Security of Air Transport Infrastructures of Europe

D4.1 - Specification of data exchanges, interfaces and log semantic

Deliverable Number	D4.1
Author(s)	ISEP, FQS, TLB, IDE, DLR, ASC, SAT, INOV, ITTI
Due/delivered Date	M14/2020-10-15
Reviewed by	DLR, KEM, ACS
Dissemination Level	PU
Version of template	1.07

Start Date of Project: 2019-05-01

Duration: 27 months

Grant agreement: 832969

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 832969

Document contributors

No.	Name	Role (content contributor / reviewer / other)
1	Alda Canito (ISEP)	Content Contributor
2	Hubert Künig (FQS)	Content Contributor
3	Kelly Burke (NIS)	Content Contributor
4	Matteo Mangini (NIS)	Content Contributor
5	François Déchelle (TLB)	Content Contributor
6	Sébastien Clavert (IDE)	Content Contributor
7	Jens Hampe (DLR)	Content Contributor
8	Thomas Oudin (ACS)	Content Contributor
9	Leonidas Perlepes (SAT)	Content Contributor
10	Souzanna Sofou (SAT)	Content Contributor
11	Filipe Apolinário (INOV)	Content Contributor
12	Marcin Przybyszewski (ITTI)	Content Contributor
13	Corinna Köpke (FHG)	Content Contributor
14	David Lancelin (ACS)	Technical Review
15	Vasileios Kazoukas (KEMEA)	Security Review
16	Meilin Schaper (DLR)	Quality Review

Document revisions

Revision	Date	Comment	Author
V0.1	2019-12-09	Table of contents	Alda Canito
V0.2	2020-01-06	Added some message specifications provided by partners	Alda Canito
V0.3	2020-01-12	Added INOV and FHG contribution	Filipe Apolinário Corinna Köpke
V0.4	2020-02-14	Added ITTI and TLB contribution	Marcin Przybyszewski François Déchelle
V0.5	2020-03-17	Added Incident Management Portal description	Thomas Oudin
V0.6	2020-04-17	Added Syslog Messages description	Alda Canito
V0.7	2020-04-21	Syslog review	Alda Canito
V0.8	2020-04-27	Edited the part about RIS	Kelly Burke
V0.9	2020-06-16	Edits	Alda Canito
V0.9	2020-06-25	Final technical check and approval for submission	David Lancelin, Technical Manager
V0.10	2020-06-26	Adapting document to SAB review comments	Alda Canito
V0.11	2020-06-29	Final security check and approval for submission	Vasilis Kazoukas, Project Security Officer
V1.0	2020-06-25	Final quality check and approval for submission	Meilin Schaper, Quality Manager
V1.1	2020-10-01	Changes in chapter 4.	Alda Canito
V1.1	2020-10-14	Final security check and approval for submission	Vasileios Kazoukas, Project Security Officer
V2.0	2020-10-15	Final quality check and approval for submission	Meilin Schaper, Quality Manager

Executive summary

The goal of this deliverable is to describe the process through which an ontology was developed for the SATIE project. The focus of this ontology is to facilitate the interoperability between the SATIE's systems by defining clear semantics for all messages exchanged between the systems, their interfaces and logs.

An exploration of existing ontologies both the field of cyber-security and airport security was conducted. The ontologies better suited for SATIE's needs were selected, combined and extended as necessary. The development process began with an analysis of the expectations of each of the SATIE's tools by assessing their inputs, outputs and responsibilities. From here, an extraction of the most relevant ideas and concepts was undertaken and compared to those of the studied ontologies in order to establish the necessary extensions. This domain extension is guided by the necessities of the different tools and is described in detail. The ontology and its logics can be used by the existing tools to aid with knowledge representation and reasoning processes these may wish to employ.

Table of Contents

List of Figures

List of Tables

List of Acronyms

Acronym	Definition
ALCAD	Application Layer Cyber Attack Detection
AODB	Airport Operational Data Base
ASDL	Aviation Scenario Definition Language
ATM	Air Traffic Management
ATMONTO	Air Traffic Management Ontology
BP-IDS	Business Process-based Intrusion Detection System
CAPEC	Common Attack Pattern Enumeration and Classification
CAS	Crisis Alerting System
CCTV	Close-Circuit Television
CEF	Common Event Format
COTS	Commercial Off-the-Shelf
CVE	Common Vulnerabilities and Exposures
CVE	Common Vulnerability Exposure
DAML+OIL	Defence Advanced Research Projects Agency Agent Markup Language with Ontology Integration Layer
DARPA	Defence Advanced Research Projects Agency
EDXL	Emergency Description Exchange Tool
EMCR	Emergency Message Content Router
EMRC	Emergency Message Content Router
FPL	Flight Plan
GLPI	Gestionnaire Libre de Parc Informatique, or "Open Source IT Equipment Manager" in English
GUI	Graphical User Interface
HMI	Human-Machine Interface
IDS	Intrusion Detection System
IE	Innovation Element
IMDEF	Intrusion Detection Message Exchange Format
INSPIRE	"INcreasing Security and Protection through Infrastructure Resilience" Project
IODEF	Incident Object Description Exchange Format
ISQ NOTAM	“Intelligent Semantic Query of Notices to Airman” Project

JSON	JavaScript Object Notation
MET	Meteorological
NAS	US National Airspace System
NASA	National Aeronautics and Space Administration
NOTAM	Notices to Airman
OASIS CAP	OASIS' Common Alerting Protocol
ODE	Ontology Development Environment
OntoSec	Security Ontology
OntoVul	Vulnerability Ontology
OVM	Ontology for Vulnerability Management
OWL	Web Ontology Language
RDF	Resource Description Framework
REST	Representational State Transfer
RIS	Risk Integrated Service
SAO	Situation Awareness Ontology
SCADA	Supervisory Control and Data Acquisition
SECCO	Security Core Ontology
SIEM	Security Information Event Management
SMS	Short Message Service
SMS-I	Security Management Solutions – Investigation Tool
SOC	Security Operation Center
STIX	Structured Threat Information eXpression
SWIM	System-Wide Information Management
TMI	Air Traffic Management Initiatives
UCO	Unified Cybersecurity Ontology
URL	Uniform Resource Locator
VIP	Vulnerability Intelligence Platform
VuMS	Vulnerability Management Systems (by AIRBUS and Teclib)
WP	Work Package
XML	Extensible Markup Language

1 Introduction

The SATIE project aims to develop a cyber-security toolkit to face cyber-physical threats in a coordinated and effective way, supported by a shared situational awareness system. In order to do so, several systems must communicate and cooperate, exchanging data between themselves in a coordinated way. Which communications are possible within the SATIE system, along with what messages are exchanged and what the contents of these mean must to be established as soon as possible in order to fully achieve SATIE's goals.

As such, this deliverable focuses on describing the work developed under the Task 4.1 "Specification of Data Exchanges, Interfaces and Log Semantic", resulting in an ontology that defines the several cyber-security concepts that can be used to describe the contents of the message exchanged between the different systems of SATIE. To achieve and agree upon this ontology, it was necessary to analyse all incoming and outgoing messages for each of the systems, extracting the concepts and contents mentioned in these and establishing the relationships between them. Existing ontologies in the cyber-security domain were researched, evaluated and measured against the needs of the systems. The remainder of the deliverable is organized as follows:

Chapter 2 describes the state of the art on ontologies for the domains of cyber-security and airports, pointing to existing standards and their applicability to SATIE.

Chapter 3 presents the results of a questionnaire of the expected inputs and outputs of each SATIE system in order to assess the real necessities of each, and further establish which systems effectively exchange messages with one another. The results of this questionnaire serve as a starting point for the extraction of concepts that are shared among the majority of the systems and thus must be agreed upon.

Chapter 4 presents the most important concepts within an ontology to be applied in all communications between SATIE's systems. Additionally, it shows how the ontology is compatible with existing ontologies for cyber-security and how it effectively works as an extension of these.

Finally, Chapter 5 will close the document with a general assessment of the work done and some final remarks.

2 Data conceptualization: Ontologies

Any communication between two or more systems relies on an agreement: what data are being exchanged and what their meaning is. While this agreement can be implicit – and therefore not formally defined – that choice comes with a few hindrances, such as higher maintenance costs, more resistance to change, lack of explainability and making it harder for different systems to join into those communications. Explicit agreements, on the other hand, ease these problems by formalizing the semantics of the data, usually through means of ontologies.

In computer science, ontologies are commonly defined as "explicit specification of a conceptualization" (Struder, Richard, & Fensel, 1998). Here, the conceptualization refers to a rational and abstract model of a given domain, which includes the identification and description of concepts, properties and relationships between these. These must be detailed and consistently described in a way that intelligent agents can understand and reason upon. In (Brost, 1997), this definition is extended with two additional concepts, namely "formal" and "shared": through formalization, the ontology can be read, understood and processed by either humans or machines, and by being shared it means the ontology is accepted as the description of a given domain in consensus by a given group.

The main objective of the Task 4.1, "Specification of Data Exchanges, Interfaces and Log Semantic Specification of Data Exchanges, Interfaces and Log Semantic " is to define an ontology that is agreed upon by all the SATIE's partners and which will describe the contents of all messages exchanged within the SATIE system. In this chapter, existing public ontologies on the domains relevant to the SATIE project – i.e. cyber-physical security and airports – will be explored and evaluated.

2.1 Background

In this modern age, where everything is connected to the internet, there are new threats associated to the new medium of communication. More and more services are provided online, which means more and more possible weak points to be exploited. Just in the first half of 2015 (Data Breach QuickView – 2015 Data Breach Trends, 2015) , more than 200 million records were exposed. A single hacking attack exposed about 78 million of those records. It is worth mentioning that this issue is a domain-crossing one. There is not a sector which uses technology that can be considered safe from such threats. Whether it is business, education, medical, or even governmental, they are all at risk if proper precautions were not taken.

As technology is continuously changing and developing, this makes the infrastructure unstable and vulnerable. However, this does not deny that humans play a role in this as well. Therefore, there is an interaction between human and machine elements which is very important when considering situation awareness in cyber-security of systems.

Regardless of acting agents being humans or computers, any cyber-security system needs to react as soon as possible to any state change within its environment. In order to achieve that, it is necessary to collect and integrate information from different resources and systems. This information is needed to analyse events, make decisions, obtain feedback after applying those decisions, and gain knowledge to be used in future occurrences (Ulicny, Moskal, Abe, & Smith, 2014). The first challenge is that different systems use different representation of their knowledge. Therefore, an ontology that is focused on cyber-security is needed in order to provide a standard way to exchange data between the corresponding systems.

2.2 Cyber-security ontologies

2.2.1 UCO

Unified Cybersecurity Ontology (UCO) is an extension to Intrusion Detection System (IDS), which integrates different schemas from different systems to obtain data and knowledge related to cyber-security. This integration helps with the transition from reactive approach to a more proactive and eventually a predictive approach. UCO provides better understanding of cyber-security by mapping some of the existing ontologies related to this field. In addition to the domain description using the Web Ontology Language (OWL), UCO uses rules to infer new information which could not be captured otherwise by reasoners relying on description logics alone (Syed, Pädia, Finin, Mathews, & Joshi, 2016). This ontology can be considered as a semantic version of Structured Threat Information eXpression (STIX), which is an XML representation for cyber-security vocabulary. In addition to STIX, UCO has been extended with more cyber-security and general world knowledge resources. The main classes available in UCO include Means, Consequences, Attack, Attacker, Attack Pattern, Exploit, Exploit Target, and Indicator, additionally describing Vulnerabilities through Common Vulnerability Exposures (CVE). This ontology's latest version is publicly available on GitHub[1]. Figure 2.1 shows the concepts' diagram of UCO.

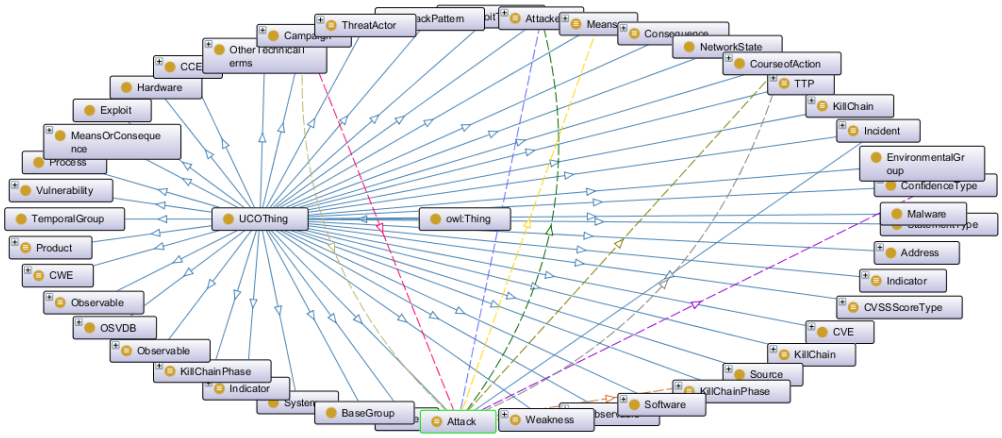

Figure 2.1: UCO ontology graph

2.2.2 SECCO

Security Core Ontology (SECCO) is a small ontology that provides key definitions of security concepts (Oltramari, Cranor, Walls, & McDaniel). Concepts of SECCO are noted to be generally domain independent and based on an intuitive understanding of security. Each concept can be used to represent different things in different domains. Some of the security-related concepts defined in SECCO are Asset, Stakeholder, Security Objective, Threat, Countermeasure, Attack, Attacker, Vulnerability, and Risk. These concepts are interlinked using relations like cause harm to, protects, implements, and place value on. Figure 2.2 shows a sample of the ontology structure.

[1] GitHub - Ebiquity/Unified-Cybersecurity-Ontology: Unified Cybersecurity Ontology." [Online]. Available: https://github.com/Ebiquity/Unified-Cybersecurity-Ontology. [Accessed: 30-Oct-2019]

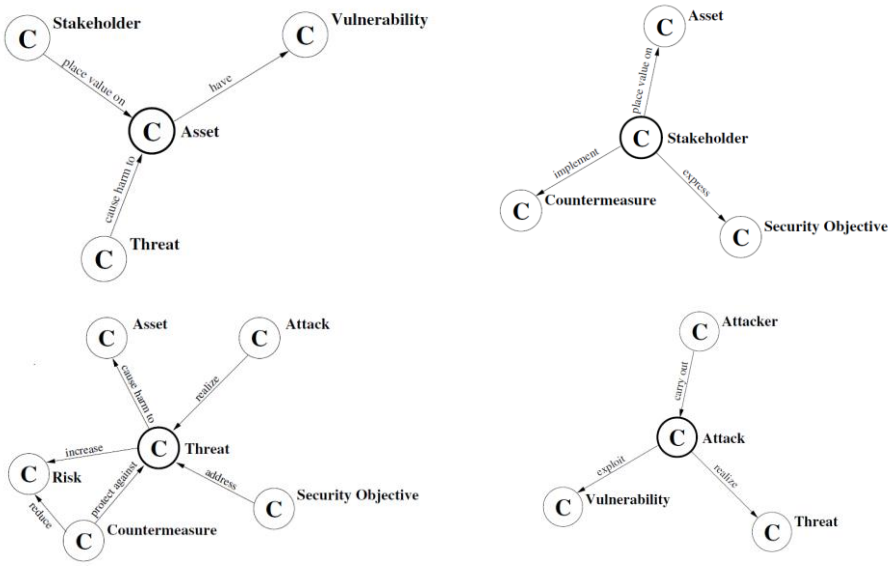

Figure 2.2: SECCO ontology concepts characterization (Oltramari, Cranor, Walls, & McDaniel)

2.2.3 Enhanced Vulnerability Ontology

(Aime & Guasconi, 2010) analysed the limits in several existing vulnerability models and came up with an enhanced vulnerability ontology. They focused on the distinction between vulnerability and threat as they found in their analysis that many frameworks confuse these concepts. The proposed ontology provides an improvement to vulnerability management and risk assessment. Some of the main concepts included in this ontology are Vulnerability, Threat, Impact, Asset, and Control. The core model of the proposed ontology can be seen in Figure 2.3.

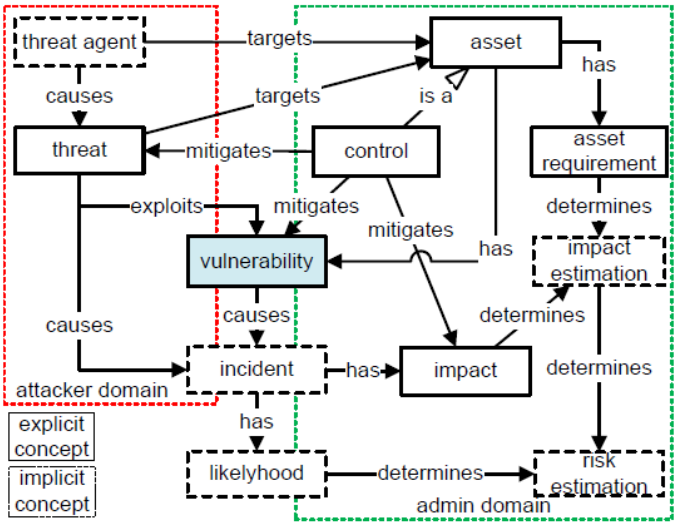

Figure 2.3: Ontology core model (Aime & Guasconi, 2010)

In this ontology, they used Threat to represent a fault that activates a dormant error that is represented by Vulnerability. This activation leads to an Impact of an Incident which is “a concrete error in the intended behaviour of the system” (Aime & Guasconi, 2010).

2.2.4 INSPIRE Security Ontology

As part of the project INcreasing Security and Protection through Infrastructure REsilience (INSPIRE), (Choraś, Kozik, Flizikowski, & Hołubowicz, 2010) worked on an ontology-based decision support engine to be used in protection of critical infrastructure. The goal of the ontology proposed for this project is to provide interdependencies description between vulnerabilities, SCADA assets, safeguards, source of attacks, and risk-categorized threats. Figure 2.4 shows the ontology's main concepts and relationships. The diagram shows that Threats can exploit available Vulnerabilities to expose important Assets. Safeguards work on reducing those Vulnerabilities in order to protect the Assets.

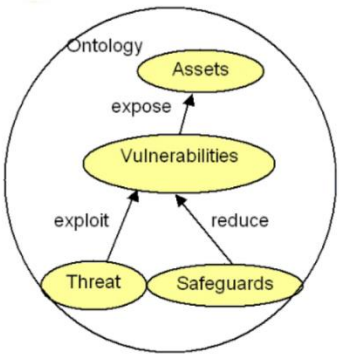

Figure 2.4: INSPIRE security ontology (Choraś, Kozik, Flizikowski, & Hołubowicz, 2010)

2.2.5 Alert ontologies

(Krauß & Thomalla, 2016) recognized the importance of quick detection and efficient reaction to attack. They proposed an ontology to model the security events, attacks, and vulnerabilities by diving the domain into three sub-ontologies: the Alert ontology, the Attack ontology and the Vulnerability ontology. The Alert ontology represents alerts parsed from logs and reports in Intrusion Detection Message Exchange Format (IDMEF) format inspired by (Cuppens-Boulahia, Cuppens, Autrel, & Debar, 2009), while the Attack ontology represents the attacks inferred by the reasoning component using information like attacker and target. The Vulnerability ontology represents vulnerabilities and security gaps' information in compliance with taxonomies and vulnerability databases. Figure 2.5 shows the Alert part of the ontology. An Alert is defined by having Target, Source, Assessment, Classification, Analyzer, Creation Time, and Additional Data.

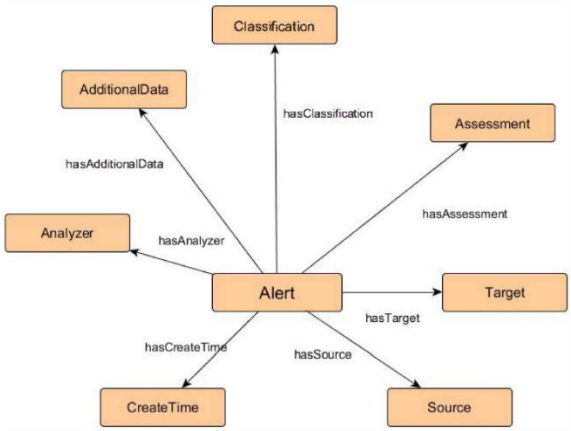

Figure 2.5: Alert ontology (Krauß & Thomalla, 2016)

2.2.6 Ontology for Vulnerability Management

The Ontology for Vulnerability Management (OVM) focuses on software vulnerability by capturing relationships between IT products, vulnerabilities, and other relevant concepts. It is based on multiple vulnerability standards like Common Vulnerabilities and Exposures (CVE) and Common Attack Pattern Enumeration and Classification (CAPEC). This ontology is rich in instances and relationships (Wang & Guo, OVM: An ontology for vulnerability management, 2009). OVM was designed for vulnerability analysis and management and it can accurately describe patterns for external threats and internal vulnerabilities. Some on the key concepts defined in OVM are Vulnerability, IT_Product, Attacker, Attack, Consequence, and Countermeasure. (Wang, Guo, & Camargo, 2010). The conceptual model for OVM can be seen, below, in Figure 2.6.

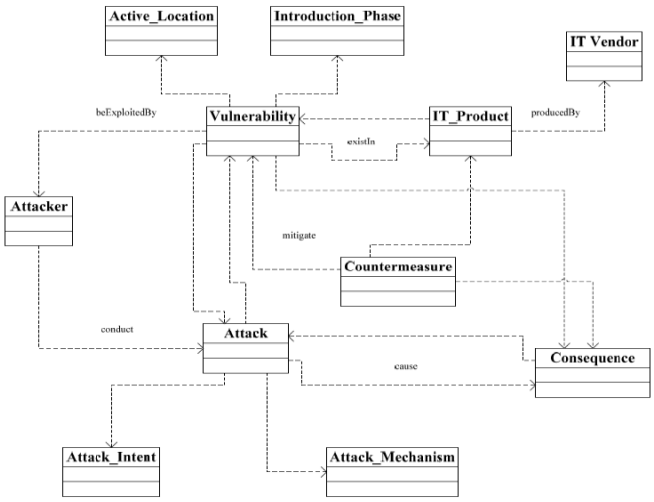

Figure 2.6: OVM conceptual model (Wang & Guo, 2009)

In OVM, an Attacker can conduct an Attack to exploit a Vulnerability in an IT_Product. To protect the IT_Product against any Consequences caused by the Attack, Countermeasures can be used to mitigate the Vulnerability.

2.2.7 OntoSec

The Security Ontology (OntoSec) (Martimiano & Moreira, 2005) is based on security incidents taxonomies and formalized using OWL. OntoSec represents the main security domain concepts into 4 levels. Starting with first/core level that has 13 concepts, each level contains sub-classes of the previous one. Main concepts that are provided by OntoSec include: Agent, Asset, Attack, Tool, Consequence, and Vulnerability. Vulnerability Ontology (OntoVul) represents the concepts and relation about the vulnerability domain. Some of the concepts are imported from OntoVul into OntoSec and they are: Vulnerability, Type, Correction, Range, and Supplier. Figure 2.7 shows the main concepts and relationships defined by the ontology.

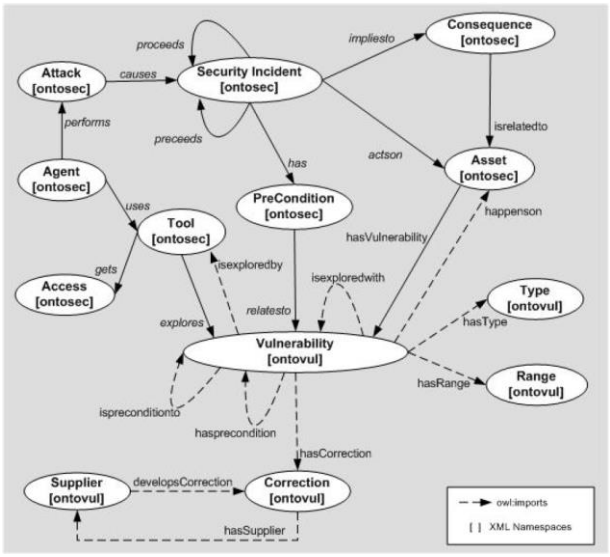

Figure 2.7: Main concept and relation of OntoSec (Martimiano & Moreira, 2005)

2.2.8 Vulnerability Ontology of Metro Operation

A more specific vulnerability analysis has been conducted by (Chen, Peng, Zhong, & Luo, 2016) for metro operation systems. They noticed that vulnerability knowledge was defined by various disciplines and contexts. Therefore, that exist different models describing the available vulnerability knowledge which in turn makes it difficult to reuse it. They applied ontology into the vulnerability analysis to establish a basis for a common knowledge base that enables information sharing. Some of the key concepts of this ontology are Vulnerability, Indicator, Control, Impact and Event. Figure 2.8 shows the conceptual model of the proposed ontology of metro operating system. The figure shows the internal and external types of vulnerabilities. The internal vulnerabilities include defects and flaws in the metro network's topology. While the external vulnerabilities that are forced by nature and humans.

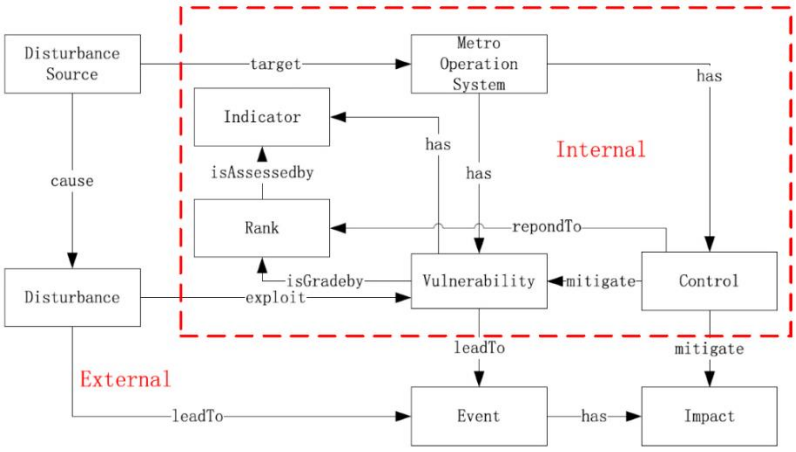

Figure 2.8: Conceptual model of metro operation system's vulnerability ontology (Chen, Peng, Zhong, & Luo, 2016)

2.2.9 Identifying common concepts in the ontologies

Each of the ontologies previously mentioned brings some contribution to the cyber-security domain, but many of them have concepts in common, even if under slightly different nomenclatures. To overcome these differences, the concepts and their descriptions were examined and aggregated. The

graph in Figure 2.9 displays the most popular concepts found in the examined ontologies; here, only concepts that appeared in at least two of the ontologies are displayed for readability.

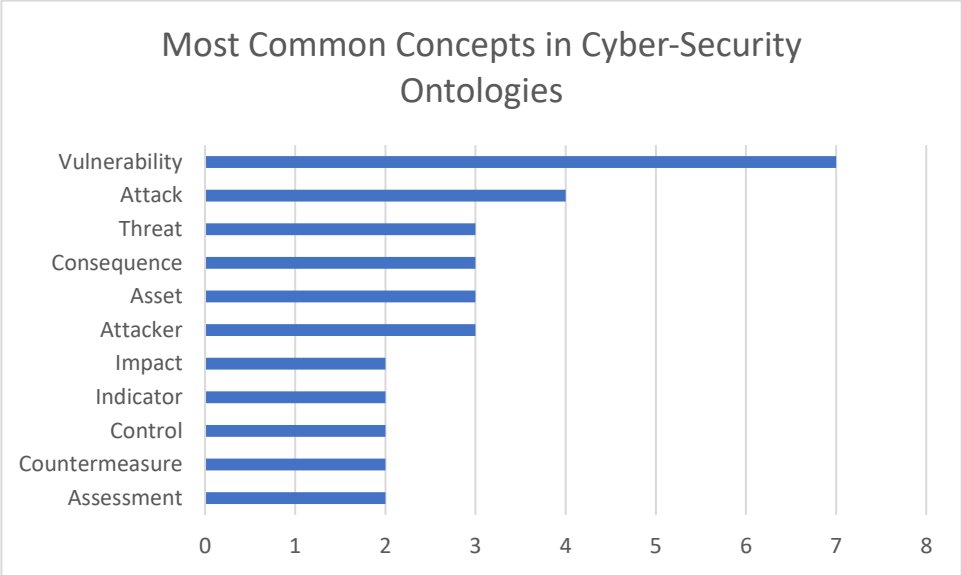

Figure 2.9: Most common concepts in cyber-security ontologies

Vulnerability is the most popular concept, being described in all but one of the ontologies (the Alert ontology). Attack is the second most popular one, although the properties and relationships it allows for vary substantially according to the ontology in question. Next, we find Threat, Consequence, Asset and Attacker. From here follows that any ontology to be chosen for application in the SATIE scenarios, or any one to be developed, should feature these concepts after some fashion and according to necessity. For example, while the concepts of Threat and Attack are very popular, they are not the main focus of any of the tools in SATIE, as will be described in further sections. The choice and application of the concepts will ultimately always rely on the effective needs of the tools in use.

2.3 Airport ontologies

2.3.1 Situation Awareness Ontology

Situation Awareness Ontology (SAO) is a specialized ontology designed using OWL to be the core of a framework to manage and reason about events, situations and actions that simplify situation awareness in airports (Tamea, Cusmai, Palo, Priscoli, & Cimmino, 2014). The main class in this ontology is Event which has two sub-classes: Low-Level Event and High-Level Event. Low-level Events refer to the Events triggered by sensors and can be used by other systems to generate other complicated high-level events. Some of the main relations provided within this ontology are *relatedEvents*, which link Events together, and *relatesWith*, which links Events with other objects like luggage. Another important class is Situation, which represents airport situations during a pre-defined time interval and can be linked events.

2.3.2 ATMONTO

The Air Traffic Management (ATM) Ontology (ATMONTO) is provided by National Aeronautics and Space Administration (NASA). ATMONTO was released in 2018 and it describes classes, properties, and relationships related to air traffic management general domain. The main entities represented by this ontology include flights, aircraft and manufactures, airport and infrastructure, airlines, US

National Airspace System (NAS) facilities, Air Traffic Management Initiatives (TMIs), surface weather conditions and forecasts, airspace components, and departure/arrival routes. NASA provides three interrelated ontologies depending on the level of details that might be required. The ontology is publicly available on the corresponding website (Keller, 2018).

Table 2.1 maps the available features in each layer of ATMONTO, while Figure 2.10 depicts the ontology graph for ATMONO equipment, as described through a Resource Description Framework (RDF) file.

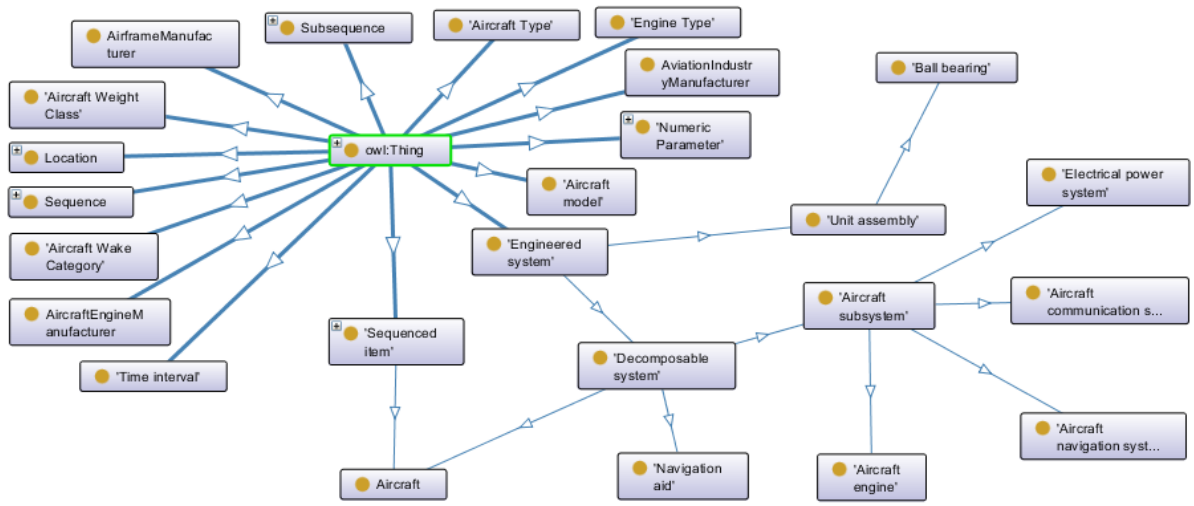

Figure 2.10: Ontograph of the Equipment subdomain of NASA's ATMONTO

Table 2.1: Layers of ATMONTO and their features

Ontology layer	ATMONTO Core	ATMONTO	ATMONTO Plus
Classes definition	Yes	Yes	Yes
Classes instances	No	Yes	Yes
Property definitions	Yes	Yes	Yes
Property values	No	Yes	Yes
Additional instances	No	No	Yes

ATMONTO has been organized into several RDF files that can be imported to any Ontology Development Environment (ODE).

2.3.3 ISQ NOTAMs Project Ontology

The Intelligent Semantic Query of Notices to Airman (ISQ NOTAMs) project included the development of an OWL ontology to be used in NOTAMs content representation. It supports retrieval and reasoning on those NOTAMs including, among others, runways, taxiways, and ground and air communications. This ontology was based on a US/EU commission standard (Bobrow, 2006) that combines Defense Advanced Research Projects Agency (DARPA) Agent Markup Language with Ontology Integration Layer (DAML+OIL). In this ontology, capabilities of high-level NOTAMs are

represented, including aviation specific environment, temporal and spatial knowledge and aviation requirements.

This ontology contains 502 concepts, and the author provided a full list of concepts in the referenced document. Figure 2.11 shows a sample of the listed concepts.

Alphabetical list of OWL Concepts used by the ISQ NOTAMs System

ATSReportingOffice	Airport
ATSRoute	AirspaceEntryPermission
ATSRouteActivation	AirspaceOrganization
ATSRouteClosed	AirspaceOrganizationStatus
ATSRouteStatusReport	AirspaceReservation
ATZStatus	AirspaceReservationActivationReport
ATZStatusReport	AirspaceReservationHead
AUSIdentifier	AirspaceReservationPhrase
Activated	AirspaceReservationStatus
Active	AlongBoundary
AerialDisplay	AltitudeMinimum
AerialDisplayStatusReport	AltitudeReservation
AerialObstacle	AltitudeReservationStatusReport
AerialObstacleStatusReport	Antenna
Aerobatics	AntennaStructureRegistration

Figure 2.11: ISQ NOTAM's ontology concepts sample (Bobrow, 2006)

2.3.4 Aviation Scenario Definition Language

On the other hand, (Jafer, Chhaya, Durak, & Gerlach, 2016) created an ontology as a first step towards developing Aviation Scenario Definition Language (ASDL). This ontology has two different parts, one describes the physical model and flights' operation, while the second one describes important control tower – pilots communications. The main base high-level concepts of this ontology are: Air_Traffic_Control, Aircraft, Airport, and Weather.

Figure 2.12 shows the Aircraft concept's hierarchy for the proposed ontology.

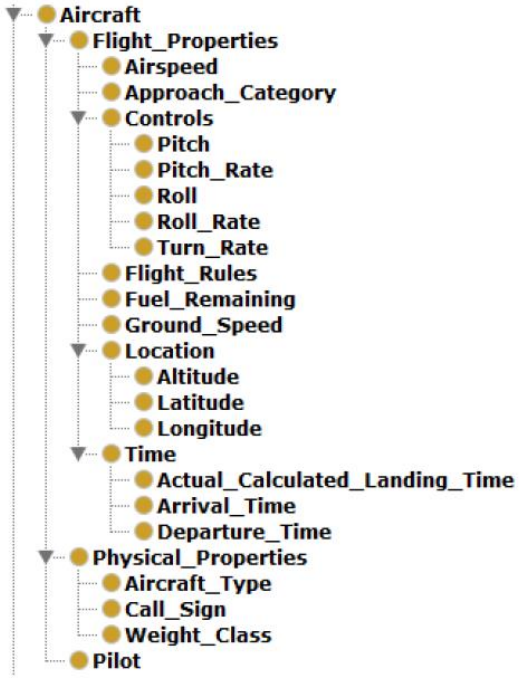

Figure 2.12: Aircraft concept in ASDL ontology (Jafer, Chhaya, Durak, & Gerlach, 2016)

2.3.5 Identifying common concepts in the ontologies

A comparative analysis of the presented ontologies is complicated to perform. These ontologies describe the aircraft domain, albeit under different lenses and for different purposes. While SAO

describes events that can occur in airports, ATMONTO is focused on describing the components and systems that comprise aircrafts. On the other hand, ISQ NOTAM is concerned with locations, routes and communication channels within an airport. Finally, the ASDL aims to describe the actual activities of flight and current positions of aircraft while moving. Comparing these ontologies is therefore a fruitless task, and their applicability to the SATIE's scenarios may be short. However, they may still have some applicability in terms of describing existing assets, particularly ATMONTO, as it can describe not only aircraft but also other airport infrastructure.

3 Data Exchanges

Before moving on to the development of the ontology or the selection of existing ontologies to work with, a general overview of the communications within the SATIE ecosystem was in order. This will allow to understand who communicates with whom and what information they expect to send and receive from other systems. This information can be used as a starting point to establish the most important concepts and how these relate to each other.

This process began by querying the SATIE partners about their system's requirements, inputs and outputs. It is important to note that the messages described below are the result of a first attempt to define the system's communications and responsibilities, which may not necessarily reflect their final structure. It is, however, interesting to present and analyse them as a starting point for the structuring of the SATIE domain through means of an ontology.

The SATIE communications ontology will be made publicly available and its development has been done with Protégé – a popular open-source ontology development tool – and described using the Web Ontology Language (OWL) format.

3.1 SATIE architecture

The definition of possible communications within the SATIE's systems begins with the analysis of the proposed architecture, which can be seen in Figure 3.1.

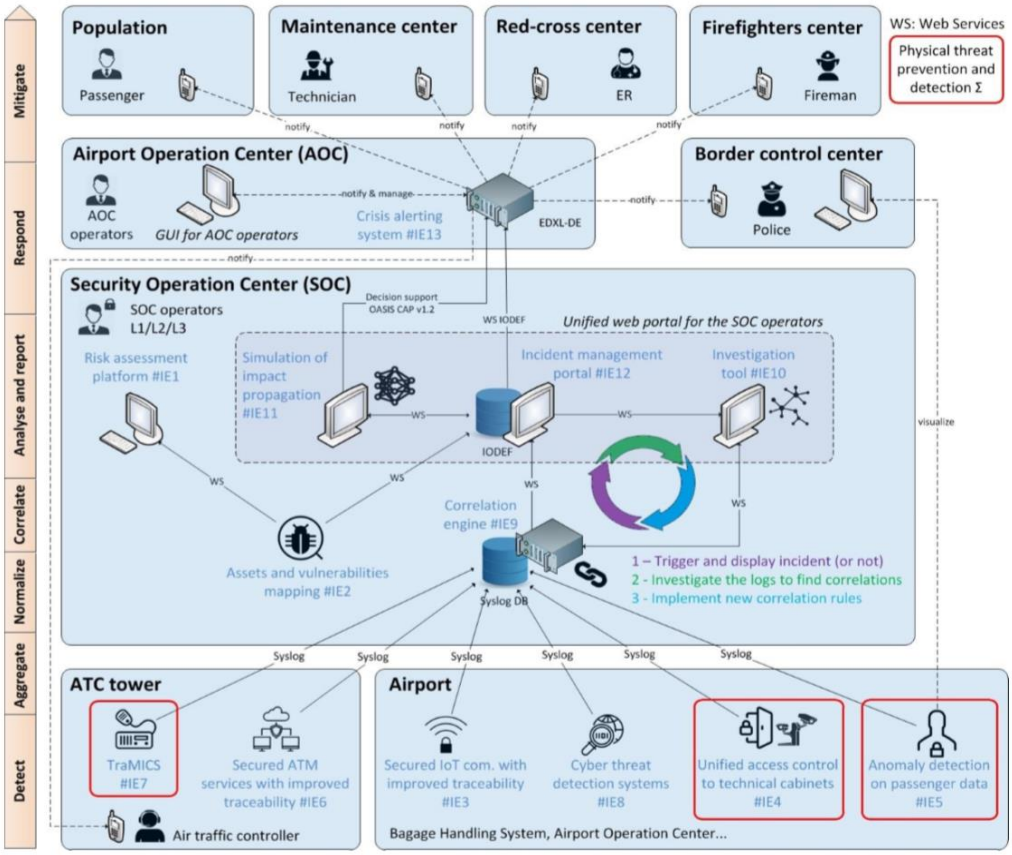

Figure 3.1: SATIE architecture elements and their communications

Detection systems in the lower layers of the architecture gather their data through sensors or existing tools, which, for the purposes of this document, can be categorized as external systems. Because these tools are already in place and using existing communication channels, any communications provided to or from these cannot be modified and therefore are out of the scope of this document. Similarly, on the upper layers of the architecture, outgoing communications (to the passengers, border control, maintenance, firefighter and red-cross centres) must follow already existing protocols and thus also categorized as communications to external systems.

3.2 SATIE systems

The second step was the analysis of the expected inputs and outputs of the remaining systems, which would help further establishing their roles and necessities within the architecture, as well as what they expect to receive and supply to each other. As such, a questionnaire has been sent to all partners for them to describe their tools and each of their expected inputs and outputs. This entailed, for each system, defining the senders and recipients of each message, the different possible messages and a textual description that should be as detailed as possible, including suggestions of existing formats that should be considered. The results of the questionnaire are described in the remainder of this chapter.

3.2.1 TraMICS - Traffic Management Intrusion and Compliance System

TraMICS intends to detect security incidents in Air Traffic Control domain, especially at the controller working position. TraMICS includes four kinds of detectors: un-authorized speakers in the controller-pilot voice communication, stress detection within the voice, non-conformance of aircraft movements and a conflict detection. Each single detector delivers its findings to the TraMICS Indications Correlation Module which calculates a correlated security indicator for each single working position TraMICS is used for. To support e.g. human operators in decision making, also the single detections themselves will be sent by TraMICS. This means, each “Correlation security alert”-message may be supplemented by a “Conflict detection details”-message and/or “Conformance monitoring details”-message and/or “Speaker verification details”-message and/or “Stress detection details”-message. If, after a specific period (duration still to be defined), the alert is still valid, new messages will be sent. The “Alive”-message might be taken to evaluate if TraMICS is up and running.

3.2.2 Secured Air Traffic Management Services

Figure 3.2 provides an overview of the ATC Tower “Secured Air Traffic Management (ATM) Services” component, its context and interfaces. The Secured ATM Services receive their input data from external systems and provide their servicer date to service consumers, such as the AODB or the ATC HMI. During service provision, ATM services provide logging information to the Correlation Engine located in the Security Operations Centre. At the same time, Secured ATM Services accept cyber-security management commands (from Incident Management Portal) to switch between states of operation.

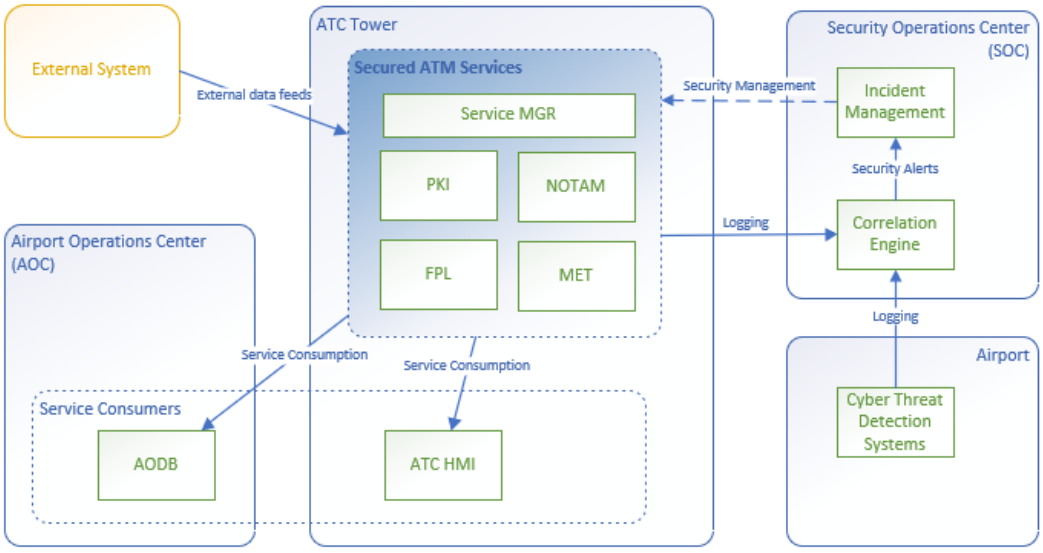

Figure 3.2: Interfaces of Secured ATM Services

3.2.3 Cyber Threat Detection Systems on Business Processes

The Cyber Threat Detection Systems monitor network communications in order to identify potential threats to the system. This module is comprised by four systems, which are explained in detail below. These systems are:

- ComSEC: Secured Communications, which verifies the integrity of exchanges;
- BP-IDS: Business Process-based Intrusion Detection System, which uses sensors to monitor the status of different processes;
- BIA: Business Impact Assessment, which simulates the propagation of threats and affected assets from a business perspective and
- ALCAD: Application Layer Cyber Attack Detection, which monitors Netflow information received from the Secured ATM Services.

3.2.3.1 ComSEC

ComSEC is a sensor network tap sensor that intercepts network communication packets exchanged between the host where ComSEC is installed and the other network devices and validates the integrity of the messages. Whenever integrity validation fails to be verified for a given network packet, ComSEC sends alerts exportation using an Apache Kafka plugin to the Correlation Engine (Table 3.1).

Table 3.1: Inputs and outputs of ComSec

Sender	Recipient	Type	Contents
ComSEC	Correlation Engine	Alert (ComSEC)	ComSEC status alert object. Contains network information profiling the ComSEC sensor.

ComSEC	Correlation Engine	Alert (ComSEC)	ComSEC incident alert object.
			Contains information about an alert raised by ComSEC concerning a network security violation. Sent information includes:
			• Network information profiling the ComSEC sensor
			• Information about the incident detected:
			  ○ List of Devices and the software applications involved in the network security violation.
			  ○ Incident classification:
			  ▪ Type of deviation detected.

3.2.3.2 Business Process-based Intrusion Detection System (BP-IDS)

Business Process-based Intrusion Detection System (BP-IDS) is a process monitoring solution that aims at the detection of incidents on technology enabled infrastructures. It operates by collecting traces from multiple sensors scattered on the monitored infrastructure that indicate execution of activities in business processes. It matches, in real time, the activities detected in the executed business process with the specified business process and specified business rules. Whenever those executed process deviate from the specification, the activity is marked as a possible incident and the infrastructure administrator is notified in real-time by BP-IDS with the causes of that anomaly (traces, affected processes, etc.). Thus, it offers broad protection against (1) cyber-security incidents (such as, intrusions or forgery of equipment behaviour) and (2) operational security incidents (like, equipment and network failure, human error, or natural disasters).

BP-IDS is architected as a distributed system composed by:

- One monitoring core, which is composed by: Configuration Manager responsible for configuring the several sensors to capture the activities; and the Verification Engine, which analyses the captured activities according to the specification and produces alerts whenever incidents occur. Internally, the Verification Engine's also contains an inner component Event Output Engine, whose responsibility is to export the alerts produced to different data formats (such as Syslog, XML or JSON) for better integration with others Security Information Event Management (SIEM) systems;
- Several sensors, which depending on their characteristics can be classified as: network-based sensors, when the extracted traces of activities are collected by inspecting network traffic; or host-based sensors when they are gathered from logs stored in the infrastructure's systems. Sensors used by BP-IDS are typically extended versions of COTS sensors (e.g. Snort as network sensor and Ossec as host sensor) that contain additional to their software, the BP-IDS Sensor Plugin that serves as interface between the monitoring core and the sensor. BP-IDS Sensor Plugin adapts the specification sent by the Configuration Manager into configuration parameters specifically used for sensor configuration, and is also responsible for identifying the activities executed based on process traces captured by the sensor, and send them to monitoring core's Verification Engine component for further analysis;
- Two management applications, which are: Administration Interface that allows the organization's system administrators to setup business process specification; and Monitoring Interface, that allows the administrators to analyse the results obtained from BP-IDS monitoring.

BP-IDS will provide alert exportation capabilities by supporting the Common Event Format (CEF) data format and automatic exportation of alerts using an Apache Kafka plugin. Moreover, to facilitate accessing BP-IDS alerts for other tools in WP4, hyperlinks to the new web-based BP-IDS online visual interface will be included on the alerts.

Table 3.2: Outputs of the Business Process-based Intrusion Detection System

Sender	Recipient	Type	Contents
BP-IDS	Correlation Engine	Alert (BP-IDS)	BP-IDS status alert JSON object. Contains information status of BP-IDS, including: • Network information profiling the BP-IDS (core and sensors) • Information describing the BP-IDS status

BP-IDS	Correlation Engine	Alert (BP-IDS)	BP-IDS incident alert JSON object.
			Contains information about an incident detected by BP-IDS and the deviations on business process detected. Sent information includes:
			• Network information about BP-IDS components involved on the detection of this deviation:
			  ○ sensor that identified the activity traces that caused this deviation.
			  ○ monitoring core that identified the deviation.
			• Information about the incident detected:
			  ○ List of Devices and the software applications involved in the incident.

			○ Incident classification: ▪ Type of deviation detected. ▪ Description created by BP-IDS with incident description

As seen in Table 3.2, every time BP-IDS changes its operative status (first row of the table), or detects incidents (second row of the table), it will send to the Correlation Engine an alert in JSON format.

3.2.3.3 Business Impact Assessment

Business Impact Assessment (BIA) simulates how cyber threats propagate to the organization assets and assesses the impact caused on the organization business-processes and goals. The tool receives as user input for simulation the initial conditions to start the simulation. These initial conditions include:

- Information about airport infrastructure, namely: airport asset lists; network connections between assets, and threats present on assets;
- Information about business processes related to each asset (provided by SATIE Cyber threat detection system on business processes on Task 4.3);
- Threat affecting a given asset to simulate impact propagation.

The methodology first identifies threat propagation based on organization infrastructure and threats, by searching for paths from the initial compromised asset to the organization targets (specifically the mission assets), based on network connectivity and asset threats. Then, the methodology proceeds with impact assessment by analysing the business-processes each compromised asset is involved. Table 3.3, below, describes the inputs and outputs of this tool.

Table 3.3: Inputs and outputs of Business Impact Assessment tool

Sender	Recipient	Type	Contents
BIA	Correlation Engine	Threat for simulation	BIA will receive the asset and threat required to simulate threat propagation and impact assessment.
Correlation Engine	BIA	Business impact assessment	BIA results. Contains the simulated threat propagation path and business processes affected.

3.2.3.4 ALCAD - Cyber threat detection on critical networks and business processes

As part of the innovation element IE8, ITTI will adapt a cyber-attack detection system. Please refer to D1.2 for the Application Layer Cyber Attack Detection (ALCAD) general architecture (SATIE project, 2019). ALCAD modules are connected using Apache Kafka distributed streaming platform. It serves two purposes: reactive, event-driven communication (instead of request-response approach) and real-time event processing. Currently, the majority of algorithms are implemented on Apache Spark systems. This framework provides an engine that processes big data workloads. Currently, ITTI has implemented several modules allowing detection of malware and botnet presence based on the network traffic analysis. ALCAD can be further queried in order to extract relevant patterns and perform visual analytics.[2]

ALCAD is expected to ingest NetFlow data provided associated to Secured ATM Services (provided by FQS) and provide alerts to the Correlation Engine. Moreover, discussions are ongoing on ingesting network data from SEA. Table 3.4, below, presents the overall description of the type and contents of the messages/data exchanged.

Table 3.4: Inputs and Outputs of ALCAD general description

Sender	Recipient	Type	Contents
ALCAD	Correlation Engine	Alerts	IP address, time of generation, time window of the detected abnormal behaviour (will conform to formats to be defined by the Correlation Engine)
Secured ATM Services	ALCAD	Netflow Information	Ingress Information Source IP address Destination IP address IP protocol Source port for UDP or TCP, 0 for other protocols Destination port for UDP or TCP, type and code for ICMP, or 0 for other protocols IP Type of Service

3.2.4 Unified Access Control and Anomaly Detection on Passenger Records

The Unified Access Control and Anomaly Detection On Passenger Records systems are solutions constructed by IDEMIA on its product Augmented Vision that allows real time analysis of video feeds with biometrics detection and identification. The association of biometrics identification with physical token validation for access control allows for extended protection against intrusion. This system will collect information from the token reader and the CCTV feed to elaborate a decision that is send to the control system to grant access or not. It matches token's user biometrics to a list of pre-enrolled user to validate its identity and its ownership. The system also analyses the context of the request, this allows biometrics identification of any individual trying to get access to the restricted zone either to facilitate access and thus avoid to match each token of each individual or to avoid that any non-allowed individual try to profit the "grant access" decision to infiltrate the

[2] The current view in SATIE is that ALCAD will operate as a service. Although it would be possible to include the dedicated ALCAD GUI, it is currently considered to be out of the SATIE project's scope.

restricted zone. In addition, the Unified Access Control will try to detect any coercion made onto the token's user to avoid unwanted access authorization.

The Unified Access Control system sends an audit of each access request as an info or an alert depending of the threat evaluation to the Correlation Engine. Content of the log are available in Table 3.5.

Table 3.5: Inputs and Outputs of the Unified Access Control system

Sender	Recipient	Type	Contents
Unified Access Control	Correlation Engine	Info	Info audit send when an access is granted / denied to a zone to a known person (in case of denied person does not have rights to access the zone but is known by the system) Syslog message containing: - type - action timestamp - origin (station) - identifier of known person - Decision
Unified Access Control	Correlation Engine	Alerts	Info audit send when an access is required by an unknown person (badge ok, biometrics not) Syslog message containing: - type - action timestamp - origin (station) - identifier of badge - decision - confidence score - additional data (if required, for the moment none)
Unified Access Control	Correlation Engine	Alerts	Alert audit log sent when an unauthorized access is detected into a zone (shadowing) Syslog message containing: - type - action timestamp - origin (station) - confidence score - additional data (if required, for the moment none)

IDEMIA's solution to detect anomaly in passenger data ("Anomaly Detection On Passenger Records") uses travel document information of a passenger, collected during its check-in, to match it against a list of people of interest or against business rules to evaluate its threat level. In addition, the system takes a capture of a passenger baggage to extend its identity. This allows threat sharing between a passenger and its baggage or recovery of baggage owner using simple picture taking if the tag of the baggage is damaged or destroyed. The Anomaly Detection On Passenger Records system will send log to the Correlation Engine each time a control is made onto passenger information either as an alert or info depending of the threat level. In addition, a log is sent for each enrolment of baggage. Table 3.6 lists the messages.

Table 3.6: Inputs and Outputs of the Anomaly Detection On Passenger Records

Sender	Recipient	Type	Contents
Anomaly Detection On Passenger Records	Correlation Engine	Info	info audit log sent when an anomaly detection is performed on a passenger through its traveller document info Syslog message containing: - type - action timestamp - origin (station) - identifier of passenger - control result summary - Decision
Anomaly Detection On Passenger Records	Correlation Engine	Alerts	Alert audit log sent when an anomaly is detected (replacing info message) Syslog message containing: - type - action timestamp - origin (station) - identifier of passenger - control result summary - decision - Action to follow - alert identifier
Anomaly Detection On Passenger Records	Correlation Engine	Info	Info audit log when an enrolment of a baggage is performed Syslog message containing: - type - timestamp - origin (station) - baggage identifier - passenger id
Anomaly Detection On Passenger Records	Correlation Engine	Info	Info audit log when a matching for unidentified baggage is performed. Passenger identifier list may be null if no match Syslog message containing: - type - timestamp - origin (station) - decision - list of passenger id (may be null)
Anomaly Detection On Passenger Records	Correlation Engine	Info	Info audit log when an agent accesses an identified anomaly onto passenger data Syslog message containing: - type - timestamp - origin (station) - alert identifier - agent identifier

3.2.5 Correlation Engine for cyber-physical threat detection

The Correlation Engine is a rule-based system who will correlate incoming Syslog messages from different SATIE systems, namely:

- TraMICS;
- Anomaly Detection On Passenger Records;
- Unified Access Control;
- Secured ATM Services;
- cyber threat detection systems (i.e., Malware Analyser, ComSEC, BP-IDS and ALCAD).

Furthermore, it will query the VuMS (Vulnerability Management System by Airbus and Teclib) for additional information on any event it receives from other tools, enriching the information and allowing for more interesting correlations. The engine will correlate events through the application of a set of rules; should a rule be triggered, an alert is sent to the Incident Management Portal.

3.2.6 Vulnerability Management System

The Vulnerability Management System by Airbus and Teclib (VuMS) is composed of two subsystems, namely the GLPI (Gestionnaire Libre de Parc Informatique, or "Open Source IT Equipment Manager" in English) and the Vulnerability Intelligence Platform. The GLPI deals mostly with inventory, making requests about their vulnerabilities and criticalities to the Risk Integrated Service (RIS), information which is then used to enrich the events it receives from the Correlation Engine. The Vulnerability Intelligence Platform, on the other hand, infers and exposes new vulnerabilities, which are fed to the RIS and to the Incident Management Portal.

The VuMS is based on GLPI (Gestion Libre de Parc Informatique), Teclib's open source solution for IT Service Management. Vulnerability management is based on GLPI inventory functionalities, implemented by both the GLPI Inventory Agent and the GLPI Inventory Plugin. VuMS communicates as a message recipient with vulnerability information sources, which can be either:

- The Vulnerability Intelligence Platform (VIP) by CCS;
- Any external system which provides vulnerability information under the CVE format.

The CVE (Common Vulnerability Exposure) format is a (not completely specified) data format providing for an identified vulnerability the information that is needed to determine whether an asset is affected by this vulnerability or not. In particular, a CVE vulnerability entry contains for affected software a description of affected versions.

The main task of the VuMS is to perform for each asset of the inventory a match between the softwares installed on this asset and the known vulnerabilities. If a match is found, it means the asset is vulnerable and an alert must therefore be raised.

VuMS communicates as a message sender with the cyber-physical Incident Management Portal which receives the vulnerability alerts. The alert message will follow the SATIE ontology and may contain for specific information an IODEF (Incident Object Description Exchange Format) object.

3.2.7 RIS (Risk Integrated Service) - Risk Assessment Platform with Cyber-Physical Threat Analysis

The RIS (Risk Integrated Service) solution provided by NIS allows the carrying out of assessments of the risks that insist on assets and operations within the analysis perimeter. The RIS interface allows one to receive input data provided by a human user but can also be integrated with third-party software capable of providing data to be used in the risk assessment process. In particular, two types of data that can be received relate to the assets making up the scope of the assessment and the vulnerabilities associated with these assets.

Within the SATIE platform the module capable of producing this information is the VuMS for ICS and OT systems (GLPI). RIS will therefore have to integrate with GLPI by receiving input information on the inventory of the detected assets and the vulnerabilities associated with them. The technology used for the integration will be REST web services with payloads containing information on assets and vulnerabilities.

An important note to highlight is that the input data provided by GLPI and the data usually managed by RIS work at a different level of detail, which is why a mapping between received inputs and data to be managed in the risk assessment process will have to be applied, both at the asset level and at the vulnerability level.

The outputs produced by RIS can be visualised by SOC operators to obtain information on the presence of risks within the perimeter of analysis, both at the individual asset, operation or specific threat level.

3.2.8 Impact Propagation Simulation for anticipated impact assessment

The Impact Propagation Simulation is a hybrid simulation tool that combines a network modelling approach with a flow model and an agent-based model. The resulting tool requires various inputs which are described in the following.

The Impact Propagation Simulation receives offline information from the airports such as network information for relevant airport systems, dependencies between assets in case of an incident and corresponding recovery procedures and related properties of assets. Further, to evaluate resilient behaviour of assets and systems respective performance functions need to be defined.

The online information that is needed for the Impact Propagation Simulation is incident and alert information consisting of affected assets, time stamp and additional information that still needs to be specified. This information will be provided by the Incident Management Portal.

The Impact Propagation Simulation produces time series data of performance and information on failed, degraded or recovered assets as a function of time. Further, the developed tool analysis the systems' resilience considering uncertainties and various mitigation strategies (such as e.g. the order of repair of broken components in the case of an incident). This output will be provided to the Incident Management Portal and the CAS (Crisis Alerting System).

3.2.9 Cyber-physical Incident Management Portal

The Incident Management Portal helps operator to analyse, respond and remediate to the incident.

Alerts are sent by the Correlation Engine to the Incident Management Portal. When an alert is received by the Incident Management Portal, its related events can be automatically or manually be classified as incident. When this takes place, the incident information, along with the originating events, are sent to the CAS.

Table 3.7: Inputs and Outputs of the Incident Management Portal

Sender	Recipient	Type	Contents
Correlation Engine	Incident Management Portal	Alerts	Syslog message sent from Correlation Engine to the Incident Management Portal when a rule is triggered, content IP address, port, timestamp, etc..
Incident Management Portal	Crisis Alerting System	Incident	Message sent from the Incident Management Portal to the CAS when an operator changes an alert to an incident.

3.2.10 Investigation Tool SMS-I

The Security Management Solutions - Investigation Tool (SMS-I) serves as a unifier of physical and cyber security investigations, supporting fast recovery in case of incidents. To do so the SMS-I collects Syslog data from the Correlation Engine, analysing contextual and semantic data, to identify possible causes for security events and threats, as illustrated in Figure 3.3, below. The Syslog data should include alerts and incidents from different systems that occur, its timestamp, severity and the correlations between them. Also, the assets impacted should be known. With all this information the Investigation Tool will find correlations that will help to find evidence of the causes of an attack.

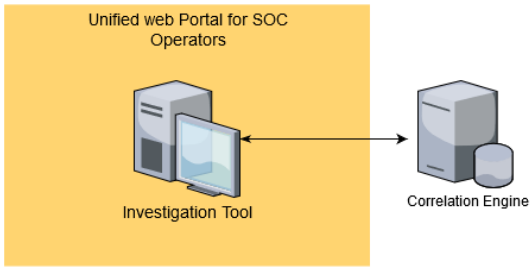

Figure 3.3: Communication with the SMS-I

The results of these analyses will be included in the Correlation Engine ruleset, if new rules (or improvements to existing ones) are discovered. The rules will allow us to protect us from a new set of attacks. A rule states that if a set of incidents occur and there exists a certain relationship between them, then an alert should be raised. A very simple example can be if there are five failed login attempts for the same IP an alert should be raised w.r.t that IP.

Table 3.8: Inputs and Outputs of the SMS-I

Sender	Recipient	Type	Contents
SMS-I	Correlation Engine	Rules	SMS-I will generate correlation rules that will be integrated in the Correlation Engine. The rules will define what should be done if a given set of actions occur, e.g., if there are five failed login attempts for the same IP an alert should be raised w.r.t that IP.

Sender	Recipient	Type	Contents
Correlation Engine	SMS-I	Security Logging	The Investigation Tool will use the information stored in the Correlation Engine to find evidence of the causes of an attack. For this, the Investigation Tool needs to know the incidents from different systems that occur, its timestamp, severity and the correlations between them. Also, the assets impacted should be known.

Events and threats will be made available through an intelligent dashboard, supporting the SOC in their analysis of activities and threats in real-time and allowing for a dynamic definition/customization of correlation rules.

3.2.11 Crisis Alerting System (CAS) for coordinated security and safety responses

The Crisis Alerting System (CAS) performs two tasks. The first task is to create common operational picture by combining information received by the Incident Management Portal and the Impact Propagation Simulation. The communication channel among the two systems belonging to the Incident Management Portal and the CAS, is implemented through a REST Web Service.

The second task performed by CAS is the notification and alerting of the airport stakeholders and passengers, as well as population adjacent to the airport facilities. The airport stakeholder's notification mechanism is implemented through a CAS component, the EMCR (Emergency Message Content Router), which is a smart message routing service by using OASIS CAP standard and EDXL family messages (i.e. EDXL SitRep). On the other hand, passengers or nearby citizens will receive notification messages through cell broadcasting, in SMS format.

Table 3.9: Inputs and Outputs of CAS

Sender	Recipient	Type	Contents
Impact Propagation Simulation	CAS	Resilience assessment	Time-series data of performance for each system during the scenario.
Impact PropagationSimulation	CAS	Mitigation strategies	Identification of critical components to manage the recovery.
Incident Management Portal	CAS	Incident	Message sent from the Incident Management Portal to the CAS when an operator changes an alert to an incident.

CAS	Law Enforcement/Border Control Center Fire Service Control Center Emergency Medical Control Center[3]	OASIS CAP message EDXL SitRep message	The communication among the Airport Operations Center and the related public and safety agencies will be supported. Information about the current situation will be exchanged. This communication will be based on emergency interoperability standards.
CAS	Citizens (Cell Broadcast Center)[3]	SMS Broadcast message	Passengers or nearby citizens will receive notification messages through cell broadcasting, in SMS format.

3.3 Concept analysis

For an easier understanding of the similarities between the contents of the messages, the results of the questionnaire have been condensed in Table 3.10, below. Messages to and from external systems are not considered here. This view offers us an idea of what concepts can be used to describe the messages in a more generic way. One immediate conclusion that can be taken from these results is that most systems will communicate in the form of alerts (info can be considered a type of low severity alert).

Table 3.10: Questionnaire results' summary

Systems	Message
TraMICS	Security Logging (Info and Alert)
Secured ATM Services	Security Logging (Info and Alert) Threat Level
ComSEC	Alerts
BP-IDS	Alerts
Business Impact Assessment	Impact Assessment Assets
ALCAD	Alerts Netflow Information (existing protocol)
Unified Access Control and Anomaly Detection On Passenger Records	Info Alert
Correlation Engine	Info
VuMS	CVE

[3] external systems

Systems	Message	
	IODEF (both existing protocols)	
GLPI	Assets: criticality Vulnerabilities	
RIS	Assets Asset: criticality	
Impact Propagation Simulation	Threat Assessment Assets	Strategies Performance
Incident Management Portal	Alerts Incidents	
SMS-I	Rules Security Logging (Info and Alert)	
Crisis Alerting System	OASIS CAP & EDXL suite standards (existing protocols)	

It is interesting to note that some existing protocols have been proposed by members of the consortium. While OASIS CAP & EDXL Suite of standards is used only by the Crisis Alerting System to communicate with external systems, the Vulnerability Management System expects to be able to send messages in some format that is compatible with both IODEF and CVE, which would require defining concepts such as Incident, Impact, Assessment and Vulnerability, among others. At least three tools will need some sort of conceptualization of Assets, which should include their criticality.

The Impact Propagation Simulation should be able to, upon receiving a reference to a threat – but not necessarily the description of one –, assess its impact on existing assets, and their expected performance loss while the threat is active; additionally, it should supply a number of mitigation strategies and the expected performance of the same assets should those be implemented.

While several of the concepts mentioned in these descriptions are present in several of the ontologies presented in chapter 2, none of the options described or combined these in a way that made them directly useable in the SATIE context. Furthermore, of the described ontologies, only OntoSec, UCO and ATMONTO are publicly available. Of these, UCO describes the domain in a richer way, especially considering it maps all the concepts of the IODEF protocol. A possible combination of UCO and ATMONTO can therefore be considered and possibly extended in order to properly describe all the system's needs. This process is described with more detail in the next chapter.

4 Proposed Ontology

An analysis of the proposals presented in chapter 3 shows that there is indeed a need for harmonization: a lot of systems need to communicate with each other, but have different expectations of how the communication will happen; namely, the inputs of one don't match the outputs of the other. Furthermore, some propose existing formats, but their description shows these may either not be sufficient or may be too complex for their needs. As a starting point, we elicited several concepts that appear several times on this section and from different systems. These will work as the foundation for the development of the ontology and for the further consolidation of the communications that would take place. That being said, the following concepts were considered: Asset, Alert (possibly of different levels), Events, Vulnerabilities and Incidents.

As a start for the extension process, we will begin with some main concepts that are essential to this domain. These concepts are: Alert, Asset, Event, Vulnerability and Incident. Different interpretations for these concepts can be found in different systems and documents. A short overview of these can be found in the Annex: Common Cybersecurity Definitions. The consensus definition is provided in Table 4.1.

Table 4.1: Definition consensus for the SATIE project

Concept	Definition
Alert	A notification that a specific event has been directed at an organization's systems. These can be either Infos, Warnings, Advisories or Alarms depending on the criticality of the Assets involved.
Asset	Information or resource which has value to an organization or person.
Event	A discrete change of stats of an Asset or group of Assets. Some of these changes can trigger Alerts.
Incident	An Event (or group of Events) that compromises an Asset. An Incident may be retroactively classified as an attack. Additionally, it has some sort of impact within the organization, which is described by its severity and completion level.

From the consensus presented in Table 4.1, we can extract the following conclusions:

- The concept of Incident is identical to UCO's Incident concept (*ucoIncident*), also equivalent to IODEF's Incident description;
- UCO's Incident provides a format that additionally allows for the description of both Assessment and Impact;
- ATMONTO provides different systems definition through the Engineering System concept, which allows for the representation of several sub-systems related to avionics (e.g. Navigation and Electrical Power Systems). These can be used, to an extent, to describe existing physical Assets in the airport, but are not sufficient;
- Descriptions of Events and Alerts need to be added to the ontology to reflect the consensus definition.

Timestamps are to be frequently exchanged in these messages, although there does not seem to be any expectations regarding automatic inference about time, such as establishing the order of the

messages. Given that only instantaneous time seems to be required, the generic TimeOntology (Hobbs & Little, 2020) can be used to provide representation for beginning and end time instants.

The relationships between these can thus be visualized in Figure 4.1.

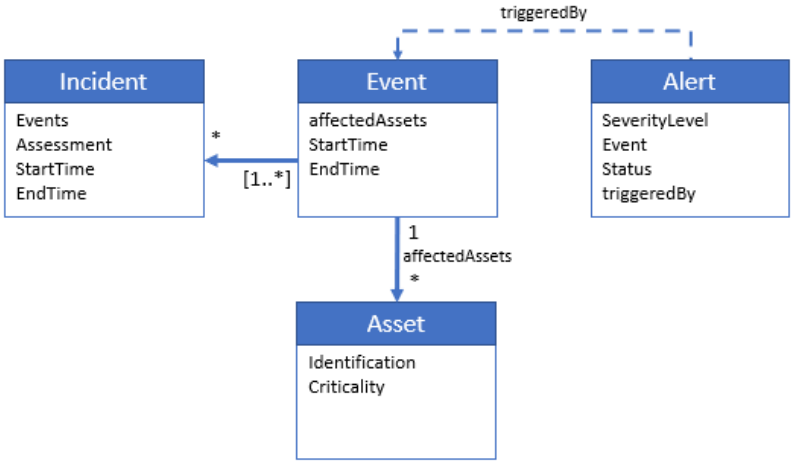

Figure 4.1: Initial concept set and proposed properties

An Event can affect (or change) one or more Assets and trigger one or more Alerts. The SeverityLevel of these should be related to the Criticality of the Assets involved; how that relationship is defined can be specified by each individual tool issuing the Alerts, or it can be inferred through the affected Assets' Criticality. Here we may introduce some sub-classes that comply with specific practical conditions, i.e. different types of Alerts: a consensus between all involved partners establishes these types as Info, Advisory, Warning and Alarm. These are used differently depending on the system in question, namely:

- The cyber threat detection systems, along with the Secured ATM Services, TraMICS, Unified Access Control and Anomaly Detection On Passenger Records report different types of Events and may raise different levels of Alerts.
- The Correlation Engine receives Events and Alerts from other systems and, similarly, outputs Events and Alerts to both the Incident Management Portal and the Investigation Tool. Additionally, it queries the VuMS for additional information about the Events it received on the topics of Assets (Inventory) and Vulnerability.
- The VuMS and its systems query the Risk Integrated Service for information regarding Assets. Additionally, it may expose new Vulnerabilities to the Risk Integrated Service and supply information regarding known Vulnerabilities to the Incident Management Portal when prompted.
- The Incident Management Portal is the only system that generates Incidents. A human operator on the Security Operations Centre (SOC) is charged with the analysis of incoming Events and will validate whether these should be considered Incidents. The information regarding the Incidents is then forwarded to both the Impact Propagation Simulation and to the Crisis Alerting System. A command with the threat level is similarly issued to the Secured ATM Services.
- Different tools supply visualization data to the Incident Management Portal via HTML links. As this information will not be processed by the system and is in visual form only, it does not require structuring and analysis and is therefore beyond the scope of this document.

Knowing these requirements, finally all possible communications between the systems can be defined, which resulted in the diagram shown in Figure 4.2.

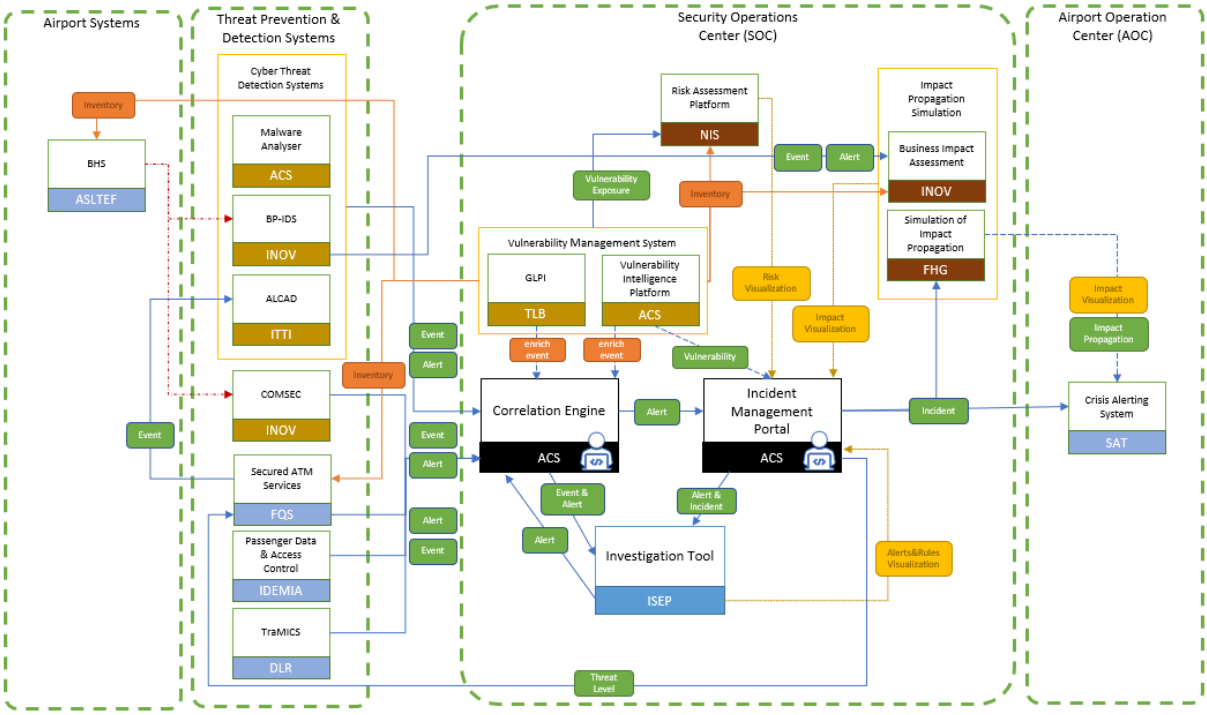

Figure 4.2: Communications between the SATIE's systems

Given these considerations, the following diagram (Figure 4.3) represents the relationship between these ontologies and the extensions under consideration.

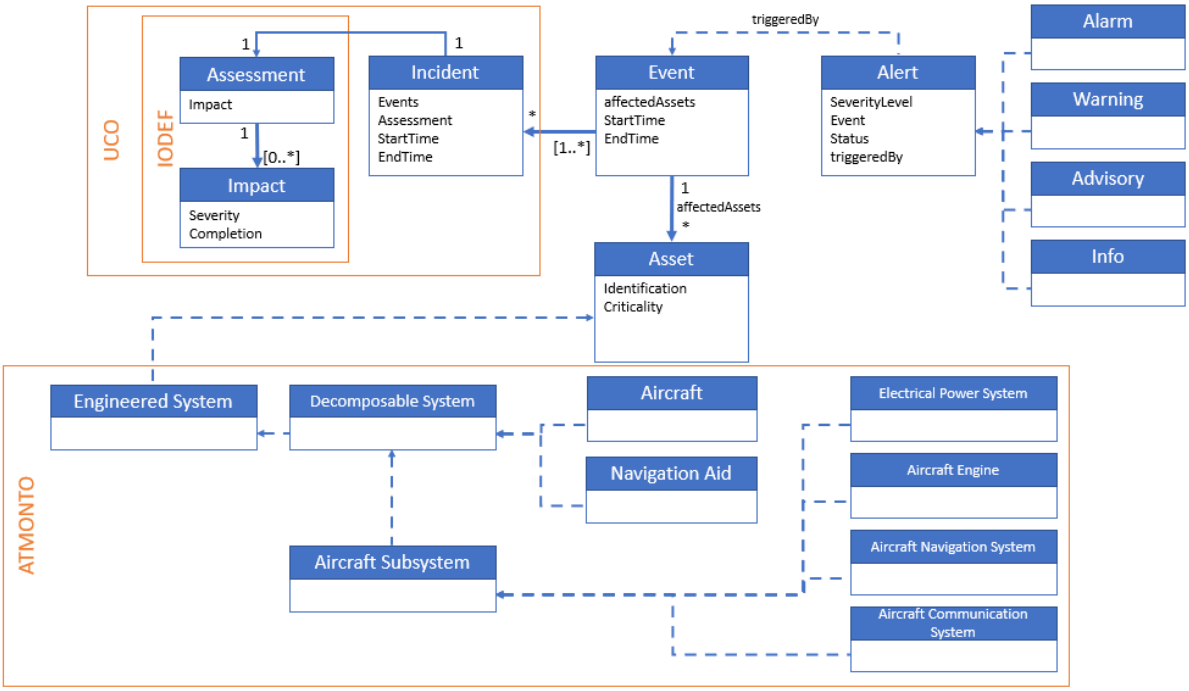

Figure 4.3: Concepts representation for the extended ontology

4.1 Domain representation

In order to further specify the contents of the messages to be exchanged, it was necessary to get a more detailed view of each system's needs. This is particularly important in the cases of GLPI, VuMS and RIS, since these systems enrich the contents of existing messages on demand and need to make their Events and Asset descriptions as specific as possible.

4.1.1 RIS specification: Asset hierarchy

RIS contains a database of cyber and physical Assets, which are distributed in different categories and subcategories. Through the analysis of the outcomes of Task 2.4 - "Definition of an impact propagation and decision support model" (SATIE project, 2019), a number of Asset hierarchies were established and added to the ontology, which will be described next. It is important to note that this document does not specify any properties or relationships between the Assets except for hierarchical relationships (*is-A* or *SubClassOf*).

Figure 4.4, Figure 4.5 and Figure 4.6 describe the subdomain of Physical Assets. The main subcategories to note here are Equipment, Location, Hardware, Transport System and Software. Equipment and Hardware describes mainly the physical equipment within the system, with Software representing its cyber counterparts. Transport Systems represented in Figure 4.5 will be related whenever possible through a *sameAs* relationship with the Engineered Systems described through ATMONTO's ontology.

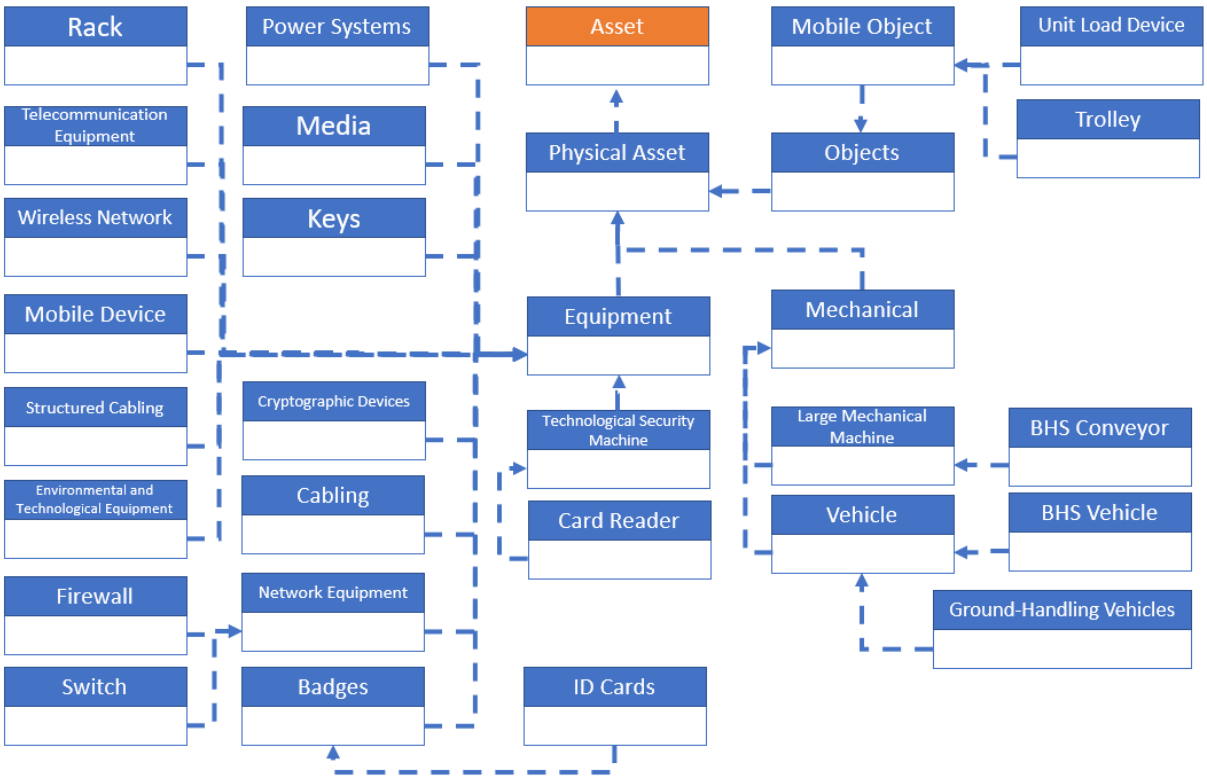

Figure 4.4: Physical Assets 1 – Equipment

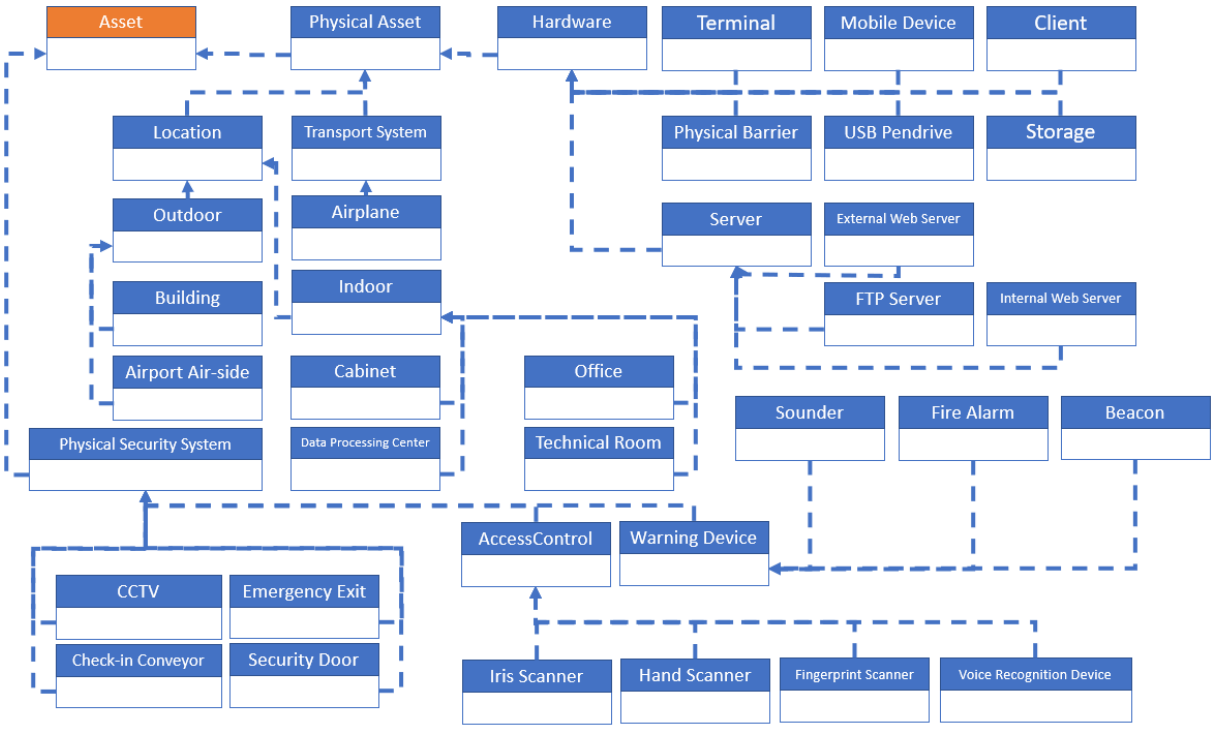

Figure 4.5: Physical Assets 2 – Location, Transport Systems and Hardware

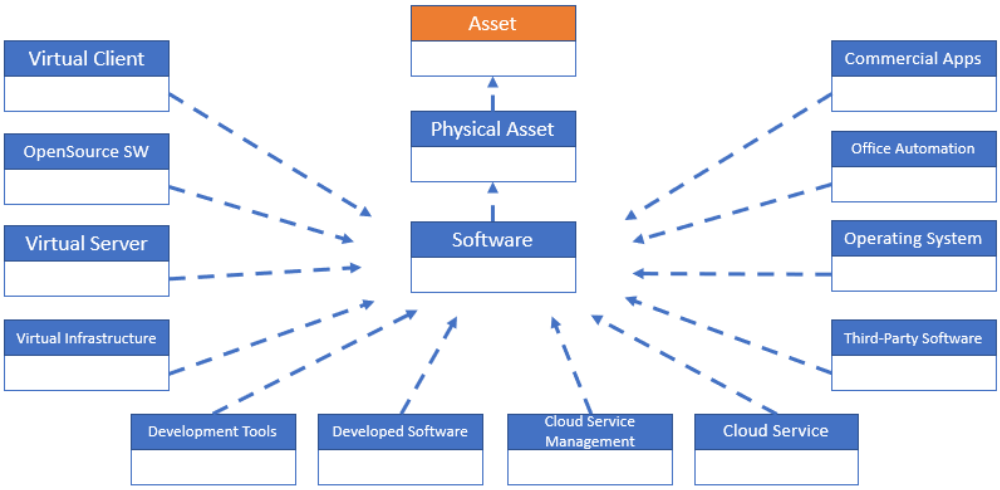

Figure 4.6: Physical Assets 3 – Software

Figure 4.7 describes the hierarchies within the subdomain of Logical Assets, which include Databases, Sensors and the logical part of Networks (contrast with its physical counterpart: Network Equipment is described in Figure 4.4). Additionally, different types of sensors can be considered.

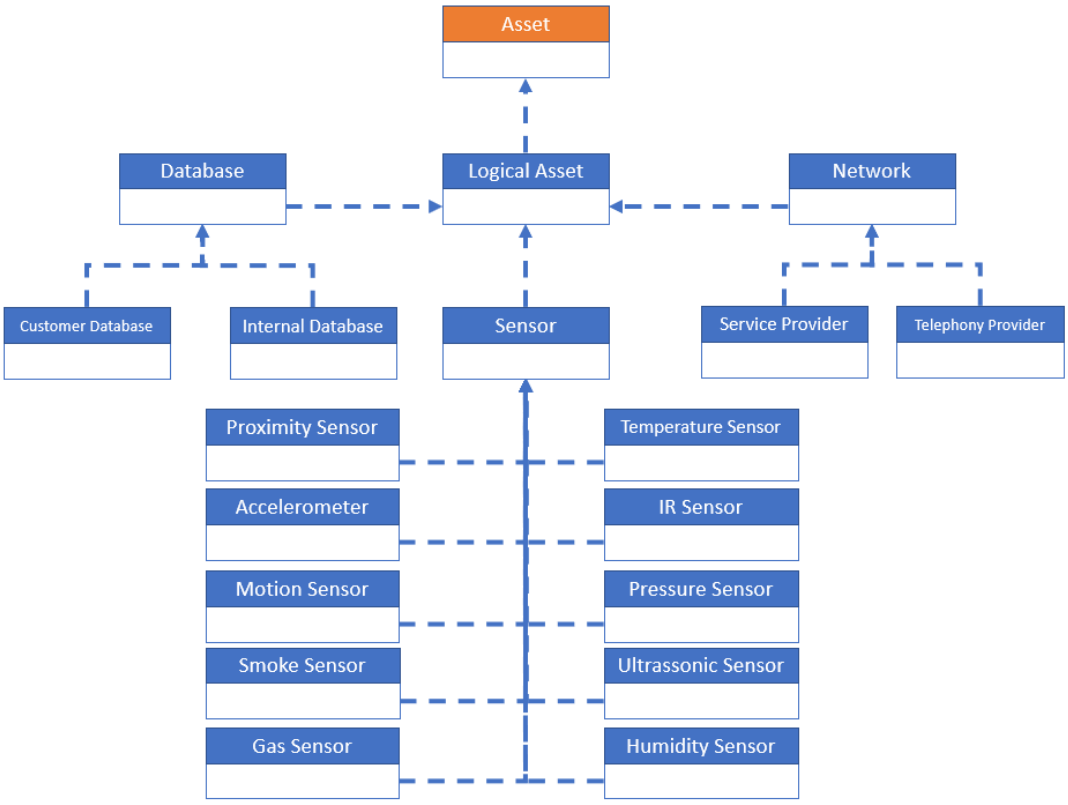

Figure 4.7: Logical Assets

Finally, Figure 4.8 describes the domains of human resources and existing data and or documents.

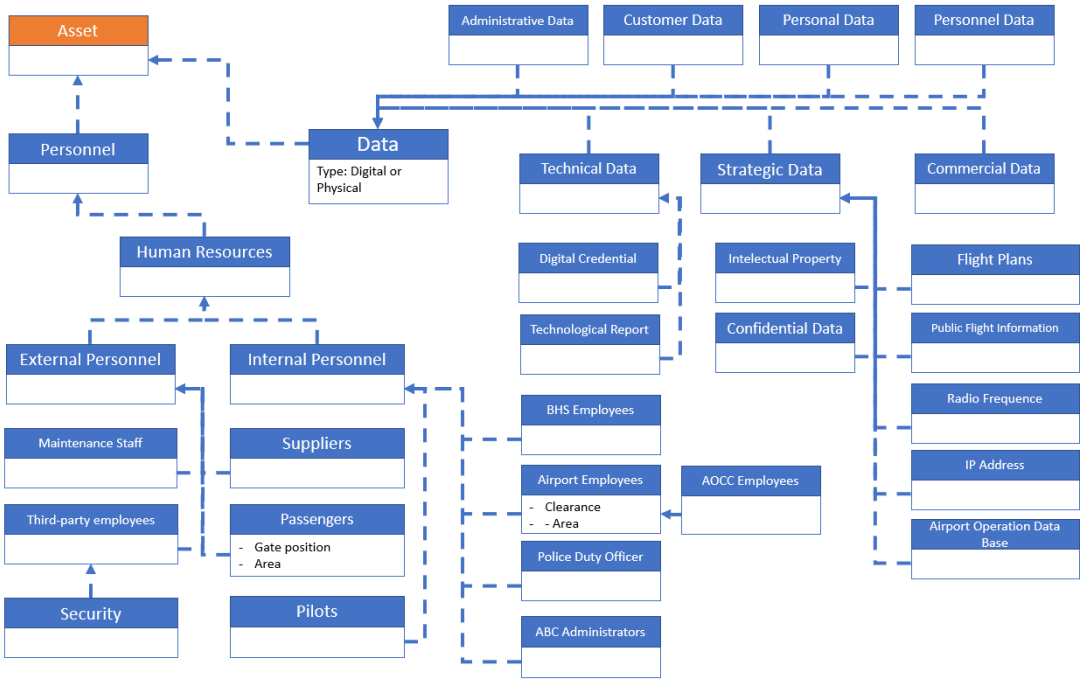

Figure 4.8: Data and Personnel

Here, it is interesting to note that subclasses of Data are not disjoint with one another and a document or file can belong to more than one subcategory simultaneously. Furthermore, data can refer to either digital or physical versions of existing documents. Passenger's status can change according to their location, particularly in regard to their relative position to the gates and whether they are in a restricted area or not.

4.1.2 GLPI: Asset hierarchy

Much like the RIS system, so does GLPI have its internal database describing Assets. These descriptions do include some properties shared by all Assets, namely: (1) ID, (2) Entities_ID (the entity the Asset belongs to) and (3) Name (short, descriptive name of an individual Asset). Assets are discriminated into three main categories: Computer, Software and SoftwareVersion. These can be added to the hierarchies established above as follows (Figure 4.9):

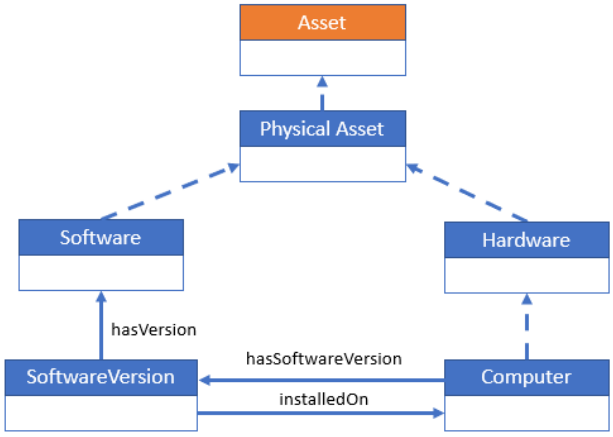

Figure 4.9: Physical Asset addictions per GLPI database contents

Both Software and SoftwareVersion concepts have properties indication their installation and modification dates (namely, data_creation and date_mod). For consistency, these properties are considered to be synonymous (using *sameAs*) with CreationDate and ModificationDate properties.

4.1.3 VuMS: Vulnerabilities and Vulnerability Exposures

The VuMS stores, manages and discovers vulnerabilities, either through its own internal tools or through other vulnerability discovery methods. Here, a Vulnerability is known to affect a particular SoftwareVersion or Configuration, which are installed in specific Assets. A Vulnerability Exposure is an Event in which a new Vulnerability has been discovered and added to the system. A Vulnerability may be known but not necessarily be an issue, so long it affects Configurations that are not installed on any specific Assets (or, at least, not on those with high criticalities). An Event that exploits a known Vulnerability may be retroactively reclassified as an Attack. As for the Vulnerability's properties, it is worth noting that CVE_ID refers to the specific Common Vulnerabilities and Exposures (CVE) ID in the cases the Vulnerability has been identified by existing tools, URL points to the online description of this Vulnerability and Score, as indicated by its name, represents its possible threat/priority level in a scale of 1 to 10.

4.1.4 Incidents, Impact and Assessment

Within the SATIE scenarios, Incidents are generated exclusively through manual means by a SOC operator. The operator goes through a list of Events and determines whether these are related and should be considered an Incident. After this assessment, it is possibly to query existing tools (namely the Impact Propagation Simulation and Business Impact Assessment tools) about what the Incident's

estimated Impact. Because this information is exclusively shared through visual means (except parts of the impact propagation graph) and because the assessment of the Incident's Impact is not one of the concerns of the SATIE's scenarios, it was agreed by all concerning partners that the Incident description provided by IODEF (in which an Incident has an Assessment, which has an Impact) was excessive. As such, a simpler version of the relationships was designed that excludes the Assessment concept.

The Impact's specification is directly related to the needs of the Business Impact Assessment and Impact Propagation Simulation tools, describing how the Performance of Assets may be affected, how Assets affect each other and suggesting possible Mitigation Strategies, each with their own expected performances. How different Events and Assets may affect each other is described by the ThreatPropagationPath and ThreatPropagationEvent concepts respectively. Through reasoning it is possible to automatically assess which Assets are affected by a given Incident, although this list may not be exhaustive: the SOC operator, through the analysis of the visualizations provided by the impact assessment tools, may add more Assets to this list *a posteriori*.

4.1.5 Event types

In order to allow the Correlation Engine to generate richer correlations between Events, these have been classified into a number of categories. These correspond roughly to the outputs of the different tools whose communications are under scrutiny in this document, but different tools may output more than one type of Event. Here, a Correlation is a type of Event that shows the relationship between two or more Events, either by showing the direct correlation between these or by showing similar Events that occurred in the past that were also correlated as a justification. The combination of *score_p* and *threshold* allows the operator to establish whether the correlation is of interest or not. Additionally, given the outputs of the Correlation Engine and of the Incident Management Portal, some Events may represent not only a change in a specific Asset, but the action through which an Asset modifies another Asset. In order to describe this situation, the properties *sourceAsset* and *targetAsset* have been added to the description of Event, and their usage is optional. On that regard, it is important to note that the properties listed here are but a selection of the most relevant ones, and this list is not exhaustive. Additional, optional properties may be added to the Events as necessary with no detriment to the communications or reasoning processes.

5 Conclusions

This document described the process through which an ontology to define the semantics of the communications between the different SATIE's systems was developed. The main goal of this ontology is to promote the interoperability between the existing systems, while facilitating the process of including new ones in the future by stipulating the semantic contents of the messages. Additionally, the application of this ontology opens the possibility of more complex reasoning processes over the exchanged contents, which may be particularly relevant for the Correlation Engine's and the Investigation Tool's purposes.

This process started with a study of existing ontologies and the requirements provided by each of the involved partners, who specified, for each tool, the tasks they are expected to perform, their inputs and outputs. From here, it was possible to extract a set of recurrent concepts which were used as the starting point for the ontology's development process. These would help establishing which of the existing ontologies previously studied were more suited for use within the SATIE's scenarios, and how these could be combined and extended to fulfil all communication's need. This resulted in a proposed bridge between the UCO and ATMONTO ontologies and, therefore, bringing together the cyber-security and airports domains by defining how Events and Alerts triggered by airport elements can be used to enhance cyber-secure solutions. These ontologies were further enriched with hierarchies both Assets and Events that reflect the needs of the SATIE's systems, but are open enough to be exploited in other scenarios.

Another consequence of the systematization process required for the development of the ontology was the definite specification of the responsibilities of each tool and the information they exchange within the system.

6 References

Aime, M., & Guasconi, F. (2010). Enhanced vulnerability ontology for information risk assessment and dependability management. *3rd International Conference on Dependability (DEPEND 2010)*, (pp. 92-97).

Bobrow, R. (2006). *Intelligent Semantic Query of Notices to Airman (NOTAMs).*

Brost, W. N. (1997). Construction of Engineering Ontologies for Knowledge Sharing and Reuse. *Enschede: Universiteit Twente*.

Chen, Y., Peng, X., Zhong, B., & Luo, H. (2016). Application of ontology in vulnerability analysis of metro operation systems. *Structure and Infrastructure Engineering, 12*, 1256-1266.

Choraś, M., Kozik, R., Flizikowski, A., & Hołubowicz, W. (2010). Ontology applied in decision support system for critical infrastructures protection. *Lecture Notes in Computer Science, 6096*, pp. 671-680.

Cuppens-Boulahia, N., Cuppens, F., Autrel, F., & Debar, H. (2009). An ontology-based approach to react to network attacks. *International Journal of Information and Computer Security, 3*, 280-305.

Data Breach QuickView – 2015 Data Breach Trends. (2015). Retrieved from Risk Based Security: https://www.riskbasedsecurity.com/2015-data-breach-quickview/

Hobbs, J. R., & Little, C. (2020, March 26). *Time Ontology in OWL.* Retrieved from W3C - World Wide Web Consortium: https://www.w3.org/TR/owl-time/

Jafer, S., Chhaya, B., Durak, U., & Gerlach, T. (2016). Formal scenario definition language for aviation: aircraft landing case study. *AIAA Modeling and Simulation Technologies Conference.*

Keller, R. M. (2018, March). *The NASA Air Traffic Management Ontology.* Retrieved from Intelligent Systems Division, NASA Ames Research Center: https://data.nasa.gov/ontologies/atmonto/ATM

Krauß, D., & Thomalla, C. (2016). Ontology-based detection of cyber-attacks to SCADA-systems in critical infrastructures. *6th International Conference on Digital Information and Communication Technology and its Applications (DICTAP 2016)*, (pp. 70-73).

Martimiano, L. A., & Moreira, E. (2005). An owl-based security incident ontology. *8th International Protégé Conference*, (pp. 43-44).

Oltramari, A., Cranor, L., Walls, R., & McDaniel, P. (n.d.). Building an ontology for cyber security. *CEUR Workshop Proceedings*, *1304*, p. 5461.

SATIE project. (2019). *D1.2 - Specification of the overall technical architecture.*

SATIE project. (2019). *D2.5 - Specification of Impact Propagation Model.*

SATIE project. (2019). *DX.X - name of the SATIE deliverable you want to reference.*

Struder, R., Richard, B. V., & Fensel, D. (1998, March). Knowledge Engineering: Principles and Methods. *Data Knowledge Engineering, 25*, pp. 161-197.

Syed, Z., Pädia, A., Finin, T., Mathews, L., & Joshi, A. (2016). UCO: A Unified Cybersecurity Ontology. *AAAI Workshop - Technical Report*, (pp. 192-202).

Tamea, G., Cusmai, M., Palo, A., Priscoli, F. D., & Cimmino, A. (2014). Situation awareness in airport environment based on Semantic Web technologies. *2014 IEEE International Conference on Cognitive Methods in Situation Awareness and Decision Support (CogSIMA 2014)*, (pp. 174-180).

Ulicny, B. E., Moskal, J. J., Abe, M. M., & Smith, J. K. (2014). Inference and Ontologies. *Advances in Information Security, 62*, pp. 167-199.

Wang, J., & Guo, M. (2009). OVM: An ontology for vulnerability management. *ACM International Conference Proceeding Series.*

Wang, J., Guo, M., & Camargo, J. (2010). An ontological approach to computer system security. *Information Security Journal*, 61-73.

7 Annex: Common Cybersecurity Definitions

Table 7.1: Common definitions for the Alert concept

Source	Description
NIST	Notification that a specific attack has been directed at an organization's information systems.
Cybrary.it	Alert Situation: An alert situation is when the interruption in an enterprise is not resolved even after the competition of the threshold stage, an alert situation requires the enterprise to start escalation procedure.

Table 7.2: Common definitions for the Event concept

Source	Description
NIST	Any observable occurrence in an information system. Any observable occurrence in a network or system. Something that occurs within a system or network. Any observable occurrence on a manufacturing system. Events can include cybersecurity changes that may have an impact on manufacturing operations (including mission, capabilities, or reputation). Security Relevant Event: Any event that attempts to change the security state of the system (e.g., change access controls, change the security level of a user, change a user password). Also, any event that attempts to violate the security policy of the system (e.g., too many logon attempts).
Cybrary.it	An Event is an action or an occurrence that a program can detect. Examples of some events are clicking of a mouse button or pressing the key, etc.

Table 7.3: Common definitions for the Incident concept

Source	Description
NIST	An occurrence that actually or potentially jeopardizes the confidentiality, integrity, or availability of an information system or the information the system processes, stores, or transmits or that constitutes a violation or imminent threat of violation of security policies, security procedures, or acceptable use policies. Anomalous or unexpected event, set of events, condition, or situation at any time during the life cycle of a project, product, service, or system. A violation or imminent threat of violation of computer security policies, acceptable use policies, or standard security practices. Cyber incident: Actions taken through the use of an information system or network that result in an actual or potentially adverse effect on an information system, network, and/or the information residing therein. See incident. See

	also event, security-relevant event, and intrusion.
Cybrary.it	An incident is an unplanned disruption or degradation of a network or system service and needs to be resolved immediately. An example of an incident is a server crash that causes a disruption in the business process. However, if the disruption is planned, say, a scheduled maintenance, it is not an incident.
ANSSI	A security incident is an event that affects the availability, confidentiality or integrity of a property. Examples: Illegal use of a password, theft of computer equipment, intrusion into a file or application, etc.
HR NCSS	Computer security incident: one or more computer security events that have disturbed or are disturbing the security of the information system.
CNCS GNS	An event with a real adverse effect on the security of networks and information systems.

Table 7.4: Common definitions for the Vulnerability concept

Source	Description
NIST	Weakness in an information system, system security procedures, internal controls, or implementation that could be exploited or triggered by a threat source. A flaw or weakness in system security procedures, design, implementation, or internal controls that could be exercised (accidentally triggered or intentionally exploited) and result in a security breach or a violation of the system's security policy. A security exposure in an operating system or other system software or application software component. A variety of organizations maintain publicly accessible databases of vulnerabilities based on the version numbers of software. Each vulnerability can potentially compromise the system or network if exploited. Software Vulnerability: A security flaw, glitch, or weakness found in software that can be exploited by an attacker.
ANSSI	Faulty, malicious or clumsy, in the specifications, design, realization, installation or configuration of a system, or in the way of using it. Notes: A vulnerability can be used by an exploit code and lead to an intrusion into the system.
CULSIT	It represents an intrinsic weakness or due to conditions of exercise or lack of controls, which can be exploited by a threat to cause damage.
CNCS GNS	Weakness of an asset or control that may be exploited by a threat.

Table 7.5: Common definitions for the Threat concept

Source	Description
NIST	An event or condition that has the potential for causing asset loss and the undesirable consequences or impact from such loss. Note: The specific causes of asset loss, and for which the consequences of asset loss are assessed, can

	arise from a variety of conditions and events related to adversity, typically referred to as disruptions, hazards, or threats. Regardless of the specific term used, the basis of asset loss constitutes all forms of intentional, unintentional, accidental, incidental, misuse, abuse, error, weakness, defect, fault, and/or failure events and associated conditions. A possible danger to a computer system, which may result in the interception, alteration, obstruction, or destruction of computational resources, or other disruption to the system. Cyber Threat: An event or condition that has the potential for causing asset loss and the undesirable consequences or impact from such loss. Note: The specific causes of asset loss, and for which the consequences of asset loss are assessed, can arise from a variety of conditions and events related to adversity, typically referred to as disruptions, hazards, or threats. Regardless of the specific term used, the basis of asset loss constitutes all forms of intentional, unintentional, accidental, incidental, misuse, abuse, error, weakness, defect, fault, and/or failure events and associated conditions.
Cybrary.it	A threat is a possible danger that might exploit a vulnerability to violate security protocols and thus, cause possible harm. A threat can be either deliberate (example, an individual cracker or a criminal organization) or accidental (example, the possibility of a computer malfunctioning, or the possibility of a natural disaster such as an earthquake, a fire, or a tornado) or otherwise a circumstance, capability, action, or event.
CULSIT	The threat is defined as an event of malicious or accidental nature which, by exploiting a vulnerability of the system, could cause damage.
IT NCSS	We define the cyber threat as the complex malicious conducts that can be exercised in and throughout cyberspace, or against cyberspace and its fundamental elements.
GNS	Potential cause of an unwanted incident that could cause damage to a system, individual or organization.