ASCLEPIOS

Advanced Secure Cloud Encrypted Platform for Internationally Orchestrated Solutions in Healthcare

Project Acronym: **ASCLEPIOS**
Project Contract Number: **826093**

Programme**: Health, demographic change and wellbeing**
Call: **Trusted digital solutions and Cybersecurity in Health and Care to protect privacy/data/infrastructures**
Call Identifier: **H2020-SC1-FA-DTS-2018-2020**

Focus Area: **Boosting the effectiveness of the Security Union**
Topic**: Toolkit for assessing and reducing cyber risks in hospitals and care centres**
Topic Identifier: **H2020-SC1-U-TDS-02-2018**

Funding Scheme: **Research and Innovation Action**

Start date of project: 01/12/2018 Duration: 36 months

Deliverable:
D3.1 ASCLEPIOS Security and Policies Model

Due date of deliverable: 30/11/2019 Actual submission date: 08/12/2019

WPL: ICCS

Dissemination Level: Public

Version: final

1 Table of Contents

2 List of Figures and Tables

Figures

Tables

3 Status, Change History and Glossary

Status:	Name:	Date:	Signature:
Draft:	ASCLEPIOS Security and Policies Model	30/10/2019	
Reviewed:	Antonis Michalas (TUT)	22/11/2019	
Approved:	Tamas Kiss	08/12/2019	

Table 1: Status Change History

Version	Date	Pages	Author	Modification
v1.0	23/10/2019	74	Yiannis Verginadis, Dimitris Apostolou, Ioannis Patiniotakis, Gregoris Mentzas, Jenny Psarra (ICCS)	Context – aware security model provided
v1.1	5/11/2019	74	Yiannis Verginadis, Dimitris Apostolou, Ioannis Patiniotakis, Gregoris Mentzas, Jenny Psarra (ICCS)	Updated Context – aware security model based on partners' comments
v1.2	11/11/2019	103	Yiannis Verginadis, Dimitris Apostolou, Ioannis Patiniotakis, Gregoris Mentzas, Jenny Psarra (ICCS)	ASCLEPIOS Policies Model provided
v2.0	15/11/2019	103	Yiannis Verginadis, Dimitris Apostolou, Ioannis Patiniotakis, Gregoris Mentzas, Jenny Psarra (ICCS)	Pre-final version ready for internal review
v3.0	22/11/2019	103	Antonis Michalas (TUT)	Comments of internal review
Final	03/12/2019	81	Yiannis Verginadis, Dimitris Apostolou, Ioannis Patiniotakis, Gregoris Mentzas, Jenny Psarra	Final version

Table 2: Deliverable Change History

Glossary

ABE	Attribute-Based Encryption
SSE	Symmetric Searchable Encryption
FE	Functional Encryption
ABAC	Attribute-Based Access Control
NDC	National Drug Code
HL7	Health Level Seven
CPT	Current Procedural Terminology
NUCC	National Uniform Claim Committee
SNOMED	Systematized Nomenclature of Medicine
LOINC	Logical Observation Identifiers Names and Codes
ICD	International Classification of Diseases
ATC	Anatomical Therapeutic Chemical
DDD	Defined Daily Dose
WHOCC	World Health Organization Collaborating Centre
CDA	Clinical Document Architecture
MIME	Multipurpose Internet Mail Extensions
GP	General Practitioner
JAseHN	Joint Action supporting the eHealth Network
EU	European Union
AU	Africa Union
USAN	Union of South American Nations
EAEU	Eurasian Economic Union

CSME	Caribbean Single Market and Economy
POI	Point of Interest
IMEI	Mobile Station Equipment Identity
IMSI	International Mobile Subscriber Identity
UCE	Unsolicited Commercial E-mail
MAPS	Mail Abuse Prevention System
RBL	Real-time Blackhole List
CA	Certificate Authority or Certification Authority
her	Electronic Health Records
DNA	Deoxyribonucleic Acid
RNA	Ribonucleic Acid
UML	Unified Modelling Language
XACML	eXtensible Access Control Markup Language

Table 3: Glossary

4 Executive Summary

The need of a trusted environment in which only authorized users are permitted to access a system was of imperative importance since the early days of cloud computing. Even nowadays, a lot of users seem to be reluctant to store their personal data in the cloud. Especially, the users are hesitant to upload their sensitive data in the cloud and specifically the data related to bank accounts and the health care domain. Our goal is to enhance the access control mechanisms that can be used in the healthcare domain and raise security awareness. In this way, users' trust will be extended and their unwillingness to manage and share their health data through cloud-based applications, will be hopefully diminished. In this work, we present a context-aware security model which consists of classes and properties that can serve as background knowledge for creating and enforcing access control rules for electronic health records (EHRs). We consider two different layers of authorization control based on the current context: i) the Attribute Based Access Control (ABAC layer which permits or denies access and/or editing rights to (encrypted) EHRs; and ii) the Attribute-Based Encryption (ABE layer which handles the way sensitive data should be decrypted.

5 Introduction

A promising approach for alleviating the security risks associated with cloud computing is to define effective context-aware security controls for the sensitive data of cloud applications. The ASCLEPIOS project will hinge upon an adequate access control scheme, one that takes into account the inherently dynamic nature of cloud environments and captures the knowledge that lurks behind such a scheme (e.g. actions, subjects, locations, environmental attributes, etc.) This access control scheme calls for the incorporation of the notion of context in access control policies, i.e. the consideration of dynamically-changing contextual attributes that may characterise data accesses. Context can be perceived as *any information that can be used to characterize the situation of an entity (person, place, or object) that is considered relevant to the interaction between a user and an application, including the user and applications themselves* [1]. In fact, the use of contextual information makes it possible to apply access control policies by mainly considering the circumstances under which access requests to sensitive data, should be granted. This characteristic, which involves the development of a re-usable and generic context-aware security model, is an important part of our work, presented in this deliverable.

Access control protocols are responsible for deciding if a user has the right to execute a certain operation such as read or write, on a specific object. Objects can be a server, a service, an application, an entire relational database, a single row in a table or even an entire wide column in a NoSQL datastore. The user is considered as the active element and is called subject. A permission associates an object with an operation. Static access control models, usually, provide a list of permissions that each subject has on certain objects. Commonly used access control models are the Mandatory Access Control (MAC), the Discretionary Access Control (DAC) and the Role-Based Access Control (RBAC) [2]. All these models are known as identity-based access control models where user (subjects) and resources (objects) are identified by unique names [3]. In the literature, a fourth type has been identified, the ABAC [4] which is by nature dynamic. In ABAC, there are no static lists of permissions that associate subjects with objects, but instead there are “snapshots” of such associations that can be generated and dynamically change based on the current context. In ASCLEPIOS, the process of granting/denying access on data artefacts is based on dynamically changing parameters, thus we rely on an ABAC model, which goes beyond the traditional security models that are usually context insensitive. The context parameters are individual for every single user or access request, so for granting access it is necessary to regard the single user, the object that s/he is requesting to access and any external information that should be considered for enhancing the security.

In the ASCLEPIOS project, apart from ABAC, the extra ABE layer that considers additional important aspects of access control is used. More precisely, ABE in ASCLEPIOS is not used for directly encrypting/decrypting health records. Instead it is only used to protect the actual symmetric key used for encrypting users’ data. In case a user (e.g. a general practitioner) would like to access a patient’s Electronic Health Records (EHR), s/he should be able to transmit a request along with certain attributes (expressing the current context). In order to be successfully authorized these attributes should satisfy a policy, defined by the data owner beforehand. This ABE policy was used by the data owner to encrypt the symmetric key with which the sensitive medical data have been encrypted. Thus, a successful ABE authorization implies the decryption of a key, which then can be used for acquiring the plaintext of the encrypted medical records.

The ABAC and ABE layers, which are both based on policies, are combined in an innovative way and enforced in order to satisfy the need for advanced access control for cloud persisted health data. The ABAC layer provides a fine-grained access control by evaluating rules, while the ABE layer authorizes access by decrypting the symmetric key. The final result on this couple of layers is the decrypted data.

WP3 focuses on defining and evaluating contextual information, e.g., the identity of a user, his/her role, patterns of access, connection type etc. that should be considered before granting any data access request and to the attributes that characterise sensitivity levels of data. Therefore, the ASCLEPIOS context-aware security model conceptualises, through an appropriate vocabulary, all the facets, that must be taken under consideration during the development and enforcement of a data access control policy. The existence of a common vocabulary in a context model is of paramount importance for its functionality. This vocabulary is also extended with medical taxonomies that amplify its cohesiveness and reusability. Based on this vocabulary, enforcement rules can be modelled as the most elementary structural elements of policies. Indicatively, attributes that are organized in a hierarchical structure may include concepts related to: i) the device from which there is an access attempt, ii) the actor that tries to access the data (e.g. location, IP, role in the healthcare use case, etc.) and iii) historic data that reveal patterns of access (e.g. frequency, usual dates or hours of access, usual duration of access, previously accessed data, etc.). Such concepts along with a number of properties that interrelate them, serve as background knowledge for the ASCLEPIOS access control policies.

5.1 Objectives

The primary goal of this deliverable is to enhance the access control mechanisms that can be used in the healthcare domain and raise security awareness. The deliverable documents a context-aware security model which consists of classes and properties that can serve as background knowledge for creating access control rules for EHR. The authorization control hinges upon two different layers of authorization control: i) the ABAC layer which permits or denies access and/or editing rights to (encrypted) EHRs; and ii) the ABE layer, which handles the way sensitive data should be decrypted. To illustrate our model, we apply it to support secure access to Electronic Health Record (EHR) data.

Specifically, we relied on the existing context aware security model of the PaaSword H2020 project. This model was first conceived as quite generic for serving as background knowledge when creating (ABAC) access control rules in cloud applications. With this work, we try to extend it, considering three main dimensions: i) covering the additional peculiarities of the healthcare domain, ii) supporting multiple advanced authorization methods (i.e. both ABAC and ABE) and iii) creating security awareness. Thus, in this work we aim to extend PaaSword's context model by enriching its classes of Subject, Object and Connectivity.

Additionally, this deliverable undertakes the research and development of a policy model for formally describing dynamically-generated context-based access control policies, as well as static policies concerning the manner in which medical data can be decrypted according to the ABE paradigm. This policy model comprises two critical parts: i) an XACML-based part for defining declaratively authorization policies for permitting or denying access requests to sensitive data, in real-time; and ii) an ABE-oriented part for declaring the attribute-based dependency between actors' private keys and ciphertexts which upon matching are to be decrypted. The first part of this policy model encourages the separation of the access decisions from the points of use. This will pave the way for the development of an interpreter for dynamically interpreting authorization policies in healthcare scenarios and also for developing a healthcare-oriented context-aware ABAC enforcement mechanism. The latter part of this model will allow for the design and implementation of the ASCLEPIOS ABE mechanism.

5.2 Relationship to ASCLEPIOS Deliverables

The notion of context is considered an important element in the ASCLEPIOS approach. This deliverable documents the context-aware security model that constitutes the background knowledge for establishing authorization control over accessing EHR, based on ABAC and

ABE. Figure 1 illustrates the tasks documented in deliverable D3.1 and how the work documented herein feeds the development of the ASCLEPIOS model editor and interpretation mechanism (D3.2) as well as the context aware ABAC enforcement mechanism (D3.3).

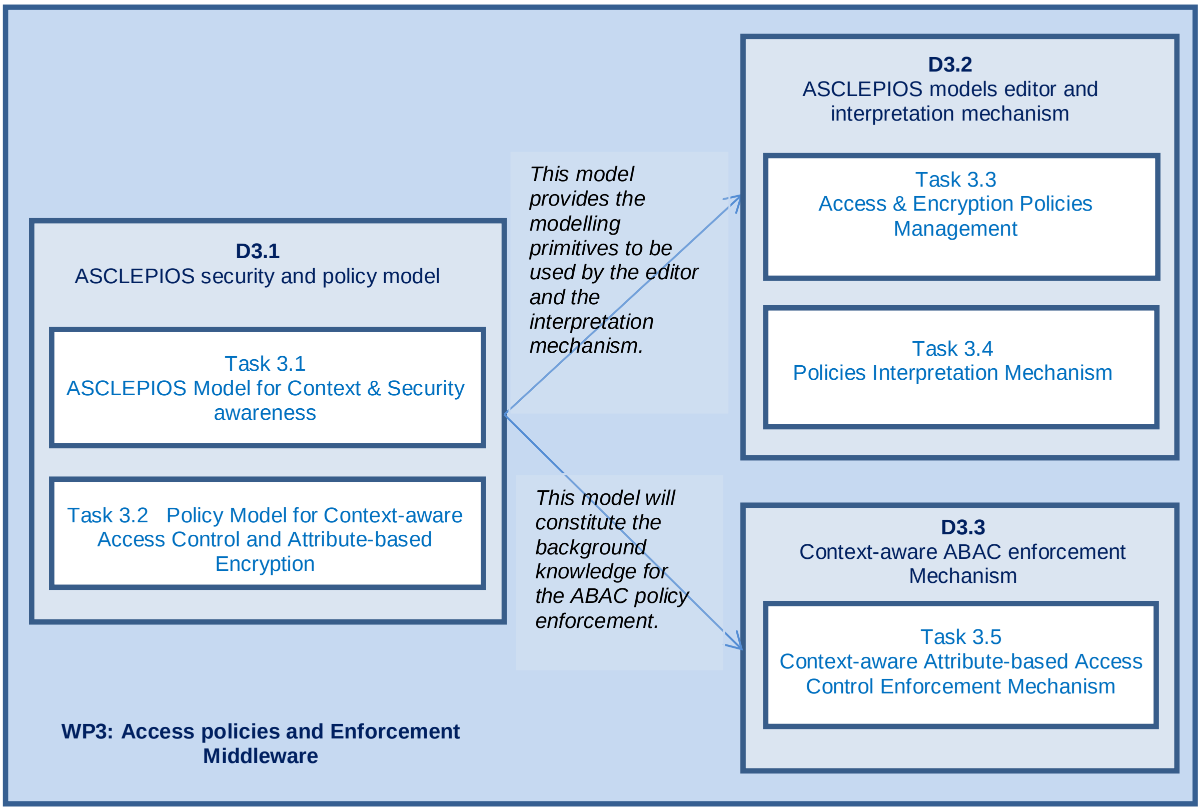

D3.1
ASCLEPIOS security and policy model
Task 3.1
ASCLEPIOS Model for Context & Security awareness
Task 3.2 Policy Model for Context-aware Access Control and Attribute-based Encryption
This model provides the modelling primitives to be used by the editor and the interpretation mechanism.
This model will constitute the background knowledge for the ABAC policy enforcement.
D3.2
ASCLEPIOS models editor and interpretation mechanism
Task 3.3
Access & Encryption Policies Management
Task 3.4
Policies Interpretation Mechanism
D3.3
Context-aware ABAC enforcement Mechanism
Task 3.5
Context-aware Attribute-based Access Control Enforcement Mechanism
WP3: Access policies and Enforcement Middleware

Figure 1: Deliverables of WP3 in ASCLEPIOS project

The outcomes of WP3 will drive the next work packages WP4 and WP5 in terms of Access Policies and Enforcement Middleware. The model presented herein will constitute the necessary background knowledge layer for enabling the ABAC and ABE paradigms in ASCLEPIOS. Specifically, policies will be determined through a number of attributes that specify valuable security-related details of the entity that is requesting access to sensitive data, the data itself and its ambient environment. Hence, the model will serve the development of the ASCLEPIOS Data Access Policies Interpretation and Enforcement mechanism.

5.3 Methodology and Deliverable Organization

To develop the ASCLEPIOS context-aware security model, we started by performing an extensive literature review of existing relevant frameworks and models. Moreover, we researched works that dealt with security of healthcare systems. We identified and collected attributes, characteristics, features, roles and any other entity that was highlighted in the literature and appeared relevant to security of healthcare systems. We synthesised our findings into a baseline ASCLEPIOS security model. Then, we presented this model to a group of healthcare experts from within and outside the ASCLEPIOS project and collected their feedback with respect both to the fit of the proposed model and its constituents, as well as to possible additional entities not included in the baseline model. Subsequently, a revised security model was developed and presented to the expert group for a second revision. Finally, we developed the policy model based on the model resulted from the PaaSword H2020 project, which has been revised and extended to fit the specific needs of healthcare systems. The policy model has been described and several examples of policy access rules have been defined. We discussed the policy model and examples with the expert group and we made the necessary revisions. This document presents the resulting security and policy models. These models will be utilised by the ASCLEPIOS software components and will be further validated and possibly revised during the course of the project.

This document begins with a table of contents (chapter 1). In chapter 2, lists of figures and tables are provided, while an additional list about status, change history and glossary is presented in chapter 3. Additionally, chapter 4 provides an executive summary, while chapter 5 contains an introduction that describes the development of the context aware security model along with the policy model. Chapter 6 reports on context models for ABE in the literature, while chapter 7 provides detailed information about ASCLEPIOS context aware security model. Chapter 8 presents the policies model. In addition to this, chapter 9 provides information about the exploitation authorization policies. The information presented in this document is summarized in the final chapter (chapter 10).

6 Context Models for ABE in the Literature

This section presents scientific works which propose contextual attributes using ABE that can be taken into account for the development of our context and policy model. Sahai Amit and Waters Brent [5] introduced the term ABE, in which an identity is a set of descriptive attributes. The authors proposed biometric information such as the iris, as attributes that could be used for authorizing access to a sensitive dataset. Bethencourt John et al. [6] introduced Ciphertext - Policy ABE - and proposed attributes such as the Name and the City of location of a person to determine access to a specific dataset. Muller Sascha et al. [7] considered, among others, as attribute if someone is underage or not. According to Moffat Steve et al. [8] the role, gender and country of a subject are relevant attributes. Wang Shangping et al. [9] in their approach, about searchable and revocable multi-data owner at scheme with hidden policy in cloud storage, proposed as attribute the role of Employee. As an example, if an employee is fired or promoted, the corresponding attribute needs to be updated. Premkamal P. K. et al. [10] considered as attribute, among others, the role of Researcher. Han Jinguang et al. [11] in their approach, according to fine-grained information flow control using attributes, proposed as attributes, among others, a subject's age and gender.

In the Healthcare domain, Lewko Allison et al. [12] proposed as attributes the role of Doctor and Researcher. In the example given, a party might want to share medical data only with a user who has the attribute of Doctor or the attribute of Researcher. Kumar Praveen et al. [13] analyzed an example in personal health information system where the patient would like to share his medical record only with the concerned doctor, concern department nurse, and insurance people. In their work, in case a doctor wishes to consult expert from a different hospital hospital, the doctor will authorize the access control to them. In this case the following roles exist: Patient, medical staff like Concerned Doctor, Concerned Department Nurse and Expert from another hospital, and the insurance people who have rights upon data and can access them. In addition to this, attributes are considered to be the patient's medical records which constitute sensitive data. Liu Zechao et al. [14] propose as attributes the roles of: Surgeon, Medical Researcher and Rehabilitation Doctor. As an example, in an e-Health system, a patient may like to share medical data with a user who has the attribute Surgeon issued by a hospital and the attribute Medical Researcher issued by a clinical research center. In addition to this, a patient should define an access policy as ("Surgeon" AND "Medical Researcher") before encrypting his/her data under this policy. In this scenario two authorities exist: the hospital and the clinical research center. In case of a surgeon's resignation from the hospital, s/he loses the attribute Surgeon and cannot decrypt previously shared data anymore. Apart from that, when the patient needs rehabilitation guidance, s/he needs to update the encrypted medical data to give Rehabilitation Doctor permission to access the data with the new policy ("Rehabilitation Doctor" AND "Medical Researcher").

Domingo – Ferrer Josep et al. [15] proposed as attributes, among others, the social security number, the age, the gender and the health condition of a subject. In an example given, a ninety-five years old female doctor who lives in a bad neighborhood is defined by the combination of these attributes. Zhang Leyou et al. [16] considered as subjects the Doctor, the Patient and the Patient's Family Members. In their approach, the patient intents to permit some doctors and family members to access his/her personal health record and simultaneously he may keep the medical condition secret from others. In another example given in the same approach, the access policy contains "cardiopathy" and "DC hospital". Attrapadung Nuttapong [17] in his approach, proposed three categories of sets/policies: Person:{Trainee, Doctor}, Place:{Paris, Zip:75001}, Content:'(Kidney and Disease) or Emergency', with a "composition policy" such as 'Person or (Place and Content)', which plays the role of concluding the whole policy. A ciphertext could be associated to Person: 'Senior and Doctor', Place: 'Paris or London', Content: {Kidney, Disease, Cancer}. Hong

Jiaojiao et al. [18] propose as attributes, among others, the role of: Physician, Nurse and Researcher. Xu Qian et al. [19] consider, among others, as attributes the role of Doctor and the role of Researcher. In the example given, they proposed the following context expression: (“Doctor” or “Researcher”). Li Qi et al. [20] considered, among others, as attribute the role of Doctor and the role of Researcher. Liang Pengfei et al. [21] in their approach, proposed as attributes, among others, the doctor’s specialty and the associated hospital. Specifically, they proposed the following context expression: (“Cardiologist” and (“Hospital A” or “Hospital B”)).

7 ASCLEPIOS context-aware security model

Our work relies on the existing context-aware security model of PaaSword which we extend by enriching its classes of Subject, Object and Connectivity. More precisely the original PaaSword context aware security model is composed by five main classes. These classes which described below, are subclasses of the class SecurityContextElement that refers to several contextual attributes that may be associated with the subject and/or the objects of a request as well as with the request itself [22][23].

- **Object**: contains types of sensitive data;
- **Subject**: represents a requestor who intends to access the object content. The requestor can be a person, an organization, a group or a software;
- **Location**: tracks the exact location of a subject who requests access to data;
- **DateTime**: tracks the exact date and time of a subject who requests access to data;
- **Connectivity**: tracks the device type, the connection type, the connection security and the connection metrics.

The main classes of PaaSword context-aware security model are shown in the security element overview, in the Figure 2, as part of the ASCLEPIOS project.

7.1 Security Context Element Overview

The ASCLEPIOS context model formally describes classes and properties with respect to: i) associations between types of access to sensitive data and situations under which this access should be permitted; ii) cryptographically matching policies between ABE private keys and ciphertexts for permitting decryption of sensitive data; iii) concepts that capture and highlight possible cyber security threats for enhancing the security awareness of actors in hospitals and care centres. This model constitutes the necessary background knowledge layer for enabling the ABAC and ABE paradigms. Specifically, such situations and policies are determined through a number of attributes that specify valuable security-related details of the entity that is requesting access to sensitive data, the data itself and its ambient environment. Such attributes are organized in a hierarchical structure that may include concepts related to: i) the device from which there is an access attempt, ii) the actor that tries to access the data (e.g. location, IP, role in the healthcare use case, etc.) and iii) historic data that reveal patterns of access (e.g. frequency, usual dates or hours of access, usual duration of access, previously accessed data, etc.). We reuse similar context models from the cloud security domain (e.g. PaaS Security Context model) and extend them in order to cope with concepts, challenges and standards from the healthcare domain.

This context-aware security model is based on the vocabulary of PaaSword context model which is the base for a generic and structured context model. We managed to extend it such as to serve the quite demanding, structured and specialized field of healthcare. Our context model has enriched the classes of Subject, Object and Connectivity to consider important concepts from the medical sector. In the class Subject, its subclasses Person, Organization and Authentication Method were enriched. The class Person was enriched with classes: Technical Staff, Contact Person / Legal Guardian, Administrative Staff, Medical Force, Researcher, Employer and Patient. The class Organization was enriched with classes: Insurance Company, GP (General Practitioner's) Office, Hospital, Diagnostic Centre and Research Institution. In class Authentication Method a subclass Biometric Information was added. In class Object, a subclass Medical Artefact was created. The class Medical Artefact has the following subclasses: EHR, Medical Process and Medical Report. In the subclass Security Protocol of the class Connectivity, the subclass Security Protocol Certificate was added. The class Connectivity was enriched so as to enhance the model's security and safety. Figure 2 demonstrates the enhancements of our model (dark blue colour denotes new classes), enriched with external medical ontologies (green colour), to the PaaSword context model (light blue colour). Figure 6 depicts precisely the enhancements of class Object. Finally, Figure 9 demonstrates analytically the enhancements of class Subject.

We provide an elaboration of ASCLEPIOS model in the form of facets. For each of these context model facets, we present an overview (i.e. class/sub-class) diagram, a list of its core concepts and properties (in tabular format), and a UML class diagram that formalises it. In Table 4, we provide the legend for the Overview and the UML Class diagrams that are used in the following sub-sections.

Class	Class defined in the PaaSword Context-aware Security Model. It might be a subclass of another Class. (Appears in Overview diagrams)
Class	Class defined in the ASCLEPIOS Context-aware Security Model. It might be a subclass of another Class. (Appears in Overview diagrams)
Class	Class defined in the ASCLEPIOS Context-aware Security Model. It is a Class imported directly from an external ontology. In our case this external class represents the imported medical taxonomies. (Appears in Overview diagrams)
SubClass of ——▷	Captures a subClassOf relation. The arrowhead indicates the parent class. (Appears in Overview diagrams)
Property ——►	Captures a named property between two Classes or between two Individuals. The arrow starts from the domain Class and the arrowhead indicates the range of this property. (Appears in Overview diagrams)
Local Class{prefix} property1{prefix}:External Class property2{prefix}:datatype	Class defined in the *ASCLEPIOS Context-aware Security Model*. It might be a subclass of another Local class. (Appears in UML class diagrams)
Local Class{prefix} <External Parent Class property1{prefix}:External Class property2{prefix}:datatype	Class defined in the *ASCLEPIOS Context-aware Security Model*. It is a subclass of an external class (i.e. imported from external ontology). It might also be subclass of Local class too. (Appears in UML class diagrams)
External Class{prefix} property1{prefix}:External Class property2{prefix}:datatype	Class imported directly from an external ontology. It is depicted in the UML diagrams in order to present the reuse of available vocabularies in the *ASCLEPIOS Context-aware Security Model*. (Appears in UML class diagrams)
Individual{prefix}:Any Class	Individual (i.e. instance) of a Local or External class. (in UML class diagrams)
——▷	Captures a subClassOf relation. The arrowhead indicates the parent class. (Appears in UML class diagrams)

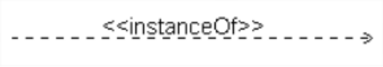
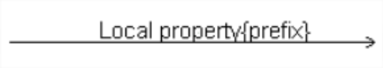
	Captures an instance-of relation. The arrowhead indicates the Class whereas the arrow starts from the Individual. (Appears in UML class diagrams)
	Captures a property between two Classes or between two Individuals. The arrow starts from the domain Class and the arrowhead indicates the range of this property. (Appears in UML class diagrams)

Table 4: Legend of the Overview and UML diagrams

7.2 Security Context Element Details

The security context element [24] (Figure 2) refers to the following five top-level concepts:

- **Location**
- **DateTime**
- **Connectivity**
- **Object**
- **Subject**

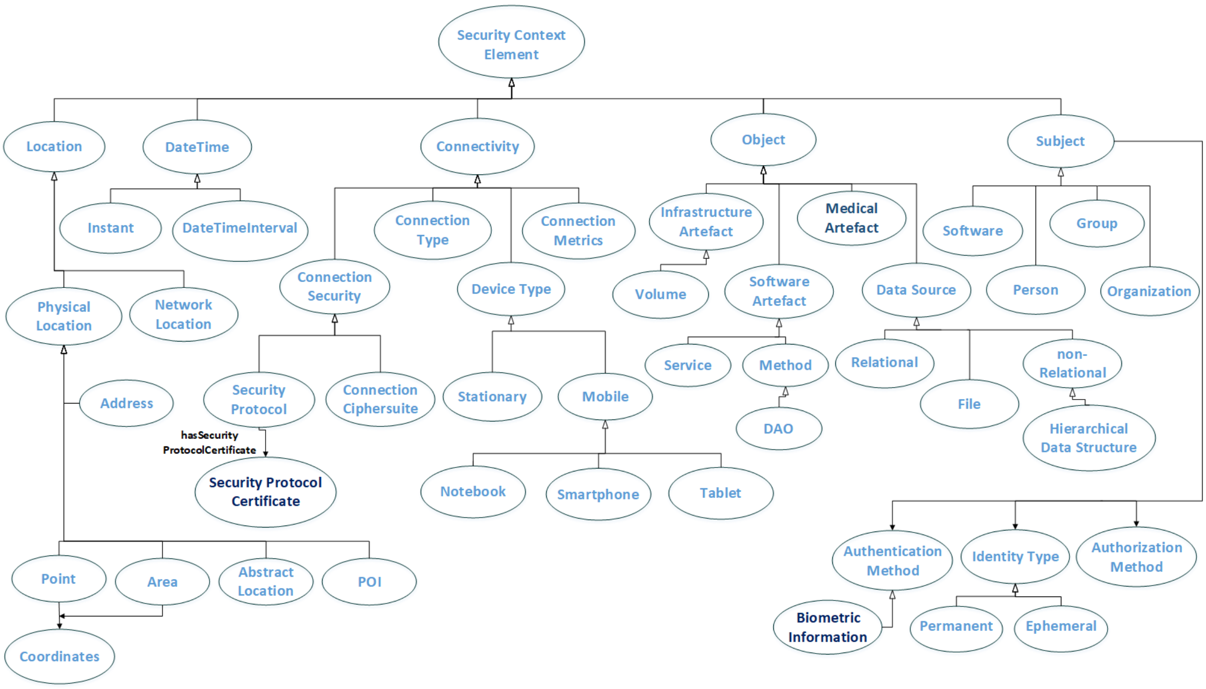


Figure 2: ASCLEPIOS Security Context Element overview diagram

For each of these top-level core classes we provide a description, their respective subclasses and their main associated properties. In the following description, we denote in the form of *Parent Class/Child 1 Class/…/Child N Class* the hierarchy details to reveal the full path of each class in our model. Classes or properties that are re-used or extended from external ontologies, are denoted accordingly. We note that in cases that we extend already existing classes then the description is also adjusted using the original description of the relevant concept. In Table 5, the details of the Location context element are presented.

Class Path (Hierarchically)	Class Description	Property / Description

Location	This class describes a physical and/or a network location where data are stored or from which a particular entity is requesting to access data.	
Location/ **PhysicalLocation**	A physical location is a point or area of interest where data are stored or from which a particular entity is requesting to access data. Physical locations might involve: an address, a geographical position, an area, an abstract location and/or a Point of Interest. Extends http://purl.org/goodrelations/v1#Location , http://schema.org/Place	
Location / PhysicalLocation / **Point**	A specific spot where data is stored or from which a particular entity is requesting to access data.	**hasPointCoordinates**: This property associates a Point with the specific Coordinates class (outlined in the next row).
Location / PhysicalLocation / Point / **Coordinates**	Refers to the positioning of an entity using the geographic coordinate system (we note that this class is not a subclass of Point, but we include its details here because they can be used in several security context elements). Imported from http://schema.org/GeoCoordinates	**Latitude**: The latitude of a location in decimal format or in degree/minute/second (DMS) format. The range of this property is the class: http://schema.org/latitude **Longitude**: The longitude of a location in decimal format or in degree/minute/second (DMS) format. The range of this property is the class: http://schema.org/longitude **Elevation**: The elevation of a physical location (either a number or text). The range of this property is the class: http://schema.org/elevation
Location / PhysicalLocation / **Area**	This class describes a geographical region from which a data access request can originate.	**hasAreaCoordinates**: This property associates an Area with specific coordinates (class). **hasCircularRadius**: This property declares the diameter of a circular area with a centre pointed by the hasAreaCoordinates property. (e.g. diameter 1km) **hasRectangularRangeWidth**: This property describes the width of the Rectangular area with top-left corner pointed by the hasAreaCoordinates property (e.g. 1km) **hasRectangularRangeHeight**: This property describes the height of the Rectangular area with top-left corner pointed by the hasAreaCoordinates property (e.g. 2Km)
Location / PhysicalLocation / **Address**	Physical address where data are stored or from which a particular entity is requesting to access sensitive data. Extends https://schema.org/PostalAddress	**hasAddressCountry**: The range of this property is the class: https://schema.org/addressCountry **hasAddressLocality**: The range of this property is the class: https://schema.org/addressLocality

		hasAddressRegion: The range of this property is the class: https://schema.org/addressRegion
		hasStreetAddress: The range of this property is the class: https://schema.org/streetAddress
		hasBuildingNumber: This property associates an Address with a number that identifies a building.
		hasRoomNumber: This property associates an Address with a number that identifies a room.
		hasFloorNumber: This property associates an Address with a number that identifies a floor.
		refersToContinentalUnion: The range of this property may include: EU, Africa Union (AU), Union of South American Nations (USAN)
		refersToEconomicUnion: The range of this property may include the individuals: EU, Eurasian Economic Union (EAEU), Mercosur, Caribbean Single Market and Economy (CSME)
Location / PhysicalLocation / **AbstractLocation**	Conceptual characterization of a physical location (e.g. non-organization premises, building, room, section, department etc.)	**hasName**: This property associates an Abstract Location with a string that denotes its name.
		hasArea: This property associates an Abstract Location with the class Area.
Location / PhysicalLocation / **POI**	Points of interest (POI) that might be meaningful for controlling access to sensitive data.	**hasName**: This property associates a POI with a string that denotes its name.
		hasAddress: This property associates a POI with the class Address.
Location / **NetworkLocation**	An identifier for a node or network telecommunication interface from which a particular entity is requesting to access data.	**hasIPAddress**: This property associates a Network Location with a string that identifies its IP address.
		hasDomain: This property associates a Network Location with a string that identifies its domain.
		hasSubnet: This property associates a Network Location with a string that identifies its subnet mask.
		hasZone: This property associates a Network Location with a string that identifies its zone.
		hasIPAddressRange: This property associates a Network Location with a string that identifies the IP address range of the network.
		hasPort: This property associates a Network Location of server with a positive integer that identifies its port number.

Table 5: Details of the Location context element

In the following Figure 3, we provide the UML Class diagram that provides the classes, subclasses object and data properties of the top-level class of the Location Security Context Element [25].

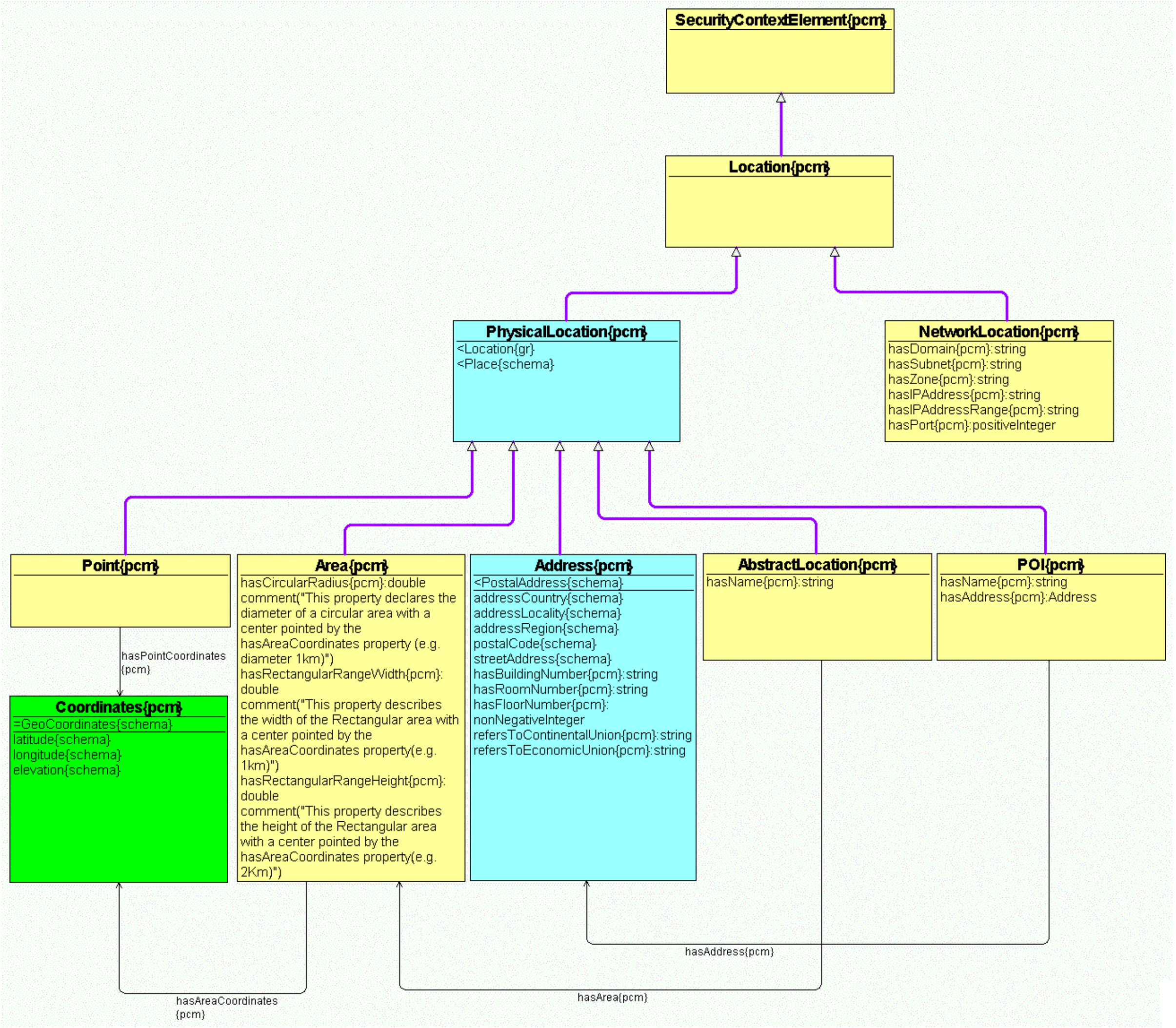

SecurityContextElement{pcm}
Location{pcm}
PhysicalLocation{pcm}
<Location{gr}
<Place{schema}
NetworkLocation{pcm}
hasDomain{pcm}:string
hasSubnet{pcm}:string
hasZone{pcm}:string
hasIPAddress{pcm}:string
hasIPAddressRange{pcm}:string
hasPort{pcm}:positiveInteger
Point{pcm}
hasPointCoordinates {pcm}
Coordinates{pcm}
=GeoCoordinates{schema}
latitude{schema}
longitude{schema}
elevation{schema}
Area{pcm}
hasCircularRadius{pcm}:double
comment("This property declares the diameter of a circular area with a center pointed by the hasAreaCoordinates property (e.g. diameter 1km)")
hasRectangularRangeWidth{pcm}: double
comment("This property describes the width of the Rectangular area with a center pointed by the hasAreaCoordinates property(e.g. 1km)")
hasRectangularRangeHeight{pcm}: double
comment("This property describes the height of the Rectangular area with a center pointed by the hasAreaCoordinates property(e.g. 2Km)")
Address{pcm}
<PostalAddress{schema}
addressCountry{schema}
addressLocality{schema}
addressRegion{schema}
postalCode{schema}
streetAddress{schema}
hasBuildingNumber{pcm}:string
hasRoomNumber{pcm}:string
hasFloorNumber{pcm}: nonNegativeInteger
refersToContinentalUnion{pcm}:string
refersToEconomicUnion{pcm}:string
AbstractLocation{pcm}
hasName{pcm}:string
POI{pcm}
hasName{pcm}:string
hasAddress{pcm}:Address
hasAddress{pcm}
hasAreaCoordinates {pcm}
hasArea{pcm}

Figure 3: UML Class diagram for the Location context element

In Table 6, the details of the DateTime context element are presented.

Class Path (Hierarchically)	Class Description	Property / Description
DateTime	This class specifies the datetime at which an access request takes place or a data artefact is stored.	**before**: This property associates a DateTime instance with another earlier DateTime instance. *Extends* owl-time: before
		after: This property associates a DateTime instance with another later DateTime instance. Extends owl-time: after
		hasTimezone: This property associates a DateTime with a string that corresponds to the relevant time zone.
DateTime / **Instant**	A precise point in time used to specify the datetime at which an access request takes place or a data artefact is stored. Extends owl-time: instant	**hasYear**: This property associates an Instant with a positive integer that corresponds to a year.
		hasMonth: This property associates an Instant with a positive integer that corresponds to a

		month.
		hasDay: This property associates an Instant with a positive integer that corresponds to a day.
		hasHour: This property associates an Instant with a non- negative integer that corresponds to an hour.
		hasMinute: This property associates an Instant with a non- negative integer that corresponds to a minute.
		hasSecond: This property associates an Instant with a non- negative integer that corresponds to a second.
DateTime / **DateTimeInterval**	A period of time bounded by two xsd:dateTime or by two gr:DaysOfWeek instances (e.g. gr:Monday).	**hasBeginning**: This property associates the DateTimeInterval with an xsd:dateTime that denotes the start of an interval period of time.
		hasBeginningDay: This property associates the DateTimeInterval with the gr:DayOfWeek that denotes the start of an interval period of time.
		hasEnd: This property associates the DateTimeInterval with an xsd:dateTime that denotes the end of an interval period of time.
		hasEndDay: This property associates the DateTimeInterval with the gr:DayOfWeek that denotes the end of an interval period of time.

Table 6: Details of the DateTime context element

The details of the DateTime context element are captured in the following UML Class diagram presented in Figure 4 [26].

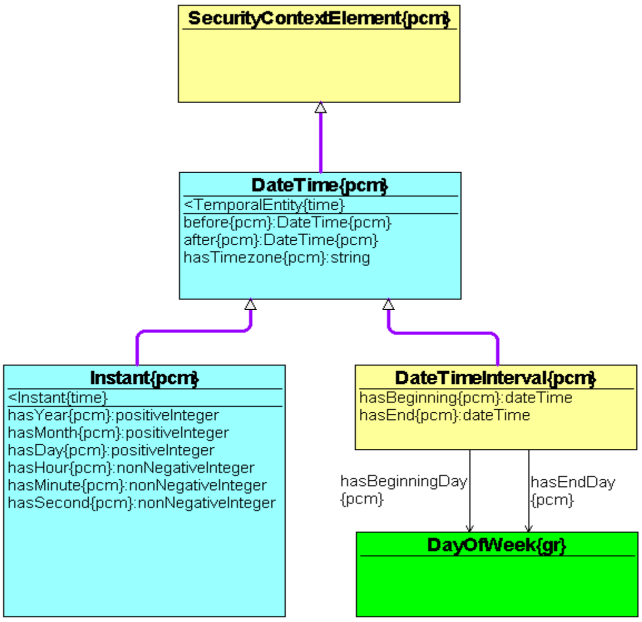

SecurityContextElement{pcm}
DateTime{pcm}
<TemporalEntity{time}
before{pcm}:DateTime{pcm}
after{pcm}:DateTime{pcm}
hasTimezone{pcm}:string
Instant{pcm}
<Instant{time}
hasYear{pcm}:positiveInteger
hasMonth{pcm}:positiveInteger
hasDay{pcm}:positiveInteger
hasHour{pcm}:nonNegativeInteger
hasMinute{pcm}:nonNegativeInteger
hasSecond{pcm}:nonNegativeInteger
DateTimeInterval{pcm}
hasBeginning{pcm}:dateTime
hasEnd{pcm}:dateTime
hasBeginningDay {pcm}
hasEndDay {pcm}
DayOfWeek{gr}

Figure 4: UML Class Diagram for the DateTime context element

In Table 7, the details of the Connectivity context element are presented.

Class Path (Hierarchically)	Class Description	Property / Description

Connectivity	This class captures the information related to the connection used by the subject for accessing sensitive data.	
Connectivity / **DeviceType**	This class describes a device used when requesting access to sensitive data. Extends http://purl.org/goodrelations/v1#ProductOrService	
Connectivity / DeviceType / **Stationary**	This class refers to an immobile device used when requesting access to sensitive data.	**hasStationaryOS**: This property associates a Stationary device with a string that denotes the operating system used by that device. **supportsEncryptedStorageStationary**: This property associates a Stationary device with a string that denotes the kind (if any) of encrypted storage is supported by that device.
Connectivity / DeviceType / **Mobile**	This class refers to a portable device used when requesting access to sensitive data.	**hasMobileOS**: This property associates a Mobile device with a string that denotes the operating system used by that device. **hasIMEI**: This property associates a Mobile device with a string that denotes the international mobile station equipment identity (IMEI). **hasIMSI**: This property associates a Mobile device with a string that denotes the international mobile subscriber identity (IMSI) of that device. **supportsEncryptedStorageMobile**: This property associates a Mobile device with a string that denotes what kind (if any) of encrypted storage is supported by that device.
Connectivity / DeviceType / Mobile / **Notebook**	This class refers to laptop computers used to access sensitive data elements.	
Connectivity / DeviceType / Mobile / **Tablet**	This class refers to mobile computers that are primarily operated by touching the screen and are used to access sensitive data elements.	
Connectivity / DeviceType / Mobile / **Smartphone**	This class refers to mobile phones used to access sensitive data elements, which combine features of a personal computer with other features useful for handheld use.	
Connectivity / **ConnectionType**	This class refers to different ways of transmitting an access request (e.g. LTE, 3G, WiFi, Cable, Satellite).	**hasTelecommunicationsProvider**: This property associates a ConnectionType with a string that denotes a telecommunication provider. **hasConnectionMetric**: This property associates a ConnectionType with a

		ConnectionMetric.
Connectivity / **ConnectionMetric**	This class provides quantitative characteristics of the connection type used for accessing sensitive data.	**hasUploadRate**: This property associates a ConnectionType with a non-negative number that denotes the upload rate for a certain network connection.
		hasDownloadRate: This property associates a ConnectionType with a non-negative number that denotes the download rate for a certain network connection.
		hasMetricUnit: This property associates a Connection Metric with a string that denotes the measurement unit.
Connectivity / **ConnectionSecurity**	This class provides details on the level of security in the established connection for accessing sensitive data.	**hasHostedIP**: This property represents a network host which is a computer or other device connected to a computer network. A host may work as a server offering information resources, services, and applications to users or other hosts on the network.
		hasWhiteListRange: This property refers to the range of a white list which is a list of e-mail addresses or domain names which should are consider safe.
		hasBlackListRange: This property refers to the range of a blackhole list, which is sometimes simply referred to as a blacklist, which is the publication of a group of ISP addresses known to be unsafe.
Connectivity / ConnectionSecurity / **SecurityProtocol**	This class reveals the security technology or protocol adopted for establishing an encrypted link between a server and a client (e.g. None, SSLv1v3, TLS v1.0-1.2, IPSec, SSH, S/MIME). None, SSLv1v3, TLS v1.0-1.2, IPSec, SSH, S/MIME).	**hasSecurityProtocolImplementation:** This property associates a Security Protocol with a string that denotes its implementation type (e.g OpenSSL).
Connectivity / ConnectionSecurity / SecurityProtocol / **SecurityProtocolCertificate**	This class checks the Security Protocol Certificates for enabling or denying the access control in the system. A certificate authority or certification authority (CA) is an entity that issues digital certificates. A digital certificate certifies the ownership of a public key by the named subject of the certificate.	**hasSecurityProtocolCertificate**: This is an object property which has as domain the Security Protocol class and as range the Security Protocol Certificate class.

Connectivity / ConnectionSecurity / **ConnectionCiphersuite**	This class refers to the ciphersuite used for establishing a secure connection (e.g. TLS_PSK_WITH_AES_128_GCM_SHA256).	

Table 7: Details of the Connectivity context element

The details of the Connectivity context element are captured in the following UML Class diagram presented in Figure 5 [27].

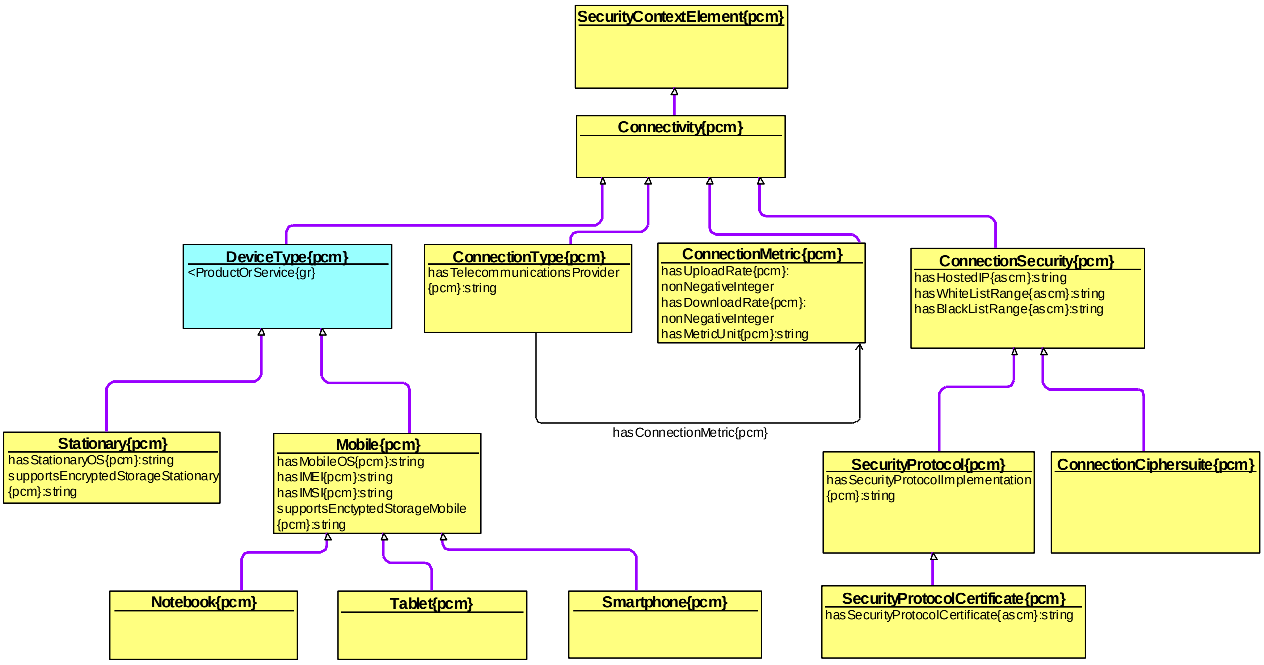

SecurityContextElement{pcm}
Connectivity{pcm}
DeviceType{pcm}
<ProductOrService{gr}
ConnectionType{pcm}
hasTelecommunicationsProvider {pcm}:string
ConnectionMetric{pcm}
hasUploadRate{pcm}: nonNegativeInteger
hasDownloadRate{pcm}: nonNegativeInteger
hasMetricUnit{pcm}:string
ConnectionSecurity{pcm}
hasHostedIP{ascm}:string
hasWhiteListRange{ascm}:string
hasBlackListRange{ascm}:string
hasConnectionMetric{pcm}
Stationary{pcm}
hasStationaryOS{pcm}:string
supportsEncryptedStorageStationary {pcm}:string
Mobile{pcm}
hasMobileOS{pcm}:string
hasIMEI{pcm}:string
hasIMSI{pcm}:string
supportsEnctyptedStorageMobile {pcm}:string
SecurityProtocol{pcm}
hasSecurityProtocolImplementation {pcm}:string
ConnectionCiphersuite{pcm}
Notebook{pcm}
Tablet{pcm}
Smartphone{pcm}
SecurityProtocolCertificate{pcm}
hasSecurityProtocolCertificate{ascm}:string

Figure 5: UML Class Diagram for the Connectivity context element

In Table 8 and Figure 6 [28], the details of the Object context element are presented. The class Object is extended by our context model by the main class Medical Artefact which is analysed to the three main classes EHR, Medical Process and Medical Report. We note that the following classes: Environment, Genetics, History, Treatment, Encounters, Diagnostics, Diagnoses, Socioeconomic and Symptoms, Lifestyle, Social Network existed in the proposed types of data of an EHR by Weber et al [29].

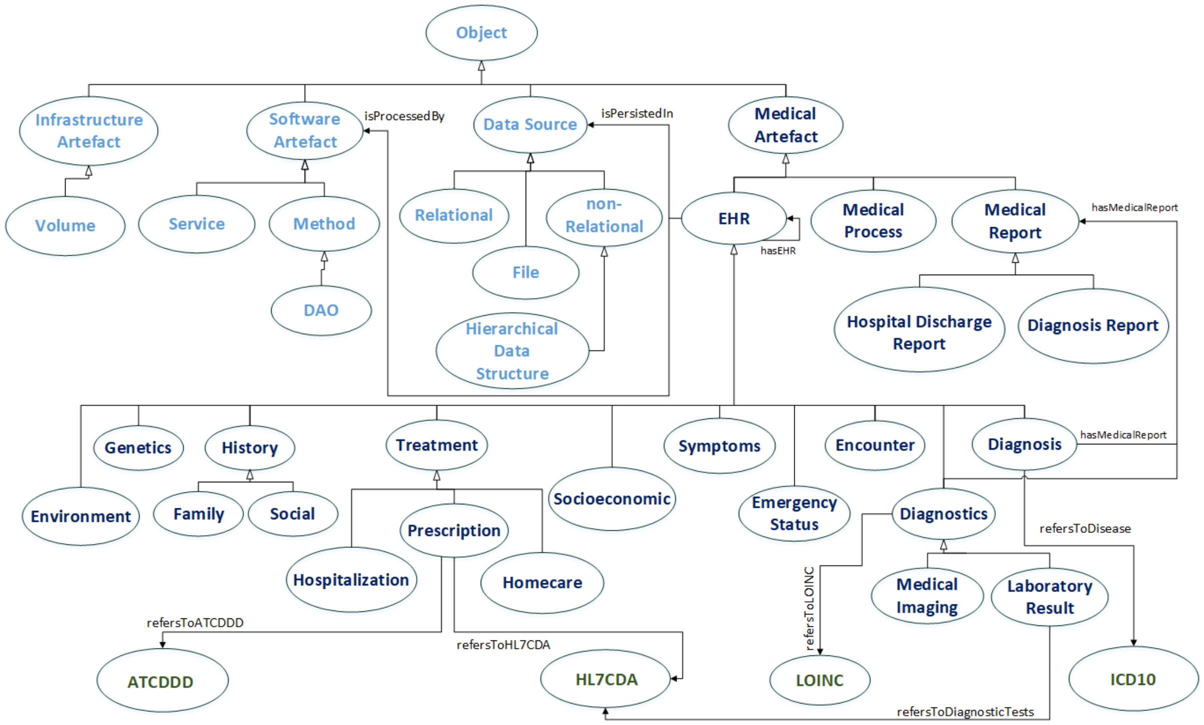


Figure 6: Extension of Security Context Element Object overview diagram

Class Path (Hierarchically)	Class Description	Property / Description
Object	This class refers to any kind of artefact that should be protected based on their sensitivity levels. These artefacts may refer to relational or other databases, files, software artefacts that manage sensitive data or even infrastructure artefacts used.	**hasTarget**: This property associates an *Object* and any of its subclasses with a string that denotes its access identifier.
Object / **DataSource**	This class refers to any sensitive data entities stored in schema-based or schema-less databases that should be protected from unauthorised access.	
Object / DataSource / **Relational**	This class refers to accessible data entities that exist in relational databases, i.e. structured to recognise relations between stored items of information (i.e. Database Table, Table Row, Table Column, and Table Cell).	**hasRelationalType**: This property associates a Relational instance with a string that denotes the type of the relational database element (e.g. RDBMS, NewSQL).
Object / DataSource / **non-Relational**	This class refers to accessible data entities that exist in non-relational (schema-less) databases (i.e. Document, Graph, Key-Value, Wide Column, Time-Series).	**hasNonRelationalType**: This property associates non-Relational with a string that denotes the type of the non-relational database element.
Object / DataSource / non-Relational / **HierarchicalDataStructure**	This class refers to hierarchical data formats used for encoding information in both human and machine-readable text (e.g. json).	**isChildOf**: This property associates a *Hierarchical Data Structure* instance with its parents forming a hierarchical structure among class instances.

Object / DataSource / **File**	This class refers to any kind of sensitive information that is stored in a file based system.	**hasFileType**: This property associates a File instance with a string that denotes the type of file that is considered sensitive (e.g. Local, Distributed). **hasFileName**: This property associates a *File* instance with a string that denotes the name of file that is considered sensitive. **hasFilePath**: This property associates a *File* instance with a string that denotes the path where the file can be retrieved.
Object/ **SoftwareArtefact**	This class refers to any computerised medium that manages or gives access to sensitive data. It is a subclass of: http://purl.org/goodrelations/v1#ProductOrService , http://schema.org/Product	
Object/SoftwareArtefact/ **Service**	This class refers to a service provided by an organization that may allow the manipulation of data artefacts. Extends: http://schema.org/Service	**category**: It associates the Service with a Category class. The range of this property is the class: http://schema.org/category
Object/SoftwareArtefact/ **Method**	This class refers to a programmatic method used for accessing or managing data artefacts.	**hasName**: This property associates a Method instance with a string that denotes its linguistic identifier.
Object/SoftwareArtefact / Method/**DAO**	This class refers to a *Data Access Object*, an object that provides an abstract interface to some type of database or other persistence mechanism.	
Object/ **InfrastructureArtefact**	This class refers to any hardware artefact used for storing and managing sensitive data. It is a subclass of: http://purl.org/goodrelations/v1#ProductOrService , http://schema.org/Product	
Object/ InfrastructureArtefact/ **Volume**	This class refers to a single accessible storage area.	**hasVolumeType**: This property associates a *Volume* instance with a string that denotes the kind of volume that is used for storing sensitive data (e.g. block storage, object storage). **isVolumeEncrypted**: This property associates a *Volume* instance with a Boolean value that denotes whether or not the volume is encrypted. **volumeInputSpeed**: This property associates a *Volume* instance with a string that denotes the input speed and its unit of measurement (e.g. 85000 IOPS).

		volumeOutputSpeed: This property associates a *Volume* instance with a string that denotes the output speed and its unit of measurement (e.g. 60000 IOPS)
Object/**MedicalArtefact**	This class refers to any medical sensitive data entities stored in schema-based or schema-less databases that should be protected from unauthorised access.	
Object/MedicalArtefact/ **MedicalProcess**	This class represents the administrative medical process of the medical condition of each patient during his or her treatment.	
Object/MedicalArtefact/ **MedicalReport**	This class represents the *Medical Report*. *Medical Report* is the report of the results of a medical examination of a patient.	
Object/MedicalArtefact/ MedicalReport/ **HospitalDischargeReport**	This class represents a clinical report prepared by a physician or other health professional at the end of a hospitalization or after a series of treatments.	
Object/MedicalArtefact/ MedicalReport/ **DiagnosisReport**	This class refers to the results of a medical examination of a certain patient.	
Object/MedicalArtefact/ **EHR**	This class represents the patient's Electronic Health Records (*EHR*). *EHR* represents a digital collection of medical information about a person. It includes information about a patient's health history, such as diagnoses, medicines received, tests, allergies, immunizations, and treatment plans.	**hasEHR**: This property has as domain and as range the class *EHR*. Its purpose is to model the complex structure of an EHR object than may contain other *EHR* objects in a nested manner.
Object/MedicalArtefact/ EHR / **Environment**	This is a subclass of *EHR* and refers to the patient's *Environment*, described as the conditions in which a person lives and operates.	
Object/MedicalArtefact/ EHR / **Genetics**	This is a subclass of *EHR* and refers to information about patient's genetics data, i.e. personal data related to the inherited or acquired genetic characteristics of a person which can be retrieved through the analysis of a biological sample (e.g. chromosomal, deoxyribonucleic acid (DNA) or ribonucleic acid (RNA) analysis).	
Object/MedicalArtefact/ EHR / **History**	This is a subclass of *EHR* and refers to information that may directly or indirectly affect the current or future health condition of the patient.	
Object/MedicalArtefact/ EHR / History / **Family**	This is a subclass of *History* and refers to the past data about medical or mental health conditions in patient's family members (e.g. cardiovascular diseases, autoimmune disorders, mental disorders, diabetes, cancer).	

Object/MedicalArtefact/ EHR/History/ **Social**	This is a subclass of *History* and refers to the patients' records which are associated to the patient's social life and behaviour, including substances abuse (e.g. alcohol, tobacco, illicit drugs), occupation, sexual preference, imprisonment, travel, exercise, diet.	
Object/MedicalArtefact/ EHR/**Treatment**	This is a subclass of *EHR* and refers to information about medical advices, operations or in general help received by a general practitioner or any other health professional.	**hasStartDate**: This is an object property with a range the *DateTime/Instant* class of our model. It represents the exact date of the beginning of the patient's treatment.
		hasEndDate: This is an object property with a range the *DateTime/Instant* class of our model. It represents the exact date of that a patient's treatment is completed.
		hasDuration: This data property associates the *Treatment* class with a positive integer that represents the duration of the patient's treatment in days.
Object/MedicalArtefact/ EHR/Treatment/ **Hospitalization**	This is a subclass of *Treatment* and refers to details of a patient's treatment that was offered through a public or private hospital.	
Object/MedicalArtefact/ EHR/Treatment/ **Prescription**	This is a subclass of *Treatment* and refers to any written instruction, provided by a physician or other qualified healthcare practitioner that authorizes a patient to be issued with a medicine or treatment.	**refersToATCDDD**: This object property associates the class Prescription with the taxonomy *ATC/DDD* (Anatomical Therapeutic Chemical Classification System / Defined Daily Dose) which is a standard for international drug utilization monitoring and research [30].
		hasATCCode: This object property associates the class Prescription with the ATC code that refers to the several classification levels provided by the *ATC/DDD* Taxonomy (e.g. G04BX15).
		hasName: This object property associates the class Prescription with the ATC Name as a descriptive title of the chemical substance used in a certain prescription (e.g. pentosan polysulfate sodium).

		hasDDD: This object property associates the class Prescription with the Defined Daily Dose of the *ATC/DDD* Taxonomy, for describing the dose (e.g. 0.3), defined in a prescription.
		hasUnit: This object property associates the class Prescription with the Unit for defining the dose unit in a prescription (e.g. milligram, millilitre etc.).
		hasAdministrationRoute:This object property associates the class *Prescription* with the Administration Route of the *ATC/DDD* taxonomy for describing how a prescribed drug should be consumed (e.g. inhalation, nasal, oral etc.).
		refersToHL7CDA: This property associates the class *Prescription* with the taxonomy Health Level Seven (HL7) Clinical Document Architecture (CDA) Release 2 Level 3 and Level 1 (PDF7/A) [31]. CDA is a document markup standard that specifies the structure and semantics of "clinical documents" for the purpose of exchange between healthcare providers and patients and developed by HL7.
		hasPrescriptionAuthentication: This property associates the class *Prescription* with the *Authentication Of The Prescription* class of the HL7 CDA taxonomy [32] in order to describes information such as: Prescription ID and Issue Date.
		hasIdentificationOfThePrescribedProduct: This property associates the class *Prescription* with the class *Identification Of The Prescribed Product* of HL7 CDA [32] in order to provide prescribed product information such as: Name, Identifier and Strength of the item.

		hasPrescriptionInformation: This property associates the class Prescription with the class *Prescription* Information of HL7 CDA [32] in order to provide with information such as: Pharmaceutical formulation (e.g. tablet, solution, ointment).
Object/MedicalArtefact/ EHR/Treatment/ **Homecare**	This is a subclass of *Treatment* and refers to any details about a patient's treatment that was provided in patient's home.	
Object/MedicalArtefact/ EHR/**Socioeconomic**	This is a subclass of *EHR* and refers to any information relevant to a patient's social or economic status.	
Object/MedicalArtefact/ EHR/**Symptoms**	This is a subclass of *EHR* and refers to the physical or mental features which are regarded as indicative of a health-related condition.	
Object/MedicalArtefact/ EHR/**EmergencyStatus**	This is a subclass of *EHR* and refers to any situation that poses or posed an immediate risk to patient's health or life.	
Object/MedicalArtefact/ EHR/**Encounter**	This is a subclass of *EHR* and represents any registered interaction between a patient and healthcare provider(s) for the purpose of providing healthcare service(s) or assessing the health status of a patient.	
Object/MedicalArtefact/ EHR/**Diagnostics**	This is a subclass of *EHR* and represents the practice or techniques of diagnosis. It involves a number of subclasses and properties that help formally describe the means used for examining a patient.	**hasMedicalReport**: This is an object property with range the class *Medical Report* of our model. **refersToLOINC**: This property associates the class *Diagnostics* with the taxonomy of Logical Observation Identifiers Names and Codes (LOINC). LOINC taxonomy is a database and universal standard for identifying medical laboratory observations [33]. **hasComponentAnalyte**: This property associates the class *Diagnostics* with Component / Analyte of LOINC for registering the name of the component or analyte measured (e.g. Potassium, Hemoglobin, Hepatitis C antigen) [33]. **hasSystem**: This property associates the class *Diagnostics* with the System of LOINC to describe the type of sample (e.g. Urine, Blood) [33].

		hasMethod: This property associates the class Method of LOINC to describe the technique used to produce the diagnostics result (e.g. Rapid Plasma Reagin, Branched chain DNA (bDNA)) [33].
Object/MedicalArtefact/ EHR/Diagnostics/ **MedicalImaging**	This is a subclass of *Diagnostics* and refers to the process of producing a digital image of any part of the human body based on radiographic techniques.	
Object/MedicalArtefact/ EHR/Diagnostics/ **LaboratoryResult**	This is a subclass of Diagnostics and refers to the part of patients' records that present the result of any diagnostic test performed in a laboratory e.g. result of blood group test.	**refersToDiagnosticTests**: This property associates the class LaboratoryResult with the HL7 CDA taxonomy and specifically the Diagnostic Tests class (blood tests, urine test, electrocardiogram, etc.) of HL7 CDA taxonomy [34].
Object/MedicalArtefact/ EHR/**Diagnosis**	This is a subclass of *EHR* and refers to the conclusion of an examination concerning a potential health issue. Medical diagnosis is the act of determining a person's pathological status from an available set of findings [35].	**hasMedicalReport**: This is an object property with range the class *Medical Report* of our model. **refersToDisease**: This object property associates the class *Diagnosis* with the ICD-10 taxonomy which is the international standard (diagnostic classification) for reporting diseases and health conditions [36]. **hasICD10Code**: This property is used for registering the ICD-10 code used for classifying the diagnosed disease (e.g. 1A07.Z). **hasICD10Title**: This property describes the name if the disease diagnosed (e.g. typhoid fever).

Table 8: Details of the Object context element

The details of the Object context element are captured in the following UML Class diagram presented in Figure 7 and Figure 8. The complete UML Object class diagram is given at the following source [37].

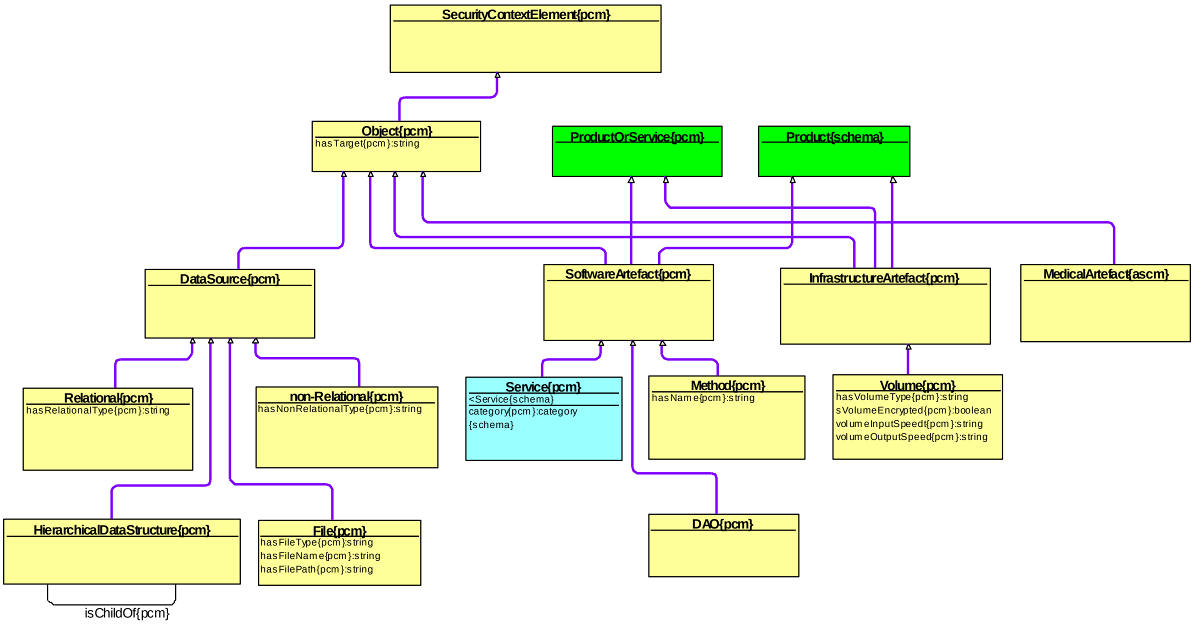


Figure 7: ASCLEPIOS UML class diagram for the Object context element

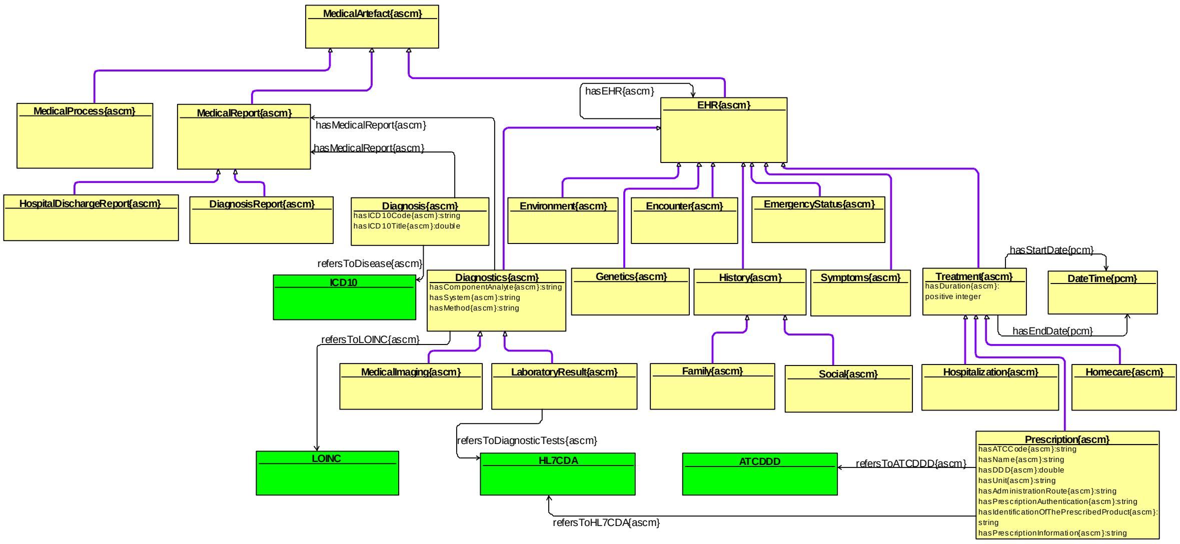


Figure 8: UML Class diagram for the Medical Artefact context element

In Table 9 and Figure 9 [38], the details of the Subject context element are presented. We note that although both Object and Subject may participate as individual concepts in access rules or policy expressions, we consider them as part of our model, in order to cover some of the valuable contextual information that usually accompany them. Such contextual information can enhance the access control where it might be not enough to identify who is the entity that is requesting access and what is the target object. Additional information about both of them can lead to the enforcement of even more dynamic and context-aware access controls (e.g. how has the entity been authenticated? What type of data does the access request target?).

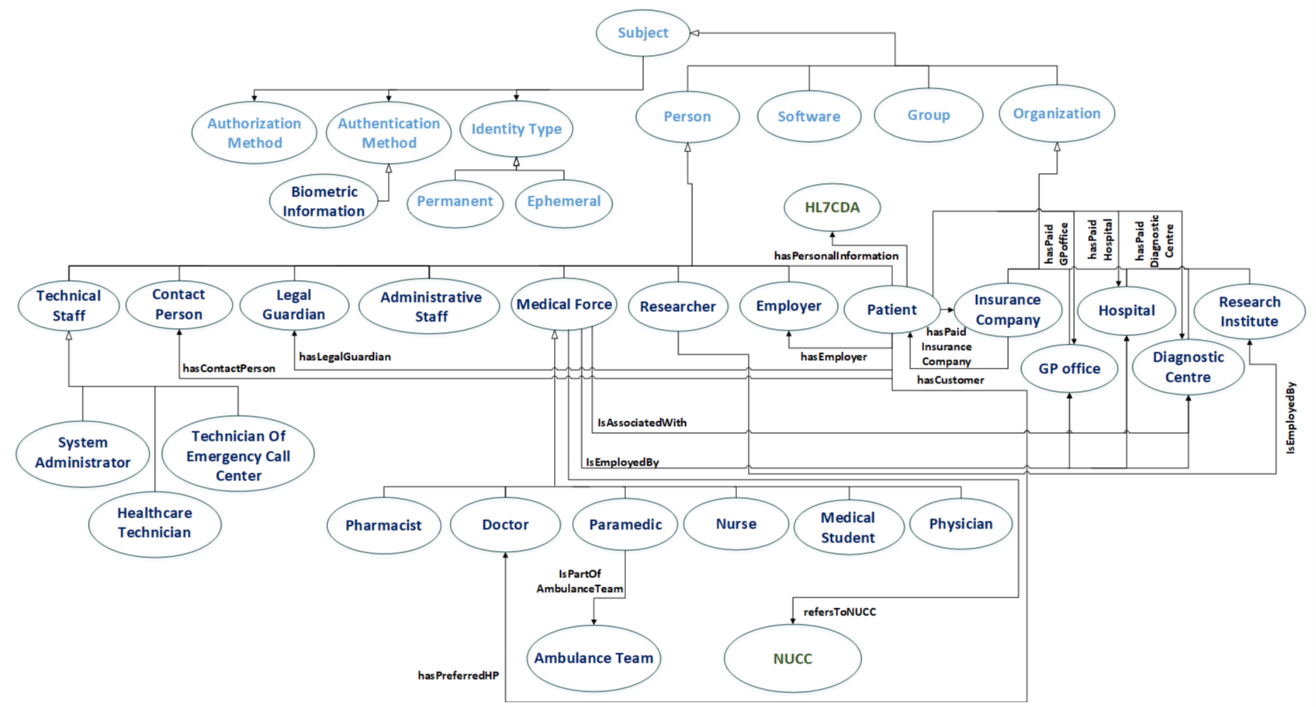


Figure 9: Extension of Security Context Element Subject overview diagram

Class Path (Hierarchically)	Class Description	Property / Description
Subject	An instance of this class represents the agent seeking access to a particular data artefact. This can be an organization, a person, a group or a service. **Extends** foaf:Agent http://purl.org/goodrelations/v1#BusinessEntity http://purl.org/goodrelations/v1#ProductOrService	**hasRole**: This property associates a *Subject* with the class usdl-core:Role. It is a sub-property of the usdl-core:withRole. The class *Role* constitutes a grouping mechanism for categorizing Subjects based on various properties (e.g. job title, user functions, responsibilities etc.).
		hasIdentityType: This property associates a Subject with the class *IdentityType*.
		hasAuthenticationMethod: This property associates a *Subject* with the class AuthenticationMethod.
		hasAuthorizationMethod: This property associates a *Subject* with the class *AuthorizationMethod*.
		name: This property associates a *Subject* instance with a piece of text that identifies it. This property is semantically equivalent to dcterms:title, rdfs:label and gr:name. **Imported from** http://purl.org/goodrelations/v1#name
		description: This property associates a *Subject* instance with a short textual description. This property is semantically equivalent to rdfs:comment and gr:description. **Imported from** http://purl.org/goodrelations/v1#description

		category: This property associates a *Subject* instance with the name of a category (if any) that the instance belongs to: **Imported from** http://purl.org/goodrelations/v1#category
		hasBrand: This property associates a *Subject* with a brand, which is the identity of a specific product, service, or business. **Imported from** http://purl.org/goodrelations/v1#Brand
		hasDUNS : This property associates a *Subject* with the Dun & Bradstreet DUNS number. The Dun & Bradstreet DUNS is a nine-digit number used to identify legal entities. **Imported from** http://purl.org/goodrelations/v1#hasDUNS
		legalName: This property associates a *Subject* with the legal name drawn from the class gr:BusinessEntity. **Imported from** http://purl.org/goodrelations/v1#legalName
		hasConnectivity: This property associates the *Subject* with the *Connectivity* class of our model to register the connection details used by the entity that requests access to sensitive data.
IdentityType	This class refers to permanent/static and ephemeral information that can be used for identifying a *Subject*.	
IdentityType / **PermanentId**	This class refers to infrequently-changing information that can be used for identifying a *Subject* (e.g. Name, Username, Id number, URI, SSN, Phone number, Email).	
IdentityType / **EphemeralId**	This class refers to frequently-changing information that can be used for identifying a *Subject* (e.g. Electronic Token, Biometric Token Cookie). Authentication	
Authentication Method	This class reveals the technological means used for validating the identity of a Subject during an access request (e.g. OS, LDAP, OpenID, Anonymous access)	
Authentication Method / **Biometric Information**	This class represents the measurement and statistical analysis of people's unique physical and behavioral characteristics e.g. fingerprint, iris.	
Authorization Method	This class indicates the framework used for granting access rights to a *Subject* (e.g. OS, LDAP, OAuth).	
Subject/**Group**	This class represents a collection of individual agents. Imported from foaf:Group	
Subject / **Organization**	This class represents a kind of Agent corresponding to social institutions such as companies, societies etc. Imported from foaf:*Organization*	

Subject / Organization / **InsuranceCompany**	This is a subclass of Organization which represents a financial institution which underwrites the risk of and compensates for the loss of, or damage to, personal and business assets (general insurance) and life or limb (life and accident insurance).	
Subject / Organization / **GPOffice**	This is a subclass of Organization which represents the business entity of the general practitioner.	
Subject / Organization / **Hospital**	This is a subclass of Organization which represents an institution dedicated to medical and surgical treatment and nursing care for sick or injured people.	
Subject / Organization / **DiagnosticCentre**	This is a subclass of Organization which represents a place that offers diagnostic services to the medical profession or general public.	
Subject / Organization / **ResearchInstitute**	This is a subclass of Organization which represents an institute where researchers gain, correct or improve knowledge about phenomena by using scientific methods and techniques, based on empirical or measurable observations [42].	
Subject / **Software**	This class represents software that attempts to access sensitive data.	
Subject / **Person**	This class represents people.	
Subject/Person/ **TechnicalStaff**	This is a subclass of *Person* which represents any entity with technical capabilities and responsibilities. It involves three subclasses: *System Administrator, Healthcare Technician and Technician of Emergency Call Centre.*	
Subject/Person/ TechnicalStuff/ **SystemAdministrator**	This is a subclass of *Person* which represents the entity managing the operation of a computer system or particular electronic communication service.	
Subject/Person/ TechnicalStuff/ **TechnicianOf EmergencyCallCentre**	This is a subclass of *Person* which represents employees who work at emergency call centres.	
Subject / Person/ TechnicalStuff/ **Heathcare Technician**	This is a subclass of *Person* which involves skilled personnel who works in a specialized area within the health care industry e.g., microbiology lab assistant, x-ray machine technician.	
Subject/Person/ **AdministrativeStaff**	This is a subclass of *Person* which represents any entity in charge of administrative responsibilities in healthcare provider organisation.	

Subject/Person/ **MedicalForce**	This is a subclass of *Person* which represents the medical staff who conducts research; improves or develops concepts, theories and operational methods; and applies scientific knowledge relating to medicine, nursing, dentistry, veterinary medicine, pharmacy, and promotion of health. Competent performance in most occupations in this sub-major group requires skills at the fourth ISCO skill level [39].	**isEmployedBy**: This object property associates the class *MedicalForce* with the organization in which a medical force instance is employed by (i.e. hospital, general practitioner's (GP) office and a diagnostic centre).
		isAssociatedWith: This object property associates the class *MedicalForce* with the organization which a medical force instance is associated with (i.e. hospital, general practitioner's (GP) office and a diagnostic centre). This property differentiates from the previous in the sense that it describes an external collaboration with a certain organisation and not a fixed employment contract.
		refersToNUCC: This object property associates the class *MedicalForce* to the National Uniform Claim Committee (NUCC) taxonomy [40].
		hasCode: This property associates the class *MedicalForce* with the taxonomy's *NUCC Code* which is a ten character code [40], (e.g. "1223P0221X"), which maps to the clinicians' medical specialty.
		hasGrouping: This property associates the class *MedicalForce* with the taxonomy's Grouping of NUCC taxonomy for describing the general group of the specialty (e.g. Dental Providers) [40].
		hasClassification: This property associates the class *MedicalForce* with the *Classification* which describes the specialty (e.g. Dentist, Oral Medicinist, Dental Hygienist, Dental Therapist) of a certain medical force instance [40].
		hasDefinition: This property associates the class *MedicalForce* with the Definition of NUCC for providing a short text that describes a medical force instance's specialty (e.g. "An age-defined specialty that provides both primary and comprehensive preventive and therapeutic oral health care for infants and children through adolescence, including those with special health care needs") [40].
Subject/Person/ MedicalForce/ **Pharmacist**	This is a subclass of MedicalForce and represents the person who is licensed by the appropriate state regulatory agency to engage in the practice of pharmacy. The practice of pharmacy includes, but is not limited to, assessment, interpretation, evaluation, and implementation, initiation, monitoring or modification of medication and or medical orders; the compounding or dispensing of medication and or medical orders; participation in drug and device procurement, storage, and selection; drug administration; drug regimen reviews; drug or drug-related research; provision of patient education and the provision of those	

	acts or services necessary to provide medication therapy management services in all areas of patient care [40].	
Subject/Person/ MedicalForce/ **Doctor**	This is a subclass of *MedicalForce* with instances that correspond to a person who studies, diagnoses, treats and prevents illness, disease, injury, and other physical and mental impairments in humans through the application of the principles and procedures of modern medicine, and additionally he plans, supervises and evaluates the implementation of care and treatment plans by other health care providers, and conduct medical education and research activities [41].	**hasDoctorEncounter**: This object property associates the class *Doctor* with the class *Encounter* of our model (subclass *Object*) for registering any interaction of a Doctor with a patient.
Subject / Person/ MedicalForce/ **Paramedic**	This is a subclass of MedicalForce with instances that correspond to a person who is an individual trained and certified to perform advanced life support (ALS) in medical emergencies based on individual state boards [40].	**isPartOfAmbulanceTeam**: This object property associates the class Paramedic with the class AmbulanceTeam for identifying the paramedic workers who belong in the ambulance team.
Subject/Person/ MedicalForce/ **Nurse**	This is a subclass of MedicalForce with instances that correspond to a person who provides treatment and care services for people who are physically or mentally ill, disabled or infirm, and others in need of care due to potential risks to health including before, during and after childbirth, and additionally assumes responsibility for the planning, management and evaluation of the care of patients, including the supervision of other health care workers, working autonomously or in teams with medical doctors and others in the practical application of preventive and curative measures [41].	**hasNurseEncounter**: This object property associates the class Nurse with the class Encounter of our model (subclass Object) for registering any interaction of a Nurse with a patient.
Subject/Person/ MedicalForce/ **MedicalStudent**	This is a subclass of MedicalForce with instances that correspond to a person who is enrolled in an organized health care education/training program leading to a degree, certification, registration, and/or licensure to provide health care [41].	
Subject/Person/ MedicalForce/ **Physician**	This is a subclass of MedicalForce with instances that correspond to Physician who diagnoses, treats and prevents illness, disease, injury, and other physical and mental impairments in humans through application of the principles and procedures of modern medicine, and additionally he does not limit his	

	practice to certain disease categories or methods of treatment, and may assume responsibility for the provision of continuing and comprehensive medical care to, and the maintenance of general health of, individuals, families and communities [41].	
Subject/Person/ **Researcher**	This is a subclass of *Person* with instances that correspond to a person who conducts scientific research involving background research, constructing a hypothesis, testing it, analysing data and concluding the results [42].	**isEmployedBy**:This object property associates the *Researcher* with the *ResearchInstitute* class for our model.
Subject/Person/ **Employer**	This is a subclass of *Person* with instances that correspond to those workers who, working on their own account or with one or a few partners, hold the type of job defined as a self-employed job, and in this capacity, on a continuous basis have engaged one or more persons to work for them in their business as employees [43].	
Subject/Person/ **LegalGuardian**	This is a subclass of *Person* that represents a person who legally assists and supports minor children, mentally disabled persons or incapacitated older adults in their personal life. They can manage their property, help with daily financial administration and assist with the ward's medical or social needs [44].	
Subject/Person/ **ContactPerson**	This is a subclass of *Person* that represents someone who undertakes the responsibility of communication among healthcare providers or emergency call team in cases of an unconscious patient.	
Subject / Person / **Patient**	This is a subclass of *Person* with instances that correspond to a person who is a recipient of healthcare, that is services received by individuals or communities to promote, maintain, monitor or restore health. Additionally, he is referred to as patient, rather than as client, tenant or consumer, although it is recognized that recipients such as a healthy pregnant woman or a child undergoing immunization may not be regarded, or regard themselves, as patients [45].	**hasLegalGuardian**: This object property associates the class Patient with the class LegalGuardian of our model [34] for denoting the legal representative of the patient.
		hasContactPerson: This object property associates the class Patient with the class ContactPerson [34].
		hasEmployer: This object property associates the class Patient with the class Employer.
		hasPatientEncounter: This object property associates the class Patient with the class Encounter which is subclass of class Object.
		IsInpatient: This object property is used to denote that a certain person is being treated in the premises of a GP office, a Hospital or a Diagnostic Centre.
		IsOutpatient: This object property is used to denote that a certain person is being treated outside the premises of a GP office, a Hospital or a Diagnostic Centre (usually at his home).

		isMentallyIncapable: This data property associates the class Patient with a Boolean which represents if a patient is mentally capable or not.
		isPhysicallyIncapable: This data property associates the class Patient with a Boolean which represents if a patient is physically capable or not.
		hasBodyHeight: This data property represents the patient's Height in centimeters (cm) (e.g. 170) [32].
		hasBodyWeight: This data property represents the patient's weight in kilograms (kg). (e.g. 70) [32].
		hasPaidHospital: This object property associates the class *Patient* with the class Hospital and denotes the successful payment of the received healthcare services.
		hasPaidGPOffice: This object property associates the class *Patient* with the class *GPOffice* and denotes the successful payment of the received healthcare services..
		hasPaidDiagnosticCentre: This object property associates the class *Patient* with the class *DiagnosticCentre* and denotes the successful payment of the received healthcare services.
		hasPaidInsuranceCompany: This object property associates the class *Patient* with the class *InsuranceCompany* and denotes the successful payment of the received insurance services.
		hasPreferredHP: This object property associates the class *Patient* with the class *Doctor* to denote a preference in the health practitioner.
		hasMedicalReport: This property associates the class *Patient* with the class *MedicalReport*.
		hasPrescription: This property associates the class *Patient* with the class *Prescription*.
		patientHasEHR: This property associates the class Patient with the class *EHR* which is subclass of the class *Object*.
		hasPersonalInformation: This object property associates the class Patient with the HL7 CDA taxonomy to register all the required personal Information of the patient [34] (e.g. social security number).
		hasInsuranceInformation: This property is used for registering the patient's private insurance information (e.g. insurance number - QQ 12 34 56 A) [34].

Table 9: Details of the Subject context element

The details of the Subject context element are captured in the following UML Class diagram presented in Figure 10, Figure 11 and Figure 12. The complete UML Subject class diagram is given at the following source [46].

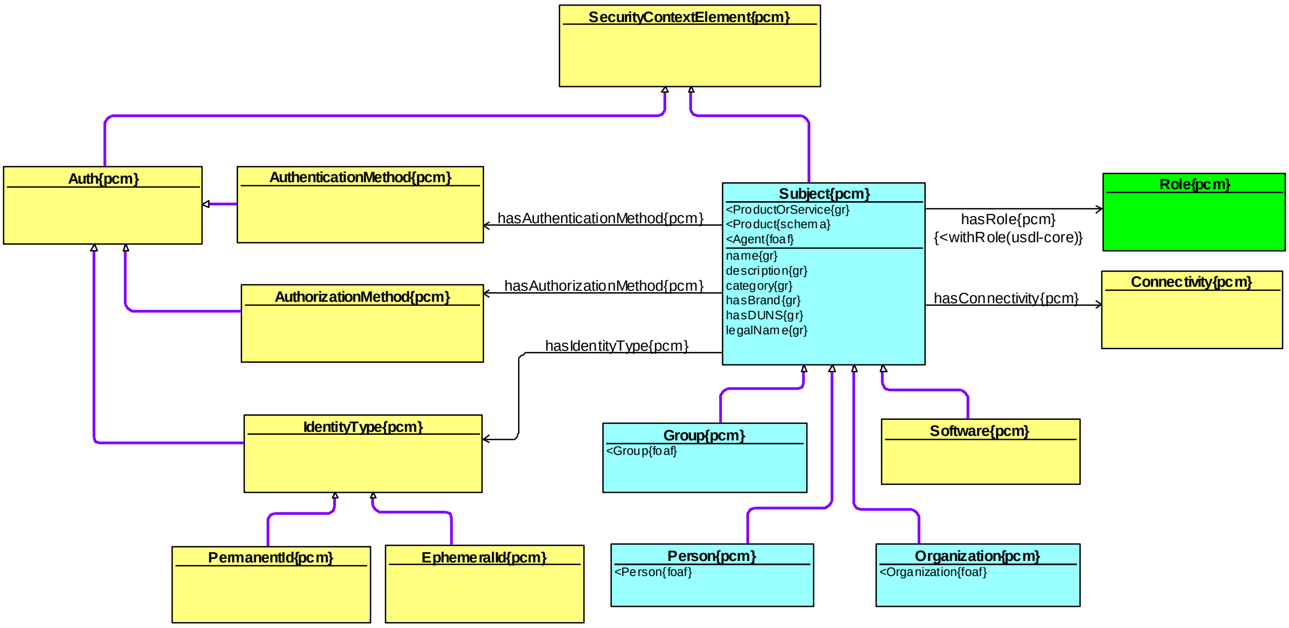


Figure 10: UML Class diagram for the Subject context element

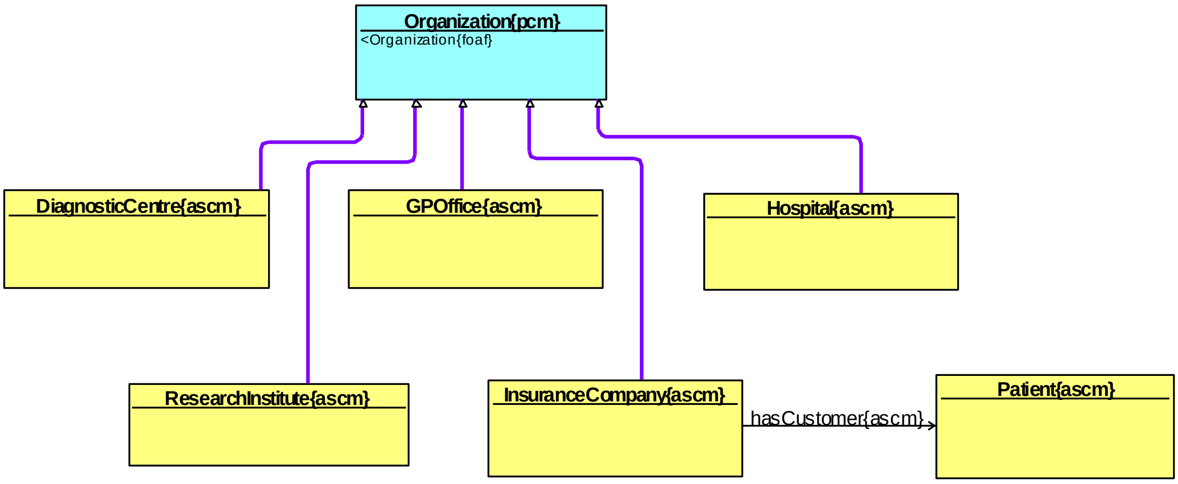


Figure 11: UML Class diagram for the Organization context element

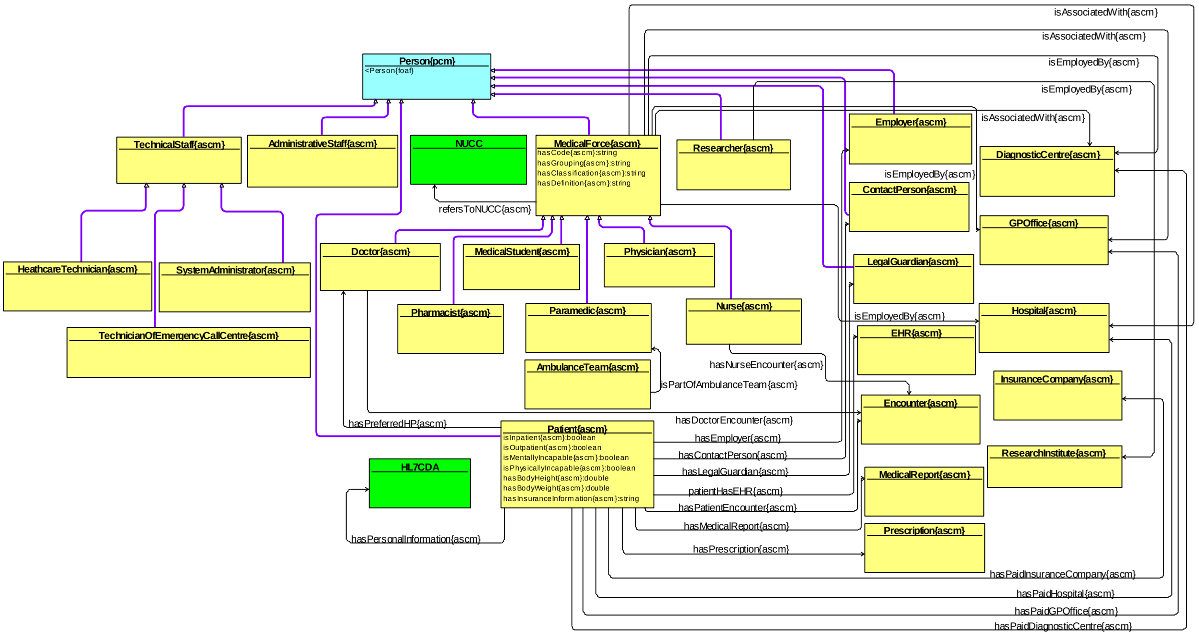


Figure 12: UML Class diagram for the Person context element

With respect to concepts related to security awareness we investigated the Common Attack Pattern Enumeration and Classification (CAPEC) taxonomy [47]. In the following Figure 13 [48] are demonstrated the main classes of this taxonomy.

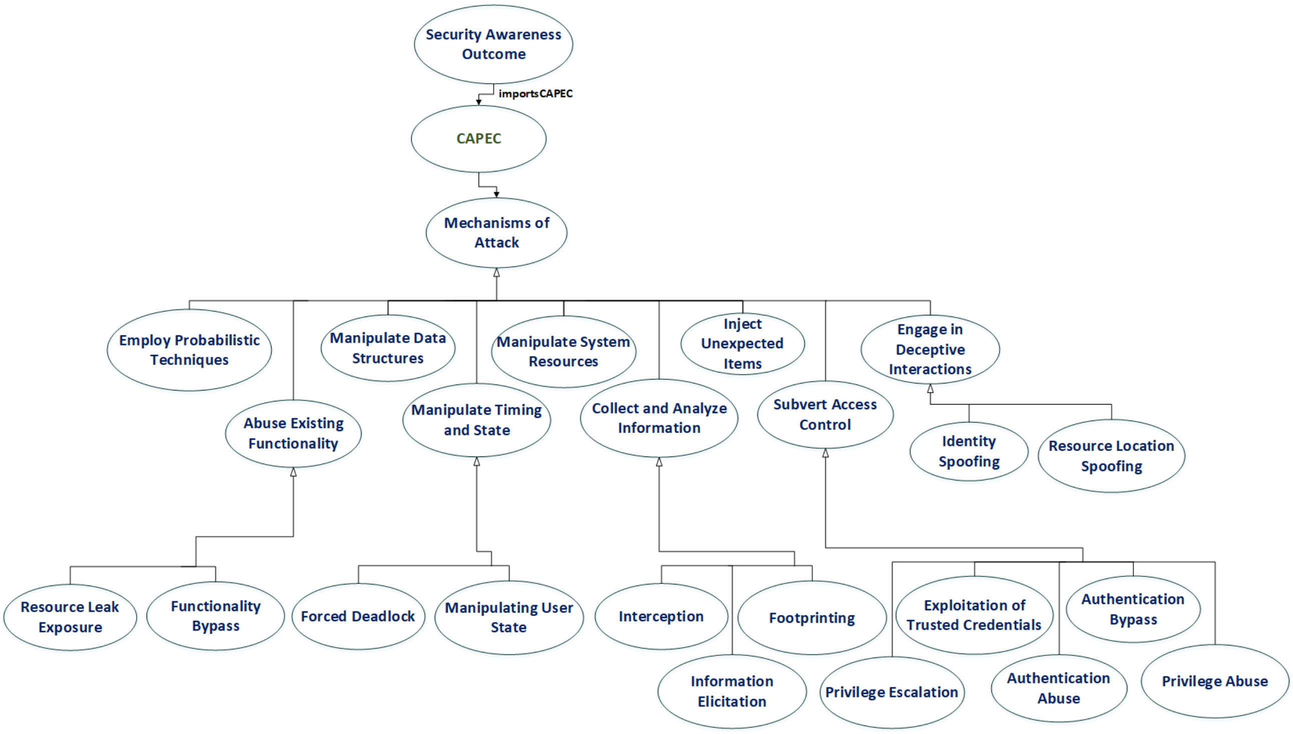


Figure 13: Security awareness component

In the following Table 10 there are analytically described the classes of CAPEC taxonomy:

Class Path (Hierarchically)	Class Description	Property / Description

MechanismsOf Attack	An instance of this class organizes attack patterns hierarchically based on mechanisms that are frequently employed when exploiting vulnerability. The categories that are members of this view represent the different techniques used to attack a system. They do not, however, represent the consequences or goals of the attacks. There is the potential for some attack patterns to align with more than one category depending on one's perspective. [47].	
MechanismsOf Attack / **Employ Probabilistic Techniques**	This is a subclass of *MechanismsOfAttack* with instances that correspond to an attacker who utilizes probabilistic techniques to explore and overcome security properties of the target [47].	
MechanismsOf Attack / **AbuseExisting Functionality**	This is a subclass of *MechanismsOfAttack* with instances that correspond to an adversary who uses or manipulates one or more functions of an application in order to achieve a malicious objective not originally intended by the application, or to deplete a resource to the point that the target's functionality is affected This comprises a broad class of attacks wherein the adversary is able to alter the intended result or purpose of the functionality and thereby affect application behavior or information integrity. [47].	
MechanismsOf Attack / AbuseExisting Functionality / **Resource Leak Exposure**	This is a subclass of *AbuseExistingFunctionality* with instances that correspond to an adversary who utilizes a resource leak on the target to deplete the quantity of the resource available to service legitimate requests. Resource leaks most often come in the form of memory leaks where memory is allocated but never released after it has served its purpose, however, theoretically, any other resource that can be reserved can be targeted if the target fails to release the reservation when the reserved resource block is no longer needed [47].	
MechanismsOf Attack / AbuseExisting Functionality / **Functionality Bypass**	This is a subclass of *AbuseExistingFunctionality* with instances that correspond to an adversary who attacks a system by bypassing some or all functionality intended to protect it. Often, a system user will think that protection is in place, but the functionality behind those protections has been disabled by the adversary [47].	
MechanismsOf Attack / **ManipulateData Structures**	This is a subclass of *MechanismsOfAttack* with instances that correspond to attack patterns in this category which manipulate and exploit characteristics of system data structures in order to violate the intended usage and protections of these structures. This is done in such a way that yields either improper access to the associated system data or violations of the security properties of the system itself due to vulnerabilities in how the system processes and manages the data structures [47].	
MechanismsOf Attack / **Manipulate TimingAndState**	This is a subclass of *MechanismsOfAttack* with instances that correspond to an attacker who exploits weaknesses in timing or state maintaining functions to perform actions that would otherwise be prevented by the execution flow of the target code and processes. An example of a state attack might include manipulation of an application's information to change	

	the apparent credentials or similar information, possibly allowing the application to access material it would not normally be allowed to access [47].	
MechanismsOf Attack / Manipulate TimingAndState / **ForcedDeadlock**	This is a subclass of *ManipulateTimingAndState* with instances that correspond to the adversary who triggers and exploits a deadlock condition in the target software to cause a denial of service. A deadlock can occur when two or more competing actions are waiting for each other to finish, and thus neither ever does. Deadlock conditions can be difficult to detect [47].	
MechanismsOf Attack / Manipulate TimingAndState / **Manipulating UserState**	This is a subclass of ManipulateTimingAndState with instances that correspond to the adversary who modifies state information maintained by the target software in user-accessible locations. If successful, the target software will use this tainted state information and execute in an unintended manner. State management is an important function within an application. User state maintained by the application can include usernames, payment information, browsing history as well as application-specific contents such as items in a shopping cart [47].	
MechanismsOf Attack / **Manipulate System Resources**	This is a subclass of *MechanismsOfAttack* with instances that correspond to attack patterns within this category which focus on the adversary's ability to manipulate one or more resources in order to achieve a desired outcome. This is a broad class of attacks wherein the attacker is able to change some aspect of a resource's state or availability and thereby affect system behavior or information integrity. Examples of resources include files, applications, libraries, infrastructure, and configuration information [47].	
MechanismsOf Attack / **Collect AndAnalyze Information**	This is a subclass of *MechanismsOfAttack* with instances that correspond to attack patterns within this category which focus on the gathering, collection, and theft of information by an adversary. The adversary may collect this information through a variety of methods including active querying as well as passive observation. By exploiting weaknesses in the design or configuration of the target and its communications, an adversary is able to get the target to reveal more information than intended. Information retrieved may aid the adversary in making inferences about potential weaknesses, vulnerabilities, or techniques that assist the adversary's objectives [47].	
MechanismsOf Attack / Collect AndAnalyze Information / **Interception**	This is a subclass of *CollectAndAnalyzeInformation* with instances that correspond to an adversary who monitors data streams to or from the target for information gathering purposes. This attack may be undertaken to solely gather sensitive information or to support a further attack against the target. This attack pattern can involve sniffing network traffic as well as other types of data streams (e.g. radio). The adversary can attempt to initiate the establishment of a data stream or passively observe the communications as they unfold. In all variants of this attack, the adversary is not the intended recipient of the data stream. [47].	

MechanismsOf Attack / Collect AndAnalyze Information / **Information Elicitation**	This is a subclass of *CollectAndAnalyzeInformation* with instances that correspond to an adversary who engages an individual using any combination of social engineering methods for the purpose of extracting information. Accurate contextual and environmental queues, such as knowing important information about the target company or individual can greatly increase the success of the attack and the quality of information gathered. Authentic mimicry combined with detailed knowledge increases the success of elicitation attacks [47].	
MechanismsOf Attack / Collect AndAnalyze Information / **Footprinting**	This is a subclass of *CollectAndAnalyzeInformation* with instances that correspond to an adversary who engages in probing and exploration activities to identify constituents and properties of the target. Footprinting is a general term to describe a variety of information gathering techniques, often used by attackers in preparation for some attack. It consists of using tools to learn as much as possible about the composition, configuration, and security mechanisms of the targeted application, system or network [47].	
MechanismsOf Attack / **Inject Unexpected Items**	This is a subclass of *MechanismsOfAttack* with instances that correspond to attack patterns within this category which focus on the ability to control or disrupt the behavior of a target either through crafted data submitted via an interface for data input, or the installation and execution of malicious code on the target system. The former happens when an adversary adds material to their input that is interpreted by the application causing the targeted application to perform steps unintended by the application manager or causing the application to enter an unstable state. [47].	
MechanismsOf Attack / **Subvert AccessControl**	This is a subclass of *MechanismsOfAttack* with instances that correspond to an attacker who actively targets exploitation of weaknesses, limitations and assumptions in the mechanisms a target utilizes to manage identity and authentication as well as manage access to its resources or authorize functionality. Such exploitation can lead to the complete subversion of any trust the target system may have in the identity of any entity with which it interacts, or the complete subversion of any control the target has over its data or functionality [47].	
MechanismsOf Attack / Subvert AccessControl / **Privilege Escalation**	This is a subclass of *SubvertAccessControl* with instances that correspond to an adversary who exploits a weakness enabling them to elevate their privilege and perform an action that they are not supposed to be authorized to perform [47].	
MechanismsOf Attack / Subvert AccessControl / **ExploitationOf Trusted Credentials**	This is a subclass of SubvertAccessControl with instances that correspond to attacks on session IDs and resource IDs that take advantage of the fact that some software accepts user input without verifying its authenticity. For example, a message queuing system that allows service requesters to post messages to its queue through an open channel (such as anonymous FTP), authorization is done through checking group or role membership contained in the posted message. However, there is no proof that the message itself, the	

	information in the message (such group or role membership), or indeed the process that wrote the message to the queue are authentic and authorized to do so [47].	
MechanismsOf Attack / Subvert AccessControl / **Authentication Abuse**	This is a subclass of *SubvertAccessControl* with instances that correspond to an attacker who obtains unauthorized access to an application, service or device either through knowledge of the inherent weaknesses of an authentication mechanism, or by exploiting a flaw in the authentication scheme's implementation. In such an attack an authentication mechanism is functioning but a carefully controlled sequence of events causes the mechanism to grant access to the attacker. This attack may exploit assumptions made by the target's authentication procedures, such as assumptions regarding trust relationships or assumptions regarding the generation of secret values [47].	
MechanismsOf Attack / Subvert AccessControl / **Authentication Bypass**	This is a subclass of *SubvertAccessControl* with instances that correspond to an attacker who gains access to application, service, or device with the privileges of an authorized or privileged user by evading or circumventing an authentication mechanism. The attacker is therefore able to access protected data without authentication ever having taken place. This refers to an attacker gaining access equivalent to an authenticated user without ever going through an authentication procedure. This is usually the result of the attacker using an unexpected access procedure that does not go through the proper checkpoints where authentication should occur [47].	
MechanismsOf Attack / Subvert AccessControl / **Privilege Abuse**	This is a subclass of *SubvertAccessControl* with instances that correspond to an adversary who is able to exploit features of the target that should be reserved for privileged users or administrators but are exposed to use by lower or non-privileged accounts. Access to sensitive information and functionality must be controlled to ensure that only authorized users are able to access these resources. If access control mechanisms are absent or misconfigured, a user may be able to access resources that are intended only for higher level users. An adversary may be able to exploit this to utilize a less trusted account to gain information and perform activities reserved for more trusted accounts [47].	
MechanismsOf Attack / **Engage InDeceptive Interactions**	This is a subclass of *MechanismsOfAttack* with instances that correspond to attack patterns within this category which focus on malicious interactions with a target in an attempt to deceive the target and convince the target that it is interacting with some other principal and as such take actions based on the level of trust that exists between the target and the other principal. These types of attacks assume that some piece of content or functionality is associated with an identity and that the content / functionality is trusted by the target because of this association. Often identified by the term "spoofing", these types of attacks rely on the falsification of the content and/or identity in such a way that the target will incorrectly	

	trust the legitimacy of the content [47].	
MechanismsOf Attack / Engage InDeceptive Interactions / **Identity Spoofing**	This is a subclass of *EngageInDeceptiveInteractions* with instances that correspond to identity spoofing which refers to the action of assuming (i.e., taking on) the identity of some other entity (human or non-human) and then using that identity to accomplish a goal. An adversary may craft messages that appear to come from a different principle or use stolen / spoofed authentication credentials. Alternatively, an adversary may intercept a message from a legitimate sender and attempt to make it look like the message comes from them without changing its content [47].	
MechanismsOf Attack / Engage InDeceptive Interactions / **Resource Location Spoofing**	This is a subclass of *EngageInDeceptiveInteractions* with instances that correspond to an adversary who deceives an application or user and convinces them to request a resource from an unintended location. By spoofing the location, the adversary can cause an alternate resource to be used, often one that the adversary controls and can be used to help them achieve their malicious goals [47].	

Table 10: Details of the classes of CAPEC taxonomy

8 Policies Model

This chapter provides an overview of the policy types introduced in the ASCLEPIOS framework and provides a generic ontological meta-model for their abstract representation. It also discusses the advantages of this approach.

8.1 Purpose and Types of Policy Models in ASCLEPIOS

The purpose of policies is to determine the way entities of a particular domain behave, by specifying certain constraints. These constraints are enforced at runtime by policy enforcement mechanisms. A point of paramount importance is the separation of policies from the policy enforcement mechanisms implementation, because this way it is possible to use different constraints for the domain entities without having to change the implementation of the enforcement mechanisms. Such a separation requires a *declarative* representation of policies rather than an imperative one[1].

In ASCLEPIOS we will use a semantic (and declarative) representation of policies, because it offers the following advantages over a syntactic approach[2]:

i. It leverages the policy portability and reusability;
ii. It enables the translation to/from other policy models;
iii. It enables the construction of *policy-validation mechanisms* that can automatically reason about the validity and compliance of policies to certain meta-constraints. Such reasoning can be performed independently to any particular syntactic representation of the policies.

The semantic representation acts as an abstract, formal language in which meanings are represented. It requires the definition of models that capture all relevant domain entities, their interrelations while it adds meanings to them. Domain entities are represented as classes and their relations as properties between the classes. Therefore, policies must be modelled according to a semantic representation in order to have their information falling into certain classes or properties, hence having particular meanings. Thus, the semantic representation provides a meta-model used to describe and assign meanings to policies captured as policy models. From an object-oriented (OO) point of view, one could parallel the semantic representation to an OO class model or Entity-Relation Diagram, and the policy models (that capture policies) as graphs of class instances (i.e. objects). It is worth mentioning that a policy model is a rational decision-making scheme based on certain constraints. A policy model can have a syntactic representation but also a semantic one that gives meaning to its entities. From the OO perspective, a policy model can be an instantiation of a policy semantic representation.

In the context of ASCLEPIOS, we focus on authorization policies and specifically on policies with constraints based on the use of contextual attributes, for making decisions on granting or denying access to protected resources or access operations. We specifically consider two types of policies and hence we provide two relevant semantic representations; one for ABAC policies and one for ABE policies.

[1] ***Declarative*** means knowledge about facts. It is explicit and easy to verbalize by capturing facts, events, rules or constraints and their interrelations. It is typically captured separately from programs (e.g. configuration files). Programs must know how to work with this type of knowledge. ***Procedural*** (or ***imperative***) means knowledge about actions to take or how to do something. It is implicit and refers to tasks, processes, methods, algorithms. It tends to be realized as programs (code), thus it suffers certain limitations like reduced portability and capability for reasoning on the knowledge.

[2] Syntactic refers to the grammatical structure of representation whereas Semantic refers to the meaning of representation terms (vocabulary)

A focal concept in these policy models is that of *Contextual Attribute* (or for simplicity just *attribute*). The former term (*contextual*) refers to any interesting information describing the situation (condition, state etc.) of entities relevant to a domain (including their environment and their relations). The latter term (*attribute*) refers to a quality or characteristic of a domain entity or entity relation. Therefore, a contextual attribute captures the *situation* or *state* of an entity's characteristic at a given moment. Time and location are typical examples of contextual attributes. The full set of contextual attributes (relating to an entity) is the entity's context (or group of entities or a whole system). We note that the contextual attributes of interest for ASCLEPIOS have been described in the context–aware security model (see chapter 7). Contextual attributes are used in ASCLEPIOS policy models (either individually or combined) to formulate constraints. Therefore, policies place constraints to the values attributes or sets of attributes can have in order to allow or deny access.

Apart from the authorization policies capturing the (authorization) constraints within which entities can interact with ASCLEPIOS framework, another import aspect is being able to validate these policies using a comprehensive and extensible "tool". Since policies can use several different attributes as well as expressions containing attributes, the policy validation "tool" must be generic enough and also being able to leverage the policy semantics. For this reason, a generic *Policy Validation* ontological model is also proposed. Based on this model two concrete policy validation types will be defined; namely *Policy Inspection* and *Security Awareness Assessment*. The former will investigate whether given policies abide to certain organisation-wide meta-constraints in order to allow them put in production, whereas the latter will check policies for security-related deficiencies with regard to known attack vectors, and issue warnings.

The Policy Validation model must be compatible with the ABAC and ABE policy models that will be presented in the next sections. Furthermore, ABAC policy model (derived from PaaSword [22], [23]) and ABE policy model (proposed in ASCLEPIOS) must also be compatible with each other thus enabling interoperability. For this reason, a meta-model that binds everything together, and defines their common concepts and relations is necessary. We call this meta-model, "ASCLEPIOS Authorization Policy & Validation Meta-Model."

8.2 ASCLEPIOS Authorization & Validation Meta-Model

The purpose of the meta-model is to introduce concepts abstract enough, which will enable deriving ABAC and ABE related concepts, as specializations (sub-classes or sub-properties) of the abstract concepts of meta-model. In cases where the use of concrete concepts (classes, properties) is already known, these are included in the meta-model.

The next figure provides an overview of the ASCLEPIOS Authorization Policy & Validation Meta-Model. Ovals with white fill represent classes imported from ASCLEPIOS security context model, presented in chapter 7.

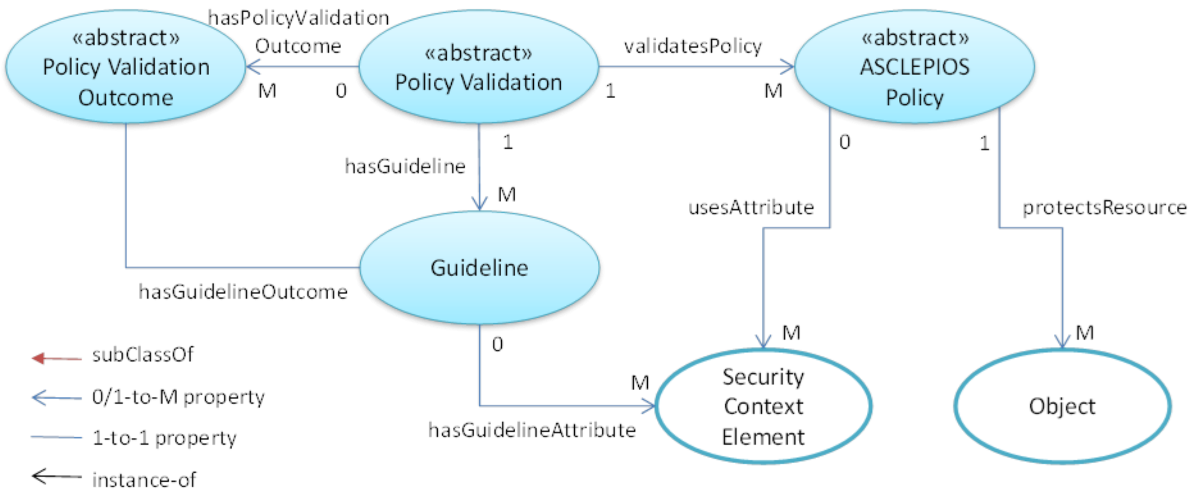


Figure 14 : ASCLEPIOS Authorization Policy & Validation Meta-Model

The abstract classes and relations (properties) of the Meta-Model are briefly described in the next table

Class Path (Hierarchically)	Class Description	Property / Description
ASCLEPIOS Policy	Represents an abstract policy class. In ABAC and ABE policy model, this class must be specialized (sub-classed).	***usesAttribute:*** It is an abstract relation between a generic policy and attributes, represented by *Security Context Elements* instances. It expresses the fact that policies use attributes. This relation can be specialized (sub-property) in policy models. A policy can use zero, one or many attributes.
		protectsResource: It is a relation between a generic policy and resources being protected, which are represented as instances of *Object* class. It captures the fact that a concrete policy must protect at least one resource.
Policy Validation	Represents an abstract policy validation that can be applied to policies. It must be specialized (sub-classed). A policy validation must contain meta-constraints, called *Guidelines*, which are checked against given policies. Each meta-constraint can yield an outcome. The combined outcomes of meta-constraints yield the overall outcome of a policy validation.	***validatesPolicy:*** It is an abstract relation that captures the fact that policy validations are used for checking policies. It must be specialized (sub-property) in concrete policy validation models in order to restrict the type of policy validations and type of policies involved, thus also specializing (narrowing) the meaning of this relation.
		hasGuideline: It is a relation capturing the fact that a policy validation must use one or more meta-constraints (called Guidelines) in order to check the given policies. It can be specialized in the concrete policy validation models.

		hasPolicyValidationOutcome: It is an abstract relation capturing the fact that a policy validation can have one or more outcomes. It can be specialized in the concrete policy validation models.
Guideline	Represents meta-constraints that can be included in policy validations. It must be specialized (sub-classed) in the policy validation models, in order to provide concrete checks; for example *Attribute Exists* guideline or *Attribute Expression Exists* guideline.	***hasGuidelineOutcome:*** It is a relation that captures the fact that a policy validation can have zero or more outcomes, depending on the policy validation type.
		hasGuidelineAttribute: It is a relation that captures the fact a guideline can (optionally) have an outcome.
Policy Validation Outcome	Represents the outcome of a policy validation. It can be the result, either of an individual meta-constraint, or it can be the combined result of a policy validation. The outcome type varies among concrete policy validation types (i.e. Inspection and Security awareness outcome). Therefore, it must be specialized (sub-classed) in the concrete policy validation models, in order to provide outcomes relevant to the validation type; for example *Valid* or *Invalid* in Policy Inspection.	
Security Context Element	This is a concrete class imported from *ASCLEPIOS Security Context Model* presented in chapter 7. It is used for representing any contextual attribute that can be used either in policies or policy validations.	
Object	This is a concrete class imported from *ASCLEPIOS Security Context Model* presented in chapter 7. It is used for representing any resource that can be protected by a policy.	

Table 11: Authorization Policy and Validation Meta-Model classes and properties

In the following sections the ABAC and ABE policy models, as well as the Policy Inspection and Security Awareness Assessment validation model, are provided as specializations of abstract classes of this meta-model.

8.3 ABAC Policy Model

In this section the ABAC policy model is presented. The purpose of this model is to capture in a formal and generic way the concepts related to ABAC authorization. Concrete ABAC policies are instances of the ABAC policy model classes.

The ABAC policy model used in ASCLEPIOS is based on the corresponding policy model of PaaSword project [23]. Although it is defined as a specialization of the Authorization Policy & Validation Meta-Model, presented in section 8.2, it is additionally mapped onto the classes and properties of the PaaSword policy model.

The ASCLEPIOS ABAC policy model derives from the *ASCLEPIOS Policy* abstract class by specializing it with *ABAC Policy* class, as well as the related properties. Additional classes are also introduced, which capture concepts and operations specific to ABAC policies.

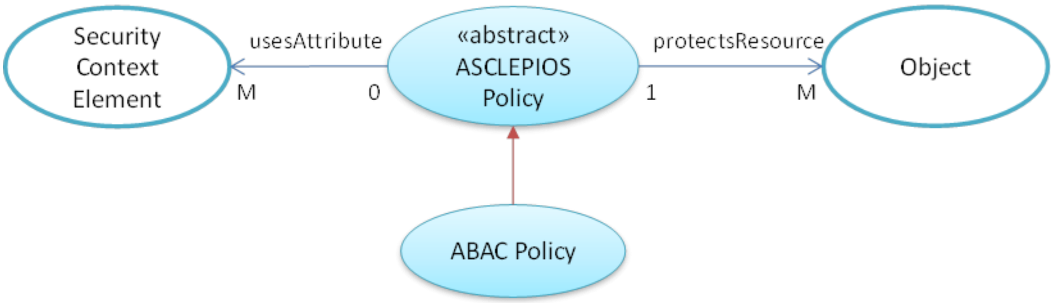


Figure 15: ABAC Policy Model as specialization of Authorization Meta-Model

Apart from Authorization Meta-Model of ASCLEPIOS, the ABAC Policy Model also derives from PaaSword policy model, as it will be explained in the subsequent sections.

8.3.1 ASCLEPIOS ABAC Policy Model

Practically, an ABAC engine protects resources by intercepting attempts/requests to access or use them, checks the characteristics (i.e. attributes) of the access requests as well as the resource (and possibly of the environment) against a set of ABAC policies, and eventually allows or blocks further processing of access requests. The characteristics of requests, resources and access operations can be dynamically-evolving contextual attributes.

Each ABAC policy comprises a number of ABAC rules as well as a combining algorithm. In fact, all ABAC policy models consider these two types of entities and their semantic representations:

- ABAC Rules, and
- ABAC Policies.

An ABAC rule is the most basic structural element of policies. It comprises a number of attributes, which typically describe entities attempting to access a protected object, the attempted actions, environmental conditions that must hold and the protected objects. Each rule can refer to one or more resources (to be protected), one or more access operations (actions) considered, any number of requestors and yield exactly one outcome that is either positive or negative; i.e. permit access or deny access. When the attributes of an incoming request (including requestor and action), as well as those of resource meet those specified in the rule, the rule is triggered and the associated outcome (positive or negative) is yielded. Moreover, a context expression can be used in a rule to capture the state of any number of contextual attributes in order to trigger the rule. The outcomes of all triggered rules in a policy are then combined using the specified policy combing algorithm.

According to the previous discussion an ABAC rule takes the form:

Requestor with **Context Expression** has **Authorisation** for **Action** on **Resource**

It defines a generic structure, in terms of relevant attributes, to which all ABAC rules in the ASCLEPIOS framework adhere. It comprises several attributes which will be elaborated below.

The next figure provides an overview of the ABAC authorization policy model classes and their interrelations. The classes depicted with ovals with white fill represent classes imported from the Security Context model presented in chapter 7. Ovals with cyan fill represent classes introduced in the ABE Policy model. Moreover, classes from PaaSword policy model are indicated with ovals with green fill.

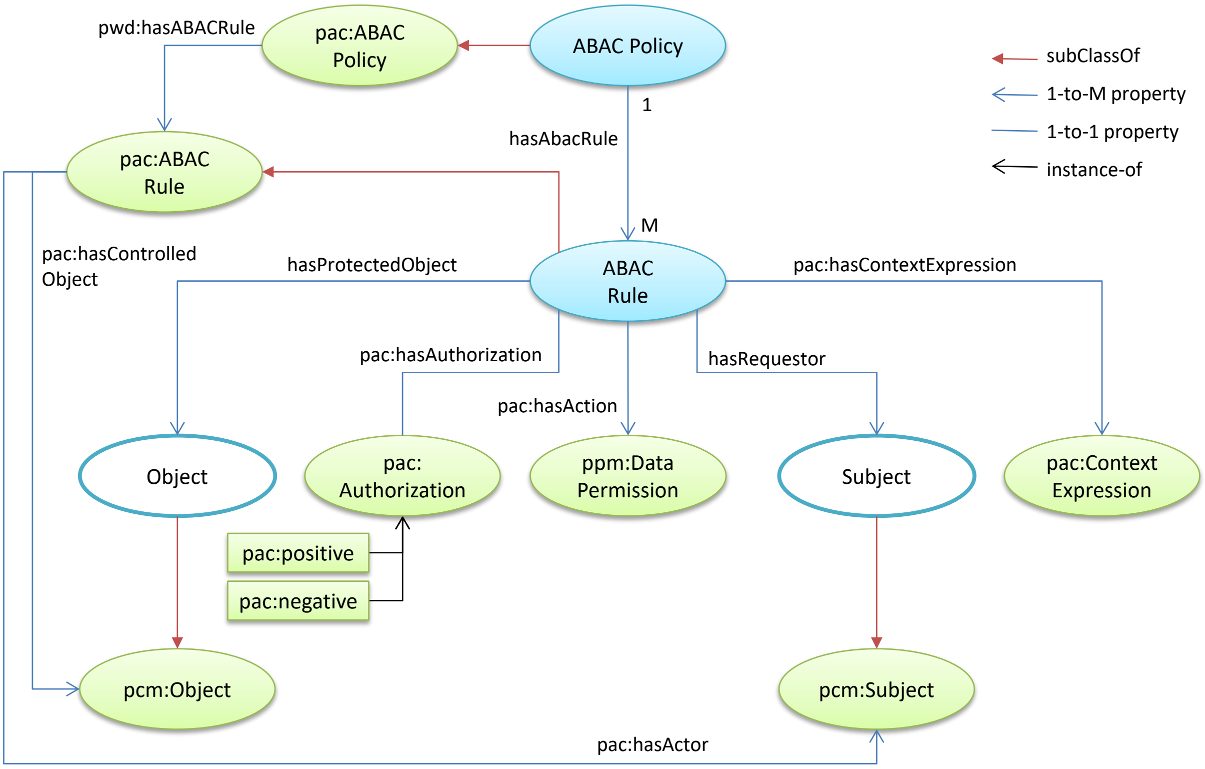


Figure 16: ABAC Policy Model and its mapping onto PaaSword policy model

The classes and properties of the ABAC Policy Model are described next:

Class Path (Hierarchically)	Class Description	Property / Description
ABAC Policy class	Instances of this class represent ABAC policies. An ABAC policy instance acts as the "container" of any number of ABAC rules as well as a combining algorithm. ABAC Policy specializes the ASCLEPIOS Policy in Authorization Meta-Model as well as the PaaSword `pac:ABAC Policy`.	***hasAbacRule:*** It is a sub-property of PaaSword `pwd:hasABACRule` object property, associating an ABAC Policy instance with an ABAC rule. An ABAC Policy instance can reference any number of instances of ABAC Rules.
ABAC Rule class	Represents an ABAC rule, which is the building block for creating ABAC policies. A rule includes a context expression that checks the state of the used contextual attributes and yields a certain results (positive or negative). ABAC Rule specializes PaaSword `pac:ABAC Rule`.	***hasProtectedObject:*** It is a sub-property of the object property `pac:hasControlledObject` of PaaSword, associating an ABAC rule instance with an instance of *Object* class of ASCLEPIOS security context model, which is sub-class of `pac:Object` of PaaSword. An ABAC Rule can reference one or more instances of Object class.
		pac:hasAuthorization: It is an object of PaaSword, associating an ABAC rule instance with an instance of `pac:Authorization`. An

		ABAC Rule can reference exactly one instance of Authorization; either `pac:positive` or `pac:negative`. ***pac:hasAction:*** It is an object property of PaaSword, associating an ABAC rule instance with an instance of class `pac:Data Permission` of PaaSword. An ABAC Rule can reference any number of Authorization class instances. ***hasRequestor:*** It is a sub-property of `pac:hasActor` object property of PaaSword, associating an ABAC rule instance with an instance of *Subject* class of security context model of ASCLEPIOS, which in turn is sub-class of `pac:Subject` of PaaSword. An ABAC Rule can reference any number of instances of Subject class. ***pac:hasContextExpression:*** It is an object property of PaaSword, associating an ABAC rule instance with an instance of class `pac:Context Expression` of PaaSword. An ABAC Rule can reference exactly one instance of *Context Expression* class.
Object class	Represents a protected resource. It sub-classes the `pcm:Object` of PaaSword, thus providing interoperability with it. See also Table 8 in chapter 7	
pcm:Object class	Represents a protected resource in PaaSword security context model.	
Subject class	Represents a requestor (an entity that attempts or requests access to a protected resource). It specializes the `pcm:Subject` of PaaSword, thus providing interoperability with it. See also Table 9 in chapter 7	
pcm:Subject class	Represents a requestor or any other entity in PaaSword security context model.	
pac:Authorization class	Represents the outcome of a rule that has been evaluated (triggered). It has two instance : `pac:positive` and `pac:negative`.	
ppm:Data Permission class	Represents an access operation (action) that can be attempted on a protected resource. Examples of such actions are read, write of a file, and execution of an	

	online service. It is part of *PaaSword permission model (ppm)*.	
pac:Context Expression class	Represents a Boolean expression that identifies the conditions that must hold in order to permit or deny an access request to a protected object. The required conditions are captured as expressions involving attributes, which are instances of classes of security context model presented in chapter 7. When a context expression is evaluated to true it triggers the corresponding ABAC rule, thus the associated authorization is returned. Context Expression class is part of *PaaSword access control model (pac)*.	

Table 12: ABAC Policy Model classes and properties

More information on PaaSword policy model classes and properties can be found in deliverable D2.2, section 2.2.

8.3.1.1 Ontological Representation of a Context Expression

A context expression appears as an instance of the class pac:ContextExpression. It specifies a number of constraints on the values of one or more instances drawn from the pcpm:ContextPattern and pcm:SecurityContextElement vocabularies defined in [22,23]. The class pac:ContextExpression is associated with these vocabularies through the object properties pac:hasPatternParameter and pac:hasParameter respectively – see Figure 17.

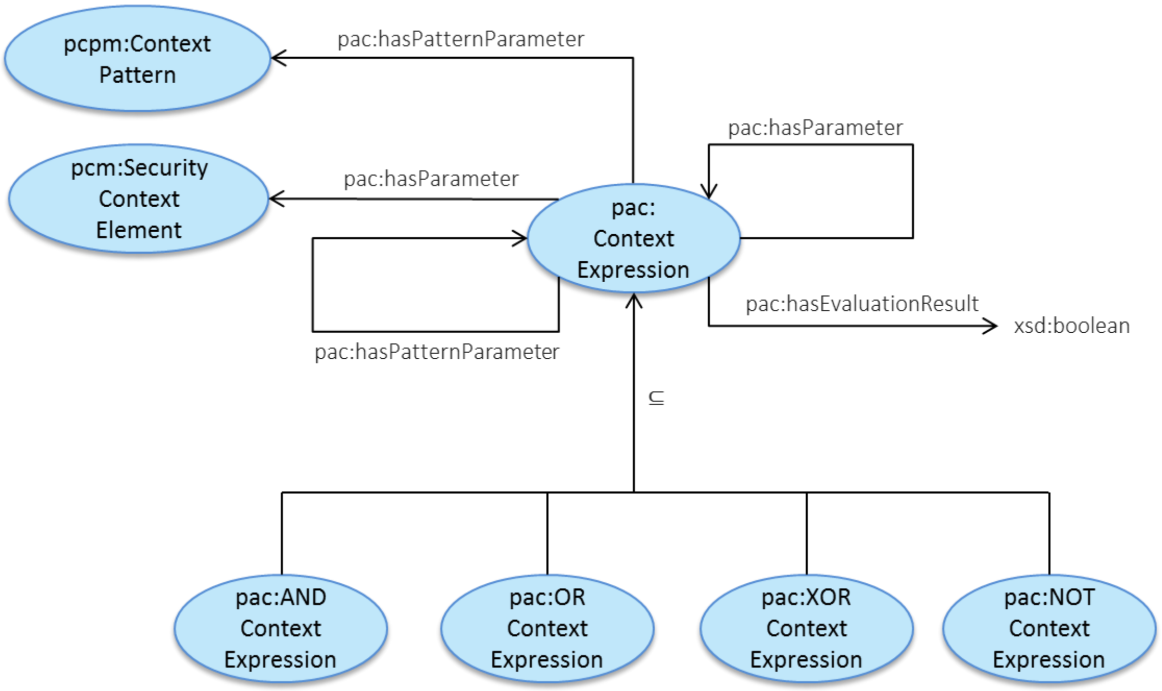


Figure 17: PaaSword Context expression template [22,23]

8.3.1.2 Combining Algorithms

An ABAC policy may encompass several ABAC rules which can potentially yield different results (permit and deny). This situation requires a way of reconciling the conflicting decisions and determines the overall outcome for the ABAC policy. For this reason, an algorithm is named *combining algorithm* [49] is used. An example of a combining algorithm is the *deny-overrides* algorithm, where an ABAC policy evaluation results to *deny* if at least one of its rules results to *deny*, or if none of the rules evaluates to *permit*. A number of combining algorithms are defined in XACML standard (an implementation of ABAC). In ABAC Policy Model a combining algorithm takes the form of an instance of class `pac:PolicyCombiningAlgorithm` of PaaSword. A combining algorithm is attached to an ABAC policy instance through the object property `pac:hasPolicyCombiningAlgorithm`.

8.4 ABE Authorization Policy Model

In this section the ABE authorization policy model is presented. The purpose of this model is to capture in a formal and generic way the concepts related to ABE authorization. It corresponds to a formal and abstract description, therefore concrete ABE policies (used to protect certain resources) can then be seen as instantiations of the ABE policy model.

In ontological terms ABE Policy Model derives from the Authorization Meta-Model presented in section 8.2. Specifically, the ASCLEPIOS Policy abstract class is specialized to ABE Policy, as well as the related properties. Moreover, additional classes are added that capture concepts and functioning specific to ABE policies; for instance ABE Expressions.

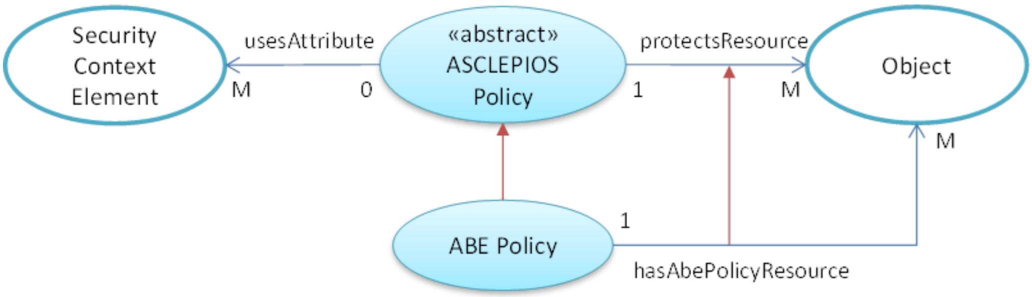


Figure 18: ABE Policy Model as specialization of Authorization Meta-Model

The use of a policy model for describing ABE policies enables their semantic uplifting, and allows for several actions; for instance, inferencing on the validity and compatibility of policies. In addition, a policy model makes possible the use of ABE-specific concepts in other ontological domains.

ABE enforces the protection of resources through encryption and authorizes access to them only when certain predetermined attributes are featured. These attributes (the required values) are defined in the form of an ABE policy. In ASCLEPIOS, ABE policies are used to derive a decryption key used to decipher the encryption/decryption symmetric key of a resource [50].

The next figure provides an overview of the ABE authorization policy model classes and their interrelations. The classes depicted with ovals with white fill represent classes imported from the Security Context model presented in chapter 7. Ovals with cyan fill represent classes introduced in the ABE Policy model. All the elements of the ABE Policy Model (Figure 19) are explained in Table 13.

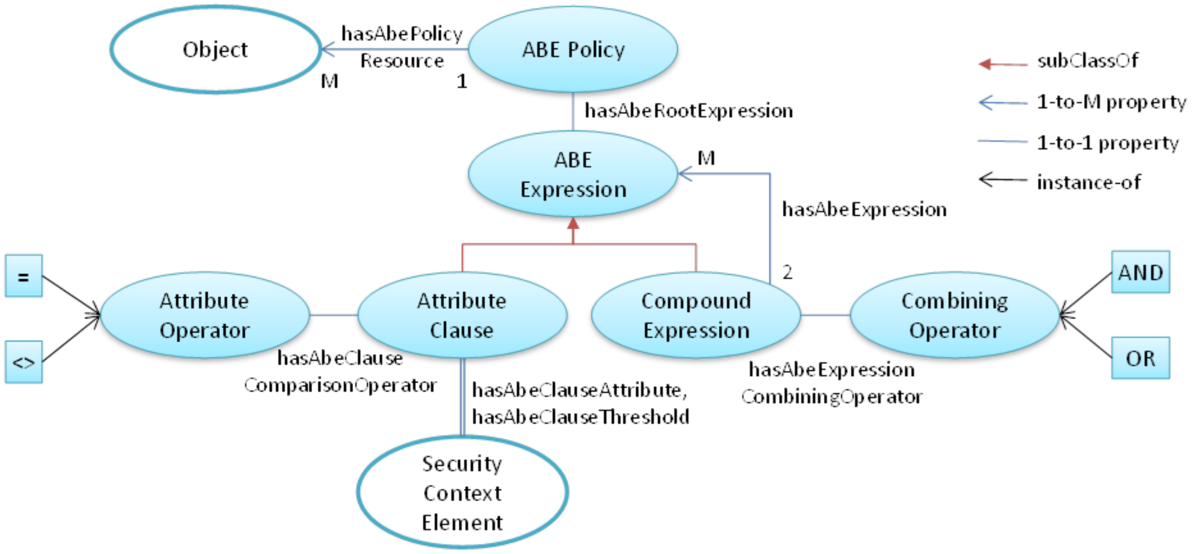


Figure 19: ABE Policy Model

The classes of the ABE Policy Model are described next:

Class Path (Hierarchically)	Class Description	Property / Description
ABE Policy class	Instances of this class represent ABE policies. An ABE policy instance acts as the "container" of an ABE expression that defines the actual ABE policy content (i.e. the attribute set and the way they must be combined to derive the decryption key). An ABE Policy is applied on a specific resource or set of resources.	***hasAbeRootExpression:*** It is an object property associating an ABE Policy class instance with an ABE expression, which defines the policy in ABE terms. An ABE Policy instance can reference exactly one ABE Expression instance. ***hasAbePolicyResource:*** It is an object property associating an ABE Policy class instance with an instance of Object class (from ASCLEPIOS Security Context model), representing a protected resource. An ABE Policy instance can reference one or more Object instances.
ABE Expression class	Represents a Boolean expression of attributes. Attributes act as placeholders or variables that are replaced with concrete values when expression is evaluated. An expression can be compound if it contains other (sub-)expressions, or it can be a simple clause involving a comparison operation on a single attribute. Simple clauses are combined with Boolean operators like AND and OR to formulate compound expressions. A compound expression may also encompass other compound expressions apart from simple clauses. These two types of expressions are captured in the ABE policy model as two subclasses of ABE Expression (see next).	

	As already mentioned, an ABE Policy is effectively defined by a corresponding ABE expression that captures the attribute set used to derive the decryption key and the way they must be combined.	
ABE Expression / *Attribute Clause* class	Represents a clause capturing a comparison operation regarding a single attribute. Therefore, the clause also captures the corresponding operator and possibly a constant value. Examples of operators are <, >, ≤, ≥, =, ≠, BETWEEN, IN. The Security Context Elements, presented in section 8, are used as attributes. As already mentioned the Attribute clause operators are used in Attribute expressions and always yield a Boolean result.	***hasAbeClauseComparisonOperator:*** is an object property associating an Attribute Clause instance with exactly one Attribute Clause Operator instance. ***hasAbeClauseAttribute:*** It is an object property associating an Attribute Clause instance with exactly one instance of a Security Context Element (from ASCLEPIOS Security Context model). ***hasAbeClauseThreshold:*** It is an object property associating an Attribute Clause instance with a threshold used to compare the attribute value against. Threshold instances must be of the same class, or a sub-class, as the Attribute instance of that clause.
ABE Expression / *Compound Expression* class	Represents a complex expression made up of Attribute Clauses or other Compound Expressions. It involves at least two sub-expressions or Attribute clauses and an operator, which specifies how the component expressions & clauses are combined. Typically, compound expression operators are the Boolean operators AND, OR.	***hasAbeExpressionComparisonOperator:*** It is an object property associating an Attribute Compound Expression instance with exactly one Compound Expression Operator instance. ***hasAbeExpression:*** It is an object property associating an Attribute Compound Expression instance with at least two other ABE Expression instances.
Attribute Clause Operator class	As already mentioned Attribute clause operators are used in Attribute expressions and always yield a Boolean result. This class captures the concept of the Attribute clause operator. In the context of ASCLEPIOS we will consider only equality (=) and non-equality (≠) attribute clause operators but the existence of this class enables the definition of additional operators, provided the mechanism underpinning ABE is capable of handling them.	
Compound Expression Operator class	This class captures the notion of Compound expression operators. These are logical operators used to combine sub-expressions and simple clauses, therefore they always yield Boolean results. In the context of ASCLEPIOS we consider only AND and OR however more	

	operators can be defined (by extending ABE policy model) if the mechanism underpinning ABE can handle them.	

Table 13: ABE Policy Model classes and properties

8.5 ABAC and ABE Policy Validation

In this section two types of policy validations are presented and discussed. These validations regard both ABAC and ABE policies, and they are built on top of the ABAC and ABE policy models. Since both an ABAC policy and an ABE policy are required to complete the ASCLEPIOS authorization scheme, the policy validations must be performed on the pair of ABAC and ABE policy considered.

Policy validations verify that a given policy pair meets certain requirements imposed by legislation or corporate guidelines to reach a minimum level of quality. They can be performed when developing the authorization polices pair or just before putting them into effect.

8.5.1 Policy Inspection

It is a type of policy validation requiring that a given pair of ABAC and ABE policies abide to a set of predetermined constraints. Practically, an instantiation of policy inspection will be a set of constraints captured as Guidelines. If a single guideline is violated by a given policy pair, then the pair is considered as not valid. Validity in this context means “not meeting the predetermined requirements”.

Guidelines can be perceived as rules that comprise a condition (a Boolean expression) and an outcome (Valid or Invalid). In the context of ASCLEPIOS three types of Guidelines are considered; Attribute existence checks, Attribute absence checks and Attribute expression existence checks. They can be applied either only on ABAC policy, or only on ABE policy or on both policies. Examples of guidelines can require that “policies consider/include user id attribute”, “ABE policy considers/includes resource creation time” and “ABAC policy requires that expression *‘user communication protocol is HTTPS’* is included in policy rules.”

8.5.2 Security Awareness Assessment

This policy validation identifies whether a given policy pair contains certain security-related weaknesses and reports them. Its outcome is a collection of security awareness alerts that can help in improving the given policies. A security awareness assessment instantiation encompasses a number of Guidelines and each Guideline is associated to a security awareness alert. In the context of ASCLEPIOS the alerts are taken from CAPEC classification [47]. When a guideline is violated the corresponding alert is added in security awareness assessment outcome. An example of security awareness assessment can investigate whether ABAC policy designed include an attribute expression for checking if the communication is over HTTPS, and raise an alert otherwise, like *”CAPEC-102: Session Sidejacking. Make sure that HTTPS …”* that will be added to the security awareness assessment outcome. Obviously, an empty alert collection will be returned for a policy pair with no security-related issues.

8.5.3 Policy Inspection and Security Awareness Assessment Model

In this section the Policy Inspection and Security Awareness Assessment model is presented. The purpose of the model is to capture the concepts related to ABAC and ABE policy validation. Moreover, concrete Guideline checks are defined in this model. We have decided to introduce a single model to accommodate both policy validation types (i.e. policy inspection and security awareness assessment) since they have similar features.

In ontological terms the model derives from the Authorization Meta-Model presented in section 8.2. Specifically, the *Policy Validation* abstract class is specialized to *Policy Inspection* and *Security Awareness Assessment* classes, i.e. one concrete class for each policy validation type. The same applies for *Policy Validation Outcome* abstract class, which is specialized to *Inspection Outcome* and *Security Awareness Outcome* classes. depicts the specialization of Authorization Meta-Model classes and properties to concrete Policy Validation model classes and properties.

Figure 21 provides the Policy Validation model, without including the specialization relations of Figure 20 for the sake of simplicity. This model introduces a few additional classes specializing *Guideline* class, which regard three new concrete policy checks; namely *Attribute Exists Guideline*, *Attribute Not Exists Guideline* and *Attribute Expression Exists Guideline*.

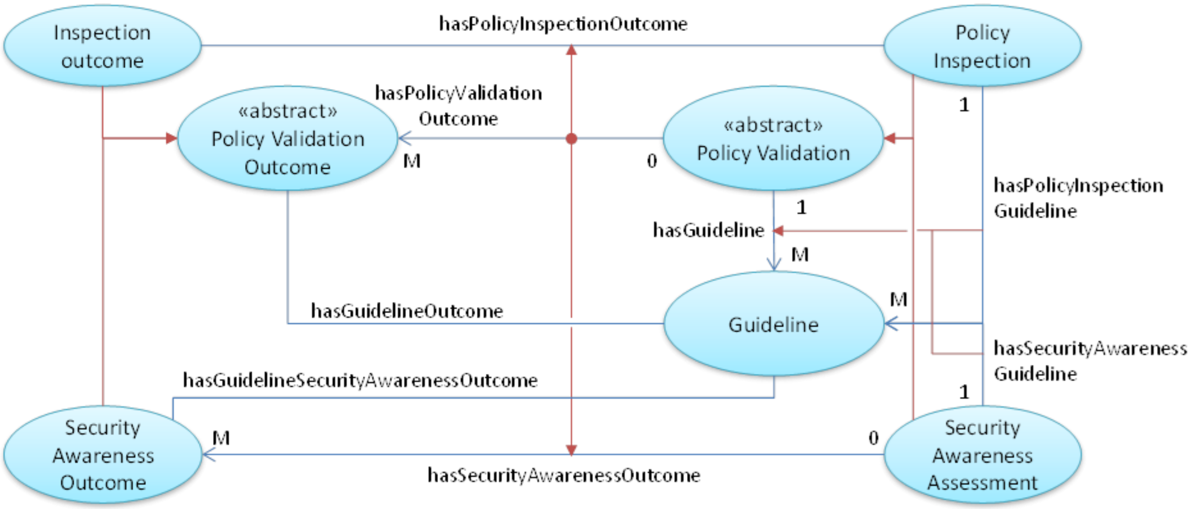

Inspection outcome
hasPolicyInspectionOutcome
Policy Inspection
hasPolicyValidation Outcome
«abstract» Policy Validation Outcome
M
0
«abstract» Policy Validation
1
hasPolicyInspection Guideline
1
hasGuideline
M
hasGuidelineOutcome
Guideline
M
hasSecurityAwareness Guideline
hasGuidelineSecurityAwarenessOutcome
1
Security Awareness Outcome
M
0
Security Awareness Assessment
hasSecurityAwarenessOutcome

Figure 20: Derivation of Policy Inspection and Security Awareness Assessment Model from Authorization Meta-Model

The next figure provides an overview of the Policy Inspection and Security Awareness Assessment model classes and their interrelations.

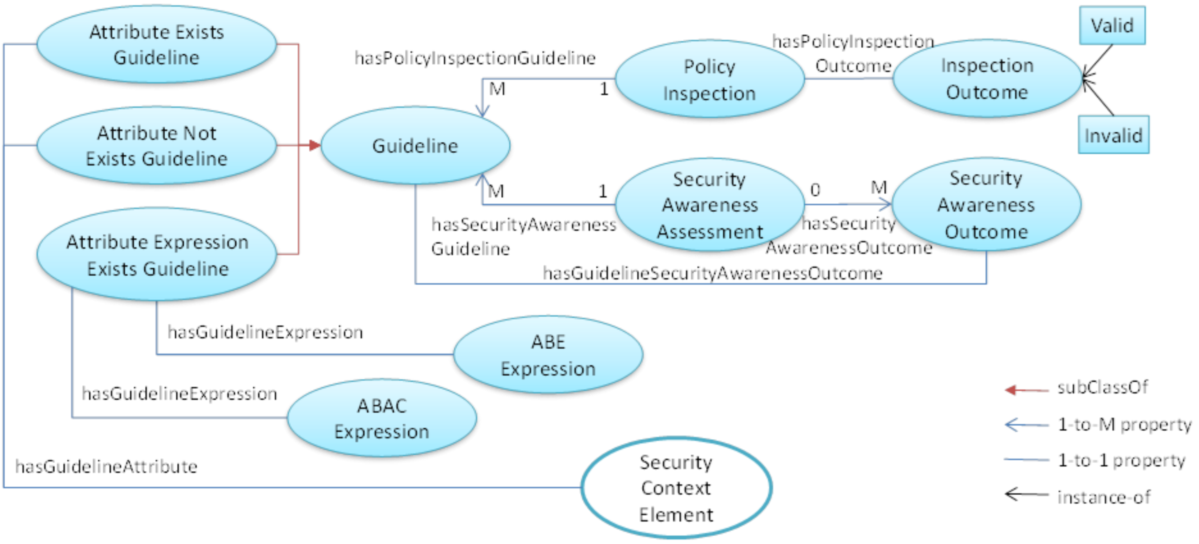

Attribute Exists Guideline
Attribute Not Exists Guideline
Attribute Expression Exists Guideline
hasPolicyInspectionGuideline
M
1
Guideline
Policy Inspection
hasPolicyInspection Outcome
Inspection Outcome
Valid
Invalid
M
1
Security Awareness Assessment
0
M
Security Awareness Outcome
hasSecurityAwareness Guideline
hasSecurity AwarenessOutcome
hasGuidelineSecurityAwarenessOutcome
hasGuidelineExpression
ABE Expression
hasGuidelineExpression
ABAC Expression
hasGuidelineAttribute
Security Context Element
subClassOf
1-to-M property
1-to-1 property
instance-of

Figure 21: Policy Inspection and Security Awareness Assessment Model

The classes of the model are briefly described in the next table.

Class Path (Hierarchically)	Class Description	Property / Description

Policy Validation	An abstract class providing a common parent for all types of policy validations. In the context of ASCLEPIOS we consider two types; Policy Inspection and Security Awareness Assessment.	***hasGuideline:*** It is an abstract property that relates an abstract policy validation to one or more *Guideline* class instances representing specific checks. One policy validation class instance can be associated with one or more *Guideline* instances. ***hasPolicyValidationOutcome:*** It is an abstract object property that associates an abstract policy validation to one *Policy Validation Outcome* class instance. One policy validation class instance can have zero or one *Outcome* instances.
Policy Validation / ***Policy Inspection***	Represents Policy Inspection validation. It is a collection of constraints, called *Guidelines*, which must all be met in order to consider a given policy pair as valid. The outcomes of policy inspection are instances of Inspection Outcome class. A policy inspection (as an operation) takes as input a pair of ABAC and ABE policies, and checks them against the defined Guidelines.	***hasPolicyInspectionGuideline:*** is a sub-property of the abstract *hasGuideline* property and narrows down its domain to *Policy Inspection*. It also narrows the semantics of *hasGuideline* to referring to the checks of a concrete policy inspection instance. ***hasPolicyInspectionOutcome:*** is a sub-property of the abstract property *hasPolicyValidationOutcome* and narrows down its domain to *Policy Inspection*. It also narrows its semantics into referring to the result of a concrete policy inspection instance.
Policy Validation / ***Security Awareness Assessment***	Represents an instance of Security Awareness Assessment. This validation involves a collection of Guidelines, too, which if not met raise certain Security Awareness alerts. In this case guidelines are applied individually, i.e. a single guideline will raise a specific alert if not satisfied.	***hasSecurityAwarenessGuideline:*** is a sub-property of the abstract *hasGuideline* property and narrows down its domain to *Security Awareness Assessment*. It also narrows the semantics of *hasGuideline* to referring to the checks of a concrete security awareness assessment instance. ***hasSecurityAwarenessOutcome:*** is a sub-property of the abstract property *hasPolicyValidationOutcome* and narrows down its domain to *Security Awareness Assessment*. It also narrows its semantics into referring to the result of a concrete security awareness assessment instance.
Guideline	Represents an individual check or meta-constraint. If used in Security Awareness Assessment instances, each guideline must also reference a specific security awareness alert captured as instance of Security Awareness Outcome class. We are currently considering three types of Guidelines; Attribute existence check, Attribute non-existence check, and Attribute expression existence check. There can be three variations of	***hasGuidelineSecurityAwarenessOutcome:*** is an object property associating a Guideline instance used in a security awareness assessment, with a security awareness outcome instance. A guideline can have at most one such outcome.

	Guidelines. Those applied only on ABAC policies, those applied only on ABE policies, and those applicable on both policies (see example at section 9.2).	
Guideline / ***Attribute Exists Guideline*** class	Represents a check for the existence of a specific attribute in a given policy or policy pair. If the specified attribute is contained then this Guideline is met.	***hasGuidelineAttribute:*** It is an object property associating an *Attribute Exists Guideline* instance, with a *Security Context Element* instance, representing an attribute. An Attribute Exists Guideline can have exactly one security context element.
Guideline / ***Attribute Not Exists Guideline***	Represents a check for the absence of a specific attribute in a given policy. If the specified attribute is not contained in the given policy or policy pair, then this Guideline is met. This guideline introduces a kind of *Forbidden-Attribute* check.	***hasGuidelineAttribute:*** It is an object property associating an *Attribute Not Exists Guideline* instance, with a *Security Context Element* instance, representing an attribute. An Attribute Not Exists Guideline can have exactly one security context element.
Guideline / ***Attribute Expression Exists Guideline***	Represents a check for the existence of an expression of attributes. If the attribute expression exists, then the Guideline is met.	***hasGuidelineExpression:*** It is an object property associating an *Attribute Expression Guideline* instance, with an ABAC or *ABE Expression* instance, representing an attribute expression. An Attribute Expression Guideline must refer to exactly one ABAC or ABE Expression.
Policy Validation Outcome	It is an abstract class providing a common parent for all types of policy validation outcomes. In the context of ASCLEPIOS we consider two types; Policy Inspection Outcome and Security Awareness Outcome.	
Policy Validation Outcome / ***Inspection Outcome***	This class represents the outcome of policy inspection checks. In the context of ASCLEPIOS its instances are `Valid` and `Invalid`.	
Policy Validation Outcome / ***Security Awareness Outcome***	This class represents the outcome of security awareness assessments, which are the security awareness alerts raised by Guidelines. The outcome of a single security awareness assessment is a collection of alerts (many can be raised). The instances of this class can be taken from any security awareness classification. In the context of ASCLEPIOS we have selected CAPEC classification [47].	

Table 14: Policy Validation Model classes and properties

9 Exploitation Authorization Policies

This chapter provides an overview of the authorization process and the functioning of the ABAC and ABE policies (presented in the previous chapter) in the context of ASCLEPIOS framework. It also provides an indicative example including ABAC and ABE policies, sample access requests as well as policy validation checks.

9.1 Authorization based on ABAC and ABE Policies

In the context of ASCLEPIOS, we propose the combined use of ABAC and ABE policies for authorization. We consider a two-step process where ABAC policy is first applied on access attempts to resources (either data or functionality) and subsequently, if an ABAC permit is granted, ABE policy is applied in order to recover the resource decryption key. Naturally, ABE is applicable only for data resources persisted in an encrypted form. Dynamically generated data or software functionality (e.g. offered through APIs) cannot be protected with ABE.

This approach brings forth certain advantages over the individual use of ABAC or ABE policies. Some of these are:

- Mitigation of ABE inability to distinguish between read and modification operations. When the resource decryption key is recovered (using ABE), the user can virtually perform any kind of operation on the resource
- Enhancement of the limited set of operations usually implemented in ABE with a rich set of operations already defined and standardized for ABAC (especially with regards to XACML implementation of ABAC).
- Enhanced security on persisted data even if ABAC mechanism is bypassed, since data are stored in an encrypted form.

An outline of the proposed authorization process would be:

1. Intercept (and temporarily block) an attempt to access a protected resource. The access attempt can be an HTTP request.
2. Collection of attributes pertaining to the requestor, the data owner, the resource being accessed, the attempted action as well as other environmental attributes. Attributes are acquired from a trusted attribute authority.
3. Evaluation of the access attempt against ABAC policy (or policies).
    a. If evaluation yields `Deny` then the access attempt is permanently blocked and an error is returned to the user.
    b. If evaluation yields `Permit` then
        i. If the protected resource is a service or functionality, then the access attempt is allowed to proceed, and authorization process completes.
        ii. If the protected resource is a persisted (and encrypted) dataset then the authorization process continues to ABE.
4. Use of the ABE policy of the resource along with the acquired attribute values, in order to recover the resource ABE decryption key.
5. ABE decryption key is used to decipher the encrypted symmetric key that decrypts the dataset.
    a. If decryption fails, then an error is returned to the user and access attempt is essentially blocked.
    b. If symmetric key decryption succeeds, the user can continue with dataset decryption.
6. Symmetric key is used to decipher the protected dataset.

The following BPMN diagram depicts the aforementioned authorization process.

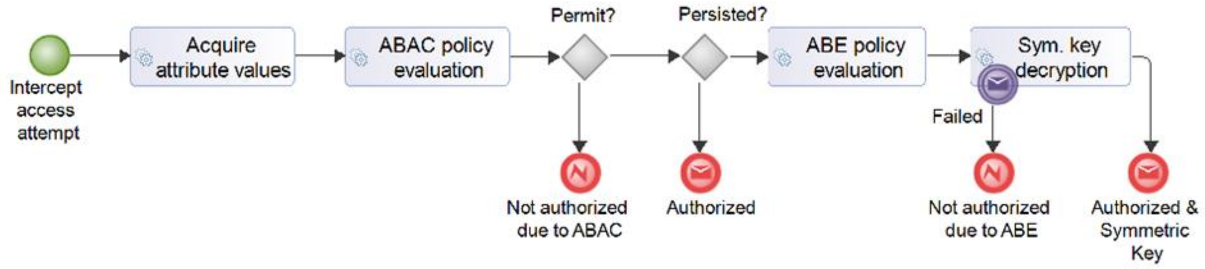


Figure 22: Authorization process (BPMN view)

9.2 Examples of ABAC and ABE Policies, and Policy Validations

In this section an illustrative example is provided to demonstrate the use of ABAC and ABE policies for authorization in ASCLEPIOS. Several authorization cases will be given and discussed. Eventually, a policy pair validation example is provided including both policy inspection and security awareness assessment.

9.2.1 Scenario of the example

Let a physician needs access to patient data in order to perform his/her work. A (patient) data controller, i.e. a person responsible for collecting and making the data available, imposes certain restrictions on data usage (due to GDPR and corporate policies). Specifically:

- Only users who are physicians with “emergency radiology” classification can access data;
- Access to data is possible during a specific period of time;
- Data controller can, at any time, modify data (in order to remove records of a patient that has withdrew his/her consent for using them).

To make the example more concrete let’s assume that:

- User Id of physician is `Physician#45`
- User Id of data controller is `DC#3`
- The dataset path starts with `"/datasets/DS12345/"`
- Access period is from `2019-10-01 00:00:00Z` till `2019-12-31 23:59:59Z`
- The user role is given by `"user-role"` attribute and user classification by `"user-classification"`.

In the following subsections the relevant ABAC and ABE authorization policies are discussed while guidelines for policy inspection and security awareness assessment are also provided.

9.2.2 ABAC Policy

ABAC policy will be used to authorize write operations of data controller and deny write operations of anyone else. Since ABE policies cannot easily distinguish between read and write operations ABAC policy is used for this purpose. ABE policy is applied after ABAC and is responsible for providing the decryption key of the dataset. Therefore, in order a user to access the dataset he/she must first be granted access by ABAC policy and then ABE will provide the dataset decryption key, provided the user has been successfully authenticated and has all necessary attribute values. An operation can be allowed by ABAC policy, but ABE policy might subsequently block it, for instance if the requestor is not a physician.

In practice the ABAC policy will be captured using XACML language and will be evaluated using an XACML-capable authorization engine.

Rule #1, permits write operations on data, only if user is the data controller:

```
IF (user-action="WRITE") AND (user-id="DC#3") AND
    (resource-path STARTS WITH "/datasets/DS12345/")
THEN permit
```

Rule #2, grants access only during specific period:

```
  IF (current-timestamp NOT BETWEEN
      '2019-10-01 00:00:00Z' AND '2019-12-31 23:59:59Z')
  THEN deny
```

Rule #3, grants read access to any user:

```
  IF (user-action="READ") AND
     (resource-path STARTS WITH "/datasets/DS12345/")
  THEN permit
```

Rule #4, denies access in any other case:

```
  IF true THEN deny
```

ABAC policy combining algorithm is "First Applicable", meaning that the first rule that will match (i.e. its IF part evaluates to true) will provide the final result of the policy.

Table 15: ABAC policy example

9.2.3 ABE Policy

In this example ABE policy is responsible for granting access only to legitimate users (physician or the specific data controller). If all constraints are met, then the SSE key is decrypted and based on it the relevant health records.

```
(User-id="DC#3")
OR
(User-role="Physician" AND User-classification="Emergency radiology")
```

Table 16: ABE policy example

User-id="DC#3", User-role="Physician", and User-classification="Emergency radiology" are Attribute Clauses that cannot be decomposed any further. These are used to build the compound expressions, like (User-role="Physician" AND User-classification="Emergency radiology"). In fact, the whole ABE policy is a compound expression.

An alternative representation of an ABE policy can take the form of a syntactic tree.

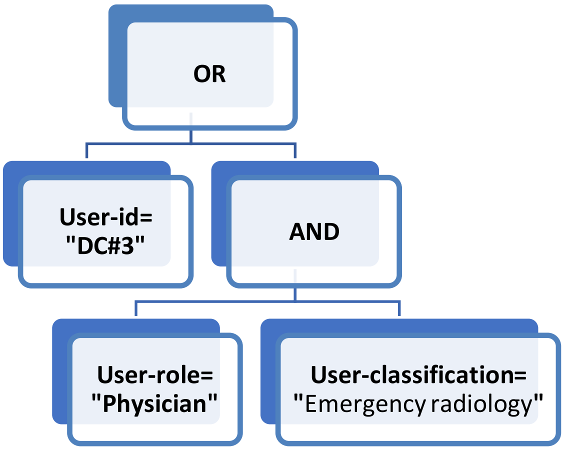


Figure 23: ABE policy example (as syntactic tree)

9.2.4 *Dataset access and Authorization examples*

Before continuing with the discussion of Policy Inspection and Security Awareness Assessment, it would be beneficial to give a few concrete examples of how a access to sensitive data attempts would be checked against the two policies described in the two previous subsections.

9.2.4.1 *Data controller attempts to modify dataset*

Let data controller attempts to modify (write) a dataset record. The related request can be described by the following attributes, which are provided by the designated attribute providers:

- `User-id: DC#3`
- `User-role: Data Controller`
- `User-action: WRITE`
- `Resource-path: /datasets/DS12345/REC98765/FLD2`
- `User-location: Hospital building X, floor Y, office Z`
- `Current-timestamp: 1571590351000 (i.e. 2019-10-20 16:52:09,000Z)`

Request is first checked against ABAC policy. Since ABAC policy utilizes a “Deny unless Permit” combining algorithm, policy rules are checked sequentially until one returns “permit”. In that case ABAC policy will grant access and the request will next be checked against ABE policy. If ABAC policy denies the access then the request is immediately rejected.

Rule 1:

```
IF (user-action="WRITE") AND (user-id="DC#3") AND
   (resource-path STARTS WITH "/datasets/DS12345/")
THEN permit
```

Replacing attribute placeholders with actual attribute values:

```
IF ("WRITE"="WRITE") AND ("DC#3"="DC#3") AND
   ("/datasets/DS12345/REC98765/FLD2" STARTS WITH "/datasets/DS12345/")
THEN permit
```

Obviously, the expression in bold type evaluates to `true` and therefore the rule yields `permit`. Subsequently ABAC policy permits request, due to "`Deny unless Permit`" combining algorithm.

Next, request is checked against ABE policy.

```
(User-id="DC#3")
OR
(User-role="Physician" AND User-classification="Emergency radiology")
```

Replacing attribute placeholders with actual attribute values:

```
("DC#3"="DC#3")
OR
("Data Controller"="Physician" AND ""="Emergency radiology")
```

The first clause before `OR`, evaluates to `true`, therefore the whole expression evaluates to `true`. That means that according to the CP-ABE key creation algorithm, the symmetric key needed to decipher the SSE key will be decrypted.

Eventually this request will result in returning the dataset decryption key to the data controller.

9.2.4.2 *Physician attempts to modify dataset*

Let physician attempts to modify (write) a dataset record. The related request can be described by the following attributes, which are provided by the designated attribute providers:

- User-id: Physician#45
- User-role: Physician
- User-classification: Emergency radiology
- User-action: WRITE
- Resource-path: /datasets/DS12345/REC98765/FLD2
- User-location: Hospital building X, floor Y, office Z
- Current-timestamp: 1571590351000 (i.e. 2019-10-20 16:52:09,000Z)

Request is first checked against ABAC policy. Since ABAC policy utilizes a “Deny unless Permit” combining algorithm, also policy rules are checked sequentially until one returns “permit”. In that case ABAC policy will grant access and the request will next be checked against ABE policy. If ABAC policy denies access then the request is immediately rejected.

Rule 1:

```
IF (user-action="WRITE") AND (user-id="DC#3") AND
   (resource-path STARTS WITH "/datasets/DS12345/")
THEN permit
```

Replacing attribute placeholders with actual attribute values:

```
IF ("WRITE"="WRITE") AND ("Physician#45"="DC#3") AND
   ("/datasets/DS12345/REC98765/FLD2" STARTS WITH "/datasets/DS12345/")
THEN permit
```

Obviously, the expression in bold type evaluates to `false` and therefore this rule does not yield a result. Subsequently the next ABAC policy rule will be evaluated.

Rule 2:

```
IF (current-timestamp NOT BETWEEN
    '2019-10-01 00:00:00Z' AND '2019-12-31 23:59:59Z')
THEN deny
```

Replacing attribute placeholders with actual attribute values:

```
IF ('2019-10-20 16:52:09,000Z' NOT BETWEEN
    '2019-10-01 00:00:00Z' AND '2019-12-31 23:59:59Z')
THEN deny
```

Obviously the expression in bold type evaluates to `false` and therefore this rule does not deny further evaluation. Subsequently the next ABAC policy rule will be evaluated.

Rule 3:

```
IF (user-action="READ") AND
   (resource-path STARTS WITH "/datasets/DS12345/")
THEN permit
```

Replacing attribute placeholders with actual attribute values:

```
IF ("WRITE"="READ") AND
   ("/datasets/DS12345/REC98765/FLD2" STARTS WITH "/datasets/DS12345/")
THEN permit
```

Obviously the expression in bold type evaluates to `false` and therefore this rule does not yield a result. Subsequently the next ABAC policy rule will be evaluated.

Rule 4:

```
IF true THEN deny
```

Obviously this rule yields `deny`. Subsequently ABAC policy denies request, due to `"Deny unless Permit"` combining algorithm, and hence the request is rejected.

Physician will not be able to write to dataset, even if he already has the dataset decryption (and re-encryption) key, since ABAC policy engine will block his/her attempt to modify the resource.

9.2.4.3 A third user attempts to read dataset

Let a third party user attempts to read dataset records. The related request can be described by the following attributes, which are provided by the designated attribute providers:

- `User-id: SomeUser#999`
- `User-role: Unknown`
- `User-action: READ`
- `Resource-path: /datasets/DS12345/REC98765/FLD2`
- `User-location: Hospital building X, floor Y, office Z`
- `Current-timestamp: 1571590351000 (i.e. 2019-10-20 16:52:09,000Z)`

Request is first checked against ABAC policy. Since ABAC policy utilizes a “Deny unless Permit” combining algorithm, also policy rules are checked sequentially until one returns “permit”.

Rule 1:

```
IF (user-action="WRITE") AND (user-id="DC#3") AND
   (resource-path STARTS WITH "/datasets/DS12345/")
THEN permit
```

Replacing attribute placeholders with actual attribute values:

```
IF ("READ"="WRITE") AND ("SomeUser#999"="DC#3") AND
   ("/datasets/DS12345/REC98765/FLD2" STARTS WITH "/datasets/DS12345/")
THEN permit
```

Obviously the expression in bold type evaluates to `false` and therefore this rule does not yield a result. Subsequently the next ABAC policy rule will be evaluated.

Rule 2:

```
IF (current-timestamp NOT BETWEEN
    '2019-10-01 00:00:00Z' AND '2019-12-31 23:59:59Z')
THEN deny
```

Replacing attribute placeholders with actual attribute values:

```
IF ('2019-10-20 16:52:09,000Z' NOT BETWEEN
    '2019-10-01 00:00:00Z' AND '2019-12-31 23:59:59Z')
THEN deny
```

Obviously the expression in bold type evaluates to `false` and therefore this rule does not deny further evaluation. Subsequently the next ABAC policy rule will be evaluated.

Rule 3:

```
IF (user-action="READ") AND
   (resource-path STARTS WITH "/datasets/DS12345/")
THEN permit
```

Replacing attribute placeholders with actual attribute values:

```
IF ("READ"="READ") AND
   ("/datasets/DS12345/REC98765/FLD2" STARTS WITH "/datasets/DS12345/")
THEN permit
```

Obviously the expression in bold type evaluates to `true` and therefore the rule and whole ABAC policy yield `permit`. Subsequently request is authorized by ABAC policy and no more ABAC rules will be evaluated.

Next, request is checked against ABE policy.

```
(User-id="DC#3")
OR
(User-role="Physician" AND User-classification="Emergency radiology")
```

Replacing attribute placeholders with actual attribute values:

```
("SomeUser#999"="DC#3")
OR
("Unknown"="Physician" AND ""="Emergency radiology")
```

The clauses in bold type evaluate to `false` thus causing the whole expression to evaluate to `false`. This means that no combination of the attribute values, using the CP-ABE key creation algorithm, will provide the right secret key needed to decipher the dataset decryption key. Therefore, the third party user will not be able to decipher and read the dataset.

9.2.4.4 Physician attempts to read dataset

Let physician attempts to read a dataset record. The related request can be described by the following attributes, which are provided by the designated attribute providers:

- `User-id: Physician#45`
- `User-role: Physician`
- `User-classification: Emergency radiology`
- `User-action: READ`
- `Resource-path: /datasets/DS12345/REC98765/FLD2`
- `User-location: Hospital building X, floor Y, office Z`
- `Current-timestamp: 1571590351000 (i.e. 2019-10-20 16:52:09,000Z)`

Request is first checked against ABAC policy. Since ABAC policy utilizes a “Deny unless Permit” combining algorithm, also policy rules are checked sequentially until one returns “permit”.

Rule 1:

```
IF (user-action="WRITE") AND (user-id="DC#3") AND
   (resource-path STARTS WITH "/datasets/DS12345/")
THEN permit
```

Replacing attribute placeholders with actual attribute values:

```
IF ("READ"="WRITE") AND ("Physician#45"="DC#3") AND
   ("/datasets/DS12345/REC98765/FLD2" STARTS WITH "/datasets/DS12345/")
THEN permit
```

Obviously the expression in bold type evaluates to `false` and therefore this rule does not yield a result. Subsequently the next ABAC policy rule will be evaluated.

Rule 2:

```
IF (current-timestamp NOT BETWEEN
    '2019-10-01 00:00:00Z' AND '2019-12-31 23:59:59Z')
THEN deny
```

Replacing attribute placeholders with actual attribute values:

```
IF ('2019-10-20 16:52:09,000Z' NOT BETWEEN
    '2019-10-01 00:00:00Z' AND '2019-12-31 23:59:59Z')
THEN deny
```

Obviously the expression in bold type evaluates to `false` and therefore this rule does not deny further evaluation. Subsequently the next ABAC policy rule will be evaluated.

Rule 3:

```
IF (user-action="READ") AND
   (resource-path STARTS WITH "/datasets/DS12345/")
THEN permit
```

Replacing attribute placeholders with actual attribute values:

```
IF ("READ"="READ") AND
   ("/datasets/DS12345/REC98765/FLD2" STARTS WITH "/datasets/DS12345/")
THEN permit
```

Obviously the expression in bold type evaluates to `true` and therefore this rule as well as the whole ABAC policy yield `permit`. Subsequently request is authorized by ABAC policy and no more ABAC rules will be evaluated.

Next, request is checked against ABE policy.

```
(User-id="DC#3")
OR
(User-role="Physician" AND User-classification="Emergency radiology")
```

Replacing attribute placeholders with actual attribute values:

```
("Physician#45"="DC#3")
OR
("Physician"="Physician" AND "Emergency radiology"="Emergency
radiology")
```

The second clause after `OR`, evaluates to `true`, therefore the whole expression evaluates to `true`. This means that the secret key needed to decipher the dataset decryption key is available for decrypting the requested healthcare data.

9.2.5 ABAC and ABE Policy Inspection

Next, we continue the discussion of ABAC and ABE policy inspection and security awareness assessment. It is important to stress that these validations are performed at design time, before putting them into effect. In this example, policy inspection ensures that all valid pairs of ABAC & ABE policies have the following features:

- Both ABAC and ABE policies contain at least one rule or clause including the `user-id` attribute.
- ABAC policy contains a rule or clause including an expression of the form `"(current-timestamp BETWEEN timestamp1 AND timestamp2)"`.

The rationale behind of these checks is to ensure that user identity is taken into consideration by both policies and that access request period is also checked.

The former requirement can be checked with two Attribute Existence Check inspection guidelines. For instance:

```
Guideline 1:
  ABAC RULE EXISTS WITH (ATTRIBUTE "user-id")

Guideline 2:
  ABE CLAUSE EXISTS WITH (ATTRIBUTE "user-id")
```

The latter requirement can be checked with an Attribute Expression Existence guideline. For instance:

```
Guideline 3:
  ABAC RULE EXISTS WITH EXPRESSION "(current-timestamp BETWEEN
<timestamp> AND <timestamp>)")
```

The validation of the two example policies would be performed as described next.

1. Guideline 1 is evaluated first. `ABAC RULE EXISTS` forces iterating through ABAC policy rules until one rule satisfies expression `WITH (ATTRIBUTE "user-id").`
   - Rule 1 of ABAC policy contains `user-id` attribute and satisfies the component guideline. No more rules are checked
2. Guideline 2 is evaluated next. `ABE CLAUSE EXISTS` forces traversing the syntactic tree of the policy until expression `WITH (ATTRIBUTE "user-id")` is satisfied.
   - The first clause of ABE policy `User-id="DC#3"` satisfies this component guideline
3. Guideline 3 is evaluated last. Since `ABAC RULE` has been specified it will check if ABAC satisfies this guideline.

a. It iterates through ABAC policy rules until one rule satisfies expression `WITH EXPRESSION "(current-timestamp BETWEEN <timestamp> AND <timestamp>)"`.
   - Rule 1 of ABAC policy is checked to determine whether it contains an expression of the form `"(current-timestamp BETWEEN <timestamp> AND <timestamp>)"`. It does not; hence the next rule is also checked.
   - Rule 2 of ABAC policy is also checked and it does contain such an expression. Therefore no more rules are checked.

b. Since ABAC policy contains the wanted expression this guideline is satisfied too.

Since all three guidelines are satisfied, this pair of ABAC policy and ABE policy is valid.

9.2.6 ABAC and ABE Policy Security Awareness Assessment

Security awareness assessments do not prohibit a policy pair from being used but they raise security related alerts in case any of the guidelines is not met. Such alerts will be presented to ASCLEPIOS DevOps for raising the security awareness. Contrary to policy inspection, security awareness assessment guidelines act independently, i.e. each guideline raises its own alert if not satisfied. Therefore, multiple alerts can be raised. In this example, security awareness assessment involves:

- Checking if ABAC policy requires that user communication protocol is over HTTPS.
- Checking if ABE policy contains a user role check.

The former requirement can be checked with an Attribute Expression Exists guideline. For instance:

```
Guideline 1:
  NO ABAC RULE EXISTS WITH EXPRESSION "(user-communication-protocol=
'HTTPS')"
ALERT:
  "CAPEC-102: Session Sidejacking. Make sure that HTTPS is used to
communicate with the target system. Alternatively, use VPN if
possible. It is important to ensure that all communication between
the client and the server happens via an encrypted secure channel. "
```

The latter requirement can be checked with an Attribute Expression Exists guideline. For instance:

```
Guideline 2:
 NO ABE RULE EXISTS WITH EXPRESSION "(user-role = <any-string>)"
ALERT:
  "CAPEC-180: Exploiting Incorrectly Configured Access Control
Security Levels. Configure the access control correctly"
```

Guideline evaluation is similar to policy inspection. The former guideline will raise a security alert `"CAPEC-102: Session Sidejacking…"` because neither of the rules of the ABAC policy contains a clause that checks if user communication protocol is HTTPS. However, the latter guideline will not raise a security alert since the ABE policy contains a clause for checking user roles.

10 Conclusions

To conclude, in this deliverable we reported on the development of a context model for i) formally describing classes and properties with respect to associations between types of access to sensitive data and situations under which this access should be permitted, and ii) matching policies between private keys and ciphertexts for permitting decryption of sensitive data, and concepts that capture and highlight possible cyber security threats for enhancing the security awareness of actors in hospitals and care centres. Moreover, this model constitutes the necessary background knowledge layer for enabling the ABAC and ABE paradigms by incorporating policies that can be used with ABE schemes based on the specific needs of healthcare organizations. In order to achieve this, we extended the PaaSword context-aware security model in order to cope with concepts, challenges and standards from the healthcare domain.

Finally, we undertook the research and development of a policy model for formally describing dynamically-generated context-based access control policies, as well as policies concerning the manner in which medical data were decrypted according to the ABE paradigm. This policy model comprised three critical parts: i) an XACML-based part for defining declaratively authorization policies for permitting or denying access requests to sensitive data, in real time; ii) an ABE-oriented part that allowed the declaration of attribute-based dependency between actors' private keys and ciphertexts which upon matching are to be decrypted and iii) security awareness part, for informing the ASCLEPIOS users about the security guarantees implied by the authorization policies defined.

11 References

1. Dey, A. 2001: Understanding and Using Context. Personal Ubi Comp (2001) https://doi.org/10.1007/s007790170019.
2. Ferrari, E. (2010). Access Control in Data Management Systems. Synthesis Lectures on Data Management, 2010, Vol. 2, No. 1, Morgan & Claypool Publishers
3. Khan, A., (2012). Access control in cloud computing environment. ARPN Journal of Engineering and Applied Sciences.
4. Hu, V. C., Ferraiolo, D., Kuhn, R., Schnitzer, A., Sandlin, K., Miller R., and Scarfone K., (2104). Guide to Attribute Based Access Control (ABAC) Definition and Considerations. NIST, 2014.
5. Sahai Amit, and Waters Brent. Fuzzy identity-based encryption. Annual International Conference on the Theory and Applications of Cryptographic Techniques. Springer, Berlin, Heidelberg, 457-473 (2005).
6. Bethencourt, J., Sahai, A., & Waters, B. (2007, May). Ciphertext-policy attribute-based encryption. In 2007 IEEE symposium on security and privacy (SP'07) 321-334. IEEE.
7. Müller, Sascha, Katzenbeisser Stefan, and Eckert Claudia. Distributed attribute-based encryption. International Conference on Information Security and Cryptology. Springer, Berlin, Heidelberg, 20-36 (2008).
8. Moffat, Steve, Mohammad Hammoudeh, and Robert Hegarty. "A survey on ciphertext-policy attribute-based encryption (cp-abe) approaches to data security on mobile devices and its application to iot." Proceedings of the International Conference on Future Networks and Distributed Systems. ACM, 2017.
9. Wang, S., Gao, T., & Zhang, Y. (2018). Searchable and revocable multi-data owner attribute-based encryption scheme with hidden policy in cloud storage. PloS one, 13(11), e0206126.
10. Premkamal, P. K., Pasupuleti, S. K., & Alphonse, P. J. A. (2019, July). Traceable CP-ABE for Outsourced Big Data in Cloud Storage. In International Conference on Computing and Information Technology (pp. 213-226). Springer, Cham.
11. Han, J., Chen, L., Susilo, W., Huang, X., Castiglione, A., & Liang, K. Fine-grained information flow control using attributes. Information Sciences, 484, 167-182 (2019).
12. Lewko Allison and Brent Waters. Decentralizing attribute-based encryption. In: Annual international conference on the theory and applications of cryptographic techniques. Springer, Berlin, Heidelberg, 568-588 (2011).
13. Kumar, Praveen, and P. J. A. Alphonse. Attribute based encryption in cloud computing: A survey, gap analysis, and future directions. Journal of Network and Computer Applications 108: 37-52 (2018).
14. Liu, Z., Jiang, Z. L., Wang, X., & Yiu, S. M. Practical attribute-based encryption: Outsourcing decryption, attribute revocation and policy updating. Journal of Network and Computer Applications 108: 112-123 (2018).
15. Domingo-Ferrer, J., Farràs, O., Ribes-González, J., & Sánchez, D. Privacy-preserving cloud computing on sensitive data: A survey of methods, products and challenges. Computer Communications, 140, 38-60 (2019).
16. Zhang Leyou, Yilei Cui, and Yi Mu. Improving Security and Privacy Attribute Based Data Sharing in Cloud Computing. IEEE Systems Journal (2019).
17. Attrapadung, N. Unbounded Dynamic Predicate Compositions in Attribute-Based Encryption. In Annual International Conference on the Theory and Applications of Cryptographic Techniques (pp. 34-67). Springer, Cham. (2019, May)
18. Hong, J., Liu, B., Sun, Q., & Li, F. A combined public-key scheme in the case of attribute-based for wireless body area networks. Wireless Networks, 25(2), 845-859 (2019).
19. Xu, Q., Tan, C., Zhu, W., Xiao, Y., Fan, Z., & Cheng, F. (2019). Decentralized attribute-based conjunctive keyword search scheme with online/offline encryption

and outsource decryption for cloud computing. Future Generation Computer Systems.

20. Li, Q., Zhu, H., Xiong, J., Mo, R., Ying, Z., & Wang, H. Fine-grained multi-authority access control in IoT-enabled mHealth. Annals of Telecommunications, 1-12 (2019).
21. Liang, P., Zhang, L., Kang, L., & Ren, J. Privacy-preserving decentralized ABE for secure sharing of personal health records in cloud storage. Journal of Information Security and Applications, 47, 258-266 (2019).
22. S. Veloudis, Y. Verginadis, I. Patiniotakis, I. Paraskakis and G. Mentzas. Context-aware Security Models for PaaS-enabled Access Control. 6th International Conference on Cloud Computing and Services Science (CLOSER 2016), Rome, Italy, April 23-25, 2016
23. Veloudis, S., Paraskakis, I., Verginadis, Y., Patiniotakis, I., & Mentzas, G. (2017, June). Ontological Templates for Regulating Access to Sensitive Medical Data in the Cloud. In 2017 IEEE 30th International Symposium on Computer-Based Medical Systems (CBMS) (pp. 805-810). IEEE.
24. ASCLEPIOS Security Context Element overview diagram. Available online at: https://www.asclepios-project.eu/wp-content/uploads/2019/11/ASCLEPIOS_Security_Context_Element_overview_diagram.png
25. ASCLEPIOS Security Context Element Location overview diagram. Available online at: https://www.asclepios-project.eu/wp-content/uploads/2019/12/UML_Location_class_diagram.jpg
26. ASCLEPIOS Security Context Element DateTime overview diagram. Available online at: https://www.asclepios-project.eu/wp-content/uploads/2019/12/UML_DateTime_class_diagram.jpg
27. UML Class Diagram for the Connectivity context element. Available online at: https://www.asclepios-project.eu/wp-content/uploads/2019/11/UML_Class_Diagram_for_the_Connectivity_context_element.svg
28. ASCLEPIOS Security Context Element Object overview diagram. Available online at: https://www.asclepios-project.eu/wp-content/uploads/2019/11/ASCLEPIOS_Security_Context_Element_Object_overview_diagram.png
29. Weber, G. M., Mandl, K. D., & Kohane, I. S. (2014). Finding the missing link for big biomedical data. Jama, 311(24), 2479-2480.
30. Anatomical Therapeutic Chemical Classification System / Defined Daily Dose (ATC/DDD), Available online at: https://www.whocc.no/ddd/list_of_ddds_for_3_years_revisio/
31. Clinical Document Architecture (CDA). Available online at: http://www.hl7.org/implement/standards/product_brief.cfm?product_id=7
32. European Guideline on the electronic exchange of health data under CrossBorder Directive 2011/24/EU Release 2 ePrescriptions and eDispensations. Available online at: https://ec.europa.eu/health/sites/health/files/ehealth/docs/ev_20161121_co091_en.pdf
33. Logical Observation Identifiers Names and Codes (LOINC). Available online at: https://loinc.org/faq/structure-of-loinc-codes-and-names/
34. European Guideline on the electronic exchange of health data under CrossBorder Directive 2011/24/EU Release 2 Patient Summary for unscheduled care. Available online at: https://ec.europa.eu/health/sites/health/files/ehealth/docs/ev_20161121_co10_en.pdf
35. Steimann, F., & Adlassnig, K. P. (1998). Fuzzy medical diagnosis. Handbook of fuzzy computation, G13.

36. International Statistical Classification of Diseases and Related Health Problems 10th Revision (ICD-10), World Health Organization (WHO). Available online at: https://icd.who.int/browse10/2016/en
37. ASCLEPIOS UML class diagram for the Object context element. Available online at: https://www.asclepios-project.eu/wp-content/uploads/2019/12/ASCLEPIOS_UML_Object_class_diagram.svg
38. Extension of Security Context Element Subject overview diagram. Available online at: https://www.asclepios-project.eu/wp-content/uploads/2019/11/ASCLEPIOS_Security_Context_Element_Subject_overview_diagram.png
39. European Skills / Competences, qualifications and Occupations (ESCO), Current version ESCO v1.0.3 (Last update 26/04/2018), Available online at: https://ec.europa.eu/esco/portal/occupation?uri=http%3A%2F%2Fdata.europa.eu%2Fesco%2Fisco%2FC22&conceptLanguage=en&full=true#&uri=http://data.europa.eu/esco/isco/C22#&uri=http://data.europa.eu/esco/isco/C22
40. National Uniform Claim Committee(NUCC), Version 19.1, 7/1/19, Available online at: http://www.nucc.org/index.php/code-sets-mainmenu-41/provider-taxonomy-mainmenu-40/csv-mainmenu-57
41. International Labour Organization (ILO), Updating the International Standard Classification of Occupations (ISCO), ISCO-08 Group Definitions: Occupations in Health, Available online at: http://www.ilo.org/public/english/bureau/stat/isco/docs/d7b.doc
42. European Skills / Competences, qualifications and Occupations (ESCO). Available online at: https://ec.europa.eu/esco/portal/skill?uri=http://data.europa.eu/esco/skill/ed3f3dba-3a35-4ed5-b113-67f4d10ef4c8&conceptLanguage=en&full=true&resetLanguage=true&newLanguage=en&skillFilterIndex=0#&uri=http://data.europa.eu/esco/skill/ed3f3dba-3a35-4ed5-b113-67f4d10ef4c8&treeSelection=C4
43. International Labour Organization (ILO). Available online at: https://www.ilo.org/global/about-the-ilo/lang--en/index.htm
44. European Skills / Competences, qualifications and Occupations (ESCO). Available online at: https://ec.europa.eu/esco/portal/occupation?uri=http%3A%2F%2Fdata.europa.eu%2Fesco%2Foccupation%2F021663ca-a367-472e-a1f0-6d8ab5b2d860&conceptLanguage=en&full=true#&uri=http://data.europa.eu/esco/occupation/021663ca-a367-472e-a1f0-6d8ab5b2d860#&uri=http://data.europa.eu/esco/occupation/021663ca-a367-472e-a1f0-6d8ab5b2d860
45. Conceptual Framework for the International Classification for Patient Safety, Version 1.1, World Health Organization (WHO). Available online at: https://www.who.int/patientsafety/taxonomy/icps_full_report.pdf
46. ASCLEPIOS UML class diagram for the Subject context element. Available online at: https://www.asclepios-project.eu/wp-content/uploads/2019/12/ASCLEPIOS_UML_Subject_class_diagram.svg
47. Common Attack Pattern Enumeration and Classification (CAPEC), MITRE. Available online at: https://capec.mitre.org/
48. Security awareness component. Available online at: https://www.asclepios-project.eu/wp-content/uploads/2019/11/CAPEC_ASCLEPIOS_Security_Awareness_Component.png
49. “eXtensible Access Control Markup Language (XACML) Version 3.0.,” 22 January 2013. [Online]. Available: http://docs.oasis-open.org/xacml/3.0/xacml-3.0-core-spec-os-en.html

50. Michalas, A. The Lord of the Shares: Combining Attribute-Based Encryption and Searchable Encryption for Flexible Data Sharing. In Proceedings of the 34th ACM/SIGAPP Symposium On Applied Computing (SAC'19). Limassol, Cyprus, April 08 – 12, 2019.
51. National Drug Code (NDC), Food and Drug Administration (FDA), Available online at: https://www.fda.gov/drugs/drug-approvals-and-databases/national-drug-code-directory
52. Health Level Seven Version 3 Standard (HL7 V3): Patient Administration, Person Registry, Release 1. Available online at: https://www.hl7.org/implement/standards/product_brief.cfm?product_id=376
53. Current Procedural Terminology (CPT), American Medical Association. Available online at: https://www.ama-assn.org/practice-management/cpt/cpt-overview-and-code-approval
54. Systematized Nomenclature of Medicine - Clinical Terms. Available online at: http://www.snomed.org

Annex I. Medical Taxonomies

In general, medical terms could have different meaning among some members of the medical community, or it could be possible to use different terms which have the same meaning. Under these circumstances, there was the need of a cohesive and structured vocabulary. In order to enable the medical community worldwide to communicate under a common vocabulary, with the same meaning and without vagueness, a medical international nomenclature was established. Thus, medical taxonomies were defined. There are a lot of such taxonomies which serve this cause and ensure the cohesiveness of the medical terms. We briefly present below some of the most used ones.

A classification of drugs internationally was assigned by the Food and Drug Administration of USA which designated specific codes for drugs that constitute the National Drug Code (NDC) taxonomy [51]. First of all, NDC taxonomy includes the following categories: the “Product ID” which is a forty character code, the “Product NDC” which is an eight digit code which also defines the first eight digits of the product id, the “Product Type Name” which describes the general type of the drug e.g. “Human OTC Drug” (OTC drugs are those sold directly to a consumer without the prescription from a healthcare professional as opposed to prescription drugs), the “Proprietary Name” which is the brand name e.g. “aspirin adult low dose aspirin”, the “Proprietary Name Suffix”, the “Non Proprietary Name” which is an official generic given to a pharmaceutical drug in order to make communication more precise e.g. “Aspirin”, the “Dosage Form Name” e.g. “tablet, delayed release”, the “Route Name” which is the way of taking the drug e.g. “oral”, the “Start Marketing Date” e.g. “20070112”, the “End Marketing Date”, the “Marketing Category Name” e.g. “OTC monograph final”, the “Application Number” e.g. “part343”, the “Labeller Name” e.g. “Liberty Pharmaceuticals, Inc”., the “Substance Name” e.g. “Aspirin”, the “Active Numerator Strength” e.g. “81”, the “Active Ingrid Unit” e.g. “mg/1”, the “Pharm Classes” which describes the drug’s pharmacological classification like its chemical type, the “DEA Schedule” which is a classification by United States Drug Enforcement Administration, according to the need of prescription for controlled substances, the “NDC Exclude Flag” which indicates whether the product has been removed/excluded from the NDC Directory and the “Listing Record Certified Through” which indicates the date when the drug will expire if not updated or certified by the firm.

The Health Level Seven Version 3 (HL7 V3) Normative Edition [52] which is a suite of specifications based on HL7’s Reference Information Model (RIM), provides a single source that allows implementers of V3 specifications to work with the full set of messages, data types, and terminologies needed to build a complete implementation. The Version 3 Normative Edition represents a new approach to clinical information exchange based on a model driven methodology that produces messages and electronic documents expressed in XML syntax. The V3 specification is built around subject domains that provide storyboard descriptions, trigger events, interaction designs, domain object models derived from the RIM, hierarchical message descriptors (HMDs) and a prose description of each element. Implementation of these domains further depends upon a non-normative V3 Guide and normative specifications for: data types; the XML technical specifications (ITS) or message wire format; message and control "wrappers", and transport protocols. This classification of patient demographic information, which refers to a set of international standards for transfer of clinical and administrative data between software applications used by various healthcare providers, includes the following categories: the “Partial or complete patient name” which is printed on the patient record or is told by the patient, the “Patient ID” which may be obtained from printed barcode or a bed-side chart etc., the “Partial ID entry or scan”, the “Date of birth / age range” and the “Bed ID”.

The classification of Procedures in healthcare area is set by the American Medical Association through the Current Procedural Terminology (CPT) [53] which is a taxonomy that describes medical, surgical, and diagnostic services and is designed to communicate uniform information about medical services and procedures among physicians, coders, patients, accreditation organizations, and payers for administrative, financial, and analytical purposes. According to this classification, procedures are defined by a ten digit code number and they are classified in the following main sections: “Anesthesia”, “Surgery”, “Radiology”, “Pathology and Laboratory procedures”, “Medicine Services and Procedures”, “Evaluation and Management Services”, “Category II Codes”, “Multianalyte Assay”, “Category III Codes” and the “Laboratory Analyses”.

A classification of medical specialties was assigned by the National Uniform Claim Committee (NUCC) [40]. This NUCC taxonomy includes the following categories:

- the “Code” which is a ten character code
  e.g. “1223P0221X”
- the “Grouping” which describes the general group of the specialty
  e.g. “Dental Providers”
- the “Classification” which describes the specialty
  e.g. “Dentist”, “Oral Medicinist”, “Dental Hygienist”, “Dental Therapist”
- the “Specialization”
  e.g. “Pediatric Dentistry”
- the “Definition”
  e.g. “An age-defined specialty that provides both primary and comprehensive preventive and therapeutic oral health care for infants and children through adolescence, including those with special health care needs”
- “Notes”
  e.g. “Source: Council on Dental Education and Licensure, American Dental Association”

Systematized Nomenclature of Medicine - Clinical Terms (SNOMED CT) [54] is a systematically organized computer processable collection of medical terms providing codes, terms, synonyms and definitions used in clinical documentation and reporting. SNOMED CT is considered to be the most comprehensive, multilingual clinical healthcare terminology in the world. SNOMED CT is maintained and distributed by SNOMED International, an international non-profit standards development organization. SNOMED CT was created in 2002.Below is a list of the Top Level Concepts with a brief description of the content represented in their branch of the hierarchy.

- Clinical finding - the result of a clinical observation, assessment or judgment (e.g. asthma, headache).
- Procedure - activities performed in the provision of health care (e.g. appendectomy, physiotherapy, subcutaneous injection).
- Situation with explicit context - concepts in which the clinical context is specified as part of the definition of the concept itself (e.g. endoscopy arranged, past history of myocardial infarction, family history of glaucoma).
- Observable entity - a question or assessment which can produce an answer or result (e.g. systolic blood pressure, colour of iris, gender).
- Body structure - normal and abnormal anatomical structures (e.g. mitral valve structure, adenosarcoma).
- Organism - organisms of significance in human and animal medicine (e.g. streptococcus pyogenes, beagle).

- Substance - general substances, the chemical constituents of pharmaceutical/biological products, body substances, dietary substances and diagnostic substances (e.g. methane, insulin, albumin).
- Pharmaceutical/biologic product - drug products (e.g. amoxicillin 250mg capsule, paracetamol + codeine tablet).
- Specimen - entities that are obtained (usually from the patient) for examination or analysis (e.g. urine specimen, prostate needle biopsy specimen).
- Special concept - concepts that do not play a part in the formal logic of the concept model of the terminology, but which may be useful for specific use cases (e.g. navigational concept, alternative medicine poisoning).
- Physical object - natural and man-made physical objects (e.g. vena cava filter, implant device, automobile).
- Physical force - physical forces that can play a role as mechanisms of injury (e.g. friction, radiation, alternating current).
- Event - occurrences excluding procedures and interventions (e.g. flood, earthquake).
- Environments and geographical locations - types of environments as well as named locations such as countries, states and regions (e.g. intensive care unit, academic medical centre, Denmark).
- Social context - social conditions and circumstances significant to health care (e.g. occupation, spiritual or religious belief).
- Staging and scales - assessment scales and tumour staging systems (e.g. Glasgow Coma Scale, FIGO staging system of gynaecological malignancy).
- Qualifier value - the values for some SNOMED CT attributes, where those values are not subtypes of other top level concepts. (e.g. left, abnormal result, severe).
- Record artefact - content created for the purpose of providing other people with information about record events or states of affairs. (e.g. patient held record, record entry, family history section).
- SNOMED CT Model Component - contains technical metadata supporting the SNOMED CT release.

Logical Observation Identifiers Names and Codes (LOINC) [33] is a database and universal standard for identifying medical laboratory observations. A fully specified name in LOINC includes

Field	Description	Examples
Component (analyte)	The name of the component or analyte measured	Potassium, Hemoglobin, Hepatitis C antigen.
Property measured	The characteristic of how the component is being measured	A mass concentration, Enzyme activity (catalytic rate).
Timing	Whether the measurement is an observation at a moment of time, or an observation integrated over an extended duration of time	24-hour urine.
System	The type of sample	Urine, Blood
Scale	Whether the measurement is: 1. quantitative (a true measurement) 2. ordinal (a ranked set of options) 3. nominal (that do not have a natural ordering) 4. narrative	1. The mm diameter of the inhibition zone. 2. An antimicrobial susceptibility that can be reported as resistant, intermediate, susceptible. 3. E. coli, Staphylococcus aureus. 4. Dictation results from x-rays.
Method	Where relevant, the methodology used to produce the result or other observation.	Rapid Plasma Reagin, Branched chain DNA (bDNA).

Table 17: LOINC taxonomy's fields - example

International Classification of Diseases (ICD) [36] is an international standard (diagnostic classification) for reporting diseases and health conditions. It defines the universe of diseases, disorders, injuries and other related health conditions, listed in a comprehensive, hierarchical fashion that allows for: easy storage, retrieval and analysis of health information for evidenced-based decision-making; sharing and comparing health information between hospitals, regions, settings and countries; data comparisons in the same location across different time periods. The uses of ICD include monitoring of the incidence and prevalence of diseases, observing reimbursements and resource allocation trends, and keeping track of safety and quality guidelines. They also include the counting of deaths as well as diseases, injuries, symptoms, reasons for encounter, factors that influence health status, and external causes of disease.
ICD-10 taxonomy, which is the latest version, includes the following fields:

Field	Description	Example
ICD10Chapter	Chapter in which this entity is located	01
ICD10Code	ICD-10 code for the entity. Note that the groupings do not have a code.	1A07.Z
ICD10Title	Title of the entity	Typhoid fever, unspecified
ICD10ClassKind	Class kind for the ICD-10 entity. It is one of the three (chapter, block, category). Chapter is top level classification entities. Blocks are high level groupings that do not bear a code. Categories are entities that have a code.	Category

Table 18: ICD-10 taxonomy's fields – example

The ATC/DDD taxonomy [30] is a standard for international drug utilization monitoring and research. The Anatomical Therapeutic Chemical (ATC) is a classification system. Defined Daily Dose (DDD), which serves as a measuring unit, is the assumed average maintenance dose per day for a drug used for its main indication in adults. It is developed by the Norwegian Institute of Public Health “World Health Organization Collaborating Centre (WHOCC) for Drug Statistics Methodology” as a modification and extension of the European Pharmaceutical Market Research Association (EphMRA) classification system.
The ATC taxonomy includes the following fields:

- ATC code - the several classification levels are depicted in the code
- Name - the name of the chemical substance or the name of the ATC level
    - 1st level - main anatomical or pharmacological group
    - 2nd level - pharmacological or therapeutic group
    - 3rd level - chemical, pharmacological or therapeutic subgroup
    - 4th level - chemical, pharmacological or therapeutic subgroup
    - 5th level - the chemical substance

ATC code	Name
A	Alimentary tract and metabolism (1st level, anatomical main group)
A10	Drugs used in diabetes (2nd level, therapeutic subgroup)
A10B	Blood glucose lowering drugs, excl. insulins (3rd level, pharmacological subgroup)
A10BA	Biguanides (4th level, chemical subgroup)
A10BA02	metformin (5th level, chemical substance)

Table 19: ATC taxonomy's fields – example

The combined ATC/DDD classification includes the following additional fields:

- DDD (Defined Daily Dose) - the number of units of the dose
- Unit - the unit in which the dose is measured e.g. milligram, millilitre etc.
- Route of administration - the way of taking a drug e.g. inhalation, nasal, oral etc.

ATC code	ATC level name or generic name	DDD	Unit	Administration Route
G04BX15	pentosan polysulfate sodium	0.3	g	O (Oral)

Table 20: ATC/DDD taxonomy's fields – example

The Clinical Document Architecture (CDA) [31] is a document markup standard that specifies the structure and semantics of "clinical documents" for the purpose of exchange between healthcare providers and patients. It is developed by Health Level Seven International (HL7). HL7 is a not-for-profit standards developing organization, dedicated to providing a comprehensive framework and related standards for the exchange, integration, sharing, and retrieval of electronic health information that supports clinical practice and the management, delivery and evaluation of health services. More specifically, there is a range of complexity allowed within the specification and users must set their own level of compliance. CDA introduces the concept of incremental semantic interoperability. A minimal CDA consists of a small number of XML-encoded metadata fields such as provider name, document type, document identifier, and a body which can be any commonly-used Multipurpose Internet Mail Extensions (MIME) type such as pdf or doc or even a scanned image file.

The "GUIDELINE on the electronic exchange of health data under Cross-Border Directive 2011/24/EU Release 2 – Patient Summary for unscheduled care" [34] utilizes this taxonomy and has the following fields:

- Patient Administrative data include:
  - Identification
  - Personal information
  - Contact information
  - Insurance information

- Patient Clinical data include:

  - Alerts
  - Medical history
  - Medical problems
  - Medication summary
  - Social history
  - Pregnancy history
  - Physical findings
  - Diagnostic tests

- Metadata include:
  - Country
  - Patient Summary (PS)
  - Nature of the PS
  - Author organization

The "GUIDELINE on the electronic exchange of health data under Cross-Border Directive 2011/24/EU Release 2 – ePrescriptions and eDispensations" [32] utilizes this taxonomy and has the class ePrescriptions.
The Clinical information for cross-border exchange about ePrescription/eDispensation data are set according to the provisions in the "GUIDELINE on the electronic exchange of health data under Cross-Border Directive 2011/24/EU Release 2 – ePrescriptions and eDispensations" which was adopted by the eHealth Network on 21 November 2016.

- ePrescriptions data include:
  - Identification of the patient
  - Authentication of the prescription
  - Identification of the prescribing health professional
  - Identification of the prescribed product
  - Prescription information