Levels of Privacy for eHealth Systems in the Cloud Era

Sinica Alboaie *abss@axiologic.ro, sinica.alboaie@romsoft.eu*
Axiologic Research
RomSoft
Iasi, Romania

Lenuta Alboaie *adria@info.uaic.ro*
Faculty of Computer Science, Alexandru Ioan Cuza University of Iasi
Iasi, Romania

Andrei Panu *andrei.panu@info.uaic.ro*
Faculty of Computer Science, Alexandru Ioan Cuza University of Iasi
Iasi, Romania

Abstract

Enforcing in code privacy laws, internal company rules and principles like Privacy by Design is recognized as a challenge for the IT industry. In this paper we analyze the steps required and propose a guide towards this major goal. Our proposal is to emphasize the need to overcome the limits of service orchestration and create strong privacy and security enabling architectures based on two main ideas. The first idea is to use a semantic firewall that is capable to check privacy properties for the communication between applications and cloud and between cloud's sub-systems. The second idea is to improve current SOA architectures with architectures based on executable choreographies that can be formally verified. In this paper we identify three types of executable choreographies. New types of abstraction which machines can verify and humans can trust are enabled by executable choreographies that act like truly verifiable environments for cloud applications.

Keywords: executable choreography, semantic firewall, privacy levels, SOA for eHealth.

1. Introduction

Ensuring privacy and security of the information contained in electronic health records (EHR) is an important matter for patients and healthcare providers. In order to achieve the potential benefits of eHealth (e.g. better health outcomes, smarter spending, healthier people), each individual must trust the systems involved and the electronic exchange of information between them. Patients must have confidence that their electronic health information is complete, accurate, private and secure [13]. If information is disclosed, patients may be financially (e.g. stolen credit card numbers, social security numbers) or emotionally affected. Also, software providers can face fines and have their reputation damaged if their applications/systems get compromised.

A first step in ensuring patient privacy was made from a legal point of view. Given the fact that governments are responsible for national healthcare, ensuring confidentiality, integrity and availability of patient data became a legal requirement, meaning that every system that deals with health data must comply with the legal reglementations. For example, in the United States of America, the Health Insurance Portability and Accountability Act (HIPAA) and the Health Information Technology for Economic and Clinical Health Act (HITECH) set national rules that apply to companies that work with sensitive patient data [17]. In the European Union there are laws that standardize the provision of cross-border eHealth services [9] and also each member state has its own national laws on electronic health records [18]. There are also many other countries that have implemented patient confidentiality laws or are planning to. This is a good starting point, but it is not enough, because it is difficult for anyone to verify the privacy compliance of a certain technological solution [7].

The next layer of protection is ensured by different technologies implemented in software products, like encryption and de-identification [6]. But given these technologies, software applications that implement them and the underlying IT infrastructure must also be well protected. This task is very difficult, because a lot of people have access to the data (healthcare personnel, IT personnel). There are software applications for privacy surveillance that automatically detect risky behaviors made by users who are accessing patient data [12], but monitoring EHR applications' usage is not enough, because an administrator or a programmer with access to the system can pass by the monitoring solution and directly access the information in the databases. Also data can be leaked through emails, chat, removable media etc. Thus, the underlying IT systems and employee communication must also be monitored and different technologies must be used in order to minimize the potential security breaches [16].

With current architectures, private data losses can go unobserved because there is always a high risk that many breaches evade any detection mechanisms. The growth of cloud and IoT (Internet of Things) applications exacerbate the privacy issues. Human only verification does not represent a feasible solution, except in those cases where verifications are triggered by high impact public scandals involving important persons or by experts discovering huge privacy breaches or stolen data. In e-health systems we create new risks if the technology for storing and processing in cloud those huge amounts of sensitive data do not stay on a very solid technological foundations provided by soundly designed systems.

The concept of privacy should not be perceived in an "all or nothing" manner. It has a gradual nature, thus we should not only think about having privacy or not, instead we should analyze the levels or risk introduced by a software system (estimating the privacy level as a measurement of the risk that data can be copied and used outside its context). If a data breach occurs, it may affect the private information of a single person or may have a major social impact, thus it is normal to differently assess the potential risks and to categorize systems in different levels of privacy [8].

An important proposal we make is for innovations and architectures in the form of powerful abstractions that can be trusted and manageable by people that are not experts in IT. The cloud software architectures should offer transparency regarding how they manage private data and should be verifiable by authorities. In our research we identified a few areas and some specific ideas that make it perfectly feasible to create new architectures for software applications that respect Privacy by Design and Privacy by Default principles.

2. Privacy as a Risk Management Problem

In this paper, we propose an approach based on risk assessment regarding unauthorized access to users' private data. The unauthorized access notion covers both the direct access to data such as private life records, and the access to sensitive information that can jeopardize the safety and security of a larger community.

First, we identify a series of 8 risk areas, each having a unique label that will further be used to describe 7 risk levels that we identified to be assignable to current software architectures and potentially to future ones:

1. The *DEV* risk area: it includes people with access to the system's source code. These may be programmers, administrators or other individuals inside the organization or organizations involved in the development of the system.
2. The *ADMIN* risk area: it includes individuals from inside the organization that hosts and administers the system. This area includes unauthorized data access by individuals with access to the source code or databases. It covers various situations like a person's incompetency that could lead to accessing their personal passwords or access keys by some third party, as well as corruption (e.g. bribe) in order to gain access to the systems they administer for extracting sensitive data.

3. The *HACKERS* risk area: it includes random Internet individuals able to exploit, in commercial or destructive purposes, certain bugs in the technological stack a software solution is based on.
4. The *INDIVIDUAL* risk area: it includes risks associated with the identification of facts or knowledge concerning the private life of a certain individual. The risk covers the ability to directly use private data concerning a certain individual or using Record Linkage techniques able to disclose anonymous data. The most important personal data risks reside in using social engineering or in compromising the various computers or devices one uses to log in the system. These risks cannot be efficiently controlled in the application's architecture or by the cloud provider. An important aspect of our analysis is that the application's provider must have a very good understanding, possibly with a formal proof, of the global risks regarding the data transmitted by the system into the cloud. The current implementation methods are based effectively on the unverifiable competence and assessment of the analysts, programmers and architects involved in creating the system.
5. The *SOCIAL* risk area: it includes risks associated with unauthorized access to sensible data concerning national security or the security of a large community. The medical data of every citizen of a town, region or country is obviously an example of such a special risk category.
6. The *USER TRUST* risk area: it includes the economic and social risks of systems violating the user's privacy expectations. Cloud technologies pose the risk of being rejected by users because they don't know what is shared and there is no formal method to check what private data are shared and with whom. We are talking about rational and justified concerns and not about unjustified paranoia, rumors, etc.
7. The *FORMAL TRUST* risk area: it includes the economic and social risks of systems that were not algorithmically verified, but were designed to comply with a predefined set of privacy properties. The complexity of cloud systems necessitates the development of architectures that are able to support formal verification methods of privacy properties. This area includes risks regarding transmissions of data that aren't anonymized or using databases that contain information that is not or is incorrectly anonymized (linkage techniques may deanonymize them).
8. The *BAD AGENCY* risk area: it includes the risk that certain influential organizations obtain unauthorized information (without the consent of the data owner). These organizations may have very advanced technologies at their disposal and can be capable of placing backdoors in operating systems or hardware equipment. Thus, we have risks regarding the use of technologies without complying with the existing laws, as well as regarding the corruption of one or several individuals with access to the legit activities and technologies of espionage agencies. Furthermore, such a backdoor created by an agency in justifiable legal circumstances, may be disclosed and illegally used by other entities.

3. Privacy Stack

The analysis of the abovementioned risk areas lead us to create and propose a privacy levels stack. These privacy levels are obtained by estimating the risk that unauthorized persons access private data. The risk assessment methods are based on formal code analysis enabling the identification of different types of individuals inside the organization who may eventually access private data outside their job description and the assessment of various external attacks categories resulting in information leaks.

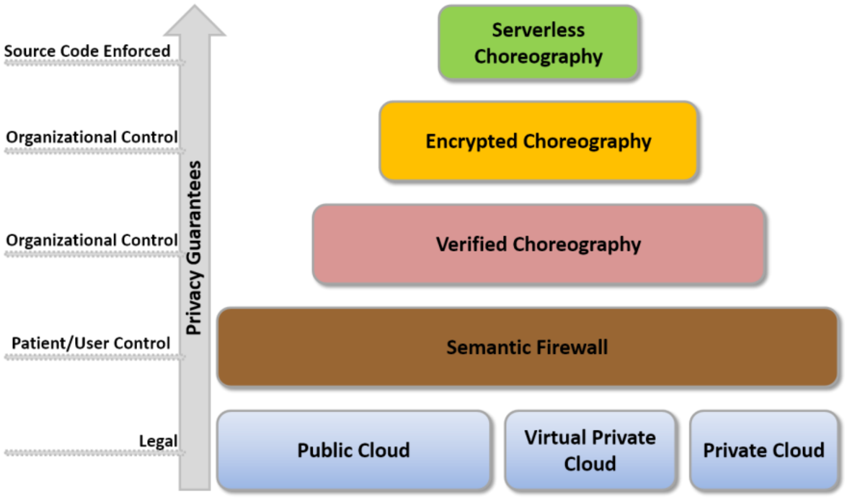

Fig. 1. Privacy stack.

3.1. Level 1 of Privacy or Public Cloud Hosting

This privacy level is offered by the usual PaaS systems that rent resources to tenants such as hospitals, private practices, etc. Although the major corporations are offering certain privacy guarantees, everything is covered by dense legal agreements, the practical applicability being difficult to prove. The rules and information reaching these systems, as well as the information leaks towards other entities are almost impossible to be verified and understood by the end user. Basically, both the tenant and the user are offering a blank cheque to the provider of the certain solution who is, at the same time, the administrator of the software solution. This kind of solutions carries the risk of theft and mass analysis of large data volumes by the provider's administrators and programmers that administer the system. In these situations an audit does not exist and there aren't any other barriers except those offered by the operating systems and the legal provisions. These offer an especially weak protection against some code injected by a programmer or against the actions of system or database administrators. These systems do not usually offer formal verification methods or enable the use of personal data protection technologies. Recently, companies such as Google have begun to implement techniques such as Differential Privacy [11] that ensure some formal guarantees. However, from a user or a client company perspective, should the data be managed by companies in different countries, the trust relationship is asymmetrical and impossible to validate.

3.2. Level 2 of Privacy or Virtual Private Cloud Hosting

This privacy level refers to using virtual machines, usually connected through VPN systems, that ensure a certain additional isolation against outside attacks. These systems are usually developed and administrated by different teams. Using a VPN between subsystems offers a better isolation against the outside world and may lower the level of risk deliberately induced by the developers. The risk level is lower because of the increased chance for the attacks to be detected. In all other perspectives, a private virtual cloud is just as inefficient in the protection of private medical data as the 1st level.

3.3. Level 3 of Privacy or Private Cloud Hosting

Using a private cloud reduces risks given the existence of a number of people involved in system administration but otherwise, it is almost identical to the 2nd level. It is possible that users have an increased confidence knowing that the system is not entirely hosted in a public cloud, but rather administrated by a trusted institution. In certain situations, the interest, motivation and competence level of the system administrators, as well as of the software solution providers, might make the 3rd level to be inferior to the 1st or 2nd levels. Nevertheless, in many cases, the users may experience a higher level of trust when compared against the 2nd and the 1st level.

3.4. Level 4 of Privacy or Software Architectures Built Around a Semantic Firewall

With this level we go from the area of widely used approaches in the industry to an experimental research area. Intuitively, a semantic firewall understands enough about the data transmitted to the cloud by a patient's connected devices or sensors, with the purpose of warning the user if sensitive data regarding privacy is sent. Thus, the user may decide if he allows the application to transmit the private data outside. For such a semantic firewall to properly operate, ontologies that help identify risks and generate relevant warning messages for the end user must exist. Initiatives on using ontologies in creating a semantic firewall may be found in [22]. A semantic firewall may offer additional guarantees to the user covering the type of data transmitted in cloud, it may deduce relevant information on the anonymization level and the record linkage risks the users' data is exposed to. In the 4th chapter we will further develop our approach on how to achieve this privacy level.

3.5. Level 5 of Privacy or Software Architectures Build Around Verified Choreography

The usage of formal verification models of source code is not widely applied in the industry. However, we suggests that instead of trying an exhaustive verification of the program's correctness, one may proceed to verify limited system properties. These properties may be about the way the system backend stores private data or what information is sent to the exterior. Thus, static and dynamic code analysis methods can be applied, automatically proving violation of invariants or private data usage rules. The composition of Web services or services in general can be done in a centralized way, under the authority of a single legal entity, this being the case of orchestration, or otherwise decentralized, involving the cooperation of several individual entities, through executable choreographies. Currently there are few executable choreography solutions (e.g. [1] [2] [3] [5]). Static code analysis can be easily applied to the executable choreography systems based on swarm communication paradigm [swarm-communication], because they offer two separate layers for APIs and composition. By analyzing the use of the data models presented in the 3rd chapter, we will present in the next chapter a simple example of inference on private data, realized on a swarm executable choreography. Similar formal verification ideas for privacy properties may be developed for other types of orchestrations and choreographies. These verifications cannot entirely eliminate the risks of unauthorized private data use, but may significantly decrease them.

3.6. Level 6 of Privacy or Software Architectures Built Around Encrypted Choreography

The 6th level is based on the idea that using cryptographic methods in an executable choreography, it is possible to guarantee that different legal entities or different departments of a company have access to data in a way that enables them to detect unauthorized access to protected information. The private data will be anonymized and managed on different servers and administrated by different individuals in different departments. The data on a certain server will not make any sense on themselves and only by executing a choreography involving data on 3 or 4 such systems may the information be of use. In such a system, obtaining unauthorized access requires individuals from different institutions or departments to conspire.

Executing choreographies that are verified using level 5 techniques or composing encrypted data only on users' end-systems will significantly decrease the risk that system administrators or programmers obtain access to unauthorized data. This approach is not possible for all applications. To store medical imaging data or medical tests data we developed prototypes proving the viability of such a system. Various patient identification data are anonymized and different information regarding relationships between data are stored encrypted on different servers, while the processing is made also on different nodes. Our experiments showed that a Swarm choreography system enables both manual and automated verification of the fact that each sub-system may only access a specific data type and has never access to another data type [19].

3.7. Level 7 of Privacy or Public Cloud Hosting

The 7th privacy level implies that applications are no longer hosted by a single legal entity and that the hosting location is controlled by the user or by the provider of the medical solution. In this approach, the cloud services providers will offer only general hosting services for light virtualization systems such as Linux containers (e.g. Docker [10]). They will have almost no information about the type and the kind of the applications that are using their infrastructure. If the applications use the techniques from the 6th level, the cloud services providers will only see entirely anonymized data or simply encrypted data. By “serverless” we mean that the server side code is not hosted by the author of the application but it will be running anonymously in cloud in a transparent way (that can be eventually user chosen or randomly chosen among competing cloud providers). For each user, private data can be stored and computed in a different place, making almost impossible for an attacker to access huge amounts of data without compromising very large parts of the cloud. Even in such improbable cases, an attacker will have serious difficulties because he will be required to use huge computational resources to understand and make sense of the collected data.

This is the only level able to give cryptographic guarantees that the data truly belongs to the patient and without the decryption key stored on the patient’s own devices, the data may not be decrypted. These are the only systems able to lead the way for safely sharing high risk medical information.

In Table 1 we present a synthesis with our proposal for levels of privacy that potential software architectures for cloud systems could and should sustain.

Table 1. Levels of privacy and the associated risks.

Levels & Risks	**DEV**	**ADMIN**	**HACKER**	**NO FORMAL TRUST**	**LOOSE USER TRUST**	**SOCIAL**	**INDIVID UAL**	**BAD AGENCY**	**Risk Score**
Level 1 *Public Cloud*	high	high	high	high	high	high	high	high	24
Level 2 Virtual Private Cloud	may be	high	may be	high	high	may be	high	high	21
Level 3 *Private Cloud*	low	high	may be	high	may be	may be	high	high	20
Level 4 *Semantic Firewall*	low	low	may be	may be	low	may be	high	high	10
Level 5 *Verified Choreography*	low	low	may be	low	low	may be	may be	low	9
Level 6 *Encrypted Choreography*	no	no	low	no	no	low	may be	may be	6
Level 7 *Serverless Choreography*	no	no	low	no	no	no	may be	low	4

Legend:

- no (0): there is no risk or extremely low risks;
- low (1): there is some risk, not zero but fairly acceptable;
- may be (2): there is a medium risk, even a meaningful risk in some scenarios but not high;
- high (3): there is a significant risk, privacy loose can happen anytime fairly easy and without noticing.

4. A Privacy Properties Verification Model

In order to verify the executable choreography and the semantic firewall properties described in section 3.4, we propose an abstract model based on predicate logic (first-order predicate calculus). The model is based on resources, privacy areas or access areas, and predicates that interpreted are offering a database of parent-son relationships between the resources and the access areas. Intuitively, the predicates represent facts from a knowledge base about the code used to describe models or executable choreographies. We may thus have facts such as "parent(R1, RP2)" or "parent(Z1, ZP2)" and facts such as "grant(Z,R)". Parent means that RP2 is a parent resource for R1, the fact that ZP2 is a parent zone for Z1 or that the Z zone has access right on the R resource and its children. Additional predicates of the "allow(R,Z)" type may be inferred from this facts, based on the following inference rules:

```
1. grant(R,Z) -> allow(R,Z)
2. grant(R,Z1) AND parent(Z1,Z2) -> allow(R,Z2)
3. parent(R1,R2) AND grant(R2,Z) -> allow(R1,Z)
```

The resources may have the parent resource with the intuition that the rights on the parent resources are automatically transferred on the children resources. The zones may have parents on their turn, with the intuition that the rights of a parent zone over other resources are automatically transmitted to the children.

The "allow" type predicates represent practically a conclusion resulted from the grant and parent relationships according to the knowledge base detected through static or dynamic code analysis.

There are several approaches and even languages for specifying privacy preferences and policies [4] [14] [15]. Our approach is novel because it aims at demonstrating properties in an executable choreography system where it is important to determine which association between identification data and private data (represented by resources) reach the actors participating to the choreography (represented by the access zones).

5. Techniques to Accomplish the Privacy Stack

5.1. Semantic Communication Firewall for eHealth Systems

In this section, we are proposing a semantic firewall concept especially useful for guaranteeing privacy properties of applications that communicate private medical data from mobile phones. The communication methods widely employed in the last years that are using web services or web sockets (SOAP, REST, etc.) are based on the assumption that they are communicating objects. Our approach is based on the same fact – the communication is made by serializing a number of objects, as they are defined in OOP (object oriented programming). An object is a collection of members (fields). Our analysis identified that beside the necessity to label an object's members with information on their type as they are represented in the memory in order to be used by algorithms, it is necessary to also label the object's members with information regarding privacy. We thus determined that there are data helping to identify the user (Identification Fields) and data sensible for the user (Personal Fields).

According to [ontology-firewall], as "Identification Fields" we may list:

- name, first name, date of birth and place of birth;

- contact information: postal address, phone number, email address;
- IMSI: International Mobile Subscriber Identity;
- IMEI: International Mobile Equipment Identity;
- directory of contacts;
- location data: GPS, IP address, Cell Id;
- data on patient related events and activities;
- certificates and cryptographic keys.

As "Personal Fields" we have:

- patient financial data;
- any data provided by sensors concerning patient's general health condition;
- encrypted data.

All the fields and the objects they compose represent, according to the formalism described in section 4, resources. Any data transmission to a third party means, in the interpretation of the mentioned formalism, the communication with an access zone. Any communication comprising a combination of Identification Fields and Personal Fields, directly or by a series of objects transmitted in succession, is creating a new type of resource which we will identify by a Privacy Risk type resource.

Some information such as the IP address may be transmitted implicitly, that's why any proper implementation of a semantic firewall must hide this kind of information by using anonymization techniques of the source address (like proxies that hide the IP address or other techniques such as the TOR project [20]).

The problem of determining the confidentiality and risk levels is reduced to determining the existence of the facts (the truth value of the predicates) concerning "allow" (Zones, Privacy Risk resources). If the application communicates with a single Zone, the problem is quite simple, but we may easily imagine systems communicating with different access zones, controlled by different legal entities, such as described in section 3.7. Thus, the semantic firewall may both detect the private data transfer from the user's devices (and asks for approval), as well as allow using several zones to reduce the risks of unauthorized private data leaks.

5.2. Advanced Executable Choreographies

In this section, we propose an interpretation model for resources and zones applicable for the implementation of the 5th to 7th levels of the privacy stack. As presented above, the verification model suggested in the 4th chapter is enabling the 3 distinct categories of executable choreographies.

If the access zones are representing departments, teams or servers under the control of a single or various companies, we may extend the verification model beyond the simple communication with the cloud. The verification model identifies all the risk areas by detecting the "allow" predicates between these zones and the Privacy Risk resources. For example, such a system may automatically identify that by using external Web services there is the risk of transmitting private information. Furthermore, it is possible to know exactly which departments, servers, individuals (all representing access Zones) have access to private information and which combinations of private and identification information are available to them. Using this type of choreography does not stop data leaks caused by the inside personnel, but rather makes it easier for the investigation process in case of possible incidents and allows implementing a system access policy meant to minimize the risks.

The encrypted choreography implies that beside verification, the system is built out of verifiable subsystems in order to guarantee that no single department or individual holds at the same time the encrypted data and the decryption keys concerning the same Privacy Risk resources. Thus, the administrators or the programmers of these subsystems represent a reduced risk. Even if in the case of certain applications this approach cannot be used, the encrypted choreography may minimize the risks induced by the inside individuals that must administer or program these systems. There is a formal guarantee that there are as few as possible access zones (ideally never a single one) to Privacy Risk resources. This form of choreography is useful because it enables companies to guarantee by code the compliance with the legislation or the provisioned security regulations.

The serverless choreography implies the development and a certain level of maturity of the virtualization techniques, as well as new economic application hosting models. For example, the Hidden Service Protocol concept from TOR [21] provides an incipient model for such systems. The serverless choreography implies the cloud execution and storage is made using encrypted choreography, while the access zones are not controlled by a single company. Using this type of choreography we are able to achieve a context wherein the hosting companies and the individuals physically administrating the hosting infrastructure have practically no means to significantly influence the sub-components of the hosted applications. The serverless choreography brings forth the opportunity that such systems that are able to formally guarantee advanced privacy properties may be widely available.

6. Conclusions

By analyzing and implementing integration systems in the medical environment, we noticed that from the perspective of obtaining good privacy properties we have to create novel approaches regarding systems architecture at the communication level , data storage and cloud computation. In this paper we propose 7 privacy levels applicable to analyze use cases and applications in the e-health field.

In the context of the present article we approach two kinds of applications: classical enterprise and cloud applications that manage medical data generated by specialized health devices equipping private practices or hospitals, and applications that generate medical data using sensors and communicate data in cloud by means of patients' own mobile devices. Although the two application categories are requiring different approaches and assessments from the perspective of private data protection, this paper aims at a unified approach that can be applied differently depending on the type of the application. While ideas presented in this article could be extended to other areas, we focus ourselves in this paper on analyzing mobile and Web applications that share and analyze medical data.

Executable choreography will create transparent and verifiable environment for cloud computation by providing new high level abstractions (verifiable executable choreographies) that machines and humans can directly monitor. Our proposal is to emphasize the need to overcome the limits of service orchestration and create strong privacy and security enabling architectures based on executable services choreography concepts. To make the patient the real owner of his medical data we need large scale choreography enabling distributed computation systems that provide safe and performant execution of the signed code belonging to the "serverless" applications of the future. We foresee that is possible to create dynamic and static code verification technologies that automatically reject privacy breakage consume of unjustified resources or access to private data at large scale. This layer could also automatically inform the user on how their private data is used and stored, that without automate verification can be different from what the app provider declares or even believes. The privacy levels presented in this article can provide a guidance to follow developing medical applications that have user friendly behavior regarding privacy.

Acknowledgements

This work was supported by the European Commission under the Horizon 2020 Programme (H2020), as part of the OPERANDO project (Grant Agreement no. 653704).

References

[1] Akkawi, F., Fletcher, D.P., Cottenier, T., Duncavage, D.P., Alena, R. L. and Elrad, T. 2006. "An executable choreography framework for dynamic service-oriented architectures", Aerospace Conference. IEEE

[2] Alboaie, L., Alboaie, S., Barbu, T. 2014. "Extending swarm communication to unify choreography and long-lived processes", 23rd International Conference on Information Systems Development (ISD 2014)

[3] Alboaie, L., Alboaie, S. and Panu, A. 2013. "Swarm Communication - A Messaging Pattern Proposal for Dynamic Scalability in Cloud", 15th IEEE International Conference on High Performance Computing and Communications (HPCC 2013). IEEE, pp. 1930 - 1937.

[4] Ashley, P., Hada, S., Karjoth, G., Powers, C., and Schunter, M. (2003). "Enterprise privacy authorization language (EPAL 1.2)". Submission to W3C.

[5] Besana, P. and Barker, A. 2009. "An Executable Calculus for Service Choreography", in Proceedings of the Confederated International Conferences, CoopIS, DOA, IS, and ODBASE. Springer, pp. 373-380.

[6] Cavoukian, A., and El Emam, K. 2014. "De-identification Protocols: Essential for Protecting Privacy".

[7] Cavoukian, A., and Jutla, D. 2014. "Privacy Policies Are Not Enough: We Need Software Transparency".

[8] Cavoukian, A., Shapiro, S. and Cronk, R. J. 2014. "Privacy Engineering: Proactively Embedding Privacy, by Design".

[9] Directive 2011/24/EU on patients' rights in cross-border healthcare. http://ec.europa.eu/health/cross_border_care/policy/index_en.htm.

[10] Docker. http://www.docker.com

[11] Erlingsson, U., Pihur, V. and Korolova, A. 2014. "RAPPOR: Randomized Aggregatable Privacy-Preserving Ordinal Response", proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security (CCS 2014). ACM, pp. 1054-1067.

[12] FairWarning. http://www.fairwarning.com/

[13] Health Information Privacy, Security, and Your EHR. http://www.healthit.gov/providers-professionals/ehr-privacy-security.

[14] Kolter, J.P. 2010. "User-centric Privacy: A Usable and Provider-independent Privacy Infrastructure". Josef Eul Verlag GmbH.

[15] Moritz Y. Becker, M. Y., Malkis, A. and Bussard, L. 2010. "A practical generic privacy language", in Proceedings of the 6th international conference on Information systems security (ICISS'10). Springer-Verlag, pp. 125-139.

[16] NitroSecurity and FairWarning. "White Paper: Security and Privacy of Electronic Medical Records". http://www.himss.org/files/himssorg/content/files/securityandprivacyofelectronicmedicalrecords.pdf

[17] OnRamp. "What is HIPAA Compliance?". http://www.onr.com/solutions/compliant-hosting/hipaa-compliant-hosting/what-is- hipaa-compliance/

[18] Overview of the national laws on electronic health records in the EU Member States. http://ec.europa.eu/health/ehealth/projects/nationallaws_electronichealthrecords _en.htm.

[19] Alboaie, S. 2015. "Semantic firewall prototype". https://github.com/salboaie/semantic-firewall/

[20] Tor Project. https://www.torproject.org/

[21] Tor: Hidden Service Protocol. https://www.torproject.org/docs/hidden-services.html.en

[22] Vincent, J., Porquet, C., Borsali, M., Leboulanger, H. 2011. "Privacy Protection for Smartphones: An Ontology-Based Firewall", Workshop in Information Security Theory and Practice. Springer, pp.371-380.