Meeting EHR Security Requirements: Authentication as a Security Service

Basel Katt[1] Thomas Trojer[1] Ruth Breu[1] Thomas Schabetsberger[2] Florian Wozak[2]

[1] Research Group Quality Engineering, University of Innsbruck, Austria.
[2] ITH-Icoserve, Innsbruck, Austria.

Abstract: Electronic Health Record (EHR) is a promising concept to collect and manage electronic health information of all citizens. Integration the Heathcare Enterprise (IHE) was one of the first initiatives that aims at standardizing the way healthcare systems exchanging information in a distributed environment. Based on EHR concepts and IHE profiles different approaches have been introduced in the industry and the literature to implement and apply solutions for different stakeholders in the healthcare domain (see e.g., `http://www.ith-icoserve.com/`). Due to the sensitivity of the data dealt with in these systems, security is a major concern that must be considered. In previous work we have presented a general architectural solution to apply the evolving *Security as a Service* (*SeAAS*) paradigm in distributed architectures for EHR in conformance to IHE–proposed profiles. While our architecture proposed is generic and covers all security requirements, we focus in this work on one security requirement, namely, authentication and show how it can be offered as a service while adhering to IHE profiles. [1]

1 Introduction

Information and communication technologies have been involved in most sectors of our lives, and healthcare is not an exception. Electronic Health Record (EHR) systems have been proposed and researched recently aiming at decreasing healthcare costs, increasing healthcare quality and reducing medical errors. IHE was one of the first initiatives started in 1998 with a main goal of building a framework that seamlessly enables the exchange of health information across multiple healthcare institutions and enterprises. While IHE does not create new standards, it proposes profiles that specify precisely how current standards can be used to reach it goals. Due to the sensitivity of the information that healthcare system are dealing with, security is one of the major concerns that must be tackled. Despite of the fact that IHE has recognized the importance of security by introducing few profiles that tackle different security requirements, however they are oversimplified, vague and do not consider architectural design [KTB+10].

IHE IT infrastructure profiles use the Service Oriented Architecture (SOA) paradigm in

[1] This work is paritally supported by the Austrian Federal Ministry of Economy as part of the Laura-Bassi —Living Security Models —project FFG 822740

its design, thus, IHE based systems can be featured as a highly heterogeneous and distributed. The current main practice to offer security functionalities in such highly dynamic and distributed environments is based on *end point security* concept. End point security is based on putting security functionality exclusively at end points, which means that each actor —functional component of the healthcare enterprise —in any domain must implement, maintain, and manage its own security related functions. Recent study [HMB09] shows that this methodology is inadequate and ineffecient in distributed and heterogenous systems. The proposed alternative to end point security is the *Security As A Service (SeAAS)* paradigm. SeAAS aims at extracting all security functionalities and mechanisms from end points in one domain and offer these functions as a central services for the whole end points in that domain. In [KTB+10] we proposed a general architectural solution to apply SeAAS concepts in IHE based healthcare systems that are based in theirdesign on the Cross-Document Sharing (XDS) profile [tHEI09a, tHEI09b]. We proposed to offer secuirty functionalities as services for each XDS affinity domain —a group of healthcare enterprises that have agreed to work together using a common set of policies and share a common infrastructure —without discussing the details of each security service. In this paper we move a head and discuss how authentication can be offered to an affinity domain based on the general architecture we proposed previously.

The rest of this paper is organized as follows. In Section 2 we discuss the concept of *Security as a Service*. In Section 3 we focus our study on the authentication service and present how brokered authentication can be offered as a service. Finally, we conclude and discuss future work in Section 4.

2 Security as a Service for Distributed EHR Systems

SeAAS, unlike end point security, provides security functionality centrally for endpoints within a common domain. In most scenarios nowadays endpoint security is applied. Setting all security mechanisms at the endpoint in such distributed and heterogeneous environment (like IHE-based infrastructures) increases dramatically the processing overhead applied on each endpoint. This yields the management and maintenance of these decentralized security mechanisms an exhausting tasks, and poses interoperability challenges [MHB09]. The benefits of our SeAAS (please refer to [KTB+10] for more details on the genera architecture) solution can be gained in the following issues:

- Performance: In critical systems like healthcare systems that deals with people's lives, performace is one of the key factors that should be considered by architects and designer. Security services involve performance costly functions that affects the perfromance of the whole system. An emperical study conducted in the context of Sectissimo project (http://www.sectisimmo.info) showed that SeAAS prototype performed better than end point security and was at least 1.2 time faster (More details can can be found in a paper to appear soon in the context mentioned project). Thus, the first advantage of SeAAS is performance.
- Maintainance and policy management: The maintainance and the managment of

security solutions for a dynamic distributed and heterogenous systems is comlex task. With security functions done at end points, security mechanisms are spread over the system infrastructure and involve all functional services. Thus, in order to keep the solutions updated with (i) the new functions or policy changes required due to the changing of security requirements, or (ii) upgrades of current solutions to cope with new security risks and threats, changes must be propogated to each end point. With a larg number of services and end points, maintenance and management tasks will be very inefficient and complex. Central solution for security services that provide central secuirty services eases the updating tasks as they are done once and do not require any propogation.

- Configurability: Our solution allows for two main types of configuration at two layers. First, at the upper layer, we have the composition policy that indicate which sercurity services to invoke and in which order. Second, the configuration of each security service in order to offer more than one security pattern, more about security patterns can be found in [DFLPW07, ESP07, RGFMP06]. Security patterns provide different solution for each security service based on different requirements. For example, authentication can be either direct authentication, borkered authentication, distributed authentication (fedirated identity), or centralized authentication [Erl09].

The concept of SeAAS bases on two modules, namely the *SeAAS engine* and *security services*. The SeAAS engine is responsible of orchestrating security functionality according to requests of secured endpoints. Deciding what are the needed secuirty requirements, i.e., secuirty services that must be invoked, and in which order these services must be invoked is done using a declarative policies called *Composition Policies*. Furthermore, security services can be classified into two types. First, *primitive security services* implement basic security or security related functionality, like (de-)encryption, signature, and time stamping. Second, *composed security services* utilize multiple primitive security services according to a general security requirement to be fulfilled. Based on this concept we introduced in [KTB+10] a SeAAS architecture to an IHE-based healthcare system. Security services to realize e.g., *authorization*, *non-repudiation*, *monitoring* and *authentication* are briefly mentioned.

Based on IHE XDS profile each affinity domain contains of four main actors: one document registry, one or more document repositories, one patient ID service,a gateway, a document consumer, and a document source, more details about XDS profile can be found in [tHEI09a]. Assuming an affinity domain with three document respositories, Figure 1 shows how this domain can be extended with the SeAAS components. Upon receiving a request from another domain by the gateway ①, the gateway forwards this request to the SeAAS engine ②. Based on the composition policy that corresponds the received request ③, the SeAAS engine invokes the required composed security services ④ in the order mentioned in the policy (policy can be defined as WS-BPEL [OAS], or WS-Policy). While composed security services are executed, primitive security services can be invoked ⑤. Finally, after all required security services are executed and the security requirement are fulfilled, the SeAAS engine returns the final decision to the gateway ②. Upon a positive decision the gateway forwards the functional request to the corresponding actor ⑥.

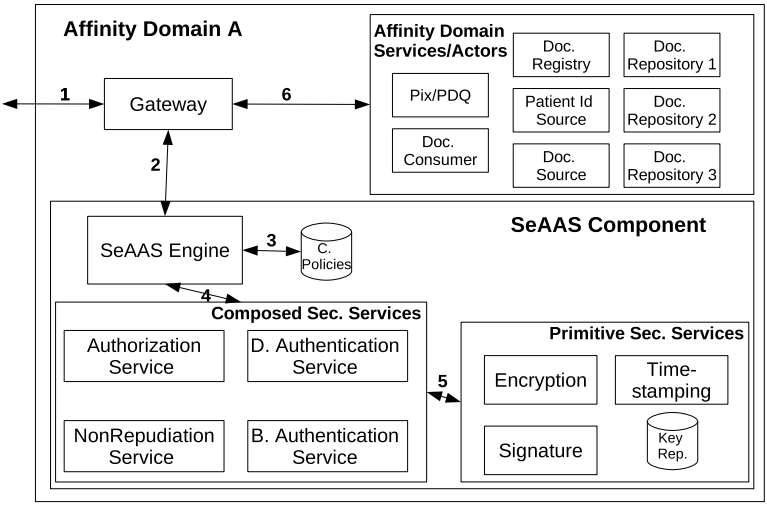

Figure 1: SeAAS architecture in an affinity domain.

It can be noticed that for any composed security service, the SeAAS component can offer more than one security pattern, (mechanism or solution). For example, Figure 1 shows that two authentication can be offered either as *Direct (D.)* or *Brokered (B.)* authentication. We show these two service for authentication protocol in separate components, however, in reality we will have one service that can be configured to act as direct of brokered authentication service.

3 Authentication Security Service

Authentication service is the service that aims at the verification of identiy. Authentication can be considered from two perspectives, node and user authentication, the first authenticate the node, using Transport Layer Secuirty (TLS) for example, and the latter verifies the authentication of users.

Transaction ITI-19 [tHEI09a] suggests the mutual authentication of nodes. Authentication of nodes is useful to provide trusted channels for specific transactions using functionality or data provided by multiple nodes. Nodes are therefore provided with trusted certificates and validation of those is covered by credentials validation services. On the other hand, transactions ITI-2,ITI-2, and ITI-4 [tHEI09a] suggest user authentication based on a challenge and response mechanism to verify the identity of an individual communication with the enterprise. *Kerberos* protocol was suggested to be used [NT94]. Kerberos user/password authentication is available for users within a protected domain, beside more sophisticated means of identity provisioning like smart cards or biometrics available to protected domains and external ones.

In this work we focus only on user authentication to be offered as a service. Two main drawbacks can be identifed in the IHE profiles related to authentication (cf. Section 2). First, it only proposes one authentication pattern and technology. Different health care institution apply different authentication services and protocols, based on different authentication patterns. For example brokered authentication with kerberos, X509 PKI, or STS (Security Token Service) options, or distributed authentication. Proposing only one solution oversimplifies the problem and decreases the viability of this service. Second, each IHE actor must implement and take care of the authentication mechanism by its own, which dereases the overall performance of the system. Applying SeAAS allows (i) offering multiple authentication patterns for the authentication service due to the configurability feature, and (ii) remove the security functionality form the end points and apply then in a dedicated services for authentication mechanism, thus enhancing the performance.

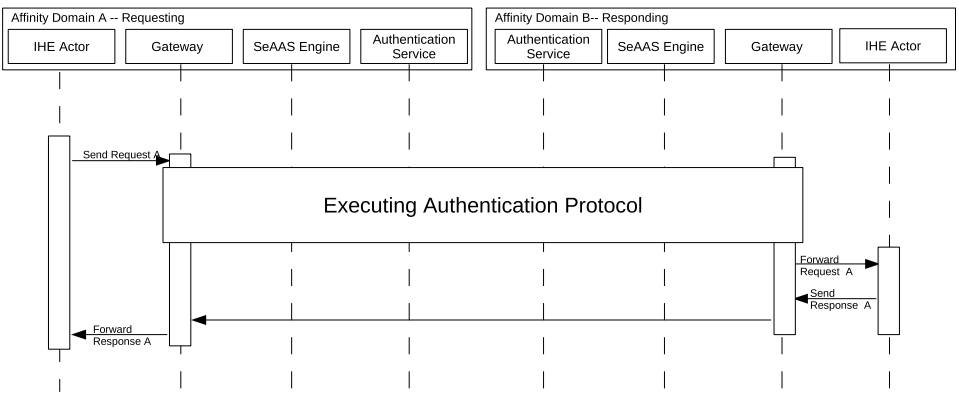

Figure 2: Sequence diagram shows how authentication functionality is moved from the IHE actors' endpoints and executed by the SeAAS components.

Figure 2 shows a how the execution protocol is moved from the IHE actors (document repository, registry etc.) to the SeAAS engine and authentication service. After validating the identity of the user the result is sent to the gateway. If the user is authenticated, the gateway forward the rquest further to the ITH actor, other wise send back an error message to the requesting gateway. The figure does not show the details of the authentication service, which might support different authentication patterns. The selection of the suitable one to execute is done by the SeAAS engine based on the *composition policy*. In the following we discuss the authentication service using a general borker authentication pattern. Please note that authroization that must be checked after authentication is out of scope of this work and is not shows in Figure 2. Furthermore, we assume that the response that is sent back to the requesting domain does not need any authentication check, which is the normal situation. That is why the response message is sent directly from the gateway of the responding domain to the gateway of the requesting domain.

3.1 Brokered Authentication

Brokered authentication pattern is used when the both the service consumer and the service provider do not trust each other and the consumer require an access to multiple services. An authentication broker in this case is responsible for authenticating the consumer and issuing a secuirty token to the consumer. This security token is used by the consumer to the access the service.

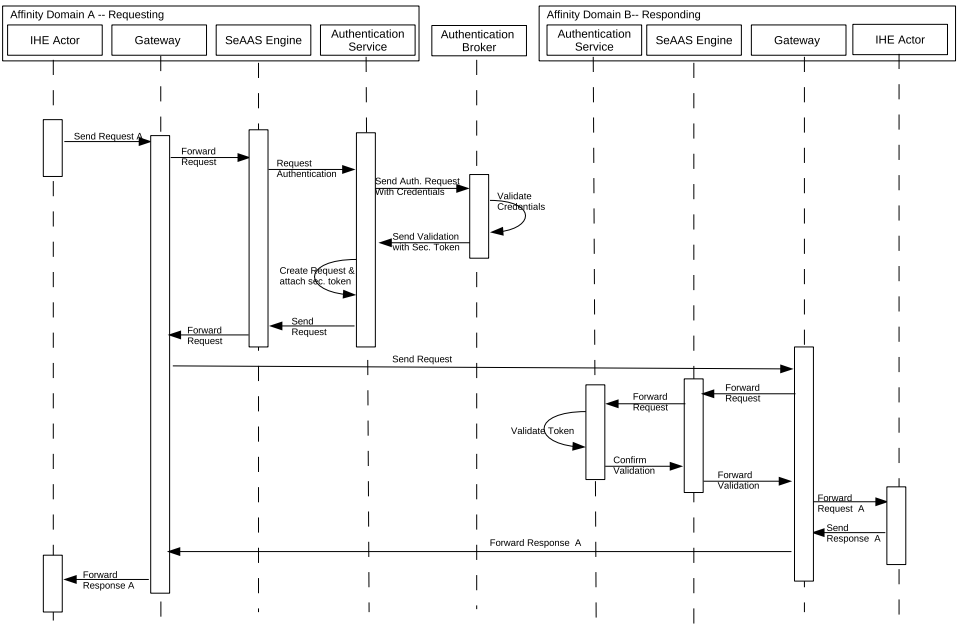

Figure 3: Brokered authentication service.

Figure 3 shows the execution sequence when brokered authentication is used to authenticate a user in an affinity domain A that is trying to get an access to a service in an affinity domain B. After the gateway of the requesting domain receives the request from the IHE actor, it forword the request to the SeAAS engine, which in turn forward the request to the required security services, authentication in our case. Authentication service send an authentication request to the authentication broker with the credentials of the requesting user. After the authentication broker validate the credentials it sends the validation with a security token to the authentication service. In the case of Kerberos, this will be the service ticket. Using the received token and the original request, the authentication service creates a new request with the security token attached to it. This request is forwarded to the SeAAS engine, which will send it forward to the requesting gateway. At this stage the request is created with the required security attachment and ready to be sent to the domain B. The requesting gateway send this request to the responding gateway, which in turn forwards the request to the SeAAS engine at domain B. After checking the required security service that need to be invoked, in our case only the authentication service, it forwards this

request to the authentication service. The authentication service validates the token that is sent with the request and confirms the validation to the SeAAS engins, which in turn forward the validation to the gateway in domain B. Upon a positibe validation, the gateway at the responding domain forwards the request to the IHE actor. The actor proccess the request and send back a response to the requesting IHE actor through the gateways in both domains.

4 Conclusion and Future Work

In this work we present an architectural solution for applying the evolving *SeAAS* paradigm to secure healthcare systems focusing on one security measure, namely, user authentication. SeAAS methodology overcomes the shortcomings of the current widely adapted *endpoint security* solutions with respect to management, maintainability and performance. In the future we plan to tackle other security requirements and develop a proof of concept prototype of the SeAAS framework.

References

[DFLPW07] N. Delessy, E.B. Fernandez, M.M. Larrondo-Petrie, and J. Wu. Patterns for access control in distributed systems. In *Proceedings of the 14th Conference on Pattern Languages of Programs*, pages 1–11. ACM, 2007.

[Erl09] T. Erl. *SOA Design Patterns*. Prentice Hall PTR, Upper Saddle River, NJ, USA, 2009.

[ESP07] R. Erber, C. Schlager, and G. Pernul. Patterns for Authentication and Authorisation Infrastructures. 2007.

[HMB09] M. Hafner, M. Memon, and R. Breu. SeAAS-A Reference Architecture for Security Services in SOA. *Journal of Universal Computer Science*, 15(15):2916–2936, 2009.

[KTB+10] B. Katt, T. Trojer, R. Breu, T. Schabetsberger, and F. Wozak. Meeting EHR Security Requirements: SeAAS Approach. In *EFMI STC 2010. Accepted*, June 2010.

[MHB09] M. Memon, M. Hafner, and R. Breu. Security As A Service: A Reference Architecture for SOA. In *7th International Workshop on Security in Information Systems (WOSIS 2009)*, Milan, Italy, May 2009. Springer, Springer.

[NT94] B.C. Neuman and T. Ts'o. Kerberos: an authentication service for computer networks. *Communications Magazine, IEEE*, 32(9):33–38, September 1994.

[OAS] OASIS. Web Services Business Process Execution Language (WSBPEL) TC. http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wsbpel.

[RGFMP06] D.G. Rosado, C. Gutierrez, E. Fernandez-Medina, and M. Piattini. Security patterns and requirements for internet-based applications. *Internet Research*, 16(5):519–536, 2006.

[tHEI09a] Integrating the Healthcare Enterprise (IHE). *IT Infrastructure (ITI) Technical Framework, Volume 1, Integration Profiles*. IHE, August 2009.

[tHEI09b] Integrating the Healthcare Enterprise (IHE). *IT Infrastructure (ITI) Technical Framework, Volume 2a, Transactions ITI–1 through ITI–28*. IHE, August 2009.