A Taxonomy Model for Single sign-on Oriented towards Cloud Computing

Glauber C. Batista[1], Maurício A. Pillon[1], Guilherme P. Koslovski[1], Charles C. Miers[1], Marcos A. Simplício Jr.[2] and Nelson M. Gonzalez[3]
[1]*Santa Catarina State University (UDESC), Joinville, Brazil*
[2]*University of São Paulo (USP), São Paulo, Brazil*
[3]*IBM Watson Research Center, Yorktown Heights, U.S.A.*

Keywords: Cloud Computing, Single Sign-On, Taxonomy.

Abstract: Clouds can be seen as a natural evolution of the Internet, allowing the utilization of computing capabilities maintained by third parties for optimizing resource usage. There are several elements that compose the cloud infrastructure and its services, and all of them must operate harmoniously. In particular, to allow the creation and deployment of services resilient to internal and external threats, the observance of security aspects is essential. This includes the deployment of authentication and authorization mechanisms to control the access to resources allocated on-demand, a strong requirement for any cloud-based solution. With this issue in mind, several providers have recently started using some form of Single Sign-On (SSO) mechanism to simplify the process of handling credentials inside the cloud. In this work, aiming to provide a structured overview of the wide variety of mechanisms that can be employed with this purpose, we propose a classification of SSO systems for cloud services, which can be used as a model for comparing current and future designing instances of such mechanisms. In addition, to validate the usefulness of the proposed taxonomy, we provide a classification of existing cloud-oriented SSO solutions.

1 INTRODUCTION

Cloud computing is a model that supports ubiquitous, convenient, on-demand access to a shared pool of configurable resources that can be rapidly provisioned and released with minimal effort (Mell and Grance, 2011). As such, clouds can be seen as the natural evolution of the Internet, allowing the utilization of computing capabilities maintained by third parties in an optimized manner, potentially reducing costs (Velte et al., 2009). In order to provide a suitable service, the multiple elements that compose the cloud infrastructure must operate harmoniously. In particular, to allow the creation and deployment of services resilient to internal and external threats, the observance of security aspects is essential. This includes the deployment of authentication and authorization mechanisms to control the access to resources allocated on-demand, a key requirement for any cloud-based solution (Tavizi et al., 2012).

Whereas authentication is widely deployed in several cloud systems, each independent cloud services may end up adopting its own authentication mechanism. For example, it is not uncommon for cloud services to perform some basic authentication using an identity (e.g., a username) and a secret (e.g., a password), which is straightforward to deploy and use. However, such uncoordinated authentication approach usually harms the usability of these systems, at the same time that it hinders any possibility of integrating different services. Even worse, the lack of integration is also likely to impair the system's security itself, since users who are forced to remember several pieces of information to access different services are often compelled to set the same secret for most (if not all) of them. In addition, from the services' standpoint each authentication operation corresponds to a separate process, requiring the allocation of additional system resources and increasing the possibility of information leakage (You and Zhu, 2012).

Aiming to avoid such issues, many existing cloud services have adopted some form of SSO mechanism (Chadwick et al., 2013; Sette and Ferraz, 2014). The core characteristic of SSO solutions is to provide

Batista, G., Pillon, M., Koslovski, G., Miers, C., Simplício Jr., M. and Gonzalez, N.
A Taxonomy Model for Single Sign-on Oriented towards Cloud Computing.
DOI: 10.5220/0006784205730581
In *Proceedings of the 8th International Conference on Cloud Computing and Services Science (CLOSER 2018)*, pages 573-581
ISBN: 978-989-758-295-0

a unique, system-wide identifier to each user. Several SSO-enabled services can then rely on this identifier, providing pervasive authentication and authorization for its users (Urueña et al., 2014). The interest in such approach can be verified by the fact that SSO mechanisms have become prevalent across numerous services, especially among major providers such as Google (Google, 2017), Facebook (Facebook, 2017), and Microsoft (Microsoft, 2006).

Despite the comprehensive reach of SSO mechanisms for cloud services, the literature still lacks a clear organization or a taxonomy to classify these mechanisms, thus allowing clear comparisons between them. Indeed, even though standardization initiatives have been proposed for authenticating users toward cloud solutions (Ates et al., 2011; Hamlen et al., 2011), these efforts do not take into account the integration of SSO with services *inside* the cloud. Outside the cloud context, Clercq (Clercq, 2002) analyzed current SSO architectures, classifying them as simple and complex approaches, based on the description of the protocols used for each scenario. Pashalidis and Mitchell (2003) proposed a taxonomy for existing SSO solutions, organizing them in pseudo-SSO and true SSO systems, depending on whether the authentication process is performed directly with each service or centralized in an SSO entity, respectively (Pashalidis and Mitchell, 2003). Bhargav-Spantzel et al. (2006) proposed a taxonomy for user-centric SSO systems, focusing on credentials and their relationships (Bhargav-Spantzel et al., 2006). Albeit interesting, the first two works are quite high level, so very distinct solutions may end up under the same category, making the taxonomy less useful for fine-grained comparisons. In contrast, the latter is too focused on one particular type of system to be considered comprehensive, so it often fails to provide specific categories for complex environments where users have low control over their own identities, which is the case of the cloud (Ates et al., 2011).

Aiming to address this gap, in this paper we propose a classification of SSO systems suitable for cloud services, to be used as a model for current and future implementations of such systems. In addition, we provide a classification of existing SSO solutions which can be considered cloud-oriented, exemplifying the usefulness of the proposed taxonomy when compared with existing SSO categorizations. The rest of this paper is organized as follows. Section 2 gives an overview of SSO systems. Section 3 briefly discusses the studied taxonomies that are related to this work (albeit not focused on cloud computing). Section 4 provides a comparison between the existing taxonomies, discussing their limitations. Section 5 describes the proposed taxonomy and applies it in the classification of existing authentication solutions.

2 SINGLE SIGN-ON: OVERVIEW

The concept of authentication is based on the correct association of a subject to an identity (Bishop, 2004). In other words, via the authentication process, a subject claims to be the rightful owner of an identity, providing credentials that prove such ownership, and the system then verifies whether or not this claim is valid. This process can be carried out using several different types of credentials, such as something that the subject knows (e.g., a password), possesses (e.g., a smart card), is (e.g., static biometrics), does (e.g., dynamic biometrics), and/or some other verifiable property (e.g., the subject's location) (Bishop, 2004). For a higher security level, these credentials can be combined, leading to a multi-factor authentication mechanism. In this context, an SSO is an authentication mechanism that provides a unique identifier for a subject, whether it is a user or a (virtual) machine, so it can be authenticated on several services, potentially in a transparent manner. To provide this service, SSO mechanisms assume basically the existence of three elements:

- **Identity Provider (IdP)**: the authentication server responsible for issuing digital identities in a secure manner, whenever necessary;
- **Service Provider (SP)**: the application that requires the authentication of subjects (e.g., users or machines) before providing the service; and
- **Subject**: the end-user/application that accesses the service after being authenticated with the IdP's aid.

Figure 1 illustrates the basic architecture of SSO solutions and the interactions among these entities. The SSO-based authentication process starts with the subject accessing the SP and indicating which IdP should be used for authentication. The authentication agent embedded into the subject's application then obtains the credentials from the IdP *(2)*, where the subject's credentials are stored. This IdP then executes the authentication process toward the SP, granting access to the services the subject is authorized to use *(3)*. From the SP's standpoint, there is little difference between this process and a traditional authentication: as long as the SP supports the SSO mechanism, it still gains access to some information that represents the subject's credentials, the main difference being that in this case the entity providing these credentials is the IdP rather than the subject (Volchkov, 2001).

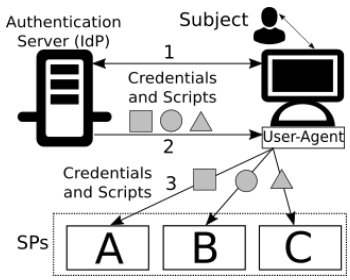

Figure 1: SSO system. Based on (Volchkov, 2001).

The authentication performed in this manner is said to be centralized in an IdP (Urueña et al., 2014; Sette and Ferraz, 2014), since a single authentication toward this entity grants access to several computing resources and applications in a distributed environment. In other words, after the IdP verifies that the user's claim over an identity is valid, the IdP itself can act on the subject's behalf whenever an authentication is required. In systems that adopt passwords as part of its authentication mechanism, for instance, a user does not need to remember several passwords for different services, nor ignore security best practices advising against password reuse on multiple sites: a single password, intended for the IdP, is all that is needed. The IdP can then protect those passwords accordingly, such as using some password hashing scheme (Andrade et al., 2016) for thwarting dictionary attacks in case the database is stolen, as well as captchas and/or throttling mechanisms for avoiding online password cracking attempts (Zhu et al., 2014). The IdP may also consider the user's password as a "master secret", using it to derive cryptographic keys for encrypting the database, providing further protection against database breaches. Similarly, in systems that rely on biometrics, the IdP is the only entity that collects and stores the user's biometric information, reducing the risk of its exposure and subsequent attacks (Galbally et al., 2012). Whenever the credentials need to be updated, this can be done directly with the IdP, which then becomes responsible for synchronizing this change toward all services.

Combined, these elements create an Identity Metasystem, i.e., an interoperable architecture of digital identities. This metasystem allows subjects to manage several digital identities using different underlying technologies, possibly from different implementations and providers (Hamlen et al., 2011). Indeed, SSO solutions have constantly evolved over time and currently rely on different mechanisms for synchronization and credential storage. Examples include the use of certificates (e.g., X.509), tokens (e.g., Kerberos), and attribute exchange mechanisms (e.g., OpenID).

3 EXISTING TAXONOMIES

Historically, different SSO protocols have been proposed, often targeted at different systems and requirements. This has led to several taxonomies, created to classify these protocols for specific contexts. Even though these taxonomies do not focus specifically on cloud computing, they provide a good overview of SSO solutions. For this reason, they are analyzed in this section.

3.1 Pashalidis and Mitchell

The taxonomy presented in (Pashalidis and Mitchell, 2003) is somewhat classical, being adopted for the classification of several SSO systems (Tiwari and Joshi, 2009; Clercq, 2002; Linden and Vilpola, 2005). It separates SSO systems in two main categories:

- **Pseudo-SSO** systems use an SSO component to manage the authentication credentials for each SP. The user authenticates against this component, which is responsible for conveying the operation to the different SPs. The relationship between identity and SP is $n:1$ – each identity is related to a single SP and the user can have multiple identities for a single SP.
- **True SSO** systems use an Authentication Service Provider (ASP), a role that is usually played by the IdP, to establish a relationship with each SP. This relationship requires some sort of trust level that is supported by a contractual agreement (Pashalidis and Mitchell, 2003). The authentication process occurs exclusively between users and the ASP, and SPs are then notified about the result via authentication assertions. These assertions contain the user's identity and the authentication state at the SP. The relationship between identities and SPs can then be $n:m$ – the user may have multiple identities in each SP, but may also prefer to use a same identity in different SPs.

Both pseudo-SSO and true SSO systems can operate in local mode or in proxy-based mode, leading to the four categories presented in Table 1.

3.2 Clercq

(Clercq, 2002) classifies SSO architectures as simple or complex, which are illustrated in Figure 2.

Simple SSO architectures use a single authentication authority with a same set of credentials for the user. Some examples include IBM Tivoli and Microsoft RADIUS. Having a single authority does not mean necessarily that there is only one authentication

Table 1: SSO taxonomy by (Pashalidis and Mitchell, 2003).

	Local	Proxy
Pseudo-SSO	**Local Pseudo-SSO**: The pseudo-SSO component resides in the user's machine, acting as a repository of credentials and scripts for authenticating the user. The user starts the authentication process toward the component and then the process is conveyed to the SPs.	**Proxy Pseudo-SSO**: The pseudo-SSO component resides in an external proxy server, which must be trusted by the user for storing its credentials and authentication scripts. Authentication is started by the user and redirected to the proxy, which executes the authentication process.
True SSO	**Local True SSO**: User authenticates toward locally managed ASP, which then sends authentication assertions to the SPs. There must an adequate trust level between the local ASP and the SPs, as well as a proper security infrastructure in place.	**Proxy True SSO**: The ASP resides in an external server that operates as a broker between users and SPs, issuing assertions regarding the authentication. The ASP must be trusted, otherwise it could easily impersonate a user by sending an assertion to a SP.

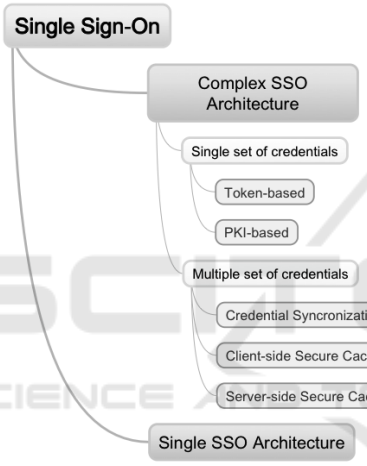

Figure 2: Clercq Architecture (Clercq, 2002).

server and one credential database available, though. In fact, for improved performance and scalability, a single authentication authority might consist of multiple authentication servers and several replicated credential databases (Clercq, 2002).

Complex SSO architectures involve multiple authentication authorities with one or many sets of credentials for each user, which allows them to cover distinct administrative domains implemented in different platforms (potentially managed by multiple organizations). They are further divided in two subclasses: systems that use a single set of credentials and systems that employ multiple sets. The former usually rely on two types of credentials:

- **Token-based**: Users are authenticated toward the authentication authority and receive a token, which can be used in subsequent authentications toward services from the same domain or to prove the user's identity toward a secondary authority (like in Kerberos (MIT, 2015));
- **PKI-based**: Rely on a Public Key Infrastructure (PKI), which means that the user first registers in a trusted Certification Authority (CA) or in a Registration Authority (RA). During the authentication process, the user is identified via a set of credentials, leading to the generation of an asymmetric pair of cryptographic keys. The possession of the public key certificate together with the corresponding private key allows the user to generate signed tokens for authentication (e.g., in a challenge-response protocol).

Architectures with multiple sets of credentials, on their turn, can be subdivided in basically three types:

- **Credential Synchronization**: The users' credentials are synchronized among several applications, so they all see the same set of credentials (e.g., a same password). Usually, the credentials are stored in a single database that is accessed by those applications. As a result, when the credentials are updated, the synchronization between services occurs in a seamless manner.
- **Client-side Secure Caching**: Users are able to locally store service credentials, which are usually protected by a master password.
- **Server-side Secure Caching**: Users store credentials in a remote server, which becomes the primary database of credentials. This database maps primary and secondary credentials of the user, such as a master password and a service-specific password, respectively; the latter can then be stored by the corresponding services, which constitute secondary authentication domains.

It is interesting to notice that, even though Clercq's taxonomy for SSO solutions includes more classes than (Pashalidis and Mitchell, 2003), they actually use similar concepts in their classification. For example, Clercq's client and server-side credential storage classes are analogous to the pseudo-SSO and true SSO based on proxies from Pashalidis and Mitchell's taxonomy.

3.3 Bhargav-Spantzel Taxonomy

The taxonomy by Barghav-Spantzel et al. (2006) addresses the SSO solution from the user's standpoint (Bhargav-Spantzel et al., 2006). For this reason, it is currently well-accepted for classifying SSO systems in the context user-centered identity management solutions (Han et al., 2010; Suriadi et al., 2009; Zhang and Chen, 2010; Zhang and Chen, 2011). Basically, the taxonomy proposes a classification based on two main paradigms: relationship-focused and credential-focused, discussed in what follows.

3.3.1 Relationship-focused

A relationship-focused SSO system manages only the relationship among IdP, SP, and user. For each transaction, the user queries an IdP with which it is registered and dynamically obtains information. Relationship-focused systems use short-lived tokens, valid only during a limited set of transactions. The pros and cons of this type of system are as follows:

- **Pros:**
  - Short-lived tokens limit the risk and potential damage in case they are stolen or intercepted;
  - As long as the IdP is online, it can keep the authentication information up-to-date;
  - In general, they are lightweight systems and do not required a robust user-side client application; and
  - They only require an effective asymmetric cryptographic solution.
- **Cons:**
  - The need of having an always-online IdP turns it into a Single Point of Failure (SPF);
  - As the IdP is always involved in the transactions, the user activities can be monitored, so the IdP needs to be fully trusted; and
  - If there is a lot of token transitivity, the risk of impersonation issues grows. A token is considered transitive if the component that receives the token can somehow use it to impersonate its owner (e.g., if the authentication mechanism assumes the bearer of the token is its rightful owner for any authentication context). Several existing SSO systems include mechanisms to prevent this issue; for example, SAML (OASIS, 2015) and Liberty (Liberty Alliance, 2015) explicitly define who is the entity to which the token is intended, so any attempt to forward the received token to a third party would lead to its rejection as the token's recipient information would not match that of the third party.

3.3.2 Credential-focused

A credential-focused system is characterized by the fact that it directly manages the credentials. This includes, for instance, a client application that stores the user's long-term tokens in a local database: in this case, the user can reuse the credentials issues by the IdP for multiple transactions, even without contacting the IdP once again. To be practical, such systems must use long-lived credentials (e.g., X.509 certificates (Cooper, 2008)), as otherwise the locally stored credentials would become quickly useless. Some pros and cons of credential-focused solutions are:

- **Pros:**
  - By definition, the IdP is offline during the transactions. Hence, it is unable to trace the users' activities and, if it goes offline, the authentication system's availability is not critically affected (except, obviously, that new tokens cannot be issued).
  - The tokens generated from the long-term credentials are necessarily not transitive, as otherwise the users would be vulnerable to impersonation attacks. Such non-transitivity is ensured by mechanisms such as challenge-response protocols involving nonces, so a token generated for a certain context cannot be reused in a different context.
- **Cons:**
  - Credential theft or undesired sharing could lead to severe risk and damage to the system. Hence, it is fundamental to avoid any sort of credential sharing in such systems.
  - Credential loss requires a mechanism to revoke it. If a credential is revoked, the user cannot access any service associated to it until a new valid credential is issued.
  - Credential-focused systems usually lead to a higher workload on the client application, thus requiring more robust client applications (and client-side computational resources).

3.3.3 Limitations

Even though Bhargav-Spantzel's taxonomy model leads to a quite succinct and comprehensive abstraction for SSO mechanisms, the fact that it focus on user-centered systems make it less useful for analyzing identity-centered architectures. In other words, this taxonomy is interesting to evaluate SSO systems in which the users can have multiple identities to log on several services, potentially using different identity providers, as it is the common case of the Internet. However, it does not cover well solutions that rely on a locally centralized identity provider, in which it is important to ensure that each entity is identified unequivocally for administrative purposes; this is the case of corporate or federated systems, and also of many cloud environments.

4 COMPARISON AND EVALUATION OF EXISTING TAXONOMIES

A first aspect that must be considered in a comparison among the three presented taxonomies refers to its coverage of the user-centered and/or identity-centered paradigms. Namely, while the taxonomy by Bhargav-Spantzel et al. exclusively focuses on user-centered solutions, the other two taxonomies address the whole SSO spectrum.

Another interesting aspect refers to the core method or categories employed in the construction of the proposed taxonomies for SSO solutions. Pashalidis and Mitchell's taxonomy classifies the systems mainly in terms of locality, distinguishing pseudo-SSO and true SSO systems (local or remote). Clercq architecture, in turn, uses complexity as the primary metric for classifying SSO systems, and then consider the amount of credential information handled by them (single set or multiple set of credentials). Finally, the Bhargav-Spantzel taxonomy focuses on the interactions between users and the IdP, distinguishing systems where the interactions are strong and the tokens are short-lived (relationship-focused) from those with less frequent interactions and long-lived tokens (credential-focused).

Even though it is not possible to define which taxonomy is best suited for all scenarios, the analysis of the literature shows that the taxonomy by Pashalidis e Mitchell is the most comprehensive and well-accepted. Therefore, it is reasonable to use it as a reference for defining more detailed sub-classes. Indeed, the relationship between (Pashalidis and Mitchell, 2003) and (Clercq, 2002) architectures has already been described in (Linden and Vilpola, 2005): as illustrated in Table 2, Pashalidis and Mitchell's taxonomy (in the rows and columns) can be approximately mapped to the classes defined in Clercq's taxonomy (in the table's internal cells). The main limitation of this mapping is that, even though credential synchronization is accounted for in (Pashalidis and Mitchell, 2003) original article, there is not a specific category for it in their taxonomy.

Table 2: Comparison between Pashalidis and Mitchell's taxonomy and the taxonomy by Clercq. Adapted from (Linden and Vilpola, 2005).

	Local SSO systems	Proxy-based SSO systems
Pseudo-SSO Systems	Secure client-side credential caching	Secure server-side credential caching
True SSO Systems	PKI-based SSO systems	Token-based SSO systems

The relationship between the taxonomy by (Bhargav-Spantzel et al., 2006) and the other two taxonomies hereby described is illustrated in Figure 3. Figure 3 shows that the credential-focused and relationship-focused systems can be seen as subtypes of the PKI-based and token-based systems, respectively. Therefore, Pashalidis and Mitchell's taxonomy (and, consequently, Clercq's) can be seen as more comprehensive, including categories for the classification of specific systems as those covered by (Bhargav-Spantzel et al., 2006).

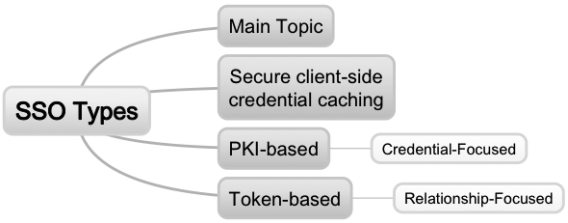

Figure 3: Relationship among the taxonomies hereby investigated.

Whereas a generic taxonomy allows several systems to be classified under its structure, the lack of more fine-grained classes available in more specific taxonomies may reduce their usefulness when a detailed analysis and comparison is necessary. After all, systems with quite different properties end up in the same (high-level) category, the comparison between them will not rely on the taxonomy itself, but in details not covered by it. On the other hand, a user-centered taxonomy approach as the one proposed by Bhargav-Spantzel et al. does not necessarily cover well SSO systems that are not user-centered themselves. In the specific case of cloud systems, this may hinder the classification and comparison of different authentication models, since they must be able to address (Hamlen et al., 2011):

- Several trust relationships;
- Several access control policies, based on roles and attributes;
- The provisioning of services in real time;
- Authorization services; and
- Audit and accountability services.

In addition, one particularity of the identity management in cloud systems is that in such environments the identity of users is fragmented in silos, and usually is not entirely on the user's control (Ates et al., 2011). Such specificity motivates the creation of a cloud-oriented SSO taxonomy that allows the identification of authentication systems that conciliate those properties (e.g., by returning the control of the system's identities to the users).

5 PROPOSED TAXONOMY

The proposed taxonomy is based on the one presented by Pashalidis and Mitchell (Pashalidis and Mitchell, 2003), but adopts a hierarchical approach instead of a horizontal one to account for particularities of cloud systems. As a result, this approach remains concise in its initial levels, but in its deeper levels allows a more detailed classification of SSO systems even without focusing on a particular (e.g., user-centered) scenario. Similarly to the work proposed in (Pashalidis and Mitchell, 2003), it is possible to use the proposed taxonomy as a starting point for further more detailed analysis of cloud SSO systems. The resulting hierarchical structure is illustrated in Figure 4, which shows the following categories:

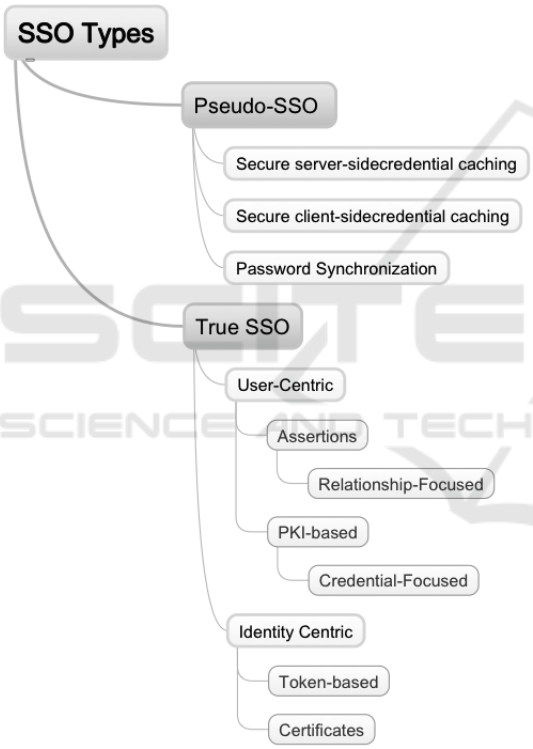

Figure 4: Proposed taxonomy model for Single Sign-On oriented towards cloud computing.

- **Pseudo-SSO**: Pseudo-SSO systems can be classified as: local, proxy-based, or synchronization-based. As noted in Section 4, this explicit inclusion of synchronization-based systems avoids the need of classifying them as local or proxy-based, conciliating the taxonomies by Pashalidis and Mitchell, and Clercq.
  - *Password synchronization*: Systems based on password synchronization between the services. They are typically represented by systems that use a master database for such synchronization, thus implementing transparent and seamless integration between any other credential databases.
  - *Secure client-side credential caching*: Systems that allow local management of credentials. Credentials are stored in a local database via a master password or other authentication mechanism (e.g., biometrics).
  - *Secure server-side credential caching*: Systems that allow remote storage and management of credentials. These credentials are replicated and secured by a trusted remote server, including secondary credentials used to authenticate other service domains.
- **True SSO**: True SSO systems are organized in local and proxy-based. The proposed taxonomy explicitly divides these categories, but the analysis of currently existing solutions shows that PKI-based systems end up being local, while systems based on tokens are proxy-based.
  - *User-centric*: Systems that are focused on the user instead of the SPs. In this model, the user may choose the IdP that will be responsible to process the authentication request. User-centric systems are based on assertions about the user, which are used for the communication between the IdPs and SPs, or employ PKI for that purpose. Systems that are based on assertions are typically relationship-focused, while those based on PKI are usually credential-focused.
  - *Identity-centric*: Systems focused on the user's identity in the SP. These systems are also called "centralized authentication solutions", as in those cases the users are limited to authenticate against a particular IdP.

5.1 Classifying Cloud-oriented SSO Solutions

Aiming to illustrate the usefulness of the proposed taxonomy, it is interesting to discuss how it addresses the classification of SSO systems used in cloud computing environments. First, we note that user-centered solutions such as OpenID (OpenID, 2015) and SAML (OASIS, 2015) are widely adopted and supported by cloud services, both in private clouds such as OpenStack (Openstack, 2017) and in public ones such as Amazon Web Services (AWS) (Amazon, 2015). In this context, as in Bhargav-Spantzel et al.'s taxonomy, it is possible to distinguish user-centered mechanisms that are relationship-focused from those

that are credential-focused.

The taxonomy also allows the characterization of SSO systems in different specificity levels, depending on the how much detail is necessary. For example, in a high level, Kerberos and OpenID are both true proxy-based SSO systems. However, under a finer-grained perspective, Kerberos can also be classified as identity-centered, using tokens for the communication between the system's elements. In contrast, OpenID is a user-centered proxy-based true SSO system, using assertions (typically conveyed via tokens) for this communication. Hence, even though both Kerberos and OpenID rely on tokens and centralize the authentication process on an SSO entity, the approaches adopted by them can be clearly distinguished using the proposed taxonomy.

Whereas the proposed taxonomy aims to be quite comprehensive, we emphasize that it might not remain as complete with the development of new systems. Nevertheless, as long as novel approaches can still be classified in the high-level categories covered, it can be extended to accommodate such solutions in its hierarchy. In addition, some SSO solutions combine different mechanisms for the same domain, leading to a more complex system that does not fit directly into a single category of the proposed taxonomy. For example, OpenStack running its Identity API version 3 (Openstack, 2015) is able to use both OpenID and Kerberos.

Table 3: Classification of existing SSO solutions using the proposed taxonomy.

Solution	Main category	Subcategories
Azure Active Directory (Microsoft, 2015a)	Pseudo-SSO	Credential synchronization
CardSpace (Microsoft, 2006)	Pseudo-SSO	Secure client-side credential caching
Entrust Cloud PKI (Entrust, 2015)	Pseudo-SSO	Secure server-side credential caching
Facebook Login (Facebook, 2017)	True SSO	User-centric; Assertions; Relationship-focused
FIDO (FIDO, 2015)	Pseudo-SSO	Secure client-side credential caching
Kerberos-based (MIT, 2015)	True SSO	Identity-centric; Tokens
Liberty Alliance (Liberty Alliance, 2015)	True SSO	User-centric; Assertions; Relationship-focused
Microsoft Passport (Microsoft, 2015b)	True SSO	User-centric
OpenID (OpenID, 2015)	True SSO	User-centric; Assertions; Relationship-focused
SAML (OASIS, 2015)	True SSO	User-centric; Assertions; Relationship-focused
Shibboleth (Shibboleth,)	True SSO	User-centric; Assertions; Relationship-focused
X.509 (Cooper, 2008)	True SSO	User-centric; PKI-based; Credential-focused

Consequently, since the resulting authentication system is based on two distinct models, it can only be placed on a higher level category, e.g., being classified generically as a proxy-based true SSO system.

For a more complete view of the authentication ecosystem in the cloud environment, Table 3 shows the classification of several SSO systems using the proposed taxonomy.

In this table, SSO systems are classified first using a coarse-grained level, and then into the fine-grained level of the proposed taxonomy. Several SSO systems used in cloud computing services and solutions are covered, even when they were developed for particular contexts. Therefore, its fine-grained levels should aid in the comparison of distinct solutions to find the most suitable to a particular cloud service and context.

Compared to the other taxonomies investigated in this paper, the one hereby proposed allows a more precise classification of SSO systems, since it is the result of combining generic architectures with more specific ones.

6 CONSIDERATIONS & FUTURE WORK

SSO solutions have been widely adopted throughout the years, in especial to solve the complex problem of credentials management. In addition, SSO mechanisms have also contributed for identity management among several services, enabling authentication in a more user-friendly manner. This characteristic is of particular interest in cloud environments, in which the development of new services leads to a constant need of integrating different systems.

Aiming to facilitate the comparison among SSO mechanisms that can be employed in the cloud scenario, this article presents a taxonomy for existing solutions, conciliating categories from different taxonomies available in the literature. More specifically, the proposed taxonomy builds upon the work in (Pashalidis and Mitchell, 2003), taking advantage of its generality and extending it with finer-grained categories, in special those from the taxonomies proposed in (Bhargav-Spantzel et al., 2006; Clercq, 2002). As a result, it allows a clearer analysis and evaluation of several characteristics of the target SSO solutions, as well as a comparison with alternative approaches for identifying the best suited for a specific scenario.

One important characteristic of the proposed taxonomy is the extensibility applied to base categories provided in (Pashalidis and Mitchell, 2003), which should allow it to cover new solutions as they are created and implemented. To further explore this capability, as future works we plan to investigate other SSO solutions that might lead to an enhanced taxonomy, with even finer-grained categories.

ACKNOWLEDGEMENTS

The authors would like to thank the support of the LabP2D (Laboratory of Parallel and Distributed Processing) / UDESC (Santa Catarina State University), providing its facilities and resources to the accomplishment of this research.

This work was in part supported by the Brazilian National Council for Scientific and Technological Development (CNPq) under grant 301198/2017-9.

REFERENCES

Amazon (2015). Amazon web services.

Andrade, E., Jr, M. S., Barreto, P., and Santos, P. (2016). Lyra2: Efficient password hashing with high security against time-memory trade-offs. *IEEE Transactions on Computers*, 65(10):3096–3108.

Ates, M., Ravet, S., Ahmat, A., and Fayolle, J. (2011). An identity-centric internet: Identity in the cloud, identity as a service and other delights. In *6th Int. Conf. on Availability, Reliability and Security (ARES)*, pages 555–560.

Bhargav-Spantzel, A., Camenisch, J., Gross, T., and Sommer, D. (2006). User centricity: A taxonomy and open issues. In *Proc. of the 2nd ACM Workshop on Digital Identity Management*, DIM '06, pages 1–10, New York, NY, USA. ACM.

Bishop, M. (2004). *Introduction to Computer Security*. Addison-Wesley Professional, 1st edition.

Chadwick, D. W., Siu, K., Lee, C., Fouillat, Y., and Germonville, D. (2013). Adding federated identity management to OpenStack. *Journal of Grid Computing*, 12(1):3–27.

Clercq, J. D. (2002). Single sign-on architectures. In *Proc. of the Int. Conf. on Infrastructure Security (InfraSec'02)*, pages 40–58, London, UK, UK. Springer-Verlag.

Cooper, D. (2008). Internet X.509 public key infrastructure certificate and certificate revocation list (CRL) profile.

Entrust (2015). Entrust.

Facebook (2017). Facebook login.

FIDO (2015). FIDO Alliance.

Galbally, J., Ross, A., Gomez-Barrero, M., Fierrez, J., and Ortega-Garcia, J. (2012). From the Iriscode to the iris: A new vulnerability of iris recognition systems. Technical report, Black Hat USA. Available: https://media.blackhat.com/bh-us-12/Briefings/Galbally/BH_US_12_Galbally_Iris_Reconstruction_WP.pdf.

Google (2017). Google identity platform.

Hamlen, K., Liu, P., Kantarcioglu, M., Thuraisingham, B., and Yu, T. (2011). Identity management for cloud computing: Developments and directions. In *Proc. of the 7th Annual Workshop on Cyber Security and Information Intelligence Research*, CSIIRW '11, pages 32:1–32:1, New York, NY, USA. ACM.

Han, J., Mu, Y., Susilo, W., and Yan, J. (2010). A generic construction of dynamic single sign-on with strong security. In *Security and Privacy in Communication Networks*, pages 181–198. Springer.

Liberty Alliance (2015). Liberty Alliance project.

Linden, M. and Vilpola, I. (2005). An empirical study on the usability of logout in a single sign-on system. In *Information Security Practice and Experience*, number 3439 in LNCS, pages 243–254. Springer Berlin Heidelberg. DOI: 10.1007/978-3-540-31979-5_21.

Mell, P. and Grance, T. (2011). The NIST definition of cloud computing.

Microsoft (2006). Introducing Windows CardSpace.

Microsoft (2015a). Azure active directory.

Microsoft (2015b). Microsoft passport.

MIT (2015). Kerberos: The network authentication protocol.

OASIS (2015). Saml xml.

OpenID (2015). Openid.

Openstack (2015). Identity API v3.

Openstack (2017). Openstack – open source software for creating private and public clouds.

Pashalidis, A. and Mitchell, C. (2003). A taxonomy of single sign-on systems. In *Information security and privacy*, pages 249–264. Springer.

Sette, I. and Ferraz, C. (2014). Integrating cloud platforms to identity federations. In *2014 Brazilian Symposium on Computer Networks and Distributed Systems (SBRC)*, pages 310–318.

Shibboleth. Shibboleth.

Suriadi, S., Foo, E., and Jøsang, A. (2009). A user-centric federated single sign-on system. *Journal of Network and Computer Applications*, 32(2):388–401.

Tavizi, T., Shajari, M., and Dodangeh, P. (2012). A usage control based architecture for cloud environments. In *26th International Parallel and Distributed Processing Symposium Workshops PhD Forum*, pages 1534–1539. IEEE.

Tiwari, P. B. and Joshi, S. R. (2009). Single sign-on with one time password. In *1st Asian Himalayas Int. Conf. on Internet*, pages 1–4. IEEE.

Urueña, M., Muñoz, A., and Larrabeiti, D. (2014). Analysis of privacy vulnerabilities in single sign-on mechanisms for multimedia websites. *Multimedia Tools and Applications*, 68(1):159–176.

Velte, T., Velte, A., and Elsenpeter, R. (2009). *Cloud computing, a practical approach*. McGraw-Hill, Inc.

Volchkov, A. (2001). Revisiting single sign-on: a pragmatic approach in a new context. *IT Professional*, 3(1):39–45.

You, X. and Zhu, Y. (2012). Research and design of web single sign-on scheme. In *IEEE Symposium on Robotics and Applications (ISRA)*, pages 383–386.

Zhang, Y. and Chen, J.-L. (2010). Universal identity management model based on anonymous credentials. In *2010 IEEE International Conference on Services Computing*, pages 305–312. IEEE.

Zhang, Y. and Chen, J.-L. (2011). A delegation solution for universal identity management in SOA. *Services Computing, IEEE Transactions on*, 4(1):70–81.

Zhu, B. B., Yan, J., Bao, G., Yang, M., and Xu, N. (2014). Captcha as graphical passwords: A new security primitive based on hard AI problems. *IEEE Transactions on Information Forensics and Security*, 9(6):891–904.