
*Integrated cyber-physical security for health services*

# Specification of the Central Database

Deliverable 6.4

Lead Author: CSI

Contributors: LINKS, CCS, ASLTO5, AMC, AP-HM, ENC, KUL

Deliverable classification: PU

Co-funded by the Horizon 2020 Framework Programme of the European Union

**Version Control Sheet**

| | |
|---|---|
| Title | *Specification of the Central Database* |
| Prepared By | *CSI* |
| Approved By | |
| Version Number | *7.0* |
| Contact | |

Revision History:

| Version | Date | Summary of Changes | Initials | Changes Marked |
|---|---|---|---|---|
| 0.0 | 11/06/2019 | Initial Version | CSI | |
| 1.0 | 14/10/2019 | Provision of paragraphs 3, 4, 5, 6, 7 | CSI | |
| 2.0 | 29/10/2019 | Revised version including comments from partners | CSI | |
| 3.0 | 30/10/2019 | Revised version including comments from partners Airbus, SPF and ISEP | CSI | |
| 4.0 | 05/11/2019 | Revised version | CSI | |
| 5.0 | 06/11/2019 | Revised version including comments from partner LINKS | CSI | |
| 6.0 | 13/11/2019 | Revised version after OpenMaint evaluation and modification to HAMS_T_AVAILABILITY table by LINKS | CSI | |
| 7.0 | 19/11/2019 | Revised version after CNAM request | CSI | |

*The Research leading to these results has received funding from the European Union's Horizon 2020 Research and Innovation Programme, under Grant Agreement no 787002*

# Contents

## LIST OF FIGURES

## List of Acronyms and Definitions

| | |
|---|---|
| BMS | Building Management System |
| BTMS | Building Threat Monitoring System |
| CDB | Central Database |
| CTMS | Cyber Threat Monitoring System |
| DB | Database |
| DDoS | Distributed Denial of Service |
| DoA | Document of Agreement |
| E-health | Digital health |
| Ext JS | Extended JavaScript |
| GIS | Geographic Information System |
| GNU GPL | "GNU's Not Unix "General Public License |
| HAMS | Hospital Availability Management System |
| ID | Identification |
| IPM | Impact Propagation Model |
| IT | Information Technology |
| JSON | JavaScript Object Notation |
| MQTT | Message Queuing Telemetry Transport |
| OS | Open Source |
| OT | Operational Technology |
| PC | Personal Computer |
| REST | Representational State Transfer |
| RSE | Remote Support Engineer |
| SMS | Short Message Service |
| SOA | Service Oriented Architecture |
| SOC | Security Operation Center |
| TRAS | Threat Response and Alert System |
| WP | Work Package |

# The SAFECARE Project

Over the last decade, the European Union has faced numerous threats that quickly increased in their magnitude, changing the lives, the habits and the fears of hundreds of millions of citizens. The sources of these threats have been heterogeneous, as well as weapons to impact the population. As Europeans, we know now that we must increase our awareness against these attacks that can strike the places we rely upon the most and destabilize our institutions remotely. Today, the lines between physical and cyber worlds are increasingly blurred. Nearly everything is connected to the Internet and if not, physical intrusion might rub out the barriers. Threats cannot be analysed solely as physical or cyber, and therefore it is critical to develop an integrated approach in order to fight against such combination of threats. Health services are at the same time among the most critical infrastructures and the most vulnerable ones. They are widely relying on information systems to optimize organization and costs, whereas ethics and privacy constraints severely restrict security controls and thus increase vulnerability. The aim of this project is to provide solutions that will improve physical and cyber security in a seamless and cost-effective way. It will promote new technologies and novel approaches to enhance threat prevention, threat detection, incident response and mitigation of impacts. The project will also participate in increasing the compliance between security tools and European regulations about ethics and privacy for health services. Finally, project pilots will take place in the hospitals of Marseille, Turin and Amsterdam, involving security and health practitioners, in order to simulate attack scenarios in near-real conditions. These pilot sites will serve as reference examples to disseminate the results and find customers across Europe.

# Executive Summary

The challenge of SAFECARE is to bring together the most advanced technologies from the physical and cyber security spheres to achieve a global optimum for systemic security and for the management of combined cyber and physical threats and incidents, their interconnections and potential cascading effects. The project focuses on health service infrastructures and works towards the creation of a comprehensive protection system, which will cover threat prevention, detection, response and, in case of failure, mitigation of impacts across infrastructures, populations and environment.

The SAFECARE comprehensive protection system includes a single central database (CDB), which stores static data from assets involved in the project and constitutes the pillar stone in order to build added-value from storing dynamic data, generated by domino effects of different incidents, both physical and cyber.

Cross connecting data expands the capacity to create either more consistent results or innovative results. The central database will therefore include a dynamic data store and a static data store.

In the following, the document provides specifications of the central database in order to produce and design the conceptual model of the central database, which will be used in the future deliverable D6.5 “Central database” to implement the software application to manage it.

# 1. Introduction

SAFECARE project aims at leveraging most updated technologies in IT and OT fields, in order to combine information coming from physical and cyber monitoring systems in a unique place where such information can be grouped, managed and accessed in a secure way. This is the minimum requirement that allows innovative software to elaborate response that can effectively improve the security of healthcare critical infrastructure.

According to the SAFECARE vision, all this information is being stored in the Central database, that is logically divided into two parts:

- Static data store, containg static data, such as assets,
- Dynamic data store, containing dynamic data, such as incidents and impacts, and various responses/elaborations coming from the decisional modules as: impacts, threat responses, hospital availabilities.

The central role of the database is well described by the Figure 1, designed under the task T6.1 and described in the deliverable D6.1 "Specification of the global architecture". This figure reports the whole SAFECARE system and highlights the central role of the Central database and the role of the Data exchange layer, that acts as an interface among the central database and the other modules.

The SAFECARE global architecture described in Figure 1 shows the integration among the different SAFECARE system modules:

- The WP4 Physical Security Solutions
- The WP5 Cyber Security Solutions
- The WP6 Integrated Solutions,

All of these modules (including the decisional ones) interact with the intercommunication level performed by the Data exchange layer and, through this, with the Central Database.

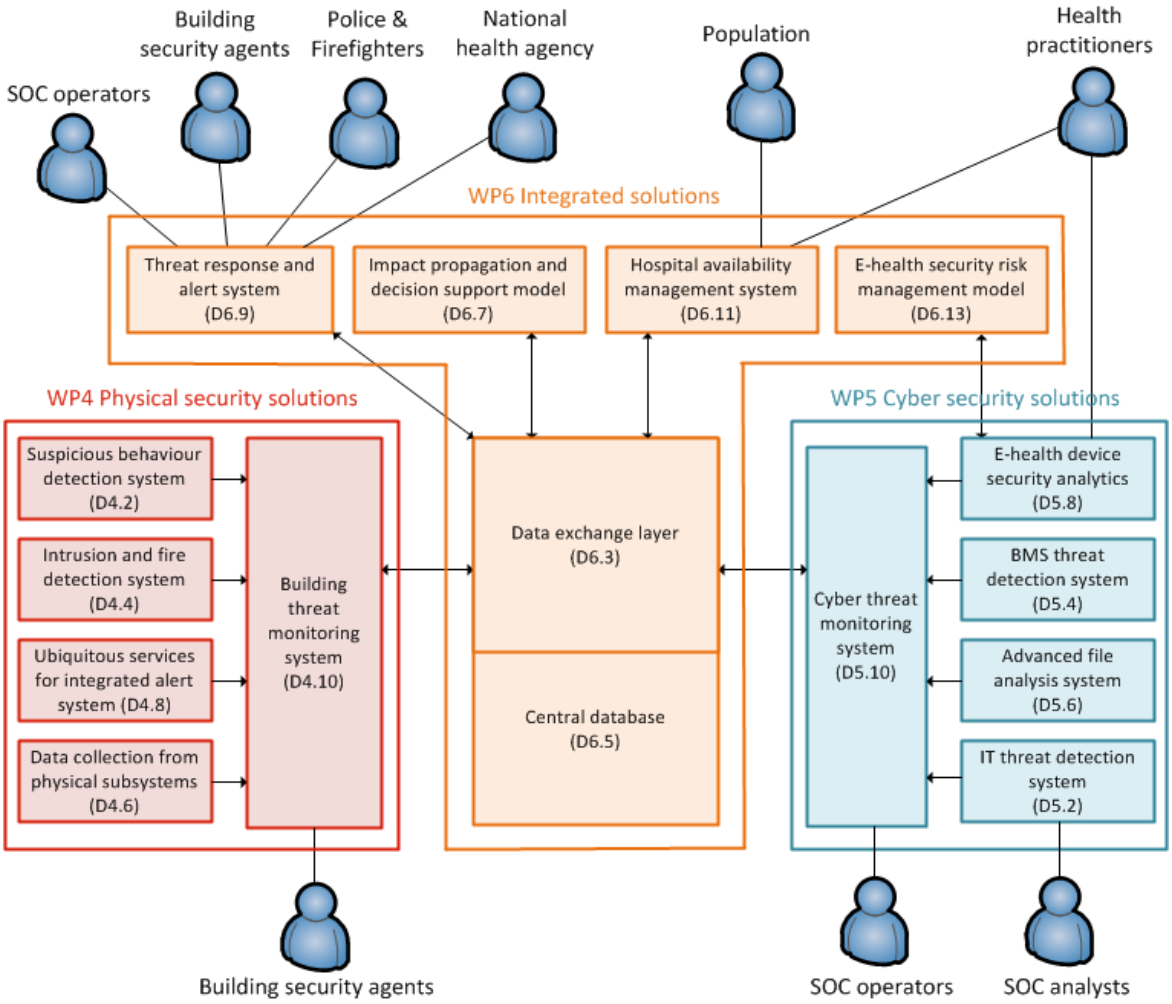


*Figure 1. Global Architecture*

## 1.1. Deliverable 6.4

The aim of this document is to provide the specification of the Central Database, describing the design of the structure of static and dynamic data and their correlations, in order to provide the data model of the central database. Section 2 describes the adopted methodology. Section 3 defines the role of the central database in the whole SAFECARE system. Section 4 describes the identified conceptual schema, beginning with the interactions, through sequence diagrams, among logical entities and the components of the system. Following these definitions, the section provides the logical model of the central database, from which derives the physical model, described in Section 5. Finally, while Section 6 reports the conclusions and some indications about the future works.

# 2. Methodology

At the very beginning of the project, SAFECARE partners agreed on the issue that INCIDENTs as main logical entity should be stored in the central database. According to this starting point, we will collect the object of the domain. We will provide a definition for each of them. Afterwards we will provide the conceptual data model: objects and relations among them.

In the conceptual scheme, we provide the logical entity and component of the system. In order to explain better the relations among central database and other external elements of SAFEARE system, we provide also logical entities NOT stored in the database, but useful for a better understanding of how the information system will work.

# 3. Role of the Central Database in the Global Architecture

As described in Section 1, the Central Database is the main database in the SAFECARE system, and it stores mainly two different types of data:

- Static data
- Dynamic data

Static data contains a census of all the medical devices, physical security devices, cyber security devices, doors, windows, rooms, buildings, computers, networks, servers, etc. All these objects are generically called "assets". All the assets involved in the project will be stored in Central Database, and for each of them it's assigned an ID code, valid in the whole SAFECARE system, which has nothing to do with the fabric ID of the asset nor with the ID assigned by the hospital. From the moment of storage in CDB, in all messages sent through data exchange layer every asset will refer only to the "SAFECARE ID". Components like BTMS or CTMS, that manage directly assets and sensors, should have conversion tables in order to compose the messages with the correct ID code.

According to the DoA of SAFECARE, the Central Database will also store static data as static referential of data which can be non-exhaustively critical physical and cyber assets, names of the buildings, names of the rooms, normalized scales defining asset status and impact gravity.

Static data can be extracted from the Central database through dedicated REST APIs that will be developed in the Data Exchange Layer[1].

Dynamic data are all information and messages generated by other modules of the SAFECARE project, such as incidents, impacts, threat responses and availability data.

Dynamic data will be stored in the Central Database, and extracted through the Data Exchange Layer (see D6.2 document, "Specification about the Data Exchange Layer"). Data Echange Layer will implement web services in order to dynamically check the data format and the data content before storing, by checking static referential of data (e.g. list of critical assets, scale of impacts, names of rooms and buildings). The Data Exchange Layer will allow other project components to extract added-value information on demand from the central database.

Moreover, the Central Database could contain data eventually inserted directly by operators of SAFECARE systems, such as information about the assets, locations and/or sensors present in the hospital. Data contained in the Central Database will be available for analysis by other components, accessible through REST API.

As already described in the deliverable D6.1 "Specification of the global architecture", WP4 Physical Security modules produce EVENTS. If if the event is confirmed as a potential security risk by Building Threat Monitoring System (BTMS), it becomes an ALERT. After a human check, each alert, as an INCIDENT, is forwarded to the Central Database through the Data Exchange Layer and stored as dynamic data.

Similarly, for the WP5 Cyber Security modules, when EVENTS are pointed out and if the event is confirmed as a potential security risk by the Cyber Threat Monitoring System (CTMS), it is transformed into an ALERT. After a human check, each alert, as an INCIDENT, is forwarded via the Data Exchange Layer to the Central Database and stored as dynamic data, as well.

[1] See deliverable D6.2 "Specification of the Data Exchange Layer"

## 3.1 Analysis of OpenMaint for the management of assets

For the management of the static data in the Central Database, CSI has evaluated if to take advantage also of a specific software named openMAINT, developed by Tecnoteca SrL, open to reuse, following the rules of GNU Affero GPL.

OpenMaint is an OS application for the management of buildings, installations, movable assets and related maintaining activities.

The main functionalities that have been tested to be used in SAFECARE, are space and assets inventory, and GIS support. For each asset, openMAINT manages both predefined standard and customized information. The application has a user interface in order to store and query data.

OpenMAINT basic functionalities and technical features are:

• Service Oriented Architecture (SOA), organized in components and services, which cooperate also with external tools through webservice;

• Ajax user interface (Ext JS frameworks);

• components in the server realized with Java Enterprise environment;

• PostgreSQL database.

OpenMAINT uses only open source components: PostgreSQL database and PostGIS, GeoServer and OpenLayers for the GIS functionalities support.

CSI verified the advantages deriving from use of OpenMaint toward the availability, organization and amount of the data to be treated, coming from the local hospital structures.

During the WP6 analysis phase, the partners have defined a data structure suitable for the Impact Propagation Model, for what it pertains the assets management. During the test phase of OpenMaint, the data structure of OpenMaint has been compared with the one previously defined for IPM. We verified directly in the OpenMaint database structure, that assets management in OpenMaint is based on data organized in one asset table only, while the IPM analysis has been based on a flexible data organization, in which the attributes of the asset is stored and described in different tables, in order to make easier to change attribute's properties without any change of database structure.

As assets management in Central Database must fully cope with the Impact Propagation Model asset description, to completely benefit from the OpenMaint adoption, OpenMaint has been considered insufficient and therefore discarded.

It' important now to recall the definitions of objects pertaining to the SAFECARE datamodel, that are described in the following sections:

- EVENT: any security fact, both physical and cyber, that is noticed by human surveillance or technical instruments
- ALERT: any event checked by automatic devices, that is suitable to be forwarded to the BTMS or CTMS.
- INCIDENT: any alert checked by humans, suitable to be forwarded to the central database through the Data Exchange Layer
- STATIC DATA: static referential of data concerning physical and cyber security, as name of buildings, address, information on assets, personal references, etc

- THREAT RESPONSE: result of an application of the threat response module: action to be implemented when threat arises.
- IMPACT: result of an application of the impact propagation module, coming from physical and cyber incidents on assets.
- AVAILABILITY DATA: result of the application of the HAMS module.

# 4 Identification of the conceptual schema

The first step to design the SAFECARE Central Database concern the definition of a conceptual schema that summarize and organize the main entities of the domain and their relations.

Figure 2 shows a sequence diagram that describes how SAFECARE system process an incident and which are the interactions among different software modules involved in the processing.

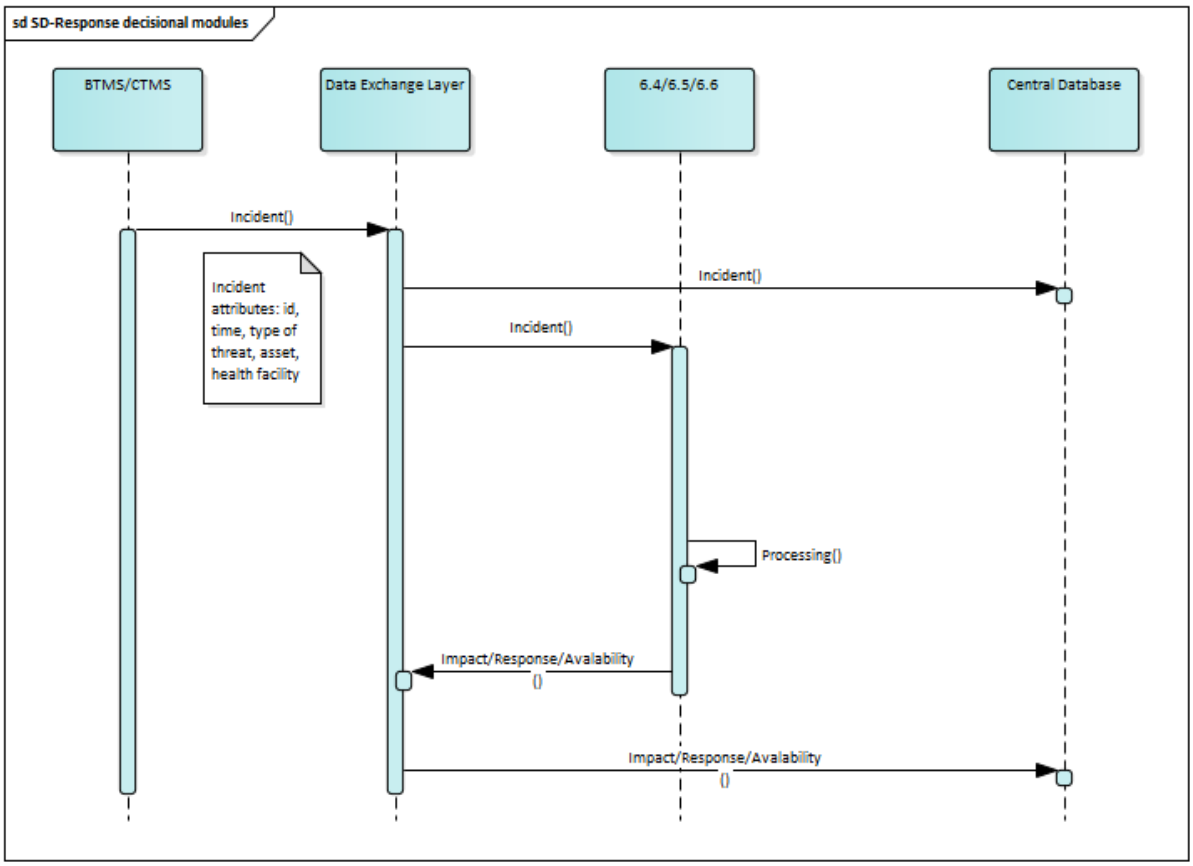


*Figure 2. Sequence diagram of an incident processing*

Entity BTMS is a module of physical security and it's the unique channel from which the physical security modules are connected to the Central Database. Entity CTMS is a module of cyber security and it's the unique channel from which the cyber security solutions are connected the Central Database.

Incidents coming from the BTMS and CTMS are stored through Data Exchange Layer to the Central Database. At the same time, Data Exchange Layer sends notification to the decisional modules (impact propagation model, threat response system model and HAMS) informing them that a new incident has been stored in the CDB.

When an incident is sent to the Data Exchange Layer using MQTT, all the subscribers that are listening to the predefined topic receive the incident.IPM (T6.4) receives the incidents, requires static data from CDB, and, using its correlation rules, elaborates an impact, that is sent to all other components and stored in CDB. TRAS (T6.5) receives the impact, and, using its logic rules, generates a response (stored in CDB) and a notification. In the meantime, HAMS (T6.6), using data from the incident, the impact and the static data from CDB, elaborates an availability message,

that is stored in the CDB. The availability message contains the updated availability status of all the assets involved in the incident.

Results of such elaborations are stored in Central Database as impact, threat response and availability.

From the analysis of the domain, the logical model of the Central Database has been designed.

The logical model, depicted in figure 3 is a representation of entities and relationships of the scope of the project. Squares represent entities, circles are attributes, rhombuses are relationships using the paradigm of Enterprise Architect TM product.

The logical model is divided into two parts:

1. the static data;
2. the incident.

The static data includes two master entities: the “health facility” and the “asset”.

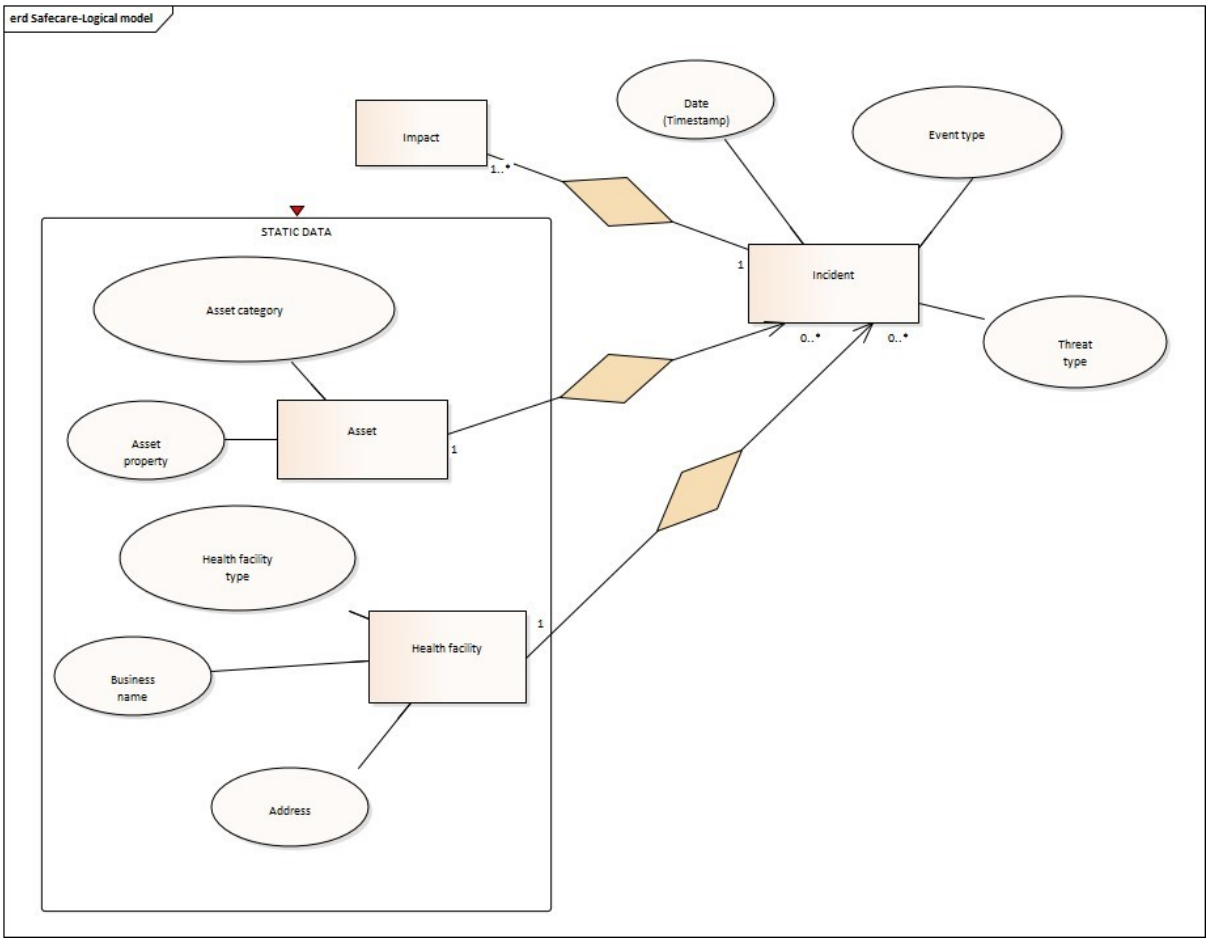


*Figure 3. Logical Model of the Central Database*

“Asset” are the potential targets of threats either cyber or physical; the connection with entity “health facility” that describes the physical structure of a healthcare company, serves to locate the incident.

The incident represents an event potentially dangerous from healthcare company.

## 4.1 Entities and attributes

Entities represented in Figure 3 have the following attributes that characterize them. For the attributes, in some cases necessary, are provided some examples of values.

**Asset**

Asset category:

1. specialist personnel;
2. building and terrain;
3. Appliances;
4. IT.

Asset property.

**Health facility**

Business name;
Address;
Health facility type:

1. Hospital;
2. laboratory;
3. other.

**Incident**

Date of event;
Type of threat:

1. cyber;
2. physical.

Type of event:

1. access to restricted area;
2. digital access to network/systems.

Impact:

1. data breach/loss of data;
2. loss of reputation;
3. damage data center;
4. energy breakdown;
5. contamination of structures;
6. other.

Object of threat (asset and health facility).

# 5. Physical data model

This Section describes the physical data model of the Central Database. The physical data model derives from the logical model. Entities and relationships become table and relational tables. It provides the structure of the data base (both static and dynamic data) following the paradigm of relational database (in attachment to the deliverable is provided the schema in A3 format to use for an easier reading).

Figure 4 describes in detail the physical data model of the Central Database. In the center of this model there is the incident entity, that derives from one or more threat and is linked with one or more adversary. The incident has one or more target and is measured with a different level of severity. Each incident has one or more impact, effecting assets that are linked with alerts by a cross reference table. Each alert is also connected with detector.

The table names are defined as follows:

- The tables named SC_D_xxxxx contain identifier and description.
- The tables named SC_T_xxxxx are supplied from “BTMS” or “CTMS” module.
- The tables named ASS_x_xxxxx contain information of assets.
- The tables named HAMS_T_xxxxx are supplied with information of application of HAMS module.
- The tables named IPM_T_xxxxx are supplied with information of application of Impact module.

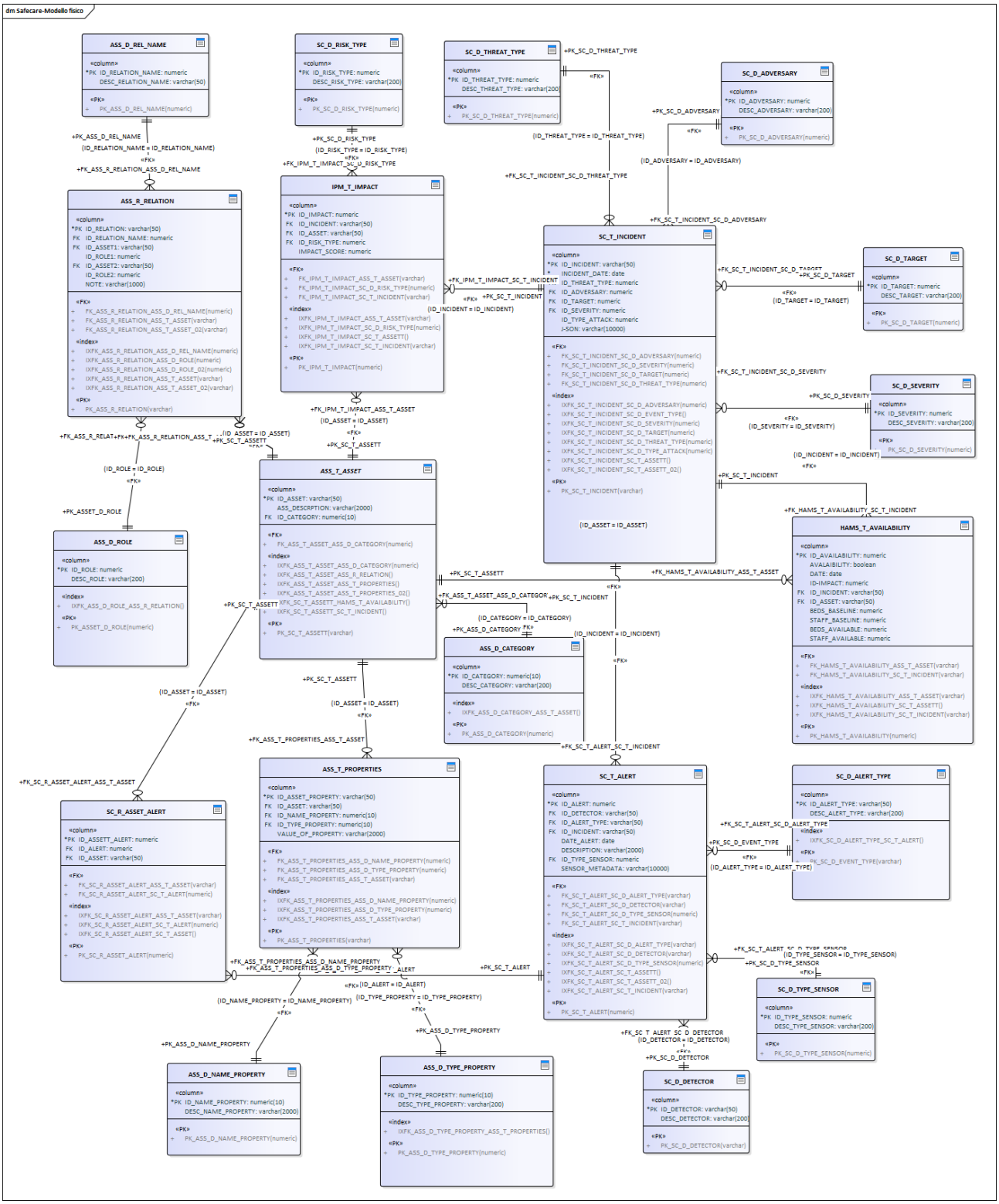


*Figure 3. Physical model of the Central Database*

The tables of model are:

HAMS_T_AVAILABILITY
The Table contains results of incident processing by module HAMS T6.6

| COLUMN NAME | DATATYPE | NOT NULL | COMMENTS |
|---|---|---|---|
| ID_AVALAIBILITY | numeric | True | |
| AVALAIBILITY | bool | True | Full availability or not full availability of the asset |
| DATE | Date | True | Date of last data update |
| ID_IMPACT | numeric | False | Reference to last impact that update data |
| ID_INCIDENT | Varchar(50) | False | Reference to last incident that update data |
| ID_ASSET | numeric | False | Reference to involved asset |
| BEDS_BASELINE | numeric | False | Total number of beds in the asset |
| STAFF_BASELINE | numeric | False | Total number of staff in the asset |
| BEDS_AVAILABLE | numeric | False | Number of beds available |
| STAFF_AVAILABLE | numeric | False | Number of staff available |

| PRIMARY KEY NAME | COLUMNS | COMMENTS |
|---|---|---|
| PK_HAMS_T_AVAILABILITY | ID_AVAILABILITY | |

IPM_T_IMPACT
The Table contains impacts coming from incident processing performed by module Impact Propagation Model T6.4.

| COLUMN NAME | DATATYPE | NOT NULL | COMMENTS |
|---|---|---|---|
| ID_IMPACT | numeric | True | |
| ID_INCIDENT | Varchar(50) | False | |
| ID_ASSET | numeric | False | |
| ID_RISK_TYPE | numeric | False | |
| IMPACT_SCORE | numeric | False | |

| PRIMARY KEY NAME | COLUMNS | COMMENTS |
|---|---|---|
| PK_IPM_T_IMPACT | ID_IMPACT | |

SC_D_ADVERSARY

The table SC_D_ADVERSARY contains the list of possible adversaries responsible of an attack.

| COLUMN NAME | DATATYPE | NOT NULL | COMMENTS |
|---|---|---|---|
| ID_ADVERSARY | numeric | True | The attribute ID_ADVERSARY can take the following values:<br>0-individual or small group;<br>1-political group;<br>2-organized crime;<br>3-terrorists;<br>4-nation states;<br>5-hacker or cyber criminal;<br>6-activists. |
| DESC_ADVERSARY | Varchar(200) | False | |

| PRIMARY KEY NAME | COLUMNS | COMMENTS |
|---|---|---|
| PK_SC_D_ADVERSARY | ID_ADVERSARY | |

SC_D_ALERT_TYPE

The table contains a list of possible types of alert.

| COLUMN NAME | DATATYPE | NOT NULL | COMMENTS |
|---|---|---|---|
| ID_ALERT_TYPE | Varchar(50) | True | |
| DESC_ALERT_TYPE | Varchar(200) | False | The attribute can take the next list of values:<br>• fire;<br>• malicious tool;<br>• intrusion;<br>• identity theft;<br>• credential theft;<br>• ..... |

| PRIMARY KEY NAME | COLUMNS | COMMENTS |
|---|---|---|
| PK_SC_D_EVENT_TYPE | ID_ALERT_TYPE | |

SC_D_DETECTOR

The table contains the list of types of "WP4 physical security solution" or "WP5 cyber security solution".

| COLUMN NAME | DATATYPE | NOT NULL | COMMENTS |
|---|---|---|---|
| ID_DETECTOR | Varchar(50) | True | |
| DESC_DETECTOR | Varchar(200) | False | The attribute can take the next list of values:<br>• suspicious behavior detection system (WP4);<br>• intrusion and fire detection system (WP4);<br>• ubiquitous services for integrated alert system (WP4);<br>• data collection from physical subsystem (WP4);<br>• e-health device security analytics (WP5); |

| COLUMN NAME | DATATYPE | NOT NULL | COMMENTS |
|---|---|---|---|
| | | | • BMS threat detection system (WP5);<br>• advanced file analysis system (WP5);<br>• IT threat detection system (WP5). |

| PRIMARY KEY NAME | COLUMNS | COMMENTS |
|---|---|---|
| PK_SC_D_DETECTOR | ID_DETECTOR | |

SC_D_RISK_TYPE
The table contains a list of possible types of risk.

| COLUMN NAME | DATATYPE | NOT NULL | COMMENTS |
|---|---|---|---|
| ID_RISK_TYPE | numeric | True | |
| DESC_RISK_TYPE | Varchar(200) | False | |

| PRIMARY KEY NAME | COLUMNS | COMMENTS |
|---|---|---|
| PK_SC_D_RISK_TYPE | ID_RISK_TYPE | |

SC_D_SEVERITY

The table SC_D_SEVERITY contains the possible values of severity of an incident.

| COLUMN NAME | DATATYPE | NOT NULL | COMMENTS |
|---|---|---|---|
| ID_SEVERITY | numeric | True | The attribute id_severity can take the following values:<br>0-very low;<br>1-low;<br>2-significant;<br>3-high;<br>4-very high. |
| DESC_SEVERITY | Varchar(200) | False | |

| PRIMARY KEY NAME | COLUMNS | COMMENTS |
|---|---|---|
| PK_SC_D_SEVERITY | ID_SEVERITY | |

SC_D_TARGET

The table SC_D_TARGET contains a list of possible targets of an attack.

| COLUMN NAME | DATATYPE | NOT NULL | COMMENTS |
|---|---|---|---|
| ID_TARGET | numeric | True | The attribute id_target can take the following values:<br>0 - untargeted attack;<br>1 - targeted attack. |
| DESC_TARGET | Varchar(200) | False | |

| PRIMARY KEY NAME | COLUMNS | COMMENTS |
|---|---|---|
| PK_SC_D_TARGET | ID_TARGET | |

SC_D_THREAT_TYPE

The table SC_D_Threat_Type contains the possible types of threat (from WP4 or WP5).

| COLUMN NAME | DATATYPE | NOT NULL | COMMENTS |
|---|---|---|---|
| ID_THREAT_TYPE | numeric | True | The attribute id_threat_type can take the following values:<br>1-cyber threat<br>2-physical threat. |
| DESC_THREAT_TYPE | Varchar(200) | False | |

| PRIMARY KEY NAME | COLUMNS | COMMENTS |
|---|---|---|
| PK_SC_D_THREAT_TYPE | ID_THREAT_TYPE | |

SC_D_TYPE_SENSOR
The table contains a list of possible types of sensor.

| COLUMN NAME | DATATYPE | NOT NULL | COMMENTS |
|---|---|---|---|
| ID_TYPE_SENSOR | numeric | True | |
| DESC_TYPE_SENSOR | Varchar(200) | False | The attribute desc_type_sensor can take the next list of values:<br>1. camera;<br>2. fire detection system;<br>3. mobile app;<br>4. antivirus;<br>5. firewall;<br>6. ..... |

| PRIMARY KEY NAME | COLUMNS | COMMENTS |
|---|---|---|
| PK_SC_D_TYPE_SENSOR | ID_TYPE_SENSOR | |

SC_R_ASSET_ALERT

The table contains the list of assets connected with an alert.

| COLUMN NAME | DATATYPE | NOT NULL | COMMENTS |
|---|---|---|---|
| ID_ASSET_ALERT | numeric | True | |
| ID_ASSET | numeric | False | |
| ID_ALERT | numeric | False | |

| PRIMARY KEY NAME | COLUMNS | COMMENTS |
|---|---|---|
| PK_SC_R_ASSET_ALERT | ID_ASSET_ALERT | |

SC_T_ALERT

The table contains the alert.

| COLUMN NAME | DATATYPE | NOT NULL | COMMENTS |
|---|---|---|---|
| ID_ALERT | numeric | True | |
| ID_DETECTOR | Varchar(50) | False | |
| ID_ALERT_TYPE | Varchar(50) | False | |
| ID_INCIDENT | Varchar(50) | False | |
| DATE_ALERT | date | False | |
| DESCRIPTION | Varchar(2000) | False | |
| ID_TYPE_SENSOR | numeric | False | |
| SENSOR_METADATA | bigserial | False | |

| PRIMARY KEY NAME | COLUMNS | COMMENTS |
|---|---|---|
| PK_SC_T_ALERT | ID_ALERT | |

SC_T_INCIDENT

The table contains the incident.

| COLUMN NAME | DATATYPE | NOT NULL | COMMENTS |
|---|---|---|---|
| ID_INCIDENT | Varchar(50) | True | |
| INCIDENT_DATE | date | True | |
| ID_THREAT_TYPE | numeric | False | |
| ID_ADVERSARY | numeric | False | |
| ID_TARGET | numeric | False | |
| ID_SEVERITY | numeric | False | |
| ID_TYPE_ATTACK | numeric | False | |
| J-SON | bigserial | False | |

| PRIMARY KEY NAME | COLUMNS | COMMENTS |
|---|---|---|
| PK_SC_T_INCIDENT | ID_INCIDENT | |

ASS_T_ASSET

The table contains the list of assets (type, building, technologies….) identified by an "id"; the attributes are the category (from table ASS_D_CATEGORY) and a short description.

| COLUMN NAME | DATATYPE | NOT NULL | COMMENTS |
|---|---|---|---|
| ID_ASSET | Varchar(50) | True | |
| ASS_DESCRIPTION | Varchar (2000) | False | |
| ID_CATEGORY | Numeric (10) | False | |
| ID_ASSET_PROPERTY | Numeric (10) | False | |

| PRIMARY KEY NAME | COLUMNS | COMMENTS |
|---|---|---|
| PK_ASS_T_ASSET | ID_ASSET | |

ASS_D_CATEGORY

The table contains the list of possible categories of assets.

| COLUMN NAME | DATATYPE | NOT NULL | COMMENTS |
|---|---|---|---|
| ID_CATEGORY | Numeric (10) | True | The id_category can assume, for example, the following values:<br>1-Medical device;<br>2-IT asset;<br>3-..... |
| DESC_CATEGORY | Varchar (200) | False | |

| PRIMARY KEY NAME | COLUMNS | COMMENTS |
|---|---|---|
| PK_ASS_D_CATEGORY | ID_CATEGORY | |

ASS_T_PROPERTIES

The table contains, for every asset, the list of properties. Every property is organized as follows:

1. a name of property (described from the table ASS_D_NAME_PROPERTY);
2. a type of property ((described from the table ASS_D_TYPE_PROPERTY);

3. the value of property (string).

| COLUMN NAME | DATATYPE | NOT NULL | COMMENTS |
|---|---|---|---|
| ID_ASSET_PROPERTY | Varchar(50) | True | Identify uniquely a property of an asset. |
| ID_ASSET | Varchar(50) | False | |
| ID_NAME_PROPERTY | numeric(10) | False | |
| ID_TYPE_PROPERTY | numeric(10) | False | |
| VALUE_OF_PROPERTY | Varchar(2000) | False | The attribute contains the value of property link with the asset category. For example: if the asset category is "medical device" and the name of property is name and the property type is "string", then value of property is "Digital radiography system". |

| PRIMARY KEY NAME | COLUMNS | COMMENTS |
|---|---|---|
| PK_ASS_T_PROPERTIES | ID_ASSET_PROPERTY | |

ASS_D_NAME_PROPERTY
The table contains the list of possible name of property that describe an asset.

| COLUMN NAME | DATATYPE | NOT NULL | COMMENTS |
|---|---|---|---|
| ID_NAME_PROPERTY | numeric(10) | True | |
| DESC_NAME_PROPERTY | Varchar(200) | False | Example of values:<br>1. name;<br>2. type;<br>3. model;<br>4. model version; |

| COLUMN NAME | DATATYPE | NOT NULL | COMMENTS |
|---|---|---|---|
| | | | 5. vulnerabilities;<br>6. other..... |

| PRIMARY KEY NAME | COLUMNS | COMMENTS |
|---|---|---|
| PK_ASS_D_NAME_PROPERTY | ID_NAME_PROPERTY | |

ASS_D_TYPE_PROPERTY

The table contains a list of possible types of a property.

| COLUMN NAME | DATATYPE | NOT NULL | COMMENTS |
|---|---|---|---|
| ID_TYPE_PROPERTY | numeric(10) | True | |
| DESC_TYPE_PROPERTY | Varchar(200) | False | The attribute can take the following values:<br>1. string;<br>2. array;<br>3. number;<br>4. other.... |

| PRIMARY KEY NAME | COLUMNS | COMMENTS |
|---|---|---|
| PK_ASS_D_TYPE_PROPERTY | ID_TYPE_PROPERTY | |

ASS_D_REL_NAME

The table contains the types of relation that exist between assets.

| COLUMN NAME | DATATYPE | NOT NULL | COMMENTS |
|---|---|---|---|
| ID_RELATION_NAME | numeric | True | |
| DESC_RELATION_NAME | varchar(50) | False | The attribute can assume, for example, the value "located in". |

| PRIMARY KEY NAME | COLUMNS | COMMENTS |
|---|---|---|
| PK_ASS_D_REL_NAME | ID_RELATION_NAME | |

ASS_D_ROLE
The table contains a list of possible roles between two assets

| COLUMN NAME | DATATYPE | NOT NULL | COMMENTS |
|---|---|---|---|
| ID_ROLE | numeric | True | |
| DESC_ROLE | varchar(200) | False | The attribute can assume the following values (for example):<br>1. target;<br>2. source;<br>3. ..... |

| PRIMARY KEY NAME | COLUMNS | COMMENTS |
|---|---|---|
| PK_ASSET_D_ROLE | ID_ROLE | |

ASS_R_RELATION

The table contains the relations between two assets and the role for every asset in the relation.
For example:
Identification --> progressive number;
Relation name --> located in;
asset1 --> PC;
role1 --> source;
asset2 --> printer;
role2 --> target;
note --> .....

| COLUMN NAME | DATATYPE | NOT NULL | COMMENTS |
|---|---|---|---|
| ID_RELATION | varchar(50) | True | |
| ID_RELATION_NAME | numeric | False | |
| ID_ASSET1 | varchar(50) | False | |
| ID_ROLE1 | numeric | False | |
| ID_ASSET2 | varchar(50) | False | |
| ID_ROLE2 | numeric | False | |
| NOTE | varchar(1000) | False | |

| PRIMARY KEY NAME | COLUMNS | COMMENTS |
|---|---|---|
| PK_ASS_R_RELATION | ID_RELATION | |

# 6. Conclusion

SAFECARE will provide cyber-physical integrated solutions mainly relying on a data exchange layer to share information between physical and cyber security solutions, a database to centralize physical and cyber incidents, and an impact propagation model to better apprehend the aftermaths of the combination of physical and cyber incidents. The impact propagation model will be the cornerstone of the project achievements because it will interconnect both the physical and the cyber world.

The present document provides specifications of the central database in order to produce and design the conceptual model of the central database, which will be used to implement the software application to manage it.