CrowdHEALTH

Collective Wisdom Driving Public Health Policies

Del. No. – D4.17. Integrated Holistic Security and Privacy Framework v1

Project Deliverable

This project has received funding from the European Union's Horizon 2020 Programme (H2020-SC1-2016-CNECT) under Grant Agreement No. 727560

Del. No. – D4.17. Integrated Holistic Security and Privacy Framework v1

Work Package:	WP4
Due Date:	31/10/2017
Submission Date:	07/11/2017
Start Date of Project:	01/03/2017
Duration of Project:	36 Months
Partner Responsible of Deliverable:	UPRC
Version:	1.1
Status:	☒ Final ☐ Draft ☐ Ready for internal Review ☐ Task Leader Accepted ☒ WP leader accepted ☐ Project Coordinator accepted
Author name(s):	Stefanos Malliaros (UPRC), Christos Xenakis (UPRC), George Moldovan (SIEMENS)
Reviewer(s):	Antonio De Nigro (ENG), Kostas Perakis (SILO)
Nature:	☒ R – Report ☐ D – Demonstrator
Dissemination level:	☒ PU – Public ☐ CO – Confidential ☐ RE – Restricted

REVISION HISTORY			
Version	Date	Author(s)	Changes made
0.1	17/09/2017	UPRC	Initial ToC
0.2	04/10/2017	UPRC	Added content
0.3	09/10/2017	UPRC	Added Content
0.4	17/10/2017	UPRC	Added Content / Review Version
0.7	24/10/2017	SIEMENS	Added Content
0.9	30/10/2017	UPRC	Addressed Comments from Peer Review
0.9.1	31/10/2017	SIEMENS	Added Content
0.9.2	31/10/2017	UPRC	Addressed Comments from Peer Review
1.0	03/11/2017	SIEMENS	Revised Authors
1.1	07/11/2017	ATOS	Quality control and Submission to EC

List of acronyms

ABAC	Attribute Based ACcess Control
DPO	Data Protection Officer
EHR	Electronic Health Record
ENG	ENGineering - Ingegneria Informatica SPA
ECA	Event Condition Action
GDPR	General Data Protection Regulation
IT	Information Technology
ISO	International Organization for Standardisation
JSON	JavaScript Object Notation
JOSE	JavaScript Object Signing and Encryption
JWT	JSON Web Token
OAuth 2.0	Open Authentication version 2.0
PETs	Privacy Enchancing Technologies
PPDP	Privacy Preserving Data Publishing
REST	REpresentational State Transfer
RBAC	Role Based Access Control
SAML	Security Assertion Markup Language
SILO	SIngular LOgic
SSO	Single Sign On
TLS	Transport Layer Security
URID	Unique Random IDentifier
UPRC	University of Piraeus Research Centre
XACML	eXtensible Access Control Markup Language
XML	eXtensible Markup Language
XSL	XML Stylesheet Language

Table of Contents

Table of Figures

1. Executive Summary

This document is part of the WP4 *Information and Knowledge Acquisition and Management* of the CrowdHEALTH project. The purpose of this report is to describe the current status of the Holistic Security and Privacy Framework of CrowdHEALTH, which is crucial for the protection of the CrowdHEALTH's resources and data. This document presents briefly the regulatory requirements of CrowdHEALTH, and presents the technologies and protocols that will be used to fulfil the relevant requirements. The next version of this deliverable will include technical and implementation details that will be agreed within the project consortium.

The structure of this document is as follows. Section 1 introduces the reader to the scope of this document, while section 2 investigates the data protection requirements that CrowdHEALTH is obliged to follow due to regulatory legislation. Next, section 3 presents a summary of the CrowdHEALTH's holistic security and privacy framework, and Section 4 is the main section of this document and presents the technologies that will be used by the security and privacy framework. This section will be enhanced with technological and implementation details at the next versions of this deliverable. Finally, section 5 concludes by summarizing the achievements of the CrowdHEALTH's security and privacy framework.

2. Introduction

To reap the promise of digital health information to achieve better health outcomes, smarter spending, and healthier people, providers and individuals alike must trust that an individual's health information is private and secure. In order for patients to disclose their health information with a healthcare provider, they need to trust that the (EHRs) will be protected and that the confidentiality of their personal information is not at risk.

CrowdHEALTH will employ a security and privacy framework that assures the confidentiality, integrity, and availability of the data managed and processed within the scope of the project. Any interaction on the data will be handled from the integrated holistic security and privacy framework that provides: (i) trust management to quantify the trustworthiness of the participating users and healthcare ecosystem entities; (ii) data anonymization to ensure privacy; and (iii) access control and authorization to facilitate both integrity and authorized data disclosure. Due to the nature of e-health data CrowdHEALTH's security and privacy framework protects the individually identifiable health information, such as the individual's past, present or future physical and mental condition, or the provision of health care to the individual. Such information is of crucial importance and technical measures need to be taken, since health information can be used in order to identify an individual.

The structure of this document is as follows. Section 1 introduces the reader to the scope of this document, while section 2 investigates the data protection requirements that CrowdHEALTH is obliged to follow due to regulatory legislation. Next, section 3 presents a summary of the CrowdHEALTH's holistic security and privacy framework, and Section 4 is the main section of this document and presents the technologies that will be used by the security and privacy framework. This section will be enhanced with technological and implementation details at the next versions of this deliverable. Finally, section 5 concludes by summarizing the achievements of the CrowdHEALTH's security and privacy framework.

3. Regulatory assessment

The European Union's General Data protection Mechanism (GDPR), also known as Regulation (EU) 2016/679, is a very important legislation that will come into effect on 25th May 2018. Compliance to GDPR applies to the processing of personal data wholly or partially by automated means and to the processing other than by automated means of personal data which form part of a filling system or are intended to form part of a filling system [1]. Currently, the EU member states have a diverse legislation regarding data protection. By having GDPR coming into force, tougher fines for non-compliance and breaches will be introduced, and moreover individuals will have more control over what companies can do with the personal data they have.

Figure 1: GDPR key changes

The following key changes are proposed by GDPR:

- **Personal data:** GDPR enforces a broader aspect of personal data than the existing Directive 95/46/EC [2], by regulating also genetic, medical, economic cultural or social data. The organizations that possess data must qualify which data are considered personal, where they are physically stored, and whether they are encrypted, unencrypted or anonymized.
- **Consent:** GDPR introduces new rules regarding the consent of the user that is required to let the organization process his personal data. Consent forms will be created and will be easily accessible, and written in an easily understandable manner.
- **Data protection officer:** Any organization that has more than 250 employees, or, if it processes more than 5000 personal data records in any given year shall set a data protection officer (DPO).

- **Privacy impact assessments:** The GDPR introduces Privacy Impact Assessments (PIAs) as a mean to identify risks to the privacy rights of individuals, when processing their personal data. The organization then addresses the identified risks, by employing technical controls, such as encryption, anonymization or pseudoanonymization. The PIA shall happen before the process of personal data, and it should focus on topics, such as the system's description and the processing activity.
- **Data breach notification:** GDPR introduces the data breach notification regulations and changes in liability will have a profound impact on the supply chain. Data processors shall alert and inform controllers immediately after a data breach or without undue delay. Another impact of GDPR is that contacts negotiated with suppliers will also need to be future-proofed for GDPR.
- **Right to be forgotten:** GDPR enables the data subjects to apply for the deletion of his/her data collected or shared to other service providers.
- **Data portability:** The right of data portability allows individuals to reuse their personal data for their purpose across different services. For example, it allows the transfer of personal data between different IT systems in a safe and secure way.
- **Protect by design:** Protect by design, also known as Privacy by design, in a service is addressed by GDPR, as a legal obligation for data controllers and processors, making an explicit reference to data minimization and the possible use of pseudoanonymization [3]. This will render data protection as a default practice of systems and services.

CrowdHEALTH will employ Privacy by design, since it promotes data and privacy protection from the phases of design and implementation of an IT system. Designing IT systems in such way will can lead to some significant benefits, such as increased awareness of privacy and data protection and also organizations are more likely to meet the regulatory requirements.

3.1. Persons in charge of the processing

The processing or personal data is performed only by individuals authorized to have access and process personal data. Their activity is limited precisely to the scope defined by the scope of CrowdHEALTH.

3.2. Processing of personal data

In accordance to [2, 1, 4], CrowdHEALTH will process personal data in compliance to the following statements:

- The personal data within CrowdHEALTH are relevant and adequate to the project. This requirement is fulfilled, since the CrowdHEALTH partners are responsible for importing relevant information to the project.
- Personal data are protected against accidental modification or loss by employing appropriate technical measures. User authentication and authorization are used in order to identify that an individual has the right to access the system's resources, and Attribute Based Access Control is exploited in order to allow or prohibit interactions to

system resources, based on the attributes of the user, the attributes of the resource, the environmental conditions, and the relevant access control rules

- All sensitive data are anonymized before entering the CrowdHEALTH system in a way that no subject is identifiable. The CrowdHEALTH databases will include *k-anonymous* versions of e-health data in order to avoid any subject identification.
- Personal data cannot be communicated between organizations and especially in different countries, without an appropriate protection mechanism. This is achieved by using end to end encryption between the different entities of CrowdHEALTH.

4. CrowdHEALTH Security and Privacy Framework summary

The CrowdHEALTH Security and Privacy framework outlines a structure which administers the security and privacy requirements of the project. A Security and Privacy Framework contains guidelines that fulfil the security requirements, and aim to protect the Confidentiality, Integrity, and Availability of the data, resources, services, and users of a system. Due to the nature of CrowdHEALTH, and the data that will be maintained, processed, and stored, a Security and Privacy framework is of vital importance to mitigate potential threats and risks related to users' privacy.

The CrowdHEALTH Security and Privacy framework contains Privacy Enhancing Technologies (PETs) to conform with the EU laws of data protection. The objective of PETs is to protect and ensure the confidentiality and secure management of the personal information. CrowdHEALTH will employ PETs to perform user authentication, authorization, and access control, trust evaluation and modelling, as well as to achieve data anonymization of the e-health data that will be managed by CrowdHEALTH.

User authentication is the key to secure a computer system and in CrowdHEALTH will be the first step before letting the user access the resources and services. CrowdHEALTH will employ state-of-the-art authentication protocols to protect users against security threats. Particularly, CrowdHEALTH will exploit federated identity management, by employing a secure Single Sign-On mechanism, which enables the user identification from entities that rely on the result of the authentication process. This has a significant advantage, since a user registered to a system A shall be able to authenticate, and use services of a third-party system B by exploiting the identity of system A. CrowdHEALTH will employ OpenID connect [5], which is the state of the art authentication protocol, as it provides flexibility, scalability, and lightweight user authentication. After a successful authentication, user authorization and access control are the next two steps before a user can access the requested service or resource.

Authorization is the process of determining whether an already authenticated user can access the information resources. For instance, if an authenticated nurse would like to access the medical record of a patient on a file server, then it will be responsibility of the file server to determine whether the user is allowed for this type of access. CrowdHEALTH will employ authorization by exploiting OAuth 2.0 [6], which is token-based open standard for user authorization. OAuth 2.0 provides a process for resource owners to authorize third-party access to their resources without having to perform authentication and maintain user credentials.

The last step before an authenticated and authorized user can access a requested resource is to apply an access control mechanism to verify whether the requested access to the resource permitted or prohibited. CrowdHEALTH will employ the attribute based access control (ABAC) [7] mechanism to build effective and efficient access control policies. ABAC is a scalable mechanism, which relies on the user attributes, the resource attributes, and the access control rules defined by system administrators to permit or forbid access to a requested resource.

CrowdHEALTH will employ a Trust evaluation model that will provide the ability of users to compute their trust rating based on several parameters. The set of rules creating these two models – namely the trust and the reputation model, are the core part of the Trust and Reputation Modelling component. Moreover, the Trust and Reputation Model includes a third model – namely a Reaction Model, which specifies what kind of event the (trust and reputation) mechanisms should generate and propagate to the system. The Trust evaluation model that will provide a mechanism which, when used by individual observers, can enable them to produce their own trust rating. The rating can refer to single or sets of data streams (i.e. measurements), to the services providing the measurements, or to the devices as a whole. By merging such individual trust ratings according to a Reputation Model, for each entity (i.e. data stream, services or devices) a reputation rating can be then computed.

5. Holistic Security and Privacy Framework Components

The security components that form the CrowdHEALTH's security and privacy framework ensure the compliance of the CrowdHEALTH platform to the security principles, which are confidentiality, integrity, and availability. More specifically, the objective of the employed security measures is to protect the personal data within CrowdHEALTH, from unauthorized disclosure (i.e. confidentiality), to prevent unauthorised modification (i.e. integrity), and to assure that the data are available when required (i.e. availability).

In CrowdHEALTH's architecture, data are stored and maintained in the Data store. The data are transferred between different services, and as a rule data are encrypted during transfer. To assure the confidentiality, integrity, and availability of data, CrowdHEALTH applies security mechanisms to achieve effective authentication, authorization, access control, anonymization, and trust reputation. Each one of these mechanisms are presented and elaborated in the following subsections.

5.1. User Authentication and Authorization

5.1.1. Introduction to user authentication and authorization

User authentication is an act, where a user delegates his/her authority to exploit identity information to another entity. New technologies have emerged, which provide the ability to exchange identity information by using a short string token. Security Assertion Markup Language (SAML) [8] was one of the most conspicuous technologies, which uses an artefact as a tool to exchange identity information. However, a significant limitation of SAML is that it does not offer schemes for managing user access control, which is a significant requirement of CrowdHEALTH.

Following SAML, OAuth is an open standard for identity delegation and authorization. There exist two versions of OAuth [9] [10], which OAuth 2.0 not being backwards compatible with OAuth 1.0. OAuth 1.0 is a protocol for identity delegation, while OAuth 2.0 is a framework, which focuses on providing authorization for web and desktop applications, as well as mobile phones and smart devices. OAuth 2.0 does not provide encryption, digital signatures, or client verification services, but instead it uses the TLS protocol to offer a degree of confidentiality and server authentication.

5.1.2. OpenID Connect

OpenID Connect [11] is a protocol which is based on OAuth 2.0 and exploits a JSON/REST-based identity built-in functionality, alongside with JSON Web Tokens (JWT) [12]. To be more specific, OpenID Connect consists of an identity layer on top of the OAuth 2.0 framework, which enables clients to perform identity verification, based on the authentication performed by an authorization server. Moreover, some basic profile information is obtained about the identified person in an interoperable REST-like manner. The main idea of OpenID Connect is

to create an API which provides seamless authentication and authorization, to be able to build lightweight to implement for applications.

Based on the D2.1 State of the Art and Requirements Analysis v1 three distinguished roles are defined:

- **End-user:** End user is a person, who accesses protected resources of other end-users. Each end user has his own unique ID, for example username or e-mail address, and this attribute is used to distinguish and contact different end-users. In OpenID Connect, End users are distinguished into two different categories: delegators and delegates. A delegator is either the owner of a resource or he/she has rights to delegate privileges on it to other end users. A delegate is an end-user that has been provided with privileges by a delegator to access his resources. It is essential for both delegators and relegates to have a trust relationship.
- **Client:** A client is an entity that gives access to restricted resources, which are managed by authorized end-users. This means a Client may give, deny or revoke access to resources containing personal information based on its security policies, if the Client verifies the appropriateness of the end-user requesting the access.
- **OpenID Connect Provider:** OpenID Connect Provider is an entity which is responsible for authenticating users by exploiting authentication methods, and producing an assertion of the completion of the authentication, which might also include some basic information about the end-user. An OpenID Connect Provider is also responsible for supervising the delegation authorization of a delegator to a delegate and for informing the relevant Clients.

There are two key differences between SAML and OpenID Connect. The first is that OpenID Connect applies most of the complexity to the OpenID Connect Provider, whereas SAML distributes this complexity to both the provider and the client. Also, OpenID Connect migrated from XML to JSON, which has already been supported by all modern programming environments. Moreover, JWT which also supports Signing and Encryption (JOSE) [13] allows for more practical and compact tokens by using the XML language.

	SAML 2.0	OpenID Connect
Service Provider	Client libraries	Client libraries
Identity Provider	Identity Provider libraries	OpenID Connect Provider libraries
Attribute Provider	Attribute provider provides further detail to enrich SAML assertion Requires further step to populate assertion with user attributes	OpenID Connect Provider The userinfo endpoint returns claims about the end-user
Attributes	SAML attributes	OpenID connect scopes
Discovery Service	Requires pre-agreed metadata	Single discovery service for client allowing sites & apps can validate your users
Privacy	Yes	JSON Object Signing and Encryption (JOSE)
Signing	Yes	JSON Web Token (JWT)
Mobile Apps	No, SAML web profile for web browser only	Both web browser & mobile apps
Support for SSO	Web SSO only	Yes
Form Rendering	Both client and identity provider	Normally Identity provider

Figure 2: OpenID Connect and SAML 2.0 comparison

5.1.3. OpenID Connect in CrowdHEALTH

CrowdHEALTH's authentication protocol enables its users' to seamlessly authenticate and use different services without the need of reauthentication, and without the need of a separate user account. The user account contains the attributes of the user, and among them the mandatory attributes are: Name, Surname, Nationality, Email and Password. Regarding the password, it is essential to enforce a password policy, since it is one of the basic security measures to prevent unauthorized access. Since, most users tend to select easy to remember passwords, and they do not want to change it, a well-defined password policy is the first keystone to protect the system from unauthorized access. CrowdHEALTH's password policy will include the following rules:

- Password Strength: the strength of the passwords is one of the most critical properties of a password. The password strength depends on several parameters and these are the minimum password length, and the character set that the password is comprised of. In CrowdHEALTH, a password can be no shorter than 8 characters, and the users have to select passwords than contain both uppercase and lowercase letters, numbers and symbols.
- Password expiration time: By setting up a password expiration time, users are enforced to change their password in regular time intervals. CrowdHEALTH will employ a password expiration time of 1 year.

- Trivial password selection: CrowdHEALTH will not accept passwords that can be easily guessed. A password cannot contain trivial words, such as the word *password*, and it can also not contain the users' name or surname intact.
- Uniqueness of passwords: The uniqueness of passwords specifies the number of new passwords that the user has to select before being able to reuse a previously used password. In CrowdHEALTH, the users cannot use the same passwords until they have changed their password three times.

The OpenID Connect protocol follows a specific execution flow to achieve authentication and authorization of the CrowdHEALTH end-users. The execution flow is described briefly below:

- Step 1: The CrowdHEALTH end-user requests access to some resources via the client.
- Step 2: The client redirects the session of the end-user's web browser to the OpenID Connect Provider's system for authentication.
- Step 3: The OpenID Connect Provider authenticates the CrowdHEALTH end-user by using either username or password, or even by using a two-factor authentication.
- Step 4: After the authentication has been performed, the OpenID Connect provider performs a redirection of the end-user's web browser or application to the client, including an authorization code.
- Step 5: The client sends a POST request to the OpenID Connect provider along-side with the authorization code provided by the end-user
- Step 6: The OpenID Connect Provider responds to the client with an ID token, which contains details about the attributes of the authenticated use in JWT form, and an optional Access token which is used for accessing resources. Apart from user identification, the OpenID Connect Provider also identifies the client which requested the initial authentication.
- Step 7: If in the previous step the OpenID Connect Provider also sends an Access token, then the client may send it back to the OpenID Connect Provider to request further profile information for the CrowdHEALTH end-user.
- Step 8: The OpenID Connect Provider returns the user profile to the client containing the requested information (e.g. email).

After the completion of the execution flow of the OpenID protocol, the end-user has been authenticated to the OpenID Connect Provider, and authorized to the Client. Also, the client has been authorized to access the protected resources by using the token obtained by the OpenID Connect Provider. In CrowdHEALTH, the client is the different applications (e.g. web site) that that are used by the end-users to access the offered services, while the OpenID Connect provider is an internal part of the CrowdHEALTH system and is responsible for authenticating users, and issuing tokens that are passed to clients, so that clients can request data on behalf of the end-users.

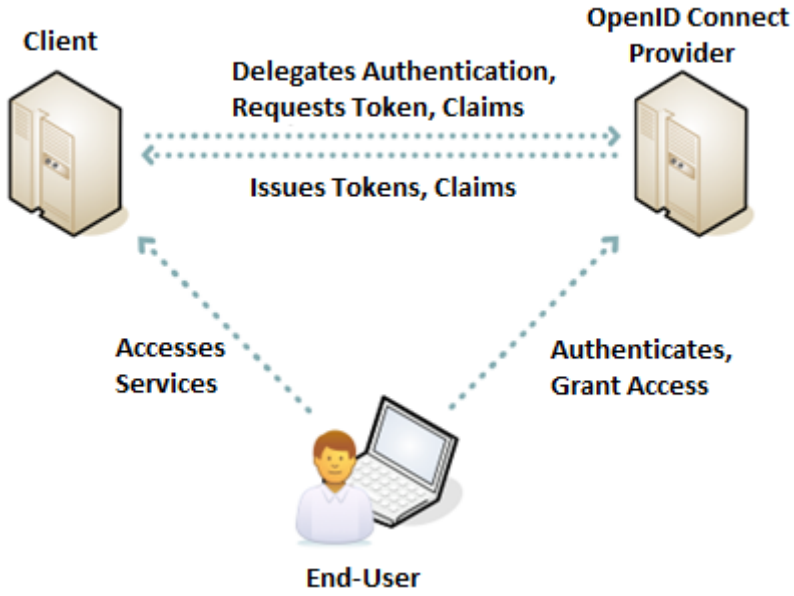

Figure 3: OpenID Connect execution flow

5.2. User Access Control

5.2.1. Access control preliminaries

Authentication of users is the first step towards effective authorization, and effective verification of the user's permissions and privileges. Access control is one of the main methodologies used to perform the verification of the authorizations of an end user requesting access to specific restricted resources. Since CrowdHEALTH is a cross border system that manages and analyses anonymized health data from several data sources, it is crucial to integrate support of attribute-based access control (ABAC) in the CrowdHEALTH platform to be able to perform effective and efficient access control policies. By exploiting both the end-users and the resource attributes defined between different organizations, ABAC does not rely on explicit authorizations that are required prior to the access request to a resource. Also, it is scalable for large enterprises, where the management of other access control mechanisms, such as roles based access control and access lists, would be time inefficient. Maintaining access control over attributes based on both the end-user and the resource, authentication and authorization functionalities can be maintained in the same or separate infrastructures, yet maintaining appropriate security levels.

Access lists and role based access control (RBAC) are special cases of ABAC in terms of the used attributes [14], since access lists rely on the unique identifier of the end-user, while RBAC relies on the role or group of the end user. An example of a framework that employs ABAC is XACML [15]. To sum up, ABAC does not require directly assignments to end-users or their roles or groups before the request is performed by the end-user. Upon an end-user performing a request, ABAC can decide based on the assigned attributes of the end-user in

combination with the attributes of the resource, and other policies specified for the specific end-user and resource.

5.2.2. ABAC components

ABAC relies on the evaluation of subject attributes, object attributes, environment conditions and the access control policy defining allowable operation for subject-object combinations, as shown in Figure 4. These components are mandatory for every ABAC implementation, ranging from a small isolated system, to a cross border complex system with multiple data sources and user types. Suppose an example in which a hospital holds a patient's record. The health record has several attributes, such as date of creation, name, surname, past health issues, last modification date, etc. Upon the creation of new health records, these attributes must be set to apply efficient ABAC rules.

Every end-user that uses an ABAC system has some attributes defined, such as unique identifier, and these attributes are set by an authority within the organization of the end-user. Every resource of an ABAC system should have policies that define the access rules based on the combinations of the end-subject, environment conditions, and operations to the resource. The policy is derived from documented functional requirements. For instance, in the case of a hospital, based on the documented rules, only authorized medical personnel should be able to access a patient's medical record. One way to implement this into ABAC is to consider the medical record as the resource, to which the subject requests read permissions. Next, the subject's personnel type is identified, and based on the specified policy, if the subject is a non-medical staff, the operation Read access will be denied.

The ABAC policies can be highly complex, and their functionality is scoped to the degree introduced by the computational language, and the richness of the available attributes [7]. Due to this fact, it is feasible to allow connections between subjects and objects, without specifying explicit relationships between them. The provision of attributes on objects and subjects is a process that needs to follow some rules that state the acceptable operations for all subjects and users. These rules are not mandatory to be changed as new end-user accounts are crated within the system.

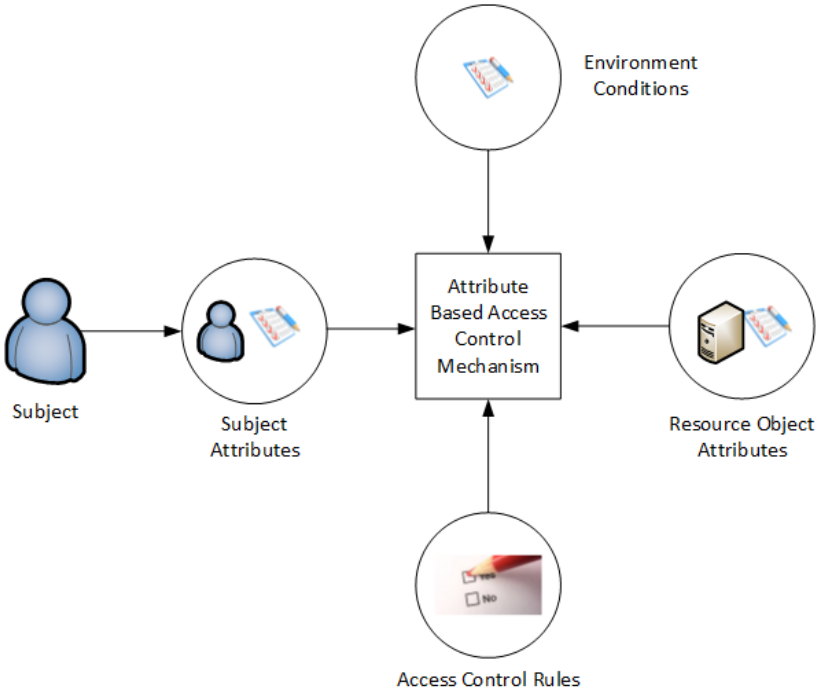

Figure 4: ABAC Architecture

5.2.3. ABAC integration in CrowdHEALTH

CrowdHEALTH is an ecosystem that will be used from several stakeholders from different EU member states. The stakeholders could be citizens, policy makers, healthcare professionals and organizations etc. Users of different groups have different level of access within CrowdHEALTH due to security policy of the OpenID Connect provider. For example, a doctor may not have access to the patient's data from a different country, while the policy makers should have necessary permissions to perform analytics and create new policies.

CrowdHEALTH will integrate ABAC with OpenID Connect to create an efficient, effective, and scalable infrastructure [16] for authorization of the end-users. In general, the ABAC architecture consists of three entities: the end-user, the issuer, and the verifier. The end-user is the subject that wants to access the resources of the system. The authentication is performed by exploiting credentials or other attributes that have been issued by the issuer, which is the OpenID Connect Provider of CrowdHEALTH. The verifier, also called client in CrowdHEALTH, is an entity that can request attributes of the end-user on behalf of the end-user, based on an authentication performed by an authentication server.

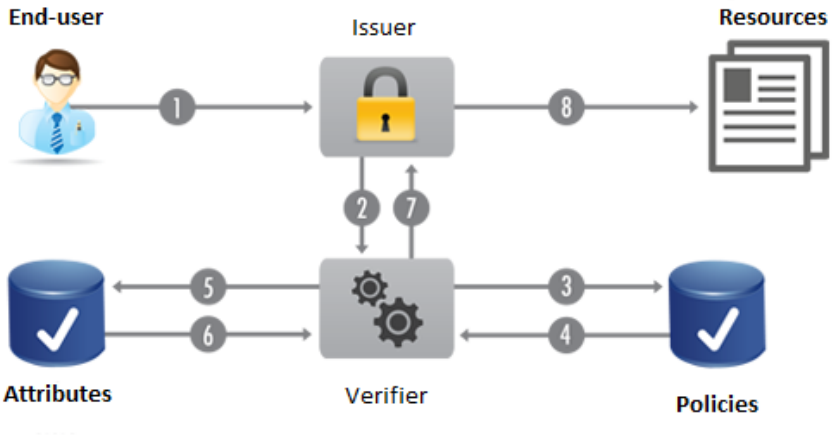

Figure 5: ABAC execution flow

CrowdHEALTH's security and privacy framework interacts with other CrowdHEALTH's components that rely on user authentication and authorization (e.g. Big Data Platform). The users of CrowdHEALTH will be able to access the CrowdHEALTH's database based on the result of an ABAC request. The ABAC component of the security and privacy framework shall state the permission that the users' have on the requested resource (e.g. CrowdHEALTH's database) as follows:

- Read-Only: The user has read-only access to the requested resource
- Write-Only: This user (or application) has write-only access to the indicated resource. This is mostly for applications or layers that will be adding information to the indicated resource but don't need to retrieve information.
- Read-Write: The user has read and write access to the requested resource.
- Admin: The user is the administrator of the resource and is allowed to modify both the content and the metadata of a resource. For example, an administrator is allowed to change the database schema, as well as from adding content to the database.

5.3. Data Anonymization

5.3.1. Introduction to Data anonymization

Data anonymization is a procedure followed by organizations to remove personal information from collected user data, to use, archive and share such data with other organizations [17].There exist several approaches, tools, and algorithms that can be exploited to perform data anonymization depending on the types of data. Some techniques are more effective, while others pose security issues, in terms of re-identification of subjects from anonymized data.

The sharing of anonymized data has great benefits, especially in clinical research, since best practices can be identified. Apart from the benefits of data sharing, there is a possibility that some anonymized information may be linked to the original subject, resulting in the re-

identification of a subject the loss of privacy protection. Due to this risk, some organizations which share anonymized data, also sign a data usage agreement, which forbids the one who holds the anonymized data to attempt to re-identify the subjects.

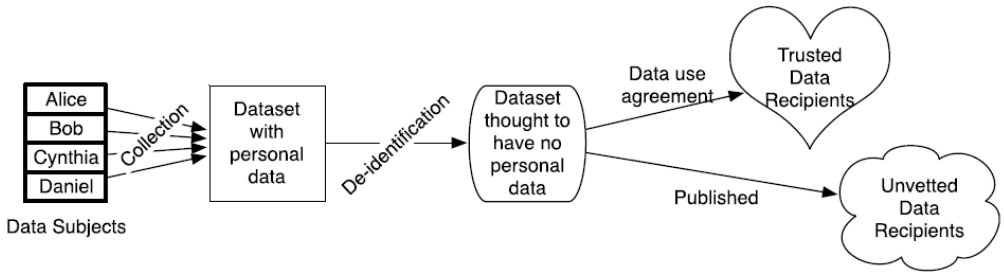

Figure 6: Data anonymization flow model

Data anonymization is the mainly used methodology to achieve Privacy Preserving Data Publishing (PPDP) [18]. In this methodology data are processed to create a new anonymized dataset that will be distributed or published, allowing researchers to use the anonymized data as input for their purposes. Thus the aim of PPDP is to offer data of high utility, without threating the privacy of the data subjects.

5.3.2. CrowdHEALTH anonymization procedure

In CrowdHEALTH, the procedure that will be used to anonymize the data will consist of two stages. The first one aims to remove the attributes that directly identify an individual, while the second aims in pseudoanonymizing the rest data, in a way that the individual's privacy is not threatened. The anonymization will take place at the source of data to avoid any potential security threats during the transmission of data to the CrowdHEALTH system, and to also avoid identification from information disclosure. The data that can directly identify a person are called direct identifiers, while the data that can identify indirectly a person are called indirect identifiers, as shown in Figure 7.

Direct identifiers			Indirect identifiers				
Name	Surname	Social Security Number	Birth Date	Sex	Weight	Disease	Religion

Figure 7: A hypothetical table containing direct and indirect identifiers

ISO 25237:2008 [19], and its latest version issued in 2017 [20] specifies explicitly, the direct identifiers. Some examples of direct identifiers are names, social security numbers, email addresses, ID card numbers, passport numbers, address, etc. ISO 25237 also advises to treat medial record numbers and phone numbers as direct identifiers, since these attributes are

widely used, and re-identification based on these attributes is easy. In the bibliography, the direct identifiers are either suppressed or systematically replaced with pseudonyms. In CrowdHEALTH, we will remove the direct identifiers from the anonymized dataset to avoid re-identification incidents, as done for instance in [21].

Indirect identifiers introduce a significant issue that the anonymization procedure will face. Indirect identifiers hold useful information, such as the measurements of a medieval test that can be used for analytics and policy creation. CrowdHEALTH will employ “suppression” and “generalization” as the main two methodologies for anonymizing indirect identifiers. In suppression, the value or a part of the value of the indirect identifier is replaced by an asterisk. For example, a Greek ZIP code states a small number of blocks in which the subject live. By suppressing the last 2 numbers to find in which municipality the subject lives, which is useful for analytics that include geographical data. In generalization, the values of the indirect identifiers are not removed, but are replaced by a broader category. For instance, the attribute age can be split into several intervals, such as <20, 20 ≤ Age < 25, 25 ≤ Age < 30, and ≥ 30.

CrowdHEALTH will achieve *k-anonymity* [22] in the datasets that it will process, which is one of the most popular data anonymization methodologies. An anonymized *k-anonymous* dataset refers to a dataset, that for every combination of indirect identifiers, there are at least *k* matching records. For instance, if a dataset contains the age and the gender of subjects and is *k=3* anonymous, means that for every combination of age and gender there are at least three matching records. For CrowdHEALTH to achieve *k-anonymity*, the following steps will be performed:

- **Attribute identification:** An expert states the attributes that can be used as direct identifiers.
- **Removal of direct identifiers:** The direct identifiers are removed and not contained in the created anonymized dataset.
- **Unique Random IDentifier:** CrowdHEALTH will create a Unique Random IDentifier (URID) for every patient that the database has information about. This is crucial, since different sets of data are stored in different files, and all the information is related with each other by the URID. The usage of URID cannot break *k-anonymity*, due to the amount of data that will be imported into the CrowdHEALTH system
- **Threat model:** A threat model consists of adversaries, security threats, alongside with the information they might hold, that can be exploited to re-identify the data subjects.
- **Utility of anonymized data:** The utility of the created anonymized dataset will be determined.
- **Create the anonymized dataset:** The initial dataset underdoes the anonymization process, thus creating the anonymized dataset.

5.4. Trust and Reputation Modelling

5.4.1. Introduction

The Trust Model's task is to provide a mechanism and set of rules which, when used by individual observers, can enable them to produce their own trust rating reflecting their individual trust in specific entities or information received. This resulting trust rating can refer to single data stream, to sets of data streams (i.e. measurements), to the services providing the measurements, or to the devices as a whole. By merging such individual trust ratings according to a Reputation Model, for each entity (i.e. data stream, services or devices) a reputation rating can be then computed.

The set of rules creating these two models – namely the trust and the reputation model - are the core part of the Trust and Reputation Modelling component. In addition to defining and adjusting the said models, it is important to note and specify how the system is to react to the generated ratings. Towards this end, the Trust and Reputation Model includes a third model – namely a Reaction Model, which specifies what kind of event the (trust and reputation) mechanisms should generate and propagate to the system. - Such events can include a plethora of definable actions, from simple annotation of data stored within the platform's back-end to dispatching alerts to various subscribed components.

5.4.2. Requirements towards the Trust and Reputation Model

This section presents some of the requirements for the Trust and Reputation Model. A detailed list of requirements can be found in D2.1 State of the Art and Requirements Analysis. We note that these requirements are expected to change during the project's lifetime, and that we will present any further changes and provide a context for each one during further updates of the deliverable.

- **Service Level Trust**: Services within the CrowdHEALTH platform should be able to query reputation ratings for specific services (e.g. a heart rate monitoring service)
- **Measurement Level Trust**: Services within the CrowdHEALTH platform should be able to query reputation ratings for specific measurements (e.g. a single or a set of heart rate monitoring readings)
- **Device Level Trust**: Services within the CrowdHEALTH platform should be able to query reputation ratings for specific devices (e.g. a wearable heart rate monitoring device)
- **Multiple Evaluation Criteria**: The Reputation Model should be based on multiple criteria, as defined and required by the entities processing the data (and the device-platform interactions). This includes the requirement for being able to interactively define and update the Trust and Reputation Models during runtime, in order to have the platform's component flexible and configurable.

- **Multiple Observers**: The Trust and Reputation Model should support and mitigate reports and ratings from multiple observers. This includes the ability to define new observers and evaluation rules.
- **Configurable Reaction Events**: The Trust and Reputation Model should be able to configure and dispatch reaction-based events according to the subscribers' specification. I.e., changes in trust and reputation ratings greater than 25% over a specific time interval.

5.4.3. Interactions of the Trust and Reputation Model

Figure 8 presents an overview of the interactions between these three models previous presented.

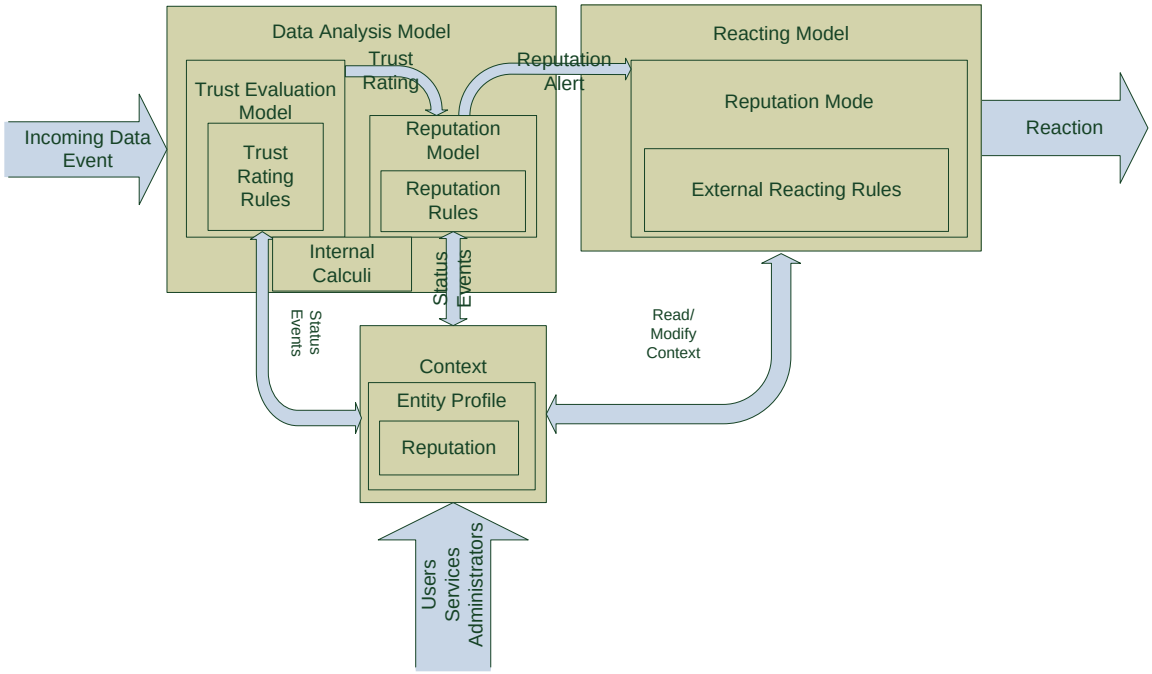

Figure 8 Trust and Reputation Model components

The Trust, Reputation and Reaction Models are all interacting components, driven by internal rules for processing information. We can distinguish between two types of modalities: active and passive.

In the active case, the mechanisms would compute and actively drive any interaction with the data and data producing entities. For example, the CrowdHEALTH system could require proofs or additional information during the interactions with data producers which have a low trust or reputation level.

In the passive case, the system is meant to determine what behaviour is considered “normal” or “anomalous” without using the trust and reputation ratings to drive any interactions. Only subsequent internal data processing steps would adjust depending on the existing ratings.

Within all these components, we identify four main specific values which determine the trustworthiness of a specific measurement, service or device:

- An observer - which is, the entity of the platform which observers and monitors the data streams and compares the observed behaviour with the expected one and therefore produces the trust ratings. A continuous, systematic evaluation of the data streams is required for computing correct trust ratings.
- Indicator: an Observer is usually unable to immediately categorise an entity as acting correctly, incorrectly or maliciously (i.e. based on a sudden spike in temperature or heart rate values). Instead, the Observer is detecting deviations from the expected values or (known) trends, specific to the entity observed. These deviations are instead considered as possible indicators for some kind of errors. The opposite is correct as well – value within expected ranges is considered correct behaviour.
- Trust Ratings: Indicators of abnormal/erroneous behaviour may determine the Observer to update his appreciation of an entity – more specifically: confidence, reliability or trustworthiness of an entity decreased. This information is quantified as a value, the *trust rating.* This value can be either a number, or a complex data structure.
- Reputation Ratings: Reputation Managers have the task of merging multiple trust ratings in *reputation ratings*. This is a global view and evaluation of multiple trust ratings.

Therefore, while the trust rating reflects the direct trust of an Observer into a specific element, the reputation rating reflects the global, indirect trust.

5.4.4. Statistic Evaluation of Trust and Reputation

The basic mechanism for evaluating streams is statistics methods such as averages of streams, jumps/deviations, density, or quantiles. These reference calculi are used in order to generate estimators for detecting on-the-fly abnormal service behaviour.

As an initial proof of concept, based on the RERUM [23] proposed mechanism, we employ the expert system CLIPS [24] to demonstrate and implement many of the rules mentioned.

The role of this initial proof of concept is to evaluate and demonstrate the streaming mode processing capabilities and the (relative) simplicity of the calculations performed. Further iterations of the Trust and Reputation Model plan to employ a more efficient evaluation system, but still similar to CLIPS.

As an example, the following rule (employed in RERUM) generates an alarm if the observed values are outside of an interval [*minA*, *maxA*]:

```
;=======================

; Rule valC-mima

;=======================
(defrule valC-mima "checks valC (str val)a-priori boundary conditions
 of each observer [ 0 < valC < 40 ]"
      (a-valC-mima (obsN ?obsN)(strN ?strN) (ruID ?ruID)(minA
?minA)(maxA ?maxA))
      (a-str (strN ?strN)(valC ?valC) (timC ?timC))
      (test (or (< ?valC ?minA)  (> ?valC ?maxA)))
      =>
      (assert (bad ?strN ?obsN "valC-mima" ?ruID (getTime ?timC)))
      ;(printout t "Alert! "?obsN"'s "?strN" values are abnormal"
crlf)
)
```

5.4.5. Reacting System

The reacting system's goal is to evaluate the Reaction Model and to trigger certain defined events. Of special interests are the definition and execution of non-trivial rules, which are to be expected in complex systems. A formally expression of rules for reacting systems is through the Event Condition Action (ECA) paradigm, which is typical for event-driven architectures and active database systems (through their support for database triggers). The ECA paradigm defines a structure of rules, which consist of three main entities:

- An event, which defines the stimuli which triggers the rule.
- A condition, which is evaluated and of which the execution of rule depends on.
- An action, which defines the actions undertaken.

ECA rules have are expressed in a high-level, declarative language. It is important therefore to have the language of choice complemented by sufficient knowledge models, so as to be able to have a system which can define events, conditions and actions of a complex high enough for the system to efficiently perform is tasks, There are several proposed approaches, such as using XML/ XSL for modelling constraints and entities [25].

6. Conclusions

CrowdHEALTH is highly related to security, and its' approach is based on privacy-by design. The CrowdHEALTH Integrated Holistic Security and Privacy framework fulfils the security needs of an e-health cross-border system, due to the adoption of security mechanisms, such as user authentication, user authorization, access control, data anonymization, trust management and reputation modelling. The framework will go under thorough testing to ensure its robustness and its acceptable performance.

The CrowdHEALTH consortium will be continuously informed about the related EU legislation and will adapt the holistic security and privacy framework accordingly. Possible changes or additions will be addressed and documented in the deliverable D4.18 which is a revision of D4.17.

7. References

[1] "Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016," *Official Journal of the European Union,* 2016.

[2] "Directive 95/46/EC of the European Parliament and of the Council," *Official Journal of the European Communities,* 1995.

[3] E. U. A. f. N. a. I. Security, "Data Protection," [Online]. Available: https://www.enisa.europa.eu/topics/data-protection/privacy-by-design. [Accessed 2 October 2017].

[4] "European Convention on Human Rights," [Online]. Available: http://www.echr.coe.int/Documents/Convention_ENG.pdf. [Accessed 2 October 2017].

[5] "OPenID Connect - Welcome to OpenID Connect," [Online]. Available: http://openid.net/connect/.

[6] "OAuth 2.0," [Online]. Available: https://oauth.net/2/.

[7] "Attribute Based Access Control," [Online]. Available: https://nccoe.nist.gov/sites/default/files/library/sp1800/abac-nist-sp1800-3b-draft.pdf. [Accessed 2017].

[8] "Security Assertion Markup Language (SAML)," [Online]. Available: http://saml.xml.org.

[9] "The OAuth 1.0 Protocol," [Online]. Available: https://tools.ietf.org/html/rfc5849.

[10] "The OAuth 2.0 Authorization Framework," [Online]. Available: https://tools.ietf.org/html/rfc6749.

[11] "OpenID Connect," [Online]. Available: http://openid.net/connect/.

[12] "JSON Web Token (JWT)," [Online]. Available: https://tools.ietf.org/html/rfc7519.

[13] "JSON Object Signing and Encryption (JOSE)," IANA, 2017. [Online]. Available: http://www.iana.org/assignments/jose/jose.xhtml.

[14] S. Carter, "RBAC vs ABAC Access Control Models - IAM Explained," Identity Automation, 19 January 2017. [Online]. Available: http://blog.identityautomation.com/rbac-vs-abac-

access-control-models-iam-explained.

[15] “OASIS eXtensible Access Control Markup Language (XACML) TC,” OASIS, [Online]. Available: https://www.oasis-open.org/committees/tc_home.php?wg_abbrev=xacml.

[16] E. C. a. T. R. Weil, “ABAC and RBAC: Scalable, Flexible, and Auditable Access Management,” *IT Professional,* vol. 15, no. 3, pp. 14-16, 2013.

[17] S. L. Garfinkel, “De-Identification of Personal Information,” INST.

[18] B. C. M. a. W. K. a. C. R. a. Y. P. S. Fung, “Privacy-preserving Data Publishing: A Survey of Recent Developments,” *ACM Comput. Surv.,* vol. 42, no. 4, 2010.

[19] “ISO/TS25237:2008 Health informatics -- Pseudoanonymization,” International Organization for Standardization, [Online]. Available: https://www.iso.org/standard/42807.html.

[20] “ISO 25237:2017 Health informatics -- Pseudoanonymization,” International Organization for Standarization, [Online]. Available: https://www.iso.org/standard/63553.html.

[21] “New York taxi details can be extracted from anonymized data, researchers say,” The Guardian, [Online]. Available: https://www.theguardian.com/technology/2014/jun/27/new-york-taxi-details-anonymised-data-researchers-warn.

[22] L. Sweeney, “k-anonymity: a model for protecting privacy,” *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems,* vol. 10, no. 5, pp. 557-570, 2002.

[23] D. R. López, J. Cuellar, R. Weber, P. Kasinathan, S. Suppan, R. C. Staudemeyer, H. C. Pöhls, A. Fragkiadakis, P. Charalampidis and E. Tragos, “RERUM D3.3 – Modelling the trustworthiness of the IoT,” RERUM, 2016.

[24] CLIPS, “A Tool for Building Expert Systems,” CLIPS, [Online]. Available: http://www.clipsrules.net/. [Accessed 29 October 2017].

[25] J. Bailey, G. Papamarkos, A. Poulovassilis and P. T. Wood, “An Event-Condition-Action Language for XM,” in *Web Dynamics: Adapting to Change in Content, Size, Topology and Use*, Berlin, Heidelberg, Springer Berlin Heidelberg, 2004, pp. 223--248.