Bruno Augusti Mozzaquatro

Mestre em Ciência da Computação

Security Management Framework for the Internet of Things

Dissertação para obtenção do Grau de Doutor em

Engenharia Electrotécnica e de Computadores

Orientador: Ricardo Luís Rosa Jardim Gonçalves, Professor Catedrático, Faculdade de Ciências e Tecnologia da Universidade Nova de Lisboa

Co-orientador: João Francisco Martins, Professor Associado, Faculdade de Ciências e Tecnologia da Universidade Nova de Lisboa

Júri

Presidente: Doutor Luís Manuel Camarinha de Matos

Arguentes: Doutor Bruno Andó
Doutora Teresa Cristina de Freitas Gonçalves

Vogais: Doutor Ricardo Luís Rosa Jardim Gonçalves
Doutora Anabela Monteiro Gonçalves Pronto
Doutor Carlos Manuel Melo Agostinho
Doutora Manuella Kadar

FACULDADE DE CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE NOVA DE LISBOA

Novembro, 2020

Security Management Framework for the Internet of Things

Este documento foi gerado utilizando o processador (pdf)LaTeX, com base no template "unlthesis" [1] desenvolvido no Dep. Informática da FCT-NOVA [2].
[1] https://github.com/joaomlourenco/unlthesis [2] http://www.di.fct.unl.pt

To my family.

Acknowledgements

I would like to start my acknowledgements to thank my wife, Karine Fontana Alves, to encouragement and support to face all the difficulties met in his period.

For my family, especially my parents Claurindo and Aneli and my brothers Nilton and Nádia, who supported me all those years far away from home during this doctorate. My advisor, Dr. Ricardo Gonçalves for believing in my abilities and giving me his advice for the successful completion of this work, and the great opportunity to work in FCT/Uninova. I would like to thank Prof. João Martins for his support to complete this work. I also want to acknowledge Carlos Agostinho contribution to this work. I grew a lot while working with him in these years at Uninova. The experience that this period gave was very valuable.

To all my colleagues, especially to José Ferreira, Rui Lopes and Pedro Magalhães, who helped me and believed me at this moment, although it often seemed that I was not going to arrive, pushing me to do a better job and successfully complete this dissertation. For my friends, Ricardo Martini, Rafael Pereira, Leandro Freitas, Renato de Azevedo, Thais Baldissera, Cristiano de Faveri, Vagner Schaefer and Walter Priesnitz, for always giving me support and motivation to finally finish this dissertation, which gave me hope and believed in me all the time, made me believe in myself, although I do not always show how much I appreciate to both of you.

Finally, my great thanks to my institution, FCT for hosting this research. I also acknowledge the support by CAPES - Ciência Sem Fronteiras Program, BEMundus Erasmus Program, and Universidade Federal de Santa Maria - which provided me the resources needed to undertake my PhD work.

Abstract

The increase in the design and development of wireless communication technologies offers multiple opportunities for the management and control of cyber-physical systems with connections between smart and autonomous devices, which provide the delivery of simplified data through the use of cloud computing. Given this relationship with the Internet of Things (IoT), it established the concept of pervasive computing that allows any object to communicate with services, sensors, people, and objects without human intervention. However, the rapid growth of connectivity with smart applications through autonomous systems connected to the internet has allowed the exposure of numerous vulnerabilities in IoT systems by malicious users.

This dissertation developed a novel ontology-based cybersecurity framework to improve security in IoT systems using an ontological analysis to adapt appropriate security services addressed to threats. The composition of this proposal explores two approaches: (1) design time, which offers a dynamic method to build security services through the application of a methodology directed to models considering existing business processes; and (2) execution time, which involves monitoring the IoT environment, classifying vulnerabilities and threats, and acting in the environment, ensuring the correct adaptation of existing services.

The validation approach was used to demonstrate the feasibility of implementing the proposed cybersecurity framework. It implies the evaluation of the ontology to offer a qualitative evaluation based on the analysis of several criteria and also a proof of concept implemented and tested using specific industrial scenarios. This dissertation has been verified by adopting a methodology that follows the acceptance in the research community through technical validation in the application of the concept in an industrial setting.

Keywords: Cybersecurity Framework; Internet of Things; Security Ontology; Security service Provisioning; Industry 4.0

Resumo

O aumento no projeto e desenvolvimento de tecnologias de comunicação sem fio oferece múltiplas oportunidades para a gestão e controle de sistemas ciber-físicos com conexões entre dispositivos inteligentes e autônomos, os quais proporcionam a entrega de dados simplificados através do uso da computação em nuvem. Diante dessa relação com a Internet das Coisas (IoT) estabeleceu-se o conceito de computação pervasiva que permite que qualquer objeto possa comunicar com os serviços, sensores, pessoas e objetos sem intervenção humana. Entretanto, o rápido crescimento da conectividade com as aplicações inteligentes através de sistemas autônomos conectados com a internet permitiu a exposição de inúmeras vulnerabilidades dos sistemas IoT para usuários maliciosos.

Esta dissertação desenvolveu um novo framework de cibersegurança baseada em ontologia para melhorar a segurança em sistemas IoT usando uma análise ontológica para a adaptação de serviços de segurança apropriados endereçados para as ameaças. A composição dessa proposta explora duas abordagens: (1) tempo de projeto, o qual oferece um método dinâmico para construir serviços de segurança através da aplicação de uma metodologia dirigida a modelos, considerando processos empresariais existentes; e (2) tempo de execução, o qual envolve o monitoramento do ambiente IoT, a classificação de vulnerabilidades e ameaças, e a atuação no ambiente garantindo a correta adaptação dos serviços existentes.

Duas abordagens de validação foram utilizadas para demonstrar a viabilidade da implementação do framework de cibersegurança proposto. Isto implica na avaliação da ontologia para oferecer uma avaliação qualitativa baseada na análise de diversos critérios e também uma prova de conceito implementada e testada usando cenários específicos. Esta dissertação foi validada adotando uma metodologia que segue a validação na comunidade científica através da validação técnica na aplicação do nosso conceito em um cenário industrial.

Palavras-chave: Framework de cibersegurança; Internet das Coisas; Ontologia de Segurança; Provisionamento de Serviços de Segurança; Indústria 4.0

Contents

List of Figures

List of Tables

Listings

Acronyms

6LowPAN Low power Wireless Personal Area Network.

AMQP Advanced Message Queuing Protocol.

API Application Programming Interface.

BPMN Business Process Model and Notation.

CA Certificate Authority.

CoAP Constrained Application Protocol.

CPS Cyber-Physical System.

CVE Common Vulnerabilities and Exposures.

CWE Common Weaknesses Enumeration.

DDoS Distributed Denial of Service.

DTLS Datagram Transport Layer Security.

EA* Extended Diagram Star.

ECC Elliptic Curve Cryptographic.

GPRS General Packet Radio Service.

GSM Global System for Mobile.

IDMEF Intrusion Detection Message Exchange Format.

IDS Intrusion Detection System.

IEEE Institute of Electrical and Eletronics Engineers.

IIoT Industrial Internet of Things.

IoT Internet of Things.

IP Internet Protocol.

IPSec IP Security Protocol.

ITU International Telecommunications Union.

JVM Java Virtual Machine.

M2M Machine-to-Machine.

MAC Medium Access Control.

MDSEA Model-Driven Service Engineering Architecture.

MQTT Message Queuing Telemetry Transport.

NFC Near Field Communication.

NVD National Vulnerability Database.

OWL Ontology Web Language.

PaaS Platform-as-a-Service.

QoS Quality of Service.

RDF Resource Description Framework.

RFID Radio-Frequency Identification.

RPL Routing Protocol for Low Power and Lossy Networks.

SOA Service-Oriented Architecture.

SPARQL SPARQL Protocol and RDF Query Language.

SSL Secure Socket Layer.

SWRL Semantic Web Rule Language.

TLS Transport Layer Security.

UDP User Datagram Protocol.

VPN Virtual Private Network.

W3C World Wide Web Consortium.

XMPP Extensible Messaging and Presence Protocol.

CHAPTER 1

Introduction

The popularization of many emerging technologies and wireless devices enables them to provide better products and personalized services to achieve the needs of the customers. The advances of wireless communication systems enabled users to become easily connected to the global network, but also allowed an interlink between the physical world and the cyberworld, in particular through the use of Cyber-Physical Systems (CPS) and Internet of Things (IoT) [1, 2]. CPS refers to the use of physical components (e.g., actuators) to gather data through sensors, to be processed, and used in the cyber world. CPS has affected the society and humans with provisioning of the interacting networks of physical and computational equipment of smart services and critical infrastructure. The term IoT became popular in the late 1990s after having several technologies associated with sensor development, machine control, connected to the World Wide Web [3, 4].

However, recent developments in wireless sensor networks and Industry 4.0 motivated the expansion of IoT applications to different domains such as industrial internet [5], smart cities [6], smart grid [7], and healthcare [8]. Some emerging technologies of IoT, as well as RFID, are allowing the identification of objects using tags, which have the capabilities to store small quantities of data on the industrial environments [9]. Also, the IoT in the industrial context is known as the Industrial Internet of Things (IIoT) or Industry 4.0. However, the popularization of smart sensors connected to the Internet is a characteristic of the IoT, in which the number of things surpassed the human population in 2010, also, there is a prediction that it is reaching 50 billion things in 2020 [10].

However, smart devices are becoming even more relevant to our lives and changing habits with activities automation. For instance, in the case of the elderly, small sensors have transformed their lives with tasks of self-monitoring using a combination of wearable devices to allow seniors to age with comfort and health. In another case, industries have less time of production with the use of sensors in the shop floor, creating the next level of productivity, enabling manufacturers to both better align with customer

preferences and build a collaborative ecosystem of partnerships. Some industries have exploited all activities deploying IoT applications for creating intelligent transportation, monitoring process, service sector, and manufacturing systems [9]. According to these scenarios, IoT technologies are becoming central between different contexts, and cybersecurity is of extreme importance.

The development of new IoT technologies enables the emergence of threats, and new approaches on security management are required to establish more secure and available smart devices and services in the scenario of IoT. Nowadays, 70% of the IoT devices have a vulnerability, at least one piece of personal information [11]. These vulnerabilities are potential to prompt security incidents, especially in critical sectors (e.g., health, military, industrial), because it could result in dangerous problems for humans' lives.

There are many security and privacy challenges to IoT [12] such as user privacy and data protection, authentication, and identity management, trust management and policy integration, authorization and access control, end-to-end security, and attack resistant security solution. To address these challenges and to protect smart devices, some requirements are dealing with security challenges in IoT network [12]: lightweight and symmetric solutions to support resource-constrained devices and establishment of trust relationships; cryptographic techniques to protect data processing; use distributed computing and critical management to maintain information decentralized.

1.1 Central Challenge and Motivation

One of the most critical aspects regarding the complete adoption of the Internet of Things in the real world is cybersecurity [13]. Faced with new wireless technologies, IoT presents several opportunities for an expansion of services, creating more effective real-time functions, and providing diversification of business models with global visibility. However, the connectivity of numerous IoT systems imparts several challenges and possible threats. The protection of the IoT has several cybersecurity challenges associated, which increases the task for security experts because it involves security provisioning services to billions of objects. Also, the high number of incidents with IoT technologies is one of the main challenges that academia must have attention when discussing the future of cybersecurity.

As shown in the scenarios presented in Figure 1.1, there are several threats present within IoT systems such as spoofing, traffic sniffing, manipulation of sensitive information, code injections, and unauthorized access. Besides, the attacks can occur at different points in the IoT system, which stresses the importance of cybersecurity within IoT systems. According to [14], the design of IoT systems must consider the operation under a unified view of safety and security characteristics because both deal with the physical world and in a lot of times with critical activities with unsafe impacts.

IoT cybersecurity involves the inherent complexity of the IoT, which is further aggravated by the multiple heterogeneous exchanges of information, which occurs

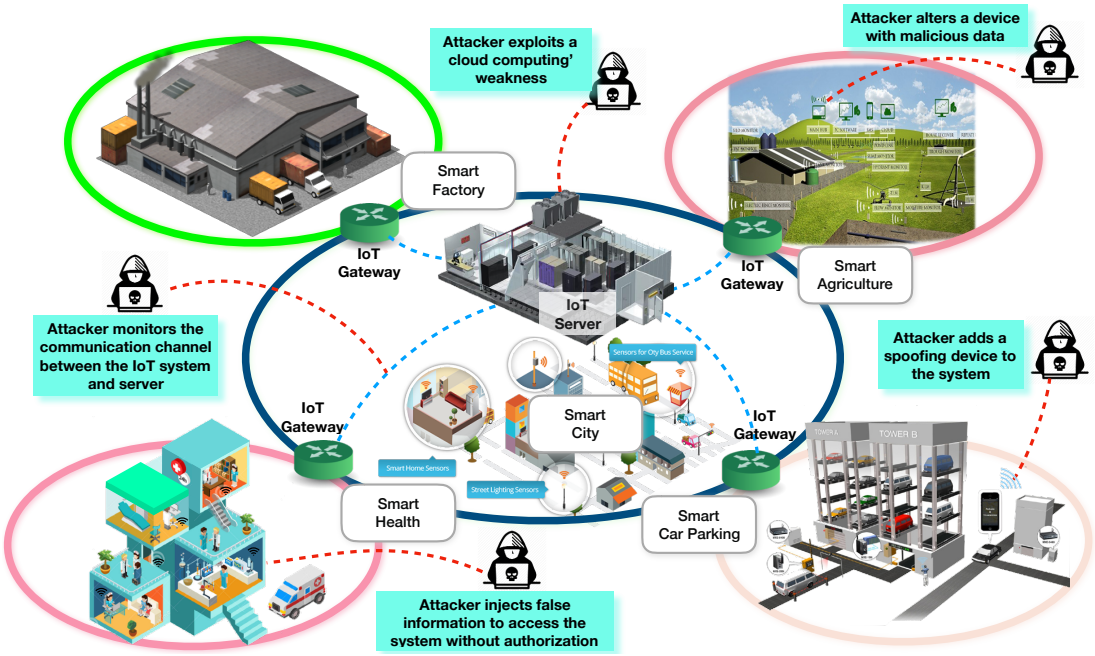


Figure 1.1: Potential security challenges for IoT ecosystem.

between IoT devices. Due to these characteristics, IoT devices become increasingly exposed to computer networks. The latter holds due to the ever-changing risk of new attacks and undiscovered vulnerabilities present within the IoT [15]. The interconnectivity present amongst different devices may potentially undergo significant changes in the form of new features by a third-party not involved in the process, which may yield potential security weaknesses. On the other hand, some devices have a specific design to work in local networks or indirect internet access. In this case, IoT cybersecurity requirements must be accounted to ensure the security of IoT devices within networks.

The major problem of cybersecurity within the IoT ecosystem involves a lack of knowledge of the essential components of cybersecurity: Assets, Threats, Security Mechanisms, Vulnerabilities, and Security Properties. A plausible explanation for the latter is the fact that different IoT systems require distinct protection mechanisms to avoid intrusions from the physical and cyber world [16]. As a result, there are several threats to exploit sensitive information, which can endanger and compromise the adoption of IoT systems and their growth rate. In other words, a compromised IoT device can be looked upon as an entry point for malicious users to gather user sensitive information, which results in losing integrity and confidentiality.

As technological progress occurs and the development of the IoT becomes more prevalent than the last decade, there is an increase of connected devices that can network and communicate with other web-enabled devices. As a higher amount of data is shared, there is a greater emphasis placed on data security. Although cybersecurity has always been a matter of extreme relevance, data security has earnt increased relevance as a

result of the growing use of intelligent systems by businesses in various industries worldwide. Therefore, the concept of IoT cybersecurity can be looked upon as critical when identifying approaches to ensure data security during the rise of IoT.

This crucial challenge affects the adoption of IoT in multiple domains. Indeed, a 2016 survey conducted by Dell company, found that cybersecurity professionals were 49% more likely to spend additional time securing their data with sufficient information if they correctly understood the risks and threats faced by cybersecurity within the next five years [17]. There are several traditional security mechanisms to mitigate specific threats. Despite this, the use of intelligent capabilities in IoT systems allows us to gather security information from network probes, and consequently processed, which allows for situation-based security solutions in the form of services and tools. The adoption of IoT devices and sensors without accounting for the presence of possible security weaknesses to perform everyday actions need, more attention to avoid potential security problems because these devices are vulnerable to eavesdropping, spoofing, tampering, jamming, and so forth.

1.2 Vision

The research work envisions the proposal of an ontology-based framework focused on the provisioning of suitable security solutions using an appropriate formal conceptualization of security components through services. The purpose of this approach is to provide service provisioning using the novel IoTSec ontology while considering adjustments to deal with the dynamism and flexibility of IoT systems. The main contributions refer to an identification of the cybersecurity challenges of the existing IoT systems, the implementation of a new cybersecurity framework to provide suitable security solutions using the proposed IoTSec ontology, and the validation of model-driven services provisioning.

This approach should also support the cybersecurity management with a knowledge base to increase the information available for threat analysis, allowing better prediction, prevention, and mitigation of cyber threats. The proposed framework will be able to share, collect, and aggregate information from distinct sources and use data collection from sensors to provide controlled actions to actuators.

As part of the research, the author also wants to address service provisioning, resulting in the expansion of various socio-economical factors within the global market. The ontology-based approach allows a correlation between the cybersecurity components and proposes solutions to mitigate weaknesses and potential attacks. This approach has the potential to evaluate a collection of data sensors with individual security requirements due to dominant ontology languages, which allows for the machine processing of information with sufficient expressive power to support reasoning in formal semantics.

1.3 Adopted Research Method

The scientific method is the process to build and develop an approach following a systematic observation, measurement, and experiment based on formulation, testing, and modification of hypotheses. The scientific research process is known by which scientists, collectively and over time, endeavor to construct a real representation based on experiments without influences of the researchers' bias on the outcome of the experiment. Figure 1.2 depicted classical phases of the scientific method:

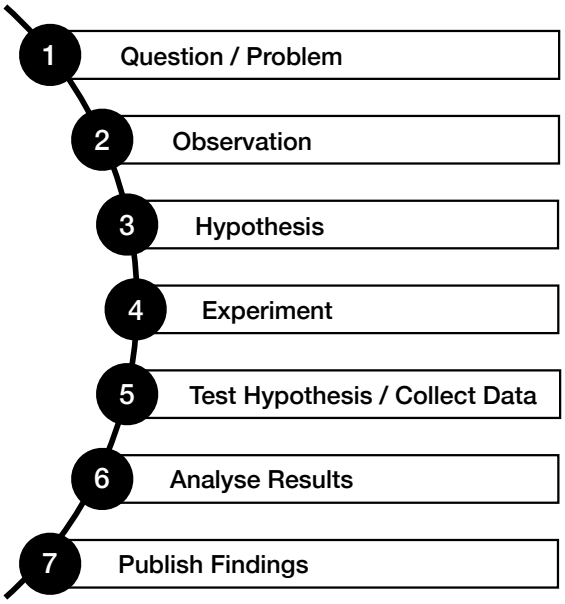


Figure 1.2: Classical phases of the scientific method.

The scientific method adopted in this dissertation is based on prior knowledge to formulate the research question and to investigate the results found. The previous knowledge can contribute with formulation and path to explore the research topic. Hence, several iterations between phases are defined to give support to the research results. Thereafter, Figure 1.3 presents the scientific method selected to produce results based on experiments with a focus on the investigation.

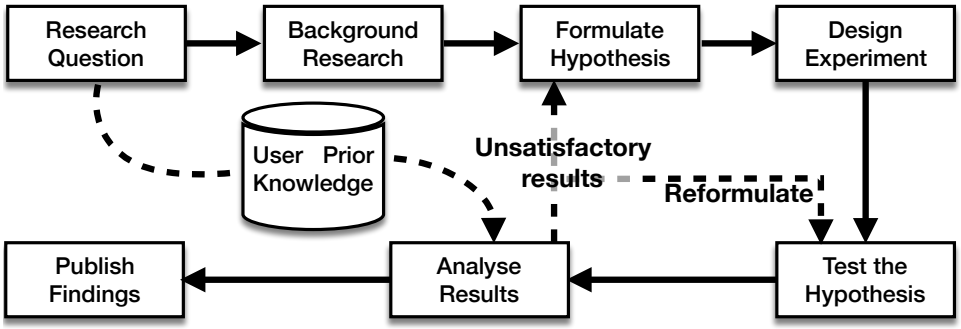


Figure 1.3: Adopted research method.

The steps of the adopted scientific method are described as follows:

Research question: It is the most crucial step in research to encompass the scope of the work and follow the systematic steps until a conclusion is achieved based on the analyzed results. This research question must be confirmed or refuted according to the test of the hypothesis.

Background research: This step explores similar works to distinguishes that the researcher wants to do. Typically, a study of the art is conducted by research and join discussion groups to verify if the work has been done previously, to see similar approaches, and construct an approach with proper perspective to make an impact on this research.

Formulate hypothesis: One or more scientific hypotheses can be defined to test each research question. Usually, it brings clarity, specificity, and focus on a research problem [18]. In the case of unsatisfactory results during the more advanced steps, hypotheses are reformulated.

Design experiment: This step demonstrates that the study is feasible. Engineering research often includes the design of prototypes or system architectures to establish variables to be manipulated and measured as well as the research outcomes must be measurable [18]. Nevertheless, thesis validation must be achieved with the design experiment in this step.

Test Hypothesis: The testing of the hypothesis includes the implementation of the prototype, collection of data, and execution tests according to the pre-defined validation method. This step demonstrates signals of needs to adapt the prototype design and confirm the result by retesting. Also, ethical issues need to be considered.

Analyse results: A quantitative and qualitative analysis enables to evaluate the outcomes of the experiments. In this step, an insightful analysis must be conducted to create discussions about the means of the results to the research question. If the conclusion fails the tests, it must be rejected or modified for testing again, returning the step 3. In the case of the abandoned, the PhD must return to step 1.

Publish findings: Research results represent the advance to the community, and it is highly recommended to publish final findings and provide peers a critical analysis of the subject. Typically, conferences are published with intermediate results to collect feedback. The focus on the publications in reference journals is the consolidated results.

During this PhD work, the classical steps of the scientific method will be followed. During these steps, common findings will target publications at recognized conferences and journals, and backward loops will apply if required.

1.4 Research Problem

Ubiquitous developments of IoT technologies have the potential to impact the global economy and create real economic value. On the first view, the potential IoT technologies in the global economy is to identify correctly areas within which IoT technologies can

have the most significant value within the economy. For instance, Manyika [19] argue that IoT technologies can reach a total economic value of $11.1 trillion by 2025, a value that is equivalent to approximately 11% of the world economy. As an increase in the adoption of Industry 4.0 arises, productivity will boost among the manufacturing sectors. In the case of Germany, one of the first countries to adopt Industry 4.0, this ranges between 15% and 25% [20].

Therefore, IoT cybersecurity is a factor that must be highlighted when identifying the security implications of the emergence of industrial IoT [21]. From an economic perspective, this does not deter the impact that it has on the productivity of a country's workforce, a key macroeconomic variable that is linked with a country's level of economic growth. However, the presence of security vulnerabilities' within IoT technologies is a predominant aspect that could hinder its potential within specific business organizations [22]. Such organizations deal with a high volume of user sensitive data that highlights the privacy concerns of IoT technologies.

The risk management strategies implemented by enterprises must be tailored depending on the potential risk presented. Firstly, the ability of IoT to increase the efficiency of manufacturing processes and supply chain networks relies heavily upon interoperability. The benefits yielded by interoperability in terms of increased labor productivity must be contrasted with the predominant threat of data exposure. This problem often occurs due to the absence of investment by enterprises to ensure that the IoT solutions implemented have been correctly tested during design time for potential vulnerabilities [23].

The greater use of IoT technologies by businesses and the greater adoption of Industry 4.0 is estimated to boost productivity across the manufacturing sector, with values ranging between 90 and 150 billion euros in Germany alone. Industry 4.0 is also estimated to generate a significant impact on gross production share. This impact is better understood when we place our scope on Germany and its manufacturing industries. Given this scenario, Rüßmann et al. [20] discusses that other industries are likely to account for 55% of gross production share in Germany due to the enhanced and more efficient manufacturing processes in Industry 4.0. However, there are a variety of different challenges concerning cybersecurity present within Industry 4.0 that pose a risk to the productivity gains that may be earn as a consequence of the technological advancements in IoT technologies [24].

The research question is established to be validated to attend the objectives of this dissertation, namely, the research problem:

Can ontologies enrich the capabilities of security management in IoT-enabled industrial environments, to enhance confidence in Industry 4.0?

1.5 Hypothesis

As new wireless technologies are adopted, new exploits are appearing in the IoT ecosystem with a prevalent focus on increasing the exposition of data to potential threats. As a result, there must be more excellent knowledge of the elements in IoT systems to understand correctly how the prevalent cybersecurity issues at hand can potentially impact the function of the system. As depicted in Figure 1.1, some threats try to manipulate data, access sensitive information, and monitor communication channels. Therefore, the author claims that "**If knowledge about known cybersecurity issues (e.g., vulnerabilities, known threats), and the corresponding prevention measures should integrate cybersecurity information in a comprehensive ontology that is accessible to run time monitoring and actuation tools, then security systems could be improved to automatically detect threats to the IoT network and dynamically propose or implement suitable protection services.**"

This dissertation presents an ontology-based cybersecurity framework to verify this hypothesis and improve IoT cybersecurity based on the problems as mentioned earlier. The main contributions of this dissertation are listed below.

Main contributions:

- The cybersecurity framework itself: an integrated technical framework using ontology and knowledge reasoning to address the security aspect of the Internet of Things within industrial environments. The framework focuses on the enterprise (company-side) monitoring, security analysis, and the subsequent security service design and provisioning to improve business processes and technology assets;
- The IoTSec ontology, which is a core component of the framework and a continued work of the authors (see [25]), gathering cybersecurity knowledge about alerts and possible threats and providing reasoning capabilities to discover implicit data from the contextual information of security issues;
- Design and orchestration method to implement and provide suitable security services in the IoT environments through the application of the Model-Driven Service Engineering Architecture (MDSEA) methodology (see [26]);
- Runtime security monitoring and actuation services integrated with the IDMEF standard (see [27]).

1.6 Dissertation Outline

This dissertation is organized into 8 chapters that compose the overall contents.

- **Introduction:** this chapter established the guidelines of the presented research work.

- **Internet of Things:** all the relevant information about the Internet of Things is presented in this chapter.
- **Cybersecurity in the Internet of Things:** the Internet of Things' requirements and challenges are identified and explained in detail to address the vulnerabilities, threats, and security mechanisms involved in this domain of interest.
- **Semantic Web and Ontology Management:** the semantic web and ontology concepts are presented together with cybersecurity ontologies proposed in the literature. The IoTSec ontology is also presented in detail with the modeling process to create a reference ontology.
- **Framework to Enable Cybersecurity in IoT Ecosystems:** the proposed framework to improve cybersecurity in the IoT domain is presented, and its implementation considerations are described in detail.
- **Proof-of-Concept Implementation:** the validation of the ontology-based security framework for IoT is performed considering several technologies to improve IoT cybersecurity improvements.
- **Implementation Assessment and Hypothesis Validation:** the hypothesis validation's main contributions are presented to improve IoT cybersecurity, considering the scientific and industrial validation.
- **Final Considerations and Future Works:** the final considerations of the developed work are established. Some future potential issues on what can be done using the present work to promote further developments are presented.

CHAPTER 2

Internet of Things

The term Internet of Things (IoT) has suffered several modifications because technologies are changing over time. In 1999, IoT was defined by Kevin Ashton in Auto-ID Centre at MIT as uniquely addressable things connected to form a dynamic worldwide network through sensors and a platform with the potential to improve everyday activities of our lives.

IoT has gained interest in industry and academy in recent years with emerging technology, and it has become a reality over the popularization of many different wireless devices. The IoT integrates part of Future Internet, and it is considered the new wave of information industry after the computer and the Internet [28]. This attribution belongs to the growth of many different technologies and services. Several things around us are combining several standards to provide new business and opportunities (M2M – Machine-to-Machine architecture) such as RFID, Zigbee, Wi-Fi, 3G, embedded devices, and sensing devices to ensuring the benefits of smart environmental monitoring, weather forecasting, transportation, healthcare, business.

The adoption of a new vision has distinct differences between companies because the interests of the organizations and research centers guide the approval of the use of technologies for specific purposes. According to Chase [29], IoT is defined as: "*The IoT creates an intelligent, invisible network fabric that can be sensed, controlled and programmed. IoT-enabled products employ embedded technology that allows them to communicate, directly or indirectly, with each other or the Internet*". This scope allows creating new services and applications to connect smart objects, integrating network technologies, devices, software technologies, infrastructure to build new business and services. The first technology for IoT was the Radio-Frequency IDentification (RFID) that allows identifying objects by tags, which have capabilities to storing small quantities of data. The industries have exploited all industrial activities deploying IoT applications for creating intelligent transportation, monitoring process, service sector, and manufacturing systems [9].

IoT is much more than these technologies and refers to a revolution of the Internet that

involves the connection of everything globally that through wired and wireless devices can interact with each other and cooperate to make the network of "things". IoT refers to an emerging paradigm consisting of a continuum of uniquely addressable things communicating one another to form a dynamic worldwide network [4].

2.1 The Thing Lifecycle

The IoT ecosystem consists of a network of interconnected nodes that performs many functionalities in different IoT applications and scenarios. These nodes, also called things, are resource-constrained devices such as luminaires and sensors. Some of them use a battery to operate and require sleep capabilities during their operation to save energy. Figure 2.1 shows the generic phases of a simplified model of the thing lifecycle. Initially, the users need to install the thing according to their manufacturer specifications in a bootstrapping phase. The interoperability is essential in this phase because the things used in the scenario are must not of the same manufacturer, which requires a trust bootstrapping to proceed with the sharing of the secret keys and the device identity to be used in regular operation. The operational phase is already ready to be used by multiple system users.

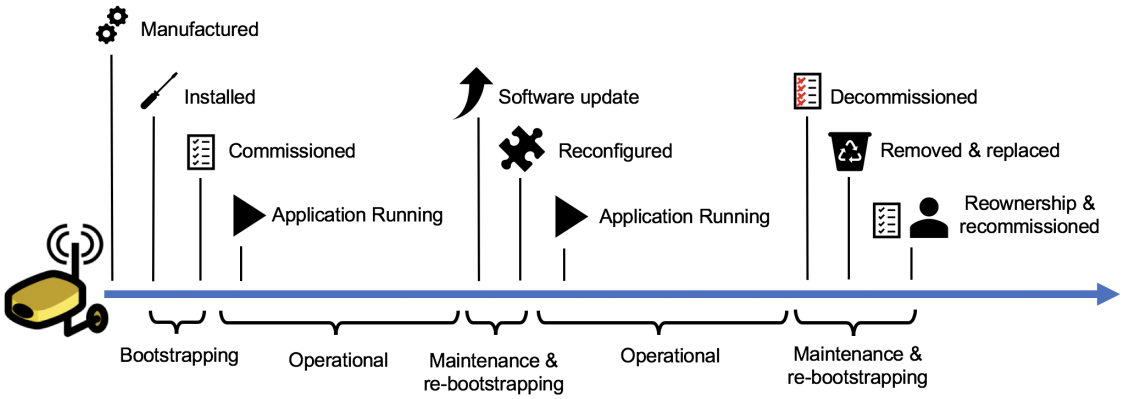


Figure 2.1: The lifecycle of a thing in the Internet of Things [30].

Each vendor has their maintenance policies to provide continuous software updates. In this phase, the thing can be updated or reconfigured. Many times these tasks require re-bootstrap at the end of a maintenance cycle to pass to the operational phase. Several times repeats this phase until the end of its lifecycle when the decommissioning of the device occurs. However, the end-of-life of a device denotes a need to replace or upgrade the network things for additional capabilities. Therefore, the thing can be removed and re-commissioned for a different IoT ecosystem under a different owner, thereby starting the lifecycle all over again.

2.2 IoT System Model

The exponential growth of Internet-connected devices consists of the use of straightforward sensors to a great variety of objects to multiple cloud servers, which involves the concept of IoT. The system model of IoT involves the network connectivity to allow all objects to connect to the Internet and exchange data, resulting in an integrated environment with the real world and less human intervention. The connectivity among IoT devices uses various protocols to provide a 'common' language of the IoT devices in terms of the message's format and a set of rules and regulations to organize the way of processing data and exchange messages between all involved parties. This system model represents the real meaning of the complexity to connect IoT devices to the cloud infrastructure through the Internet (Figure 2.2).

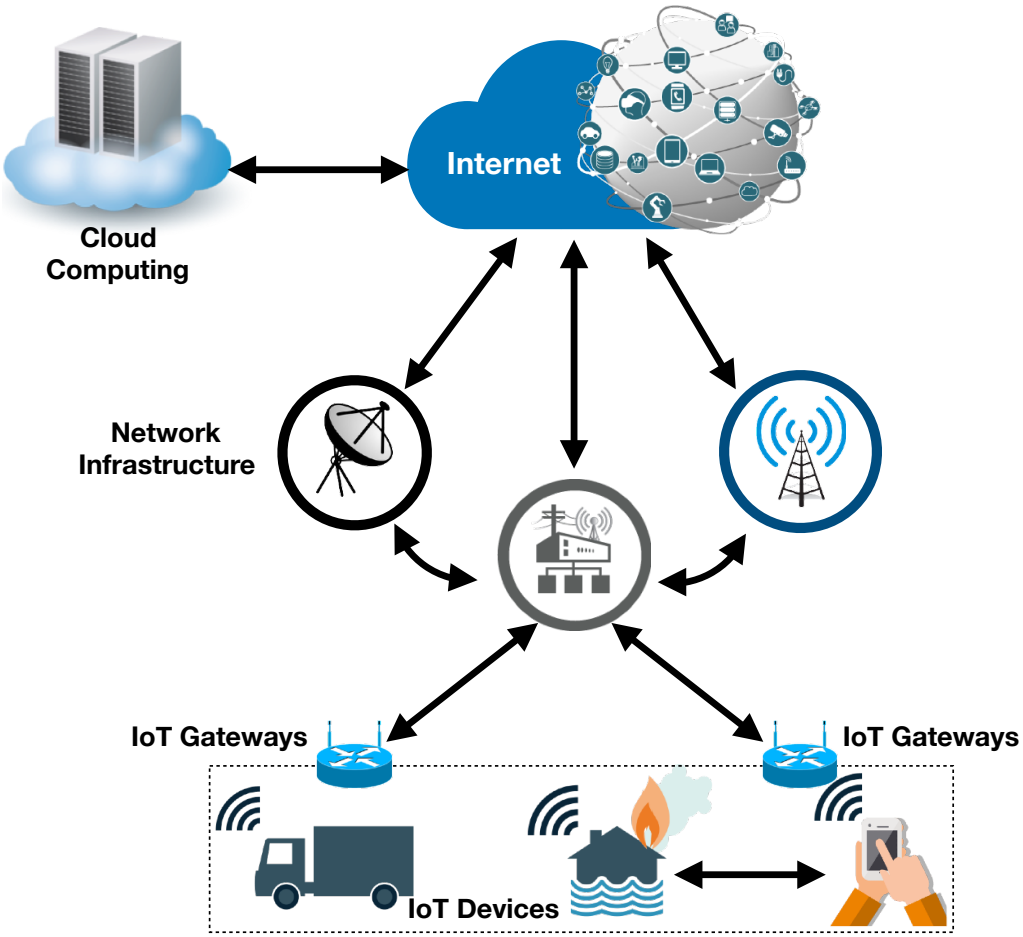


Figure 2.2: A system model of IoT.

Several technologies are associated with the IoT ecosystem, in which different communication models could be used to connect of IoT devices [31], such as device-to-device, device-to-cloud, device-to-gateway, and back-end data-sharing. The device-to-device communication model contains two or more devices connected using specific communication technologies such as Bluetooth, Z-Ware, or Zigbee. On the other

hand, several communication models use an intermediate server establishing direct communication between other devices and application servers (e.g., Cloud Computing). The device-to-cloud communication model uses IP-based or non-IP-based technologies, depending on the vendor that offers the IoT product. In a scenario with an application server to provide interoperability between smart devices of different manufacturers are used different Internet protocols like HTTP, TLS, and TCP by one device and the other uses CoAP, DTLS, and UDP.

The device-to-gateway communication model uses a local gateway between smart devices and a cloud service. The IoT gateway is responsible for bridging the interoperability gap between devices themselves. A scenario where smart objects require interoperability with non-IP-based devices uses this model. Back-end data-sharing refers to an architecture that supports collaboration between the different providers of the application server. This model explores the device-to-cloud communication model with data sharing using federated cloud services to achieve interoperability of smart devices data hosted.

These communication models provide the foundations for the different variations of interaction and connection between entities. According to Roman et al. [13], IoT enables services provisioning using centralized architectures or distributed architectures. In centralized architectures, digital entities acquire, process, and provide information. Also, distributed architectures use digital entities at the edge of the network to exchange information and collaborate dynamically.

Following the IoT system model, sensors devices have their categories according to their capabilities: nano, micro, and personal nodes [32]. Nano nodes are sensor devices with minimal hardware capabilities that have no processing power. Due to it, data collection from the environment often uses these devices. One example is the GPS sensor. The autonomy of these devices allows creating a communication system with thousands of them which are capable of sensing and monitor the physical and environmental conditions such as temperature and pressure. Usually, they collect data and send through many repeaters to the base station where the data is processed and stored to the storage server. Micro nodes consist of sensor devices with limited processing capabilities to perform only simple tasks such as a comparison of two values. The connectivity of microsensors can be fixed or wireless, but it only is possible to connect through gateways. This type of device can be programmable with embedded sensors and the base station, such as the example Sun SPOT using a high portable Java VM (Squawk). Personal nodes are sensor devices with processing capabilities and direct communication with other nodes. This processing power allows run operational embedded systems and monitor and actuate using this processing from the environment. There are many examples of all-in-one chipset solutions of a personal node such as Arduino and Raspberry Pi.

2.3 Internet of Things Architecture

The heterogeneity of communication technologies at specific environments interacting with IoT devices results in different requirements for IoT systems. A study published by Cisco [10] estimated the exponential growth of the number of Internet-connected devices, which will be 50 billion Internet-connected devices by 2020. It has an impact on architecture design and led to several numbers of proposed architectures. Some projects try to propose a typical architecture based on the analysis of industry and academy.

There are specific requirements to support emerging needs for the IoT ecosystem, e.g., the connectivity and communication heterogeneity of each IoT device. Other requirements affect the existing Internet-based approaches such as the device management, data collection, scalability, and security.

The basic architecture model for IoT follows the 3-layer architecture [33]: perception layer, network layer, and application layer. Others architecture models had been proposed in the literature as alternative models such as middleware based, SOA-based [34] and 5-layer [35, 36] model depicted in Figure 2.3.

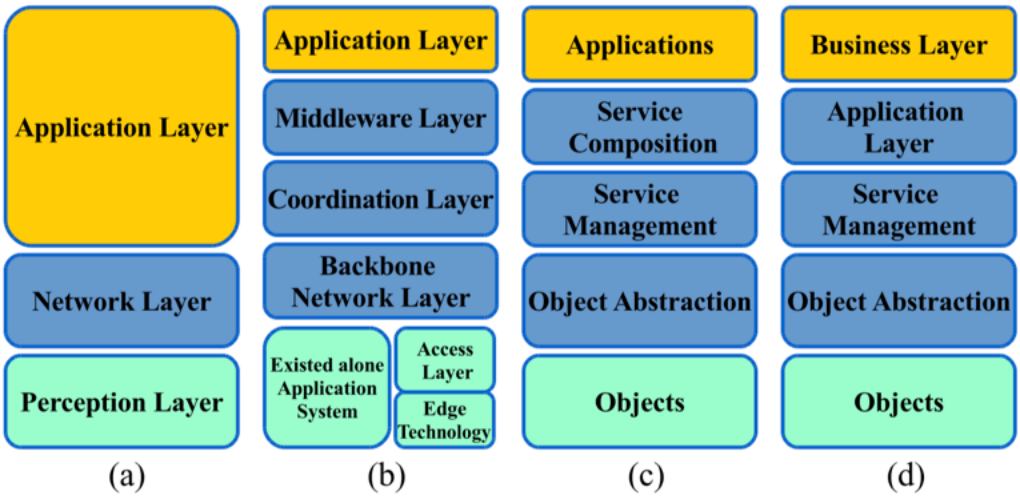


Figure 2.3: The IoT architecture. (a) 3-layer (b) Middleware. (c) SOA. (d) 5-layer [37].

The most famous architecture model uses the 5-layer architecture with different names of layers but maintains the same goals. The first layer contains objects or also considered the perception layer, which collects information about the physical devices with particular sensors. Physical devices collect data using specific sensors such as location, vibration, acceleration, humidity, temperature, and so on. After that, these devices transmit to the abstraction object layer using a secure layer for service management. This layer implements several technologies to information transfer such as Global System for Mobile communications (GSM), ZigBee, Bluetooth Low Energy, General Packet Radio Service (GPRS), infrared, Z-Wave, NFC, Wi-Fi, and Ethernet. Usually, routing protocols specify how objects communicate with each other.

Also, the service management or middleware layer is responsible for processing information and decision-making over services offered by the system. In this layer are defined interoperability services to data exchange between any platform. The application layer includes essential services to ensure the availability of content on the network and data management, such as Market-to-Market service, Quality of Service (QoS), facility management, geomatics, service-oriented architecture (SOA), cloud computing technologies, and forth. These services offer an excellent user interface to enterprise applications and end-users, such as logistics and supply, real-time monitoring, production management. By the end, the business layer has the responsibility to build a business model, graph based on analysis using the monitoring and management of the underlying four layers.

2.4 Standards IoT Protocols

Considering the future of Fieldbus protocols is IP (Internet Protocol) to explore and expand to many vertical domains, improving the interoperability with standard networking technology running over any physical layer. At this point, the Institute of Electrical and Electronics Engineers (IEEE) Committee 802 defines physical and data link technologies. Such technologies are Wi-Fi (802.11), Bluetooth (802.15.1) Zigbee, 6LowPAN (802.15.4), and Wireless Metropolitan Access Networks Broadband Wireless Access (BWA) WiMax (802.16) [38]. The 802.15.4 standards consist of a physical and link layer technology optimized for low bitrate, low duty cycle applications.

Follow the description of a set of application protocols used in IoT networks to data or message exchange between smart sensors. Each protocol has particular properties to use in specific scenarios and environments. Communication standards ensure data exchange between IoT devices on the same network sharing the same protocol [39].

Table 2.1 describes a brief description of most important protocols used in IoT applications. A comparison between these protocols is presented in [37].

Table 2.1: Traditional protocols of IoT.

OSI Layer	IoT Technology	Description
Application	AMQP	An open standard application layer protocol for the IoT application with the focus on message-oriented environments [37]. AMQP is a reliable transport protocol like TCP using appropriate queues to exchange messages. Also, AMQP supports the publish/subscribe communications model. It uses three qualities of service to message delivery guarantee such as i) at-most-once, ii) at-least-once and iii) exactly once delivery. Specification of messaging interoperability between heterogeneous platforms and message brokers. AMQP was created to provides a standard messaging API for the Java Platform.
	CoAP / CoAPs	A protocol or application layer for IoT applications defined by the IETF Constrained RESTful Environments working group. The CoAP protocol is based on representational state transfer for the simplicity of data exchange between the client-server model over HTTP. By default, COAP uses the UDP protocol, and it is suitable for IoT applications. Moreover, CoAP modified the HTTP primitives to minimize resource requirements to meet the IoT requirements [37]. The REST-CoAP proxy executes conversion between two protocols for smart devices with constraints of energy, and computational power utilizes the RESTful interactions using HTTP methods (i.e., get, post, put and delete).
	MQTT / MQTT-SN	A simple broker-based publish/subscribe messaging protocol designed in 2013 by Andy Stanford-Clark of IBM and Arlen Nipper of Arcom (now Eurotech) in 1999 and was standardized in 2013 at OASIS [40].

Table 2.1 – Continued from previous page

OSI Layer	IoT Technology	Description
	XMPP	A messaging protocol developed by the Jabber open-source community used for instant messaging standard of multimedia applications with chatting, voice, and video calling.
Session	DTLS	A security protocol derived from SSL protocol to ensure security properties like integrity, authentication, and confidentiality under UDP protocol.
Transport	UDP	A communication protocol used to the packets transport of network layer with a minimum amount of communication approach. It is the simplest protocol that uses IP services to provides a best-effort delivery mechanism.
Network	6LowPAN	A low-cost and low-power communication network which connects resource-constrained wireless devices [41].
	IPv6	The communication protocol of the Internet Protocol to provide identification for devices on networks. IPv6 was developed to have multiple addresses with 128 bits.
	RPL	A routing protocol for Low power and Lossy Networks (LLNs) with distance vector IPv6. RPL provides routing across multiple paths of link layers of the IP architecture using a function to ensure the best route using a set of metrics/constraints.
	IPSec	A suite of protocols and algorithms to make secure an untrusted network. The IPSec standards (RFC 2401) provide some security properties as confidentiality (encryption), integrity (hashing algorithm), authentication (pre-shared keys or a certificate authority), and anti-replay (unique sequencing number).

Table 2.1 – Continued from previous page

OSI Layer	IoT Technology	Description
Data link	802.15.4 MAC	The largest standard for low-rate wireless personal area networks (WPAN) with many categories like 802.15.4c (China), 802.15.4d (Japan), 802.15.4e (industrial applications), 802.15.4f (RFID), and 802.15.4g (smart grid). This standard defines the media access control (MAC) to ensure the format of the data handling.
	NFC	Near Field Communication (NFC) is a technology to connect objects automatically from the physical world to the virtual world. This technology provides the vision of a ubiquitous reality that is commonly associated with IoT [42].
	RFID	Radio Frequency Identification Technology (RFID) is a technology to identify objects using tags with a unique ID. It is composed of a tag reader to capture the energy and transfer the tag's ID using magnetic induction or electromagnetic wave. They take advantage of the RF antenna to sustain its sensing operation [43].
	ZigBee	Zigbee is a wireless technology created by the Zigbee Alliance for low-data-rate and short-range applications (in the personal operating space as well as 10m) to support simple devices that consume minimal power [44] [45]. Zigbee operates in four layers: the physical (PHY) layer, the medium access control (MAC) layer, the network (NWK) layer, and the application (APL) layer [41]. The support base of the Zigbee protocol operates on the initial version of IEE 802.15.4, which has three frequencies bands: 868 MHz, 915 MHz, and 2.4 GHz. ZigBee has three types of devices in his network: the coordinator, router, and end devices. Moreover, this technology provides some security functions across layers.

Table 2.1 – Continued from previous page

OSI Layer	IoT Technology	Description
	Z-Wave	Z-Wave is a protocol interoperable, wireless, RF-based communications technology created by ZenSys and promoted by the Z-Wave Alliance. This protocol has proposed to maintain reliable transmission of short messages between nodes from a control unit in the network. For that, it operates in five main layers: the PHY, MAC, transfer, routing, and application layers [41].
Physical	802.15.4 PHY	This technology is a short-range communication system designed to provide throughput and latency requirements for WPAN. It operates in three unlicensed bands: 868MHz, 915MHz, and 2.4 GHz. According to the geographical area where the system was developed, the transmission range is suitable.
	Wavenis	Wavenis is a wireless protocol designed by Coronis Systems. It operates with PHY, MAC, and network layers in 433MHz, 868 MHz and, 915 MHz bands [41]. An application programming interface (API) provides access to the services available. This protocol only defines a type of device that when to join in a Wavenis network to find an adequate parent.

These are some solutions to allow reliable communication between devices in an environment of the IoT. Each technology has its characteristics that are better than others, but network administrators need to be aware of that to choose suitable for their network work well.

2.5 Application Domains for Internet of Things

The emerging paradigm of the Internet of Things aims to connect several physical things exploring unique addressing to identify each other, and they represent virtual elements from a dynamic worldwide network. Smart devices interconnection allows for creating innovative applications in different domains, such as the medical domain, to sensing medical parameters from a specific patient or a natural phenomenon. According to Borgia [4], IoT applications have categories of three major domains (Figure 2.4): i) industrial domain, ii) smart city domain, and iii) health well-being domain. Therefore, each domain

specializes according some applications are more suitable for the user that which can improve our daily lives.

Figure 2.4: IoT Application Domains and Related Applications (Adapted from [4]).

The applications of IoT in the Industrial domain involve several activities around commercial and financial transactions between companies, organizations, and other entities. More specifically, applications in this context are logistics, agriculture and breeding, and industrial processing [46]. Logistics is a domain of IoT application that offers data monitoring and management from different contexts and, usually, it explores the identification and registration of entities of the system to create an intelligent and efficient use of sensing and processing of IoT technologies.

The application domain of logistics involves simplify and help to manage products and materials efficiently in supply chain management. It represents the improvement to the packaging, transportation, warehousing, and retails. For that, RFID is a vital technology tracking and identifying products to control them through radio waves. An example of the use of this technology is in military logistical support to collect the information of combat units, material storage units, weapons, and equipment status and send it to a processing center by the military network [47]. Combining technologies as RFID for IoT is possible to improve the efficiency and effectiveness of logistic support because the capabilities of sensing, transmission, and processing are essential to control and synchronize the real-time analysis, which can ensure real-time interaction and intelligence sharing when commanders make the decision.

Another application of IoT in the industrial domain is in agriculture and breeding. IoT has many capabilities to identify animals and continuous monitoring of anomalous

behavior, e.g., diseases [48]. The control of animals allows us to store information about the animals, and it can be checked by veterinaries, for example. Also, IoT can offer solutions to the industrial processes with monitoring of industrial plants to identify the specific geographical area, and sensors may manage different kinds of plants, e.g., oil plants, gas plants. In dangerous situations, automatically alarms are sent to monitoring centers.

2.6 IoT Platforms

IoT platforms have been proposed to simplify some developing challenges such as high complexity of distributed computing, unknown guidelines to simplify high-level implementation, multiple programming languages, use of distinct communication protocols. All these challenges require many efforts to manage infrastructure and handle both software and hardware layers to attend the software requirements.

Some IoT platforms have been proposed to support developments, deployment, and maintain IoT applications. Each platform explores IoT requirements with particular features as follows the description of each approach. A collection of IoT platforms includes AWS IoT from Amazon, ARM Bed from ARM and other partners, Azure IoT Suite from Microsoft, Calvin from Ericsson, HomeKit from Apple, Kura from Eclipse, and SmartThings from Samsung.

2.6.1 AWS IoT

Amazon Web Services (AWS) IoT [49] provides multi-layered security, bi-directional communication between Internet-connected devices such as sensors, actuators, embedded micro-controllers, or smart appliances and the AWS Cloud to build IoT specific services. Among them, AWS IoT Core is an IoT cloud platform available to let connect smart devices easily and securely interact with the cloud applications and other connected devices. This platform supports HTTP, Web Sockets, and MQTT, a lightweight communication protocol, other industry-standard and custom protocols. These protocols are specifically designed to tolerate intermittent connections, minimize the code footprint on devices, and reduce network bandwidth requirements. In addition, devices can communicate with each other even if they are using different protocols.

The AWS IoT architecture consists of four major components: Device Gateway, Message Broker, Rule Engine, Registry, Device Shadows, and Security and Identity service.

The Device Gateway is responsible for managing all device connections as an entry point for IoT devices connecting to AWS, ensuring multiple protocols to devices securely and efficiently communicate with AWS IoT Core. For device connections, the Device Gateway ensures low latency to the transmission of messages at any time. The Device Gateway is fully managed and scales automatically to support over a billion devices

without requiring you to manage any infrastructure. Amazon uses the MQTT protocol [50] due to several features described in Section 2.4. However, one feature essential to the platform that sensors and other embedded devices are moving and talking to the Device Gateway does not need to know who is sending data to them. This feature enables a scalable environment for low-latency, low-overhead, and bi-directional communication.

The Message Broker is a publish/subscribe broker service that manages the transmission of messages to and from all IoT devices and applications. This component sends the message to all clients that have registered or subscribe to receive messages for a specific topic.

The Rule Engine uses an SQL-based language from incoming published messages and then processes and sends the data to other subscribed devices or AWS cloud services, such as Amazon S3, Amazon DynamoDB, and AWS Lambda. Rules act according to the content of each incoming message. This component allows us to build IoT applications that collect, process, analyze, and act on data generated by connected devices globally.

The Registry component consists of the identification of each device regardless of the device type, vendor, or the way of connection. This component stores the metadata that describes the capabilities of each device to have the capability of tracking them. This information will be maintained in the Registry for a period of 7 years.

The Device Shadow consists of a virtual image of each connected device in a persistent and stored in the cloud to be available and accessible all the time. The component stores the last reported state and desired state of each device, which makes it easier to choose the desired future state of a device without accounting for the devices' current state by interacting with Device Shadows via REST API or by using the Rules Engine. This means that Device Shadow improves the application development providing a uniform and available interface of devices, even though IoT devices use heterogeneous communication and security protocols and constrained by intermittent connectivity, limited bandwidth, limited computing ability, or limited power.

The Security and Identity service provides responsibility for cybersecurity in the AWS cloud, which keeps their credentials safe to send data to the message broker securely. Using mutual authentication provided by a Certificate Authority (CA), generated and signed certificates are used in all connections of the platform. Each certificate can use distinct policies to authorize devices or applications to have access or revoke access without ever touching the device. All traffic of AWS IoT must be encrypted over Transport Layer Security (TLS), and credentials are authenticated with an X.509 certificate. The AWS IoT platform has the Device SDK that provides facilities to connect hardware devices or mobile applications to AWS IoT Core, which uses the MQTT, HTTP, or WebSockets protocols. The AWS IoT Device SDK supports developing languages C, JavaScript, and Arduino, and includes the client libraries.

The Security and Identity service of the AWS IoT platform offers many security measures to apply at every level of the technology stack. This service provides authentication at all IoT devices to confirm that the source of the transmitted data is

always known. The platform acts as a Certification Authority (CA) to digital certificates such as X.509. It verifies the certificates over SSL/TLS protocols to ensure secure authentication and check validity against a registry of certificates. The authentication provided by the AWS IoT platform is responsible for applying for access permissions by the user on his devices using access control policies.

These policies ensure the mapping of devices or applications specified in these rules can have access to the particular device, which the certificate belongs to. The rule engine has capabilities to deal with an access management system to offer secure access to objects according to the predefined rules and policies. The communication security is achieved with the encryption of all traffic over SSL/TLS protocols. These protocols ensure the confidentiality of the application protocols (e.g., MQTT and HTTP) supported by the AWS IoT platform. Several cipher suites are supported in the platform as well as *Forward Secrecy*, a property of secure communication protocols, in which it continually generates new keys and does not compromise the communication.

This IoT platform offers a fully managed service, called IoT Device Defender to continuously audit the configurations of IoT devices to ensure that devices are not deviating from security best practices and mitigate security risks. These configurations involves ensuring device identity, authenticating and authorizing devices, and encrypting device data. The platform adopts consistent security policies across IoT devices and responds quickly when devices are compromised.

AWS IoT uses a security profile with a set of behaviors such as each behavior contains a metric that specifies the normal behavior for all devices. These metrics are checked using some criteria in the definition of a behavior. Some metrics can be the number of bytes in a message, the number of messages received or sent by a device during a given time period, the number of authorization failures during a given time period, the IP address from which a device has connected to AWS IoT, and so on. The application of this metrics associated to security profiles provides the impact analysis of threats from public or private way addressed to IoT devices.

2.6.2 ARM Pelion IoT

ARM mbed IoT is a fully integrated platform to develop applications for IoT based on ARM microcontrollers [51]. It provides a device management solution composed by the operating system, gateway, device management services, and partner ecosystem to deploy of IoT solutions. Due to his operating system, this platform has the advantage over several majorities of platforms, which supports communication protocols for connecting devices with cloud computing. The architecture of this platform has device, data and connectivity management services (Figure 2.5).

The mbed OS is an open-source operating system designed for ARM Context-M microcontrollers, containing a core, security, and key IoT networking and communication technologies. The mbed OS allows developers to focus on application code, not

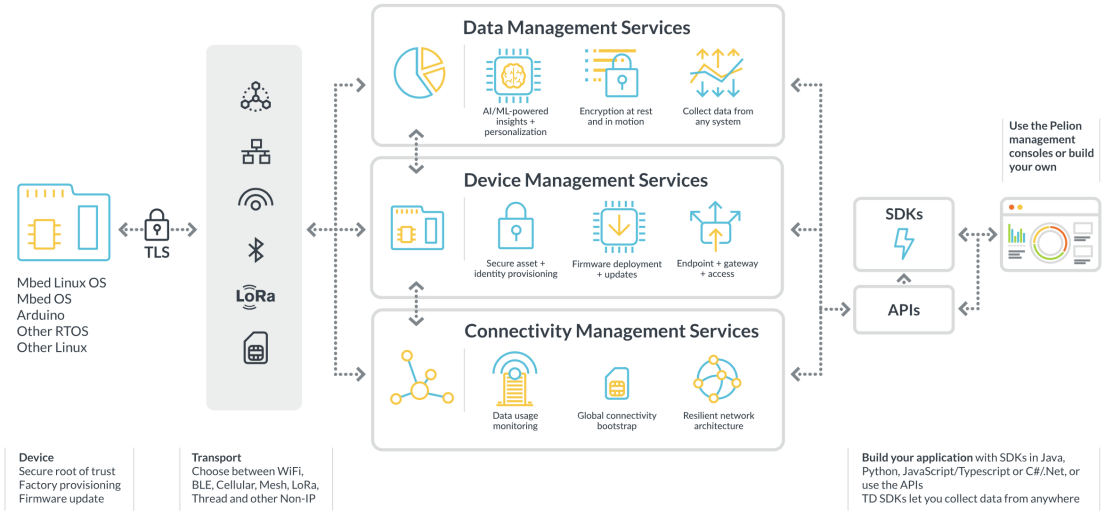


Figure 2.5: Components of the Arm Pelion IoT platform [51].

underlying complexity. It will enable us to develop IoT applications with connectivity for constrained devices using a wide variety of communication protocols.

The mbed OS makes the fast device development using the high-level C++ API to the full control of endpoint and application logic, which implements a communication stack with several transport technologies such as BLE, Cellular, Mesh, LoRA, Thread, and Non-IP [51].

The Connectivity Management Services offer monitoring and global connectivity bootstrap to efficiently connect IoT devices to the cloud, which is fully compatible with the mbed OS. Also, the platform offers provisioning devices for managing IoT ecosystem to the Internet to provide all security measures for the connection with devices IoT. Then, developers have many possibilities to implement web and IoT applications to manage cloud-connected IoT devices via REST API.

In terms of cybersecurity, the Pelion IoT platform provides a wide variety of cryptography standards, fundamental exchange mechanisms, certificate-based signatures, and symmetric and public/private key encryptions [51]. The platform provides end-to-end secure communication using TLS protocol, including cryptographic and SSL/TLS capabilities (also the related Datagram TLS) in their embedded products with a minimal code footprint.

2.6.3 Azure IoT Suite

Amazon launched the Azure IoT Suite as a platform for an IoT solution in the cloud that allows end-users to interact with IoT devices, to monitor your devices to analyze and visualize data from business outcomes by automating processes [52]. It addresses some enterprise-grade collection of preconfigured solutions built on Azure Platform-as-a-Service (PaaS) that enable an acceleration of the development of custom IoT applications.

The architecture of the Azure IoT suite is composed of some key elements of a typical IoT solution architecture (see Figure 2.6. In this platform, the IoT devices collect data and sent it to the cloud gateway. This one is responsible for making the data available for other back-end services such as Azure cloud services (e.g., Azure Machine Learning and Azure Stream Analytics). The output of processing is visualized in a customized way that meets the requirements of customers and suites their business.

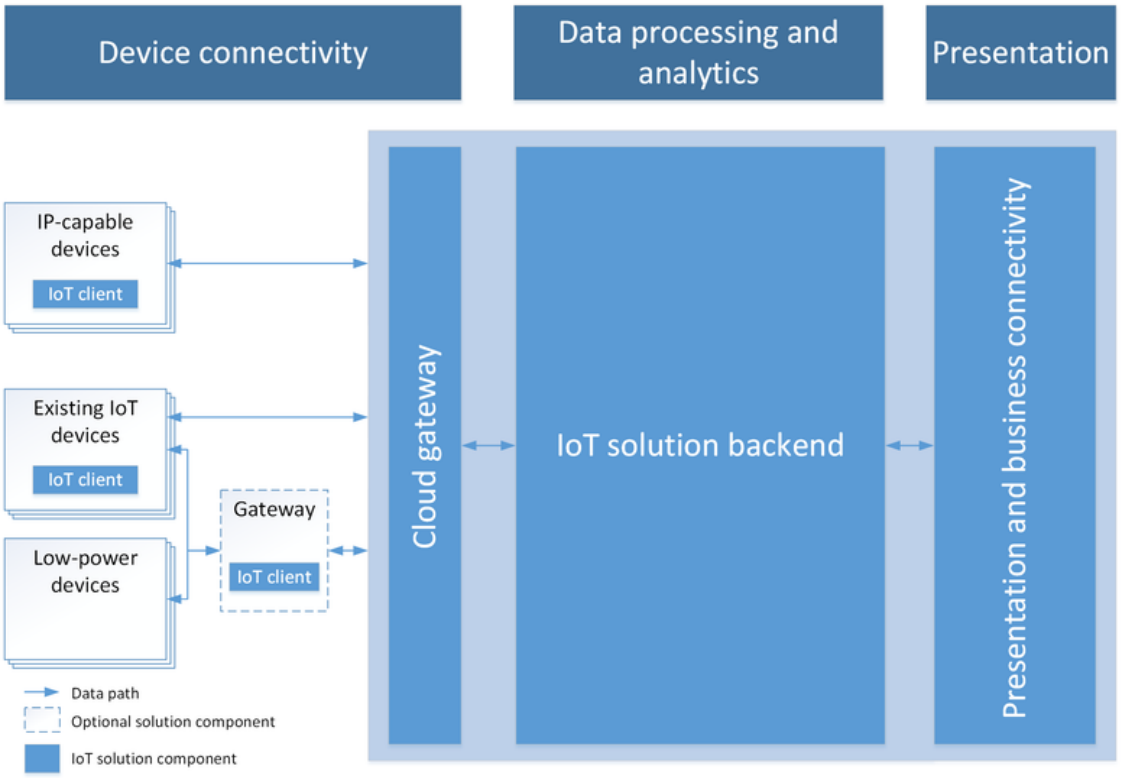


Figure 2.6: Azure IoT architecture [52].

The Azure IoT Hub is a fully managed service that enables reliable and secure bidirectional communications between millions of IoT devices and a solution back end [52]. It provides multiple communication options between device and cloud computing. The platform offers the storage, synchronization, and querying metadata and state information at a rich set of device-to-cloud and cloud-to-device communication options. The Azure IoT Hub provides authentication and secure connectivity with the identity registry of each device using per-device security keys or X.509 certificates. The IoT Hub natively supports communication over AMQPs, MQTT, or HTTP protocols [52]. Also, the device provisioning enables the right IoT Hub without human intervention of devices in a secure and scalable manner.

The SSL/TLS protocol is used to establish a secure connection between IoT devices and Azure IoT Hub, which enables encrypt of the handshaking process. The platform provides an authorization model to deal with resource access, management, and auditing using a policy-based approach. The IoT Hub uses a set of access control rules to ensure

access control policies to either IoT devices or smart applications.

This IoT platform takes advantage of threat monitoring solution to collect data from others organizations based on cloud-scale machine learning and behavioural analytics for Microsoft researches to find new threats and proactively create and implement security fixes before your IoT deployment is affected [52]. The Azure IoT uses artificial intelligence to make the threat detection and response smarter and faster. This components is provided by a cloud-native SIEM Azure Sentinel.

2.6.4 Calvin

Calvin is an open-source IoT platform to help developers develop smart applications and a runtime environment for handling the running application [53]. This platform is a hybrid framework combining ideas from the Actor model and Flow Based Computing [54]. As it was designed to simplify the development of IoT applications, Calvin makes simple and fun the development of applications as well as offering a unique form manner to the platform-dependent runtime layer with all kinds of communications between distinct IoT devices are supported. Also, this layer supports several transport layer protocols (WiFi, BT, i2c).

The development of IoT applications follows four steps, and each step has its functionality [54], such as i) *Describe* consists of functional parts of applications, including reusable components. Each component is an actor such as a device, a service, for instance. Actors can communicate with each other using tokens over ports. The description of the actions composes an actor, their input /output relations, the conditions for a particular action to be triggered, and the priority order between actions; ii) *Connect* is the process to form a directed graph by connecting the ports of some actor. It uses a declarative language to describe applications and how actors connect inside them. Then, it merely follows data on its ports, unconcerned with anything beyond them; iii) *Deploy* consists of the execution of the application, in fact. The platform provides many accessible nodes for deployment and actors executions. This distributed execution environment is flexible and can move actors to any available note considering factors such as resource, locality, connectivity, or performance; iv) *Manage* monitors the life cycle of the application. For instance, resource usage, firmware updates, migration of running actors, error recovery, and scalability.

The Calvin IoT platform authenticates its users using three different ways: local machine, external machine, and using a RADIUS server. However, only using the local and external machine is possible to proceed authorization such as using JSON files in the same machine, whereas the external device involves digital certificates in the form of X.509 standards to check JSON web tokens with the authorization request/response. After the authentication and authorization process, IoT devices connects to the M2M gateways using short-range radio protocols to deliver its data. M2M gateways use their identity to authenticate to the cloud servers through SIM cards and 3GPP standardized

Generic Bootstrapping Architecture. The IoT platform transmits data using TLS/DTLS protocol and Elliptic Curve Cryptographic (ECC) for encrypting communications and providing digital signatures.

2.6.5 Eclipse Kura

The Eclipse Kura is an Eclipse IoT project for the development of IoT gateways with remote management using a wide range of APIs to deploy IoT applications [55]. This framework is an extensible open-source IoT Edge Framework based on Java Virtual Machine (JVM)/OSGi and offers a device abstraction layer for accessing the gateway's hardware interfaces. Its objective is to provide an intelligent gateway between the private device network and the local network, public internet, or cellular network for running applications with sensing and actuating in IoT environments.

The Kura architecture provides a Java-based platform to implement applications within an OSGi container working with a standard framework for handling events, packaging code, and range of standard services. Such services are watchdog, clock, GPS position, embedded database, process, and device profile service. The network management layer provides OSGi services to access current network configurations, including WiFi access points and PPP connections. Also, the Kura provides a store-and-forward repository service to retain data temporarily on the locally or network-attached devices, sending that data using MQTT brokers and cloud services.

The security aspects from Kura involves secure sockets provided by the Java Runtime environment and the use of public-key cryptography authentication via MQTT protocol. Also, all communications are protected using digital certificates based on the SSL/TLS protocol. The network configuration component of the platform offers options to use firewall, port forwarding and network monitors.

2.6.6 SmartThings

SmartThings is a development platform released by Samsung [56] to build and integrate IoT devices, services, and existing solutions for IoT applications. All developers can create, integrate, and publish applications to connect devices, actions, and external services to create automation into the IoT ecosystem using SmartThings. The SmartThings ecosystem includes some components such as the hub/home controller, the cloud back-end, the mobile client app, and the IoT device (Figure 2.7). The hub/home controller connects all IoT devices to the cloud services using several communications protocols such as ZigBee, Z-Wave, WiFi, and BLE. The mobile client app offers an easy-access way to manage their IoT devices with support to multiple mobile operating systems such as Android and iOS. A new voice assistant (called Bixby) supports the interaction with devices on cloud back-end using capabilities to handle device control, monitoring, scheduling, location-based automation, and mode execution.

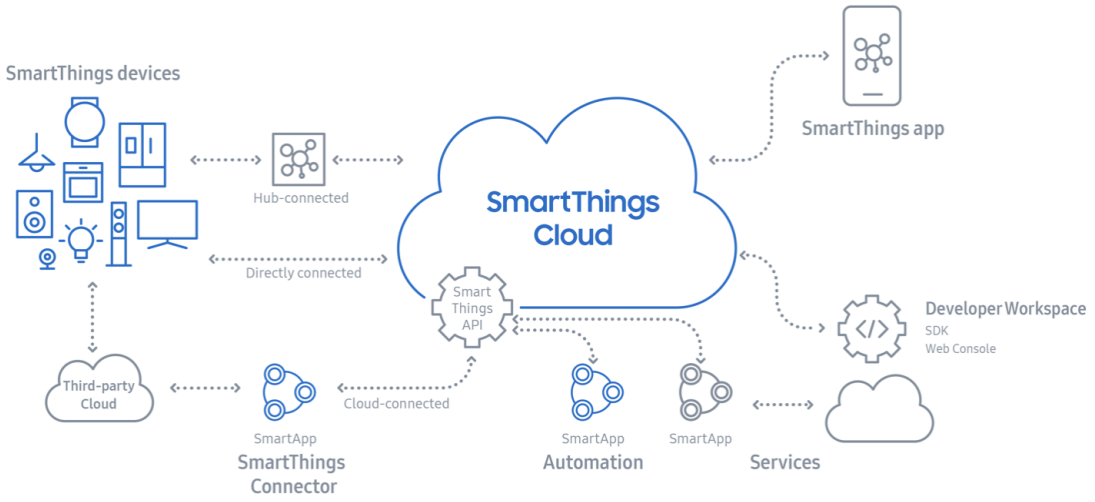


Figure 2.7: The structure of the SmartThings ecosystem [57].

Also, this app supports the installation of SmartApps as well as third party applications with the customization and new functionalities developed from third-party developers. The SmartApps has three distinct classes of applications. The first on is the device type handlers to subscribe for events and call handler methods. The Solutions Modules simplifies the management of a specific physical area in the environment and, lastly, the Service Managers support the discovering of IoT devices (SmartDevices) on the network through WiFi/IP protocol.

The security aspects from SmartThings involve the use of the OAuth/OAuth2 protocol to authenticate IoT devices and to authorize access to its capabilities. The communications between components of the SmartThings are performed using SSL/TLS protocols and encrypted using 128-bit AES.

2.6.7 Summary of Cybersecurity on IoT Platforms

A more in-depth review of a subset of commercially IoT platforms to support development of IoT applications presented some security challenges such as limited utilization of mutual authentication between the involved parties by each IoT platform, lack of security mechanisms to avoid exposures of users to significant cybersecurity risks, encryption techniques with requirements of the higher computing power.

The description of the IoT platform sections demonstrated distinct approaches to implement applications supporting IoT technologies. Most of the significant challenges are not present in these approaches when used with the IoT. There is a dependency on the commercial of the shelf microcontrollers, and most of them are used without considering security capabilities to protect sensitive information from users.

These platforms reviewed perform the protection of the data integrity and devices communications using cryptographic technologies such as SSL/TLS protocols. It ensures that any communication channel stay vulnerable to malicious stakeholders in IoT systems.

Also, the access control of users and authorization mechanisms are adopted to control which devices each users can to collect data in the environment. For that, Azure and AWS IoT platforms use its own policy-based services to manage a set of access control rules to grant or deny permissions to either IoT devices or smart apps.

However, only AWS, Pelion and Azure IoT platforms adopted device monitoring solutions to identify abnormal behavior from IoT devices. These platforms explores the components to develop IoT applications and offer the monitoring support after the deployment process keeping devices and data secure using digital certificates. Azure IoT platform explores the intelligence artificial to minimise security weaknesses at each stage of IoT deployments and provide end-to-end security and many threat monitoring solutions for all IoT devices.

2.7 Discussion

This chapter provides the basis to understand the Internet of Things concept. The thing life cycle was identified to understand each vendor's entire process for things in the IoT ecosystem. Each thing is a critical piece of the system model of IoT. The association of a network of the various devices creates an ecosystem with distinct communication models. It involves using one specific IoT architecture to support the IoT ecosystem's emerging needs, such as data collecting to business models based on data management. There are several standard IoT protocols used for these domains' specific applications, considering some characteristics of the scenarios and environments. Finally, to simplify this ecosystem's developing challenges, some IoT platforms are available to support developments, deployments, and maintaining IoT applications. Most of them apply for data protection with cryptographic technologies, authentication mechanisms to support access control from users, and authorization systems to control devices' permissions in the network.

CHAPTER 3

Cybersecurity in the Internet of Things

This chapter presents the relevant concepts concerning cybersecurity to accomplish a high level of device protection in the Internet of Things environments. The main objective is to analyse significant components that involve cybersecurity features for IoT applications. Firstly, cybersecurity in this context is directly linked to the protection of communication channels to data exchange among smart devices in IoT networks. Secondly, privacy is an essential requirement when we talk about cybersecurity in IoT, because many personal data are transmitted without encryption in public networks. Finally, the development of new products for IoT does not provide systematic mechanisms concerning security tools to protect personal information.

Cybersecurity is an area of computer science that explores ways to protect the information by limiting the impact of the risks. Cybersecurity represents the process to ensure the security of software, hardware, or environment [58] using specific types of products but is not something that can be bought off the shelf [59]. Moreover, the process of securing cannot be considered secure or non-secure. [60] defines cybersecurity as "*the protection of information and its critical elements, including the systems and hardware that use, store, and transmit that information*".

There is a substantial overlap between information security and cybersecurity, but they have no completely the same meaning. Information security is more restrict to traditional cybersecurity because it includes the protection of resources, assets, and, usually, referencing roles of the human factor in the security process. For cybersecurity, this factor has an additional dimension, namely, the humans as potential targets of cyber-attacks or even unknowingly participating in a cyber attack [59].

The International Telecommunications Union (ITU) defines cybersecurity as follows:

> "*Cybersecurity is the collection of tools, policies, security concepts, security safeguards, guidelines, risk management approaches, actions, training, best*

> *practices, assurance and technologies that can be used to protect the cyber environment and organization and user's assets. Organization and user's assets include connected computing devices, personnel, infrastructure, applications, services, telecommunications systems, and the totality of transmitted and/or stored information in the cyber environment. Cybersecurity strives to ensure the attainment and maintenance of the security properties of the organization and user's assets against relevant security risks in the cyber environment"* (ITU).

Cybersecurity is a critical requirement in IoT systems. A compromised communication channel can not endanger the privacy of users and could adversely affect the user from physical environments. IoT systems have characteristics to continuously change over an area without well-defined perimeters due to device and user mobility. A malicious user with access to the communication channel can alter the functionality of a device from a given manufacturer and leak information that is very valuable for the users.

IoT cybersecurity requires a wide range of procedures because it has an impact on attacking a system using several devices to compromise services using a Distributed Denial of Service (DDoS). These cases are common in IoT ecosystems because they rely on standard IP protocols to provide easier system integration, but this also makes threats on full applicable in other environments. However, new requirements are identified as essential to implement security.

3.1 Cybersecurity Requirements

Security management involves several mechanisms and tools for the preservation of security properties confidentiality, integrity, and availability. The NIST standard FIPS 199 lists these three security objectives for information systems. Also, the FIPS PUB 199 provides a useful characterization of these three objectives regarding requirements and definition of a loss of security in each category.

Also, other properties such as authenticity, accountability, non-repudiation, and reliability can be involved. Privacy is considered highly relevant to security, as well as other security attributes that have positive or negative influences on privacy [61]. Security mechanisms focus on the provision of protection mechanisms that include different types of protection such as authentication, access control policy, availability, confidentiality, integrity, intrusion detection, retention, storage, backup, incident response, and recovery.

Availability is the term used to assure that the systems responsible for delivering, storing, and processing information are accessible when needed and by those who need them [62]. Nevertheless, availability is a property of being accessible and usable upon demand by an authorized entity. It occurs when they can not prevent legitimate users from having reasonable access to their systems. For this, there is a need to prevent

authorized access to resources or the delaying of time-critical operations, i.e., Denial of Service [63].

Confidentiality is the term used to avoid the disclosure of information to unauthorized individuals or systems [62]. Confidentiality has the main objective of computer security, which is to stop unauthorized users from reading sensitive information. Moreover, unauthorized users should not access confidential information. Hence, privacy terms and secrecy are sometimes used to distinguish between the protection of personal data and organization data, respectively [63].

Furthermore, other security requirements are necessary for IoT systems like authentication, authorization, accountability, auditing, non-repudiation, and privacy goals. These requirements are accomplished based on security components to protect the organization's resources [64]:

- A risk analysis will identify these assets, discover the threats that put them at risk, and estimate the possible damage and potential loss a company could endure if any of these threats become real. The results of the risk analysis help management construct a budget with the necessary funds to protect the recognized assets from their identified threats and develop applicable security policies that provide direction for security activities. Security education takes this information to every employee.

- Security policies and procedures to create, implement, and enforce security issues that may include people and technology.

- Standards and guidelines to find ways, including automated solutions for creating, updating, and tracking compliance of security policies across the organization.

- Information classification to manage the search, identification, and reduction of system vulnerabilities by establishing security configurations.

- Security monitoring to prevent and detect intrusions, consolidate event logs for the future log, and trend analysis, manage security events in real-time, manage parameter security including multiple firewall reporting systems and analyze security events enterprise-wide.

- Security education to bring security awareness to every employee of the organization and teach them their security responsibility.

3.2 Cybersecurity Challenges

The IoT ecosystem is embedded with dedicated IoT devices, which are mostly integrated with control, monitoring, and communication capabilities. In contrast, these devices have sensing and control functionalities from distinct devices and require particular purposes

of scalability and maintenance. Moreover, IoT core components are usually commercial off-the-shelf products from diverse vendors having unknown incompatibilities.

Cybersecurity challenges are the most critical aspects of the complete adoption of the Internet of Things in the real world [13]. The multiple interfaces from the connections between objects and services in IoT systems become even more complex this adoption because system weaknesses are exploited to malicious users. The high number of devices and software vulnerabilities and inherent characteristics from the IoT systems are furthermore complicated when multiple heterogeneous devices exchange information with each other. These aforementioned aspects are some cybersecurity views from the adoption process of IoT systems must overcome to achieve the reality.

The research community [65, 66, 67] enumerated several cybersecurity challenges and how existing security protocols deal with these operational and technical aspects. Many initiatives that can cope with security protocols and mechanisms to understand which are limitations from distinct areas and their applications and environments. The cybersecurity requirements have the principal role in enabling the users to trust their IoT environment. Therefore, a high degree of reliability in the IoT network enables an increase in connecting smart devices to the Internet, resulting in the growth of new opportunities for business and market in IoT.

3.2.1 Resource Constraints and Heterogeneous Technologies

IoT ecosystems are composed of several devices with severe resource constraints hindering the use of Internet standards for communication and service provisioning. The main characteristics of these devices are related to CPU, memory, and energy budget available. It directly impacts the design of communication protocols for the IoT applications such as hop-by-hop fragmentation and reassembly due to the small packet size limits at the physical layer resulting in the excessive fragmentation of large packets that are required by security protocols and becoming vulnerable for state exhaustion attacks.

Security protocols have limited usage for devices with small CPUs and scarce memory limits such as public-key cryptography. There are ongoing strategies to reduce the resource consumption in most Internet security standards by using more efficient underlying cryptographic primitives such as some examples of the Elliptic Curve Cryptography (ECC) [68]: elliptic curve X25519, stream ciphers such as ChaCha, Diet HIP, and ECC groups for IKEv2. They typically admit a reduced implementation that uses few resources and inexpensive operations, and they have also been designed to minimize leakage of information through side-channels, which makes them suitable for a wide range of architectures.

Security protocols for constrained IoT ecosystems are not precisely identical to their Internet standards, and it requires some adaptations to ensure end-to-end security [66]. Protocol translators at middleboxes reduce the performance gap between Internet

protocols and the IoT, but they become major obstacles if end-to-end security measures are used. Some solutions are founded to enable IoT specific optimizations, but it requires application-specific transformations before security is applied [65].

3.2.2 Bootstrapping of a Security Domain

Bootstrapping is a phase of the thing lifecycle, and it also is applied within a security domain. This phase addresses the network operation, communication aspects, and the privacy implications for applying continuous operational capabilities. The network operation can be distributed or centralized way to form a security domain. In a centralized procedure, the nodes use a dedicated node for the security capabilities between things in the IoT environment, such as the central management of devices and cryptographic keys. On the other hand, the decentralized procedure allows to create a security domain without a central node to provide supervision, and they operate autonomously.

The network operations associate a device to another one, to a network, or a system. It relies on to bootstrapping a thing's identity, which involves mutual object authentication with identity management and key-management infrastructures that are important even more when working together with cryptography [66]. For example, shared secret keys. The object identity principles: i) to distinguish object's identity and its mechanisms; ii) to separate object's identity and several temporary identities according to its role; iii) to identify an object using its identity or its specific features; iv) objects know the identity of their owners.

Identity management addresses even deeper privacy concerns because anyone might access and track tags in IoT environments. IoT technologies and protocols should be designed to avoid theses expositions during bootstrapping and operation, such as the adoption of authorization and authentication to manage the identities of the domain of IoT security. Authorisation is a front end to manage access control policies and to perform access decisions. This security objective is the most critical to privacy protection mechanisms because it has the choice to make access to a resource. Authentication provides the authentication of users and services. It verifies the credentials are valid and returns an assertion as the outcome, which is required to use the IoT network [69]. These two functionalities are designed to authenticate a user based on provided credentials and to verify whether an assertion given by a user is valid or invalid.

Some IoT protocols offer the option to authenticate the responding host, such as TLS and DTLS. While hosts are unauthenticated, they can stay anonymous. However, the opportunity to authenticate hosts can be established to apply public-key certificates and allows to reveal of the responder's identity to possible malicious users. Also, HIP and IKEv2 protocols use public-key identities to provide authentication from the initial connection from the environment, and it can offer additional protection of the information in an encrypted packet.

3.2.3 Operational Challenges

After the bootstrapping phase, IoT devices start communicating between them to exchange information securely and in an authenticated manner. Several challenges in IoT can be grouped in devices, communication, and application vulnerabilities to allow malicious users to enter into the IoT infrastructure from unauthorized devices (endpoints). The protection of an IoT ecosystem involves protecting all communication channels using the end-to-end security with confidentiality and integrity properties assured. Some protocols provide secure communication associated with end-to-end security to protect the gap between the IoT network and the Internet. Some of them, as IKEv2, HIP, TLS, and DTLS, also include authentication services, end-to-end encryption, and integrity protection.

The operational phase requires solutions for secure group communication over local broadcast and multicast using symmetric group keys. The centralized-based solutions use a coordinator entity to distribute symmetric keys over end-to-end secure channels as an example of MIKEY architecture. It supports the possibility of establishing keys and parameters for more than one security protocol (or for several instances of the same security protocol) at the same time [70].

3.2.4 Verifying Device Behaviour

Understand how the device behavior affects the IoT ecosystem is important to maximize privacy and minimize leakage. IoT devices have different behaviors in realistic deployment scenarios because they use particular protocols, data volumes, data rates, and transmission frequencies. Each one affects the network behavior also and makes it challenging to assess the costs of bandwidth allocation and specific cybersecurity strategies and security mechanisms to protect their privacy and security threats [71].

3.2.5 Privacy Protection

Privacy concerns are one of the cybersecurity challenges to identify access control issues of the information as well as what information is shared when need to protect it and to whom access should be granted/restricted [72]. One viable solution is privacy by design to provide tools to manage personal data [66]. Moreover, privacy involves data transparency in services provisioning to identify which entities are managing their data. Technically, cryptographic mechanisms protect data, but it does not a complete solution because several resources might lack to manage such mechanisms. On the other hand, interoperability between security policies needs to be applied to a comprehensive interpretation, translating and reconciling a series of rules using different languages.

3.2.6 Data Leakage

Unauthorized entities compromise business credibility because they need to access, share, and use information, which leads to the inevitable release of confidential data. An accessible report of data leakage involved sensitive government information of the United States diplomatic cables by WikiLeaks [73]. The disclosure consisted of about 250,000 United States diplomatic cables and 400,000 military reports referred to as 'war logs'.

3.2.7 Trustworthy IoT Operation

The operational phase of thing lifecycle requires a level of confidence while interacting in the IoT network. The trust management consists of collecting user information about his reputation and calculating trust levels. It helps people overcome perceptions of uncertainty and risk and engages in user acceptance and consumption on IoT services and applications [74]. Two dimensions can be considered in IoT: trust in the interactions between entities and confidence in the system from the users' perspective.

3.3 Vulnerabilities Management

Vulnerabilities are flaws in a system or weakness of or an absence of security procedures and controls that allow an attacker to access unauthorized information, execute instructions and disclosure-sensitive information [75]. In IoT systems, software vulnerabilities are identified as weaknesses because at design moment programmers do not worry about security issues, resulting in security problems on operating systems, application software, control software like communication protocols and devices drives [12]. According to Kizza [64], these problems happen due to human weaknesses, and results of not understanding the requirements comprise starting the project without a plan, poor communication between developers and users, a lack of resources, skills, and knowledge, and failing to manage and control the system.

The rapid popularity of wireless communication has brought severe security problems of the interaction between devices. Internet technology has been vulnerable over the years due to the technologies are used by many who are not security experts [64]. Usually, many vulnerabilities are visible and available to observe but are not reported because those who use that do not know to classify as a vulnerability. Even though everyday hackers report several vulnerabilities, but it is not enough to solve the security problem. Some repositories become available several ways to improve security reporting software and hardware weaknesses like National Vulnerability Database (NVD)[1] and Common Weaknesses Enumeration (CWE)[2], and Common Vulnerabilities and Exposures (CVE)[3].

[1] https://nvd.nist.gov
[2] http://cwe.mitre.org
[3] https://cve.mitre.org

3.4 Threats to IoT Systems

Over the years, academic researchers and security professionals have identified different types of threats that affect specifics devices. Systems and applications had been designed without any security issue which becomes a relevant point of attack to the network. These threats aim to explore the system's weaknesses or vulnerable devices that have limited memory power limitation devices like sensors in the context of IoT [76] and critical infrastructure environments as well as Industrial Control Systems [77].

The attacking nature classifies two attack types: active and passive. The active attack compromises the IoT ecosystem introducing data into the system and changing data within the system. Some examples of this attack are brute force, buffer overflow, SQL injection, and so on. Passive attacks gather information for active attacks. Usually, this type of attack collects data from user records and poses a privacy threat. Some examples of this attack are Denial of Service (DoS), man-in-the-middle, session replay, and so on.

Threats of privacy issues are extensive and have many definitions and perspectives [78]. The privacy definition considers the human rights that a person has the responsibility to maintain, control, and possess personal information that defines them. In this work, we adopt a definition that combines all core privacy concepts important for us:

> *"Privacy is the faculty and right that a person has to define, preserve and control the boundaries that limit the extent to which the rest of society can interact with or intrude upon. At the same time, he or she retains full control over information generated by, and related to, him or her."* [79]

Privacy in the IoT involves i) privacy risks imposed by smart things services; ii) control over the collection of personal information by the smart things, and iii) dissemination of personal data by the entities.

3.4.1 Attack Methods

The attack method is the technique used to affect a vulnerability available in the system, and it can surpass user weaknesses or system configurations to breach a system. In this dissertation, the four most common attack methods are described to understand malicious user's motivations to implement proper security mechanisms.

- Social engineering: a technique used to delude users from the IoT ecosystem against the disclosure of personal information. Malicious users focus on social interactions to convince an individual or an organization to put their particular information. Usually, this method is disseminated with phishing emails containing a link to a webpage that requests the user to fill in personal information [80, 81].
- Configuration or implementation error: the exploitation of a misconfiguration or implementation error could pose a risk to the entire system [82].

- Software or hardware flaws: Many flaws could be available in the system like memory overflow, for instance.
- Malware: Malicious software can be used to affect vulnerability and infect the system or disrupt their services.

3.4.2 Attacks to IoT Ecosystems

A large variety of heterogeneous devices and their interconnection in IoT networks introduce new attacks to IoT ecosystems. The IoT cybersecurity has many different aspects from standard internet in terms of security and privacy. The dynamic environments, memory, and processing power are unknown factors for the internet. More than that, small sensors can be used to connect to the network using any identity or to assume an authentic node to receive sensitive data or to manipulate information with erroneous control messages. In this dissertation, the attacks to IoT ecosystems are classified into the 5-layer architecture of IoT [37]: perception layer, network layer, middleware, application layer, and business layer.

Perception Layer

The perception layer consists of a layer to collect data from sensors of the IoT environment. These devices have unique address identification and use standard communication protocols such as RFID, Bluetooth, NFC, and 6LoWPAN to deliver the information through various applications and servers. Usually, this layer is the bottom layer in IoT architecture responsible for transforming data sensors into a digital signal as processed information into the upper layer [83].

The side-channel attacks are a class of physical attacks in which a malicious user tries to exploit physical information leakages such as data movement timing and power consumption analysis [84]. For example, the health monitoring system could be affected to extract sensitive data from electromagnetic activities around medical devices. This attack class is a passive type because usually, it uses small embedded devices to exploit physical devices such as smart cards and RFIDs. The central aspect of this threat in which a malicious user attempts to conduct a timing attack to compromise a cryptosystem by analyzing the time and to bypass the Same Origin Policy to obtain sensitive information relies on the weaknesses of the system or software used such as one example in certain Apple products with the WebKit component (CVE-2017-7006).

Usually, RFID systems used in IoT environments are susceptible to many attacks ranging from passive eavesdropping to active interference. Due to IoT limitations, an attack to gather sensitive data from a pre-existing RFID tag uses a technique to clone a tag to be used to access unauthorized buildings or systems. A genuine RFID tag could be replaced with a virus coded tag that could do malicious actions against the system. This cloned or replaced tags can lead to many severe problems with financial losses, brand damage, health system. The main problem is the impossibility of distinguishing a genuine and counterfeit component in a system.

Network Layer

The network layer is responsible for transmitting the information from the perception layer and determining routes to the IoT hub. This layer integrates many communication technologies available and various devices to transmit data from different networks and applications. Most attacks from the network layer are related in the computer network or WSNs, and a few threats result from new technologies such as RPL.

Various attacks classes in the network layer try to capture network traffic, to reveal sensitive information, or to compromise the IoT network such as eavesdropping, man-in-the-middle, rogue access, and sinkhole. In the IoT networks, due to the broadcast nature of radio propagation, the eavesdropping attack uses the wireless communication channels to get access by unauthorized users, which presents a weakness of wireless sensor networks. Genuine wireless communications can be intercepted by the eavesdropper, which may apply techniques to decode the data and violate the security property confidentiality and integrity of the sensor' information communications [85]. Usually, IoT systems use encryption as a defense technique to overcome this weakness, but it is not always possible to apply in low powered devices.

In the active attack type, the man-in-the-middle attack uses a vulnerability to alter the legitimate parties' communications secretly.

Besides the traffic interception, malicious users have been authenticated messages exchanged between genuine users to access controlled networks and compromised nodes. The replay attack is one threat of the network layer used to affect the vulnerability of the secure communication mechanisms such as the example of the insulin pump One Tough Ping [86]. Another technique used to intercept the network traffic is the use of a legitimate base station for the system within a wireless network to allow a set of users to get access believing to be a valid base station, and then they capture all traffic. This threat is called the Rogue access point attack, and the malicious users often use free software to hide their presence in the IoT ecosystem. Most of the current approaches fail for detecting rogue access point attacks because the attack is applied behind security mechanisms as firewalls [87].

In the same way, the Sinkhole attack uses the normal behavior of a legitimate node with better link quality to attack users, and after that, malicious users apply several techniques to forward packets or performing leakage of information. Typically, the malicious user's goal is to concentrate a set of users through a compromised node and to tamper the application data from the path that packets are transmitted [88]. This attack is the first step for malicious users to apply the selective forwarding attack, which ensures that all traffic in the targeted area flows through a compromised node. The IoT networks are susceptible to these attacks because the communication pattern shares the same ultimate destination, then a requirement to apply these attacks is to offer a single high-quality route to the base station in influence a large number of nodes.

Middleware

The middleware layer utilizes advanced technologies to access a database directly

to store information such as cloud computing, fog computing, edge computing, and so on. Attacks addressed in this layer focus on the vulnerable application without the user's knowledge. For instance, the Cross-Site Request Forgery is an attack in which malicious users try to execute unwanted action on a web application. Usually, they utilize techniques with social engineering to force the executing actions of the attacker's choosing.

At the same time, the attack Cross-Site Scripting takes advantage of the communication process with IoT applications. Then malicious users try to injecting side scripts to bypass access controls through the RESTful-based IoT systems. These systems are vulnerable to session hijacking because IoT devices handle session connection at the web interface level, and malicious users can take over the session data and control [89].

Application Layer

The application layer is responsible for offering customized services based on user needs from intelligent applications of IoT, such as health-care monitoring, smart grids, smart homes, smart transportation, and so forth. This ability to provide customized services can compromise IoT systems when a vulnerable smart application is not configured correctly. For instance, the SQL injection poses a significant risk to the IoT devices by inserting a malformed SQL statement [90]. This attack is a threat to IoT ecosystems because a successful SQL injection allows us to modify or remove critical data in a system. Usually, malicious users use vulnerable web interfaces to take advantage of flaws in the input validation logic of web applications to obtain unrestricted access to the databases.

The input validation is a severe problem for IoT applications because vulnerable applications allow malicious users by trying all possible combinations of passwords without any protection against brute force attacks. This threat affects IoT devices over the sensor's limited computation power, such as pacemakers [89].

Business Layer

The business layer is the top layer of the IoT architecture for specific applications and services according to particular business models. It is responsible for designing, analyzing, evaluating, and develop elements from gathered data of smart services to build high-level analysis and reports considering the business model, graphs, flowcharts. In this layer, unauthorized accesses pose a risk to the confidentiality and integrity of an IoT system.

3.5 Security Mechanisms

IoT cybersecurity requires protection more than just information, or information systems resources, but an organization as a whole. It relies on the cyberspace and any of their assets that can store and transmit data, the protection of the person using resources in an IoT environment [59]. Several threats violate information privacy bypassing security

mechanisms, as the risks of unauthorized access and increases when system exchange private information between different data sources.

Usually, these data collections with the combining of distinct data sets involve third-party applications that depend on the interaction and collaboration of systems. Attackers apply many security threats in different ways that are categorized according to the architectural layers. Each security threat affects security requirements, and the system is protected using specific security countermeasures. Security mechanisms are software-based tools to designed and implemented to ensure previously defined security purposes. These tools of logical cybersecurity can be passwords, access control, cryptography, firewalls, intrusion detection systems, and intrusion prevention systems, private virtual networks, and so forth.

3.5.1 Firewall

Firewall is a tool designed to protect resources from internal network against Internet threats. Its advantages ensure the control unique point from cybersecurity and avoiding external access to sensitive information and weakness' assets. A firewall is an essential security mechanism to filter all network traffic between the Internet and local network, and it requires the direct connection of one single port (entrance or exit port) to the firewall.

The main components of a network firewall are packet filtering, proxies, network address translation, port forwarding. It can only make decisions based on traffic analyses at the network level using security rules to combine grant access permissions regarding access control policies. Only network packets authorized from security policies are forwarded to the internal network.

Firewalls of packet filtering are based on packets' parameters, which considers each part of them to drop or accept the packet. For instance, the origin port or address, network interface, protocols, and connection status. This packet analysis allows selecting to forward only a particular type of traffic to and from the public network. Usually, firewalls aggregate network address translations to convert private addresses to public addresses from resources within an internal network. The use of NAT isolates these resources from the Internet, and the border router provides this address translation. They are implemented together with the routing process to define forward rules and to block many kinds of threats. The iptables is an example of the packet filtering, and the rules are defined as the following examples:

- -A FORWARD -p tcp -m tcp -d 200.100.100.100 -i eth0 –dport 80 -j ACCEPT
- -A FORWARD -p tcp -m tcp -s 200.100.100.100 -o eth0 –sport 80 -j ACCEPT

These rules allow filtering network traffic specifying a protocol, port, network address, and interface. The parameters -d and -s specify the source IP and destination IP to filter

packets through the system. In this case, the forward policy enables to accept requests on port 80 and allows the server to respond to those requests.

There are two different types of packet filtering to analyze network traffic: stateless and stateful packet filters. The stateless packet filtering without information about the connection. Because of this, the application of the rules occurs in all packets through the machine. It only checks packets from the data link layer to the network layer with few details such as the SYN flag. However, the second type, the stateful packet filtering, which are mechanisms that store information about the packets and use it to match incoming and outgoing packets to determine which packets may be dropped or accepted in the same network communication session. This kind of filtering rule can be in the following example:

- -A FORWARD -p tcp -m tcp -m state -d 200.100.100.100 -i ppp0 –d port 80 –**state NEW** -j ACCEPT
- -A FORWARD -p tcp -m tcp -m state -s 200.100.100.100 -o ppp0 –s port 80 **–state ESTABLISHED, RELATED** -j ACCEPT

The use of these mechanisms is an alternative to avoid sophisticated threats as IP Spoofing attack or packet flooding. However, many vulnerabilities and threats at the application level are not addressed by this security mechanism. Firewalls solve these security problems situated on the application level, also called proxies. Proxies act among connections at the application level to analyze and filter all communication between internal and external devices from the network. It separates the communication from the internal network to remote devices and often aggregates other services such as cache service proxy and blacklist service to provide authentication and an economy of bandwidth limit and robust pages administration, respectively. The use of different types of firewalls constructs secure protection against threats in application servers. For instance, a stateful packet filtering would allow packets with destination to the port 80 because it is following the rules even though no one has requested it. Then, the application proxy firewall would block a malicious character sequence inside of a GET request with malicious actions to the web application as well as the SQL injection threat.

3.5.2 Virtual Private Network

Virtual Private Networks (VPN) are mechanisms used to provide security properties confidentiality and privacy and protect data in transmission. The VPN interconnects two devices using secure tunnels between two private networks using a public non-secure network as communication way such as the Internet. The protection method used for this security mechanism involves the encapsulating of one protocol in another protocol, and QoS guarantees through the service level agreements. It includes the specification of a low bound on the bandwidth available from service providers. The most important of

this mechanism is security, in which sensitive information is transmitted with insurance of data is not intercepted and modified. A VPN uses data encryption and security tools to prevent unauthorized users from sensitive data between two points with a dedicated link.

3.5.3 Intrusion Detection Systems

Intrusion detection systems (IDS) are mechanisms designed to analyze the system or network behavior to detect threats and malicious patterns [91]. The IDS is a security mechanism with a type of defense for detection, and no protection, and can be implemented in a specific system or host and they are called of host IDS (HIDS) or in a particular network (NIDS). A HIDS is designed to monitor the system calls, registry, and log files to identify evidence of malicious behaviors of a successful attack into a system. On the other side, NIDS inspects network packets, or the header information associated with each package to check intrusion attempts. In the case of early detection, this detection can detect and take action against threats before they even reach their destination computer [92].

These security mechanisms use different methodologies of detecting intrusion attempts: signature-based or anomaly-based systems. Signature-based IDSs are designed with commercial purposes considering its characteristics [93]. This type of IDSs relies on the analysis of a given piece of threat for particular elements to characterize an intrusion attempt such as a bit of code, a sequence of characters, the number of the same type of packets during a certain period of the time. A signature is this information labeled as a threat to be identified instantly. This methodology has a high probability of detecting an intrusion based on known information, which generates a low false-positive ratio because of its absolute identification of threats.

Anomaly-based IDSs are security mechanisms designed and developed for research purposes of achieving the solutions to the drawbacks to signature-based IDSs. These systems consider the “normal" system profile to flag anything that does not fit this behavior. Against the signature-based IDSs, they can detect zero-day threats because they have multiple attributes to identify possible intrusion attempts. Its flexibility ensures the detection of threats with modified versions of the same attack but requires a high level of complexity to create a “normal" behavior from the system or network. This limitation of anomaly-based IDSs generates high false-positive rates from the primary detection of unusual behaviors. There are multiple actions that a genuine user can take as malicious behavior, and it will increase the number of false positives that compromise the overall effectiveness of anomaly-based IDSs [92].

3.6 Cybersecurity Proposals for IoT Systems

The literature analysis conducted suggests that there are several initiatives to provide cybersecurity for IoT systems, predominantly through the use of frameworks [32, 94, 95, 96, 97].

Ficco [94] proposed a hybrid and hierarchical event correlation approach for intrusion detection in Cloud Computing. The author provides a complex event analysis supported by an ontology to detect intrusion symptoms in a distributed approach. It gathers several symptoms to report if a certain action is a successful attack. A complex query analyzing a sequence of required conditions is performed on the knowledge base to decide whether the particular behavior represents a potential threat. The same author has also explored this proposal as a distributed intrusion detection in Cloud environments [95]. However, our proposed solution uses the ontology, not for intrusion detection but for analysis from vulnerability and threats as a support for service design and provisioning to prevent or recover from potential attacks.

Alam et al. [32] proposed a layered architecture of IoT to provide secure access provisioning to IoT-enabled things and interoperability of security attributes between distinct administrative domains. They used a semantically enhanced overlay to interlink layers, in which the ontology reasoning and semantic rules enable the security aspects in a machine-to-machine platform. However, the authors only focus on the security requirements of the access control issues, i.e., the semantic rules are designed to ensure access authorization. In contrast, our work can identify and provide security services using the ontology, with reasoning and querying capabilities.

The authors Tao et al. [96] proposed an ontology-based security service framework for IoT-based smart homes handling heterogeneity issues such as security and privacy preservation in a novel multi-layer cloud architectural model and enabling interactions on heterogeneous devices/services. The authors adopted ontologies to model and describe the different aspects of the IoT resources and a security ontology to achieve the security and privacy preservation in the process of interactions. However, the authors designed a small ontology considering only security properties (*Integrity* and *Confidentiality*) and key carrier (*Security Token*) in the process of interactions. They do not explore the reasoning capabilities to infer implicit knowledge on the security ontology, hence limiting the design of and application of security rules. Our work uses a security ontology with a focus on the cybersecurity components to provide security services provisioning based on the reasoning capabilities and a model-driven methodology.

Finally, to conclude our analysis, Ekelhart et al. [97] proposed a framework for information security risk management to measure security through risk assessment, risk mitigation, and evaluation. This work included the presentation of a new methodology, AURUM, used to support the risk management standard using an ontological information security knowledge base to provide a consistent and comprehensive method for the risk manager. This proposal is limited in the sense that it focuses solely on risk management.

3.7 Discussion

Cybersecurity is an essential aspect for the complete adoption of the Internet of Things. This chapter provides the guidelines to understand how to provide the most appropriate knowledge to improve cybersecurity in the IoT ecosystems. An analysis of the IoT environments' main cybersecurity requirements is crucial to identify the challenges to ensure the security properties of communication channels, devices, and IoT systems. Thus, the attacks and security mechanisms are also described. In the literature, many proposals have been used to provide cybersecurity for IoT systems. However, each proposal focuses only on one aspect, i.e., security requirements of the access control issues [32], distributed intrusion detection in Cloud Computing [95].

CHAPTER 4

Semantic Web and Ontology Management

4.1 Semantic Web

Semantic Web refers to the capability to understand the meanings behind information on the web. The Semantic Web is considered an expansion of the World Wide Web in which information is shared using a standard data format (syntax), making it useful to data exchange between machines. The World Wide Web Consortium (W3C)[1] director Tim Berners-Lee defined this term in 1994, which consists of extensive documents for humans to read and machines to manipulate. The main advantage of the Semantic Web is the interoperability between systems because it allows a unique format of information that computers can be manipulated and data integration of distinct sources [98].

The W3C is a collaborative movement that promotes standard data formats on the web as well as a hierarchy of languages standardized for the Semantic Web. This team has been working to improve, extend, and standardize the idea of the Semantic Web. According to *W3C Semantic Web Activity*[2], the Semantic Web can be understood as *"a common framework that allows data to be shared and reused across application, enterprise, and community boundaries"*. This framework is organized with a collection of technologies and standards to build applications according to need to be implemented to achieve full visions of the Semantic Web. Figure 4.1 presents a Semantic Web stack of these technologies and languages standardized by W3C.

Technologies from this semantic web stack are considered a data model for the Semantic Web. According to [99], at the data level, the Resource Description Framework (RDF) is used to represent Semantic Web data. Moreover, the RDF also is considered

[1] http://www.w3.org/
[2] W3C Semantic Web Activity, http://www.w3.org/2001/sw/

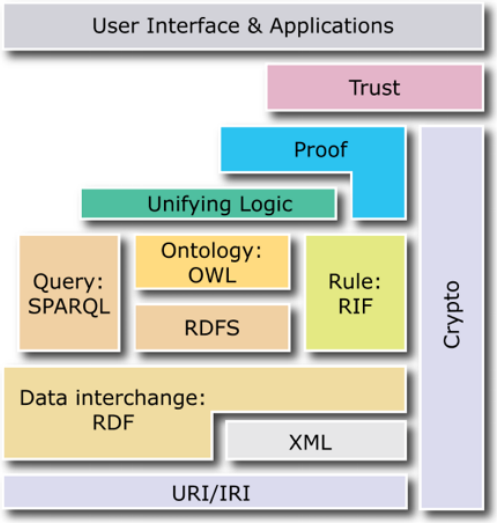


Figure 4.1: Semantic Web stack.

a standard model for data interchange on the Web to represent resources that are associating a Uniform Resource Identifier [100].

4.1.1 Resource Description Framework (RDF)

The RDF was initially created in early 1999 by W3C as a standard (also recognized as the official language) to provide the key to the whole foundation of the Semantic Web. This standard proposes the use of the metadata to describe the distributed information published on the Web and offers the machine-understandable automated processing of the related Web resources [101].

The RDF contains an abstract model to represent data into small pieces using simple rules about each one of these pieces. This model provides a flexible and straightforward method to express any fact of the world and structured enough that computer applications manipulate the expressed knowledge. This abstract model has three components: statement, resources, and predicate.

A statement represents each small piece of information and it takes the form of `Subject-Predicate-Object` (example 4.1). Therefore, an RDF statement must have the following format (this order should never be changed) as the example 4.1.

$$subject\ predicate\ object \tag{4.1}$$

There is a directed graph between the `subject` and the `object`. They are names for things in the world, and each one is called a `resource`. However, the `predicate` (also called `property`) is the name of the relation between these two things, which can be anything, concrete or abstract. This structure also can be called `triple` and represents a piece of information. A collection of statements or triples - `RDF graph` - represents some

given piece of knowledge. There is enough flexibility in this abstract model to represent any new fact to an existing graph to make it more expressive.

New facts are the association between two resources in the world, which can be located through the Uniform Resource Identifier (URI). A URI can be created to identify any resource that can be retrieved directly from the Web. It provides a globally distributed information aggregation to connect all documents containing a resource identified by the same known URI, which has a well-defined meaning as a global uniqueness format from a collection of subject and object resources in the whole world. Also, resources can describe some knowledge about that resource, producing compelling results. The example 4.2 presents the standard format of a URI in the world of RDF.

$$normal\ URI + \# + fragment\ identifier \tag{4.2}$$

An URI example in the context of this dissertation can be written as `http://www.bruno mozza.com/iotsec#Threat` or in a shorter form: `iotsec:Threat`. In a given RDF statement, the predicate must be globally identified using a URI. For instance, the resource `Threat` (subject) `hasSecurityMechanism` (predicate) `VPN` (object). An object can take a simple literal as its value. Literal values are simple text data, and they can be optionally localized, attaching a language tag, such as `"public-key encryption"@en`.

The RDF uses the XML syntax for creating and reading RDF models according to W3C specifications. There are several reasons behind the relationship between XML and RDF, such as minimal semantics. The RDF vocabulary is a set of terms and is typically used in XML with the prefix `rdf`. For example: *rdf:Description*, *rdf about*, *rdf:datatype*, and so forth. This vocabulary of terms is provided by the prefix `rdf` from the URI: http://www.w3.org/1999/02/22-rdf-syntax-ns#. List 4.1 presents the basic syntax of a RDF/XML with a simple example:

List 4.1: A simple example of RDF/XML file.

```
1 <?xml version="1.0" ?>
2 <rdf:RDF
3       xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
4       xmlns:iotsec="http://www.brunomozza.com/iotsec#">
5    <rdf:Description rdf:about="http://www.brunomozza.com/iotsec#ReplayAttack">
6       <iotsec:hasSecuritMechanism rdf:resource="http://www.brunomozza.com/iotsec#
            Kerberos"/>
7    </rdf:Description>
8 </rdf:RDF>
```

The `rdf:RDF` element presents a XML namespace declaration by using an *xmlns* attribute, which specifies two prefix and their URI references such as `rdf` and `iotsec` (Line 2-4). Line 5 contains the `rdf:Description` element that indicates the description of a resource and it uses a `rdf:about` attribute to specify the information of the resource. Also, Lines 5-7 the description about a resource `iotsec:ReplayAttack` has another resource, a security mechanism with name `Kerberos`.

Another way to express a predicate using the RDF vocabulary is to use a common term `rdf:hasType`. List 4.2 shows an example of the use of `rdf:hasType` in Line 6 to define a specification of this instance of threat.

List 4.2: The use of the term *rdf:hasType*.

```
1 <rdf:Description rdf:about="http://www.brunomozza.com/iotsec#ReplayAttack">
2    <rdf:hasType rdf:resource="http://www.brunomozza.com/iotsec#SensorAttack"/>
3 </rdf:Description>
```

The RDF/XML syntax is essential to understand, representing the capabilities of statements, and models information and knowledge to support machine-readable documents. Then, machines can able to discover new information from distributed RDF graphs from RDF/XML documents.

4.1.2 RDF Schema

RDF Schema is a standard proposed initially by W3C in April 1998, as a semantic extension of the RDF standard, to provide a data-modeling vocabulary for RDF data [102]. RDFS can be known as a vocabulary description language, and it consists of a collection of terms to define classes and properties for a specific application domain [101]. This collection of RDF resources is predefined in a namespace informally called `rdfs:` and the prefix `rdf:` to refer to the RDF namespace. These URIs identify them:

rdfs: http://www.w3.org/2000/01/rdf-schema#

rdf: http://www.w3.org/1999/02/22-rdf-syntax-ns#

The RDFS terms are divided into `Classes`, `Properties` and `Utilities`. `Classes` include terms such as `rdfs:Resource`, `rdfs: Class`, `rdfs:Literal`, `rdfs:Datatype`. For `Properties`, the terms include `rdfs:range`, `rdfs:domain`, `rdfs:subClassOf`, `rdfs:subPropertyOf`, `rdfs:label` and `rdfs:comment`. In terms of miscellaneous purposes, Utilities include terms `rdfs:seeAlso` and `rdfs:isDefinedBy`.

First of all, a way more intuitive to the definition of classes uses the `rdf:about` to specify the URI for classes as depicted in List 4.3, which the definition of class `Threat` uses this form:

List 4.3: Definition of the class *Threat*.

```
1 <?xml version="1.0"?>
2 <rdf:RDF
3          xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
4          xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
5          xmlns:iotsec="http://www.brunomozza.com/iotsec#">
6
7    <rdfs:Class rdf:about="http://www.brunomozza.com/iotsec#WebAttack">
8       <rdfs:subClassOf rdf:resource="http://www.brunomozza.com/iotsec#Threat"/>
9    </rdfs:Class>
```

```
10     [...]
11  </rdf:RDF>
```

Also, List 4.3 presents an example of class definition of the `rdf:subClassOf` property defined in RDF Schema. It assumes that the class `Threat` must have been defined in the same document. Following this example, the definition of all classes for the vocabulary can be specified. A property defines the relationship among classes, `rdf:Property` type is used, and `rdf:about` in this case specifies the URI of the property. List 4.4 present the definition of a property.

List 4.4: Definition of the property *threatens*.

```
1  <rdf:Property rdf:about="http://www.brunomozza.com/iotsec#PizzaOntology#threatens">
2     <rdfs:domain rdf:resource="#Threat"/>
3     <rdfs:range rdf:resource="#Vulnerability"/>
4  </rdf:Property>
```

In some case, the `rfds:domain` is optional, which can be used without specify the `rfds:domain` property. Is means that the property `threatens` can be used to describe any class. On the other hand, there exist the possibility to declare multiple `rfds:domain` properties, indicating that the property can be used with a resource that is an instance of every class defined by `rdfs:domain` property [101]. In the same way, the use of multiple `rdf:range` represents that its value has to be someone related to an instance of every class defined at the same time.

Also, the `rdfs:range` is used to specify the possible values of a property such as this is represented by `rdfs:Literal` class contained in RDFS vocabulary. Even though the recommended way for most cases is to use a data type such as `rdfs:DataType`, which is always useful and good practice to specify the valid value for the property. Then, the URI of an example of string datatype is given by the following:

http://www.w3.org/2001/XMLSchema#string

Notice that although the content is well-formed XML content, the structure, in general, is one of the main reasons why a given application can understand the content. Therefore, there are two properties used to provide a class/property name for humans such as `rdfs:label` and `rdfs:comment`. It is used to describe resources in a human-readable description of properties and classes.

4.2 Ontologies

Ontology is a term with origin from philosophy motived by the need for shareable and reusable knowledge bases, but it is used in computer science to describe a specific domain [103] [104]. A common definition of an ontology is *'an explicit and formal specification of a conceptualization of a domain of interest'* [105]. In the world of the Semantic Web, an ontology formally defines a standard set of terms that are used to describe and represent a

domain [106]. Also, the conceptualization should express a shared view between different sources, improving a consensus than an individual view [104].

This definition relies upon many essential aspects to elucidate around ontologies. An ontology is domain-specific and defines the terms used to describe and represent an area of knowledge. A domain represents a piece of an area of knowledge and uses particular terms of its domain. The ontology contains terms and the relationships among them. Different kinds of relationships are considered as properties. They describe the features and attributes of the concepts and are also used to associate distinct classes in the same group. Using a specific language to represent terms and relationships among these terms encodes the knowledge of the specific domain in such a way that machines can understand it.

Formally, an ontology is a composition of a finite list of concepts (also knows as *classes*), relations (*properties*), instances, and axioms. Also, an ontology could be represented by a 4-tuple `<C, R, I, A>`, where `C` is a set of concepts, `R` a set of relations, `I` a set of instances and `A` a set of axioms [107]. Formally, an ontology refers to the statement of a logical theory [105]. It consists of the explicit specification of the conceptualization used to make ontological commitments in a domain of interest. A set of representational terms associated with names of entities in a universe of discourse share a vocabulary of a typical ontology among entities within the same domain, intending to provide reasoning capabilities to discover implicit fact over the individuals. These classes are important groups of objects in the scope of the domain. Hence, these groups are related formally with each other using an explicit relationship that represents a hierarchy of classes. Moreover, one domain is characterized by defining necessary information like properties, restrictions, disjoint statements, and specifications of logical relationships between objects.

Ontologies enable us to generate, share, or consume knowledge defined as a hierarchical view of the world. Nevertheless, ontologies are more appropriate for systems interoperability enabling automatic reasoning on classified data and automatic large-scale machine processing. Many reasons are discussed to argue why ontologies are becoming more and more critical as well as increasing communication between humans and computational systems with languages to formalize the concepts. The utilization of ontologies are characterized to provide systems or people communication around of a knowledge domain but at least, no share the same concepts of the components that domain. This interoperability problem allows us to reuse and share knowledge, which is essential for the support of many existing methods, paradigms, languages, and tools of computer science [104].

Ontology is based on description logic to represents a structured knowledge based on a basic vocabulary of RDF schema with more advanced constructs to describes the semantics of RDF statements. Consequently, it represents an advance to the heterogeneity of smart devices that are used in data interchange for IoT. Usually, IoT devices communicate with each other, and the Internet using different technologies and

a straightforward and powerful representation language is required to promoting its widespread deployment of interoperability. However, it is a conceptual phase, and one of the most aspects that concern the professional and scientists are cybersecurity.

4.2.1 Web Ontology Language (OWL)

Web Ontology Language (OWL) was built upon the World Wide Web Consortium's (W3C) as a research-based revision to explore the idea of knowledge representation form the area of Artificial Intelligence (AI). The idea emerged from the utilization of the Web to make machines understand its content [101]. It results in some languages, and the OWL becomes the most popular language to create ontologies. The OWL documents became a recommended standard on February 10, 2004, in which the web ontology working group creates a formal W3C recommendation in description logic for ontology development and data exchange. One year after, W3C performs a revision with recent advances to attend user requirements while retaining excellent computational properties. The first version had several weaknesses from a user's point of view in its expressiveness, which is a critical requirement for real development work.

In September 2007, a new W3C OWL Working Group[3] was officially formed to discuss these language needs as well. A newest version, OWL 2, became a W3C standard[4] on October 27, 2009. Then, W3C's OWL 2 Primer[5] defines OWL as "*a language designed to represent rich and complex knowledge about things, groups of things, and relations between things*". This language was carefully designed for knowledge representation using a logic-based approach with capabilities for reasoning by machines to identify consistency flaws of that knowledge or to make implicit knowledge explicit. In this thesis, the expression OWL will be used to represent the last version.

As W3C standard, OWL was built as an extension to RDFS for defining classes and properties which enable more powerful reasoning and inference over relationships in an RDF language. OWL was separated into three sub-languages to improve complexity and expressiveness, such as OWL-Lite, OWL DL, and OWL Full. The vocabulary of OWL uses URIs in the RDF, RDFS, and OWL namespaces even though it uses the XML Schema literal to define literal values. Following are some relevant classes to define an ontology in OWL:

`owl:Thing`: The class of all classes and individuals OWL. The properties defined with a range or domain of `rdfs:Resource` have possibility to use any instance of individuals.

`owl:Class`: Classes represent RDF resources as instances of `owl:Class`.

[3] http://www.w3.org/2007/OWL/wiki/OWL_Working_Group
[4] http://www.w3.org/TR/2009/REC-owl2-overview-20091027/
[5] http://www.w3.org/TR/2009/REC-owl2-primer-20091027/

rdfs:subClassOf: It represents a subclass of another class and all instances of the subclass are also instances of the superclass.

owl:DatatypeProperty: It represents all properties that contains ranges and instances of `rdfs:DataType`.

owl:ObjectProperty: It represents all properties that contains ranges of instances of `owl:Class`.

rdf:XMLLiteral: It are literal values in the XML Schema specification and this is a subclass of `rdfs:Literal` and instance of `rdfs:DataType`.

rdf:type: It consists the type of a resource specifying that a resource is an instance of a class.

rdfs:domain: It represents a domain of a specific class specifying an instance of the class as the subject.

rdfs:range: It represents instances and literal range of the class as the object.

After to describe the essential classes of an ontology, OWL language offers a correct representation of the rather complex.

The functional properties are responsible for relating one object for a given subject, for instance, `has_parent`. In this example of `owl:FunctionalProperty`, each individual can be only a parent. A functional property can be specified as an inverse property to represent the inverse meaning as `parent_of`.

The expressiveness of the OWL language enables to guarantee that member of a class that is disjoint from another class have not instances in the second class.

4.3 Query and Reasoning Capabilities

There are three different levels of abstraction to query RDF documents, such as: at the syntactic level, at the structure level, and at the semantic level. The querying at the syntactic level consists of the analysis of the data structure towards the specific data model. In this level, queries there is a challenge to query on distinct data models as the case of the RDF data model is a graph that is not apparent from the XML tree structure. In querying at the syntactic level requires a hard task to formulate a query specifying exactly information like "Give me all elements grouped in a `Description` element with an `about` attribute with value 'http://www.brunomozza.com/user/Bob'."

At the structure level addresses the RDF data model independent of the specific syntax as a set of triples to represent a statement of the form Subject-Predicate-Object in any RDF document.

4.3.1 SPARQL Query Language

Simple Protocol and RDF Query Language (SPARQL) is a standardized query language for RDF graphs with powerful capabilities to filter results and generate new graphs. As a W3C recommendation, this language tries to match patterns in the graph and bind wildcard variables in the set of statements. Some clauses are available to construct queries such as `SELECT`, `CONSTRUCT`, `ASK`, and `DESCRIBE`.

The SPARQL SELECT queries are useful to query across a graph of relationships, spanning multiple triples, and searching for patterns that match the subjects and predicates. The basic query process consists of a collection of pieces of the graph in triples. A SPARQL query starts with one or more PREFIX statement to define a BASE URI to which all relative URIs are concatenated.

```
PREFIX prop: <http://dbpedia.org/property/>
```

The construction of a SELECT query involves a subset of the variables in the graph patterns whose bindings the query will return for each match. This clause contains a WHERE clause to filter only the graph pattern to match as a collection of triples. The standard to use variables in the SELECT queries starts with a question mark (`?`) or a dollar sign (`$`) without making the distinction between them. List 4.5 presents the simple example of a SPARQL SELECT query to search all threats affects security properties and has security mechanisms:

List 4.5: An example of a SPARQL SELECT query.

```
1  PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
2  PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
3  PREFIX iotsec: <http://www-usr.inf.ufsm.br/~brunomozza/iotsec#>
4  SELECT  ?label ?threat ?secProperty ?secMech
5  WHERE {
6     ?threat rdfs:label ?label .
7     ?threat rdf:type iotsec:Threat .
8     ?threat iotsec:affects ?secProperty .
9     ?threat iotsec:hasSecurityMechanism ?secMech .
10 }
```

The result of this query returns a collection of triples that were matched in the graph patterns such as threats (`?threat`), security properties (`?secProp`) and security mechanisms (`?secMech`), respectively. However, this type of query does not consider incomplete information about a resource. In these cases, the OPTIONAL clause allows to use of available information in the graph pattern but maintain solutions that are missing. SPARQL provides the FILTER operations to constrain solutions based on the requirements of a subject, predicate, or object. Table 4.1 presents some SPARQL filters within a basic graph pattern.

Then, the SELECT clause finds existing graph patterns in a collection of relationships specifying additional constraints on solution bindings using a small set of operators

Table 4.1: SPARQL filters.

Category	Operator	Examples
Logical	`!, &&, ||, =, !=, <, <=, >, >=`	`?hasPermit || ?age < 25`
Math	`*, -, *, /`	`?decimal * 10 > ?minPercent`
Existence	`EXISTS, NOT EXISTS`	`NOT EXISTS ?p foaf:mbox ?email`
SPARQL tests	`isURI, isBlank, isLiteral, bound`	`isURI(?person) || !bound(?person)`
Accessors	`str, lang, datatype`	`lang(?title) = "en"`
Miscellaneous	`sameTerm, langMatches, regex`	`regex(?ssn, "\\d3-\\d2-\\d4")`

derived from XPath 2.0[6]. Table 4.2 present the most common property paths to match triples through a graph.

Table 4.2: Regular expression operators.

Operator	Description
path1/path2	Forwards path (path1 followed by path2)
^path1	Backwards path (object to subject)
path1|path2	Either path1 or path2
path1*	path1, repeated zero or more times
path1+	path1, repeated one or more times
path1?	path1, optionally
path1m,n	At least **m** and no more than **n** occurrences of path1
path1n	Exactly **n** occurrences of path1
path1m,	At least **m** occurrences of path1
path1 ,n	At least **n** occurrences of path1

List 4.6 shows the example with a FILTER operation to provide the filter operations in the SELECT query. This filter operation allow to set up conditions of a variable's value, such as check whether a string follows the desired pattern. The SPARQL provides a regex operator and an optional set of flags to find case differences and the variable's value. For instance, a query could limit results only to threats label starting with the letter "D" using *FILTER regex(?label, '^D', 'i')*.

List 4.6: The FILTER operation in the example of SPARQL query.

```
1 PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
2 PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
3 PREFIX iotsec: <http://www-usr.inf.ufsm.br/~brunomozza/iotsec#>
4 SELECT  ?label ?threat ?secProperty ?secMech
5 WHERE {
```

[6]https://www.w3.org/TR/xpath20/

```
6     ?threat rdfs:label ?label .
7     ?threat rdf:type iotsec:Threat .
8     ?threat iotsec:affects ?secProperty .
9     ?threat iotsec:hasSecurityMechanism ?secMech .
10    FILTER regex(?label, '^Denial', 'i')
11 }
```

However, SPARQL provides the `CONSTRUCT` clause to get a solution with a list of variable bindings from a query and generate a new graph. This clause replaces exactly the `SELECT` clause. However, instead of returning a table of result values, `CONSTRUCT` returns an RDF graph. It is created by taking the results of the equivalent `SELECT` query and filling the variable's value according to the `CONSTRUCT` query form. List 4.7 shows an example of the CONSTRUCT clause to generate a new graph according to the results values.

List 4.7: The CONSTRUCT clause in standard example.

```
1  PREFIX  dc: <http://purl.org/dc/elements/1.1/>
2  PREFIX app: <http://example.org/ns#>
3  CONSTRUCT {
4     ?s ?p ?o
5  }
6  WHERE {
7     GRAPH ?g { ?s ?p ?o } .
8     { ?g dc:publisher <http://www.w3.org/> } .
9     { ?g dc:date ?date } .
10    FILTER ( app:customDate(?date) > "2018-05-28T00:00:00Z"^^xsd:dateTime ) .
11 }
```

Usually, ASK queries are used to check whether or not a query pattern matches. This clause only results in a boolean to inform whether or not a solution exists. List 4.8

List 4.8: The ASK clause in a standard example.

```
1  PREFIX foaf:    <http://xmlns.com/foaf/0.1/>
2  ASK  {
3     ?x foaf:name  "Bob"
4  }
```

SPARQL provides resource information retrieval from RDF data using the `DESCRIBE` clause. Often, this information is used to create a result set with the structure of the RDF in the data source. This result contains data from each resource identified in a solution. It can be used to get the description data about a specific resource such as `DESCRIBE <http://dbpedia.org/class/yago/>`. List 4.8 present a typical example to return information about at most one person.

List 4.9: The DESCRIBE clause in a standard example.

```
1  PREFIX foaf:   <http://xmlns.com/foaf/0.1/>
2  DESCRIBE ?x
3  WHERE {
```

```
4     ?x foaf:mbox <mailto:bob@org>
5  }
```

Sesame OpenRDF Workbench [108] is an architecture developed by Aidministrator Nederland b.v.[7] for storage and querying of RDF and RDFS information. It offers access methods to RDF data and schema information through export and querying capabilities. This powerful framework includes mechanisms to create, parse, store, make inferences, and queries over such data. Sesame translates queries into a set of calls to the SAIL (Storage and Inference Layer), which is an application programming interface with specific methods to translate these methods to calls to its specific database to the persistent storage.

4.3.2 SWRL Reasoning Language

The Semantic Web Rule Language (SWRL) is a standard language based on OWL-DL and the rule markup Language. The SWRL was designed to integrate tightly with OWL, and it provides both OWL-DL expressivity and rules from rule marked language. Many distinct rules can be specified using the SWRL as access control policies, automation rules, design rules.

These kinds of rules are expressed as implications rules to SWRL syntax because OWL expressions are based on classes and properties. SWRL rules are built in terms of OWL concepts such as classes, objects properties, data properties, and instances. A particular rule is a type of OWL axiom in the ontology and can also interact with the existing axioms present in the ontology. The syntax of SWRL follows with the validation to the antecedent is satisfied. However, the OWL expressions can occur in both antecedent and consequent (Example 4.3).

$$antecedent \rightarrow consequent \tag{4.3}$$

Both sides of the arrow are composed by conjunction and can be stated as A(x), B(x, y). Besides, A is an OWL description, B is an OWL property, and x and y can be Datalog variables, OWL instances, or OWL data values. The rule 4.4 presents a simple example using class atoms to declare that all individual of type `Person` with greater than 17 is an `Adult`. In this example, `hasAge` is an OWL object property, `?p` and `?age` are variables used as argument referring OWL individuals or data values. These variables are treated as universally quantified, with their scope limited to a given rule.

$$Person(?p), hasAge(?p, ?age), greaterThan(?age, 17) \rightarrow Adult(?p) \tag{4.4}$$

In this rule, `hasAge(?p, ?age)` presents the relationship that all individual of type `Person(?p)` has age in the variable `?age` and then it will know meanings of the other relationships. However, SWRL has a set of built-ins to allow further extensions and

[7] See `http://www.aidministrator.nl/`

flexibility for various implementations and interoperation of the language. In our example, `greaterThan(?age, 17)` is one example that the first argument (`?age`) is greater than the second argument (data value), if there is one defined for the type.

4.4 Cybersecurity Ontologies

In this chapter, the state of the art on the cybersecurity and Internet of Things is described to analyze previously published works in these topics of interest [109]. This strategy allows dealing with cybersecurity concepts around different perspectives of this field. Ontology is a potential tool utilized mainly for the structuring of an area of interest. This analysis aims to characterize the research area and to establish a base knowledge of IoT security (Figure 4.2). According to the state-of-the-art, several existing security ontologies have been proposed in the literature and point to different categories such as general security ontology [110, 111, 112, 113] and security ontology applied to a specific domain [114, 115, 116].

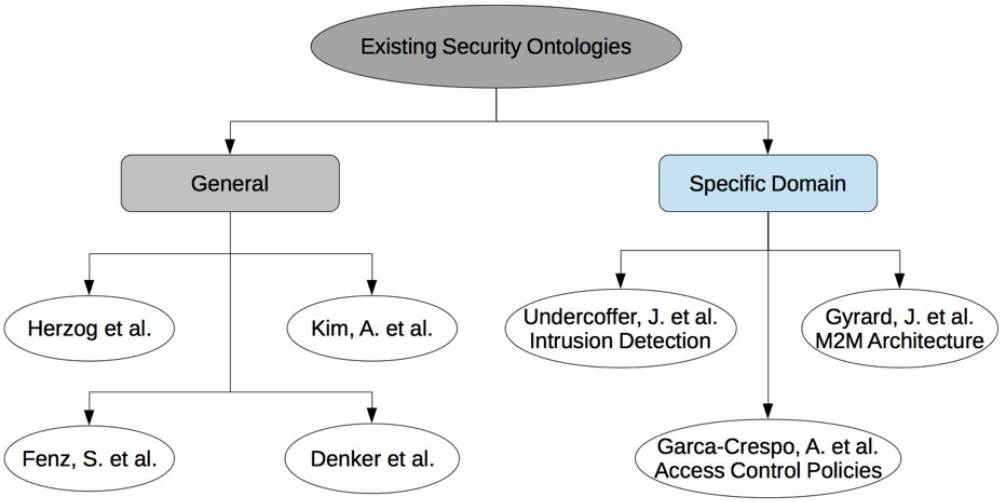


Figure 4.2: Existing security ontologies available.

Some ontologies address only one part of the security domain (e.g., computer attacks), and others explore the overview of cybersecurity. We shall present a brief description of the summary of the security ontologies found. This selection is limited in the scope that supporting the ontological structure of the cybersecurity domain and the availability of the ontology.

4.4.1 General Security Ontologies

This Section describes existing security ontologies available in the scope of this work.

4.4.1.1 Herzog et al. - "An Ontology of Information Security"

The authors propose an OWL-based ontology of the information security overview to model assets, threats, vulnerabilities, countermeasures, and their relations [112]. Ontology is useful for reasoning about relationships between entities, and it can help to answer what threats are potential to violate the assets available (Figure 4.3).

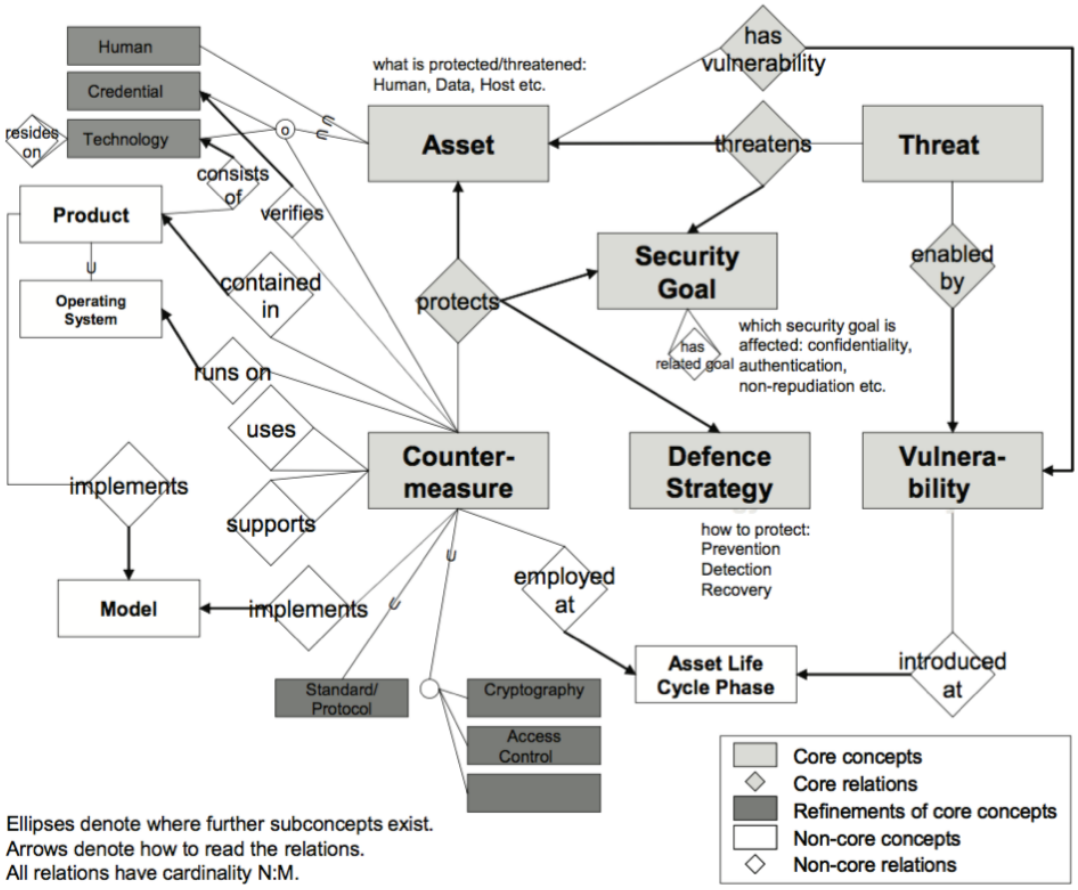


Figure 4.3: Overview of the security ontology proposed by [112].

Information security ontology comprises 88 threat classes, 79 asset classes, 133 countermeasure classes, and 34 relations between those classes. The authors describe inference and query language SPRAWL (SPARQL Protocol and RDF Query Language) to create views on the ontology. Inference also permits to explore other countermeasures to control for a given threat. Furthermore, extensions, technical implementations, and tools are working with it. For instance, the extension of the impact of a threat can be used.

4.4.1.2 Fenz and Ekelhart - "Formalizing Information Security Knowledge"

This work proposes Security Ontology to provide an unified and formal knowledge using an ontological structure for the information security domain [111]. The ontology contains 500 concepts, and 600 formal restrictions that are represented by either graphical, textual,

or description logics representation and the code ontology follow the OWL-DL (W3C Web Ontology Language) standard (Figure 4.4).

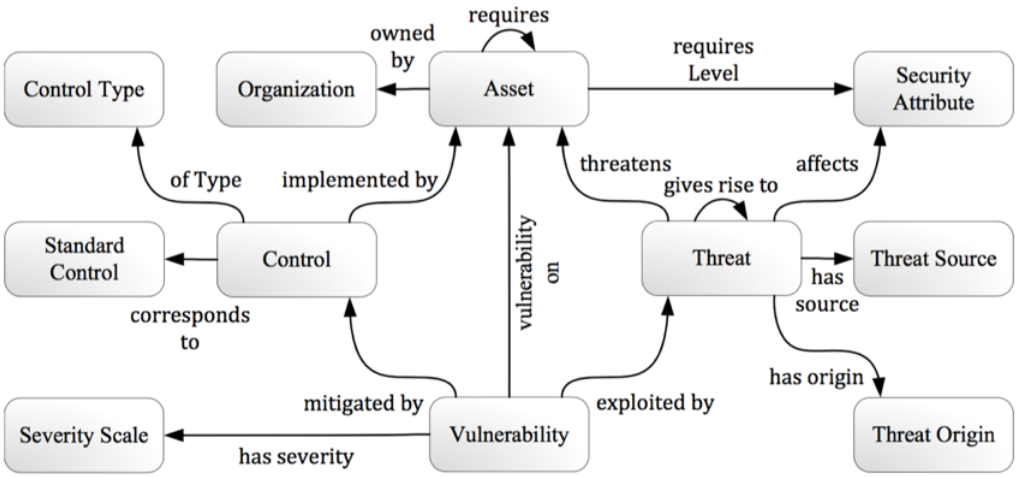


Figure 4.4: Security relationships proposed by [111].

Security ontology is composed of five sub-ontologies, such as asset, control, vulnerability, threat, and security attribute. This structure is based on Landwehr's security and dependability classification [117]. This ontology is applied in different approaches to the simulation of various attacks [118], security risk analysis [119], information security concept in risk-aware business process management [120], holistic IT-security approach for small and medium-sized enterprises [121]. The ontology focuses on providing a model for the entire cybersecurity domain, including non-core concepts such as the infrastructure of an organization as well.

4.4.1.3 Kim et al. - "Security Ontology for Annotating Resources"

This work proposes the NRL security ontology aggregate with existing ontologies in other domains and includes an accurate description of protocols, mechanisms, objectives, and other security concepts at various levels of details [113]. Figure 4.5 presents the main aspects of the security ontology proposed from [113].

This security ontology was proposed to address the limitations of existing ontologies. Regarding the organization of subclass relationships, the ontologies are not intuitive to understand the relations between them and cannot express all the security information that needs to be described. Another issue highlighted is the lack of expressiveness and the possibility of describing classes different from security-related information. Therefore, the authors improve these limitations in the NRL security ontology proposed with security information about all types of resources, the ability to annotate security information in different environments, easy to extend and provide reusability and facility to match high-level security requirements to lower-level capabilities.

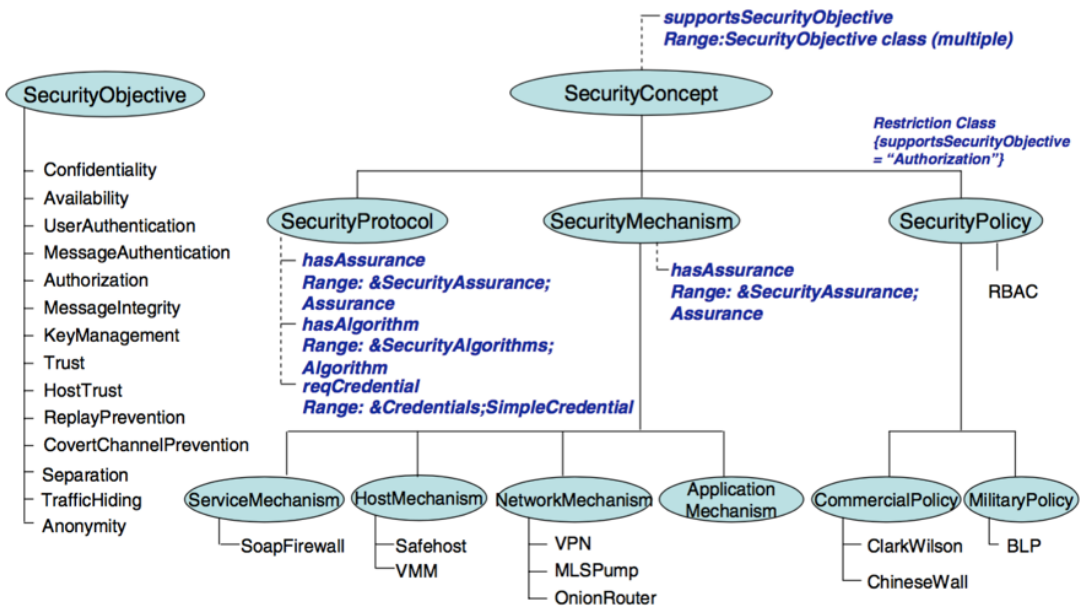


Figure 4.5: Overview of the main security ontology proposed by [113].

4.4.1.4 Denker et al. - "Security in the Semantic Web using OWL"

The authors propose an ontological approach to enhancing the Semantic Web with information security [110]. Figure 4.6 presentes the main classes of high-level security concepts and relationships between them compose the ontology "OWL-S Security and Privacy".

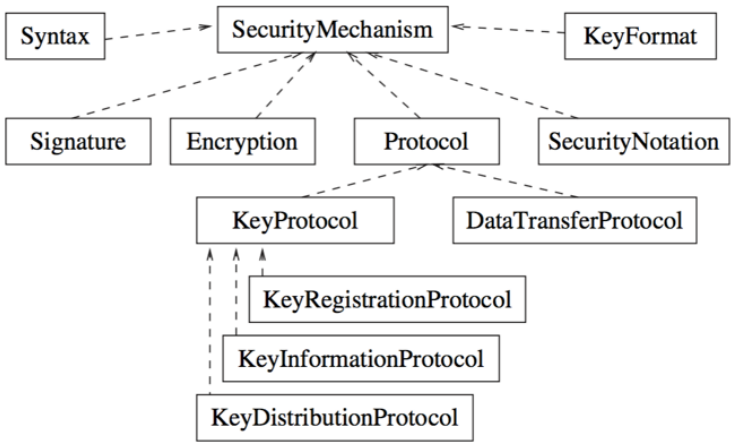


Figure 4.6: Security ontology proposed by [110].

The first sub-ontology defined is Authentication, which has subclasses related and specialized, for example, Public key, X.509 Certificate, One Time Password. The second sub-ontology is considered specific general security notations as Security Mechanisms. Some examples of second sub-ontology are Access Control, Authorization, Data Integrity, Anonymity. The proposal of this ontology provides a basis for reasoning, reads its metadata, and employing Semantic Web reasoning techniques shows services to the users.

The authors address the knowledge representation and reasoning issues for trust and security in the Semantic Web.

4.4.2 Security Ontologies Applied to Specific Domain

This segment describes existing security ontologies applied in a specific domain and available in the scope of this study.

4.4.2.1 (Undercoffer et al. - "Modelling Computer Attacks: An Ontology for Intrusion Detection"

This work states the benefit of transitioning from taxonomies to ontologies, and the authors propose an ontology specifying a model of computer attack using DAMLJessKB to implement the ontology [116]. This ontology describes the most common attacks are the result of malformed input exploiting a software vulnerability of a network, and the consequence is a denial of service. The ontology is composed of classes: host, system component, attack, input, means, input validation error, logic exploit, and consequence.

4.4.2.2 Gyrard et al. - "An Ontology-Based Approach for Helping to Secure the ETSI Machine-to-Machine Architecture"

The authors designed the STAC ontology with the state of the art of wireless communications (cellular, wireless, wired), devices (sensor or mobile phone), and applications (programming language, framework, database) [115]. This work combines existing security ontologies according to different domains to provide an approach to help software designers to secure their M2M applications. This ontology does not describe the vulnerabilities of the M2M technologies, and it only suggests countermeasures available to threats that affect a type of technology.

4.4.2.3 García-Crespo et al. - "SecurOntology: A Semantic Web Access Control Framework"

The authors present a security ontology focused on representing role-based access control policies to access control based on knowledge-oriented descriptions [114]. The ontology SecurOntology is composed of classes, properties, and rules. Classes compose a basic hierarchy of main concepts such as resources, owners, roles, permissions (read, write, and execution) and permission to the current resource, consults. The properties are the relations between the classes such as: `hasRole`, `isOnwerOf`, `itsOwnerIs`, `hasPermission`, `hasChild`, `isChildOf`, `resource`, `permission`. Finally, the rules are responsible for inferring new knowledge, which does not exist in the knowledge base.

4.5 IoTSec Reference Ontology

IoTSec ontology is a reference ontology for IoT cybersecurity developed to characterize a domain-specific with cybersecurity concepts. IoTSec ontology explores certain aspects of the relationship amongst essential components of risk analysis such as Assets, Threats, Security Mechanisms, Vulnerability, Security Properties, and Risk. Of all the factors mentioned earlier, the risk is the component that has not been previously inserted within the reference ontology because any significant change to an alternative component could have a significant impact on risk. At present, the IoTSec ontology consists of certain statements, which are summarised in Table 4.3.

Table 4.3: Number of classes, properties, axioms and annotations in the IoTSec ontology.

Ontology metric	**#**	**Ontology metric**	**#**
Classes	228	Logical Axioms	1895
Object Properties	24	Annotations	1418
Data Properties	7	Individuals	607

According to state of the art, several existing security ontologies have been proposed in the literature, but only a few are available: security overview ontology [111, 112, 113, 122], and security ontology applied to specific domain [114, 115, 116]. Some ontologies address only one part of the security domain (e.g., computer attacks), and others explore the overview of cybersecurity. The reference ontology for security in the IoT (IoTSec ontology) is the ontology harmonization based on ontology development methodology. The IoTSec ontology is different because it uses the correlation between classes to provide knowledge using reasoning capabilities to the implementation of cybersecurity mechanisms and services at design and run time.

4.5.1 Modeling Process of the Reference Ontology

In this section, the modeling process provides a methodology to create a reference ontology for IoT cybersecurity. The process uses different types of knowledge representation, such as existing ontologies, taxonomies, and the literature review as inputs for the analysis. The process of ontology development follows a set of criteria proposed by [103]. The criteria presented are:

- Clarity consists of the motivation for defining terms effectively to express the intended meaning. Definitions from the ontology should be objective.
- Coherence relies on the maintenance of consistency of the ontology with the coherence of the definitions.
- Extendibility specifies that an ontology should be designed to anticipate the uses of the shared vocabulary.

- Minimal encoding bias focus on the conceptualization should not be specified with a particular symbol-level encoding. The encoding bias only is a good practice when are made purely for the convenience of notation or implementation.

Following the criteria, a conceptual model represents the design process of the reference ontology. Figure 4.7 presents this conceptual model. The purpose of this reference ontology is helping to find new security solutions in the approximation of areas of cybersecurity and the Internet of Things. The defining terms of specific areas are related to others and need to understand the relationships among them.

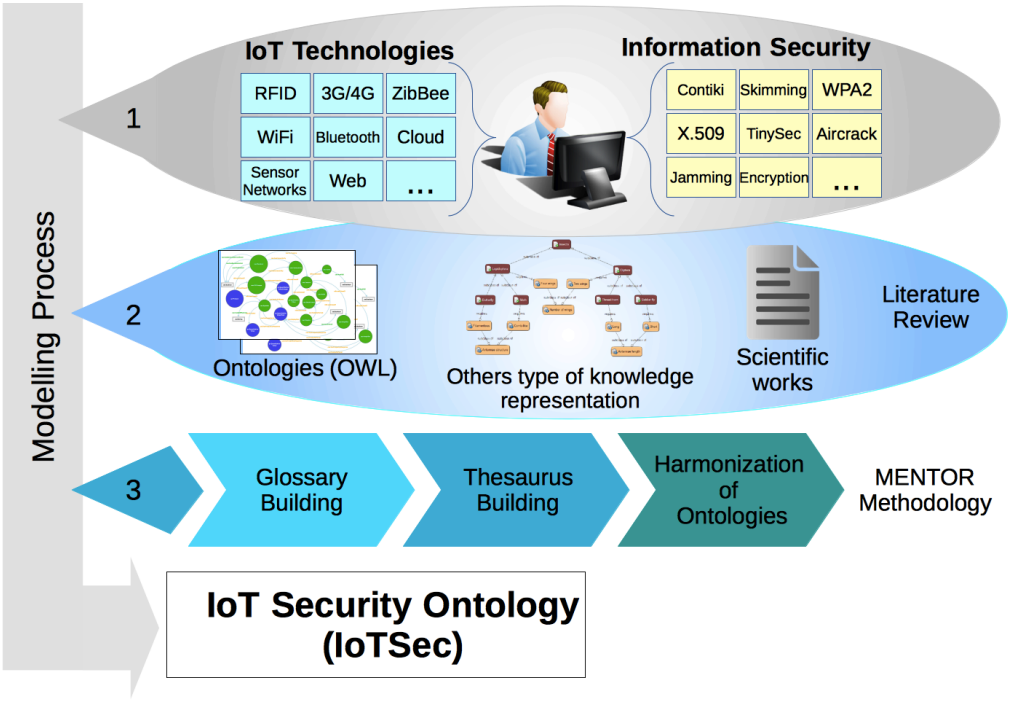


Figure 4.7: Modeling process of the reference ontology for security in IoT.

The modeling process adopted enables the creation of a reference ontology for security in IoT is composed of three steps: the 1^{st} step explores several keywords of related areas in the context of this thesis work. These keywords, many times, appear separately but are needed to approximate to create secure solutions for new IoT applications. Hence, each keyword has relevant aspects that are needed to clarify different contexts, and when they are related among different areas, it allows that software designers and users to understand new approaches to solve critical problems.

The 2^{st} step investigates the literature review crossing of the concepts in IoT and cybersecurity, identified in 1^{st} step. Different kinds of knowledge representation are analyzed to gather terms and descriptions. In this step, existing ontologies and taxonomies are collected to establish and identify similarities and differences. The main contribution of this step is breadth with knowledge about this subject to combine similarities, differences, and gaps of information.

The 3^{st} step is responsible for the harmonization and mapping process of existing ontologies in the design of a reference ontology. This step uses the MENTOR methodology to assist the construction process and avoid inconsistencies in the ontology. MENTOR has some steps that need to be highlighted in your importance to the process, such as glossary and thesaurus building. Glossary and thesaurus building are the most important tasks to identify semantic mismatches and enhance the ontology harmonization process. Another important step of the MENTOR methodology is the harmonization process of discussing the ontology structure and after for content definitions of the reference ontology.

These steps encompass the tasks needs to the modeling process of a reference ontology. We adopted them according to the analysis of state of the art of the existing ontologies proposed by [123], where many existing ontologies in the context of cybersecurity there no available anymore. Related works are essential to demonstrate positive and negative aspects in the design of proposed ontologies. The modeling process covers the most aspects of the two research areas for the design of a reference ontology for security in IoT.

4.5.1.1 Mapping of Security Ontologies to the Reference Ontology

In this section, we mapped each top-level concept of the main existing security ontologies to the reference ontology. Table 4.4 presents the comparative among main classes mapped by IoTSec. The relationships between top-level concepts were presented in Section 4.5.

Table 4.4: Mapping of security ontologies to the IoTSec.

[112]	[111]	[115]	[113]	[110]	**IoTSec**
Asset	Asset	Technology	No implemented	No implemented	**Asset**
Threat	Threat	Attacks	No implemented	No implemented	**Threat**
Vulnerability	Vulnerability	No implemented	No implemented	No implemented	**Vulnerability**
Countermeasure	Control	Security Mechanim	Security Mechanim	Security Mechanim	**Security Mechanim**
Security Goal	Security Attribute	Security Property	Security Objective	Security Notations	**Security Property**
Defense Strategy	Control Type	No implemented	No implemented	No implemented	**Type of Defense**

`Asset` is a concept of high abstraction level that is vital to the success of the organization and needs to be protected according to your value to the organization. These may include physical assets, information, software, people. The Asset class is mapped to the reference ontology to represent the valuable item of the organization. Gyrard et al. [115] defines `Technology` to define instances of many assets, which they have vulnerabilities and have specific security mechanisms implemented to protect it.

Another class mapped to the reference ontology is the `Threat` class. This class represents all kind of the threats that causes harm to an asset and therefore, an organization. These threats can occur from attacks on the systems causing information disclosure and leaking, data destruction, and other problems. A threat is successful when one or multiple existing vulnerabilities are exploited in order attackers obtain advantages of the assets. However, `Vulnerability` class describes different kinds of weaknesses in physical layout, procedures, management, administration, hardware software, or information that allow a threat to affect an asset. Threats have characteristics that cause

distinct impacts according to specific assets exploited, and more than one asset may be affected by a threat. Frequently, software vulnerabilities known are claimed by customers to vendors create the patches to fix it. The reference ontology mapped the `Vulnerability` class with the same name adopted by other ontologies that implemented it.

Vulnerabilities need to be protected by security mechanisms. Sometimes, vulnerabilities never are exploited by many reasons such as unavailability, lack of attackers' knowledge. However, if an attacker discovers methods to exploit it, then malicious actions may cause several damages to the organization. So, several measures may be adopted to protect the weakness of the asset and avoid concerns. `Security Mechanism` class mapped to the reference ontology describes tools and approaches to protect the vulnerabilities known. The most critical aspect of choosing a security mechanism depends on the valuable level of the asset to warrant some degree of protection. This class is referred to with distinct names such as Control, Countermeasures, and also Safeguard. We adopt the name of Security Mechanism to the reference ontology to emphasize the meaning of the security tools to improve the environments, and it represents best in the relation of others.

The reference ontology further distinguishes between different types of defense according to the security mechanisms adopted, and depending on which vulnerability is addressed, a type of security mechanism is best. In this case, `Type of Defense` class is mapped to the reference ontology as a set of types of the strategy to use security mechanisms such as detection, correct, or prevention threats.

4.5.1.2 Ontological Model and Knowledge Base

The IoTSec ontology was developed using the OWL language, as detailed in Figure 4.8. The utilization of IoTSec ontology in new applications involves instantiation (also called data population) and the use of a knowledge base. There is an important aspect that must be clarified regarding the difference between an ontology and a knowledge base. An ontology refers to a conceptual representation of classes and their relationships. There is an analogy present in databases because an ontology has similarities to a database schema. This said a knowledge base is an instantiation of this ontology with a population of individuals where, in the case of databases, an instantiation exists when registers are inserted in the database. Therefore, an ontology in our context is seen as a conceptual representation of the relationship between classes present in security components.

The ontological model, highlighted in grey, demonstrates the relationship between the classes, subclasses, and object properties. The model has an aggregated value only when it contains individuals correlated amongst them. The relationship between the latter will be established by an instantiation of the ontology, which will be automatically related to their respective classes. Therefore, a knowledge base requires an instantiation with a data population from different types of data sources following the requirement of the ontological model. Considering the IoT cybersecurity domain, Figure 4.8 depicts an

example of the IoTSec ontology instantiation (knowledge base).

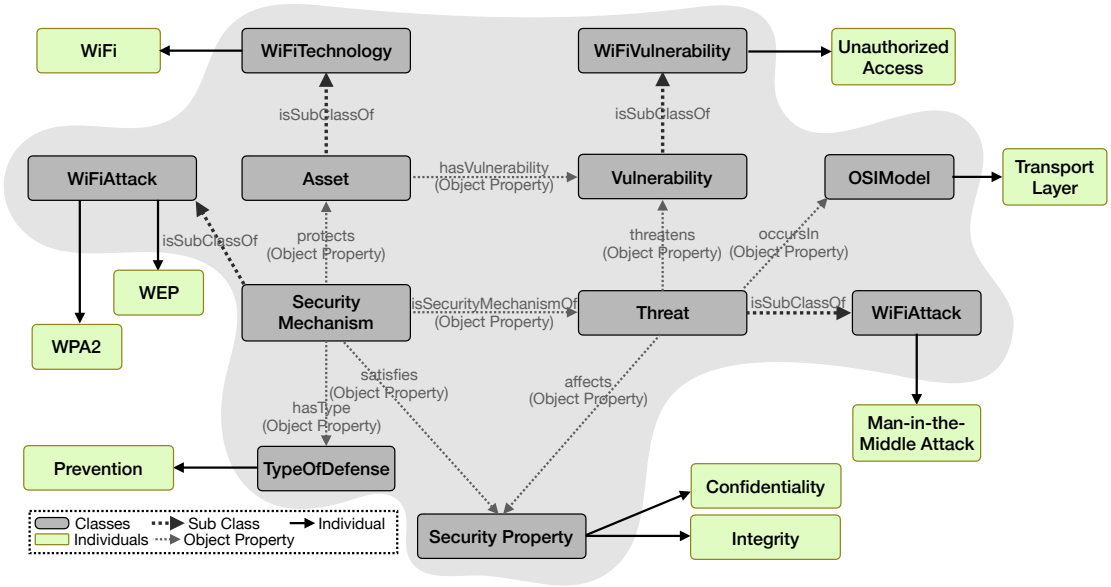


Figure 4.8: Example instantiation in the IoTSec ontology.

The example presented above demonstrates how a threat can be represented as an instance of individuals of classes in the IoTSec ontology. However, when the ontology has a composition of individuals of assets, threats, vulnerabilities, security mechanisms, and security properties, then their inference rules become more powerful, thereby resulting in inferred facts among these classes to discover new facts and relationships from the implicit knowledge.

4.5.1.3 Reasoning capabilities

Reasoning refers to the ability to identify the knowledge base uniformity, correctness of data instances, and assertions using rules. This process derives implicit facts from the existing knowledge and can be classified into logic-based context reasoning, rule-based reasoning, or deductive and inductive reasoning. Also, logic-based context reasoning may be applied. Some ontology verification processes occur as a result of reasoning. These include [107]:

- Verifying the consistency of the ontology and knowledge base.
- Verifying unintended relationships between classes.
- Automatically classifying instances in classes.

The processing of the direct and indirect relationships of the ontology model requires a descriptive language based on rules. This language also is used to enhance the descriptive ability of OWL. Specific deductive inference rules operate on the ontology

and derive new assertions and also ensure the ontology uniformity. The definition of inference rules is established with the SWRL [124].

The example 4.5 of the inference rule demonstrates new facts to be found from implicit knowledge.

$$SecurityMechanism(?sm),\ SecurityProperty(?sp),\ Threat(?t),\ affects(?t, ?sp),\\ isSecurityMechanismOf(?sm, ?t) \rightarrow satisfies(?sm, ?sp) \tag{4.5}$$

Given the inference rule presented, the rule above establishes a new relation when a set of axioms satisfies the rule requirements. The object property `isSecurityMechanismOf (?sm,?t)` provides the ability to link `SecurityMechanism(?sm)` and `Threat(?t)`, and `affects (?t, ?sp)` for linking between `Threat(?t)` and `SecurityProperty(?sp)`, respectively. Then, this association will makes explicit the facts in the object property `satisfies(?sm,?sp)` that was only implicit in theses represented facts.

When the system processes the inference rules, the reasoning engine can infer that the security mechanism protects a threat and affects some security properties. It also reinforces that this security mechanism satisfies the specified security properties. When this relation is mapped, all restrictions that were set for security properties will be applied to it. This will be useful when queries are done to identify which security mechanisms protect each security property.

Since the relationship between individuals is not explicit, the outcome of this inference rule will result in the enrichment of cybersecurity. Given the simple nature of the examples provided, it must be stressed that machine processing requires intelligence factors to provide explicitly represented facts. OWL has a high expressiveness to achieve many cases using inference rules.

4.6 Discussion

In this chapter, the semantic and ontology concepts are presented. The importance of its approaches and the advantages enabled are also explained. The ontology management is described in detail with information about the OWL language and presenting different levels of abstraction to query and reasoning capabilities wit SPARQL and SWRL languages, respectively. Existing security ontologies are explained in detail to summarize in general and specific domain ontologies. Thus, the IoTSec ontology is presented with the modeling process to create a reference ontology for IoT cybersecurity.

CHAPTER 5

Framework to Enable Cybersecurity in IoT Ecosystems

A cybersecurity framework is an essential requirement for the complete adoption of the Internet of Things by industry, academic, and domestic stakeholders. The principal aspect of establishing a cybersecurity framework is to identify what are the security-relevant capabilities of the devices to be connected and correctly used. Inside an organization, several operations compose business processes that generally contain sensitive information from customers. Cybersecurity practices are vital to business security and are a factor that has not been taken into consideration by companies as part of their risk management process [126].

5.1 Towards an Adaptive Cybersecurity for IoT Networks

Adaptive cybersecurity is an approach to adjust in real-time attributes based on the behavior to respond to new and unusual threats in critical services [127, 128]. The MAPE-K (Monitor, Analyse, Plan, Execute, Knowledge) autonomic loop was suggested by International Business Machines (IBM) to achieve the autonomic computing. Figure 5.1 presents the reference model for autonomic control loops that it has been applied to intelligent agents to collect environmental information through sensors and uses this data to determine actions to execute on the environment [131].

The MAPE-K reference model is composed of two elements: the autonomic manager and the managed element. The managed element represents any resource addressable for autonomic behavior with an autonomic manager. Smart devices and sensors, or also called probes, are responsible for collecting contextual attributes and information of the environment that are relevant to the system.

The autonomic manager represents the improvement process using high-level objectives to construct and execute plans based on an analysis of this information. It

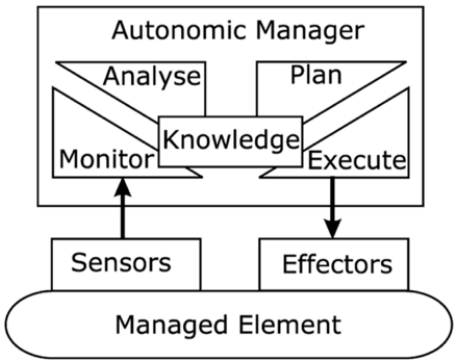


Figure 5.1: MAPE-K reference model for autonomic control loops [131].

creates low-level actions to achieve these objectives using the internal knowledge of the system and execute changes through effectors.

The data collection uses two types of monitoring in autonomic systems: active and passive monitoring. Active monitoring represents the addition of software or hardware components to gather specific information or capture functions. This type of monitoring uses techniques to control the experiment for timing quality of service or latency between devices, for example. On another side, passive monitoring is a more observational study about environment behavior. Instead of injecting code or artificial traffic, passive monitoring captures the traffic or information already on the network to analyze according to the system objectives.

Adaptive cybersecurity is a solution that learns and adapts to the changing environment at runtime in the face of changing threats and anticipates threats before they manifested. This approach is a continuous process to learn, adapt, prevent, identify, and respond to unusual and malicious behavior at run time.

The continuous cycle of security monitoring is necessary for the use of suitable mechanisms depending on the information about the context and status of IoT devices. It is appropriated for IoT scenarios because of high interactions among heterogeneous devices and the environment with critical risks to our lives. Monitoring information context allows choosing a suitable security tool for ensuring one or more cybersecurity properties. Sometimes, whether it changes the status of the secure behavior of a given situation, the system needs to change his rules to adapt, thus, ensure cybersecurity properties.

The adaptive cybersecurity model is similar to the ISO/IEC 27005 Standard's PDCA (Plan-Do-Check-Act) cycle (ISO/IEC27001:2013, 2013). The four phases are a continuous process of management of security and privacy risks to the effective reaction, based on monitoring of cybersecurity metrics and adapting according to the needs, validating the strength of the models to act. This control allows analysis of the past metrics and present metrics [133].

According to Habib and Leister [128], there is a work on adaptive cybersecurity mechanisms to secure IoT. Each work proposed explores platforms and specific aspects to

improve IoT cybersecurity as well as security policies, encryption, secure communication, and intrusion detection. There is a need for IoT cybersecurity to adapt and adjust attributes when there is a change in the context. Nevertheless, the reliability and performance of adaptive cybersecurity approaches are directly related to cybersecurity mechanisms used to identify the threats to the system.

Adaptive actions are mediated by the automated process through systems to maintain a formal model (i.e., context-aware) of the settings and relationships between them [135]. Besides it, the model-driven approach can also be considered as an ontology-driven approach, but the integration of these two approaches might result in benefits of inference support of ontological approaches and the expertise of the model-driven approach. Therefore, adaptive actions are benefited from the transformation of the OWL/RDF knowledge base into domain-centric data models [136].

According to Evesti and Ovaska [137], an adaptation approach is designed following three viewpoints: adaptation viewpoint, security viewpoint, and lifecycle viewpoint. The adaptation viewpoint is used to collect adaptation features like object to adapt, adaptation time, monitoring and analysis, planning and execution, knowledge, and self-properties. The security viewpoint explores cybersecurity issues like attributes, mechanisms, protected assets, and threats. Finally, the lifecycle viewpoint consists of the aspects of software development: architecture, extensibility, reusability, flexibility, and maturity.

Figure 5.2 present some adaptation features of an adaptive approach to monitor adaption, security, and lifecycle issues of an object, and plan actions using the knowledge to ensure cybersecurity attributes. These adaptation features use some properties to the configuring, optimization, protection, healing, and awareness.

The monitoring of the object to adapt has focused on the part of the software, protocol and product information. Different types of parameters are used in the analysis to identify relevant information to the decision-making system. For example, a cryptographic algorithm used to protect the communication channel between thin devices can be modified to improve the performance aspects. These adaptations occur when the software is starting or when the software is up and running. The runtime adaptation is more dynamic, requiring different adaptation methods such as reactively and/or proactively. The reactive adaptation takes place when the system generates an alert. On the other hand, proactive adaptation uses knowledge to identify and predicts threats and potential solutions to protect the assets.

The monitoring and execution plans are designed according to the needs of the system and/or network. Moreover, which methods are more suitable to collect information from the environment to detect anomalous behavior for making adaptation decisions. The decision plan is typically not described on an appropriate level.

The security viewpoint describes the main cybersecurity components used by the process of risk management to ensure cybersecurity. In this context, related work concentrates some information based on risk management terms in ontologies [25, 111,

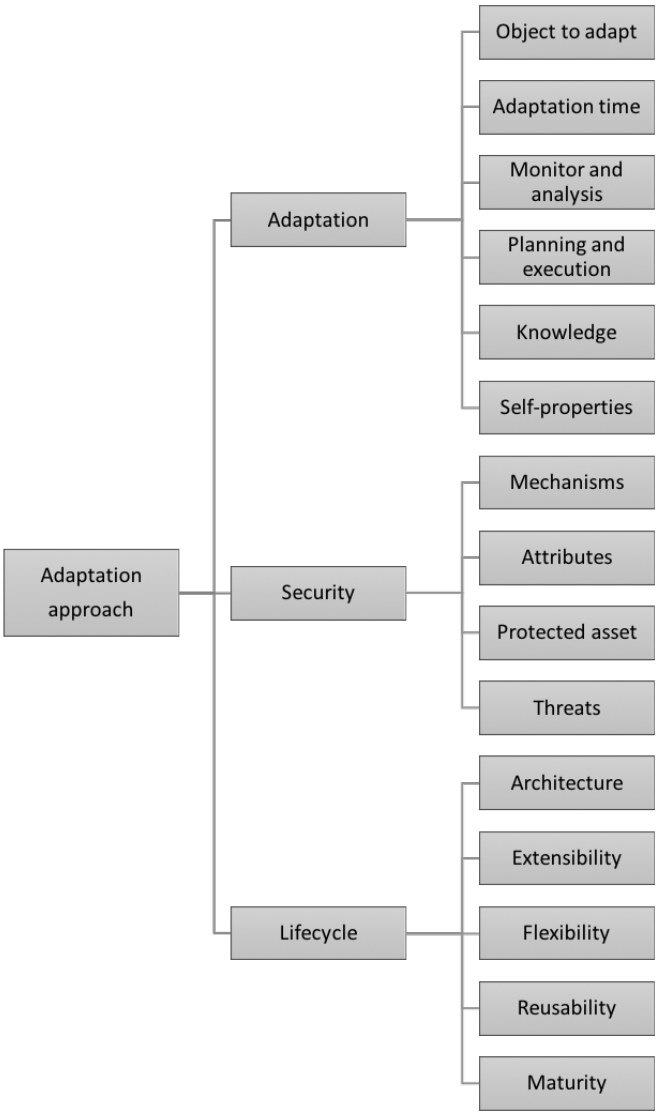


Figure 5.2: Three viewpoints of cybersecurity adaptation approaches (Adapted from [137]).

112, 115]. Several adaptation approaches manage knowledge during the decision process, but they do not offer means to manage knowledge required during the adaptation, neither the context of the necessary knowledge is described.

The lifecycle viewpoint consists of elements of the adaptation approach, and is used to execute their functionalities based on architecture inside the software or external mechanisms of software development. The extensibility is characteristic of using new components to reply to sophisticated analysis. Sometimes, security mechanisms are reused with little changes, and the reusability features of the adaptation approach can easily adjust small modifications. These changes, considered specific or generic solutions, also can be applied in different environments the adaptation architecture dictates how flexible the solution is.

A reactive approach provides adaptability based on a present situation of the

context, but due to the characteristics of the smart environments, the systems need to foresee undesired future events and make decisions of opportunities using prediction technologies. The main contribution of these environments is to incorporate the current context information further proactive behavior to predict future situations. Seven underlying principles guide proactive system design: connecting with the physical world, deep networking, macro-processing, dealing with uncertainty, anticipation, closing the control loop, and making systems personal [139, 140].

According to the proactive event-driven computing proposed by Engel et al. [141] to decide future action based on events using the pattern "detect-forecast-decide-act" illustrated in Figure 5.3. The "Detect" pattern consists of processing current events to indicate future problems. "Forecast" pattern represents events that need to be mitigated based on detected events. "Decide" pattern is the best action to mitigate an event forecasted. This pattern requires real-time decision-making capabilities with or without human intervention. By the end, "act" pattern works when an action is chosen, then an explicit distinction between consumers and actuators to adapt compatible with its needs.

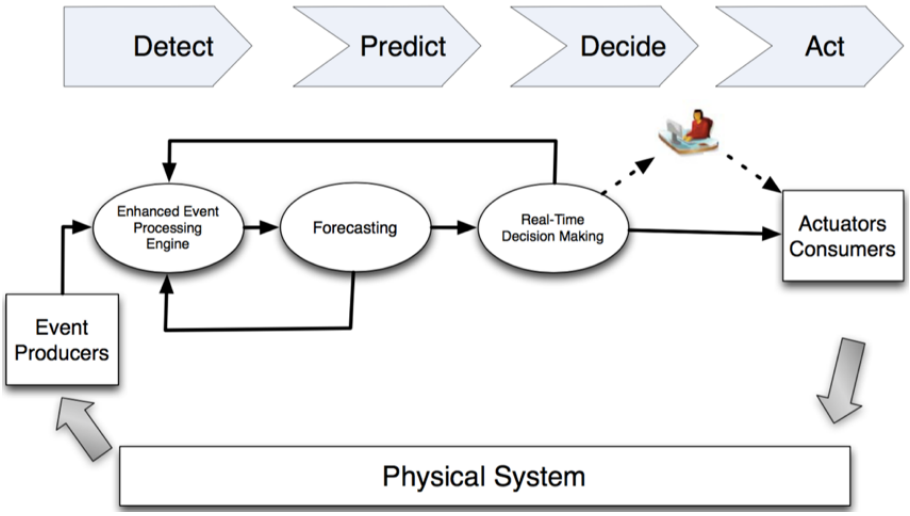


Figure 5.3: Proactive event-driven systems [141].

The decision-making systems always try to act in the current situation. Based on a proactive approach, the foresee pattern allows us to project future situations with detected events and face to a knowledge base and previous data to determine the next actions.

5.2 The Proposed Cybersecurity Framework for the IoT

As expressed in our hypothesis, a cybersecurity framework is an essential requirement for the complete adoption of the Internet of Things by industry, academia and domestic stakeholders. For that, it is needed to identify what are the security-relevant capabilities of the IoT devices in order to be connected and correctly used. Cybersecurity practices

are vital to business and comprise a factor that has not yet been taken into consideration by companies as part of their risk management process [126].

This dissertation describes an ontology-based cybersecurity framework focused on the security aspect of the Internet of Things. The framework focuses on the company-side monitoring and service provisioning to improve cybersecurity aspects around business processes and technology assets with negative impacts over the socio-economic perspectives. This Chapter presents a novel approach to improve the IoT cybersecurity focusing on the enterprise (company-side) monitoring, analysis and classification of security vulnerabilities in a knowledge base, while enabling the subsequent security service provisioning adjusted to the threats, hence improving security mechanisms around business processes and technology assets. For that purpose, the framework has three layers to deal with cybersecurity at design and run time, respectively, and an integration layer used by both (see Figure 5.4).

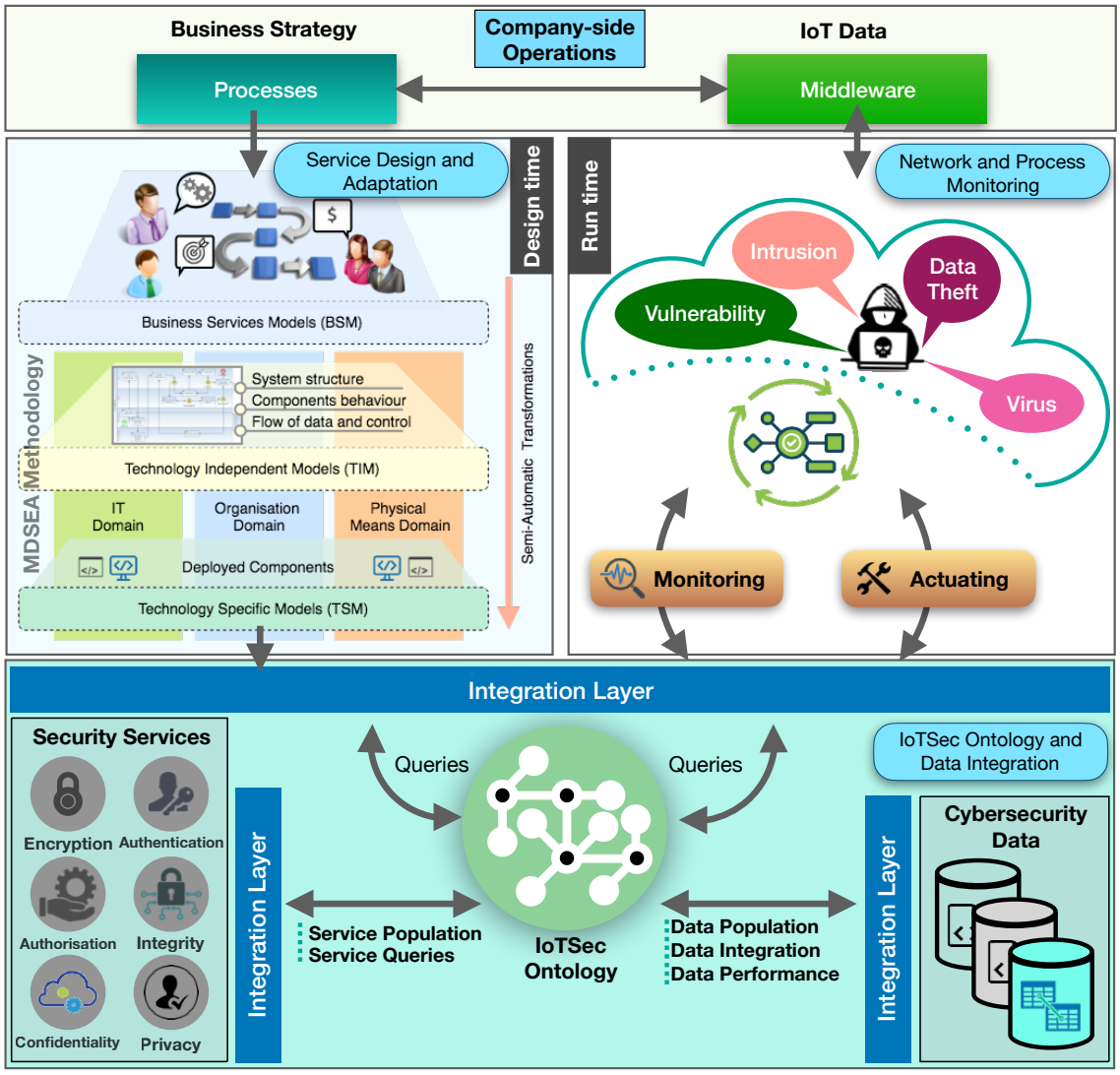


Figure 5.4: The proposed ontology-based cybersecurity framework.

At design time (top left side of the figure), the framework foresees the application of the Model Driven Service Engineering Architecture (MDSEA) model-driven methodology to build and adapt existing security services semi-automatically, reusing a high-level of abstraction security service specifications in the development of technology-specific components. At run time (top right side of the figure), network and process monitoring mechanisms collect security alerts from different cybersecurity tools identifying and classifying situations of interest (e.g., threats and vulnerabilities) in a knowledge base formalized by the IoTSec ontology (integration layer, the bottom part of the figure). Using such knowledge and its reasoning mechanisms, the ontology is able to propose suitable security services, which can or not be the ones specified at design time, adapting and actuating within IoT environments.

The company-side operations involve several business processes, which require data collection from IoT devices and sensors. Usually, these processes aim to perform parallel activities to achieve specific goals defined by the company in their business plan. The cybersecurity framework integrates the IoTSec ontology and data integration from distinct data sources into a knowledge base. This block provides data integration and population from the ontology information and access to various security services regarding several business processes and network devices, requirements, ensuring security mechanisms against threats. The following sections describe these blocks in detail.

5.2.1 Service Design and Adaptation: Design Time Layer

Security services provide different types of protection for all company-side operations. Indeed, a system composed of IoT sensors only can be considered secure when security services ensure the different security properties (e.g., confidentiality, integrity, availability).

At design-time, a company decides to implement different and specific services to address its business processes needs or adapt existing services already in place. This decision requires adaptations to attend device constraints such as the implementation of end-to-end security in the IoT, for which compressed IPsec provides a lower header overhead than link-layer security [152].

Using the run time layer modules and IoTSec, the cybersecurity framework supports minor changes in services such as the change of protocols or algorithms already designed. However, in the case of more profound adaptations that require new algorithms or protocols' development, the design time layer is required, applying the MDSEA methodology. The MDSEA supports the full-service life cycle by generating code from high-level abstraction specifications, hence accelerating the service design, adaptation, and deployment time. With this approach, business users such as company managers can collaborate with developers and participate in the specification of the necessary security mechanisms.

The service design and adaptation is a block of the proposed framework to perform model transformations from a high-level business of abstraction to code artifacts to deploy new services according to environment requirements [153]. This approach considers distinct cartridges to make deployments on heterogeneous technologies. This block provides IoTSec a pool of security services (e.g., confidentiality, privacy, authorization, encryption, integrity, and authentication) to be used in the network and process monitoring block.

Each service is composed of a set of security mechanisms specifically designed to deal with each requirement. For instance, a security service for the encryption component requires the use of mechanisms, which provide data encryption/decryption. Such mechanisms include the use of secure protocols such as the Secure Socket Layer for secure communications.

5.2.2 Network and Process Monitoring: Run Time Layer

The network and process monitoring block provides two complementary methods to explore the environment: monitoring and actuation. The first method relies on the monitoring of situations of interest that are being analyzed in each environment. It includes the detection process of possible intrusions, data theft, virus, ransomware. Monitoring tools offer information regarding different types of security alerts that are then investigated using distinct security tools, such as firewalls, intrusion detection systems, vulnerability scanners.

Each situation is then analyzed to identify whether there are suitable solutions (from the pool of security services in IoTSec) that can be applied in that specified moment to recover the system and improve cybersecurity. The second method focuses on such adaptations, i.e., to prevent threats in the business processes according to the security analysis results from the knowledge base. It consists of applying appropriate mechanisms or changing particular protocols to avoid detected security threats again.

This way, the run time layer of the proposed framework is responsible for identifying and classifying known threats in a knowledge base to offer appropriate security service to prevent future occurrences. A collection of monitoring tools provides information about intrusion detected in the environment and generates alerts using the IDMEF standard. When a situation of interest is identified, it is responsible for registering and classifying it within IoTSec according to its particular characteristics.

5.2.3 IoTSec Ontology and Data Integration Layer

The data integration layer provides cybersecurity information (e.g., threats and vulnerabilities) using the IoTSec ontology [154] with a pool of security already existing or generated using the design time layer and the MDSEA methodology. There are distinct cybersecurity data sources available that are included using an integration layer to provide data population, data integration, and data performance from the queries on

the ontology. Currently, the IoTSec ontology consists of certain statements, which are summarized in Table 5.1.

Table 5.1: Number of classes, properties, axioms and annotations in the IoTSec ontology.

Ontology Metric	**#**	**Ontology Metric**	**#**
Classes	228	Logical Axioms	1895
Object Properties	24	Annotations	1418
Data Properties	7	Individuals	607

The data integration layer of the proposed framework considers this knowledge base to perform the data integration with distinct data sources in a unified database. Using the capabilities reasoning, the correlation between main classes of the ontology provides implicit information to offer suitable security solutions from services available in the integration layer. Service adaptations are required in some situations, and then the framework uses the MDSEA methodology to transform a high-level of abstraction to minimize service deployment. The integration layer provides information from pre-build cybersecurity services that can be external or developed at design time from service design and adaptation.

5.2.3.1 Design Time Usage

The design time usage requires a procedure before start the proposed framework, which connects the IoTSec ontology with existing cybersecurity data sources using the Ontop framework [155] to provide data integration and population. However, after the data population, the design time layer may be applied before and after the network and process monitoring. The proposed framework provides support to the service development for situations that require generating new algorithms or services' functionalities that were not designed in the pool of security services. The design time layer tools can generate and provide new security services for the IoTSec pool.

5.2.3.2 Run Time Usage

The security monitoring capabilities are a feature of the proposed framework to identify and classify situations of interest (e.g., threats and vulnerabilities) in the environment. All monitoring tools used in the scenario generate alerts from situations of interest, such as known threats and intrusions. These alerts are analysed and classified according to the knowledge base. For each alert generated, queries are specified using the SPARQL Protocol and RDF Query Language (SPARQL) [156] to check suitable security mechanisms to protect the asset or process vulnerabilities. In this context, our implementation uses the Protégé Editor [157] to process these queries and collect results from the IoTSec knowledge base. According to the results, one or more security

mechanisms can be put in place, selected from the security services pool of the proposed framework.

5.2.4 Implementation Considerations

The implementation of our proposed ontology-based cybersecurity framework for IoT considers several technologies intending to achieve the requirement stressed in the concept. The implementation of the latter is described according to Figure 5.5: design time and run time.

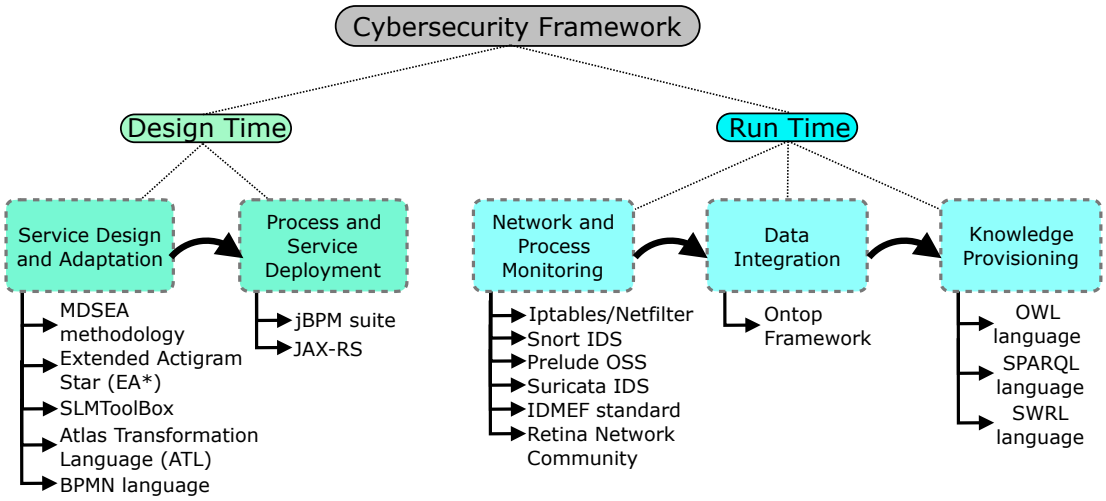


Figure 5.5: The logical relation of the application of implementation technologies and the proposed framework.

At design time, there are two steps used to build and deploy security services: service design and adaptation and process and service deployment. The first explores the Model Driven Development using the MDSEA methodology to optimize service developments [153]. The business processes are firstly designed with the Extended Diagram Star (*EA**) technology in the SLMToolBox [158]. This diagram instantiates the Business Services Model (BSM) of the MDSEA methodology and allows one to represent the regular activities flow of a specific company or network that wishes to apply the cybersecurity framework. For specific implementation of new security services (or particular functionalities), the business model is transformed (using Atlas Transformation Language) into a Technology Independent Model (TIM) in Business Process Model and Notation (BPMN) language so that developers can improve it with specific technical details about data-related operations. In this phase, any deployment aspects are still disregarded, allowing one to focus only on the functionality.

Finally, the TIM processes model is transformed to the TSM, where deployment issues are specified. The processes and service deployment step are responsible for the configuration of the jBPM suite (Kie Workbench) [159] to perform and execute specific services in the Java API for RESTful Web Services (JAX-RS). These deployments enable

the establishment of the designed security services within the business process. These services are finally made available to the knowledge base via integration of the IoTSec ontology, becoming available for future requests to address the same type of security issue.

At run time, there are three main steps responsible for the network and process monitoring, data population, and knowledge provisioning from the IoTSec ontology. The monitoring step analyses business processes and technology assets to detect threats and vulnerabilities for the IoT system. It entails specific monitoring tools to identify threats such as Iptables/Netfilter [160], Snort [161], Prelude [162], Suricata [163] and vulnerabilities such as the Retina Network Community [164]. The security alerts generated from these monitoring tools follow the Intrusion Detection Message Exchange Format (IDMEF) [27]. Alerts raised are used to classify threats and vulnerabilities in the IoTSec ontology.

The proposed framework uses SPARQL language to perform queries on the ontology to gather suitable information from potential threats in the IoT environment. The data integration step provides cybersecurity information from distinct data sources using the Ontop framework. This allows for the instantiation of a knowledge base from the IoTSec ontology.

The data population also uses Ontop to support data access through a conceptual layer rewriting the SPARQL queries (over the virtual RDF graph) to structured query language queries. Thanks to Ontop, the framework is capable of exploring the knowledge of different sources to provide data population, data integration, and data performance. Finally, the knowledge provisioning from the IoTSec ontology uses the SPARQL language to perform queries and check all information in the knowledge base. Due to language flexibility, correlations between ontology classes can be used to cross information regarding the IoT environment.

5.3 Discussion

The proposed framework to enable cybersecurity in IoT ecosystems provides the monitoring and service provisioning focused on business processes and technology assets on the company-side to minimize the impacts with the adoption of security services. The framework was designed considering the concept of adaptive cybersecurity to learn and adapt to the changing environment at run time. At the design time, the framework explores the MDSEA methodology to build and adapt existing security services based on the company-side business process. The integration with an instantiation of a knowledge base using the IoTSec ontology provides information around interest situations from alerts of security tools.

CHAPTER 6

Proof-of-Concept Implementation

6.1 Cybersecurity Framework Implementation

The implementation of our proposed ontology-based cybersecurity framework for IoT considers several technologies, intending to achieve the requirements stressed in the concept proposal to improve IoT cybersecurity. The implementation of the latter is described according to the top-down approach of each module and its proposed framework: run time, design time, pre-build services provisioning, data population, and IoTSec ontology.

6.1.1 Run Time Module

The monitoring module of the proposed framework works at run time to analyze business processes and the technology assets to identify potential threats and vulnerabilities for the IoT system. For that, specific monitoring tools were used to identify potential threats like Iptables/Netfilter [160], Snort [161], Prelude [162], Suricata [163, 165] and vulnerabilities as Retina Network Community [164]. The security alerts generated from these monitoring tools follow the Intrusion Detection Message Exchange Format (IDMEF) [27].

6.1.2 Design Time Module

The design time module uses the MDSEA methodology to expedite the software and service development from the company-side. This module uses a collection of tools to provide model transformations initializing with the language Extended Actigram Star (EA*) for the definition of a high-level of abstraction using the modeling tool SLMToolBox [158]. The use of XML metadata interchange supports some model transformations to save source and target models and mapping rules. These rules are implemented using atlas transformation language, and business models are developed in the Business Process Model and Notation (BPMN) language. After the translation process, services

implemented in Web Services (JAX-RS) are generated with little effort and reduced errors by automating model development.

6.1.3 Pre-build Services Provisioning Module

The deployment of these services is the business process management (jBPM) suite [159] to the workflow execution with service calls following the business process requirements. Also, these services are available to be adapted under knowledge support from the IoTSec ontology. The security analysis considers some assets information from the business process and IoT environment to perform queries using the SPARQL query language. Some basic needs of services are checked if they are available according to requirements or need adaptations of the design time module — for instance, an exchange of a protocol to supply a device constraint.

6.1.4 Data Population for the IoTSec ontology

The data population of the IoTSec ontology, the proposed framework chose the Ontop framework [155] to support data access through a conceptual layer rewriting the SPARQL queries (over the virtual RDF graph) to Structured Query Language queries. The framework explores the knowledge of different sources to provide data population, data integration, and data performance.

6.2 Ontology Assessment

The assessment of our ontology covers the cybersecurity domain for concrete tasks focused on improving the security of basic components present in risk analysis through the use of a framework with service provisioning against potential vulnerabilities and threats. The ontological model facilitates the expandability of the reference ontology in cybersecurity [25, 166, 167]. However, this analysis follows a quantitative evaluation of the IoTSec ontology based on SQuaRE standards for software quality evaluation.

The adopted methodology for ontology evaluation was designed with an adaptation of a software engineering standard called the OQuaRE framework [168, 169, 170]. This proposal was presented to help developers to identify weaknesses and strengths using a series of quality characteristics for ontologies according to the SQuaRE standard. The model reuses and adapts the following SQuaRE characteristics to ontology evaluation, namely, structural, functional adequacy, adaptability, reliability, transferability, maintainability, and operability. The ontology evaluation of a particular characteristic depends on the evaluation of its set of associated sub-characteristics. Similarly, a particular sub-characteristic depends on its associated metrics. The application of this assessment methodology generates the scores for the IoTSec ontology using OQuaRE metrics. Table 6.1 presents the description of OQuaRE metrics.

Table 6.1: Detail of all metrics proposed from OQuaRE methodology.

Metric	**Description**
Lack of Cohesion in Methods (LCOMOnto)	Measure the separation of responsibilities and independence of components of ontologies.
Weighted Method Count (WMCOnto)	Number of properties and relationships per class.
Depth of Subsumption Hierarchy (DITOnto)	Length of the largest path from Thing to a leaf class.
Number of Ancestor Classes (NACOnto)	Number of ancestor classes per leaf class.
Number of Children (NOCOnto)	Number of direct subclasses.
Number of properties (NOMOnto)	Number of properties per class.
Properties Richness (RROnto)	Number of properties defined in the ontology divided by the number of relationships and properties.
Relationships per class (INROnto)	Number of relationships per class.
Class Richness (CROnto)	Number of instances per class.
Tangledness (TMOnto)	Number of parents per class.

The methodology considers each pair (sub-characteristic, metric) to give a value between 1 (lowest) and 5 (highest) [168]. However, one metric may be associated with multiple sub-characteristics to provide the relevance and usefulness of the quality ontology assessment.

The *Structural* characteristic consists of the formal and semantic ontological properties that are widely used in state-of-the-art evaluation approaches. The sub-characteristics are formalization, formal relations support, cohesion, tangledness, redundancy, and consistency. The score of this characteristic was 4.25 according to quality metrics demonstrated that it represents a complete cohesion with a good domain coverage. The cohesion sub-characteristic is an excellent indication to help the identification of areas with more instances to be more closely connected. It reflects the knowledge base as a result of extracting data from separate sources. Also, the sub-characteristics present the right consistency as the quality of our ontology. Besides that, the relation of the number of properties and relationships presents a low value for the formal relations support, which can be improved using inference rules to ensure better formal relations.

The *Functional Adequacy* characteristic follows specific criteria according to the degree of accomplishment of functional requirements over different purposes. This includes reference ontology, controlled vocabulary, schema, and value reconciliation, consistent search and query, knowledge acquisition, clustering and similarity, indexing and linking, results representation, classifying instances, text analysis, guidance, and decision trees,

knowledge reuse, inferencing, and precision. The score of this characteristic was 4.05, but two metrics were not considered to the evaluation that considers the classes' attributes and annotations. As for strengths, the evaluation presents consistent search and query and knowledge reuse considering the mean number of relationships associated, number of properties per class, and length of the path from the leaf classes to the *Thing*. Also, the metric "mean number of properties per class" collaborate with knowledge acquisition because it means that probably this ontology is more useful. However, a sub-characteristic demonstrated a weakness associated with the number of instances. This aspect has no impact on the evaluation because the ontology requires an entire data population for an application in the real world.

The *Adaptability* characteristic consists of the degree to which the ontology can be adapted for different specified environments without conducting actions other than those identified for the ontology considered. The score of this characteristic was 3.67, which is acceptable for the framework requirements. The metric of the number of properties and relationships is an essential factor in providing adaptability. This measure allows creating a better understanding of how focal some classes work. Hence, the number of relationships reflects, for instance, a grouping within a class based on what they have with other instances. However, a sub-characteristic affects the purpose of adapting the ontology for distinct environments due to the most extensive path from the *Thing* to a leaf class.

The *Reliability* characteristic matches the ontology maintenance of the level of performance under stated conditions for a given period. The score of this characteristic was 3.88. Recoverability and availability are some of its sub-characteristics. The metric of the maximum depth of the hierarchy tree from *Thing* to a leaf class by the total number of paths directly influences the availability sub-characteristic.

The *Transferability* characteristic presents the degree to which the software product can be transferred from one environment to another. The score of this characteristic was 4. The metrics of the number of properties and direct subclasses allow the ontology to adapt quickly in another context. However, the metric of the length of the most extensive path affects the recoverability sub-characteristic because the higher its score, the lower is the probability to recover.

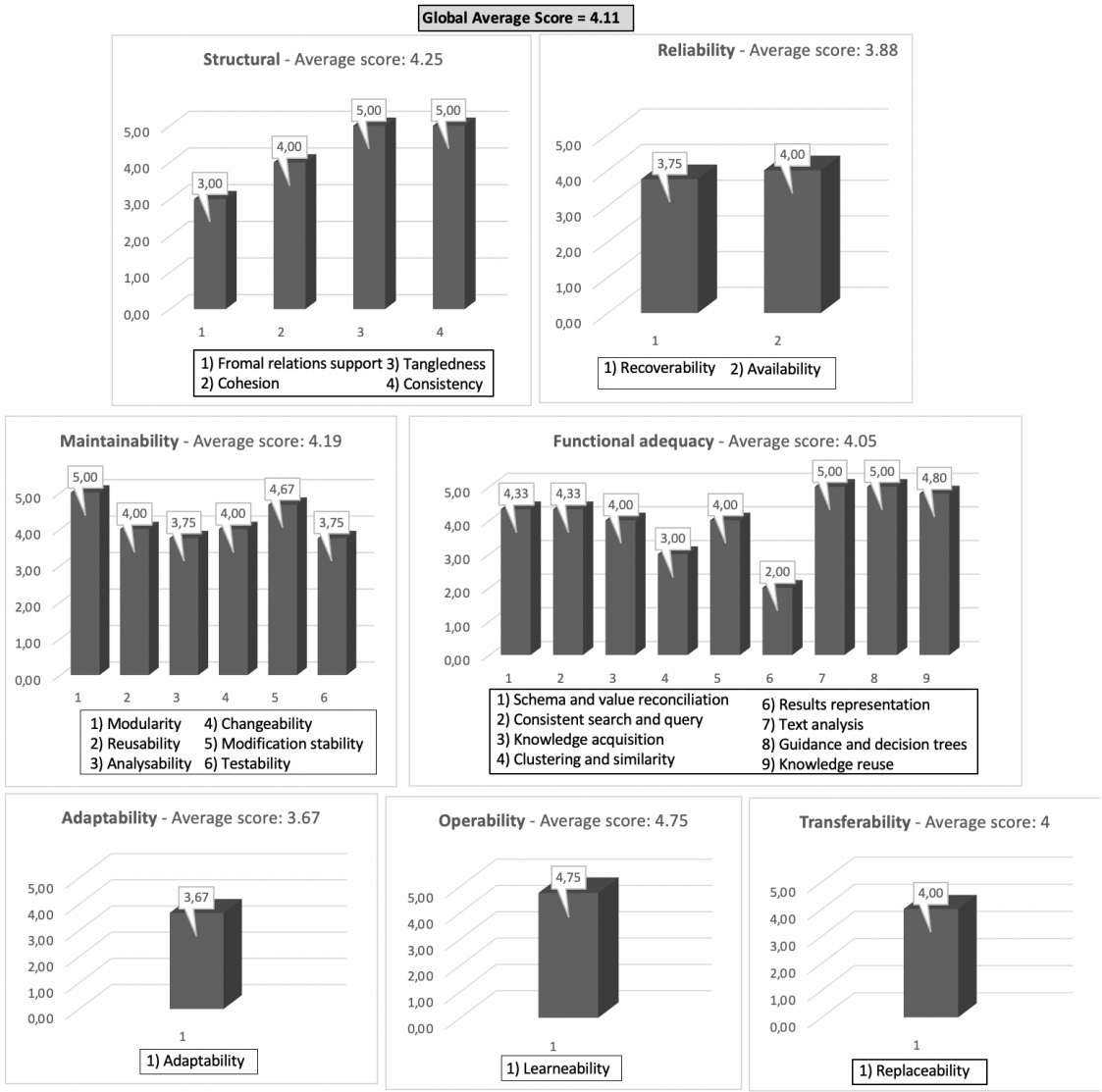
The *Maintainability* characteristic provides the ability of ontologies to adapt to changes in the environment, in terms of requirements or functional specifications. Some sub-characteristics are modularity, reusability, analysability, changeability, modification stability, and testability. The score of this characteristic was 4.19. The number of properties also impacts on reusability (of maintainability) because having a more precisely defined ontology makes its knowledge more reusable.

The *Operability* characteristic harmonizes the knowledge base necessary to use an ontology, and in the individual assessment of such use, by a single or a set of users. It is measured through the learnability sub-characteristic. The score of this characteristic was 4.75. The parameter of the number of the properties per class reflects in the schema

used at the instances level. This metric is a good indication of how well is the use of information in the extraction process. However, the maximum depth of the hierarchy tree from *Thing* to a leaf class minimizes the effectiveness.

Figure 6.1 presents these automatic scores using a series of quality characteristics for ontologies.

Figure 6.1: Complete automatic score for IoTSec ontology using OQuaRE metrics.



The results presented above demonstrate the strengths and weaknesses of the IoTSec ontology. It can be observed that the global score has a value of 4.11. In our case, some sub-characteristics were not accounted for the evaluation, such as the properties that can be directly accessed from the class and the mean number of attributes per class, which are not considered in our ontology assessment. Therefore, particular quality metrics have affected some characteristics; the general quality of our ontology is acceptable following the criteria of human evaluation; some improvements could produce a better outcome.

6.3 Validation Case Study: Industrial Scenario

In this section, a case study is used to validate the proposed cybersecurity framework for the IoT. This framework uses IoTSec ontology as an intelligent support system to improve the cybersecurity of the environment. The manufacturing scenario of this industrial case study is a pilot from the EU C2NET project, which focused on improving the supply network optimization of manufacturing and logistic assets based on collaborative aspects, production, and delivery plans. It aims to use smart devices to improve the production of product parts from different third-party organizations. The industrial scenario contains several sensors to provide IoT-based continuous data collection from supply network resources of the factory shop floor in a company of the metalworking industry, as depicted in Figure 6.2.

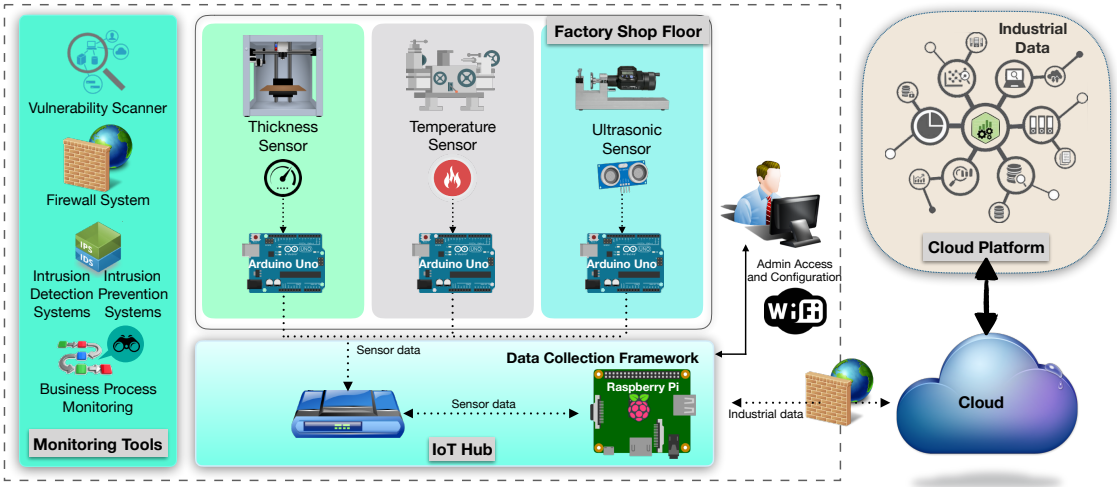


Figure 6.2: Industrial scenario in a factory shop floor.

The IoT hub represented at the bottom refers to a device that performs the data management to be processed in the cloud platform using several cloud-based tools for supporting the supply network optimization and logistic assets based on collaborative demand. As a contribution, this cloud platform provides new ways to store relevant information securely from supply network partners in a public cloud. The cybersecurity management of this scenario offers monitoring capabilities to detect and prevent threats within the shop floor networked environment.

6.3.1 Process/Environment Monitoring

There are potential cybersecurity vulnerabilities which malicious users could take advantage. These include unprotected communication channels, lack of an access control system, a private channel for sensitive information, and so forth. Furthermore, these technologies, as mentioned earlier, contain known threats, as well as security mechanisms that are essential to ensure the presence of a secure environment. The Raspberry Pi 3 is

one IoT device used in this industrial scenario (within the IoT hub) that has a variety of threats and vulnerabilities to be exploited by malicious users in the form of threats and other security hazards.

Some monitoring tools are responsible for identifying abnormal situations and generating security alerts such as the IDS, firewall system, and vulnerability scanner. These alerts provide information to be analyzed in the proposed framework. As each tool has a particular specification, in this work, the IDMEF format was adopted to make the interoperability between the monitoring tools and the framework. This security alert standard contributes to the integration layer of the framework, which offers an abstraction way to check solutions according to the knowledge representation of the IoT ontology.

6.3.2 Ontology Usage for Monitoring

The proposed cybersecurity framework uses the IoTSec ontology and knowledge base to find suitable solutions and information over services according to generated alerts. Besides, the ontology uses the reasoning capabilities to identify the knowledge base uniformity, correctness of data instances, and assertions using rules. This process derives implicit facts from the existing knowledge and can be classified into logic-based context reasoning, rule-based reasoning, or deductive and inductive reasoning.

The definition of inference rules is established with the Semantic Web Rule Language (SWRL)[124] with the Protégé editor using the reasoner Pellet [125] to make the rule processing. The reasoner manipulates the ontology logic using inference rules to reason with individuals, user-defined data types, and debugging support for ontologies.

R1: `hasPart(?x,?y), hasPart(?y,?z)` → `hasPart(?x,?z)`

R2: `isSecurityMechanismOf(?sm,?t), threatens(?t,?v)` → `mitigates(?sm,?v)`

R3: `protects(?sm,?a), requires(?a,?sp)` → `satisfies(?sm,?sp)`

R4: `mitigates(?sm,?v), threatens(?t,?v)` → `isSecurityMechanismOf(?sm,?t)`

R5: `SecurityMechanism(?sm), SecurityProperty(?sp), Threat(?t), affects(?t,?sp), isSecurityMechanismOf(?sm,?t)` → `satisfies(?sm,?sp)`

These inference rules presented above are methods, which allow new facts to be found from implicit knowledge. A graphical representation of a specific rule demonstrates the functionality to the knowledge base when the rule has a set of axioms valid. It fills an axiom that was implicit using true affirmation, such as X is a security mechanism of Y, in case of the axiom `isSecurityMechanismOf(?sm,?t)`. The inference rule R5 establishes a new relationship when a set of axioms fulfills the rule requirements. Figure 6.3 represents this relation according to the rule.

Given the graphical representation of the inference rule R5, the object property `isSecurityMechanismOf(?sm,?t)` provides the ability to link `SecurityMechanism(?sm)` and `Threat (?t)`, and `affects(?t,?sp)` for linking between `Threat(?t)` and `Security`

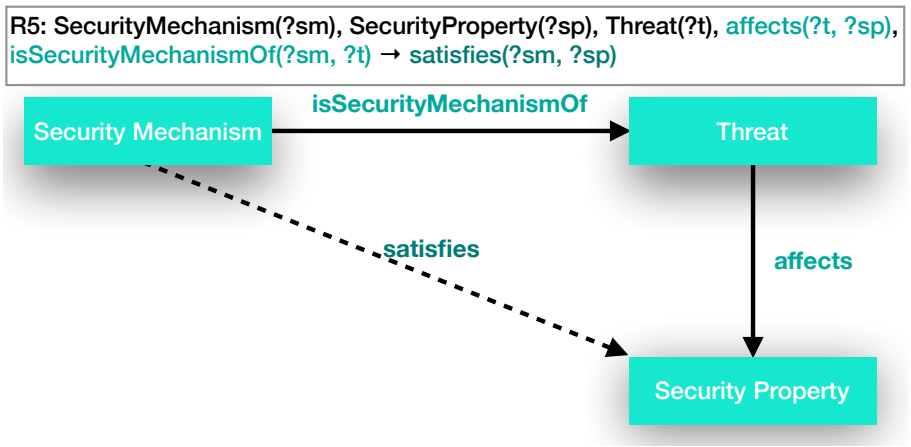


Figure 6.3: Graphical representation of the inference rule R5.

`Property(?sp)`, respectively. Then, this association enables to discover implicit facts from structured knowledge in the object property `satisfies(?sm,?sp)`.

The knowledge reasoning can infer in several cases, discovering relations among assets, vulnerabilities, threats security properties, and security mechanisms. In the industrial scenario, several threats could violate information privacy due to bypassing security mechanisms. One of the common threats of industrial networks is the protocol used in a wireless LAN (WiFi protocol), especially for systems hosting valuable information. In this case, the best way to implement an adequate level of security mechanism within a business organization is to utilize collaborative planning at a technical and organizational level with coordinated measures to ensure appropriate protection strategies. This is further stressed by the fact that single security mechanisms and separated tools cannot guarantee cybersecurity in a global setting. Attackers often convey threats in different ways, often categorized according to architectural layers. Each security threat affects one or more security requirements, and the system is protected using specific security countermeasures.

Figure 6.4 presents the application of this rule in Protégé editor to produce inference results using the reasoner Pellet. The reasoning results from the rule R5 presents new facts (dotted line) regarding the object property `satisfies` according to the goal of the rule R5.

As reasoning results, the proposed framework provided a suitable security mechanism to prevent threats to the WiFi protocol called Wi-Fi Protected Access, version 2 (WPA2) that satisfies some instances of the *Security Property* class, ensuring protection of *Authentication*, *Confidentiality* and *Integrity* (these instances have distinct nomenclatures defined). According to the data collection, the system checks the knowledge base using SPARQL [156] to identify suitable security mechanisms. SPARQL queries are used to obtain valuable knowledge of security attributes and individuals of the situation of interest in the IoT environment.

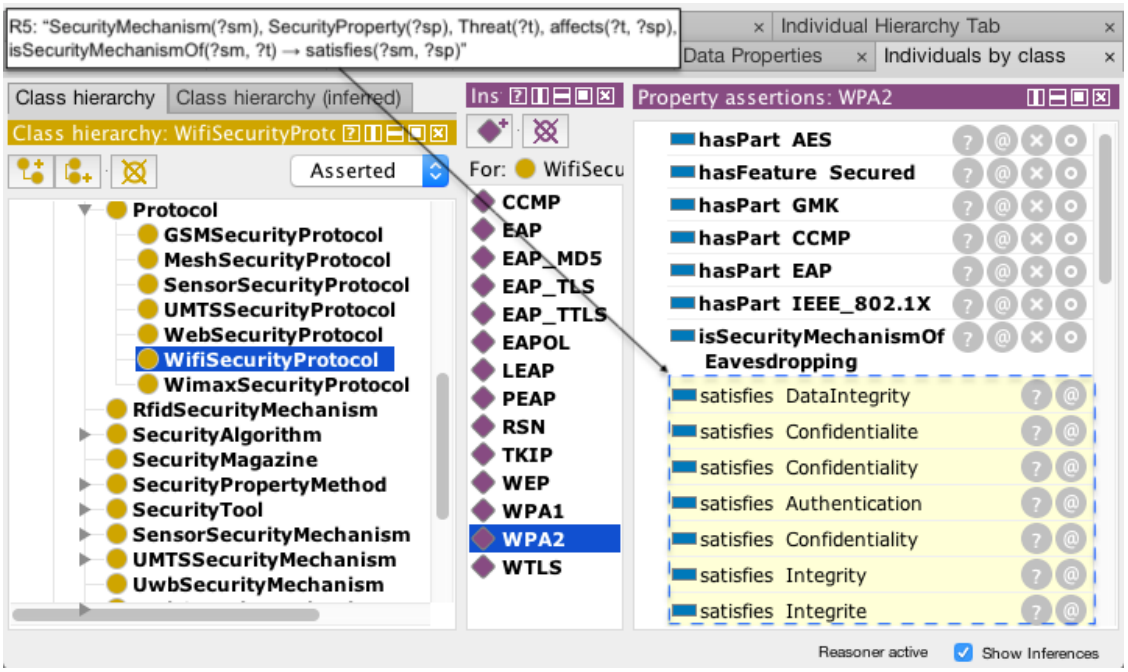


Figure 6.4: Results from the application of an inference rule.

Within the scenario implemented, the proposed framework checked the knowledge base using SPARQL queries (Listing 6.1) to provide information about a situation of interest at that moment.

List 6.1: An example of query of the proposed framework.

```
1    SELECT ?ASSET ?VULN ?THREAT ?SECPROP ?SECMEC_1 ?FEATURE_1
2    WHERE {
3    ?VULN iotsec:isVulnerabilityOf ?ASSET .
4    ?VULN iotsec:isThreatensBy ?THREAT .
5    ?THREAT iotsec:affects ?SECPROP .
6    ?SECMEC_1 iotsec:isSecurityMechanismOf ?THREAT .
7    ?SECMEC_1 iotsec:hasFeature ?FEATURE_1 .
8    ?SECMEC_1 rdfs:label ?SMLabel .
9    FILTER regex (?SMLabel, 'WEP')
10 ␣␣␣}
```

Listing 6.1 presents an SPARQL query with the association among vulnerabilities (`?VULN`), assets (`?ASSET`), threats (`?THREAT`), security property (`?SECPROP`), security mechanisms (`?SECMEC_1`), and features (`?FEATURE_1`) classes in Lines 3-8. Also, this query filters (Line 9) the results to find only labels with the expression "WEP". This vulnerability of the WEP protocol exposes weaknesses and requires suitable security solutions offered by the proposed framework.

Once a security issue in the environment is detected, the framework requests relevant information from this particular situation to the IoTSec ontology. The security alert

reported the vulnerability *Unauthorized Access*, which affects two security properties, the *Confidentiality* and *Integrity* of the WiFi technology. As a consequence, several vulnerabilities could be exploited with the *Eavesdropping* threat.

Figure 6.5 presents the results from the formal question to the cybersecurity framework to identify alternative security protocols, which are suitable solutions to be deployed in this scenario. The query identified features of security mechanisms to offer on the environment. It only filters the results regarding its features to show an alternative mechanism with status *Deprecated*.

ASSET	VULNERABILITY	THREAT	SECURITYPROPERTY	SECMEC_1	FEATURE_1	SM_2	FEAT_2
WiFi	UnauthorizedAccess	Eavesdropping	Authentication	WEP	Deprecated	WPA1	Deprecated
WiFi	UnauthorizedAccess	Eavesdropping	Authentication	WEP	Deprecated	WPA2	Secured

Figure 6.5: Results from formal question to the cybersecurity framework.

When implemented in this specific scenario, the query focused on vulnerable assets such as WiFi vulnerability, to provide a better level of security. A set of 30 executions were performed with a different number of instances involved to evaluate the computational costs during the execution of the queries. Figure 6.6 presents the average time of the SPARQL queries processing. This evaluation shows that the processing time of queries increases according to the number of results. The execution time rises considerably with over 500,000 instances. However, at 300 ms, the execution time does not appear to pose any problems to the industrial adoption of the IoTSec solution.

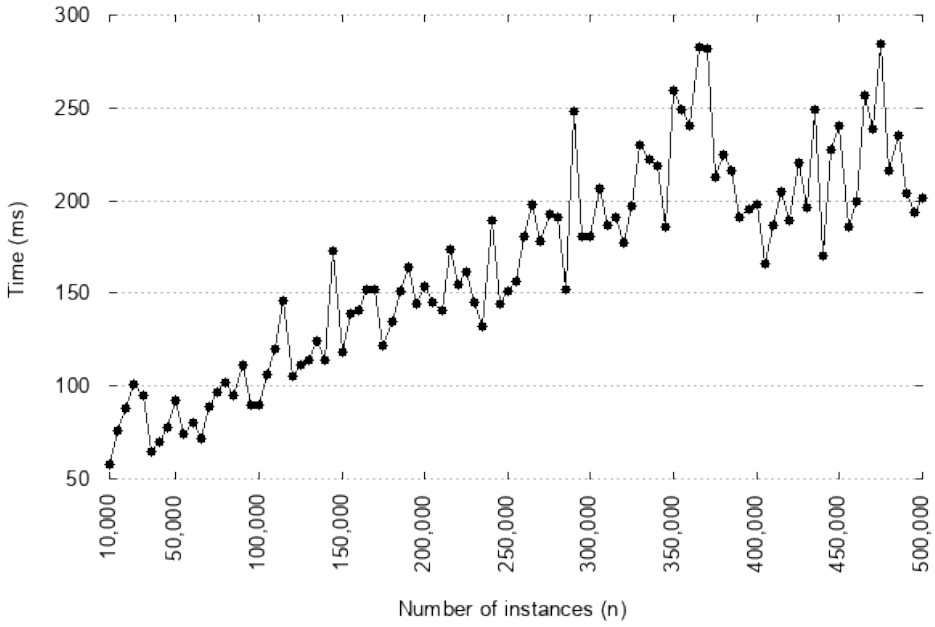


Figure 6.6: Processing time of a query with n instances in the results.

6.3.3 Service Adaptation and Actuation on the IoT Network

Based on the SPARQL query, the ontology identified a service with a security mechanism: WPA2 security protocol. This query presents protocol features from a previously used protocol (continuous line) and a suitable solution (dotted line) using the object property *hasFeature* between classes *SecurityMechanism* and *Feature*.

In this context, the framework suggests the change of the wireless security protocol for secure communication over the computer network. Considering this, the proposed cybersecurity framework requires an adaptation of the wireless communication service provided by the service design and adaptation block. The adaptation of this service requires the instantiation of the WPA2 security protocol. This service was previously designed and instantiated with the WEP protocol at design time. Since this adaption is based on a mechanism only available in the repository of security services, the suggested solution does not require the generation process of a new service using the MDSEA.

The adaptation and actuation of this service provided a reconfigurable capability that allowed the security protocol to be changed in routers and IoT devices. This security protocol uses different stream ciphers and encryption schemes that offer better protection against unauthorized access. When implemented, the security improvement of this industrial scenario solved the vulnerability of the WEP protocol as authentication forging, man-in-the-middle attacks, and brute-force dictionary. This adaptation provided a much higher level of cybersecurity for IoT users and applications.

6.4 Discussion

The main research question was defined in Section 1.4 with the corresponding hypothesis. Considering the followed proof-of-concept implementation, it can be concluded that the hypothesis has been positively validated.

The ontology assessment provides an overall evaluation of the OQuaRE methodology used to identify weaknesses and strengths using a series of quality characteristics for ontologies based on the SQUARE standard. A particular sub-characteristic depends on its associated metrics. The associating of sub-characteristics with metrics is fundamental to understanding the framework. A particular metric contributes to multiple quality properties such as "mean number of properties per class“ that contribute to knowledge acquisition because an ontology with more properties is usually more useful. It also impacts reusability because having a more precisely defined ontology makes its knowledge more reusable.

The validation by an industrial scenario with several sensors to provide IoT-based continuous data collection from supply network resources of the factory shop floor in a company of the metalworking industry was accomplished to confirm the proposed solutions' general fitness. Here the proposed approaches were globally assessed, highlighting the cybersecurity improvements with the framework that supports both

design time and run time using the knowledge base (IoTSec ontology) of the integration layer. The execution time does not appear to pose any problems to the industrial adoption of the IoTSec solution.

CHAPTER 7

Implementation Assessment and Hypothesis Validation

In this chapter, the validation of the hypothesis presented in this dissertation is assured in two ways. This chapter begins with the scientific validation by presenting the list of published articles, and then by explaining the work done in the industrial validation. Finally, hypothesis validation is the objective of this work.

7.1 Acceptance of Scientific Community

During the research and developments presented, five scientific publications were submitted to a journal and six scientific publications in conference proceedings, below is the list of papers:

- B. A. Mozzaquatro, R. Jardim-Goncalves, and C. Agostinho, "Towards a Reference Ontology for Security in the Internet of Things", in 2015 IEEE International Workshop on Measurements and Networking (M&N) Proceedings, pp. 117–122. 2015, Coimbra - Portugal, October 12-13. [25].

- A. Kozakevicius, C. Cappo, B. A. Mozzaquatro, R. C. Nunes, and C. E. Schaerer. "URL Query String Anomaly Sensor Designed with the Bidimensional Haar Wavelet Transform". In: International Journal of Information Security 14.6 (2015), pp. 561 - 581. ISSN: 1615-5270. DOI: 10.1007/s10207-015-0276-y. [171].

- G. R. Librelotto, L. O. Freitas, A. Fiorin, B. A. Mozzaquatro, L. Pasetto, R. G. Martini, R. P. Azevedo, and R. T. Pereira. "OntoHealth: A System to Process Ontologies applied to Health Pervasive Environment". In: International Journal of Computational Science and Engineering 10.4 (2015), pp. 359–367. [172]

- B. A. Mozzaquatro, R. Melo, C. Agostinho, and R. Jardim-Goncalves, "An Ontology-based Security Framework for Decision-making in Industrial Systems", in Model-Driven Enterprise Services and Applications for a Sustainable Interoperability: New Paradigms for Development in the Future Enterprise - MDE4SI, 2016, pp. 779–788. Rome - Italy, February 19-21. [166]
- B. A. Mozzaquatro, R. Jardim-Goncalves, R. Melo, C. Agostinho, R. Melo, and R. Jardim-Goncalves, "The Application of Security Adaptive Framework for Sensor in Industrial Systems", in 2016 IEEE Sensor Application Symposium, 2016, Catania - Italy, April 20-22. [173]
- B. A. Mozzaquatro, C. Agostinho, R. Melo, and R. Jardim-Goncalves, "A Model-Driven Adaptive Approach for IoT Security", in Model-Driven Engineering and Software Development: 4th International Conference, MODELSWARD, Rome - Italy, February 19-21, 2016, Revised Selected Papers, S. Hammoudi, L. F. Pires, B. Selic, and P. Desfray, Eds. Springer International Publishing, 2017, pp. 194–215. [167]
- B. A. Mozzaquatro, R. Jardim-Goncalves, and C. Agostinho, "Model Driven Implementation of Security Management Process", in Proceedings of the 5th International Conference on Model-Driven Engineering and Software Development (MODELSWARD), pp. 229–238. 2017, Porto - Portugal, February 19-21. [174]
- B. Mozzaquatro, R. Jardim-Goncalves, and C. Agostinho, "Situation Awareness in the Internet of Things", 23rd ICE/IEEE International Conference on Engineering, Technology and Innovation - International Technology Management Conference, pp. 982–990, 2017, Madeira - Portugal, June 27-29. [153]
- J. C. Nobre, B. A. Mozzaquatro, and L. Z. Granville, "Network-Wide Initiatives to Control Measurement Mechanisms: A Survey", IEEE Communications Surveys and Tutorials, 2018. [175]
- B. Mozzaquatro; C. Agostinho; D. Goncalves; J. Martins and R. Jardim-Goncalves. "An Ontology-Based Cybersecurity Framework for the Internet of Things". Sensors. 2018, 18, 3053. [176]
- J. A. Bonini, M. D. Silva, R. Pereira, B. A. Mozzaquatro, R. G. Martini, G. R. Librelotto. "An Application of Ontological Engineering for Design and Specification of Ontocancro". 14th International Conference Practical Applications of Computational Biology & Bioinformatics (PACBB 2020), 2020, p. 134-143. [177]

7.2 Industrial Validation

The viability validation of the proposed IoT cybersecurity framework in this work, which was necessary to implement, have been divided by several validations, between academic

and industrial validations. This dissertation was developed integrating the Group for Research in Interoperability of Systems (GRIS) at UNINOVA and in the C2NET EU H2020 project[1].

The C2NET project aims to create cloud-enabled tools for supporting the SMEs supply network optimization of manufacturing and logistic assets, based on collaborative demand, production, and delivery plans. The C2NET Project results and its use in a collaborative and mobile value chain, for supporting intra-plant and extra-plant processes, are depicted in Figure 7.1:

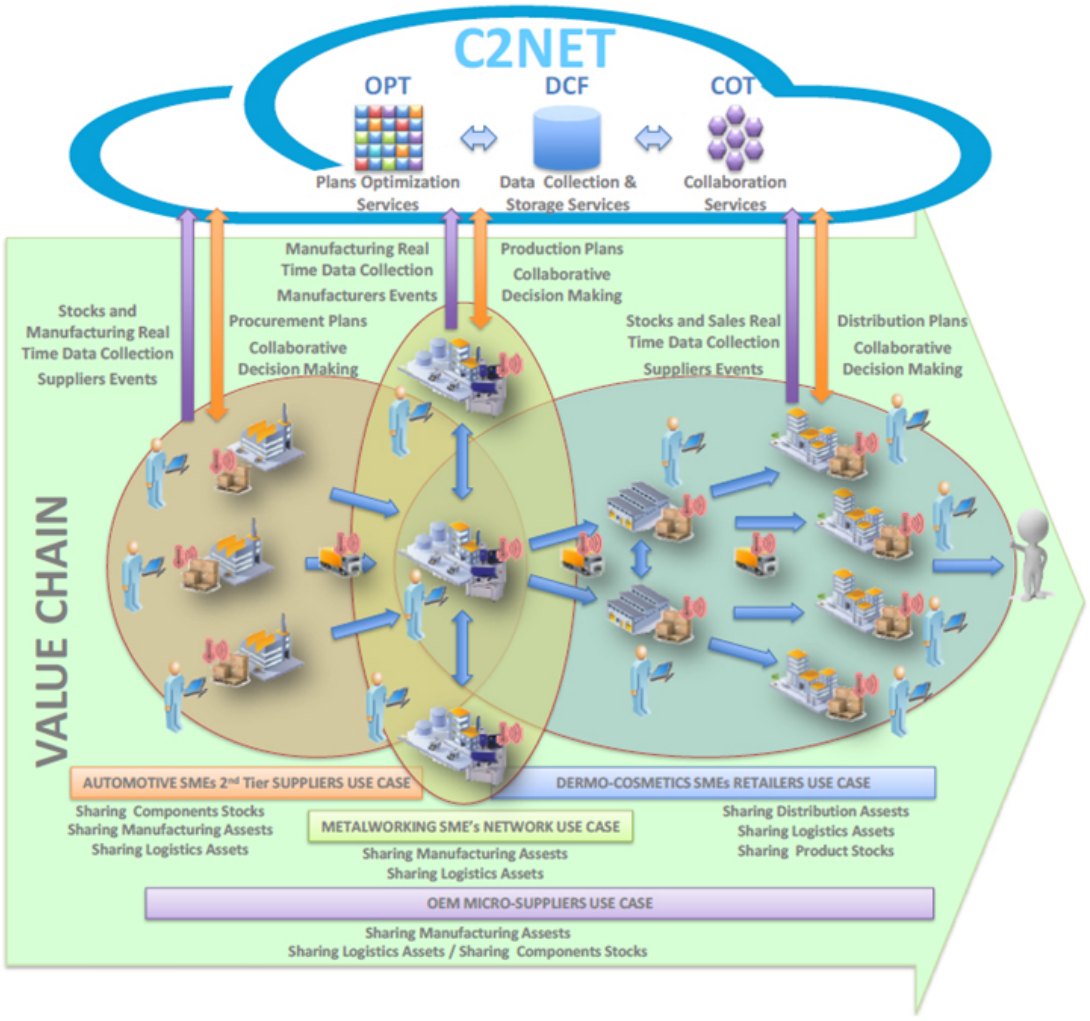


Figure 7.1: Objectives of the C2NET project [178].

The main objective of the project is to provide a new scalable real-time architecture to offer tools supporting the optimization of manufacturing networks composed mainly of SMEs and their logistic assets through demand management, production, and supply plans, considering the Collaborative Network perspective. The project is expected to generate a Cloud Architecture composed by [179]:

[1] http://c2net-project.eu/

- The Data Collection Framework (C2NET DCF) provides software components and hardware devices for IoT-based continuous data collection from supply network resources. This supports collaborative manufacturing functionality based on Cloud capabilities, which can enable solutions that are highly scalable, available, and fault-tolerant.

- The Optimizer (C2NET OPT) supports manufacturing networks in the optimization of manufacturing and logistics assets by the collaborative computation of production, replenishment, and delivery plans. This capability establishes shorter delivery times, better speed and consistency of schedules, higher use of productive resources, and energy savings.

- The Collaboration Tools (C2NET COT) provide a collection of tools to improve the collaborative processes through the Collaborative Manufacturing Network Platform.

- The Cloud Platform (C2NET CPL) integrates the data module, the optimizers, and the collaborative tools in the cloud. This platform uses the cloud facilities to allow access to process optimization resources to all the participants in the value chain to support their decisions and process enhancement.

7.3 Hypothesis Validation

The hypothesis validation addresses the research method application to attend the objectives of this dissertation, namely the research question and hypothesis. Thus, the research question is:

1. Can ontologies enrich the capabilities of security management in IoT-enabled industrial environments, to enhance confidence in Industry 4.0?

The validation of the hypothesis is "*If knowledge about known cybersecurity issues (e.g., vulnerabilities, known threats), and the corresponding prevention measures could be integrated in a comprehensive ontology that is accessible to runtime monitoring and actuation tools, then security systems could be improved to automatically detect threats to the IoT network and dynamically propose suitable protection services*". To verify this hypothesis and improve IoT cybersecurity based on the aforementioned problems, this dissertation presented an ontology-based cybersecurity framework, and the main contributions of this dissertation are listed below:

- The cybersecurity framework itself: an integrated technical framework using ontology and knowledge reasoning to address the security aspect of the Internet of Things within industrial environments (see [176]). The framework focuses on the enterprise (company-side) monitoring, security analysis, and the subsequent

security service design and provisioning, to improve business processes and technology assets;

- The IoTSec ontology, which is a core component of the framework and a continued work of the authors (see [25]), gathering cybersecurity knowledge about alerts and possible threats and providing reasoning capabilities to discover implicit data from the contextual information of security issues;
- Design and orchestration method to implement and provide suitable security services in the IoT environments through the application of the Model-Driven Service Engineering Architecture (MDSEA) methodology (see [26]);
- Runtime security monitoring and actuation services integrated with the IDMEF standard (see [27]).

From the external point of view to the validation, several scientific experts recognized and accepted the several publications that were submitted during this work.

CHAPTER 8

Final Considerations and Future Works

This dissertation introduced cybersecurity as the central challenge and one of the most crucial aspects of the complete adoption of the Internet of Things. The threat of exposing sensitive information from systems to the World Wide Web increased the complexity of the IoT cybersecurity. This is further aggravated by the risk of new threats and vulnerabilities regarding the heterogeneous connectivity of the high number of distinct IoT devices and systems. This dissertation proposed an ontology-based cybersecurity framework that addressed security concerns and increased protection of IoT devices and business processes of the Internet of Things. This involved the instantiation of an integrated technical framework using ontology and knowledge reasoning focused on enterprise monitoring, security analysis, security service design, and provisioning. The proposed cybersecurity framework was implemented, and the IoTSec ontology instantiated and assessed with knowledge about known cybersecurity issues and the corresponding prevention measures, hence validating the hypothesis. The OQuaRE methodology achieved the global average score of 4.11 in a maximum of 5.

As for the strengths of the IoTSec ontology, the assessment identified excellent consistency in its structure with an effective arrangement of the classes. In terms of functional adequacy, OQuaRE identified good individual average scores for knowledge acquisition and reuse. According to IoT cybersecurity, these characteristics associated with operability are essential to increase the amount and quality of the information in the knowledge base. Nevertheless, the assessment also identified some weaknesses in the adaptability and reliability characteristics. Even though these characteristics were significant in terms of usability, they are not relevant for the proposed framework because the proposed utilization of the IoTSec ontology does not consider changing of domains. One way to improve it would be to increase the number of data type and object properties and an average number of the direct parents as a form to create relations between different

classes with more details for different domains.

The implemented industrial scenario addressed the inherent challenges of IoT cybersecurity. The proposed framework established a knowledge base from distinct data sources using the IoTSec ontology to offer suitable security mechanisms according to known threats and vulnerabilities detected from the industrial environment. With the instantiated framework, the monitoring of assets was performed at different types of tools, collecting alerts from threats and vulnerabilities. These alerts were analyzed to discover proper security mechanisms. This implementation conducted an accurate security analysis based on the reasoning capabilities of the IoTSec ontology that determined some discoveries of unknown facts on the knowledge base. It also allowed finding implicit solutions between related classes of the ontology. Moreover, the results of the security analysis required adaptation of services, and the methodology adopted by the proposed framework provided support to accelerate the service design.

During the development of the proposed cybersecurity framework, some challenges were discovered concerning the devices' features found in the industrial environment. The utilization of IoT devices with power and performance constraints created challenges in the adoption of traditional security services, and it required the adaptation of mechanisms with specific hardware characteristics. Furthermore, the actuation requires careful specification to avoid resource consumption, such as lightweight security mechanisms.

Usually, zero-days threats or recent vulnerabilities identified from anomaly-based intrusion detection systems and vulnerability scan systems do not have security mechanisms published yet. In such cases, offering suitable solutions for "uncharted" behavior is a hard task. A particular way to manage such situations is to establish standard actions according to a particular behavior. However, the establishment of actions for each behavior category makes it challenging to select effective mechanisms of different reasoning strategies under different categories of behaviors, which further stresses the challenge of uncertainty reasoning. In this context, Bayesian networks have been used to deal with ontology uncertainty, which requires probability determination in a structured form where each state is justified mathematically and takes into consideration specific real inputs.

8.1 Main Scientific and Technical Contributions

The proposal addressed in this dissertation provided a unified technical framework to monitor business processes and technology assets using an ontology and knowledge reasoning for the IoT cybersecurity domain. In this point, some contributions can be highlighted to improve security monitoring, analysis as well as service design and provisioning to highlight particular asset constraints within an industrial environment. One of these contributions is the IoTSec ontology, which gathered cybersecurity knowledge about alerts and possible threats from the contextual information of security

issues to correlate it to vulnerabilities and security properties. The correlation among classes provided links between basic elements of cybersecurity: Assets, Threats, Security Mechanisms, Vulnerabilities, and Security Properties. These links were fundamental in the finding of implicit data from the environment, mainly through reasoning capabilities from ontology engineering.

Concerning runtime security monitoring, generated cybersecurity alerts from different probes categories could be integrated using the IDMEF standard in the same knowledge base, providing integrated actuation services and checking mechanisms. The main contribution of this aspect was the minimization of problems with heterogeneous data from distinct security mechanisms used for intrusion detection and vulnerability scanners. The design and orchestration method provided capabilities to adapt security services already existing or to generate a new one using MDSEA methodology, creating a pool of security services according to the particular needs of the application environment. Another contribution of this work was the aggregation of the runtime monitoring and the design and orchestration method that offers suitable services based on security requirements of the IoT cybersecurity.

8.2 Future Research Directions

Following the central challenge, i.e., IoT cybersecurity, there are still open research issues without ideal solutions. The IoT cybersecurity is a severe problem for society as a whole, and several insights into future trends have the potential to deal with it. Digital platforms consist of a composition of technologies formed by components directly related to IoT cybersecurity such as 5G, Edge Computing, and Low Cost Communication. These technologies will create the most unpredictable and disruptive breakthroughs for humans. They fill the gap between the device sensors and data networks to provide awareness using back-end applications to the generated data from sensors. This interoperability is a requirement that prevents the emergence of broadly accepted IoT ecosystems [180].

Artificial Intelligence (AI) is a relevant field that has received much attention with the progress of IoT because it allows developing systems that learn, adapt, and act autonomously to improve decision-making and business models within the digital market. This growth consists of several technologies such as decision trees, linear regression, and neural networks, resulting in effective implementations of physical devices and services to deliver a new class of smart applications for business scenarios. This orchestration provides intelligent devices through the use of IoT platform services or models as-a-service from an adaptive perspective [181]. AI will offer better solutions for IoT cybersecurity that aims to identify threats, even if it requires a short learning phase to establish which events are potential attacks. In our context, AI should provide alternatives to understand the system's behavior (e.g., to detect zero-day threats) and make sure that the situation of interest is happening [182, 183]. For instance, machine learning systems could be used in the future to analyze logs with statistical features to extract behavior

snapshots of the IoT network detecting threats and vulnerabilities from compromised IoT devices [184].

Ontologies provide a semantically rich knowledge base for information management in several contexts, such as business intelligence [185]. Ontology engineering is a key enabling technology to build a model of a specific domain, which has capabilities to share a common understanding and to improve the communication between people and application systems [186]. In the cybersecurity domain, as explored in this article, ontologies support the automatic establishment of security metrics based on explicit and reasoning information about situations of interest and combine knowledge from multiple security experts. Also, ontologies have improved the efficiency and effectiveness in security operations [187] and the natural language processing to help analysts to extract relevant pieces of information to characterize vulnerabilities and threats [188]. However, there are open issues that must be addressed to achieve a sufficient level of a multi-layered cybersecurity intelligence ontology, to explore smart capabilities and understand potential threats against the ever-changing cybersecurity landscape.

Finally, the IoT is a fast-growing, increasingly complex network of connected sensors and devices. One important future approach to deal with real challenges is the adoption of a continuous adaptive risk and trust assessment, which allows real-time decision-making with adaptive responses [189]. Also, the adoption of Software Defined System (SDSys) is an approach to reduce the overhead in the control and management operations of complex computing systems such as the Software Defined Networking (SDN), proposed to eliminate the rigidity present in traditional networks [190, 191]. It allows the softwarization of IoT infrastructure to improve the sensor networks' agility and flexibility. This softwarization is provided with Software Defined Security (SDS). Also, SDS can provide a flexible and centralized security solution by abstracting the security mechanisms from the hardware layer to a software layer [192]. The aggregation of SDSys such as SDN and SDS will become one of the key transformations in 5G networks, which creates new opportunities to achieve SDN-based 5G network monitoring as an alternative to traditional network-wide monitoring initiatives [175, 193]. The transition between traditional network architectures to SDN-based architectures is also an open issue. Most of these cybersecurity challenges of IoT applications are directly related to a centralized approach, such as addressed in the SDN.

Bibliography

[1] M. Conti, S. K. Das, C. Bisdikian, M. Kumar, L. M. Ni, A. Passarella, G. Roussos, G. Tröster, G. Tsudik, and F. Zambonelli. "Looking ahead in pervasive computing: Challenges and opportunities in the era of cyber–physical convergence". In: *Pervasive and Mobile Computing* 8.1 (2012), pp. 2 –21. ISSN: 1574-1192. DOI: 10.1016/j.pmcj.2011.10.001.

[2] J. H. Nord, A. Koohang, and J. Paliszkiewicz. "The Internet of Things: Review and theoretical framework". In: *Expert Systems with Applications* 133 (2019), pp. 97 –108. ISSN: 0957-4174. DOI: https://doi.org/10.1016/j.eswa.2019.05.014.

[3] J. E. Ibarra-Esquer, F. F. González-Navarro, B. L. Flores-Rios, L. Burtseva, and M. A. Astorga-Vargas. "Tracking the Evolution of the Internet of Things Concept Across Different Application Domains". In: *Sensors* 17.6 (2017). ISSN: 1424-8220. DOI: 10.3390/s17061379.

[4] E. Borgia. "The Internet of Things vision: Key features, applications and open issues". In: *Computer Communications* 54 (2014), pp. 1 –31. ISSN: 0140-3664. DOI: 10.1016/j.comcom.2014.09.008.

[5] H. Cai, L. D. Xu, B. Xu, C. Xie, S. Qin, and L. Jiang. "IoT-Based Configurable Information Service Platform for Product Lifecycle Management". In: *IEEE Transactions on Industrial Informatics* 10.2 (2014), pp. 1558–1567. ISSN: 1551-3203. DOI: 10.1109/TII.2014.2306391.

[6] E. Polycarpou, L. Lambrinos, and E. Protopapadakis. "Smart parking solutions for urban areas". In: *2013 IEEE 14th International Symposium on "A World of Wireless, Mobile and Multimedia Networks" (WoWMoM)*. 2013, pp. 1–6. DOI: 10.1109/WoWMoM.2013.6583499.

[7] E. Ancillotti, R. Bruno, and M. Conti. "The role of communication systems in smart grids: Architectures, technical solutions and research challenges". In: *Computer Communications* 36.17 (2013), pp. 1665–1697. ISSN: 0140-3664. DOI: 10.1016/j.comcom.2013.09.004.

[8] F. Delmastro. "Pervasive communications in healthcare". In: *Computer Communications* 35.11 (2012), pp. 1284 –1295. ISSN: 0140-3664. DOI: 10.1016/j.comcom.2012.04.018.

[9] L. D. Xu, W. He, and S. Li. "Internet of Things in Industries: A Survey". In: *IEEE Transactions on Industrial Informatics* 10.4 (2014), pp. 2233–2243. ISSN: 1551-3203. DOI: 10.1109/TII.2014.2300753.

[10] D. Evans. *The internet of things: How the next evolution of the internet is changing everything*. Available online. 2011. URL: https://www.cisco.com/c/dam/en_us/about/ac79/docs/innov/IoT_IBSG_0411FINAL.pdf. (Accessed on 16 January 2020).

[11] D. Miessler. *HP Study Reveals 70 Percent of Internet of Things Devices Vulnerable to Attack*. Available online. URL: http://www8.hp.com/us/en/hp-news/press-release.html?id=1744676. (Accessed on 16 January 2020).

[12] M. Abomhara and G. M. Køien. "Cyber Security and the Internet of Things: Vulnerabilities, Threats, Intruders and Attacks". In: *Journal of Cyber Security* 4 (2015), pp. 65–88. DOI: 10.13052/jcsm2245-1439.414.

[13] R. Roman, J. Zhou, and J. Lopez. "On the features and challenges of security and privacy in distributed internet of things". In: *Computer Networks* 57.10 (2013), pp. 2266–2279. ISSN: 13891286. DOI: 10.1016/j.comnet.2012.12.018.

[14] M. Wolf and D. Serpanos. "Safety and Security in Cyber-Physical Systems and Internet-of-Things Systems". In: *Proceedings of the IEEE* 106.1 (2018), pp. 9–20. ISSN: 0018-9219. DOI: 10.1109/JPROC.2017.2781198.

[15] J. Lin, W. Yu, N. Zhang, X. Yang, H. Zhang, and W. Zhao. "A Survey on Internet of Things: Architecture, Enabling Technologies, Security and Privacy, and Applications". In: *IEEE Internet of Things Journal* 4.5 (2017), pp. 1125–1142. DOI: 10.1109/JIOT.2017.2683200.

[16] R. Ande, B. Adebisi, M. Hammoudeh, and J. Saleem. "Internet of Things: Evolution and technologies from a security perspective". In: *Sustainable Cities and Society* 54 (2020), p. 101728. ISSN: 2210-6707. DOI: https://doi.org/10.1016/j.scs.2019.101728.

[17] D. Inc. *Dell Data Security Survey Finds that a Lack of Security Knowledge Limits Business Initiatives*. Available online. URL: http://www.dell.com/learn/us/en/uscorp1/press-releases/dell-data-security-survey/. (Accessed on 16 January 2020).

[18] L. Camarinha-Matos. *Scientific Research Methodologies and Techniques - Unit 2: Scientific Method*. Phd Program in Electrical and Computer Engineering. Available online. URL: www.uninova.pt/~cam/teaching/srmt.htm. (Accessed on 16 January 2020).

[19] J. Manyika. *The Internet of Things: Mapping the value beyond the hype*. McKinsey Global Institute, 2015.

[20] M. Rüßmann, M. Lorenz, P. Gerbert, M. Waldner, J. Justus, P. Engel, and M. Harnisch. "Industry 4.0: The future of productivity and growth in manufacturing industries". In: *Boston Consulting Group* 9 (2015).

[21] A.-R. Sadeghi, C. Wachsmann, and M. Waidner. "Security and Privacy Challenges in Industrial Internet of Things". In: *Proceedings of the 52Nd Annual Design Automation Conference*. DAC '15. ACM, 2015, 54:1–54:6. ISBN: 978-1-4503-3520-1. DOI: 10.1145/2744769.2747942.

[22] G. Eastwood. *4 critical security challenges facing IoT*. Available online: https://www. networkworld.com/article/3166106/internet-of-things/4-critical-security-challen ges-facing-iot.html. (accessed on 16 January 2020).

[23] R. Toney. *Risk Management in the Internet of Things*. Available online: https://erm. ncsu.edu/library/article/risk-management-in-the-internet-of-things. (accessed on 16 January 2020).

[24] H. Kagermann, W. Wahlster, and J. Helbig. "Securing the future of German manufacturing industry". In: *Recommendations for implementing the strategic initiative INDUSTRIE* 4 (2013).

[25] B. A. Mozzaquatro, R. Jardim-Goncalves, and C. Agostinho. "Towards a reference ontology for security in the Internet of Things". In: *2015 IEEE International Workshop on Measurements and Networking, M and N 2015 - Proceedings*. 2015, pp. 117–122. ISBN: 9781479918607. DOI: 10.1109/IWMN.2015.7322984.

[26] Y Ducq, C Agostinho, D Chen, G Zacharewicz, and R. Goncalves. "Generic methodology for service engineering based on service modelling and model transformation". In: *Manufacturing Service Ecosystem. Achievements of the European 7th FP FoF-ICT Project MSEE: Manufacturing SErvice Ecosystem (Grant No. 284860). Eds. Weisner S, Guglielmina C, Gusmeroli S, Doumeingts G* (2014), pp. 41–49.

[27] H. Debar, D. A. Curry, and B. S. Feinstein. *The intrusion detection message exchange format (IDMEF)*. IETF Intrusion Detection Exchange Format Working Group. Available online. URL: https://tools.ietf.org/html/rfc4765. (Accessed on 16 January 2020).

[28] L. Atzori, A. Iera, and G. Morabito. "The Internet of Things: A survey". In: *Computer Networks* 54.15 (2010), pp. 2787 –2805. ISSN: 1389-1286. DOI: 10.1016/j.comnet.2010.05.010.

[29] J. Chase. *The Evolution of the Internet of Things*. White Paper Texas Instrument. Available online. 2013. URL: http://www.ti.com/lit/ml/swrb028/swrb028.pdf. (Accessed on 16 January 2020).

[30] O Garcia-Morchon, S Kumar, and M Sethi. "State-of-the-Art and Challenges for the Internet of Things Security". In: *IRTF draft, draft-irtft2trg-iot-seccons-15* (2018).

[31] K. Rose, S. Eldridge, and L. Chapin. "The internet of things: An overview". In: *The Internet Society (ISOC)* (2015), pp. 1–50.

[32] S. Alam, M. M. R. Chowdhury, and J. Noll. "Interoperability of Security-Enabled Internet of Things". In: *Wireless Personal Communications* 61.3 (2011), pp. 567–586. ISSN: 1572-834X. DOI: 10.1007/s11277-011-0384-6.

[33] Z. Yang, Y. Yue, Y. Yang, Y. Peng, X. Wang, and W. Liu. "Study and application on the architecture and key technologies for IOT". In: *2011 International Conference on Multimedia Technology*. 2011, pp. 747–751. DOI: 10.1109/ICMT.2011.6002149.

[34] P. Spiess, S. Karnouskos, D. Guinard, D. Savio, O. Baecker, L. M. S. d. Souza, and V. Trifa. "SOA-Based Integration of the Internet of Things in Enterprise Services". In: *2009 IEEE International Conference on Web Services*. 2009, pp. 968–975. DOI: 10.1109/ICWS.2009.98.

[35] R. Khan, S. U. Khan, R. Zaheer, and S. Khan. "Future Internet: The Internet of Things Architecture, Possible Applications and Key Challenges". In: *2012 10th International Conference on Frontiers of Information Technology*. 2012, pp. 257–260. DOI: 10.1109/FIT.2012.53.

[36] M. Wu, T.-J. Lu, F.-Y. Ling, J. Sun, and H.-Y. Du. "Research on the architecture of Internet of Things". In: *2010 3rd International Conference on Advanced Computer Theory and Engineering(ICACTE)*. Vol. 5. 2010, pp. V5–484–V5–487. DOI: 10.1109/ICACTE.2010.5579493.

[37] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash. "Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications". In: *IEEE Communications Surveys Tutorials* 17.4 (2015), pp. 2347–2376. ISSN: 1553-877X. DOI: 10.1109/COMST.2015.2444095.

[38] O. Hersent, D. Boswarthick, and O. Elloumi. *The internet of things: Key applications and protocols*. John Wiley & Sons, 2011.

[39] Ángel Asensio, Álvaro Marco, R. Blasco, and R. Casas. "Protocol and Architecture to Bring Things into Internet of Things". In: *International Journal of Distributed Sensor Networks* 10.4 (2014), p. 158252. DOI: 10.1155/2014/158252.

[40] U. Hunkeler, H. L. Truong, and A. Stanford-Clark. "MQTT-S A publish/subscribe protocol for Wireless Sensor Networks". In: *Communication Systems Software and Middleware and Workshops, 2008. COMSWARE 2008. 3rd International Conference on*. 2008, pp. 791–798. DOI: 10.1109/COMSWA.2008.4554519.

[41] C. Gomez and J. Paradells. "Wireless home automation networks: A survey of architectures and technologies". In: *IEEE Communications Magazine* 48.6 (2010), pp. 92–101. ISSN: 0163-6804. DOI: 10.1109/MCOM.2010.5473869.

[42] R. Want. "Near Field Communication". In: *IEEE Pervasive Computing* 10 (July 2011), pp. 4–7. ISSN: 1536-1268. DOI: 10.1109/MPRV.2011.55.

[43] R. Want. "An Introduction to RFID Technology". In: *IEEE Pervasive Computing* 5 (Jan. 2006), pp. 25–33. ISSN: 1536-1268. DOI: 10.1109/MPRV.2006.2.

[44] P. Kinney et al. "Zigbee technology: Wireless control that simply works". In: *Communications design conference*. Vol. 2. 2003, pp. 1–7.

[45] J. S. Lee, Y. W. Su, and C. C. Shen. "A Comparative Study of Wireless Protocols: Bluetooth, UWB, ZigBee, and Wi-Fi". In: *IECON 2007 - 33rd Annual Conference of the IEEE Industrial Electronics Society*. 2007, pp. 46–51. DOI: 10.1109/IECON.2007.4460126.

[46] A. Salam and S. Shah. "Internet of Things in Smart Agriculture: Enabling Technologies". In: *2019 IEEE 5th World Forum on Internet of Things (WF-IoT)*. 2019, pp. 692–695. DOI: 10.1109/WF-IoT.2019.8767306.

[47] K. Zhang, Z. Ao, C. Tang, Y. Wang, W. Zhu, and B. Feng. "Application of Internet of Things in Combined Operation Logistics Support". In: *2012 Fourth International Conference on Computational and Information Sciences*. 2012, pp. 388–391. DOI: 10.1109/ICCIS.2012.82.

[48] A. S. Voulodimos, C. Z. Patrikakis, A. B. Sideridis, V. A. Ntafis, and E. M. Xylouri. "A complete farm management system based on animal identification using RFID technology". In: *Computers and Electronics in Agriculture* 70.2 (2010). Special issue on Information and Communication Technologies in Bio and Earth Sciences, pp. 380 –388. ISSN: 0168-1699. DOI: 10.1016/j.compag.2009.07.009.

[49] Amazon. *AWS IoT Framework*. Available online. URL: https://aws.amazon.com/iot. (Accessed on 16 January 2020).

[50] D. Locke. *MQ Telemetry Transport (MQTT) V3.1 Protocol Specification*. Available online. URL: http://www.ibm.com/developerworks/webservices/library/ws-mqtt/index.html. (Accessed on 16 January 2020).

[51] Pelion. *The Full IoT Stack - Arm Pelion*. Available online. URL: https://www.pelion.com/. (Accessed on 10 March 2020).

[52] A. Microsoft. *Azure IoT Suite*. Available online. URL: https://azure.microsoft.com/en-us/suites/iot-suite/. (Accessed on 16 January 2020).

[53] Ericsson. *Calvin IoT platform*. Available online. URL: https://www.ericsson.com/research-blog/open-source-calvin/. (Accessed on 16 January 2020).

[54] P. Persson and O. Angelsmark. "Calvin – Merging Cloud and IoT". In: *Procedia Computer Science* 52 (2015). The 6th International Conference on Ambient Systems, Networks and Technologies (ANT-2015), the 5th International Conference on Sustainable Energy Information Technology (SEIT-2015), pp. 210 –217. ISSN: 1877-0509. DOI: 10.1016/j.procs.2015.05.059.

[55] Kura. *The extensible open source Java/OSGi IoT Edge Framework*. Available online. URL: https://www.eclipse.org/kura/. (Accessed on 12 January 2019).

[56] SmartThings. *Build connected experiences for millions of SmartThings users*. Available online. URL: https://smartthings.developer.samsung.com/. (Accessed on 16 January 2019).

[57] M. Ammar, G. Russello, and B. Crispo. "Internet of Things: A survey on the security of IoT frameworks". In: *Journal of Information Security and Applications* 38 (2018), pp. 8 –27. ISSN: 2214-2126. DOI: 10.1016/j.jisa.2017.11.002.

[58] K. D. Mitnick and W. L. Simon. *The art of deception: Controlling the human element of security*. John Wiley & Sons, 2011.

[59] R. von Solms and J. van Niekerk. "From information security to cyber security". In: *Computers & Security* 38 (2013). Cybercrime in the Digital Economy, pp. 97 –102. ISSN: 0167-4048. DOI: 10.1016/j.cose.2013.04.004.

[60] M. E. Whitman and H. J. Mattord. *Principles of information security*. Cengage Learning, 2011.

[61] Z. Xiao and Y. Xiao. "Security and Privacy in Cloud Computing". In: *IEEE Communications Surveys Tutorials* 15.2 (2013), pp. 843–859. ISSN: 1553-877X. DOI: 10.1109/SURV.2012.060912.00182.

[62] J. Jang-Jaccard and S. Nepal. "A survey of emerging threats in cybersecurity". In: *Journal of Computer and System Sciences* 80.5 (2014). Special Issue on Dependable and Secure Computing, pp. 973 –993. ISSN: 0022-0000. DOI: 10.1016/j.jcss.2014.02.005.

[63] D. Gollmann. "Computer security". In: *Wiley Interdisciplinary Reviews: Computational Statistics* 2.5 (2010), pp. 544–554. ISSN: 1939-0068. DOI: 10.1002/wics.106.

[64] J. M. Kizza. *Guide to computer network security*. Springer, 2009.

[65] T. Heer, O. Garcia-Morchon, R. Hummen, S. L. Keoh, S. S. Kumar, and K. Wehrle. "Security Challenges in the IP-based Internet of Things". In: *Wireless Personal Communications* 61.3 (2011), pp. 527–542. ISSN: 1572-834X. DOI: 10.1007/s11277-011-0385-5.

[66] R. Roman, P. Najera, and J. Lopez. "Securing the Internet of Things". In: *Computer* 44.9 (2011), pp. 51–58. ISSN: 0018-9162. DOI: 10.1109/MC.2011.291.

[67] C. Lee, L. Zappaterra, K. Choi, and H.-A. Choi. "Securing smart home: Technologies, security challenges, and security requirements". In: *2014 IEEE Conference on Communications and Network Security*. 2014, pp. 67–72. DOI: 10.1109/CNS.2014.6997467.

[68] S. Blake-Wilson, N. Bolyard, V. Gupta, C. Hawk, and B. Möller. *Elliptic curve cryptography (ECC) cipher suites for transport layer security (TLS)*. Tech. rep. RFC 4492. Updated by RFC 5246. 2006.

[69] A. Bassi, M. Bauer, M. Fiedler, T. Kramp, R. Van Kranenburg, S. Lange, and S. Meissner. *Enabling things to talk*. Springer, 2013.

[70] J. Arkko, E. Carrara, F. Lindholm, K. Norrman, and M. Naslund. "Mikey: Multimedia internet keying". In: (2004).

[71] J. Ortiz, C. Crawford, and F. Le. "DeviceMien: Network Device Behavior Modeling for Identifying Unknown IoT Devices". In: *Proceedings of the International Conference on Internet of Things Design and Implementation*. IoTDI '19. Montreal, Quebec, Canada: ACM, 2019, pp. 106–117. ISBN: 978-1-4503-6283-2. DOI: 10.1145/3302505.3310073.

[72] I. Alqassem and D. Svetinovic. "A taxonomy of security and privacy requirements for the Internet of Things (IoT)". In: *2014 IEEE International Conference on Industrial Engineering and Engineering Management*. 2014, pp. 1244–1248. DOI: 10.1109/IEEM.2014.7058837.

[73] P. Karhula. "What is the effect of WikiLeaks for Freedom of Information?" In: *FAIFE Spotlight [online]* 19 (2011).

[74] Z. Yan, P. Zhang, and A. V. Vasilakos. "A survey on trust management for Internet of Things". In: *Journal of Network and Computer Applications* 42.2 (2014), pp. 120–134. ISSN: 10848045. DOI: 10.1016/j.jnca.2014.01.014.

[75] W. Stallings, L. Brown, M. D. Bauer, and A. K. Bhattacharjee. *Computer security: principles and practice*. Pearson Education, 2012.

[76] T. Roosta, S. Shieh, and S. Sastry. "Taxonomy of security attacks in sensor networks and countermeasures". In: *The first IEEE international conference on system integration and reliability improvements*. Vol. 25. 2006, p. 94.

[77] A. Hassanzadeh, S. Modi, and S. Mulchandani. "Towards effective security control assignment in the Industrial Internet of Things". In: *2015 IEEE 2nd World Forum on Internet of Things (WF-IoT)*. 2015, pp. 795–800. DOI: 10.1109/WF-IoT.2015.7389155.

[78] K. Renaud and D. Gálvez-Cruz. "Privacy: Aspects, definitions and a multi-faceted privacy preservation approach". In: *2010 Information Security for South Africa*. 2010, pp. 1–8. DOI: 10.1109/ISSA.2010.5588297.

[79] D. C. Galvez-Cruz. "An environment for protecting the privacy of e-shoppers". PhD thesis. University of Glasgow, 2009.

[80] M. Ghasemi, M. Saadaat, and O. Ghollasi. "Threats of Social Engineering Attacks Against Security of Internet of Things (IoT)". In: *Fundamental Research in Electrical Engineering*. Ed. by S. Montaser Kouhsari. Singapore: Springer Singapore, 2019, pp. 957–968. ISBN: 978-981-10-8672-4.

[81] M. Junger, L. Montoya, and F.-J. Overink. "Priming and warnings are not effective to prevent social engineering attacks". In: *Computers in Human Behavior* 66 (2017), pp. 75 –87. ISSN: 0747-5632. DOI: 10.1016/j.chb.2016.09.012.

[82] R. Sulatycki and E. B. Fernandez. "Two Threat Patterns That Exploit "Security Misconfiguration" and "Sensitive Data Exposure" Vulnerabilities". In: *Proceedings of the 20th European Conference on Pattern Languages of Programs*. EuroPLoP '15. ACM, 2015, 46:1–46:11. ISBN: 978-1-4503-3847-9. DOI: 10.1145/2855321.2855368.

[83] K. Aarika, M. Bouhlal, R. A. Abdelouahid, S. Elfilali, and E. Benlahmar. "Perception layer security in the internet of things". In: *Procedia Computer Science* 175 (2020). The 17th International Conference on Mobile Systems and Pervasive Computing (MobiSPC),The 15th International Conference on Future Networks and Communications (FNC),The 10th International Conference on Sustainable Energy Information Technology, pp. 591 –596. ISSN: 1877-0509. DOI: https://doi.org/10.1016/j.procs.2020.07.085.

[84] R. Spreitzer, V. Moonsamy, T. Korak, and S. Mangard. "Systematic Classification of Side-Channel Attacks: A Case Study for Mobile Devices". In: *IEEE Communications Surveys Tutorials* 20.1 (2018), pp. 465–488. ISSN: 1553-877X. DOI: 10.1109/COMST.2017.2779824.

[85] Q. Chi, H. Yan, C. Zhang, Z. Pang, and L. D. Xu. "A Reconfigurable Smart Sensor Interface for Industrial WSN in IoT Environment". In: *IEEE Transactions on Industrial Informatics* 10.2 (2014), pp. 1417–1425. ISSN: 1551-3203. DOI: 10.1109/TII.2014.2306798.

[86] T. Beardsley. *Multiple Vulnerabilities in Animas OneTouch Ping Insulin Pump*. Available online. URL: https://blog.rapid7.com/2016/10/04/r7-2016-07-multiple-vulnerabilities-in-animas-onetouch-ping-insulin-pump/. (Accessed on 16 January 2019).

[87] R. Beyah, S. Kangude, G. Yu, B. Strickland, and J. Copeland. "Rogue access point detection using temporal traffic characteristics". In: *IEEE Global Telecommunications Conference, 2004. GLOBECOM '04.* Vol. 4. 2004, 2271–2275 Vol.4. DOI: 10.1109/GLOCOM.2004.1378413.

[88] C. Karlof and D. Wagner. "Secure routing in wireless sensor networks: attacks and countermeasures". In: *Proceedings of the First IEEE International Workshop on Sensor Network Protocols and Applications, 2003.* 2003, pp. 113–127. DOI: 10.1109/SNPA.2003.1203362.

[89] F. Alsubaei, A. Abuhussein, and S. Shiva. "Security and Privacy in the Internet of Medical Things: Taxonomy and Risk Assessment". In: *2017 IEEE 42nd Conference on Local Computer Networks Workshops (LCN Workshops)*. 2017, pp. 112–120. DOI: 10.1109/LCN.Workshops.2017.72.

[90] W. G. Halfond, J. Viegas, A. Orso, et al. "A classification of SQL-injection attacks and countermeasures". In: *Proceedings of the IEEE International Symposium on Secure Software Engineering*. Vol. 1. IEEE. 2006, pp. 13–15.

[91] C. Jun and C. Chi. "Design of Complex Event-Processing IDS in Internet of Things". In: *2014 Sixth International Conference on Measuring Technology and Mechatronics Automation*. 2014, pp. 226–229. DOI: 10.1109/ICMTMA.2014.57.

[92] H. E. Poston. "A brief taxonomy of intrusion detection strategies". In: *2012 IEEE National Aerospace and Electronics Conference (NAECON)*. 2012, pp. 255–263. DOI: 10.1109/NAECON.2012.6531064.

[93] S. O. Amin, M. S. Siddiqui, C. S. Hong, and J. Choe. "A novel coding scheme to implement signature based IDS in IP based Sensor Networks". In: *2009 IFIP/IEEE International Symposium on Integrated Network Management-Workshops*. 2009, pp. 269–274. DOI: 10.1109/INMW.2009.5195973.

[94] M. Ficco. "Security event correlation approach for cloud computing". In: *International Journal of High Performance Computing and Networking* 7.3 (2013), pp. 173–185. DOI: 10.1504/IJHPCN.2013.056525.

[95] M. Ficco, L. Tasquier, and R. Aversa. "Intrusion detection in federated clouds". In: *International Journal of Computational Science and Engineering* 13.3 (2016), pp. 219–232. DOI: 10.1504/IJCSE.2016.078929.

[96] M. Tao, J. Zuo, Z. Liu, A. Castiglione, and F. Palmieri. "Multi-layer cloud architectural model and ontology-based security service framework for IoT-based smart homes". In: *Future Generation Computer Systems* 78 (2018), pp. 1040 –1051. ISSN: 0167-739X. DOI: 10.1016/j.future.2016.11.011.

[97] A. Ekelhart, S. Fenz, and T. Neubauer. "AURUM: A Framework for Information Security Risk Management". In: *2009 42nd Hawaii International Conference on System Sciences*. 2009, pp. 1–10. DOI: 10.1109/HICSS.2009.82.

[98] N. Shadbolt, T. Berners-Lee, and W. Hall. "The Semantic Web Revisited". In: *IEEE Intelligent Systems* 21.3 (2006), pp. 96–101. ISSN: 1541-1672. DOI: 10.1109/MIS.2006.62.

[99] N. Bikakis, C. Tsinaraki, N. Gioldasis, I. Stavrakantonakis, and S. Christodoulakis. "The XML and Semantic Web Worlds: Technologies, Interoperability and Integration: A Survey of the State of the Art". In: *Semantic Hyper/Multimedia Adaptation: Schemes and Applications*. Ed. by I. E. Anagnostopoulos, M. Bieliková, P. Mylonas, and N. Tsapatsoulis. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 319–360. ISBN: 978-3-642-28977-4. DOI: 10.1007/978-3-642-28977-4_12.

[100] G. Antoniou and F. Van Harmelen. *A semantic web primer*. MIT press, 2004.

[101] L. Yu. *A Developer's Guide to the Semantic Web*. 2nd ed. IT Pro. Springer Berlin Heidelberg, 2011, p. 608. ISBN: 978-3-662-43795-7. DOI: 10.1007/978-3-662-43796-4.

[102] D. Brickley, R. V. Guha, and B. McBride. "RDF Schema 1.1". In: *W3C recommendation* 25 (2014), pp. 2004–2014.

[103] T. R. Gruber. "Toward principles for the design of ontologies used for knowledge sharing?" In: *International Journal of Human-Computer Studies* 43.5 (1995), pp. 907 –928. ISSN: 1071-5819. DOI: 10.1006/ijhc.1995.1081.

[104] N. Guarino. "Formal ontology, conceptual analysis and knowledge representation". In: *International Journal of Human-Computer Studies* 43.5 (1995), pp. 625 –640. ISSN: 1071-5819. DOI: 10.1006/ijhc.1995.1066.

[105] T. R. Gruber. "A translation approach to portable ontology specifications". In: *Knowledge Acquisition* 5.2 (1993), pp. 199 –220. ISSN: 1042-8143. DOI: 10.1006/knac.1993.1008.

[106] W3C. *OWL Web Ontology Language Use Cases and Requirements*. Available online. URL: https://www.w3.org/TR/webont-req/. (Accessed on 16 January 2020).

[107] S. Staab and R. Studer. *Handbook on ontologies*. Springer Science & Business Media, 2010.

[108] J. Broekstra, A. Kampman, and F. van Harmelen. "Sesame: A Generic Architecture for Storing and Querying RDF and RDF Schema". In: *The Semantic Web – ISWC 2002*. Ed. by I. Horrocks and J. Hendler. Berlin, Heidelberg: Springer Berlin Heidelberg, 2002, pp. 54–68. ISBN: 978-3-540-48005-1.

[109] L. F. Sikos. "OWL Ontologies in Cybersecurity: Conceptual Modeling of Cyber-Knowledge". In: *AI in Cybersecurity*. Ed. by L. F. Sikos. Cham: Springer International Publishing, 2019, pp. 1–17. DOI: 10.1007/978-3-319-98842-9_1.

[110] G. Denker, L. Kagal, and T. Finin. *Security in the Semantic Web using OWL*. Tech. rep. 1. 2005, pp. 51–58. DOI: 10.1016/j.istr.2004.11.002.

[111] S. Fenz and A. Ekelhart. "Formalizing information security knowledge". In: *Proceedings of the 4th International Symposium on Information, Computer, and Communications Security - ASIACCS '09*. New York, New York, USA: ACM Press, 2009, pp. 183–194. ISBN: 9781605583945. DOI: 10.1145/1533057.1533084.

[112] A Herzog, N Shahmehri, and C Duma. "An ontology of information security". In: *International Journal of Information Security and Privacy (IJISP)* 1.4 (2007), pp. 1–23. ISSN: 19301650. DOI: 10.4018/jisp.2007100101.

[113] A. Kim, J. Luo, and M. Kang. "Security Ontology for Annotating Resources". In: *On the Move to Meaningful Internet Systems 2005: CoopIS, DOA, and ODBASE*. Ed. by R. Meersman and Z. Tari. Springer Berlin Heidelberg, 2005, pp. 1483–1499. DOI: 10.1007/11575801_34.

[114] Ángel García-Crespo, J. M. Gómez-Berbís, R. Colomo-Palacios, and G. Alor-Hernández. "SecurOntology: A semantic web access control framework". In: *Computer Standards & Interfaces* 33.1 (2011). Special Issue: Secure Semantic Web, pp. 42–49. ISSN: 0920-5489. DOI: 10.1016/j.csi.2009.10.003.

[115] A. Gyrard, C. Bonnet, and K. Boudaoud. "An Ontology-Based Approach for Helping to Secure the ETSI Machine-to-Machine Architecture". In: *IEEE International Conference on Internet of Things 2014 (iThings)*. 2014, pp. 109–116. DOI: 10.1109/iThings.2014.25.

[116] J. Undercoffer, A. Joshi, and J. Pinkston. "Modeling Computer Attacks : An Ontology for Intrusion Detection". In: *Recent Advances in Intrusion Detection: 6th International Symposium, RAID 2003, Pittsburgh, PA, USA, September 8-10, 2003. Proceedings*. 2003, pp. 113–135.

[117] A. Avizienis, J. C. Laprie, B. Randell, and C. Landwehr. "Basic concepts and taxonomy of dependable and secure computing". In: *IEEE Transactions on Dependable and Secure Computing* 1.1 (2004), pp. 11–33. ISSN: 1545-5971. DOI: 10.1109/TDSC.2004.2.

[118] A. Ekelhart, S. Fenz, M. D. Klemen, and E. R. Weippl. "Security Ontology: Simulating Threats to Corporate Assets". In: *Information Systems Security*. Ed. by A. Bagchi and V. Atluri. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 249–259. ISBN: 978-3-540-68963-8. DOI: 10.1007/11961635_1.

[119] A. Ekelhart, S. Fenz, M. Klemen, and E. Weippl. "Security Ontologies: Improving Quantitative Risk Analysis". In: *System Sciences, 2007. HICSS 2007. 40th Annual Hawaii International Conference on*. 2007, 156a–156a. DOI: 10.1109/HICSS.2007.478.

[120] G. Goluch, A. Ekelhart, S. Fenz, S. Jakoubi, S. Tjoa, and a. T. Muck. "Integration of an Ontological Information Security Concept in Risk Aware Business Process Management". In: *Proceedings of the 41st Annual Hawaii International Conference on System Sciences (HICSS 2008)*. 2008, pp. 377–377. DOI: 10.1109/HICSS.2008.211.

[121] S. Fenz and E. Weippl. "Ontology based IT-security planning". In: *2006 12th Pacific Rim International Symposium on Dependable Computing (PRDC'06)*. 2006, pp. 389–390. DOI: 10.1109/PRDC.2006.49.

[122] G. Denker, L. Kagal, T. Finin, M. Paolucci, and K. Sycara. "Security for DAML Web Services: Annotation and Matchmaking". In: *The Semantic Web - ISWC 2003: Second International Semantic Web Conference, Sanibel Island, FL, USA, October 20-23, 2003. Proceedings*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2003, pp. 335–350. ISBN: 978-3-540-39718-2. DOI: 10.1007/978-3-540-39718-2_22.

[123] C. Blanco, J. Lasheras, E. Fernández-Medina, R. Valencia-García, and A. Toval. "Basis for an integrated security ontology according to a systematic review of existing proposals". In: *Computer Standards & Interfaces* 33.4 (June 2011), pp. 372–388. ISSN: 09205489. DOI: 10.1016/j.csi.2010.12.002.

[124] I. Horrocks, P. F. Patel-Schneider, H. Boley, S. Tabet, B. Grosof, and M. Dean. *SWRL: A semantic web rule language combining OWL and RuleML*. Available online. 2004. URL: http://www.daml.org/rules/proposal/. (Accessed on 16 January 2020).

[125] E. Sirin, B. Parsia, B. C. Grau, A. Kalyanpur, and Y. Katz. "Pellet: A practical OWL-DL reasoner". In: *Web Semantics: Science, Services and Agents on the World Wide Web* 5.2 (2007), pp. 51 –53. ISSN: 1570-8268. DOI: 10.1016/j.websem.2007.03.004.

[126] A. Marotta, F. Martinelli, S. Nanni, A. Orlando, and A. Yautsiukhin. "Cyber-insurance survey". In: *Computer Science Review* 24 (2017), pp. 35–61. ISSN: 15740137. DOI: 10.1016/j.cosrev.2017.01.001.

[127] H. Abie. "Adaptive security and trust management for autonomic message - oriented middleware". In: *2009 IEEE 6th International Conference on Mobile Adhoc and Sensor Systems*. 2009, pp. 810–817. DOI: 10.1109/MOBHOC.2009.5336915.

[128] K Habib and W Leister. "Adaptive security for the Internet of Things reference model". In: *Proceeding of Norwegian Information Security Conference, NISK*. 2013, pp. 13–24.

[131] M. C. Huebscher and J. A. McCann. "A Survey of Autonomic Computing - Degrees, Models, and Applications". In: *ACM Comput. Surv.* 40.3 (Aug. 2008), 7:1–7:28. ISSN: 0360-0300. DOI: 10.1145/1380584.1380585.

[133] R. M. Savola, H. Abie, and M. Sihvonen. "Towards Metrics-driven Adaptive Security Management in e-Health IoT Applications". In: *Proceedings of the 7th International Conference on Body Area Networks*. BodyNets '12. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2012, pp. 276–281. ISBN: 978-1-936968-60-2.

[135] A. Soylu and P. D. Causmaecker. "Merging model driven and ontology driven system development approaches pervasive computing perspective". In: *2009 24th International Symposium on Computer and Information Sciences*. 2009, pp. 730–735. DOI: 10.1109/ISCIS.2009.5291915.

[136] A. Kalyanpur, D. J. Pastor, S. Battle, and J. A. Padget. "Automatic Mapping of OWL Ontologies into Java." In: *SEKE*. Vol. 4. Citeseer. 2004, pp. 98–103.

[137] A. Evesti and E. Ovaska. "Comparison of Adaptive Information Security Approaches". In: *ISRN Artificial Intelligence* (2013). DOI: 10.1155/2013/482949.

[139] R. Want, T. Pering, and D. Tennenhouse. "Comparing autonomic and proactive computing". In: *IBM Systems Journal* 42.1 (2003), pp. 129–135. ISSN: 0018-8670. DOI: 10.1147/sj.421.0129.

[140] S. VanSyckel and C. Becker. "A Survey of Proactive Pervasive Computing". In: *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct Publication*. UbiComp '14 Adjunct. ACM, 2014, pp. 421–430. ISBN: 978-1-4503-3047-3. DOI: 10.1145/2638728.2641672.

[141] Y. Engel, O. Etzion, and Z. Feldman. "A Basic Model for Proactive Event-driven Computing". In: *Proceedings of the 6th ACM International Conference on Distributed Event-Based Systems*. DEBS '12. ACM, 2012, pp. 107–118. ISBN: 978-1-4503-1315-5. DOI: 10.1145/2335484.2335496.

[152] R. Shahid, D. Simon, H. Joel, R. Utz, and V. Thiemo. "Secure communication for the Internet of Things - a comparison of link-layer security and IPsec for 6LoWPAN". In: *Security and Communication Networks* 7.12 (), pp. 2654–2668. DOI: 10.1002/sec.406.

[153] B. A. Mozzaquatro, R. Jardim-Goncalves, and C. Agostinho. "Situation awareness in the Internet of Things". In: *2017 International Conference on Engineering, Technology and Innovation (ICE/ITMC)*. 2017, pp. 982–990. DOI: 10.1109/ICE.2017.8279988.

[154] B. A. Mozzaquatro. *IoTSec Ontology*. Available online: http://iotsec.brunomozza.com/. (Accessed on 20 March 2020).

[155] T. Bagosi, D. Calvanese, J. Hardi, S. Komla-Ebri, D. Lanti, M. Rezk, M. Rodríguez-Muro, M. Slusnys, and G. Xiao. "The Ontop Framework for Ontology Based Data Access". In: *The Semantic Web and Web Science: 8th Chinese Conference, CSWS 2014, Wuhan, China, August 8-12, 2014, Revised Selected Papers*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2014, pp. 67–77. ISBN: 978-3-662-45495-4. DOI: 10.1007/978-3-662-45495-4_6.

[156] E. Prud'hommeaux and A. Seaborne. *SPARQL query language for RDF*. World Wide Web Consortium. Available online: www.w3.org/TR/rdf-sparql-query/. (Accessed on 19 March 2020).

[157] J. H. Gennari, M. A. Musen, R. W. Fergerson, W. E. Grosso, M. Crubézy, H. Eriksson, N. F. Noy, and S. W. Tu. "The evolution of Protégé: an environment for knowledge-based systems development". In: *International Journal of Human-Computer Studies* 58.1 (2003), pp. 89 –123. ISSN: 1071-5819. DOI: 10.1016/S1071-5819(02)00127-1.

[158] H. Bazoun, G. Zacharewicz, Y. Ducq, and H. Boyé. "SLMToolBox: An Implementation of MDSEA for Servitisation and Enterprise Interoperability". In: *Enterprise Interoperability VI*. Springer, 2014, pp. 101–111. DOI: 10.1007/978-3-319-04948-9_9.

[159] J. Koenig. "Jboss jBPM". 2004.

[160] P Ayuso. "Netfilter's connection tracking system". In: *LOGIN: The USENIX magazine* 31.3 (2006).

[161] Z. Zhou, C. Zhongwen, Z. Tiecheng, and G. Xiaohui. "The study on network intrusion detection system of Snort". In: *2010 International Conference on Networking and Digital Society*. Vol. 2. 2010, pp. 194–196. DOI: 10.1109/ICNDS.2010.5479341.

[162] P. OSS. *Prelude OSS*. Available online. URL: http://www.prelude-siem.com/en/products/prelude-os/. (Accessed on 16 January 2020).

[163] P. Kasinathan, G. Costamagna, H. Khaleel, C. Pastrone, and M. A. Spirito. "An IDS Framework for Internet of Things Empowered by 6LoWPAN". In: *Proceedings of the 2013 ACM SIGSAC Conference on Computer & Communications Security*. CCS '13. ACM, 2013, pp. 1337–1340. ISBN: 978-1-4503-2477-9. DOI: 10.1145/2508859.2512494.

[164] BeyondTrust. *Retina Network Security Scanner*. Available online. URL: https://www.beyondtrust.com/products/retina-network-security-scanner/. (Accessed on 16 January 2020).

[165] P. García-Teodoro, J. Díaz-Verdejo, G. Maciá-Fernández, and E. Vázquez. "Anomaly-based network intrusion detection: Techniques, systems and challenges". In: *Computers & Security* 28.1 (2009), pp. 18 –28. ISSN: 0167-4048. DOI: 10.1016/j.cose.2008.08.003.

[166] B. A. Mozzaquatro, R. Melo, C. Agostinho, and R. Jardim-Goncalves. "An ontology -based security framework for decision-making in industrial systems". In: *2016 4th International Conference on Model-Driven Engineering and Software Development (MODELSWARD)*. 2016, pp. 779–788. DOI: 10.5220/0005853107790788.

[167] B. A. Mozzaquatro, C. Agostinho, R. Melo, and R. Jardim-Goncalves. "A Model-Driven Adaptive Approach for IoT Security". In: *Model-Driven Engineering and Software Development*. Ed. by S. Hammoudi, L. F. Pires, B. Selic, and P. Desfray. Cham: Springer International Publishing, 2017, pp. 194–215. ISBN: 978-3-319-66302-9. DOI: 10.1007/978-3-319-66302-9_10.

[168] A. Duque-Ramos, J. T. Fernández-Breis, R. Stevens, N. Aussenac-Gilles, et al. "OQuaRE: A SQuaRE-based approach for evaluating the quality of ontologies". In: *Journal of Research and Practice in Information Technology* 43.2 (2011), pp. 159–176. ISSN: 1443-458X.

[169] A. Duque-Ramos, J. T. Fernández-Breis, M. Iniesta, M. Dumontier, M. Aranguren, S. Schulz, N. Aussenac-Gilles, and R. Stevens. "Evaluation of the OQuaRE framework for ontology quality". In: *Expert Systems with Applications* 40.7 (2013), pp. 2696 –2703. ISSN: 0957-4174. DOI: 10.1016/j.eswa.2012.11.004.

[170] A. Duque-Ramos, M. Boeker, L. Jansen, S. Schulz, M. Iniesta, and J. T. Fernández-Breis. "Evaluating the Good Ontology Design Guideline (GoodOD) with the Ontology Quality Requirements and Evaluation Method and Metrics (OQuaRE)". In: *PLOS ONE* 9.8 (Aug. 2014), pp. 1–14. DOI: 10.1371/journal.pone.0104463.

[171] A. Kozakevicius, C. Cappo, B. A. Mozzaquatro, R. C. Nunes, and C. E. Schaerer. "URL query string anomaly sensor designed with the bidimensional Haar wavelet transform". In: *International Journal of Information Security* 14.6 (2015), pp. 561–581. ISSN: 1615-5270. DOI: 10.1007/s10207-015-0276-y.

[172] G. R. Librelotto, L. O. Freitas, A. Fiorin, B. A. Mozzaquatro, L. Pasetto, R. G. Martini, R. P. Azevedo, and R. T. Pereira. "OntoHealth: a system to process ontologies applied to health pervasive environment". In: *International Journal of Computational Science and Engineering* 10.4 (2015), pp. 359–367. DOI: 10.1504/IJCSE.2015.070997.

[173] B. A. Mozzaquatro, R. Jardim-Goncalves, R. Melo, and C. Agostinho. "The application of security adaptive framework for sensor in industrial systems". In: *2016 IEEE Sensors Applications Symposium (SAS)*. 2016, pp. 1–6. DOI: 10.1109/SAS.2016.7479838.

[174] B. A. Mozzaquatro, R. Jardim-Goncalves, and C. Agostinho. "Model Driven Implementation of Security Management Process". In: *Proceedings of the 5th International Conference on Model-Driven Engineering and Software Development (MODELSWARD)*. Porto, Portugal, 2017, pp. 229–238. DOI: 10.5220/0006329602290238.

[175] J. C. Nobre, B. A. Mozzaquatro, and L. Z. Granville. "Network-Wide Initiatives to Control Measurement Mechanisms: A Survey". In: *IEEE Communications Surveys Tutorials* (2018). DOI: 10.1109/COMST.2018.2797170.

[176] B. A. Mozzaquatro, C. Agostinho, D. Goncalves, J. Martins, and R. Jardim-Goncalves. "An Ontology-Based Cybersecurity Framework for the Internet of Things". In: *Sensors* 18.9 (2018). ISSN: 1424-8220. DOI: 10.3390/s18093053.

[177] J. A. Bonini, M. D. Da Silva, R. Pereira, B. A. Mozzaquatro, R. G. Martini, and G. R. Librelotto. "An Application of Ontological Engineering for Design and Specification of Ontocancro". In: *Practical Applications of Computational Biology & Bioinformatics, 14th International Conference (PACBB 2020)*. Ed. by G. Panuccio, M. Rocha, F. Fdez-Riverola, M. S. Mohamad, and R. Casado-Vara. Cham: Springer International Publishing, 2021, pp. 134–143. ISBN: 978-3-030-54568-0. DOI: 10.1007/978-3-030-54568-0_14.

[178] C2NET. *EU Project*. Available online. URL: http://c2net-project.eu/. (Accessed on 16 January 2020).

[179] B. Andres, R. Sanchis, and R. Poler. "A cloud platform to support collaboration in supply networks". In: *International Journal of Production Management and Engineering* 4.1 (2016), pp. 5–13. DOI: 10.4995/ijpme.2016.4418.

[180] A. Bröring, S. Schmid, C. K. Schindhelm, A. Khelil, S. Käbisch, D. Kramer, D. L. Phuoc, J. Mitic, D. Anicic, and E. Teniente. "Enabling IoT Ecosystems through Platform Interoperability". In: *IEEE Software* 34.1 (2017), pp. 54–61. ISSN: 0740-7459. DOI: 10.1109/MS.2017.2.

[181] K. E. Skouby and P. Lynggaard. "Smart home and smart city solutions enabled by 5G, IoT, AAI and CoT services". In: *2014 International Conference on Contemporary Computing and Informatics (IC3I)*. 2014, pp. 874–878. DOI: 10.1109/IC3I.2014.7019822.

[182] S. Zhao, S. Li, L. Qi, and L. D. Xu. "Computational Intelligence Enabled Cybersecurity for the Internet of Things". In: *IEEE Transactions on Emerging Topics in Computational Intelligence* 4.5 (2020), pp. 666–674. DOI: 10.1109/TETCI.2019.2941757.

[183] H. Sedjelmaci, F. Guenab, S. Senouci, H. Moustafa, J. Liu, and S. Han. "Cyber Security Based on Artificial Intelligence for Cyber-Physical Systems". In: *IEEE Network* 34.3 (2020), pp. 6–7. DOI: 10.1109/MNET.2020.9105926.

[184] K. Veeramachaneni, I. Arnaldo, V. Korrapati, C. Bassias, and K. Li. "AI^2: Training a Big Data Machine to Defend". In: *2016 IEEE 2nd International Conference on Big Data Security on Cloud (BigDataSecurity), IEEE International Conference on High Performance and Smart Computing (HPSC), and IEEE International Conference on Intelligent Data and Security (IDS)*. 2016, pp. 49–54. DOI: 10.1109/BigDataSecurity-HPSC-IDS.2016.79.

[185] A. Mikroyannidis and B. Theodoulidis. "Ontology management and evolution for business intelligence". In: *International Journal of Information Management* 30.6 (2010), pp. 559 –566. ISSN: 0268-4012. DOI: 10.1016/j.ijinfomgt.2009.10.002.

[186] D. E. Forbes, P. Wongthongtham, C. Terblanche, and U. Pakdeetrakulwong. "Ontology Engineering". In: *Ontology Engineering Applications in Healthcare and Workforce Management Systems*. Cham: Springer International Publishing, 2018, pp. 27–40. ISBN: 978-3-319-65012-8. DOI: 10.1007/978-3-319-65012-8{_}3.

[187] V. Mavroeidis and S. Bromander. "Cyber Threat Intelligence Model: An Evaluation of Taxonomies, Sharing Standards, and Ontologies within Cyber Threat Intelligence". In: *2017 European Intelligence and Security Informatics Conference (EISIC)*. 2017, pp. 91–98. DOI: 10.1109/EISIC.2017.20.

[188] C. L. Jones, R. A. Bridges, K. M. T. Huffer, and J. R. Goodall. "Towards a Relation Extraction Framework for Cyber-Security Concepts". In: *Proceedings of the 10th Annual Cyber and Information Security Research Conference*. CISR '15. Oak Ridge, TN, USA: ACM, 2015, 11:1–11:4. ISBN: 978-1-4503-3345-0. DOI: 10.1145/2746266.2746277.

[189] H. Abie and I. Balasingham. "Risk-based Adaptive Security for Smart IoT in eHealth". In: *Proceedings of the 7th International Conference on Body Area Networks*. BodyNets '12. Oslo, Norway: ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2012, pp. 269–275. ISBN: 978-1-936968-60-2.

[190] A. H. Mohammed, R. M. KHALEEFAH, M. k. Hussein, and I. Amjad Abdulateef. "A Review Software Defined Networking for Internet of Things". In: *2020 International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA)*. 2020, pp. 1–8. DOI: 10.1109/HORA49412.2020.9152862.

[191] Ángel Leonardo Valdivieso Caraguay, A. B. Peral, L. I. B. López, and L. J. G. Villalba. "SDN: Evolution and Opportunities in the Development IoT Applications". In: *International Journal of Distributed Sensor Networks* 10.5 (2014), p. 735142. DOI: 10.1155/2014/735142.

[192] M. A. Salahuddin, A. Al-Fuqaha, M. Guizani, K. Shuaib, and F. Sallabi. "Softwarization of Internet of Things Infrastructure for Secure and Smart Healthcare". In: *Computer* 50.7 (2017), pp. 74–79. ISSN: 0018-9162. DOI: 10.1109/MC.2017.195.

[193] M. Liyanage, I. Ahmad, J. Okwuibe, et al. "Software Defined Security Monitoring in 5G Networks". In: *A Comprehensive Guide to 5G Security*. John Wiley & Sons, 2018, p. 231.