Securityware: Towards an Architecture for User-Centric Approach

Ali Hammami, Noëmie Simoni
TELECOM ParisTech - LTCI - UMR 5141 CNRS
46, rue Barrault F 75634 Paris Cedex 13 - France
e-mail: ali.hammami—noemie.simoni@telecom-paristech.fr

***Abstract*— With the rapid evolution of next generation networks, telecommunication field are evolving towards user-centric approach. In fact, users demand the access to any service without any technical, temporal or geographical barrier. This approach permits user to personalize their services and have a unique, dynamic and seamless session despite heterogeneity and mobility of his environment. In this context, many security issues arise. In order to overcome security challenges while respecting user-centric requirements, we propose in this paper a new security architecture that is based on a set of service components. These components ensure a secured and simplified service access in a unique and seamless session. They should be reusable, interoperable, autonomous, mutualizable, interconnected and self-managed in order to participate in a secure, personalized and dynamic service composition. We describe at the end of this paper an application scenario and show the feasibility of our proposition.**

Keywords-User-centric; Security as a service; Security architecture; Secure dynamic service composition; Secure seamless and unique session.

I. INTRODUCTION

Nowadays, end-users demand to be constantly connected anywhere, anytime and anyhow. They want to have a dynamic session according to their preferences and the desired QoS (Quality of Service). In this user-centric context, end-users require a unique, personalized and secure session. Furthermore, this session should permit them to access their services in a continuous and seamless way. And, it should offer them an end-to-end QoS. The main goal of this user-centric approach is to make the whole system serve the user without any constraints such as treatment and hardware constraints (system-centric), application constraints (application-centric) or connection constraints (network-centric).

Moreover, with the emergence of next-generation networks, heterogeneous environment (multiple terminals, access networks, operators and service providers) and mobility requirements (user, terminal, network and service mobility) present two challenges that face to ensure a user-centric session. Thus, user-centric solutions should harmonize service personalization, heterogeneity and mobility aspects.

In this context, security represents an important concern. In fact, this user-centric session should obviously be secured. However, many challenges arise. In such environment, system boundaries, which were well delimited, become increasingly open. Indeed, there are multiple services which are unknown in advance and multiple communications between services and with users.

Besides, heterogeneity of involved resources (terminals, networks and services) in the user session increases the complexity of security tasks. In addition, the different types of mobility (user, terminal, network and service mobility) affect the user-centric session that should be unique, secure and seamless and ensure continuity of services. Mobility impacts strongly on an end-to-end secured service delivery.

The main questions we need to answer in this step are: How security should be conceived to be efficient in service-oriented environments? How can security support a personalized and flexible service composition for the user who desires a unique and seamless session? How can we have a secure user-centric session in a dynamic and mobile context?

This open, heterogeneous and mobile environment presents significant risks in terms of security and becomes increasingly vulnerable. Therefore, a new security solution, which responds to user-centric requirements in NGN environment, should be provided. For this purpose, we propose a novel security architecture that respects user-centric approach, called Securityware. We adopt new criteria in Service Oriented Architecture (SOA) which ensure a dynamic and seamless service composition. This architecture is organized in four visibility levels (equipment, network, service and user) and each resource (terminal, network and service) is considered as a service component. Our solution does not afford security mechanisms in a static and centralized way. Security is provided as a service due to a set of service components (Securityware).

The remainder of this paper is organized as follows. In Section 2, we discuss some existing approaches regarding security as a service. In Section 3, we present our proposed security solution which is based on security as a service approach. First, we describe the major security service characteristics that are based not only on simple SOA principles, but also on new features to provide a secure, personalized, dynamic and seamless service composition. Second, we detail the components of our security architecture. Third, we describe how to ensure a secure service composition in user-centric context. In Section 4, we present an application scenario and show the feasibility of our proposition. Finally, conclusion and perspectives for future work are presented in Section 5.

 ISBN: 978-1-61208-209-7

II. RELATED WORK

Before presenting our proposal, we analyze the content of some related technologies and research works, which deal with security in Service Oriented Architecture. We will focus on existing approaches related to security architectures, and we will discuss important research activities on the notion of security services.

P. Qi-rui et al. propose in [1] a unified authentication and authorization solution which supports the SOA-based distributed systems. This proposal is based on Service-Access-Agent architecture. It uses a mechanism of two authentication levels to separate in-domain and inter-domain authentication for more efficiency and configuration simplicity. The general framework of this solution is composed of GAAC (Global Authentication and Authorization Centre), SAA (Service Access Agent) and Web services. GAAC ensures inter-domain authentication by authenticating SAAs and provides centralized access authorization control to Web services. SAA is responsible to control and manage the Web services within local security domain. This solution provides authentication and authorization at the service level. This is an interesting contribution but the centralized adopted approach is monolithic and vertical and has a strong coupling. This raises the following questions: Is this sufficient in our cross-organizational and ubiquitous context? How can we have more flexibility, dynamicity and extensibility in security functions?

E. Bertino et al. [2] propose a reference architectural framework based on security services such as identification, authentication and authorization services. They discuss how each component of the security pipeline is considered as a service according to SOA principles. They discuss also mechanisms for coordinating different security services. Then, an event-based approach is used to disseminate relevant security events to all the potentially affected services. To this end, a notification service is integrated in the architectural framework to notify security services of relevant events. This work deals with security as a service approach which is particularly promising for Service Oriented Architecture. Nevertheless, it does not specify how security functions could be ensured in a transparent and simplified way to user in a dynamic and user-centric service composition. This raises the question: How to provide a secure and seamless session in a dynamic environment according to user requirements and preferences?

The service-oriented security architecture which is described in [3] and [4] consists of three layers. The first layer contains the well-defined and stable service interfaces to be used by other services. The authentication interface provides operations to authenticate a user (entering username and password) and generates a temporary security token. An access control decision can be delegated to the authorization verification service interface. In functional layer, the secure token service validates the claimed identity of the service consumer. A policy decision point (PDP) evaluates policies and makes authorization decision. These functional components store and retrieve their data from components placed in the data layer below. This work focuses only on interoperability criterion while many others criteria should be specified in order to satisfy new approaches such as user-centric approach that requires personalization, dynamicity, and transparency within a secure and seamless session.

All the cited works above present the good efforts to provide a conceptual views and approaches about security as a service to be managed in a Service Oriented Architecture. According to SOA, the service layer is like a big service pool with components of different functions. This approach brings more flexibility and allows SPs to respond more quickly to users requirements. Could the current solutions actually ensure flexibility if they guard vertical constraint between client and server and if we want to take into account mobility effects? We think that is not so because it is missing some criteria such as mutualization and self-management to have effective flexibility and dynamicity. In fact, mutualization consists of having not only reusable but also shareable components in different contexts of different users. Then, security components should be mutualizable to be used in every context by many users without any constraints. Current SOA approach does not specify also self-management criterion of service components that permits to have a seamless secure session with the desired level of security and QoS requirements despite mobility and changes due to dynamic service composition.

Meanwhile, service composition only on the service layer is not enough. Nowadays, with the existence of heterogeneous networks and terminals, which can greatly affect user experiences, we should opt for a new perception able to homogenize the different resources; the components of the network layer as well as the terminal layer should be perceived as services. Thereby, the security components should consider the terminal as a service. We demonstrate, then, in our proposals, how we can manage and ensure the security and continuity of user session.

III. PROPOSITION

The challenge is how to guarantee security, continuity and uniqueness of end-users sessions in a user-oriented and dynamic context. To overcome this problem, we present in this Section a security solution which is based on security as a service approach. These security service components should satisfy a set of criteria to be used in a dynamic service composition (Section 3.1). Then, we detail the components of our security architecture (Section 3.2). Finally, we describe how to ensure a secure service composition in user-centric context (Section 3.3).

A. Security service component characteristics

The security services should have basically the same criteria as specified in SOA. The characteristics of our security services are based not only on simple SOA principles but also on new features to provide a secure, personalized, dynamic and seamless service composition.

A security component is an element of the system that performs a predefined service and is able to communicate with other components. Among the basic characteristics, a security service component should be:

 ISBN: 978-1-61208-209-7

- *Reusable*: it must be a unit of reuse. An adaptation phase may be necessary depending on the execution context.
- *Autonomous*: this feature means that a service component is functionally independent. It is self-sufficient and it is able to perform its functionalities without needing another component.

In order to ensure flexibility, dynamic service composition and personalization regarding the user-centric context, our proposition is not limited to SOA characteristics. On the contrary, the added value of our solution consists in adding the following criteria:

- Mutualizable: this feature consists of sharing, simultaneously and between several users, resources that a service can offer. This permits to optimize use of these resources. This aspect requires that all constraints related to the preservation of the service state (activation, deactivation, etc.) or to particular user data be eliminated. This means that services should be stateless. Therefore, a service should perform generic tasks to all users without considering their contexts and maintaining their specific data. This criterion helps to have a dynamic replacement of a service in a loosely coupled service composition.
- Interconnected: this feature is just added to the interoperability one and ensures thereby interoperation between several service components in order to create a global service. This approach enables the cooperation of service components while avoiding treatment conflicts and deadlocks. Interconnection between services leads to define imperatively generic interfaces and links.
- Self-managed: this feature is set by the role of the QoS agent that is integrated on each service component to supervise and control its own QoS parameters. It permits to verify whether a service behavior respects the QoS agreement.

B. Security Architecture

To ensure security during user-centric session, it is necessary to apply security mechanisms on all service components involved in this session. For this purpose, we conceive all mechanisms as services. Thus, these services should be designed with specific features as mutualization, autonomy, interoperability and self-management. Moreover, we consider elements that constitute the different layers (equipments, networks and applicative-services) as service components.

In our context, the user-centric session is a juncture of the different session parts. In order to ensure security, continuity and uniqueness of the session and to best answer the end-user's preferences, we propose a security architecture whose components are involved in the different phases of session establishment. First, in the terminal access phase, the terminal must be able to authenticate the user as the legitimate owner or user of the device. This phase is ensured by services offered by the Userware in the terminal. Second, in the network access phase, most possible attacks should be identified because remote platforms are trying to be connected across mistrusted actors. To overcome the locks of trust, we use encryption and authentication mechanisms. The final phase involves the service (application) access. Regarding user rights, authorization service performs secure access to the required services.

The Next Generation Network/Next Generation Service (NGN/NGS) context (convergence, heterogeneity, mobility, mutualization, etc.) requires new mechanisms and techniques which can simplify the different layer access. Thereby, we adopt a distributed architecture that guarantees a loose coupling in order to overcome constraints arising in centralized and monolithic architectures. Moreover, our architecture is based on the two key concepts: security as a service and visibility levels. Then, due to a set of security components, we assure to the user a secured and simplified access to his services in the terminal level and the service level.

Our proposed security architecture, as shown in Figure 1, is based on three main components: the Securityware, the Security Agent and the Security Data store.

The Security Service Provider (Securityware) is responsible of security management and control. Thus, it gathers all the security components as following:

- *Identification Service*: it enables the recognition of a user by the system. It provides an identifier for each user to be recognized by the system. According to his roles, preferences, or service providers, user can have one or more identifier types;
- *Authentication Service*: it determines if an identity is actually what it claims to be. It aims to authenticate user only one time per session (unique authentication) for all requested services;
- *Authorization Service*: it occurs when a user requests access to a service to allow (or deny) him to use service component. Indeed, authorization service evaluates the effective rights. Permission is granted according to rights associated to the user role;
- *Token Service*: it generates and updates the token that permits a unique authentication and ensures consequently a continuous session. The token is regenerated if the session lasts a long time to prevent from session hijacking attempts.
- *Session Service*: it ensures the session creation and activation and the session identifier (Session_ID) generation (upon successful authentication). This service is responsible for managing and maintaining an end-to-end secure session;
- *VPDN and VPSN Services*: the VPDN (Virtual Private Device Network) service manages the set of user terminals while considering each terminal as a service component. The VPSN (Virtual Private Service Network) service manages the network of services which interact together to offer the required global service. These services generate, respectively, the VPDN and VPSN identifiers. These identifiers are kept unique during the session, despite token regeneration, to maintain a unique session.

 ISBN: 978-1-61208-209-7

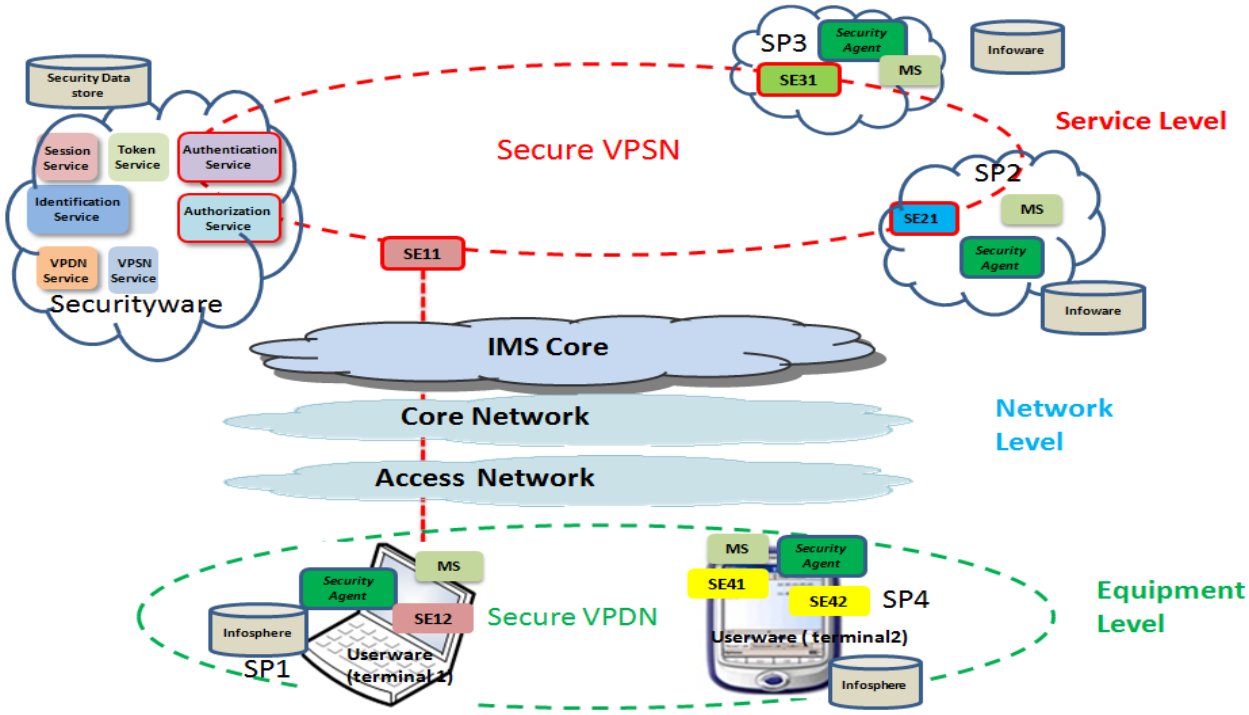

Figure 1. Security Architecture.

In order to facilitate and simplify the management of security and access control at the service and terminal levels, we propose also a Security Agent that is deployed in each service platform and terminal which is considered simultaneously as a service platform and as a service. The Security Agent intercepts requests for the protected resource (service or terminal). It sends then a request to the Securityware asking for decisions regarding the required services. So, the Security Agent secures access to service platforms by checking authorizations due to the Securityware.

Finally, the Security Datastore contains all the information needed by the Securityware, namely, user profile, device profile, VPDN profile, VPSN profile, session profile and service profile.

Our security architectural model will permit a secure vertical aggregation between terminal or equipment components and applicative-service components. Actually, the vertical view represents the user's session that must respect the User-Centric approach. The main objective of this approach is to make the whole system at the service of user unlike other approaches where user must comply with various constraints related to connection (Network Centric) or to treatment (Application Centric). So, the session must be dynamically established according to the user preferences and to services that can be offered by its environment during his movement. We should secure this session without complicating the task to the user each time he wants to add a service to his session.

In fact, our security provider regroups a collection of security components such as identification, authentication and authorization. It represents the Securityware that permits to manage the security of a set of service platforms including the terminal. We deploy also Security Agents that ensure a vertical aggregation of service and terminal levels involved in the session.

On the other hand, our security architectural model will also provide a secure horizontal composition between service components at the service level as the terminal level. Indeed, in a horizontal view, the Securityware and the Security Agent intervene during the service composition to build the VPSN from terminal or applicative services while ensuring a secure access to these services.

C. *A Secure Service composition*

In order to meet the user-centric needs, to ensure all kinds of service personalization and to follow the user's mobility, we must conceive services and manage and secure session otherwise. Thus, our proposed architecture offers a secured service composition to end-users, with dynamic changes following mobility, QoS and security requirements and depending on available service offers.

A secure service composition in service layer represents a VPSN. It includes the set of service components that are used by a user along his session which is based on a user-centric approach. So, services are composed dynamically according to user preferences. VPSN manages interaction between these components via links while respecting a logic of service. Our security services are among management services in the VPSN that are transparent to the user and aim to secure his session.

During a User Centric session, the user is asked to dynamically compose services he needs for a period of time. For example, he can add a videophone service to its voice

call or transfer a TV program from his TV to his PDA when he moves. All these changes are ensured in a seamless way within a unique and secure session. Security is provided by a set of service components that we find in a specialized platform (service provider), called Securityware. We consider in our solution two key points: first, the Userware (terminal) is considered as any service platform (service provider); second, in each service platform, a Security Agent is deployed to manage its security. We describe below how to have a secure service composition within a Virtual Private Service Network (Figure 2).

1) Session Initialization:

When establishing a user session via his terminal, the Userware (Security Agent) sends a request to the Securityware to activate the identification service and then the authentication service in order to validate the user's credentials saved in his profile (Figure 2: (1), (2) and (3)). Once the user is successfully identified and authenticated, session initialization is performed by the Session Service. Then, a token is generated by the Token Service. It contains the necessary session and user information. Afterward, the token is transmitted by the Securityware to the Security Agent in the terminal.

2) VPDN Creation:

Next, we have to verify that the user is authorized to use this terminal and to create a Virtual Private Device Network VPDN [8]. So, an authorization request regarding the use of terminal is sent by the Security Agent to the Securityware. The Authorization Service evaluates the effective rights considering information provided by the Authentication Service. Once the user is authorized, the VPDN Service generates the VPDN-ID that is the identifier of the set of terminals deployed by the user during his session forming the VPDN. The Token Service retrieves the VPDN identifier and updates the token. The terminal is registered in the VPDN profile (Figure 2: (4) and (5)).

3) VPSN Creation:

Now, the user wants to create his VPSN via his terminal. For that the Security Agent checks user's rights (user's authorizations) with the Securityware asking for decision regarding the VPSN creation. When the Securityware grants a positive response, the VPSN Service generates a VPSN-ID and the Token Service updates the token. This latter contains then all the attributes related to the VPSN, the VPDN, the session and the user. Consequently, the service session is initiated by the terminal, which is considered as a service platform, after the authorization and the VPSN-ID generation (Figure 2: (6) and (7)).

Thereafter, the user is able to compose exposable services that he wishes to have in his session according to his preferences and needs. He can select these services from a service catalog stored in his profile or after a discovery of services that are available in his environment. Then, the terminal generates a service composition workflow and a logic of service based on the selected exposable services. Next, this information are sent via the signaling protocol SIP+ (Session Initiation Protocol) [9] in a SIP+ Invite message passing through IMS (IP Multimedia Subsystem) to the service platforms called Serviceware. The SIP+ Invite message body contains also the token that is transferred to each security agent that protects a service platform visited by the user in order to ensure a single authentication during user session (Figure 2: (8) and (11)).

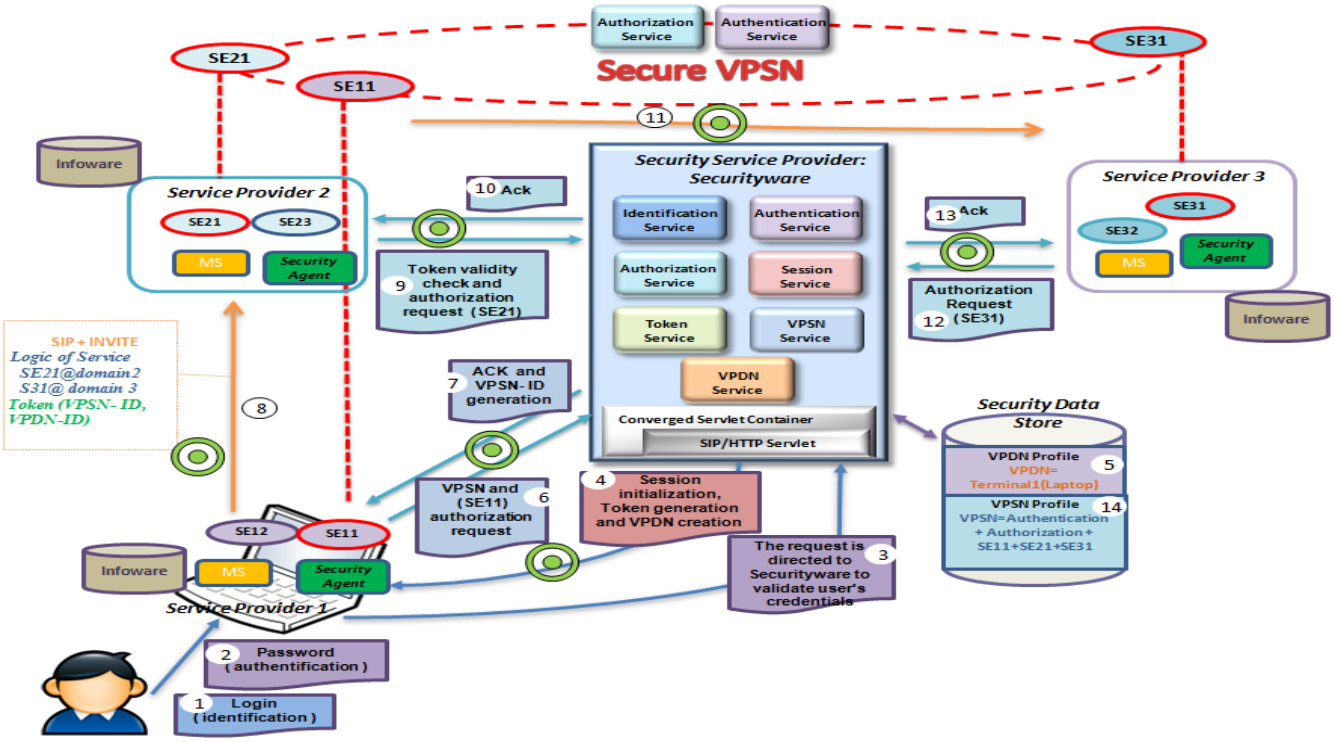

Figure 2. A Secure Virtual Private Network (VPSN).

At the reception of this message, the Serviceware translates exposable services into basic services and builds the VPSN. The Authorization service is included in this VPSN as a management service to perform access authorization to basic services. It interacts with the security agents of the different requested service platforms to verify if privileges related to the desired services correspond to user rights related to his role. In each passage from one platform to another, the security agent checks the validity of the token (Figure 2: (9), (10), (12) and (13)).

The Authentication service is also included in the VPSN in case of a higher authentication is required. For example, when one of selected services in the VPSN is a banking service (payment), a strong authentication is required.

The authentication and authorization components are basic management services which participate in pre-provisioning and provisioning of the services that compose a VPSN. These services perform their roles transparently, without user intervention. Finally, the created VPSN is saved in the VPSN profile (Figure 2: (14)).

IV. FEASABILITY

A. Scenario

For more clarity, we present in this section an application scenario that explains how our security service architecture guarantees security, seamlessness, dynamicity and uniqueness of the end-user session. To achieve automated and distributed control and management of security, we will apply our proposed vision of security as a service called Securityware based on generic, mutualizable, stateless and self-managed service components.

This scenario, as shown in Figure 3, takes place in a cross-organizational context that has multiple service platforms (terminal is also considered as a service platform). Therefore, the creation of VPSN is shared between the different involved platforms.

Considering the fact that our mobile and heterogeneous environment exposes significant risks in terms of security, it becomes increasingly vulnerable. For this reason, our scenario brings out the case when a service becomes unavailable following a malicious attack or a security fail (denial of service) in a user session.

Then, let us describe this scenario: Bob is at home. He uses his laptop. According to his needs and his preferences specified in his contract, he wants to display a movie with high resolution (SE11: Service Element 11), check his emails (SE21) and receive all SMS messages as Voice Messages (SE31).

1) A secure service composition:

As described in subsection 3-C, a secure service composition is performed and a VPSN is provisioned for this user. It contains all services that respond to Bob's request according to his preferences and the required level of security. We note that SE11 is provided by the service provider SP1 (Userware: terminal), SE21 is provided by the service provider SP2, SE31 is provided by the service provider SP3, and the security services are provided by the Securityware.

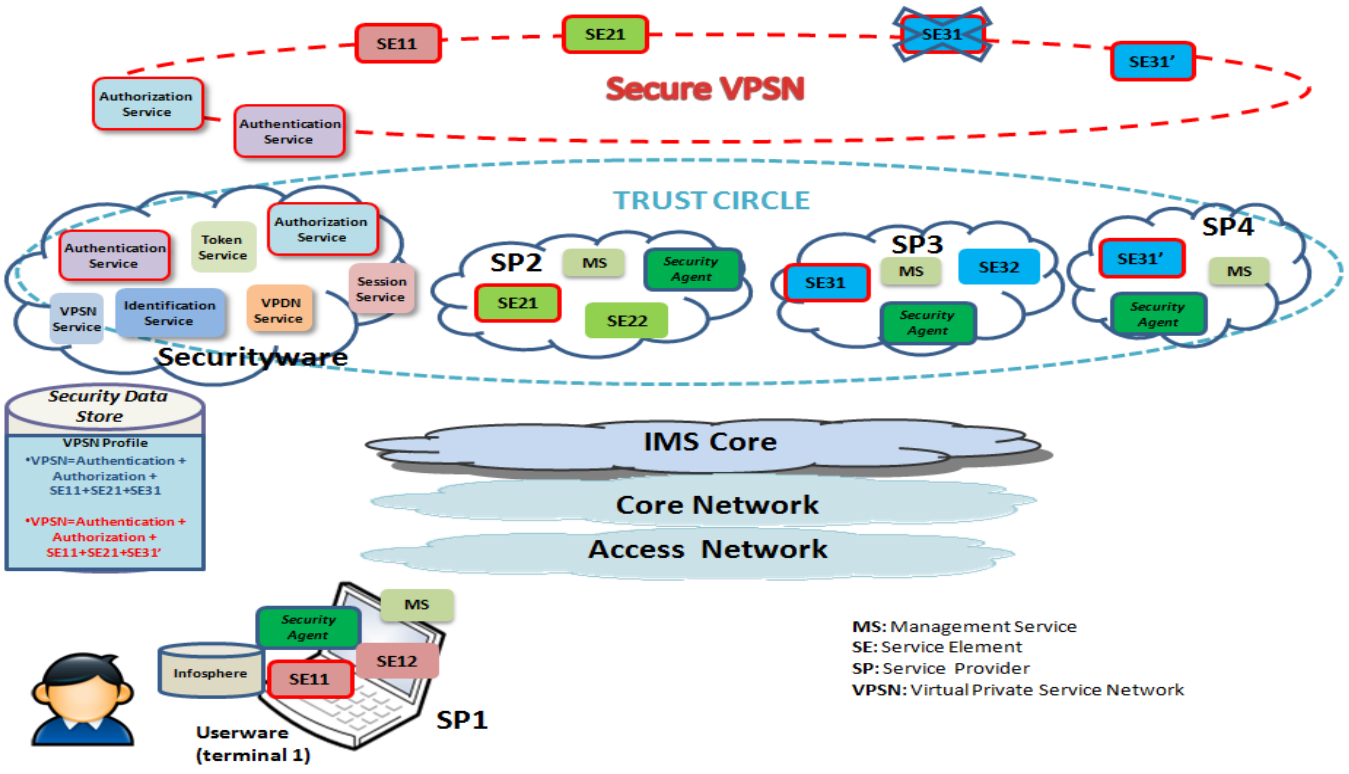

Figure 3. Scenario.

These five services are linked in the VPSN and executed according to a logic of service that presents a workflow of the chosen services and indicates in which order they should be executed.

VPSN (Bob) = Authentication Service + Authorization Service + SE11 + SE21 + SE31

The security services are imperatively interconnected and interoperable with other services in order to provide a secure global service. These security services interact also with Security Agents in each passage from one service platform to another to ensure a secure access to these service platforms using the token.

2) A continuous secure session:

Every change in the user's context as well as the changes related to a service component (availability, malicious attack, security fail) is recovered by adding, removing or replacing one or more component to the virtual service network (VPSN). To have an automated and decentralized recovery and service management, each service component is self-managed. This solution permits to detect the failure at the right time and to provision service resources without causing interruption during service use due to a periodic service discovery that maintains a set of ubiquitous services that are potentially equivalent to services involved in the active VPSN. This discovery is launched in all the federated service platforms which are in the same circle of trust. In fact, the search is based on the three following criteria:

i) Services which belong to the circle of trust,

ii) Services which have the needed functionality, and

iii) Services to which user have access authorization based on his role and privileges.

Thereby, a secure VPSN and a continuous end-user session are maintained.

For example, we suppose that SE31 undergoes a security fail (malicious attack). Thus, a new composition of services is needed to recover this fail. The unavailable component, which has been attacked, should be replaced by another that is functionally equivalent and able to provide the required security level. Consequently, the service SE31 is replaced by SE31', which has been already discovered. It belongs to the trusted service provider SP4.

As a result, a new service composition is established and the logic of service becomes as follows:

VPSN (Bob) = Authentication Service + Authorization Service + SE11 + SE21 + SE31'

3) Mutualizable, generic and stateless security services:

It can be proven that the same service component can be mutualized among different users and can be used in different contexts through the genericity of the service component. For example, the same authentication service resource can be shared between two users in two different contexts: the first user needs a strong authentication for banking transactions and the second user needs a simple authentication to watch a movie. This is possible if service components, particularly security services, are stateless. This means that service components should not require a state or information relative to clients when they are invoked. This feature leads us to have ubiquitous services that can be deployed in any environment. It also makes possible the use of different services provided by different service providers. Therefore, we can switch from a service to another even if they do not belong to the same SP or are not deployed on the same platform which makes the service composition more efficient and flexible.

B. Implementations

In order to prove the feasibility and to validate our proposition, we use our UBIS ("User centric": uBiquity and Integration of Services) project platform that contains principally Open IMS Cores and SailFin application server. The former ensures IMS session control and the latter represents control and applicative service container. We explain below how we implement and deploy our different security service components.

We use Java language to write the proposed security architecture components, namely, Securityware and Security Agent. For our Securityware, we use EJB (Enterprise JavaBeans) technology [10] to develop autonomous, loosely connected and stateless services. These services are deployed in SailFin [11] Application Server that supports various APIs (Application Programming Interfaces) such as JMS (Java Message Service), JNDI (Java Naming and Directory Interface), JDBC (Java DataBase Connectivity) and Sip servlet. Our proposed Securityware extends actually some security parts involved in OpenSSO project [12]. For our Security Agents, they are deployed in each terminal or service platform. In order to support SIP+, we deploy then Converged Application Container which is composed of SIP and HTTP (Hypertext Transfer Protocol) servlets and will permits to switch from HTTP to SIP. We note that all transactions between different components are secured using SSL (Secure Socket Layer) protocol. Our Security Datastore is an LDAP (Lightweight Directory Access Protocol) directory (OpenDS) that contains our used information and is connected to the Securityware.

Figure 4 describes how our security service provider Securityware is implemented.

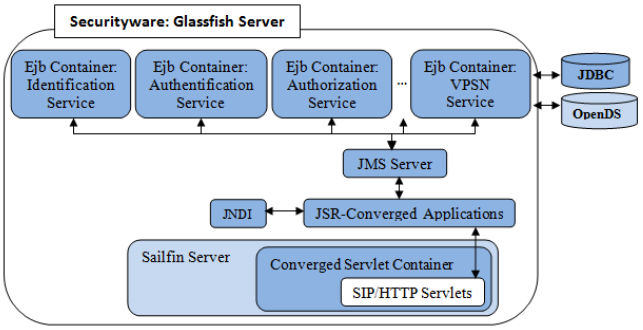

Figure 4. Securityware.

V. CONCLUSION

In this paper, we have presented our service-oriented security architecture solution, which is promising in dynamic, mobile and cross-organizational environment. We demonstrated how security can be provided as a service ensured by a set of security components offering the user a secured and a seamless access to his services. The security services are specified respecting a set of criteria, namely mutualization, autonomy, interoperability and self-management. These services intervene transparently to offer a secure and dynamic service composition within a unique and continuous session. To support our proposition, we have described an application scenario which explains the major contributions of our security solution, and we have proven its feasibility in practices. Future efforts will go into the design and implementation of federation audit and trust services.

ACKNOWLEDGMENT

The authors would like to thank all the participants in the UBIS project, financed by the French ANR VERSO 2008 in which is situated this work.

REFERENCES

[1] P. Qi-rui, W. Cheng, W. Jing, L. Jun, L. Qing, and S. Beien. An authentication and authorization solution supporting soa-based distributed systems. In Software Engineering and Service Sciences (ICSESS), 2010 IEEE International Conference on, pages 535–538. IEEE, 2010.

[2] E. Bertino and L.D. Martino. A service-oriented approach to security–concepts and issues. In Future Trends of Distributed Computing Systems, 2007. FTDCS'07. 11th IEEE International Workshop on, pages 31–40. IEEE, 2007.

[3] C. Emig, F. Brandt, S. Kreuzer, and S. Abeck. Identity as a service-towards a service-oriented identity management architecture. In Proceedings of the 13th open European summer school and IFIP TC6. 6 conference on Dependable and adaptable networks and services, pages 1–8. SpringerVerlag, 2007.

[4] C. Emig, F. Brandt, S. Abeck, J. Biermann, and H. Klarl. An access control metamodel for web service-oriented architecture. In Software Engineering Advances, 2007. ICSEA 2007. International Conference on, pages 57–57. IEEE, 2007.

[5] R. Kanneganti and P. Chodavarapu. Soa security. 2008.

[6] Z.B. Daho and N. Simoni. Towards dynamic virtual private service networks: Design and self-management. In Network Operations and Management Symposium, 2006. NOMS 2006. 10th IEEE/IFIP, pages 1–4. IEEE, 2006.

[7] C. Phan. Service oriented architecture (soa)-security challenges and mitigation strategies. In Military Communications Conference, 2007. MILCOM 2007. IEEE, pages 1–7. IEEE, 2007.

[8] A. Hammami and N. Simoni. Secure and seamless session management in mobile and heteregenous environment. In SECRYPT, ROME, 2012.

[9] H. Alaoui, P. Coude, and N. Simoni. User-centric and QoS-based Service Session. In ASPCC, Korea, 2011.

[10] http://www.oracle.com/technetwork/java/javaee/ejb/index.html

[11] Sailfin Project. https://sailfin.dev.java.net/.

[12] OpenSSO Project. https://opensso.dev.java.net/.

 ISBN: 978-1-61208-209-7