorization attributes. In order to access the services, their consumers must either transmit their own authorization attributes or those specific to service providers. In both cases, the disclosure of the authorization attributes would be a leakage of security policies informations [2]. Current federation architectures rely on common authorization attributes and the *identity mapping* for the access control. The identity mapping requires having identities of each service provider and establishing relationships between those identities. These solutions are not scalable in a service-oriented environment where services can be dynamically discovered and composed.

In this paper, we propose a cross-organizational access control method for the federation of services protected by heterogeneous access control models. Our method is based on a new federation architecture that responds to the heterogeneity of the authorization attributes through independent attributes introduced at the federation level.

The rest of the paper is organized as follows: the Section 2 introduces the basic concepts of the federation of services, as well as the challenges and limitations of existing federated access control solutions. The Section 3 describes in detail our access control method. The implementation of the proposed method is described in the Section 4. The Section 5 presents the evaluation of our method applied to a case study. We end with related work in Section 6.

II. Federation of services

In this section we introduce the background for service-oriented architecture, federation of services, the challenges of access control in the federations, the existing solutions and their limitations.

A. Background

1) Service-Oriented Architecture: SOA is an approach to organize distributed resources as autonomous and remotely accessible units of fonctionalities called services [3]. Services are discoverable and accessible to end-user applications or other distributed services via standard message interfaces and protocols. SOA has three constituent components that are determined according to three roles assumed by each component: the *service provider*, the *service registry* and the *service consumer* or client. The service provider is a software component that hosts and executes the service on the behalf of the service consumer who discovered the service description in the service registry.
Web services provide an implementation of SOA accessible via standard Internet protocols such as HTTP. A *web service* is a self-describing, self-contained software module that can perform actions on behalf of a user or application [4]. Web services rely on standard protocols such as SOAP and WSDL for the description of the service interface and communication messages. The goal of Web Services is to enable the creation of distributed applications that can be dynamically assembled by composing existing services as needed.
2) Interoperability and Federation: Interoperability is the ability of two or more systems to exchange information and access third-party functionality [5]. To ensure interoperability, SOA services can be located in independent trust domains called security domains. A *domain* is a security administration unit consisting of a set of elements, security authorities, and a security policy in which the elements are managed in accordance with the security policy [6][7]. Interoperability between domains requires common collaboration agreements and a secure and trusted environment. To achieve full interoperability between independent and autonomous domains, federation is one recommended solution. A *federation* is a set of autonomous domains that adhere to common rules and governance policies to control interactions between them [8]. The federation creates a trusted environment for the secure sharing of services between domains.

The security of the services is accomplished through the access control which ensures that only authorized users have access to the services. The federation allows the domains to have control over the security of their services. Each domain can use its own access control model. Access control begins with user authentication. In a federation, the authentication is provided by the identity federation, which allows the users to authenticate only once in one domain and access the services of others by using a unique identity. The domain that provides the identity is called the *identity provider* (IdP) and the domain that use this identity to provide the services is called service provider or *relaying party* (RP). The users authenticate with IdPs who create and transmit proof of authentication to the RPs as security tokens. A *security token* represents a set of *claims* that are declarations made by a third party about the user's identity attributes, such as his name, and his authorization attributes. Access control in domains is based on these authorization attributes.

The exchange of security tokens between IdP and RP allows the *federated single sign-on* (FSSO) between the domains. The security tokens are described using the *Security Assertion Markup Language* (SAML) standard to ensure interoperability between domains. A domain can ensure both the role of the service provider and the service consumer. The service federation allows to create distributed applications using the services provided by the domains of a federation. A *federated service* is a service accessible only to authorized users of a federation [1]. Given the federation's autonomy and flexibility requirements, the access control for federated services remains a major challenge.

B. Challenges

The access control of federated services faces two major challenges:

1) Heterogeneity of domain authorization attributes
   Each domain specifies its access control policies on its own authorization attributes. When domains use different or incompatible authorization attributes, access to services is either hindered or granted to unauthorized users.
2) Composition of federated services
   The composition of federated services must take into account the access control of each service and therefore the heterogeneity of their authorization attributes.

Access control solutions for federated services must meet the following requirements:

- *Federated single sign-on*. A user must be able to authenticate once to the federation and then use the services for which he has a valid authorization.
- *Federated single authorization*. A user must be able to acquire authorizations once from the federation and access the services on the basis of these authorizations.
- *Domain autonomy*. Each domain must be autonomous to control the access to its services.
- *Dynamic adaptation to the federation growth*. The domain's access control mechanisms should not require significant maintenance efforts during authorization changes in the federation.

- *Confidentiality of internal security information*. The authorization attributes are sensitive information and should not be disclosed beyond the domain boundaries.

C. Existing federated access control solutions

The access control consists of three essential steps: (i) identification and authorization of users; (ii) authentication of users; (iii) control of access authorizations to the services.
The federated access control authentication of users is delegated to IdPs through identity federation. The authorization of users and the control of access authorizations remain under the control of RPs. However, these latters depend on federation architectures, the main ones being Shibboleth, Liberty Alliance, and WS-Federation [9][10].
With Shibboleth, IdP and RP agree to use common authorization attributes whose semantics are defined through an LDAP schema such as the eduPerson schema. RPs specify their access control policies on the authorization attributes defined in this schema. Shibboleth also offers IdP the ability to use its own authorization attributes. In this case, the IdP must map its authorization attributes with those of the federation to access RP's services. These mappings are managed by each IdP and are therefore unreliable. This may result in unauthorized access to the services. With Liberty Alliance, the user has distinct identities with different IdPs and RPs that are connected for authentication and authorization. In this case, the access to the services does not rely on authorization attributes, but on the identity attributes such as name, email as an example.
WS-Federation supports Shibboleth and Liberty Alliance access control techniques through specialized services such as *authorization service*, *attribute service* and *pseudonym service*. WS-Federation also provides identity mapping solutions that consists of converting an identity of one domain into an identity in another domain by a third party approved by the originating domain and the end domain. However, these identity mappings solutions are not flexible because they require point-to-point negotiations between each pair of domain. Identity mapping is not applicable to service composition because services can be discovered and dynamically composed. The access control of service composition requires authorization negotiations going beyond two domains. The federated services access control requires a federation architecture that supports authorization negotiations for service composition. In the next section, we propose an access control method that addresses these needs.

III. Federated access control : our method

Our method is based on a specific federation architecture.

A. Federation architecture

The domains are federated by considering that the services in one domain are accessible to other domains based on the trust relationships and access control policies. Cross-domain access control requires that the authorization attributes of a domain are interpretable in the other domains. In order to interoperate domain access control models, we introduce the *global access control mechanism* (GACM) at the federation level. The domains no longer need to negotiate access authorization with each other (plain arrows in Fig. 1), they only need to negotiate access authorization once with GACM (dashed arrow inf Fig. 1) and then access domain services directly with these authorization as shown in the Figure 1. The GACM serves as an interface between the domain access control models.

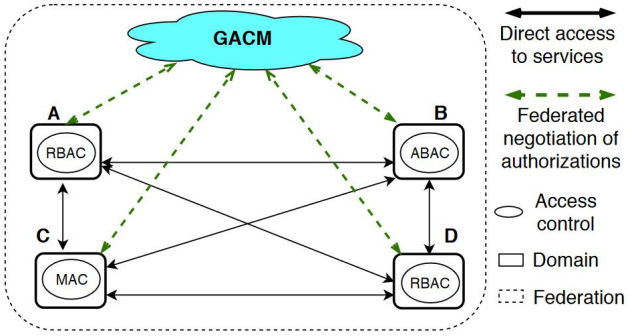

Figure 1: The proposed federation architecture

B. Interoperability between domain access control models

The purpose is to allow domains to understand the authorization attributes of others in order to determine the local access authorization for those attributes. To achieve this goal and to avoid leakage of security information, we define the GACM authorization attributes called *federated attributes* independently of the local authorization attributes of the domains. The federated attribute are public and available to all domains. Each domain negotiates with the GACM to map its local authorization attributes to the federated attributes as illustrated in the Figure 2. This first mapping, called *federated mapping*, is registered at the federation level. Locally, domains also map federated attributes to their local authorization attributes. We call this mapping, the *domain mapping*. The interactions between the domains are then performed using the federated attributes.
Using federated attributes, domains can grant access authorizations to all other domains in the federation without knowing their local authorization attributes. As a result, federated attributes allow domains to collaborate together regardless of the heterogeneity of their access control models. The advantage for the users is to obtain the access authorizations in other domains based on their original authorization attributes.

C. Access to federated services

We now present how to access the federated services.

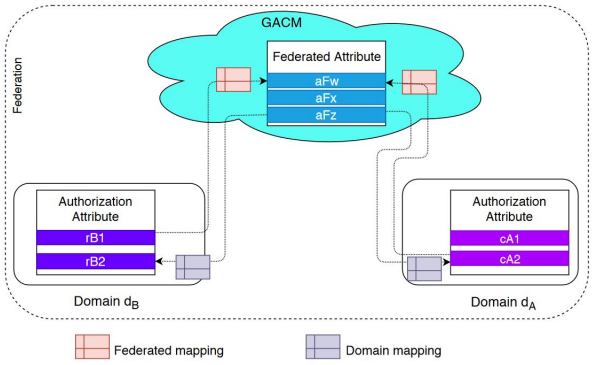

Figure 2: Federated attribute for access control models interoperability

1) Authentication and trust brokering: The access control relies on the authorization attributes asserted by a trusted third party. Each domain has its own authentication mechanism called *local token service* (LTS). The LTS authenticates users and issues a security token signed by the domain security certificate. The services of a domain are accessible only with a security token issued by the domain's LTS.
In order to establish trust between domains, we introduce in the GACM a specialized authentication mechanism called *federated token service* (FTS) for domain authentication. We identify the domains and the GACM with the public-key certificates. The domain security certificates are forwarded to the GACM which in turn transmits its certificate to the domains. The domains authenticate to the GACM with the security tokens signed with their security certificates. In response, the FTS delivers the security tokens signed by the GACM's security certificate. Consequently, the domains of the federation trust each other through the *federated security tokens*.
As shown in Figure 3, to access to a service (S_B) of domain B (d_B) from a domain A (d_A), the authentication of the user (U_A) is performed with the following steps :

1) the LTS of d_A authenticates U_A and delivers an security token (ST_A) signed with the d_A security certificate;
2) d_A authenticates to the FTS using ST_A and obtains on behalf of U_A, a federated security token (ST_F) signed with the GACM certificate;
3) the service consumer use ST_F to obtain a security token (ST_B) from d_B signed with d_B security certificate. The ST_F signature proves that d_A and U_A belong to the federation and are trustworthy.
4) finally, S_B is called on behalf of U_A with ST_B.

The authorization attributes contained in the ST_B being specific to d_B, are used for S_B access control.
2) Authorization: The security token used to invoke a service must contain the authorization attributes of the domain providing this service. The user initially has the authorization attributes of his domain. These initial authorization

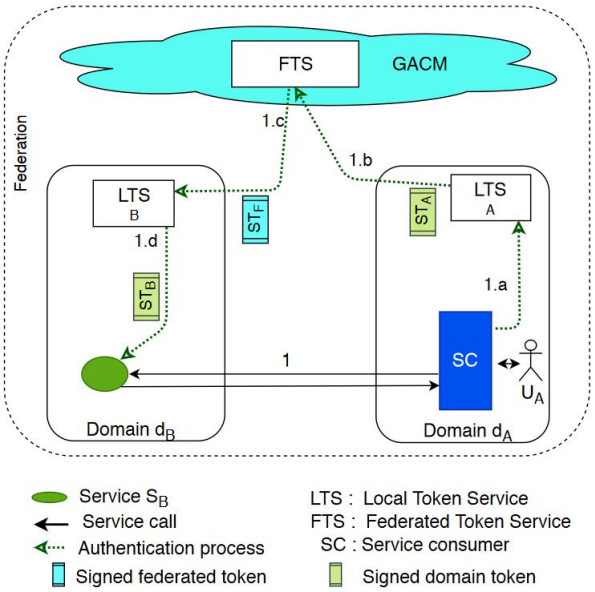

Figure 3: Principle of federated Single Sign-On

attributes must be successively mapped to the federated attributes and the target domain's authorization attributes during the authentication process. We assume that the federated mappings and domain mapping discussed in Section III-B are already established. To achieve these mappings, we define a two-level attribute mapping:

- at the GACM level, an attribute mapping provides the federated attributes corresponding to the domain authorization attributes;
- at the level of domains (providing services), an attribute mapping provides domain authorization attributes corresponding to the federated attributes.

The attribute mapping is performed during the authentication process. In the Figure 4, we illustrate the attribute mapping by considering the steps presented in III-C1. To achieve the authorization of U_A, the authorization attribute of U_A (*cA1*) is used by the FTS to compute the federated attribute *aFx* corresponding to cA1. The aFx sent to d_B, is then used by the d_B's LTS to compute the authorization attribute *rB2* corresponding to aFx. This latter is finally used to access the service targetted by U_A.

D. Composition of federated services

Each composed service has its own authorization attribute requirements. The access control of the service composition is done at two levels [11]: the composite service's access control and the composed services's access control. This creates two additional issues: (1) the specification of the composite service's access control requirements; (2) and the federated single sign-on between the composite service consumer (intial requester), the composite service and the composed services. We solve these issues by considering two

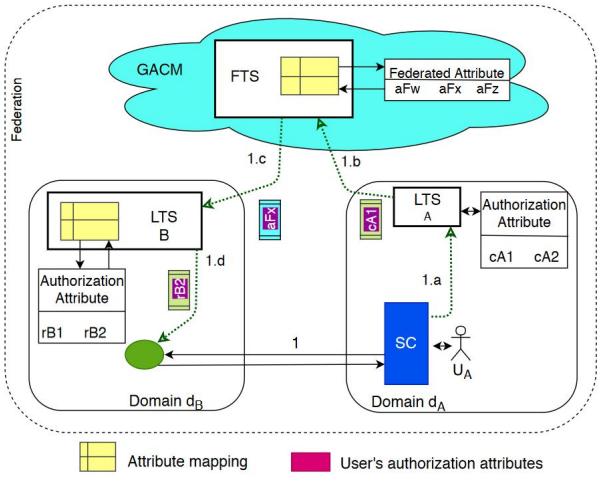

Figure 4: Sequence of cross-domain authorization

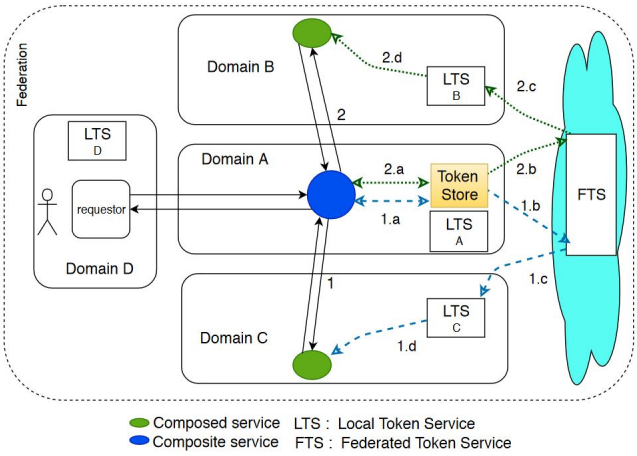

Figure 5: Invocation of composed services on the behalf of the initial requester

scenarios: (i) we invoke the composed services on the behalf of composite service; (ii) we invoke composed services on the behalf of the intial requester.

In the first scenario, the access control of the composite service is performed like in any federated service. The access control requirements of the composite service are independent of those of the composed services. To invoke composed service, the composite service follows the authentication steps described in the Section III-C1.

In the second scenario, the composite service's access control requirements depends on those of the composed services that may be different from one service to another. The composed services require a security token containing the authorization attributes of their domains. The composite service consumer must provide a security token that satisfies these requirements.

For this purpose,

1) we introduce the *token store* at the composite service level to store the security token of the initial requester (ST_{init}). The composite service must convey the ST_{init} to invoke the composed services. But, the ST_{init} contains the authorization attributes of the domain that provides the composite service.
2) we perform a new authentication process using the ST_{init} in order to have the authorization attributes corresponding to those contained in ST_{init}.

The Figure 5 illustrates the service composition with this scenario.

IV. Implementation of our method

This section presents how to implement our method to perform web services access control. The goals is to develop the required software modules; to select and customize existing security services to support our access control method.

WS-Trust, WS-SecurityPolicy and WS-Security provide the basic model of the federation of web services. WS-Trust is implemented with *Security Token Service* (STS) that provides methods for issuing, validating, transforming, and renewing security tokens. The LTS of domains and the FTS of GACM are implemented with WS-Trust STS. We have three types of STS in our architecture: the STS in the services providers domains (named STS_{SP}), the STS in the services consumers domains (named STS_{SC}) and the STS of the GACM named STS_{GACM}.

The implementation of our method involves four steps.

Step 1: *Definition of claim dialect of federated attributes*. The security requirements of federated web services must be specified using the federated attributes defined by the GACM. However, WS-SecurityPolicy does not define a claim dialect for the expression of claim requirements. We define a claim dialect (XML schema) to describe the federated attributes. Each web service specifies its claim requirements using this claim dialect.

Step 2: *Definition of federation-specific security requirements of STS_{SP} and STS_{GACM}*. First, the target web service requires a SAML token issued by the STS_{SP} of its domain with specific claims. The STS_{SP} requires a SAML token issued by the STS_{GACM} which also requires a SAML token. The STS_{GACM} does not specify the token issuer because it trusts all STS_{SC} in the federation.

Step 3: *Implementation of attribute mapping of STS_{SP} and STS_{GACM}*. The STS_{SP} and the STS_{GACM} are customized in order to implement the attribute mapping as illustrated in the Figure 6. These STS must be able to retrieve authorization attribute (claims) contained in the SAML tokens and exchange them with the corresponding attributes stored in the *mapping module* that contains the pre-established attribute mapping. We implement the mapping module with

a relational database so that it can be queried in order to easily find the desired attributes.

Step 4: *Implementation of service's access control enforcement*. Web services access control is based on XACML which has several logically distinct components, including the *Policy Enformcement Point* (PEP) and the *Policy Decision Point* (PDP). The PEP intercepts the SOAP request, extracts the authorization attributes contained in the SOAP message header and enforces acces decision made by the PDP. We assume that domains already have access control policies defined on their local authorization attributes. As a result, the existing PDP are maintained. But, we implement the PEP with *Apache CXF [12] interceptors*.

Web services access control requires the composition of several security standards, namely WS-Security, WS-Trust, SAML, XACML and possibly WS-Federation. SAML and XACML are implemented independently. But WS-Federation, WS-Trust, WS-Security are dependent. The implementation of WS-Federation depends on the implementation of WS-Trust which depends on that of WS-Security thus forming the layers of security protocols. This dependency is difficult to deal with because there are no solutions that deploy these layers together. The web service developer is constrained to deploy each of its layers separately. What is likely to generate configuration errors or impossible to make them work together. The main web services development solutions providing the security layers are *Apache CXF*, *Axis2* [13], *Glassfish Metro* [14] for those that are compatible with *JAX-WS* specification [15] and *Microsoft's WCF* [16]. These solutions do not directly integrate the access control (XACML and SAML). Solutions that integrate access control such as *WSO2 Application Server* [17], do not support WS-Federation or WS-Trust inter-domain. Finally, the implementation of web services access control becomes quickly a real challenge.

V. Application and Evaluation

We present a case study on which we experiment the proposed method.

A. Case study: Federation of Scholarship Services

The case is a federation of three institution systems involved in the payment of students scholarship. Initially, the scholarships were paid by a national treasury but three independant higher education institutions are responsible to grant the scholarship to students: the Center of University Studies (*CUS*), the Directorate of Higher Education (*DHE*) and Universities. The usual scholarship is allocated by CUS. An additional aid is allocated to disabled students by the DHE. Some universities grant on their budjets an aid to the non-scholars students. The Treasury pays the scholarships and the various aids for the account of each

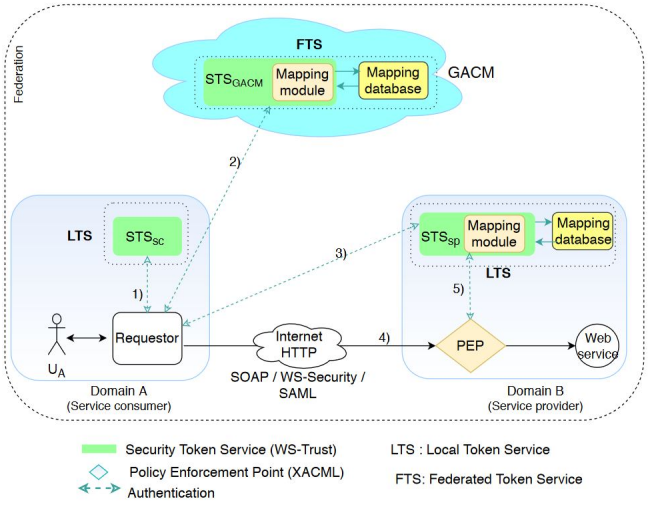

Figure 6: The required software modules and their customizations for the implementation

institution that dispatches them to students. To this end, each institution establishes and submits to the treasury a scholarship payment card consisting of a set of attribution codes (*sc-code*) and their amount. Each sc-code represents a student's scholarship. In order to facilitate the payment of scholarships, the decision is taken that all payments should be now made by the accounting departments of universities. To put into practice, the CUS, DHE and universities decided to federate their systems to share securely the scholarships attribution codes. The CUS provides the sc-codes of the usual scholarship. DHE provides the sc-codes of disabled students aids. The Universities collect the sc-codes of their students at CUS and DHE to establish their payment cards. The scholarship of a disabled student is the sum of his sc-codes. The table I describes the different domains that will participate in the federation.

Table I: Description of the domains to federate

Domain	A.C. model	Authorization attribute (role)	Web services
CUS	RBAC	financial-officer, accounting-officer, chief-accountant	scholarshipService
DHE	ABAC	cashier, accountant	disabled-grantService
UTS	RBAC	administrator, financial, accounting-secretary	

As described in Table I, the CUS and DHE are the service providers and the universities such as Univerversity of Technical Sciences (UTS) are the consumers of these services. Each domain has its access control model (A.C. model) with its authorization attributes that are the roles of the RBAC

model. The role is considered in the ABAC model as an attribute. To create his scholarship payment card, the UTS accounting department must access the web services of the CUS and DHE. This requires establishing trust between their systems and the interoperability between their access control models.

B. Federation of domains

To federate the CUS, the DHE and the universities, we follow the steps described in the Section IV.

Step 1 - Definition of federated attributes and the claims dialect. An autonomous department of DHE, the Department of Administrative Affairs (DAA) is designated to host the GACM. A security certificate is created for the DAA, CUS, DHE and all universities belonging to the federation. DAA registers the security certificates of the domains. The CUS, DHE and universities also register the DAA security certificate. The DAA and the domains of the federation can now trust each other. The DAA defines the federated attributes as shown in the Table II and creates the claim dialect to describe them. The DAA negotiates with the domains to establish the federated mapping. The CUS and the DHE establish their domain mapping. The DAA sets up its security token service, the STSDAA. It is assumed that the domains already have their STS. Otherwise, the domains install their STS. The STS of the UTS is configured to support the claims dialect.

Table II: The DAA federated attributes

Finance	finance-director finance-assistant finance-secretary ...
Administration	adminitration-director administration-adjt ...
Information technology	it-administrator ...

Step 2 - Definition of STS security requirements. The CUS and DHE specify the access control requirements of their web services and their STS using the DAA claims dialect.

Step 3 - Implementation of attribute mapping. The DAA, CUS, and DHE create a database to store federated mapping and domain mappings respectively. Then, they install in their STS, the mapping implementation which is a generic software component to query the mapping databases.

Step 4 - Web services access control. The CUS and the DHE deploy SOAP message interceptors to extract the authorization attributes from the SAML assertions and enforce the services access decision.

After these steps, the UTS and other universities can then access the web services of the CUS and the DHE to collect the students sc-codes and create their scholarship payment card.

C. Evaluation

We evaluate our web services federation architecture based on the following criteria:

- *Applicability*: the ease of implementation and integration into an existing security environment;
- *Scalability*: the adaptation to the evolution of the federation like the increase of the number of domains or users;
- *Reliability and security*: the trust in the access control;
- *Extensibility*: the support of others access control models different from RBAC and ABAC

Applicability. For example, in Section V-B, when an university —using an LDAP registry with OpenAM as the authentication mechanism and a RBAC as authorization model— participates in a federation built according to our method, the existing security mechanisms (LDAP, OpenAM and RBAC) are maintained. OpenAM is configured to support the dialect of the federation. Internal roles used for the authorizations are never disclosed to CUS and DHE. This reduces the dependencies between domains for the access control. The only change in CUS and DHE is the implementation of an STS in order to support the attribute mapping. Our architecture fits well with existing access control mechanisms of domains and its adoption requires minimal configuration efforts.

Scalability. The evolution of the federation has no effect on the access control of the CUS and DHE because of the stability induced by the domain mapping. Service providers adapt themsselves only to the evolution of the federated attributes and not directly to that of the involved domains.

Reliability and security. Federated mapping is only done at the GACM level. This ensures the reliability of the authorizations granted by the service providers through their domain mapping.

Extensibility. Our approach assumes different access control models in each domain of the federation. But only the models that group authorization such as Group-based, Role-based, and Attribute-based are supported. Other access control models such as History-based are not taken into account. The domain attribute mapping can be extended with individual authorization to allow fine-grained access control. However, our approach focuses more on the access control of external users to the domains. In the case where the federated service is used within the domain, the access control of the service will always use the GACM. Internal use of the federated service requires a new WSDL interface that does not employ the federation.

VI. Related Work

In this section, we discuss related works on inter-domain access control. Many efforts have already been made for

inter-domain access control, especially with the advent of e-commerce. In the following, the terms domain and organization are used interchangeably. The role is the authorization attribute in the RBAC model.

In [18] the authors proposed for the access control of inter-organizational workflow, an architecture that decoupled the security infrastructures of member organizations by introducing a shared role domain containing the roles each organization (RBAC) and the relationships between them. Each organization associates its local role structure with the role domain. The role domain is similar to the GACM of our architecture. However, the workflow is considered as an internal application. Unlike our method, role mapping is point-to-point and does not require trust management between organizations.

A similar approach is proposed in [19] for managing federated access to collaborative network environments. This approach introduces the notion of federated attribute as a solution to the heterogeneity between the local attributes of domains. Each domain must map local attributes to a set of federated attributes that are used in the context of the collaboration. Each domain defines its federated attributes from which permissions are assigned. Conversely, federated attributes in our approach serve as the common authorization language between domains to determine local permissions for external users.

The concepts of Private Virtual Organization (VPO) and Single Sign-On (RSSO) are introduced in [20] to help organizations maintain the control over their resources and users to have permissions in other organizations according to their role. The organizations must create as many VPOs as partners. Each VPO has different authorization policies defined on the roles of the concerned partner. The GACM in our architecture overcomes this multiplication of authorization policies. The GACM allows organizations to share the same resources with multiple partners unlike VPO. The concept of a single role is very interesting for the single federated authorization, but the disclosure of the role to the partners constitutes a leak of security information according to [2].

The federation of web services [1] [21] further complicates inter-domain access control because of trust management and federated single sign-on. In [22], the authors propose to convey the authorization attributes of the user instead of or in addition to its identity attributes (e.g. his name) in the security token. We employ the same approach, but we convey the federated attributes of the user instead of his local authorization attributes.

A two-level access control architecture is proposed in [2] for the access control of web service in which the authorization decision is made in the requester's domain and attached to the request invoking the web service. Although this approach preserves domains autonomy and avoids the leakage of security information, it is hardly applicable to a federation of multiple domains that need the information about the requester.

The WS-Federation standard provides a flexible and extensible architecture for access control of federated web services. Our approach extends the WS-Federation architecture by the GACM for the trust management and the federated single sign-on. Unlike our approach, the access control with WS-Federation is based on the user's identity attributes. Identity maping approaches proposed by WS-Federation are point-point and is very complex to apply to the service composition. The federated attributes used in our approach allow to perform the access control to the service composition without requiring complex attribute mapping.

VII. Conclusion

We proposed a cross-domain access control method for service-oriented environments. Our approach is based on a new federation architecture that allows domains to be independent and autonomous for the access control of their services. In this architecture, a domain can be both a service consumer and a service provider. We introduced a third-party entity at the federation level, the global access control mechanism (GACM) in order to establish trust between domains and interoperability between their access control models. The access control models of domains interoperate using the federated attributes.

The federated attributes are used to establish mapping between the authorization attributes of the domains. We defined a federation-level attribute mapping between domain attributes and federated attributes and a domain-level attribute mapping between federated attributes and domain attributes. These attribute mappings support federated single sign-on between domains. They also make it possible to compose the services of different domains using heterogeneous models of access control.

We proposed an implementation of our method for the access control of web services in which we detail the different steps of implementation, the necessary components and the difficulties encountered. Our method was applied to a case study in order to evaluate it according to feasibility, reliability, scalability and security criteria. However, the current implementation of our architecture is about functional testing. We have not yet deal with performance testing. We plan to experiment the access control of service composition where composed services are invoked on the behalf on initial requester. Token store integration in the orchestration engines must be also implemented. We plan to extend attribute mapping with individual access permissions to allow fine-grained cross-domain access control.

References

[1] E. Dalsgaard, K. Kjelstrøm, and J. Riis, "A federation of web services for danish health care," in *Proceedings of*

the 7th Symposium on Identity and Trust on the Internet, ser. IDtrust '08. New York, NY, USA: ACM, 2008, pp. 112–121. [Online]. Available: http://doi.acm.org/10.1145/1373290.1373305

[2] M. Menzel, C. Wolter, and C. Meinel, "Access control for cross-organisational web service composition," *Journal of Information Assurance and Security*, vol. 2, no. 3, pp. 155–160, 2007.

[3] L. Dikmans and R. Van Luttikhuizen, *SOA made simple discover the true meaning behind the buzzword that is "service oriented architecture"*. Birmingham, UK: Packt Pub, 2013, oCLC: 847034163.

[4] M. P. Papazoglou, *Web services: principles and technology*. Harlow: Pearson/Prentice Hall, 2008, oCLC: 255863191.

[5] D. Chen and N. Daclin, "Framework for enterprise interoperability," in *Proc. of IFAC Workshop EI2N*, 2006, pp. 77–88.

[6] *Web Services Federation Language (WS-Federation) Version 1.2. OASIS Standard*, 05 22 May 2009. [Online]. Available: http://docs.oasis-open.org/wsfed/federation/v1.2/os/ws-federation-1.2-spec-os.pdf

[7] *Baseline identity management terms and definitions*, International Telecommunication Union, 04 Avril 2010. [Online]. Available: https://www.itu.int/rec/T-REC-X.1252-201004-I/en

[8] N. Duan, "Design Principles of a Federated Service-oriented Architecture Model for Net-centric Data Sharing," *The Journal of Defense Modeling and Simulation: Applications, Methodology, Technology*, vol. 6, no. 4, pp. 165–176, Oct. 2009. [Online]. Available: http://journals.sagepub.com/doi/10.1177/1548512909352790

[9] U. Fragoso-Rodriguez, M. Laurent-Maknavicius, and J. Incera-Dieguez, "Federated Identity Architectures," p. 8, 01 2006.

[10] J. Kallela, "Federated identity management solutions," 2008.

[11] M. Coetzee and J. H. P. Eloff, "Towards web service access control," *Comput. Secur.*, vol. 23, no. 7, pp. 559–570, Oct. 2004. [Online]. Available: http://dx.doi.org/10.1016/j.cose.2004.05.006

[12] "Apache cxf," http://cxf.apache.org/, accessed: 2019-01-24.

[13] "Apache axis2," http://axis.apache.org/axis2/java/core/, accessed: 2019-01-24.

[14] "Metro web service stack," https://javaee.github.io/metro/, accessed: 2019-01-24.

[15] *The Java API for XML-Based Web Services (JAX-WS) 2.2*, Sun Microsystems, Inc., 12 2009. [Online]. Available: https://download.oracle.com/otn-pub/jcp/jaxws-2.2-mrel3-evalu-oth-JSpec/jaxws-2_2-mrel3-spec.pdf?AuthParam=1548322813_c2b0d630ba0646f70f84dde067145187

[16] "Windows communication foundation," https://dotnet.microsoft.com/, accessed: 2019-01-24.

[17] "Wso2 application server," https://wso2.com/products/application-server/, accessed: 2019-01-24.

[18] M. H. Kang, J. S. Park, and J. N. Froscher, "Access control mechanisms for inter-organizational workflow," in *Proceedings of the sixth ACM symposium on Access control models and technologies*. ACM, 2001, pp. 66–74. [Online]. Available: http://dl.acm.org/citation.cfm?id=373266

[19] C. E. Rubio-Medrano, Z. Zhao, A. Doupe, and G.-J. Ahn, "Federated Access Management for Collaborative Network Environments: Framework and Case Study," in *Proceedings of the 20th ACM Symposium on Access Control Models and Technologies - SACMAT '15*. Vienna, Austria: ACM Press, 2015, pp. 125–134. [Online]. Available: http://dl.acm.org/citation.cfm?doid=2752952.2752977

[20] F. Cuppens, N. Cuppens-Boulahia, and C. Coma, "O2o: Virtual Private Organizations to Manage Security Policy Interoperability," in *Information Systems Security*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, vol. 4332, pp. 101–115. [Online]. Available: http://link.springer.com/10.1007/11961635_7

[21] B. Fabian, S. Kunz, S. MüLler, and O. GüNther, "Secure federation of semantic information services," *Decis. Support Syst.*, vol. 55, no. 1, pp. 385–398, Apr. 2013. [Online]. Available: http://dx.doi.org/10.1016/j.dss.2012.05.049

[22] J. Li and A. H. Karp, "Access control for the services oriented architecture," in *Proceedings of the 2007 ACM workshop on Secure web services - SWS '07*. Fairfax, Virginia, USA: ACM Press, 2007, p. 9. [Online]. Available: http://portal.acm.org/citation.cfm?doid=1314418.1314421