Identity federation scenarios for the Cloud

N. Ben Bouazza[#1], M. Lemoudden[#2], B. El Ouahidi[#3], D. Bourget[*4]

[#] Mohammed-V University - Agdal, Faculty of Sciences, L.R.I,
B.O. 1014 Rabat, Morocco
[1] ben.bouazza.naoufal@gmail.com
[2] mouad.lemoudden@gmail.com
[3] bouabid.ouahidi@gmail.com

[*] Institut Mines Telecoms, Telecom Bretagne, Technopôle de Brest IROISE,
CS 83818, 29238 Brest Cedex, France
[4] daniel.bourget@telecom-bretagne.eu

Abstract— Traditional identity approaches in a cloud environment that demands scale and openness, suffer from a number of limits, especially when the enterprise uses multiple cloud service providers (CSPs) and user credentials are not shared with other providers. Multiple attempts to solve this problem have been proposed like federated Identity that has a number of advantages, even though it suffers from many challenges that are common to new technologies. Keeping business systems data safe and protecting the identity has never been more difficult to achieve; therefore, in this paper we tackle federated identity, its components, advantages, disadvantages, then we propose a number of useful scenarios to manage identity in the cloud.

Keyword- federated identity, cloud, security, claim, token, identity provider, SaaS, federation provider, access control

I. INTRODUCTION

Mobile apps we have today would not exist without the cloud, which explains the synergistic proliferation of those two technologies in recent years. Mobile apps rely on the cloud model that offers a number of services, from computing, networking to storage, enabling organizations to new opportunities. However, when it comes to security, and especially managing the identity, organizations are facing numerous issues: the perception of perimeter security becomes fuzzy, compared to the luxury of traditional static conceptual boundary, where all parties involved are living inside a shared domain using a directory service to manage their identities by protocols such as Kerberos and LDAP, that are provisioned by the enterprise. Thus, eliminating trust issues.

Today, most cloud providers store user credentials in different places, which are not shared with other providers. This results in users having multiple sets of credentials, which causes issues, since cloud provider policies may differ from the internal corporate policies. adding to this is the Bring Your Own Device (BYOD) to work trend (smart phones and tablets) that has increased the level of concern for many IT managers and executives, essentially because mobile devices weren't designed for enterprises (No SOAP, No SAML, No WS-*,...), since they have limited resource and no XML stack.

This paper will first talk about traditional identity approaches in section 2. Section 3 identifies federated identity and its components: authentication, authorization, access control and federated service model. Section 4 gives an overview of federated identity, examining its advantages and disadvantages. In section 5, we provide a couple of proposed scenarios to manage identity in the cloud, taking into account the increased usage of mobile devices. In section 6, we make a case for claims and token transformation capabilities. The last section contains conclusion and perspectives.

II. TRADITIONAL IDENTITY APPROACHES

In this section we will review succinctly how we do things today, the simplest case is accessing an application over the internet, the user sends a username and password over https, and the application checks those in a database that maintains them, if it finds a match, the user is logged in, which does not allow to verify the identity of the user; but inside an organization, when a user wants to access an application, he has to log into a domain using a service domain like Active Directory among others that checks the login and the password against the database it maintains, if it finds a match, the user is logged into the enterprise domain, not to any single application; to access an application in that domain if we are using Kerberos to handle identity, we need a Kerberos ticket from Active Directory, which contains relatively simple information about who the user is, what groups the user belong to, and it's digitally signed by Active Directory, which means the user can provide the Kerberos ticket to the application; then it checks the signature and make sure that it was issued by Active Directory, since the application trusts Active Directory to only issue a Kerberos ticket to people it has authenticated. It will then use the information in the ticket to know various things such what data the user is authorized to access. In this manner, we authenticate the users of the application, and provide a common way to represent the information about the identity; that is the ticket. But Kerberos, as we mentioned before, itself isn't

sufficient for broad-based applications since it's designed to function in shared domain, shared realm and shared enterprises.

To solve broad-based application problem like business to business federation there are a number of protocols that are used to connect outside resources such as SOAP, WS-* (Trust, Security, Federation,...) and SAML/XML, allowing either internal employers to connect to external resources, and external employers to connect to internal resources using SSO [15], and it works because both ends of the environment are typically tightly controlled.

One problem with using traditional identity approaches in a cloud environment arises when the enterprise uses multiple cloud service providers (CSPs) [7]. In such use case, synchronizing identity information with the enterprise is not scalable. Another set of problems arises with traditional identity approaches when migrating infrastructure toward a cloud-based solution.

Modern Information Systems are becoming user-centric, which make identity theft and forgery a common threat to perform malicious actions. For these reasons, most cloud service provider (CSPs) demand a proof of identity, by using a trusted third party model. In the next section we tackle federated identity, which is an effective foundation for identity in cloud computing.

III. FEDERATED IDENTITY COMPONENTS

Federated identity is a combination of multiple components and concepts aligning together to provide one solution; these components are the building blocks and are as important as the solution [2]. In this section, we review the components and the difference between them.

A. Authentication

Authentication is the process whereby, a computer system can verify the identity of a person or a computer to allow access to the entity of resources (systems, networks, applications). Authentication allows validating the authenticity of the entity in question and in consequence verifies its identity.

Authentication is composed of two components: first, identification and second, verification, occurring in that order. Identification is the means of claiming an identity; which can take the form of a username or an email address. Verification is the system processing the identity claim to check whether it's truthful, which can take the form of a password or a biometric identifier.

There is a wide variety of authentication techniques [9] that are supported by federated identity. Of these, we cite the combination of username and password as the most common method, biometric authentication that consists of comparing the presented physical attribute to a stored copy, digital user certificates that contain a unique associated key pair, ticketing-based authentication involving the use of symmetrical keys such as Kerberos, etc.

B. Authorization

Authorization is the process whereby, a computer system grants someone to be where they want to go, or to have information that they want to have [16]. Being the second step after authentication, it allows the user access to various resources based on the user's identity. Since it has an effect spread over the whole of an organization, the latter must possess a secure policy that dictates authorization within.

C. Access control

A security policy of information systems is an action plan that reflects the strategic vision of the direction of an organization to ensure security objectives such as confidentiality and integrity. These can be represented clearly and unambiguously by models of access control [3].

A model of access control can be defined as a formalism that allows developing and specifying the system behaviour accurately [10]. It also allows abstracting and facilitating the understanding of a security policy, and implementing mechanisms to ensure security objectives. Examples of these objectives are confidentiality in the case security can be connected to the disclosure of confidential information, or integrity in cases where security can be related to alteration of the information [8]. Below are a number of access control methods.

Discretionary access control (DAC) allows a subject to assign permissions to other subjects. This access control is flexible but it can pose some problems such as vulnerability to Trojan horses [4]. These models do not ensure security properties such as confidentiality and integrity because the permissions are "at the discretion" of the owner of an object. Indeed, the owner of an object can attribute to a malicious subject access to important information because he does not have a global vision of the system.

Mandatory access control (MAC) imposes essential rules guaranteeing the achievement of safety objectives. In this access control model, subjects can not intervene in the allocation of access rights. This model is more rigid, but safer than discretionary access control. Models of multi-level security [5] will prevent unauthorized disclosure of certain information to "untrusted" users. The mechanism of implementation of this policy is to assign security levels to the objects and subjects.

Role Based Access Control (RBAC) [6] can be considered as an alternative approach to mandatory access control (MAC) and discretionary access control (DAC). In this model, permissions assigned to specific roles instead of being directly affected subjects as is the case with previous models. Then, the subjects can be assigned to roles that generally result from the structure of an organization. This model simplifies operations such as adding or removing a subject. Indeed, the permissions are not allocated separately to subjects, and subjects can acquire these permissions only from their roles, which makes RBAC to be considered as an "ideal" system for companies having a frequent change of staff.

In Attribute-based access control (ABAC), each user and object has attributes. The user must prove his claims about his attributes to the access control engine. The access control decisions are made based on these attributes. It is very flexible, but difficult to determine in principle that is entitled to what. It's also difficult to understand the implications arising from changes in attributes. XACML (extensible access control markup language) is a standard for attribute-based access control.

D. Federated service model

The federated service model encompasses all of the components we have discussed so far. It configures the manner by which users authenticate and services are delivered. The model requires the existence of an IdP (identity provider) and a service provider (relying party); and both parties are supposed to have a trust relationship between them. Figure 1 below describes the interactions between the parties involved in the federated service model.

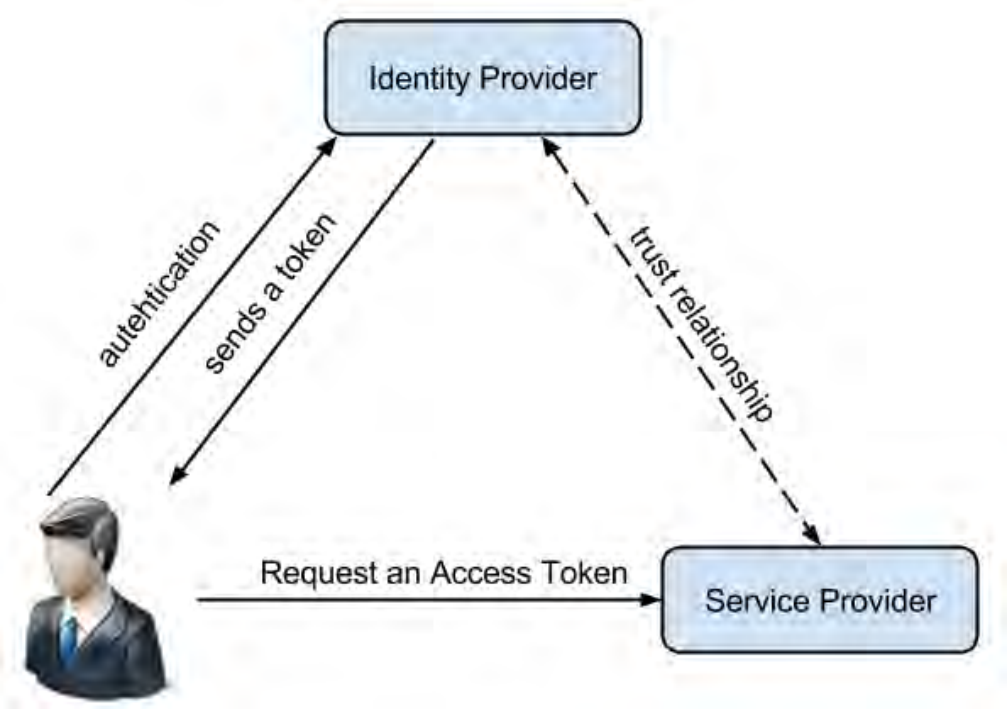

Fig. 1. Federated identity service model

IV. FEDERATED IDENTITY ADVANTAGES AND DISADVANTAGES

Federated Identity eliminates the need to keep different user credentials across multiple applications, and that results in easier administration and efficient access to resources. It plays an important role in the cloud to provision functions and services. As the relying party doesn't have access to user credentials, it receives only the identity information sent from the identity provider, which improves security. Federated identity also offers seamless user experience, increased manageability and extensibility [2].

Despite its benefits, adopting federated identity requires compliance with the complex landscape of regulations. Risks of unauthorized access to services are high enough that service providers sometimes require additional assurance from identity providers, which makes the process of establishing trust difficult as new protocols are developed and expectations are revised. Although the numbers are steadily increasing [2], they aren't many specifically proven solutions for cloud based services; therefore, we propose a number of useful scenarios to manage identity in the cloud.

V. PROPOSED SCENARIOS TO MANAGE IDENTITY IN THE CLOUD

In this section, we propose design solutions for recurring cloud identity challenges [13] (adding new identity provider, migrating to a new identity provider), and propose useful scenarios.

A. Authentication delegation

When a company develops a service and plans to offer it in the form of SaaS by moving it to the cloud, it is required that they allow their employees access wherever they are by using mobile devices. If they plan to offer this service to external users as a SaaS, the client company may not want to have identity replication; moreover, they want to have seamless access to the SaaS offering. The challenge is to figure out how authentication can work. For this scenario, we propose authentication delegation.

In order for the client or application from company B to have access to the application of company A seamlessly and without identity replication, we need to register its service domain as an identity provider for company A, as shown in figure 2:

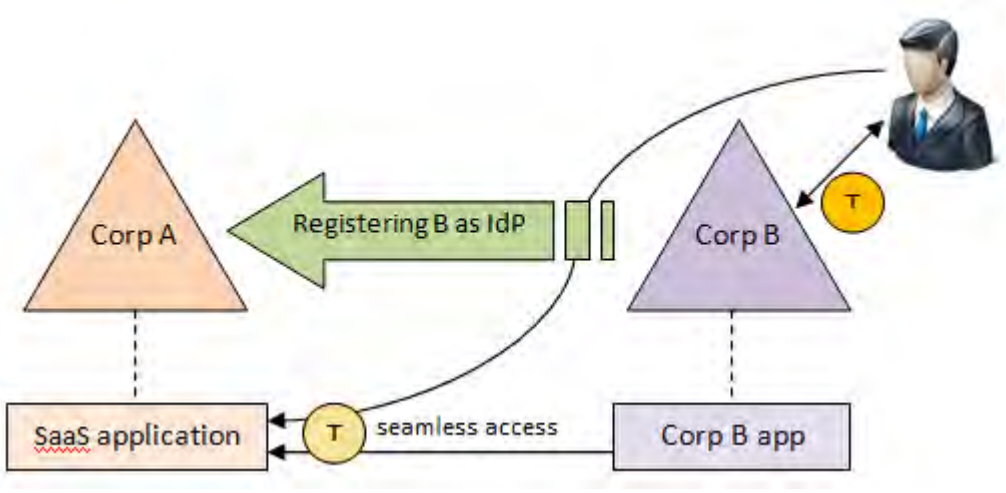

Fig. 2. Authentication delegation

The client flow is as follows:

- Registering the client organization B as an IdP (to avoid identity replication), allowing the client/application to use the token used internally to have access to an external application (authentication delegation).
- The client/application sends its token to external organization A.
- Organization A checks the token using the organization B as identity provider. If the token is valid, organization A generates a new token to be used.
- The client/application uses this new token to have access to the application in organization A.

B. Federation provider and security token management

Opening internal applications to multitenant (enterprises, users...) and multiple devices by moving them to the cloud is a challenge. Trust needs to be established across all identity providers. Consequently, enterprises face the new challenge of how to onboard the new user identities. For this scenario, we propose a federated provider across multiple identity providers.

Instead of attempting the monumental task of establishing trust across all identity providers and all relying parties, we use the layer of indirection principal by introducing a federation provider that acts as a central hub. This solution allows for a single point of integration to enterprise, as shown in figure 3:

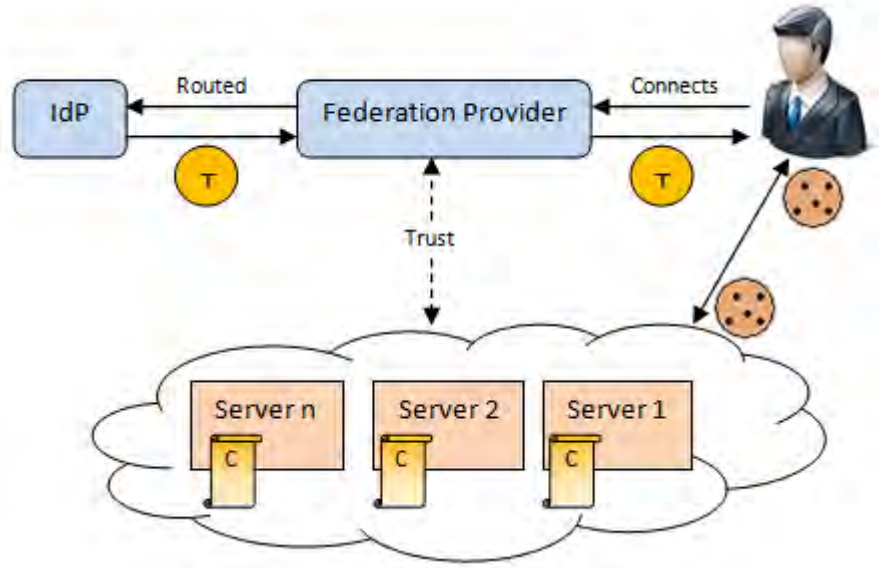

Fig. 3. Federation provider and security token management

The client flow is as follows:

- Client tries to access the SaaS offering. He gets a list of identity providers that are configured against the SaaS application.
- Client chooses a specific identity provider and passes on the credentials to their identity provider, in order to get authenticated.
- The identity provider (IdP) send a token to federation provider.
- The federation provider sends the token to client who sends it to the relaying party residing in the cloud.
- The relaying party extracts the claims presented inside the token and authorizes the user correspondently.

VI. CLAIMS AND TOKEN TRANSFORMATION

Claim-based authentication technologies are one of the prime federated identity implementation [16], but one of the common problems today is tokens mismatch and claims mismatch. Since mobile have limited resource and no XML stack (SOAP, SAML, WS-*,...) and the SaaS service might be using SAML [12] or SWT, we propose the use of claims and token transformation capabilities that allow changing incoming claims and token formats to outgoing claims and token formats by using a set of rules [11].

VII. CONCLUSION AND PERSPECTIVES

In this paper, we have highlighted the importance of digital identity that can be used to form the basis of data security; not only locally, but also in the cloud. We have successfully identified the common challenges to manage identity in the cloud, proposing a couple of design solutions for a number of scenarios that take into account the need of scale and openness in the cloud context [1]. These solutions may be implemented on any platform and technology. Managing identity in the cloud deals with protecting confidential information which makes identity theft [14][15] a serious problem to keep in mind. The proposed design scenarios focus on interoperability and scalability across the many different cloud systems, and create seamless interaction throughout the security model. Perspectives and future regarding this paper include implementing these scenarios using REST, OpenID, OAuth and JWT (JSON Web Token) to include mobile use cases.

REFERENCES

[1] R. Buyya, J. Broberg, A. M. Goscinski. “Cloud Computing: Principles and Paradigms”. John Wiley & Sons, 2011.
[2] D, Rountree. “Federated Identity Primer”. Elsevier, 2013.
[3] V. Winkler, “Securing the Cloud – Cloud Computer Security Techniques and Tactics”, Elsevier, 2011.
[4] D. Bell, L. LaPadula. "Secure Computer Systems: Unified Exposition and Multics Interpretation". The MITRE Corporation, Technical Report.1975.
[5] S. Black, V. Varadharajan. "A Multilevel Security Model for a Distributed Object-Oriented System Networks and Communications". HP Laboratories Bristol HPL, 90-74. 1990.
[6] R. Sandhu, E. Coyne, H. Feinstein, C. Youman. "Role-Based Access Control Models". IEEE Computer, 38-47. 1996.
[7] S. S. Rawat, N, Sharma. “A Survey of Various Techniques to Secure Cloud Storage”. International Journal of Computer Science and Network Security, VOL.12 No.3, March 2012.
[8] R. Ausanka-Crues. “Methods for Access Control: Advances and Limitations”. (2001): 1-5.
[9] L. Zhou, V. Varadharajan, M. Hitchens. “Enforcing role-based access control for secure data storage in the cloud”. The computer Journal, 2011.

[10] V. Goyal , O. Pandey, A. Sahai, B. Waters. “Attribute-based Encryption for Fine-grained Access Control of Encrypted Data”. In proc. of ACM CCS,2006.
[11] Blakley III, G. Robert, R. J. Cohen, I. M. Milman. "System and method for secure web server gateway access using credential transform." U.S. Patent No. 6,067,623. 23 May 2000.
[12] Security Assertion Markup Language (SAML) V2.0, Oasis, 200. [Online]. Available:, http://docs.oasis-open.org/security/saml/v2.0/saml-conformance-2.0-os.pdf
[13] J. Jensen. “Federated Identity Management Challenges”. Seventh International Conference on Availability, Reliability and Security, 2012.
[14] E. Maler and D. Reed, “The venn of identity: Options and issues in federated identity management,” Security & Privacy, IEEE, vol. 6, no. 2, pp. 16–23, 2008.
[15] Z. Ahmad, J-L. Ab Manan, S. Sulaiman. “User Requirement Model for Federated Identities Threats”. 3rd International Conference on Advanced Computer Theory and Engineering, 2010.
[16] M. Noureddinea, R. Bashroushb. “An authentication model towards cloud federation in the enterprise”. The Journal of Systems and Software, Elsevier, 2013.