A Demonstration of 4D Digital Avatar Infrastructure for Access of Complete Patient Information

Project acronym: MyHealthAvatar

Deliverable No. D3.3
Security Measures and Guidelines

Grant agreement no: 600929

Dissemination Level		
PU	Public	X
PP	Restricted to other programme participants (including the Commission Services)	
RE	Restricted to a group specified by the consortium (including the Commission Services)	
CO	Confidential, only for members of the consortium (including the Commission Services)	

COVER AND CONTROL PAGE OF DOCUMENT	
Project Acronym:	MyHealthAvatar
Project Full Name:	A Demonstration of 4D Digital Avatar Infrastructure for Access of Complete Patient Information
Deliverable No.:	D3.3
Document name:	Security measures and guidelines
Nature (R, P, D, O)[1]	R
Dissemination Level (PU, PP, RE, CO)[2]	PU
Version:	2.0
Actual Submission Date:	30/12/2015
Editor: Institution: E-Mail:	Manolis Tsiknakis TEI-C tsiknaki@epp.teicrete.gr

[1] **R**=Report, **P**=Prototype, **D**=Demonstrator, **O**=Other

[2] **PU**=Public, **PP**=Restricted to other programme participants (including the Commission Services), **RE**=Restricted to a group specified by the consortium (including the Commission Services), **CO**=Confidential, only for members of the consortium (including the Commission Services)

ABSTRACT:

This deliverable reports the work done for building and maintaining the structure of the architecture platform by investigating and reporting security issues and measures for infrastructure, resource management, data access and federation, computing resource (possible links with external HPC). It deals with all the security aspects of the technological platform, ranging from user authentication, authorization, and auditing, to data integrity and privacy, to pseudo anonymization and re identification of patient data. The security tools and policies that are developed ensure and enforce the legal and regulatory compliance and encompasses the appropriate auditing mechanisms that are needed by the legislation. This deliverable includes guidelines on how to pose these security mechanism in MHA platform and examples on how to use these guidelines for the MHA use cases.

KEYWORD LIST: Security, Authentication, Authorisation, Audit, Auditing, SAML, WS-Security, WS-Federation, WSTrust, openXDAS

The research leading to these results has received funding from the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement n° 600929.

The author is solely responsible for its content, it does not represent the opinion of the European Community and the Community is not responsible for any use that might be made of data appearing therein.

MODIFICATION CONTROL			
Version	Date	Status	Author
1.0	06/10/2015	Draft	Manolis Tsiknakis
1.1	05/12/2015	Draft	Stelios Sfakianakis, Kostas Marias
1.2	05/01/2016	Draft	Emmanouil G. Spanakis, Stelios Sfakianakis
1.4	25/01/2016	Draft	Feng Dong, Nikolaus Forgo, Sarah Jensen, Zhikun Deng
2.0	15/02/2016	Final	Manolis Tsiknakis Emmanouil G. Spanakis Stelios Sfakianakis

List of contributors

- Tsiknakis Manolis (TEI-CRETE)
- George Vavoulas (TEI-CRETE)
- Emmanouil G. Spanakis (FORTH-ICS)
- Stelios Sfakianakis (FORTH-ICS)
- Kostas Marias (FORTH-ICS)
- Nikolaus Forgó (LUH)
- Feng Dong (BED)
- Sarah Jensen (LUH)
- Zhikun Deng (BED)

Contents

Contents

1 Executive Summary

MyHealthAvatar proposes a solution for access, collection and sharing of long term and consistent personal health status data through an integrated environment, which will allow more sophisticated clinical data analysis, prediction, prevention and *in silico* treatment simulations tailored to the individual citizen. This solution is able to support: Information collection and access (Internal data repositories to store individual data for the avatars; links to external sources; model repositories, information extraction from the web and data collection using mobile apps; semantics and linked data to support the data/model searching and reasoning.), Data management and sharing, as well as Information analysis using integrated toolboxes.

MyHealthAvatar follows recommendations from relevant VPH activities on "Digital Patient". MyHealthAvatar architectural platform is designed as an integrated facility that allows multiple functionalities rather than just a data storage facility as in the previous attempts. It's distinctive features include:

- Data and model repositories to provide rich resources of data and models
- ICT utilities to support data collection with minimal user input, including web information extraction, mobile apps, etc.
- ICT toolbox to support clinical decisions by using models and visual analytics.
- Ontology and RDF repositories to support data search and reasoning.
- A cloud based ICT architecture that allows the access of data from a range of different sources, and integration of the repositories, the toolbox and the ICT utilities.
- A local cloud solution to support the computing requirement for the avatars without remote data transfer.
- A proof of market on open sources for MyHealthAvatar APIs

This deliverable reports the work done for building and maintaining the security guideline framework of MHA platform by investigating and reporting security issues and measures for infrastructure, resource management, data access and federation, computing resource (possible links with external HPC). It deals with all security aspects of the technological platform, ranging from user authentication, authorization, and auditing, to data integrity and privacy, to pseudo anonymization and re identification of patient data. The security tools and policies that are developed ensure and enforce the legal and regulatory compliance and the appropriate auditing mechanisms that are needed by the legislation for MHA. This deliverable includes guidelines on how to pose these security mechanism in MHA platform

2 Introduction

Security has been a major concern for cloud platform developers since the beginning. In our case, the significance of security for a cloud facility, that implements a storage service which handles extremely sensitive data like biomedical data, security is of vast importance and thus, an analysis of the security model and possibly a threat analysis for OpenStack and especially Swift must take place.

Several non-profit or federal organizations have analyzed security and privacy issues significant to several cloud computing models. Here, we summarize some key issues as designated by NIST and ENISA[3,4] and analyze the ways in which OpenStack deals with these issues.

- Governance: involves policies and procedures which affect information security.
- Compliance: cloud software must comply with laws, regulations, and privacy specifications which among other, affect data location and electronic discovery. The latter, deals with stored data, metadata and even non-rendered file content, a collection and processing of which, can lead to identification and collection of sensitive information.
- Trust: when data are moved to a cloud platform, even in the case of a private cloud, control over data is relinquished to the cloud provider. The relevant issues that arise with major effect on the storage potion of the cloud, can be categorized as:
    - Insider issues, which involves current or former personnel of the organization that operates the cloud, and even affiliated parties which operate at network or system level.
    - Data ownership, which can affect intellectual property rights or copyrights.
- Architecture: commonly cloud services involve several applications, abstractions and a complex under the hood setup. The combination of these, is exposed to clients and users, usually in the form of internet-aware services in order to deliver various facilities like provisioning, migration of virtual machines, imaging services and so on. Moreover, in a virtualized environment, another software layer exists between the operating system and the actual hardware platform, that of the hypervisor, which is exposed, usually in indirect ways to the clients.

  To summarize, the architectural issues can affect:

    - The attack surface of the infrastructure, with the addition of the hypervisor
    - Virtual network protection, since most platforms enable the creation of several private networks for the virtual machines.

[3] W. Jansen and T. Grance, "Guidelines on Security and Privacy in Public Cloud Computing," NIST Special Publication 800-144, 2011.
[4] D. Catteddu and G. Hogben, "Cloud Computing, Benefits, risks and recommendations for information security," EINSA, 2009.

  - Images, which can be either standard ISO images or ready-to-use customized virtual machine images effectively with data.
  - Client side applications, usually developed to deliver cloud storage services or virtual machine management, may be built on top of the cloud public API, which may have security implications.
- Identity and Access Control: Although most cloud infrastructures provide access control mechanisms over data and cloud internet services, several implications can appear due to the fact that usually a cloud provider may take advantage of federated identification services, sharing digital identities with external entities, using Security Assertion Markup Language (SAML)[5] or the OpenID standard[6].

MyHealthAvatar utilizes the latest architecture technology on cloud to lay down the foundation and allow high information security and effective integration of different components of the avatar to achieve high performance. MyHealthAvatar encourages interoperability by building as much as possible upon widely accepted security standards (e.g. SAML, Liberty-Alliance, WS-*, PKIX, XACML, OAuth 1.0 and OAuth 2.0, etc). Use case(s) guarantee data security in collecting, sharing and exploiting all kind of data linked to the Avatar through the secure API access. This deliverable aims to define the security framework, measures and guidelines of MyHealthAvatar platform. This deliverable reports the work done for building and maintaining the structure of the security platform framework by investigating and reporting security issues and measures for infrastructure, resource management, data access and federation, computing resource (possible links with external HPC). It deals with all the security aspects of the technological platform, ranging from user authentication, authorization, and auditing, to data integrity and privacy, to pseudo anonymization and re identification of patient data.

[5] "OASIS SAML Wiki," OASIS Security Services (SAML) Technical Committee, [Online]. Available: https://wiki.oasis-open.org/security/FrontPage. [Accessed 8 9 2014].
[6] O. Foundation, "Specifications," [Online]. Available: http://openid.net/developers/specs/. [Accessed 8 9 2014].

3 MyHealthAvatar platform and deployment infrastructure

MyHealthAvatar, collecting, accessing, managing and possibly sharing healthcare related data are not only important to individuals who can manage their own health, but also important for clinicians and other healthcare workers for patient monitoring and providing suitable in-time care. The consideration of cloud technology was vital to ensure the long term scalability and performance of MyHealthAvatar in data management and service management architecture.

Figure 1 shows the delivery models supported by MHA within the deployed cloud infrastructure. Their scope is to provide resources, application platforms, common infrastructure and software as services to consumer[7] shown in Figure. 1. These service models also place different levels of security requirements upon the deployed environment. As capabilities are inherited by successive models, so too are information security issues and risks.

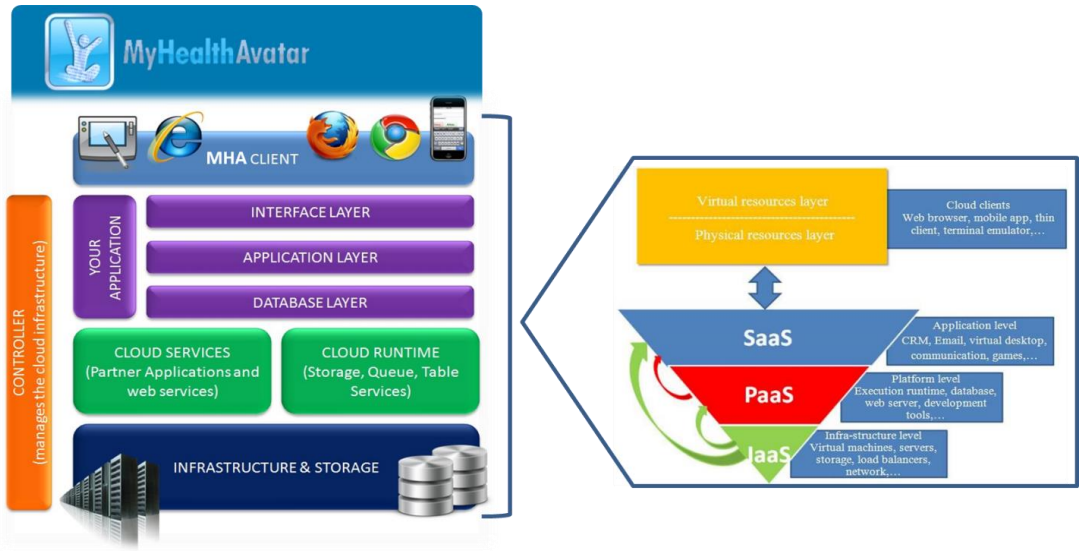


Figure 1: Cloud computing architecture considered for MHA

The main approach is to work on a locally-deployed cloud infrastructure which can utilize local computing power as well as maintain the ability to outsource the infrastructure to commercial cloud computing facilities (e.g. Amazon EC2). This way we are able to provide and sustain stable and production ready resources to support MHA needs regarding data preservation. The swift services are highly autonomous so the whole architecture is flexible enough to allow different deployment scenarios. The four main services are: Proxy Services, Object Services, Container Services and Account Services. Proxy Service plays role of the contact point (API) with users and 3rd party services, while

[7] Shey, H., R. Wang, J.P. Garbini and E. Daley, 2009. The State of Enterprise Software: 2009. Forrester Research, Inc.

other three kind of services manage files, containers and accounts so are used to manage physical data and logical structure. The major difference between locally deployed cloud (also known as private cloud) and public cloud is the control and access of the resource. In private cloud, resources are controlled and accessed by the premise only. On the other hand, in public cloud, resources are controlled by the cloud provider but are accessible for public users. Therefore, a hybrid cloud infrastructure is adopted in MyHealthAvatar with both public and private cloud facilities available. In deliverable D3.2 2nd version we have included detailed information about the private and public cloud deployment of MHA including all technical information of the supporting cloud software and services.

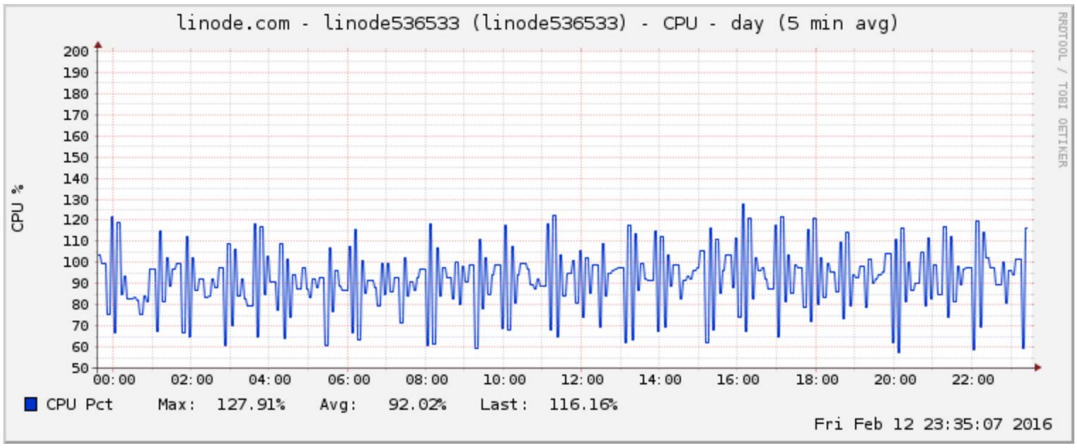
Physical Deployment

MHA platform is deployed in the premises of FORTH, in the form of a private computational and storage cloud.

Figure 2: Physical installation

In terms of hardware resources, the cloud infrastructure allows for maximum elasticity and flexibility by effectively adapting to the load of any given time. The current minimal specifications include: 300 GB of RAM, 9TB of storage and 16 cores Intel® Xeon® Processor E5-2690 and 4 cores Intel® Xeon® Processor E7520 (Dell PowerEdge R720 and SC 1425 Servers series). In terms of the software the OpenStack[15] open source cloud computing software has been installed on the machines using the Linux Ubuntu 12.04 operating system[16]. **Fort MHA we can have one or more VMs for 4GB RAM and 100GB for storage (or any other different setting for each VM), according to MHA deployment demands. We are able to serve MHA need as the project progress.**

MHA is also installed and deployed physically on a public cloud namely *Linode* which a cloud hosting service, the 8GB plan is chosen to deploy on Linode's London based servers. The virtual machine has 6 CPU cores, 192GB SSD storage, 8GB RAM. Linode provides a web based panel to monitor the virtual machine's resource usage.



MyHealthAvatar utilize the power of both Forth cloud and Linode cloud services to form a hybrid cloud which gives the platform the benefit of both world.

4 MHA privacy and legal issues

The legal framework of MHA ensures a lawful and fair data processing during the testing stage of the project, but also for the exploitation stage after the project's end. The following section provides an overview of the most important legal aspects that need to be considered for the architecture design. The further details are explained in D11.1, D11.3 and D11.4. Since sensitive health and lifestyle data can be processed in the exploitation stage after the project's end, measures need to be taken in order to protect the stored data against unauthorised disclosure or access, accidental or unlawful destruction or accidental loss or alteration according to Article 17 (1) of the Data Protection Directive (Directive 95/46/EC[8]).

4.1 Legal and data protection rules within MHA

For the testing phase of MHA it was it is crucial to know if personal data is going to be processed because the Data Protection Directive as the major source for the legal requirements is applicable only if personal data are processed. If this is the case the processing of personal data is forbidden, except if there is a legal basis or the data subject has given informed consent. Moreover, the need for fair processing requires to de-identify the data whenever possible and to use anonymised or at least pseudonymised data. According to the principle of limited retention the data must be erased as soon as it is no longer needed for the purposes for which they were collected. The principle of data minimization states that personal data should be collected only if really needed and the principle of purpose limitation that also needs to be considered means that data must generally not be processed

[8] http://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=CELEX:31995L0046:en:HTML

in a way incompatible with the purposes at collection. The architecture design must ensure for all stages of the project that personal data is protected against accidental or unlawful destruction or accidental loss, alteration, unauthorized disclosure or access by technical and organizational measures. This is especially true for the exploitation stage of the project when users can store sensitive health data. The different type of data is discussed in the following sections

4.1.1 Personal Data (citizen)

Pursuant to Article 2 (a) of the Data Protection Directive, personal data is "*any information relating to an identified or identifiable natural person ('data subject'); an identifiable person is one who can be identified, directly or indirectly, in particular by reference to an identification number or to one or more factors specific to his physical, physiological, mental, economic, cultural or social identity*". To determine whether a person is identifiable recital 26 states that account should be taken of all the means likely reasonably to be used either by the controller or by any other person to identify said person. So a person is not identifiable when *"all the means likely reasonably to be used"* do not exist and consequently the information cannot be considered as personal data.[9]

If citizens store data in their personal digital avatar on platform, e.g. data collected by apps and devices, the person behind the data is identifiable and thus the rules of the Data Protection Directive apply, amongst others that there must be no data processing without consent or another basis in law.

4.1.2 Medical Data (patient)

Since certain information is more important for the privacy of a person than other kinds of information, the Data Protection Directive provides special rules for sensitive data such as data concerning health. Here the strict rules of Article 8 must be taken into account in addition to the basic principle that the processing of personal data is forbidden, except if there is a legal basis or the data subject has given informed consent.

In this regard, Member States are permitted, subject to providing suitable safeguards, to allow processing of sensitive data or reasons of substantial public interest according to Article 8 (4) of Directive 95/46/EC. Public interest includes public health and scientific research as mentioned in recital (34) of the Directive. Although this exemption could be applicable in terms of MHA and especially with regard to the use cases, the solution of first choice is always to ask the data subject for consent. Therefore MyHealthAvatar used synthetic data to build the platform and processed personal data of volunteers only after having asked them for consent.

Regarding the exploitation phase of the platform, where real health data is processed, particular care will need to be taken to ensure maximal levels of security and control over processing operations,

[9] Opinion 4/2007, p. 15.

proportionate to the sensitivity of the data. A risk factor here is that cloud computing is deployed, as the optimum and most acceptable solution to integrate various user data from different sources, including genomics, systems biology and biomedical data mining. Although cloud computing provides several benefits such as lower costs and greater capacity and efficiency, as described in D11.1, it also raises legal and ethical issues. This is especially true in terms of public clouds because the platform as client has less overall control over operations, with the risk that it is not or only with difficulties possible to track the data access and processing. In this sense a solution will need to take proper account of the following: data control; data security, confidentiality and transfer. As described in section 3, MHA's hybrid cloud infrastructure mainly uses the FORTH private cloud which is located within the EU/EEA. So far as use will also be made of the Linode public cloud (based in the UK), this provider is subject in full measure to EU data protection and security requirements. (See also sections 5.3.7 on cloud security, and 5.4.3 on security in MHA cloud infrastructure.)

4.1.3 Clinical Data (HIS)

In general, the rules that are outlined above for sensitive data such as data concerning health also apply to clinical data that are processed by hospital information systems. A secure system is important to avoid unauthorized access. To ensure that the stored data are up to date, correct and complete, the transfer of clinical data from the hospital to the MHA platform through hospital information systems (HIS) have been investigated in D3.2 v2.0 and D11.3 and D11.4. D11.3 includes a patient data request to hospital in annex 4, p. 74 and a data transfer agreement between hospital and MHA in annex 5, p. 75 ff. For more detailed information regarding these two documents, please see D11.3, pp. 18 ff.

4.1.4 Activity Data

MHA offers the possibility to upload data by using Fitbit, Withings, and Moves. Since activity data is personal data, and as the citizen is at the center of the relevant data generation, MHA points out to the user in the paper-based consent forms that were used during the testing phase that the devices, apps and services are subject to their own third party privacy rules and that MHA has no control over data processing by such parties (please see appendix 3 of D11.1, p. 62).

For the exploitation stage, Terms and Conditions have been drafted. Clause IX. Third party services of the current General Terms and Conditions[10] (and clause XX of the extended terms and conditions[11])

[10] See annex 2, p.74 of D11.4
[11] See annex 4, p. 84 of D11.4

stipulates that MHA neither monitors nor investigates such websites and is not responsible for the content, functionality, or practices of Third Party services (clause IX. Third Party services of the current Terms and Conditions[12]).

4.2 Consent

As already pointed out, consent has always been both legally and ethically the keystone for the legal framework of MHA.. Moreover, this is in line with the core vision of MyHealthAvatar as a resource for citizen empowerment. Consequently, the MHA architecture developed and implemented a robust consent module. thatreflects all the elements necessary for valid consent. For the testing phase, the paper-based consent form has been used to reach volunteers (please see appendixes 2, 3, and 4 of D11.1, pp. 60 ff.). At a later stage of the project, when the details for an electronic consent have been solved, Terms and Conditions and a Privacy Policy (please see annexes 2 and 3 of D11.4, pp.74 ff.) have been drafted to inform interested persons about all circumstances, risks, and their rights. Please see D11.4, chapter 3.3 'Importance of consent', pp. 12 ff. for explanations regarding the different clauses.

[12] See annex 2, p. 74 of D11.4.

5 Security framework of the MyHealthAvatar Platform

5.1 Objectives

The objective of a secure system is to protect sensitive information from unauthorized access, manipulation, misuse, etc. In MyHealthAvatar the information to protect are, patients citizen personal data stored in MHA repositories, medical measurements, private patient data, medical history, activity data, medical images, model repositories data etc. The protection goals which define the requirements of a secure system are defined by the following objectives[13, 14].

Authentication: In information systems authenticity typically means the genuineness and credibility of data or an entity. The act of confirming the authenticity is called authentication. In a system where certain data or entities are required to be authentic there have to be authentication mechanisms. Authentication is based on providing some proof in the form of a credential, such as a username and password in the case of user authentication that allows the verifier to assess the genuineness of the property claimed by the credential.

Authorisation: In secure information systems access to sensitive information has to be restricted to privileged entities. Therefore, permission rights have to be specified and assigned to the privileged entities. This act is called authorization. If an entity has been granted the permission to access a specific information, the entity is said to be authorized for this access.

Confidentiality: A system ensures confidentiality, if an entity cannot learn information which is not intended for it. The assurance of this objective requires control mechanisms to ensure that information is accessible only to those who are authorized to have access to the information. There are numerous approaches to providing confidentiality, ranging from physical protection to cryptographic algorithms which render data unintelligible.

Integrity: Data integrity is preserved if it is not possible that an unauthorized entity can unnoticeably modify some information. For environments in which an unauthorized manipulation cannot be prevented a priori, mechanisms must be used to detect the manipulation a posteriori and thus, limit the damage incurred by further processing the manipulated data.

Accountability: Accountability is the ability to trace an action to a specific entity, such as a user, a process, or a device, and then hold them accountable or responsible for their actions. To ensure

[13] Claudia Eckert. *IT-Sicherheit – Konzepte, Verfahren, Protokolle*. Oldenbourg Verlag, 5th edition, 2007. ISBN 978-3-486-58270-3.

[14] Alfred J. Menezes, Paul C. van Oorschot, and Scott A. Vanstone. *Handbook of applied cryptography*. CRC Press series on discrete mathematics and its applications. CRC Press, 1997. ISBN 0-8493-8523-7.

accountability, mechanisms are required to monitor and record relevant actions performed within the system.

Non-repudiation: A system that ensures non-repudiation prevents an entity from denying previous commitments or actions. Performed actions can be retroactively traced to a unique entity which can then be held accountable for its actions. When disputes arise, due to an entity denying to have committed certain actions, some sort of proof is necessary to resolve the situation. That proof must verifiably attest that these actions were actually performed by the person concerned. Non-repudiation is particularly useful for accountability.

5.2 User data and trust

5.2.1 User data

In order to foster trust, within MyHealthAvatar we want to provide clear and objective information to users about the benefits and risks associated with the use of the system. The user should have as complete an understanding as possible of the consequences and eventual risks when using the system. However the risk that information are too detailed and might overburden the user must be taken into account as well, as this may result in users developing unfounded expectations that cannot and will not be met by the system. Furthermore, this may result in users blaming the system for not living up to their expectations. Therefore it is important that the user has the possibility to ask questions and get information repeatedly to find a balance between the risk of a loss of information and the lack of understanding.[15]. Care must also be given to the aspect *when* information should be presented. If users are presented once with some long hardly comprehensible text before they are allowed to execute a certain 'pressing' action, like acknowledging license rights before software installations, they are likely to ignore the text and pursue the action anyhow[16]. If such long texts cannot be avoided, they should be presented when the user is not engaged in routine tasks which the user expects to have finished in a few minutes. It could be helpful to *regularly remind the user of important topics by displaying brief, random pieces of information*, e.g., the user could be shown a hint when her device is busy and she cannot do anything else except watching the display and wait for the device to complete its task. Of course, such displayed information would not be complete but could help to 'educate' the user over time and also remind her of places where to look for more details.

5.2.2 Trust

MyHealthAvatar may have more than one trust model because the many use case scenarios, running in different environments with different requirements that influence their respective trust model. A

[15] Nikolaus, Forgó, Marian Arning, Tina Kruegel, Imme Petersen. Ethical and Legal Requirements for Transnational Genetic Research. C.H.Beck, p. 28. 2010.

[16] Rachna Dhamija and Lisa Dusseault. The Seven Flaws of Identity Management: Usability and Security Challenges. *IEEE Security & Privacy*, 6(2):24–29, March/April 2008.

trust model defines what can be expected from some party with respect to some item of interest. It makes explicit the set of hidden assumptions we rely on when giving guarantees to others. Trust models can be compared with respect to their relative strengths. MHA trust model also encompasses assumptions on the well-behaviour of certain entities in some regard and qualifies in which regard someone or something is trusted, similar to stating against whom anonymity or pseudonymity is directed. It is also the case that in both use cases we can have different usage scenarios, which may give rise to different trust models, too.

In this section, we present some assumptions with respect to security which are be satisfied in order to allow for the level of security we aim for. Some of the assumptions are necessary because they cannot be technically enforced or controlled by users — at least not with reasonable effort. Other assumptions are necessary because without them some of the scenarios could not be realised — at least not with a reasonable expectation of security and privacy.

Legal processing of personal data. We assume that MyHealthAvatar platform processes personal data only if the data subject has given her consent to the processing of her personal data or if there is other legal ground covering the processing.

No misrepresentation. We assume that the deployment server's operator does not deliberately misrepresent to data subjects the purpose or scope of some intended processing of personal data.

No permanent loss of secrets. We assume that reasonable steps have been taken to ensure that secret information, e.g., passwords or cryptographic keys, are regularly backed up such that the loss of the secrets, e.g., through theft or damaged media, is not permanent. Note that this does not imply that some data protected by the secret information should still be regarded as secure. The assumption merely ensures that important data does not become inaccessible for the user.

Wireless Key provisioning. In case devices need to be provisioned with secret keys over an 'air interface', we assume that steps have been taken such that attackers cannot overhear the keys without considerable effort.

Trusted personal devices (affects the portability of MyHealthAvatar to personal portable mobile smart phones and tablest). We assume that the user's device, e.g., the Android device in the inpatient scenario, acts as a trusted device, e.g., we assume that no malicious software is present on the device, e.g., key logger, viruses, Trojan horses, etc. We also assume that the device's installed software is up to date, i.e., has no vulnerability that an attacker could exploit to gain access to the device or its data. Furthermore, we assume that if the user passes the device on to another person, this person is also trusted by the device's owner.

The purpose of the assumptions above is to clearly state the limitations of the security architecture, i.e., what cannot be dealt with or what is beyond the scope of the security architecture for the purpose of the work within MHA project.

5.3 Technical Enforcement of Security

In practical systems these security objectives can be enforced by various mechanisms. In the following the most common technical means are introduced.

5.3.1 Authentication Procedures

For entities, such as users or machines, authentication technologies are subdivided into categories which are based on the knowledge of some specific information, such as a password or a PIN, personal possession, such as a key card or a software token like a digital certificate, or biometrical characteristics, such as fingerprints. To improve security and to utilize the advantages of different techniques, authentication technologies are often combined. Authenticity of data can be ensured by applying Message Authentication Codes (MAC)[17] typically based on block ciphers or cryptographic hash functions (HMAC)[18]. Another way of ensuring data authenticity is to use digital signatures [66]. Both approaches produce an authenticator which attests the authenticity of some data with respect to its originator. The difference between these techniques is that MACs and HMACs are symmetric, which means that all participants use the same secret to produce the authenticator, whereas, digital signatures are asymmetric, which means that each participant has its own secret to produce the authenticator. Both, entity and data authentication are relevant for MyHealthAvatar. Knowledge and possession-based mechanisms are more established than biometrics and thus, will be favoured for entity authentication. For data authentication several mechanisms will apply depending on the kind of data to be protected. [19]

5.3.2 Authorization & Access Control

Authorization covers the specification and the assignment of permissions to access specific resources or allowance to perform certain operations. Access control, on the other hand, is the enforcement of the permission assignment, i.e., the act of checking and granting or denying a requested action. In access control, objects are resources to protect, such as data but also processes, memory, etc. Subjects are entities, such as users or processes that want to access an object. Permissions are specific access rights on an object, such as reading or modifying a file. An Access Control Strategy determines how the permissions are granted. The concrete implementation of an access control strategy or a combination of strategies is called an Access Control Model. The following list introduces the three major strategies and some representative models:

[17] ISO/IEC 9797-1:1999. Information technology – Security techniques – Message Authentication Codes (MACs) – Part 1: Mechanisms using a block cipher, 1999.
[18] ISO/IEC 9797-2:2002. Information technology – Security techniques – Message Authentication Codes (MACs) – Part 2: Mechanisms using a dedicated hash-function, 2002.
[19] Chen CL1, Yang TT, Chiang ML, Shih TF., A privacy authentication scheme based on cloud for medical environment, J Med Syst. 2014 Nov;38(11):143. doi: 10.1007/s10916-014-0143-9. Epub 2014 Oct 15.

- In Discretionary Access Control (DAC) the authorisation originates from the owner or creator of an object and is passed on to other subjects. DAC is not suitable to enforce any system wide security policies because it is not possible to control the way an object owner manages its permissions. DAC is used, e.g., in UNIX-based systems.
- In Mandatory Access Control (MAC) permissions are managed centrally and ordinary users of the system cannot change the permissions. The disadvantage of MAC is that it can be difficult and expensive to manage large amounts of objects and users.
- In Role Based Access Control (RBAC) permissions are granted by the system not with focus on the subject itself but rather on the task or purpose a subject has. This means the subjects are categorised into roles according to their tasks and each role is then assigned to a corresponding set of permissions. RBAC is suitable for controlling systems with a large number and different types of users, like MyHealthAvatar, because organisational structures of institutions can often be mapped to roles in a straightforward manner. For example, users of a hospital's information system could be assigned to the roles *administrator*, *physician*, *nurse*, patient etc.
- OAuth 2.0 delegate access to third parties application, which reduces password sharing between users and third parties. It enable user to not only delegate access, but also revoke access. OAuth 2.0 provides specific authorization flows for various different situations, and supports web application, mobile application, desktop applications and other devices.

5.3.3 Confidentiality

Confidentiality can be achieved by using encryption mechanisms in order to transform information in such a way that an unauthorized person is not able to successfully interpret the data without knowing a relevant secret, for example the decryption key.

Encryption schemes can be categorised into two major classes:

- Symmetric cryptography refers to encryption schemes in which the same secret key is used for encryption and decryption. In the case of encrypted communication, the problem of a secure key exchange between sender and receiver arises. Symmetric cryptography mainly relates to the study of block ciphers, such as the Advanced Encryption Standard (AES)[20], and stream ciphers, such as RC4[21].
- Asymmetric cryptography, also called public key cryptography, on the other hand does not require a secure initial exchange of one or more secret keys. For asymmetric algorithms, a key

[20] U.S. National Institute of Standards and Technology NIST. Advanced Encryption Standard (AES). FIPS PUB 197, November 2001

[21] Alfred J. Menezes, Paul C. van Oorschot, and Scott A. Vanstone. *Handbook of applied cryptography*. CRC Press series on discrete mathematics and its applications. CRC Press, 1997. ISBN 0-8493-8523-7

pair — comprised of a private and a public key— is usually needed, where the private key is kept secret and the public key may be published. A message is encrypted by using the receiver's public key and the receiver uses her private key to decrypt the message. The crux of public key cryptography though is the distribution and management of public keys. Prominent representatives of public key cryptograhpy are the ElGamal encryption scheme[22] and the RSA encryption scheme[23].

Symmetric and asymmetric cryptography can also be combined to a hybrid cryptosystem, which take advantage of the fact that symmetric cryptography is more efficient than asymmetric encryption schemes. In a hybrid cryptosystem, the data is typically encrypted using a symmetric encryption scheme and the symmetric data encryption key is encrypted using an asymmetric scheme. This allows the receiver to first (asymmetrically) decrypt the data encryption key, using her private key, and subsequently decrypt the (symmetrically) encrypted data. The security of encryption schemes does not only depend on the secrecy of the secret keys but also on the lengths of the used keys. Using short keys enables adversaries to do an exhaustive search over the whole key space. With increasing technological advancements and growing computational power the cryptographic keys have to increase in length in order to resist these attacks. To ensure confidentiality in the long run, sufficient key lengths have to be chosen. An increase in key lengths on the other hand also increases the computational costs of encrypting and decrypting messages.

5.3.4 Integrity

Typically, data manipulation can be detected by using cryptographic hash functions. On the one hand, using hash functions enables the receiver of a message to detect if a message has been manipulated, given that she knows the hash value of the original data. On the other hand, an adversary is not able to bypass the hash function and thus, to manipulate a message in such a way that it cannot be detected by the receiver when she compares it to the original data's hash value. Hash function-based mechanisms to detect unauthorised data manipulation are often called Manipulation Detection Code (MDC), Message Digest, Digital Fingerprint, cryptographic checksum, or Message Integrity Check (MIC). These mechanisms cannot prevent unauthorised data manipulation but merely make it retroactively detectable. Examples of established cryptographic hash functions are RIPEMD160[24], SHA-1, and SHA-256 [25]. Integrity can also be ensured by applying Message Authentication Codes (MAC) or digital signatures.

---

[22] Taher El Gamal. A public key cryptosystem and a signature scheme based on discrete logarithms. In *Proceedings of CRYPTO 84 on Advances in cryptology*, pages 10–18, New York, NY, USA, 1985. Springer-Verlag New York, Inc

[23] Ron L. Rivest, Adi Shamir, and Leonard M. Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM*, 21(2), February 1978

[24] Hans Dobbertin, Antoon Bosselaers, and Bart Preneel. RIPEMD-160: A Strengthened Version of RIPEMD, 1996

[25] U.S. National Institute of Standards and Technology NIST. Secure Hash Signature Standard (SHS). FIPS PUB 180-2, August 2002.

5.3.5 Accountability

To ensure accountability it is necessary to monitor and record relevant actions executed within a system, e.g., when did they happen and who executed them. A lower level of accountability can typically be provided by auditing techniques, such as audit logs, which record the time and date, the performed action and who performed that action, etc. In MyHealthAvatar, accountability is a property that is especially relevant for audits as their purpose is to check whether past actions have been conducted in accordance with a given policy, e.g., laws or other regulations. Therefore, audit logs and other mechanisms are applied in full detail and the related technical implementation is presented in 5.6 section of this document.

5.3.6 Non-repudiation

Non-repudiation is a property typically achieved with the help of cryptographic methods, which prevents an individual or entity from denying having performed a particular action. Several mechanisms for non-repudiation are specified[26, 27, 28]. These can be based on symmetric techniques, such as Message Authentication Codes (MAC), as well as asymmetric cryptographic techniques, such as digital signatures. These techniques include non-repudiation certificates, tokens, and protocols and rely on additional Trusted Third Parties (TTPs), such as secure time-stamping services and Certification Authorities (CAs) of Public Key Infrastructures (PKIs).

5.3.7 Cloud Security

A private cloud allows to control the perimeter and ensure highly secured data not to be transferred out of the premise. Apart from defining the security measures a primary concern is to make sure that the administration of the private cloud will be making all required fixes, updates, and upgrades to sustain security today and for the future. Public clouds, on the other hand, provide a secure data protection layer on software and physical level. Private cloud typically resides in one premise and requires expensive proposition in order to ensure the constant service availability. Public cloud typically operates in multiple data centres at multiple geo locations, which provides high level service availability especially in the scenarios of disaster recovery. There are important differences between each model in terms of merged features, complexity and security. Service availability and disaster recovery is another key consideration.

[26] ISO/IEC 13888-1:2009. Information technology – Security techniques – Non-repudiation – Part 1: General, 2009
[27] ISO/IEC 13888-2:2010. Information technology – Security techniques – Non-repudiation – Part 2: Mechanisms using symmetric techniques, 2010
[28] ISO/IEC 13888-3:2009. Information technology – Security techniques – Non-repudiation – Part 3: Mechanisms using asymmetric techniques, 2009

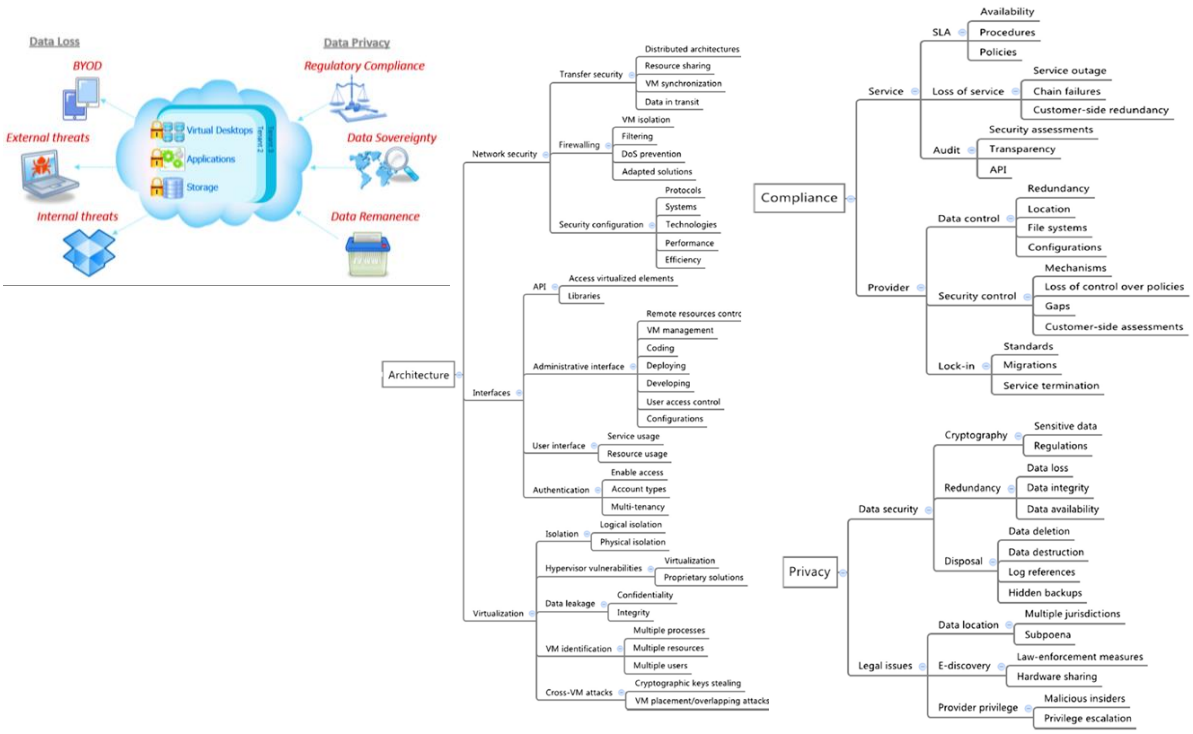


Figure 5-1: Securing data cloud infrastructure: A taxonomy

Cloud service providers can provide the basic security architecture; consumers are responsible for implementing and managing the provided security features. In deliverable D3.2 Architecture Description version 2.0 we analysed security concerns, risks and threats in respect of a secure cloud computing infrastructure[29]. From our analysis we conclude that there is a strong need and necessity to define a strong set of policies and protocols necessary to ensure secure data transmission, storage and access to your supporting services.

[29] Gonzalez, Nelson Mimura Icon ; Miers, Charles Christian Icon ; Redígolo, Fernando ; Simplicio Junior, Marcos Antonio Icon ; Carvalho, Tereza Cristina Melo de Brito Icon ; Näslund, Mats ; Pourzandi, Makan, A quantitative analysis of current security concerns and solutions for cloud computing, Journal of Cloud Computing: Advances, Systems and Applications, Heidelberg, v.1, 2012

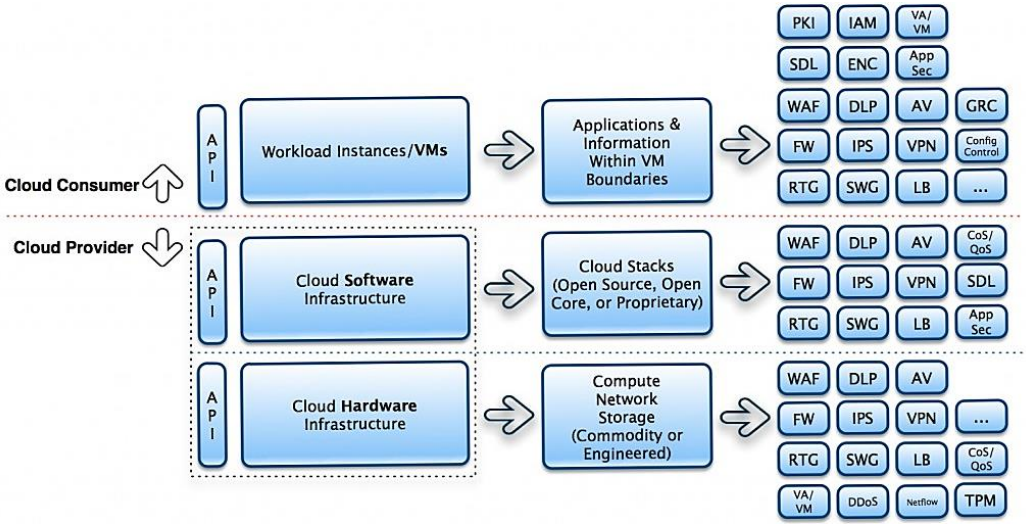


Figure 5-2: Cloud Security elements (examples)[30]

Today given that the development of cloud computing services is speeding up the rate in which the organizations outsource their computational services or sell their idle computational resources. One of the most important aspect when organizations or private companies try to exploit the possibility of deploying their services in to the cloud is security: while some cloud computing security issues are inherited from the solutions adopted to create such services, many new security questions that are particular to these solutions also arise, including those related to how the services are organized and which kind of service/data can be placed in the cloud. MHA is deployed into two different setting (private and public cloud) and is trying to provide security both for providing and consuming MHA offered platform services through related security elements.

6 Guidelines for MHA platform security measures

6.1 Web security access

6.1.1 Authentication and Authorization for MHA

The trust principles of security, availability and processing integrity of MHA system is guided by Service Organization Control (SOC 2), which allows MHA system processing to be authorized and complete.

[30] http://www.ncbi.nlm.nih.gov/pubmed/25732083

ISO/IEC 27002:2013 code of practices for comprehensive information security control and risk management is considered during the design and implement of MHA security system design.

MHA's security of password infrastructure could be further augmented by the Universal Second Factor (U2F) protocol, which adds a strong second factor to user login. The commonly used U2F application is Google Authenticator of Android platform. The "meta protocol" OAuth 2 provides a very useful foundation for other protocols (e.g. OpenID Connect, NAPS and UMA). OAuth 2 forms the security foundation of for MHA API, which delegate access to third party applications. Detail of the API security can be found from deliverable 3.6. Various cutting edge security protocols and practises are evaluated and listed in following sections with their pros and cons.

6.1.2 Transport Layer Security with HTTPS

When two software entities communicate over the network the data integrity and privacy concerns are highly important so that no third party can intercept or eavesdrop the transmitted messages. The Hypertext Transfer Protocol (HTTP) is used ubiquitously nowadays to support network communications but since it lacks any such security mechanisms HTTPS is widely adopted. HTTPS simply means that the HTTP application protocol is tunnelled through the TLS protocol[31]. TLS can be run on top of any reliable transport protocol, e.g., TCP.

Authentication in TLS requires the relying party (RP), i.e., the verifier, to verify the validity and trustworthiness of some end entity's (EE) certificate and this certificate's issuer. A certificate's issues is also called a *certificate authority* (CA). Verifying the validity of a certificate includes, at least, checking some information presented by the sender that should prove that she really is the holder of the certificate, that the certificate is not expired, and that the certificate was issued by a CA that the RP deems trustworthy. Such a trusted CA, more specifically its certificate, is usually called a *trust anchor*. It is also possible that the issuer of the EE's certificate is not some CA directly trusted by the RP but by at least one of its trust anchors. This leads to some form of 'indirect trust' because the RP trusts a certain CA to only issue certificates to trustworthy entities and, similarly, this CA trusts another CA to do the same, and so forth. In practice, this leads to a so called *certificate path* which runs from the EE's certificate over some intermediate CAs' certificates to a trust anchor of the RP. Note that a certificate path trusted by the RP *must* be terminated by a trust anchor. This also warrants the term "trust anchor" because no more checking is done beyond the point where a trust anchor is encountered on a certificate path — beyond this point only trust remains. However, if no trusted path can be built, the RP would reject the EE's certificate. Building a certificate path is what is done by default by major browsers when they encounter a URL with HTTPS in the URL's protocol part. In this case, the EE's certificate is the certificate of the site being addressed by the URL, the RP is the user/her browser, and the trust anchors are all certificates which are included in the browser's certificate store.

[31] Tim Dierks and Eric Rescorla. The Transport Layer Security (TLS) Protocol — Version 1.2. RFC 5246, August 2008

If the site requires mutual authentication the roles are reversed, i.e., the site becomes the RP, the user becomes the EE, and the trust anchors are all certificates configured in the certificate store of the RP's Web server.

Validating a certificate usually involves a revocation check, too. Any certificate, not only EE certificates but also CA certificates, can be revoked, e.g., because some guarantee given by the certificate does no longer hold. For instance, if every employee of a hospital receives a certificate when starting to work for the hospital, the certificate would normally be revoked when the employee quits, as the guarantee that the holder of the certificate is employed by the hospital no longer holds. However, a digital certificate cannot be made unusable like a credit card that is being cut when it becomes invalid. Therefore, the certificate's serial number is blacklisted by its issuing CA when the certificate is revoked. Such a blacklist is called certificate revocation list (CRL) and typically made available by the issuing CA. Now, if an RP wants to validate an EE's certificate, it will perform the certificate path building and check that the serial number of every certificate on the path has not been put on its issuing CA's CRL. Another method for checking the revocation status of a certificate are online status checks using the Online Certificate Status Protocol (OCSP). Roughly speaking, the difference between OCSP and CRL checks is that CRL checking can be done offline once the current CRL has been retrieved while OCSP requires the RP to be online to inquire the OCSP service about the status of the submitted serial number of the certificate in doubt.

Channel Security, Given that the restrictions and requirements from the previous paragraphs are satisfied, an HTTP message / payload sent over HTTPS is transmitted confidentially and the message's authenticity can also be checked by the recipient. However, TLS is a connection-oriented protocol, which means that the security guarantees are really attached to the communication channel and not to the HTTP message / payload. That is, the HTTP message / payload's authenticity can be *derived* from the authenticity of the connection but cannot be inferred from the HTTP message / payload alone. As a consequence, the security guarantees given by the communication channel are lost as soon as the message exits the channel, i.e., when it is received by the Web service. Furthermore, the security properties derived from the channel —confidentiality and authenticity— cannot be verified by a third party after the message was received. Thus, if accountability with non-repudiation is desired, TLS is not an option to achieve this. In addition, if the content of the received message was meant only for the eyes of a specific person, e.g., a patient's attending doctor, then TLS will not help either because the message will be available in clear for the receiving end, i.e., some Web server. Thus, *end-to-end security* —from the sending person to the receiving person— is not possible with TLS, either. For MyHealthAvatar, all of the above is required if HTTPS/TLS is used to secure the communication between the platform and a third party service.

6.2 Connection with third party data source for activity data retrieval and sensor

6.2.1 Mobile Devices

Apart from stationary systems, such as personal computers and servers, MyHealthAvatar aims for an integration of mobile devices, external mobile applications and web access applications. Mobile devices, such as smart phones or wireless sensors, have the major benefit of providing the users with flexibility and mobility. Mobile devices are small in size, can easily be carried with, and come with their own power supply. With today's technical progress mobile devices, such as mobile phones, have become a constant companion technically almost equal with personal computers. Thus, it seems natural for MHA to make use of existing devices and also let the patients benefit from utilizing the flexibility of mobile devices.

In contrast to stationary systems, mobile devices are exposed to several constraints having an impact on security. First of all, mobile devices have less computational power which can be an issue when considering expensive computations of cryptographic operations. Another critical constraint is the limited power of the device's battery. Therefore, it has to be taken into account that a mobile device runs out of power in critical situations. Even though the use of mobile devices provides local independence it has to be considered that some areas, especially rural ones, provide a limited bandwidth or even lack network coverage and hence, render the device useless for some applications.

Apart from technical limitations mobile devices are more strongly exposed to the threat of physical access than stationary systems, e.g. at home. Also, mobile devices are easily lost and far more subject to theft than stationary systems. Thus, it should not be overlooked that a mobile device might fall in the hands of an adversary far more easily than other platform components. Physical access, in turn, opens the door for additional attacks, such as getting direct access to the device's file system, memory card, or system memory.

Today, many mobile consumer devices, such as mobile phones or set-top boxes, enable the user to utilize online offers and services by connecting to the Internet. However, the Internet, can also be a loop hole for further threats to the device. Modern mobile operating systems are equal to their stationary counterparts in complexity and performance. Thus, mobile devices are also subject to remote attacks, such as session hijacking or the installation and execution of malicious code.

Device Classification. Mobile devices, such as laptops, which are with regard to technical capabilities not distinguishable from stationary systems can be neglected when considering particular security consideration which would typically arise for mobile devices with technical limitations. For these devices the same security considerations as for stationary systems apply.

For mobile devices with strong technical constraints, such as wireless sensors (e.g., glucose meter, SpO_2 meter, HR meter, BP meter, etc.), special assumptions concerning security have to be made. These devices are typically very limited in computational power or networking capabilities so that

some security mechanisms cannot be implemented in practice. Due to the technical constraints, however, the threat of adversary installing and executing malicious code on these devices is unlikely.

Device Security. When considering security it has to be taken into account that some mobile platforms are innately equipped with security measures that protect from attacks which require physical access. For example, some mobile operating systems provide secure storage by encrypting the device's file system. Thus, some threats can be neglected depending on the underlying platform, i.e., the mobile device's operating system. Considering the spectrum of mobile platforms, the provision of complete device security is out of the scope of MyHealthAvatar. Therefore, the security of a mobile device's operating environment is rather assumed than enforced with in the terms of MyHealthAvatar project.

6.2.2 Protocols and Standards

In this section, we present some details on protocols and related security aspects that are envisaged to be used within the scope of MyHealthAvatar mobile devices communication and medical device communication. For more details the reader is referred to the respective standard's publication.

6.2.2.1 Continua

The Continua Health Alliance is a non-profit, open industry alliance of several healthcare and technology companies. Continua's mission is to establish a system of interoperable personal solutions that offer a more independent and improved management of health and wellness. One of Continua's main objectives is develop design guidelines that will enable vendors to attend a product certification program in order to build interoperable sensors, home networks, eHealth platforms, and health and wellness services. The Continua Health Alliance's Design Guidelines[32] are based on existing standards and specifications that Continua selected for ensuring end-to-end interoperability of devices. It also contains additional guidelines for interoperability that reduce or extend options in the underlying standards or specifications. By adhering to these guidelines, products of different manufacturers can exchange health information seamlessly. For MyHelathAvatar, the major benefit of complying with the Continua Design Guidelines is the end-to-end interoperability based on established industry standards. Using Continua certified devices allows for seamless communication between heterogeneous devices over heterogeneous networks. Interoperability also helps to stimulate technological improvements and enables users to use devices of their choice. Healthcare providers are able to easily switch and integrate other device manufacturers, reduce costs and, finally, optimise the patient's treatment. The Continua Design Guidelines aim at the interoperability of devices with a

[32] Continua Design Guidelines - Version 2015, http://www.continuaalliance.org/products/design-guidelines

focus on the transport and the data level. The goal is to empower users to control how their health information is shared and used in a personal electronic health eco-system.

For MyHealthAvatar, additional concepts regarding privacy, trust, and security play an important role. As mentioned, for adoption of personal eHealth systems, trust, security and privacy are very important. The same holds for compliance to legislation like EU Directive 95/46 and HIPAA[33]. Continua acknowledges the importance of these issues amongst others through its E2E Security Task Force. Initial security and privacy issues have been addressed in Continua version 1 guidelines for the PAN and HRN interfaces. Continua version 1.5 guidelines added security features for the WAN and LAN interfaces with e.g. TLS for secure communication and SAML 2.0 tokens for authentication of AHD users. With personal eHealth systems emerging today people are able to participate in their own care supported by an open distributed system with health services. This poses new end-to-end security and privacy challenges. In Koster et. al.[34] new end-to-end security requirements and presented with a design for consent management in the context of the Continua Health Alliance architecture.

6.2.2.2 ZigBee

ZigBee is a very popular wireless communication standard for low-cost, low-power devices secure communication especially among medical devices[35]. ZigBee makes use of the security mechanisms defined by the IEEE 802.15.4 standard[36]. It also complements by providing security management functions for different key types. ZigBee supports keys for the following functions: a) key establishment (also known as key agreement); b) key transport (for master keys, network keys, link keys); c) frame protection (regarding integrity, authenticity, and confidentiality) and d) device management (configuration). In general, ZigBee distinguishes two key types, *network keys* and *link keys*. Both key types are 128 bit long and make use of the Advanced Encryption Standard (AES)[37] operated in the CCM mode (Counter with CBC-MAC[38]) which provides *authenticated encryption*. The former key type is used for broadcast communication within a device network and the latter one is

[33] http://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=CELEX:31995L0046:en:HTML

[34] Koster P, Asim M, Petkovic M., End-to-end security for personal telehealth, Stud Health Technol Inform. 2011;169:621-5.

[35] Frehill P, Chambers D, Rotariu C., Using Zigbee to integrate medical devices, Conf Proc IEEE Eng Med Biol Soc. 2007;2007:6718-21.

[36] IEEE Computer Society. IEEE Standard for Information technology – Telecommunications and information exchange between systems – Local and metropolitan area networks – Specific requirements, Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (WPANs). IEEE Std 802.15.4–2006, September 2006

[37] U.S. National Institute of Standards and Technology NIST. Advanced Encryption Standard (AES). FIPS PUB 197, November 2001

[38] Doug Whiting, Russ Housley, and Niels Ferguson. Counter with CBC-MAC (CCM). RFC 3610, September 2003

used for unicast communication between two (paired) devices. Link keys are acquired by a device or through pre-installation. A network key is shared among all devices from the same network and used to secure communication (broadcast or point-to-point) between these devices. Network keys come in two flavours, *standard* and *high-security*. The difference between the two is only how the network key is being distributed and, possibly, how frame counters for messages are initialised.

6.2.2.3 Bluetooth

Bluetooth is a well-known open standard for short range radio frequency communication, primarily used to establish wireless personal area networks (WPAN). Bluetooth is based on a so called master-slave architecture allowing point-to-point communication from the master to the slave. In Bluetooth, the level of security highly depends on the version of the standard implemented by the device because early versions use procedures which are not considered secure[39], e.g., downgrading of encryption key sizes. Bluetooth supports keys for the following purposes: device authentication and message encryption. The authentication and encryption procedures employ symmetric algorithms called E_1 and E_0, respectively, which are specific to Bluetooth and have also been found to be weak[40]. Note that Bluetooth does not offer frame or message authenticity. It only employs CRC checksums to guard against transmission errors in noisy channels. However, correct checksums can be computed by anyone and thus, messages can be injected without the recipient being able to detect this. Also note that encryption is not a replacement for proper message authentication[41].

6.3 Security in MHA Cloud infrastructure

6.3.1 Private Cloud (OpenStack)

An overview of security measure for MyHealthAvatar is summarized below (Details on the deployment of OpenStack can be found below):

Physical security

- Machine room in a restricted access area
- Smartcard & PIN access control
- Heavy duty air conditioning system
- UPS (uninterruptible power supply)
- Intrusion detection alarm
- Temperature fluctuation alarm

[39] Karen Scarfone and John Padgette. Guide to Bluetooth Security. NIST Special Publication 800121, September 2008. Recommendations of the National Institute of Standards and Technology

[40] Yi Lu, Willi Meier, and Serge Vaudenay. The Conditional Correlation Attack: A Practical Attack on Bluetooth Encryption. In Crypto 2005, number 3621 in LNCS. Springer Verlag, 2005.

[41] Michael McIntosh, Martin Gudgin, K. Scott Morrison, and Abbie Barbir. Basic Security Profile Version 1.0. Web Services Interoperability Institution (WS-I), March 2007

- Fire alarm
- Automatic CO2 fire extinguisher
- Automatic power generator in case of power shortage
- Administrative stuff available on nearby office

Network security

- Firewall
- IP based access control
- Certificate-based login (optional)

Data security

- RAID controllers (hardware)
- Automatic replication of data (software -Openstack)
- Tape backups

Figure 6-1: MHA cloud security control panel (OpenStack)

User provisioning and identity is the process of registering a new user with a given system and user de-provisioning is the process of removing a user from the system. OpenStack object storage "swift" offers significant automation of user data management tasks by using authentication/authorization systems referred to as "tempAuth" and "swAuth". The difference between "tempAuth" and "swAuth" lies in the backend storage of user data. TempAuth uses a configuration file in which user data are saved as plain text. On the other hand, swAuth is meant to be a "scalable authentication and authorization system that uses swift itself in a backing store. The characteristics of user management are based on OpenStack object storage "swift". The following characteristics are present:

- Users are not given administrative power over any other users
- Provider Admins have admin agreements with all accounts but cannot add other provider admins
- Super admins are powerful users who are able to perform all user management procedures, including adding provider admins

TempAuth and swAuth often use a username and password for the authentication process. When authentication is successfully performed, the user receives a token that will identify him to the system for a period of time. The provided token has a configurable expiration time, the default value of which is set to 4-6 h. All cloud security documents must, allow authentication by accepting confirmations in

SAML format; however, this feature is not yet available in OpenStack. Because all OpenStack projects use a password and username system to authenticate users, password strength requirements should receive greater scrutiny.

Authentication tokens play similar roles as identifiers for web applications. An API, such as an OpenStack service, is used to authenticate a user. Successful authentication generates a token that is used to authorize service requests. The password and username are given as input to the API interface. When authentication succeeds, the resulting feedback includes an authentication token and service catalogue. Note that tokens remain valid for 12 h. Issued tokens become invalid in two situations: If the token is expired or if the token has been cancelled. It is important that the authentication be executed over a secure channel, such as Transport Layer Security (TLS); otherwise, an attacker could obtain a user token by executing a man-in-the-middle-attack and remove the user who received the token from the authentication system. Lastly most cloud providers do not encrypt data before saving it to a cluster. In fact, OpenStack does not provide any data encryption at all; thus, users would need to encrypt their data before uploading it and manage their encryption keys themselves. Given the difficult of track security issues in cloud computing environments the following table provides a summary of security issues, which are divided into five categories and listed with their implications.

Security issues	Implications of Security
Trust	This is interrelated to the designated deployment modal because the control of the data and applications is directly supervised by the strict control of the owner
Availability	The capacity of a system to operate upon the demands of a certified entity. This notion implies that the system should be able to function even in the presence of authorities that disobey the regulations. Furthermore, the system must also maintain the capacity to operate even in the existence of a security breach
Integrity	Resources can only be reformed by approved individuals and through official procedures.
Software	The diverse resources include data, software and hardware
● Data (Authentication, Authorization and Access control AAA) Confidentiality	Data in cloud computing are more vulnerable because of the increase in the number
● Software	of individuals, devices and applications that use cloud computing which will in turn
● Data	increase the number of access points. Consequently, authorized individuals and systems are the only entities that are allowed to access the protected data
Privacy	An individual's need to govern the entree to his/her personal information

6.3.1.1 OpenStack Security analysis

Since it is outside the purpose of this document a full and comprehensive OpenStack security analysis, we consider the following basic, but critical aspects of security:

- Identity management
- Authorization and Access control
- Data Placement in object storage

- Data encryption

Identity management

The identity service in OpenStack performs user management and provides a list of the available services and their API endpoints.

Internally, OpenStack uses the concepts of users, tenants, domains, regions, endpoints, services and roles. With these concepts, OpenStack virtually isolates system resources, group together users and resources and controls access to services.

6.3.1.1.1

Apart from users and services which are essentially self-explanatory concepts, we need to provide a brief introduction for the rest:

- Tenants: is a configurable virtual container which isolates resources and identity objects. It usually represent users and their relevant resources
- Endpoint: the network address of an OpenStack service in URL format
- Domain: can represent an organization, a company or an individual and provide administrative boundaries to the Keystone service
- Region: represents physically or virtually isolated resources inside a domain
- Role: a set of privileges which apply to users

Most commonly, a mapping is performed between tenants, users, roles and permissions. Each available resource must be first allocated to tenants. Tenant users can access this resource if they hold the corresponding role and have the appropriate permissions.

OpenStack uses a pluggable authentication system, in the sense that it delegates control of identification to external providers, most commonly the OpenStack Keystone.

Keystone performs user and their relevant permissions tracking, provides a catalog of available services and API endpoints and issues tokens for the API calls.

In the figure below, we visualize the Keystone operation which in brief, is a series of identification

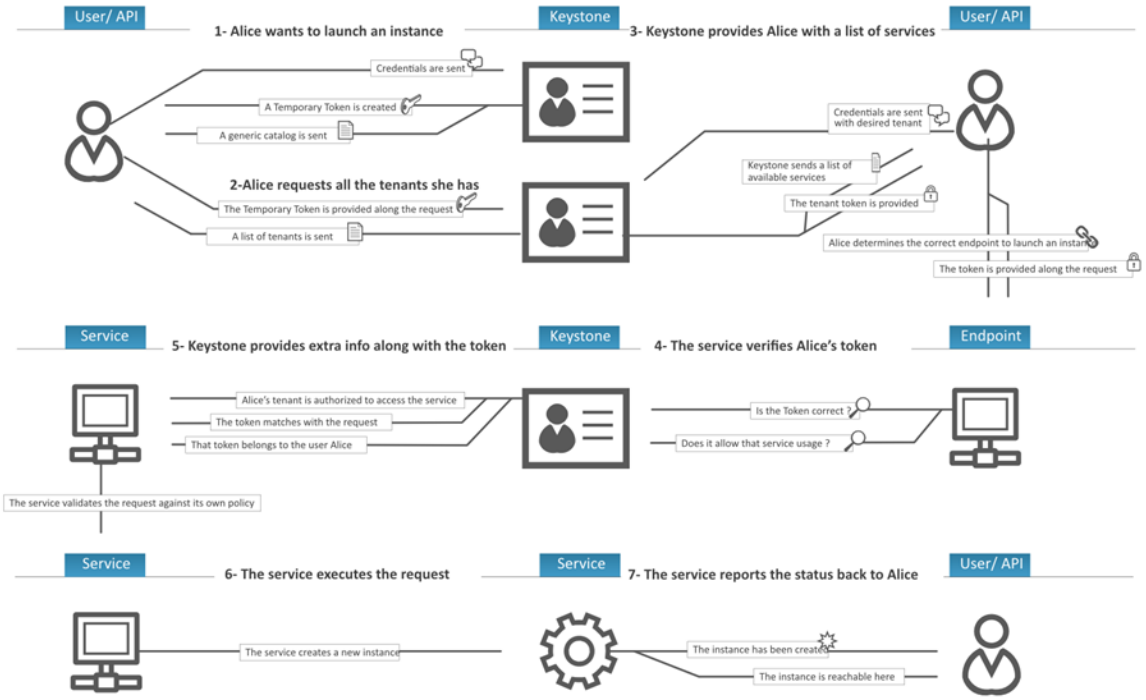


Figure 6: Keystone identity Manager(Picture taken from: http://docs.OpenStack.org/admin-guide-cloud/content/keystone-concepts.html)

retrieval based on valid credentials, a token issue and authorization of OpenStack services after a successful authentication check.

Keystone provides functionality and integration with several back ends, like Pluggable Authentication Module (PAM) and Lightweight Directory Access Protocol (LDAP). Although the default storage option for identification information is SQL databases, like SQlite3 or MySQL, the architecture of keystone enables proxying external identification systems like oAuth[42], (SAML)[43] or OpenID[44].

6.3.1.1.2

Authorization and Access Control

The design of the authorization and access control model of OpenStack is heavily based on tokens. Resources can be successfully accessed with a valid token. OpenStack deploys a number of mechanisms to handle tokens, like an expiration system, revocation procedures and PKI store.

[42] "oAuth," [Online]. Available: http://oauth.net/. [Accessed 9 9 2014].
[43] "OASIS SAML Wiki," OASIS Security Services (SAML) Technical Committee, [Online]. Available: https://wiki.oasis-open.org/security/FrontPage. [Accessed 8 9 2014].
[44] O. Foundation, "Specifications," [Online]. Available: http://openid.net/developers/specs/. [Accessed 8 9 2014].

Internally, OpenStack uses the concepts of users, tenants, domains, regions, endpoints, services and roles. With these concepts, OpenStack virtually isolates system resources, groups together users and resources and controls access to services.

In principle, any WEB service exposed to the Internet, may be a subject of malicious attacks and possibly expose weaknesses, which may prove fatal to the underlying infrastructure. Several mechanisms have been deployed in OpenStack in order to minimize the threats, like sanitization of input in API calls, secure communications of the service endpoints over SSL connections, a PKI infrastructure and more.

To enhance security, the Identity service in OpenStack operates along with a kind of policy enforcement, since services in OpenStack acquire policy rules associated with the resources this service provides.

Most OpenStack projects implement Role Based Access Control (RBAC) and Role Based Security[45], which was standardized by ANSI (ANSI INCITS 359-2004). As an example the Identity API v3 of OpenStack, uses a JSON encoded policy file which contains declarations of the form[46]:

API_NAME: RULE_STATEMENT or MATCH_STATEMENT

Where:

- RULE_STATEMENT: a RULE_STATEMENT or a MATCH_STATEMENT
- MATCH_STATEMENT: is a set of identifiers that must match between the token provided by the caller of the API and the parameters or target entities of the API call in question. A MATCH_STATEMENT is of the form:

ATTRIB_FROM_TOKEN: CONSTANT or ATTRIB_RELATED_TO_API_CALL

In order for OpenStack to comply with security standards, contributors from IBM have provided auditing functionality to OpenStack with the MTF Cloud Auditing Data Federation (CADF) standard[47].

[45] N. C. S. R. C. (CSRC), "Role Based Access Control (RBAC) and Role Based Security," [Online]. Available: http://csrc.nist.gov/groups/SNS/rbac/. [Accessed 9 9 2014].
[46] Openstack, " Identity API protection with role-based access control (RBAC)," [Online]. Available: http://docs.openstack.org/admin-guide-cloud/content/identity-service-api-protection-with-role-based-access-control.html. [Accessed 10 9 2014].
[47] C. M. Initiative. [Online]. Available: http://dmtf.org/standards/cloud. [Accessed 9 9 2014].

Data Placement in object storage

As mentioned earlier in this document, the path to an object entity is constructed as:

account/container/object

6.3.1.1.3 And referenced as a URL of the form:

<protocol>://<network address>:<port>/<auth_version>/account/container/object

For each path, OpenStack Swift generates the MD5 hash of it and places it accordingly to the Swift ring. The MD5 hashing algorithm is proved not to be collision resistant, and in fact, MD5 collision resistance can be broken in in 2^{18} time[48]. This behavior can lead to data corruption, compromise the integrity of the storage, and of course, be a subject of malicious attacks, even from validated users. This can break legal agreements and compromise the trust between the cloud provider and users.

The way to overcome this is to use a hash_path_prefix, and therefore, the actual path inside OpenStack is hashed as:

hash(path) = md5(path + hash_path_prefix)

The prefix is randomly generated and it should be unique for each Swift cluster and not known outside OpenStack.

In a normal collision attack, for a given pair of P_1 and P_2 there are S_1 and S_2 suffixes so that:

$md5(P_1 \| S_1) = md5(P_2 \| S_2)$

Given P1=P2 and assume that they represent an entity prefix (account/container), one can generate the same hashes for different objects (S_1 and S_2). If a prefix is inserted to each path, for each fixed P and S, for a successful collision attack values M_1 and M_2 must be generated in order to provide equal hashes:

$md5(P \| M_1 \| S) = md5(P \| M_2 \| S)$

If hash_path_prefix is not known, it is not feasible for such an attack to succeed[49].

[48] T. Xie, L. Fanbao and F. Dengguo, "Fast Collision Attack on MD5," Cryptology ePrint Archive, 2013.

[49] M. Stevens, A. K. Lenstra and B. De Weger, "Chosen-prefix collisions for MD5 and applications," *Int. J. Applied Cryptography,* vol. 2, no. 4, 2012.

Data encryption

6.3.1.1.4

Data are not encrypted when stored in Swift. Although data encryption is a default behavior in Swift, it is part of its functionality, for both storing and transferring data over the internet. This concept might prevent unauthorized access to the actual data, so that, even if cloud's security is compromised, a potential attacker will not be able to investigate, read or process user files. Even if a malicious action is not the case, storing unencrypted data on storage devices are at the disposal of the administrator of the storage. This can break legal and trust relationships and compromise the service.

In the case of biomedical data, data encryption can increase the level of data protection and prevent unauthorized access to sensitive information. Although server-side encryption seems a possibility, in cases where datasets of several GBs are involved, it may not be an optimal solution.

Data encryption can be applied to several levels:

- Swift object storage entities
- Data over the network
- Volumes residing on block storage

Encrypted network communication for both API calls and data transfer can be initiated with IPsec and tunneling techniques[50]. Volume encryption can be achieved by selecting the appropriate back end solution, like Logical Volume Manager (LVM)[51,52] encrypted volumes[53].

Object storage does not implement server side encryption in its upstream version, but modified and fully operational, open source Swift branches exist, which provide this functionality[54,55].

[50] S. Frankel, K. Kent, R. Lewkowski, A. D. Orebaugh, R. W. Ritchey and S. R. Sharma, "Guide to IPsec VPNs," NIST, 2005.
[51] "Logical volume management," [Online]. Available: http://en.wikipedia.org/wiki/Logical_volume_management. [Accessed 29 10 2014].
[52] H. Mauelshagen, "LVM 1.0.8 README," 17 11 2003. [Online]. Available: http://ftp.gwdg.de/pub/linux/misc/lvm/1.0/README. [Accessed 29 10 2014].
[53] C. Fruhwirth, "LUKS On-Disk Format Specification Version 1.1," 8 12 2008. [Online]. Available: http://cryptsetup.googlecode.com/svn-history/r42/wiki/LUKS-standard/on-disk-format.pdf. [Accessed 29 10 2014].
[54] "Openstack," [Online]. Available: https://wiki.openstack.org/wiki/ObjectEncryption. [Accessed 10 9 2014].
[55] Mirantis. [Online]. Available: https://www.mirantis.com/blog/on-disk-encryption-prototype-for-openstack-swift/. [Accessed 10 9 2014].

6.3.1.2 Common attacks

Network based attacks

When using insecure communication channels, a malicious attacker can intercept data, such as usernames, passwords or service tokens and use them in order to validate itself against the cloud. By default, OpenStack does not enforce SSL/TLS over public service endpoint.

6.3.1.2.1 One very common attack of this type, is the Man-In-The-Middle (MiTM) attack[56]. For this attack to succeed, an attacker must intercept traffic between two systems. In the OpenStack case, the two endpoints can be a public service endpoint and an external client application. Upon a successful attack the traffic between the two endpoints will be routed through the attacker machine and thus the attacker will be able to process all information exchanged between the client and the OpenStack. As a result, the integrity of the whole infrastructure may be compromised since the attacker will be able to gain unauthorized access to the cloud resources, depending of course on the access level the legitimate user/victim has.

One profound solution is the usage of encrypted SSL/TLS sessions between the users and the OpenStack services. Although this technique is likely to provide a solution to this problem, it is usually not enough. The MiTM attack can be easily applied on SSL connections just by creating two independent SSL sessions, one with the client and on with the server and as long as there is no validation mechanism for the digital certificates used in these SSL sessions, an MiTM attack can again take place[57].

Moreover, several vulnerabilities have been reported in the core elements of OpenStack, like Cinder. OpenStack Security Notes[58], OSSN/OSSN-0019, indicated a MiTM attack through the impersonation of a legitimate storage host[59]. Although this issue was addressed early, it indicates that a large and complex deployment like a cloud platform must be applied with caution and high alertness level.

Additional threats emerge from the RESTful nature of the OpenStack API. There are no native security mechanisms inside the REST implementation and without the proper input sanitization, given the

[56] S. Institute, "Global Information Assurance Certification Paper; Man-In-the-Middle Attack Brief," 200-2002.
[57]"OSSA-2014-005," 17 02 2014. [Online]. Available: http://openstack-security-advisories.readthedocs.org/en/latest/advisories/OSSA-2014-005.html. [Accessed 27 10 2014].
[58] "OpenStack Security Notes," [Online]. Available: https://wiki.openstack.org/wiki/OSSN. [Accessed 27 10 2014].
[59] "OpenStack Security Notes," [Online]. Available: https://wiki.openstack.org/wiki/OSSN/OSSN-0019. [Accessed 27 10 2014].

proper payload, an attacker can use injection techniques and compromise communications. In this respect, OpenStack implements input validation across all APIs.

Virtualization layer attacks

As any cloud platforms, Openstack implements virtualization in order to accommodate hardware and network resources. One leading security issue in virtualized environments, is hardware infection from inside the guest operating system to the underlying hardware layer. With this kind of attack, it is possible an instance to modify the firmware operations or take control of some part of the hardware device. This kind of infection is possible using the PCI pass-through functionality of the hypervisor[60]. In this way, the attack can affect not only the operation of the host machine, but also any other virtual machine which shares the same piece of hardware.

6.3.1.2.2

In addition to this, in a virtualized environment, the guest operating system operates upon a virtualized hardware. The majority of open source hypervisors, require QEMU a machine emulator and virtualizer[61] in order to operate. Usually, QEMU virtualizes legacy hardware and additionally it has often been subject to various attacks.

Among other security precautions, Openstack supports sVirt which provides protection against both hypervisor and virtual machine threats[62]. With sVirt, KVM provides isolation technologies for virtual machines, applying SELinux against the virtualization. In principle, sVirt isolates guest operating systems applying Mandatory Access Control (MAC security policy.

6.3.2 Public Cloud

Linode London cloud infrastructure is built based on Telecity group's Powergate data center, which is designed, built and operated by top engineers to the highest industry standards. Powergate data center features high-density capabilities, which support the power, cooling and management requirements of Linode's cloud servers. Following will refer to the physical Powergate data center as Linode cloud.

[60] Openstack, "Openstack Object Storage API v1 Reference," [Online]. Available: http://docs.openstack.org/api/openstack-object-storage/1.0/content/ch_object-storage-dev-overview.html. [Accessed 1 9 2014].
[61] "QEMU open source machine emulator and virtualizer," [Online]. Available: http://wiki.qemu.org/Main_Page. [Accessed 27 10 2014].
[62] J. Morris, "sVirt: Hardening Linux Virtualization with Mandatory Access Control," 2009. [Online]. Available: http://namei.org/presentations/svirt-lca-2009.pdf. [Accessed 27 10 2014].

Power

- Flexible and scalable power densities and configuration of up to 20kW
- Fully redundant and resilient power supplies at a minimum of N+1
- Power distribution through dedicated A&B supplies
- Separate UPS system which covers 15 minutes of grid-power failure
- Backup diesel generation at N+1 for event of grid-power failure, which is always warm up and work in seconds

Cooling

- Robust heating, ventilation and air conditioning (HVAC) system
- Minimum N+1 redundancy on water-cooling system
- Close Control Unites (CCUs) provide conditioned air at minimum N+2 redundancy
- Fully concurrently maintainable, dual-path secondary water distribution

Fire detection and suppression

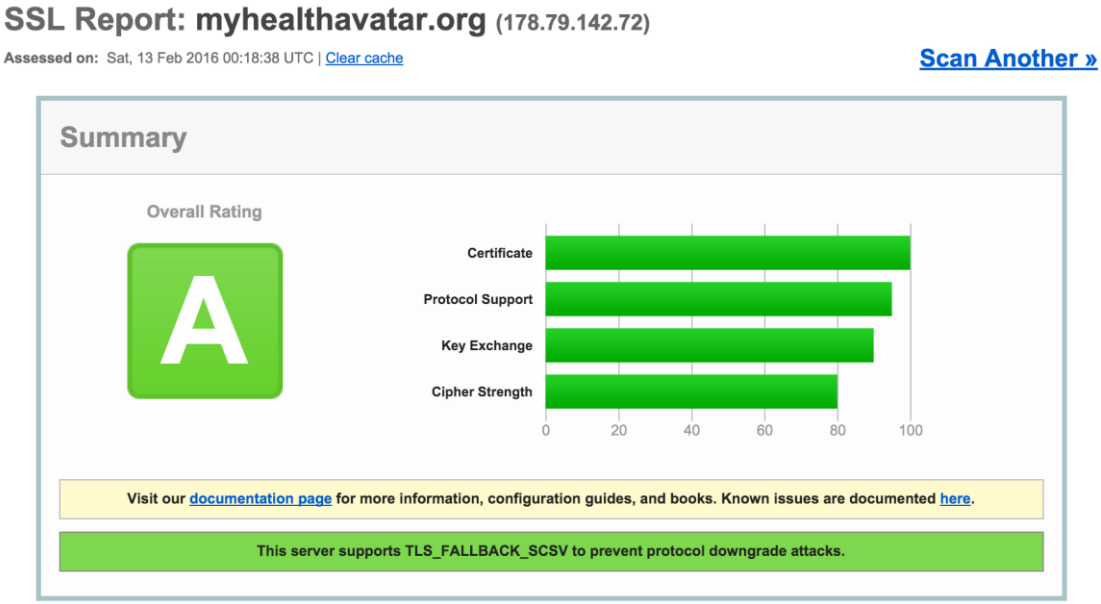
- High sensitivity smoke detection system (VESDA)
- Fully addressable two-stage fire detection and monitor system
- Water mist fire suppression system
- Network monitored fire-protection system

Security and access

- Protected by security personnel and multi-layered physical security
- Integrated digital video camera surveillance throughout the exterior and interior of data center
- Independent client-card and biometric-identification access system
- 24/7/265 manned security with unified security-breach alarm

6.4 MHA API security

MHA API is mainly based on OAuth 2 protocol, implemented with Java and Spring Security OAuth framework. HTTPS is always used for API access, strict HTTPS configuration guidelines are followed and from all four prospects, myhealthavatar.org get overall rating of A from ssllabs (https://www.ssllabs.com/ssltest/analyze.html?d=myhealthavatar.org)



MHA API's OAuth 2 supports the Authorization Code Grant and Implicit Grant flows as defined in RFC 6749. For third party application which have a web server, the Authorization Code flow is recommended, since this flow requires server-to-server communication using the third party's client secret. Mobile application that do not have web server should use the Implicit Grant flow.

Third party should never put the client secret in distributed code, e.g. client-side JavaScript or application downloaded from app store. For security considerations, MHA authorization page has built in mechanism to prevent itself been embedded as iframe, third party should not attempt to embed the authorization page in any manner. In native applications, SFSafariViewController class is recommended for iOS development, while Chrome Custom Tabs should be used by Android applications.

The OAuth 2 access token is issued for both grant flows, and refresh token is only issued to Authorization Code Grant flow. The refresh token can be used to renew access token when the access token is expired. The access token from Implicit grant flow is valid for longer period of time, however, user will need to authorize third party application again when the access token expires.

User is given total control of the API access; and user could revoke the authorization from MHA web UI. Revoked token will stop third party from further access any API on behalf of that user.

6.5 MHA API security for third party applications – CHF and oaCARE examples

OAuth 2.0 is an open authorization protocol which enables applications to access each other's data. Both CHF Alarm and oaCARE use **Implicit** grant type (which is the proper type for mobile and browser-based applications respectively).

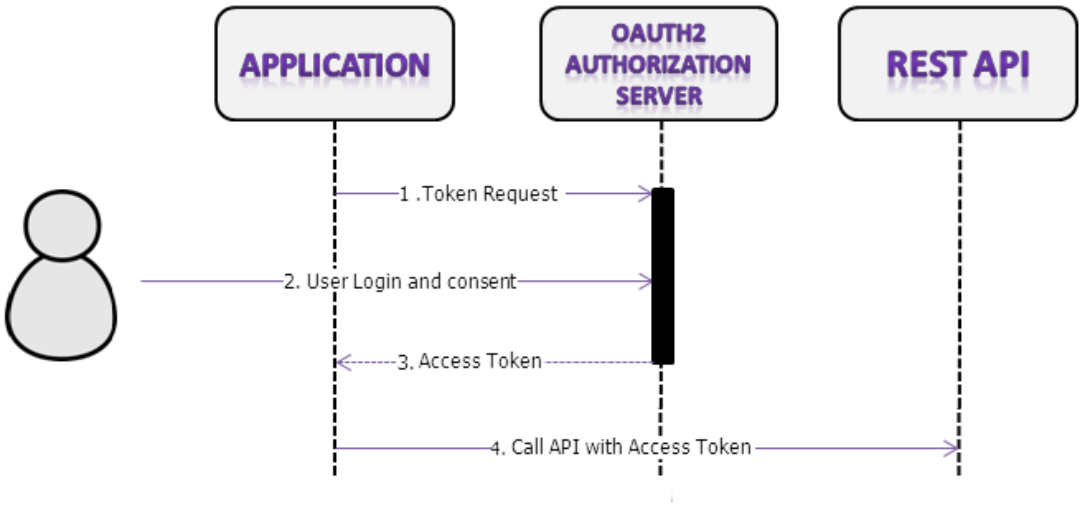


Figure 5: Flowchart for the Implicit Grant type of OAuth2.0

1) Register the application and obtain a user account for MyHealthAvatar web application (done once)

2) Obtain an access token from the OAuth 2 Authorization server:

a) Token request to the server using client_id (Figure 2)

http://myhealthavatar.org/mha/oauth/authorize?client_id=YOUR_CLIENT_ID&response_type=token&redirect_uri=YOUR_REDIRECT_URL

b) User login and consent (Figures 3 and 5)

c) Return of access token by the server

3) Call MyHealthAvatar API with the access token (Figure 4)

E.g.

url: 'https://myhealthavatar.org/mha/api/v3/profile/full',

type: 'GET',

headers: { 'Authorization':'Bearer YOUR_ACCESS_TOKEN}

Below we provide graphical examples of the applications developed:

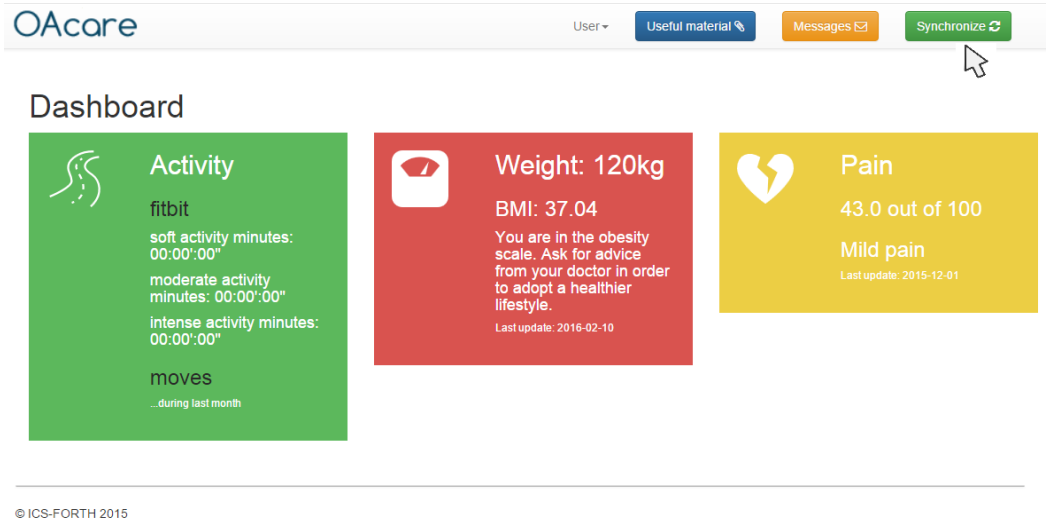


Figure 6: oaCare example – User wants to select to be synchronized with MHA (cursor on the synchronized button), then the app (following previous mentioned procedure will requests for an access token through MHA

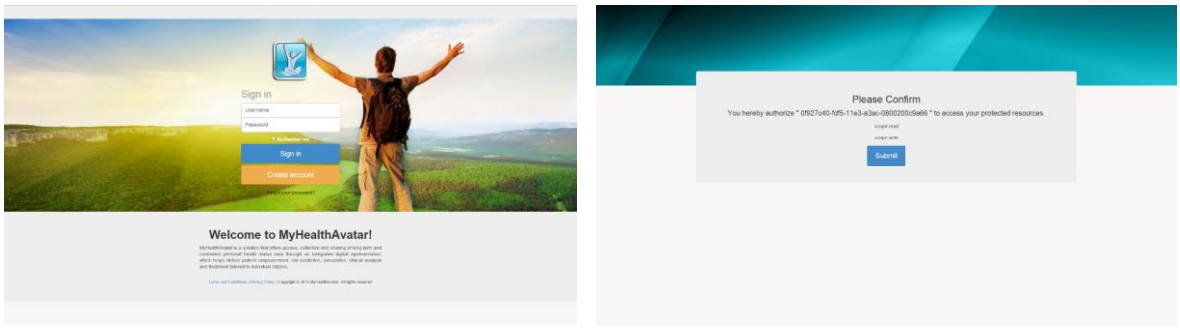


Figure 7: MHA User login/consent and access token retrieved

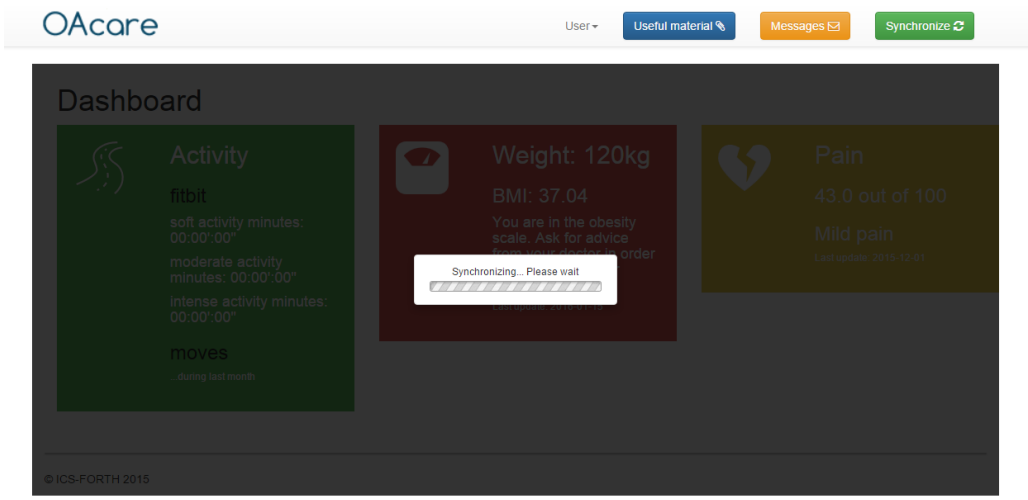


Figure 8: Synchronization of data to/from MHA (Sending and/or receiving data process)

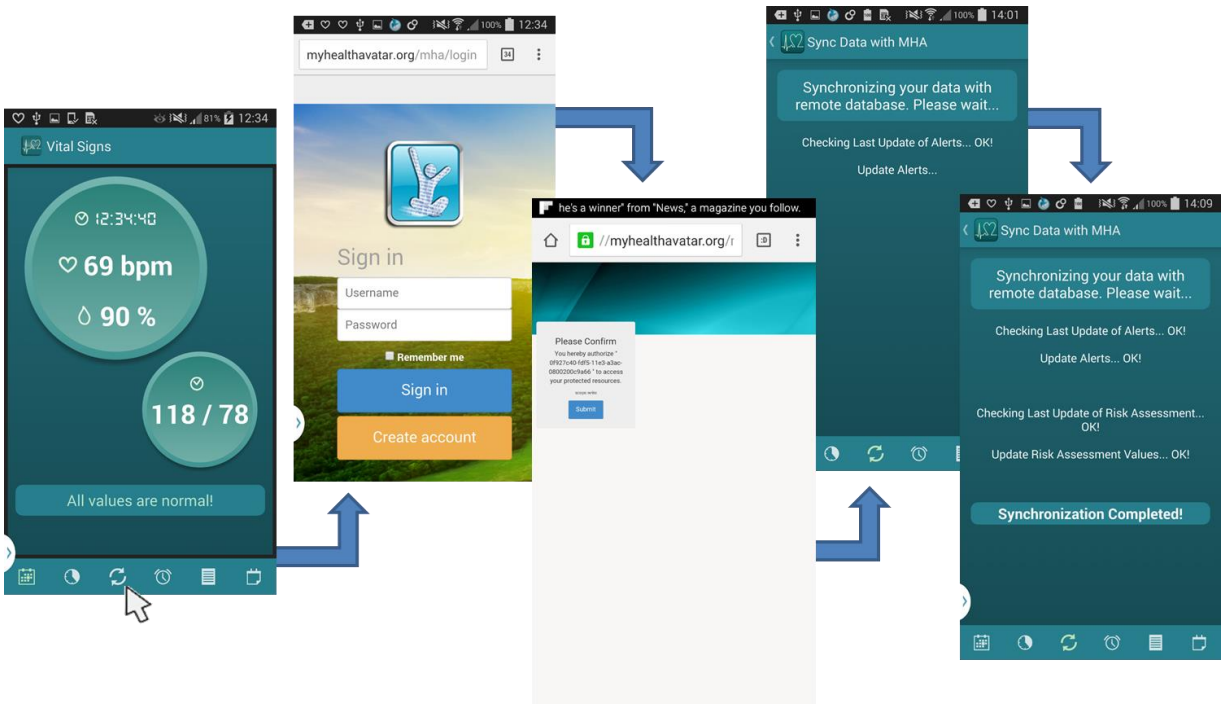


Figure 9: User request to sync data with MHA, user login and consent for CHFAlarm application use case – sequence diagram

6.6 Auditing

A successful auditing system requires high quality information on events with potential hazardous security risks. The provision of this information is primarily the responsibility of the service provider, but well-coordinated cooperation between the developers of service providers, and audit engineers is necessary to enrich the audit messages with enough data to provide clear and sufficient audit trails in a production environment. The preferred audit data model is based on openXDAS2[63] for MHA.

Below follows a description of the main elements of an audit message. The elements that are not described below are self-explanatory.

The standardized event types specified by XDAS are simple:

- XDAS_AE_CREATE_ACCOUNT - Create account
- XDAS_AE_DELETE_ACCOUNT - Delete account
- XDAS_AE_DISABLE_ACCOUNT - Disable account
- XDAS_AE_ENABLE_ACCOUNT - Enable account
- XDAS_AE_QUERY_ACCOUNT - Query account attributes
- XDAS_AE_MODIFY_ACCOUNT - Modify account attributes
- XDAS_AE_CREATE_SESSION - Create a user session
- XDAS_AE_TERMINATE_SESSION - Terminate a user session
- XDAS_AE_QUERY_SESSION - Query a user session attributes
- XDAS_AE_MODIFY_SESSION - Modify user session attributes
- XDAS_AE_CREATE_DATA_ITEM - Create data item
- XDAS_AE_DELETE_DATA_ITEM - Delete data item
- XDAS_AE_QUERY_DATA_ITEM_ATT - Query data item attributes
- XDAS_AE_MODIFY_DATA_ITEM_ATT - Modify data item attributes
- XDAS_AE_INSTALL_SERVICE - Install service or application
- XDAS_AE_REMOVE_SERVICE - Remove service or application
- XDAS_AE_QUERY_SERVICE_CONFIG - Query configuration of service or application
- XDAS_AE_MODIFY_SERVICE_CONFIG - Modify configuration of service or application
- XDAS_AE_DISABLE_SERVICE - Disable service or application
- XDAS_AE_ENABLE_SERVICE - Enable service or application
- XDAS_AE_INVOKE_SERVICE - Invoke service or application
- XDAS_AE_TERMINATE_SERVICE - Terminate service or application
- XDAS_AE_QUERY_PROCESS_CONTEXT - Query processing context
- XDAS_AE_MODIFY_PROCESS_CONTEXT - Modify processing context
- XDAS_AE_CREATE_PEER_ASSOC - Create an association with a peer

[63] http://openxdas.sourceforge.net/architecture.html

- XDAS_AE_TERMINATE_PEER_ASSOC - Terminate an association with a peer
- XDAS_AE_QUERY_ASSOC_CONTEXT - Query an association context
- XDAS_AE_MODIFY_ASSOC_CONTEXT - Modify an association context
- XDAS_AE_RECEIVE_DATA_VIA_ASSOC - Receive data via an association
- XDAS_AE_SEND_DATA_VIA_ASSOC - Send data via an association
- XDAS_AE_CREATE_DATA_ITEM_ASSOC - Create association with data item
- XDAS_AE_TERMINATE_DATA_ITEM_ASSOC - Terminate association with data item
- XDAS_AE_QUERY_DATA_ITEM_ASSOC_CONTEXT - Query context of association with data item
- XDAS_AE_MODIFY_DATA_ITEM_ASSOC_CONTEXT - Modify context of association with data item
- XDAS_AE_QUERY_DATA_ITEM_CONTENTS - Query data item contents
- XDAS_AE_MODIFY_DATA_ITEM_CONTENTS - Modify data item contents
- XDAS_AE_START_SYS - Start system
- XDAS_AE_SHUTDOWN_SYS - Shutdown system
- XDAS_AE_RESOURCE_EXHAUST - Resource exhaustion
- XDAS_AE_RESOURCE_CORRUPT - Resource corruption
- XDAS_AE_BACKUP_DATASTORE - Backup datastore
- XDAS_AE_RECOVER_DATASTORE - Recover datastore
- XDAS_AE_AUD_CONFIG - Configure audit service
- XDAS_AE_AUD_DS_FULL - Audit datastore full
- XDAS_AE_AUD_DS_CORR - Audit datastore corrupted

These events are arranged into a two-level hierarchy, the top level of which specifies the class of event type, while the leaf level is the actual event type.

Auditing is managed centrally in MHA. All services within the MHA domain may send audit log messages to the auditing service. The audit system (Figure 7) consists of four major components:

Account Management Events: This set of events is applicable to the management of principal accounts. A principal may be an end-user or a service within the system, a psuedo-user.

- Create account - the creation of an account representing a principal within a domain.
- Delete account - the deletion of an account representing a principal from a domain.
- Disable account - an action the prevents a principal account from being used within a domain.
- Enable account - an action that permits a principal account to be used within a domain.
- Query account attributes - the requesting of the attributes associated with a principal within a domain.
- Modify account attributes - the modification of the attributes associated with a principal within a domain.

User Session Events: This set of events is relevant to the creation and use of user sessions on the system.

Create a user session - the establishment of a processing environment to service an end user.

Terminate a user session - the dismantling of a processing environment associated with servicing an end user.

Query user session attributes - the requesting of the attributes associated with a user session.

Modify user session attributes - the modification of security significant attributes of the context of a processing environment servicing an end user.

Data item and Resource Element Management Events: This set of events relate to the creation and management of data items and resource elements within a domain. The type of data item or resource element is dependent upon the domain, e.g., files and directories, device special files, shared memory segments, within an operating system, tables and records within a database, messages within an email system. The term data item is used to refer to any type of resource element.

Create data item - creation of a data item within a domain.

Delete data item - deletion of a data item from a domain.

Query data item attributes - the requesting of the attributes associated with a domain data item.

Modify data item attributes - the modification of the security attributes of a domain data item such as access control attributes, ownership, aliases.

Service or Application Management Events:

This set of events relate to the management of system services and applications.

Install service or application - the installation of additional or updated software on a system, e.g., an application or system service.

Remove service or application - the deinstallation of software on a system.

Configure service or application - the modification of the configuration data associated with a software component.

Query configuration of service or application - the requesting of information about the configuration of a service or application.

Disable service or application - an action that prevents an application or system service from being used, for example, inhibiting responses to service requests. It may also involve the termination (shutdown) of application processing components that are currently providing the service.

Enable service or application - an action that permits an application or system service to be used, for example, allowing responses to service requests. This may also involve the invocation of specific application processing components (startup).

Service and Application Utilization Events: These events relate to the use of service and applications. They typically map to the execution of a program or a procedure and manipulation of the processing environment.

Invoke service or application - the invocation of a service or application (exec), e.g., operating system utility, database, accounting application, etc.

Terminate service or application component - the termination (exit) of the use of a service or application. This could be at the instigation of the application itself or by the intervention of the domain in response to user or administrative action.

Query processing context - the requesting of the attributes associated with the current processing environment.

Modify processing context - the modification of the attributes associated with the current processing environment.

Peer Association Management Events:

- Create an association with a peer - the creation of a communication channel and the processing context between system components.
- Terminate an association with a peer - the closure of a communications channel and destruction of processing context between system components.
- Query an association context - the requesting of the attributes of a context associated with a communications channel between peers.
- Modify an association context - the modification of the attributes of a processing context associated with a communications channel.
- Receive data via an association - receiving data from associated peer within current association context.
- Send data via an association - sending data to associated peer within current association context.

Data Item or Resource Element Content Access Events:

These events relate to the formation of an association between a service or application and a data item or resource element for the purpose of using its contents or services. For example, a file or directory, device special file, memory segment, communications port, etc.

- Create association with data item - create an association with (open) a data item. This creates a binding between the caller and the data item.
- Terminate association with data item - the termination of an existing association with (close) a data item.
- Query context of association with data item - the requesting of the context of an association with a data item, e.g., mode of access, size limits, access path, etc.

- Modify context of association with a data item - the modification of the context of an association with a data item or resource element.
- Query data item contents - the requesting of the contents of a domain data item (read).
- Modify data item contents - the modification of the contents of a domain data item (write, append, etc.).

Exceptional Events: These are events that are considered to be outside the generalized events listed above.

- Start system - the action of booting a system host or of changing the processing state of a system host to an operational mode.
- Shutdown System - the action of halting the processing by a system host or of changing the processing state of a system host to a maintenance mode.
- Resource exhaustion - the detection of resource exhaustion which has a potential impact on system operations, perhaps based upon a configurable threshold, e.g., data storage resources, communication end points.
- Resource corruption - the detection of an integrity failure of a system resource, for example data storage resource.
- Backup datastore - the action of making a backup copy of a datastore for the purposes of protecting availability and integrity of the data it contains.
- Recover datastore - the action of restoring the contents of a datastore from a previously made backup copy for the purposes of restoring the availability of the contents, or the integrity of the contents, or both.

Audit Service Management Events: These are events of specific relevance to the audit service itself.

- Configure audit service - the modification of the parameters controlling the operation of the audit service, for example, audit event filtering criteria.
- Audit datastore full - the detection of resource exhaustion for the particular instance of the resource used to store the log of audit event records.
- Audit datastore corrupted - the detection of a datastore integrity failure for the particular instance of the resource used to store the log of audit event records.

Audit Events may be specifically referenced by an Event Number. A set of Audit Events may be referred to by an Event Class. The concept of an Event Class is included in the XDAS solely as an administrative convenience. It provides an efficient and convenient reference to sets of audit events so that audit filters can be easily defined. An audit event record only includes the Event Number. It does not include any reference to Event Class for two reasons: its inclusion leads to redundant information in the audit record; and the mapping of event classes across administrative domains is problematic. When specified in filtering selection criteria, an event class is translated internally into the individual event numbers.

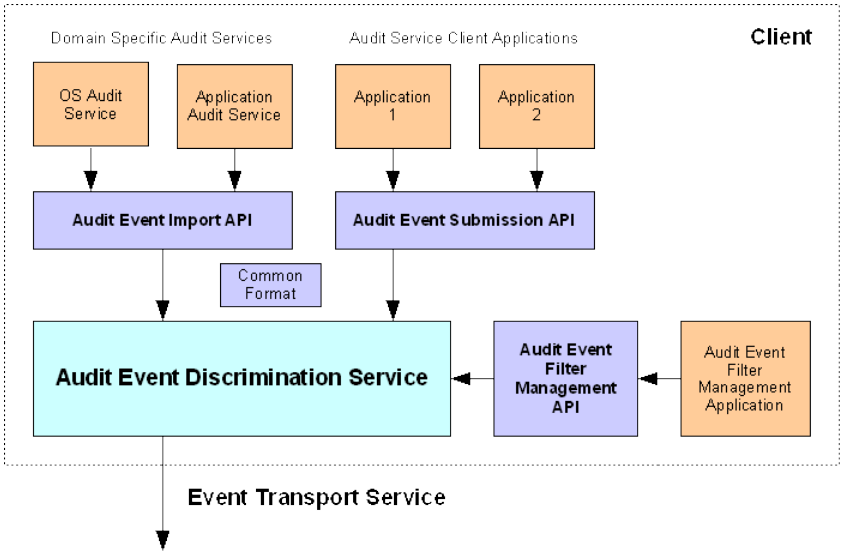


Figure 10: Audit Conceptual Architecture

7 Conclusion

This deliverable aims to define the security framework, measures and guidelines of MyHealthAvatar platform. This deliverable reports the work done for building and maintaining the structure of the security platform framework by investigating and reporting security issues and measures for infrastructure, resource management, data access and federation, computing resource (possible links with external HPC). It deals with all the security aspects of the technological platform, ranging from user authentication, authorization, and auditing, to data integrity and privacy, to pseudo anonymization and re identification of patient data. The security tools and policies developed ensure and enforce the legal and regulatory compliance and encompass the appropriate auditing mechanisms that are needed by the legislation. This deliverable includes guidelines on how to pose these security mechanism in MHA platform, provides details on the legal privacy and legal issues associated with the platform and gives the general security framework infrastructure of the system.

8 Appendix 1 – Abbreviations and acronyms

ADaM Analysis Data Model CDISC standard supporting efficient generation, replication, and review of analysis results

AES Advanced Encryption Standard a specification for the encryption of electronic data based on a design principle known as a Substitution permutation network

AGPL Affero General Public License refers to two free software licenses. Affero General Public License, Version 1 and GNU Affero General Public License, version 3.

API application programming interface is a particular set of rules ('code') and specifications that software programs can follow to communicate with each other

ASCII American Standard Code for Information Interchange a character-encoding scheme based on the ordering of the English alphabet

BMP Bitmap a raster graphics image file format used to store bitmap digital images

BSD Berkeley Software Distribution is a Unix operating system derivative developed and distributed by the Computer Systems Research Group (CSRG) of the University of California, Berkeley

CAS Central Authentication Service a single sign-on web protocol

CA Certification Authority an entity that issues digital certificates

CBC Cipher-Block Chaining a cryptographic mode of operation in which each block of plaintext is XORed with the previous ciphertext block before being encrypted

CCM Counter with CBC-MAC Mode a mode of operation for cryptographic block ciphers

CCZero Creative Commons licenses are several copyright licenses that allow the distribution of copyrighted works

CDASH Clinical Data Acquisition Standards Harmonization CDISC standard describing the basic recommended (minimal) data collection fields for 18 domains, including common header fields, and demographic, adverse events, and other safety domains that are common to all therapeutic areas and phases of clinical research

CDA Clinical Document Architecture is an XML-based markup standard intended to specify the encoding, structure and semantics of clinical documents for exchange

CDE Clinical Document Architecture an XML-based markup standard defined by HL7 intended to specify the encoding, structure and semantics of clinical documents for exchange

CDISC Clinical Data Interchange Standards Consortium - a global, open, multidisciplinary, non-profit organization that has established standards to support the acquisition, exchange, submission and archive of clinical research data and metadata

CDMI Cloud Data Management Interface - defines a functional interface that applications can use to create, retrieve, update and delete data elements from the Cloud

CDS Clinical Decision Support decision support software designed to assist physicians and other health professionals with decision making tasks, as determining diagnosis of patient data

CMWG Cloud Management Work Group focused on standardizing interactions between cloud environments by developing specifications that deliver architectural semantics and implementation details to achieve interoperable cloud management between service providers and their consumers and developers

CRISP-DM Cross Industry Standard Process for Data Mining a data mining process model that describes commonly used approaches that expert data miners use to tackle problems

CRL Certificate Revocation List a list of certificates that have been revoked, and therefore should not be relied upon

CSS Cascading Style Sheets is a style sheet language used to describe the presentation semantics (the look and formatting) of a document written in a markup language

CSV Comma-Separated Values a set of file formats used to store tabular data in which numbers and text are stored in plain-text form that can be easily written and read in a text editor

CWM Common Warehouse Metamodel a specification for modeling metadata for relational, non-relational, multi-dimensional, and most other objects found in a data warehousing environment

CeCILL CEA CNRS INRIA Logiciel Libre is a free software license adapted to both international and French legal matters, in the spirit of and retaining compatibility with the GNU General Public License

CellML Cell Markup Language is an XML based markup language for describing mathematical models

DICOM Digital Imaging and Communications in Medicine - a standard for handling, storing, printing, and transmitting information in medical imaging

DTMF Distributed Management Task Force brings the IT industry together to collaborate on the development, validation and promotion of systems management standards

EHR Electronic health record is an evolving concept defined as a systematic collection of electronic health information about individual patients or populations

EULA End-user licensing agreements An EULA is a legal contract between the manufacturer and/or the author and the end user of an application

EUPL European Union Public License the first European Free/Open Source Software (F/OSS) license

FMA F Foundational Model of Anatomy it is concerned with the representation of classes or types and relationships necessary for the symbolic representation of the phenotypic structure of the human body in a form that is understandable to humans and is also navigable, parseable and interpretable by machine-based systems

FieldMLField Markup Language is an XML based markup language for describing field models

GAS Gridge Authorization Service provides functionality that would be able to fulfill most authorization requirements of grid computing environments

GCM Galois/Counter Mode a mode of operation for symmetric key cryptographic block ciphers that has been widely adopted because of its efficiency and performance

GEM Guideline Elements Model an XML-based guideline document model that can store and organize the heterogeneous information contained in practice guidelines

GNU Gnu's Not Unix is a Unix-like computer operating system developed by the GNU project, ultimately aiming to be a "complete Unix-compatible software system" composed wholly of free software.

GO Gene Ontology is a major bioinformatics initiative with the aim of standardizing the representation of gene and gene product attributes across species and databases

GPL General Public License is the most widely used free software license, originally written by Richard Stallman for the GNU Project

GridFTP GridFTP is an extension of the standard File Transfer Protocol (FTP) for use with Grid computing

HL7 Health Level Seven is an all-volunteer, non-profit organization involved in development of international healthcare informatics interoperability standards

HMAC Hash-based Message Authentication Code a mechanism for message authentication using cryptographic hash functions

HTML Hypertext Markup Language is the predominant markup language for web pages. HTML elements are the basic building-blocks of webpages.

HTTPS Hypertext Transfer Protocol Secure is a combination of the Hypertext Transfer Protocol (HTTP) with SSL/TLS protocol to provide encrypted communication and secure identification of a network web server

IBM International Business Machines

ID-FF Liberty Identity Federation Framework an approach for implementing a single sign-on with federated identities based on commonly deployed technologies

ID-WSF Liberty Identity Web Services Framework a framework for identity-based web services in a federated network identity environment

IEC International Electrotechnical Commission is the world's leading organization that prepares and publishes International Standards for all electrical, electronic and related technologies

IEEE Institute of Electrical and Electronics Engineers is a non-profit professional association headquartered in the United States that is dedicated to advancing technological innovation and excellence

IETF Internet Engineering Task Force a large open international community of network designers, operators, vendors, and researchers concerned with the evolution of the Internet architecture and the smooth operation of the Internet

IHE Integrating the Healthcare Enterprise - an initiative by healthcare professionals and industry to improve the way computer systems in healthcare share information

IPSec Internet Protocol Security a protocol suite for securing Internet Protocol (IP) communications by authenticating and encrypting each IP packet of a communication session

ISBN International Standard Book Number is a unique numeric commercial book identifier based upon the 9-digit Standard Book Numbering (SBN) code created by Gordon Foster

ISO International Organization for Standardization is an international standard-setting body composed of representatives from various national standards organizations

InSilicoML InSilico Markup Language is a markup language that can explicitly describe the multi-level hierarchical structures of the physiological functions in mathematical models

JDMP Java Data Mining Package an open source Java library for data analysis and machine learning

JPEG Joint Photographic Experts Group is a commonly used method of lossy compression for digital photography

JSDL Job Submission Description Language is an extensible XML specification from the Global Grid Forum for the description of simple tasks to non-interactive computer execution systems

KNIME Konstanz Information Miner a user-friendly and comprehensive open source data integration, process, analysis and exploration platform

LGPL Lesser General Public License is a free software license published by the Free Software Foundation

LOINC Logical Observation Identifiers Names and Codes is a database and universal standard for identifying medical laboratory observations

MAGE-ML Microarray and Gene Expression - Markup Language markup language format for the representation of gene expression data from microarrays to facilitate the exchange of information between different data systems

MAGE-OM Microarray and Gene Expression - Object Model data exchange model for the representation of gene expression data from microarrays to facilitate the exchange of information between different data systems

MAGE-TAB Microarray and Gene Expression - Tabular tabular format for the representation of gene expression data from microarrays to facilitate the exchange of information between different data systems

MIAME Minimum Information About a Microarray Experiment needed to enable the interpretation of the results of the experiment unambiguously and potentially to reproduce the experiment

MIASE Minimal Information About a Simulation Experiment common set of information a modeller needs to provide in order to enable the execution and reproduction of a numerical simulation experiment, derived from a given set of quantitative models

MIASE Minimum Information About a Simulation Experiment is an effort to list the common set of information a modeller needs to provide in order to enable the execution and reproduction of a numerical simulation experiment, derived from a given set of quantitative models.

MIBBI Minimum Information for Biological and Biomedical Investigations maintains a web-based, freely accessible resource for ”Minimum Information” checklist projects, providing straightforward access to extant checklists (and to complementary data formats, controlled vocabularies, tools and databases), thereby enhancing both transparency and accessibility

MIT MIT License is a free software license originating at the Massachusetts Institute of Technology

ML Markup Language is a modern system for annotating a text in a way that is syntactically distinguishable from that text

MOF MetaObject Facility the foundation of OMG's industry-standard environment where models can be exported from one application, imported into another, transported across a network, stored in a repository and then retrieved, rendered into different formats

MPL Mozilla Public License is a free and open source software license

MS Microsoft is an American public multinational corporation headquartered in Redmond, Washington

MTOM Message Transmission Optimization Mechanism is the W3C Message Transmission Optimization Mechanism, a method of efficiently sending binary data to and from Web services

MedLEE Medical Language Extraction and Encoding system System to extract, structure, and encode clinical information in textual patient reports so that the data can be used by subsequent automated processes

NeuroML Neuro Markup Language is an XML (Extensible Markup Language) based model description language that aims to provide a common data format for defining and exchanging models in computational neuroscience

OASIS Organization for the Advancement of Structured Information Standards a not-for-profit consortium that drives the development, convergence and adoption of open standards for the global information society

OBO Open Biomedical Ontologies is an effort to create controlled vocabularies for shared use across different biological and medical domains

OGSA-BES Open Grid Services Architecture - Basic Execution Services defines Web Services interfaces for creating, monitoring, and controlling computational entities such as UNIX or Windows processes, Web Services, or parallel programswhat we call activities within a defined environment

OGSA-DAI Open Grid Service Architecture-Data Access and Integration allows data resources (e.g. relational or XML databases, files or web services) to be federated and accessed via web services on the web or within grids or clouds. Via these web services, data can be queried, updated, transformed and combined in various ways.

OSI Open Source Initiative is an organization dedicated to promoting open source software

OS Operating System is a set of programs that manages computer hardware resources, and provides common services for application software

OWL-S Ontology Web Language for web Services an ontology of services to discover, invoke, compose, and monitor Web resources offering particular services and having particular properties

OWL Web Ontology Language is a family of knowledge representation languages for authoring ontologies.

OpenID Open Identity provider of web-based SSO services

PAOS Reverse HTTP Binding for SOAP a binding that enables HTTP clients to expose services using the SOAP protocol, where a SOAP request is bound to a HTTP response and vice versa

PATO PATO an ontology of phenotypic qualities, intended for use in a number of applications, primarily defining composite phenotypes and phenotype annotation.

PHP PHP: Hypertext Preprocessor is a general-purpose server-side scripting language originally designed for web development to produce dynamic web pages

PKIX Public-Key Infrastructure Working Group was established in the fall of 1995 with the goal of developing Internet standards to support X.509-based Public Key Infrastructures

PMML Predictive Model Markup Language an XML-based language which provides a way for applications to define statistical and data mining models and to share models between PMML compliant applications

PNG Portable Network Graphics is a bitmapped image format that employs lossless data compression

POST POST is one of many request methods supported by the HTTP protocol used by the World Wide Web

RAD Rapid application development is a software development methodology that uses minimal planning in favor of rapid prototyping

RDF Resource Description Framework is a family of World Wide Web Consortium (W3C) specifications originally designed as a metadata data model

REST Representational state transfer is a style of software architecture for distributed hypermedia systems such as the World Wide Web

RFC Request for Comments is a memorandum published by the Internet Engineering Task Force (IETF) describing methods, behaviors, research, or innovations applicable to the working of the Internet and Internet-connected systems

RICORDO RICORDO is focused on the study and design of a multiscale ontological framework in support of the Virtual Physiological Human community to improve the interoperability amongst its Data and Modelling resources

RIM Reference Information Model is the cornerstone of the HL7 Version 3 development process and an essential part of the HL7 V3 development methodology

SAML Security Assertion Markup Language a standard, XML-based framework for creating and exchanging security information between online partners

SAS Business analytics software and service developer, and independent vendor in the business intelligence market

SAWSDL Semantic Annotations for WSDL defines mechanisms using which semantic annotations can be added to WSDL components

SBML System Biology Markup Language is a representation format, based on XML, for communicating and storing computational models of biological processes

SDTM Study Data Tabulation Model CDISC defining a standard structure for human clinical trial (study) data tabulations that are to be submitted as part of a product application to a regulatory authority

SED-ML Simulation Experiment Description Markup Language an XML-based format for encoding simulation experiments, following the requirements defined in the MIASE guidelines

SHA Secure Hash Algorithm a number of cryptographic hash functions published by the National Institute of Standards and Technology as a U.S. Federal Information Processing Standard

SLO Single Log-Out termination of a SSO action

SNIA Storage Networking Industry Association not-for-profit trade organization for companies and individuals in various sectors of the storage industry

SNOMED-CT Systematized Nomenclature of Medicine - Clinical Term is a systematically organised computer processable collection of medical terminology covering most areas of clinical information such as diseases, findings, procedures, microorganisms, pharmaceuticals etc

SOAP Simple Object Access Protocol is a protocol specification for exchanging structured information in the implementation of Web Services in computer networks

SOAP Simple Object Access Protocol a lightweight XML-based protocol for exchange of structured information in a decentralized, distributed environment

SOA Service-Oriented Architecture s a set of principles and methodologies for designing and developing software in the form of interoperable services

SPARQL SPARQL Protocol and RDF Query Language - query language for RDF

SQL Structured Query Language a standard language for accessing and manipulating databases

SSH Secure Shell is a network protocol for secure data communication, remote shell services or command execution and other secure network services between two networked computers that it connects via a secure channel over an insecure network

SSL Secure Sockets Layer a cryptographic protocol that provides communication security over the Internet, predecessor of TLS

SSO Single Sign-On a mechanism whereby a single action of user authentication and authorization can permit a user to access all computers and systems where he has access permission, without the need to enter multiple passwords

TCP/IP Transmission Control Protocol/Internet Protocol the first two networking protocols defined in the Internet Protocol Suite standard

TDD Test-driven development is a software development process that relies on the repetition of a very short development cycle.

TLS Transport Layer Security a cryptographic protocol that provides communication security over the Internet, successor of SSL

UML Unified Modelling Language a specification defining a graphical language for visualizing, specifying, constructing, and documenting the artifacts of distributed object systems

VDM Vienna Development Method is one of the longest-established Formal Methods for the development of computer-based systems

VPH-NoE Virtual Physiological Human - Network of Excellence is a project which aims to help support and progress European research in biomedical modelling and simulation of the human body

VPH Virtual Physiological Human is a methodological and technological framework that, once established, will enable collaborative investigation of the human body as a single complex system

WAV Waveform Audio File Format is a Microsoft and IBM audio file format standard for storing an audio bitstream on PCs

WS-* Web Services-* common prefix for the family of Web Services specifications

WSDL Web Services Description Language a way to describe the abstract functionalities of a service and concretely how and where to invoke it

WSMO Web Service Modelling Ontology ontology for describing Semantic Web Services

XFree86 A freely redistributable open-source implementation of the X Window System

XHTML Extensible HyperText Markup Language is a family of XML markup languages that mirror or extend versions of the widely-used Hypertext Markup Language (HTML), the language in which web pages are written

XML Extensible Markup Language - a format for encoding documents in machine-readable form, similar in syntax to HTML

XTS XEX-based Tweaked Codebook a mode of operation for cryptographic block ciphers

caBIG Cancer Biomedical Informatics Grid a virtual network of interconnected data, individuals, and organizations that work together to redefine how cancer research is conducted