SPoC: Protecting Patient Privacy for e-Health Services in the Cloud

Lu Fan, William Buchanan, Owen Lo, Christoph Thümmler
Alistair Lawson, Omair Uthmani, Elias Ekonomou, Abou Sofyane Khedim
Edinburgh Napier University, Edinburgh, UK
{L.Fan; B.Buchanan; O.Lo}@napier.ac.uk

Tabassum Sharif
Flexiant Ltd.
Livingston, UK
TSharif@flexiant.com

***Abstract*—The use of digital technologies in providing health care services is in general subsumed under the term e-Health. The Data Capture and Auto Identification Reference (DACAR) project provides an open e-Health service platform that reinforces the integrity, security, confidentiality and auditability of medical data throughout their life-cycle. This paper presents the design and implementation of the core component of this platform, namely the Single Point of Contact (SPoC). A SPoC is essentially a security authority that provides claim-based authentication and authorisation functionalities, and facilitates the development and integration of secure e-Health services hosted within a Cloud Computing environment.**

***Index Terms*—Single Point of Contact, e-Health, Privacy, Security, Cloud Computing**

I. Introduction

The use of modern communication infrastructures in medicine, and the ubiquitous provision of health care services, are collectively known as e-Health [1]. Currently, many countries are keen to shift their traditional health care services to this new paradigm, in order to improve the quality of care and reduce the health care delivery cost [2], [3], [4], [5].

The Cloud Computing technology [6] appears well-suited to meet such demands, as it is able to reduce the capital and operational expenditures for the development and provision of e-Health applications. A Cloud adopts a Service Oriented Architecture (SOA) [7] and supports the functionalities of an integrated e-Health system as a number of coarse-grained and inter-operable software services. These services may exchange and share medical data with each other in order to improve the overall quality of care offered to the patients.

However, confidential health care information is often subject to a variety of risks, and an inconsistency and loss of such information can result in severe consequences [8]. Hence, a *patient-centric* e-Health system must provide the patients with control over the utilisation and dissemination of their own private information [9]. Unfortunately, traditional security mechanisms are insufficient to meet the requirements of patient-centric e-Health services in an open, dynamic Cloud Computing environment, mainly due to:

- **Platform-dependent** – Traditional security mechanisms often rely on specific operating systems or protocols, and thus it is difficult for them to interact and co-operate.
- **Isolated** – It is difficult for service providers to federate across security domains. As a result, users need to manage multiple identifiers for multiple service providers.
- **Cumbersome** – Traditional security mechanisms often rely on firewalls and expect users to access protected network resources over a VPN connection.
- **Inflexible** – A security authority only delegates access rights according to a user's identity and group, without considering other attributes of the user.

The aim of the Data Capture and Auto Identification Reference (DACAR) project [10] is to develop, implement, validate and disseminate a novel, secure, Cloud-based e-Health service platform that reinforces the integrity, security, confidentiality and auditability of sensitive medical data throughout their life-cycle. Our previous work has provided an overview of the DACAR platform [11], and this paper further elaborates on the design and implementation of its core component, namely the ***Single Point of Contact*** (**SPoC**). A SPoC is essentially a security authority, which protects patients' privacy in e-Health applications by providing a claim-based authentication and authorisation functionality [16], and facilitating secure communication between an e-Health service and its clients.

The remainder of this paper is organised as follows. Section II presents the background of this research. Section III discusses the design of the SPoC, including its internal architecture (Section III-A), supports for authentication (Section III-B), authorisation (Section III-C), secure Web Services (Section III-D), and three representative application scenarios (Section III-E). Section IV outlines the implementations of the SPoC and proof-of-concept applications. Finally, Section V draws the conclusions and sketches the future work.

II. Related Work

Benzschawel et. al. pointed out that the main expectations of e-Health are to provide better ways to exchange and share medical information and to improve the quality of services offered to patients [9]. A multi-level architecture is proposed to protect patient privacy, which uses: a Central Medical Registry (CMReg) for authentication and authorisation purposes; a Centralised Medical Data Repository for storing anonymised medical documents; and a Document Management System for authorised users to associate medical documents with real patient identities. The design of the SPoC shares many good characteristics with the CMReg, such as the anonymisation of medical data and the use of pseudonyms to enhance contextual privacy. The main difference between the CMReg and the SPoC is that the CMReg only focuses on people's access to confidential medical data, while the SPoC also facilitates the development and integration of secure e-Health services.

Zhang et. al. have identified a set of security requirements for e-Health services hosted by a Cloud computing environment, including authentication, authorisation, ownership of information, and integrity, confidentiality and availability of data [12]. A model is proposed to address the security and privacy issues relating to access to and management of Electronic Health Records (EHRs). The design of the SPoC also takes these requirements into consideration, but in addition it aims to be more generic, and able to support a wider range of application scenarios beyond the sharing of EHRs.

Kilic et. al. have proposed the sharing of EHRs among multiple e-Health communities over a peer-to-peer network [13]. A super-peer is used to represent an e-Health community, which is responsible for routing messages and adapting different meta data vocabularies used by different communities. This super-peer design is similar to the design of a SPoC, yet a SPoC provides additional claim-based authentication and authorisation functionalities. Multiple SPoCs may also keep contact with each other in a peer-to-peer fashion to form a *Circle of Trust*.

Claim-based identity management and access control is proposed to overcome the disadvantages of conventional security mechanisms [14]. It thus abstracts from concrete formats and protocols of identity systems and provides a platform-independent way of presenting identity information [15]. Windows Identity Foundation (WIF) [16] is a typical example of this approach, which consists of the following components:

- **User** – A user is a subject of access control, which can either be a human, or a non-human entity.
- **Claim** – A claim is a statement about a subject made by another subject and can relate to any type of identity attribute. A claim is essentially a cryptographically protected security token, and its format is usually standard, e.g. Security Assertion Markup Language (SAML) [17].
- **Security Token Service** (STS) – A STS is an issuer that accepts requests and creates security tokens containing claims. If a STS is used to verify user credentials and certify user identities, the STS is referred to as an Identity Provider. If a STS is used to certify a user's attributes other than identity, the STS is referred to as an Attribute Provider. If a STS accepts a claim and translates it into another application-specific claim, the STS is referred to as a Resource STS (R-STS).
- **Relying Party** (RP) – A RP is a service provider, or an application, which relies on an issuer to provide information about its users' identities and attributes.
- **Client** – A client is a software agent that implements protocols like WS-Trust [18] and WS-Federation [19] to request and pass around claims on behalf of a user.

The SPoC adopts a claim-based approach for both authentication and authorisation. However, its role may change depending on different situations. Firstly, a SPoC is able to issue security tokens by itself. In this sense, a SPoC is a STS. Secondly, a SPoC is able to authenticate internal users, who have accounts in the SPoC's local domain, as well as to certify the attributes that the user has. Therefore, a SPoC can be both an Identity Provider and an Attribute Provider. Thirdly, a SPoC relies on trustworthy issuers to provide information about external users, who do not have accounts in the SPoC's local domain. In this case, the SPoC becomes a RP. Finally, a SPoC is able to translate a claim from another SPoC in the circle of trust, and thus a SPoC can also be a R-STS.

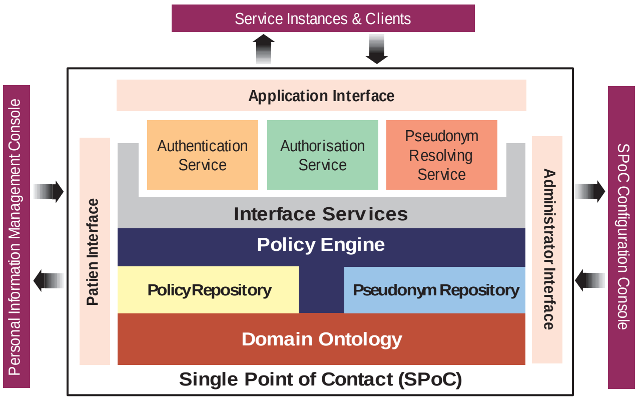

Fig. 1. The internal architecture of a SPoC

Another crucial challenge for a patient-centric e-Health platform is to obtain a variety of patient consents in an electronic way. Coiera et. al. have identified four levels of e-consent, including: *general consent*; *general consent with specific exclusions*; *general denial*; and *general denial with specific consents* [20]. The information sharing policy syntax used by the SPoC is able to express all of the above, as well as *service authorisation*, *service subscription* and *investigation*. Furthermore, Pruski has identified the requirements for an e-consent language to capture *specific grantees*, *operations*, *purposes* and *period of validity*, and proposed a novel language called e-CRL [21]. The SPoC's policy syntax is as expressive as e-CRL, and has been successfully applied to other domains beyond health care, such as police and social care [22], [23].

III. Design

A. Internal Architecture

The internal architecture of a SPoC consists of the following modules, as shown in Figure 1:

1) Domain Ontology – A SPoC maintains a dynamic set of domain ontologies using an internal database. This provides the necessary vocabulary for the SPoC to issue various kinds of claims, and for the users and SPoC administrators to create a range of authentication and authorisation policies. Concepts and their relationships can be established and modified conveniently using the *SPoC Configuration Console*. The most notable concepts in the domain ontology include:

- *Domain*: This refers to a distinct business area that is administered by a single organisation. An e-Health application may involve multiple domains, such as hospitals, pharmacies, insurance companies, and research institutions. Typically, a domain is represented by one SPoC, and the SPoCs for multiple cooperative domains communicate with each other to form a Circle of Trust (CoT). In a CoT, each SPoC keeps a list of services provided by other SPoCs, as well as a table for translating concepts from foreign domain ontologies to native ones.

- *User*: This refers to a consumer of an e-Health application, which can be a person or an impersonated service. A user must be a member of at least one domain, which is able to certify the user's identity and attributes.
- *Object*: This refers to any entity that is managed by an e-Health application, such as patients and medical devices. An object is identified by a unique identifier (UID) assigned by its owner domain. To withstand contextual privacy attacks [24], opaque pseudonyms are often used in place of transparent UIDs [9].
- *Attribute*: This refers to an atomic unit of information that is used to describe an object. The SPoC supports flexible customisation of application-specific attributes, but it is recommended to use standard medical attributes defined by HL7 [25] or CHH [27] whenever possible. The DACAR e-Health platform stores attributes using Data Buckets [11].
- *Service*: A SPoC maintains the identity, public key, communication end-point and dependent attributes of the e-Health services provided by the local domain. A SPoC also maintains a list of services that are provided by other trustworthy domains in a CoT, as discussed above.

2) Policy Engine – On top of the Domain Ontology module is the Policy Engine, which comprises of a *Policy Repository* and a *Pseudonym Repository*.

The DACAR e-Health platform provides a novel and consistent policy syntax for patients to give explicit consent on the utilisation and dissemination of their own medical data. Considering that patients may have limited IT skills, a friendly user interface, i.e. the Personal Information Management Console, is provided for the patients to give their consents using structured natural language. Then, an interpretor converts these consents into formal policy syntax and uploads them to the Policy Repository. Technical details about the design and implementation of DACAR's policy syntax are provided in [11], [22] and [23]. Briefly, the policy syntax can be used for the following:

- ***Service Authorisation Policy***: This allows or denies an individual with certain identity, roles or application-specific attributes to consume an e-Health service.
- ***Service Subscription Policy***: This represents a patient's subscription to an e-Health service, and allows the service to access or modify a set of the patient's medical data, so that the service is able to perform its functionality. This set of medical data is referred to as the *dependent attributes* of that service.
- ***Specific Consent***: This allows or denies an individual with certain identity, roles or attributes, to access or modify a patient's medical data in a fine-grained manner.
- ***General Consent***: Sometimes it could be difficult for a patient to name the grantees of a specific consent, because they are unclear, unknown, or difficult to describe. In this case, a general consent is used to express the patient's willingness to share his/her medical data with e-Health services, from a certain domain, for a certain purpose, or in a certain application context.
- ***Audit***: This obligates information sharing in exceptional situations, such as a medical incident investigation.

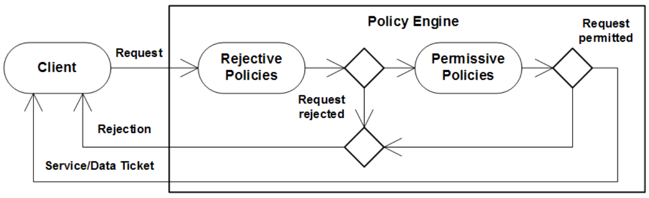

Fig. 2. Decision-making process of the Policy Engine

Figure 2 depicts the decision-making process of a Policy Engine. When it receives a request for an e-Health service or medical data, it firstly checks existing rejective policy rules in the Policy Repository. If an explicit rejection is found, the Policy Engine rejects the request immediately. Otherwise, it continues to check existing permissive policy rules. Unless an explicit permission is found, the request would still be rejected. This decision-making process allows a rejective rule to override a permissive rule, when both conflicting rules coexist in a Policy Repository. Furthermore, because a decision is made when the Policy Engine identifies an explicit rejective or permissive policy rule, the total number of relevant rules to analyse and the way they are ranked may have an impact on the average time that the Policy Engine takes to make a decision. Alternative decision-making strategies and performance improving methods will be investigated in future work.

3) Interface Services – The SPoC offers three interface services to e-Health applications and their clients, including:

- *Authentication Service*: This service verifies users' identities using flexible methods and issues claims about users' identities and attributes.
- *Authorisation Service*: This service accepts requests for e-Health services or medical data, analyses the requests using the Policy Engine and issues security tokens such as *Service Tickets* and *Data Tickets*.
- *Pseudonym Resolving Service*: This service uses the Pseudonym Repository to resolve opaque pseudonyms into transparent UIDs, so that privileged users or applications can associate anonymised medical data with real patient identities.

B. Authentication

An authentication mechanism enables an entity to prove to a remote end its identity using a cryptographic protocol.

It is a fundamental building block for service oriented e-Health systems. A SPoC thus provides flexible methods for authenticating internal and external users.

Internal users refer to the members of a SPoC's local domain. Usually, a security infrastructure is already set up to manage internal user accounts and attributes. In this circumstance, a SPoC can be integrated to this existing infrastructure. For example, a SPoC may authenticate internal users of a Windows domain using Active Directory, and look up their attributes using LDAP. If such a security infrastructure did not exist, a SPoC would manage user accounts and attributes using its Domain Ontology database, and employ *Federated Identity Providers* for authentication purposes. Currently, multiple technologies are available for federated identity management, such as U-Prove [28] and OpenID [29]. The SPoC adopts U-Prove, because it offers stronger cryptographic algorithms, separates the retrieval of identity information from the release of this information to destination sites, prevents the issuing organisations from tracking and linking user actions, and thus protects patients' privacy better.

When an internal user requests an e-Health service that is provided by the same domain, the SPoC supports a single sign-on and does not ask for the user's credentials repeatedly. In the case that the service is provided by a different domain, the SPoC issues a claim about the user's identity and attributes, and forwards it to the SPoC in charge of the foreign domain.

External users refer to people who do not have an account in a SPoC's local domain. In this circumstance, the users should firstly log on to the local SPoC to obtain a claim about their identity and attributes. This claim, together with a service request, is forwarded to the foreign SPoC in charge of the target service. The foreign SPoC translates the external user's attributes into its local domain ontology, and then proceeds to the authorisation process.

C. Authorisation

An authorisation mechanism endows different entities in a system with different access rights to sensitive information and resources. A SPoC uses its Policy Engine to match a request for an e-Health service or medical data to existing security policies, and determines whether the request should be permitted or not. If the request is permitted, the SPoC shall issue a security token, which entitles the requester to consume the service, or to Create, Read, Delete, and Update (CRUD) corresponding medical data as appropriate.

1) Service Authorisation: The Policy Engine requires a service request and a claim regarding the requester in order to make an authorisation for service access. The service request provides information about the target e-Health service, including its qualified name and favourite locations where the requester prefers a service instance to be created in the Cloud. The claim provides information about the requester's identity, role and other application-specific attributes. The Policy Engine analyses existing ***Service Authorisation Policies*** in the Policy Repository, as discussed in Section III-A. If the request is permitted, the SPoC issues the requester a *Service Ticket*, which is essentially a security token signed by the SPoC and encrypted by the requester's public key, and the target service's public key, respectively.

The contents of a service ticket include:

- The communication end-point of the e-Health service instance that the SPoC has initialised in the Cloud.
- Opaque pseudonyms of the requester's identity and attributes. The reason for including these in a service ticket will be explained in the following paragraphs.
- A symmetric session key for the service instance and its client to encrypt subsequent application-level messages.
- A time stamp and period of validity of the service ticket.

2) Data Authorisation: The DACAR platform uses Data Buckets [11] to provide long-term persistence of atomic medical data and associated meta-data. Each Data Bucket provides a CRUD service as an interface for e-Health services to access and modify medical data being stored in that bucket.

The Policy Engine requires a data request and a claim regarding the application service and its consumer in order to make an authorisation for CRUD operations on data. The data request provides information about the target Data Bucket, including the qualified name of the medical attribute, a query string that narrows down the result set, and the intended CRUD operations to be carried out on the results. The claim provides the identity of the application service, and the identity and attributes of the service consumer. In practice, a *Service Ticket* is reused for this purpose. This is why a service ticket contains opaque pseudonyms of a service consumer's identity and attributes. On the one hand, this facilitates an application service to impersonate its consumer while requesting medical data. On the other hand, the service consumer's privacy is also protected, as the service cannot reveal its consumer's real identity and attributes from the opaque pseudonyms.

A data authorisation process involves two steps. Firstly, the Policy Engine analyses existing ***Service Subscription Policies*** and ***General Consents*** in the Policy Repository to find out whether a patient has subscribed to this service, and whether the service is trusted in general. If so, the service identity would be sufficient for data access rights to be granted. Otherwise, the Policy Engine resolves the service consumer's identity and attribute pseudonyms and analyses existing ***Specific Consents*** to find out whether this particular service consumer is trusted by the patient. If so, the data access rights should be granted. Access rights to medical data is represented by a *Data Ticket*, which is essentially a security token signed by the SPoC and encrypted by the public keys of the application service and the target CRUD service respectively. The contents of a data ticket include:

- The communication end-point of the CRUD service.
- The approved operations over the result set.
- A symmetric session key for the application service and the CRUD service to encrypt application-level messages.
- A time stamp and period of validity of the data ticket.

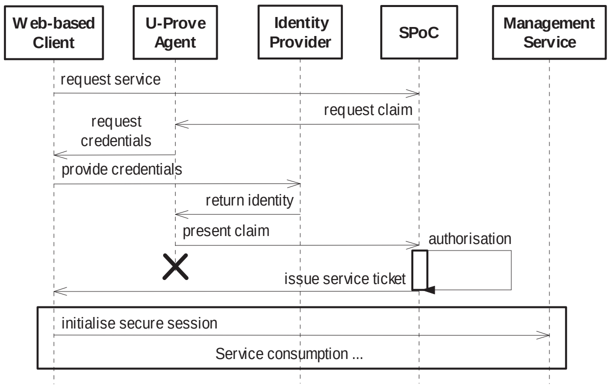

Fig. 3. Application scenario 1: A SPoC authenticates & authorises an internal user to consume an e-Health service

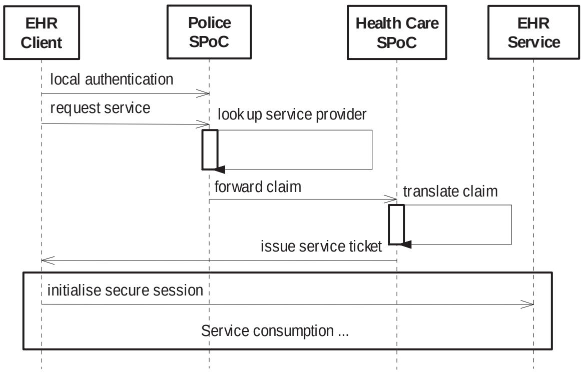

Fig. 4. Application scenario 2: A SPoC authorises an external user to consume an e-Health service

D. Secure Web Service

The DACAR platform provides a software toolkit for programmers to develop Web services with message-level security. As discussed in previous sections, both the *Service Ticket* and the *Data Ticket* carry a symmetric session key, which can be used by an e-Health service, its clients and the CRUD service of a Data Bucket to encrypt confidential application messages with a strong, yet efficient, algorithm, such as the Advanced Encryption Standard (AES) [30].

This security mechanism is designed to be a platform-independent solution for application developers to establish a secure communication model conveniently. It can be applied alone, or on top of any existing transport-level or message-level security mechanisms as a reinforcement.

E. Application Scenarios

This section provides a comprehensive view of the SPoC's work flow by presenting three concrete application scenarios:

Scenario 1: In this scenario, Deirdre, a patient of Chelsea & Westminster Hospital (C&W), wants to update her home address registered with the C&W SPoC using the Personal Information Management Console (PIMC), i.e. a Web-based front-end of the SPoC management service. The work flow of this application scenario is depicted by Figure 3.

Firstly, Deirdre opens the PIMC website in a browser. Because only C&W's internal users are allowed to consume this service, the SPoC needs to authenticate the user's identity. Instead of using a local identity management infrastructure, in this scenario the SPoC uses Federated Identity Providers and redirects Deirdre to a U-Prove Agent.

Secondly, the U-Prove Agent displays a list of trustworthy Identity Providers and Deirdre chooses to log on from one of them, e.g. Windows Live ID. Deirdre enters her credentials on the log on page, and the Identity Provider issues a claim about Deirdre's identity, e.g. "Deirdre@hotmail.com", to the U-Prove Agent. However, it should be noted that the Identity Provider cannot find out that Deirdre is using this identity to mange her health care information through the C&W SPoC, so Deirdre's privacy is protected.

Thirdly, the U-Prove Agent forwards the claim to the SPoC, which in turn starts the authorisation process. The SPoC works out that "Deirdre@hotmail.com" has a patient account with the hospital, and a ***Service Authorisation Policy*** permits the *Patient* role to consume the SPoC management service. Hence, the SPoC issues a *Service Ticket* to the PIMC client, which contains the communication end-point of the SPoC management service in the Cloud and a valid session key.

Fourthly, the PIMC client establishes a secure session with the SPoC management service, and Deirdre uses the Web-based user interface to update her home address.

Scenario 2: In this scenario, David, a police officer of Lothian & Borders Police, wants to view the Electronic Health Record (EHR) of a victim in a traffic accident investigation. The EHR service is managed by the C&W SPoC, which has a trust relationship with the Police SPoC. The work flow of this application scenario is depicted by Figure 4.

Firstly, David logs on from his local SPoC for the police domain and requests the EHR service. The Police SPoC realises that the service being requested is not provided by the local domain, and looks it up in the list of services provided by other SPoCs in the circle of trust. It turns out that the EHR service is offered by the C&W SPoC, so the Police SPoC issues a claim about the requester's identity and attributes, e.g. "**Name**:*David*; **Role**:*Police Officer*". This claim is encrypted, so that only the C&W SPoC is able to view its contents.

Secondly, the Police SPoC forwards the claim, together with an EHR service request, to the C&W SPoC, which translates the claim into local domain ontology, e.g. "**Name**:*David*; **Role**:*Data Viewer*; **Level**:*6*". A ***Service Authorisation Policy*** permits the *Data Viewer* role to consume the EHR service, so the C&W SPoC issues the requester a *Service Ticket* containing the communication end-point of an EHR service instance in the Cloud and a valid session key.

Finally, the EHR client establishes a secure session with the EHR service instance, and David obtains part of the victim's health care information that is classified at or below *Level 6*.

Scenario 3: In this scenario, Kate, a nurse of Chelsea & Westminster Hospital (C&W), wants to set up a number of medical sensors controlled by a handheld device to upload

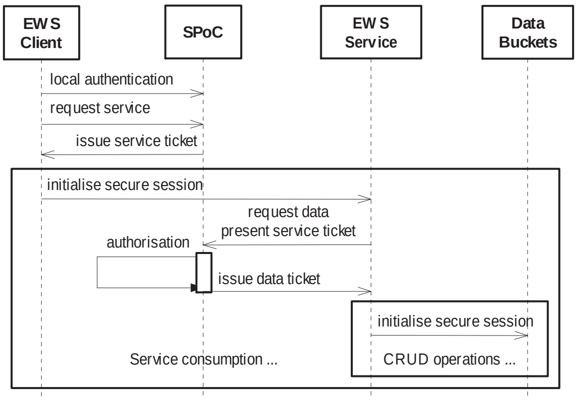

Fig. 5. Application scenario 3: A SPoC authorises an application service to access medical data

Deirdre's six physiological vital signs to the Early Warning Score (EWS) service. The work flow of this application scenario is depicted by Figure 5.

EWS is a clinical service widely used in UK hospitals. Traditionally, EWS requires medical staff to record a patient's blood pressure, heart rate, body temperature, respiration rate, Oxygen saturation and pain level on a paper-based observation chart periodically, and to calculate a risk score according to predefined equations. In the case that a patient is evaluated to be *at risk*, the medical staff should take appropriate actions immediately. This traditional approach is prone to mistakes, as the measurement, recording and calculation work all need to be done manually. DACAR's EWS e-Health service fully automates this process by capturing vital signs using medical sensors, transmitting the values to data buckets using smart hand-held devices, monitoring patient status constantly in real-time, and pushing alerts to medical staffs' mobile phones.

Firstly, Kate starts the EWS client running on a smart handheld device, logs on to the local SPoC and requests the EWS service. A ***Service Authorisation Policy*** permits the *Nurse* role to consume the EWS service, and thus the SPoC issues a *Service Ticket* containing the communication end-point of an EWS service instance in the Cloud, a valid session key and pseudonyms of Kate's identity and role.

Secondly, Kate sets up the application context by scanning her RFID staff card, the RFID label of the ward, and Deirdre's RFID wristband. The EWS client submits the *Service Ticket* to the EWS service instance, and establishes a secure session, captures vital sign values using medical sensors, marks each value with a set of meta-data, such as who captured the measurement for which patient using which device, and at what location and time, and then uploads the data samples to the EWS service.

Thirdly, the EWS service monitors Deirdre's status constantly by analysing the vital sign values it has received. The service also stores these data samples into corresponding Data Buckets, so that other e-Health services, such as Audit Trail, Electronic Health Records and Evidence-based Medicine, may reuse this data. To store data, the EWS service requires access rights to the CRUD services of the Data Buckets. Hence, it sends a data request and the original *Service Ticket* to the SPoC. The SPoC analyses existing ***Specific Consents***, ***Generic Consents*** and ***Service Subscription Policies*** to find out whether Deirdre allows Kate, the *Nurse* role, or the EWS service in general to upload her vital sign data. If so, the SPoC issues a *Data Ticket*, which contains the communication end-points of corresponding CRUD services in the Cloud, the permitted operations and valid session keys.

Finally, the EWS service instance establishes secure sessions with the CRUD services and starts to upload data samples to the Data Buckets.

IV. Implementation

Currently, a prototype of the SPoC has been implemented using Microsoft .NET Framework 4.0, Windows Communication Foundation (WCF) and Windows Identity Foundation (WIF). The *Authentication*, *Authorisation* and *Pseudonym Resolving* services are deployed as self-hosting network services running in Windows Server 2008. The SPoC is integrated with a Windows security domain, authenticates internal users with Kerberos and X.509 certificates, and issues claims about user identities and attributes with Active Directory Federation Service (ADFS) 2.0. The SPoC can also authenticate external users with Federated Identity Providers using U-Prove, but this feature is not mature, as the U-Prove technology is still under development. The SPoC's Domain Ontology database is implemented using SQL Server 2008, and the information sharing policy syntax and the Policy Engine are implemented in Java.

Furthermore, a number of client software and demonstration applications have also been implemented. The SPoC administrator's Configuration Console is implemented with Windows Presentation Foundation (WPF) as both a standalone application and a XAML Browser Application (XBAP). The patient's Personal Information Management Console is implemented with ASP.net. The Early Warning Score (EWS) service is implemented as a WCF service hosted by IIS 7 Web server, and a variety of EWS clients are developed on computer, iPhone and Windows mobile platforms.

The DACAR platform and its proof-of-concept e-Health services have been deployed on the Flexiscale public Cloud [26] for testing and demonstration purposes. In the mean time, a private Cloud infrastructure is being built at Chelsea & Westminster Hospital in London, and selected e-Health services will be evaluated in a real-life clinical environment.

V. Conclusion & Future Work

The main expectations of e-Health are to provide better ways to exchange and share medical information, and to improve the quality of services offered to the patients. However, a significant challenge is to protect patients' privacy and ensure that sensitive medical data is never lost or misused. The Data Capture and Auto Identification Reference (DACAR) project aims to provide a Cloud-based secure e-Health service platform, of which the core component is the **Single Point**

of Contact (SPoC). A SPoC facilitates the development and integration of e-Health services by addressing the most fundamental security requirements, including authentication, authorisation, secure data transmission and persistence.

A SPoC authenticates the users of an e-Health system in many flexible ways. Firstly, it can be integrated with an existing security infrastructure to authenticate internal users, who have accounts in the SPoC's local domain. Secondly, it can manage user identities with Federated Identity Providers using the U-Prove protocol. Last but not least, multiple SPoCs can form a circle of trust and authenticate external users from other trustworthy domains in a claim-based fashion.

A SPoC supports dynamic creation of domain ontologies, which provide a necessary vocabulary for medical information management and access control. Furthermore, a SPoC supports a patient-centric health care model and provides user friendly interfaces for patients to create security policies to govern the utilisation and dissemination of their own medical data. As a security authority, a SPoC issues *Service Tickets* for privileged users to consume various kinds of e-Health services, and *Data Tickets* for privileged e-Health services to create, read, update and delete (CRUD) patients' medical data in Data Buckets. Both the service and data tickets contain symmetric session keys for e-Health services, their clients and CRUD services of Data Buckets to encrypt application-level messages. Hence, the integrity and confidentiality of sensitive medical data are guaranteed in transmission. This security mechanism is platform-independent, and thus can be used alone, or on top of any existing transport-level or message-level security mechanisms as a reinforcement.

Currently, a prototype of the DACAR platform and its demonstration applications have been implemented and deployed on the Flexiscale public Cloud. A more comprehensive evaluation of the platform in a real life clinical environment will be carried out at Chelsea & Westminster Hospital in London. In future work, the design and implementation of the SPoC's policy engine and the U-Prove authentication module will be improved. Another avenue of future work is to build bridges between DACAR and other e-Health service platforms, e.g. Microsoft Health Vault, to enable secure sharing of health care information on a larger scale, and ultimately to integrate primary, secondary and home care.

Acknowledgement

This research is partially supported by grant from UK's Technical Strategy Board and EPSRC. Also, we acknowledge the hard work of our research partners, including Microsoft Corp., HoIP CIC, Flexiant Ltd., Chelsea & Westminster NHS Foundation Trust, Imperial College London, Kodit Database Ltd., GS1 UK Ltd. and CipherLab UK Ltd.

References

[1] D. Slamanig and C. Stingl, "Privacy Aspects of eHealth," in *Proc. of ARES'08*. IEEE, March 2008, pp. 1226–1233.

[2] H. J. Cheong, N. Y. Shin, and Y. B. Joeng, "Improving Korean Service Delivery System in Health Care: Focusing on National E-health System," in *Proc. of eTELEMED'09*. IEEE, 2009, pp. 263–268.

[3] "Federal Health IT Initiatives," [Online] http://www.hhs.gov/healthit

[4] "Canada Health Infoway," [Online] http://www.infoway-inforoute.ca

[5] J. Dzenowagis and G. Kernen, "Global vision, local insight," World Health Organization Press, Report for the World Summit on the Information Society, 2005.

[6] R. B. Prasad, C. Eunmi, and L. Ian, "A Taxonomy and Survey of Cloud Computing Systems," in *Proc. of NCM'09*. IEEE, 2009, pp. 44–51.

[7] M. Endrei, J. Ang, A. Arsanjani, S. Chua, P. Comte, P. Krogdahl, M. Luo, and T. Newling, *Patterns: Service-Oriented Architecture and Web Services*. IBM Redbooks, July 2004.

[8] M. Smith, W. Buchanan, C. Thuemmler, D. Bell, and R. Hazelhoff, "Analysis of Information Governance and Patient Data Protection within Primary Health Care," Int. Journal for Quality in Health Care, 2010.

[9] S. Benzschawel and M. D. Silveira, "Protecting Patient Privacy when Sharing Medical Data," in *Proc of eTELEMED'11*. IEEE, 2011.

[10] DACAR. Data capture and auto identification reference project. TSB/EPSRC project No. 400092. [Online] www.dacar.org.uk

[11] L. Fan, W. Buchanan, C. Thuemmler, O. Lo, A. S. Khedim, O. Uthmani, A. Lawson, and D. Bell, "DACAR Platform for e-Health Services Cloud," in *Proc of CLOUD'11*. IEEE, July 2011, pp. 1–8.

[12] R. Zhang and L. Liu, "Security Models and Requirements for Healthcare Application Clouds," in *Proc. of CLOUD'10*. IEEE, 2010, pp. 268–275.

[13] O. Kilic, A. Dogac, and M. Eichelberg, "Providing Interoperability of eHealth Communities Through Peer-to-Peer Networks," *IEEE TITB*, vol. 14, Issue 3, pp. 846–853, 2010.

[14] W. A. Alrodhan and C. J. Mitchell, "Enhancing User Authentication in Claim-based Identity Management," in *Proc. of CTS'10*. IEEE, 2010, pp. 75–83.

[15] I. Thomas and C. Meinel, "Enhancing Claim-Based Identity Management by Adding a Credibility Level to the Notion of Claims," in *Proc. of SCC'09*. IEEE, 2009, pp. 243–250.

[16] D. Baier, V. Bertocci, K. Brown, E. Pace, and M. Woloski, *A Guide to Claims-based Identity and Access Control*, Patterns & Practices. ISBN: 9780735640597, Microsoft Corp., Jan. 2010.

[17] N. Ragouzis, J. Hughes, R. Philpott, and E. Maler, "Security Assertion Markup Language (SAML) V2.0 Technical Overview," OASIS, Tech. Rep., Oct. 2006.

[18] A. Nadalin, M. Goodner, M. Gudgin, A. Barbir, and H. Granqvist, "WS-Trust Specification v1.4," OASIS Standard, Tech. Rep., Feb. 2009.

[19] "Web Services Federation Language (WS-Federation) v1.1," BEA, BMC, IBM, Layer 7 Technologies, Microsoft, Novell and VeriSign, Tech. Rep., Dec. 2006.

[20] E. Coiera and R. Clarke, "e-Consent: the Design and Implementation of Consumer Consent Mechanism in an Electronic Environment," *JAMIA*, vol. 11, no. 2, pp. 129–140, 2004.

[21] C. Pruski, "e-CRL: A Rule-Based Language for Expressing Patient Electronic Consent," in *Proc. of eTELEMED*. IEEE, 2010, pp. 141–146.

[22] O. Uthmani, W. Buchanan, A. Lawson, C. Thuemmler, and L. Fan, "Novel Information Sharing Syntax for Data Sharing Between Police and Community Partners, Using Role-Based Security," in *Proc. of ECIW'10*. IEEE, 2010, pp. 394–402.

[23] B. Buchanan, L. Fan, A. Lawson, R. Scott, B. Schafer, C. Tuemmler, and O. Uthmani, "Interagency Data Exchange Protocols as Computational Data Protection Law," *JURIX*, vol. 223, pp. 243–147, Dec. 2010.

[24] X. Lin, R. Lu, X. Shen, Y. Nemoto, and N. Kato, "Sage: a Strong Privacy-preserving Scheme Against Global Eavesdropping for eHealth Systems," *IEEE J-SAC*, vol. 27, Issue 4, pp. 365–378, May 2009.

[25] "Health Level Seven," [Online] www.hl7.org

[26] "Flexiscale public Cloud," [Online] http://www.flexiant.com/products/flexiscale/

[27] "The Compound Healthcare Headings Model - What Is It and How to Use It," Clinical Data Structures & Implementation Support, NHS Scotland Data Recording Advisory Service, Tech. Rep., June 2011.

[28] C. Paquin and G. Thompson, "U-Prove CTP White Paper," Microsoft, Tech. Rep., March 2010.

[29] "OpenID Authentication 2.0," OpenID Foundation, [Online] http://openid.net/specs/openid-authentication-2_0.html

[30] J. Daemen and V. Rijmen, *The Design of Rijndael: AES - The Advanced Encryption Standard*. ISBN:3-540-42580-2, Springer, 2002.