tecnun
Universidad
de Navarra

Pº Manuel Lardizabal, 13.
20018 Donostia-San Sebastián, Spain
Tel.943 219 877
Fax 943 311 442
www.tecnun.es

Santiago Figueroa Lorenzo TOWARDS DECENTRALIZED AND SCALABLE ARCHITECTURES FOR ACCESS CONTROL SYSTEMS FOR IIoT SCENARIOS

TOWARDS DECENTRALIZED AND SCALABLE ARCHITECTURES FOR ACCESS CONTROL SYSTEMS FOR IIoT SCENARIOS

DISSERTATION
submitted for the Degree of Doctor of Philosophy by

Santiago Figueroa Lorenzo
under the supervision of
Saioa Arrizabalaga Juaristi and
Javier Añorga Benito

Donostia-San Sebastián, September 2021

tecnun
Universidad
de Navarra

TOWARDS DECENTRALIZED AND SCALABLE ARCHITECTURES FOR ACCESS CONTROL SYSTEMS FOR IIoT SCENARIOS

Ph.D. dissertation

Santiago Figueroa Lorenzo

Asociación Centro Tecnológico CEIT
Information and Communication Technologies Division
Data Analysis and Information Management Group
TECNUN, University of Navarra

Donostia-San Sebastián, September 2021

TOWARDS DECENTRALIZED AND SCALABLE ARCHITECTURES FOR ACCESS CONTROL SYSTEMS FOR IIoT SCENARIOS

Dissertation submitted for the Degree of Doctor of Philosophy

Under the supervision of

Saioa Arrizabalaga Juaristi

Javier Añorga Benito

Asociación Centro Tecnológico CEIT
Information and Communication Technologies Division
Data Analysis and Information Management Group
TECNUN, University of Navarra

Donostia-San Sebastián, September 2021

De utilitate credendi
Augustine of Hippo

AGRADECIMIENTOS

Acaba esta etapa de la que sólo tengo gratos recuerdos y tengo que agradecer a la Asociación Centro Tecnológico Ceit y la Universidad de Navarra la oportunidad de haber realizado esta Tesis, que me ha hecho mejor profesional y persona en muchos aspectos. Hace unos años emprendí un camino sin la certeza de a lo que me enfrentaba y he encontrado en Ceit y Tecnun, una segunda casa, y sobre todo amigos e increíbles personas que han hecho estos años maravillosos para mí.

En primer lugar, quiero agradecer a mis directores de tesis Saioa Arrizabalaga y Javier Añorga su confianza, apoyo, paciencia, ayuda y cariño durante estos años. Ha sido un placer compartir esta etapa junto a ustedes. Saioa, gracias por la oportunidad y gracias por apostar por mí. Javi, gracias por cada día motivarme y mostrarme el valor y el camino del conocimiento.

En segundo lugar, quiero agradecer especialmente a un conjunto de personas de Ceit que a lo largo de esta etapa han estado muy cerca, haciendo cada día especial. A Ion Irizar agradecer su apoyo, enseñanzas y sinceridad en todo momento, ha sido una maravilla trabajar a su lado y poder contar con su sabiduría. A Jon Goya, agradecer su paciencia e innumerables consejos dedicados. Gracias también por la oportunidad de compartir asignatura, y ayudarme a crecer también en el ámbito docente en los dos años de trabajo. Agradecer también a todas y cada una de las personas que integran o han integrado el fantástico grupo de Análisis de Datos y Gestión de la Información (DAIM) a lo largo de estos años. Particularmente a los integrantes del grupo de WhatsApp @Tiernos divirtiéndose: Mario Monterde, Meritxell Arsuaga, Itxaro Errandonea, Eneko Lizardi, Ioanes Ceballos, Jaione Arrizabalaga, Urko Larrañaga e Ibai Alberdi. Os habéis convertido en más que compañeros de trabajo: en amigos. Agradecer a Damián Caballero por todo el cariño recibido. Nunca olvidaré esa Nochebuena en La Habana compartiendo en familia. Agradecer a Miriam Garmendia, Isabel Gastaminza, Belén Berdeal y Amaya Alcorta, las primeras personas que día tras día encuentro al llegar a Ceit. Gracias, siempre habéis tenido una sonrisa; lo mejor que se puede dar para iniciar cada día. Quiero agradecer a Patricio Smith, "el gran Patri", por toda la ayuda que me ha brindado. A tod@s, gracias por tanta gentileza y cariño.

En tercer lugar, agradecer a las siguientes personas de Tecnun: a Héctor Solar y a Roc Berenguer por la oportunidad y confianza depositada para formar parte del claustro de

profesores del Máster de Telecomunicaciones, a Dani Valderas por los innumerables consejos dedicados y finalmente, a Manu Sánchez y Javi García Muñoz por todo el cariño. Adicionalmente, quiero reconocer especialmente a Iker Muñiz, Migsheng Yin, Xabier Eizaguirre, Jon Alberdi, Mikel Culla, Jianni Song y Jon Corral el haber decidido realizar sus Proyectos de Final de Grado conmigo. Sus trabajos han contribuido en mayor o menor medida con algún aspecto de esta Tesis.

En un plano más personal, quiero agradecer a Jon Lecanda. Nunca olvidaré que fue la primera persona con la que compartí en Donostia. Gracias por las enseñanzas, consejos, paseos y buenos momentos. Agradecer a Dani Fernández, Héctor Solar, Joseba Meana, y demás "compis" que me acogieron como uno más al llegar a esta maravillosa ciudad que es Donostia. Hicieron que esos primeros seis meses fueran algo realmente inolvidable.

Por último, a mi familia en su conjunto y en especial, agradecer a mi hermano Andy, no importa lo lejos que estemos, te quiero, y sé que tienes un gran futuro por delante. A por ello máquina, a comerse el mundo. Agradecer a mis abuelos Lorenzo y Marta desde la distancia. Sé que no podréis leer estas líneas, pero estáis cada segundo en mi corazón. Agradecer a mi "papi" Santiago, por el ejemplo de esfuerzo y sacrificio que representa. Quiero dedicar unas breves líneas a nuestra preciosa mascota Bella por ser cada día un motivo de alegría y felicidad. Agradecer a Keila, mi esposa, mujer, amiga, confidente... Gracias por apoyarme cada segundo de mi vida, gracias por estar ahí a cada instante en ese sin fin de locuras que es nuestra vida. Gracias por ese "que locos somos …, hasta donde andamos …". Sin ti este camino hubiera sido imposible de recorrer. Por último y más importante, a mi persona, mi apoyo, mi amiga y confidente, mi ejemplo; la persona a la que le debo todo lo bueno que me pasa y sin la que no me imagino la vida, gracias a ti mami, eres lo que más quiero en el mundo.

Santiago Figueroa Lorenzo, Donostia, 07 de septiembre de 2021

SUMMARY

The Industrial Internet of Things (IIoT) architecture is complex due to, among other things, the convergence of protocols, standards, and buses from such heterogeneous environments as Information Technology (IT) and Operational Technology (OT). IT – OT convergence not only makes interoperability difficult but also makes security one of the main challenges for IIoT environments. In this context, this thesis starts with a comprehensive survey of the protocols, standards, and buses commonly used in IIoT environments, analyzing the vulnerabilities in assets implementing them, as well as the impact and severity of exploiting such vulnerabilities in IT and OT environments. The Vulnerability Analysis Framework (VAF) methodology used for risk assessment in IIoT environments has been applied to 1,363 vulnerabilities collected from assets implementing the 33 protocols, standards and buses studied.

On the other hand, Access Control Systems emerges as an efficient solution to mitigate some of the vulnerabilities and threats in the context of IIoT scenarios. Motivated by the variety and heterogeneity of IIoT environments, the thesis explores different alternatives of Access Control Systems covering different architectures. These architectures include Access Control Systems based on traditional Authorization policies such as Role-based Access Control or Attribute-based Access Control, as well as Access Control Systems that integrate other capabilities besides Authorization such as Identification, Authentication, Auditing and Accountability. Blockchain technologies are incorporated into some of the proposals as they enable properties not achievable in centralized architectures, at different levels of complexity: they can be used just as a verifiable data registry, executing simple off-chain authorization policies, up to scenarios where the blockchain enables on-chain an Identity and Access Management System, based on Self-Sovereign Identity.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER IV

CHAPTER V

CHAPTER VI

LIST OF TABLES

GLOSSARY

Notation	Description
ABAC	Attribute-Based Access Control
ABACP	Attribute-Based Access Control Policy
ACH	Authentication Chaincode
ACL	Access Control List
ACS	Access Control System
ACPE	Access Control Policy Execution
AECH	Attribute-Based Access Control Execution Chaincode
AIC Triad	Triad Availability, Integrity, and Confidentiality
AMRP	Automatic Meter Reading Protocol
AO	Attribute Object
AP	Attribute for Permission
APCI	Application Protocol Control Information
APDU	Application Protocol Data Unit
AppUT	Application Under Test
ArchUT	Architecture Under Test
AS	Attribute Subject
ASDU	Application Service Data Unit
APL	Application Layer
AuthN	Authentication
AuthZ	Authorization
AI	Artificial Intelligence
AM	Access Management
AWS	Amazon Web Service
BEX	Base Exploitation Level
BLE	Bluetooth Low Energy
BSL	Base Severity Level
BTL	Bluetooth
BAP	Building Automation Protocol
CA	Certificate Authority
CAPEC	Common Attack Pattern Enumeration and Classification
CIA Triad	Triad: Confidentiality, Integrity, and Availability
CPE	Common Platform Enumeration
CIoT	Cellular Internet of Things
CP	Channel Privacy
CPS	Cyber Physical System
CVE	Common Vulnerabilities and Exposures
CVD	Coordinated Vulnerability Disclosure

CWE	Common Weakness Enumeration
DBMS	Database Management System
DiD	Defense in Depth
DLL	Data Link Layer
DLT	Distributed Ledger Technology
EC2	Elastic Compute Cloud
ECU	Electronic Control Unit
EPS	Evolved Packet System
E-UTRAN	Evolved UMTS Terrestrial Radio Access Network
FAP	Factory Automation Protocol
FWK	Framework Layer
GSSE	Generic Substation State Events
HF	High Frequency
HFB	Hyperledger Fabric Blockchain
ICS	Industrial Control Systems
ICSP	Industrial Control System Protocol
IAAA	Identification, Authentication, Authorization and Accountability
IM	Identity Management
IAM	Identity and Access Management
IIoT	Industrial Internet of Things
IoT	Internet of Things
IPM	Information Privacy Management
ICS	Industrial Control System
ICT	Information and Communication Technologies
IT	Information Technologies
IVSS	Industrial Vulnerability Scoring System
LF	Low Frequency
MAC	Media Access Control Layer
MWF	Microwave Frequency
M2M	Machine to Machine
NWK	Network Layer
NFC	Near Field Communication
NIST	National Institute of Standards and Technology
NIDD	Non-IP Data Delivery
NTFS	New Technology File System
OOB	Out of Band
OSN	Ordering Service Network
OT	Operation Technologies
PA	Performance Analysis
PAP	Process Automation Protocol
PAN	Personal Area Network
PCH	Policy Chaincode
PDC	Private Data Collection

PDLM	Private Data Local Management
PHY	Physical Layer
PoW	Proof of Work
PROFINET	PROcess FIeld NETwork
PSAP	Power System Automation Protocol
PT	Public Transactions
RBAC	Role-Based Access Control
RCH	Registration Chaincode
RFID	Radio Frequency Identification
SER	Shared External Resources
SDK	Software Development Kit
SGW	Smart Gateway
SRN	Stochastic Reward Nets
SELinux	Security-Enhanced Linux
SHAP	Smart Home Automation Protocol
SSL	Secure Socket Layer
SUT	System Under Test
StateDB	State Database
TBAC	Transaction-Based Access Control
TEMSL	Threat, Exposure, Mission, Safety and Loss
TLS	Transport Layer Secure
TRT	Transport Layer
UHF	Ultra-High Frequency
VAF	Vulnerability Analysis Framework
VAP	Vehicular Automation Protocol
ZB	ZigBee
ZBHA	ZB Home Automation
ZBHC	ZB Health Care
ZBBA	ZB Building Automation

CHAPTER I

INTRODUCTION AND OBJECTIVES

This chapter includes the context where the thesis has been carried out. It starts by introducing the overall scope of security in the Industrial Internet of Things (IIoT) which is the focus of this thesis. Then, the state-of-the-art background related to the convergence of Information Technologies (IT) and Operational Technologies (OT), Access Control Systems and blockchain as distributed ledger technologies is described. The thesis motivation is then highlighted, to establish the main objectives of this work. The scientific contributions that have targeted these objectives are introduced, together with the thesis outline. Finally, the projects that have supported this research are referred.

1. OVERVIEW

The accelerated development of Information and Communication Technologies (ICT) has resulted in the transformation of the industrial production, traditionally based on Operation Technology (OT), towards the incorporation of information technologies (IT) to achieve further automation of manufacturing into the industry 4.0 paradigm, which attempts to build a controllable, credible, scalable, secure and efficient interaction of physical devices in the industrial scene, trying to fundamentally change the traditional human understanding of the industry. Industry 4.0 is merely a term for the fourth industrial revolution, which includes interoperability, autonomy, information transparency, technical assistance and distributed decisions, making traditional manufacturing smart, representing, in addition, a broader concept that encompasses both smart manufacturing and the Industrial Internet of Things (IIoT) [1].

IIoT is defined by Enisa as IoT applied in the industrial environment [1]. IIoT represents a subset of IoT, encompassing the domains of machine-to-machine (M2M) and industrial communication technologies (ICT). IIoT, therefore, is paving the way for enhanced manufacturing process performance, improving operational efficiency. In this regard, the enabling technologies that ensure the industrial transformation represented by IIoT, integrating virtual space with the physical world, are mainly cyber-physical systems (CPS) and the Internet of Things (IoT). CPS interconnects physical assets with computational capabilities, while IoT can be considered a global network infrastructure composed of several connected devices based on sensory, communication, networking and information processing technologies. By the end of 2020, IIoT projected more than 10 billion devices, accounting for 57% of IoT spending [2].

In particular, the role of IIoT is to connect all industrial assets, including machines and control systems, with the corresponding information systems and business processes, i.e., to adapt business and operating models by integrating information technologies (IT) and operation technologies (OT). IIoT devices are therefore responsible for monitoring, collecting, exchanging and analyzing information so that they can control their behavior without human intervention. This huge amount of data collected can feed analytical solutions and lead to optimal industrial operations and all that information needs to be collected and analyzed to create significant information for future decision-making.

Despite the benefits of IIoT for Industry 4.0, the technology faces several challenges in terms of energy management, evolution to new generations of industrial systems to cope with the

complexity of production in cyber-physical environments and the capacity to perform diagnoses, such as health management of machine tools, enhancing productivity and increasing the quality of processes and services, based on data analytics and artificial intelligence (AI). However, one of the main challenges presented by the IIoT is security. The scientific literature is clear in highlighting, on the one hand, that data confidentiality, cyber-physical systems integrity and device management are challenges inherited from the IoT [3]. On the other hand, it defines specific challenges related to industrial security, such as the security of industrial control systems (ICS), connectivity between field devices and gateways, the increasing integration of IT monitoring of physical production processes, but above all, the fact that IIoT has blurred the traditional boundaries of IT and OT infrastructures, enabling the convergence of both environments [4]. In this regard, the threats and vulnerabilities, not only of design but also of implementation and configuration, in both IoT and IIoT environments have been properly documented [5].

Considering that the severity of these threats on the IIoT environment can be characterized around three tenets: confidentiality, integrity and availability, so that, on the one hand, the impact of vulnerability exploitation can be measured, depending on the security requirements of an organization from them and on the other hand, these three tenets, allow to evaluate the security controls, i.e., the countermeasures that are adopted to protect IIoT environments, where, access control system emerges as one of the few common solutions for the protection of these three tenets [6]. In this sense, Access Control Systems can be part of scenarios that include only authorization processes, i.e., the data protection against unauthorized access, use, or disclosure while in storage, in-process or in transit and therefore considered simply as enforcement of access control model, as well as scenarios consisting of Access Control Systems, where in addition to the usual authorization, the phases of identification, authentication and accountability are involved.

Additionally, these Access Control Systems are integrated into the blockchain, which is a disruptive technology consisting of states, operations and transactions replicated in a distributed system, on which security is guaranteed based on cryptographic techniques both to preserve the integrity (e.g. digital signatures) and to preserve privacy, which in turn, uses protocols for the establishment of consensus between the operations performed by the nodes of the distributed network and finally presents a business layer that represents the logic embedded in the peers, e.g., smart contracts that ensure the development of decentralized applications, enabling decentralized Access Control Systems.

The next section provides more detailed background on the following three topics: IT-OT convergence, Access Control Systems and blockchain as a Distributed Ledger Technology.

2. BACKGROUND

After establishing security in IIoT environments as the global scope of the thesis, this section defines the technologies and their context to build a background, allowing to establish, in turn, the motivation for the development of this doctoral thesis. For this reason, firstly, the IT - OT convergence is defined in section 2.1, as one of the most important challenges of security in IIoT environments. Secondly, section 2.2 analyzes Access Control Systems based on the application of access control models and their integration as part of Access Control Systems involving identification, authentication, auditing and accountability to the existing authorization process. Finally, section 2.3 introduces blockchain as a disruptive technology, as well as the integration of access control and identity management systems into the blockchain, which is one of the pillars of this Ph.D. Thesis.

2.1. IT-OT CONVERGENCE: A CHALLENGE FOR IIOT ENVIRONMENTS SECURITY

The increase in computing power, as well as ubiquitous connectivity and the evolution of data analysis techniques, paved the way for the convergence of controls systems, i.e., operational technologies (OT) and business systems, i.e., information technologies (IT) over the Internet. This convergence, defined as IT-OT convergence, integrates IT systems applied to data-centric computing with OT systems used to monitor events, processes and devices and adjust in enterprise and industrial operations. IT is composed of those hardware and software systems that allow for corresponding information processing. OT is supported by physical devices, i.e., switches, sensors, power distribution networks, valves, motors and software that allow for control and monitoring of a plant and its associated equipment [7]. This convergence has increased the productivity and efficiency of operational processes. However, it has led to systems designed to be isolated being exposed to attacks with a higher impact when considering the operational side of IIoT environments. Significant aspects include unprotected network connections, deployment of technologies with identified vulnerabilities that bring together formerly unidentified risks into the OT environment and inadequate perception of requirements for ICS settings [8]. These scenarios can result in the brand and reputational damage, material financial loss and potential damage to critical infrastructures.

In this sense, because of their importance in the context of security management, confidentiality, integrity, and availability evaluate both the vulnerabilities and risks as well as the security controls applicable to a given context. These three principles are considered the most important ones within the security domain, although the importance of each principle

for a particular organization depends on the organization's security objectives and requirements. In this way, through these three parameters, it is possible to give a perspective of the impact of IT-OT convergence. Thus, IT systems, tend to follow the CIA triad, prioritizing confidentiality and integrity over availability; however, OT systems tend to follow the AIC triad, where availability, in general, is prioritized and integrity is valued over confidentiality. Considering that the tenets of confidentiality, integrity and availability are used not only to measure the impact of an exploited vulnerability but also to evaluate the security controls to enable IIoT protection, authorized voices such as the "industrial internet consortium" define access control as a fundamental part of security management in IIoT environments, both at the level of communication and connectivity protection as well as at the level of data protection and endpoint protection [9]. Therefore, access control constitutes a common countermeasure to ensure confidentiality as well as integrity and availability [6].

The following section provides a brief overview of the state-of-the-art Access Control Systems and their role to secure IIoT environments.

2.2. ACCESS CONTROL SYSTEM

Section 2.2.1 includes the first approach of traditional Access Control Systems, focused on authorization through enforcement of access control policies. A second approach, through section 2.2.2, analyzes Access Control Systems evolution, integrating not only authorization policies execution but also other phases of access management, such as identification, authentication and accountability. Finally, section 2.2.3 highlights the value of Access Control Systems with a focus on IIoT scenarios.

2.2.1. ACCESS CONTROL SYSTEM: A TRADITIONAL APPROACH

Objects are defined as passive elements in a security relationship, e.g., files, computers, network connections and applications and subjects are active elements in a security relationship, e.g., users, programs and computers. Then, access control is, by definition, the management of the relationship between subjects and objects. Thus, it is possible to define Access Control Systems in terms of access control policies, models and mechanisms, restricting their function to enforcing those policies. The access control policies are characterized as high-level requirements that specify how and when a user or a process, can access a resource and are enforced through an access control mechanism, which is responsible for granting or denying access. In this regard, an access control model is a collection of implementations of access control mechanisms, based on established access

control policies [10]. Therefore, access control models are one of the main security measures for systems and assets, which by default is associated only with authorization, i.e., the execution of security policies. Compared to authentication and identity management, which typically constitutes the primary defense layer, access control usually appears as the second defense layer. In that sense, an essential requirement of access control architectures is their ability to support changing policies. Therefore, an important consideration is that the system has to be as flexible as possible to be able to dynamically grant and define privileges based on attribute values in real-time. In general, four main architectures of access controls can be used to control access to resources [11]:

1. Attribute or Object-based Access Control (OBAC or ABAC) models use rules that can include several attributes. This distinctive feature enables more flexibility than others Access Control Models.
2. Role-based Access Control (RBAC) assigns system policies to user roles or groups.
3. Rule-based Access Control (RuBAC) applies global rules to all subjects. These rules can be seen as restrictions or filters.
4. Discretionary Access Control (DAC) is an access control model where every object has an owner and this owner can grant or deny access to any other subject.
5. Mandatory Access Control (MAC) is a rule-based system for restricting access, where labels are applied to both subjects and objects.

Table 1: Common access control models.

Access Control Model	Applications/Environments	References
ABAC/OBAC	Software-Defined Networks (e.g., CloudGenix software-defined wide area network (SD-WAN))	[12]
RBAC	Microsoft Windows groups, Microsoft Azure, AWS, OpenStack	[13]
RuBAC	Firewalls, ACLs	[6]
DAC	NTFS	[14]
MAC	Linux Security Modules (e.g., AppArmor, SELinux)	[15]

Considering adaptability as the reference property of the access control models, Table 1 lists these five access control models, establishing also common applications where they are used.

2.2.2. Access Control System based on IAAA

As mentioned in section 2.2.1, Access Control Systems by default are commonly associated with authorization. Nevertheless, with the appearance of paradigms such as cloud computing and IoT, authentication and authorization technologies have become key elements to

overcome security and privacy issues. For an access control system to currently satisfy the main security requirements, i.e., confidentiality, integrity and availability, it must include not only the aforementioned functions of authentication and authorization but also other functions such as identification and accountability. In this regard, the IAAA comprises Identification, Authentication, Authorization and Accountability and is also known as access management [16]. Each IAAA component includes the following phases [6]:

1. Identification is the initial step involved in any request for getting access to an object. It is the process in which the subject provides an identification, e.g., username.
2. Authentication allows to verify the identity, ensuring that it is the one it claims to be, the preregistered information is stored and compared when access to an object is requested.
3. Authorization: Identification and authentication allow subjects to access the system. However, not all subjects can access all parts of the objects. A common approach is, for instance, the use of roles (RBAC) to authorize access to elements of an object, granting certain privileges to an authenticated identity.
4. Accountability enables the tracking of unauthorized identity access from auditing mechanisms such as logging. Logs enable the tracking of identity activity and, through identification and authentication processes, can demonstrate accountability.

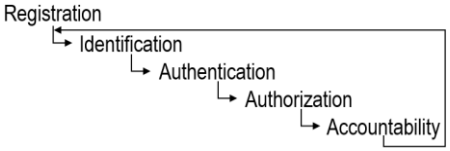

Figure 1: Stages in Access Control Systems based on IAAA.

In addition to these factors, there are other two factors in the Access Control Systems. On the one hand, auditing is the capability that ensures the recording of logs from subject events related to the systems and therefore constitutes a fundamental piece of the accountability process. On the other hand, the registration process is the other capability to be considered and occurs when a user, device, or thing is assigned an identity [6]. Figure 1 shows an interaction between all components.

2.2.3. ACCESS CONTROL SYSTEMS IN IIOT

In section 2.1, access control has been introduced as a key piece of security in IIoT environments, both at the level of communication and connectivity protection, as well as at the level of data and endpoint protection. To justify this statement in this section, based on

these three levels, further examples of Access Control Systems for IIoT environments are analyzed, divided into, Access Control Systems based only on the execution of authorization policies and, Access Control Systems including other phases such as identification, authentication, auditing and accountability. It should be noted that for the second case, scenarios that integrate Authentication and Authorization (AA) and Authentication, Authorization and Accountability (AAA) are also summarized. Table 2 includes examples of Access Control Systems are applicable in IIoT environments for different IIoT ecosystem level.

Table 2: Access control systems in IIoT environments.

IIoT Ecosystem Levels	Access Control System				
	Access Control Model				AA, AAA, IAAA
	ABAC	RBAC	RuBAC	Access Control Scheme	
				CP-ABE	
Endpoint		[17]			[18]
Data	[19], [20]			[21]	[22], [23]
Communication & Connectivity			[24]		[24]

From the scientific literature studied there are some cases of Access Control Systems designed for IoT environments, which are considered applicable to IIoT environments. In that sense, the RFID-enabled Supply Chain (ReSC-2) solution aims to protect endpoint devices in an IoT context [17]. ReSC-2 addresses security challenges such as device authentication, hardware theft and the integration of access control models, such as RBAC, into the IoT hardware. The ReSC-2 solution is fully compliant with the IoT environment. In addition, the authors of the reference [18] design and implement an access control system solution to authenticate and authorize both endpoints and gateways. The proposed solution also verifies the identity of the user that controls the endpoints via Real-Time Identity Monitor. Moreover, the authors of the reference [19] have proposed a system to ensure data protection in case the IoT device suffers loss or theft based on an ABAC access control model, which provides flexible policies' enforcement for unauthorized access detection. Furthermore, a framework design for IoT data management, which is also fully compliant with IIoT environments, has been proposed by the authors of the reference [22]. This framework includes an access control mechanism that enables both authentication and authorization via access control list (ACL), i.e., RuBAC as an access control model. The authors describe data privacy, data classification and confidentiality as the main strength of the framework. Additionally, a Simple Messaging and Access Control (SMAC) is designed by the authors of the reference [24], to prevent unauthorized communications in IoT

environments, where access control lists (ACL) are used as access control mechanism (i.e., a RuBAC from the classification provided in Table 1).

Additionally, the review of the scientific literature determined a set of scenarios where Access Control Systems have been applied directly over IIoT environments. Thus, a comprehensive Access control framework (CACF) is designed to ensure service privacy and protect data confidentiality [20]. The CACF is based on an ABAC model and has been prototyped based on Apache Active MQ, demonstrating that latency results are compatible with SCADA services' requirements. In addition, an access control scheme for fog IIoT (called SAC-FIIoT) has been proposed not only to protect endpoints, since it allows gateway securely to access endpoints, but also to establish a secure session to exchange data between gateways and endpoints in a secure way [23].

The Ciphertext-Policy Attribute-based Encryption (CP-ABE) is described as a technique that associates user encryption keys to attributes and employs an access policy to encrypt data. This policy is a tree-based structure composed of descriptive attributes related through logical operators (e.g., AND, OR) that determine who may access the data in plaintext. The CP-ABE primitive is natively designed to enforce access control on the data itself, offering a more intuitive and secure way of sharing information [26]. CP-ABE is included in this analysis because of its growing importance, as an access control model for IIoT environments, concerning data protection [25]. In this regard, the authors of the reference [21] design a secure industrial data access control scheme for cloud-assisted IIoT, which enables participants to enforce access control policies for their data at rest via CP-ABE. The CP-ABE scheme guarantees privacy protection to prevent leakage problems.

Despite the works mentioned above represent important contributions regard Access Control Systems, in general, they focus on design, most of them lacking both implementation and evaluation phases.

Although the analysis performed so far has been agnostic to who or what has root access to change permissions on the system, traditional Access Control Systems (e.g., examples included in Table 2) have a centralized approach, which implies that access rights are granted by a centralized entity which constitutes a single point of failure. Unfortunately, traditional Access Control Systems are not suitable for some IIoT scenarios mainly due to a massive scale, ubiquitous connectivity and distributed nature [27]. In contrast, decentralized Access Control Systems lack administrators who manage or grant access to individual users, devices, or things, which control their credentials. Theoretically, without a centralized root access point, decentralized Access Control Systems are supposed to be hack-proof. To further

study decentralized Access Control Systems the section 2.3 discusses, on the one hand, the details of blockchain technology and on the other hand its contribution to Access Control Systems, which constitutes one of the objects of this Ph.D. Thesis.

2.3. BLOCKCHAIN AS DISTRIBUTED LEDGER TECHNOLOGY

Blockchain belongs to distributed ledger technology (DLT) family. The disruption of a DLT lies in the fact that the ledger database is distributed, spread across all nodes in the network. Each node has an identical copy of the ledger, which is updated independently by all nodes. In this regard, the nodes participating in the network reach an agreement to determine a true copy for the ledger through a procedure called consensus. Once consensus is reached, the DLT is automatically updated and the agreed true copy of the ledger is added to each node separately. Particularly, a blockchain network is a P2P network of computers that run the blockchain protocol, allowing them to maintain a copy of the ledger which includes transactions, grouped into chained blocks to each other, starting from a genesis block. This inclusion of blocks in the chain is agreed by consensus, which dictates the rules and participates in the validation of the transactions. Once consensus is reached, the blocks are added to the ledger. This process is performed without intermediaries' participation [29].

To connect the components mentioned throughout the blockchain definition, section 2.3.1 organizes them around a reference architecture. Next, section 2.3.2 defines blockchain types, while section 2.3.3 provides the details of the smart contract. Relating blockchain types to the processing involved in developing smart contracts, section 2.3.4 details the types of blockchain applications and section 2.3.5 describes some of the state-of-the-art applications of blockchain-supported Access Control Systems. Finally, section 2.3.6 introduces the identity and access management systems based on blockchain.

2.3.1. BLOCKCHAIN ARCHITECTURE

Blockchain includes a set of components that are part of the blockchain network. They are briefly described below while being related to the reference architecture shown in Figure 2:

1. Peer network and nodes: The node or peer itself runs a client (e.g., Ethereum geth) that is hosted (e.g., on a Docker container) and therefore is placed in the lower layer of the reference architecture. In addition, the P2P network is formed by peers responsible for validating transactions, organizing them into blocks and broadcasting them to the blockchain network, which uses protocols (e.g., a Gossip Protocol) for distributing and propagating information across the network layer.

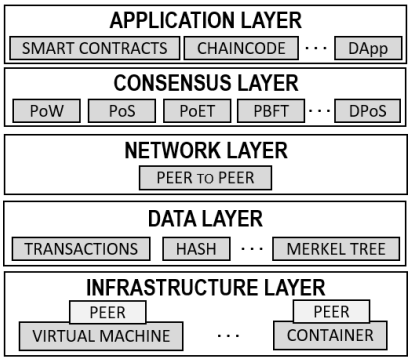

Figure 2: Blockchain reference architecture.

2. Ledger: Blockchain is a decentralized, massively replicated database, where transactions are arranged in a block, which ensures the immutability of transaction history, right from the genesis block to the current block. In this regard, the data structure of a blockchain can be represented by two primary components: pointers and a linked list of blocks where transactions are ordered. The pointers are the variables, which refer to the location of another variable and the linked list is a list of chained blocks, where each block has data and pointers to the previous block

3. Consensus: The consensus algorithm is the core for the existence of blockchain platforms, being the most critical and crucial layer for any blockchain. Consensus is responsible for validating the blocks, ordering the blocks and ensuring that everyone agrees on them.

4. Smart contracts or chaincode constitute the logic executed on a blockchain network. Participating nodes or blockchain clients can issue transactions against that business logic. Thus, the ledger will store not only the immutable transactions but also the immutable code.

2.3.2. BLOCKCHAIN TYPES

According to the reference [30], blockchain systems can be characterized by two dimensions: access and validation. Access refers to permissions for the execution of transactions. For instance, public access implies that anyone can perform transactions. Validation refers to participation in the consensus mechanism and validation of transactions. For instance, permissioned only refers to who can operate the network and run the validator nodes. This context enables four scenarios for blockchain types divided into: public and permissionless,

public and permissioned, private and permissionless and private and permissioned. Table 3 illustrates blockchain examples by category.

Table 3: Blockchain types.

		Validation	
		Permissionless	Permissioned
Access	Public	Bitcoin Ethereum Veres One IoTA	Sovrin Hyperledger Indy Alastria LaCChain EBSI
	Private		Corda Hyperledger Fabric Hyperledger Besu

1. A public-permissionless blockchain is fully open and transparent, offering disintermediation, anonymity and immutability. However, they are trustless since anybody can participate on the blockchain network, i.e., a user can deploy a blockchain client, e.g., Ethereum geth and then join the blockchain network. Thus, the trust is placed in the consensus, which is responsible for validation and synchronization of the transactions to be added to the blockchain once they are replicated to each participating node, allowing trustless parties to execute transactions with confidence. Although transactions can be read by anyone, the identities of users are protected, ensuring anonymity.

2. A public-permissioned blockchain is cost-effective, transparent and offers disintermediation and anonymity. Public-permissioned blockchain allows anyone to read transactions, but only a few permissioned users can write transactions. Alternatively, it can allow a few to read transactions and everyone to write transactions. These types of blockchains are intended for use cases where individuals or authorities can authorize a transaction with data that is visible to the public, currently, these types of blockchains support privacy layers. Considering that not everyone participates in the validation, i.e., validators are elected, transaction validation is neither slow nor expensive compared to a public-permissionless blockchain.

3. A private-permissionless blockchain is rarely implemented as it implies that only individual or selected members can run an entire node to execute, validate and read transactions. A few can write transactions and validate transactions, while all can read. It can be applied for use cases such as audits and is mostly adopted by companies that want to explore blockchain within the enterprise.

4. A private-permissioned blockchain (a.k.a. private consortiums) addresses the enterprise requirements that the public blockchain cannot address. In this regard, since the role of nodes in the DLT can be elected and the number of nodes is smaller than in the public blockchain, the throughput and the performance are generally greater than in public-permissionless blockchains. Additionally, this type of network ensures data privacy among network participants, even if they all are in the same blockchain network. Moreover, the consensus is controlled by a predefined set of nodes, leading to a faster, low-cost network.

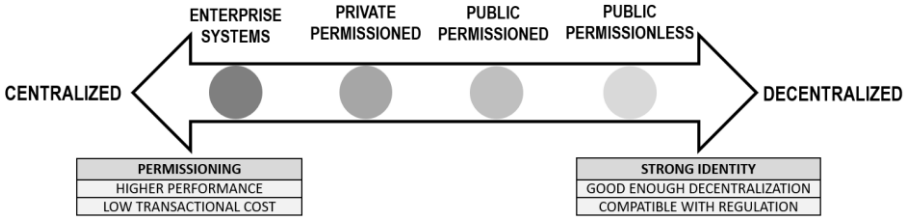

Figure 3: Permissioning and identities approach.

By the way of summary, Figure 3 shows a current trend to position blockchain types, on the one hand, network governance and on the other hand identity decentralization. Moving closer to the left implies increasingly centralizing the solution, obtaining improvements in performance levels and transaction costs. Moving to the right implies further decentralization not only of the network but also of the identity itself [31].

2.3.3. SMART CONTRACTS

A smart contract is a set of code and data deployed by cryptographically signed transactions on the blockchain network (e.g., Hyperledger Fabric's chaincode and Ethereum's smart contracts). Smart Contracts are performed by blockchain network nodes; all nodes executing the smart contract must obtain the same execution results, i.e., must be deterministic, which are recorded on the blockchain. Being deterministic implies that the same input will always produce the same output and therefore, the nodes that have executed the code shall agree with the new state obtained after the execution. The code, since it is on the blockchain, is tamper-resistant and can therefore be used, for instance, as a trusted third party [32].

In many blockchain implementations, nodes execute the smart contract code in parallel when publishing new blocks. There are other blockchain implementations where publishing nodes do not execute the smart contract code but validate the results of nodes that do. In permissionless smart contract-enabled networks (e.g., Ethereum) the issuer (e.g., a user) will have to pay the cost of executing the code. In the case of permissioned smart contract

networks (e.g., Hyperledger Fabric), there is no need for users to pay for the execution of the smart contract code. These scenarios are determinant in the fact of how much information will be handled within or without the blockchain, which can be referred to as on-chain and off-chain respectively, constituting the topic of the next section.

2.3.4. OFF-CHAIN VS. ON-CHAIN

Blockchain applications are categorized into two types: on-chain blockchain ([33]) and off-chain blockchain ([34]). In turn, both types can be a combination of computation and storage, i.e. on-chain computation, on-chain storage, off-chain computation and off-chain storage. On-chain storage uses the blockchain as storage for transaction results such as the outcome of a smart contract. On-chain computation focus on a smart contract that operates in an isolated environment of the blockchain network, determining the rules for both the creation and the composition of transaction attributes [35]. In general, on-chain computation and storage are typically associated with a consortium or public-permissioned blockchains and allows to implement mechanisms such as on-chain signature verification. In contrast, off-chain storage exploits the blockchain as storage for fingerprints and off-chain computation aims to handle as many interactions as possible outside the blockchain [36]. In general, off-chain computation and storage are used in a public-permissionless environment to reduce transaction and storage costs and can use resources such as the collection of events emitted from the blockchain as a mechanism that ensures a permanent index on the blockchain.

The blockchain properties mentioned throughout section 2.3 make the technology suitable for the Access Control Systems integration. The following section discusses the state-of-the-art Access Control Systems - Blockchain integration in IIoT environments.

2.3.5. ACCESS CONTROL SYSTEM ON BLOCKCHAIN TO PROTECT IIOT ENVIRONMENTS

In section 2.3, blockchain technology was introduced and, although authorized voices such as "industrial internet consortium" indicates that blockchain use cases in IIoT environments are in their infancy, a set of use cases are available for different IIoT sectors such as the Pharmaceutical supply chain, Alibaba food traceability (transportation), Amazon managed blockchain (cloud computing), Dun & Bradstreet Business Identity (Finance & Banking), Everledger Diamonds (Mining & Metals) and so on [37]. In addition, the same organization summarizes typical IIoT use cases where blockchain is decisive, including device monitoring, device analytics and edge autonomy [38]. In this sense, blockchain is a suitable technology to provide access control, since blockchain-based access control offers security and transparency, avoiding the involvement of third parties. Access control systems based on

blockchain, while providing data confidentiality, also ensure data integrity. In addition, the fact to be a tamper-proof technology enables it as a chain of custody, increasing the security of the data stored in the devices. In this section, using as a basis the approach to Access Control Systems summarized in section 2.2.3 some examples of blockchain-based Access Control Systems for securing IIoT environments are described. Table 4 summarizes these scenarios, considering the IIoT ecosystem levels defined.

Table 4: Access control systems on blockchain to protect IIoT environments.

IIoT Ecosystem Levels	Access Control System based on blockchain			
	Access Control Model			IAAA
	ABAC	RBAC	OrBAC	
Endpoint	[39]	[40]	[40]	[41], [42]
Data				[43], [42]
Communication & Connectivity				[42]

Similar to section 2.2.3, there is a set of proposals generated for IoT environments, but with applicability in an IIoT context. In this regard, authors of reference [39] propose an ABAC scheme for IoT, where each endpoint can be described by a set of attributes, e.g., its identity. The blockchain is used as a verifiable data registry, to record the distribution of attributes. Each endpoint, i.e., a device executes the access control policy off-chain, which involves, as authors indicate, simple signature and hash operations. BorderChain constitutes the second approach of an access control framework based on blockchain to protect IoT endpoints [41]. Unlike the previous solution, BorderChain uses Ethereum blockchain not only as a verifiable data registry but also as on-chain enforcement of authentication and authorization (AA). The authors also introduce the IoT Domain concept like a zone where the owners authenticate gateways and devices as endpoints, previous to open the endpoints to other parties. Similar to BorderChain, BlendMAS is a solution that uses blockchain beyond a verifiable data registry. The smart contract enables both data sharing as well as authentication and access control [43]. In this regard, BlendMAS constitutes a microservice architecture to enable Smart Public Safety. Unlike both BorderChain and BlendMAS projects, the authors of the reference [40] are fully focused on the dynamic and distributed access control policy where each device has to be capable of handling authorization, i.e., off-chain level. As an important contribution, the authors emphasize the dynamism of the access control model, referring to both RBAC and OrBAC, which is defined as a rich access control model in terms of components and applicability to real environments [44]. Finally, Fabric-iot constitutes an Access Control System solution that provides security, thanks to three types of smart contracts, not only to communication level but also to endpoint and data level [42]. Fabric-iot focuses on the ABAC model as an authorization mechanism.

In contrast to the works analyzed in section 2.2.3, most of these works do complete a life cycle of design, implementation and evaluation, regard Access Control Systems on blockchain environments, showing an increasing interest in this type of research.

2.3.6. IDENTITY AND ACCESS MANAGEMENT SYSTEM BASED ON BLOCKCHAIN

The perspective of Access Control Systems integrated with blockchain technology is framed as part of access management within the Identity and Access Management Framework [45]. However, the contributions of this thesis also include the identity management perspective, so this section introduces blockchain into the Identity and Access Management Framework.

Stakeholders define blockchain as an enabling technology for both identity management (IM) and access management (AM). Thus, in the identity management field, Blockchain-based Identity Management Systems (BIMS) are considered a rising technology, as they differ from Traditional Identity Management Systems (TIMS). BIMS systems enable the custody of identity-related information of users or devices, while TIMS records information (e.g., access keys) of these [46]. In this regard, the City of Zug in Switzerland has adopted a BIMS based on uPort [47]. Finally, the provinces of British Columbia and Ontario in Canada use a BIMS based on the Sovrin blockchain that runs, a Hyperledger Indy instance [48]. In the field of access management, blockchain technology has been included as part of several scenarios, mentioned throughout this chapter, involving all phases of Access Control Systems: identification, authentication, authorization, auditing and accountability [49]. At this point, the Self-Sovereign Identity paradigm, which refers to the ability of a user or device to control and manage its identity and associated data, authorizing access to the information strictly necessary for the verification of such identity and which uses blockchain technology as its core, emerges as a key technology for both contexts: identity and access management [50]. In this sense, approaches such as the one presented by Sovrin in [51] comply with identity and access management, indicating how SSI addresses the main challenges of IoT, such as identification, authentication, authorization and auditing, while ensuring data privacy and integrity through a secure channel. Despite these advances, there are numerous challenges faced by SSI in the context of IIoT, including authentication and authorization in machine-to-machine (M2M) environments [52], being one of the problems to be studied by this thesis.

2.4. THESIS MOTIVATION

Throughout the background section, we have presented the enabling technologies in the development of this thesis. However, in this section, we pretend to emphasize the reasons that have motivated the development of this thesis. As already mentioned, IIoT environments

are specialized in improving the productivity and efficiency of industrial processes and are particularly related to Industry 4.0. For these environments, security constitutes one of the main challenges so that numerous issues have been identified that bring associated risks, among which are the security of industrial control systems, the connectivity between field devices and gateways, the increasing integration of IT monitoring of physical production processes. However, the major security risk is represented by the fact that IIoT has blurred the traditional boundaries of IT and OT infrastructures, allowing the convergence of both environments. In that sense, while IIoT convergence is central to the capabilities that IIoT builds, breaking down the boundaries between these two worlds exposes significant aspects such as unprotected network connections, deployment of technologies with identified vulnerabilities that bring together formerly unidentified risks into the OT environment and inadequate perception of requirements for ICS settings. Therefore, these scenarios can result in brand and reputational damage, material financial loss and potential damage to critical infrastructures. Considering that confidentiality, integrity and availability are parameters that emerge as mechanisms with applicability in the context of categorization of both risks and protection mechanisms, the first motivation of this thesis is the need for a thorough analysis of the vulnerabilities and their impact on both IoT and IIoT.

It has also been mentioned that access control emerges as enabling technology in the security management of IIoT environments, both at the level of communication and connectivity protection and at the level of data and endpoint protection [9]. Hence, the second motivation of the thesis is the exploration of the Access Control Systems for IIoT scenarios due to the current heterogeneity and variety of this environment. For this purpose, Access Control Systems are studied with an approach related to the execution of authorization policies based on traditional models such as RBAC or ABAC, as well as Access Control Systems that involve not only authorization capabilities but also identification, authentication, auditing and accountability. Additionally, it is also considered of interest the research not only from the access management perspective but also from the identity management perspective. For the scenarios analyzed and proposals realized, the discussion in terms of the degree of decentralization, information storage and performance will be also relevant.

3. THESIS OBJECTIVES

Security is one of the main challenges of Industrial Internet of Things environments. The growth of interconnected assets and devices has raised the incidence of vulnerabilities in both design and configuration as well as implementation. In this regard, access control emerges as a solution to mitigate some vulnerabilities and threats. However, the effectiveness

of these solutions requires a comprehensive analysis that includes, based on experimental results, benefits and limitations of their design, implementation and verification in IIoT environments. With the goal of address these scenarios, the objectives of this doctoral thesis are presented below based on two major general objectives and a set of specific objectives associated with them:

O_1. Review IIoT technologies (protocols, standards and buses) and provide a mechanism to evaluate the impact of their vulnerabilities in IT and OT environments. This objective can be subdivided into the following particular objectives:

$O_{1.A.}$ To provide a mechanism to study the set of vulnerabilities documented in assets that use the protocols, standards and buses studied.

$O_{1.B.}$ To characterize both the severity and the impact of studied vulnerabilities, enabling a comparison of the behavior in both IT and OT environments.

$O_{1.C.}$ To analyze the most used attack pattern and the more exploited weaknesses.

O_2. Design, implement and evaluate the feasibility of a variety of Access Control Systems applicable to different IIoT environments, based on different architectures. This objective can be subdivided into the following particular objectives:

$O_{2.A.}$ To design, implement and evaluate the feasibility of an ACS based on a centralized architecture as a single entity responsible for endpoint authorization management in a performance-constrained industrial environment.

$O_{2.B.}$ To design, implement and evaluate the feasibility of an ACS decentralized architecture based on Ethereum's public blockchain technology in a non-performance-constrained environment, paying attention to the transaction cost of the architecture.

$O_{2.C.}$ To design, implement and evaluate the feasibility of a decentralized architecture based on permissioned blockchain enabling the blockchain as more than a verifiable data registry that performs an Access Control Systems in a performance-constrained industrial environment.

$O_{2.D.}$ To design, implement and evaluate the feasibility of a decentralized architecture based on permissioned blockchain enabling the blockchain as more than a verifiable data registry that performs an on-chain identity and access management based on self-sovereign identity in performance-constrained industrial environments.

The results of this thesis include six main contributions and the other five additional contributions (see details in **APPENDIX A**). This thesis book includes the six main

contributions (from C_1 to C_6) that have targeted the aforementioned objectives. Table 5 details the relationship between general and specific objectives and contributions.

Table 5: Objectives and main contributions.

General Objective	Specific Objective	Contributions					
		C_1	C_2	C_3	C_4	C_5	C_6
O_1	$O_{1.A}$	X					
	$O_{1.B}$	X					
	$O_{1.C}$	X					
O_2	$O_{2.A}$		X				
	$O_{2.B}$			X	X		
	$O_{2.C}$					X	
	$O_{2.D}$						X

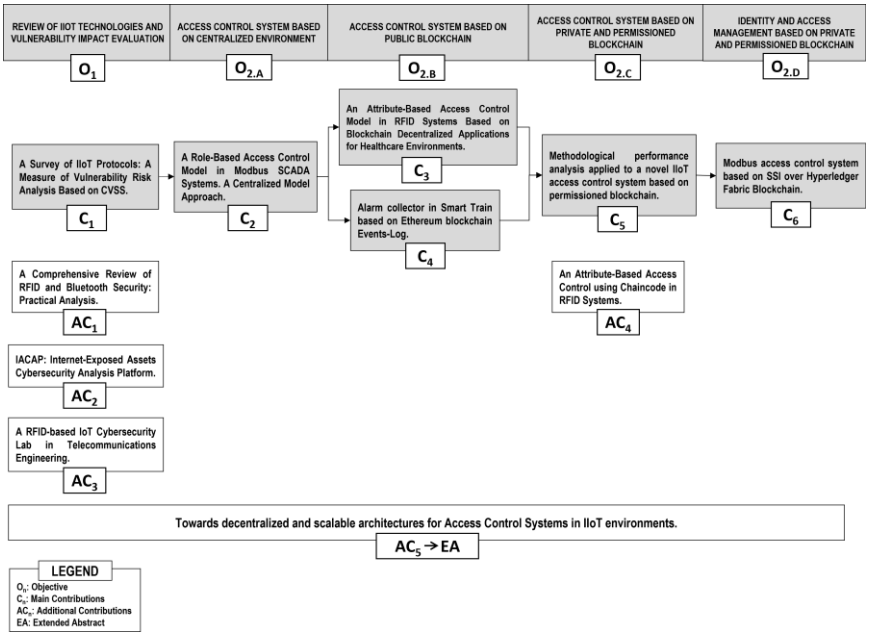

Figure 4*:* Ph.D. Thesis roadmap.

Figure 4 describes visually the relationship between the objectives and all contributions including also the additional contributions (AC_n) listed in APPENDIX A.

4. THESIS OUTLINE

This Ph.D. Thesis is organized by articles as follows:

CHAPTER II: This chapter includes the C_1 journal contribution, which reviews IIoT technologies (protocols, standards and buses) and provides a mechanism to evaluate the

impact of their vulnerabilities in IT and OT environments. Particularly, it provides a mechanism to study the set of vulnerabilities documented in assets that use the protocols, standards and buses studied. This contribution addresses the three objectives of O_1. **C_1 journal contribution** can be found in:

S. Figueroa-Lorenzo, J. Añorga and S. Arrizabalaga, "A Survey of IIoT Protocols: A Measure of Vulnerability Risk Analysis Based on CVSS," ACM Computer Surveys, vol. 53, no. 2, p. 53, 2020, doi: doi.org/10.1145/3381038. [**JCR. 7.990**, **Q1**].

CHAPTER III: This chapter includes the **C_2 journal contribution** related to a centralized architecture as a single entity responsible for endpoint authorization management in performance-constrained industrial environments. Particularly, it includes the design, implementation and evaluation of a centralized Access Control System for a constrained industrial manufacturing application. This contribution addresses the objective $O_{2.A}$. **C_2 journal contribution** can be found in:

S. Figueroa-Lorenzo, J. Añorga and S. Arrizabalaga, "A Role-Based Access Control Model in Modbus SCADA Systems. A Centralized Model Approach," Sensors, MDPI, vol. 19, no. 20, p. 4455, 2019, doi: doi.org/10.3390/s19204455. [**JCR. 3.576**, **Q1**].

CHAPTER IV: This chapter includes the **C_3 journal contribution** related to the use of Ethereum's public blockchain as a verifiable data registry in non-performance-constrained environments. Particularly, it includes the design, implementation and evaluation of a decentralized access control system based on the Ethereum blockchain for a non-constrained application in the healthcare sector. Although this contribution fully addresses $O_{2.B}$, the high cost of transactions requires further research for the exploration of cost-effective solutions. **C_3 journal contribution** can be found in:

S. Figueroa-Lorenzo, J. Añorga and S. Arrizabalaga, "An Attribute-Based Access Control Model in RFID Systems Based on Blockchain Decentralized Applications for Healthcare Environments," Computers, MDPI, vol. 8, no. 3, p. 19, 2019, doi: doi.org/10.3390/computers8030057. [**SJR. 3.3, Q2**].

CHAPTER V: This chapter includes the **C_4 journal contribution,** also related to the use of Ethereum's public blockchain as a verifiable data registry in non-performance-constrained environments. However, in this case, it includes the design and implementation of an alarm collection system based on the emission of ETH event logs in the transport sector. This contribution complements **C_3** contribution to addressing $O_{2.B}$ objective in terms of providing a more cost-effective solution. **C_4 journal contribution** can be found in:

S. Figueroa-Lorenzo, Santiago; Goya, Jon; Añorga, Javier; Adin, Iñigo; Mendizabal, Jaizki; Arrizabalaga, "Alarm collector in Smart Train based on Ethereum blockchain events-log," IEEE Internet of Things Journal, vol. 08, no. 17, pp. 13306 – 13315, 2021, doi: doi.org/10.1109/JIOT.2021.3065631. [JCR. 9.471, Q1].

CHAPTER VI: This chapter includes the **C_5 journal contribution** related to the use of a decentralized architecture based on permissioned blockchain enabling the blockchain as more than a verifiable data registry that performs an Access Control Systems in a performance-constrained industrial environment. Particularly, it includes the design, implementation and evaluation of a decentralized Access Control System based on consortium blockchain for an engine assembly line. This contribution addresses $O_{2.C}$. **C_5 journal contribution** can be found in:

S. Figueroa-Lorenzo, J. Añorga and S. Arrizabalaga, "Methodological performance analysis applied to a novel IIoT access control system based on permissioned blockchain," Information Processing & Management., vol. 58, no. 4, p. 102558, 2021, doi: doi.org/10.1016/j.ipm.2021.102558. [JCR. 6.222, Q1].

CHAPTER VII: This chapter includes the **C_6 journal contribution** related to the use of a decentralized architecture based on permissioned blockchain enabling the blockchain as more than a verifiable data registry that performs an on-chain identity and access management system based on self-sovereign identity in performance-constrained industrial environments. Particularly, it includes the design, implementation and evaluation of decentralized identity and access management systems based on consortium blockchain for a constrained-performance Modbus environment. This contribution addresses $O_{2.D}$. **C_6 journal contribution** can be found in:

S. Figueroa-Lorenzo, J. Añorga and S. Arrizabalaga, "Modbus access control system based on SSI over Hyperledger Fabric Blockchain," Sensors, MDPI, vol. 21, no. 16, p. 5438, 2021, doi: doi.org/10.3390/s21165438. [JCR. 3.576, Q1].

CHAPTER VIII: This chapter discusses the main contributions of this Ph.D. Thesis, by starting with an overview of all the contributions together with a more specific discussion for each of the contributions. Finally, some future research lines are identified.

5. FRAMEWORK

This Ph.D. Thesis has been carried out at Data Analysis and Information Management (DAIM) Group that belongs to the Information and Communication Technology Division of

CEIT and the funding for this thesis has come from different research & development projects:

1. SEKUTEK: "SEKUrtasun TEKnologiak" with project number KK-2017/00044, supported by the Elkartek program of the Basque Government (2017 – 2018).
2. CORMORAN: "Conversión de energía y monitorización de la red de alimentación del sistema ferroviario" with project number TSI-100505-2016-8 supported by the program "Acción Estratégica de Economía y Sociedad Digital (AEESD)" of the "Ministerio de Industria, Energía y Turismo", Spain Government (2016 – 2018).
3. CYBERPREST: "CyberPrest - Cybersegurtasunerako gaitasun osoa" with project number KK-2018/00058, supported by the Elkartek program of the Basque Government (2018 – 2019).
4. CONTRACK: "Container Tracking: Trazabilidad Avanzada del Envase" with project number KK-2018/000057 supported by the Hazitek program of the Basque Government (2018 – 2020).
5. SENDAI: "SEgurtasun integrala iNDustria AdImentsurako" with project number KK-2019/00072 supported by the Elkartek program of the Basque Government (2019 – 2020).
6. TRUSTIND: "Creating Trust in the Industrial Digital Transformation" with project number KK-2020/00054 supported by the Elkartek program of the Basque Government (2020 – 2021).

6. REFERENCES

[1] European Union Agency for Cybersecurity (ENISA), Baseline Security Recommendations for IoT in the context of Critical Information Infrastructures, no. November. 2017.

[2] J. Navarro-Ortiz, S. Sendra, P. Ameigeiras, and J. M. Lopez-Soler, "Integration of LoRaWAN and 4G/5G for the Industrial Internet of Things," IEEE Commun. Mag., vol. 56, no. 2, pp. 60–67, Feb. 2018, doi: doi.org/10.1109/MCOM.2018.1700625.

[3] V. Sklyar and V. Kharchenko, "ENISA Documents in Cybersecurity Assurance for Industry 4.0: IIoT Threats and Attacks Scenarios," in 2019 10th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS), 2019, vol. 2, pp. 1046–1049, doi: doi.org/10.1109/IDAACS.2019.8924452.

[4] X. Yu and H. Guo, "A Survey on IIoT Security," in 2019 IEEE VTS Asia Pacific Wireless Comm. Symposium (APWCS), 2019, pp. 1–5, doi: doi.org/10.1109/VTS-APWCS.2019.8851679.

[5] F. Meneghello, M. Calore, D. Zucchetto, M. Polese, and A. Zanella, "IoT: Internet of Threats? A Survey of Practical Security Vulnerabilities in Real IoT Devices," IEEE Internet Things J., vol. 6, no. 5, pp. 8182–8201, Oct. 2019, doi: doi.org/10.1109/JIOT.2019.2935189.

[6] M. Chapple, J. M. Stewart, and D. Gibson, (ISC)2 CISSP Certified Information Systems Security Professional Official Study Guide. 2018.

[7] S. Mantravadi, R. Schnyder, C. Møller, and T. D. Brunoe, "Securing IT/OT Links for Low Power IIoT Devices: Design Considerations for Industry 4.0," IEEE Access, vol. 8, pp. 200305–200321, 2020, doi: doi.org/10.1109/ACCESS.2020.3035963.

[8] ENISA, Good Practices for Security of Internet of Things in the context of Smart Manufacturing, no. November. 2018.

[9] S. Schrecker et al., "Industrial Internet of Things Volume G4 : Security Framework," Ind. Internet Consort., pp. 1–173, 2016, doi: doi.org/10.13140/RG.2.2.28143.23201.

[10] S. Salonikias, A. Gouglidis, I. Mavridis, and D. Gritzalis, "Access Control in the Industrial Internet of Things," in Security and Privacy Trends in the Industrial Internet of Things, C. Alcaraz, Ed. Cham: Springer International Publishing, 2019, pp. 95–114.

[11] I. Alsmadi and I. Alsmadi, Cyber Intelligence Analysis. 2019.

[12] K. Riad and Z. Yan, "EAR-ABAC: An Extended AR-ABAC Access Control Model for SDN-Integrated Cloud Computing," Int. J. Comput. Appl., vol. 132, no. 14, pp. 9–17, 2015, doi: doi.org/10.5120/ijca2015907649.

[13] S. Bhatt, F. Patwa, and R. Sandhu, "An Attribute-Based Access Control Extension for OpenStack and Its Enforcement Utilizing the Policy Machine," in 2016 IEEE 2nd International Conference on Collaboration and Internet Computing (CIC), 2016, pp. 37–45, doi: doi.org/10.1109/CIC.2016.019.

[14] D. Rountree, "Chapter 2 - What Is Federated Identity?," in Federated Identity Primer, D. Rountree, Ed. Boston: Syngress, 2013, pp. 13–36.

[15] Amith Raj MP, A. Kumar, S. J. Pai, and A. Gopal, "Enhancing security of Docker using Linux hardening techniques," in 2016 2nd International Conference on Applied and Theoretical Computing and Communication Technology (iCATccT), 2016, pp. 94–99, doi: doi.org/10.1109/ICATCCT.2016.7911971.

[16] A. Ouaddah, H. Mousannif, A. Abou Elkalam, and A. Ait Ouahman, "Access control in the Internet of Things: Big challenges and new opportunities," Comput. Networks, vol. 112, pp. 237–262, 2017, doi: doi.org/10.1016/j.comnet.2016.11.007.

[17] K. Yang, D. Forte, and M. M. Tehranipoor, "Protecting endpoint devices in IoT supply chain," 2015 IEEE/ACM Int. Conf. Comput. Des. ICCAD 2015, pp. 351–356, 2016, doi: doi.org/10.1109/ICCAD.2015.7372591.

[18] M. W. Condry and C. B. Nelson, "Using Smart Edge IoT Devices for Safer, Rapid Response with Industry IoT Control Operations," Proc. IEEE, vol. 104, no. 5, pp. 938–946, 2016, doi: doi.org/10.1109/JPROC.2015.2513672.

[19] M. Hemdi and R. Deters, "Using REST based protocol to enable ABAC within IoT systems," in 2016 IEEE 7th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON), 2016, pp. 1–7, doi: doi.org/10.1109/IEMCON.2016.7746297.

[20] L. Duan, C. Sun, Y. Zhang, W. Ni, and J. Chen, "A Comprehensive Security Framework for Publish/Subscribe-Based IoT Services Communication," IEEE Access, vol. 7, pp. 25989–26001, 2019, doi: doi.org/10.1109/ACCESS.2019.2899076.

[21] S. Qi, Y. Lu, W. Wei, and X. Chen, "Efficient Data Access Control with Fine-Grained Data Protection in Cloud-Assisted IIoT," IEEE Internet Things J., vol. 8, no. 4, pp. 2886–2899, 2021, doi: doi.org/10.1109/JIOT.2020.3020979.

[22] N. Kaliya and M. Hussain, "Framework for privacy preservation in iot through classification and access control mechanisms," in 2017 2nd International Conference for Convergence in Technology (I2CT), 2017, pp. 430–434, doi: doi.org/10.1109/I2CT.2017.8226166.

[23] M. Wazid, M. S. Obaidat, A. K. Das, and P. Vijayakumar, "SAC-FIIoT: Secure Access Control Scheme for Fog-Based Industrial Internet of Things," in GLOBECOM 2020 - 2020 IEEE Global Comm. Conference, 2020, pp. 1–6, doi: doi.org/10.1109/GLOBECOM42002.2020.9322212.

[24] A. Saxena, P. Duraisamy, and V. Kaulgud, "SMAC: Scalable Access Control in IoT," Proc. - 2015 IEEE Int. Conf. Cloud Comput. Emerg. Mark. CCEM 2015, pp. 169–176, 2016, doi: doi.org/10.1109/CCEM.2015.22.

[25] M. V Tripunitara and N. Li, "A Theory for Comparing the Expressive Power of Access Control Models," J. Comput. Security., vol. 15, no. 2, pp. 231–272, 2007.

[26] J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext-Policy Attribute-Based Encryption," in 2007 IEEE Symposium on Security and Privacy (SP '07), 2007, pp. 321–334, doi: doi.org/10.1109/SP.2007.11.

[27] S. Shafeeq, M. Alam, and A. Khan, "Privacy aware decentralized access control system," Futur. Gener. Comput. Syst., vol. 101, pp. 420–433, 2019, doi: doi.org/10.1016/j.future.2019.06.025.

[28] S. Nakamoto, "Bitcoin: A Peer-to-Peer Electronic Cash System," Www.Bitcoin.Org, p. 9, 2008, doi: doi.org/10.1007/s10838-008-9062-0.

[29] V. Acharya, A. Eswararao Yerrapati, and N. Prakash, Oracle Blockchain Quick Start Guide. Packt Publishing, 2019.

[30] S. Gerth and L. Heim, "Trust through Digital Technologies: Blockchain in Online Consultancy Services," in Proceedings of 2020 The 2nd International Conference on Blockchain Technology, 2020, pp. 150–154, doi: doi.org/10.1145/3390566.3391662.

[31] J. Ruiz, C. R. Technical, and R. Alastria, "To cite this version : Public-Permissioned blockchains as Common-Pool Resources 1 . Introduction to Public-Permissioned," 2020.

[32] D. Yaga, P. Mell, N. Roby, and K. Scarfone, "Blockchain Technology Overview," arXiv, 2019, doi: doi.org/10.6028/NIST.IR.8202.

[33] E. Ben-Sasson et al., "Zerocash: Decentralized anonymous payments from bitcoin," Proc. - IEEE Symp. Security and Privacy, pp. 459–474, 2014, doi: doi.org/10.1109/SP.2014.36.

[34] G. Malavolta, P. Moreno-Sanchez, A. Kate, M. Maffei, and S. Ravi, "Concurrency and privacy with payment-channel networks," arXiv, 2019.

[35] В. Носов, "Разработка и тестирование смарт-контрактов Hyperledger Fabric," November 27, 2018, at 11:27 AM, 2018. [Online]. Available: https://habr.com/en/post/426705/. [Accessed: 04-Apr-2021].

[36] J. Eberhardt and J. Heiss, "Off-Chaining Models and Approaches to Off-Chain Computations," in Proceedings of the 2nd Workshop on Scalable and Resilient Infrastructures for Distributed Ledgers, 2018, pp. 7–12, doi: doi.org/10.1145/3284764.3284766.

[37] The consortium, "Distributed Ledgers in IIoT," 2020.

[38] Industrial Internet Consortium, "Implementation Aspect: IIoT and Blockchain," pp. 1–7, 2020.

[39] S. Ding, J. Cao, C. Li, K. Fan, and H. Li, "A Novel Attribute-Based Access Control Scheme Using Blockchain for IoT," IEEE Access, vol. 7, pp. 38431–38441, 2019, doi: doi.org/10.1109/ACCESS.2019.2905846.

[40] A. OUTCHAKOUCHT, H. ES-SAMAALI, and J. Philippe, "Dynamic Access Control Policy based on Blockchain and Machine Learning for the Internet of Things," Int. J. Adv. Comput. Sci. Appl., vol. 8, no. 7, pp. 417–424, 2017, doi: doi.org/10.14569/ijacsa.2017.080757.

[41] Y. E. Oktian and S.-G. Lee, "BorderChain: Blockchain-Based Access Control Framework for the Internet of Things Endpoint," IEEE Access, p. 1, 2020, doi: doi.org/10.1109/ACCESS.2020.3047413.

[42] H. Liu, D. Han, and D. Li, "Fabric-iot: A Blockchain-Based Access Control System in IoT," IEEE Access, vol. 8, pp. 18207–18218, 2020, doi: doi.org/10.1109/ACCESS.2020.2968492.

[43] R. Xu, S. Y. Nikouei, Y. Chen, E. Blasch, and A. Aved, "BlendMAS: A blockchain-enabled decentralized microservices architecture for smart public safety," Proc. - 2019 2nd IEEE Int. Conf. Blockchain, Blockchain 2019, pp. 564–571, 2019, doi: doi.org/10.1109/Blockchain.2019.00082.

[44] A. A. E. Kalam et al., "Organization based access control," in Proceedings POLICY 2003. IEEE 4th International Workshop on Policies for Distributed Systems and Networks, 2003, pp. 120–131, doi: doi.org/10.1109/POLICY.2003.1206966.

[45] A. Caballero, "Chapter 24 - Information Security Essentials for Information Technology Managers: Protecting Mission-Critical Systems," in Computer and Information Security Handbook (Third Edition), Third Edit., J. R. Vacca, Ed. Boston: Morgan Kaufmann, 2017, pp. 393–419.

[46] L. Lesavre, P. Varin, P. Mell, M. Davidson, and J. Shook, "A taxonomic approach to understanding emerging blockchain identity management systems," arXiv, 2019, doi: doi.org/10.6028/NIST.CSWP.07092019-draft.

[47] A. Offerman, "Swiss City of Zug issues Ethereum blockchain-based eIDs," 27/02/2018, 2018. [Online]. Available: https://joinup.ec.europa.eu/collection/egovernment/document/swiss-city-zug-issues-ethereum-blockchain-based-eids. [Accessed: 17-Jan-2021].

[48] VON, "Verifiable Organizations Network," 2019. [Online]. Available: https://vonx.io/. [Accessed: 17-Jan-2021].

[49] I. Butun and P. Österberg, "A Review of Distributed Access Control for Blockchain Systems Towards Securing the Internet of Things," IEEE Access, vol. 9, pp. 5428–5441, 2021, doi: doi.org/10.1109/ACCESS.2020.3047902.

[50] A.-E. Panait, R. F. Olimid, and A. Stefanescu, "Identity Management on Blockchain - Privacy and Security Aspects," CoRR, vol. abs/2004.13107, 2020.

[51] D. Reed, "Introducing the Trust over IP Foundation," 2020. [Online]. Available: https://trustoverip.org/wp-content/uploads/sites/98/2020/05/toip_introduction_050520.pdf. [Accessed: 14-Jan-2021].

[52] A. Offerman, "Swiss City of Zug issues Ethereum blockchain-based eIDs," 27/02/2018, 2018. [Online]. Available: https://joinup.ec.europa.eu/collection/egovernment/document/swiss-city-zug-issues-ethereum-blockchain-based-eids. [Accessed: 17-Jan-2021].

CHAPTER II

A SURVEY OF IIOT PROTOCOLS: A MEASURE OF VULNERABILITY RISK ANALYSIS BASED ON CVSS

This chapter was published in *S. Figueroa-Lorenzo, J. Añorga, and S. Arrizabalaga, "A Survey of IIoT Protocols: A Measure of Vulnerability Risk Analysis Based on CVSS," ACM Computer Surveys, vol. 53, no. 2, p. 53, 2020, doi: doi.org/10.1145/3381038. [JCR. 7.990, Q1].*

This chapter introduces a comprehensive survey of the protocols, standards, and buses most used in the IIoT environment, in addition, to enable a comprehensive review of the tools available to characterize CVSS in industrial environments as well as an exhaustive collection of OSINTs, data sources and integration tools that allow carrying out vulnerability studies. Finally, it presents a Vulnerability Analysis Framework (VAF) to analyze vulnerabilities associated with the protocols, buses, and standards. VAF allows determining that both the severity and impact of a vulnerability are higher if the asset is part of an Operational Technology (OT) environment than of an Information Technology (IT) environment.

1. ABSTRACT

Industrial Internet of Things (IIoT) is present in many participants from the energy, health, manufacturing, transport, and public sectors. Many factors catalyze IIoT, such as robotics, artificial intelligence, and intelligent decentralized manufacturing. However, the convergence between IT, OT, and IoT environments involves the integration of heterogeneous technologies through protocols, standards, and buses. However, this integration brings with it security risks. To avoid the security risks, especially when systems in different environments interact, it is important and urgent to create an early consensus among the stakeholders on the IIoT security. The default Common Vulnerability Scoring System (CVSS) offers a mechanism to measure the severity of an asset's vulnerability and therefore a way to characterize the risk. However, CVSS by default has two drawbacks. On the one hand, to carry out a risk analysis, it is necessary to have additional metrics to the one established by CVSSv3.1. On the other hand, this index has been used mostly in IT environments and although there are numerous efforts to develop a model that suits industrial environments, there is no established proposal. Therefore, we first propose a survey of the main 33 protocols, standards, and buses used in an IIoT environment. This survey will focus on the security of each one. The second part of our study consists of the creation of a framework to characterize risk in industrial environments, i.e., to solve both problems of the CVSS index. To this end, we created the Vulnerability Analysis Framework (VAF), which is a methodology that allows the analysis of 1,363 vulnerabilities to establish a measure to describe the risk in IIoT environments.

2. INTRODUCTION

Industrial Internet of Things, also known as IIoT, transforms industrial and business operations by adding smart connectivity to machines, peoples, and processes. IIoT integrates technical areas such as network connectivity (e.g., low energy wireless protocols, edge computing, and cloud computing), low cost to apply machine learning in both sensors and computing, big data produced by sensors, m2m (machine to machine) communications, and the traditional automation technologies, i.e., traditional industrial control systems (ICS). Although the term Industry 4.0 is associated with how the manufacturing sector is transformed by technologies such as big data, data analytics, and IoT, this concept can be also associated with the interconnection of cyber physical systems, which enables us to use the term IIoT and Industry 4.0 indiscriminately. For this reason, several factors are considered as catalysts for both IIoT and Industry 4.0, such as the convergence of information technologies (IT) and operational

technologies (OT), robotics, data, artificial intelligence, smart decentralized manufacturing, auto-optimizing systems, and the digital supply chain information management [1]. Although IIoT architectures have many uses case-specific variations, in Figure 2 we consider a basic IIoT architecture model, which includes four subsystems: a sensor network (e.g., communication over Wi-Fi and BLE), a controller or aggregator, an edge gateway, and the business application (through Cloud IIoT Platform). To contextualize this architecture, in Figure 1, the SCADA network is connected to the cloud via an edge gateway. From a higher range connectivity system based, for example, on LoRa, it is possible that many windmills are controlled by the ICS/Scada system. The data received from the turbines are sent to the data center for cloud analysis. The turbine data flows through an edge device, which could be a gateway, central concentrator, or edge controller. This edge device will have to support many protocols. Therefore, a distinctive feature of IIoT environments is the convergence of protocols, standards, and buses of different technologies, i.e., the integration. Although, this integration is not the only area of an IIoT system, given the convergence of protocols, standards, and buses is a representative feature. For that reason, the process of convergence can be described around the protocols, standards, and buses that form the following three paradigms: OT (e.g., Modbus, EtherNet/IP, and OPC-UA), IT (e.g., WebSocket, HTTP, and XMPP) and IoT (e.g., Wi-Fi, RFID, and Bluetooth). Therefore, this is the first focus of attention of our work. The digital connectivity (promoted by the integration of protocols, standards, and buses) of industrial machinery and industrial equipment with any physical asset with an IT platform is a unique advance that sets a social precedent of business opportunities. This convergence of the physical world and cyber on an industrial scale allows operations to be handled in thousands of ways. An example of a use case is to prevent critical machine failures through both predictive and proactive detection and maintenance. Another example of a use case is to provide a digital tracking capability in supply chain assets, to name a few use cases. Many other uses cases could be mentioned.

However, these benefits bring associated risks, because cyberthreats are a break point in ubiquitous connectivity and are now a critical constraint on the adoption of IIoT technology. These cyber threats exploit the weaknesses of IIoT environments, causing vulnerabilities. According to IETF RFC 4949 [2] a system can have three types of vulnerabilities: vulnerabilities in design or specification; vulnerabilities in implementation; and vulnerabilities in operation and management. The design vulnerabilities are inherent to a protocol specification, present even in perfect implementations (e.g., a specification using weak cryptography has a design vulnerability) [3]. In addition, implementation vulnerabilities are inherent to assets and correspond to how a particular proto col, standard, or bus is

implemented in an asset [3]. The operation vulnerabilities are any event that alter the correct functioning of the asset despite the correct design and implementation of the system.

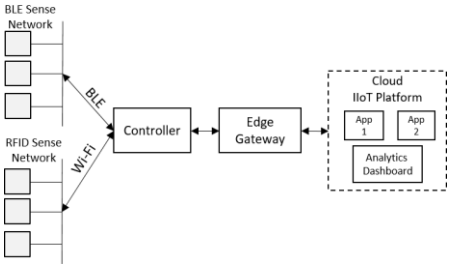

Figure 1: General IIoT Architecture.

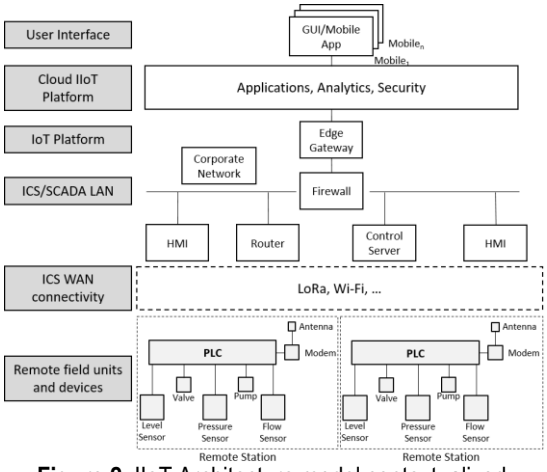

Figure 2: IIoT Architecture model contextualized.

According to NIST 800-82r2 Reference [4], there are a wide range of threats and vulnerabilities in IIoT environments. These can be classified in general way into policy and procedural vulnerabilities (e.g., there is no formal security training for ICS or OT), architecture and design vulnerabilities (e.g., no security perimeter defined), configuration and maintenance vulnerabilities (e.g., poor remote access controls), physical vulnerabilities (e.g., lack of backup power), software development vulnerabilities (e.g., improper data validation) and communication and network configuration (e.g., data flow controls not employed). For this reason, it is important that IIoT environments include risk analysis to identify and detect malfunctions, as well as to prevent incidents and accidents, i.e., be prepared. The risk is the result of uncertainty about objectives, which considers the probability of an event occurring along with the impact, i.e., the effects, of that event if it happens. Precise product and system design, including design reviews and tests, should prevent malfunctions and improve the system's robustness to possible or potential events identified in the risk evaluation [5]. As we have mentioned, the starting point (phase 1) of our work is based on an analysis of the integration of IIoT protocols, standards, and buses from the perspective of the OT, IT and IoT paradigms. Therefore, our second line of research focuses on providing a risk characterization mechanism for an IIoT environment (phase 2) based on the analysis of vulnerabilities that may affect the protocols, standards, and buses that are considered during phase 1 of our work. Summarizing, the major contributions of this work are (1) a comprehensive survey of the 33 protocols, standards, and buses most commonly used in IIoT environment, (2) a comprehensive review of the tools available to characterize CVSS in

industrial environments, (3) an exhaustive collection of OSINT, data sources, and integration tools that allow carrying out vulnerability studies, (4) a presentation and deployment of a Vulnerability Analysis Framework (VAF) to analyze 1,363 vulnerabilities and from which the two major gaps of CVSS are solved, i.e., (1) CVSS alone is not enough to determine risk, i.e., it is not a risk measure; (2) CVSS introduces different complexities when calculating the severity of a vulnerability in industrial environments. The remainder of the manuscript is organized as follows: Section 3 carries out a comprehensive survey of most used thirty-three protocols, standards, and buses in IIoT environments. Section 4 analyses CVSS as evaluation tool for ICS system. Section 5 provides the proposed methodological framework, and the information data sources. Section 6 summarizes the results obtained after VAF application. Section 7 provides conclusions and future research lines.

3. SURVEY OF IIOT PROTOCOLS, STANDARDS, AND BUSES

Until now, we have focused on the overall architecture of an IIoT environment. However, the essential feature of this environment is that it has a high level of integration in terms of communication protocols, standards, and buses. For this reason, first, the study of the 33 protocols, standards, and buses most used in IIoT environments will be introduced. To this end, we will divide our section into three subsections according to the categories established in Table 1: IT, IoT, and OT, and from this point on, the IIoT, protocols associated with each category will be analyzed. Although common criteria are analyzed for each protocol, which will be listed in the following paragraph, the reference is security. Due to the importance of the security of the protocols, standards, and buses in the survey, the need to carry out a risk analysis for these protocols, standards, and buses is detected.

Since we have mentioned at this first stage, we will focus our attention on the IIoT connectivity, which is related by the integration in the same environment (IIoT) of OT, IT, and IoT protocols. For this reason, Table 1 summarizes the protocols, standards, and buses, according to categories to which they belong, i.e., IT, OT, and IoT, as well as the classification according to the industrial area in which they have been applied. To classify the industrial context, the following categories are established: Process Automation, Industrial Control System, Building Automation, Power System Automation, Automatic Meter Reading, and Home Automation. Certain protocols must be excepted as they are on the border between the main categories: IT, IoT, and OT. For instance, XMPP is an instant messaging protocol, i.e., recently having a huge adoption in IoT environments, however, since its application was originally in the IT environment, we have added it in this category.

Table 1: IIoT protocols classification.

Cat.	Protocol/Standard/Bus	FAP	PAP	ICSP	BAP	PSAP	AMRP	SHAP	VAP
IT	REST/HTTP/TLS							[6]	
	WebSocket							[6]	
	JMS		[7]						
	DDS		[8]			[9]			[10]
	XMPP					[11]		[12]	
IoT	Wi-Fi		[13]		[14]			[15]	
	4G/LTE	[16]		[17]	[18]			[19]	
	5G	[16]						[20]	[20]
	MQTT		[21]					[22]	
	CoAP				[23]	[24]		[25]	
	AMQP		[21]						
	Bluetooth		[26]		[27]			[28]	[29]
	ZigBee		[13]		[30]	[31]	[32]	[33]	
	RFID/NFC	[34]						[35]	
	LoRa				[36]		[36]		
	NB-IoT	[37]					[38]	[39]	
	WirelessHart	[39]	[40]						
	Z-Wave				[41]		[42]	[41]	
OT	PROFINET	[43]	[44]						
	Niagara AX				[45]				
	CAN				[46]				[47]
	Siemens S7	[48]	[49]						
	EtherNet/IP	[50]	[51]						
	Hart/IP		[52]						
	BACnet				[53]				
	Modbus		[49]	[54]	[55]	[56]	[57]		
	OPC-DA			[58]					
	OPC-UA			[59]					
	DNP3/IEC-62351	[60]				[56]			
	ANSI C12.22						[61]		
	IEC-61850/IEC-62351					[44]			
	IE-60870-5-104/IEC-62351					[62]			
	IEC-60870-6-503/IEC-62351					[63]			

FAP: Factory Automation Protocol, **PAP**: Process Automation Protocol, **ICSP**: Industrial Control System Protocol, **BAP**: Building Automation Protocol, **PSAP**: Power System Automation Protocol, **AMRP**: Automatic Meter Reading Protocol, **SHAP**: Smart Home Automation Protocol, **VAP**: Vehicular Automation Protocol

Similar cases are found in the Wi-Fi and 4G/LTE standards that although their initial adoption was in the IT and IoT environments, both have a strong presence in industrial environments. Another consideration has been a protocol such as AMQP, designed to support a broad range of both messaging applications as well as communication patterns from banking organizations efficiently; however, its current use is in IoT environments. The last case to

consider is DDS, which by default addresses the needs of applications like aerospace, defense, and air-traffic control. At present, DDS participates in others application context such as autonomous vehicles, smart grid management, power or energy generation, simulation and testing environments, health devices, transport systems, and others real-time applications. Therefore, DDS could perfectly be included in an IoT or OT environment, but given the closeness to the user it has been added within the IT paradigm. Next, the 33 main IIoT protocols, standards, and buses are analyzed in this section. A summary of the protocol, as well as its topology, architecture, types of messages/frames exchanged, and security are used as common criteria in the survey. However, the TCP or UDP port associated with the protocols, standards, and buses is defined by the Internet Assigned Numbers Authority (IANA) recommendations [64].

3.1. IT PROTOCOLS AND STANDARDS

Based on the common criteria established in the introduction to the section, this sub-section analyzes the protocols, standards, and buses associated with the IT category according to Table 1.

3.1.1. REST/HTTP/TLS (TCP: 80)

REST (REpresentational State Transfer) is the dominant model for Web application design, which aims to reduce latency and network communication, maximizing the scalability of independent component deployments. Reference [65] invokes an in-depth study of the use of API REST in IoT environments. The three defined aspects: URI, MIME, and the set of operations relate RESTful (web services based on HTTP and REST fundamentals) with IoT, since the sensors send the information in a MIME format, RESTful sets up a URL for each resource. As Reference [66] mentions, it is simple to retrieve sensor information as a web resource. The security, in HTTP, has typically been delegated to the lower layer via SSL (Secure Sockets Layer) or Transport Layer Security (TLS), and both cases are referred such as HTTPS. These two protocols provide both clients and servers HTTP, security based on asymmetric cryptography for authentication enabling the exchange of keys, and symmetric cryptography for confidentiality. Currently, because SSL is considered unsecure, TLS is mostly used in its latest version 1.3. In addition to SSL/TLS, Reference [67] is a review of mechanisms used by REST to provide security to IoT environments through HTTP authentication schemes. These are divided into basic authentication schemes, token-based authentication, OAuth, and OpenID. However, since all the mentioned applications use SSL/TLS, which guarantees the security of the transport and not of the applications, vulnerabilities and attack patterns must be considered, mainly due to implementation failures. For instance, Reference

[68] demonstrates a proof of concept of JSON injection, Reference [69] reported serious logic flaws in OpenID systems, Reference [70] reported cross-site request forgery for OAuth Clients and injection attacks in Web Services are analyzed in Reference [71]. Given that the Open Web Application security Project (OWASP) introduces in Reference [72] some different kinds of REST attack and the how-to prevents, this resource must be used for both penetration test and design phases.

3.1.2. WebSocket (TCP: 80)

WebSocket protocol is a low latency (real-time), full duplex (bidirectional), long running (persistent), single connection (TCP) between a client and server. WebSocket provides benefit for real-time, live text chat, video conferencing, VoIP, IoT control and monitoring. The full-duplex aspect of WebSocket allows a server to initiate communication with a client whenever it would like, which is contrary to other protocols, such as REST/HTTP, in which the client must initiate communication. Bidirectional connection allows the server to update the client application without an initiating request from the client. WebSocket not only allows for low-latency communication between a server and client but also reduces network traffic by eliminating the need for a client to send a packet just to request data from the server (the server can send data as soon as they are available or when states have changed without the need to issue a request). For this reason, WebSocket is commonly used in applications like traffic reports, browser-based multiplayer games, and IoT application, e.g., to control servomotors [73]. In addition, WebSocket is used like transport layer of messaging IoT protocols, standards, and buses like XMPP, AMQP, MQTT, and JMS [74]. Since WebSocket is a very young technology, the security best practices around WebSocket are still evolving. The common security levels used in WebSockets not only include WSS (WebSockets over TLS) but also include avoid tunneling, validate client input, validate server data, authentication/authorization, and origin header [75]. However, Reference [76] demonstrates attacks such as fingerprinting and fuzzing, JavaScript overload and denial of service (DoS) to both client and server.

3.1.3. JMS (TCP)

Java Message Service (JMS) is a Java Message Oriented Middleware (MOM) API used to create, sending, receiving, and reading messages between two or more clients. JMS is a component of the enterprise edition of the Java EE Platform, and it was defined according to the JSR 914 specification, supported by the community. Therefore, JMS enables communication among the different distributed application components to be coupled, asynchronous and reliable as well as, supports publish-and-subscribe and point-to-point

routing [77]. The main JMS constraints are because it is a standardized API for JAVA only and it does not define a wire protocol. Therefore, the JMS deployments from several vendors are not interoperable, for instance, with Microsoft NMS (.Net Messaging Service). In that sense, two programs written in two different programming languages cannot communicate with each other over asynchronous messaging. The JMS message objects can be configured to include more information in the message. This information is useful when more complex business logic is required in enterprise messaging applications [78]. JMS does not provide an API to control messages' integrity and privacy, and neither specifies how distributed digital signatures or keys are. Each JMS provider manages security specifically. For instance, the JMS provider TIBCO EMS supports Java clients that can use either the Java Secure Sockets Extension (JSSE) Java package, or an SSL implementation [79]. JMS's leading vendors provide several levels of security. Typically, this means supporting facilities for client authentication and access control. As well as HTTP, JMS supports TLS. Java Authentication and Authorization Service (JAAS) supports several of the major JMS implementations to provide authentication and authorization [80]. In spite of that, McAfee describes in Reference [81] penetration-testing techniques to assess the security of ActiveMQ (based on Enterprise Messaging Systems (EMS) written using JMS API), as well as demonstrates vulnerabilities as insecure communication, insecure password storage, and weak encryption password. However, deserialization vulnerability to JMS is presented by Reference [80].

3.1.4. DDS (UDP: 7400, TCP: 7400)

Data Distribution Service (DDS) is a data-focused publishing and subscription protocol that grew out of the aerospace and defense environments and was developed by the Object Management Group (OMG). DDS has been designed to handle applications critical to the enterprise such as air traffic control, financial trading, and intelligent network management. The current application environments range from autonomous vehicles through smart grid management to power generation (Table 1). DDS provides to both publishers and subscribers with a scalable, real-time, reliable, high-performance, and interoperable data exchange. In addition, DDS specifications set important protocols as part of the DDS suite: Data Centric Publish Subscribe (DCPS); DDS Interoperability Wire Protocol (DDSI) and Real-Time Publish-Subscribe (RTPS). DCPS provides a set of tools that targeting real-time information-availability [82], while DDSI ensures portability and interoperability application [83]. Finally, RTPS manages the discovery process [84]. By default, DDS uses UDP but supports other options like IP multicast [85] and TCP [86]. Since DDS is language and operating system independent, it is suitable for running on both embedded devices and large-scale enterprise systems [87]. The DDS security specification determines both the Security

Model (SM) as well as the Service Plug-in Interface (SPI) architecture [88]. DDS-SM is implemented through the invocation of SPIs. The SPI enables to DDS users to customize the behavior for authentication, access control, encryption, data tagging and digital signing logging [88]. Some vulnerabilities are tested in Reference [89], e.g., listing the devices that are communicating using DDSI-RTPS is enabled via an unauthenticated client in both passive and active mode. DoS attack is achieved in the same reference. In addition, Reference [90] shows how DDS can be manipulated to support malicious activity through client-side attacks.

3.1.5. XMPP (TCP: 5222, 5269, 5280, 5281)

The Extensible Messaging and Presence Protocol (XMPP) is a semi-real-time message exchange protocol that transmits XML elements. By default, XMPP is used to deploy instant messaging applications, multimedia, lightweight middleware, social networking services, and IoT applications. The fast and asynchronous exchange of small payloads of structured information between entities is possible by XML streams and XML stanzas [91]. An XML stream is a container that exchanges XML elements between two entities through the network [91]. The XML stanza is a discrete semantic unit of structured information that is sent between entities through an XML sequence [91]. The XMPP client device nodes establish asynchronous communication with the XMPP server, which is an intermediary component that provides routes among both type of clients, senders, and receivers, i.e., entities [92]. The XMPP core specification includes security features. By default, the client must establish an XML stream with a server authenticating itself via the credentials of an account registered through SASL's negotiation with Public-Key Infrastructure X.509 (PKIX) certificates to provide strong authentication [91]. In addition, the XMPP specification introduces the concept of a security label, or confidentiality label, which allows a structured representation of sensitive information [93]. The security labels are used in combination with an entity authorized to access and a security policy to control access to each piece of information. For instance, a message could be tagged as "SECRET" and therefore require that both, the sender, and recipient are authorized to access the "SECRET" information. In addition, the specifications establish requirements such as that all communication must be done via properly encrypted links and the data must be encrypted using industry standard encryption on all links and end-to-end. In that sense, given the existing limitation of end-to-end encryption techniques due to the lack of full stanza encryption, they promote the use of Double ROT-13 as a transparent encryption that provides excellent interoperability benefits [94]. However, vulnerabilities and backdoors appear due to poor quality code implementations (e.g., buffer overflow attacks), poisoning blacklists, attacking the Domain

Name System (DNS) infrastructure, amplifying network traffic, and hijacking the TCP communication for both clients and servers are referred in Reference [95].

3.2. IoT Protocols and standards

Based on the common criteria established in the introduction to the section, this sub-section analyzes the protocols, standards, and buses associated with the IoT category according to Table 1.

3.2.1. Wi-Fi

The IEEE 802.11 standard was designed to be the comparable, i.e., equivalent, to the physical and MAC layers of the Ethernet standard (IEEE 802.3), so the distinction between a Wi-Fi and an Ethernet network is the way the frames are transmitted. The IEEE 802.11 standard contains other standards as IEEE 802.11b, IEEE 802.11g, and IEEE 802.11n over 2.4 GHz with speeds of up to 11 Mbit/s, 54 Mbit/s, and 300 Mbit/s, respectively. IEEE 802.11ac is another included standard that operates in the 5GHz band and although it has a smaller range (approximately 10%) with respect to the other standards, there is less interference, because there are no other technologies as established as Bluetooth and ZigBee in this frequency. To ensure security in a Wi-Fi network there are several alternatives ranging from using encryption protocols such as WEP, WPA and WPA2 (IEEE 802.11i) to MAC filtering and IP tunnels (IPSEC). In addition, there areWPA2 Enterprise solutions that use RADIUS back-end servers to store user credentials and even relay the authentication process response to the network access server (NAS), granting access to network resources [96]. However, it is also known that vulnerabilities exist at several levels of the security algorithms. For instance, in WEP most vulnerabilities are introduced because of a weak encryption scheme was implemented while in WPA (2)-PSK it is the authentication mechanism that introduces one significant vulnerability in this algorithm. However, there are different types of attacks on Wi-Fi networks to the extent that there are numerous tools to carry out penetration tests such as the one detailed in Reference [97]. The following two references show examples of some of the most common attacks on Wi-Fi networks. Since, WPA2 still uses a password as the key to authenticate the user; it is susceptible to password-based attacks such as brute force, dictionary, rainbow and the more prominently: the phishing [98]. However, Reference [99] mentions other security issues and commons attacks over Wi-Fi networks such as non-authorized access to target information, replay, DoS, and Pseudo-AP interference. In addition, Wi-Fi can also be used as vector attack since that Wi-Fi offloading contributes to mitigate the gap between cellular network capacity and mobile data traffic. Therefore, Reference [100] demonstrates a method to build a low-rate DoS attacks via the

offloading architecture. Finally, other possible attack in Wi-Fi is the deauthentication attack, which addresses the communication between a router and the device for the effective deactivation of the Wi-Fi connection on the device. The deauthentication attacks use a deauthentication frame, which is sent from a router to a device, forcing the device to disconnect, therefore this attack does not require credentials.

3.2.2. 4G/LTE

To address the growing use of mobile data through video, image, audio, and text enabled by applications such as social networks, 3GPP specified both Long Term Evolution (LTE), and LTE-Advanced (LTE-A) standards that became mobile broadband wireless technologies [101]. LTE divides the protocol stack into user plane protocols, which support routing of users' data between UEs and S-GWs and control plane protocols that are used for exchanging signaling messages between various devices within the network [102]. On the air interface (Uu), the user equipment (UE) functionalities are controlled by the Mobility Management Entity (MME). However, the communication between UE and MME is established via the evolved node base stations (eNodeB) [102]. The eNodeB supports both user plane and control plane protocols. The user plane protocols include the packet data convergence protocol (PDCP), the Radio link control (RLC), the medium access control (MAC), and the physical (PHY) layer protocols. In addition, the protocols of the control plane include the radio resource control (RRC) protocols [102]. The Uu interface is further divided into two levels of protocols: the access stratum (AS) and the non-access stratum (NAS). The MME signaling lies in the NAS level but is transported within the network using AS protocols. The LTE-A system specified by 3GPP LTE Release 10 was created to improve LTE systems, i.e., to handle significantly superior data usage, even smaller latencies, and higher spectral efficiency [103]. Additionally, both systems were designed to handle features such as IP compatibility, complete interoperability with other wireless networks and different kinds of base-stations (BS), as well as nodes retransmission in a large cellular, i.e., macro cellular network [101]. LTE/LTE-A improved security over its predecessor Universal Mobile Telecommunication System (UMTS). For example, to ensure mutual authentication between the UE and the MME, LTE improved both the authentication process and key handle with respect to UMTS. The MME is the termination point in the network for ciphering/integrity protection for NAS signaling and handles the security key management. Additionally, the key hierarchy and handover key management mechanisms were introduced to improve the access security and the mobility process in the LTE architecture [104]. Additionally, the "key hierarchy" and "handover key management" mechanisms were introduced to improve the access security and the mobility process in the LTE architecture [104]. The introduction of

new entities such as Machine-type Communication (MTC) [105], Home eNodeB (HeNB) [106] and Relay nodes [107] have been another mechanism used by LTE-A to improve security with respect to LTE. Despite this, several vulnerabilities have appeared in LTE/LTE-A technologies. However, Reference [101] summarizes vulnerabilities of the LTE system architecture, the LTE Access Procedure, the LTE Delivery Procedure, the IMS Security Mechanism and the MTC Security Architecture. For each case, the same reference ([101]) includes attack patterns. Although the same reference ([101]) proposes solutions for the issues listed, the experts of this reference argue that many security issues remain in LTE and LTE-A networks.

3.2.3. 5G

The fifth-generation mobile network (5G) aims to address the limitations of previous cellular standards, which makes it a key factor for IIoT environments. In general terms, the architecture to be deployed must be between the implementation of a complete autonomous 5G (SA) network that provides experience with end-to-end (E2E 5G) or implementation of a non-autonomous 5G (NSA) network will be complemented and supported by the LTE network [108]. The stand-alone 5G network introduces both, a new 5G-air interface, the "new Radio Access Network" (RAN) that focuses on defining the new radio access, which is flexible enough to support two frequency bands: lower than 6 GHz and higher than 24 GHz and a 5G Core (5GC). Due to the broad range of carrier frequencies supported, orthogonal frequency division multiplexing (OFDM) is the base of the 5G new radio (NR) air interface. This model requires interaction with the LTE network to cover areas outside 5G and to integrate 5G users with non-5G users [108]. The non-standalone (NSA) 5G network contains 5G NR cells connected to the Evolved Packet Core (EPC), which provides LTE as the core. Therefore, 5G cells are fully dependent on the LTE network for control functions and additional services. The NSA architecture functions as a master-slave where the 4G node is the master and the 5G access node is the slave [108]. However, the 5G efficiency includes enhancements to the self-organized networks 5G (SON) and Big Data capabilities, MIMO enhancements, improved power consumption, support for exchange of device capabilities, and a support study for non-orthogonal multiple access (MOMA) [109]. Considering the depth of 5G, for example, in IIoT environments, as analyzed in Reference [110], security becomes an imperative factor, becoming a threat surface for 344 5G. For this reason, 5G has considered security in its design phase through the SA3 Working Group (WG), which is responsible for security and privacy in systems, determining both security and privacy requirements, and the specification of security architectures and protocols as well as the availability of cryptographic algorithms that must

form part of the specifications. Reference [111] sets out the main security features in 5G such as increased home control, the unified authentication framework, the security anchor function (SEAF), the subscriber identifier privacy (SUPI), the subscription concealed identifier (SUCI), the subscription identification security, the subscription identifier de-concealing function (SIDF), the permanent equipment identifier, the globally unique temporary identifier, the procedure for using the temporary subscriber identifier, the subscriber's privacy, the secure steering of roaming, the security edge protection proxy (SEPP), the UE assisted network-based detection of false base station, the network redundancy at the 5G core, as well as the network slicing. In addition, 3GPP has included for 5G an entire security architecture based on elements such as network access security, network domain security, application domain security, SBA domain security, and security visibility and configurability. Despite the high number of security features introduced in 5G, Reference [111] sets out the landscape of the 5G threats, divided into threats in terminals, network or access nodes, core network and external and internal services and applications. Reference [111] illustrates a scenario where adversaries exploit zero-day vulnerabilities in devices or terminals belonging to the massive IoT (MIoT) from which they perform a DDoS attack on a 5G RAN. In addition, Reference [111] mentions the rogue base station (RBS) threat, where an attacker can use the RBS directive to launch different attacks on mobile users and networks. The RBS disguises itself like a real, i.e., "legitimate," base station to address a MITM ("Man-In-The-Middle") among the mobile UE and the mobile network. Finally, in Reference [111], threat scenarios, both the privacy of the subscriber and the heart of the network, are carefully analyzed.

3.2.4. MQTT (TCP: 1883, 8883)

MQTT is a message transport protocol for client-server environments, which includes as main features to be lightweight, designed to be easy to use, and intended for use in very large environments, including limited environments, e.g., m2m, IoT and IIoT, i.e., scenarios that require a tiny code and/or high bandwidth [112]. The MQTT protocol is covered by "ISO/IEC 20922:2016, which represents theMQTTv3.1.1 according to [112]. In addition, MQTT version 5.0 was released in 2019 and includes new features like enhanced authentication, flow control, maximum packet size, user properties, server keep alive, assigned client ID, and others [113]. MQTT works over a transport protocol, which offers bidirectional, ordered and loss-free connections, e.g., TCP. MQTT also uses a publication/subscription messaging pattern that provides one to many distributions of messages and application decoupling. MQTT, is agnostic to the payload content, which presents three QoS attributes for message transmission: (QoS 0) "At most once," (QoS 1)

"At least once," and (QoS 2) "Exactly once" [114]. The QoS level zero guarantees a best-effort delivery, the QoS level 1 guarantees that a message is delivered at least one time to the receiver and the QoS level 2 guarantees that each message is received only once by the intended recipients. In addition, an important feature is that the protocol provides a small transport overhead and minimized packet exchanges to decrease network traffic [112]. MQTT provides their own authentication mechanism, using CONNECT Packet that supports username and password. In addition to the username and password, MQTT clients provide other information that can be used for authentication such as client identifier and X.509 client certificate. The authorization is further another security mechanism used by the protocol via access control list (ACL) [115] and role base access control (RBAC) [116]. External authentication schemes such as OAuth 2.0 or LDAP are also supported by MQTT [117]. Other important security feature supported by the protocol is data encryption, which can be independently managed via SSL or TLS. The port 8883 is exclusively reserved for MQTT over TLS [118]. Despite the numerous security mechanisms supported by MQTT some issues such as lack of authentication, authorization, confidentiality, integrity are analyzed in [119], as well as different attack scenarios are examined in Reference [120].

3.2.5. CoAP (UDP: 5683)

The Constrained Application Protocol, CoAP is a protocol focused on web traffic for the use of nodes and networks with restricted resources. These nodes, for example, have eight-bit microcontrollers with low quantities of memory, whereas a restricted network are, e.g., 6LoWPANs, which provides a transition for the IPv6 over WPAN, and address both high-rate packet error and high-throughput. Since CoAP is built for m2m environments, e.g., "smart energy" and "building automation," it is oriented on the client-server paradigm with an architecture grounded in RESTful. It therefore adopts RESTful concepts, where the resources are "abstractions" controlled by the server, which put it to disposition by a process that uses a Universal Resource Identifiers, also known as URI, for identification [23]. In addition, the protocol CoAP was designed over UDP, it uses Datagram Transport Layer Security (DTLS) to provide the same properties that SSL/TLS provides to TCP connections, i.e., authentication and end-to-end security [121]. Since, some networks do not forward UDP packets, recently, CoAP has been introduced over TCP, WebSocket and even incorporates TLS [122]. In addition, the protocol defines four security modes, which are detailed in Reference [123]: "NoSec," "PreshardKey," "RawPublicKey," and "Certificates." Reference [124] proposes CoAP security implementations based on DTLS and IPSec. However, even though CoAP supports multicast connections, DTLS only secure unicast message [123]. The biggest challenge to CoAP is to maintain high levels of performance

while keeping security standards and ensuring protection [123]. By default, DTLS is the application layer security protocol that CoAP can deploy, however it is not exempt from constraints and additional problems, including compression due to message length and handshake [125], as well as not adapting to "CoAP proxy modes" [126]. Reference [127] demonstrates a series of attacks against CoAP and DTLS such as MITM attack, Sniffing, Spoofing, DoS, Hijacking, Cross-Protocol attacks, Replay attacks, and Relay attacks.

3.2.6. AMQP (TCP: 5673)

Advanced Message Queuing Protocol (AMQP) is an application layer protocol of open-standard, "message-oriented," which is used commonly in middleware. AMQP grew out from the financial market with the purpose of releasing users from non-interoperable and proprietary messaging systems. It is an enterprise-oriented messaging protocol designed to ensure security, reliability, and interaction with other systems [128]. AMQP was designed to support both operational models "request/response" and "publish/subscribe" [129]. In addition, as detailed in Reference [128], the protocol allows a wide variety of messaging related utilities, such as "reliable queuing," "topic-based publish-and-subscribe messaging," "flexible routing" and "transactions." Additionally, its communication system involves the publisher/consumer setting up an "exchange" with identification through a name and broadcasting it. Publishers and consumers then use the name to discover each other [130]. AMQP will exchange messages in a variety of forms: directly, in fan-out form, by topic, or based on headers [130]. The AMQP frame normally requires a fixed 8-byte header with small and customizable payloads with a size that based on the broker/server, or the programming language used. Reference [131] describes how AMQP enables a security based on different models of TLS negotiation: Single-port TLS Model, Pure TLS, and WebSockets Tunnel TLS Model. Therefore, AMQP explicitly allows the integration of both TLS (e.g., TLS virtual server extensions, also known as, SNI) and Simple Authentication and Security Layer, also known as SASL [131]. TLS is mostly associated with encrypting the connection and SASL with authenticating the connection. Access control based on authentication and authorization are other security mechanisms supported by AMQP [132]. However, Reference [133] illustrates how AMQP's security is often affected because of the code is "poorly written" by developers. In addition, Reference [134] analyzes known AMQP vulnerabilities and the threats/attacks associated to these vulnerabilities such as replay attack, masquerade, messages modification and DoS.

3.2.7. Bluetooth

At 2018, around 4 billion Bluetooth (BTL) devices were deployed as Reference [135] indicates. BTL technology opens up more markets, including "automotive," "smart buildings," "smart cities," and "smart homes," highlighted by the recently launched BTL mesh. The topology and architecture of BTL are described by [136], from which, two forms of BTL are mentioned: "Basic Rate/Enhanced Data Rate" (BR/EDR) and "Bluetooth Low Energy" (BLE). The first topology involves a star network topology in Piconet [24]. In addition, BR/EDR includes Scatternet topology, with every Piconet holding a unique master and several slaves, where others Piconets participate of a time-division multiplexing base. The second topology enables a BLE device to perform central and peripheral role. The device with the role of central initiates the connection establishment; if another device accepts, then this connection will have the role of peripheral. BTL Core system includes a host, a primary controller and none or some secondary controllers [136]. Furthermore, BTL specification provides interoperability among independent BTL systems because it defines protocol messages exchanged between corresponding layers and defining a common and specific interface between both controllers and hosts [136]. The Host layer contains the Security Manager (SM) module. It defines the mechanisms, i.e., the methods and protocols to pair the devices and to distribute the keys. The Security Manager Protocol (SMP) sets the format of the frame and type of the pairing command, as well as the frame structure and the time-out restriction. Key distribution methods used by SM provides both identity as well as encryption [137]. Pairing is carried out to exchange the keys that are used to encrypt the channel. The use of the keys can be extended to encrypt the link at reconnect the devices; verification of information, specifically, signed data; and carry out identification of address. Reference [137], defines three phases for the pairing: (1) "Pairing Feature Exchange," (2) "LE Legacy Pairing" or "LE Secure Connections," and (3) "Transport Specific Key Distribution." Through an exchange of messages, the intervening devices, i.e., central, and peripheral, examine the input and output (I/O) capabilities of the other, making possible to determine the appropriate pairing mechanism: "LE Legacy pairing" or "LE Secure Connection." "LE Legacy pairing" includes the methods "Just Works," "Passkey," and "Out of-Band" (OOB). "LE Secure Connection" includes the three previous methods and introduces "Numerical Comparison" [138]. However, the BTL protocol for both forms described has security limitations. For BLE in particular, tools such as Ubertooth One, Kismet, Wireshark, and Crackle [139] allow to exploit design and implementation vulnerabilities based on eavesdropping attack, packet decoding and packet injection [140]. A complete revision of BTL weakness is presented by the NIST Reference [141].

3.2.8. ZigBee

The ZigBee (ZB) standard was designed for wireless short-range communications (e.g., in the order of 10 to 100 m). As BTL, the physical layer is built on the IEEE 802.15.4 standard. ZB focuses on "low-power" environments and handles different profiles (e.g., "ZB Home Automation (ZBHA)," "ZB Health Care (ZBHC)," "ZB Building Automation (ZBBA)„" etc.), each device is designed in accordance with the needs of the environment [142], [143]. Specific profiles can be found in Table 1. A ZB network consists of three logical devices: the coordinator, the router, and the end device. The coordinator, manages the overall network; the routers, manages the whole Personal Area Network (PAN) and serves as intermediate nodes among the coordinator and the end devices; and terminal equipment or end device, is the simplest type of device on a ZB network, and it is often low power or battery-power. ZB protocol stack contains four layers [143]. Like other wireless protocols, ZB architecture comprises four layers: physical (PHY), media access control (MAC); network (NWK) and application (APL). NWK performs important functionalities such as network topology construction, network topology maintenance, as well as binding and naming. APL contains Application Support Sublayer (APS), ZB Device Object (ZDO) and software application. ZDO is in charge of managing the entire device, while APS supports both ZDO and ZB applications [142], [143]. By default, ZB provides security mechanisms in three protocol stack layers: the MAC, NWK, and APL [143]. ZB supports distributed and centralized network architectures, as well as associated security models. A distributed network is made up of two devices: a router and an end device. The router will be in charge of distributing the network key, which will be used by all network devices to protect the messages. This centralized model includes a third device, usually the coordinator, which represents the trust center (TC). The TC enables the other two devices, i.e., the router and the terminal equipment to connect to the network with the appropriated credentials. The TC can only issue encryption keys and it also sets unique TC link keys to each device on the network [143]. A tradeoff that the ZB Alliance chose to make between security and simplicity is the distribution of a unique network key when a device joins the network first time. There is a security issue in ZB HA 1.2 specification if an attacker captures ZB network traffic at the same time that a new device is being joined to the network [142]. This method has been removed in the ZB 3.0 specification and replaced with a process that requires a per-device installation code (like OOB in Bluetooth), i.e., used to generate a unique joining key, which is then used to acquire the ZB network key. The security of a ZB network is built on the ability to manage encryption keys properly [144]. However, the security of the ZB protocol is likely to be limited by physical capabilities, so lower-performance devices are also lower secure [145]. Significant

security risks such as: key management issues and secure routing issues (e.g., defaults keys) are mentioned in Reference [144] and sleep deprivation are mentioned in Reference [146].

3.2.9. RFID/NFC

The RFID allows objects automatic identification by radio waves, supporting contactless between devices. RFID tags include a micro-chip and a transceiver or antenna, which is only activated by a RFID reader that receives a signal from the tag. The RFID technologies are categorized according to three criteria: (1) operating frequency, (2) power supply, and (3) memory type. The Operating frequencies supported are in the following four frequencies bands: low (LF), high (HF), ultra-high (UHF), and microwave (MWF). The power source classification includes type of tags active, passive and battery-assisted passive. The memory type includes "read only," "write once read many," "read and write." There are passives RFID tags in all frequency's bands mentioned. The Near Field Communication (NFC) is a standard associated with passive high frequency tags. Reference [147] provides a representation of the NFC architecture, including all protocols that comprise the standard. In addition, NFC includes the following standards: ISO/IEC 14443, FeliCa, and ISO/IEC 18092. Beyond RFID/NFC technologies, an RFID/NFC system goes through the integration of these technologies in IIoT environments. In addition to the already mentioned RFID/NFC tags and readers, the more general architecture involves RFID/NFC middleware and the business layer [148]. The RFID/NFC middleware performs functions such as reader management, information collection and integration, data filtering, while the business layer integrates systems such as enterprise resource planning (ERP), customer relationship management (CRM) and track and trace applications. For this reason, the security threats contained in RFID systems are different from traditional wireless security threats and can be grouped into: (1) physical components of RFID (e.g., clone tags, reverse engineering, tag modification), (2) the communication channel (e.g., eavesdropping, skimming, repeat attack), and (3) threats to the global system (e.g., spoofing, Denial of Service (DoS), and tracking and tracing) [149]. A detailed analysis of the vulnerabilities affecting RFID systems for each category mentioned can be found in Reference [138], while in particular for NFC some vulnerabilities due to a Cryptographically Weak Pseudo-Random Number Generator (PRNG), relay attack, and eavesdropping are analyzed, respectively, in references [150][151][152].

3.2.10. LoRa

The LoRa is a proprietary technology of the Low Power Wide Area (LPWA) family, as well as a patented wireless technology of Semtech Corporation. The specification included in Reference [153], indicates the frequency ranges for LoRa, where regulators control the

proper use of assigned frequency ranges [154]. Since the wireless architecture and hierarchical organization used by LoRa is quite simple, it is suitable for IoT environments [36]. Since Lora defines only the PHY layer, LoRaWAN is the protocol developed to define the top layers of the network. LoRaWAN is a cloud-based MAC protocol, which acts primarily as a NWK protocol for managing communication, i.e., the network built between LPWAN gateways and end node devices [36]. The network topology is star, enabling the interconnection between terminal nodes and base stations/gateways. The gateways allow messages to be exchanged over the internet with the LoRa network servers. These servers ensure the information exchanged by the terminal devices. LoRa defines several kinds of devices, keys, and encryption capabilities to ensure the network. The cipher suite relies on AES-128 working in CTR mode. Throughout the LoRaWAN network architecture multiple layers are encrypted with the same scheme [153]. LoRaWAN uses different encryption keys to protect devices, the network layer, and the application layer. In this way, the technology ensures the security of the lower layers, i.e., intermediate nodes such as gateways and cloud routers, to perform network routing-and maintenance at the same time as ensuring application data confidential. Al though LoRaWAN provides extreme-to-extreme (end-to-end) security, which include the use of keys in both layers APL and NWK, an attacker that get physical access will be able to compromise the LoRa terminal devices and therefore not only devices but also NWK keys. In addition, Reference [154] mentions that LoRa is susceptible to jamming, replay and wormhole attacks.

3.2.11. NB-IoT

NB-IoT ("Narrowband-Internet of Things"), such as LoRa, is an LPWA technology. It was designed to support a broad variety of different IoT devices as well as services. As Table 1 shown, NB-IoT is used in services such as AMR (e.g., smart metering) and SHA (e.g., smart parking). Since the LTE standard is the base of NB-IoT, we can say that it is a light-weight version of LTE [193]. However, NB-IoT is as simple as possible in order to minimize costs of devices and reduce the battery consumption, and thus it removes many capabilities/features of LTE, such as handover, monitoring the channel quality, "carrier aggregation", and "dual connectivity" [194]. The NB-IoT core network is based on the evolved packet system (EPS) and is defined for the cellular Internet of Things (CIoT) both the optimization of the user plane CIoT EPS and the optimization of the control plane CIoT EPS [194]. The cellular access procedure of an NB-IoT user is like that of LTE. In order to optimize the control plane CIoT EPS, the "evolved terrestrial radio access network" UMTS (E-UTRAN) is responsible for radio communications among the EU and the MME and includes the eNodeB [195]. The data was then transmitted to the packet data network

gateway (PGW) using the service gateway (SGW) [195]. If the data is not IP, the service capability exposure Function (SCEF) is transferred to the node, which provides machine-type data through the control plane and supplies an abstract interface for the services. From EPS optimization of the CIoT user plane, IP and non-IP data can be transmitted by radio carriers through SGW and PGW to the application server. Therefore, for NB-IoT both the existing/current E-UTRAN network architecture for LTE and the backbone are re-used [195]. NB-IoT also adopts security features from LTE such as authentication and encryption. A security feature applied is Data over NAS (DoNAS), which enables the network to transmit user data via the MME into NAS signaling messages. DoNAS is used to carry IP as well as non-IP traffic. Therefore, the entity's data is encrypted, ensuring the integrity through the same mechanism/process reserved for network signaling [196]. Another security feature is transporting data via Non-IP Data Delivery (NIDD) using SCEF, which provides a means to securely display service and network capabilities/features through network "application programming interfaces" (APIs) [196]. The use of VPN together with private, secure access point (APN) names dedicated to specific entities to keep their data communications isolated from the rest of the traffic is another security feature [196]. However, like the reference [193] mentions, perceptron layer is susceptible to both active and passive attacks. In passive attack, the attacker will simply monitor the network traffic while an active attack will compromise the integrity of message, as well as the forgery of encrypted data. At the perceptron layer, each node is able to communicate with the base station directly so routing security issues are prevented during networking [193]. In addition, the reference [197] establishes proof of concept of an attack based on scan using malicious User Equipment (UE).

3.2.12. WirelessHart

WirelessHart is a specification to communicate Hart protocol (Hart-IP is detailed in Section 3.3.6), which is a digital transmission method in process instrumentation through a wireless connection. It was issued by HART Communication Foundation (HCF) as HART Version 7 in September 2007 [40]. From the OSI model WirelessHart, communication stack contains five layers: PHY, data link layer (DLL), NWK, transport (TRT) and APL. The PHY is built on "IEEE 802.15.4-2006 2.4 GHz DSSS," i.e., direct sequence spread spectrum. This layer includes two important features: channel hopping and transmit power. DLL is based on "time division multiple access" (TDMA), which applies a distinct feature of this layer: a rigid collection of timeslots with a time duration of 10 ms, which are merged to generate one or more superframes [160]. A superframe groups a sequence of consecutive time slots. All superframes starts with Absolute Slot Number (ASN) 0. All nodes must synchronize with

their neighbors to determine the exact time of transmitting or receiving, according to the ASN. A time interval (timeslot) ensures the communication of the data and its acknowledgement. Before the transmitter begins to transmit, the receptor node activates the radio in reception mode for a period, also known as, guard time. If the receptor node is not receiving data during this period, then the node returns to standby (sleep) mode and expects the next time interval (timeslot). On the opposite, it waits for the end of the transmission, then it validates it and transmits the acknowledgement to the transmitter node. As we mention, WirelessHart includes other layers such as network and transport layers. The key function of the network layer is to transfer information between a source and a destination promptly and reliably. This process is called routing, which includes some ways to route data. For instance, there are four approaches established to route: "source routing," "graph routing," "superframe routing," and "proxy routing." Reference [161] defines in detail each of routing approach. Finally, APL defines several device commands, responses, data types, and reports of the network and devices status [161]. We consider important to mention when the sensor network is deployed, the devices are organized to build a mesh topology. In addition, on top of the network are both a central gateway and central network manager. WirelessHart is a protocol with security by design. The protocol provides important security features, such as confidentiality, authentication, and integrity in both communication "hop-to-hop" as well as "end-to-end," through the cipher suite AES 128-bit key combined with CCM mode [162]. Since it is a mesh network, the nodes depend on their neighbors to send the data to the network administrator. Although at the "hop-to-hop" level, the data link layer does not encrypt the information, this layer guarantees both the authenticity and integrity of the information, protecting the network from attackers [163]. Additionally, the network layer provides "end-to-end" security, where only the frame header is not encrypted [164]. Despite WirelessHart standard provides several security mechanisms that make the network more robust to attacks, Reference [160] performs attacks against the network such as Jamming and Advertisement-based Attacks.

3.2.13. Z-Wave

Z-Wave is a proprietary, low-energy wireless protocol that builds a mesh network to communicate from one device to another. Z-Wave reliably transmits short messages from the control device to one or more network devices with minimal noise. Each device or node of the network must distinguish from the Network ID or Home ID the nodes of its network from the nodes of the neighboring networks while from the Node ID distinguishes the nodes of its network. Since all nodes participate in the mesh, network sends and receives control commands allowing Z-Wave to cover a larger area. Additionally, the mesh network uses the

intermediate nodes to find a route where the nodes connect to each other to reach the destination. However, the smart home networks, which implement Z-Wave, contain up to 232 devices divided into controllers and slaves. The slaves receive and execute commands from the controller and do not contain a routing table, although they can have a network map with routes to the devices. Although there may be more than one central controller that handles both routing and security of the network, a single controller provides reliable information about the network topology. Z-Wave is composed of an architecture based on four layers: APL, Routing, MAC, and Transfer [165]. Although security for Z-Wave is in the development phase, for example, from the design of devices that allow the capture of radio packets and software, some steps have been taken in the detection of vulnerabilities. Reference [166] provides the presentation of the open-source tool "EZ-Wave," which allows penetration tests on Z-Wave networks and shows "a rapid process for destroying florescent lights." Since the insecure mode of Z-Wave is based on a unique identifier and does not introduce encryption, Reference [167] (using the "Waving-Z" tool, which encodes and decodes Z-Wave frames) allows the Home ID/Network ID to be modified. Although secure mode supports encrypted communications, these are not supported by all devices, are not enabled by default, require additional action for activation, and present poor information for clients. For this reason, Reference [168] has detected vulnerabilities from a manufacturer's implementation error. A vulnerability was discovered in "AES-encrypted" Z-Wave door locks, which would be exploited remotely to unlock doors despite unknown "encryption keys" and because of the changed keys, succeeded network messages/commands will be bypassed by the established controller of the network. However, as of 2016, Z-Wave Alliance announced strengthened security mechanisms, which include encryption standards for transmissions between nodes, and involves new pairing mechanism for each device, based on unique PIN or QR codes on each device, i.e., an OOB mechanism.

3.3. OT PROTOCOLS, STANDARDS AND BUSES

Based on the common criteria established in the introduction to the section, this sub-section analyzes the protocols, standards, and buses associated with the OT category according to Table 1.

3.3.1. PROFINET

PROFINET (acronym for "Process Field Net") is an open Industrial standard Ethernet-based, i.e., built and maintained by PROFIBUS & PROFINET International (PI). This is contained in the standards: "IEC 61158" and "IEC 61784." PROFINET satisfies the requirements for

automation (i.e., "plant and machine manufacturers") and fully compatible with Ethernet according to IEEE 802.3 and multi-protocol parallel Ethernet operation. PROFINET offers two approaches: PROFINET CBA ("Component-based Automation") and PROFINET IO (IO acronyms mean: "discrete input," "discrete output," "analog input" and "analog output"). PROFINET CBA is appropriate for communication m2m via TCP/IP, and it is also used for real-time (RT) communication [169]. PROFINET IO contains both RT communication and real-time isochronous communication (IRT) [170]. PROFINET uses TCP/IP (or UDP/IP) communications for certain non-time critical tasks, such as configuration, parameterization, and diagnostics [171]. This occurs because when a packet is transmitted from one PROFINET node to another via TCP/IP, the delay occurs in packing and unpacking across layers, this is a non-deterministic process, which produces jitter. Therefore, the TCP/IP communication is unsuitable for time-critical environments [171]. PROFINET RT handles time-critical data exchange. An arriving PROFINET RT Ethernet frame has the PROFINET EtherType: 0×8892. Upon arrival, the frame is directed to the PROFINET application directly from Layer 2 to Layer 7 [171]. Unlike TCP/IP, this process is deterministic. For the most demanding applications, PROFINET can use additional techniques for even faster performance with the PROFINET IRT channel. PROFINET IRT is a step beyond PROFINET RT. PROFINET is based on a phased security concept [172]. The protocol specifies an optimized security model according to the application environment, security zones defined. However, different attacks modalities and how to take a control of a PROFINET IO node is described in Reference [173].

3.3.2. Niagara AX (TCP: 1911)

The "guide specification to include smart buildings" describes the capabilities and functions of the Niagara Framework, which is scaled on any systems connected network, locally or remotely, and accessible via the internet via web browsers over OT and IIoT networks [174]. The Niagara Framework was designed to allow integrators and developers to connect, manage and control any device, regardless of manufacturer, using any protocol. The Niagara Framework is described by Reference [174] like a Framework Architecture for Edge-to-Cloud technology. Niagara designed the Tridium Fox protocol to drive tunnels to SCADA networks [175]. It is applied in building automation environments (Table 1). As Reference [175] also indicates; in contrast to other protocols, Tridium Fox has not a straight communication with an industrial asset, however, it facilitates interaction among workstations and devices (e.g., Modbus, BACnet, and DNP3). Tridium includes four key parts: "Niagara AX Architecture," "Niagara Structures," "Niagara Protocols," and "Niagara Platforms" [176]. Niagara AX is an open framework based on Java, which could connect both to practically

any device (for example, embedded systems) and independent communication protocols, i.e., is able to integrate several manufacturers. We can therefore conclude that it is an agnostic communication framework. It contains a full set of graphical tools that allow users to create sophisticated applications in a drag-and-drop environment to easily handle and manage assets via a web browser. The latest release of Niagara AX was version 3.8, which support TLS 1.2 [177]. However, some attacks due to improper input validations, improper access control and clear text passwords are described by Reference [178].

3.3.3. CAN

Currently, all vehicles use the CAN bus, Controller Area Network, as the main serial communication protocol between electronic control units (ECU) for reasons such as ease of adding and removing nodes, failure of a node does not bring down the bus, is implemented with a relative low amount of wires and allows independence between nodes [179]. Another functional strength of the CAN bus is to be a standard that allows, for example, microcontrollers and devices/equipment to communicate with other applications without a central host. Its use extends to applications in autonomous vehicles. For instance, within the implementation proposed in Reference [179], the CAN bus allows the transmission and reception of data to the main card, as well as the communication between two different ECU through only two CANL (Low CAN) and CANH (High CAN) wires, thus reducing the number of cables in the vehicle. In general, we should mention that the CAN bus is a multi-master serial bus that connects ECU, also known as nodes, therefore, a CAN network will need at least two nodes. Each node/ECU requires a central processing unit (CPU), a controller and a transceiver. In this way, the node can send and receive messages, although in a half-duplex way. Each frame contains an identifier (ID), which signifies the message priority, eight bytes of data, cyclic redundancy check (CRC), an acknowledgment field (ACK), and overload. There is an extended version of 64 bytes of data per frame (CAN FD). The devices that typically interconnect CAN are sensors, actuators, and control devices. CAN has no built-in default security mechanisms, such as encryption, therefore organizations are expected to implement their own security mechanisms such as authentication of network commands or devices. If an inadequate implementation is done or a security mechanism is not implemented, then the protocol is susceptible to, for example, package interception as well as MITM, because CAN messages are broadcast. Additionally, Reference [180] mentions other possible attacks like DoS or replay attacks and concludes that the opponent will be able to send valid messages and thereby control the physical components of the vehicle or system. Reference [180] establishes an authentication model based on SHA-1 due to the efficiency and speed for the HMAC, also implemented the timestamp for the verification of

the message. This implementation did not imply a severe impact on the transmission speed and latency and the prevention of both the DoS attack and the replay attack was demonstrated. However, the implementation was carried out on a test network with only two nodes.

3.3.4. Siemens S7 (TCP: 102)

PLCs are responsible for controlling critical processes and they are considered essential to field automation [181]. Siemens Simatic S7 or Simatic STEP 7 is a PLC product line that succeeded to Simatic S5. Siemens PLCs use two protocols to communicate via Ethernet through communication processors (CP): Open TCP/IP and S7 Protocol. For instance, CP 1543-1 is a communication module to support TCP/IP and multicast over UDP connection [182]. When the PLC S7 (i.e., S7-1500) uses the CP module (i.e., CP1543-1) is always the server (passive connection establishment) [182]. "Open TCP/IP" is a TCP/IP protocol implementation, dedicated to connect PLCs with non-Siemens hardware [183]. S7 protocol is known as the backbone of Siemens communications. This Ethernet deployment is based on "ISO TCP (RFC1006)" that, for design, is a block-oriented system. S7 Protocol, ISO TCP and TCP/IP use the encapsulation model defined by OSI in which the protocol data is the "payload" of the subsequent protocol [183]. S7 is a command/reply-based protocol, in which each interaction is a command or a reply. S7 uses PROFINET, which is based on Ethernet, and is currently regarded as the most reliable "fieldbus." The PROFINET protocol is used for communication between PLCs and IO modules. Basically, PROFINET infrastructure are "industrial Ethernet cables," known as industrial "Cat5" or "two-pair Cat5," for connecting industrial fieldbus systems via TCP/IP. They are desirable for fixed or flexible and dynamic industrial automation applications, as they offer outstanding resistance to both active and passive electrical interference, as demanded by PROFINET and Cat5e specifications. Siemens recommends security measures based on defense in depth (DiD) strategies to minimizing risk of products like S7-300/400/WinAC/1200/1500 [184]. However, S7 is sensitive to attacks such as "spoofing," "session hijacking," and "denial of service" [185]. Other scenarios use Frameworks like Metasploit to carry out attacks as replay [181].

3.3.5. EtherNet/IP (TCP: 44818, UDP: 2222)

Rockwell Automation designed EtherNet/IP or "Ethernet Industrial Protocol" in the decades of the 1990s. It combines Ethernet with "Common Industrial Protocol" (CIP). CIP covers a complete set of messages and services for a wide range of manufacturing and process automation applications (Table 1). Open DeviceNet Vendor Association, Inc. (OVDA) maintains both EtherNET/IP and CIP. EtherNet/IP provides a certified standard for the

development of automation devices with the below properties: (1) It employs Ethernet, (2) it is based on a broadly accepted CIP protocol layer, and (3) it is a verifiable standard. CIP is a protocol used in the transfer of automation data between two devices. The CIP protocol represents each device in the network as a set of objects, where each object is just a gathering of data values related to a device [50]. CIP defines three types of objects: application objects, required objects, and vendor-specific objects. The required objects contain three other types of objects: identity object, message router object, and network object [50]. The protocol lacks built-in security protections and the cybersecurity for EtherNet/IP and CIP is built on a defense-in-depth approach based on external mechanism [154]. ODVA defines some best practices for different types of industrial network installations for EtherNet/IP [186], where it includes isolating the control network with a single and multiple controllers, VLAN, Firewalls, and DMZ. However, some vulnerabilities such as improper input validation have been confirmed and exploited by Reference [187].

3.3.6. Hart-IP (TCP: 5094)

The HART protocol "Highway Addressable Remote Transducer" is a universal standard to send and receive digital information via analog 4–20 mA cables between smart devices and control systems [188]. More recently, HART has been extended to include communication across IP networks (HART-IP) [188] as well as wireless mesh network (Wireless HART) [189]. HART-IP was built to enable HART devices over Ethernet. This implementation includes a conventional client/server architecture [190]. The client can be either a host system or a host application while servers can be Wireless HART gateways, HART multiplexers, HART Remote I/O or individual HART devices [191]. Client/server communication utilizes either/both UDP and TCP transport. In addition, servers support a minimum of two simultaneous client sessions [192]. Since HART-IP is predominantly used within the plant perimeter, security measures should be employed to protect the data during transport: firewalls, Virtual Private Network (VPN) tunneling, SSL, and remote authentication [193]. In addition, it is recommended to restrict HART-IP to internal networks. Traffic from HART-IP (usually UDP or TCP port 5094) should be restricted to a management VLAN segment with strong network controls. Some vulnerabilities, including DoS, in HART-IP networks are demonstrated by Reference [194].

3.3.7. BACnet (UDP: 47808)

The "American Society of Heating, Refrigerating and Air Conditioning Engineers," also known as ASHRAE, designed BACnet, which is the acronym for "Building Automation and Control Network." In general, BACnet is a communication protocol and a standard designed for both

building automation and control systems (see Table 1). The BACnet protocol determines the way and which messages (data frames) may be transmitted from a device/system to other. BACnet's architecture is formed by four layers (OSI model oriented): "Physical Layer" (PL), "Data Link Layer" (DLL), "Network Layer" (NL), and "Application Layer" (AL), although only NL and AL are purely BACnet. At a data link and physical layers level, BACnet uses several protocols, including "Ethernet," "BACnet/IP," "ARCNET," "MS/TP," "Lon Talk," or "PTP" (point-to-point) as detailed in Reference [195]. BACnet/Ethernet is used directly with Ethernet IEEE 8802.3 networks. It may run on different physical supports, such as cable or optic fiber. It is limited to physical infrastructure that only uses MAC addresses to establish communications. BACnet MS/TP is based in the master-slave model or the token passing model in the data link layer [196]. This BACnet variant uses a serial channel, typically RS-485, for communications [196]. The BACnet PTP type of media access control is only used over telephone networks [197]. The EIA-232 type of direct connection is less and less used, and Ethernet is usually preferred in its place. BACnet over ARCNET is variant allows using BACnet over a coaxial cable or a RS-484 serial cable. It improves slightly the BACnet MS/TP features, but few manufacturers support it. BACnet/IP is a standard that uses IP addresses and ports, which allows it to run over the available Ethernet architecture, as well as to use of VLAN networks [198]. The messages are transported using UDP. It is different from BACnet/Ethernet in the fact that it uses IP addresses instead of MAC addresses. BACnet uses broadcast messages. For instance, to identify connected devices, BBMDs ("BACnet/IP broadcast management devices") are needed [199]. BACnet protocol security includes security keys, encryption models or authentication systems [200]. BACnet defines six different types of security keys to be used depending on the task to be carried out. Keys are distributed to all devices from the network security key server, some of which are even distributed jointly. It is not necessary to assign the security key server function to a specific device, but there must be a server containing all keys and a list of all devices to be managed [200]. BACnet applies message security at the network layer. BACnet non-encrypted messages are placed in the data section of a new secure message. This encryption is subject to four network security policies depending on whether hardware security, protocol security or no security is applied [200]. In all cases either where a message is deprived of security measures, in the target device or because the target device has a different security policy, messages must be authenticated. Such authentication consists of validation of source MAC address, an unique message ID, a time stamp, and the message signature [200]. Reference [201] mentions BACnet devices implementations could be vulnerable to malformed packets and other types of attacks, so they can be considered un-robust and unreliable for handling irregular traffic. Techniques of attacks against BACnet as attacks on BACnet routing, network

mapping, DoS, and spoofing are included by the tool BAF (BACnet Attack Framework) [202]. BACnet Anomaly Detection Framework (BADF) offers a convenient approximation of BACnet's current attack surface [203].

3.3.8. Modbus (TCP: 502, TLS: 802)

Modbus was established in 1979 by Modicon as a serial communication protocol, open standard to be used by programmable logic controllers (PLCs). Modbus actually is open protocol for industrial networks, which recently includes several environments such as "building automation" or "energy management systems" (see Table 1). The Modbus functions are to control PLC, HMI, and I/O devices or sensors. Modbus can be used over Ethernet as well as serial cable. There are three established variations of the Modbus protocol: "Modbus ASCII," "Modbus RTU," and "Modbus TCP/IP." Modbus was originally developed using ASCII character to encode messages and this version of the protocol is still in use today. Modbus RTU is by far the most common implementation, based on the use of binary code and CRC error validation. Modbus RTU devices typically use one of three electrical interfaces: RS232, RS485, RS422, and a master slave architecture. SCADA/HMI systems typically would be the master, communicating with a series the Modbus slaves' device (e.g., PLCs). A Modbus serial network has a master device, which issues commands to the slave devices. The slaves will not transmit information unless they receive a command to do so. The big difference with Modbus TCP/IP is that a Modbus Application Header (MBAP) is inserted at the beginning of each message. Modbus TCP/IP uses the terms client and server instead of master and slave. Modbus clients send the commands (e.g., SCADA/HMI) [204]. Modbus has two strong constraints: (1) it has a limit to 240 devices per network, (2) although the protocol is handled and defined by the organization itself, numerous vendor extensions are owned but without documentation, leading in interoperability problems. The Modbus organization released security specifications, which provide robust protection by combining TLS with the traditional Modbus protocol. TLS will encapsulate Modbus packets to enable both authentication and ensure message-integrity [205]. The new protocol/mechanism takes advantage of the X.509v3 digital certificates for server and client authentication [205]. Reference [206], developed a proof of concept by implementing TLS over a Modbus channel for smart grids. The results determined that the solution accomplished request/response times significantly below the 16.67 ms period of the 60 Hz grid cycle, demonstrating a minor effect on smart grid applications. In addition, Reference [207], addresses the security problems of the Modbus protocol, through a new secure version of a RBAC model, which takes advantage of the authentication provided by TLS, as well as granting authorization of the client on the server, as well as of the Modbus frame. However, Modbus is sensitive to

classic information security threats as described in Reference below [172]. A summary of the attacks according to their threat categories, targets and impact on the control system assets is presented by Reference [208]. Other attacks such as MITM and DoS attacks are demonstrated by Reference [209].

3.3.9. OPC-DA (TCP: 135)

One of the greatest attempts for automation software standardization in the last years has been the access to device automation data, with several protocols, different bus systems and interfaces are available. The result was OPC-DA (Open Platform Communications-Data Access). After, other two important OPC specification was developed: Alarm and Event, abbreviated A&E [210], and Historical Data Access, abbreviated HDA [211]. The client/server architecture in the Microsoft components was the disruptive aspect of OPC. An OPC server encloses the information generated by the industrial process and makes it available through its interface. An OPC client connects to the OPC server to access the available information (see Reference [212]). Object mapping performed by OPC Classic is implemented via Microsoft technologies such as COM ("Component Object Model") and DCOM ("Distributed Component Object Model"). OPC Classic does not define security as part of the specifications, so it is delegated to DCOM/COM protocols [213]. The OPC Foundation establishes guidelines to configure the DCOM/COM layer to provide security mechanism [214]. In general, OPC Classic supports signing of the data flow and encryption among systems, identification and securing applications and user access rights. However, OPC tunneling solution provides security robustness [215]. OPC tunneling aims to eliminate DCOM, which is usually performed by exchanging the DCOM network protocol with TCP. The connection is set up between the OPC client and the OPC tunneling application that acts as an OPC server [215]. Other security recommendations to reduce the attack surface, layering defenses and defense in depth are analyzed through a practical example [216]. OPC Classic can be set up to provide security, but these security features are provided by the functionalities in Windows and DCOM/COM. However, it requires a lot of configuration knowledge, besides some environments do not support security configuration and some applications could not support some security configuration [213]. For example, OPC Classic uses DCOM, which employs RPC internally. By default, RPC uses dynamic port mapping. This means that it is very difficult to set up a firewall, since a large number of ports must be opened [217].

3.3.10. OPC-UA (TCP: 4840)

Aimed at the SOA (Service Oriented Architecture) paradigm, the OPC Foundation has created the new standard OPC-UA (Open Platform Communications-Unified Architecture) that unifies the entire specification of OPC Classic (OPC-DA): "Data Access," "Alarms & Events," "Historical DA," and "Complex Data and Commands." OPC-UA (Unified Architecture) can be implemented in a wide range of device types such as 16-bit up to 64-bit architectures x86, ARM, and PowerPC. It is also agnostic to the Operating System (OS), so it can be deployed in Windows, Linux, VxWorks, and different Real Time (RTOS) [218]. The UA workgroup set up a binary protocol built on TCP, which supplies the communication stack in three default-programming languages: "NET," "ANSI C," and "JAVA." OPC UA separates services from the implementation language protocol, which is the basis for flexibility and usability in domains even outside the classical communication model, SCADA-PLC, HMI-PLC, e.g., integration of the OPC-UA server with field devices can be with much reduced footprint [219]. Smart Grids is a field of action of OPC-UA with application in a wide range of devices ranging from controllers of wind or photovoltaic power plants to systems like SCADAs or Energy Management Systems (EMS) [220]. OPC UA is secure-by-design addressing security next concerns: (1) authentication of Users, application instances (Software), (2) confidentiality and integrity by signing and encrypting messages, (3) availability by minimum processing before authentication, and (4) auditability by defined audit events for OPC UA operations [221]. However, a full test of the reference implementation of the OPC UA communications stack, reported through Reference [222], revealed that some bugs detected in the dynamic code analysis have a significant negative impact on server availability due to memory utilization or failures. In addition, sequenceNumber is not proven in UA Secure Conversation, i.e., a security vulnerability. The proof of concept of the vulnerability exploitation is referenced in [222]. However, it is recommended to carry out specific proof of concept (PoC) of: (1) "Replay attacks exploiting missing tests of the sequenceNumber," (2) "Exploitation of compromised certificates whose validity is not detected due to missing tests of CRLs," and (3) "DoS attacks exploiting memory leaks" [222].

3.3.11. DNP3 (TCP: 20000)

DNP3 is made up of a suite of protocols employed in electrical grid automation environments. The Distributed Network Protocol was developed by GE-Harris Canada in 1990, has been extensively implemented in utilities such as electrical, water, sewage, oil, and gas. It was created for SCADAs interaction environments; the protocol enhances the data acquisition information and control commands transmission from master (control centers)

to remote stations (remote computers) via event-based data reports. In addition, DNP3 was designed so that data acquisition and control equipment can interact. In addition, it is widely employed for communications from "master stations" to "remote terminal units" (RTUs) or "intelligent electronic devices" (IEDs) [223]. The DNP3 stack is divided in three layers: link, transport, and application. The ability to transport generic data is given by the independence with respect to the channel, for example: serial or TCP/IP. Application Data Service Units (ASDUs) are entities that transmit a combination of functions code and objects through the application layer in a standardized data format, so that it can be used by the lower layers [224]. ASDU messages are divided into fragments. The maximum fragment size is associated with the implementation as it is defined by the buffer size of the receiver device, e.g., a normal range is 2 to 4 KB. A message larger than a fragment will require some fragments. Message fragmentation is delegated to the application layer. The DNP3 protocol provides a secure authentication (DNP3-SA), which is a one-side authentication (TLS with certificate authentication) procedure employed to protect the DNP3 messages transferred among interconnected stations are secure from unauthorized applications. The system works in two modes: "Non-aggressive" and "Aggressive" modes. DNP3 is an early "standardized SCADA protocol" that aims to enable a cryptographic security embedded in the operations [225]. Another way to provide a TLS-based connection to DNP3 is through IEC 62351-5, with which DNP3 is compliant. DNP3 has integrated the IEC-62351 security requirements/capabilities in the "IEEE 1815 DNP3" standard [226]. However, DNP3 vulnerabilities was demonstrated using attack patterns like fuzzing [227] and crafted malformed frames [228]. Since 2013 to 2014, over 33 CVEs related to input validation with DNP3 implementations [229].

3.3.12. ANSI C12.22 (TCP: 1153, UDP: 1153)

Since the datasets, data structures, and communications protocols for electricity meters were all exclusively proprietary, ANSI (American National Standards Institute) standards were established to describe the datasets and data structures of the meters (C12.19), and to enable an optical point-to-point optical communications protocol (C12.18) that would enable them to interface with ANSI standard meters. To allow the transmission and reception of this information, C12.18 was adapted to set up C12.21 for telephone modems, using point-to-point communication. Later, C12.22 was structured to include other communication networks such as TCP/IP or UDP/IP and SMS on GSM [230]. The ANSI C12.22/IEEE 1703 protocol defines the scope of "ANSI C12.22/IEEE 1703 Advanced Metering Infrastructure" (AMI) Application Layer message transport over a TCP/IP network in the smart grid environment [230]. The C12.22 standard specifies both a transportation-

independent application-level protocol for information exchange among nodes and a physical and data link protocol for meter connection and communications technology [231]. The "C12.22 IP" communications system include several elements, as Reference [232] detailed "C12.22 IP Node," "C12.22 IP Network Segment," "C12.22 IP Relay," and "C12.22 Gateway." The protocol units are ANSI C12.22 / IEEE STD 1703, IETF RFC 6142, IGMP with UDP multicast, TCP or UDP transport, Abstract Syntax Notation One (ASN.1) [233], and Object identifiers (OIDs) [234]. In the ANSI C12.22 frame, it is possible to verify the authentication mechanism used, as well as the user information exchanged through an EPSEM message. "Extended Protocol Specification for Electronic Measurement" (EPSEM) provides the ability to link commands to manage communications across a multi-node medium. A full deployment of the ANSI C12.22 security and the authentication process, coupled with the ANSI C12.19 event logger, enables a utility to achieve all the required features. As Reference [231] details, within the security mechanisms that can be included are: "encryption," "authentication," "credential management," "intrusion detection," "logging," and "auditing of all changes in data and configuration." However, Reference [235] describes the next attack scenarios against EAX-prime, a "standard security function" used by ANSI C12.22 as authenticated encryption: "attacks exploit the wrong tweaking method of CMAC (Cipher-based Message Authentication Code) in EAX," "plaintext Recovery Attacks," "distinguishing attack," and "forgery attack.".

3.3.13. IEC-61850/IEC-62351 (TCP: 102)

IEC-61850 is a popular protocol in the smart grid sector. The IEC-61850 standard was initially designed for substation automation but has expanded into other domains such as wind power plants, hydropower plants, microgrids and distribution automation domains [236]. The protocol was built to enable interoperability between vendors, allowing devices to define its intrinsic functionality and simplifying its communication. IEC-61850 unifies the different functions such as measurement, control, protection, and monitoring. End devices to IEC-61850 are intelligent electronic devices (IEDs), which are classified based on their function such as relay devices, voltage regulators, circuit breakers, and so on [237]. A typical IEC-61850 substation architecture includes two kinds of communication bus: Substation Bus (SB) and Process Bus (PB) in the substation, which connects all the IEDs. Both SB and PB mapped over Ethernet medium, however they may have different bandwidths, e.g., SB (10/100/1000 Mbps) and PB (0.1/1/10 Gbps). SB handles the requests/responses and general event substation messages. Generally, there is only one global SB. However, PB interfaces IEDs to traditional devices such as merge units. There can be more than one PB inside the substation [238]. In a very simple way, an IED is a physical device (PD) that hosts all logical

devices (LD). PD connects to the network via network address. LD is a collection of LN (Logical Nodes) (e.g., a breaker). LN is the core that constitutes a single functional unit to power automation environments. LNs are stand-alone devices and can be set up flexibly on any IED. LN contains data (e.g., position (pos) and operation count (opcnt)) and objects [239]. IEC-61850 substation interactions can be grouped into following three categories: data monitoring/reporting, data gathering/setting, and event logging. To realize the above-mentioned interactions IEC-61850 standard has defined a fairly complicated communication structure that defined five types of communication profiles: Generic Object-Oriented Substation Events (GOOSE), Sample Value (SMV), Simple Network Time Protocol (SNTP), Abstract Communication Services Interface (ACSI), and Generic Substation State Events (GSSE) [238]. IEC-61850 includes several basic security features, although these differ according to the part of the protocol being inspected. Thus, no security options exist in this IEC-61850 version 1 and 2 [240]. The use of a secured tunneling protocol such as TLS (with client certificates) or VPN guidelines can be found in the IEC-62351 standards. The IEC-62351-4 enables security over MMS and the IEC-62351-6 defines over GOOSE and SNTP. IEC-62351-6 recommends that the use of VLANs is required for GOOSE [240]. IEC-62351-7 emphasizes on Network and System Management (NSM) of the "information infrastructure," which defines data objects for the power system operational environment, which contains the information needed to manage the "information infrastructure" as reliably as the power system infrastructure is managed. The NSM data objects can be assigned to IEC-61850 [226]. However, Reference [241] identified some weaknesses in the IEC-62351standard, which were documented as penetration tests to perform the protocol. Within these weaknesses with an attack pattern associated are found Replay After "stNum" Reset in the GOOSE Protocol, Cross Receiver Replay in the Sampled Values Protocol and Attacks on the Simple Network Time Protocol [241].

3.3.14. IEC-60870-5-104/IEC-62351 (TCP: 2404)

IEC-60870-5 is one between the six parts of IEC-60870 standard, which sets mechanisms applied to tele-control systems especially SCADA systems in power system automation and electrical engineering. Part 5 describes the communication module used to send tele-control messages between two directly linked systems. Tele-control refers to the sending of monitoring data and requests for data collection to control power transmission grids. This part contains seven documents defining the standard tele-control, tele-protection, and related telecommunications for electrical power systems. IEC-60870-5-101 (IEC-101) and IEC-60870-5-104 (IEC-104) are the protocols that meet these standards. The IEC-104 protocol is an analogy to the IEC-101 protocol that adapts the functions defined by IEC 101

to a TCP/IP network [242]. The IEC-104 telegram structure is composed of three sub-layers (1) Application Protocol Control Information (APCI), (2) Application Service Data Unit (ASDU), and (3) Application Protocol Data Unit (APDU). Since the IEC-104 protocol transmits clear text messages without any authentication mechanism, it is the target of different attack patterns [243]. Clear text data transmission is a potential risk for "eavesdropping," "sniffing," and "tampering" for substation [243]. The lack of authentication of the commands questioning, remote control and remote tuning, allows potential exploiters to gain non-authorized access to SCADA systems, breaking the integrity and the availability, as well as releasing spoofing attacks, replay attacks and MITM attacks [243]. However, one way to provide a TLS-based connection to IEC-104 is through IEC 62351-5 (as with DNP3) [226]. In addition, IEC-104 is adopting IEC-62351 (as is the case with IEC-61850). A sample of this is that NSM data objects can be assigned to IEC-104 [226].

3.3.15. IEC-60870-6-503/IEC-62351 (TCP: 102)

The IEC-60870-6-503 (also known as TASE.2 or Inter Control Center Protocol (ICCP)), connects control centers (e.g., "Independent System Operators" (ISO), "Regional Transmission Operators" (RTO), and some generators [244]). ICCP is a complete modern client/server protocol, i.e., relies on TCP/IP [244]. ICCP defines a mechanism for critical data sharing among locations. ICCP enables both "real-time commands" and "historical monitoring" by incorporating an "object-oriented" layout in which devices are objects with related behaviors. Objects could be specific devices (e.g., "transformers" and "relays") or abstract data structures (e.g., transfer sets). By default, IEC-60870-6 ICCP/TASE.2 is not secure, since it is a protocol that transmits in plain text without any mechanisms to ensure confidentiality or integrity. However, it promotes IEC-62351-4 like secure TASE.2. Then, some strategies to securing ICCP like VPN are analyzed in Reference [244]. However, ICCP can include mechanisms to ensure communication such as 1,024-bit asymmetric key length implementations and multi-certified by link. Asymmetric key length implementations of 1,024 bits are broadly supported now and are usable without the obsolete hashes and cyphers. Certificates by link allow certificate expirations to overlap to permit certificate updates with minimized effect on data transmissions. Like DNP3 and IEC-60850, ICCP complies with IEC-62351-3, protecting against eavesdropping and replay attacks through TLS encryption, against human security risk in the environment through message authentication and against spoofing through security certificates (node authentication); while like IEC-60850, it complies with IEC-62351-4 for using MMS [226]. However, from Reference [227], it is established that ICCP vulnerabilities were demonstrated using attack patterns such as fuzzing.

We can conclude that all protocols, standards, and buses analyzed during the survey have the security issues as a common factor, i.e., vulnerabilities associated with either design, implementation, or operation. At this point, we are unable to recommend which protocol, standard, or bus is better or worse for a category (e.g., IT or OT) or IIoT classification (e.g., FAP, PAP, ISCP, BAP, PSAP, AMRP, SHAP, or VAP). For this reason, an in-depth study of the vulnerabilities, which affect these IIoT protocols, standard and buses is required. This study is carried out via methodological framework. The results returned once the framework is applied can be used as a parameter to evaluate the risk in assets. In that sense, the Common Vulnerability Scoring System (CVSS), is an essential component of our framework as an evaluation tool for industrial environments. Since the CVSS categorizes the vulnerabilities detected in a specific asset, based on the implementation and operation of the protocol, and not from the perspective of the design of the protocol itself, our framework focuses on the analysis of implementation and operational vulnerabilities of the 33 protocols analyzed during the survey.

4. CVSS AS EVALUATION TOOL FOR ICS SYSTEMS

CVSS is a scoring system that offers a standard and opened method to estimate the severity of a vulnerability. Its use is widespread, especially for IT environments, but when this framework is used to determine the severity of vulnerabilities that affect the industrial devices, different problems arise. To illustrate the problem mentioned next an example. If it is considered, on the one hand, a pacemaker programmer Medtronic 2090 Carelink, to which was associated the vulnerability CVE-2018-10596 and, on the other hand, a digital backbone (Schneider Electric ExoStruxure), on which the vulnerability CVE-2018-7797 has been found. Although the first vulnerability can be used to kill someone and the second used to execute a phishing attack, the CVSSv3 base categorizes the first with a severity of 7.1 and the second with a severity of 7.4. For this reason, the second stage of our manuscript establishes a methodology to describe a vulnerability for an industrial environment, additionally allows us to establish a comparison of the impact of suffering the same vulnerability in an industrial environment (OT) regarding an IT environment. To achieve this, we will first analyze the pillars of cybersecurity due to their high value for the CVSS, then we will briefly analyze the CVSSv3 with respect to CVSSv2, then introduce the improvements to CVSSv3 proposed by CVSSv3.1, and finally, we will review other proposals of methodologies that include strategies to improve CVSS for an industrial environment.

4.1. CYBERSECURITY PILLARS

As mentioned in section 2, the benefits of convergence between IT and OT bring associated risks. The first point at which risk manifests is the importance that each environment assigns to the three-cybersecurity pillars. The Table 2 illustrates the degree of importance of each pillar according to the environment: IT or OT. Therefore, for an IT environment, it is more important that information is not made accessible or distributed to non-authorized individuals, entities, or processes, whereas in OT environments, it is a requirement on-demand, timely, and reliable access to and use of information by an authorized user, i.e., available. However, beyond the fact that the pillars contribute to accentuate the differences between environments (IT and OT), these and other metrics are the base of the CVSS score, which is briefly analyzed in the following section.

Table 2: IT-OT cybersecurity pillars comparison.

Order	IT	OT
1	Confidentiality	Availability
2	Integrity	Integrity
3	Availability	Confidentiality

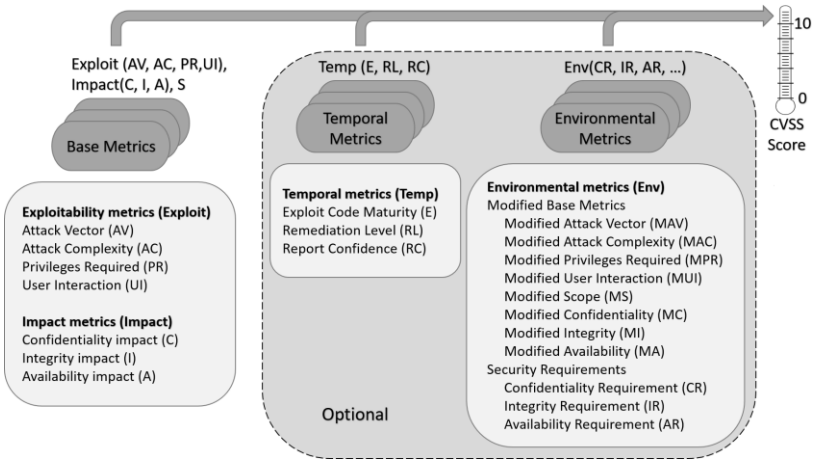

Figure 3: CVSS metrics and equations.

4.2. CVSS IN INDUSTRIAL ENVIRONMENTS

The CVSS scoring system allows us to standardize a vulnerability scoring methodology that is agnostic to any platform, so it is an open framework, which provides visibility to individual characteristics and the methodology used to obtain a score. The CVSS is composed of three metric groups: Base (BM), Temporary (TM), and Environmental (EM).

Table 3: CVSSv2, CVSSv3 and CVSSv3.1 comparison.

CVSSv2	CVSSv3	CVSSv3.1
Vulnerabilities are rated according to the overall impact on the platform.	Vulnerabilities score according to the impact on the impacted component.	Vulnerabilities score according to the impact on the impacted component.
There is no knowledge of instances where a vulnerability in one application impacted other applications on the same system.	The Scope metric allows to indicate if the vulnerability affected other components of the system.	The metric Scope, Vulnerable Component and Impacted Component concepts are reformulated to clarify them.
Access Vector can combine local system access and physical hardware attacks.	Local and Physical values are separated in the Attack Vector metric.	The descriptions of the values (Network, Adjacent, Local and Physical) of the Attack Vector (AV) metric are reformulated.
Access Complexity combined system configuration and user interaction.	Access Complexity has been separated in Attack Complexity and User Interaction.	The access complexity metric of the attack eliminates an ambiguity in its description.
Authentication metric leaves out many aspects of vulnerability.	The required privileges reflect the privileges necessary to affect the attack.	When scoring impact, to consider the privileges the attacker has prior to exploiting a vulnerability and compare those to the privileges they have after exploitation.
Impact metrics reflected percentage of impact caused to a vulnerable application.	Impact metric values reflect the degree of impact, and are renamed to None, Low and High.	It is specified that only the increase in access, privileges gained, or other negative outcome as a result of successful exploitation are considered to score the impact metrics of a vulnerability.
The environmental metrics are not considered.	The environmental metric allows to understand how vulnerability is reflected in an application environment.	Change to Modified Impact Sub-formula in Environmental Metric Group.

According to CVSSv3.1 specification (released on June 2019), the BM reflects the severity of a vulnerability according to its intrinsic characteristics, which are constant over time and have a reasonable impact in the worst case in the different environments deployed. The BM includes the set of metrics the exploitability metrics (EM) and the impact metrics (IM), where the EM measure the technical means and facility with which vulnerability is exploited and the IM involves the three-cybersecurity pillars (Table 2). The TM adjusts the BM severity of a vulnerability based on factors that change over time, such as the availability of exploitation code. The EM adjusts BM and TM severity to a specific environment. The Figure 3 contains all the metrics that make up BM, TM, and EM, which by default are represented by a string to which a numerical value is associated and then, allow to compute each metric score from of the equations included in the specifications. Since it is typical that only the BM are published, since these do not change over time and are common to all environments, it is therefore common to face two challenges when working with the CVSS. On the one hand,

the calculation of TM and EM, and on the other hand, the fact that have been determined only for CVSSv2 and not for CVSSv3 or for CVSSv3.1. The Table 3 highlights the main improvements that CVSSv3 introduces over CVSSv2. In addition, it should be noted that changes between CVSSv3.0 and CVSSv3.1 clarify and improve the standard without introducing new metrics or metric values, and without making major changes to existing equations, hence the need to adopt it.

4.3. EFFORTS TO ADAPT CVSS TO INDUSTRIAL ENVIRONMENT: RELATED WORK

To overcome the aforementioned problems presented by CVSS in section 4 to describe vulnerabilities in industrial environments, several experts in the sector are looking for alternatives among which are RSS [245], TEMSL [246], and IVSS [247]. RSS (Risk Scoring System) is an exclusive alternative for vulnerabilities in the health, aviation, and weapon sectors, which proposes to incorporate new factors, such as the duration of the attack or the chain of exploitation. IVSS (Industrial Vulnerability Scoring System) is a calculator and a description of the metrics and factors evaluated in it such as Base Severity Level (BS); Base Exploitation Level (BEX); accessibility, impact, and consequences; base score and IVSS final score, made up of all the previous values. However, IVSS is a project still in development. TEMSL (Threat, Exposure, Mission, Safety, and Loss) performs the calculation associated with the severity of vulnerability with decision trees. These trees qualify threats as evidence of exploitation of vulnerability; exposure as the extent of access (external or local network); mission as operation, service delivery, and data protection; physical security as injuries or deaths and losses as costs associated with exploitation of vulnerability beyond mission or physical security. However, as the associated footer shows, there is no formal (e.g., a white paper) or scientific reference associated with the project yet. As a summary of this section, we can conclude that there is a need to establish an index that can characterize the severity of a vulnerability in industrial environments, for that reason, numerous projects such as RSS, IVSS and TEMSL are making efforts to develop a model that meets the requirements. However, since there is no established methodology yet, in the next section, we will present our methodology through the VAF framework.

5. METHODOLOGY AND THE INFORMATION DATA SOURCE

Before proceeding to make our proposal, the advantage provided by working with CVSSv3.1, which is the basis of our methodology, should be emphasized, because as we have mentioned it leads to a risk analysis (sub-section 4.2). Therefore, through CVSSv3.1 our methodology solves two concrete problems: (1) the framework allows for risk analysis; (2) the framework

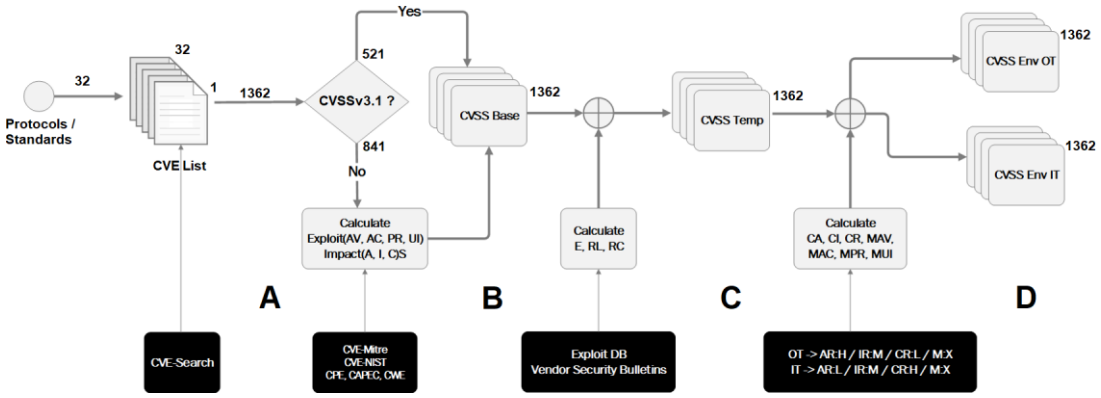

Figure 4: Methodology and information data sources.

is coupled to an industrial environment. To perform a risk analysis from CVSSv3.1, both temporal and environmental metrics must be calculated and then complemented by external factors such as exposure and threat.

To link the analysis to an industrial environment, the environmental metrics are contextualized for both an IT and an OT environment, and the behaviors that would have the same vulnerability if exploited in one or the other environment are analyzed. A methodology known as the Vulnerability Analysis Framework (VAF) was designed to enable the vulnerability analysis (Figure 4). VAF is powered by open-source data sources, which are described throughout this section. These data sources are classified into dictionaries, Open-Source Intelligence (OSINT) and integration tools.

1. CVE-Mitre (OSINT) is the acronym for "Common Vulnerabilities and Exposures" and is a set of items that includes an ID number (e.g., CVE-2017-2681), a descriptive statement, and a minimum of one published reference for well-known cybersecurity vulnerabilities [248]. Mitre Corporation supports CVE-Mitre.

2. CAPEC (dictionary) is the acronym for "Common Attack Pattern Enumeration and Classification." It is a complete dictionary and categorization of known attacks used by cybersecurity researchers, programmers, auditors (or cybersecurity testers) and academics to further enhance community awareness and improve defenses [249].

3. CWE (dictionary) is the acronym for "Common Weakness Enumeration," which is a set of common software security weaknesses created by the community. CWE is a standard used to describe the security weaknesses of software in architecture, design, or code. It is also a benchmark for software security tools and a guideline for identifying, mitigating, and preventing weaknesses [250].

4. CPE (dictionary) is the acronym for "Enumeration of Common Platforms." It is a structured naming framework and a uniform approach to describe and to identify several types of applications, OS, and hardware devices contained in an organization's data-processing assets [251].

5. EDB (OSINT) is the acronym for "Exploitation Database." It is a public exploits file compatible with CVE and the associated vulnerable software. EDB is supported by Offensive Security [252]. EDB has been developed for penetration testers and cybersecurity researchers [253]. The aim is to store the complete collection of exploits and POC collected via direct post, mailing lists. This collection is released as part of an open-source repository [253].

6. MB (Integration tool) is the acronym for "Microsoft Bulletin," which is a Microsoft Security Response Center (MSRC) initiative [254]. MSRC publishes monthly security bulletins, which cover security vulnerabilities in Microsoft software. These bulletins describe both the issue and the fix to the vulnerabilities as well as provide links to updates relevant to the affected software.

7. In addition to CVE-Mitre, CVE-NVD is classified as integration tool. The NVD is the U.S. "National Vulnerability Database." NVD is a standards-based vulnerability database managed by the Security Content Automation Protocol (SCAP) [255]. NVD includes a comprehensive report, consisting of the following elements: a brief description of the existing vulnerability; impact metrics, references to alerts, solutions, and tools; technical details including the type of weakness and known affected software configurations.

8. CVE-Search (integration tool) is a tool that incorporates lists of assets, dictionaries, and vulnerabilities, including those: CVE, CWE and CPE. These elements are inserted into a MongoDB database to simplify the searching and processing of CVE [256]. The objective of the tools is to search for vulnerabilities in a local database. In addition to the backend where the data at rest is, CVE-Search includes a user-friendly web interface for vulnerability searching and managing, a set of tools to access the system, and a web API interface.

9. CVE-Details (integration tool) provides a compliant to use web interface to CVE vulnerability data [257]. Allows access to vendor information, associated assets, firmware releases, OS and CVE inputs, and associated vulnerabilities. It enables us to see vendor, products, and version statistics. CVE-Details data is taken from CVE-NVD and various sources such as EDB exploits, vendor declarations, and supplemental data provided by vendors. Metasploit modules are issued as well as NVD-CVE data.

10. Vulnerability Analysis Framework (VAF) methodology starts with CVE List that contains all CVE numbers from origin to mid-2018 for each protocol, standard and bus (Figure 4). CVE List is an output of CVE-Search tool. Because of this first step, 1,363 vulnerabilities are collected, with quantities distributed very heterogeneously between protocols, standards, and buses (Figure 7). VAF then determines whether the vulnerability has the CVSSv3.1 calculated. CVSSv3.1 was calculated for 841 of the 1,363 vulnerabilities analyzed, assigning base metrics values, following analyst-scoring strategies based on both recovering and analyzing information from data sources such as CVE-Mitre, CVE-NIST, CPE, CAPEC, and CWE (Figure 4). With the base metrics standardized to CVSSv3.1, again scoring analyst strategies are followed to determine the submetrics E, RC, RL (Figure 3). Both EDB and vendors security bulletins (e.g., Microsoft) were the data sources (Figure 4). As a result, temporal metrics are normalized as well as base metrics. VAF uses the customizable nature of environmental metrics. Keeping the rest of the metrics constant, from Table 2, values have been assigned according to the importance of the cybersecurity pillar for the environment. In this way, it is determined whether the severity of a given vulnerability affects an IT environment more than an OT or vice versa. To customize the OT environment, a high value to AR, a medium value to IR and a low value to CR are assigned, while to customize the IT environment, a high value to CR, a medium value to IR and a low value to AR are assigned (Figure 4). As a result, the data analyzed in the following section expands to 1,363 custom vulnerabilities to OT environment and 1,363 custom vulnerabilities to IT environment (Figure 4).

5.1. RELATIONSHIP BETWEEN CVE, CWE AND CAPEC

It should be noted that there is interaction between some of the information sources, for instance, CVE, CWE, and CAPEC. In this section, we will mention the interaction between CVE, CWE, and CAPEC, which will be used in our research within the results analysis section. To understand the relationship between CVE, CWE, and CAPEC, first, we must understand the relationship between vulnerability, weakness, and exploit.

As shown in Figure 5, there are weaknesses discovered, characterized, exploitable and possibly with mitigations, which are grouped within the CWE. However, there are also weaknesses in assets and protocols that have not been characterized. If a weakness is exploited from an exploit, then it becomes in a vulnerability. The vulnerabilities can be reported, publicly known and exposed through the CVE. However, unreported, or undiscovered vulnerabilities may also exist. If previously unmitigated weaknesses are

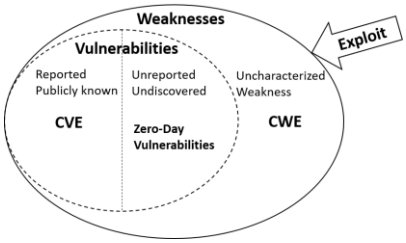

Figure 5: Vulnerability, weakness, and exploit.

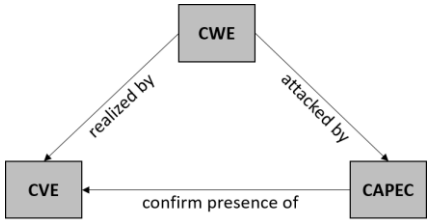

Figure 6: Relationship between CVE, CWE and CAPEC.

exploited with little or no warning, then they become in zero-day vulnerabilities. Since CAPEC describes the common attributes and techniques used by adversaries to exploit known weaknesses (e.g., SQL Injection, XSS, Session Fixation, Clickjacking), as Figure 6 illustrates, the relationship between CWE, CVE, and CAPEC is given, because a vulnerability (CVE) is the materialization of a weakness (CWE) through a known attack pattern (CAPEC).

6. ANALYSIS RESULTS

This section summarizes the results obtained through the application of VAF on each of the 32 protocols, standards, and buses commonly used in IIoT environments. The Figure 7 represents a distribution of documented vulnerabilities until the first half of 2018 (Q2, 2018) for 32 of the 33 protocols analyzed. In addition, it includes a division into categories based on the distribution (IT, IoT, and OT) of Table 1. Since 5G is a technology in the process of implementation and deployment, the following section presents the results found in the security field, because there is not any CVE yet. As Figure 7 illustrates, the distribution is highly heterogeneous.

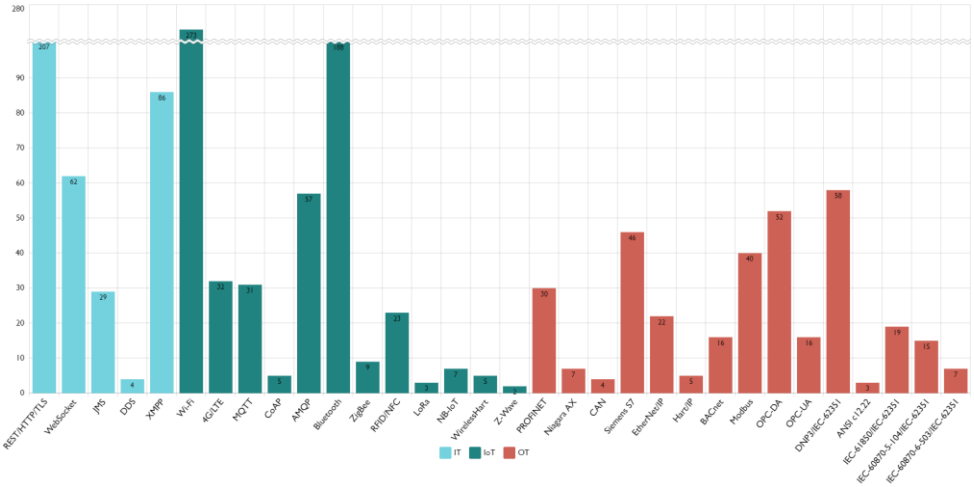

Figure 7: Distribution of vulnerabilities by protocol, standard and bus.

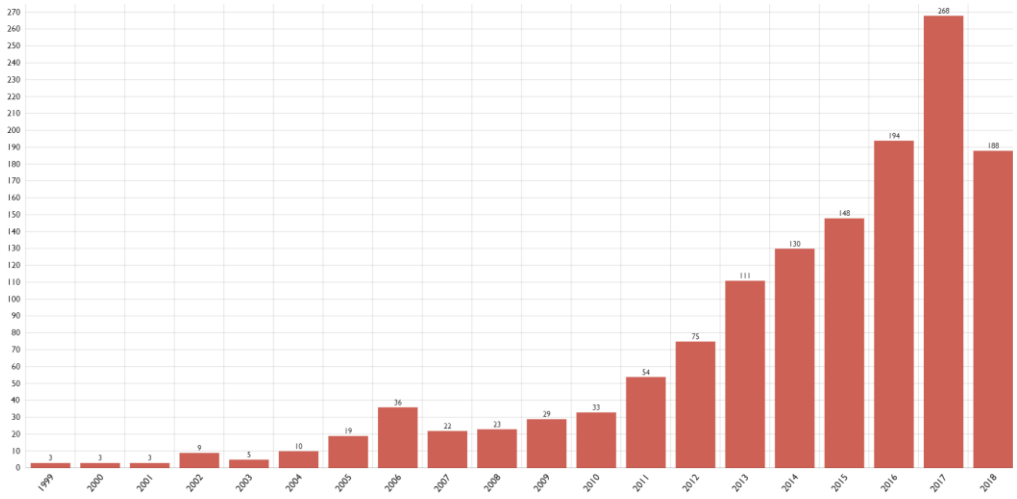

Figure 8: Distribution of vulnerabilities by year.

Therefore, we can find a set of protocols, standards, and buses such as Bluetooth, REST/HTTP/ TLS, and Wi-Fi where a large number of vulnerabilities are documented; others such as DDS, LoRa, and ANSI c12.22 with few identified vulnerabilities and another set, where WebSocket, AMQP, and MQTT are found, which maintains an intermediate number of documented vulnerabilities. We conclude by saying that the susceptibility of the protocol, standard, or bus depends to a high degree on its usability, i.e., if DDS were so popular and were on mobile devices, wearables, routers, among others, such as Bluetooth, then definitely the number of documented vulnerabilities would be greater. Therefore, the increased use of protocols through the integration proposed by IIoT will increase the interest of security researchers to discover new vulnerabilities in devices that implement these protocols, standards, and buses. However, Figure 8 provides a distribution by year of the 1,363 vulnerabilities studied. The contribution of each protocol per year allowed us to identify in the first place the age of protocols, standards, and buses, which entails security challenges, as well as, in the second place, to confirm the increase in the detection of vulnerabilities at present, highlighting the need to consider security from the design phase. First, OPC-DA reported vulnerabilities detected since 1999. Although in 2006 there was an increase in detection, the behavior was relatively stable until 2010. The disruptive year 2011marks the beginning of a consistent growth in vulnerability detection. It should be noted that our analysis was performed until the middle of 2018, so the value reflected in the graph is lower. To analyze each of the 1,363 vulnerabilities, the methodology proposed by the VAF framework has been applied. The main purpose of the VAF application on 32 protocols, standards, and buses is to perform a risk analysis. To this end, we have divided our analysis into four steps

that are derived throughout the application of VAF (5.2–5.6) and a classification proposed by NIST (5.6). Additionally, we have focused the first point of our analysis on the 5G technology, because currently there is no vulnerability of implementation documented with CVE, so in this case it has been impossible to apply VAF on this technology. Therefore, the analysis of results is divided into (1) recent attacks in 5G; (2) comparison between base and temporal metrics; (3) comparison between contextualized environmental metrics for both OT and IT environments; (4) impact of vulnerabilities on the pillars of cybersecurity: availability, confidentiality, and integrity; (5) impact of the cybersecurity pillars on IIoT categories; (6) attack patterns and weakness associated with vulnerabilities; (7) classification of vulnerabilities according to NIST Reference [4].

6.1. RECENT ATTACKS IN 5G

Currently, many countries have released or are close to releasing commercially 5G services, because the 3GPP group has developed the 5G standards where security procedures are included, as was mentioned previously in section 3.2.3 [258]. As mentioned in section 3.2.3, air security between mobile phones and mobile phone towers has been improved to overcome several threats, for example, through certain security measures fake base-station-type attacks (also known as IMSI or Stingray sensors) can be mitigated [258]. However, some of 4G's wireless features are reused in 5G. For instance, the 3GPP standards group has designed various capabilities in 4G and 5G specifications to support a wide range of applications such as smart homes, critical infrastructure, industrial processes, autonomous vehicles, and so on. This type of mechanism indicates to the network the type of device, i.e., a mobile device, a vehicle, an IoT device, so that it can receive specialized services and connectivity. A group of researchers through the works referenced in Reference [259] have found the following vulnerabilities:

1. A "protocol vulnerability" in specifications of 4G and 5GTS 33.410 [260] and TS 33.501 [261] allow a false base station to steal information about the device and mount identification attacks.
2. An "implementation vulnerability" in equipment of the cellular network operators that can be exploited during the "registration phase" of a device.
3. A "protocol vulnerability" in 1271 the first version of LTE NB-IoT that affects the "battery life" of low-power devices.

To each one of the vulnerabilities cited, the researchers associated the following attacks, respectively: mobile network mapping attack (MNmap) (active or passive), "bidding down"

via MITM, and "battery drain" via MITM. The attacks demonstrations were announced at a prestigious security conference [259]. The vulnerabilities and attack patterns mentioned received responsible disclosure and were notified to GSMA through its Coordinated Vulnerability Disclosure Program (CVD) [262]. In addition, the researchers notified 3GPP, responsible for the design of the 4G/5G security specifications and the mobile network operators concerned via CVD-2019-0018 [258].

6.2. COMPARISON BETWEEN BASE METRICS AND TEMPORAL METRICS

Based on the methodology proposed by VAF, following the risk analysis criteria of CVSSv3.1, the first step is to determine the base and temporal metrics. These steps correspond to stages B and C of Figure 4. From this point, an analysis is established between the behaviors of each protocol, standard and bus. Figure 9 summarizes the results obtained, which are divided into mapping of the vulnerabilities in Figure 7, distribution of the severity of the vulnerabilities for both the base metric (BM) and the temporal metric (TM), and determination of the difference between the average severity for the base metric and the temporal metric, which corresponds to the average BM and average TM index.

As shown in Figure 9 for each protocol, standard, or bus, the sum of the vulnerabilities distributed according to their severity coincides with Figure 7. For example, REST/HTTP/TLS contains 207 vulnerabilities analyzed. In addition, for each protocol, standard, or bus the vulnerabilities are divided according to their severity into critical, high, medium, and low for both base and temporal metrics. For example, of the 207 vulnerabilities analyzed for REST/HTTP/TLS for the base metrics, 35 are critical, 67 are high, 86 medium, and 19 low; while for the temporal metric 15 are critical, 61 are high, 102 medium, and 29 low. Finally, to characterize the behavior of the protocol, standard, or bus in a general way, the average value of the severity obtained is determined for both the base and the temporal metrics. This result is shown in the middle of Figure 7. For example, for REST/HTTP/TLS the average severity of the vulnerabilities for the base metric is 6.7, while the average severity for the temporal metric is 6.2, resulting in 0.5 as the difference between them. For each of the protocols, standards, and buses, Figure 9 shows that the severity of the base metric is greater than the severity of the temporal metrics. The number of critical vulnerabilities decreases practically for each one of the protocols independently of the category IT, IoT, and OT. For instance, for REST/HTTP/TLS (IT) it decreases from 35 to 15, AMQP (IoT) decreases from 8 to 1, and Modbus (OT) decreases from 12 to 5. The result is expected, because the temporal metric has the "vulnerability remediation level factor," which measures the vulnerability patching level and indicates if the vulnerability has not been patched, presents a

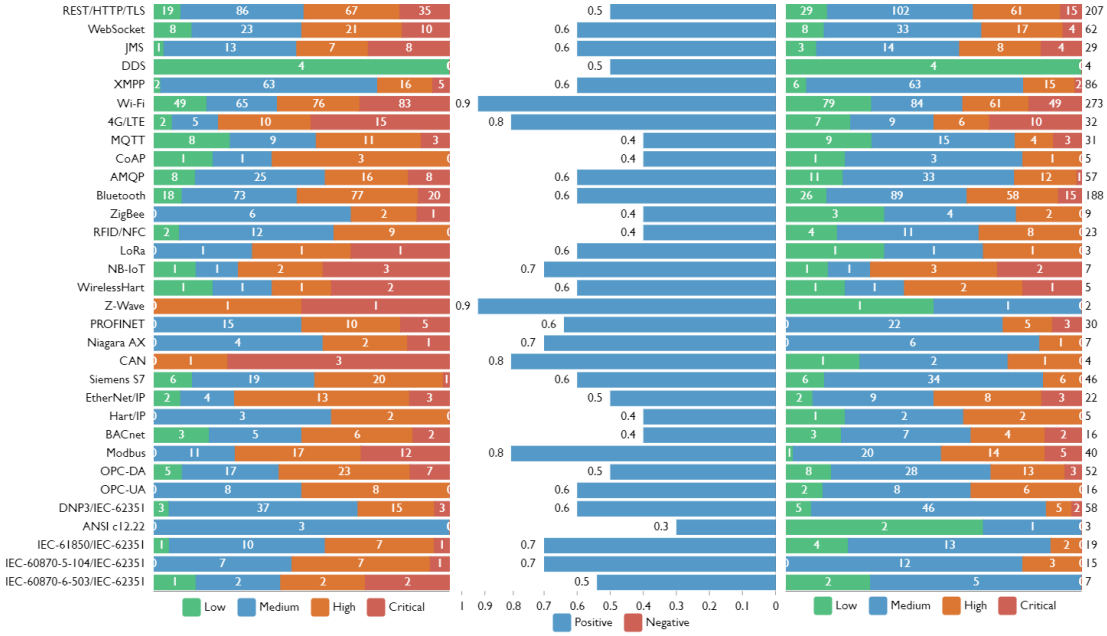

Figure 9: From left to right: (a) BM, (b) Avg (BM) - Avg (TM), and (c) TM.

temporary solution or hotfixes that offers a temporary solution until a patch or official update is issued. If the temporal metric is closer to the base metric, then it will show a lower level of patching. The ANSI C12.22 protocol is a case where severity does not decrease between base and time metrics. Although we could argue that the severity of the detected vulnerabilities is medium and that it presents a small number of detected vulnerabilities (only three vulnerabilities), the truth is that at the time of the study, there was a high risk that some of the vulnerabilities could be exploited in some of the devices where it was detected. To establish which of the protocols, standards, or buses analyzed have the best results in this comparison, we have decided to compare the mean severities for the base and temporal metrics. Therefore, the protocols, standards, or buses that present a higher index, i.e., the difference of the average base severity is much greater than the average temporal severity, would theoretically be more secure. It should be noted that, although the result provides us with a classification it is not determinant, because although a protocol has a high index it is the result of the average of severities, which does not mean that one of the severities that has been averaged is, for example, critical and permanently affects the availability of the asset. Therefore, it is recommended not to take the following result out of context. From our index, we can affirm that the least susceptible protocols would be Wi-Fi (0.9), Z-Wave (0.9), CAN (0.8), and Modbus (0.8); while the most susceptible would-be ANSI C12.22 (0.3), CoAP (0.4), MQTT (0.4), ZigBee (0.4), RFID/NFC (0.4), Hart/IP (0.4), and BACnet (0.4).

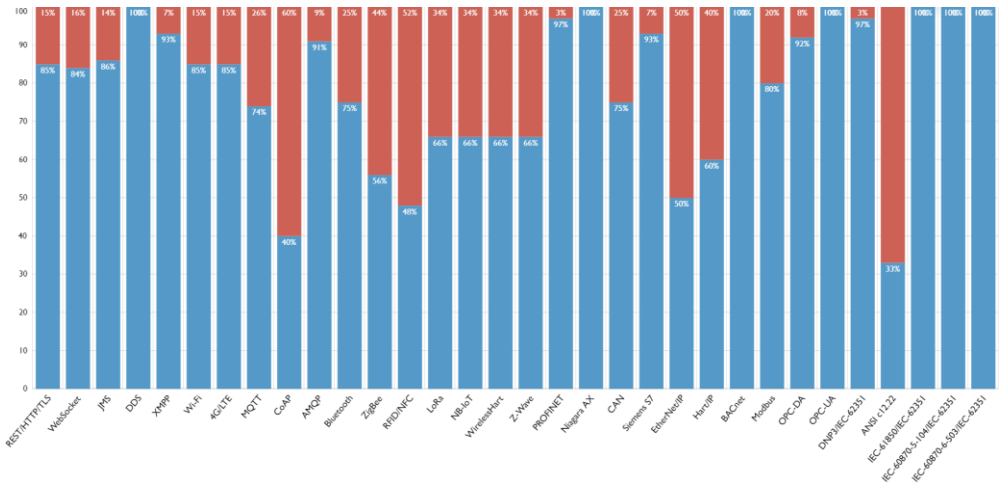

Figure 10: Remediation level.

Two paragraphs above mention that the remediation level (RL) of vulnerability is the main factor that determines the behavior that the temporal metric will have.

For this reason, Figure 10 is a representation of that factor, illustrating what percentage of the vulnerabilities of a specific protocols, standards, or buses has been patched or updated, presents a provisional solution, i.e., "workaround" or "hotfixes," against the percentage that does not present any type of solution. Each of these corresponding stages adjusts the temporal metric down, revealing the reduction in vulnerability level as the remediation is made final. Figure 10 indicates the patched, updated, workaround, or hotfixes vulnerability percentages in blue color. The protocols ANSI C12.22, CoAP, ZigBee, RFID/NFC, EtherNet/IP, Hart/IP, and BACnet, which according to Figure 9, present the worst values of our index, also present the worst remediation levels (RL) in Figure 10. In addition, Wi-Fi, 4G/LTE, Z-Wave, CAN, and Modbus have high remediation level (RL). For instance, the Modbus protocol shows that 80% of the detected vulnerabilities have patch, update, workaround, or hotfixes. The DDS protocol is a particular case with four vulnerabilities with low base metric severity (Figure 9), therefore once applied the remediation level (RL), which is 100%, as shown in Figure 10, the severity of the vulnerabilities are kept at a low level; hence, the severity index is higher than that of the previously mentioned protocols as less secure. All the examples analyzed above confirm the robustness of the proposed analysis.

6.3. COMPARISON BETWEEN ENVIRONMENTAL METRICS CONTEXTUALIZED TO OT AND IT ENVIRONMENT

Once the second stage of the risk analysis has been completed, where emphasis has been placed on base metrics, temporal metrics, and comparisons between them, this new phase,

which corresponds to stage D in Figure 4, VAF uses environmental metrics as a risk analysis mechanism. As mentioned above, organizations must specify environmental metrics, because they are the most appropriate to measure the potential impact of a vulnerability within their own computational environment. Therefore, Environmental Metrics adjust the Base (BM) and Temporal (TM) severities to a specific environment. Figure 3 establishes the set of submetrics that takes environmental metrics into account. The strategy used in this research will be to keep the values of the submetrics MAV, MAC, MPR, MUI, MS, MC, MI, and MA unchanged and to customize the values of the "security requirements" (CR, IR, and AR) according to the order of importance given in Table 2 to the cybersecurity pillars. This strategy aims to customize the CVSS score according to the relevance of the environment. From our results, we will be able to determine if an asset presents a certain risk of being compromised and this vulnerability is exploited in which environment (IT or OT) the impact will be greater. Figure 11 shows the results of applying stage D of Figure 4 to VAF. These results are divided into distribution of the severity of environmental metrics in both OT and IT environments and determination of the difference between the average severity for operational environmental metric (EM_{OT}) and information environmental metric (EM_{IT}), which corresponds to the average EM_{OT} index and average EM_{IT}. It should be noted that as with the results of Figure 9 for each protocol, standard, or bus, the sum of the vulnerabilities distributed according to their severity coincides with Figure 7. Figure 11 shows that the impact of applying compensation for OT environments is greater than applying compensation for IT environments in most protocols, standards, and buses. This behavior indicates that if an asset has a vulnerability and it is exploited, then the impact will be greater in an OT environment than in an IT environment. For example, 4G/LTE has 19 vulnerabilities with critical severity for EM_{OT} and 7 vulnerabilities with critical severity for EM_{IT}. Additionally, Bluetooth has 38 vulnerabilities with critical severity for EM_{OT} and 15 vulnerabilities with critical severity for EM_{IT}. The most representative case is that of Wi-Fi, which has 90 critical vulnerabilities for EM_{OT}, while 33 vulnerabilities with critical severity for EM_{IT}, which represents the most significant difference between all the protocols, standards, and buses analyzed. Similar results are achieved for XMPP, CoAP, AMQP, MQTT, ZigBee, RFID/NFC, LoRa, NB-IoT, WirelessHart, Z-Wave, PROFINET, Niagara AX, CAN, Siemens S7, EtherNet/IP, Hart/IP, BACnet, Modbus, OPC-DA, OPC-UA, DNP3, ANSI c12.22, IEC-61850, IEC-60870-5-104, and IEC-60870-6-503. The DDS protocol is a case to consider, since it has no vulnerabilities with critical severity, the reason why the analysis of the behaviors once applied the requirements of OT and IT, must extend to vulnerabilities of high and medium severity. Despite this, it is observed that severity is greater when OT requirements are applied than when IT is applied. In addition to DDS, the WebSocket protocol has a

different behavior. As it can be seen in Figure 11, the number of vulnerabilities with critical severity is the same (four) for both EM_{OT} and EM_{IT}. Therefore, the analysis should be extended to vulnerabilities with high severity, where if the number for EM_{OT} is greater than for EM_{IT}. However, there are two protocols, which do not follow this pattern: REST/HTTP/TLS and JMS. As shown in Figure 11, the number of vulnerabilities with critical severity is greater for EM_{IT} than for EM_{OT}. The explanation for this result comes from the fact that these protocols are mostly used in IT environments and that they are recently finding applicability in IoT environments. Like the analysis applied in the previous section to establish which of the protocols, standards, or buses analyzed have the best results, we have decided to compare the average severities for EM_{OT} and EM_{IT}. Therefore, the protocols, standards, or buses that have a higher index, i.e., the difference in average EM_{OT} severity is much greater than the average EM_{IT} severity, as a vulnerability is exploited it will have a greater impact on an OT environment than on an IT environment. As a conclusion of this phase, we can say that the implication of this study is that if the same vulnerability were detected in both an industrial and an information environment, the consequence of the exploitation of this vulnerability would be greater in the industrial environment. This result guarantees the adaptation of our framework to an industrial environment.

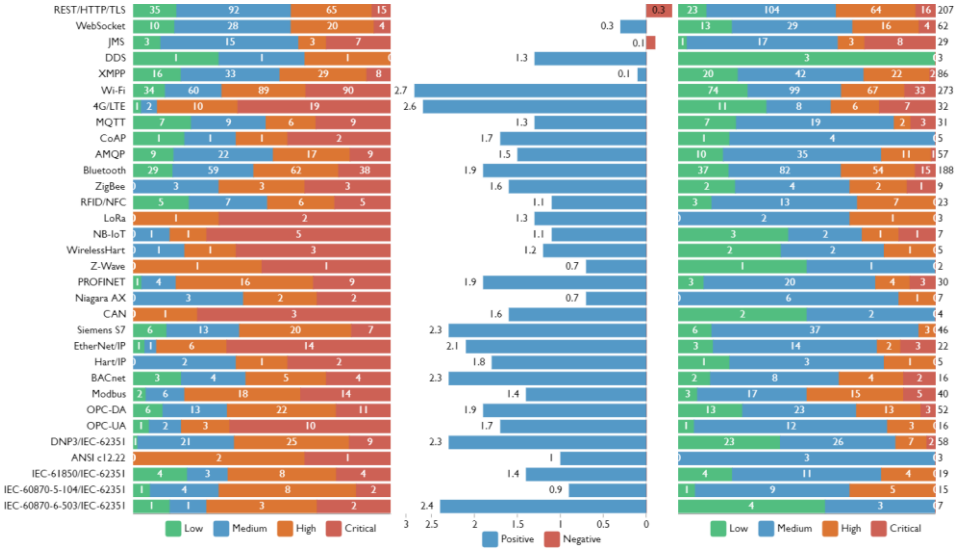

Figure 11: From left to right: (a) EM_{OT}; (b) Avg (EM_{OT}) - Avg (EM_{IT}) and (c) EM_{IT}.

6.4. IMPACT OF THE VULNERABILITIES ON SECURITY PILLARS

In the two previous sections, we have given a measure that describes how secure a protocol is based on the patches and updates it presents, as well as a measure of the impact of a

vulnerability on an OT or IT environment. To this end, we have adopted the methodology of the VAF framework. Since our framework is based on CVSSv3.1 and this indicates that risk analysis goes beyond the measurement of base, temporal and environmental metrics, the following analysis focuses on how the cybersecurity pillars of Table 2 have been impacted according to each protocol, standard, or bus. Exploring the impact of vulnerabilities on the three pillars of cybersecurity is one of VAF's aims. Table 4 provides the number of times availability (A), integrity (I), and confidentiality (C) have been fully affected with respect to the number of vulnerabilities (NoV) of the protocol, standard, or bus. This means, for example, that for REST/HTTP/TLS, of the 207 vulnerabilities examined, 77 of them have fully affected the availability, 76 vulnerabilities have totally affected the integrity, and 85 vulnerabilities have totally affected the confidentiality. It should be noted that a vulnerability could affect more than one pillar at a time. Analyzing the results provided by Table 4 shows that traditional industrial protocols (e.g., DNP3, IEC-61850, IEC-60870-5-104, and IEC-60870-6-503), IoT protocols (e.g., Wi-Fi, 4G/LTE, Bluetooth, ZigBee, and RFID/NFC), including IoT messaging protocols (e.g., MQTTT, CoAP, and AMQP) have availability as the most affected parameter. From column "A (%)," it is also possible to see how other OT protocols such as PROFINET, OPC-DA, OPC-UA, and DNP3 have the worst impact on availability. However, other protocols and standards, such as REST/HTTP/TLS, WebSocket, JMS, and DDS, present confidentiality as the most affected parameter. These protocols are of IT nature, i.e., recently have a significant utility in IIoT environments. Finally, the XMPP protocol should be mentioned, which, although it has a similar origin to the protocols, the most affected pillar is integrity. Table 5 provides an overview of the behavior of each of the categories associated with our protocols, standards, and buses in Table 1, i.e., IT, IoT and OT. Table 5 summarizes the behavior of the cybersecurity pillars for each of these paradigms (IT, IoT, and OT), considering that it has only been represented when the pillar has been totally compromised. The results allow to distinguish as for the protocols, standards, or buses associated to the IT environment the order of impact is confidentiality, integrity, and availability, while for both IoT and OT paradigms the order of impact is availability, confidentiality, and integrity.

6.5. IMPACT OF THE SECURITY PILLARS ON IIOT CATEGORIES

The previous section provided an overview of the behavior of the pillars of cybersecurity (confidentiality, availability, and integrity) on each of the protocols, allowing the protocols to be grouped according to the paradigm IT, IoT, and OT. In this section, we would like to move one step further in the risk analysis evaluation provided by our research. To this end, we will analyze the relationship that exists between the IIoT categories defined in Table 1

and the cybersecurity pillars. Contextualizing, again only the values of confidentiality, integrity and availability that have been totally compromised will be represented. Table 6 defines the

Table 4: Vulnerability impact on security pillars.

Protocol	NoV	A	I	C	A (%)	I (%)	C (%)
REST/HTTP/TLS	207	77	76	91	37	37	44
WebSocket	62	23	16	31	47	26	50
JMS	29	15	13	16	52	45	55
DDS	4	0	1	3	0	25	75
XMPP	86	19	31	27	22	36	31
Wi-Fi	273	135	75	87	49	27	32
4G/LTE	32	16	9	10	50	28	31
MQTT	31	11	8	4	35	26	13
CoAP	5	3	0	0	60	0	0
AMQP	57	22	13	13	38	23	23
Bluetooth	188	99	95	93	53	51	44
ZigBee	9	4	3	1	44	33	11
RFID/NFC	23	9	8	9	39	35	39
LoRa	3	2	1	0	67	33	0
NB-IoT	7	4	2	1	57	29	14
WirelessHart	5	4	2	0	80	40	0
Z-Wave	2	1	0	1	50	0	50
PROFINET	30	24	7	6	80	23	20
Niagara AX	7	4	3	4	43	43	57
CAN	4	3	2	3	75	50	75
Siemens S7	46	29	3	4	63	7	9
EtherNet/IP	22	17	5	4	77	23	18
Hart/IP	5	2	1	1	40	20	20
BACnet	16	5	3	5	31	19	31
Modbus	40	27	20	21	68	50	53
OPC-DA	52	33	17	19	63	33	37
OPC-UA	16	10	3	5	63	19	31
DNP3	58	42	9	8	72	16	14
ANSI C12.22	3	3	0	0	100	0	0
IEC-61850	19	9	4	4	47	21	21
IEC-60870-5-104	15	9	5	4	60	33	27
IEC-60870-6-503	7	4	0	0	57	0	0

Table 5: Summary of the vulnerability impact on security pillars.

Category	NoV	A	I	C
IT	388	134	137	168
IoT	635	310	216	219
OT	340	221	82	88

number of vulnerabilities (NoV), as well as the pillars of cybersecurity compromised for each of the IIoT categories defined. The number of vulnerabilities that affect each category is a function of the number of protocols, standards, or buses that are associated with it as well as the vulnerabilities that have been analyzed for each of the protocols, standards, or buses. Due to the wide application context, as well as the growing number of protocols that implement this category, Smart Home Automation Protocol (SHAP) contains the largest number of associated vulnerabilities. For SHAP it should be noted that although availability is the most affected pillar, confidentiality is the second most affected, which is manifested, for example, through the high risk of leakage of personal information by users who use the Smart Home. Although all IIoT classification are relevant, particularly Factory Automation Protocols (FAP) and Process Automation Protocols (PAP) present availability as the essential pillar, which for both categories in Table 6, is the most impacted pillar. Finally, it is important to highlight that the IIoT categories where the least difference between the numbers of compromised pillars exists is in "Vehicle Automation Protocols" (VAP), where confidentiality, integrity, and availability are around 100 times compromised.

Table 6: Impact of cybersecurity pillars on IIoT categories.

IIoT Classification	NoV	A	I	C
FAP	223	145	45	42
PAP	739	389	246	253
ICSP	140	86	49	55
BAP	579	299	211	225
PSAP	243	117	73	68
AMRP	64	41	26	24
SHAP	925	401	323	355
VAP	196	102	98	99

6.6. ATTACK PATTERNS AND WEAKNESS ASSOCIATED TO THE VULNERABILITIES

As mentioned in section 5.1, there is a close relationship between weakness (CWE), vulnerability (CVE), and attack pattern (CAPEC). In this section, we take advantage of that relationship to add a new layer to the risk analysis. A criterion mentioned as important within the risk analysis as a complement to the CVSSv3.1 analysis are the weaknesses and patterns of attacks, so the VAF allows a relationship between vulnerability, weakness, and pattern of attack. The methodology applied through VAF 1455 has been extracted for each of the 1,363 vulnerabilities, analyzing the weakness or weaknesses exploited and associating them with the corresponding attack pattern. Table 7 associates to each protocol, standard, or bus, the weakness exploited more times, as well as the attack pattern used more times. However, it is not necessary to have a match between the weaknesses and the attack patterns shown

in Table 7, because as mentioned, a weakness can have several attack patterns associated with it and a pattern of attack with several vulnerabilities.

For instance, the CAPEC-125: Flooding has two associated weaknesses; CWE-404: Improper Resource Shutdown or Release; and CWE-770: Allocation of Resources without Limits or Throttling. Since the flooding attack is the most used, it is detailed below. The flooding attack allows an adversary to consume the resources of a target due to the high amount of targeted interactions. This attack commonly exposes a weakness in rate or flow limitation. If performed correctly, then this attack impedes the access of legitimate users to the service and may result in the target being locked out. The number of requests the attacker makes in a certain period is the main factor in a flood attack. The use of the flood attack pattern is very common. For instance, EtherNET/IP allows that an attacker can send malformed packet CIP to Port 44818/TCP, Modbus allows an attack pattern by sending specifically designed (crafted) packages to port 502/TCP, and MQTT allows to send crafted CONNECT packets. For all three protocols, the flooding category used is TCP flood. An illustration of the connection between vulnerability, weakness, and attack pattern is demonstrated through AMQP, which it presents input validation errors (e.g., does not properly restrict incoming client connections). This weakness allows flooding attack (TCP Flood) and the possibility of DoS (see CVE-2012-2145). Another point to consider is that for wireless IoT technologies such as 4G/LTE, LoRa, NB-IoT, and WirelessHart, the most common attack pattern is the jamming. In this type of attack, an attacker would use noise or radio signals to disrupt communications. By intentionally overwhelming system resources with illegitimate traffic, legitimate traffic from authorized users is denied service. Two exceptions need to be mentioned: BACnet and XMPP. BACnet is a protocol used in building automation, and nevertheless it presents an attack pattern that is mostly used in web technologies.

This is because authentication vulnerabilities and cross-site request forgery (CSRF) were identified on KMC Controls' BACnet Conquest routers via their web interface. XMPP has a social-engineering-like attack pattern. Although XMPP is now linked to IIoT environments, it is a technology associated also with instant messaging. Tools like Pidgin are based on XMPP protocol. In this case, a Pidgin plugin called "Message Carbon" located in several XMPP clients enables a remote attacker to masquerade as either user, even contacts. Another exception occurs with Z-Wave where the weakness and attack pattern are not of a wireless nature and this is because the vulnerabilities, for example, are found in a device that implements the protocol and does not implement HTTP Public Key Pinning (HPKP). This weakness, allows to obtain the commands that are transmitted to the controller using a fake certificate in a

proxy server, enabling to control each node of the HUB, reaching to get the Z-Wave network key.

Table 7: Weakness and Attack Pattern Associated to Each Protocol.

Protocol	Weakness	Attack Pattern
REST/HTTP/TLS	Information exposure	API manipulation
WebSocket	Improper restriction[1]	TCP Flood
JMS	Information exposure	Object injection
DDS	Resource management errors	Flooding
XMPP	Improper input validation	Social Engineering
Wi-Fi	Improper input validation	Traffic injection
4G/LTE	Improper input validation	Cellular Jamming
MQTT	Improper Input Validation	TCP Flood
CoAP	NULL point dereference	Traffic injection
AMQP	Improper input validation	TCP Flood
Bluetooth	Information exposure	Flooding
ZigBee	Improper access control	Flooding
RFID/NFC	Information exposure	Flooding
LoRa	Improper input validation	Jamming
NB-IoT	Improper input validation	Cellular Jamming
WirelessHart	Improper input validation	Jamming
Z-Wave	Improper Certificate Validation	Malicious Root Certificate
PROFINET	Improper input validation	Flooding
Niagara AX	Improper authentication	Authentication Abuse
CAN	Improper restriction[1]	Code Injection
Siemens S7	Resource management errors	Flooding
EtherNet/IP	Improper restriction[1]	TCP Flood
Hart/IP	Improper restriction[1]	Overflow Buffers
BACnet	Improper restriction[1]	Cross site request forgery
Modbus	Information exposure	TCP Flood
OPC-DA	Improper restriction[1]	Flooding
OPC-UA	Improper input validation	TCP Flood
DNP3	Improper input validation	Input Data Manipulation
ANSI C12.22	Improper input validation	Target Programs[2]
IEC-61850	Improper access control	Flooding
IEC-60870-5-104	Improper input validation	Input Data Manipulation
IEC-60870-6-503	Improper restriction[1]	TCP Flood

[1] Improper Restriction of Operations within the Bounds of a Memory Buffer
[2] Target Programs with Elevated Privileges

6.7. VULNERABILITIES CLASSIFICATION

As we mentioned in Section 2, the NIST 800-82r2 Reference [4] proposes a general vulnerabilities classification into: policy and procedural vulnerabilities, architecture and design vulnerabilities, configuration and maintenance vulnerabilities, physical vulnerabilities, software development vulnerabilities, and communication and network configuration vulnerabilities. In addition, each category includes a group of vulnerabilities associated, i.e., a specific

vulnerabilities classification. Table 8 summarizes both the most exploited general vulnerability and the most exploited specific vulnerability for each of the 32 protocols, standards, and buses analyzed. The first conclusion reached after analyzing Table 8 is that there are a set of vulnerabilities that present similar behaviors as those associated with wireless IoT protocols, standards, and buses of the IoT category such as Lora and WirelessHart. In general, the most common vulnerability they face is the physical type. From a more specific point of view, these technologies are mostly affected by radio frequency vulnerabilities, since the hardware used for control systems is vulnerable to radio frequency. The impact can vary from temporal interruption of command and control to continuous failure to circuit boards. However, outside that group are protocols such as Bluetooth, RFID/NFC, Wi-Fi, and ZigBee. The first three (Bluetooth, RFID/NFC, and Wi-Fi) present vulnerabilities of the type of inappropriate information protection among "wireless clients" and "access points," which are not protected. The fourth (ZigBee) presents mostly vulnerabilities of the type of inadequate access controls applied, which belongs to the category of configuration and maintenance vulnerabilities. Another group of protocols, standards, and buses with vulnerabilities of a similar nature are those belonging to the OT category, i.e., the classic ICS. In protocols and standards such as Siemens S7, EtherNet/IP, Hart/IP, Modbus, OPC-DA, DNP3, ANSI C12.22, IEC-61850, IEC-60870-5-104, and IEC-60870-6-503, the most common type of vulnerabilities reported are architecture and design. There is also similarity in the type of specific vulnerabilities for most of these protocols and standards, with an inadequate incorporation of security inappropriate into architecture and design, which is justified by the fact that most have migrated to TCP/IP environments and therefore lack default security mechanisms, which is the reason they lack identification, authorization, and authentication mechanisms, i.e., access control, as well as mechanisms that guarantee the configuration and integrity of the system. Another group is made up of PROFINET, Niagara AX, CAN, and OPC-UA, with a general classification of software-development-type vulnerabilities and a specific classification of the type of inadequate data validation. In addition, there is an important group formed by protocols and standards IT nature, such as REST/HTTP/TLS, WebSocket, JMS, and so on, that present as major vulnerabilities the improper input validation, which is characterized because of the IIoT software does not validate user entries or received data correctly to guarantee their authenticity. Therefore, invalid data can lead to several vulnerabilities, among which are cross-site scripting, path traversals, command injections, and buffer overflows.

Table 8: Vulnerabilities classification according to the classification proposed by [4].

Protocol	Cat.	General Vulnerability Classification	Specific Vulnerability Classification
REST/HTTP/TLS	IT	Software Development Vulnerabilities	Improper Data Validation
WebSocket		Software Development Vulnerabilities	Improper Data Validation
JMS		Software Development Vulnerabilities	Improper Data Validation
DDS		Software Development Vulnerabilities	Improper Data Validation
XMPP		Software Development Vulnerabilities	Improper Data Validation
Wi-Fi	IoT	Communication and Network Configuration	Inadequate data protection between wireless clients and access points
4G/LTE		Communication and Network Configuration	Inadequate data protection between wireless clients and access points
MQTT		Software Development Vulnerabilities	Improper Data Validation
CoAP		Software Development Vulnerabilities	Improper Data Validation
AMQP		Software Development Vulnerabilities	Improper Data Validation
Bluetooth		Communication and Network Configuration	Inadequate data protection between wireless clients and access points
ZigBee		Configuration and Maintenance Vulnerabilities	Inadequate access controls applied
RFID/NFC		Communication and Network Configuration	Inadequate data protection between wireless clients and access points
LoRa		Physical Vulnerabilities	Radio frequency, electromagnetic pulse (EMP), static discharge, brownouts, and voltage spikes
NB-IoT		Communication and Network Configuration	Inadequate data protection between wireless clients and access points
WirelessHart		Physical Vulnerabilities	Radio frequency, electromagnetic pulse (EMP), static discharge, brownouts, and voltage spikes
Z-Wave		Architecture and Design Vulnerabilities	Inadequate incorporation of security into architecture and design
PROFINET	OT	Software Development Vulnerabilities	Improper Data Validation
Niagara AX		Software Development Vulnerabilities	Improper Data Validation
CAN		Software Development Vulnerabilities	Improper Data Validation
Siemens S7		Architecture and Design Vulnerabilities	Inadequate incorporation of security into architecture and design
EtherNet/IP		Architecture and Design Vulnerabilities	Inadequate incorporation of security into architecture and design
Hart/IP		Architecture and Design Vulnerabilities	Inadequate incorporation of security into architecture and design
BACnet		Software Development Vulnerabilities	Improper Data Validation
Modbus		Architecture and Design Vulnerabilities	Inadequate incorporation of security into architecture and design
OPC-DA		Architecture and Design Vulnerabilities	Inadequate incorporation of security into architecture and design
OPC-UA		Software Development Vulnerabilities	Improper Data Validation
DNP3		Architecture and Design Vulnerabilities	Inadequate incorporation of security into architecture and design
ANSI C12.22		Architecture and Design Vulnerabilities	Inadequate incorporation of security into architecture and design
IEC-61850		Architecture and Design Vulnerabilities	Inadequate incorporation of security into architecture and design
IEC-60870-5-104		Architecture and Design Vulnerabilities	Inadequate incorporation of security into architecture and design
IEC-60870-6-503		Architecture and Design Vulnerabilities	Inadequate incorporation of security into architecture and design

7. CONCLUSIONS

The IIoT architecture is complex, largely due to the convergence of protocols, standards, and buses. For this reason, a comprehensive survey of the 33 most useful protocols, standards, and buses in the IIoT environment is performed based on characteristics such as architecture, topology, messages/data, and security. From this analysis, we detect the existence of security problems in each of the protocols, standards, and buses. Therefore, an extensive assessment is necessary to measure the risk of existing documented vulnerabilities. Since the CVSS offers a way to collect the main properties of a vulnerability and generate a numbered score that indicates its severity, it is presented as the fundamental bedrock of our approach. At this point, we find two situations, on the one hand, to carry out the risk analysis it is necessary to complement the CVSSv3.1 with other elements and, on the other hand, CVSS presents problems to characterize the industrial environments. For this reason, we present the VAF methodological framework. For this purpose, an exhaustive compilation of OSINT tools such as CVE-Mitre, dictionaries such as CPE, and integration tools such as CVE-Search is performed. These data sources, our experience as analysts, and the practical context provided by CVSSv3.1 are the core of the Vulnerability Analysis Framework (VAF). This methodological framework enabled the analysis of 1,363 vulnerabilities from 33 protocols, standards, and buses. From the VAF, we divide our analysis into seven steps: (1) Vulnerability search (CVE) for the IT/OT/IoT standards, protocols and buses; (2) comparison between base and temporal metrics; (3) comparison between contextualized environmental metrics for both OT and IT environments; (4) impact of vulnerabilities on the pillars of cybersecurity: availability, confidentiality and integrity; (5) impact of the cybersecurity pillars on IIoT categories; (6) attack patterns and weakness associated with vulnerabilities; (7) classification of vulnerabilities according to NIST reference. It should be noted that was added a step only for 5G due to the novelty of the technology does not present active with CVE, but only a CVD confirmed by 3GPP: recent attacks in 5G. Each of these steps constitutes a phase of our risk analysis. The comparison between base and temporal metrics enabled to establish how the level vulnerability severity could be reduced through the application of security updates and at the same time, it grows through the deployment of exploits. In this sense, the impact of the amount of fixed vulnerabilities over Temporal Metrics were examined considering the existing relationship with the remediation level. The comparison between environmental metrics revealed that if an asset uses one or a combination of the 33 protocols, standards, and buses analyzed, when suffering one of the 1,363 vulnerabilities analyzed, the severity will be higher if the asset is located as part of an OT environment (e.g., Control and Operations Domain for IIoT functional domains) than of an IT environment

(e.g., Information and Applications Domain for IIoT functional domains). The impact on three security pillars (availability, integrity, and confidentiality) was determined that for the protocols, standards, or buses associated to the IT environment the order of impact is confidentiality, integrity, and availability, while for both IoT and OT paradigms the order of impact is availability, confidentiality, and integrity. We have also been able to evaluate the impact of the pillars of cybersecurity in the established IIoT classification, where although availability is the most affected parameter for each of them, which is critical for Fabric and Process automation Protocols, for various environments such as Smart Home Automation Protocols and Vehicular Automation Protocols confidentiality is the second most affected parameter, being very close to availability in both cases. An accurate integration between attack pattern, vulnerability, and weakness was obtained, which demonstrated that "flooding" is the most used attack pattern and "improper input validation" is a more exploited weakness. In addition, it is also confirmed that the most common attack used in IoT wireless technologies, such as 4G/LTE, LoRa, and NB-IoT, is jamming. Based on the NIST 800-82r2 recommendation, which establishes a classification of vulnerabilities in industrial environments, we classify the vulnerabilities that most affect our IIoT protocols, standards, and buses. Although there are numerous exceptions, we can detail three major groups associated with the classification established for IIoT in IT, IoT, and OT. In the case of IT, vulnerabilities related to improper data validation predominate. In the case of IoT, physical and network configuration and communication, vulnerabilities predominate. In the case of OT, inadequate incorporation of security into architecture and design vulnerabilities predominate. The future research focuses on fully automating the VAF methodology, incorporating searches from the Common Platform Enumeration (CPE), so that we can associate to a given asset all the results that VAF offers.

The future research focuses on fully automating the VAF methodology, incorporating searches from the Common Platform Enumeration (CPE) so that we can associate to a given asset all the results that VAF offers.

8. ACKNOWLEDMENT

This research was supported by the Basque Government (Elkartek program) through both projects "CYBERPREST-Cybersegurtasunerako gaitasun osoa" with project number KK-2018/00076 and "SENDAI-SEgurtasun integrala iNDustria AdImentsurako" with project number KK-2019/00072.

9. REFERENCES

[1] S. Bhattacharjee, Practical Industrial Internet of Things Security. Packt Publishing Ltd, 2018. [Online]. Available: https://www.packtpub.com/eu/business/practical-industrial-internet-things-security.

[2] R. Shirey, "Internet Security Glossary, Version 2," 2007. [Online]. Available: https://tools.ietf.org/html/rfc4949.

[3] S. Whalen, M. Bishop, and S. Engle, "Protocol Vulnerability Analysis," Citeseer, p. 14, 2005. [Online]. Available: https://pdfs.semanticscholar.org/cb46/7b25e76e309b15fef603882c8b9892a2ddc7.pdf 7

[4] K. Stouffer, V. Pillitteri, S. Lightman, M. Abrams, and A. Hahn. 2015. NIST special publication 800-82: Guide to industrial control systems (ICS) security. Retrieved from https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-82r2.pdf.

[5] R. Martin et al. 2016. Industrial Internet Security Framework Technical Report, Second. Highland Avenue Needham, MA. Industrial Internet Consortium. Retrieved from https://www.iiconsortium.org/pdf/IIC_PUB_G4_V1.00_PB-3.pdf.

[6] A. Hasibuan, M. Mustadi, I. E. Y. Syamsuddin, and I. M. A. Rosidi, "Design and implementation of modular home automation based on wireless network, REST API, and WebSocket," in 2015 International Symposium on Intelligent Signal Processing and Communication Systems (ISPACS), 2015, pp. 362–367, doi: doi.org/10.1109/ISPACS.2015.7432797.

[7] V. M. Trifa, D. Guinard, and M. Koehler, "Messaging methods in a service-oriented architecture for industrial automation systems," in 2008 5th International Conference on Networked Sensing Systems, 2008, pp. 35–38, doi: doi.org/10.1109/INSS.2008.4610893.

[8] J. Yang, K. Sandström, T. Nolte, and M. Behnam, "Data Distribution Service for industrial automation," in Proceedings of 2012 IEEE 17th International Conference on Emerging Technologies & Factory Automation (ETFA 2012), 2012, pp. 1–8, doi: doi.org/10.1109/ETFA.2012.6489544.

[9] A. Alaerjan and D. Kim, "Configuring DDS features for communicating components in smart grids," in 2017 IEEE International Conference on Smart Energy Grid Engineering (SEGE), 2017, pp. 162–169, Doi: doi.org/10.1109/SEGE.2017.8052793.

[10] J. Rodríguez-Molina, S. Bilbao, B. Martínez, M. Frasheri, and B. Cürüklü, "An Optimized, Data Distribution Service-Based Solution for Reliable Data Exchange Among Autonomous Underwater Vehicles," Sensors (Basel)., vol. 17, no. 8, p. 1802, Aug. 2017, Doi: doi.org/10.3390/s17081802.

[11] A. A. Khan and H. T. Mouftah, "Secured web services for home automation in smart grid environment," in 2012 25th IEEE Canadian Conference on Electrical and Computer Engineering (CCECE), 2012, pp. 1–4, Doi: doi.org/10.1109/CCECE.2012.6335018.

[12] Y. Wenbo, W. Quanyu, and G. Zhenwei, "Smart home implementation based on Internet and WiFi technology," in 2015 34th Chinese Control Conference (CCC), 2015, pp. 9072–9077, doi: doi.org/10.1109/ChiCC.2015.7261075.

[13] G. Afifi, H. H. Halawa, R. M. Daoud, and H. H. Amer, "Dual protocol performance using WiFi and Zigbee for industrial WLAN," in 2016 24th Mediterranean Conference on Control and Automation (MED), 2016, pp. 749–754, doi: doi.org/10.1109/MED.2016.7535854.

[14] K. Khanchuea and R. Siripokarpirom, "A Multi-Protocol IoT Gateway and WiFi/BLE Sensor Nodes for Smart Home and Building Automation: Design and Implementation," in 2019 10th International Conference of Information and Communication Technology for Embedded Systems (IC-ICTES), 2019, pp. 1–6, Doi: doi.org/10.1109/ICTEmSys.2019.8695968.

[15] G. V Vivek and M. P. Sunil, "Enabling IOT services using WIFI - ZigBee gateway for a home automation system," in 2015 IEEE International Conference on Research in Computational Intelligence and Communication Networks (ICRCICN), 2015, pp. 77–80, doi: doi.org/10.1109/ICRCICN.2015.7434213.

[16] S. A. Ashraf, I. Aktas, E. Eriksson, K. W. Helmersson, and J. Ansari, "Ultra-reliable and low-latency communication for wireless factory automation: From LTE to 5G," in 2016 IEEE 21st International Conference on Emerging Technologies and Factory Automation (ETFA), 2016, pp. 1–8, Doi: doi.org/10.1109/ETFA.2016.7733543.

[17] P. Orosz, P. Varga, G. Soós, and C. Hegedűs, "QoS Guarantees for Industrial IoT Applications over LTE - a Feasibility Study," in 2019 IEEE International Conference on Industrial Cyber Physical Systems (ICPS), 2019, pp. 667–672, Doi: doi.org/10.1109/ICPHYS.2019.8780308.

[18] A. Ahmed, M. M. Khan, and W. Ahmed, "Cloud based network management and control for building automation," in 2016 19th International Multi-Topic Conference (INMIC), 2016, pp. 1–6, Doi: doi.org/10.1109/INMIC.2016.7840123.

[19] X. Feng, S. Zhao, Y. Wang, and T. Qi, "The Application on 4G-LTE System with Evaporation Duct," in 2018 2nd IEEE Advanced Information Management,Communicates,Electronic and Automation Control Conference (IMCEC), 2018, pp. 1970–1975, Doi: doi.org/10.1109/IMCEC.2018.8469369.

[20] G. A. Akpakwu, B. J. Silva, G. P. Hancke, and A. M. Abu-Mahfouz, "A Survey on 5G Networks for the Internet of Things: Communication Technologies and Challenges," IEEE Access, vol. 6, pp. 3619–3647, 2018, Doi: doi.org/10.1109/ACCESS.2017.2779844.

[21] D. Bezerra, R. R. Aschoff, G. Szabo, and D. F. H. Sadok, "An IoT Protocol Evaluation in a Smart Factory Environment," in 2018 Latin American Robotic Symposium, 2018 Brazilian Symposium on Robotics (SBR) and 2018 Workshop on Robotics in Education (WRE), 2018, pp. 118–123, Doi: doi.org/10.1109/LARS/SBR/WRE.2018.00030.

[22] R. K. Kodali and S. Soratkal, "MQTT based home automation system using ESP8266," in 2016 IEEE Region 10 Humanitarian Technology Conference (R10-HTC), 2016, pp. 1–5, Doi: doi.org/10.1109/R10-HTC.2016.7906845.

[23] C. Bormann, A. P. Castellani, and Z. Shelby, "CoAP: An Application Protocol for Billions of Tiny Internet Nodes," IEEE Internet Comput., vol. 16, no. 2, pp. 62–67, 2012, Doi: doi.org/10.1109/MIC.2012.29.

[24] I. Shin, D. Eom, and B. Song, "The CoAP-based M2M gateway for distribution automation system using DNP3.0 in smart grid environment," in 2015 IEEE International Conference on Smart Grid Communications (SmartGridComm), 2015, pp. 713–718, Doi: doi.org/10.1109/SmartGridComm.2015.7436385.

[25] D. Halabi, S. Hamdan, and S. Almajali, "Enhance the security in smart home applications based on IOT-CoAP protocol," in 2018 Sixth International Conference on Digital Information, Networking, and Wireless Communications (DINWC), 2018, pp. 81–85, Doi: doi.org/10.1109/DINWC.2018.8357000.

[26] M. Grover, M. E. I. I. Year, S. K. Pardeshi, N. Singh, and S. Kumar, "Bluetooth Low Energy for Industrial Automation," no. Icecs, pp. 512–515, 2015, Doi: doi.org/ 10.1109/ECS.2015.7124960.

[27] E. J. Sebastian, A. Yushev, A. Sikora, M. Schappacher, and J. A. Prasetyo, "Performance investigation of 6Lo with RPL mesh networking for home and building automation," in 2016 3rd International Symposium on Wireless Systems within the Conferences on Intelligent Data Acquisition and Advanced Computing Systems (IDAACS-SWS), 2016, pp. 127–133, Doi: doi.org/10.1109/IDAACS-SWS.2016.7805801.

[28] P. H. B. Shinde, A. Chaudhari, P. Chaure, M. Chandgude, and P. Waghmare, "Smart Home Automation System using Android Application," pp. 2408–2411, 2017. [Online]. Available: https://www.irjet.net/archives/V4/i4/IRJET-V4I4604.pdf.

[29] X. Shen, X. Wang, and M. Jia, "Design and implementation of traffic information detection equipment based on bluetooth communication," in 2017 IEEE 2nd Information Technology, Networking, Electronic and Automation Control Conference (ITNEC), 2017, pp. 1595–1601, Doi: doi.org/10.1109/ITNEC.2017.8285063.

[30] H. P. Lin, S. C. Cheng, D. B. Lin, C. H. Chung, and R. S. Hsiao, "Integrating ZigBee lighting control into existing building automation systems," IET Int. Conf. Inf. Sci. Control Eng. 2012 (ICISCE 2012), pp. 3.48-3.48, 2012, Doi: doi.org/10.1049/cp.2012.2445.

[31] W. Yang, H. Jiang, J. Wu, and C. Zhang, "Research on the hybrid network technology of industrial ethernet and Zigbee for monitoring the ship power system," in 2010 2nd International Asia Conference on Informatics in Control, Automation and Robotics (CAR 2010), 2010, vol. 1, pp. 460–463, Doi: doi.org/10.1109/CAR.2010.5456798.

[32] S. Rana, H. Zhu, C. W. Lee, D. M. Nicol, and I. Shin, "The Not-So-Smart Grid: Preliminary work on identifying vulnerabilities in ANSI C12.22," 2012 IEEE Globecom Work. GC Wkshps 2012, pp. 1514–1519, 2012, Doi: doi.org/10.1109/GLOCOMW.2012.6477810.

[33] F. M. Schaefer, T. Groß, and R. Kays, "Energy consumption of 6LoWPAN and Zigbee in home automation networks," IFIP Wirel. Days, pp. 13–15, 2013, Doi: doi.org/10.1109/WD.2013.6686463.

[34] C. Swedberg, "General Motors Factory Installs Smart Bolts in Engine Blocks, Cylinder Heads," RFID J., pp. 1–2, 2014. [Online]. Retrieved: 29-Mar-2019. [From: https://www.rfidjournal.com/articles/view?11329]

[35] O. Bindroo, K. Saxena, and S. K. Khatri, "A wearable NFC wristband for remote home automation system," in 2017 2nd International Conference on Telecommunication and Networks (TEL-NET), 2017, pp. 1–6, Doi: doi.org/10.1109/TEL-NET.2017.8343563.

[36] A. Haidine, A. Aqqal, and A. Dahbi, "Performance Evaluation of Low-Power Wide Area based on LoRa Technology for Smart Metering," in 2018 6th International Conference on Wireless Networks and Mobile Communications (WINCOM), 2018, pp. 1–6, Doi: doi.org/10.1109/WINCOM.2018.8629693.

[37] K. A. Nsiah, Z. Amjad, A. Sikora, and B. Hilt, "Performance Evaluation of Latency for NB-LTE Networks in Industrial Automation," in 2019 IEEE 30th Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC), 2019, pp. 1–7, Doi: doi.org/10.1109/PIMRC.2019.8904407.

[38] S. Chen, Y. Li, M. H. Memon, and F. Lin, "Design and Implementation of Cell Search in NB-IoT Downlink Receiver," in 2018 IEEE International Conference on Integrated Circuits, Technologies and Applications (ICTA), 2018, pp. 20–21, Doi: doi.org/10.1109/CICTA.2018.8705949.

[39] J. Du and X. Liu, "Design and Implementation of Smart Socket Based on NB-IoT," in 2019 IEEE 3rd Information Technology, Networking, Electronic and Automation Control Conference (ITNEC), 2019, pp. 1033–1037, Doi: doi.org/10.1109/ITNEC.2019.8729284.

[40] H. Hayashi, T. Hasegawa, and K. Demachi, "Wireless technology for process automation," in 2009 ICCAS-SICE, 2009, pp. 4591–4594, IEEE: https://ieeexplore.ieee.org/document/5333003.

[41] M. Kuzlu, M. Pipattanasomporn, and S. Rahman, "Review of communication technologies for smart homes/building applications," in 2015 IEEE Innovative Smart Grid Technologies - Asia (ISGT ASIA), 2015, pp. 1–6, Doi: doi.org/10.1109/ISGT-Asia.2015.7437036.

[42] S. Ahmad, "Smart metering and home automation solutions for the next decade," in 2011 International Conference on Emerging Trends in Networks and Computer Communications (ETNCC), 2011, pp. 200–204, Doi: doi.org/10.1109/ETNCC.2011.5958516.

[43] P. Ferrari, A. Flammini, F. Venturini, and A. Augelli, "Large PROFINET IO RT networks for factory automation: A case study," in ETFA2011, 2011, pp. 1–4, Doi: doi.org/10.1109/ETFA.2011.6059160.

[44] J. Vasel, "One plant, one system: Benefits of integrating process and power automation," in 2012 65th Annual Conference for Protective Relay Engineers, 2012, pp. 215–250, Doi: doi.org/10.1109/CPRE.2012.6201235.

[45] K. T. Smith and C. Architect, "Cybersecurity and the IoT — Threats , Best Practices and Lessons Learned," pp. 1–10, 2017. [Online]. Retrieved: 03-Apr-2019 [From: https://bit.ly/2Ua6tMq].

[46] T. C. Hooi, M. Singh, Y. K. Siah, and A. R. bin Ahmad, "Building low-cost intelligent building components with controller area network (CAN) bus," in Proceedings of IEEE Region 10 International Conference on Electrical and Electronic Technology. TENCON 2001, 2001, vol. 1, pp. 466–468 vol.1, Doi: doi.org/10.1109/TENCON.2001.949636.

[47] D. Liaw, C. Yu, and K. Wu, "A CAN-based design for the control of electric vehicle," in 2014 14th International Conference on Control, Automation and Systems (ICCAS 2014), 2014, pp. 1233–1237, Doi: doi.org/10.1109/ICCAS.2014.6987745.

[48] A. I. Abashar, M. A. Mohammedeltoum, and O. D. Abaker, "Automated and monitored liquid filling system using PLC technology," in 2017 International Conference on Communication, Control,

Computing and Electronics Engineering (ICCCCEE), 2017, pp. 1–5, Doi: doi.org/10.1109/ICCCCEE.2017.7866699.

[49] S. S. Khuzyatov and R. A. Valiev, "Organization of data exchange through the modbus network between the SIMATIC S7 PLC and field devices," 2017 Int. Conf. Ind. Eng. Appl. Manuf. ICIEAM 2017 - Proc., pp. 15–17, 2017, Doi: doi.org/10.1109/ICIEAM.2017.8076369.

[50] J. Rinaldi, "An Overview of EtherNet/IP. An Application Layer Protocol for Industrial Automation," Wauwatosa WI, 2003. [Online]. Retrieved: 03-Apr-2019 [From: https://www.rtautomation.com/technologies/ethernetip/].

[51] S. Rao, G. V Chatrapathi, and T. Yashashwini, "EtherNet/IP + FDI: Value in Process Automation," 2017 2nd Int. Conf. Emerg. Comput. Inf. Technol., pp. 1–5, 2017, Doi: doi.org/10.1109/ICECIT.2017.8453324.

[52] Thomas Hilz, "HART at the speed of Ethernet," 2015. [Online]. Retrieved: 03-Apr-2019 [From: http://www.controlengeurope.com/article/106724/HART-at-the-speed-of-Ethernet.aspx].

[53] S. H. Hong, "Development of a BACnet-ZigBee gateway for demand response in buildings," 2013 Pan African Int. Conf. Inf. Sci. Comput. Telecommun. PACT 2013, pp. 19–23, 2013, Doi: doi.org/10.1109/SCAT.2013.7055082.

[54] H. Dachao, H. Yu, and C. Shaokuan. 2007. Research and application of sinec L2 and modbus plus networks on industrial automation. In 2007 International Conference on Mechatronics and Automation. 3424--3428. Doi: doi.org/10.1109/ICMA.2007.4304113.

[55] T. Tenkanen and T. Hamalainen, "Security Assessment of a Distributed, Modbus-Based Building Automation System," IEEE CIT 2017 - 17th IEEE Int. Conf. Comput. Inf. Technol., pp. 332–337, 2017, Doi: doi.org/10.1109/CIT.2017.38.

[56] Triangle Microworks, "Scada Data Gateway," 2018. [Online]. Retrieved: 03-Apr-2019 [From: http://www.trianglemicroworks.com/products/scada-data-gateway/iccp-tase-2].

[57] A. C. D. Bonganay, J. C. Magno, A. G. Marcellana, J. M. E. Morante, and N. G. Perez, "Automated electric meter reading and monitoring system using ZigBee-integrated raspberry Pi single board computer via Modbus," 2014 IEEE Students' Conf. Electr. Electron. Comput. Sci. SCEECS 2014, 2014, Doi: doi.org/10.1109/SCEECS.2014.6804531.

[58] Y. Shimanuki, "OLE for process control (OPC) for new industrial automation systems," IEEE SMC'99 Conf. Proceedings. 1999 IEEE Int. Conf. Syst. Man, Cybern. (Cat. No.99CH37028), vol. 6, pp. 1048–1050, 1999, Doi: doi.org/10.1109/ICSMC.1999.816721.

[59] H. Haskamp, F. Orth, J. Wermann, and A. W. Colombo, "Implementing an OPC UA interface for legacy PLC-based automation systems using the Azure cloud: An ICPS-architecture with a retrofitted RFID system," Proc. - 2018 IEEE Ind. Cyber-Physical Syst. ICPS 2018, pp. 115–121, 2018, Doi: doi.org/10.1109/ICPHYS.2018.8387646.

[60] K. S. Manoj, Industrial Automation with SCADA : Concepts, Communications and Security, First Edit. Notion Press, 2019. [Online]. Available: https://books.google.es/books?id=FgCRDwAAQBAJ&lpg=PP1&pg=PP1#v=onepage&q&f=false.

[61] S. Kim, H. Chng, and T. Shon, "Survey on security techniques for AMI metering system," ISOCC 2014 - Int. SoC Des. Conf., pp. 192–193, 2015, Doi: doi.org/10.1109/ISOCC.2014.7087691.

[62] B. Chen, M. Chen, H. Tian, and L. Chen. 2017. Advanced application of IEC60870-5-101 protocol on feeder terminal unit. In Proceedings of the International Conference on Anti-Counterfeiting, Security and Identification (ASID'18), vol. 2017-Octob. 142--145. Doi: doi.org/10.1109/ICASID.2017.8285761.

[63] N. V. Mago, J. D. Moseley, and N. Sarma, "A methodology for modeling telemetry in power systems models using IEC-61968/61970," 2013 IEEE Innov. Smart Grid Technol. (ISGT Asia), pp. 1–6, 2013, Doi: doi.org/10.1109/ISGT-Asia.2013.6698713.

[64] I. A. N. A. (IANA), "Service Name and Transport Protocol Port Number Registry," 2019. [Online]. Retrieved: 04-May-2019. [From: https://bit.ly/346PQCN].

[65] Z. B. Celik, E. Fernandes, E. Pauley, G. Tan, and P. McDaniel, "Program Analysis of Commodity IoT Applications for Security and Privacy: Challenges and Opportunities," ACM Comput. Surv., vol. 52, no. 4, pp. 74:1--74:30, 2019, Doi: doi.org/10.1145/3333501.

[66] X. Zhang, Z. Wen, Y. Wu, and J. Zou, "The implementation and application of the internet of things platform based on the REST architecture," in 2011 International Conference on Business Management and Electronic Information, 2011, vol. 2, pp. 43–45, Doi: doi.org/10.1109/ICBMEI.2011.5917838.

[67] B. N. Nakhuva and T. A. Champaneria, "Security provisioning for RESTful web services in Internet of Things," in 2017 4th International Conference on Advanced Computing and Communication Systems (ICACCS), 2017, pp. 1–6, Doi: doi.org/10.1109/ICACCS.2017.8014642.

[68] H. Lee and M. R. Mehta, "Defense Against REST-based Web Service Attacks for Enterprise Systems," Commun. IIMA, vol. 13, no. 1, pp. 57–68, 2013.

[69] R. Wang, S. Chen, and X. Wang, "Signing Me onto Your Accounts through Facebook and Google: A Traffic-Guided Security Study of Commercially Deployed Single-Sign-On Web Services," in 2012 IEEE Symposium on Security and Privacy, 2012, pp. 365–379, Doi: doi.org/10.1109/SP.2012.30.

[70] Egor Homakov, "The Most Common OAuth2 Vulnerability," 2012. [Online]. Retrieved: 05-May-2019. [From: http://homakov.blogspot.com/2012/07/saferweb-most-common-oauth2.html].

[71] V. Clincy and H. Shahriar, "Web service injection attack detection," in 2017 12th International Conference for Internet Technology and Secured Transactions (ICITST), 2017, pp. 173–178, Doi: doi.org/10.23919/ICITST.2017.8356371.

[72] Open Web Application Security Project, "REST Security Cheat Sheet," 2018. [Online]. Retrieved: 05-May-2019 [From: https://www.owasp.org/index.php/REST_Security_Cheat_Sheet].

[73] G. C. Fernandez, E. S. Ruiz, M. C. Gil, and F. M. Perez, "From RGB led laboratory to servomotor control with websockets and IoT as educational tool," in Proceedings of 2015 12th International Conference on Remote Engineering and Virtual Instrumentation (REV), 2015, pp. 32–36, Doi: doi.org/10.1109/REV.2015.7087259.

[74] F. Greco, "API Design and WebSocket," Sep 25, 2014, 2014. [Online]. Retrieved: 05-May-2019 [From: https://www.slideshare.net/grecof/api-designandweb-socket?from_action=save].

[75] H. D. Center, "WebSocket Security," 14 August 2019, 2019. [Online]. Retrieved: 05-May-2019. [From: https://devcenter.heroku.com/articles/websocket-security].

[76] M. Shema, S. Shekyan, and V. Toukharian, "Hacking with WebSockets," 2012. [Online]. Retrieved: 05-May-2019 [From: https://bit.ly/38SmPxt].

[77] G. Chen, Y. Du, P. Qin, and L. Zhang, "Research of JMS Based Message Oriented Middleware for Cluster," in 2013 International Conference on Computational and Information Sciences, 2013, pp. 1628–1631, Doi: doi.org/10.1109/ICCIS.2013.426.

[78] IBM, "Developing client applications," Monday, 25 February 2019, 2019. [Online]. Retrieved: 07-May-2019. [From: https://www.ibm.com/support/knowledgecenter/en/SSWMAJ_5.0.0/com.ibm.ism.doc/Developing/develop_guide.html].

[79] IBM Integration Bus, "Configuring the integration node to use SSL with JMS nodes," 2019-03-30 08:15:45, 2019. [Online]. Retrieved: 03-May-2019. [From: https://ibm.co/36ENEDH].

[80] O. Corporation, "Using JAAS-Based Authentication," 2010. [Online]. Retrieved: 07-May-2019. [From: https://docs.oracle.com/cd/E19879-01/820-6740/gepfq/index.html].

[81] G. S. Kalra, "A Pentesters Guide to Hacking ActiveMQ-Based JMS Applications," 2014. Retrieved: 03-Jun-2019. [From: https://bit.ly/2P43Vg6].

[82] Object Management Group (OMG). 2019. The Real-Time Publish-Subscribe Wire Protocol DDS Interoperability Wire Protocol (DDSI-RTPS) Version 2.3. Retrieved from https://www.omg.org/spec/DDSI-RTPS/2.3/PDF.

[83] OMG, "Extensible and Dynamic Topic Types for DDS," Needham, MA, 2012. Retrieved: 03-Jun-2019. [From: https://www.omg.org/spec/DDS-XTypes/About-DDS-XTypes/].

[84] L. Bertaux, A. Hakiri, S. Medjiah, P. Berthou, and S. Abdellatif, "A DDS/SDN Based Communication System for Efficient Support of Dynamic Distributed Real-Time Applications," in 2014 IEEE/ACM 18th International Symposium on Distributed Simulation and Real Time Applications, 2014, pp. 77–84, Doi: doi.org/10.1109/DS-RT.2014.18.

[85] G. Yoon, J. Choi, H. Park, and H. Choi, "Topic naming service for DDS," in 2016 International Conference on Information Networking (ICOIN), 2016, pp. 378–381, Doi: doi.org/10.1109/ICOIN.2016.7427138.

[86] A. Alaerjan, D. Kim, H. Ming, and K. Malik, "Using DDS Based on Unified Data Model to Improve Interoperability of Smart Grids," in 2018 IEEE International Conference on Smart Energy Grid Engineering (SEGE), 2018, pp. 110–114, Doi: doi.org/10.1109/SEGE.2018.8499513.

[87] P. Peniak and M. Franekova, "Open communication protocols for integration of embedded systems within Industry 4.0," in 2015 International Conference on Applied Electronics (AE), 2015, pp. 181–184.

[88] OMG, "DDS Security," Needham, MA, 2018. [Online]. Retrieved: 05-Jun-2019. [From: https://www.omg.org/spec/DDS-SECURITY/About-DDS-SECURITY/].

[89] T. White and M. N. Johnstone, "An investigation into some security issues in the DDS messaging protocol," in AUSTRALIAN INFORMATION SECURITY MANAGEMENT CONFERENCE, 2017, pp. 132–139, Doi: doi.org/10.4225/75/5a84fcff95b52.

[90] M. J. Michaud, T. Dean, and S. P. Leblanc, "Attacking OMG Data Distribution Service (DDS) Based Real-Time Mission Critical Distributed Systems," in 2018 13th International Conference on Malicious and Unwanted Software (MALWARE), 2018, pp. 68–77, Doi: doi.org/10.1109/MALWARE.2018.8659368.

[91] P. Saint-andre, "Extensible Messaging and Presence Protocol (XMPP): Core," Parker, CO, 2011. [Online]. Available: https://xmpp.org/rfcs/rfc6120.html.

[92] Peter Saint-Andre, "Extensible Messaging and Presence Protocol (XMPP): Core." Saint-Andre, pp. 1–73, 2014. [Online]. Retrieved: 07-Jun-2019. [From: https://xmpp.org/rfcs/rfc6120.html].

[93] K. Zeilenga, "XEP-0258: Security Labels in XMPP." [Online]. Retrieved: 09-Jun-2019. [From: https://xmpp.org/extensions/xep-0258.html].

[94] C. Davidland and L. George, "XEP-0419: Improving Baseline Security in XMPP," 2019. [Online]. Retrieved: 09-Jun-2019. [From: https://xmpp.org/extensions/xep-0419.html].

[95] P. Saint-andre, "XEP-0205 : Best Practices to Discourage Denial of Service Attacks," Parker, CO, 2018. [Online]. Retrieved: 09-Jun-2019. [From: https://xmpp.org/extensions/xep-0205.html].

[96] B. Chifor, S. Teican, M. Togan, and G. Gugulea, "A Flexible Authorization Mechanism for Enterprise Networks Using Smart-Phone Devices," in 2018 International Conference on Communications (COMM), 2018, pp. 437–440, Doi: doi.org/10.1109/ICComm.2018.8484268.

[97] A. Esser and C. Serrao, "Wi-Fi Network Testing Using an Integrated Evil-Twin Framework," in 2018 Fifth International Conference on Internet of Things: Systems, Management and Security, 2018, pp. 216–221, Doi: doi.org/10.1109/IoTSMS.2018.8554388.

[98] C. Sudar, S. K. Arjun, and L. R. Deepthi, "Time-based one-time password for Wi-Fi authentication and security," in 2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI), 2017, pp. 1212–1216, Doi: doi.org/10.1109/ICACCI.2017.8126007.

[99] B. Li, H. Yu, and F. Tan, "Wireless Network Security Detection System Design Based on Client," in 2018 International Conference on Intelligent Transportation, Big Data & Smart City (ICITBS), 2018, pp. 227–230, Doi: doi.org/10.1109/ICITBS.2018.00066.

[100] Z. Liu and J. Zhang, "Launching Low-Rate DoS Attacks with Cache-Enabled WiFi Offloading," in 2018 14th International Conference on Mobile Ad-Hoc and Sensor Networks (MSN), 2018, pp. 171–176, Doi: doi.org/10.1109/MSN.2018.00028.

[101] J. Cao, M. Ma, H. Li, Y. Zhang, and Z. Luo, "A Survey on Security Aspects for LTE and LTE-A Networks," IEEE Commun. Surv. Tutorials, vol. 16, no. 1, pp. 283–302, 2014, Doi: doi.org/10.1109/SURV.2013.041513.00174.

[102] Y. Mehmood, C. Görg, M. Muehleisen, and A. Timm-Giel, "Mobile M2M communication architectures, upcoming challenges, applications, and future directions," EURASIP J. Wirel. Commun. Netw., vol. 2015, p. 250, Nov. 2015, Doi: doi.org/10.1186/s13638-015-0479-y.

[103] A. Ghosh, R. Ratasuk, B. Mondal, N. Mangalvedhe, and T. Thomas, "LTE-advanced: next-generation wireless broadband technology [Invited Paper]," IEEE Wirel. Commun., vol. 17, no. 3, pp. 10–22, 2010, Doi: doi.org/10.1109/MWC.2010.5490974.

[104] T. Specification, "Security architecture (3GPP TS 33.401 version 12.13.0 Release 12)," vol. 0, 2015. [Online]. Available: https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=2296.

[105] H. Shariatmadari et al., "Machine-type communications: Current status and future perspectives toward 5G systems," IEEE Commun. Mag., vol. 53, no. 9, pp. 10–17, 2015, Doi: doi.org/10.1109/MCOM.2015.7263367.

[106] 3GPP, "Service requirements for Home Node B (HNB) and Home eNode B (HeNB)," 2012. [Online]. Available: https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=626.

[107] 3GPP, "Evolved Universal Terrestrial Radio Access (E-UTRA) and Evolved Universal Terrestrial Radio Access Network (E-UTRAN)," 2017. [Online]. Retrieved: 13-Jun-2019 [From: https://bit.ly/2tfmBBr].

[108] S. Teral, "5G Best Choice Architecture," IHS Markit Technol., no. January, pp. 1–17, 2019. [Online]. Available: https://cdn.ihs.com/www/prot/pdf/0519/IHSMarkit_5G_Best_Choice_Architecture.pdf.

[109] 5G Americas, "The evolution of security in 5G," 2018. [Online]. Retrieved: 13-Jun-2019. [From: https://www.5gamericas.org/wp-content/uploads/2019/08/5G-Security-White-Paper-07-26-19-FINAL.pdf].

[110] M. Suryanegara, A. S. Arifin, and M. Asvial, "The IoT-based Transition Strategy Towards 5G," in Proceedings of the International Conference on Big Data and Internet of Thing, 2017, pp. 186–190, Doi: doi.org/10.1145/3175684.3175728.

[111] 5G Americas. 2018. The evolution of security in 5G. Retrieved from https://www.5gamericas.org/wp-content/uploads/2019/08/5G-Security-White-Paper-07-26-19-FINAL.pdf.

[112] Raphael J Cohn; Richard J Coppen, "MQTT Version 3.1.1," United Kingdom, 2015. [Online]. Retrieved: 13-Jun-2019. [From: http://docs.oasis-open.org/mqtt/mqtt/v3.1.1/csprd02/mqtt-v3.1.1-csprd02.html].

[113] M. Version, "MQTT Version 5.0," United Kingdom, 2019. [Online]. Retrieved: 13-Jun-2019. [From: https://docs.oasis-open.org/mqtt/mqtt/v5.0/mqtt-v5.0.html].

[114] M. B. Yassein, M. Q. Shatnawi, S. Aljwarneh, and R. Al-Hatmi, "Internet of Things: Survey and open issues of MQTT protocol," in 2017 International Conference on Engineering & MIS (ICEMIS), 2017, pp. 1–6, Doi: doi.org/10.1109/ICEMIS.2017.8273112.

[115] M. S. Harsha, B. M. Bhavani, and K. R. Kundhavai, "Analysis of vulnerabilities in MQTT security using Shodan API and implementation of its countermeasures via authentication and ACLs," in 2018 International Conference on Advances in Computing, Communications and Informatics (ICACCI), 2018, pp. 2244–2250, Doi: doi.org/10.1109/ICACCI.2018.8554472.

[116] M. Erber, "Role Based Access Control to Secure an MQTT Broker," 2019. [Online]. Retrieved: 15-Jun-2019. [From: https://www.hivemq.com/blog/rbac-for-the-control-center-with-ese/].

[117] A. Niruntasukrat, C. Issariyapat, P. Pongpaibool, K. Meesublak, P. Aiumsupucgul, and A. Panya, "Authorization mechanism for MQTT-based Internet of Things," in 2016 IEEE International Conference on Communications Workshops (ICC), 2016, pp. 290–295, Doi: doi.org/10.1109/ICCW.2016.7503802.

[118] T. H. Team, "TLS/SSL - MQTT Security Fundamentals," May 11, 2015, 2015. [Online]. Retrieved: 17-Jun-2019. [From: https://www.hivemq.com/blog/mqtt-security-fundamentals-tls-ssl/].

[119] S. H. Ramos, M. T. Villalba, and R. Lacuesta, "MQTT Security : A Novel Fuzzing Approach," Wirel. Commun. Mob. Comput., vol. 2018, p. 11, 2018.

[120] S. Andy, B. Rahardjo, and B. Hanindhito, "Attack scenarios and security analysis of MQTT communication protocol in IoT system," in 2017 4th International Conference on Electrical Engineering, Computer Science and Informatics (EECSI), 2017, pp. 1–6, Doi: doi.org/10.1109/EECSI.2017.8239179.

[121] V. Lakkundi and K. Singh, "Lightweight DTLS implementation in CoAP-based Internet of Things," in 20th Annual International Conference on Advanced Computing and Communications (ADCOM), 2014, pp. 7–11, Doi: doi.org/10.1109/ADCOM.2014.7103240.

[122] C. Bormann, S. Lemay, H. Tschofenig, K. Hartke, B. Silverajan, and B. Raymor. 2018. CoAP (constrained application protocol) over TCP, TLS, and WebSockets. Retrieved from https://tools.ietf.org/html/rfc8323.

[123] R. A. Rahman and B. Shah, "Security analysis of IoT protocols: A focus in CoAP," in 2016 3rd MEC International Conference on Big Data and Smart City (ICBDSC), 2016, pp. 1–7, Doi: doi.org/10.1109/ICBDSC.2016.7460363.

[124] T. A. Alghamdi, A. Lasebae, and M. Aiash, "Security analysis of the constrained application protocol in the Internet of Things," in Second International Conference on Future Generation

Communication Technologies (FGCT 2013), 2013, pp. 163–168, Doi: doi.org/10.1109/FGCT.2013.6767217.

[125] A. Capossele, V. Cervo, G. De Cicco, and C. Petrioli, "Security as a CoAP resource: An optimized DTLS implementation for the IoT," in 2015 IEEE International Conference on Communications (ICC), 2015, pp. 549–554, Doi: doi.org/10.1109/ICC.2015.7248379.

[126] J. Granjal, E. Monteiro, and J. S. Silva, "Security for the Internet of Things: A Survey of Existing Protocols and Open Research Issues," IEEE Commun. Surv. Tutorials, vol. 17, no. 3, pp. 1294–1312, 2015, Doi: doi.org/10.1109/COMST.2015.2388550.

[127] S. Arvind and V. A. Narayanan, "An Overview of Security in CoAP: Attack and Analysis," in 2019 5th International Conference on Advanced Computing & Communication Systems (ICACCS), 2019, pp. 655–660, Doi: doi.org/10.1109/ICACCS.2019.8728533.

[128] N. Naik, "Choice of effective messaging protocols for IoT systems: MQTT, CoAP, AMQP and HTTP," in 2017 IEEE International Systems Engineering Symposium (ISSE), 2017, pp. 1–7, Doi: doi.org/10.1109/SysEng.2017.8088251.

[129] N. S. Han, "Semantic service provisioning for 6LoWPAN : powering internet of things applications on Web," Institut National des Télécommunications, 2015.

[130] M. Phillips, P. Adams, D. Rokicki, and E. Johnson, "URI Scheme for Java(tm) Message Service 1.0," 2011. [Online]. Available: https://tools.ietf.org/html/rfc6167.

[131] R. Cohn, "A Comparison of AMQP and MQTT," 2012. [Online]. Available: https://lists.oasis-open.org/archives/amqp/201202/msg00086/StormMQ_WhitePaper_-_A_Comparison_of_AMQP_and_MQTT.pdf.

[132] RabittMQ, "Authentication, Authorisation, Access Control," 2019. [Online]. Retrieved: 19-Jun-2019. [From: https://www.rabbitmq.com/access-control.html].

[133] D. Braue, "Small, unsophisticated developers perpetuating IoT security lapses," 2019. [Online]. Retrieved: 21-Jun-2019. [From: https://www.cso.com.au/article/560521/small-unsophisticated-developers-perpetuating-iot-security-lapses-ibm/].

[134] I. N. Mcateer, M. I. Malik, Z. Baig, and P. Hannay, "Security vulnerabilities and cyber threat analysis of the AMQP protocol for the internet of things," Aust. Inf. Secur. Manag. Conf., pp. 70–80, 2017, Doi: doi.org/10.4225/75/5a84f4a695b4c.

[135] I. Bluetooth SIG, "Bluetooth Core Specification Version 5.0," Washington, USA, 2016. [Online]. Retrieved: 21-Jun-2019. [From: https://www.bluetooth.org/docman/handlers/DownloadDoc.ashx?doc_id=421043].

[136] Bluetooth SIG Proprietary, "Bluetooth Core Specification v5.1," Washington, 2019. [Online]. Retrieved: 21-Jun-2019. [From: https://bit.ly/2REkBwf].

[137] K. Ren, "Bluetooth Pairing Part 3 Low Energy Legacy Pairing Passkey Entry," 2016. [Online]. Retrieved: 23-Jun-2019 [From: https://blog.bluetooth.com/bluetooth-pairing-passkey-entry].

[138] S. Figueroa Lorenzo, J. Añorga Benito, P. García Cardarelli, J. Alberdi Garaia, and S. Arrizabalaga Juaristi, "A Comprehensive Review of RFID and Bluetooth Security: Practical Analysis," Technologies, vol. 7, no. 1, p. 15, 2019, Doi: doi.org/10.3390/technologies7010015.

[139] M. Ryan, "Crackle." p. 1, 2015. [Online]. Retrieved: 23-Jun-2019. [From: https://lacklustre.net/projects/crackle/].

[140] P. Cope, J. Campbell, and T. Hayajneh, "An investigation of Bluetooth security vulnerabilities," 2017 IEEE 7th Annu. Comput. Commun. Work. Conf. CCWC 2017, pp. 1–7, 2017, Doi: doi.org/10.1109/CCWC.2017.7868416.

[141] J. Padgette et al. 2017. NIST Special Publication 800-121 Revision 2 Guide to Bluetooth Security. Retrieved from https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-121r2.pdf.

[142] Zigbee Alliance, "ZigBee Specification v 1.0," San Ramon, CA, 2005. [Online]. Retrieved: 23-Jun-2019. [From: https://www.zigbee.org/wp-content/uploads/2014/11/docs-05-3474-20-0csg-zigbee-specification.pdf].

[143] Zigbee Alliance, "Zigbee Specification v2," San Ramon, CA, 2012. [Online]. Retrieved: 25-Jun-2019. [From: https://zigbee.org/download/zigbee-3-0-base-device-behavior-specification/].

[144] T. Zillner and S. Strobl, "ZigBee Exploited. The Good, the Bad and the Ugly," in Black Hat USA, 2015, p. 47. [Online]. Retrieved: 25-Jun-2019. [From: https://zigbee.org/download/new-white-paper-zigbee-securing-the-wireless-iot/].

[145] O. Olawumi, K. Haataja, M. Asikainen, N. Vidgren, P. Toivanen, and K. Campus. 2014. Three practical attacks against zigbee security: Attack scenario definitions, practical experiments, countermeasures, and lessons learned. In 2014 14th International Conference on Hybrid Intelligent Systems. 199--206. Doi: doi.org/10.1109/HIS.2014.7086198.

[146] X. Cao, D. M. Shila, S. Member, Y. Cheng, and S. Member. 2016. Ghost-in-zigbee: Energy depletion attack on zigbee-based wireless networks. IEEE Internet Things J. 3, 5 (2016), 816--829. Doi: doi.org/10.1109/JIOT.2016.2516102.

[147] NFC Forum, "Core Protocol Technical Specifications," 2017. [Online]. Retrieved: 25-Jun-2019 [From: https://nfc-forum.org/our-work/specifications-and-application-documents/specifications/protocol-technical-specifications/].

[148] S. Figueroa, J. Añorga, S. Arrizabalaga, I. Irigoyen, and M. Monterde, "An Attribute-Based Access Control using Chaincode in RFID Systems," in 2019 10th IFIP International Conference on New Technologies, Mobility and Security (NTMS), 2019, pp. 1–5, Doi: doi.org/10.1109/NTMS.2019.8763824.

[149] S. Figueroa, J. Añorga, and S. Arrizabalaga, "An Attribute-Based Access Control Model in RFID Systems Based on Blockchain Decentralized Applications for Healthcare Environments," Computers, vol. 8, no. 3, 2019, Doi: doi.org/10.3390/computers8030057.

[150] F. D. Garcia et al., "Dismantling MIFARE classic," Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics), vol. 5283 LNCS, pp. 97–114, 2008, Doi: doi.org/10.1007/978-3-540-88313-5-7.

[151] L. Francis, G. Hancke, K. Mayes, and K. Markantonakis, "Practical relay attack on contactless transactions by using NFC mobile phones," Cryptol. Inf. Secur. Ser., vol. 8, pp. 21–32, 2012, Doi: doi.org/10.3233/978-1-61499-143-4-21.

[152] G. P. Hancke, "Eavesdropping Attacks on High-Frequency RFID Tokens," in Proceedings of the 4th Workshop on RFID Security (RFIDsec'08), 2008, no. July, pp. 1–36.

[153] LoRa Alliance Technical Commitee, "LoRaWAN 1.1 Specification," LoRaWAN 1.1 Specif., no. 1.1, p. 101, 2017. [Online]. Available: https://lora-alliance.org/resource-hub/lorawanr-specification-v11.

[154] E. Aras, G. S. Ramachandran, P. Lawrence, and D. Hughes, "Exploring the Security Vulnerabilities of LoRa," in 2017 3rd IEEE International Conference on Cybernetics (CYBCONF), 2017, pp. 1–6, Doi: doi.org/10.1109/CYBConf.2017.7985777.

[155] S. Chacko and M. D. Job, "Security mechanisms and Vulnerabilities in LPWAN," IOP Conf. Ser. Mater. Sci. Eng., vol. 396, no. 1, 2018, Doi: doi.org/10.1088/1757-899X/396/1/012027.

[156] R. S. Sinha, Y. Wei, and S.-H. Hwang, "A survey on LPWA technology: LoRa and NB-IoT," ICT Express, vol. 3, no. 1, pp. 14–21, 2017, Doi: doi.org/: https://Doi: doi.org/.org/10.1016/j.icte.2017.03.004.

[157] C. B. Mwakwata, H. Malik, M. M. Alam, Y. Le Moullec, S. Parand, and S. Mumtaz, "Narrowband internet of things (NB-IoT): From physical (PHY) and media access control (MAC) layers perspectives," Sensors (Switzerland), vol. 19, no. 11, pp. 1–34, 2019, Doi: doi.org/10.3390/s19112613.

[158] M. Iot and S. Report, "Security Features of LTE-M and NB-IoT Networks.". [Online]. Retrieved: 29-Jun-2019. [From: https://www.gsma.com/iot/resources/security-features-of-ltem-nbiot].

[159] F. L. Coman, K. M. Malarski, M. N. Petersen, and S. Ruepp, "Security Issues in Internet of Things: Vulnerability Analysis of LoRaWAN, Sigfox and NB-IoT," in 2019 Global IoT Summit (GIoTS), 2019, pp. 1–6, Doi: doi.org/10.1109/GIOTS.2019.8766430.

[160] D. Raposo, A. Rodrigues, S. Sinche, J. S. Silva, and F. Boavida, "Securing WirelessHART: Monitoring, Exploring and Detecting New Vulnerabilities," in 2018 IEEE 17th International Symposium on Network Computing and Applications (NCA), 2018, pp. 1–9, Doi: doi.org/10.1109/NCA.2018.8548060.

[161] R. Budampati and S. Kolavennu, Industrial Wireless Sensor Networks. Monitoring, Control and Automation, First., vol. 53, no. 9. Cambridge: Elsevier, 2016.

[162] S. Raza, A. Slabbert, T. Voigt, and K. Landernäs, "Security considerations for the WirelessHART protocol," in 2009 IEEE Conference on Emerging Technologies & Factory Automation, 2009, pp. 1–8, Doi: doi.org/10.1109/ETFA.2009.5347043.

[163] C. Alcaraz and J. Lopez, "A Security Analysis for Wireless Sensor Mesh Networks in Highly Critical Systems," IEEE Trans. Syst. Man, Cybern. Part C (Applications Rev., vol. 40, no. 4, pp. 419–428, 2010, Doi: doi.org/10.1109/TSMCC.2010.2045373.

[164] L. Bayou, D. Espes, N. Cuppens-Boulahia, and F. Cuppens. 2017. Security analysis of wirelessHART communication scheme bt—Foundations and practice of security. (2017), 223--238. Retrieved from https://hal.archives-ouvertes.fr/hal-01411385/document.

[165] C. Gomez and J. Paradells, "Wireless Home Automation Networks: A Survey of Architectures and Technologies," IEEE Commun. Mag., no. June, pp. 92–101, 2010.

[166] M. Smith, "EZ-Wave: A Z-Wave hacking tool capable of breaking bulbs abusing Z-Wave devices," 2018. [Online]. Retrieved: 29-Jun-2019 [From: https://www.csoonline.com/article/3024217/ez-wave-z-wave-hacking-tool-capable-of-breaking-bulbs-and-abusing-z-wave-devices.html].

[167] L. Rouch, "A Universal Controller to Take Over a Z-Wave Network," Black Hat Present., 2017. [Online]. Retrieved: 29-Jun-2019. [From: https://ubm.io/2U4WKau].

[168] B. Fouladi and S. Ghanoun, "Security Evaluation of the Z-Wave Wireless Protocol," Black hat, p. 6, 2013. [Online]. Retrieved: 29-Jun-2019. [From: https://bit.ly/2PurBZS].

[169] P. N. e. V. (PNO), "PROFINET System Description. Technology and application," Karlsruhe, 4.131, 2014. [Online]. Retrieved: 29-Jun-2019. [From: https://bit.ly/2vmu3ey].

[170] M. Yang and G. Li, "Analysis of PROFINET IO communication protocol," in Proceedings - 2014 4th International Conference on Instrumentation and Measurement, Computer, Communication and Control, IMCCC 2014, 2014, pp. 945–949, Doi: doi.org/10.1109/IMCCC.2014.199.

[171] C. Henning, "PROFINET for Network Geeks (and those who want to be)," JANUARY 14, 2014, 2014. [Online]. Retrieved: 29-Jun-2019 [From: https://us.profinet.com/profinet-network-geeks-want/].

[172] Z. Drias, A. Serhrouchni, and O. Vogel, "Taxonomy of attacks on industrial control protocols," Int. Conf. Protoc. Eng. ICPE 2015 Int. Conf. New Technol. Distrib. Syst. NTDS 2015 - Proc., 2015, Doi: doi.org/10.1109/NOTERE.2015.7293513.

[173] J. Åkerberg and M. Björkman, "Exploring security in PROFINET IO," in Proceedings - International Computer Software and Applications Conference, 2009, vol. 1, pp. 406–412, Doi: doi.org/10.1109/COMPSAC.2009.61.

[174] Tridium Europe Limited, "Tridium Niagara Framework Smart Buildings Guide Specification," West Sussex, 2017.

[175] A. Mirian et al., "An Internet-wide view of ICS devices," in 2016 14th Annual Conference on Privacy, Security and Trust, PST 2016, 2016, pp. 96–103, Doi: doi.org/10.1109/PST.2016.7906943.

[176] P. Zito, "What is Tridium, Part 1," Jul 3, 2017, 2017. [Online]. Retrieved: 29-Jun-2019 [From: http://buildingautomationmonthly.com/what-is-tridium/].

[177] Tridium, "Niagara AX 3 . 8u3 Features overview," 2017. [Online]. Retrieved: 29-Jun-2019. [From: https://bit.ly/2rhOx6R].

[178] B. Rios, "Owning a Building. Exploiting Access Control and Facility Management Systems," in Black Hat USA, 2014, p. 89. [Online]. Retrieved: 29-Jun-2019. [From: https://ubm.io/2YurdyO].

[179] A. M. Elshaer, M. M. Elrakaiby, and M. E. Harb, "Autonomous Car Implementation Based on CAN Bus Protocol for IoT Applications," in 2018 13th International Conference on Computer Engineering and Systems (ICCES), 2018, pp. 275–278, Doi: doi.org/10.1109/ICCES.2018.8639206.

[180] Z. King and S. Yu, "Investigating and securing communications in the Controller Area Network (CAN)," in 2017 International Conference on Computing, Networking and Communications (ICNC), 2017, pp. 814–818, Doi: doi.org/10.1109/ICCNC.2017.7876236.

[181] D. Beresford, "Exploiting Siemens Simatic S7 PLCs," in Black Hat USA, 2011, pp. 1–26. [Online]. Retrieved: 01-Jul-2019. [From: https://media.blackhat.com/bh-us-11/Beresford/BH_US11_Beresford_S7_PLCs_WP.pdf].

[182] Siemens, "S7-1500 - Industrial Ethernet CP 1543-1," Munich, Germany, 2013. [Online]. Retrieved: 01-Jul-2019. [From: https://support.industry.siemens.com/cs/attachments/76476576/GH_CP1543-1_76_en-US.pdf].

[183] D. Nardella, "Snap 7." Sourceforge, Bari, Italy, p. 1, 2018. [Online]. Retrieved: 01-Jul-2019. [From: http://snap7.sourceforge.net/].

[184] Siemens AG, "Security with SIMATIC controller," Munich, Germany, 2016. [Online]. Retrieved: 01-Jul-2019. [From: https://support.industry.siemens.com/cs/attachments/90885010/77431846_Security_SIMATIC_DOKU_V20_en.pdf].

[185] A. Timorin, "SCADA deep inside: protocols and security mechanisms," in Hacktivity, 2014, no. October 2014, p. 84. [Online]. Retrieved: 01-Jul-2019. [From: https://bit.ly/2YF0KPa].

[186] ODVA, "Securing EtherNet/IP Networks," Ann Arbor, Michigan, USA, 2016. [Online]. Retrieved: 03-Jul-2019. [From: https://bit.ly/2Ywb1go].

[187] F. Tacliad, T. D. Nguyen, and M. Gondree, "DoS Exploitation of Allen-Bradley's Legacy Protocol through Fuzz Testing," in CEUR Workshop Proceedings, 2017, vol. 2065, no. 2, pp. 54–57, Doi: doi.org/10.1145/nnnnnnn.nnnnnnn.

[188] Hart Communication Fundation, "Hart Communication. Application guide," Austin, TX 78759, 2013. [Online]. Retrieved: 03-Jul-2019. [From: https://www.fieldcommgroup.org/sites/default/files/technologies/hart/ApplicationGuide_r7.1.pdf].

[189] M. Duijsens, "WirelessHART: a security analysis," Eindhoven University of Technology, 2015. [Online]. Retrieved: 05-Jul-2019. [From: https://pdfs.semanticscholar.org/6d53/d09b602a6b315dc79bc746efa5bd4ba13e02.pdf].

[190] F. C. Group, "Digital Transformation in the age of IIoT," Austin, TX, 2016. [Online]. Retrieved: 05-Jul-2019. [From: https://bit.ly/36jjUga].

[191] S. Shah and P. Bhargava, "HART over IP for industrial automation networks," San Jose, CA, 2013. [Online]. Retrieved: 05-Jul-2019. [From: https://www.einfochips.com/images/in_sight/HART_Over_IP_EEIOL_2013JAN30_NET_TA_01.pdf].

[192] F. C. Group, "Hart Technology. Leading the digital transformation," Austin, Texas, 2017. [Online]. Retrieved: 05-Jul-2019. [From: https://www.fieldcommgroup.org/sites/default/files/technologies/hart/HART%20brochure%20web%20view.pdf].

[193] S. I. A. GmbH, "HART-IP solution communicates at Ethernet speed," Munich, Germany, 2015. [Online]. Retrieved: 05-Jul-2019. [From: https://industrial.softing.com/fileadmin/sof-files/pdf/de/ia/Articles/HART-IP_IEB-1502.pdf].

[194] A. Bolshev, "HART as an Attack Vector: from Current Loop to Application Layer," in S4x14, 2014, p. 40. [Online]. Retrieved: 15-Jul-2019. [From: https://documents.pub/document/hart-as-an-attack-vector-from-current-loop-to-application-layer.html].

[195] BACnet, "BACnet : Answers to Frequently Asked Questions," HPAC Heating/Piping/AirConditioning, no. March, pp. 47–51, 1997.

[196] R. Automation, "BACnet MS/TP Adapter," Milwaukee, 2006. [Online]. Retrieved: 15-Jul-2019. [From: https://bit.ly/38PkxiL].

[197] L. K. Haakenstad, "The Open Protocol Standard for Computerized Building Systems: BACnet," Proc. 1999 IEEE Int. Conf. Control Appl. (Cat. No.99CH36328), vol. 2, pp. 1585–1590, 1999, Doi: doi.org/10.1109/CCA.1999.801208.

[198] H. Merz, T. Hansemann, and C. Hübner, "BACnet BT - Building Automation: Communication systems with EIB/KNX, LON and BACnet," H. Merz, T. Hansemann, and C. Hübner, Eds. Cham: Springer International Publishing, 2018, pp. 209–302.

[199] B. Isler, "BACnet to Leverage IT. A Whitepaper on BACnet/IT," 2016. [Online]. Retrieved: 17-Jul-2019. [From: http://www.bacnet.org/Bibliography/BACnet_IT_WhitePaper_2016121.pdf].

[200] D. Robin et al. 2010. Ashrae Standard BACnet—A data communication protocol for building automation and control networks. Retrieved from http://www.bacnet.org/Addenda/Add-135-2008t.pdf.

[201] J. Kaur, J. Tonejc, S. Wendzel, and M. Meier. 2015. Securing BACnet's Pitfalls BT - ICT Systems security and privacy protection. In SEC 2015: ICT Systems Security and Privacy Protection. 616--629. Doi: doi.org/10.1007/978-3-319-18467-8_41.

[202] B. Bowers, "How to Own a Building BacNET Attack Framework," in Shmoocon, 2013, p. 1. [Online]. Retrieved: 17-Jul-2019. [https://archive.org/details/Shmoocon_2013_How_to_Own_a_Building_BacNET_Attack_Framework]
.

[203] Z. Pan, S. Hariri, and Y. Al-Nashif, "Anomaly Based Intrusion Detection for Building Automation and Control Networks," in 2014 IEEE/ACS 11th International Conference on Computer Systems and Applications (AICCSA), 2014, pp. 72–77.

[204] N. I. Corporation, "The Modbus Protocol In-Depth," Mar 19, 2019, 2017. [Online]. Retrieved: 17-Jul-2019. [From: http://www.ni.com/white-paper/52134/en/].

[205] M. Organization, "Modbus / TCP Security," Hopkinton, MA, 2018. [Online]. Retrieved: 19-Jul-2019. [From: http://modbus.org/docs/MB-TCP-Security-v21_2018-07-24.pdf].

[206] M. K. Ferst, H. F. M. de Figueiredo, G. Denardin, and J. Lopes, "Implementation of Secure Communication with Modbus and Transport Layer Security protocols," in 2018 13th IEEE International Conference on Industry Applications (INDUSCON), 2018, pp. 155–162, Doi: doi.org/10.1109/INDUSCON.2018.8627306.

[207] S. Figueroa-Lorenzo, J. Añorga, and S. Arrizabalaga, "A Role-Based Access Control Model in Modbus SCADA Systems. A Centralized Model Approach," Sensors , vol. 19, no. 20. 2019, Doi: doi.org/10.3390/s19204455.

[208] P. Huitsing, R. Chandia, M. Papa, and S. Shenoi, "Attack taxonomies for the Modbus protocols," Int. J. Crit. Infrastruct. Prot., vol. 1, no. C, pp. 37–44, 2008, Doi: doi.org/10.1016/j.ijcip.2008.08.003.

[209] B. Chen, N. Pattanaik, A. Goulart, K. L. Butler-purry, and D. Kundur, "Implementing attacks for modbus/TCP protocol in a real-time cyber physical system test bed," in 2015 IEEE International Workshop Technical Committee on Communications Quality and Reliability (CQR), 2015, pp. 1–6, Doi: doi.org/10.1109/CQR.2015.7129084.

[210] O. Foundation, "OPC Alarms and Events Custom Interface," Scottsdale, AZ, 2002. [Online]. Retrieved: 19-Jul-2019. [From: http://advosol.com/OpcSpecs/OPC%20AE%201.10%20Specification.pdf].

[211] OPC Foundation, "OPC Historical Data Access Specification," Scottsdale, AZ, 2003. [Online]. Retrieved: 19-Jul-2019. [From: http://advosol.com/OpcSpecs/OPC%20HDA%201.20%20Specification.pdf].

[212] O. Foundation, "OPC Overview," Scottsdale, AZ, 1998. [Online]. Retrieved: 25-Apr-2019. [From: https://invent.ge/2Pp0ek2].

[213] B. P. Hunkar, "OPC UA vs OPC Classic," Hudson, Ohio, 2017. [Online]. Retrieved: 25-Jul-2019. [From: http://www.dsinteroperability.com/OPCClassicVSUA.pdf].

[214] D. P. E. Byres, "OPC Security White Paper # 1 Understanding OPC and How it is Deployed," Llantzville, bc, 2009. [Online]. Retrieved: 25-Jul-2019. [From: http://www.opcti.com/opc-security-white-paper-1.aspx].

[215] B. Mcilvride and A. Thomas, "OPC Tunnelling – Know Your Options," Mississauga, ON, 2008. [Online]. Retrieved: 25-Jul-2019. [From: https://bit.ly/2RwlxCp].

[216] D. Kominek, "Effective OPC Security for Control Systems - Solutions you can bank on," Edmonton, Canada, 2011. [Online]. Retrieved: 25-Jul-2019. [https://bit.ly/37yZtgf].

[217] Microsoft, "How to configure RPC dynamic port allocation to work with firewalls," Apr 17, 2018, 2018. [Online]. Retrieved: 27-Jul-2019 [From: https://bit.ly/2GHS8ix].

[218] U. Steinkrauss. 2010. Whitepaper—Overview: OPC unified architecture. Technical overview and short description. Retrieved from http://www.ascolab.com/images/stories/ascolab/doc/ua_whitepaper_technicaloverview_e.pdf.

[219] J. Imtiaz and J. Jasperneite, "Scalability of OPC-UA down to the chip level enables 'internet of Things,'" in IEEE International Conference on Industrial Informatics (INDIN), 2013, pp. 500–505, Doi: doi.org/10.1109/INDIN.2013.6622935.

[220] M. D. Wolfgang Mahnke, Stefan-Helmut Leitner, OPC Unified Architecture, 1st ed. Springer-Verlag Berlin Heidelberg: Springer; 2009 edition (May 4, 2009), 2009.

[221] N. Pocock, D. Kominek, and P. Hunkar, "OPC UA Security - How It Works," in Microsoft Conference Center, 2014, p. 54.

[222] Bundesamt für Sicherheit in der Informationstechnik (BSI), "OPC UA Security Analysis," Bonn, Germany, 2017. [Online]. Retrieved: 27-Jul-2019. [From: https://opcfoundation.org/wp-content/uploads/2017/11/OPC-UA-Security-Advise-EN.pdf].

[223] C. Wester, N. Engelman, T. Smith, K. Odetunde, B. Anderson, and J. Reilly, "The role of the SCADA RTU in today's substation," in 2015 68th Annual Conference for Protective Relay Engineers, 2015, pp. 622–628, Doi: doi.org/10.1109/CPRE.2015.7102199.

[224] Ken Curtis, "A DNP3 Protocol Primer Introduction," Calgary, AB, 2005. [Online]. Retrieved: 27-Jul-2019. [From: https://www.academia.edu/35049898/A_DNP3_Protocol_Primer].

[225] R. Amoah, "Formal Security Analysis of the DNP3-Secure Authentication Protocol," Queensland University of Technology Australia, 2016.

[226] F. Cleveland. 2016. IEC TC57 WG15: IEC 62351 Security standards for the power system information infrastructure. Retrieved from http://iectc57.ucaiug.org/wg15public/Public%20Documents/White%20Paper%20on%20Security%20Standards%20in%20IEC%20TC57.pdf.

[227] G. Devarajan, "Unraveling SCADA Protocols: Using Sulley Fuzzer," in Def Con 15, 2007, p. 28. [Online]. Retrieved: 27-Apr-2019. [From: https://bit.ly/2PAhK54].

[228] C. S. Pe, A. Crain, C. Sistrunk, and A. Crain, "Master Serial Killer," in S4x14, 2014. [Online]. Retrieved: 27-Jul-2019. [From: http://blog.cci-es.org/2014/01/review-of-master-serial-killer-project.html].

[229] S. Bratus et al., "Implementing a vertically hardened DNP3 control stack for power applications," in S4x16, 2016, p. 9. Doi: doi.org/10.1145/3018981.3018985.

[230] IEEE, IEEE Standard for Local Area Network/Wide Area Network (LAN/WAN) Node Communication Protocol to Complement the Utility Industry End Device Data Tables, 5th ed., no. June. New York, NY: IEEE, 2012.

[231] A. F. Snyder and M. T. G. Stuber, "The ANSI C12 protocol suite - updated and now with network capabilities," 2007 Power Syst. Conf. Adv. Metering, Prot. Control. Commun. Distrib. Resour. PSC 2007, pp. 117–122, 2007, Doi: doi.org/10.1109/PSAMP.2007.4740906.

[232] C. Greer et al. 2014. NIST framework and roadmap for smart grid interoperability standards, release 3.0. Retrieved from https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.1108r3.pdf.

[233] I. T. Union, "X.237 bis," Geneva, Switzerland, 1999. [Online]. Retrieved: 29-Jul-2019. [From: https://bit.ly/2rrr96O].

[234] F. D. R. Inc, "Energy Communications Management Exchange AMI / SmartGrid Engineering and Consulting Services," June 30, 2017, 2017. [Online]. Retrieved: 29-Jul-2019. [From: https://www.ecmx.org/public/].

[235] K. Minematsu, S. Lucks, H. Morita, and T. Iwata, "Attacks and security proofs of EAX-prime," Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics), vol. 8424 LNCS, pp. 327–347, 2014, Doi: doi.org/10.1007/978-3-662-43933-3_17.

[236] Y. Chen, Z. Zhu, B. Xu, K. Fan, and K. Wang, "The use of IEC61850 for distribution automation," in 2016 China International Conference on Electricity Distribution (CICED), 2016, pp. 1–4, Doi: doi.org/10.1109/CICED.2016.7576251.

[237] Y. Liang and R. H. Campbell, "Understanding and Simulating the IEC 61850 Standard," Ieee Trans. Power Deliv., vol. 22, pp. 1482–1489, 2007.

[238] International Standard. 2003. IEC 61850-7-1: Communication networks and systems in substations–Part 7-1: Basic communication structure for substation and feeder equipment–Principles and models. Retrieved from https://webstore.iec.ch/p-preview/info_iec61850-7-1%7Bed1.0%7Den.pdf.

[239] S. Patel, "IEC-61850 Protocol Analysis and Online Intrusion Detection System for SCADA Networks using Machine Learning," University of Victoria, 2017.

[240] O. S. G. Platform, "Introduction to the Open Smart Grid Platform. IEC-61850," 20AD. [Online]. Retrieved: 29-Jul-2019 [From: http://documentation.opensmartgridplatform.org/Protocols/IEC61850/index.html].

[241] M. Strobel, N. Wiedermann, and C. Eckert, "Novel weaknesses in IEC 62351 protected Smart Grid control systems," 2016 IEEE Int. Conf. Smart Grid Commun. SmartGridComm 2016, pp. 266–270, 2016, Doi: doi.org/10.1109/SmartGridComm.2016.7778772.

[242] IEC 2006, "International Standard IEC 60870-5-104," vol. 2006–06, 2006, Doi: doi.org/IEC 61672-1.

[243] Y. Yang, K. McLaughlin, T. Littler, S. Sezer, B. Pranggono, and H. F. Wang, "Intrusion Detection System for IEC 60870-5-104 based SCADA networks," in 2013 IEEE Power & Energy Society General Meeting, 2013, pp. 1–5, Doi: doi.org/10.1109/PESMG.2013.6672100.

[244] J. T. Michalski, A. Lanzone, J. Trent, and S. Smith. 2007. Secure ICCP integration considerations and recommendations. Retrieved from https://www.energy.gov/sites/prod/files/oeprod/DocumentsandMedia/19-Secure_ICCP_Integration.pdf.

[245] QED Secure Solutions, "Risk Scoring System," 2019. [Online]. Retrieved: 01-Ago-2019. [From: https://www.riskscoringsystem.com/].

[246] A. Manion, "TEMSL," 2019. [Online]. Retrieved: 01-Ago-2019. [From: https://www.youtube.com/watch?v=-6cThOCm9co&feature=youtu.be&t=1303].

[247] C. Bodungen, "Industrial Vulnerability Scoring System (IVSS)," 2019. [Online]. Retrieved: 03-Ago-2019. [From: https://securingics.com/IVSS/IVSS.html].

[248] MITRE Corporation, "Common Vulnerabilities and Exposures (CVE)," March 20, 2019, 2018. [Online]. Retrieved: 03-Ago-2019 [From: https://cve.mitre.org/].

[249] MITRE Corporation, "Common Attack Pattern Enumeration and Classification (CAPEC)," April 04, 2019, 2018. [Online]. Retrieved: 03-Ago-2019. [From: https://capec.mitre.org/index.html].

[250] MITRE Corporation, "Common Weakness Enumeration: CWE," April 03, 2018 , 2018. [Online]. Retrieved: 05-Ago-2019. [From: https://cwe.mitre.org/index.html].

[251] MITRE Corporation, "Common Platform Enumeration: CPE Dictionary," 2018. [Online]. Retrieved: 05-Ago-2019. [From: https://cpe.mitre.org/].

[252] Offensive Security, "Offensive Security," 2017, 2019. [Online]. Retrieved: 07-Ago-2019. [From: https://www.offensive-security.com/].

[253] O. Security, "The Exploit Database - Offensive Security," April, 2019, 2019. [Online]. Retrieved: 07-Ago-2019 [From: https://www.exploit-db.com/].

[254] Micrososoft, "Security Bulletins," 10/11/2017, 2019. [Online]. Retrieved: 17-Ago-2019 [From: https://docs.microsoft.com/en-us/security-updates/SecurityBulletins/securitybulletins].

[255] National Institute of Standards and Technology (NIST), "NATIONAL VULNERABILITY DATABASE," 2015, 2018. [Online]. Retrieved: 17-Ago-2019 [From: https://nvd.nist.gov/vuln].

[256] A. Dulaunoy, "CVE-Search (Common Vulnerabilities and Exposures)." GitHub, Luxembourg, p. 1, 2018. [Online]. Retrieved: 17-Ago-2019 [From: https://www.cvedetails.com/].

[257] S. Özkan, "CVE-Details (Common Vulneravility Exposure)," 2012, 2018. [Online]. Retrieved: 17-Ago-2019 [From: www.cvedetails.com].

[258] R. Borgaonkar, "New vulnerabilities in 5G Security Architecture & Countermeasures," 2019. [Online]. Retrieved: 07-Dec-2019. [From: https://infosec.sintef.no/en/informasjonssikkerhet/2019/08/new-vulnerabilities-in-5g-security-architecture-countermeasures/].

[259] A. Shaik and R. Borgaonkar, "Black Hat 2019: 5G Security Flaw Allows MiTM, Targeted Attacks," 2019. [Online]. Retrieved: 02-Dec-2019. [From: https://blacklakesecurity.com/black-hat-2019-5g-security-flaw-allows-mitm-targeted-attacks/].

[260] 3GPP, "3GPP System Architecture Evolution (SAE); Security architecture," 2018. [Online]. Retrieved: 07-Dec-2019. [From: https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=2296].

[261] 3GPP, "Security architecture and procedures for 5G System. Technical Specification (TS) 33.501.," 2018. [Online]. Retrieved: 07-Dec-2019. [From: http: //www.3gpp.org/DynaReport/33501.htm].

[262] G. Association, "GSMA Mobile Security Hall of Fame," 2019. [Online]. Retrieved: 07-Dec-2019. [From: https://www.gsma.com/security/gsma-mobile-security-hall-of-fame/].

CHAPTER III

A ROLE-BASED ACCESS CONTROL MODEL IN MODBUS SCADA SYSTEMS. A CENTRALIZED MODEL APPROACH

This chapter was published in S. Figueroa-Lorenzo, J. Añorga, and S. Arrizabalaga, "A Role-Based Access Control Model in Modbus SCADA Systems. A Centralized Model Approach," Sensors, MDPI, vol. 19, no. 20, p. 4455, 2019, doi: doi.org/10.3390/s19204455. [JCR. 3.576, Q1].

This chapter introduces a role-based access control (RBAC) system, based on the security recommendations established by the Modbus Organization. In addition to the role-based authorization included in the X.509 certificate, the designed system promotes mTLS-based mutual authentication. Comprehensive performance testing demonstrates the viability of RBAC in a centralized environment.

1. ABSTRACT

Industrial Control Systems (ICS) and Supervisory Control systems and Data Acquisition (SCADA) networks implement industrial communication protocols to enable their operations. Modbus is an application protocol that allows communication between millions of automation devices. Unfortunately, Modbus lacks basic security mechanisms, and this leads to multiple vulnerabilities, due to both design and implementation. This issue enables certain types of attacks, for example, man in the middle attacks, eavesdropping attacks, and replay attack. The exploitation of such flaws may greatly influence companies and the general population, especially for attacks targeting critical infrastructural assets, such as power plants, water distribution and railway transportation systems. In order to provide security mechanisms to the protocol, the Modbus organization released security specifications, which provide robust protection through the blending of Transport Layer Security (TLS) with the traditional Modbus protocol. TLS will encapsulate Modbus packets to provide both authentication and message-integrity protection. The security features leverage X.509v3 digital certificates for authentication of the server and client. From the security specifications, this study addresses the security problems of the Modbus protocol, proposing a new secure version of a role-based access control model (RBAC), in order to authorize both the client on the server, as well as the Modbus frame. This model is divided into an authorization process via roles, which is inserted as an arbitrary extension in the certificate X.509v3 and the message authorization via unit id, a unique identifier used to authorize the Modbus frame. Our proposal is evaluated through two approaches: A security analysis and a performance analysis. The security analysis involves verifying the protocol's resistance to different types of attacks, as well as those certain pillars of cybersecurity, such as integrity and confidentiality, are not compromised. Finally, our performance analysis involves deploying our design over a testnet built on GNS3. This testnet has been designed based on an industrial security standard, such as IEC-62443, which divides the industrial network into levels. Then both the client and the server are deployed over this network in order to verify the feasibility of the proposal. For this purpose, different latencies measurements in industrial environments are used as a benchmark, which are matched against the latencies in our proposal for different cipher suites.

2. INTRODUCTION

Modbus is an application layer message exchange protocol, which provides client-server communication between devices connected on different sorts of buses or networks [1].

Modbus has been known as industry's serial de facto standard since 1979 and keeps on enabling millions of automation devices to communicate [2]. The Internet community can access Modbus at a reserved system port 502 on the TCP/IP stack. Modbus is a request/reply protocol, and offers services specified by function codes. The Modbus protocol allows an easy communication within all types of network architectures. Every type of device (PLC, HMI, Control Panel, Driver, Motion control, I/O Device, and so on) can use the Modbus protocol to initiate a remote operation [2].

The Figure 1 allows identifying action fields of Modbus based on the ISA 95 model and related standards. The Modbus protocol is part of the first two levels of this layered model. However, Modbus TCP/IP is mostly used in the data sharing between the field device level (e.g., PLC, CAN J1939 to the Modbus Gateway) and the SCADA system level. Although Modbus TCP/IP as a protocol could support communication between field devices via TCP, i.e., between sensors, actuators, and PLCs, at this point there is an additional requirement: The behavior as a real-time system (RTS). Real-Time support is a requirement mainly for devices at the field level (e.g., sensors and actuators).

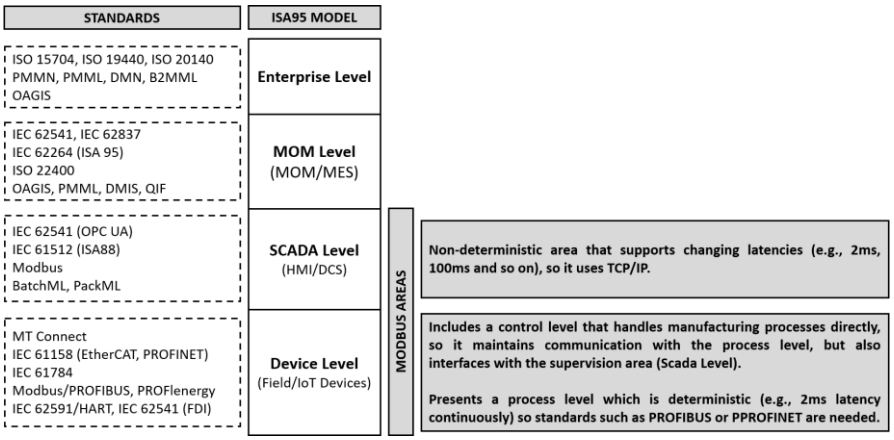

Figure 1: Action fields of Modbus based on the ISA 95 model and related standards [3], [4].

In an RTS, if the time constraints are not fulfilled, it can be said that the system has failed. As we have defined, the Modbus protocol TCP/IP implementation is in the application layer, therefore, considering the buffering problem, and given that the frames are queued FIFO (First Input First Output), unless a priority setting mechanism is used, as implemented by PROFINET RT, the process is not deterministic, and therefore is prone to introduce delay. In order to contextualize our work, Figure 2 shows an example of Modbus Network Architecture, where Modbus TCP/IP is not on the field device level, such as Modbus RTU (serial communication over RS-232, RS-485 and RS-422) or Modbus HDLC (MB+).

Therefore, Modbus TCP/IP is suitable to enable the communication, for example, between two gateways, an HMI with a PLC, a gateway with a HMI, or an Input/output (I/O) device without RTS requirements.

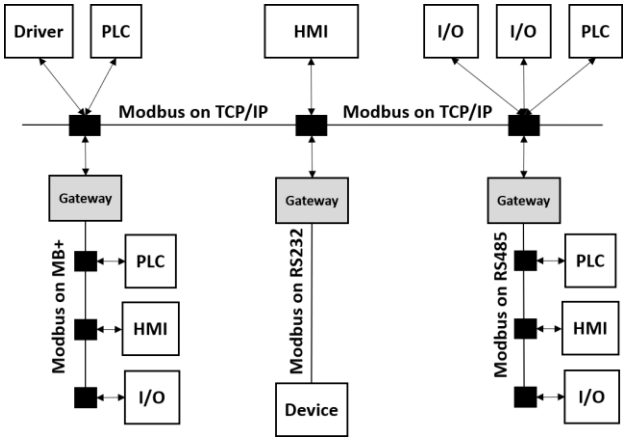

Figure 2: Example of Modbus Network Architecture [2].

Therefore, our work focuses on level two in Figure 1 the high usability of the Modbus protocol at this level, in both Operational Technologies (OT) environments and within industry 4.0 or the Industrial Internet of Things (IIoT), (due to its ability to integrate into industrial processes such as process automation, industrial automation, building automation, power system automation and automatic meter reading), makes the security the main concern of the Modbus protocol. In that sense, we can affirm that the Modbus TCP/IP security problem has its focus on the protocol design. Modbus RTU was scaled to Modbus TCP/IP because the controllers could manage the bandwidth more efficiently, where a client (e.g., a SCADA) could support the connection with multiple servers; however, the Modbus frame TCP/IP was scaled without considering security.

The Figure 3 helps to understand the statement. As is shown, by default, Modbus PDU includes the Function Code field and Data payload. This function code indicates to the server which kind of action to perform. All function codes are found in the specifications [2]. When the Modbus TCP/IP frame was defined, only the Slave ID field was changed to an Application Protocol (MBAP) header, and the check error field was removed. MBAP contains only seven bytes [2]. Therefore, the frame does not include any mechanism to provide authentication or access control. In addition, the default Modbus specification [2] does not include a mechanism to provide integrity or confidentiality, using, for instance, end-to-end encryption. Therefore, if we adopt the criteria shown in [5], where it is established that design vulnerabilities are inherent to a protocol specification, present even in perfect

implementations, we can confirm that Modbus security problems are related to the protocol design.

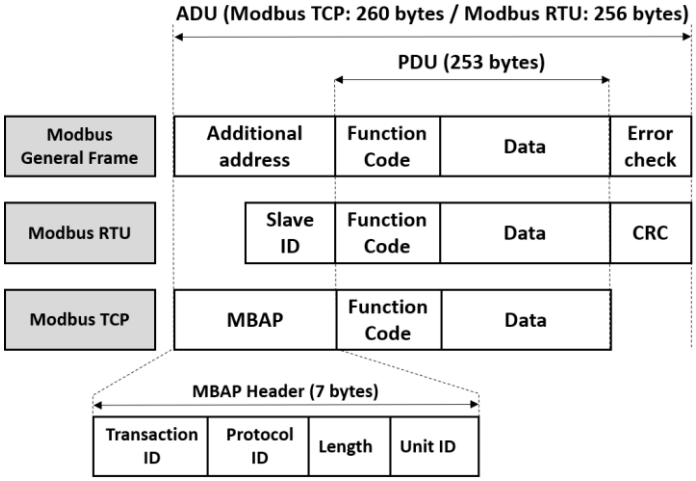

Figure 3: The Modbus frame [2].

However, until a few years ago it was not a problem since industrial networks (traditional OT) were isolated. Now, it is the age of new factory (the industry 4.0) floor platforms, new technologies such as OPC-UA ([6] enhances the value of OPC-UA for industry 4.0), new paradigms, such as the Internet of Things (IoT) or IIoT, and the integration between IT and OT environments. Security is no longer a privilege, it is a necessity, and therefore mandatory.

Despite this, several vulnerabilities are found in devices that support the Modbus protocol, which are classified into vulnerabilities by protocol implementation (i.e., exploitation on a specific device because of, e.g., firmware error) or by protocol design (i.e., exploitation on any device using the protocol).

For instance, in the CVE-2017-6819 vulnerability, during the communication between the operator and the PLC through the Modbus 0x5A function, it is possible for an attacker to send specially crafted packets to consume the PLC resources, and hence freeze it. The product affected was the Modicon M340 PLC. The next reference details the Schneider Electric Report [7].

In addition, at the Defconf security conference in 2018, a study was presented where an injection attack was made upon three types of PLC (one Modicon, another Allen-Bradley and the third based on an open-source PLC, known as OpenPLC [8]) which supported the Modbus protocol. To perform the injection attack, the same crafted frame was sent to each of the PLC, causing the same result: Denial-of-Service (DoS) [9].

In order to include security mechanisms to the protocol, in October 2018 the Modbus organization released security specifications [10], which provide robust protection through the blending of Transport Layer Security (TLS) [11] with the traditional Modbus protocol. TLS will encapsulate Modbus packets to provide both authentication and message-integrity protection. The security features leverage X.509v3 digital certificates for the authentication of both the server and the client. The protocol also supports the transmission of role-based access control (RBAC) information using an X.509v3 extension to authorize the request of the client.

Although the implementation of the system provides protocol security, authorized voices in the automation world argue that instead of securing Modbus, organizations should invest in technology to deploy a protocol that provides security by design, such as OPC-UA [12]. However, considering the number of devices supporting Modbus on the network, the option of providing security to the protocol is a solution that many organizations will adopt.

Although the specifications are a guideline to provide security for the Modbus protocol, they have a general approach to implementation, leading to new proposals. For this reason, the objective of our work is to improve the security of the Modbus protocol, based on the recommendations [10], i.e., our proposal is a way to contextualize the specifications [10] and to demonstrate the viability of it through both a security and a performance analysis. Within the implementation process, we introduce on the one hand the way in which the implementation is carried out, i.e., the authorization process, and on the other hand, the message authorization of the Modbus frame. Both points are not detailed in the specifications. To this end, this study proposes a role-based access control model (RBAC), which allows the server (e.g., PLC) to authorize a client (e.g., SCADA system) and that once this process has been carried out, and the Modbus frames flow through a secure channel, i.e., they are encrypted, they are also authorized. Since by default, TLS provides authentication of the server and client. Therefore, we are talking about an Authentication and Authorization (AA) model for the Modbus protocol. The authorization process is via a role-based access control. The roles are included as an arbitrary extension in the X.509v3 certificate and validated through a query from the server to a secure database, which has been populated by the client, e.g., the organization that own the SCADA, via an out-of-band (OOB) mechanism such as a secure web form. The authorization process takes place within the handshake phase of establishing a TLS connection, more precisely when the server receives the certificate from the client. Therefore, once this phase is over, any client-server communication will be secure, i.e., encrypted. At this point, the Modbus frame is also authorized, because the Modbus TPC frame contains a unique identifier (unit id) as part of

its header, and also since this is transmitted in a secure way. So, it can be used to authorize the frame, validated also via a query from the server to the secure database (the same used in the entity authorization process). In order to demonstrate the viability of our proposal, we provide both a security and performance analysis. The security analysis demonstrates the resistance of the proposal to different kinds of attacks. The performance analysis examines the latency behavior, not only for the cipher suites established by the security specifications ([10]), but also for others that involve a more complex processing and consequently higher latency measures. The remainder of the manuscript consists of related work in order to provide security to Modbus, followed by our implementation proposal. In addition, the corresponding security analysis is carried out. To evaluate our proposal, we will deploy our model on a GNS3 (Graphical Network Simulator-3) network built under the IEC-62443 standard that will allow traffic capture in a controlled environment. This represents our performance analysis. Finally, we provide the results obtained, the conclusions and the future research lines.

3. RELATED WORK

Several efforts have been made to provide security to the Modbus protocol. In order to establish a balance around the analyzed proposals, they are divided between offensive security proposals and defensive security proposals.

On the offensive security side, the first works analyzed was the reference [13]. It presents a formal model for evaluating the security of the Modbus protocol based on a formal demonstration of the existence of man-in-the-middle (MiTM) attacks in Modbus-based systems. An additional work analyzed is the reference [14] which adopts a penetration testing approach using a penetration-testing tool based on Intrusion Detection Systems (IDS) to examine the insider threat, as well as the external threat through internal and external penetration testing, respectively. The work presented by the reference [15] involves an automated tool to generate malicious SCADA Modbus traffic to be used to evaluate such systems. Additionally, the work [16], demonstrates the attacks to the authentication protocol initially presented by [17]. A deep analysis of the Modbus protocol specification in order to distinguish the possible attacks was presented by [18]. The same work ([18]) identifies several taxonomies, divided into the serial transmission mode and Modbus TCP/IP protocol. All of them consider the existence of a Modbus sniffer or a packet injector.

Other work analyzed was the reference [19]. It investigates the impact of malware attacks on Modbus-based SCADA networks, such as Code Red, Nimda, Slammer and Scalper. The authors also developed specialized malware to attack Modbus TCP/IP devices. One of them

performs DoS attacks to the SCADA system by injecting valid but malicious Modbus packages, consuming bandwidth without alarming a possible IDS system that monitors the network.

On the defensive security side, i.e., mechanisms for detecting and preventing attacks, we have also been able to find numerous works. For instance, the article [20], proposes a special smart fuzzing technology for Modbus TCP/IP which satisfies the requirement of the vulnerability detection for Modbus TCP/IP. In addition, an abnormal traffic detection mechanism by tracing Modbus TCP/IP transactions is proposed by [21]. The proposed method ([21]) enables a response immediate and fast, not only to Denial-of-Service (DoS) attacks, but also to various types of malfunctions, such as routing loops, misconfigured devices, and human mistakes. In addition, an authentication model, based upon the one-way property of cryptographic hash functions is proposed by [17]. Additionally, the article [22], investigates unauthorized, malicious, and suspicious SCADA Modbus activities by leveraging the Darknet address space in order to establish attack prevention models. A solution based on SCTP (Stream Control Transmission Protocol) and HMAC (Hash-based Message Authentication Code) named ModbusSec, is presented by the reference [23]. The SCTP is a transport layer protocol that provides a reliable message-oriented communication channel, with features such as congestion control and multi-homing. A new secure version based on the TLS protocol which addresses some security problems of the Modbus is proposed by [24]. The experimental results show that it is feasible to implement TLS by using it as a benchmark of power grid applications. Finally, the work addressed by the reference [25] is the only precedent of a role-based access control (RBAC) system for Modbus. Additionally, this reference performs a detailed review of the root causes of vulnerabilities in industrial environments. The access control is done on the client side, since they developed a security-hardened architecture for delivering enhanced security for SCADA remote terminal (RTU) devices, i.e., not focused on Modbus TCP. However, it is an interesting proposal, because in addition to presenting the access control approach, they protect the frames cryptographically and check for the existence of a CRC, even though, it does not check for a valid CRC.

From the related work, we can conclude on the one hand that there is evidence of several efforts made, both offensive and defensive security, in order to provide security to the Modbus protocol in the TCP/IP version. In particular, there are two works closely related to our approach, [24], which proposes an implementation of TLS for Modbus TCP/IP, and [25], which proposes an RBAC model for Modbus RTU. However, these two schemes, and in general the rest of the efforts, are outside the context of the specification [10], which is the starting point of our proposal. On the other hand, we can affirm that our proposal presents

novelty, because from the recommendations and guidelines provided by the specification [10], we propose a scheme that includes the following points. First, follows the recommendations of the specification, since it provides the implementation of an RBAC model, over a centralized system, which uses the X.509v3 certificate, for the server to authorize the client, i.e., the role of the client. All of the above procedure is performed within the framework of the TLSv1.3 handshake between the client and the server, which is not a condition established in the specification. Secondly, since the communication channel between the client and the server is protected after the handshake phase ends, i.e., encrypted, we perform the Modbus frame authorization from a unique identifier (unit id) found in the MBAP frame header (Figure 3). The second point of the proposal is not also addressed in the specification [10]. In order to demonstrate the viability of our proposal, our analysis includes two stages: Security analysis and performance analysis. The security analysis examines the resistance of the proposal to different kinds of attacks, such as eavesdropping, replay, forgery, and so on. The performance analysis verifies the implementation of the proposal, based on concrete and objective variables, which is measured with respect to changing conditions. These results are compared with established or adopted references. For our proposal, the variable is the latency; the changing conditions are the cypher suites, which, on one hand are defined by the specifications [10], and on the other hand, we propose to use other cipher suites which are more complex in terms of operations processing, i.e., resource consumption and the benchmarking are latencies and jitter of some industrial services.

4. PROPOSAL OF AN RBAC MODEL ON A CENTRALIZED ARCHITECTURE

Our proposal consists of a Role-based Access Control (RBAC) model, which is based on a centralized architecture. Figure 4 shows the general architecture of the proposal. The general architecture of the proposal is formed by five sub-systems: The client, the MBAPS handler, the MBAP handler, the AC module, and the Roles Database. It should be noted that the server is composed of three of the five sub-systems: The MBAPS handler, the MBAP handler and the AC module. MBAPS is the acronym for Modbus Application Protocol Secure. Below is a breakdown of how each of these entities interact as part of our proposal. The clients are sub-systems that send the connection request, e.g., a SCADA. Each client must store an X.509v3 certificate. The extension RoleSpecCertIdentifier has been added (Figure 4) to this certificate. We associate the OID ("1.3.6.1.4.50316.802.1") provided by the Modbus organization on the security specifications [10]. Additionally, our extension contains three fields. These fields are: roleName (e.g., operator), roleCertIssuer (e.g., client),

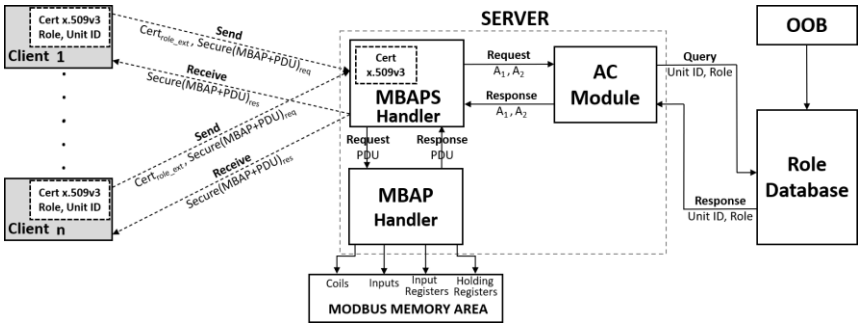

Figure 4: Role-based access control model (RBAC) based on centralized architecture.

roleCertSerialNumber (where we will store the Unit ID field). More details about the X.509v3 certificates can be found in the ITU recommendations [26]. The MBAPS is the entity or sub-system responsible for establishing the secure connection with the client, i.e., it is the entity that receives the client's secure connection request, authenticates the client via certified, as part of the mutual-authentication process of TLS; hence, it also needs direct communication with the AC module (Figure 4). Once the secure connection has been established and the frame has been authorized, the MBAPS handler interacts with the MBAP handler to which it sends the Modbus frame. Therefore, this module participates in both the authorization and authentication processes. The AC module is in charge of executing the policies found performing the corresponding verifications in the role database (Figure 4)The trigger to perform its functions is received from the MBAPS handler module, so in addition to the role database, the AC Module only interacts with this entity (MBAPS Handler).

The role database is a very simple entity whose interaction is based on one side with an out-of-band (OOB) mechanism through which the client populates the database before the connection request, and on the other side, with the AC module, which queries the stored data. This data enables one to perform access control policies (Figure 4). The MBAP handler module is the entity that communicates directly with the Modbus Memory Area (Figure 4) and for this, it must receive the Modbus frames from the MBAPS handler. Once the frames have been received, this module separates them according to their function code.

These entities interact as part of the RBAC model, which is composed of two authorization phases. The first uses the role extension of the client's certificate to authorize it within the TLS handshake stage, and the second authorization phase enables each of the Modbus frames based on the unit_id. Additionally, as part of the TLS handshake, it executes an authentication process of the client and server entities.

Therefore, the next section analyzes both the authorization and authentication phases as part of the handshake. In addition, the section 4.2 analyzes the second authorization phase.

4.1. AUTHENTICATION PHASE VIA TLS AND ENTITY AUTHORIZATION PHASE VIA ROLE ON X.509V3 CERTIFICATE

As Figure 5 illustrates, the first step in the process is to populate the Role Database. We assume that there is an OOB mechanism (e.g., via a secure web form). The interaction of the role database with the OOB system is shown in Figure 4. Through this system, the client will be able to insert the Role that they will have in the server, as well as Unit ID that the

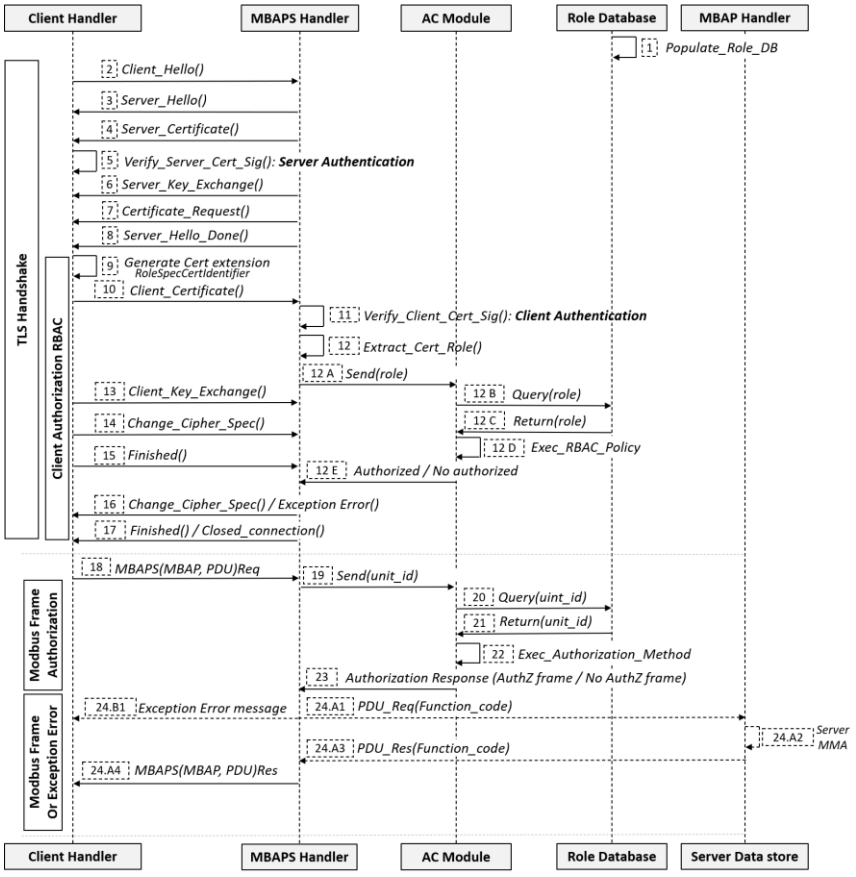

Figure 5: Sequence Diagram of Modbus Transport Layer Security (TLS) handshake and RBAC client authorization.

Modbus frames will have in their header. The client then sends a request to establish a secure connection to the server. Until step 8 in Figure 5 a normal handshake, this process is performed between a client and a server as part of a TLS implementation. However, this normal handshake includes an important feature of TLS, the first step of the mutual authentication, i.e., the server authentication through the verification of the server certificate

(step 5). In step 9, the client adds an extension with the corresponding role to its certificate, as mentioned in the previous section, before sharing it with the server. All of this as part of the TLS session management.

Once, the MBAPS handler has received the X.509v3 certificate (step 10), it verifies the certified (step 11). This process is the second step of the mutual authentication process, i.e., client authentication (step 11). In addition, the server extracts from it the role; sends it to the AC module, which queries the Roles Database and executes the RBAC policy (step 12 A–step 12 D). This RBAC policy validates that the role stored in the database coincides with the one included in the certificate. Without knowing if their role will be authorized, the client sends relevant information to the server (steps 13–15). These steps are also common within the TLS session generation process. Until the server receives the "Finished" message from the client (step 15), the MBAPS handler will retain the authorization or non-authorization response received from the AC Module. If the role is authorized, the MBAPS handler sends the messages to confirm the cipher specifications and finishes, which steps are also regular in the generation of a TLS session. If the role has not been authorized, the handshake phase ends when the MBAPS handler sends an exception error message to the client and closes the connection.

4.2. MESSAGE AUTHORIZATION PHASE

Once the authorization phase between client and server is over, frames can be exchanged securely, using symmetric encryption. These frames will contain the traditional Modbus frames (MBAP + PDU (Figure 3)). In the Figure 5, when the MBAPS Handler receives the frame request (MBAP + PDU) (step 18), it extracts the unit_id from the MBAP header and sends it to the AC module (step 19), which queries the Role Database (steps 20–21) and executes the authorization method. The authorization method validates that the unit_id stored in the database matches with the one included in the MBAP header field (step 22). Then the AC module responds by authorizing (or not) the frame (step 23), from which MBAPS handler sends an exception, error message to the client (step 24.B1) or sends the frame to the MBAP handler (step 24.A1) that processes the frame from the function code and interacts with the four areas of the Modbus Memory Area (MMA). The Figure 4 shows the four areas of the MMA.

5. IMPLEMENTATION PHASE

From the sequence diagram of the Figure 5, in the Figure 6 we divided into logical actions these design phases in order to simplify the implementation phase. There are three

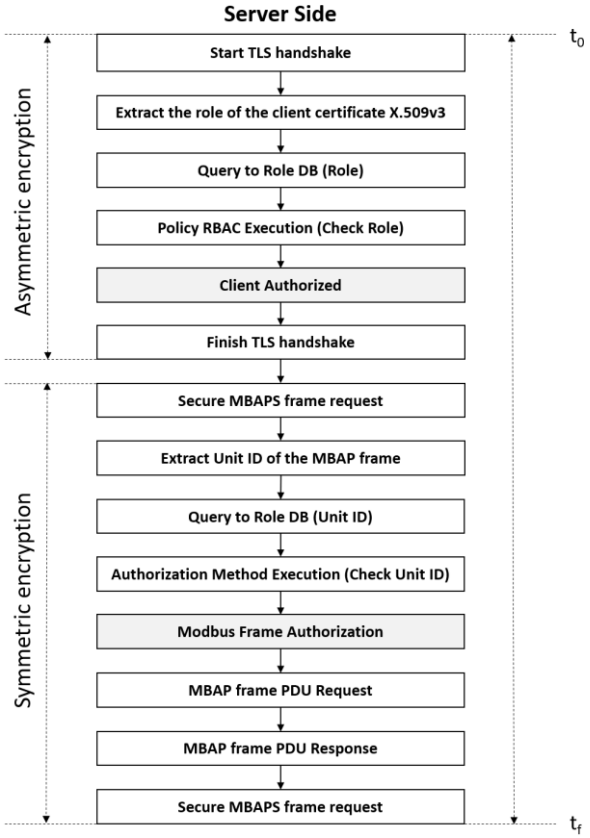

Figure 6: Step-by-step, successful server-side; the two-authorization process.

challenges at this point: (1) Select the base library (core library) on which to implement, (2) analyze the mechanism to implement encryption and to enable other security features and (3) analyze the cipher suites to implement.

Our Zenodo reference contains the implementation details that include the Role Database, the arbitrary extension configuration file, and the video demonstration [27].

5.1. SELECTION OF THE CORE LIBRARY

In order to accomplish our implementation, we used the open-source library, supported by the Community pymodbus [28]. Although as part of the evaluation phase is shown through performance analysis details of latencies between client and server, without any encryption, i.e., using the unmodified pymodbus library, at this point we can mention that these measurements of latencies was the first criterion used for the selection of the library (see column TCP of both tables of section 7.1). The second criterion considered was the language

in which it was written: Python; and in third place was analyzed the broad community that has pymodbus, hence, the troubleshooting is simples.

5.2. MECHANISM TO IMPLEMENT THE ENCRYPTION, AND OTHER SECURITY REQUIREMENTS

Once the core library was selected, we proceeded to implement each of the steps shown in Figure 6, where we show the followed server-side procedure. In order to generate the handshake session a TLS socket was used. Over this socket, the client sends the certificate with the added role extension. Next, the role of the client's certificate is extracted, the database is queried to recover the associated role to the specific OID, the policy is executed, i.e., it is checked that the role stored in the database, as well as the extracted role from the client certificate, do in fact match, determining the client's authorization. At this point, the handshake stage and the asymmetric encryption stage are completed.

Therefore, when a Modbus frame is received, the ID unit is extracted from the header, the database is queried again, and the authorization method is executed, i.e., the unit id associated with the OID of the certificate of the client is verified to match with the frame. Finally, the frame authorization is determined, allowing the core library to perform the function code over the Modbus Memory Area (MMA).

Next, we detail the three most important points during the implementation stage: (1) The library used to generate the TLS socket, (2) the certificate generation process and (3) the database where the roles are stored.

The implementation of TLS was based on the recommendations provided by Python Software Foundation (PSF) from the wrap_socket handler [29]. This module provides access to TLS (a.k.a., "Secure Sockets Layer") encryption and peer authentication facilities for network sockets, both client-side and server-side. This module uses the OpenSSL library. This library supports the use of TLSv1.3, and the management of X.509v3 certificates, where both are essential requirements in our implementation. Another advantage of the library is that it presents the SSLContext.set_ciphers() method, which enables the efficient management of the cipher suites to be selected. The SSL module disables certain weak ciphers by default, but it is possible to restrict the choice of ciphers even further. The following subsection analyzes the used encryption suites.

In order to generate an arbitrary extension, i.e., an extension with a custom OID, a configuration file (openssl.conf) is previously generated, and it is loaded in the command line to generate the certificate. The expression (1) shows the command used to generate the certificate with the added extension. This extension is contained in the openssl.conf

configuration file. For more detail, our Zenodo reference contains these implementation details.

Finally, the Role DB database has been implemented using SQLite database. For instance, expression (2) is the simple instruction used to create the Table over our "RoleDB.db" to perform the proof of concept (PoC). It is composed of a table with the attributes: Role, oid, and unit_id, which are gotten from the same client, and are used to authorize the client and to authorize the Modbus frame, respectively.

```
openssl req -newkey rsa:2048 -nodes -keyout {0} -extensions RoleSpecCertIdentifier -out {1}
-subj "/C=NA/ST=NA/L=NA/O=OT/CN={2}/description={3}" -config ./openssl.cnf          (1)
```

```
CREATE TABLE roleTable (ID INTEGER PRIMARY KEY, unit_id INTEGER, role
text, oid STRING),                                                                  (2)
```

5.3. CIPHER SUITES TO IMPLEMENT

The specifications [10] set the lowest boundary of the cipher suites to be used, in terms of resource consumption due to processing, while maintaining minimum security levels. However, as cipher suites will be used as a resource in our evaluation phase for performance analysis, we have decided to perform an analysis of other cipher suites and to establish a comparative analysis with respect to those defined by the specification [10]. The cipher suite represents which cryptographic algorithms and methods should be used, and it is defined by [30]. Currently, there are 339 suites officially supported, that target different applications and security levels. The cipher suite name contains the key exchange, the authentication method, the key exchange algorithm, and the symmetric algorithm for an authenticated encryption of application data transfer between entities.

For instance, TLS_ECDH_ECDSA_WITH_AES_128_CBC_SHA (0xC004) states that the Elliptic-curve Diffie-Hellman (ECDH) will be used as key exchange, while Elliptic Curve Digital Signature Algorithm (ECDSA) will be used as digital signature, and AES_CBC_SHA will be used to construct the symmetric authenticated encryption (Advanced Encryption Standard (AES, a.k.a. Rijndael) with Cipher Block Chaining (CBC) mode for encryption and HMAC with SHA for the construction of the Message Authentication Code). Table 1 sets the encryption suites recommended by the specifications [10] and Table 2 sets the cipher suites that are additionally sampled. As it can be noticed when comparing both tables, the cipher suites proposed in Table 2 are more complex than those proposed in Table 1, i.e., in terms of the computational cost required to implement each. However, in order to select the

suites that we have proposed in Table 2, we have examined the corresponding relationship with those presented in Table 1, with the objective of stressing the system and obtaining measures of higher latencies. For instance, if we compare 0xC02B with 0xC02C, a similar structure is visible in terms of the key exchange, the authentication method, and the symmetric algorithm construction for encryption, however, on the one hand, 0xC02B uses AES_128_GCM while 0xC02C uses AES_256_GCM, and on the other hand, 0xC02B uses SHA256 while 0xC02C uses SHA384.

Table 1: Cipher suites defined by the specification [10].

Cipher Mode	Cipher Suite	Number [1]
Null	TLS_RSA_WITH_NULL_SHA256	0x003B
CBC	TLS_RSA_WITH_AES_128_CBC_SHA256	0x003C
GCM	TLS_RSA_WITH_AES_128_GCM_SHA256	0x009C
	TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256	0xC02B

[1] IANA Format (0xC0, 0x2D) equal to Number (0xC02D).

Table 2: Cipher suites used as additional samples.

Cipher Mode	Cipher Suite	Number [1]
Null	TLS_RSA_WITH_AES_256_CBC_SHA	0x0035
CBC	TLS_RSA_WITH_AES_256_CBC_SHA256	0x003D
GCM	TLS_RSA_WITH_AES_256_GCM_SHA384	0x009D
	TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384	0xC02C

[1] IANA Format (0xC0, 0x2D) equal to Number (0xC02D).

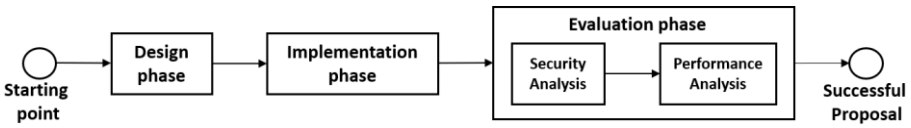

Figure 7: Proposal roadmap.

6. EVALUATION PHASE

At this point, we consider it important to analyze the phases of the proposal in order to understand the evaluation phase. The Figure 7 contains the roadmap followed. Once the design and implementation phase has been completed, we will divide the evaluation phase into two stages. The first, corresponding to the next section, exposes our proposal to a security analysis, while the second corresponds to the performance analysis of the proposal. The purpose of the evaluation phase is to determine whether the proposal is feasible.

6.1. SECURITY ANALYSIS

The security analysis of a system traditionally starts with the definition of the attacker's model. Since we do not strictly propose a protocol, but rather we base our analysis upon the improvements proposed by the specifications [10], we have divided our security analysis into some issues of an attacker model, as well as the analysis with respect to other parameters and the pillars of cybersecurity. At this point we should mention that in our demonstration the database is local. However, in the specification [10], it is indicated that it can be external to the server, so in that case it would be secured via a TLS socket. The attacker model used is the classic Dolev-Yao [31]. The following analysis is a consequence of the analysis of the exchange of messages between a client and a server through the network.

1. Mutual Entity Authentication: Our proposal contains one mechanism where authentication is implemented. It is the TLS authentication proper, which happens when the endpoints verify the validity of the certificates. These certificates have been previously installed, i.e., they are factory-installed.

2. Confidentiality and Message Authentication: As the recommendations [10] indicate, providing security to MBAP through the implementation of TLS via the construction of symmetric encryption, i.e., encryption + MAC, a.k.a., authenticated encryption (e.g., in Table 1, AES 128 CBC will be used for encryption) ensures the confidentiality of the information.

3. Integrity: TLS provides a security-focused protocol alternative to MBAP (to see MBAP sub-system in Figure 4) by adding data integrity via certificates, and authorization via information embedded in the certificate, such as user and device roles. So, the integrity is provided by on one hand the public-key cryptography in the TLS handshake process, and on the other hand by symmetric encryption (encryption + MAC).

4. Replay attack: TLS properties guarantee freshness, so TLS protects against replay attacks. In addition, an attacker cannot replay a message that has been logged in previous sessions, because both the oid and the unit id change in every session. Moreover, it must be considered that the generation of the oid, the role and the unit id is performed through an OOB mechanism (Figure 4).

5. Man-in-the-Middle (MiTM) attacks: Because all of our client-server communications are encrypted through TLS, mutual entity authentication is required before performing a transaction. Mutual entity authentication exists, because according to TLS, it requires that each endpoint send its domain certificate chain to the remote endpoint (step 5 and

step 11 of Figure 5) Subsequent communication between entities over authenticated encryption (symmetric encryption algorithms) also provides a MAC algorithm, which protects the communication against MiTM attacks.

6. Eavesdropping attack: Thanks to the implementation of TLS, our proposal guarantees the confidentiality and integrity of the information. For this reason, we consider our proposal resistant to any eavesdropping attack.

6.1.1. Implementation test

Summarizing, the demonstration of resistance to the attacks mentioned above, as well as the guarantee that the pillars of cybersecurity are not compromised, is because TLS, by default, provides security to MBAP by adding data confidentiality, data integrity, anti-replay protection, end-point authentication via certificates and authorization via information embedded in the certificate, such as user and device roles. Additionally, we provide a mechanism to authenticate the Modbus frame.

Therefore, from a design point of view, we can justify our analysis, which means that if there was a failure it would be due to the implementation. For this reason, we carry out a new demonstration by using a framework to measure the quality of the software.

For Python, there is a well-known library called Pytest. Pytest automatically catches warnings during test execution. In addition, pytest can be synchronized with Allure. Allure Framework is a flexible lightweight multi-language test report tool that not only shows a very concise

```
# ---------------------------------------------------------------------#
# Test TLS Client
# ---------------------------------------------------------------------#

def testSyncTLSClientInstantiation(self):
    client = ModbusTLSClient()
    self.assertNotEqual(client, None)

def testBasicSyncTLSClient(self):
    ''' Test the basic methods for the TLS sync client'''

    # receive/send
    client = ModbusTLSClient()
    client.socket = mockSocket()
    self.assertEqual(0, client._send(None))
    self.assertEqual(1, client._send(b'\x00'))
    self.assertEqual(b'\x00', client._recv(1))

    # connect/disconnect
    self.assertTrue(client.connect())
    client.close()

    # already closed socket
    client.socket = False
    client.close()

    self.assertEqual("ModbusTLSClient(192.168.127.153:802)", str(client))
```

Figure 8: Test TLS client via the Pytest module.

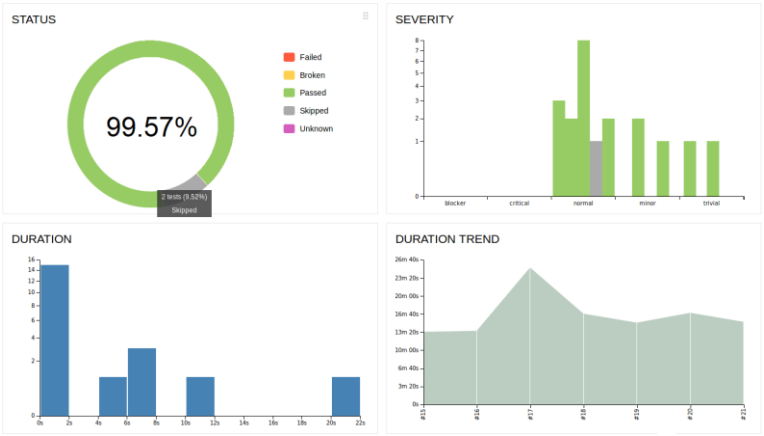

Figure 9: Report sample generated through Allure.

representation of what has been tested in a neat web report form, but also allows everyone participating in the development process to extract a maximum of useful information from the everyday execution of tests. Therefore, Allure-Pytest is our Software Quality Framework. Figure 8 contains the most basic part of the test we apply to TLS, where (the client's ip that appears in Figure) matches the client's ip of the evaluation architecture, which will be analyzed in the section 6.2.1. Multiple tests have been carried out, including tests for connection and disconnection, data transmission and data receipt. The Figure 9 shows a part of the report that is generated through Allure, where it is possible to monitor the number of tests performed and their success rate. Although there are more rigorous frameworks, Pytest gives a good measure of the correct implementation of our security proposal.

6.2. PERFORMANCE ANALYSIS

The performance analysis is given by the deployment of an evaluation architecture, from a test-network, which is built on GNS3 and is based on the industrial standard IEC-62443, which divides the network into levels. Both the client and the server are deployed on this architecture in order to measure the latencies generated for different cipher suites. The results of these measurements are compared with the time constraints collected for several industrial services. Therefore, each of the sections below are presented according to the logical order mentioned, i.e., (1) evaluation architecture, (2) tool for measuring the latency, (3) test scenarios to be executed and (4) time constraints in industrial services.

6.2.1. Evaluation architecture

In order to evaluate our proposals, models that use an in-depth defense approach are used, aligned with industrial security standards such as IEC-62443 ICS Security and NIST 800-82

Industrial Control System (ICS) security [32]. IEC-62443 divides the network of an industrial environment into levels. Based on the application environment defined in Section 2 we consider that our implementation should be based upon levels one and two of the Purdue Model. Level 1 contains all of the controlling equipment. The main purpose of the devices (e.g., our server) in this level is to open valves, move actuators, start motors, and so on. Typically, in Level 1 we find devices such as PLCs, Variable Frequency Drives (VFDs), dedicated proportional-integral-derivative (PID) controllers, and so on [33]. In addition, Level 2 specifies parts of the system to be monitored and managed with HMI systems (e.g., our client), which allows to start or stop the machine and see some basic running values and manipulate machine specific thresholds and set points [33].

According to this requirement we generate a testnet in GNS3, which includes a firewall (ASA 5505), switches L2 (cisco 2960), routers (cisco 1941) and Docker containers to simulate the Modbus TLS Client and the Modbus TLS Server. The Cisco Adaptive Security Appliance (ASA) is an advanced network security device that integrates a stateful firewall, VPN, and other capabilities. This testnet employs an ASA 5505 to create a firewall and protect an internal industrial network from external intruders, while allowing internal hosts' access to the Internet.

The ASA creates three security interfaces: Outside, Inside, and DMZ. It provides Outside users limited access to the DMZ, and no access to inside resources. Inside users can access the DMZ and outside resources. For this reason, the Modbus TLS client has access to the Modbus TLS server.

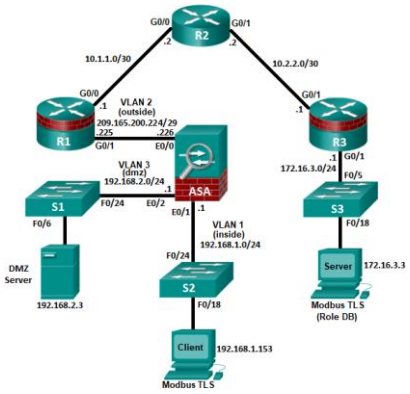

Figure 10: Diagram of the architecture deployed on GNS3.

Testnet configurations include NAT, VLAN and Access Control over the ASA 5505. We share all configurations (ASA 5505, R1, R2, and R3) as an additional resource using a Zenodo reference [27]. The architecture designed is shown in the Figure 10.

6.2.2. Tool for measuring latencies

Once the architecture is deployed and considering that our development was based on python, we use the *time* module. The python docs say that clock should be used for benchmarking. In the time module, there are two timing functions: These are time and clock. In **expression (4)**, the variable t_f is elapsed CPU seconds since t_0 was started (**expression (3)**). By moving the start and end within the code we can obtain the results of interest. We have recorded the latency times for the client, i.e., when the client requests a function (e.g., read coils), this function is executed in the server and until the response is received.

$$t_0 = \text{time.clock()}, \quad \textbf{(3)}$$

$$t_f = \text{time.clock()} - t_0, \quad \textbf{(4)}$$

6.2.3. Test scenarios for performance evaluation

On the one hand, the Modbus specifications [2] list the set of functions that are implemented, and additionally describes therefore the specific functions to be tested in the PI-MBUS-300 guidelines [34]. These three specific functions should be executed over the maximum number of coil (65535), or register (123) allowed by the protocol. These three functions are: Read Coils (0x01), Read Holding Registers (0x03) and Read-Write Multiple Register (0x17).

On the other hand, in order to evaluate the contribution of the RBAC model to the latencies, the following two approaches have been tested. First, Modbus over TLS, but without the RBAC model, is used to perform all of the measurements. This means that some steps that were included in **Figure 6** are skipped for this first approach. More specifically, in the first phase of the extraction of roles from the certificate, the database query and the execution of the RBAC policy are removed, while in the second phase the extraction of the unit ID, the database query, the authorization mechanism and the authorization response, are removed. These results will be analyzed in **section 7.1**.

In the second approach to be tested, all the steps also including the RBAC model (described in **Figure 6**) are executed at each test. These results will be analyzed in **section 7.2**, and as it has more steps than the measurements without RBAC policy (approach 1), it is expected that obtained latency measurements will be higher in the second approach.

In both approaches, there are several options for selecting the cipher suite to be used, as established in the specifications [10] (see **Table 1** and **Table 2**). Hence, all of these cipher

suites will also be tested in order to analyze the contribution to the final latencies in both approaches. Additionally, as we are based on Modbus TCP and it is nondeterministic, each test will be repeated 1,000 times. The Table 3 summarizes all the tests that have been carried out.

Table 3: Total samples analyzed.

Tested Approach	Tested Cipher Suites	Tested Modbus Functions	Number of Tests
Without RBAC	9	3	27000
With RBAC	8	3	24000
Total	17	6	51000

As an example, the implementation used to evaluate the performance using the Read Holding function to the TCP samples is shown in the Figure 11. In order to execute the function on TLS, instead of using the ModbusTcpClient function, the ModbusTLSClient function is used. The cycle's variable is in charge of monitoring the number of samples that are collected (to consider that the client's ip, Figure 11, also matches the client's ip of the evaluation architecture, Figure 10).

```
# ------------------------------------------------------------------------- #
# Modify the parameters below to control how we are testing the client:
#
# * workers - the number of workers to use at once
# * cycles  - the total number of requests to send
# * host    - the host to send the requests to
# ------------------------------------------------------------------------- #
workers = 10
cycles = 1000
host = '192.168.127.153'

def single_client_test(host, cycles):
    """
    :param host: The host to connect to
    :param cycles: The number of iterations to perform
    """
    logger = log_to_stderr()
    logger.setLevel(logging.DEBUG)
    logger.debug("starting worker: %d" % os.getpid())

    try:
        count = 0
        #client = ModbusTLSClient(host, port=802)
        client = ModbusTcpClient(host, port=502)
        #client = ModbusSerialClient(method="tcp",
        #                            port="/dev/ttyp0", baudrate=9600)
        while count < cycles:
            with _thread_lock:
                client.read_holding_registers(10, 1, unit=1).registers[0]
                count += 1
    except:
        logger.exception("failed to run test successfully")
    logger.debug("finished worker: %d" % os.getpid())

# ------------------------------------------------------------------------- #
```

Figure 11: Read Holding function used to evaluate the performance.

Finally, it is necessary to specify the measurement process with regard to the TLS session and the number of times that a function will be executed per session, since this is a requirement that we have established. As we mentioned before, the Figure 6 illustrates how for a TLS session, the process by which a single function (e.g., read coil (0x01)) is executed

once, is composed of two phases. A phase where the TLS session is established (asymmetric encryption, Figure 6) and another where the Modbus frame is authorized, as well as the interaction between the MBAP handler and the MMA, i.e., the execution of the read coil function (0x01), under symmetric encryption, i.e., encryption + HMAC (Figure 6). Although it would be possible to execute many functions within the same TLS session, in the tests only one function is executed: In order to obtain the worst-case latency, we have generated a new session for each one of the functions to be executed.

6.2.4. Latencies benchmarking

Once the measurements are carried out, we set latency constraints in order to reach conclusions about proposals feasibility. In order to achieve this, we take as a reference an ITU appendix: "Technical and operational aspects of Internet of Things and Machine-to-Machine applications by systems in the Mobile Service" [35]. Table 4 shows the latencies and jitter of some industrial services. ITU defines latency such as a "parameter for characterizing the communication service delay from an application point of view". In addition, it defines jitter, such as "variation of latency".

Table 4: End-to-end latency constraints [35].

Service	End-to-End Latency	Jitter
Factory automation (motion control)	1 ms	1 µs
Factory automation	10 ms	100 µs
Process automation (remote control)	50 ms	20 ms
Process automation (monitoring)	50 ms	20 ms
Electricity distribution (medium voltage)	25 ms	10 ms
Electricity distribution (high voltage)	5 ms	1 ms
Intelligent transport systems (infrastructure backhaul)	10 ms	2 ms
Remote Control	5 ms	1 ms

Table 5: Modbus use cases.

Service	Reference
Process automation	[36]
Industrial automation	[34]
Building automation	[37]
Power System Automation	[38]
Automatic Meter Reading	[39]

It is necessary to emphasize firstly that most of the services listed in Table 4 are deterministic in nature, and as we have mentioned in the introduction section, our work focuses on Modbus TCP/IP, which is non-deterministic. However, we will adopt these values as latency constraints to be fulfilled. Secondly, it is important to remark that Modbus can be used in

more application areas (see Table 5) than those identified in Table 4. Our proposal will be feasible if it finally fulfills the latency constraints defined for the specific use case, which might be even higher that those presented in Table 4.

7. RESULTS

Although security analysis has determined that our proposal is resistant to different attacks and that pillars of cybersecurity have not been compromised, it is also relevant to analyze its performance in order to assure the feasibility of our proposal (see Figure 7). For this reason, this section discusses the results of the tests described in Section 6.2.3.

As a reminder, the first section includes the performance analysis for the tests carried out when using different cipher suites but without applying RBAC model, for the different Modbus functions.

The second section includes the performance analysis for the tests carried out when applying RBAC model, comparing also results for different cipher suites and different Modbus functions. In both approaches, each test was repeated 1,000 times.

7.1. COMPARISON BETWEEN CIPHER SUITES LATENCIES WITHOUT APPLYING RBAC MODEL

Table 6 shows the average and standard deviation of the 1,000 measurements of latencies obtained for each combination of cipher suite and Modbus function.

As shown in Table 6 the fastest tested null-encryption suite with the secure hash function, TLS-RSA-WITH-NULL_SHA256 (0x003B), is in average 0.09 ms (Read Coils), 0.14 ms (Read Holding Registers) and 0.18 ms (Read-Write multiple Registers) faster than the lower latency encrypted option (0x0035) which implements TLS-RSA-WITH-AES_256_CBC_SHA256. In addition, the average latency of the former cipher suite (0x003B), is 0.44 ms (Read Coils), 0.43 ms (Read Holding Registers) and 0.55 ms (Read-Write multiple Registers) slower than the insecure Modbus (Modbus TCP). This shows that the highest latency is provided by asymmetric encryption implemented by the TLS handshake.

Although it is difficult to establish a comparison between CBC and Galois/Counter Mode (GCM) with these data, given that our approach depends on the requirements of the specification [10], GCM beats CBC categorically. For instance, Table 6 contains the encryption suite (0x003D), while Table 7 contains the encryption suite (0x009D). Clearly, it is observed that despite presenting a more complex hash function the latencies of 0x009D

are lower. Table 7 shows that despite the high requirements (AES_256) and (SHA384), TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 would be faster than either of those, use a lot less bandwidth, and be more secure.

Standard deviation is also calculated, considering that the process that handled the communication is not deterministic. If it is taken into consideration that a low standard deviation indicates that the data points tend to be close to the mean of the set, while a high standard deviation indicates that the data points are spread out over a wider range of values. From Table 6 and Table 7 we conclude that as the operations complexity tends to increase, the standard deviation increases and therefore the latency values tend to be further away from the mean. For instance, if the cipher suite 0x009C is compared with 0x009D, both share operations, but the operations from 0x009D are more complex. This situation is reflected also in each value of the standard deviation.

Table 6: Average latencies (ms) and standard deviation in Cipher Block Chaining (CBC) cipher mode (without RBAC model).

	Read Coils (0x01)		Read Holding Registers (0x03)		Read-Write Multiple Register (0x17)	
Cipher Suite Code	**Ave.**	**Std.**	**Ave.**	**Std.**	**Ave.**	**Std.**
TCP	0.93	0.56	1.03	0.61	1.13	0.72
0x003B	1.37	0.73	1.46	1.09	1.68	1.26
0x003C	1.65	0.83	1.77	0.89	2.05	1.56
0x0035	1.46	0.80	1.60	0.87	1.86	1.43
0x003D	1.69	0.91	1.81	0.98	2.17	1.67

Table 7: Average latencies (ms) and standard deviation in Galois/Counter Mode (GCM) cipher mode (without RBAC model).

	Read Coils (0x01)		Read Holding Registers (0x03)		Read-Write Multiple Register (0x17)	
Cipher Suite Code	**Ave.**	**Std.**	**Ave.**	**Std.**	**Ave.**	**Std.**
TCP	0.93	0.56	1.03	0.61	1.13	0.72
0x009C	1.55	0.91	1.65	1.13	2.11	1.75
0xC02B	1.47	0.83	1.54	1.24	2.03	1.53
0x009D	1.81	1.26	2.03	1.35	2.48	2.36
0xC02C	1.56	0.93	1.64	1.31	2.11	1.77

The Figure confirms the analysis performed, once the maximum, minimum and mean values of the latencies associated with each of the three test functions have been recovered. As both the processing of the operations of the encryption suites, as well as the processing of the Modbus function increase, the latency values also increase. If we compare the three functions associated with 0x003C, the greatest increase in latencies is given through the "read-write multiple records" function. Additionally, if we compare 0x003B without

Figure 12: Boxplot (maximum, average, minimum) of latencies (ms) for each of the cipher suites (without applying RBAC model).

symmetric encryption and 0x003C with AES_CBC_128 for the same function, e.g., "read-write multiple records", we observe that latencies increase due to the processing introduced by symmetric encryption. Despite the additional computational complexity, even the slowest cipher suite achieves transaction times below to those shown in Table 4 related to Factory automation (not for motion control), Process automation (remote control and monitoring), Electricity distribution (medium voltage) and intelligent transport systems. Anyhow, encryption algorithms with block cipher operation, such as AES-128-CBC, show the worst performance. Therefore, stream operated ones should be preferred, such as AES-256-GCM, for high parallelizable devices.

7.2. RBAC MODEL RESULTS

Table 8 shows the average and standard deviation of the 1,000 measurements of latencies obtained for each combination of cipher suite and Modbus function, where at each execution of the test the entire process of the RBAC model was carried out as described in the Figure 6.

The increase of the latencies in Table 8 with respect to the latencies analyzed in Table 6 and Table 7 is coherent. On one hand, because more steps are carried out: When the RBAC model is executed, in addition to the TLS handshake and the symmetric encryption, in the server side the role is extracted from the certificate, two queries to Role database are executed and two policies are verified. On the other hand, the client must generate the certificate extension. All these additional steps imply extra processing.

A conclusion from Table 8 is the implication of the execution of Modbus functions on latency. For instance, for all cases it is fulfilled that the difference between the average times

of the reading operations is less than the difference between the average times of either of the two reading operations with respect to the read-write operations.

Table 8: Average latencies (ms) and standard deviation for each of the cipher suites when applying the RBAC model.

	Read Coils (0x01)		Read Holding Registers (0x03)		Read-Write Multiple Register (0x17)	
Cipher Suite Code	**Ave.**	**Std.**	**Ave.**	**Std.**	**Ave.**	**Std.**
0x003B	11.31	3.01	13.34	3.45	17.55	4.54
0x003C	17.17	3.34	20.12	3.65	25.27	4.89
0x009C	19.01	3.29	22.13	3.61	26.11	4.73
0xC02B	18.14	2.79	21.07	3.43	25.24	4.12
0x0035	12.15	3.55	14.41	3.79	19.45	5.01
0x003D	20.31	3.89	24.04	3.99	30.01	5.14
0x009D	23.18	4.03	26.23	4.22	31.43	5.31
0xC02C	20.19	2.91	23.22	3.58	27.33	4.35

With respect to the standard deviation, the conclusion reached in the previous section is verified, where it was determined that the standard deviation increased depending on the complexity of the operations to be performed. This increase occurs both when the complexity of the Modbus function increases, as well as when the complexity of the cipher suite increases. An analysis of the results in Table 8 for the cipher suites compared in the previous section shows that the behavior is the same. These conclusions can be clearly seen in the Figure 13, where for instance, it is evident from 0x009C, the latencies generated by "read-write multiple registers" is higher than "read coils". In addition, if we compare the behaviors of the 0x009D cipher suite with the 0x003D cipher suite, for the same function, e.g., "read-write multiple registers", we can see that the latency generated by 0x009D is higher, which is associated with a higher hash function despite using GCM, while 0x003D uses CBC.

Figure 13: Boxplot (maximum, average, minimum) of latencies (ms) for each of the cipher suites when applying RBAC model.

Although latency increases with respect to the values analyzed in Table 6 and Table 7, it is demonstrated that our proposal is feasible to be used in remote control and monitoring in Process automation, as it fulfills with the latency constraints shown in Table 4. However, at this point it must be reminded that all processes included in Table 4 are deterministic, which is not a requirement for our application context. Finally, it is important to highlight that these results were for the worst-case scenario, where for each execution of the Modbus function a new session was established.

8. CONCLUSIONS

Applications that use SCADA systems rely upon protocols such as Modbus to enable their operations. Many protocols widely deployed lack basic security mechanisms, such as confidentially and the authenticity of transmitted data. When deployed in critical infrastructure assets, these applications enable advanced control possibilities. The exposition of those systems by incorrect deployment or existing vulnerabilities, both in design and implementation, create new attack scenarios. Based on the security recommendations established by the Modbus organization, our manuscript includes a role-based access control model (RBAC) as an access control mechanism, in order to authorize and authenticate systems based on Modbus. This model is divided into an authorization process and an authentication process. The authentication process is provided by TLS, because by default this implements mutual authentication. The authorization process includes both the entity authorization as well as the message authorization. The entity authorization is via roles, which are included as an arbitrary extension in the X.509v3 certificate. The roles are validated with a value stored in a secure database, populated from an out-of-band mechanism. The message authorization process is via a unit id, where it comes from a unique identifier containing the frames, which is validated from a query to the secure database (same database used by the authorization process). In order to evaluate our implementation that we have based on a security analysis, which indicates the attacks against which our implementation is resistant, also justifying the correct implementation. In addition to the security analysis, we perform a performance analysis, since the cipher suites support the mechanism mentioned above different cipher suites were analyzed, establishing a comparison with benchmarks, arriving at the conclusion of the feasibility of the model presented.

Since one of our lines of research is access control, based on attributes (ABAC) in decentralized systems ([40], [41]) and this model of access control based on roles (RBAC) has been applied in a centralized environment (Role Database), the first future line of research is to apply this model in a decentralized environment based on blockchain.

Additionally, we want to provide results to systems based on both Hyperledger Fabric Blockchain and Ethereum blockchain. Thirdly, the creation of a robust mutual authentication RFID protocol that works together with our ABAC blockchain system in order to build a secure supply chain system.

9. REFERENCES

[1] Joelianto, E. Performance of an industrial data communication protocol on ethernet network. In Proceedings of the 2008 5th IFIP International Conference on Wireless and Optical Communications Networks (WOCN '08), Surabaya, Indonesia, 5–7 May 2008; pp. 1–5.

[2] Modbus Organization. Available online: http://www.modbus.org/docs/Modbus_Application_Protocol_V1_1b.pdf (accessed on 21 July 2019).

[3] Bullema, J.E. Available online: https://www.researchgate.net/publication/321770622_2017_-_Bullema_-_Smart_Manufacturing_-_not_only_for_greenfield_high_tech_factories (accessed on 22 July 2019).

[4] Lu, Y.; Morris, K.C.; Frechette, S. Standards landscape, and directions for smart manufacturing systems. In Proceedings of the 2015 IEEE International Conference on Automation Science and Engineering (CASE), Gothenburg, Sweden, 24–28 August 2015; pp. 998–1005.

[5] Whalen, S.; Bishop, M.; Engle, S. Protocol Vulnerability Analysis; Technical Report CSE-2005-04; Department of Computer Science, University of California: Davis, CA, USA, 2005.

[6] Rinaldi, J. OPC UA - Unified Architecture: The Everyman's Guide to the Most Important Information Technology in Industrial Automation, 1st ed.; Independent Publishing Platform: Scotts Valley, CA, USA, 2016; pp. 18–41.

[7] Schneider Electric. Available online: https://www.schneider-electric.com/en/download/document/SEVD-2019-134-05/ (accessed on 21 July 2019).

[8] Alves, T.R.; Buratto, M.; de Souza, F.M. OpenPLC: An open-source alternative to automation. In Proceedings of the IEEE Global Humanitarian Technology Conference (GHTC 2014), San Jose, CA, USA, 10–13 October 2014; pp. 585–589.

[9] Thiago Alves. Available online: https://www.slideshare.net/cisoplatform7/hacking-plcs-and-causing-havoc-on-critical-infrastructures (accessed on 23 July 2019).

[10] Modbus Organization. Modbus TCP Security. Available online: http://modbus.org/docs/MB-TCP-Security-v21_2018-07-24.pdf (accessed on 25 July 2019).

[11] Rescorla, E. The Transport Layer Security (TLS) Protocol Version 1.3. Available online: https://tools.ietf.org/html/rfc8446 (accessed on 27 July 2019).

[12] Rinaldi, J. Available online: https://www.rtautomation.com/rtas-blog/modbus-security-2/ (accessed on 29 July 2019).

[13] Nardone, R.; Rodríguez, R.J.; Marrone, S. Formal security assessment of Modbus protocol. In Proceedings of the 2016 11th International Conference for Internet Technology and Secured Transactions (ICITST), Barcelona, Spain, 5–7 December 2016; pp. 142–147.

[14] Luswata, J.; Zavarsky, P.; Swar, B.; Zvabva, D. Analysis of SCADA Security Using Penetration Testing: A Case Study on Modbus TCP Protocol. In Proceedings of the 2018 29th Biennial Symposium on Communications (BSC), Toronto, ON, Canada, 6–7 June 2018; pp. 1–5.

[15] Al-Dalky, R.; Abduljaleel, O.; Salah, K.; Otrok, H.; Al-Qutayri, M. A Modbus traffic generator for evaluating the security of SCADA systems. In Proceedings of the 2014 9th International Symposium on Communication Systems, Networks & Digital Sign (CSNDSP), Manchester, UK, 23–25 July 2014; pp. 809–814.

[16] Phan, R.C.W. Authenticated Modbus Protocol for Critical Infrastructure Protection. IEEE Trans. Power Delivery 2012, 27, 1687–1689.

[17] Liao, G.Y.; Chen, Y.J.; Lu, W.C.; Cheng, T.C. Toward Authenticating the Master in the Modbus Protocol. IEEE Trans. Power Delivery 2008, 23, 2628–2629.

[18] Huitsing, P.; Chandia, R.; Papa, M.; Shenoi, S. Attack taxonomies for the Modbus protocols. Int. J. Crit. Infrastruct. Prot. 2008, 1, 37–44.

[19] Fovino, I.N.; Carcano, A.; Masera, M.; Trombetta, A. An experimental investigation of malware attacks on SCADA systems. Int. J. Crit. Infrastruct. Prot. 2009, 2, 139–145.

[20] Xiong, Q.; Liu, H.; Xu, Y.; Rao, H.; Yi, S.; Zhang, B.; Jia, W.; Deng, H. A vulnerability detecting method for Modbus-TCP based on smart fuzzing mechanism. In Proceedings of the 2015 IEEE International Conference on Electro/Information Technology (EIT), Dekalb, IL, USA, 21–23 May 2015; pp. 404–409.

[21] Kim, B.K.; Kang, Y. Abnormal Traffic Detection Mechanism for Protecting IIoT Environments. In Proceedings of the 2018 International Conference on Information and Communication Technology Convergence (ICTC), Jeju, South Korea, 17–19 October 2018; pp. 943–945.

[22] Fachkha, C. Cyber Threat Investigation of SCADA Modbus Activities. In Proceedings of the 2019 10th IFIP International Conference on New Technologies, Mobility and Security (NTMS), Canary Islands, Spain, 24–26 June 2019; pp. 1–7.

[23] Hayes, G.; El-Khatib, K. Securing Modbus transactions using hash-based message authentication codes and stream transmission control protocol. In Proceedings of the 2013 Third International Conference on Communications and Information Technology (ICCIT), Beirut, Lebanon, 19–21 June 2013; pp. 179–184.

[24] Ferst, M.K.; de Figueiredo, H.F.; Denardin, G.; Lopes, J. Implementation of Secure Communication with Modbus, and Transport Layer Security protocols. In Proceedings of the 2018 13th IEEE International Conference on Industry Applications (INDUSCON), São Paulo, Brazil, 12–14 November 2018; pp. 155–162.

[25] Graham, J.; Hieb, J.; Naber, J. Improving cybersecurity for Industrial Control Systems. In Proceedings of the 2016 IEEE 25th International Symposium on Industrial Electronics (ISIE), Santa Clara, CA, USA, 8–10 June 2016; pp. 618–623.

[26] X.509: Information technology - Open Systems Interconnection - The Directory: Public-key and attribute certificate frameworks. Available online: http://www.itu.int/rec/T-REC-X.509-201610-I/en (accessed on 1 August 2019).

[27] Figueroa Lorenzo, S.; Añorga, J.; Arrizabalaga, S. A Role-Based Access Control model in Modbus SCADA systems. A Centralized Model Approach. Available online: https://doi.org/10.5281/zenodo.3366479 (accessed on 2 August 2019).

[28] Collins, G. Pymodbus Documentation. Available online: https://buildmedia.readthedocs.org/media/pdf/pymodbus/latest/pymodbus.pdf (accessed on 3 August 2019).

[29] The Python Software Foundation. TLS/SSL Wrapper for Socket Objects. Available online: https://docs.python.org/3/library/ssl.html (accessed on 3 August 2019).

[30] Transport Layer Security (TLS) Parameters, TLS ClientCertificateType Identifiers. Available online: https://www.iana.org/assignments/tls-parameters/tls-parameters.xhtml (accessed on 29 July 2019).

[31] Herzog, J. A computational interpretation of Dolev–Yao adversaries. Theoretical Computer Science Volume 340, Issue 1, 13 June 2005, Pages 57-81.

[32] Stouffer, K.; Falco, J.; Scarfone, K. Guide to Industrial Control Systems (ICS) Security. Gaithersburg, MD National Inst. Stand. Technol. (NIST) 2011, 800, 16. [Google Scholar]

[33] Pascal, A. Industrial Cybersecurity Governance. Efficiently Secure Critical Infrastructure Systems; Packt Publishing Ltd.: Birmingham, UK, 2017; pp. 16–22.

[34] Dachao, H.; Yu'an, H.; Shaokuan, C. Research and Application of Sinec L2 and Modbus Plus Networks on Industrial Automation. In Proceedings of the 2007 International Conference on Mechatronics and Automation, Harbin, China, 5–8 August 2007; pp. 3424–3428.

[35] Groups, R.S. Technical and operational aspects of Internet of Things and Machine-to-Machine applications by systems in the Mobile Service (excluding IMT) Geneva, 2017. Available online: https://www.itu.int/dms_pub/itu-r/md/15/wp5a/c/R15-WP5A-C-0469!N36!MSW-E.docx (accessed on 4 August 2019).

[36] Khuzyatov, S.S.; Valiev, R.A. Organization of data exchange through the Modbus network between the SIMATIC S7 PLC and field devices. In Proceedings of the 2017 International Conference on Industrial Engineering, Applications and Manufacturing (ICIEAM), St. Petersburg, Russia, 16–19 May 2017; pp. 15–17.

[37] Tenkanen, T.; Hamalainen, T. Security Assessment of a Distributed, Modbus-Based Building Automation System. In Proceedings of the 2017 IEEE International Conference on Computer and Information Technology (CIT), Helsinki, Finland, 21–23 August 2017; pp. 332–337.

[38] Triangle Microworks. Available online: http://www.trianglemicroworks.com/products/SCADA-data-gateway/iccp-tase-2 (accessed on 5 August 2019).

[39] Bonganay, A.C.D.; Magno, J.C.; Marcellana, A.G.; Morante, J.M.E.; Perez, N.G. Automated electric meter reading and monitoring system using ZigBee-integrated raspberry Pi single board computer via Modbus. In Proceedings of the 2014 IEEE Students' Conference on Electrical, Electronics and Computer Science, Bhopal, India, 1–2 March 2014; pp. 1–6.

[40] Figueroa, S.; Añorga, J.; Arrizabalaga, S.; Irigoyen, I.; Monterde, M. An Attribute-Based Access Control using Chaincode in RFID Systems. In Proceedings of the 2019 10th IFIP International Conference on New Technologies, Mobility and Security (NTMS), Canary Islands, Spain, 24–26 June 2019; pp. 1–5.

[41] Figueroa, S.; Añorga, J.; Arrizabalaga, S. An Attribute-Based Access Control Model in RFID Systems Based on Blockchain Decentralized Applications for Healthcare Environments. Computers 2019, 8, 57.

CHAPTER IV

AN ATTRIBUTE-BASED ACCESS CONTROL MODEL IN RFID SYSTEMS BASED ON BLOCKCHAIN DECENTRALIZED APPLICATIONS FOR HEALTHCARE ENVIRONMENTS

This chapter was published in S. Figueroa-Lorenzo, J. Añorga, and S. Arrizabalaga, "An Attribute-Based Access Control Model in RFID Systems Based on Blockchain Decentralized Applications for Healthcare Environments," Computers, MDPI, vol. 8, no. 3, p. 19, 2019, doi: doi.org/10.3390/computers8030057. [SJR. 3.3, Q2].

This chapter focuses on the development of an ABAC access control system over an IIoT-RFID environment, which is more flexible and scalable than RBAC. The ABAC policy, i.e., authorization, is executed on a decentralized application, i.e., off-chain, where each of the endpoints running the DApp (RFID Reader) have control of the keys according to the Ethereum (ETH) public network rules, which decentralizes the identity significantly. Proof-of-concepts conducted over the ETH Ropsten test network demonstrated the technical feasibility of the proposal with acceptable latency levels over the test network compared to the latency levels achieved when a smart contract is deployed on local ETH nodes. Despite the successful technical evaluation of the proposal, the high transaction cost implies the need for further research to overcome this shortcoming.

1. ABSTRACT

The growing adoption of Radio-frequency Identification (RFID) systems, particularly in the healthcare field, demonstrates that RFID is a positive asset for healthcare institutions. RFID offers the ability to save organizations time and costs by enabling data of traceability, identification, communication, temperature, and location in real time for both people and resources. However, the RFID systems challenges are financial, technical, organizational and above all privacy and security. For this reason, recent works focus on attribute-based access control (ABAC) schemes. Currently, ABAC are based on mostly centralized models, which in environments such as the supply chain can present problems of scalability, synchronization, and trust between the parties. In this manuscript, we implement an ABAC model in RFID systems based on a decentralized model such as blockchain. Common criteria for the selection of the appropriate blockchain are detailed. Our access control policies are executed through the decentralized application (DApp), which interfaces with the blockchain through the smart contract. Smart contracts and blockchain technology, on the one hand, solve current centralized systems issues as well as being flexible infrastructures that represent the relationship of trust and support essential in the ABAC model in order to provide the security of RFID systems. Our system has been designed for a supply chain environment with a use case suitable for healthcare systems, so that assets such as surgical instruments containing an associated RFID tag can only access to specific areas. Our system is deployed in both a local and Testnet environment in order to stablish a deep comparison and determining the technical feasibility.

2. INTRODUCTION

The healthcare field is aware of the essential need to adopt and use healthcare information technology (IT) successfully. Radio-frequency Identification (RFID) provides several opportunities for healthcare transformation [1]. The same reference before argues that RFID provides an enhanced method to decrease errors in patientcare, to improve tracking and tracing for both patients and equipment, as well as to enable better management of health assets and improving the audit process and predictability.

In general terms, four sub-systems describe an RFID system's architecture (Figure 1): (1) a transponder or tag, which contains the identification data, (2) a reader to interact directly with the tag exchanging information with it, (3) a RFID middleware and (4) a business and/or information management layer. RFID middleware supports RFID tag data management by

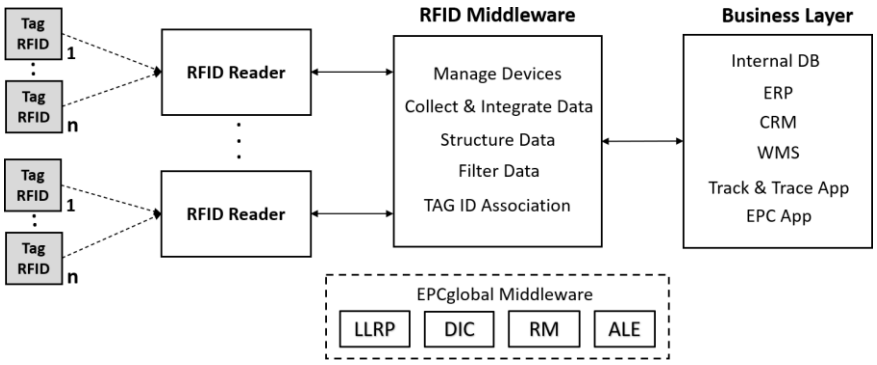

Figure 1: General architecture of Radio-frequency Identification (RFID) systems.

handling devices, filtering, collecting, integrating, and constructing data. The business layer also includes applications such as back-end databases (DBs), enterprise resource planning (ERP), customer relationship management (CRM), warehouse management solutions (WMS), tracking and tracing and electronic product code (EPC) applications.

The GS1 (Whilst "GS1" is not an acronym it refers to the organization offering one global system of standards) standards ([2]) address three wide categories: identify, capture and share [3][4]. The capture process could be performed through sub-systems (1) and (2), using EPC-enabled for RFID tags (i.e., through EPC, GS1 also provides a construction to write and read unique identifiers on RFID tags). The identification process would be covered by sub-system (3) and an identification number is performed, for instance, when it is encoded (e.g., to GTIN (Global Trade Item Number)) or it is decoded (e.g., from RFID Tag EPC). Finally, the sub-system 4) carried out the sharing category. In particular, GTIN describes a data structure that uses 14 digits with the option to encode in some combinations. GTIN is currently used in both barcodes and RFID [2]. The structure of the GTIN number is shown below:

urn:epc:id:sgtin:CompanyPrefix.ItemReference.SerialNumber, **(1)**

These fundamental principles are used to explain how the GS1 standards system can be used to enable traceability solutions, where RFID systems are involved in both data capture and data sharing. In addition, RFID systems are able to achieve traceability in a variety of supply chains such as fresh food, health, technical industries, transportation, and logistics. The supply chain in the healthcare sector will be taken as use case. In that sense, RFID is the industry-leading technology used by medical device manufacturers to enable smart devices to provide higher-quality patient care, the most common RFID applications include [5]:

1. Tracking and tracing of trust device to individual patients.
2. Ensuring appropriate sterilization.
3. Control of servicing and calibration of medical equipment.
4. Invoicing procedures, to associate patients with medical device and prescription use.
5. Stock management.
6. Decreased time spent by staff tracking articles and devices.

In this way, a model based on control and traceability of assets is a determining factor in safety. Based on the analysis of the six points mentioned above, interviews with specialists were carried out in order to determine the needs existing in institutions, where a specific use case related to point 1) has been identified. Hospitals employ large numbers of assets (e.g., surgery medical instruments (SMI)), which can flow through constant cycles such as sterilization department, surgery room, laboratories, etc. A location mistake could risk the patients' lives. In addition, the lack of detailed asset records causes asset losses.

However, given that RFID is one of the most well positioned technologies to perform the data capture and sharing process, the biggest challenge for any RFID systems is its security. The security threats encountered in RFID systems are distinct from traditional wireless security threats, which can be grouped into: (1) physical components of RFID (e.g., cloning tags, reverse engineering, tag modification), (2) the communication channel (e.g., eavesdropping, skimming, replay attack), and 3) global system threats (e.g., spoofing, Denegation of Service (DoS) and "tracing and tracking"). Other examples and details can be obtained from reference [6]. Therefore, our proposal must focus on both safety risks and security risks.

Access control (AC) is a core piece of any organization's security infrastructure. In particular, AC has popularized as a solution for some of the threats mentioned [7]. Below is an overview of our proposal. Based on GS1, SMI are tagged (passive RFID tag) with GTIN. The coding scheme (see **expression (2)**) contains a company prefix (e.g., Hospital A: 000389), an article reference (product type) to categorize the asset (e.g., scissors: 000162) and, finally, a serial number to identify a specific asset (e.g., serial number: 000169740). **Figure 2** helps to detail how our healthcare system works. The source room (e.g., sterilization department) sends some assets (e.g., SMI) to the destinations rooms (e.g., surgery $room_0$, surgery $room_1$). Since $asset_1$ has been assigned to destination $room_1$ (e.g., surgery $room_1$) and due to human mistake (e.g., in transportation) attempts to access to destination $room_0$ (e.g., surgery $room_0$),

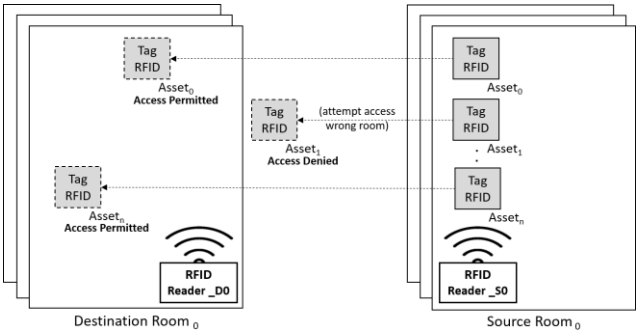

Figure 2: Healthcare system.

our system establishes access denied status to $asset_1$. In short, our proposal is an access control system of a healthcare asset (e.g., SMI) in order to prevent unwanted assets from entering the wrong area (e.g., room) because of human error or an external security threat. Therefore, our proposal is a prevention system that provides a security-safety solution.

By considering the general model presented in Figure 1, RFID middleware commonly deploys the access control mechanism (ACM) in an RFID system [8]. For instance, the EPC global community standardizes four main layers within the middleware (Figure 1): low level read protocol (LLRP), discovery initialization and configuration (DCI), read management (RM) and application-level event standard (ALE) [9]. ALE is the sub-system that applies AC policies.

The Table 1 contains different implementations of the ALE sub-system, included on EPC global middleware, as it is established by the following specifications [10][11]. These traditional AC systems present two major challenges for supply chain application environments: (1) almost all include role-based access control (RBAC) as AC model and (2) the implementations are based on centralized architectures. Therefore, from the technical point of view, our proposal consists of an ABAC system for RFID systems that executes access control (AC) policies from a decentralized application (DApp) based on a blockchain architecture. Our proposal integrates several technologies, which allow in the first place the tracking of assets, i.e., an asset (e.g., SMI) is associated to a GTIN code. The system allows us to verify the existence of a certain asset based on the coding scheme presented. Finally, the proposal makes it possible to permit or deny access of an asset to a certain area (e.g., surgery room). For this, smart contracts are used as an interface between the DApp and the blockchain, i.e., all these functionalities, including the AC policy, are executed from the DApp, which interacts with the smart contract, which, in turn, interacts directly with the blockchain, e.g., by a method to insert assets or a method to query certain attributes. The remainder of the article includes the related work section, which is the keystone for the design of our system based on the reviewed literature, allowing us to arrive at conclusions. From this point

onwards, we present the technical proposal, followed by the evaluation methodology, the results obtained and finally the conclusions and future research lines.

Table 1: Electronic product code (EPC) global standard: application-level event standard (ALE) version and access control model [12].

Middleware	ALE Version	Access Control Model
IBM WebSphere RFID	1.1	RBAC
Oracle sensor edge server	1.0	RBAC
Rifidi Edge	1.1	RBAC
Fosstrak	1.1	-
Chuck	1.1	RBAC
Aspire	1.1	Access Control API[1]
WinRFID	1.1	-

[1] API: Application Programming Interface.

3. RELATED WORK

In this section, we present the literature justification that allows us to establish the use of an access control model based on attributes (ABAC) over an access control model based on roles (RBAC) for our use case. In addition, we indicate the preference of decentralized architectures over centralized architectures for supply chain environments and our use case. Additionally, we justify the blockchain selection within the set of distributed ledger technologies (DLT). Next, the type of blockchain that best fits our proposal is analyzed. Once the blockchain has been selected, if this is a public blockchain, a business model must be associated with it for implementation to be feasible. For this reason, we present a proposal based on an asset tokenization model for healthcare environments. However, the implementation of the tokenization model is established for future research lines. The related work section concludes with a discussion sub-section.

3.1. Attribute-Based Access Control (ABAC) vs. Role-Based Access Control (RBAC)

Although the RBAC model is well established, Gartner predicts that by 2020, 70% of companies will use ABAC to protect critical assets [13]. In addition, the following references [14][15][16], provide some clear limitations for the RBAC model such as:

1. It is not possible to configure rules using parameters, which are unknown to the system.
2. Permissions can only be assigned to user roles, not to objects and operations.
3. Since the RBAC model is predominantly based on static organizational positions, there are problems in particular RBAC architectures where dynamic AC decisions are needed.

4. It is possible to restrict access to specific system actions but not to data model.
5. RBAC does not support multi-factor decisions (e.g., decisions that depend on location and timestamp).

On the other hand, the ABAC model presents important benefits that are adapted to our use case (Section 3) such as:

1. ABAC provides access based on the attributes of each system component and not based on the user function [14].
2. ABAC supports AC decisions without previous understanding of the object by the subject or understanding of the subject by the object owner [15].

A comparison that includes other features can be analyzed in Table 2. As a conclusion, we can establish that the ABAC model is suitable for our case of use based on the supply chain, i.e., applications that require flexibility and scalability.

Table 2: Comparison between RBAC and ABAC [14].

Characteristic	RBAC	ABAC
Flexibility	Yes (For small and medium-sized organizations)	Yes
Scalability	No	Yes
Simplicity	Easy to establish roles and permissions, hard to maintain the system for a big company	Hard to establish all the policies at the start, easy to maintain and support
Support simple rules	Yes	Yes
Support complex rules	Yes	Yes
Support rules with dynamic parameters	No	Yes
Customizing user permissions	No (Every customization requires creating a new role)	Yes
Granularity	Low	High

3.2. DECENTRALIZED MODEL VS CENTRALIZED MODEL

As it can be analyzed in reference [9], most middleware implementations are based on centralized architectures. In order to examine the disadvantages of this model we use a common application environment for RFID systems such as the supply chain. Although the centralized approach is well adopted, it is not scalable, introduces bottlenecks and makes difficult to synchronize information, e.g., product status among different parts with their centralized DBs or to add new elements [17]. In addition, this model does not provide the

degree of trust that must exist between the parties and therefore someone who is accountable for the data shared [18].

Different works focus on attribute-based access control models on centralized architectures. For instance, the reference [19] presents an AC model for IoT, in which it is established a coupling between ABAC and trust concepts. In addition, the reference [20] promotes an ABAC mechanism, which is applied to give the system the ability to implement policies to detect any unauthorized entry.

On the other hand, a decentralized model provides a solution to the aforementioned problems: firstly, the supply chain adopts a method in products can be tracked through every step of the chain, from suppliers, through manufacturers, to end users and secondly, with a certain degree of trust between the parties. Although a model based on decentralized architecture is the solution, the type of architecture to be used must be studied in depth, above all, to establish selection criteria.

3.2.1. Blockchain over Other Distributed Ledger Technologies

Blockchain technology is now entering a maturity stage that determines the use cases where the technology is applicable, which determines even the type of blockchain to be used. However, blockchain is not the only type of DLT, e.g., directed acyclic graphs (DAG) is considered another way to represent the data structure with advantages over the blockchain approach [21]. Therefore, we want to emphasize below the reason why the blockchain is suitable as a decentralized solution. First, it is clear that all data are not located on a central server but are decentralized. These are distributed across all devices connected to the blockchain, so the blockchain can be thought of as a network of nodes from peer-to-peer where a device (e.g., miner device) connected to the blockchain as the node, which talks to all the other nodes. In addition, this device will share the same responsibilities as the other peers, and it will get a copy of all the data that is shared across the blockchain. All of this data is contained in packets of records called blocks that are chained together to create a "public" ledger and all of the network nodes work together to ensure that all of the “public” ledger data remains secure and unchanged and this is important for an AC application. The blockchain is fundamentally a DB and because all nodes communicate with each other in the blockchain, it is a network, so instead of the traditional centralized model, it is possible to think on a blockchain as a network and a DB all in one [22]. Once it is determined that blockchain is the type of decentralized architecture to implement, we focus on defining the type of blockchain suitable for our use case.

3.2.2. Selecting the blockchain type

Although the technical criteria of selection is fundamental, we first review the existing proposals in the literature and then the technical selection criteria. In that sense, there are some proposals that use AC based on blockchain, including RBAC. For example, the reference [23], proposes an approach based on blockchain to publish policies expressing the right to access a resource and to allow the distributed transfer of that right between users. In addition, the reference [24] includes a dynamic access control scheme for direct data communication between Internet of Things (IoT) devices. The reference [25] presents a RBAC using Smart Contract to realize trans organizational utilization of roles. Finally, transaction-based access control (TBAC) is a platform that integrates the ABAC model and the blockchain, combining four types of transactions and Bitcoin-type cryptographic scripts to describe the TBAC access control procedure corresponding to subject registration, object holding and publication, access request and grant [26]. By analyzing existing proposals that combine access control models in decentralized architectures, we conclude that these are mostly based on RBAC models. In addition, the proposal found based on ABAC uses Bitcoin as a blockchain. The technical criteria are detailed below.

Table 3: Comparison between popular blockchain types and a centralized database [17].

Description	Public	Permissioned	Private	Centralized DB
Participation	Anyone	Members of organizations	Members of organizations	Limited
Write permissions	Granted	Restricted	Restricted	Restricted
Read permissions	Granted	Granted	Restricted	Granted
Speed	Slow	Fast	Fast	Slow
Identity	Anonymous	Anonymous	Known	Known
Security	Impervious to security attacks	Impervious to security attacks	Impervious to security attacks	Vulnerable to security attacks
Transparency	Visible across all supply chain nodes	Visible across all supply chain nodes	Restricted to specific supply chain nodes	Restricted to specific supply chain nodes
Traceability	Yes	Yes	Restricted	No

The selection of the type of blockchain depends on factors such as the use case, the technical requirements and even the business model. For this reason, firstly it is considered the most recent Gartner recommendations, which indicate that to ensure a successful blockchain project, it is necessary to focus on the business problem, not on the technology solution [27]. According to the use case and the characteristics of the technological project, it is necessary to select between: a model based on governance with some trust between the

parties and a certain level of centralization, represented by the Hyperledger Fabric Blockchain (HFB) or a model where there is no trust between the parties and fully decentralized, represented by the Ethereum (ETH) blockchain. The Table 3 presents common criteria to establish a comparison among different blockchain types. From our use case of a supply chain based on healthcare environments and considering, the project scalability, community support, skill availability, multi-functionality, and adaptability, we consider deploy our DApp on the ETH blockchain.

From the decision (type of blockchain) based on technical criteria and considering the literature review carried out, we can affirm that our proposal contains high novelty value. The following sub-section summarizes all the points analyzed throughout the related work section and based on our selection criteria and the revised bibliography; we arrive at conclusions and introduce a model.

3.3. DISCUSSION

From the analysis performed throughout this section, we determined that ABAC is a suitable access control model for applications requiring flexibility and scalability. In addition, we analyzed that our use case is not optimal to build a centralized application for two reasons. Since our system is based on the ABAC model, asset attributes can change at any time, so that decision support is highly scalable. In addition, all the code in the application could change at any time and this means that the rules in AC policy also could change. Additionally, blockchain has been selected as DTL, thanks to features such as immutability, necessary to ensure both AC and a reliable history of asset attributes behavior. Based on common criteria of selection, we determined the type of blockchain suitable to our proposal: the ETH blockchain.

However, if we take up Gartner's recommendation [27] to be able to deploy our proposal on the main ETH network, even if the project is technically feasible, it needs to be endowed with a business model that makes it viable. For that reason, we present below one based on asset tokenization.

The tokenization through a blockchain platform (the most used is ETH) enables us to leave not only the use of expensive and complex transactions, but also the exchange itself. Any person enrolled in the blockchain could potentially act as an issuer of a legitimate asset that he or she would like to tokenize [28]. Applied to healthcare, tokenization can contribute to reducing the cost of private medical treatment by transferring the ability to maintain and hold data from intermediaries, like insurance companies, hospitals, and pharmacies to

patients. In the existing scheme, neither of these subjects share information with patients, and patients are unable to verify the data's correctness. Through tokens, both patients and the general public can keep their data and share it with anyone they want [29]. Tokenization can also automate the payment process. In addition, since tokens are a secure and protected way to make transactions, the payment system is simplified. However, the main challenge is that, so far, no nation has a strong regulation for cryptocurrency. As a result, tokens do not have legal rights to property and are not protected by law. Therefore, legislative changes are required to adapt these new business models [30].

Therefore, a business model based on tokenization is applicable to a public blockchain such as ETH, which it contextualized to healthcare through the supply chain and, therefore, to our model based on control and traceability. A tokenization model is applicable to both assets (e.g., SMI) and for patient information (e.g., blood pressure sensors). Thus, a security model based on AC is highly suitable and applicable in these environments.

Since our article is a proposal, the technical behavior of our implementation evaluates firstly in a local environment (i.e., our own ETH node, without joining the main ETH network) and secondly scales to an ETH Testnet. Ropsten Ethereum, also known as "Ethereum Testnet", is a testing network that runs the same protocol as Ethereum and is used for testing purposes before deploying on the main network (Mainnet). In order to scale our proposal to the Mainnet, we will propose a tokenization-based model, which we introduced above and is part of our future research lines.

4. PROPOSAL

As we mentioned in the introduction, our proposal consists of a complete system in which several technologies converge. However, we want to start this section with a basic architecture that enables a general understanding of how our system performs ABAC.

4.1. DECENTRALIZED SYSTEM ARCHITECTURE

Figure 3 represents the general architecture of the proposed ABAC model based on ETH blockchain. The physical node is composed of the RFID Reader Control (RFID-RC), the DApp and the smart contract. When a medical instrument (previously tagged with an RFID tag) attempts to gain access to a room, the RFID-RC sends a request for access to the DApp. The DApp sends a query via smart contract to the blockchain network, which returns some attributes related with the asset (e.g., company prefix, product type, serial number). In addition, the DApp receives other attributes (e.g., timestamp) from the RFID-RC. Then, the

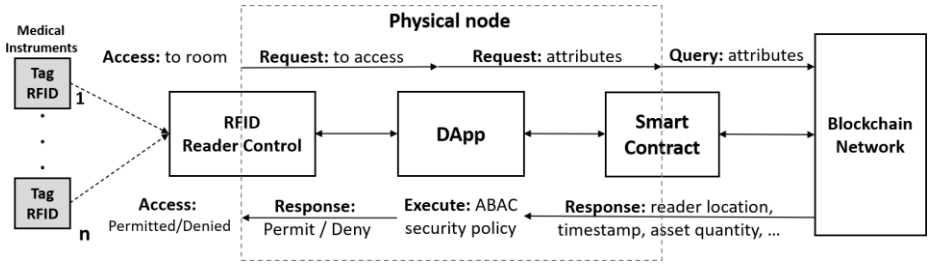

Figure 3: Decentralized system architecture based on Ethereum (ETH) blockchain.

DApp uses the attributes to execute the ABAC security policy, which determines whether tag access is permitted or denied. The next section details the implementation framework used. Additionally, we need to confirm that one of the main advantages of a decentralized system is scalability, so this physical node (smart oracle) can be replicated in a way that establishes a new connection with the blockchain (via smart contract), without affecting any of the existing nodes.

4.2. ACCESS CONTROL MECHANISM (ACM)

For a subject to be able to execute a policy on an object (e.g., permit or deny access), ABAC access control mechanism (ACM) must be enabled. ACM includes the next steps: (1) check the subject's attributes, (2) check the AC policies (rules), (3) check the object's attributes, and 4) check the environmental conditions. Although it is normal to expect that the subject is a human, a non-person entity (NPE), such as an autonomous service or an application, could also occupy the subject's role, as the reference [15] indicates. In our case, the reader requests the DApp for the tag RFID (associated with a SMI) access.

Before analyzing the AC policy, some boundary conditions are established for the transfer of an asset from the source room (e.g., sterilization area) to the destination room (e.g., surgery room) and vice versa should be mentioned:

1. The transaction that authorizes the transfer of an asset is invoked by an authorized employee through a trusted application connected to DApp (Figure 4).
2. The tag uses an EPC code with a pattern similar to the expression (1) and illustrated in expression (2):

The process that is performed by DApp when it receives an access request is described next. The variables' names used to define the AC policy are included. (1) The subject (reader) is verified based on two attributes: reader name (variable 01: "rdr_nm", e.g., rdr_nm: "roomA") and location (variable 02: "loc", e.g., loc: "41.40338, 2.17403")). (2) The company

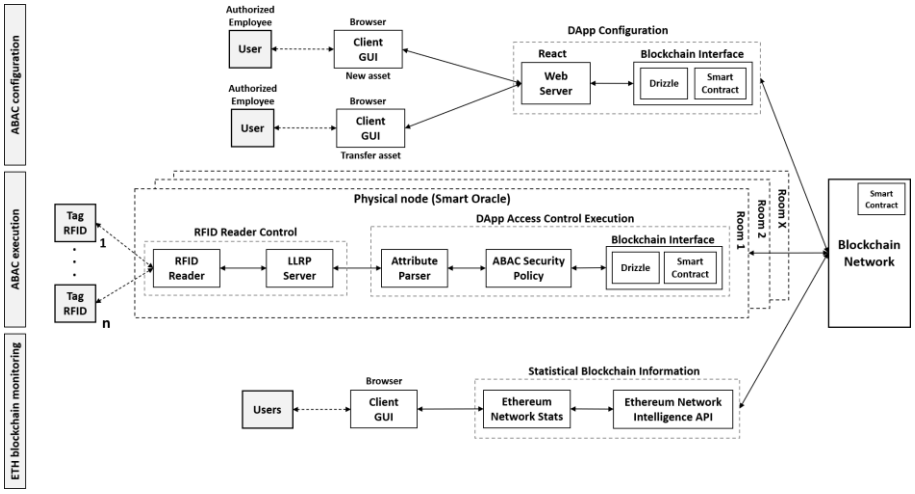

Figure 4: Details of the system architecture.

$$\underset{\text{Header}}{01}.\underset{\text{CompPrefix}}{0000389}.\underset{\text{Product type}}{000162}.\underset{\text{Serial Number}}{000169740}, \quad \textbf{(2)}$$

prefix (variable 03: "cmp_prf", e.g., cmp_prf: 000389), the product type (variable 04: "item_ref", e.g., item_ref: 000162), the serial number of a specific asset (variable 05: "ser_nmb", e.g., ser_nmb: 000169740) and the asset status (variable 06: "st", e.g., st: "STERILIZED") are verified. (3) The environmental condition is verified based on the time elapsed since an asset is sent to an existing reader in a medical room (through a transaction) and that reader receives the request for access to that asset (tag). The environmental condition is approved if the interval is less than 10 min (600 s). This time is set for moving assets between locations once the transaction has been invoked. In that sense, variable 07: "time_in" (e.g., time_in: 1,560,209,335) is the time record once the transaction is completed and variable 08: "time_out" (e.g., time_out: 1560209455) is the time given when the reader requests access to this RFID tag.

Based on the AC policy notation established in the reference [31], our AC policy C is defined in the expression (**3**).

We decided to implement the AC policy on the DApp and not as part of the smart contract for two reasons. Firstly, as we indicated in **Table 3**, one of the constraints of a public blockchain is the speed, so if the AC policy is executed as part of the smart contract, it would lead to a delay. Secondly, since smart contracts are public the AC policy would be exposed. In this way, one of the future research lines is the implementation of this model in a private

$$C = \begin{cases} True, \quad if: (\text{rdr\_nm} = \text{“roomA”} \cap \text{loc} = \text{“41.40338, 2.17403.”} \cap \\ \text{cmp\_prf} = 000389 \cap \text{item\_ref} = 000162 \cap \text{ser\_nmb} = 000169740 \cap \\ \text{st} = \text{“STERILIZED”} \cap \text{time\_out} - \text{time\_in} <= 600) \\ False, \quad otherwise: \end{cases} \tag{3}$$

blockchain (e.g., HFB); so that the AC policy can be located within the smart contract (chaincode) in order to analyze these results.

Table 4: Technologies used.

Sub-System	Block	Implementation	Technology/Library
ABAC Configuration	New asset	X	ReactJS
	Verify ID	X	ReactJS
	Transfer asset	X	ReactJS
	Blockchain Interface	X	Truffle-Drizzle
ABAC Execution	Rifidi Virtual Reader	[33]	Java
	LLRP Server	X	llrp-nodejs [34]
	Attribute Parser	X	node-epc [33]
	ABAC Security Policy	X	Java Script
	Blockchain Interface	X	Truffle-Drizzle
Blockchain monitoring	ETH Network Stats	[35]	AngularJS
	ETH Network Intelligence API	[36]	Java Script
Blockchain local node	Smart Contract	X	Solidity
	ETH Network	X	Geth
Legend	*“X” implies that we implements the block based on the technology or library*		
	“[ ]” implies that we use the project.		

4.3. TECHNICAL IMPLEMENTATION DETAILS

The two previous sub-sections allowed respectively to define the general functioning of our ABAC model and to detail how the AC policy is executed in the DApp; therefore, it is time to present our system in detail. In order to better understand it, it begins with a summary of the main technologies used. Table 4 summarizes the technologies used in each sub-system and their associated blocks. Table 4 follows the Figure 4 design principles. For a better understanding of our work, the reference [32] is a film reference included as external document. In order to analyze the technical implementation details, Figure 4 shows the specific architecture of our system, which consists of three sub-systems: ABAC configuration, ABAC execution and ETH blockchain monitoring.

4.3.1. ABAC configuration

The ABAC configuration sub-system includes a graphical interface (GUI), based on ReactJS web technology that is launched from a browser (Figure 5). This GUI includes two views (Figure 5). Since a demonstration environment is presented, the two views are included within the same browser window, however as it is detailed, each view has its own functionality. The first view allows an authorized employee to add new assets to the system. This employee introduces the code of the company prefix, the code of the product type, the asset ID (e.g., serial number) and so on (Figure 5). Each time a new asset is stored in the ETH blockchain, a new transaction is generated. In order to transfer an asset between rooms the authorized employee first needs to verify the ID (e.g., serial number) of the asset, through a simple query to the blockchain via smart contract. To do this, the authorized employee uses the button ("VERIFY ID") of the second view. This blockchain query does not generates transactions. Next, the same second view enables the transfer assets from the source room (e.g., sterilization area) to the destination room (e.g., surgery room), before attributes values, such as asset status (e.g., "STERILIZED") and timestamp are updated (Figure 5). This action is carried out from "TRANSFER ASSET" button. Since asset transfer involves changes (e.g., room, status, timestamp) new transactions are generated via smart contract. Details of blockchain interface operation are analyzed in the following sub-section.

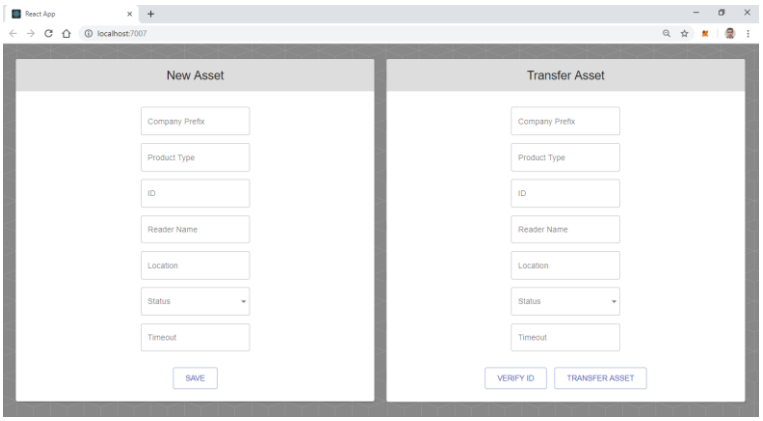

Figure 5: Graphical interface (GUI) of ABAC configuration sub-system.

4.3.2. ABAC execution

The ABAC execution sub-system contains a smart oracle to permit or deny asset access and it is located in each of the medical rooms. Our smart oracle includes the RFID reader, the LLRP server, the attribute parser (AP), the ABAC security policy (ABAC-SP) and the blockchain interface (BI). The AP, the ABAC-SP and the BI comprise the DApp access

control execution (**Figure 4**, ABAC execution sub-system) and RFID-RC includes the RFID reader and the LLRP server (**Figure 4**, ABAC execution sub-system).

The RFID reader interacts directly with the tagged assets and the LLRP server. LLRP is a protocol that EPC global ratified as a standard that constitutes an interface between the reader and its software or control hardware [37]. The protocol sends XML (eXtensible Markup Language) messages between the client (e.g., RFID reader) and the server (e.g., LLRP server). To develop our proof of concept (PoC) we use an open-source tool, as known as Rifidi ([38]), that create a virtual reader and RFID tags based on SGTIN96 standard. Details of the project that supports it, as well as the getting started guide are located in [33]. In addition, since our LLRP server is based on the standard LLRP, it is agnostic to any RFID reader that supports the LLRP protocol such as Motorola FX7400, Intermec IF61 and Impinj Speedway.

The AP receives the RFID Tag EPC from the LLRP server and uses a GTIN conversion system, based on a NodeJS library [34], which allows transforming the RFID TAG EPC code to the EPC Tag URI (Uniform Resource Identifier) (e.g., **expression (4)**). AP filters the attributes: Company Prefix (variable "cmp_prf" in AC policy), Product Type (variable "item_ref" in AC policy) and Serial Number (variable "ser_nmb" in AC policy). In addition, AP controls other attributes such as timestamp (variable "time_out" in AC policy), reader name (variable "rdr_nm" in AC policy) and location (variable "loc" in AC policy).

RFID Tag EPC: 3074257bf7194e4000001a85
EPC Tag URI: urn:epc:tag:sgtin-96:3.0614141.812345.6789, **(4)**

The BI is built based on the truffle framework, using the drizzle library to interact with the web3.js server. Drizzle is a collection of front-end libraries that enable writing DApp front-end in an easier way [39]. The communication is performed between the parties via GET and POST methods. For instance, ABAC-SP determines whether asset access is permitted or denied, it sets a variable, which is sent via POST method to the LLRP server. Therefore, the LLRP server sends an XML "keepAlive" message (**Figure 6**) to maintain the interaction with the RFID tag or simply disconnects it.

```
<?xml version="1.0" encoding="UTF-8"?>
<llrp:SET_READER_CONFIG_RESPONSE xmlns:llrp="http://www.llrp.org/ltk/
  <llrp:LLRPStatus>
    <llrp:StatusCode>M_Success</llrp:StatusCode>
    <llrp:ErrorDescription />
  </llrp:LLRPStatus>
</llrp:SET_READER_CONFIG_RESPONSE>
```

Figure 6: (eXtensible Markup Language) XML keepAlive messages to permit the access.

To execute the AC policy established by the expression (3), ABAC-SP matches the attributes from the AP with the attributes queried from the blockchain.

4.3.3. Ethereum (ETH) blockchain monitoring

Although this sub-system is an integral part of our implementation ([32]), how the following section is dedicated i.e., it enumerates and describes the monitoring tools of our system in order to verify its feasibility, we have preferred to set the analysis of this sub-system as part of the successive sections. In that sense, Figure 7 represents the monitoring tool ETH Network Stats, which as part of this sub-system.

Figure 7: ETH Developer tools List.

5. EVALUATION METHODOLOGY

In order to evaluate the feasibility of our proposal, it is necessary to indicate first that our model has been deployed in two environments, one based on local blockchain and the other based on a Testnet blockchain.

In the first case, an ETH node was deployed, although it included the property of no discover, making it impossible to connect to the Mainnet. The expression (5) is a sample of the command deployed based on geth client (main ETH client).

```
geth --datadir data --unlock 0x8a6d63ea98e05a550b01f8aa4a19021e43bd43f0 --networkid 123456 ----ws –wsaddr 192.168.127.95 --wsport 8546 --wsorigins "*" –rpc –rpcaddr 192.168.127.95 --rpcport 8545 --rpccorsdomain "*" --nodiscover console 2>> ETH.log,          (5)
```

In the second case, to scale our system, as mentioned above, we use Ropsten as Testnet. Some of the advantages that have allowed us to select this network over others like Kovan, Rinkeby and Sokol are:

1. It better reproduces the current production environment, i.e., the system and network conditions in the Mainnet, since it employs the proof of work (PoW) as the consensus algorithm between the nodes.

2. It can be used with both geth client and parity client.
3. It enables to join its own node to the network, i.e., to participate in the PoW or simply request the ether from a faucet.

ropsten.infura.io/v3/fa42299dbea54014801bc4145d7a1a1e, **(6)**

To access the network, it is necessary to create an Infura project, which generates the endpoint URL (for example, the expression (**6**)) used in the configuration files of our system (truffle-config.js). Next is a detail of the tools used and then the features that are measured:

5.1. EVALUATION TOOLS

First, we present the tools used to evaluate our system: ETH Network Stats, Etherscan, Truffle Test, and Infura Dashboard. These tools have been deployed both for the local environment and for the Testnet. The exception is Infura which is a tool that is only associated with the Testnet. Below is a brief description of the tools.

ETH Network Stats is a tool composed by a front-end Ethereum Network Stats [35] and a back end Ethereum Network Intelligence API [36]. This is a visual interface for tracking Ethereum network status. It uses WebSockets to receive stats from running nodes and output them through an angular interface. Both servers are installed locally. This tool was presented as part of the sub-system ETH blockchain monitoring. ETH Network Status ([40]) is the equivalent to ETH Network Stats but used to the Ropsten Testnet.

Etherscan Ropsten Testnet Network ([41]) is a tool that we will use to monitor the state of the blockchain and the transactions that are stored in it. This tool presents an equivalent for the local environment, which it is installed as a server.

Infura Dashboard is a response to developer demand for a better understanding of how to improve DApps. The following reference [42] mentions that it has been recently updated, enabling us to obtain relevant information about calls to web3.js methods, which allow for some type of interaction (e.g., generating a transaction) with Ropsten Testnet.

Truffle test is combined with the data obtained from contract migration process in order to improve the data analysis. Truffle comes standard with an automated testing framework to make testing the smart contracts easy. This framework lets it write simple and manageable tests in JavaScript or Solidity. The JavaScript way is used from the outside world, just like an application. The Solidity way is used in bare-to-the-metal scenarios. Truffle test has been deployed for both the local environment and the Ropsten Testnet.

The Table 5 summarizes these tools with application environment and the main features measured.

Table 5: Tools used to test the proposal.

Tool	ETH Network Stats/ETH Network Status	Etherscan	Truffle Test	Infura Dashboard
Features	Network monitoring	Blockchain monitoring	Smart Contract monitoring	Bandwidth monitoring
Local Environ-ment	Network monitoring of our local node (Figure 7)	Local ETH Blockchain monitoring (e.g., contract addresses, transactions, blocks)	Testing the smart contract interaction with local blockchain	-
Ropsten Testnet	Network monitoring of Ropsten	Testnet blockchain monitoring (e.g., contract	Testing the smart contract interaction	It allows seeing the bandwidth behavior for each web3.js method used

5.2. MEASUREMENTS

We consider the analysis as integral because can test each part of the implementation, i.e., from network monitoring, with features like the number of nodes and the network hashrate to the delay of the smart contract application and the bandwidth consumption for each

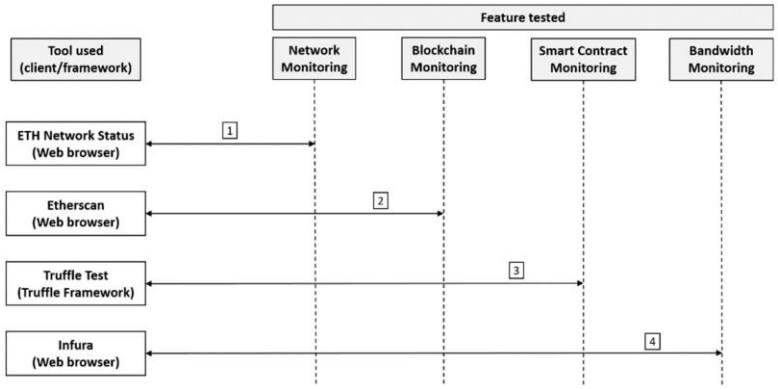

Figure 8: Sequence diagram of tested features and used tools.

web3.js methods. Below we present the main parameters that can be monitored through the tools listed in the previous sub-section. Considering the main characteristics associated with each tool, Figure 8 establishes the logical order of use of these tools with respect to the features analyzed.

ETH Network Stats and ETH Network Status allow measurement of a wide range of parameters within the ETH network. These parameters focus mostly on network status

(Table 5). Some of the parameters that can be measured are number of successfully mined blocks, presence of uncle blocks, mined time of the last mined block, average mined time, average network hash-rate, difficulty, active nodes, gas price, gas limit, page latency, uptime, node name, node type, node latency, peers connected to one node and some others. Figure 7 illustrates the tool in use, accessed from the browser from a private IP on port 3000.

As we mentioned Etherscan allows extraction of information relative to the blockchain (Table 5). Within the parameters that can be obtained are account balance, account info, transaction hash, block number, type of token (e.g., Erc20), average gas used, transaction costs and transaction fee.

Infura Dashboard allows obtaining a wide range of parameters such as: the total number of methods called, the bandwidth consumed by each of the methods used and the total bandwidth consumed. Therefore, the main feature measured is the bandwidth (Table 5).

As mentioned, the truffle test is a framework that allows running tests on smart contracts. For our case of use, the parameters we measure are time of data query, time and cost of data insertion and time of full test. In addition, these data are combined with the data obtained from contract migration process. Therefore, other parameters measures are gas and time spend to deploy the contract. The results obtained are presented below.

6. RESULTS

Based on the analysis performed in the previous section, the results are presented for each of the tools.

ETH Network State and ETH Network Status enables monitoring the network all the time. For example, at the time of analysis the Testnet Ropsten has mined 5,931,224 blocks, has 14 active nodes, the average block time is 14.04 s, the average network hashrate is 120.1 MH/s, and the difficulty is 2.16 GH. These parameters can be contrasted with those shown in Figure 7 for our locally deployed blockchain. For instance, at the time of analysis our local blockchain has mined 6967 blocks, has only one active node (our node), the average block time is 27.45 s, the average network hashrate is 142 KH/s and the difficulty is 1.43 MH. As a conclusion, it is visible that the power of mining and therefore the resources available to our local device are much less than those presented by the public network. This is an expected result.

Etherscan Ropsten Testnet Network ([41]), which allows us to have a view of all transactions that have been executed from our test address (e.g.,

0xe8d5487caebfb3f3e93304161cad0d5d3078b033). Other attributes that can be verified are the status of each of the transactions, the block where the transaction has been assigned, the gas percentage used (e.g., average gas used 66.67% of the established limit value), transaction costs and fee, as well as the nonce used in the PoW. Similar behaviors are obtained for the tool used locally.

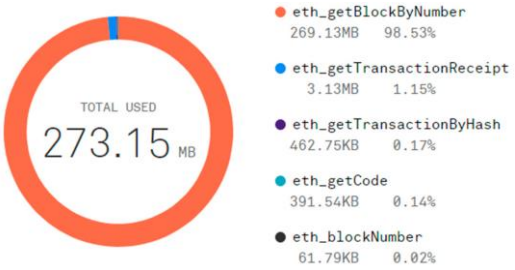

Figure 9: Infura dashboard tool: top five methods call bandwidth usage.

Figure 9 is taken from the Infura dashboard, and it details the main methods called by the web3.js library in order to interact with the blockchain via Smart Contract, as well as the bandwidth they spend. Clearly, there is a relationship between the method that infura detects and the method we use in our blockchain interface (BI) based on the truffle-drizzle framework, only that Infura perceives JSON (JavaScript Object Notation) RPC (Remote Procedure Call) API (https://github.com/ethereum/wiki/wiki/JSON-RPC) methods based on web3.js library and, since truffle-drizzle works with promises and web3.js works with callback, truffle-drizzle framework uses functions like cacheSend. Calling the cacheSend function on a contract will send the desired transaction and return a corresponding transaction hash so the status can be retrieved from the store. The procedure mentioned at web3.js level is performed by eth_getTransactionByHash, however since we work at a higher level, our cacheSend function agglutinates this and other methods. On the other hand, when

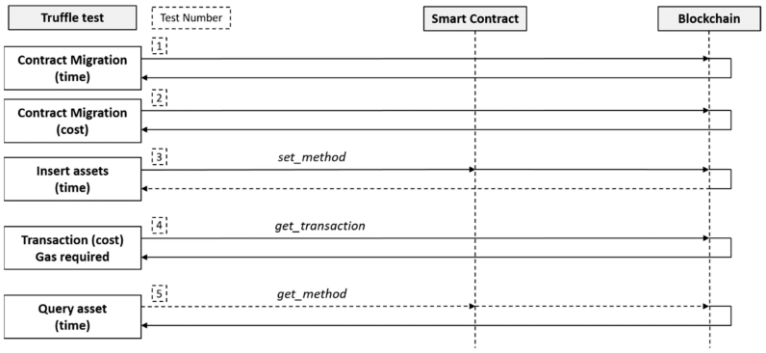

Figure 10: Sequence diagram of truffle test and contract migration.

performing debugging via node inspect tool on the migration process (truffle migrate), it is determined that both methods: getTransactionReceipt and eth_getCode are used. This highlights the importance of the Infura dashboard and details the consumptions made by the methods at a low level. In addition, Infura dashboard includes other relevant information such as peak (e.g., 183.33 MB) and average (e.g., 9.11 MB) hourly bandwidth usage and so on.

$$(\text{Local\_network\_time}/\text{Ropsten\_network\_time}) \times 100, \quad (7)$$

In order to examine our smart contract, we have established a combination of tests between the truffle test frame and the data obtained from the contract migration process. Therefore, Figure 10 shows the succession of the applied mechanism in order to check the feasibility. The data received are shown in Table 6, which compares data insertion time, data query, time gas used and time to pass the full test. Expression (7) is used to calculate the percent. Since, the times achieved are not deterministic, was taken both best and worst times. The test process performed is described below.

Table 6: Truffle test results, local network vs. Ropsten network.

Monitored Feature	Local	Ropsten	Local vs Ropsten (%)
Gas used to deploy	732,151	732,151	-
Cost to deploy the contract	0.01464302 ETH	0.01464302 ETH	-
Total migration cost	0.01987088 ETH	0.01987088 ETH	-
Contract Migration time	15,703 ms	225,001 ms	6.98 %
Data query (best time)	112 ms	855 ms	13.1%
Data query (worst time)	160 ms	1,705 ms	9.38%
Data insertion (best time)	184 ms	12,225 ms	1.505%
Data insertion (worst time)	205 ms	40,646 ms	0.46%
Gas used to data insertion	43,255	43,255	-
Time passing full test (best time)	814 ms	54,000 ms	1.507%
Time passing full test (worst time)	902 ms	180,000 ms	0.501%

Our model initiates by recovering the migration time from the contract, which is not deterministic, and it establishes a considerable delay between migrations in a local environment and migration in the Ropsten Testnet. Although it is a time to consider, it is not decisive to evaluate the feasibility of the system, since its implementation is prior to the deployment of the system. Because the same smart contract is employed in both environments, the costs and computational power required for the deployment match.

Another essential requirement is the insertion time of the asset attributes. The set_method is used to send attributes to the deployed smart contract, and it waits for blockchain

response. This procedure is equivalent to the mentioned one for inserting data via GUI (ABAC configuration sub-system, Figure 5). Although the delay between the local environment and the Ropsten Testnet is evident (Table 6), this metric will not cause a delay in the execution of the AC policy. As mentioned above, data insertion involves the generation of transactions and, therefore, associated costs, which is an indispensable measurement, so the get_transaction method is applied. We consider that since the same smart contracts deployed in different environments, the transaction cost is equivalent.

The decisive metric (delay requirement over AC policy) is the query of assets data. Therefore, it is essential to execute the get_method function and wait for the delay value (Table 6). For this reason, we can conclude that the implementation of our system is technically feasible. Since the Ropsten Testnet is less stable than the Mainnet because of the smaller number of nodes joining that network, and, therefore, less computational power, as discussed in tool one (ETH Network Status), these delays would be less in a Mainnet.

7. CONCLUSIONS

The growing adoption of RFID systems in healthcare is evident. Based on interviews with specialists we determine the implementations needs of a trust tracking and tracing system of medical assets. Our proposal is an access control system of a healthcare asset in order to prevent unwanted assets from entering the wrong area because of human error or an external threat. Therefore, it is a prevention system that aims to solve both security and safety risks. Traditional Access Control Systems are based on role-based access control (RBAC) and centralized architecture. From the technical point of view, our proposal consists of an attribute-based access control (ABAC) system for RFID systems that executes access control (AC) policies from a decentralized application (DApp) based on a blockchain architecture. This model is a proof of concept in both a local environment (single node) and in a public environment (Ropsten Testnet) and although the technological feasibility for its eventual production implementation is demonstrated, it requires a viable underlying business model. In order to demonstrate the implementation feasibility were used four recommended tools: ETH Network Status, Etherscan Ropsten Testnet Network, Infura dashboard and truffle test.

Future research lines are firstly, to establish a comparison between systems based on Hyperledger Fabric Blockchain and other with Ethereum blockchain. One of the common criteria in order to establish a comparison is the ABAC policy as part of the contract (Chaincode and Smart Contract). Secondly, to consider an application environment based on the public blockchain with a base on a tokenization environment. Thirdly, the creation of

a robust mutual authentication RFID protocol that works together with our ABAC blockchain system in order to build a secure supply chain system. Finally, to extend ABAC and RBAC blockchain concept to industrial manufacturing and automation environments. Recently, modbus.org has established security requirements, which include RBAC authentication based on X.509v3 certificates.

8. REFERENCES

[1] Wamba, S.F.; Anand, A.; Carter, L. A literature review of RFID-enabled healthcare applications and issues. Int. J. Inf. Manag. 2013, 33, 875–891.

[2] Thiesse, F.; Floerkemeier, C.; Harrison, M.; Michahelles, F.; Roduner, C. Technology, Standards, and Real-World Deployments of the EPC Network. IEEE Internet Comput. 2009, 13, 36–43.

[3] Guinard, D. GS1 blog series: Everythng and GS1 in a nutshell. 2017. Available online: https://evrythng.com/platform/evrythng-gs1-in-a-nutshell/ (accessed on 25 March 2019).

[4] GS1. GS1's Framework for the Design of Interoperable Traceability Systems for Supply Chains. GS1 Glob. Traceabil. Stand. 2017, 102, 1–58. Available online: https://www.gs1.org/sites/default/files/docs/traceability/GS1_Global_Traceability_Standard_i2.pdf (accessed on 25 March 2019).

[5] Hrabina, M. Taking Advantage of RFID's Expanding Role in Medical Devices. 2017. Available online: https://www.meddeviceonline.com/doc/taking-advantage-of-rfid-s-expanding-role-in-medical-devices-0001 (accessed on 10 March 2019).

[6] Figueroa Lorenzo, S.; Añorga Benito, J.; García Cardarelli, P.; Alberdi Garaia, J.; Arrizabalaga Juaristi, S. A comprehensive review of RFID and bluetooth security: Practical analysis. Technologies 2019, 7, 15.

[7] Griffin, P.H. Secure authentication on the Internet of Things. In Proceedings of the IEEE SoutheastCon 2017, Charlotte, NC, USA, 30 March–2 April 2017; pp. 1–5.

[8] Floerkemeier, C.; Roduner, C.; Lampe, M. RFID application development with the Accada middleware platform. IEEE Syst. J. 2007, 1, 82–94.

[9] The Global Language of Business. GS1, EPCglobal. 2019. Available online: https://www.gs1.org/epcglobal (accessed on 9 June 2019).

[10] EPCglobal. The Application Level Events (ALE) specification. Interface 2009, 1–229. Available online: https://www.gs1.org/sites/default/files/docs/epc/ale_1_1_1-standard-core-20090313.pdf (accessed on 9 June 2019).

[11] EPCglobal. The Application Level Events (ALE) Specification Version 1.1.1 Part II: XML and SOAP Bindings. 2009, pp. 1–119. Available online: https://www.gs1.org/sites/default/files/docs/epc/ale_1_1_1-standard-XMLandSOAPbindings-20090313.pdf (accessed on 9 June 2019).

[12] Tounsi, W.; Cuppens-Boulahia, N.; Cuppens, F.; Garcia-Alfaro, J. Fine-grained privacy control for the RFID middleware of EPCglobal networks. In Proceedings of the Fifth International Conference on Management of Emergent Digital EcoSystems, Luxembourg, 28–31 October 2013; pp. 60–67.

[13] Contu, R.; Kavanagh, K.M. Market Trends: Cloud-Based Security Services Market, worldwide. 2013. Available online: https://www.gartner.com/en/documents/2607617/market-trends-cloud-based-security-services-market-world (accessed on 9 June 2019).

[14] Ekran System. Role-Based Access Control vs Attribute-Based Access Control: How to Choose. 4 February 2019. Available online: https://www.ekransystem.com/en/blog/rbac-vs-abac (accessed on 11 June 2019).

[15] Hu, V.C.; Friedman, A.R.; Lang, A.J.; Cogdell, M.M.; Scarfone, K.; Kuhn, R. Guide to Attribute Based Access Control (ABAC) Definition and Considerations; Spec. Publ. 800-162; National Institute of Standards and Technology: Gaithersburg, MA, USA, 2019.

[16] Coyne, E.; Weil, T.R. ABAC and RBAC: Scalable, flexible, and auditable access management. IT Prof. 2013, 15, 14–16.

[17] Ong, M.T.; Sridharan, R.V.; Nakamura, J.; Ohmura, R.; Sidorov, M.; Khor, J.H. Ultralightweight mutual authentication RFID protocol for blockchain enabled supply chains. IEEE Access 2019, 7, 7273–7285.

[18] Ouaddah, H.; Mousannif, A.; Elkalam, A.; Ait Ouahman, A. Access control in the Internet of Things: Big challenges and new opportunities. Comput. Netw. 2017, 112, 237–262.

[19] Ouechtati, H.; Ben Azzouna, N. Trust-ABAC towards an access control system for the Internet of Things. In Proceedings of the International Conference on Green, Pervasive, and Cloud Computing, Cetara, Italy, 11–14 May 2017; pp. 75–89. [Google Scholar]

[20] Hemdi, M.; Deters, R. Using REST based protocol to enable ABAC within IoT systems. In Proceedings of the IEEE 7th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON), Vancouver, BC, Canada, 13–15 October 2016; pp. 1–7.

[21] Pervez, H.; Muneeb, M.; Irfan, M.U.; Haq, I.U. A comparative analysis of DAG-based blockchain architectures. In Proceedings of the 12th International Conference on Open-Source Systems and Technologies (ICOSST), Lahore, Pakistan, 19–21 December 2018; pp. 27–34.

[22] Gatteschi, V.; Lamberti, F.; Demartini, C.; Pranteda, C.; Santamaría, V. To Blockchain or not to Blockchain: That is the question. IT Prof. 2018, 20, 62–74.

[23] Di Francesco Maesa, D.; Mori, P.; Ricci, L. Blockchain based access control. Lect. Notes Comput. Sci. 2017, 10320 LNCS, 206–220.

[24] Hwang, D.; Choi, J.; Kim, K.H. Dynamic access control scheme for IoT devices using blockchain. In Proceedings of the International Conference on Information and Communication Technology Convergence (ICTC), Jeju, Korea, 17–19 October 2018; pp. 713–715.

[25] Cruz, P.; Kaji, Y.; Yanai, N. RBAC-SC: Role-based access control using smart contract. IEEE Access 2018, 6, 12240–12251.

[26] Zhu, Y.; Qin, Y.; Gan, G.; Shuai, Y.; Chu, W.C.C. TBAC: Transaction-based access control on blockchain for resource sharing with cryptographically decentralized authorization. In Proceedings of the IEEE 42nd Annual Computer Software and Applications Conference (COMPSAC), Tokyo, Japan, 23–27 July 2018; Volume 1, pp. 535–544.

[27] Gartner. Blockchain: What's Ahead? 2019. Available online: https://www.gartner.com/en/information-technology/insights/blockchain (accessed on 10 June 2019).

[28] Blockchain, C. Tokenization of Financial Assets: Financial Blockchain Revolution. 30 January 2019. Available online: https://medium.com/@credits/tokenization-of-financial-assets-financial-blockchain-revolution-bc632e75c8 (accessed on 10 June 2019).

[29] Yasri, D. Revolutionizing Healthcare with Tokenization. 2018. Available online: https://medium.com/pikciochain/revolutionizing-healthcare-with-tokenization-d4d36a2ca6fe (accessed on 10 June 2019).

[30] Sazandrishvili, G. Asset Tokenization on Blockchain Explained in Plain English. 19 May 2018. Available online: https://medium.com/coinmonks/asset-tokenization-on-blockchain-explained-in-plain-english-f4e4b5e26a6d (accessed on 10 June 2019).

[31] Samarati, P.; de Vimercati, S.C. Access Control Policies, Models, and Mechanisms. In International School on Foundations of Security Analysis and Design; Springer: Berlin/Heidelberg, Germany, 2011.

[32] Lorenzo, S.F.; Añorga, J.; Arrizabalaga, S. An Attribute-Based Access Control model in RFID systems based on blockchain Decentralized Applications for healthcare environments (video demonstration). 2019. Available online: https://zenodo.org/record/3339217 (accessed on 10 July 2019).

[33] Transcends. RIFIDI Community Wiki. 30 March 2019. Available online: http://wiki.rifidi.net/index.php/Main_Page (accessed on 11 June 2019).

[34] Serme, node-epc. GitHub, Romania, 2016; p. 1. Available online: https://github.com/sarme/node-epc (accessed on 10 June 2019).

[35] Cubedro, M.O. Ethereum Network Stats. GitHub, Romania, 2016. Available online: https://github.com/cubedro/eth-netstats (accessed on 10 June 2019).

[36] Cubedro, M.O. Ethereum Network Intelligence API. Romania, 2016. Available online: https://github.com/cubedro/eth-net-intelligence-api (accessed on 10 June 2019).

[37] llrp.org. LLRP Toolkit. 1 December 2008. Available online: http://llrp.org/ (accessed on 11 June 2019).

[38] Huebner, A.; Facchi, C.; Janicke, H. Rifidi toolkit: Virtuality for testing RFID systems. In Proceedings of the Seventh International Conference on Systems and Networks Communications (ICSNC 2012), Lisbon, Portugal, 18–23 November 2012; pp. 1–6.

[39] GitHub. Truffle Blockchain Group 2019. Drizzle. 2019, p. 1. Available online: https://github.com/trufflesuite/drizzle (accessed on 11 June 2019).

[40] Cubedro, M.O. Ropsten Stats. 2019. Available online: https://ropsten-stats.parity.io/ (accessed on 7 June 2019).

[41] Ethereum Community. Etherscan Ropsten Testnet Network. 2019. Available online: https://ropsten.etherscan.io/ (accessed on 13 June 2019).

[42] Wuehler, M. Infura Dashboard Update. Infura Blog. 2018. Available online: https://blog.infura.io/infura-dashboard-update-9f02d0643eb3 (accessed on 6 July 2019).

CHAPTER V

ALARM COLLECTOR IN SMART TRAIN BASED ON ETHEREUM BLOCKCHAIN EVENTS-LOG

This chapter was published in S. Figueroa-Lorenzo, Santiago; Goya, Jon; Añorga, Javier; Adin, Iñigo; Mendizabal, Jaizki; Arrizabalaga, "Alarm collector in Smart Train based on Ethereum blockchain events-log," IEEE Internet of Things Journal, vol. 08, no. 17, pp. 13306 – 13315, 2021, doi: doi.org/10.1109/JIOT.2021.3065631. [JCR. 9.471, Q1].

This chapter focuses on the design and implementation of an alarm collection system based on the emission of Ethereum Event-Logs over a smart train-based environment, which integrates both private data collection and alarm collection. The main objective pursued is to demonstrate the use of Event-Logs as the most efficient technology in terms of gas costs, as well as to determine the capacity of our system to manage a high number of concurrent alarms. This research arises from the need detected from the C_3 publication: " *An Attribute-Based Access Control Model in RFID Systems Based on Blockchain Decentralized Applications for Healthcare Environments*" as a solution to the problem of high transaction costs.

1. ABSTRACT

The European Union is moving towards the "Smart" era having as one of the key topics the Smart Mobility. What is more, the EU is moving towards Mobility as a Service (MaaS). The key concept behind MaaS is the capability to offer both the traveler's mobility and goods transport solutions based on travel needs. For example, unique payment methods, intermodal tickets, passenger services, freight transport services, etc. The introduction of new services implies the integration of many IoT sensors. At this point, security gains a key role in the railway sector. Considering an environment where sensor data is monitored from sensor-events, and alarms are detected and emitted when events contain an anomaly, this document proposes the development of an alarms collection system which ensures both traceability and privacy of these alarms. This system is based on Ethereum blockchain events-log, as an efficient storage mechanism, which guarantees that any railway entity can participate in the network, ensuring both entity security and information privacy.

2. INTRODUCTION

European Union's (EU) aim is to foster the railway sector to become an attractive transport method. The most remarkable example is Shift2Rail (S2R) Joint Undertaking (JU) initiative, where the EU promotes the competitiveness of the European rail industry and meets the EU transport needs. This work is carried out by joining efforts to seek focused research and innovation (R&I) and market-driven solutions, under H2020 initiative, to develop the necessary technology to complete the Single European Railway Area (SERA) [1]. Therefore, Shift2Rail activities should support the rapid and broad deployment of advanced traffic management and control systems, by offering improved functionalities and standardized interfaces, based on common operational concepts, facilitating the migration from legacy systems, decreasing overall costs, adapting it to the needs of the different rail segments as well as to the needs of a multimodal Smart Mobility system. In this regard, Smart Mobility concept includes, e.g., new payment methods, smart ticketing, intramodalities and multimodalities, smart stations, passenger services, etc. [2].

Blockchain is a disruptive technology that is playing a key role in several fields. At the beginning, it was associated with the financial sector, however, the appearance of smart contracts in Ethereum blockchain has opened a universe of possibilities for use cases such as smart mobility. Although the scientific community has developed use cases in railway

environments such as blockchain approach to digital ticketing, we can affirm that blockchain technology is in its infancy in both smart mobility and railways.

Considering that initiatives like S2R propose smart freight wagon concepts to face the main challenges in rail freight transport, for which the wagons are provided with a high number of sensors, a reliable recording mechanism for sensor-events is mandatory. Once these sensor-events are analyzed and overcome a defined threshold, alarm detection becomes a necessary mechanism to implement. However, this leads to the need to establish a solution that allows the reliable collection of alarms, ensuring traceability while maintaining the privacy of information. Considering that Ethereum blockchain provides properties such as immutability, security by cryptography and transparency in operations, the main contribution of this manuscript is to present an alarm collection system in smart trains based on Ethereum blockchain events-log, which involves the collection of these alarms. Our system provides alarm traceability, guaranteeing reliable information of each alarm, preserving information privacy.

The article is organized as follows. Section II reviews the blockchain applications for railway, as well as justifies both the selection of blockchain type according to our use case and Ethereum events-log. Section III proposes the alarm collection system based on Ethereum events-log, emphasizing in smart contract design. Section IV discusses the implementation criteria performed. Section V evaluates the proposal and analyses the results of the experiments conducted. Section VI discusses the limitations of our proposal as well as the possible security risks to consider. Section VII proposes countermeasures based on IEC-62443 and finally, conclusions and future research lines are addressed.

3. RELATED WORK

This section first, performs a comprehensive review about the use cases of blockchain in the railway context, followed by justification of blockchain type selection according to our use case, enabling differences, e.g., with permissioned blockchains. Finally, Ethereum events-log selection is also justified.

3.1. BLOCKCHAIN APPLICATION FOR RAILWAY

Rail is in a privileged position to become the backbone of an intermodal "Mobility as a Service" for passengers and "Delivery as a Service" for goods [3]. The analysis of new technologies could allow faster deployment of some of these initiatives and make the railway a more competitive and attractive transport system. Thus, one of the technologies that are

already improving the way of operating railways is the Internet of Things (IoT). The use of low power interconnected sensors and positioning systems have already improved the mobility concept by generating huge quantities of information which are mainly used by the railway operators and infrastructure managers, but also by the passengers.

All this means that being aware of the new technologies that are being game changers can provide a big push to Smart Mobility and particularly to the railway sector. In this regard, blockchain is one of these technologies that recently emerged and is providing new alternatives to cope with problems that were not solved yet.

3.1.1. E-Ticket

Theoretically, crypto-currencies should improve e-ticket systems through charges and taxes restructuration, smart pricing based on the use and proof of identity & loyalty programs. However, an e-ticket system based on crypto-currency has two major challenges ahead. On the one hand, it is susceptible to the current high volatility of the cryptocurrency market, as well as the number of transactions currently supported by Ethereum, i.e., currently, only about 7 transactions per second are performed in Ethereum, which produces a scalability problem. Despite the challenges ahead, the first steps in e-tickets are being taken through initiatives such as the Shenzhen metro in China, which recently has released an invoicing platform based on blockchain to check, verify, and trace each invoice, i.e., to manage the cycle of tax invoices [4]. Besides, the reference [5] presents a digital ticketing system based on the permissioned blockchain Hyperledger Fabric that distributes the tickets among all the participating organizations. An important contribution of this platform is that the governing organizations maintain both the right to establish the rules of use of the platform and to access the data for statistical purposes. Finally, despite not being an e-ticket system but a reward system, Go-Ahead Group PLc, a UK transport provider, joined the start-up DOVU to release a token reward system for its customers [6].

3.1.2. Traceability and asset management

Another important application where the blockchain is being directly applied in the railway industry and more particularly to the freight railways is the supply chain traceability. In this regard, the initiative as part of Australia's rail freight network, Pacific National has been using blockchain for supply chain management of perishable goods, with the aim of reducing the administrative burden of their businesses, offering the best services to their customers [7]. In addition, the reference [8] indicates that State Railway of Thailand (SRT) invests in both blockchain and IoT to develop a dedicated communications system to increase the accuracy of its railway itinerary. Another important contribution is the union of organizations such as

Canadian Pacific Railway (CPR) to the Standards Council of the Blockchain In Transport Alliance (BiTAS) [9], which aims to provide suitable information to enable the quick determination of a shipment's location [10].

3.1.3. Security, auditability, and accountability

Cybersecurity in the railway domain is a key issue. In this regard, considering that train accidents are potentially fatal and frequently related to bad practice on the tracks, the Swiss Government, in compliance with security standards and registration requirements, has been able to determine who has worked on the tracks and whether they were certified to do so, via the Swiss Federal Railways (SBB), which is applying blockchain to handle the digital employee identity system [11]. Also, the RailChain project [12] has a first use case dealing with the development of blockchain-based identities for assets such as individual vehicle components (e.g., brakes, individual sensors, etc.) to ensure the trustworthiness of the corresponding asset. The second use case of RailChain project [12] is the implementation of a data logger without real-time requirements, which allows data such as delay information or sensor readings and diagnostic messages are to be recorded and stored reliably and tamper-proof on a blockchain. The third use case of the RailChain project [12] is Juridical Recording Unit (JRU) based on blockchain like a solution with real-time requirements (<1 second block cycle time) for testing in a suitable test environment. In addition to the counterfeit protection of the data storage, attention is also paid to interoperability requirements between several manufacturers (e.g., standardized communication protocols). In this step, communication encryption for train control systems, secured by blockchain, as well as the topic of over the air software updates via blockchain must also be tested.

3.2. BLOCKCHAIN TYPE SELECTION JUSTIFICATION

Although the RailChain project, introduces evidence of the use of blockchain technology, this does not imply that blockchain is applicable in any railway context. Thus, important organizations, e.g., Gartner in the reference [13], indicate that the type of blockchain must be related with the use case, i.e., not every blockchain is applicable in every context. For that purpose, this section analyzes the selection of the appropriate blockchain in our railway environment. Although there are currently some blockchain types, we will basically focus on two types: permissioned blockchains and permissionless blockchains. Permissioned blockchains include projects such as Hyperledger Fabric Blockchain [14] oriented to scenarios of high transaction throughput performance and low latency of transaction confirmation. However, permissioned blockchains and particularly Hyperledger Fabric operates under a governance model, i.e., based on trust between participants. Considering the dispersion that

exists between the organizations that management the railways, it is not feasible to have a scenario where there is even partial trust between the parties, so representing a model based on governance is impossible under current conditions. Additionally, if we consider time constraints represent a requirement for the blockchain type selection, it should be noted that our use case focuses on alarm collection, i.e., we collect the alarm emitted for the purpose of providing traceability while maintaining the data privacy, which is an independent system of the decision that will be taken from this alarm, therefore, our system does not involve constraint environments. For this reason, we have selected the Ethereum blockchain, which will allow us to take advantage of the benefits offered by a public blockchain, which presents its greatest strength in the network size, as well as in the transparency of operations and information exchanged, in addition to maintain the blockchain properties such as security by cryptography, consensus algorithm, smart contract and immutability. It should be noted that to our use case the low transaction throughput and the resource consumption are not a constraint because, on the one hand, we are focusing on the alarm's traceability, i.e., a scenario without immediate decision. On the other hand, we do not run a full node or even a light node, but we use a provider and deploy an off-chain node for alarm collection purposes. Regarding the issue of transaction costs, our system is based on Ethereum events-log, the cheapest mechanism to use the blockchain. Next section details this Ethereum blockchain resource.

3.3. ETHEREUM BLOCKCHAIN EVENTS-LOG

In traditional environments, applications can use logs to capture and describe an instant. Logs can be used for instance, to notify that something happened. Thus, logs can also be used to describe an event (events-log) within a smart contract. The Ethereum Virtual Machine defines five opcodes for emitting events-log to create log records (LOG0, LOG1, LOG2, LOG3 and LOG4), which consist of both topics and data [15]. The opcode LOG0 is composed only by data and other opcodes contain both topics and data [15]. The major difference between the two is that topics are searchable, and data is not, but using data is cheaper than topics. To address the data search, simply trace back from the last Ethereum block where data has collected for a specific smart contract. Additionally, considering any transaction starts at 21000 gas, being gas, a measure of Ethereum work to perform transactions or any network interaction; a partial payment for a log operation is 375 gas, 8 gas is paid for each byte in a log operation's data and 375 gas is paid for each topic of a log operation [15]. Given the difference in costs between operations with topics and operations with data, and since that subsequent search of data in the logs is an efficient process, in our proposal we will use operations only with data. Considering, aforementioned (section 3.1) use cases are not in

production, but they are initiatives for future implementations or prototypes, which although reflect the community's spirit for blockchain use, they demonstrate that the adoption of blockchain in the railway sector is in its infancy. Additionally, blockchain-based Ethereum events-log systems mostly exist intrusion detection systems [16] or supply chain application [17], therefore, the contribution of our alarm collection system, based on the Ethereum event-log, which performs an alarms reliable traceability, ensuring data privacy, presents a high novelty.

4. PROPOSAL

This section analyses the design of our proposal. To do so, we analyze the general structure of the system's operation, as well as the entities involved. From there, we focus on the design of the three key pieces of our proposal: the smart contract, private data collection, and off-chain node.

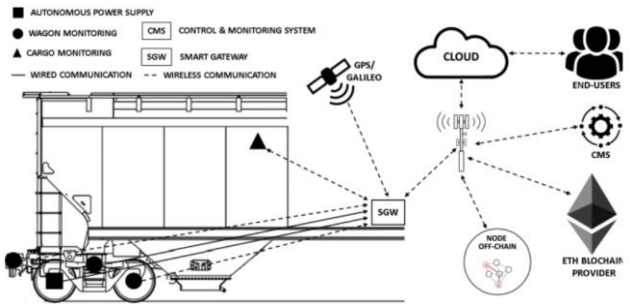

Figure 1: Overall structure of system operations.

4.1. OVERVIEW OF THE SYSTEM'S OPERATION

Figure 1 illustrates the general operation of the system. We have included a group of both wired and wireless sensors that are monitoring the cargo, the wagon systems as well as the autonomous power supply. This set of sensors constantly sends information, at different frequencies, by different media, i.e., wired, or wireless, via several protocols (e.g., CAN bus, RS232/422/485 interfaces), to Smart Gateway (SGW). The SGW concentrates the sensor data and sends it to both the Control and Monitoring System (CMS) and cloud infrastructure. For this purpose, IoT protocols such as MQTT, are used via cellular infrastructure (e.g., LTE). CMS, represents the entity responsible for receiving the events, comparing with the threshold, sending the alarm, and requesting the procedures execution.

Figure 2 involves participating entities of the Alarm Manager Module (AMM), which like other modules, is part of the SGW. The CMS Alarm Detection Module (CMS-ADM) belongs to

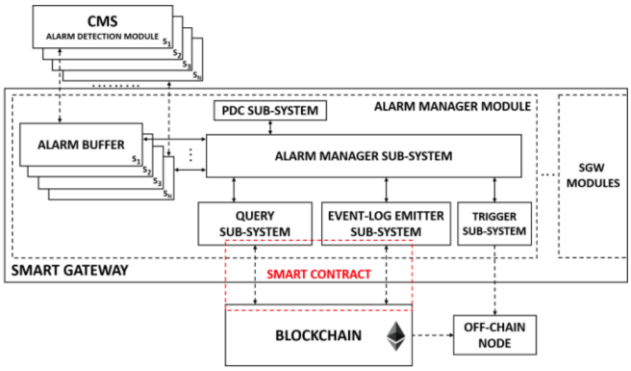

Figure 2: Alarm manager module.

the CMS entity of Figure 1. For our use case, the CMS-ADM constitutes a black box with only one function, sends alarms to Buffers, where S1, S2, …, SN represent a FIFO queue associated to each sensor (S1, S2, …, SN) that CMS has enabled. The SGW enables a connection with the blockchain provider, i.e., an entity that enables access to the blockchain network through a secure mechanism. Thus, sub-systems such as Query Sub-System (QS) and Events-Log Emitter Sub-System (EMS) interacts with Ethereum blockchain via smart contract, which is represented in Figure 2 as an abstraction, because once the smart contract has been deployed, it is fully contained in Ethereum blockchain, so it constitutes an interface between the sub-systems and the blockchain. The Alarm Manager Sub-System (AMS) integrates important functions: (1) reading alarms from Buffers; (2) verifying the correctness of alarm via QS; (3) forwarding alarms data to the Private Data Collection (PDC) Sub-System, the entity that ensures the information privacy; (4) emitting Ethereum event-log via EES to guarantee traceability and; (5) sending requests to off-chain node via Trigger Sub-System (TS), from which the off-chain node retrieves the events-log, ensuring a reliable alarm traceability collection.

In the remainder of the section, we focus on the smart contract design, private data collection design and off-chain node design.

4.2. SMART CONTRACT DESIGN

Figure 2 illustrates the smart contract interaction with both sub-systems QS and EES. Considering smart contract as the key piece of the alarm collection system in this section we focus on design criteria. On the one hand, it provides to QS the capacity to verify the alarm that CMS has sent to the alarm buffer. For this purpose, when contract is deployed, the relation between the sensors identification ($sensor_{ID}$) and the alarm thresholds are defined. On the other hand, it provides the capacity to EES to emit the events-log for each

alarm verified. The following guidelines have been applied: (1) the alarm thresholds are deployed in the constructor, which implies that they are initialized and cannot be intentionally modified since access is allowed only to get and set functions, which can only be executed by the Externally Owned Account (EOA) that deployed the smart contract; (2) the establishment of run-time limits based on the retrieval of the number of blocks mined at the time of the constructor's deployment, to establish the time limits within which a method can be executed; (3) modifiers constitutes a mechanism to apply the criteria (1) and (2), which represent a condition applied before calling a function, relying on properties such as inheritance and use of the "require()" function to avoid redundant-code and possible errors; (4) the use of bytes-type variables to reduce unnecessary storage costs, i.e., the fixed-size byte arrays define a variable by using the keyword "bytesX" where X represents the sequence of bytes from 1 up to 32. For instance, sensor and alarm identifiers are merged in bytes32 data type. Thus, $sensor_{ID}$ uses bytes12 data type and $alarm_{ID}$ uses bytes20 data type; (5) since the aim of our system is to leave a verifiable alarm mark, i.e., only queried for further analysis, the smart contract emits events-log instead of recording all information to reduce gas costs. To this end, we enable a key-value pair between the alarm and sensor identifiers to store relevant information of the alarms, maintaining low levels of blockchain transaction cost. As we mentioned in **section 3.1.3**, events-log can include topics, these will not be declared because when events-log are emitted even through these allow filtering and searching for attributes in a more detailed way, it is more expensive than using a raw data format, so the off-chain node handles this processing. Although the events-log are not properly stored in the ledger, they are reproducible, verifiable, and identifiable due to the use of blooming filters to receive significant blocks in the blockchain [18]. The smart contract presents three main functions: *getThreshold*, *setThreshold* and *triggerAlarmHash*. The first two to receive and send alarms from the constructor and the third to emit events-log. The *triggerAlarmHash* function contains the alarmTrigger event which receives two data types bytes32 as arguments. The first represents the combination of $sensor_{ID}$ and $alarm_{ID}$ (to determine $Event_{ID}$), while the second represents the $HASH_{SHA256}$ returned by PDC.

4.3. PRIVATE DATA COLLECTION DESIGN

Private Data Collection (PDC) is a key sub-system of AMM, which includes the functionality of to guarantee information privacy of alarms. To achieve this goal, PDC applies a hash operation over alarm data. However, PDC not only computes hash but as well, stores the information, creating two class of storage: short term and long term. Short term storage constitutes a previous step used by EMM to perform transactions sequentially to avoid errors. Once transactions are performed, inputs are deleted from short-term storage to

retrieve the proper indexed transaction, considering that events-log are public, and this information will be available. Short term uses a key-value pair, relating the event identification ($Event_{ID}$) with computed hash ($HASH_{SHA256}$). Additionally, long-term storage guarantees data privacy also in form of key-value pair, associating the hash computed ($HASH_{SHA256}$) and the alarm object ($alarm_{object}$).

4.4. OFF-CHAIN NODE DESIGN

Off-chain node is a sub-system independent of AMM, but with a key role in our system. The main function of the off-chain node is to recover alarms events-log to provide a reliable alarms traceability, for which, not only it queries events-log in blockchain but also organizes event-log data once collected. Thus, one key point is to determine, which useful information must be retrieved from blockchain to guarantee both an efficient traceability, as well as a successful operation of the off-chain node. Thus, $Event_{ID}$ and $Hash_{SHA256}$ are mandatories values since, they enable an appropriate traceability, provided that the entity accessing this data is authorized in the PDC. In addition, the log_{ID} and the last block number are data used by off-chain node to operate properly. Log_{ID} represents the identifier to each emitted event. Additionally, last block number indicates, the block number included in blockchain at the time of emitting the event. Both data, log_{ID} and last block number allow to determine the following log to be collected. It should be noted, data are collected in long-term storage.

5. IMPLEMENTATION OF ALARM COLLECTION SYSTEM

This section includes information about the designed architecture, as well as define the three main phases of the alarm collection system, i.e., data privacy collection, event-log emission, and alarm collection. Additionally, this section analyzes the implementation of the sub-systems, which compose AMM such as alarm buffer and EES. Finally, it analyzes the implementation of the off-chain node.

5.1. SETTING UP THE ENVIRONMENT

To set-up our system, a railway entity, with an EOA, must include the Alarm Manager Module, in the SGW, ensuring decentralization. Additionally, the off-chain node must be included as an external entity. Our environment starts when buffer receives an alarm from CMS. However, we consider useful clarify the process to better understanding. The first criteria are sensors included along the wagon. For instance, orientus ([19]) is a ruggedized orientation sensor, which combine accelerometers, gyroscopes, and magnetometers to deliver accurate orientation. In this regard, reference [20] includes the dataset used, including

variables used in our implementation such as velocity, acceleration, and pitch. From Figure 2, a sensorID is associated to each alarm buffer between S_1-S_N. Thus, the CMS constantly processes the information, i.e., sensor-events, received from orientus, determining whether an alarm is generated. In this regard, alarms can be generated for each one of variables sent by orientus in each interval time. When an alarm is detected the JavaScript Object Number (JSON) alarmobject is sent to the corresponding alarm buffer (S_1-S_N). The first two fields of this alarmobject contain the alarmID and an alarm index indicating the variable from which the alarm was generated. Since the objective of AMS is alarm traceability, alarm buffer puts data at rest, while AMS extract them sequentially following a FIFO queue. When the contract is deployed (Section 4.2), a mapping between a bytes12 and a dynamic array is included in the constructor associating to each sensorID (bytes12) the alarmindex (dynamic array) that represents the threshold corresponding to the sensor instruments.

Table 1: Hardware and software environments.

Hardware		
CPU	i5-6500 3.20GHz	
Memory	8G DDR4	
Hard Disk	500 GB	
Sensor	AIMS Navigation IMU 0821111408	
Software		
Name	**Version**	**Applications**
OS	Ubuntu Server 18.04	
Node-Express	v4.17.1	AMS, EES, OFF-CHAIN
Node-Axios	v0.19.0	TS
Node-Crypto	v13.10.1	AMS
Node-Postgres	v7.18.2	AMS, EES, OFF-CHAIN
Node-File Sys	v13.12	ALARM BUFFER
Web3.js	v1.2.6	EES, QS
Truffle	v5.1.19	SMART CONTRACT
Solidity	>=v0.4.22 <v0.7.0	SMART CONTRACT
Testnet Ropsten	Infura.io	SMART CONTRACT
Postgres	v9.4.26	PDC, OFF-CHAIN
Postman	v7.21.1	AMS, EES, OFF-CHAIN

Once, boundary conditions are established, we define, in Table 1 both, hardware and software resources. Considering Figure 2, as reference architecture, Table 1 shows the relationship between technologies and applications, considering that a technology is used more than once, e.g., AMS, EES and OFF-CHAIN use Node-Express, Node-Postgres and Postman. To implement and migrate the smart contract we have used truffle, a development environment for blockchain and finally to deploy the contract we have used an Infura Ropsten provider. The application process is performed by the Alarm Manager Module, and

it is divided mainly into generating of private data collection and event-log emission. The flowcharts of both processes are represented in Figure 3 and Figure 4, respectively. Additionally, Figure 5 represents a flowchart of off-chain node implementation. To achieve the linearity shown in the flow charts, the resource async/await is used as part the JavaScript implementation.

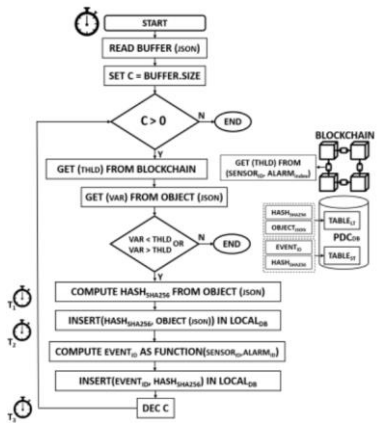

Figure 3: Private Data Collection flow chart.

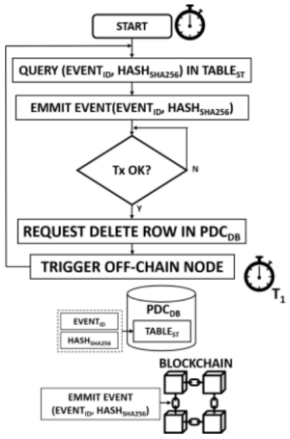

Figure 4: Event-log emission flow chart.

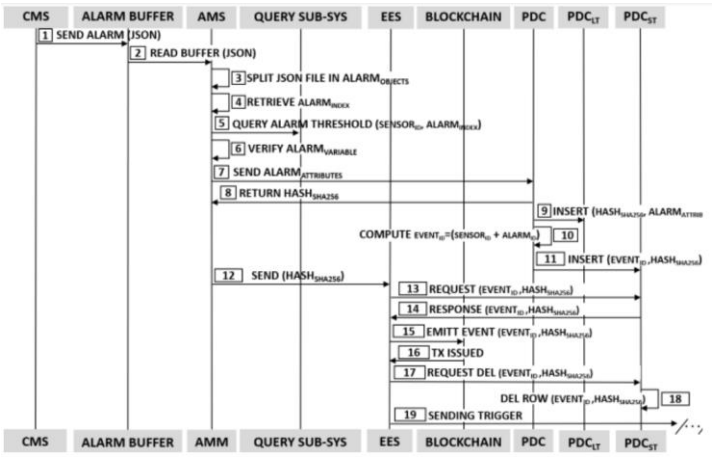

Figure 5: Workflow Event-Log Emission.

5.2. PRIVATE DATA COLLECTION IMPLEMENTATION

Figure 3 shows the flowchart of the PDC process. Since data is at rest in the alarm buffer in JSON format, we read them, determining the dimension of this alarm buffer, i.e., the number of objects contained in the file, at the time of reading it and thus, to establish a counter that

will be progressively decreased. At this point, each object ($alarm_{OBJECT}$) is retrieved sequentially. From sensorID and alarmindex we recover the alarm threshold to compare indexed variable in the object. Once the alarm is verified, the object is sent to PDC which returns the $HASH_{SHA256}$ associated. Next, PDC stores the key-value pair $Hash_{SHA256}$-$Alarm_{OBJECT}$ in long-term storage ($Table_{LT}$). Additionally, PDC unifies sensorID and $alarm_{ID}$, creating an $Event_{ID}$ of 32 bytes. Finally, PDC stores another key-value pair comprised of $Event_{ID}$ and $Hash_{SHA256}$ in a short-term storage ($Table_{ST}$). Now, the counter is decreased, and a new object is retrieved. The exposed mechanism ensure information privacy since the alarm susceptible information remains private and available to authorized roles.

5.3. EVENT-LOG EMISSION IMPLEMENTATION

Figure 5 shows the flowchart of the alarm emission based on blockchain events. The system designed can manage the concurrent alarms sent to short-term storage ($Table_{ST}$). This mechanism avoids using a transaction batch and possible transaction errors. EES interacts with the blockchain via smart contract, as well as the Trigger Sub-System and PDC. $Table_{ST}$ is managed as FIFO queue and EES recovers key-value pair ($Event_{ID}$, $Hash_{SHA256}$) and invokes a transaction. At this point, async/await functions are used to wait (Tx OK) for the callback of transaction (Tx) and therefore the correct event-log emission without stressing the channel. Once the transaction is successfully performed, EES request to PDC to delete row of data emitted. Finally, the trigger is issued to off-chain node. Additionally, Figure 5 provides a process perspective, focused the integration of the sub-systems. The workflow of the entire process starts when a sensor alarm is detected by CMS, sending it to the alarm buffer until event-log is emitted once the alarm is verified, passing through private data collection process. First, CMS sends alarmobject identified with alarmID to specific alarm buffer identified with sensorID (step 1). AMS reads sequentially each alarm buffer (step 2), always waiting to finish with one to start the next one. When AMS receives JSON object array from a specific buffer, processes sequentially object by object (step 3), analyzing the second attribute which contain an $alarm_{index}$ (step 4), which indicates the variable affected, thus AMS verifies correctness of alarm retrieving threshold value from blockchain via *getThreshold* smart contract method, using as argument sensorID and $alarm_{index}$ (step 5-6). Next, AMS sends $alarm_{OBJECT}$ to PDC, which returns a HashSHA256 (steps 7-8) and collect the alarm attributes in a long-term storage (step 9). PDC also collects in a short-term storage a key-value pair formed by EventID and HashSHA256 (step 10-11). Next, EES requests eventID and $Hash_{SHA256}$ to PDC short-term storage (step 13-14) and invokes the *triggerAlarmHash* method, which emits the event using both EventID and $Hash_{SHA256}$ as arguments (steps 15-16). At this point, EES sends a request to PDC to delete the row associated to last EventID

and $Hash_{SHA256}$ recorded in blockchain (step 17-18) and finally EES sends the trigger to the off-chain node (step 19). This implementation ensures reliable traceability by leaving a mark on the blockchain.

5.4 ALARM COLLECTION IMPLEMENTATION

Alarm collection is the third of the main functions mentioned, and the only one which does not belong to AMM, but to the off-chain node. Alarm collection recovers events-log emitted, and organizes them locally, improving the traceability process due to efficient collection of the alarms emitted. Figure 6, contains the flowchart that characterizes the alarm collection process. Considering the off-chain node was that was introduced in Figure 2, alarm collection is performed from the event-log collection. As Figure 6 shows, off-chain node is listening for a trigger emitted by the Trigger Sub-System (Figure 2). Then, off-chain node storage is queried to retrieve the block associated to the last stored event. To start the event collection, the blockchain is queried to retrieve the last block added. The number of events to be collected depends on the number of pending events between the time interval of the last stored block and the last block added. Once the event pool has been recovered, a counter (C) is initialized, which will be increased and compared with the pool size. For each event, the identifier (log_{ID}), the block in which it was emitted, the EventID and the associated $Hash_{SHA256}$ are retrieved. These attributes are inserted in a long-term storage, from where another authorized system, will be able to retrieve the data, i.e., obtain the EventID and the associated $Hash_{SHA256}$ to query to PDC to retrieve the corresponding attributes.

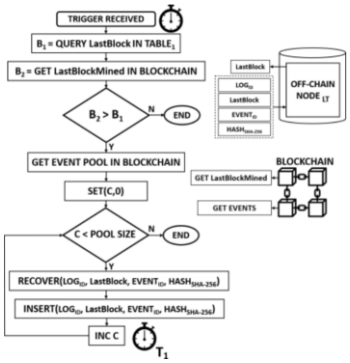

Figure 6: Alarm collection flow chart.

6. EVALUATION AND RESULTS

To demonstrate our proposal viability, we evaluate implementation of both Alarm Manager Module and off-chain node. Thus, on the one hand, we start evaluating different smart

contract implementation to demonstrate that our selection is more efficient with respect to gas consumption. On the other hand, we analyze the delays caused by the implemented systems, illustrated in Figure 3, Figure 4, and Figure 6 respectively, demonstrating the viability of the proposal concerning the private data collection, event-log emission, and alarms collection.

6.1. SMART CONTRACT EVALUATION

As we have mentioned, costs are important criteria in the use of the public blockchain, from which we have derived the main mechanisms implemented to save gas, i.e., the appropriate use of the variables in the smart contract and the emission of events-log. Thus, four smart contracts implementations are compared in Figure 7. From left to right, we find: (1) cost of the transaction using a regular method, i.e., using the blockchain to store the EventID and associated $HASH_{SHA256}$, for which, mappings of "uint256" and "string" as data types are used; (2) the transaction cost of the same method and strategy, but changing data types to "bytes32", i.e., using a more appropriate way to manage data, but again, using blockchain as storage system; (3) the transaction cost to emit an event using like variables EventID and $HASH_{SHA256}$, defining data types "uint256" and “string” respectively and (4) the transaction cost to emit an event, using like variables EventID and $HASH_{SHA256}$, with data types: “bytes32” to both cases. Considering the analysis made in section 3.1.3 where the gas consumption base was established when events are emitted in 21000 gas, we can affirm that the contract deployed in (4) is highly efficient in terms of cost, since, each time an event is emitted, our system will spend 21750 gas.

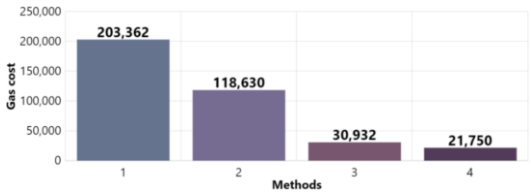

Figure 7: Smart Contract Evaluation.

6.2. PRIVATE DATA COLLECTION EFFICIENCY

As illustrated in Figure 1 and Figure 2 private data collection plays a key role in Alarm Manager Module. However, Figure 3 illustrates that PDC integrates several procedures such as two types of storage: long-term and short term, which can introduce delays. Thus, to determine impact of in different points of PDC implementation will allow to demonstrate solution viability. As Figure 3 shows, three classes of timers are used to measure delays in different

steps: (1) delay from the start until the $HASH_{SHA256}$ is computed; (2) delay from the start until long-term storage is used and (3) the delay from the start until the end, including EventID computation, as well as short-term storage. In our code, we use the JavaScript Date.now() function and the calculation of the difference between the times. Figure 8 illustrates viability of implementation, since the most complex procedure introduces only 73 ms of delay.

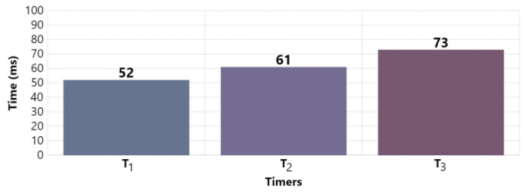

Figure 8: Delay of private data collection from timers of **Figure 3**.

Considering that the procedures performed by PDC are sequential, we consider relevant to analyze the efficiency of the PDC not only by collecting a single alarm, but also by performing the same procedure for higher alarm buffer sizes. Figure 9 shows the results of the experiment measuring the delays for the entire process, i.e., T3, according to Figure 3, for the alarm buffer sizes: 1, 10, 100, 1000, 10000. The results confirm the viability of the process and demonstrate an increase in efficiency as the buffer size is higher.

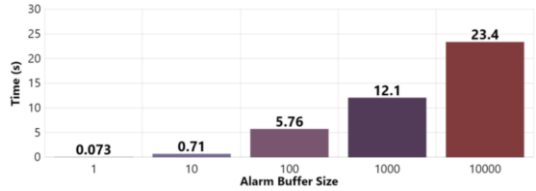

Figure 9: Delay of private data collection for different alarm buffer sizes.

6.3. CONCURRENT ALARMS

Our system has been designed to work sequentially, so that it does not invoke a transaction to emit a new event-log, without having finished the previous one, which also depends on the short-term storage of the PDC. Hence, it is necessary to analyze the viability of the proposal at this point, for which, the concept of alarm concurrence is used, which consists in assuming that we have in the alarm buffer a certain number of alarms to be emitted. Considering that PDC works as an independent sub-system to EES, these alarms have been previously processed by PDC. Thus, EES acquires information directly from short-term storage. To conduct the experiments, we relate the alarms amounts to those established in Figure 9, i.e., 1, 10, 100, 1000, 10000. In addition, Figure 4 contains a timer to illustrate the

measuring time interval also by using JavaScript Date.now() function. The results shown in Figure 10, demonstrate proposal viability, so that when the number of alarms to be issued increases, the efficiency of the system also increases. Additionally, we consider that in our use case the purpose of the event-log is to ensure traceability and not real-time decisions.

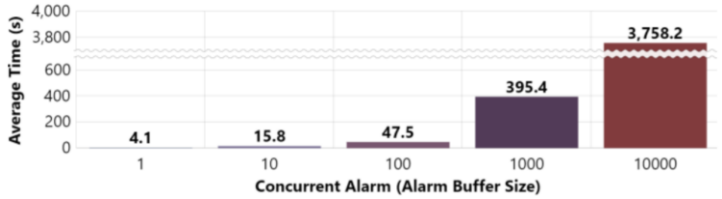

Figure 10: Concurrent Alarms.

6.4. EXAMINING EMITTED EVENTS

The off-chain node participates in the alarm collection processes as indicated in section 5.3, hence demonstrating its correct operation and viability is relevant. The logic used during the event-log emission consists in sending a trigger once the event-log is emitted. Hence, as Figure 6 shows, to demonstrate the event collection viability, it would be enough to analyze the time required to collect an event. However, to demonstrate the efficiency of the process, it is necessary to vary the size of the pool of events to be collected to determine whether it is more feasible to collect one by one or to wait for a certain size of the pool to perform the collection. Timer in Figure 6 illustrates time interval to be measured again by JavaScript Date.now() function.

Figure 11 compares off-chain node delays to different pool sizes, starting in one, i.e., every time an event is emitted it is collected. For the following pool sizes: 5, 10, 50, 100 and 1000, we collect request sent from Trigger Sub-System and requests an increment counter. Figure 11 illustrates results when comparing different pool sizes (Event Pool). An event collection, i.e., a pool size equal to 1, in 224 ms, allows the demonstration of the feasibility of implementation. Additionally, as the number of pools increases, the efficiency of the process increases. Thus, collecting one event each time it is emitted requires 224 ms, while collecting five events requires 312 ms, which is an increase of 88 ms, and five more events have been

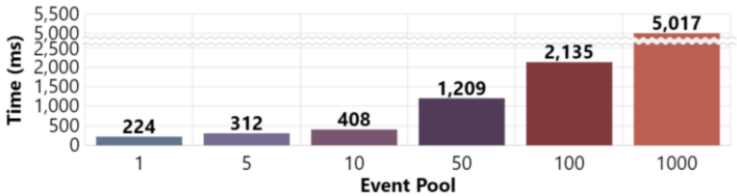

Figure 11: Time of collected events for some event pools.

collected. This same trend is maintained for the rest of the pools size, so we consider that depending on the use case, it may be more convenient to increase the pool size.

7. LIMITATIONS AND SECURITY RISKS

Although the viability of this proposal has been demonstrated, it presents a limitation in terms of costs, despite events are the most efficient form of "storage" in Ethereum blockchain, as we demonstrate in section 6.1. One solution to avoid transaction costs could be to join an alliance-type network based on the Ethereum Quorum, e.g., the Alastria project [21] since it includes high transaction throughput, increasing the number of transactions per second from dozens to hundreds per second and technological compatibility with Ethereum projects.

When analyzing Figure 1: Overall structure of system operations., it is necessary to mention the security risks our system have. On the one hand, we assume trusted system inputs, e.g., orientus sensors. However, we are aware that Gyroscopes may be susceptible to Microelectromechanical systems (MEMS) ultrasound attacks [22] or the GPS may be susceptible to spoofing attack [23], in this way the Control and Monitoring System is susceptible to attacks and therefore to the introduction of false positives. On the other hand, Smart Gateway integrates internal connections between Node.JS servers, which are assumed secure, but are susceptible to Denial of Service, JSON injection and Traceless Routing Hijacking [24] Additionally, communications between entities, i.e., Smart Gateway, Control and Monitoring System, blockchain provider and off-chain node via trusted cellular infrastructures, however reference [25] details vulnerabilities detected in 4G/LTE and 5G networks. Finally, communication via Ethereum blockchain provider are also assumed to be trusted. However, blockchain provider has reported threats of falsification of both identity and communication channels [26].

8. COUNTERMEASURES BASED ON SMART GATEWAY COMPLYING WITH IEC 62443

Considering the risks analyzed in Section 7, we consider that it is mandatory to establish security recommendations, which are contextualized to our case of use, so we will put the focus on the Smart Gateway. In this regard, although perimeter security levels are added in smart trains, if there is a critical aspect, it must be ensured from the origin. A countermeasure to avoid several of the risks mentioned in the previous section is to develop a Smart Gateway compliant with the IEC 62443 standard. Some of the measures implemented by a compliant SGW are (1) employ identity management system; (2) integrate access control system in

order to request identification, authentication and authorization for administration process; (3) avoid hardcode password; (4) create password-less system; (5) consider fully outgoing connections; (6) prevent system from being visible from outside; (7) establish a recovery process and maintain backups of recorded data; (8) encrypt data network, based on TLS; (9) encrypt data in device, strong encryption to store sensible data based on NodeJS key-store; (10) limit physical access to communication network; (11) set up intrusion detection solution to detect hijacking attempts or other advanced measures preventing spoofing and (12) include the EOA in the SGW together with its private key properly secured, so it has the ability to interact with the blockchain provider (Testnet Ropsten, Table 1), which enables a security mechanism to include the SGW EOA as part of the address whitelist [26].

9. CONCLUSIONS

This manuscript analyses the use cases of blockchain technology in railway environments, emphasizing three applications: e-Ticket, Traceability and Asset Management, and Security and Privacy. Considering our application context based on the smart wagon, we present the selection criteria of Ethereum as blockchain to implement our alarm collection system based on the event-log emission, which involves the private data collection, and alarm collection. In order to demonstrate the viability of the alarm manager sub-system, several experiments were performed, allowing (1) to justify the use of events-log as the most efficient technology in terms of gas costs, (2) to determine the capacity of our system to manage a high number of concurrent alarms, (3) to demonstrate viability of the private data collection and therefore guaranteeing data privacy, (4) to demonstrate the viability of event-log emission, which provides alarm traceability and (5) to demonstrate efficiency of alarm collection. Finally, limitations, security risks and countermeasures based on IEC-62443 are defined.

Future research lines aim to integrate our system with a blockchain-based access control system, such as proposed by references [27] and [28], as well as to develop smart train use cases on a permissioned blockchain, such as, Hyperledger Fabric, pluggable in constrained environments [29].

10. ACKNOWLEDGEMENTS

This research was supported in part by the Basque Government (Elkartek program) through projects “SENDAI-SEgurtasun integrala iNDustria AdImentsurako” with project number KK-2019/00072 and “TRUSTIND-Creating Trust in the Industrial Digital Transformation” with project number KK-2020/00054.

11. REFERENCES

[1] Shift2Rail, "About - Shift2Rail," 2017.

[2] Shift2Rail, "Shift2Rail Master Plan (v1.0)," 2015.

[3] U. M. N.Mazzino (Anslado STS), X.Perez (CAF), M. B. (RSSB) (Deutsche Bahn), R.Santoro (Ferrovie dello Stato Italiane), L. D. (UITP) J.Schlaht (Siemens), C.Chéron (SNCF), H.Samson (Strukton Rail), and C. H. (UITP) N.Furio (UNIFE), "Rail 2050 Vision: Rail - The Backbone of Europe's Mobility," 2017.

[4] O. O. Emmanuel, "Shenzhen Adopts Blockchain Technology for Transportation," 2019. [Online]. Available: https://bit.ly/33Vw9PH. [Accessed: 15-Mar-2020].

[5] J. D. Preece and J. M. Easton, "Blockchain Technology as a Mechanism for Digital Railway Ticketing," in 2019 IEEE International Conference on Big Data (Big Data), 2019, pp. 3599–3606, doi: doi.org/10.1109/BigData47090.2019.9006293.

[6] F. Memoria, "Leading UK Public Transport Provider to Launch Blockchain-Based Loyalty Program," 2019. [Online]. Available: https://www.cryptoglobe.com/latest/2019/02/leading-uk-public-transport-provider-to-launch-blockchain-based-loyalty-program/. [Accessed: 15-Mar-2020].

[7] O. Probert, "Pacific National takes part in experimental blockchain delivery," 2018. [Online]. Available: https://www.railexpress.com.au/pacific-national-takes-part-in-experimental-blockchain-delivery/. [Accessed: 15-Mar-2020].

[8] Asia Blockchain Review, "Thailand: The Land of Blockchain," 2019. [Online]. Available: https://www.asiablockchainreview.com/thailand-the-land-of-blockchain/. [Accessed: 15-Mar-2020].

[9] R. Kastelein, "Canadian Pacific Railways Joins Blockchain in Transport Alliance (BiTA)," 2019. [Online]. Available: https://medium.com/@expathos/canadian-pacific-railways-joins-blockchain-in-transport-alliance-bita-771e810ac267. [Accessed: 15-Mar-2020].

[10] B. Kothari, "BiTAS Tracking Data Framework Profile," 2019.

[11] M. V. Rodríguez, "Swiss Federal Railways Develop a Digital Identity Pilot on Ethereum," 2018. [Online]. Available: https://www.cryptoworldjournal.com/swiss-federal-railways-develop-a-digital-identity-pilot-on-ethereum/. [Accessed: 15-Mar-2020].

[12] Federal Ministry of Transport and Digital Infrastructure (BMVI), "RailChain," 2019. [Online]. Available: https://railchain.berlin/.

[13] R. Kandaswamy and D. Furlonger, "Blockchain Technology Spectrum," Stamford, USA, 2019.

[14] Hyperledger, "hyperledger-fabricdocs Documentation, Release master," 2019.

[15] G. Wood, "Ethereum: a secure decentralized generalized transaction ledger," Ethereum Proj. Yellow Pap., pp. 1–32, 2014, doi: doi.org/10.1017/CBO9781107415324.004.

[16] R. M. A. Ujjan, Z. Pervez, and K. Dahal, "Snort Based Collaborative Intrusion Detection System Using Blockchain in SDN," in 2019 13th International Conference on Software, Knowledge, Information Management and Applications (SKIMA), 2019, pp. 1–8, doi: doi.org/10.1109/SKIMA47702.2019.8982413.

[17] M. Westerkamp, F. Victor, and A. Küpper, "Blockchain-Based Supply Chain Traceability: Token Recipes Model Manufacturing Processes," in 2018 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData), 2018, pp. 1595–1602, doi: doi.org/10.1109/Cybermatics_2018.2018.00267.

[18] V. Buterin, "A next-generation smart contract and decentralized application platform," Ethereum, no. January, pp. 1–36, 2014, doi: doi.org/10.5663/aps.v1i1.10138.

[19] Advanced Navigation Pty, "Orientus Reference Manual," 2013.

[20] S. Figueroa-Lorenzo, "Alarm collector in Smart Train based on Ethereum blockchain events-log." Zenodo, 2020, doi: doi.org/10.5281/zenodo.3961294.

[21] A. Junio, "Memoria de actividades," Cuad. Relac. Laborales, vol. 20, no. 2, pp. 457-463–463, 2019.

[22] A. Pan, B. Yang, S. LI, W. Kang, and W. Zhengbo, "Sonic Gun to Smart Devices: Your Devices Lose Control Under Ultrasound/Sound," Black Hat USA, 2017.

[23] K. Wang, S. Chen, and A. Pan, "Is Your TimeSpace Safe?," Black Hat EU, 2015.

[24] M. Siman, "The Node.js Highway : Attacks are at Full Throttle," in Black Hat EU, 2015, p. 43.

[25] S. Figueroa-Lorenzo, J. Añorga, and S. Arrizabalaga, "A Survey of IIoT Protocols: A Measure of Vulnerability Risk Analysis Based on CVSS," ACM Comput. Surv., vol. 53, no. 2, 2020, doi: doi.org/10.1145/3381038.

[26] M. Godsey, "API Security section now available in the Infura Dashboard," 2019. [Online]. Available: https://blog.infura.io/api-security-section-now-available-in-the-infura-dashboard-9f8cfae64044/. [Accessed: 29-Mar-2020].

[27] Y. Zhang, S. Kasahara, Y. Shen, X. Jiang, and J. Wan, "Smart Contract-Based Access Control for the Internet of Things," pp. 1–11, 2018, doi: doi.org/10.1109/JIOT.2018.2847705.

[28] S. Figueroa, J. Añorga, and S. Arrizabalaga, "An Attribute-Based Access Control Model in RFID Systems Based on Blockchain Decentralized Applications for Healthcare Environments," Computers, vol. 8, no. 3, 2019, doi: doi.org/10.3390/computers8030057.

[29] S. Figueroa-Lorenzo, J. Añorga, and S. Arrizabalaga, "Methodological performance analysis applied to a novel IIoT access control system based on permissioned blockchain," Information Processing & Management., vol. 58, no. 4, p. 102558, 2021, doi: doi.org/10.1016/j.ipm.2021.102558.

CHAPTER VI

METHODOLOGICAL PERFORMANCE ANALYSIS APPLIED TO A NOVEL IIOT ACCESS CONTROL SYSTEM BASED ON PERMISSIONED BLOCKCHAIN

This chapter was published in S. Figueroa-Lorenzo, J. Añorga, and S. Arrizabalaga, "Methodological performance analysis applied to a novel IIoT access control system based on permissioned blockchain," Information Processing & Management., vol. 58, no. 4, p. 102558, 2021, doi: doi.org/10.1016/j.ipm.2021.102558. [JCR. 6.222, Q1].

This chapter focuses on the design and development of an on-chain Access Control System based on Hyperledger Fabric Blockchain to achieve high-performance levels in the RFID-IIoT environment. In this regard, the registration, authentication, and attributed-based authorization (ABAC) phases are on-chain. Based on a methodological framework proposed, we have demonstrated the feasibility of the Access Control System in a performance constrained IIoT environment such as an engine assembly line. In addition, we have designed the registration phase of our ACS based on HFB private data collection, which promotes a novel reliable data privacy model applied to an ACS. Additionally, we have demonstrated the feasibility of using HFB's private data collection over a private data local management, and finally, we have selected the most suitable combination of network elements and resources for optimal deployment of the HFB network associated with our use case.

1. ABSTRACT

Considering that RFID technology presents a significant growth in IIoT environments, industrial manufacturing is being one of the most benefited by this growth. As growth implies increased security risks, Access Control Systems have emerged as an essential solution for IIoT environments and particularly in RFID systems. Considering Hyperledger Fabric Blockchain as a modular project oriented to environments with high level of performance in terms of speed and scalability, our manuscript proposes a performance analysis based on a methodological framework to demonstrate the viability of a comprehensive access control system which includes Identification, Authentication, Authorization and Accountability/Auditing based on the permissioned blockchain Hyperledger Fabric Blockchain. Our proposal promotes a novel approach to reliable data privacy, based on private data collection solution promoted by Hyperledger Fabric to implement the registration phase of our access control system. In this regard, the feasibility of using private data collection with respect to a private data local management solution is demonstrated. Finally, thanks to the modularity promoted by Hyperledger Fabric Blockchain, we define the optimal network model for our use case. To demonstrate these approaches, several experiments are conducted, based on a proposed methodological performance framework.

2. INTRODUCTION

Currently, the blockchain type to be used in a project depends, according to authorized voices such as Gartner [1], on both the use case and the business model. Focusing on scenarios in which Industrial Internet of Things (IIoT) plays a decisive role (e.g., in high-tech assembly lines where RFID provides several opportunities in terms of data collection), it is necessary to consider time constraints as a requirement. In these environments, blockchain technologies with high transaction throughput, i.e., a measure of valid transactions, are fundamental. In terms of performance, private and permissioned blockchains accumulate advantages over the public and permissionless blockchains, however, a more comprehensive analysis is needed. Specific criteria are analyzed below to highlight the value of Hyperledger Fabric Blockchain (HFB) even within the scope of permissioned blockchains [2].

The first consideration is related to transaction validation approach where HFB differs not only from permissionless blockchain such as Ethereum but as well as from permissioned blockchain such as Tendermint, Chain and Quorum. The approach used by the mentioned blockchains is based on an order-execute architecture, in which the consensus protocol: (1) validates and orders transactions and then, propagates them to all peer nodes and (2) each

peer executes transactions sequentially. The fact all transactions are executed sequentially by all nodes is a performance and scalability constraint. For this reason, HFB introduces a new approach called execute-order-validate which introduces flexibility, scalability, and performance because it separates transaction processes, i.e., flows, in three steps: (1) execute a transaction and verify its correctness, i.e., endorsing a transaction, (2) order transactions through consensus protocol pluggable, and (3) validate transactions through a specific endorsement policy before committing them to the ledger. Particularly, endorsement policy details the peer nodes, or at least, how many of them, need to ensure the correct execution of a given smart contract (chaincode). Therefore, transactions need only to be endorsed by the subset of the peers defined in the transaction's endorsement policy. This feature allows increasing performance and network scalability due to parallel execution.

The second consideration is related to HFB storage system which is oriented towards high-performance, since, on the one hand it contains a world state that describes the state of the ledger at a given time and is name-spaced according to its chaincode, thus smart contracts in different chaincode cannot directly access each other's world state (i.e., Database Management System (DBMS) generates different world state databases). On the other hand, the ledger can be configured to support a variety of DBMSs, i.e., the ledger has a replaceable data store for the world state. The DBMS can be configured to be embedded or separate, depending on its type. For GoLevelDB, the DBMS is fully embedded within the peer, but for CouchDB, the DBMS is an independent process.

The third consideration is related with ordering of transactions, which is performed by a modular component for consensus that is obviously separated from the peers that execute transactions and manage the ledger. Specifically, this pluggable ordering service establishes a consensus protocol to be used, e.g., Raft, for the transactions and for broadcasting blocks to peers.

The last consideration is related to HFB capacity, which introduces private data collection. This solution enables specific storage space for the private data and establishes the hash of private data as "public" to organizations in the same channel. It represents a high degree of integration, improving performance in comparison to public blockchain solutions, in which it is necessary to integrate external storage resources, such as IPFS ([3]), and then emit the transaction based on key-value pair (using the hash of externally stored data as 'value') to a public blockchain such as Ethereum.

These four considerations address the use of HFB in IIoT environments with high performance levels, e.g., high-tech assembly lines, beyond other blockchains. High-tech

assembly lines environments, as part of the industrial manufacturing, are fully integrated with IIoT technologies, adopting RFID systems successfully. For instance, in engine assembly line, the information received of RFID sensors not only helps the plant ensuring the correct processes are being performed in the proper order in real time, but also provides historical data in events of a factory recall or faulty part, helping to identify all engines assembled at a specific time or by a particular machine [4].

Since RFID provides several opportunities for industrial manufacturing transformation to improve efficiency and productivity, it also increases the risk of being affected by different kinds of threats. For instance, authors of reference [5] establish a classification of threat types in RFID systems according to the physical RFID threats, RFID channel threats and the RFID system threats. In this way, access control emerges as a core piece of any organization's security [6]. For this reason, our use case integrates an access control system (ACS) in high-tech engine assembly line. The ACS is based on HFB, and involves all access control phases, i.e., Identification, Authentication, Authorization and Accountability (also often referred to as Auditing) (IAAA). In this sense, a traditional RFID on a centralized environment (which includes subsystems: RFID tag, RFID reader, RFID middleware and business layer [7]) would be transformed into a decentralized environment, where access control policies (ACP), which are regularly executed by the RFID middleware [8], as part of the authorization phase, are now performed on-chain by the chaincode. In fact, chaincode not only performs on-chain the ACP, but also performs on-chain the rest of IAAA processes: registration, identification, authentication, auditing, and accountability, of the assets identified with RFID tags.

From the introduced scenario, the main contributions of our manuscript are: 1) to demonstrate the viability of proposed novel access control system based on HFB for an IIoT environment, by applying a methodological performance framework, 2) to implement the registration phase of our access control system based on the private data collection solution promoted by Hyperledger Fabric, which promotes a novel reliable data privacy model applied to an access control system, 3) to demonstrate the feasibility of using HFB's private data collection over a private data local management and 4) by considering the modularity of HFB, to select the most appropriate combination of network elements and resources for our use case for an optimal deployment of the HFB network.

The remainder of the manuscript is distributed as follows. The second section discusses the most relevant scientific literature related to HFB performance to justify the novelty of our work. The third section establishes the design of the evaluation model, which includes the presentation of our performance framework methodology. The fourth section includes the

use case definition, the benchmarks criteria for the use case evaluation and finally, chaincode design and implementation, including the data privacy solution. The fifth section designs the experiments to be conducted and discusses the results obtained, ensuring that objectives are achieved, by following the methodology we propose with the performance framework. Additionally, as part of section 5, we include an overview of experiment results and a comparison of different access control proposals, with the aim of simplifying and highlighting the novelty of our work. Finally, conclusions and future research lines are addressed.

3. RELATED WORK

Several works applying performance analysis in permissioned blockchain focus on HFB with the caveat that new versions imply improved functionality and new features that impact performance. Considering that the current version of HFB is 2.x, we will focus related work analysis on HFB versions range 0.6 to 1.4x.

The reference [9] evaluates the performance of two HFB versions (0.6 and 1.0), deploying up to twenty nodes for each HFB version in a local environment, demonstrating in terms of execution time, latency, throughput, and scalability that HFB version 1.0 outperforms HFB version 0.6 when the number of nodes is increased. The reference [10] evaluates the performance only for HFB version 1.0, considering the impact of several metrics such as block sizes, endorsement policies, number of channels, resource allocation (number of vCPUs) and DBMS type. In addition, the experiments were conducted in a local but powerful environment, which dedicates important resources to each organization. The reference [11] evaluates the performance based on Stochastic Reward Nets (SRN) to determine the throughput, utilization, and queue length for each peer of HFB version 1.0. To conduct a comprehensive analysis, Hyperledger Caliper is used to deploy up to twenty clients on a local environment, including several scenarios such as two consensus algorithm and block size variation. The reference [12] characterizes the performance of HFB version 1.0, focusing on two measures: throughput and latency. The authors conduct experiments to scale chaincode, channels and number of peers. It should be noted that, although the experimental setup was entirely local, the authors used four separate hardware devices and the Hyperledger Caliper tool to generate the client workload. The reference [13] focuses on Hyperledger Fabric v1.1 and conducts experiments hosting several nodes on IBM Cloud. The HFB setup includes a single channel, Kafka consensus algorithm and different endorsements options. The authors presented the impact of block size, number of vCPUs, number of peers and SSD vs RAM disk on latency and throughput. Additionally, the reference [14] offers

important contributions since it conducts the performance analysis based on HFB version 1.4, determining a wide range of variables such as latency, throughput and scalability and deploying the network in a single machine in Amazon Web Service (AWS). The tests performed are based only on the number of transactions. Like other studies analyzed, the authors use Hyperledger Caliper tool to generate the client workload.

Considering there is not a framework for performance analysis and, based on the set of metrics defined by the Hyperledger Performance and Scale Working Group in [15] and the HFB specifications in [16], the novelty of our work is justified based on the definition of a methodological framework for performance. What is more, it is used for demonstrating the viability of the access control system we propose for an IIoT environment. Additionally, both the use of the HFB's private data collection solution as part of the registration phase of our access control system, and the use of private data collection instead of a private data local management solution based on hashing operations and external storage, lend novelty to our work. Finally, the establishment of the optimal network architecture associated with our use case also confers novelty. In this regard, unlike the HFB versions analyzed as part of the related work, our analysis focuses on HFB version 2.0 because this version includes substantial improvements around: 1) Raft consensus algorithm natively as ordering service; 2) private data collections are significantly improved from previous versions; 3) State database (StateDB) cache improves performance for CouchDB and 4) Kafka-based ordering service is established as obsolete [17].

4. DESIGN OF PERFORMANCE EVALUATION METHODOLOGY

The starting point of the evaluation methodology is the definition of the performance framework (Figure 1), which is an abstraction of the Hyperledger Fabric layers. Our framework complies with the framework presented by the authors of reference [18]. This performance framework is applied through a methodology which allows to evaluate the impact of the different possible configurations for each of the different layers. The Figure 1 shows the HFB layers related to performance, where:

1. Infrastructure refers to the available computational resources dedicated to build the HFB, i.e., the underlying infrastructure which hosting the HFB modules, including details such as the HFB deployment over Kubernetes or bare metal.
2. Architecture refers to how the nodes over the infrastructure (peer, orderer, client) are connected and interact, considering the scalability (e.g., behavior when peer number is increased) and optimization (e.g., the best StateDB).

3. Protocol refers to configuration of the modules which integrate the nodes, which directly affects the performance (e.g., increasing the number of channels, set the block size, the endorsement policies, the strategies of CouchDB and so on).
4. Chaincode refers to strategies and best practices for smart contract design in order to improve the performance result.
5. Software Development Kit (SDK) refers to one mechanism responsible for orchestrating all the transaction lifecycle in client side, used for the interaction via Application Programming Interface (API).

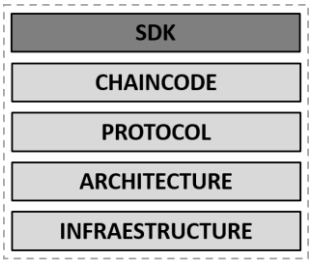

Figure 1: Performance framework.

In this section, the chaincode and SDK layers are excluded from our analysis. The reason for this decision is that on the one hand, chaincode depends on the use case and it will be analyzed after presenting it in next section. On the other hand, in the performance evaluation we use the benchmark framework Hyperledger Caliper [19] which bypasses the SDK layer. Hence, the remainder of this section analyzes the infrastructure, architecture, and protocol layer.

4.1. INFRASTRUCTURE LAYER

The starting point to consider is that underlying infrastructure to deploy the Fabric network is key in the performance. In this regard, independently of the underlying infrastructure (e.g., Kubernetes or bare metal) we ensure to have, in general terms, enough hardware resources to our Fabric network and, specifically, enough vCPUs in the infrastructure to accommodate the network. At this point, it is key to establish a consensus regarding the scope of the implementation, this is because several references such as [20] indicates that a Kubernetes based infrastructure introduce overload into the HFB network. However, as the HFB network scales, the management of a bare metal-based environment becomes more complex despite the better ratio performance/hardware. Considering that the purpose of our manuscript is not to bring the HFB network into production, four organizations, each

composed of three hosts, the HFB network will be deployed on Amazon Elastic Compute Cloud (EC2) instances running on Amazon Web Services (AWS) [21].

4.2. ARCHITECTURE LAYER

The architecture layer underlies the infrastructure layer and is essentially composed by deployed nodes, as well as the mechanisms adopted to optimize their performance and ensure scalability. In order to analyze the performance of the architecture layer the reference [15] establishes System Under Test (SUT) concept. As Figure 2 shows, our SUT is composed of the Application Under Test (AppUT) and Architecture Under Test (ArchUT). The AppUT will be the subject of future sections, although it should be noted that Caliper tool ([400]) is used for this purpose. In this regard, this section focuses on defining the ArchUT to be deployed for proper performance evaluation, i.e., the architecture layer established in Figure 2.

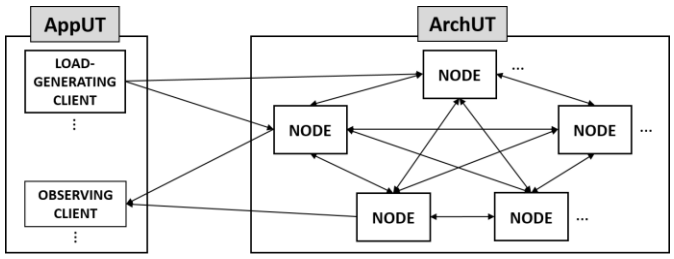

Figure 2: System Under Test (SUT).

ArchUT represents the several configurations that can be adopted by the HFB network, so that performance can be evaluated comprehensively. The nodes in an HFB network can play three roles: client, peer and orderer. Regarding the client role, it submits the transaction proposal, receives the transaction response endorsed and broadcasts the transaction for ordering. With regard to peer, it executes transaction proposal, validates transactions, maintains the blockchain ledger, as well as the last ledger state; however, the peer only executes the transaction proposals, whether it is an endorsing peer, as specified by the policy of the chaincode to which the transaction pertains. Finally, with respect to orderer, it establishes the total order of all transactions in HFB, enabling the block creation, which is crucial to ensure determinism in the ledger. At this point, it is necessary to clarify the operations required to evaluate performance with respect to the architecture layer: A) scaling the endorsing peers, B) scaling non-endorsing peers, C) scaling the ordering nodes, D) distinguish StateDB (DBMS) between CouchDB and GoLevelDB, E) scaling number of channels and F) modifying ordering service network (OSN) between Raft and Solo. Table 1 summarizes the operation that could be performed on each architecture component to

evaluate the HFB performance. The ordering service modification, based on the consensus algorithm modification, promotes the greatest variation in the network architecture, since it modifies the way in which the client broadcasts the endorsed transaction response, forcing different ArchUT scenarios. For that reason, the next sections analyze the HFB network deployment for two ordering services: Raft and Solo.

Table 1: Architecture layer configurations.

Architecture component	Operation
Endorsing Peer	Scaling
Non-endorsing Peer	Scaling
StateDB	Comparison CouchDB and GoLevelDB
Orderer	Scaling
Channels	Scaling
Ordering service Nodes	Modifying Network: Raft or Solo

4.2.1. ArchUT based on Raft Ordering Service

In the previous section, it was mentioned how ordering service plays a key role in HFB, because it receives endorsed transaction responses via clients broadcasting and arranges them in order into a block, which is validated by the peer in a deterministic fashion, providing consistency in the ledger. In this section, we will assess the first Raft-based ArchUT.

The Figure 3 shows several organizations created as part of ArchUT. We have deployed four organization and therefore twelve hosts. Each host contains different types of nodes. By default, all of them include an orderer and a peer, which contains the ledger and the chaincode; however, in each organization one host will run the organization's certification authority (CA) and finally the first host will contain an additional orderer (e.g., ORDERER 13), which represents the leader assigned as part of the Raft ordering service. It should be noted that Raft is a Crash Fault Tolerant (CFT) ordering service based on Raft protocol [22],

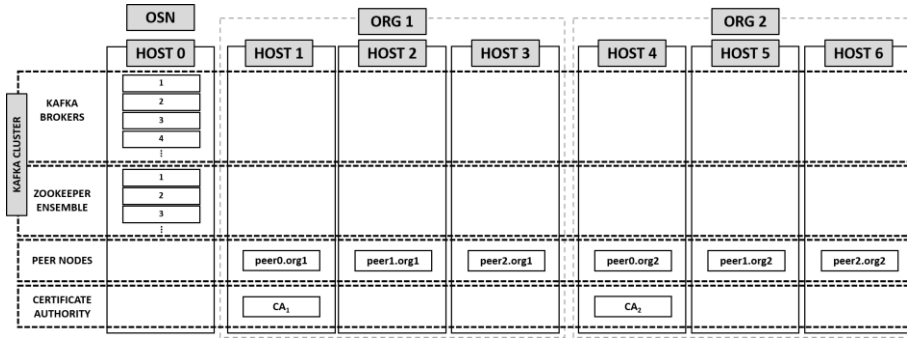

Figure 3: Raft-based Ordering Service.

which follows a "leader" (elected per channel) and "follower" model. For instance, in the configuration of the Figure 3, the "ORDERER 1" node is the leader.

4.2.2. ArchUT based on Solo Ordering Service

The second ordering service analyzing is Solo, which features only a single ordering node, as Figure 4 shows, and therefore never will be fault tolerant. For that reason, the Hyperledger Fabric documentation ([23]) indicates to use Solo ordering service only as part of proof of concepts (PoC), i.e., not for production. In addition, Figure 4 shows only one host dedicated to Solo ordering service, since according to Hyperledger Fabric documentation ([23]), Solo is applicable for networks with only a single ordering node.

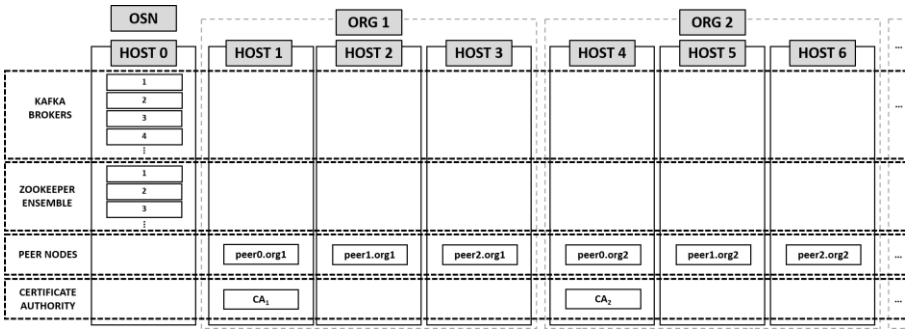

Figure 4: Solo-based Ordering Service.

4.3. PROTOCOL LAYER

So far, the impact on performance of the infrastructure and architecture layers has been presented (Figure 1). In this section, the impact on performance of the protocol layer is analyzed. To this end, Table 2 relates architecture components to the parameter or protocol applicable to our use case, where: A) endorsement policies refer to the set of organizations required to endorse a transaction; B) private data collection allows organizations to manage private data without having to create a separate channel; C) block size refers to the block size in MB, D) consensus type enables the ordering service type; E) cache size enables local cache reads in order to reduce reading delays when external CouchDB is used as StateDB.

Table 2: Protocol layer configurations.

Architecture component	Protocol parameter/operation
Peer	Endorsement policies
Channel	Private Data Collection
Orderer	Block Size
Orderer	Consensus type
StateDB	Cache Size

The remaining layer of the framework represented in Figure 1 is Chaincode, however, the details of its intervention in performance will be analyzed when defining the use case. It should be mentioned that several parameters are analyzed such as the need to follow a correct design and best implementation practices as requirements to improve performance.

5. USE CASE FOR PERFORMANCE EVALUATION: NOVEL IIOT ACCESS CONTROL SYSTEM IN MANUFACTURING ASSEMBLY LINE

On the one hand, this section describes both the use case in which the access control system is applied and the definition of the benchmark requirements for it, and, on the other hand, it analyzes the design of the access control system as well as the design and implementation criteria of the chaincode.

The Figure 5 shows a manufacturing assembly line composed of five phases or stages of a motor assembly process, where an asset is assembled in each of them, thus achieving an assembly cycle. The IIoT environment involves the phases of sensing (RFID tag), control (RFID reader) and actuation (robotic arm) [7]. RFID Readers (RR_1, RR_2, RR_3, RR_4 and RR_5) provide the entry point for asset registration, as well as RFID Readers (RAA_1, RAA_2, RAA_3, RAA_4 and RAA_5) provide the entry point for asset authentication and authorization. Robotic Arms (A_1, A_2, A_3, A_4 and A_5) recover the assets from the boxes (B_1, B_2, B_3, B_4 and B_5) and bring them closer to the industrial transport belt to assemble them. Assets are identified through a RFID tag, which represents the digital asset identification (e.g., "0x1A", "0x2B", "0x5E" in Figure 5). In the initial state all assets identified are in the boxes and are registered in blockchain, once they are detected by the corresponding RFID Readers (RR_1, RR_2, RR_3, RR_4 and RR_5). Then, the robotic arm A_1 approaches to the industrial transport belt, the asset with category 1 identified as Figure 5 shown as "0x1A" and once the RFID reader RAA_1 detects it, the access control system authenticates the asset and authorizes A1 to place it on the industrial transport belt. Then, A_2 brings the asset with category 2 identified as "0x2B" to the industrial transport belt and once RAA_2 detects it, the system authenticates it and authorizes the assembly operation, which executes A_2, coupling it with the asset "0x1A". The

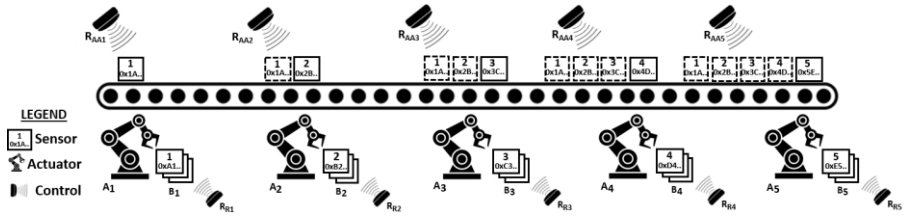

Figure 5: Use case definition for IIoT environment.

cycle ends, i.e., an engine is completed when the category 5 asset identified as "0x5E" is assembled with the set of assets from phase 4.

5.1. PERFORMANCE CRITERIA FOR OUR USE CASE'S FEASIBILITY

The reference [24] enables a framework to minimize the non-productive activities on an engine assembly line. Therefore, Table 3 shows the benefits of applying this framework in terms of improved engine assembly time. To align this analysis with our use case, we should be aware of some considerations:

1. The values obtained are an approximation because the number of RFID tags (284) is a particular data on every Gen 5 six and eight-cylinder engines, in order to track the assembly process [4].
2. The assembly time of each asset is an average value, because the time spent to each assembly phase is heterogeneous. The most restrictive time benchmark will be used to validate the latency requirements, which involve authentication, authorization, and assembly of an identified asset operations. For the access control system that we propose to be feasible to implement, it is necessary that time of set of operations that are executed along an assembly phase (i.e., authentication of an identified asset and authorization of assembly) are significantly less than 15.50 seconds, which is the average assembly time for each of the phases according to Table 3.

Table 3: Optimized parameters of an engine assembly line for establishing benchmark time for our use case [24].

Parameter	Value
Number of motors assembled in shifts	87
Assembly time of an engine	73.40 min
Number of RFID assets in an engine	284
Assembly time for each phase	15.50 ms

5.2. ACCESS CONTROL SYSTEM DESIGN

Figure 6 presents the access control system which involves all access control phases, i.e., Identification, Authentication, Authorization and Accountability/Auditing (IAAA), which constitutes an HFB network abstraction because key elements such as endorsing and non-endorsing peers, ordering nodes and ledger are omitted, i.e., the HFB network with the four defined components are omitted. This way, we focus on representing the interaction between the participating entities (RFID reader, robotic arm, user application and smart gateway) with the blockchain network through the chaincode. This interaction is conducted

through the four types of chaincode observed in the figure: Policy Chaincode (PCH), Registration Chaincode (RCH), Authentication Chaincode (ACH) and ABAC Execution Chaincode (AECH), which are analyzed in detail in the following section.

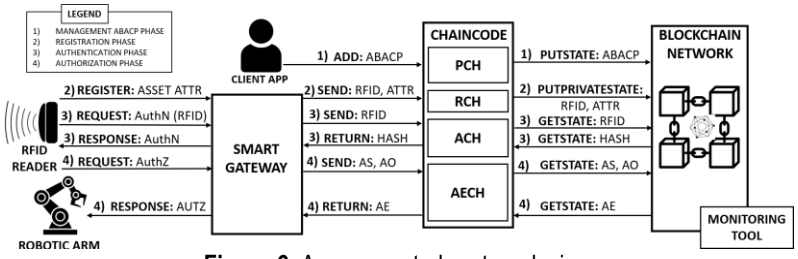

Figure 6: Access control system design.

Table 4 relates all the chaincodes, methods and APIs used, being the API, the mechanism used by the method to interact with the blockchain via SDK. For instance, a method that uses the *PutState* API, requires a key and a value (a byte array) which are inserted in the StateDB, while a method that uses the *GetState* API, requires only a key to return the associated value. Each of the chaincodes are associated with a stage or phase of our access control model. Before describing each phase, from the reference [25], we can determine four categories of attributes applied in Attributed-based Access Control Policy (ABACP). First, attributes of subjects (AS), concerning the identities and features of the entities that initiate the access request (e.g., actuator ID, person's ID, etc.). Second, attributes of objects (AO) concerning the attribute of the accessed resource (e.g., RFID tag ID, IP address, URL, etc.). Third, attributes for permissions (AP): concerning the operation of the subject on the object (e.g., reading, writing, executing, etc.). Fourth, attributes of environment (AE): concerning the environment information when the access request is generated (e.g., time, location, etc.). In this regard, the phase 1) allows a trusted client to enable the ABACP in the blockchain via *PutState* API, which updates the StateDB channel of peers joined to a channel. The phase 2) involves the asset registration, which starts when the RFID Reader detects the asset in the initial state and sends the asset attributes to the smart gateway (SGW). SGW sends asset Identifier (RFID) as well as asset attributes to RCH, which enables the key-value pair, composed of the asset identifier (RFID) and the hash of the attributes, in the blockchain via *PutPrivateState*. It updates both the Private StateDB with the private attributes, available only for specific organizations into a channel and the StateDB with the hash of the private attributes available for all members of the channel (section 5.4 details the data privacy solution). The phase 3) involves the asset authentication, starting when the RFID Reader identifies an asset, and it requests the AuthN. Next, SGW sends the asset identification (RFID) and the ACH uses API *GetState* to recover the hash associated and the API

GetPrivateState to recover private attributes. The SGW verifies the hash and responds to RFID Reader. The phase 4) involves the asset assembly authorization (AuthZ), which starts when RFID Reader emits the AuthZ request. SGW sends AS and AO to AECH. AECH uses *GetState* API to send AS and AO. Next, AECH evaluates AE to determine AuthZ sending the response to SGW, which in turn permits or denies the assembly operation. Finally, Figure 6 includes the monitoring toolbox, which is part of the accountability/auditing phase of our access control system. Thus, the HFB documentation ([16]) establishes a set of metrics used to monitor, control, audit, and account for: containers, nodes, ledger, chaincode, consensus and DBMS. These metrics are exported for consumption by the open-source monitoring solution Prometheus ([26]).

Table 4: Relation between chaincode, methods and API of our use case.

Chaincode	Method	API	Description
PCH	addPolicy	PutState	Insert an ABAC policy on-chain in the HFB.
	verifyPolicy	GetState	Verify an ABAC policy inserted in HFB via Policy ID.
	queryPolicy	GetState	Query a policy inserted in HFB via AS and AO.
	updatePolicy	PutState	Update an existing policy.
	deletePolicy	PutState	Delete a policy in the HFB.
RCH	addAsset	PutPrivateState	Enables a key and sends attributes to Private StateDB.
ACH	getAsset	GetState	Recovers value via key.
	getPrivateAsset	GetPrivateState	Recovers private attributes via key.
AECH	verifyAccess	GetState	Query an ABAC Policy on-chain, via queryPolicy method and then verify the AE correctness.

5.3. CHAINCODE DESIGN

Throughout section 4, the layers included in the performance framework have been detailed except for the chaincode layer, since the use case needed to be introduced first. In this section, design criteria of the chaincode associated to our use case is analyzed. Our chaincode system is divided into four smart contracts mentioned in the previous section (section 5.2) and summarized in Table 4: 1) PCH, 2) RCH 3) ACH and 4) AECH, each of which is composed of different methods. The PCH contains five methods: *addPolicy*, *verifyPolicy*, *queryPolicy*, *updatePolicy* and *deletePolicy*. The *addPolicy* method adds the ABACP to the StateDB. The request for adding ABACP is verified through the method *verifyPolicy*, which analyses whether the policy contains the four types of attributes (AS, AO, AP, and AE) and whether these in turn meet strict requirements. The *updatePolicy* method allows the

administrator modifying the ABACP overwriting the values defined by the addPolicy method. The *deletePolicy* method is performed when the administrator invokes this method, as well as, when the *verifyAccess* method is executed and the endTime attribute has expired. The *queryPolicy* method queries the ABACP on-chain, via AS or AO attributes. The RCH contains the method addAsset, used as part of the identification phase, which finish when the pair (rfid, hash) is inserted into the blockchain. The *addAsset* method represents an innovation point because it involves the data privacy solution included in HFB (which will be inspected in depth in the next section). The ACH contains the method *getAsset*, which uses a key field to recover the value associated to determine the asset authentication via attributes verification and the method *getPrivateAsset*, which recovers the private attributes from the Private StateDB. The AECH contains the method *verifyAccess* which is a core mechanism to perform the ABACP, i.e., authorization process. It uses the attributes AS, AO, as arguments and invokes the *queryABAC* method to recover the corresponding ABACP. Finally, it compares each AE to determine if the attributes match and allow the assets to be assembled.

5.4. DESIGN CRITERIA BASED ON DATA PRIVACY SOLUTION

In our context, some attributes used to perform both authentication and authorization (ABAC policy execution) are recovered in the registration phase (section 5.2) and contain a sensitive set of data that must not be shared with the participants of external organizations and channels. This private data is stored on the Side Database (Side DB) instead of on the blockchain, but a hash of the private data is stored on the blockchain, which serves as an evidence of the private data and is used to validate the private data. Our proposal trusts the robustness of transactions associated with private data collections, which we use only as part of the registration phase, through addAsset method by associating the asset identifier with its attributes. The purpose of this solution is that even though there are several organizations joined to our network, only the peers of the organization that will evaluate the access control of a specific asset will have access to the attributes of that asset, while the peers belonging to the rest of the organizations will only be able to access to the hashes of these data, keeping information as private. Previously, for private data collections, when a group of organizations in a channel needed to maintain the privacy of data from other organizations in that channel, they could only create a new channel that included only the organizations that needed to access the data. However, as Hyperledger Fabric's specifications indicate, this creation of separate channels in each of these cases creates additional administrative overheads and exclude use cases where all channel participants can see a transaction while keeping a portion of the data private.

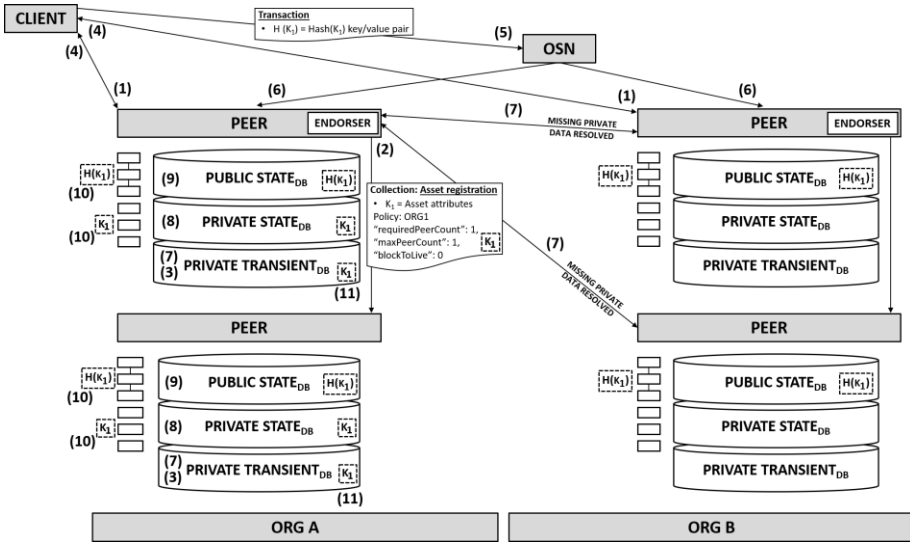

Figure 7: Chaincode of registration phase.

Figure 7 details how the side DB solution is applied in the asset registration phase of our use case. (1) Client sends proposal to endorsing peers, (2) endorsing peers simulates transaction and distributes private details collection data based on collection policy; (3) peers store received data in Transient DB (only peer belonging to ORG A); (4) endorsing peers sends proposal response back to client with public data; (5) client submits transactions with public data to ordering services nodes (OSN), which includes private data hash, i.e., key-value pair; (6) OSN distributes the blocks with transactions to all peers; (7) peers validate private data against public hashes, while the missing private data is resolved with pull request to other peers via gossip protocol; (8) commit private data to private StateDB; (9) commit hashes to public StateDB; (10) commit public block and private set storage; (11) delete transient data.

5.5. CHAINCODE IMPLEMENTATION CRITERIA ADOPTED

Considering that the chaincode implementation significantly affects the HFB performance, this section focuses on describing some followed criteria to achieve the best performance scenario. The starting point is the chaincode implementation in the data privacy solution. Our use case only needs one private data collection (see asset registration in Figure 6) which instead of a normal *PutState* API we use the *PutPrivateData* API. To configure the collection properties, the fields: name, policy, *requiredPeerCount*, *maxPeerCount* and *blockToLive* are defined in a separate file (JSON file) from the chaincode. As shown in Figure 7, the field policy (dissemination policy) determines which organizations have access to the private data,

the field *maxPeerCount* considers the necessary endorsing peers to distribute the private data and finally the *blockToLive* field represents how long the data would be in the private database before being purged. In our case we set *blockToLive* to "0" to keep private data indefinitely, i.e., to never purge private data. The second implementation criteria adopted, avoids Multi-Version Concurrency Control (MVCC) conflicts. MVCC is a mechanism commonly used by DBMS to provide concurrent access to the database, which is susceptible to fail when the high-rate transactions involve two transactions attempting to write the same key in the ledger at the same time. Our data model performs every new transaction from a completely different key since we associate the key to the unique asset identifier. This interaction is only possible by performing the *addAsset* method, which belongs to the registration phase. The third consideration is to minimize the number of *GetState* and *PutState* in the same transaction, as well as, to separate possible read operations from write operations. In this regard, we strictly separate the methods in different smart contracts, associated to each use case phase. Finally, we consider that CouchDB supports complex queries, however, it needs to be validated based on the capabilities offered by GoLevelDB.

From the nine methods designed (see Table 4), there are only two directly involved in authorization phase of ACS: *getAsset* (ACH) and *verifyAccess* (AECH). To ensure the feasibility of the ACS, the sum of the average latencies must be significantly less than the value set as the assembly benchmark time (15.50 s) in section 5.1.

6. EXPERIMENTS AND COMPARISON

This section introduces experiments conducted, as well as the results obtained to achieve the defined objectives: 1) to demonstrate the viability of proposed access control system based on HFB for an IIoT environment, from performance evaluation metrics; 2) to implement the registration phase of our access control system based on the private data collection solution promoted by HFB, which promotes a novel reliable data privacy model applied to an ACS; 3) to demonstrate the feasibility of using HFB's private data collection (PDC) over a private data local management (PDLM) and 4) to select the most appropriate combination of network elements and resources according to our use case for an optimal deployment of the HFB network.

To conduct each of the experiments listed in Table 5, the SUT is defined, composed of the AppUT and the ArchUT. Regarding ArchUT, we have a base ArchUT or testbed, from which we adjust the configuration parameters according to the experiment. Table 6 contains the default deployment features of the testbed. With regard to AppUT, we have defined

two major configurations showed in Figure 8 and Figure 9. The flowchart in Figure 8 involves an abstraction of our network for executing a submit function, which is a method that invokes transactions, i.e., it allows to call the methods that use the *PutState* API. The flowchart in Figure 9 also involves an abstraction of our network for executing an evaluation function, which is a method that queries the blockchain, i.e., *GetState* API.

Table 5: Relation between objective, framework layer and experiment conducted.

Obj	FWK Layer	Experiment	Brief description
1	Chaincode	Feasibility of access control system in terms of latency	Demonstration of the viability of the ACS based on latencies of getAsset and verifyAccess methods.
2	Chaincode	PutState vs PutPrivateState	Demonstration of the viability of applying the addAsset method via PutPrivateState API based on performance criteria.
3	Protocol	Feasibility of data privacy solution	Demonstration of the viability of applying a solution based on PDC, over one based on PDLM, based on performance criteria.
4	Architecture	Optimizing HFB network based on StateDB	Selection of the appropriate StateDB for our use case based on performance criteria.
	Protocol	Optimizing HFB network based on block size	Selection of the appropriate Block Size for our use case based on performance criteria.
	Architecture	Optimizing HFB network based on ordering consensus	Selection of the appropriate ordering consensus for our use case based on performance criteria.
	Protocol	Optimizing HFB network based on endorsement policy	Selection of the appropriate endorsement policy for our use case based on performance criteria.

Table 6: ArchUT or testbed by default.

Configuration parameter	Value
Endorsing Peer	1 per ORG
Number of ORG	4 ORG
Non-endorsing Peer	3 per ORG
StateDB	GoLevelDB
Orderer	1 per ORG
Channel number	1
OSN	Raft
Endorsement Police	OR [a, b, c, d], where “a”, “b”, “c”, and “d” are ORG names
Batch size	100
Block size	2 MB

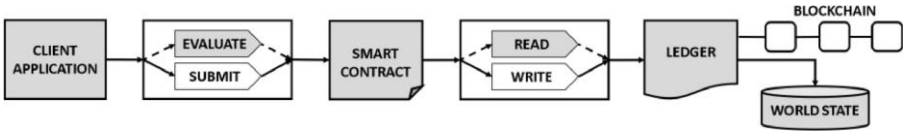

Figure 8: AppUT of the experiment conducted to evaluate performance querying StateDB.

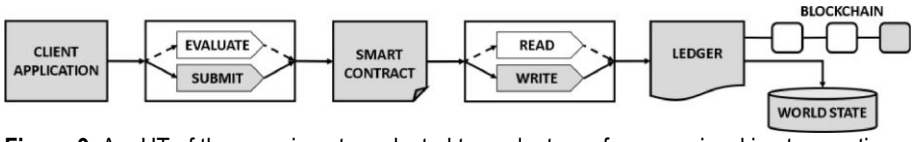

Figure 9: AppUT of the experiment conducted to evaluate performance invoking transactions.

6.1. FEASIBILITY OF ACCESS CONTROL SYSTEM IN TERMS OF LATENCY

In section 5.1, reference measurements associated with the use case in production were established. Adopting the most restrictive value, to determine the viability of the proposal implicates that each phase of engine assembly has average latency levels under 15.50s. We established in section 5 that once the robotic arm has approached the industrial transport belt, the new asset to be assembled has first to be authenticated and then authorized. Therefore, in order to determine the feasibility of the proposed access control system, the latencies of each of the method directly involved in both authentication and authorization need to be performed. From section 5.3, we can determine the methods under test: *getAsset* (ACH chaincode) and *verifyAccess* (AECH chaincode). Again, the default testbed (Table 6) and the scenario defined in Figure 8 is used for the testing purpose. Both methods getAsset and *verifyAccess* method uses *GetState* API to submit transactions via client application based on Hyperledger Caliper to evaluate the Smart Contract which read the information from the Private StateDB and Ledger, respectively. An input transaction rate of 100 tps has been applied when measuring average latencies.

Figure 10 shows the average latencies obtained after the application of each method. It is observed that both latencies are significantly lower than the established benchmark time

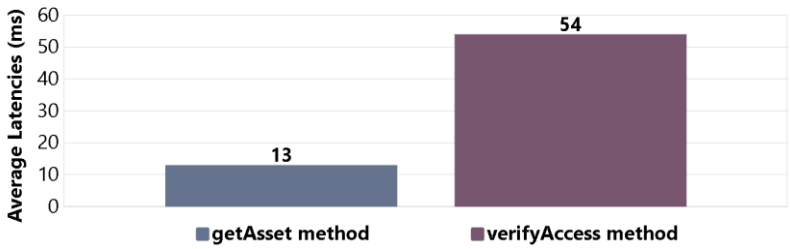

Figure 10: Evaluation of latencies for getAsset and verifyAccess methods.

(15.50 s), so that the sum of both average latencies (67 ms) would be insignificant compared to such requirement (67 ms << 15500 ms). In addition, it is noted that the time difference between both methods (41 ms) is remarkable, which is explained because the *verifyAccess* method performs the operation of comparison of environmental attributes (EA), despite both use *GetState* API.

6.2. PUTSTATE VS PUTPRIVATESTATE

Private data collection has emerged as a usable solution based on functional and privacy requirements. For instance, it is used when only a subset of the organizations belonging to a channel should have access to some (or all) of the data in a transaction. In the context of our use case, private data collection is only used as part of an organization because it contains sensitive data, i.e., attributes that are used in the authentication and authorization phases. Furthermore, since private data are disseminated between peers and not through blocks, private data collections maintain confidentiality in OSN, since only the hash of attributes is disseminated in "public" transactions. Therefore, the objective of this section is to evaluate the performance of private data collection transactions with respect to "public" transactions within a channel. In section 5.2 it was mentioned that private data collection is used in the registration phase from RCH, which implements the *addAsset* method. In addition, *addAsset* method uses the API *PutPrivateState*. Given that the *PutPrivateState* API receives as arguments a key and an additional set of attributes for which the hash operation is computed, in order to demonstrate the viability of the private data collection, the *addAsset* method and consequently the use of the *PutPrivateState* API must be compared with a method that uses a key-value pair, i.e., the *PutState* API. The use of the *addPolicy* method is determined from Table 4.

To conduct the experiment the default testbed (Table 6) is used. Considering, the referred methods is used the test environment presented in Figure 8 to compare the experimental results to submit transactions via Hyperledger Caliper Tool both over the PCH chaincode and ACH chaincode, i.e., the method *addPolicy* which uses *PutState* API and the method

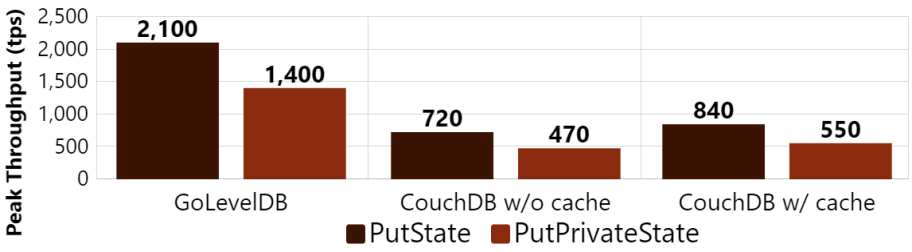

Figure 11: Effects of throughput evaluation of PutState vs. PutPrivateState.

addAsset which uses *PutPrivateState* API. The methods are tested under an input transaction rate of 2100 tps. Figure 11 shows how the throughput peaks achieved with the *PutState* API compared with those achieved using the *PutPrivateState* API. Considering that in section 5.4, the complexity of the life cycle of a private data collection was defined, we can conclude that the reduction in throughput of around 45% is comprehensible.

6.3. FEASIBILITY OF DATA PRIVACY SOLUTION

As mentioned in section 5.4, the data privacy solution interviews in the registration phase of our proposal. Considering the registration phase is performed as part of the initial state, its contribution to the delay of the process cannot be compared with the reference values defined in section 5.1. For that reason, to demonstrate the feasibility of the application of the data privacy solution we have designed the following experiment to evaluate the performance. Figure 7 shows in a methodological way the steps that followed by the data until they are recorded on the private StateDB. These processes introduce some delay in the registration process, which has been demonstrated by evaluating throughput in section 5.2, evaluating submitted transactions via the *PutState* API and the *PutPrivateState* API. However, to demonstrate the feasibility of the data privacy solution an experiment, which includes HFB-external hashing operations and processing of storage, must be considered. In this regard, as shown in Figure 12, a private data local management solution has been created exclusively for verifying the feasibility of our solution based on private data collection, i.e., created only for demonstration purposes. From an implementation point of view, when the asset attributes are obtained from the RFID reader, the Smart Gateway (see Figure 6) computes the $hash_{sha256}$ of the attributes, invokes a transaction via an *addAssetTest* method, which uses the *PutState* API instead of the *PutPrivateState* API. Finally, the SGW records the asset's RFID identifier in a local database. To compare scenarios, for private data collection, latency is considered as the time taken from a client sending the transaction proposal until the transaction is committed; while for private data local management, latency is considered as the interval in which all the operations illustrated in Figure 12 are performed, including the transactions through the *PutState* API. It should be noted that to conduct the experiment the default testbed (Table 6) is used.

The results first show the feasibility of applying the private data collection as solution over the local storage solution (Private data local management). On the one hand, Figure 13 shows (in the left side), the average latency comparison between a private data collection (which applies the *PutPrivateState* API) and a private data local management (which applies the

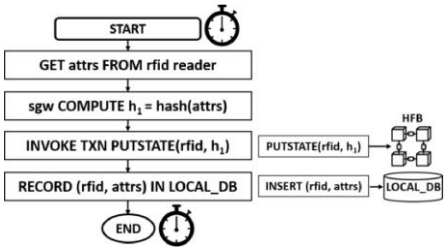

Figure 12: Experiment to demonstrate the feasibility of private data collection.

PutState API). It is confirmed that the latency value obtained for the private data collection is higher, due to the complexity of the procedures we have illustrated in Figure 7. On the other hand (right side of Figure 13), when comparing the latency of the private data collection with the application of the entire process illustrated in Figure 12, it can be seen that the introduction of HFB-external processing around local storage and hashing causes an increase in the latency values.

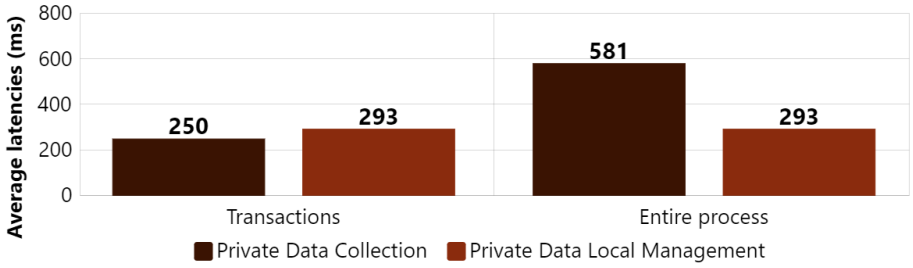

Figure 13: Effect of transaction latency comparison of private data collection and private data local management.

6.4. OPTIMIZING THE NETWORK FOR OUR USE CASE

This section aims to select the most appropriate combination of network elements and resources for our use case for an optimal deployment of the HFB network. For this reason, this section focuses on the Architecture and Protocol layer of the performance framework (Table 6). The first section analyses the variation of the StateDB, the second analyses the variation of the block, the third analyses the variation of the ordering consensus and the last the variation of the endorsement policy.

6.4.1. Optimizing HFB network based on StateDB

From ArchUT's point of view, this experiment follows the default testbed (Table 6). In addition, Figure 8 illustrates the AppUT flowchart of the experiment conducted to evaluate StateDB performance. Hyperledger Caliper emulates a client application that submits

transactions via the *addPolicy* method (PCH). As Table 4 illustrates, the *addPolicy* method uses *PutState* API, i.e., a writing operation. To conduct the experiment, we use an input transaction rate of 2100 tps. The life cycle of the transaction includes the participation of both World State DB and blockchain. However, the World State DB involves different behaviors in terms of throughput when the World State DB is set to: GoLevelDB, CouchDB with cache and CouchDB without cache.

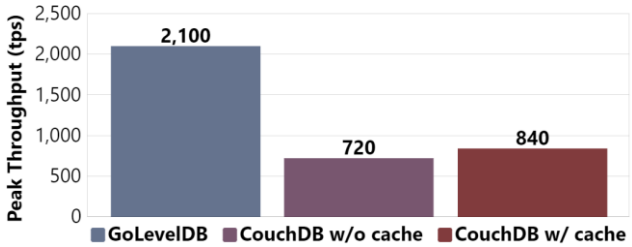

Figure 14: Effect of throughput evaluation via Hyperledger Caliper.

Figure 14 shows the experimental results to submit transactions via Hyperledger Caliper Tool over the method *addPolicy* of PCH chaincode. As Table 4 illustrates, *addPolicy* is based on *PutState* API. The highest throughput peak is achieved by the GoLevelDB DBMS, followed by the CouchDB DBMS with cache and finally CouchDB achieved the worst throughput peak. The cache size is set to 64 MB to test the Couch DB with cache.

To determine the causes of the behaviors, we need to analyze the resource consumption of a peer in terms of CPU utilization and disk write.

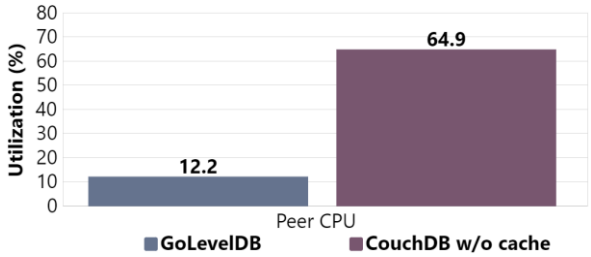

Figure 15: Effect of evaluation of a peer's CPU utilization.

On the one hand, Figure 15 analyzes how the utilization of CPU is higher in CouchDB than GoLevelDB, promoted by the higher CPU contention for Validation System Chaincode (VCSS), which is a piece of code used to determine whether a transaction is valid and to verify if the transaction satisfies the endorsement policy. It should be noted that this result is obtained both when the peer with GoLevelDB and the peer with CouchDB achieved 720 tps. On the other hand, Figure 16 shows that CouchDB is doing more work for writes. The

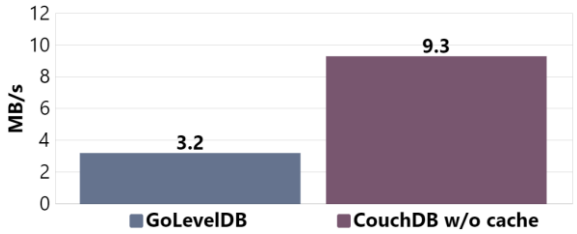

Figure 16: Effect of evaluation of the writing on a peer's disk.

fact that CouchDB requires more processing both in terms of CPU and number of writes to disk, explains GoLevelDB's higher throughput.

6.4.2. Optimizing HFB network based on block size

In all the experiments conducted, we have been used the same network architecture based on the default testbed (Table 6), with the same configuration parameters, e.g., the block size was set to 2MB. In this section, we analyze the first criteria that allows us to optimize the network architecture to the use case, in terms of latency and throughput. Block size is a variable used to determine number of transactions into a block. The *addPolicy* (PCH) method has been used to perform the tests via *PutState* API, so the Figure 8 shows the AppUT used. In this test scenario, the input transaction rate has been defined as 2100 tps for each of the defined block size 0.5 – 4 MB.

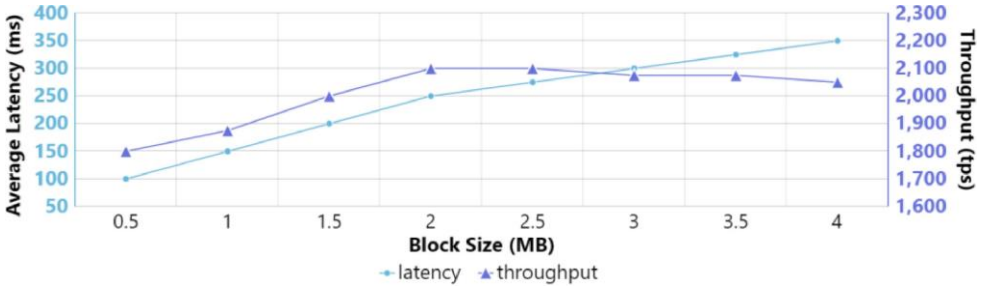

Figure 17: Effects of block size on transaction latency.

Figure 17 shows the size block in a range of 0.5 MB – 4 MB, as well as the behavior of average latency and throughput when transactions are performed via *PutState* API for the values in range. In this regard, the lowest block size implies less processing for both the ordering nodes and the peers, and therefore a reduction in latency. However, the higher block size implies more successful transactions and therefore increased throughput. Both behaviors are visualized in Figure 17. However, it can be noted that from a block size of 2 MB the throughput tends to decrease, which can be explained by the fact that the transaction queue to be processed can be saturated. Considering that 2 MB also represents a point from

which the latency grows with a lower slope, we can consider that the optimal size should be in a reduced neighborhood of 2 MB.

6.4.3. Optimizing HFB network based on ordering consensus

Ordering consensus was identified as a key parameter in section 4.2. Considering that Kafka-based ordering service is deprecated from HFB 2.0 [17] and Raft provides the equivalent trust model and is easier to manage and operate, in this section we focus on comparing environments with both Raft and Solo. Solo is used only as reference, since as HFB documentation indicates [16], it is not crash fault tolerant and it is fully centralized (see Figure 4). Unlike Solo, Raft-based ordering service integrates all components in the orderer node itself, therefore, it makes sense to include a node in each organization, which represents a totally decentralized environment (see Figure 3). Raft is supported natively from HFB version 1.4. To optimize the ordering consensus to our network, we use the AppUT scenario showed in Figure 18 from *addPolicy* method (PCH) to submit transactions via *PutState* API and analyze the throughput behavior, when input transaction rate is increased. Thus, the experiment is based on the default testbed (Table 6), but the OSN is set to Raft and Solo. In this test scenario, the input transaction rate has been varied between 100 tps and 3500 tps for each OSN.

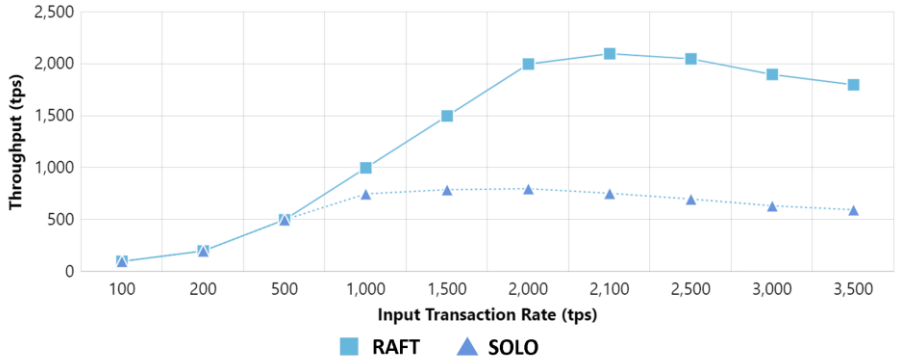

Figure 18: Effects of ordering consensus on transaction throughput.

As Figure 18 shows, the experiment conducted a set of transactions are invoked (input transaction rate), obtaining throughput behavior. Thus, under Raft-based ordering service, network can commit up to 2100 transactions. From this point, due to the saturation of the queues, the number of transactions successfully committed decreases. On the other hand, Solo-based ordering service behavior is worse since it can commit up to only 800

transactions per second since it is only including one node to process all transactions (Figure 4).

6.4.4. Optimizing HFB network based on endorsement policy

In all experiments conducted, the endorsement policy of the chaincode was set such that one peer from each organization needed to endorse every transaction. In section 2, it was mentioned the fact that HFB introduced the approach execute-order-validate, where the execution phase is applied by peer set as endorser which simulate the transactions. For this reason, whether the number of endorser peers is increased, must be obtained a throughput decrease and increase in latency.

From the default testbed (Table 6), we will change the endorsement policy as Table 7 illustrates, where 'a', 'b', 'c' and 'd' denotes four different organizations, maintaining the rest of configuration parameters.

Table 7: Configuration to identify impact of endorsement policies.

	ENDORSEMENT POLICY (AND/OR)
1	OR [a, b, c, d]
2	OR [AND(a,b), AND(a,c), AND(a,d), AND(b, c), AND(b,d), AND(c,d)]
3	OR [AND(a , b, c), AND(a, b, d), AND(b, c, d), AND(a, c, d)]
4	AND [a, b, c, d]

From the AppUT point of view, the experiments conducted are based on the Figure 8 scenario and focus on determining the effects of varying endorsement policies on transaction throughput and latency. The *addPolicy* (PCH) method has been used to perform the tests via *PutState* API. In this test scenario, the input transaction rate has been varied between 100 tps and 3500 tps. Thus, the Figure 19 confirms the expected results, so that as endorsement

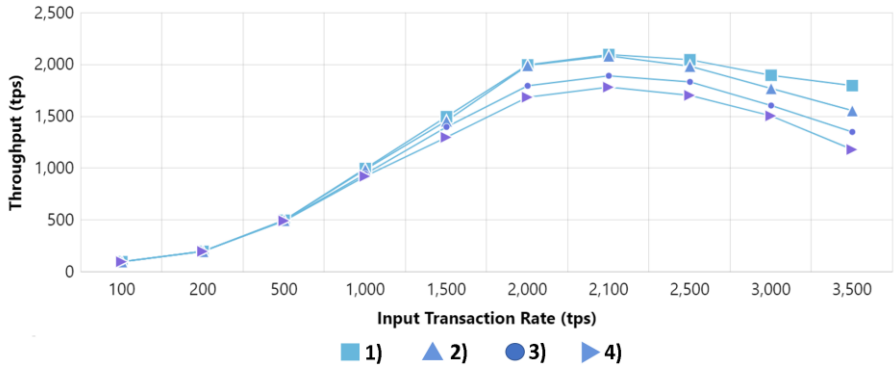

Figure 19: Effect of endorsement policy on transaction throughput.

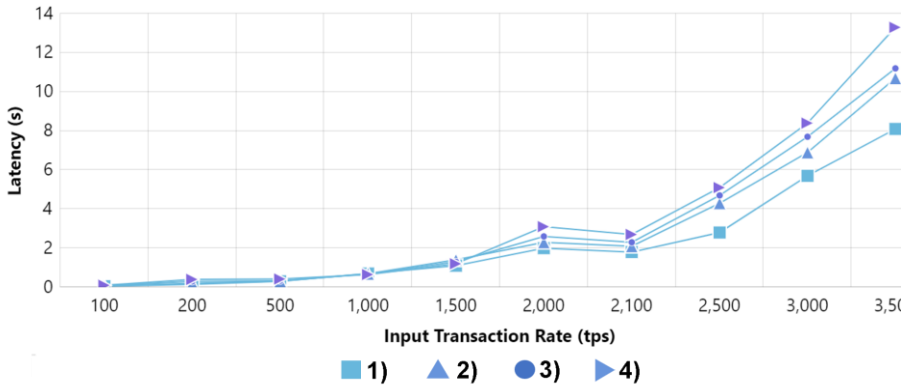

Figure 20: Effect of endorsement policy on latency.

policies need the approval of more organizations, the lower the transaction throughput. Additionally, the Figure 20 also confirms the latency increase when more endorsement policies are needed. After conducting the experiments of section 6.4, we can affirm that the optimal network to deploy our use case is composed by four organization, which use Raft as Ordering Consensus, uses a block size of 2MB and implements an endorsement policy based on the endorsement of a unique organization (OR [a, b, c, d]).

6.5. OVERVIEW OF EXPERIMENT RESULTS AND COMPARISON OF ACCESS CONTROL PROPOSALS

The aim of this section is on the one hand, to summarize each of the results obtained during the experimental phase and on the other hand, to simplify and highlight the novelty of our manuscript with respect to the scientific literature.

Table 8 defines an overview of the results of the experiments, taking into consideration the framework layer, the type of experiment and a brief description of the results obtained.

Table 9 establishes a comparison based on the following topics: 1) execution of the access control policy on-chain or off-chain, 2) information privacy management, and 3) proposal performance analysis.

The importance of the first topic is that adding the on-chain access control policy ensures that this code is executed on the blockchain through the smart contract or chaincode preventing it from being modified or altered.

The second topic allows us to identify information privacy management among the different proposals. In this regard, it is important to highlight that our proposal is the only one based on private data collection (PDC), as the rest includes other alternatives: private data local

management (PDLM), channel privacy (CP) and public transactions (PT). Although private data local management is also feasible as shown in section 6.3, it has the disadvantages that on the one hand the hashing operation is carried out externally from the smart contract and on the other hand, storage is performed in a centralized way. Channel privacy implies that only Hyperledger Fabric's channel participants have access to that channel's information, however, creating separate channels implies additional administrative overhead (maintaining chaincode versions, policies, MSPs), and additionally does not allow for use cases in which you want all channel participants to see a transaction while keeping a portion of the data private. Finally, a public transaction (PT) is the one in which the payload of the transaction is visible to all the participants of the network, so it lacks a mechanism that ensures information privacy.

Table 8: Overview of experiment results.

FWK layer	Experiment	Brief Results description
Chaincode	Feasibility of access control system in terms of latency	The time to recover the asset plus the time to verify asset access is significantly less than the latency time adopted as a reference for an engine assembly phase.
Chaincode	PutState vs PutPrivateState	The throughput achieved in private data collections is lower than regular transactions, keeping similar behavior regardless of StateDB.
Protocol	Feasibility of data privacy solution	The latencies reached in private data collection is higher than regular transactions, however these are lower than the latencies reached by a system that guarantees privacy based on hash operations and local storage (Figure 12).
Architecture	Optimizing HFB network based on StateDB	GoLevelDB offers better throughput than other StateDBs. This behavior is validated by analyzing the lower CPU consumption, as well as the lower processing of writing to a peer's disk.
Protocol	Optimizing HFB network based on block size	2MB is the selected block size as it combines the highest network throughput with an acceptable latency level.
Architecture	Optimizing HFB network based on ordering consensus	A Raft-based ordering service provides the highest performance for a given input transaction rate.
Protocol	Optimizing HFB network based on endorsement policy	The endorsement policy which ensures best performance is OR [a, b, c, d].

The third topic categorizes the performance analysis (PA) of the proposals based on the fact that tools, e.g., Hyperledger Caliper are used for performance framework, as well as the simulation environment isolation with dedicated resources such as cloud environments (e.g.,

AWS), local deployment with dedicated resources, shared external resources (SER) and non-specified resources.

As Table 9 indicates, our proposal ensures novelty based on the established points, since in addition to implementing an on-chain access control policy, it solves information privacy management based on private data collection and includes a detailed performance analysis, based on Hyperledger Caliper tool and an isolated deployment on AWS cloud environment.

Table 9: Comparison table of access control proposals.

Reference	ACPE	IPM	PA	
			Performance Tool	**Simulation Environment Isolation**
Our Proposal	On-Chain	Private Data Collection	Hyperledger Caliper	AWS
[27]	On-Chain	Private Data Local Management	Self-implementation on SDK	Local Resources
[8]	Off-Chain	Public Transactions	Non-specified	Local Resources and Shared External Resources
[25]	Off-Chain	Channel Privacy	Non-specified	Non-specified
[28]	Off-Chain	Public Transactions	Non-specified	Local Resources
[29]	On-Chain	Channel Privacy	Self-implementation on SDK	Local Resources
[30]	On-Chain	Channel Privacy	Self-implementation on SDK	Local Resources
[31]	On-Chain	Channel Privacy	Self-implementation on SDK	Local Resources
[32]	Off-Chain	Public Transactions	Non-specified	Non-specified
[33]	Off-Chain	Public Transactions	Self-implementation on SDK	Shared External Resources

ACPE: Access Control Policy Execution
IPM: Information Privacy Management
PA: Performance Analysis

7. CONCLUSIONS

Hyperledger Fabric Blockchain is a modular project, oriented to environments with high requirements of performance in terms of speed and scalability, preserving the essential properties of a blockchain such as immutability, security through cryptography and the execution of smart contracts (chaincode). Considering that RFID technology presents a significant growth in IIoT environments, being industrial manufacturing one of the most benefited by this growth, our use case is focused on the assembly of engines. Since access

control systems have emerged as an essential solution for IIoT environments and particularly in RFID systems, our proposal consists of a comprehensive solution that covers all facets of access control: Identification, Authentication, Authorization and Accountability/Auditing, i.e., IAAA. Therefore, based on the performance analysis of the Hyperledger Fabric Blockchain network, we have demonstrated the viability of proposed novel access control system based on Hyperledger Fabric Blockchain for an IIoT environment, by applying a methodological performance framework. In addition, we have implemented successful the registration phase of our access control system based on the private data collection solution, which promotes a novel reliable data privacy model applied to an access control system. Additionally, we have demonstrated the feasibility of using Hyperledger Fabric Blockchain's private data collection over a private data local management and finally we have selected the most appropriate combination of network elements and resources for an optimal deployment of the Hyperledger Fabric Blockchain network associated to our use case.

Future lines of research focus on the development of self-sovereign identity models over industrial protocols such as Modbus based on Hyperledger Fabric Blockchain and Hyperledger Indy.

8. ACKNOWLEDGEMENTS

This research was supported in part by the Basque Government (Elkartek program) through projects "SENDAI-SEgurtasun integrala iNDustria AdImentsurako" with project number KK-2019/00072 and "TRUSTIND-Creating Trust in the Industrial Digital Transformation" with project number KK-2020/00054.

9. ACRONYM LIST

ABACP: ABAC Policy
ACH: Authentication Chaincode
ACS: Access Control System
AECH: ABAC Execution Chaincode
AO: Attribute Object
AP: Attributes for Permission
ACPE: Access Control Policy Execution
AppUT: Application Under Test
ArchUT: Architecture Under Test
AS: Attribute Subject
AWS: Amazon Web Service
CA: Certification Authority

CP: Channel Privacy
DBMS: Database Management System
EC2: Elastic Compute Cloud
FWK: Framework
HFB: Hyperledger Fabric Blockchain
IAAA: Identification, Authentication, Authorization and Accountability
IIoT: Industrial Internet of Things
IPM: Information Privacy Management
OSN: Ordering Service Network
PA: Performance Analysis
PCH: Policy Chaincode
PDC: Private Data Collection
PDLM: Private Data Local Management
PT: Public Transactions
RCH: Registration Chaincode
SDK: Software Development Kit
SER: Shared External Resources
SGW: Smart Gateway
SRN: Stochastic Reward Nets
SUT: System Under Test
StateDB: State Database

10. REFERENCES

[1] R. Kandaswamy and D. Furlonger, "Blockchain Technology Spectrum," Stamford, USA, 2019.

[2] X. Xu, G. Sun, L. Luo, H. Cao, H. Yu, and A. V. Vasilakos, "Latency performance modeling and analysis for hyperledger fabric blockchain network," Inf. Process. Manag., vol. 58, no. 1, p. 102436, 2021, doi: doi.org/10.1016/j.ipm.2020.102436.

[3] J. Benet, "IPFS - Content Addressed, Versioned, P2P File System," Jul. 2014.

[4] C. Swedberg, "General Motors Factory Installs Smart Bolts in Engine Blocks, Cylinder Heads," RFID J., pp. 1–2, 2014.

[5] S. Figueroa Lorenzo, J. Añorga Benito, P. García Cardarelli, J. Alberdi Garaia, and S. Arrizabalaga Juaristi, "A Comprehensive Review of RFID and Bluetooth Security: Practical Analysis," Technologies, vol. 7, no. 1, p. 15, 2019, doi: doi.org/10.3390/technologies7010015.

[6] D. Berdik, S. Otoum, N. Schmidt, D. Porter, and Y. Jararweh, "A Survey on Blockchain for Information Systems Management and Security," Inf. Process. Manag., vol. 58, no. 1, p. 102397, 2021, doi: doi.org/10.1016/j.ipm.2020.102397.

[7] S. Figueroa-Lorenzo, J. Añorga, and S. Arrizabalaga, "A Survey of IIoT Protocols: A Measure of Vulnerability Risk Analysis Based on CVSS," ACM Comput. Surv., vol. 53, no. 2, 2020, doi: doi.org/10.1145/3381038.

[8] S. Figueroa, J. Añorga, and S. Arrizabalaga, "An Attribute-Based Access Control Model in RFID Systems Based on Blockchain Decentralized Applications for Healthcare Environments," Computers, vol. 8, no. 3, 2019, doi: doi.org/10.3390/computers8030057.

[9] Q. Nasir, I. A. Qasse, M. Abu Talib, and A. B. Nassif, "Performance Analysis of Hyperledger Fabric Platforms," Secur. Commun. Networks, vol. 2018, p. 3976093, 2018, doi: doi.org/10.1155/2018/3976093.

[10] P. Thakkar, S. Nathan, and B. Viswanathan, "Performance benchmarking and optimizing hyperledger fabric blockchain platform," Proc. - 26th IEEE Int. Symp. Model. Anal. Simul. Comput. Telecommun. Syst. MASCOTS 2018, pp. 264–276, 2018, doi: doi.org/10.1109/MASCOTS.2018.00034.

[11] H. Sukhwani, N. Wang, K. S. Trivedi, and A. Rindos, "Performance modeling of hyperledger fabric (permissioned blockchain network)," NCA 2018 - 2018 IEEE 17th Int. Symp. Netw. Comput. Appl., pp. 1–8, 2018, doi: doi.org/10.1109/NCA.2018.8548070.

[12] A. Baliga, N. Solanki, S. Verekar, A. Pednekar, P. Kamat, and S. Chatterjee, "Performance characterization of hyperledger fabric," Proc. - 2018 Crypto Val. Conf. Blockchain Technol. CVCBT 2018, pp. 65–74, 2018, doi: doi.org/10.1109/CVCBT.2018.00013.

[13] E. Androulaki et al., "Hyperledger Fabric: A Distributed Operating System for Permissioned Blockchains," Proc. 13th EuroSys Conf. EuroSys 2018, vol. 2018-Janua, 2018, doi: doi.org/10.1145/3190508.3190538.

[14] M. Kuzlu, M. Pipattanasomporn, L. Gurses, and S. Rahman, "Performance analysis of a hyperledger fabric blockchain framework: Throughput, latency and scalability," Proc. - 2019 2nd IEEE Int. Conf. Blockchain, Blockchain 2019, pp. 536–540, 2019, doi: doi.org/10.1109/Blockchain.2019.00003.

[15] The Hyperledger White Paper Working Group, "Hyperledger Blockchain Performance Metrics," Hyperledger.org, pp. 1–17, 2018.

[16] Linux Foundation, "hyperledger-fabricdocs Documentation, Release master," Jan 13, 2021, 2019. [Online]. Retrieved: 14-Jan-2021. [From: https://hyperledger-fabric.readthedocs.io/_/downloads/en/release-2.2/pdf/].

[17] J. Yellick, "Deprecate the Kafka-based orderer," 2020. [Online]. Retrieved: 17-Jun-2020. [From: https://jira.hyperledger.org/browse/FAB-16408].

[18] G. Chung et al., "Performance Tuning and Scaling Enterprise Blockchain Applications," 2019.

[19] N. Lincoln, "Hyperledger Caliper," 2019. [Online]. [From: https://hyperledger.github.io/caliper/].

[20] A. de la Rocha, "Performance Best Practices in Hyperledger Fabric II: Infrastructure and Architecture," March 29, 2020, 2020. [Online]. Retrieved: 13-Jun-2020. [From: https://adlrocha.substack.com/p/adlrocha-performance-best-practices-72e].

[21] I. Amazon Web Services, "Amazon Elastic Compute Cloud. User Guide for Linux Instances," 2020.

[22] L. Lamport et al., "In Search of an Understandable Consensus Algorithm," Atc '14, vol. 22, no. 2, pp. 305–320, 2014, doi: doi.org/10.1145/1529974.1529978.

[23] Hyperledger, "hyperledger-fabricdocs Documentation, Release master," 2019.

[24] A. R. Thakre, D. A. Jolhe, and A. C. Gawande, "Minimization of Engine Assembly Time by Elimination of Unproductive Activities through 'MOST,'" in 2009 Second International Conference on Emerging Trends in Engineering & Technology, 2009, pp. 785–789, doi: doi.org/10.1109/ICETET.2009.149.

[25] S. Figueroa, J. Añorga, S. Arrizabalaga, I. Irigoyen, and M. Monterde, "An Attribute-Based Access Control using Chaincode in RFID Systems," in 2019 10th IFIP International Conference on New Technologies, Mobility and Security (NTMS), 2019, pp. 1–5, doi: doi.org/10.1109/NTMS.2019.8763824.

[26] B. Kochie and B. Plumber, "Prometheus," 2020. [Online]. Retrieved: 06-Mar-2020. [From: https://hyperledger-fabric.readthedocs.io/en/release-1.4/metrics_reference.html#id1].

[27] H. Liu, D. Han, and D. Li, "Fabric-iot: A Blockchain-Based Access Control System in IoT," IEEE Access, vol. 8, pp. 18207–18218, 2020, doi: doi.org/10.1109/ACCESS.2020.2968492.

[28] D. R. Putra, B. Anggorojati, and A. P. Pratama Hartono, "Blockchain and smart-contract for scalable access control in Internet of Things," in 2019 International Conference on ICT for Smart Society (ICISS), 2019, vol. 7, pp. 1–5, doi: doi.org/10.1109/ICISS48059.2019.8969807.

[29] M. A. Islam and S. Madria, "A Permissioned Blockchain Based Access Control System for IOT," in 2019 IEEE International Conference on Blockchain (Blockchain), 2019, pp. 469–476, doi: doi.org/10.1109/Blockchain.2019.00071.

[30] Y. E. Oktian and S.-G. Lee, "BorderChain: Blockchain-Based Access Control Framework for the Internet of Things Endpoint," IEEE Access, p. 1, 2020, doi: doi.org/10.1109/ACCESS.2020.3047413.

[31] S. Ding, J. Cao, C. Li, K. Fan, and H. Li, "A Novel Attribute-Based Access Control Scheme Using Blockchain for IoT," IEEE Access, vol. 7, pp. 38431–38441, 2019, doi: doi.org/10.1109/ACCESS.2019.2905846.

[32] X. Ding and J. Yang, "An Access Control Model and Its Application in Blockchain," in 2019 International Conference on Communications, Information System and Computer Engineering (CISCE), 2019, pp. 163–167, doi: doi.org/10.1109/CISCE.2019.00044.

[33] J. P. Cruz, Y. Kaji, and N. Yanai, "RBAC-SC: Role-based access control using smart contract," IEEE Access, vol. 6, pp. 12240–12251, 2018, doi: doi.org/10.1109/ACCESS.2018.2812844.

CHAPTER VII

MODBUS ACCESS CONTROL SYSTEM BASED ON SSI OVER HYPERLEDGER FABRIC BLOCKCHAIN

This chapter was published in S. Figueroa-Lorenzo, J. Añorga, and S. Arrizabalaga, "Modbus access control system based on SSI over Hyperledger Fabric Blockchain," Sensors, MDPI, vol. 21, no. 16, p. 5438, 2021, doi: doi.org/10.3390/s21165438. [JCR. 3.576, Q1].

This chapter focuses on the design and implementation of a Self-Sovereign Identity (SSI) system, over Modbus IIoT an ACS based on SSI over HFB that, through a decentralized identity system that promotes, at the chaincode level, not only on-chain authentication, and authorization but also advanced operations such as signature verification. The system designed is an identity and access management system itself, that provides security to Modbus connections and ensures scalability in environments with more than one organization. The latency and throughput levels achieved for a network of up to 32 clients and one server demonstrate the feasibility to secure the channel for Modbus transactions while ensuring simplicity, compatibility, and interoperability.

1. ABSTRACT

Security is the main challenge of the Modbus IIoT protocol. The systems designed to provide security involve solutions that manage identity based on a centralized approach by introducing a single point of failure and with an ad hoc model for an organization, which handicaps the solution scalability. Our manuscript proposes a solution based on self-sovereign identity over hyperledger fabric blockchain, promoting a decentralized identity from which both authentication and authorization are performed on-chain. The implementation of the system promotes not only Modbus security, but also aims to ensure the simplicity, compatibility and interoperability claimed by Modbus.

2. INTRODUCTION

Modbus became one of the most widely used protocols in Industrial Internet of Things (IIoT) environments and nowadays it is implemented by hundreds of different vendors and thousands of devices [1]. Modbus enables the data exchange between different parts of the industrial process, not only Programable Logic Controller (PLCs) and field devices but also PLCs and Supervisory Control and Data Acquisition (SCADAs). The three main features that enable Modbus success are simplicity, compatibility and interoperability. However, security is its main deficiency. In this regard, vulnerabilities have been detected both in design and configuration as well as in implementation, which are exploited using different attack patterns [1]. Hence, several solutions add security layers while trying to keep it simple, compatible, and interoperable. Some of these solutions belong to identity and access management domains. For instance, this approach provides Modbus with trusted identity and mutual authentication based on X.509 certificates, as well as secure authorization based on a Role-based Access Control (RBAC) [2]. These types of common solutions, based on centralized systems, are optimal in single-organization environments, being the scalability their main drawback (i.e., solution feasibility is affected when multiple organizations are involved). In this sense, the authors of [3] justify scalability problems in terms of PKI infrastructure costs in Smart Grid environments, while the authors of [4] justify it based on scenarios such as certificate revocation. However, a concern with central authorities controlling identities is that if they are compromised in some way, those identities can be used in malicious ways: e.g., a hack of Dutch Certificate Authority allowed supposedly secure encrypted data going across the internet to be intercepted and accessed by hackers [5]. The decentralization promoted by blockchain emerges as a solution to both scalability and centralized identity challenges, since it introduces an environment with multiple organizations, while enabling the

devices' self-custody of identifiers and credentials. In that sense, different works introduce innovative solutions for identity and access control management based on blockchain for IoT environments [6],[7]. However, none of them natively support the use of Self-Sovereign Identity (SSI) for decentralized authentication and authorization and less for an IIoT environment represented by Modbus. The manuscript's contribution is the design and implementation of a Modbus Application Protocol Secured based on Self-Sovereign Identity (mbapSSI) over Hyperledger Fabric Blockchain (HFB) that ensures not only on-chain authentication and authorization but also the other phases of an access control system: identification, auditing, and accountability, constituting an identity and access management system. To verify the feasibility of mbapSSI, performance and scalability analyses are carried out in constrained Modbus-IIoT environments. The rest of this manuscript is structured as follows: Section 3 introduces the context related to blockchain-SSI as an identity and access management system, examining then, the application of SSI in the context IoT and IIoT as part of the related work. Section 4 provides a background including Modbus protocol, SSI, and HFB, i.e., mbapSSI enabling technologies. The design and implementation of our system are described in Section 5. Section 6 describes the testbed and conducted experiments to determine the feasibility of the proposal. Section 7 discusses the results of conducted experiments. Finally, the conclusions of the manuscript are included in Section 8.

3. RELATED WORK

Enabling the aforementioned features for mbapSSI implies going beyond an access control system or an identity management system. For this reason, this section discusses the integration of both concepts. Based on the Identity and Access Management (IAM) Framework [8], the scientific community, governments and enterprises establish blockchain as an enabling technology for both Identity Management (IM) and Access Management (AM). In the identity management domain, Blockchain Identity Management Systems (BIMS) are considered as an emerging technology, which differs of Traditional Identity Management System (TIMS) since blockchain enables the custody of identity information, while TIMS stores credentials (e.g., password) about users and devices which they interact [9]. Currently, governments and organizations are working hard to create guidance on BIMS. Thus, some countries, such as Estonia, are experimenting with BIMS for electronic medical records [10]. In the access management domain, blockchain technology has been included as part of several scenarios that involve all phases of Access Control Systems: identification, authentication, authorization, auditing and accountability [11]. At this point, the SSI paradigm, does not need blockchain to be implemented. Nevertheless, blockchain offers benefits that

can be exploited by an integrated SSI-blockchain solution: blockchain can be used as a distributed ledger to establish immutable records of lifecycle events for globally unique decentralized identifiers (DIDs). In addition, SSI-blockchain integration can be the enabler for identity and access management. Although the authors of [12] classify SSI only as an emerging identity management system, the fact that SSI ensures not only authentication, but also authorization [13], implies that it is also involved in the context of access management. Once having discussed that SSI is suitable for IAM, we will focus on the related work in the application of SSI in the context of IoT and IIoT due to the recent integration of both concepts.

Thus, the following manuscript [14] contrasts existing identity approaches, such as digital certificates, with open standards for SSI: decentralized identifiers (DIDs) and verifiable credentials. The same work also analyzes the advantages and challenges of both standards for ensuring authentication in IoT environments, although these are only proposals that do not result in the design or implementation of a specific system but are part of future research lines. Additionally, the work in [15] compares SSI-blockchain solutions such as Sovrin, UPort and Jolocom, while promoting the integration of SSI in IoT architectures, enabling comparisons with OpenID Connect. However, again, this work is only a very useful proposal (not implementation) in terms of background and analysis of possible use cases. Furthermore, the Sovrin approach indicates, from a non-technical perspective, how SSI addresses main IoT challenges such as identification, authentication, authorization and auditing, while ensuring data privacy and data integrity over a secure channel [16]. However, this manuscript also focuses on proposals rather than specific designs or implementations of SSI. Finally, the authors of [17] present a novel solution for IoT device IM based on SSI and backed by the security offered by the IoTA Tangle DLT. Although the authors even present a practical use case focused on car rental industry, there are no technical details neither performance analysis of the implementation of the solution. Despite the progress described above, numerous challenges face SSI in both the IoT and IIoT contexts, including authentication and authorization in machine-to-machine (M2M) environments [13]. Therefore, mbapSSI aims to address these challenges, not only from a theoretical design in a representative IIoT use case, but also from the implementation and feasibility analysis point of view, trying to solve the mentioned gaps. Thus, mbapSSI does not only guarantee the decentralized identity management that blockchain promotes, but also includes the design and implementation of authentication and authorization on-chain. The mbapSSI approach consists of authorizing the use of a resource and giving access to a Modbus service, putting in value the role of blockchain technology for SSI, as a trust mechanism that allows controlling access to data, allowing the Modbus device to decide with whom, when and how it shares information.

4. BACKGROUND

The three main concepts of mbapSSI are discussed in this section: Modbus protocol, SSI and Hyperledger Fabric Blockchain.

4.1. MODBUS PROTOCOL

Modbus is an Industrial Internet of Things (IIoT) protocol with applications in scenarios such as building automation or energy management systems [1]. Modbus is useful for the management and control of industrial devices such as Programable Logic Controller (PLCs), Supervisory Control and Data Acquisition (SCADAs), sensors and actuators. The three most common versions of the protocol are Modbus ASCII, Modbus RTU and Modbus TCP/IP, where the first two use serial line interfaces and are associated with deterministic transmissions. Modbus TCP/IP arises with the objective of solving requirements such as the limit to 240 devices per network, however, the non-deterministic nature of TCP/IP means that it is common to see this version of Modbus in interactions between supervisory level and field level devices (e.g., SCADAs and PLCs) rather than interactions between field devices with each other such as a PLC and a sensor. From performance perspective, Modbus-IIoT environments can be considered as constrained scenarios, which can result in latencies of up to 100 ms according to [18]. It is well known that the major shortcomings of Modbus are in the area of security, for this reason references such as [19] present an extensive taxonomy of attacks for both Modbus serial and Modbus TCP. Considering that standard Modbus protocol cannot be secured, e.g., providing authentication to the Modbus frame, whatever security layer provided, it must not limit the simplicity, compatibility and interoperability of the Modbus protocol.

4.2. SELF-SOVEREIGN IDENTITY (SSI)

SSI follows the basic premise that people should control their own identity with regard to relationships and interactions with other people, organizations, and things [20]. However, the evolution of the concept of SSI extends the control of their identity to the context of things and machines, as mentioned in Section 3. The principles guiding SSI have been presented in the past by Christopher Allen [21], however, approaches that extend these principles have been analyzed by the authors of [12], including concepts such as (1) sovereignty, (2) data access control, (3) data storage control, (4) security, (5) privacy, (6) flexibility, (7) accessibility and (8) availability.

The SSI ecosystem assumes three key roles (issuer, holder, and verifier) in addition to a verifiable data and status registry (VDSR). An issuer creates and issues credentials to a holder. A holder receives credentials from an issuer, holds them and when required, it shares these credentials with a verifier. A verifier receives and verifies credentials presented by a holder. A VDSR is a trusted mediator to manage and verify relevant information, e.g., identifiers. The verifiable credentials' specification defines examples of VDSR such as a trusted database, decentralized databases, and distributed ledgers [22].

SSI integrates some standards such as Verifiable Credential (VC) [22] and Decentralized IDentifier (DID) [23], which are proposed to create a cryptographically verifiable digital identity that is fully controlled by its owner [24]. A VC is an attestation of qualification issued by a third party (e.g., issuer) with de facto authority to an entity (e.g., holder) [24]. In this regard, VCs are JSON documents constructed and digitally signed by an authority (e.g., issuer) which includes possible correlating values such as a holder identifier, the signature value and the claim value. The DID is a permanent, universally unique identifier and cannot be taken away from its owner who owns the associated private key, which is completely different from other identifiers such as an IP address and domain name [24]. Two other important concepts associated with DID should be analyzed: DID Method and DID Document. DID Methods define the types or classes of DIDs and represents the second part of the DID format. Several types of DID Methods are set out in [25], including Ledger-based DIDs, Ledger Middleware, Peer DIDs, Static DIDs and Alternative DIDs. Since Ledger-based DIDs are the most commonly used, some examples are shown next: btcr (Bitcoin), ethr (Ethereum), and sov (Sovrin). DID Document is a data structure that contains basic information that is needed to connect with a subject (e.g., a verifier). The DID Document is resolved through a DID and consumed by digital identity applications (e.g., wallet), so that each DID has exactly one DID Document associated with it [25].

4.3. HYPERLEDGER FABRIC BLOCKCHAIN

HFB is a private and permissioned distributed ledger and smart contract framework maintained by Linux Foundation [26]. Considering that HFB is pivotal to our proposal, we will justify its selection as a VDSR of our SSI proposal from three perspectives. First, the use of HFB as part of the IAM context, highlighting its suitability in SSI scenarios. Second, the justification of using HFB over other SSI implementations. Thirdly, from a performance perspective given our focus related to IIoT environments.

In the identity management domain, HFB is involved in several use cases. For instance, the work in [27] uses HFB as a Certification Control System to enforce Certificate Signing

Request (CSR) validation. In addition, HFB is also used to provide transparency for better Certificate Authority (CA) accountability [28]. In the access management domain, there are also several use cases, including HFB as part of SSI systems. Thus, several approaches use Access Control Systems based on HFB to ensure phases such as identification, authentication, authorization and accountability in IIoT environments [15],[29],[30]. Moreover, in the context of SSI, although the scientific literature offers few examples involving HFB in SSI systems, some steps have been taken towards this direction. Thus, the contribution in [31] designs and implements both an SSI and an access management system for smart vehicles, using HFB as a key piece.

References such as [15] analyze SSI solutions including Sovrin, Civic, UPort, Jolocom, Veres One, etc. These solutions mostly implement two types of blockchain technologies: Ethereum and Hyperledger Indy, which will be the objects of our comparison with HFB. Although solutions such as uPort (Ethereum) seem promising [32], on-chain environments are limited from both solidity capabilities and order–execute architecture. In contrast, HFB through the execute–order–validate model eliminates non-determinism and enables standard programming languages such as Golang, which amplifies the on-chain capability of the solution. Additionally, Hyperledger Indy is a purpose-built ledger for Identity, which can be considered a public-permissioned blockchain [33]. However, since Indy does not have the ability to run smart contracts, authentication and authorization are based on information stored in the ledger using static and predefined rules. From the perspective of mbapSSI, the on-chain dynamism provided by a chaincode, allowing complex computational operations such as on-chain signature verification, exceeds the performance of Indy, a scenario that implies the need to strengthen the off-chain implementation.

From a performance perspective, according to [34], blockchain addresses challenges defined as key for IIoT environments such as system scalability and interoperability, security and data quality. Particularly, it has been demonstrated the feasibility of an HFB-based access control system in a performance-constrained IIoT environment such as an engine assembly line [35].

5. DESIGN AND IMPLEMENTATION

This section establishes the design and implementation criteria defined for mbapSSI. For a better understanding of these criteria, we contextualize the background provided for SSI in terms of key concepts for mbapSSI.

5.1. KEY CONCEPTS FOR MBAPSSI

5.1.1. Modbus End-Devices Analysis for mbapSSI

Section 4.1 mentioned the low latency requirements for a Modbus-IIoT environment. Considering the network architecture recommended by NIST [36], mbapSSI puts the focus on the interaction between field control devices with local human–machine interface (HMI), i.e., TCP/IP connections. In this way, a Modbus client could be visualized as a Scada, while the Modbus server as a PLC. In addition to network communication, Modbus devices store a key pair associated with the DID as part of a wallet.

5.1.2. DID Approach for mbapSSI

Expression (1) shows the Ledger-based DID contextualization for mbapSSI on a three-element identifier separated by a colon: scheme identifier "did", DID Method identifier and DID method-specific identifier. The approach adopted in mbapSSI for three-parameter definition is based on [25], following the DID syntax based on Augmented Backus-Naur Form (ABNF) Syntax Specification [37] as DID Standard defines [23]. In this regard, the DID Method indicates the ledger used as VDSR, defining as DID work with a specific blockchain. The specific identifier of the DID Method satisfies the property of being unique in the namespace of the DID Method, since it is the SHA-256 hash function of the public key registered in the VDSR, being the entity identified by the DID, the holder of the private key. Additionally, with regard to the DID infrastructure, mbapSSI promotes a key-value database, where key is the DID and value is the DID Document. Thus, we can confirm that the DID adopted by mbapSSI complies with the properties of being permanent, resolvable, cryptographically verifiable and decentralized [38], being therefore a very close approximation to the DID standard by satisfying some of the core properties of the standard.

did:hfb:1fb352353ff51248c5104b407f9c04c3666627cf5a167d693c9fc84b75964e2, **(1)**

Considering that mbapSSI aims to improve the shortcomings of the PKI infrastructure presented for Modbus in terms of scalability, the first objective of the DID will be to integrate it into X509 certificates, to replace the functionality of the Certification Authority (CA) associated with a PKI infrastructure, as a base for the security of Modbus devices in centralized environments according to the Modbus security specifications [39]. The purpose behind not discarding the use of X509 certificates is that it is a mandatory requirement for securing the channel using TLS. In this way, the DID of the Modbus device is recorded as the Subject Alternative Name (SAN) extension within the X.509 certificate [40], allowing

the VDSR to become the authority for verifying identity rather than a CA when establishing a secure channel. In this way, when Modbus devices exchange certificates as part of TLS, they will be able to verify the DID through the VDSR. The second objective of DID is to return the associated DID Document when the VDSR is properly queried, thus fulfilling the resolvable property.

5.1.3. DID Document Approach for mbapSSI

Our mbapSSI uses the "resolvable" property of DID to record, in the VDSR, the key-value type for the DID-DID Document relation. Thus, our DID Document contains a context, the authentication mechanism supported, the Modbus service definition, and the digital signature of the entity that creates it. The Modbus service definition is fully compliant with Modbus specifications [41], so it includes the service endpoint, the function code, the starting address and the offset. As expression (2) indicates, service endpoint is constituted by the Modbus application protocol secured (mbaps) frame type, as well as IP address and access port. Each of the Modbus services to be exposed must be defined in the DID Document. The DID Document can be constantly updated by adding or removing Modbus services.

```
"serviceEndpoint": "mbaps://{}:{}".format(ipaddress, port),          (2)
```

5.1.4. SSI without VC Approach for mbapSSI

Although in Section 4.2, VCs were defined, their standard indicates that their use is not mandatory [22]. Thus, mbapSSI does not follow the VC model but focuses on complying with the SSI properties established by [12], particularly in data access control, data storage control, security and privacy. In that sense, mbapSSI promotes a decentralized identity that highlights the property of sovereignty in the fact that the Modbus server has the ability to share its data (through a Modbus service) with the authorized Modbus client. In mbapSSI, the Modbus client provisioning on the authorization whitelist is performed through the intervention of a user. It is identified as next objective the design and implementation of an SSI model based on M2M authentication proposed by OAuth in [42], but it is out of the scope of this work.

The authorization whitelist includes as input an out-of-band (OOB) mechanism provided by mbapSSI to associate the client's public key to a Modbus service included in the DID Document. This authorization whitelist will be updated in the VDSR and constitutes the single point of user intervention.

5.2. OVERVIEW OF MBAPSSI

In order to integrate the concepts defined in the Section 5.1, as well as to provide an overview of mbapSSI, Table 1 relates the parts that compose an SSI system with the mbapSSI system phases. In this regard, the registration phase provides the Modbus device (holder) with a decentralized identity, ensuring that this identity is registered in the VDSR. In this phase, the DID has been integrated into the X509 certificate. In the provisioning phase, the Modbus server's owner (user) matches the Modbus clients with the Modbus services to be used, so that the authorization whitelist is updated. It should be noted that the provisioning phase is unrelated to the others, so it can be repeated at any time. The channel securing phase provides the Modbus devices (both being holders and verifiers as it will be clarified later on), with on-chain authentication, as well as the establishment of a secure TLS channel [43]. The verification phase authorizes the Modbus client (holder) to use a resource, i.e., a Modbus service, through a proof of identity provided by the on-chain query of the authorization whitelist, where the Modbus server (verifier) is involved. Finally, Modbus transactions phase enables the exchange of Modbus frames through a secure channel between Modbus client and server (both holders). The VDSR is involved in all phases of mbapSSI except when Modbus devices transact with each other to ensure both low levels of latency and high throughput.

According to [44], in an SSI context each entity can have multiple roles, e.g., holders can also be verifiers. Extrapolating this concept to a Modbus context throughout the mbapSSI phases, the devices supporting the Modbus protocol, i.e., the Modbus client and the Modbus server, may behave as holders or verifiers depending on the phase requirement. In this regard, we define a Modbus client acting as a holder as C_H, Modbus client acting as a verifier as C_V, Modbus server acting as a holder as S_H and Modbus server acting as a verifier as S_V. Table 2 summarizes the behaviors per phase.

Table 1: Participation of SSI parties on mbapSSI's phases.

Phase	User	Holder	Verifier	VSDR
Registration		X		X
Provisioning	X			X
Channel securing		X	X	X
Verification		X		X
Modbus transaction		X		

Figure 1 shows an overview of the interaction between parties of the SSI context and Modbus devices of the Modbus context. As Table 2 indicates, the parties involved in the registration phase (C_H and S_H) register their identities in the VDSR (1). Next, Modbus

service(s) to be exposed are registered in the VDSR by S_H (2). At this point and as part of the provisioning phase, in the step (3), the server's owner relates Modbus client information to the Modbus service(s) exposed in step (2), and then S_H updates the whitelist (4).

Table 2: Modbus devices' behavior.

Phase	Device
Registration	C_H, S_H
Channel securing	C_H, S_H, C_V, S_V
Verification	C_H
Modbus transaction	C_H, S_H

The channel securing phase is composed of two steps. On the one hand, C_H and S_H start to ensure the communication channel via TLS (5). On the other hand, C_V and S_V must verify the identities through the VDSR (6). At this point, C_H must provide access proof for services through the VDSR (7), based on the information defined in steps (3–4). Finally, Modbus devices (C_H and S_H) exchange Modbus transactions for authorized services (8).

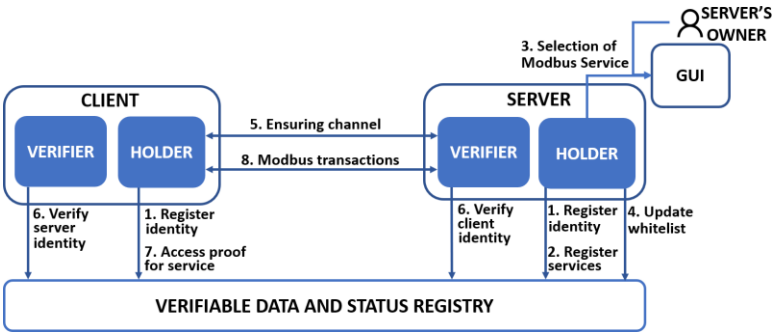

Figure 1: Overview of mbapSSI.

The correlation between Modbus and SSI converges in mbapSSI, where the three initial phases (Registration, Channel Securing and Verification), excluding the Provisioning phase, constitute the necessary mechanism, based on SSI, to provide a secure communication channel, based on TLS, between a client and a Modbus server. The fourth phase is thus isolated to ensure the simplicity, compatibility, and interoperability of Modbus transactions.

5.3. CHAINCODE DESIGN

The chaincode is the most important piece of mbapSSI. Its strength lies in the fact that complex operations such as signature verification can be performed on-chain, something practically impossible in public-permissionless blockchains such as Ethereum, due to the high transactions cost. Therefore, it is possible to assert that the role of HFB for mbapSSI goes beyond a simple VDSR. From a design perspective, the main feature of the chaincode is the

presence of a proxy, which is an approach pioneered by UPort [45], although UPort deploys a new smart contract proxy for each identity. In that sense, the design of the proxy as part of our chaincode adopts the identity verification functionality as well as the routing capability to other functionalities. However, the mbapSSI chaincode does not need to generate a new Smart Contract for each identity but relies on on-chain mechanisms such as signature verification to perform identity verification. Therefore, in our case the identity is stored in the ledger and can be queried from the proxy itself. Figure 2 describes the functionalities of mbapSSI's chaincode: *Get Identity*, *Set Identity*, *Update Whitelist*, *Get Entity's Identity*, *Set DID Document* and *Get DID Document*. The default design criterium is that any input to the chaincode from the different entities (C_H, C_V, S_H, and S_V) must have a DID and payload structure as shown in expression (3). For payload processing by the chaincode, it must meet two conditions: (1) the sender identity must be registered and (2) the payload must be signed by the sender.

```
{
did: "did:hfb:1fb352353ff51248c5104b407f9c04c3666627cf5a167d693c9fc84b75964e2",
payload: "eyJhbGdvcml0aG0iOiJQUzI12In0.eyJmdW5jdGlvbiI6ImNyZWF0ZVNlbGZJZGVudG"
},                                                                              (3)
```

To enable these conditions, a proxy structure has been designed that basically verifies both identity and signature, and then deserializes the payload in order to obtain the service to be executed. For this purpose, the proxy relies on other structures, such as the signature verifier and the identity registry, which are involved in the other functionalities except for the *Set Identity* where the payload of expression (3) only contains the public key of the entity. In this regard, the *Get Identity* functionality is only executed within the chaincode and is called several times in all functionalities, in order to retrieve the public key for an entity with a registered identity. In addition, the *Update Whitelist* functionality stores the set of relations of the public key with the Modbus service index in the DID Document. On the other hand, the *Get Entity's Identity* functionality returns to off-chain side the public key associated to an identity. Additionally, the *Set DID Document* functionality allows to create and later update the DID Document. Finally, the *Get DID Document* functionality allows to retrieve the DID document, for which, a JSON Web Token (JWT) must be included to prove not only the identity of the client, but also the validity of the authorization given to use the service defined in the whitelist. Thus, the public key associated to the C_H identity is used to retrieve the Modbus service(s) index(es) from the whitelist, returning to off-chain side, the DID Document associated to the C_H-related services in provisioning phase.

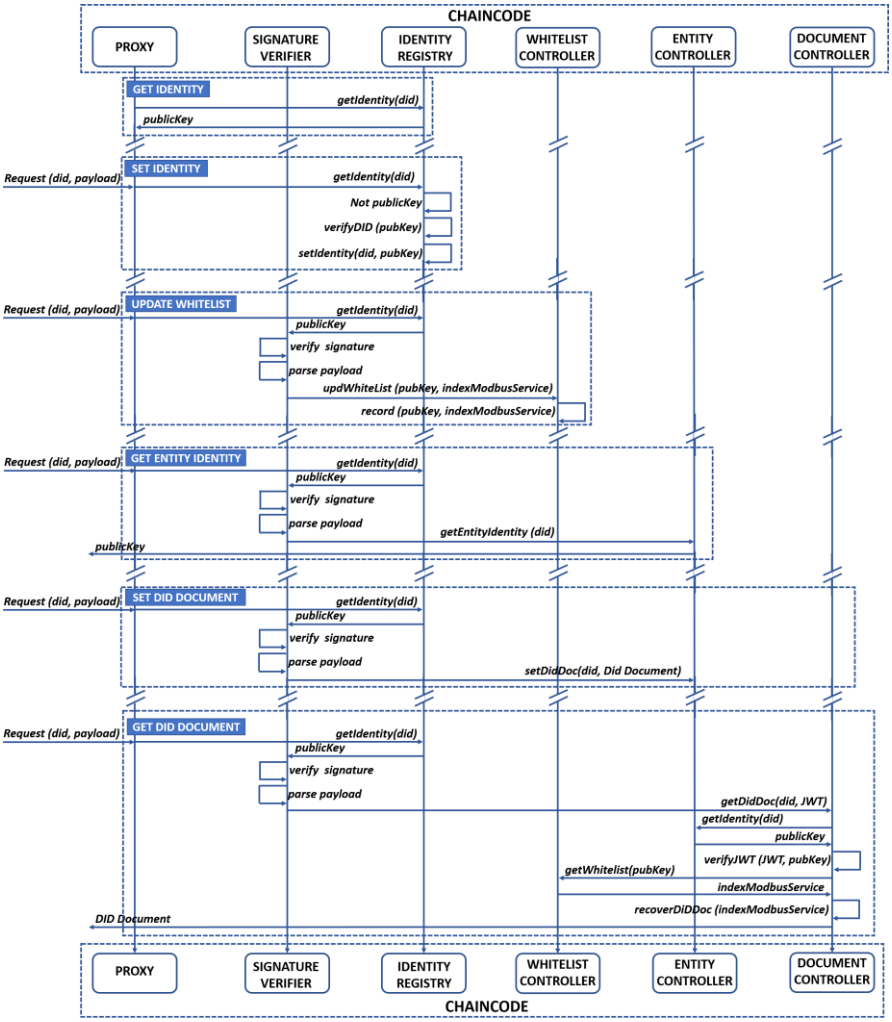

Figure 2: Chaincode design.

5.4. IMPLEMENTATION

The mbapSSI system is composed of the aforementioned chaincode and a Software Development Kit (SDK) to enable the integration and interaction with HFB network. Thus, while the chaincode manages the interactions in the on-chain side, the mbapSSI-SDK manages the interactions in the off-chain side. This SDK, written in python is imported into Modbus devices, i.e., client and server, to expose all functionalities required to interact with the HFB network, enabling these devices to become C_H, C_V, S_H, and S_V. However, mbapSSI-SDK imports three key pieces of code for the different interactions throughout the mbapSSI

lifecycle. Firstly, mbapSSI-SDK makes use of the HFB SDK for python to invoke (write operations) and query (read operations) transactions on HFB [46]. Secondly, mbapSSI-SDK requires the use of the python SSL library, to build TLS/SSL wrapper for socket objects [47]. Finally, to transact Modbus frames from clients and servers, mbapSSI-SDK imports the pymodbus library [48], a full Modbus protocol coded in python. Since python is not one of the official HFB languages, it makes it necessary to use the HFB version to 1.4.x since the python HFB-SDK updates are delayed for languages such as JavaScript. Thus, to run mbapSSI, both the client and server must include python 3.8 and mbapSSI-SDK into docker images.

The Figure 3 shows a flowchart, which details interactions between entities (C_H, C_V, S_H, and S_V), including the information exchanged with the VDSR, corresponding to the phases of mbapSSI. Although the figure uses generic names for easy understanding of the functionalities (e.g., Request Did Document), each interaction with the blockchain follows the format defined by expression (3).

Now, as part of the registration phase both S_H and C_H create and register their DID in the VDSR via the setIdentity method. Next, S_H is able to create the DID Document via the *setDidDocument* method, although it may be updated each time the server has to expose a new service using the same method. The only requirement for the provisioning phase to take place is that S_H has registered the DID Document, so that the set of services to be exposed are selectable by the server's owner. At this point, the user relates the public key of the Modbus client (C_H) and the service(s) he wants to access, using an OOB mechanism such as a graphical interface (GUI). This relationship is updated in the blockchain via the *updWhiteList* method. As part of channel securing phase, the C_H request TLS connection and after some steps of TLS specification [43] S_H sends its certificate (X.509), where, according to Section 5.1.2, both C_H and S_H have included DID as part of the certificate's SAN extension. In this way, both client and server, with verifier role, i.e., C_V and S_V will be able to verify each other's identity (via *getIdentity* method), since blockchain (VDSR) plays the role of CA. Once the mutual TLS is completed, the channel is secured. As part of the verification phase, C_H computes off-chain JWT, which is included as argument of *getDidDoc* method. Thus, the VDSR requests on-chain, the authorization whitelist returning the service(s) enabled to C_H. In this way, S_H authorizes the use of a resource, granting access to a Modbus service, without the need for interaction with C_H, putting in value the role of blockchain technology for SSI, as a trusted mechanism that allows regulating the access to the S_H data, enabling it to decide with whom and when it shares its information. C_H and S_H are ready to transact Modbus frames.

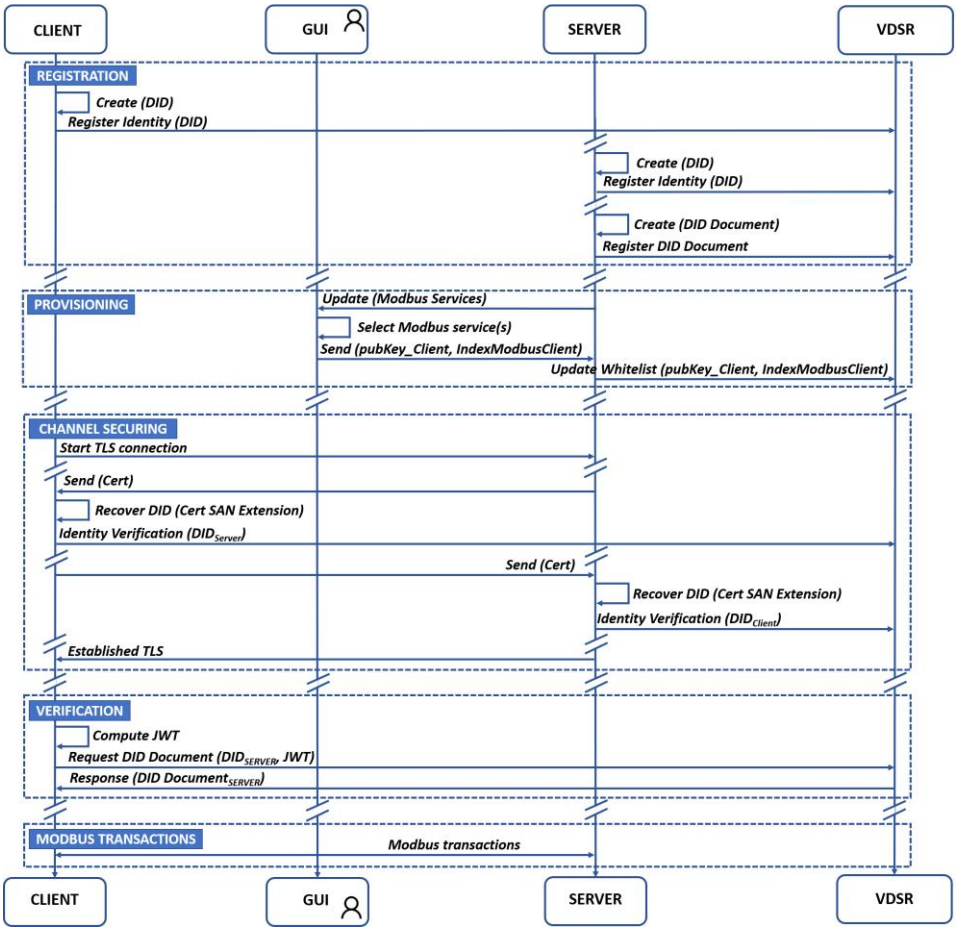

Figure 3: Interaction between entities and information processing by VDSR.

In order to clarify the off-chain process, when the client requests the DID Document within the verification phase, the mbapSSI-SDK will be used in the off-chain side, in or der to create an object containing the *getDidDoc* method in JSON format. This method uses as arguments a JWT and the server's DID. This plain text object is firstly encoded in base 64, constituting the first part of the payload of expression (3) which uses the dot as a delimiter. Secondly, the same object is signed and then encoded in base 64, constituting the second part after the delimiter. At this point, the client DID and payload are sent as transaction arguments.

6. PERFORMANCE EVALUATION

This section focuses on the evaluation of mbapSSI in terms of performance to determine its feasibility. The performance tests aim at isolating the first three phases from the fourth phase.

This is because the value of mbapSSI lies in the fact that the three initial phases, excluding the provisioning phase, constitute the necessary mechanism (based on SSI), to provide a secure communication channel (based on TLS) between a client and a Modbus server. Therefore, the fourth phase is abstracted from the others to ensure the simplicity, compatibility and interoperability of Modbus transactions.

Three performance metrics (processing time, latency, and throughput) are used for the performance evaluation. Processing time represents the time required for a process (e.g., a piece of code) to handle a given request. Latency is a measure of round-trip delay. Throughput constitutes the ratio of the total transactions committed into the ledger in 1 s (not all the sent transactions are always committed in the same second and are pooled in the ordering node). The three metrics are evaluated in different ways, so processing time is used to measure the performance of the mbapSSI phases: registration, channel securing and verification (hereinafter referred to as "three-phases"); latency is used to measure the performance of Modbus transactions; throughput is used to measure the performance of mbapSSI interactions with HFB. Similarly, scalability is another important point in the evaluation of mbapSSI performance, to determine the behavior of mbapSSI due to the increase of organizations and therefore of requests, an n:1 connection pattern is followed, i.e., "n" clients to "1" server. The provisioning phase is not measured since it constitutes an OOB mechanism isolated from the other phases. The remainder of this section examines the testbed used and the experiments conducted to demonstrate the feasibility of mbapSSI.

6.1. TESTBED DESCRIPTION

Figure 4 shows the test network to be deployed. It is composed of organizations, each of which contains a set of entities such as Modbus entity, orderer, peer0.org and CA. Each organization constitutes a module and as needed new modules are added to the testbed, which provides scalability. To achieve integration within a module the deployment flexibility offered by the Docker infrastructure is used so that all elements represented for an organization (Modbus entity, orderer, peer0.org and CA) are deployed from Docker images. These containers are integrated into Docker Swarm worker nodes. In this way each organization constitutes a worker node which will be orchestrated using Docker swarm.

The default testbed would be two modules (organizations) including 1 Modbus-client (e.g., Modbus-client1) and 1 Modbus-server. To obtain accurate results, an isolation environment is needed, which is provided by an Amazon Elastic Compute Cloud (EC2) instance, whose main feature is scalable performance from a basic level of CPU and memory. The mbapSSI testbed included the deployment of a t2.2xlarge instance, which contains 8 intel AVXT CPUs

(3.0 GHz) and 32 GB of memory [49]. It should be noted that the main decision criteria for the deployment of a testbed with these characteristics has been that the HFB-network is deployed using Raft as an ordering service network instead Kafka or Solo since it introduces less centralization [35].

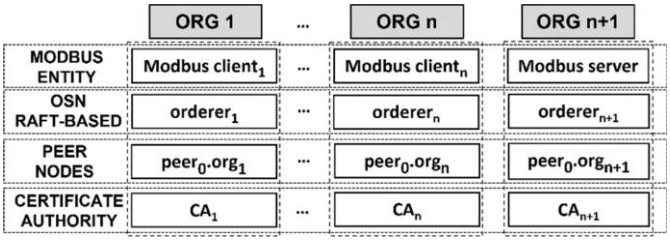

Figure 4: Testbed applied on performance measure.

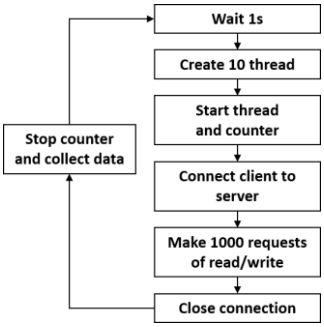

Figure 5: Flow chart of Modbus performance test.

6.2. EXPERIMENTS CONDUCTED

This section describes the experiments to be performed to demonstrate the feasibility of mbapSSI and determine the overhead that mbapSSI introduces to the HFB network. For this purpose, four types of experiments are conducted: Section 6.2.1 describes the experiment that measures the processing time of the three-phases and the latency of the Modbus transaction phase with a client–server 1:1 ratio; Section 6.2.2 describes the experiment that measures the behavior of the processing times of the three-phases of mbapSSI considering the scaling of the client–server ratios as follows: 4:1, 8:1, 16:1 and 32:1; Section 6.2.3 describes the experiment measuring the latency behavior of the Modbus transaction phase of mbapSSI in isolation, considering the scaling of client–server ratios as follows: 4:1, 8:1, 16:1 and 32:1; Section 6.2.4 describes the experiments to determine how many transactions per second are supported by the deployed HFB network.

6.2.1. Performance of mbapSSI Phases at 1:1 Ratio

This experiment focuses on measuring the processing time of the three-phases of mbapSSI that provide a decentralized identity, a secure channel, and the authorization to use the Modbus resource, as well as the latency of the Modbus transaction phase. From the testbed of Figure 4, this experiment requires the deployment of two organizations, one holding the client and the other holding the server. In order to measure the processing time, since mbapSSI is based on python 3.8, we use the "process_time" tool of the time library. Expressions (4) and (5) are the ways of using the tool, i.e., inserting them in the code at the beginning and at the end of each phase, respectively. To determine the latency of the Modbus transactions, a performance test was designed based on the flowchart shown in Figure 5, which represents a piece of code written in Python running on each Modbus client, which simultaneously opens 10 connections (threads) to the server, performing 1000 requests for each of the threads, collecting latency introduced by each of these connections, so that the maximum and minimum value of this metric can be evaluated.

$$start = time.process\_time\ (), \tag{4}$$

$$end = time.process\_time\ () - start, \tag{5}$$

6.2.2. Performance of the First Three Phases of mbapSSI based on n:1 Ratio

This experiment aims to measure the behavior of the processing time of the three-phases of mbapSSI under the stimulus of scaling the number of Modbus clients connected simultaneously in ratios of 4:1, 8:1, 16:1 and 32:1, following the architecture of Figure 4. The maximum number of clients is 32, since this is the number commonly supported by real devices such as the MGate MB3170/3270 as shown in its technical documentation [50]. Unlike the previous section, this experiment focuses on analyzing only the behavior of processing time for three-phases of mbapSSI, using the same tools as in the first experiment, i.e., expressions (4) and (5). For this purpose, 10 processing time measurements are collected for each of the three-phases.

6.2.3. Performance of the Modbus transaction Phase at n:1 Ratio

This experiment aims to measure the behavior of the latencies of the Modbus transaction phase under the stimulus of scaling the number of simultaneously connected Modbus clients in ratios of 4:1, 8:1, 16:1 and 32:1, following the architecture of Figure 4. As in Section 6.2.1, this experiment uses the performance test illustrated in Figure 5. Considering that mbapSSI

Modbus frames are transacted over a secure channel, the experiment includes, as a benchmark, the same performance test applied to Modbus TCP.

6.2.4. Measuring transaction throughput over the HFB network

This experiment aims to determine the number of transactions per second that the different network architectures deployed are able to support, so that it is possible to analyze whether mbapSSI can cause overhead in the HFB network. For this purpose, 1000 transactions are issued in a variable range, for each of the networks (1:1, 4:1, 8:1, 16:1 and 32:1) defined based on Figure 4. The Hyperledger Caliper tool [51], simplifies the transaction evaluation workflow. Thus, a comparison between five types of networks created from docker swarm clusters containing different architectures is carried out. Then, the maximum transaction throughput achieved will be compared with the number of interactions performed by clients and server with HFB, assuming that all devices are connected simultaneously. All nodes in the HFB-network deployed have the same participation in transaction endorsement.

7. DISCUSSION OF RESULTS

7.1. PERFORMANCE OF MBAPSSI PHASES AT 1:1 RATIO

This section contains the results of the experiment described in Section 6.2.1 and its main objective is to determine a set of benchmark metrics to be used as a starting point for the analysis in the other sections, since it constitutes the best case. Table 3 and Table 4 summarize the maximum and minimum values of the processing time, as well as the number of interactions with the blockchain for both the client and the server for each of three-phases of mbapSSI. The registration phase presents a longer processing time for S_H than the C_H, since it not only creates and registers its identity but also creates and registers the DiD Document. The channel securing phase presents similar behavior for both entities, as Figure 3 illustrates, based on the same number of interactions of both entities (C_V and S_V) with the blockchain, as well as a number of symmetric interactions between C_H and S_H. It should be noted that only C_H participates in the verification phase. The processing time of the verification phase involves the generation of a JWT by C_H, as well as a single interaction with the VDSR, hence it is as simple as C_H's processing time in the registration phase.

At this point, the performance test of the Figure 5 is performed on both the mbapSSI secure connection and insecure Modbus TCP connection, which is used for benchmark purposes. Table 5 collects the maximum and minimum latencies for both connections. Since the latencies achieved are below the time defined as typical reaction time for an industrial TCP

connection, i.e., 100 ms [18], it can be considered that for this mbapSSI baseline case, the achieved latency values are acceptable.

Table 3: Processing time measurements of three-phases of mbapSSI for the client.

Phase	Modbus Client Time Min	Modbus Client Time Max	HFB invocations	HFB queries
Registration	12.1ms	19.7ms	1	0
Channel securing	26,2ms	33,8ms	0	1
Verification	14.6ms	19,2ms	0	1
Total	**52,9ms**	**71.7ms**	**1**	**2**

Table 4: Processing time measurements of three-phases of mbapSSI for the server.

Phase	Modbus Server Time Min	Modbus Server Time Max	HFB invocations	HFB queries
Registration	23,9ms	31,2ms	2	0
Channel securing	28,7ms	39,6ms	0	1
Total	**51.6ms**	**70.8ms**	**2**	**1**

Table 5: Latencies measurements of the fourth phase of mbapSSI.

Phase	Client-Server TCP Min Time	Client-Server TCP Max Time	Client-Server TLS Min Time	Client-Server TLS Max Time
Modbus transaction	0,49ms	0,54ms	1,12ms	1,37ms

7.2. Performance of the Three-Phases of mbapSSI Based on n:1 Ratio

In this section we analyze the effects, in terms of maximum and minimum processing times, when the number of mbapSSI client scales follows the next ratios: 4:1, 8:1, 16:1 and 32:1. In this way, a comparison with the results of Section 6.1 is established. Figure 6 shows the behavior of processing time when the number of clients deployed for the registration phase is increased. For each architecture, only one server is deployed and, since C_H and S_H do not interact with each other at this phase, the processing times obtained for the server are practically the same as those achieved in Section 6.1. In this way, the server processing time is closer to the processing time for registering 8 clients if the maximum and minimum processing times are considered. Figure 7 also shows the behavior of the processing time when the number of clients deployed for the securing channel phase is increased. Unlike the registration phase, in this phase there is a client–server interaction, hence the behavior of the server will depend on the number of clients simultaneously interacting with it. The results show that the higher the number of clients, the higher the maximum processing time for the server, which can be explained by the overhead accumulated after processing interactions with some clients. Despite this scenario, when comparing the extreme cases for the server, i.e., the reference architecture (1:1) and the worst case, i.e., the 32:1 architecture, the

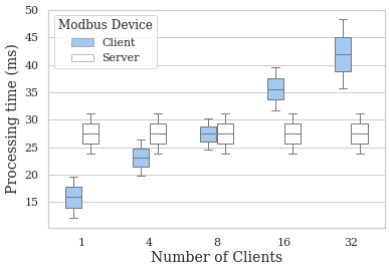

Figure 6: Processing time of Registration phase.

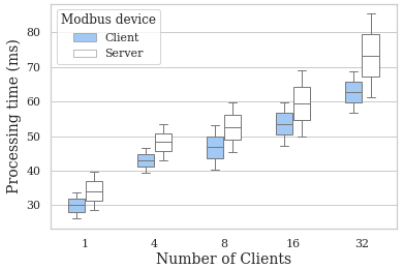

Figure 7: Processing time of Channel securing phase.

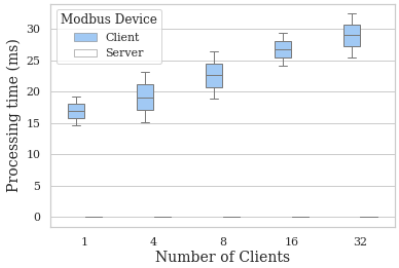

Figure 8: Processing time of Verification phase.

differences between the maximum and minimum processing time values are still acceptable: 45.8 ms for the maximum processing time and 32.7 ms for the minimum processing time. Figure 8 shows the behavior of the processing time for the verification phase when the number of clients increases. When comparing the minimum and maximum processing time values between clients of the reference architecture, i.e., 1:1, and the 32:1 architecture, the difference is 10.9 ms for the minimum processing time and 13.2 ms for the maximum processing time, keeping acceptable performance levels in both cases.

7.3. Performance of the Modbus Transaction Phase at N:1 Ratio

The isolation of the fourth phase of mbapSSI aims to ensure that once the first three phases of mbapSSI have been successfully completed, the Modbus transactions maintain the requirements of simplicity, compatibility, and interoperability. For this purpose, using as a reference the latency achieved by applying the performance test in Figure 5 over Modbus TCP, the same experience is compared over Modbus TLS. Figure 9 and Figure 10 include the repetition for the experiment (latencies' measurements for Modbus transactions) for the next client–server architectures: 4:1, 8:1, 16:1 and 32:1.

The best- and worst-case scenarios are compared to reach conclusions, i.e., the latency of the reference architecture 1:1 and the worst case, i.e., the 32:1 architecture. Thus, 1 TCP

client has a maximum latency of 0.54 ms, while 1 TLS client has a maximum latency of 1.37 ms. Using the same measurement criteria per client, 32 TCP clients have a latency of 39.3 ms, while 32 Modbus TLS have a latency of 63.3 ms. All these times achieved are below the time defined as typical reaction time for an industrial TCP connection, i.e., 100 ms [18], hence it can be considered that mbapSSI allows to keep the aforementioned Modbus properties.

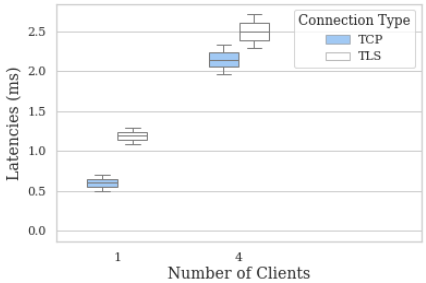

Figure 9: Latencies measurements in the fourth phase for the following architectures: 1:1 and 4:1.

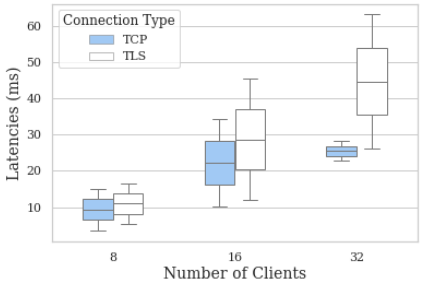

Figure 10: Latencies measurements in the fourth phase for the following architectures: 8:1, 16:1 and 32:1.

7.4. MEASURING TRANSACTION THROUGHPUT OVER THE HFB NETWORK

As mentioned in Section 6.2.4, 1000 transactions are sent at variable rate for each of the five possible network scenarios to determine the performance of these transactions. Based on the chaincode design of Figure 2, the setIdentity method was selected as the test method because it is performed by the participant entities at boot time. Considering that the block size is fixed at 100 transactions per block or 2 MB [35] and that the validation policy forces all organizations to validate transactions, i.e., all organizations have the same weight in the consensus, Figure 7 shows the throughput behavior for each of the defined architectures.

The saturation point should be highlighted for each of the five cases, since it represents the point from which the number of transactions per second stops growing. Analyzing the extreme cases, 1:1 and 32:1, with the best and worst saturation point, respectively, the 1:1 architecture reaches saturation for a throughput of 129.6 tps, i.e., of the 150 transactions sent, 129.6 transactions are successfully committed into the ledger in 1 s, while the 32:1 architecture reaches saturation for a throughput of 37.2 tps, i.e., of the 50 transactions sent, 37.2 transactions are successfully committed into the ledger in 1 s. The remaining transactions sent will be committed in subsequent intervals. To determine the overhead that the mbapSSI boot time can cause on HFB, from Figure 5 the number of interactions are computed until all identities have been registered. Thus, for this interval, the identity registration of the C_H and the S_H will be considered as a concurrent transaction. Table 6

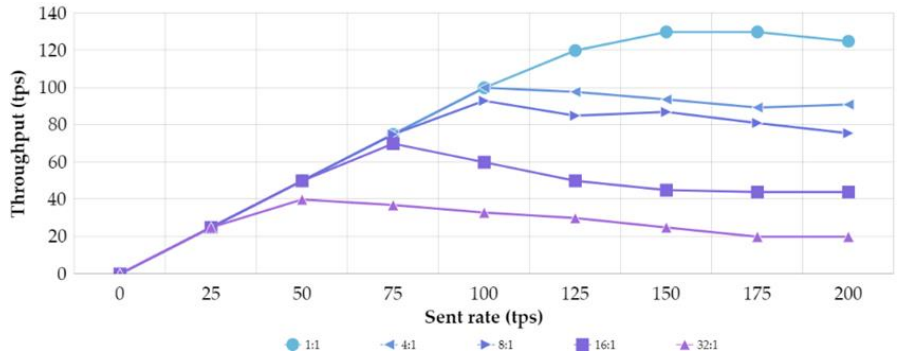

Figure 11: mbapSSI throughput behavior for different architectures.

collects for each architecture the concurrent transactions, the saturation point, and the sending rate associated with that saturation. Hence, in the 32:1 scenario, where the simultaneous registration of the 32 clients and 1 server (which requires 66 concurrent transactions), reach the saturation point, the remaining pending transactions would be committed in the next second. However, considering that this registration is carried out only once and is not interfering in the transaction execution time, the boot-time overhead of mbapSSI over HFB might not be considered a problem.

Table 6: Determination of the boot-time overhead of mbapSSI over HFB.

Architecture	mbapSSI concurrent transactions	Saturation point (tps)	Sent rate (tps)
1:1	2	129.6	150
4:1	10	100	100
8:1	18	97	100
16:1	34	70	75
32:1	66	37.2	50

Nevertheless, there are also ways to avoid this overhead. It is necessary on the one hand to decrease the block size, since performance analyses show decreasing the block size implies better throughput and lower latency [52]. On the other hand, reducing the number of endorsing peers in the HFB-network allow fewer entities to execute transactions and therefore transaction could be ordered into the block quickly; however, this implies that there must be greater trust between the organizations.

This results discussion section has demonstrated the feasibility of mbapSSI in terms of performance and scalability. The achievement of reasonable processing times in each of the phases that ensure Modbus transactions, the latency levels below the benchmark for industrial environments achieved in Modbus transactions and the optimal performance of

mbapSSI interactions with HFB, for environments with up to 32 deployed organizations attest that statement.

8. CONCLUSIONS

Modbus is a widely used IIoT protocol based on three main features: simplicity, compatibility, and interoperability, but which lacks security. In this regard, Access Control Systems emerge as a solution. However, common access control solutions are based on centralized systems that include well known drawbacks: a single point of failure and limited scalability. This manuscript examines an Access Control System based on Self-Sovereign Identity over Hyperledger Fabric Blockchain from an Identity and Access Management perspective. The designed decentralized identity system supports on-chain authentication and authorization. Hence, it provides not only security for Modbus connections, but also ensures scalability in environments with more than one organization. The performed experiments and a subsequent critical discussion demonstrate that processing times achieved for the registration, channel securing and verification phases, as well as the latency achieved for Modbus transactions and the throughput achieved for mbapSSI transactions over HFB guarantee both the feasibility and scalability of mbapSSI and the simplicity, compatibility, and interoperability of Modbus. However, self-sovereign identity in machine-to-machine (IIoT) environments is in its infancy, therefore, our next approach is to eliminate the need for user involvement, achieving a fully machine-to-machine interaction, for which we are currently studying the OAuth machine-to-machine scheme to guarantee access to resources [42]. This approach should be fully compliant with DID standard [23]. Likewise, authorization mechanisms based on verifiable credentials are under study, a line in which Siemens is undertaking important steps to support selective disclosure based on Zero Knowledge Proofs [53]. These are the subjects of future lines of research. Additionally, mbapSSI supports the default accountability provided by blockchain, so we are also developing specific chaincodes for monitoring logs and event emission that integrate Access Control Systems.

9. ACKNOWLEDGEMENTS

This research was funded by Elkartek program of the Basque Government through the project: “TRUSTIND-Creating Trust in the Industrial Digital Transformation” with grant number KK-2020/00054..

10. REFERENCES

[1] S. Figueroa-Lorenzo, J. Añorga, and S. Arrizabalaga, "A Survey of IIoT Protocols: A Measure of Vulnerability Risk Analysis Based on CVSS," ACM Computer Survey, vol. 53, no. 2, 2020, doi: doi.org/10.1145/3381038.

[2] S. Figueroa-Lorenzo, J. Añorga, and S. Arrizabalaga, "A Role-Based Access Control Model in Modbus SCADA Systems. A Centralized Model Approach," Sensors , vol. 19, no. 20. 2019, doi: doi.org/10.3390/s19204455.

[3] Smith, S.W. Cryptographic scalability challenges in the smart grid (extended abstract). In Proceedings of the 2012 IEEE PES Innovative Smart Grid Technologies (ISGT), Washington, DC, USA, 16–20 January 2012; pp. 1–3.

[4] Slagell, A.; Bonilla, R.; Yurcik, W. A survey of PKI components and scalability issues. In Proceedings of the 2006 IEEE International Performance Computing and Communications Conference, Phoenix, AZ, USA, 10–12 April 2006; pp. 10–484.

[5] The Weakest Link in the Chain: Vulnerabilities in the ssl Certificate Authority System and What Should Be Done about Them. Available online: https://www.accessnow.org/cms/assets/uploads/archive/docs/Weakest_Link_in_the_Chain.pdf (accessed on 22 July 2021).

[6] H. Liu, D. Han, and D. Li, "Fabric-iot: A Blockchain-Based Access Control System in IoT," IEEE Access, vol. 8, pp. 18207–18218, 2020, doi: doi.org/10.1109/ACCESS.2020.2968492.

[7] S. Figueroa, J. Añorga, and S. Arrizabalaga, "An Attribute-Based Access Control Model in RFID Systems Based on Blockchain Decentralized Applications for Healthcare Environments," Computers, MDPI, vol. 8, no. 3, p. 19, 2019, doi: doi.org/10.3390/computers8030057.

[8] A. Caballero, "Chapter 24 - Information Security Essentials for Information Technology Managers: Protecting Mission-Critical Systems," in Computer and Information Security Handbook (Third Edition), Third Edit., J. R. Vacca, Ed. Boston: Morgan Kaufmann, 2017, pp. 393–419.

[9] L. Lesavre, P. Varin, P. Mell, M. Davidson, and J. Shook, "A taxonomic approach to understanding emerging blockchain identity management systems," arXiv:1908.00929v2, 2019, doi: doi.org/10.6028/NIST.CSWP.07092019-draft.

[10] Martinson, P. Estonia—The Digital Republic Secured by Blockchain; PricewaterhouseCoopers: London, UK, 2019; pp. 1–12.

[11] I. Butun and P. Österberg, "A Review of Distributed Access Control for Blockchain Systems Towards Securing the Internet of Things," IEEE Access, vol. 9, pp. 5428–5441, 2021, doi: doi.org/10.1109/ACCESS.2020.3047902.

[12] Naik, N.; Jenkins, P. Governing Principles of Self-Sovereign Identity Applied to Blockchain Enabled Privacy Preserving Identity Management Systems. In Proceedings of the 2020 IEEE International Symposium on Systems Engineering (ISSE), Vienna, Austria, 12 October–12 November 2020; pp. 1–6.

[13] Bartolomeu, P.C.; Vieira, E.; Hosseini, S.M.; Ferreira, J. Self-Sovereign Identity: Use-cases, Technologies, and Challenges for Industrial IoT. In Proceedings of the 2019 24th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA), Zaragoza, Spain, 10–13 September 2019; pp. 1173–1180, doi: doi.org/10.1109/ETFA.2019.8869262.

[14] Fedrecheski, G.; Rabaey, J.M.; Costa, L.C.P.; Calcina Ccori, P.C.; Pereira, W.T.; Zuffo, M.K. Self-sovereign identity for IoT environments: A perspective. In Proceedings of the 2020 Global Internet of Things Summit (GIoTS), Dublin, Ireland, 3 June 2020..

[15] Kulabukhova, N.; Ivashchenko, A.; Tipikin, I.; Minin, I. Self-Sovereign Identity for IoT Devices. In Computational Science and Its Applications—ICCSA 2019; Springer: Cham, Switzerland, 2019; pp. 472–484.

[16] Self-Sovereign Identity and IoT. 2020. Available online: https://sovrin.org/wp-content/uploads/SSI-and-IoT-whitepaper.pdf (accessed on 14 January 2021).

[17] Gebresilassie, S.K.; Rafferty, J.; Morrow, P.; Chen, L.; Abu-Tair, M.; Cui, Z. Distributed, Secure, Self-Sovereign Identity for IoT Devices. In Proceedings of the 2020 IEEE 6th World Forum on Internet of Things (WF-IoT), New Orleans, LA, USA, 2–16 June 2020; pp. 1–6.

[18] Lin, Z.; Pearson, S. An Inside Look at Industrial Ethernet Communication Protocols Strategic Marketing Manager Texas Instruments Strategic Marketing Manager Texas Instruments; White Paper; Texas Instruments: Dallas, TX, USA, 2018.

[19] Huitsing, P.; Chandia, R.; Papa, M.; Shenoi, S. Attack taxonomies for the Modbus protocols. Int. J. Crit. Infrastruct. Prot. 2008, 1, 37–44.

[20] Allen, C. Self-Sovereign Identity: Ideology & Architecture. 16 March 2020. Available online: https://ssimeetup.org/self-sovereign-identity-why-we-here-christopher-allen-webinar-51/ (accessed on 26 July 2021).

[21] Allen, C. Self-Sovereign Identity Principles. 2016. Available online: https://github.com/ChristopherA/self-sovereign-identity/blob/master/self-sovereign-identity-principles.md (accessed on 26 July 2021).

[22] Sporny, M.; Longley, D.; Chadwick, D. Verifiable Credentials Data Model 1.0. 19 November 2019. Available online: https://www.w3.org/TR/vc-data-model/ (accessed on 11 February 2021).

[23] Reed, D.; Sporny, M.; Longley, D.; Allen, C.; Grant, R.; Sabadello, M. Decentralized Identifiers (DIDs) v1.0. 11 February 2021. Available online: https://www.w3.org/TR/did-core/#method-schemes (accessed on 11 February 2021).

[24] Khovratovich, D.; Law, J. Sovrin: Digital Identities in the Blockchain Era; Github Commit by jasonalaw; Sovrin Foundation: Northampton, MA, USA, 2016.

[25] Preukschat, A.; Reed, D. Self-Sovereign Identity Decentralized Digital Identity and Verifiable Credentials, 1st ed.; Manning Publications Co.: Shelter Island, NY, USA, 2021; ISBN 9781617296598.

[26] Linux Foundation, "hyperledger-fabricdocs Documentation, Release master," Jan 13, 2021, 2019. [Online]. Available: https://hyperledger-fabric.readthedocs.io/_/downloads/en/release-2.2/pdf/. (Accessed: 14-Jan-2021).

[27] Kinkelin, H.; von Seck, R.; Rudolf, C.; Carle, G. Hardening X.509 Certificate Issuance using Distributed Ledger Technology. In Proceedings of the NOMS 2020-2020 IEEE/IFIP Network Operations and Management Symposium, Budapest, Hungary, 20–24 April 2020; pp. 1–6, doi: doi.org/10.1109/NOMS47738.2020.9110311.

[28] Madala, D.S.V.; Jhanwar, M.P.; Chattopadhyay, A. Certificate transparency using blockchain. In Proceedings of the 2018 IEEE International Conference on Data Mining Workshops (ICDMW), Singapore, 17–20 November 2018; pp. 71–80, doi: doi.org/10.1109/ICDMW.2018.00018.

[29] Ding, S.; Cao, J.; Li, C.; Fan, K.; Li, H., "A Novel Attribute-Based Access Control Scheme Using Blockchain for IoT," IEEE Access, vol. 7, pp. 38431–38441, 2019, doi: doi.org/10.1109/ACCESS.2019.2905846.

[30] Figueroa, S.; Añorga, J.; Arrizabalaga, S.; Irigoyen, I.; Monterde, M., "An Attribute-Based Access Control using Chaincode in RFID Systems," in 2019 10th IFIP International Conference on New Technologies, Mobility and Security (NTMS), 2019, Canary Islands, Spain, 24–26 June 2019, pp. 1–5, doi: doi.org/10.1109/NTMS.2019.8763824.

[31] Terzi, S.; Savvaidis, C.; Votis, K.; Tzovaras, D.; Stamelos, I, "Securing Emission Data of Smart Vehicles with Blockchain and Self-Sovereign Identities," in 2020 IEEE International Conference on Blockchain (Blockchain), 2020, pp. 462–469, doi: doi.org/10.1109/Blockchain50366.2020.00067.

[32] Panait, A.E.; Olimid, R.F.; Stefanescu, A. Analysis of uPort Open, an Identity Management Blockchain-Based Solution; Springer International Publishing: Berlin/Heidelberg, Germany, 2020; Volume 12395, ISBN 9783030589851.

[33] Shcherbakov, A. Understanding the Hyperledger Indy Distributed Ledger. 2019. Available online: https://wiki.hyperledger.org/display/RU/Understanding+the+Hyperledger+Indy+Distributed+Ledger (accessed on 23 July 2021).

[34] Yalcinkaya, E.; Maffei, A.; Akillioglu, H.; Onori, M. Empowering ISA95 compliant traditional and smart manufacturing systems with the blockchain technology. Manuf. Rev. 2021, 8, 15.

[35] Figueroa-Lorenzo, S.; Añorga, J.; Arrizabalaga, "Methodological performance analysis applied to a novel IIoT access control system based on permissioned blockchain," Information Processing & Management., vol. 58, no. 4, p. 102558, 2021, doi: doi.org/10.1016/j.ipm.2021.102558.

[36] Fabro, M.; Gorski, E.; Spiers, N., "Recommended Practice: Improving Industrial Control System Cybersecurity with Defense-in-Depth Strategies Industrial Control Systems Cyber Emergency Response Team", St. Elizabeths West Campus: Washington, DC, USA, 2016.

[37] Crocker, D.; Overell, P. Augmented BNF for Syntax Specifications: ABNF; No. 5234; RFC Edi8or: Vashon, WA, USA, 2008..

[38] Reed, D. Webinar: Decentralized Identifiers (DIDs) SSIMeetup Objectives. Available online: https://ssimeetup.org/decentralized-identifiers-did-fundamental-block-self-sovereign-identity-drummond-reed-webinar-2/ (accessed on 11 February 2021).

[39] Modbus/TCP Security; Modbus Organization: Hopkinton, MA, USA, 2018; Available online: https://modbus.org/docs/MB-TCP-Security-v21_2018-07-24.pdf (accessed on 2 March 2021).

[40] Boeyen, S.; Santesson, S.; Polk, T.; Housley, R.; Farrell, S.; Cooper, D.I., "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile," no. 5280. RFC Editor, May-2008, doi: doi.org/10.17487/RFC5280, Available online: https://www.rfc-editor.org/info/rfc5280 (accessed on 2 March 2021).

[41] Modbus Organization. Modbus Application Protocol Specification; Modbus Organization: Hopkinton, MA, USA, 2012.

[42] Peyrott, S. Machine to Machine Communications. Available online: https://auth0.com/blog/using-m2m-authorization/ (accessed on 17 June 2021).

[43] Dierks, T.; Rescorla, E. The Transport Layer Security (TLS) Protocol Version 1.3; RFC Editor: Vashon, WA, USA, 2018.

[44] Baliga, A. The Nuts and Bolts of Decentralized Identity. 23 February 2021. Available online: https://aratibaliga.substack.com/p/the-nuts-and-bolts-of-decentralized (accessed on 2 March 2021).

[45] Naik, N.; Jenkins, P. uPort Open-Source Identity Management System: An Assessment of Self-Sovereign Identity and User-Centric Data Platform Built on Blockchain. In Proceedings of the 2020 IEEE International Symposium on Systems Engineering (ISSE), Vienna, Austria, 12 October–12 November 2020; pp. 1–7.

[46] D. Enyeart, "Hyperledger Fabric-SDK-Py," 2021-02-23, 2021. [Online]. Available: https://fabric-sdk-py.readthedocs.io/en/latest/index.html, (accessed on 14 May 2021).

[47] The Python Software Foundation. TLS/SSL Wrapper for Socket Objects. 2017. Available online: https://docs.python.org/3/library/ssl.html (accessed on 14 May 2021).

[48] RiptideIO. PyModbus-A Python Modbus Stack; GitHub: Santa Barbara, CA, USA, 2018; p. 2.

[49] Documentation Team. Amazon Elastic Compute Cloud User Guide for Windows Instances; Samurai Media Limited: Thames Ditton, UK, 2018; ISBN 978-9-88-840815-3I.

[50] Modbus, E. MGate MB3170/MB3270 Series. Available online: https://www.moxa.com/en/products/industrial-edge-connectivity/protocol-gateways/modbus-tcp-gateways/mgate-mb3170-mb3270-series#overview (accessed on 5 March 2021).

[51] Lincoln, N. Hyperledger Caliper. 2019. Available online: https://hyperledger.github.io/caliper/ (accessed on 5 March 2021).

[52] Thakkar, P.; Nathan, S.; Viswanathan, B. Performance benchmarking and optimizing hyperledger fabric blockchain platform. In Proceedings of the 2018 IEEE 26th International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (MASCOTS), Milwaukee, WI, USA, 25–28 September 2018; pp. 264–276, doi: doi.org/10.1109/MASCOTS.2018.00034.

[53] Pfaff, O.; Kind, A. Does Industrial Asset Management Provide Good Use Cases for Verifiable Credentials and Distributed Ledgers? What Is an Industrial Automation Component? 2021. Available online: https://hgf2021.sched.com/event/j3fM/does-industrial-asset-management-provide-good-use-cases-for-verifiable-credentials-and-distributed-ledgers-oliver-pfaff-andreas-kind-siemens-ag (accessed on 17 June 2021).

CHAPTER VIII

DISCUSSION AND FUTURE RESEARCH LINES

This chapter discusses the main contributions of this Ph.D. Thesis, by starting with an overview of all the contributions from different perspectives together with a more specific discussion for each of the contributions. Finally, some future research lines are identified.

1. DISCUSSION

This thesis starts with C_1 contribution related to the VAF methodological framework, which enables the evaluation of not only the vulnerabilities related to IIoT technologies but also their impact in IT and OT environments. Then, the other contributions (C_2-C_6) provide different solutions (including the design, implementation and evaluation) within Access Control Systems topic for a variety of scenarios, covering different use cases, latency-constrained vs non-latency-constrained environments, types of access control models and Access Control Systems, degree of decentralization, and use of types of blockchain technologies, which will be described and compared next.

IIoT is very extensive and enables different business scenarios which different requirements in terms of access control. Certainly, latency and performance levels determine the scenario of access control applications, as can be appreciated in Table 1 and Figure 1.

Table 1: Relationship of use cases and contributions.

ACS Contribution	Use case	IIoT		Latency constraint
		IoT	OT	
C_2	ICS		X	X
C_3	Healthcare	X		
C_4	Railway	X		
C_5	Engine Assembly Line		X	X
C_6	ICS		X	X

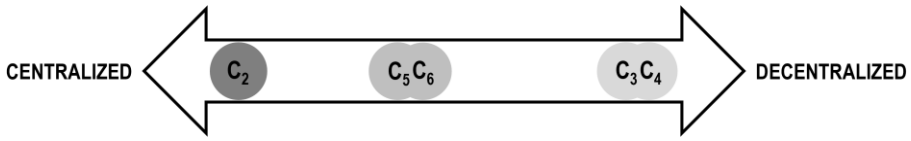

Figure 1: Architecture decentralization level.

Table 1 describes the relationship between uses cases, types of IIoT environments and latency constraints, while Figure 1 illustrates the decentralization level of the architecture deployed for the contribution. As it can be seen, there is a close relationship between the latency constraint, the use case, and the IIoT environment. While C_2, C_5, and C_6 contributions enable performance-constrained environments with latency requirements below 100 ms and therefore the IIoT environment is close to OT, C_3, and C_4 contributions enable environments without a specific latency requirement and therefore the IIoT environment is close to IoT. Regarding the technologies and protocols involved in the use cases, the C_2 contribution describes an ICS use case based on a fully centralized architecture for the

Modbus protocol. The **C_3 contribution** deploys the architecture designed as a DApp, that uses the RFID protocol on the Ethereum public-permissionless blockchain network for the Healthcare use case. The **C_4 contribution** also deploys the architecture designed as a DApp on the Ethereum public-permissionless blockchain network for the Railway use case. The **C_5 contribution** shows an Engine Assembly Line use case (where the RFID protocol is used) and the **C_6 contribution** is related to an ICS use case (where the Modbus protocol is used). Both architectures introduce a certain degree of centralization due to the use of a private permissioned network such as Hyperledger Fabric Blockchain.

In addition, an analysis of the Access Control Systems contributions related to the IIoT ecosystem-level can be performed, as summarized in **Table 2**.

Table 2: Relationship of the IIoT ecosystem level and ACS contributions.

IIoT Ecosystem Level	ACM	IAA	IAAA	
	ABAC	RBAC	ABAC	-
Endpoint	C_3	C_2	C_5	C_6
Data			C_5	
Communication		C_2		C_6

Table 2 shows how the **C_2 contribution** presents an Access Control System based on roles that impact the endpoint protection (in this case, the Modbus server endpoint) thanks to the identity based on X509 certificates, the authentication provided by TLS, and the authorization provided by the role included on the X.509 certificate. Similarly, the contribution impacts the protection of the communication channel thanks to the security provided by TLS between endpoints, i.e., Modbus client and Modubs server. Additionally, the **C_3 contribution** presents an Access Control System based on attributes that impact endpoint (RFID tagged healthcare assets) protection thanks to the off-chain execution of the access control policy that authorizes the endpoints based on immutably recorded attributes on the blockchain. The **C_5 contribution** involves endpoint (RFID tagged engine part) protection thanks to the on-chain identification, authentication and authorization based on attributes performed on the blockchain. In addition, **C_5** ensures protection at the data level (i.e., attributes used to perform both authentication and authorization) thanks to the private data collections provided by Hyperledger Fabric Blockchain. The **C_6 contribution** performs on-chain identification, authentication and authorization to protect endpoints (i.e., Modbus client and Modbus server). Similarly, the contribution enhances the protection of the communication channel thanks to the security provided by TLS between endpoints.

Finally, **Table 3** lists the uses of blockchain for the contribution in terms of computation and storage, considering that the type of blockchain-based Access Control System also depends

on the performance and the scalability of the IIoT environment to be protected. In this regard, the contributions can be grouped according to the type of blockchain used. On the one hand, the public-permissionless scenario aims to reduce storage and transaction costs through mechanisms that do not reintroduce trust as an assumption and also require careful consideration so as not to impair responsiveness.

Table 3: Blockchain uses in terms of computation and storage.

Blockchain Type	Computation		Storage	
	On-Chain	Off-Chain	On-Chain	Off-Chain
Public-Permissionless		C_3	C_3	
Public-Permissionless		C_4		C_4
Private-Permissioned	C_5		C_5	
Private-Permissioned	C_6		C_6	

On the other hand, the private-permissioned scenario ensures trust and availability of information, increasing processing costs and usability of resources. Thus, the C_3 contribution uses off-chain computation so that the access control policy is executed completely off-chain which helps to reduce processing redundancy and thus scalability while reducing costs per transaction. Only attributes relevant for decision support are stored on-chain on the Ethereum blockchain. The C_4 contribution focuses not only on storing data off-chain to reduce storage consumption by clients and to improve confidentiality by hiding data from other nodes in the network but also on processing the information off-chain, reducing transactions costs over the Ethereum blockchain. The C_5 and C_6 contributions enable both on-chain computation and storage on Hyperledger Fabric Blockchain while maintaining high levels of performance. Thus, the C_5 contribution adds a privacy layer using the Hyperledger Fabric Blockchain's private data collections, while executing on-chain all phases of an Access Control System, i.e., IAAA. The C_6 contribution performs on-chain complex operations such as signature verification, while not only implementing a decentralized identity but also executing all phases of an Access Control System, i.e., IAAA.

Once having globally discussed the contributions of the thesis, a more specific discussion is included for each of the contributions in the following sections.

1.1. Discussion of C_1 contribution (Chapter II)

The Industrial Internet of Things (IIoT) architecture is complex due to, among other things, the convergence of protocols, standards, and buses. This convergence not only makes interoperability difficult but also makes security one of the main challenges of IIoT. Because of this reason, the C_1 contribution conducts a comprehensive survey of the thirty-three most useful protocols, standards, and buses in IIoT environments regarding features such as

architecture, topology, protocol, and security. The existence of security problems, not only in the design or configuration, but also in the implementation of the assets that support these protocols, standards, and buses was determined from this analysis. Therefore, it was necessary to perform a comprehensive assessment to measure the impact of existing documented vulnerabilities on assets implementing the referred protocols, standards, or buses. Since the CVSS offers a way to collect the main properties of a vulnerability and generate a numbered score that indicates its severity, we design a vulnerability analysis framework (VAF) to process 1363 vulnerabilities corresponding to the thirty-three protocols, standards, and buses studied to determine that both the severity and impact of a vulnerability are usually higher if the asset is part of an Operational Technology (OT) environment than of an Information Technology (IT) environment.

Hence, objective O_1 together with partial objectives $O_{1.A}$, $O_{1.B}$ and $O_{1.C}$ have been successfully committed. As a reminder, the O_1 states: "*To review IIoT technologies (protocols, standards, and buses) and provide a mechanism to evaluate the impact of their vulnerabilities in IT and OT environments"*, and the specific objectives are also fulfilled:

$O_{1.A}$: "To *provide a mechanism to study the set of vulnerabilities documented in assets that use the protocols, standards, and buses studied."*

In the contribution, the VAF methodological framework has been designed based on the CVSSv3 metric, to analyze documented vulnerabilities in assets using each of the protocols, standards, and buses. VAF framework is based on data that were collected from different data sources such as CVE, CAPEC, and CPE, and permits the study of the vulnerabilities of the thirty-three protocols, standards and buses analyzed.

$O_{1.B}$: *"To characterize both the severity and the impact of studied vulnerabilities, enabling a comparison of the behavior in both IT and OT environments".*

EM_{IT} and EM_{OT} metrics included in the VAF methodology have been calculated to analyze the impact of the vulnerabilities and compare them in IT and OT environments. In particular, the comparison between environmental metrics revealed that if an asset uses one or a combination of the thirty-two protocols, standards and buses analyzed, when suffering one of the 1363 vulnerabilities analyzed, the severity will be usually higher if the asset is located in an OT environment instead of in an IT environment.

$O_{1.C}$: "*To analyze the most used attack pattern and the more exploited weaknesses".*

An accurate integration between attack pattern, vulnerability, and weakness demonstrated that "flooding" is the most used attack pattern and "improper input validation" is a more

exploited weakness. In addition, it is also confirmed that the most common attack used in IoT wireless technologies such as 4G/LTE, LoRa, and NB-IoT is "jamming".

1.2. DISCUSSION OF C_2 CONTRIBUTION (CHAPTER III)

Based on the security recommendations established by the Modbus Organization, the **C_2 contribution** designs, implements, and evaluates a role-based access control (RBAC) system, to authenticate and authorize Modbus IIoT endpoints for performance-constrained industrial environments. Mutual authentication between endpoints is provided by TLS. The authorization process includes the endpoint authorization performed through entity roles (RBAC), which are included as an arbitrary extension in the X.509v3 certificates. The roles are validated from values stored in a secure external database. Comprehensive performance tests demonstrate the feasibility of the RBAC system in a centralized environment with performance constraints. In this sense, latency measurements when applying the RBAC policy over a secure channel, i.e., Modbus TLS, are compared, using as benchmarks latencies of both Modbus TCP transactions and, latencies of standard industrial processes.

Hence, the **C_2 contribution** addresses the objective $O_{2.A}$: "To *design, implement, and evaluate the feasibility of an ACS based on a centralized architecture as a single entity responsible for endpoint authorization management in a performance-constrained industrial environment*".

1.3. DISCUSSION OF C_3 CONTRIBUTION (CHAPTER IV)

The **C_3 contribution** focuses on the design, implementation, and evaluation of decentralized attribute-based access control (ABAC) system over an IIoT-RFID non-constrained healthcare scenario using the Ethereum blockchain as a verifiable data registry, while the access control policy is executed off-chain. The ABAC policy (i.e., authorization) is executed on a decentralized application (DApp), where each of the endpoints (i.e., RFID Readers) running the DApp have control of the keys according to the Ethereum (ETH) public network rules, which significantly decentralizes the identity. Proofs-of-concept were conducted over the ETH Ropsten test network, and they demonstrated the technical feasibility of the proposal with acceptable latency levels over the test network compared to the latency levels achieved when deploying local ETH nodes ("geth").

Hence, the **C_3 contribution** addresses the objective $O_{2.B}$: "*To design, implement and evaluate the feasibility of an ACS decentralized architecture based on Ethereum's public blockchain technology in a non-performance-constrained environment, paying attention to the transaction cost of the architecture*". Despite the technical feasibility of the proposal, the high cost of

transactions leads to exploring other alternatives where transaction costs might be reduced, in the research carried out within the C_4 contribution that is described next.

1.4. DISCUSSION OF C_4 CONTRIBUTION (CHAPTER V)

The **C_4 contribution** designs, implements, and evaluates an alarm collection system based on the ETH Event-Logs emission, which integrates both private data collection and alarm collection. It is explored as an alternative to the on-chain storage of **C_3 contribution**, to decrease transaction costs while keeping the same off-chain computation strategy. The results justify the use of events-log as the most efficient technology in terms of gas costs, as well as determine the capacity of our system to manage a high number of concurrent alarms. However, these results imply to maintain a private data collection (private data local management, i.e., centralized external storage to the blockchain), so that the smart contract is only used to emit a key-value pair as part of the events-log, enabling traceability by leaving a mark on the blockchain. Due to this, the proposal is not able to ensure real-time traceability due to the delays introduced by the ETH network but is feasible for non-performance-constrained environments.

Hence, the **C_4 contribution** complements **C_3 contribution** to fully address the objective $O_{2.B}$: *"Designing, implementing and evaluating the feasibility of an ACS decentralized architecture based on Ethereum's public blockchain technology in a non-performance-constrained environment, paying attention to the transaction cost of the architecture*".

1.5. DISCUSSION OF C_5 CONTRIBUTION (CHAPTER VI)

The **C_5 contribution** presents an on-chain Access Control System, designed on an RFID-IIoT environment that includes all phases of access control: Identification, Authentication, Authorization, Auditing and Accountability. The proposal is designed over a Hyperledger Fabric Blockchain (HFB) network to achieve high-performance levels. In this regard, the registration, authentication and attributed-based authorization (ABAC) phases are on-chain (i.e., blockchain is more than a Verifiable Data Registry). From the performance analysis of the HFB network, based on a methodological framework, we have demonstrated the feasibility of the Access Control System in a performance-constrained environment such as an engine assembly line. In addition, we have designed the registration phase of our Access Control System based on HFB private data collection, which promotes a novel reliable data privacy model applied to an Access Control System. Additionally, we have demonstrated the feasibility of using HFB's private data collection over a private data local management and

finally, we have selected the most suitable combination of network elements and resources for optimal deployment of the HFB network associated with our use case.

Hence, the **C_5 contribution** addresses the objective $O_{2.C}$: *"To design, implement, and evaluate the feasibility of a decentralized architecture based on permissioned blockchain enabling the blockchain as more than a verifiable data registry that performs an on-chain Access Control System in a performance-constrained industrial environment"*.

1.6. DISCUSSION OF C_6 CONTRIBUTION (CHAPTER VII)

The **C_6 contribution** designs, implements and evaluates an Access Control System based on Self-Sovereign Identity over Hyperledger Fabric Blockchain for a performance-constrained environment that, through a decentralized identity system that promotes, at the chaincode level, not only on-chain authentication and authorization but also advanced operations such as signature verification (i.e., blockchain is again more than a Verifiable Data Registry), representing an identity and access management system itself. The designed identity and access management system enables an entity with a decentralized identity (e.g., the Modbus server), the capability to provide access permissions to its data, ensuring that the information is accessed only by authorized Modbus clients. The designed system not only provides security to Modbus connections via mTLS but also ensures scalability in environments with more than one organization, over which, the achieved latency and throughput levels within a network involving up to 32 clients and one server are well below 100 ms, thus demonstrating the viability of the contribution while ensuring simplicity, compatibility and interoperability, essential requirements for Modbus protocol.

Hence, the **C_6 contribution** addresses the objective $O_{2.D}$: "*To design, implement and evaluate the feasibility of a decentralized architecture based on permissioned blockchain enabling the blockchain as more than a verifiable data registry performed on-chain identity and access management based on self-sovereign identity in performance-constrained industrial environments*".

2. FUTURE RESEARCH LINES

The contributions of this thesis have some limitations that could be overcome through future research, together with the new open issues that could also be investigated.

Some of these research topics for extending this work are identified next:

- Regarding the VAF framework, further development for automating data collection from OSINT data sources together with automating the VAF methodology

application could enable an updated status of IIoT-related vulnerabilities and related impact information.

- VAF framework could be used together with an automated inventory of assets to extract real and current vulnerabilities (and their environmental impact) of the assets, providing real-time alerts regarding assets' vulnerabilities.
- Regarding access control system proposals, further study needs to be carried out to analyze how these solutions can be integrated into legacy architectures in current manufacturing environments.
- Further research is needed in the field of decentralized management of IIoT devices' identity and access so that devices could be able to manage their identity without the need for intervention. In this line, some of the proposals could be further explored:
    - OAuth machine-to-machine scheme to ensure access to resources of devices could prevent the user intervention into Self-sovereign identity through the machine-to-machine (IIoT) lifecycle, achieving a fully machine-to-machine interaction.
    - Authorization mechanisms based on verifiable credentials, to support selective disclosure based on Zero-Knowledge Proof.
- Study of decentralized protocols and architectures that constitute an underlying layer enabling the exchange of information across several blockchains that provide scalability and flexibility on full decentralized architecture.

APPENDIX A

PUBLICATIONS IN JOURNALS AND CONFERENCES

This section analyzes the set of articles published in both journals and conferences during the Ph.D. Thesis, which are divided into articles that contribute directly to the Ph.D. Thesis and other complementary contributions in journals and conferences.

A.1. THESIS CONTRIBUTIONS

I. S. Figueroa-Lorenzo, J. Añorga, and S. Arrizabalaga, "A Survey of IIoT Protocols: A Measure of Vulnerability Risk Analysis Based on CVSS," ACM Computer Surveys, vol. 53, no. 2, p. 53, 2020, doi: doi.org/10.1145/3381038. [JCR. 7.990, Q1].

II. S. Figueroa-Lorenzo, J. Añorga, and S. Arrizabalaga, "A Role-Based Access Control Model in Modbus SCADA Systems. A Centralized Model Approach," Sensors, MDPI, vol. 19, no. 20, p. 4455, 2019, doi: doi.org/10.3390/s19204455. [JCR. 3.576, Q1].

III. S. Figueroa Lorenzo, J. Añorga, and S. Arrizabalaga, "An Attribute-Based Access Control Model in RFID Systems Based on Blockchain Decentralized Applications for Healthcare Environments," Computers, MDPI, vol. 8, no. 3, p. 19, 2019, doi: doi.org/10.3390/computers8030057. [SJR. 3.3, Q2].

IV. S. Figueroa-Lorenzo, Santiago; Goya, Jon; Añorga, Javier; Adin, Iñigo; Mendizabal, Jaizki; Arrizabalaga, "Alarm collector in Smart Train based on Ethereum blockchain events-log," IEEE Internet of Things Journal, vol. 08, no. 17, pp. 13306 – 13315, 2021, doi: doi.org/10.1109/JIOT.2021.3065631. [JCR. 9.471, Q1].

V. S. Figueroa-Lorenzo, J. Añorga, and S. Arrizabalaga, "Methodological performance analysis applied to a novel IIoT access control system based on permissioned blockchain," Information Processing & Management., vol. 58, no. 4, p. 102558, 2021, doi: doi.org/10.1016/j.ipm.2021.102558. [JCR. 6.222, Q1].

VI. S. Figueroa-Lorenzo, J. Añorga, and S. Arrizabalaga, "Modbus access control system based on SSI over Hyperledger Fabric Blockchain," Sensors, MDPI, vol. 21, no. 16, p. 5438, 2021, doi: doi.org/10.3390/s21165438. [JCR. 3.576, Q1].

A.2. OTHER CONTRIBUTIONS

VII. S. Figueroa, J. Añorga, S. Arrizabalaga, I. Irigoyen, and M. Monterde, "An Attribute-Based Access Control using Chaincode in RFID Systems," in 2019 10th IFIP International Conference on New Technologies, Mobility and Security (NTMS), 2019, pp. 1–5, doi: doi.org/10.1109/NTMS.2019.8763824. [CONFERENCE].

VIII. S. Figueroa, J. F. Carías, J. Añorga, S. Arrizabalaga, and J. Hernantes, "A RFID-based IoT Cybersecurity Lab in Telecommunications Engineering," in 2018 XIII

Technologies Applied to Electronics Teaching Conference (TAEE), 2018, pp. 1–8, doi: doi.org/10.1109/TAEE.2018.8475973. [CONFERENCE].

IX. S. Figueroa Lorenzo, J. Añorga Benito, P. García Cardarelli, J. Alberdi Garaia, and S. Arrizabalaga Juaristi, "A Comprehensive Review of RFID and Bluetooth Security: Practical Analysis," Technologies, 2019, vol. 7, no. 1, p. 15, doi: doi.org/10.3390/technologies7010015. [JCR. 0.73, Q2].

X. Y. S. Mingsheng, S. Figueroa-Lorenzo, J. Añorga, S. Arrizabalaga, Y. Sun, "IACAP: Internet-exposed Assets Cybersecurity Analysis Platform," International Journal of Interdisciplinary Telecommunications and Networking (IJITN), 2020, vol. 12, no. 4, p. 14, doi: doi.org/10.4018/IJITN.2020100109. [JCR. 0.13, Q4].

XI. S. Figueroa-Lorenzo, S. Arrizabalaga, J. Añorga, "Towards decentralized and scalable architectures for Access Control Systems in IIoT environments," VI Jornadas Nacionales de Investigación en Ciberseguridad (JNIC), 2021, vol. 34, no. 33, p. 145, doi: doi.org/10.18239/jornadas_2021.34.33. [CONFERENCE].

APPENDIX B

LIST OF ERRATA

This section contains the list of errata found in the six main contributions of the Thesis. Possible errata of the five additional contributions have not been included as part of this Thesis.

I. **Contribution C_2**: S. Figueroa-Lorenzo, J. Añorga, and S. Arrizabalaga, "A Role-Based Access Control Model in Modbus SCADA Systems. A Centralized Model Approach," Sensors, MDPI, vol. 19, no. 20, p. 4455, 2019, doi: doi.org/10.3390/s19204455.

   a. The boxplots have been created using only maximum and minimum values of latencies. Therefore, the information provided by, e.g., quartiles cannot be used. The boxplots are in **Figure 12** and **Figure 13** of the contribution. The **Contribution C_2** is placed in **Chapter III** of this Thesis.

II. **Contribution C_5**: S. Figueroa-Lorenzo, J. Añorga, and S. Arrizabalaga, "Methodological performance analysis applied to a novel IIoT access control system based on permissioned blockchain," Information Processing & Management., vol. 58, no. 4, p. 102558, 2021, doi: doi.org/10.1016/j.ipm.2021.102558.

   a. Bar chart groups have been incorrectly named (X-axis). The legend used in the Bar chart is incorrect (bellow the X-axis). The bar chart is in **Figure 13** of the contribution. The **Contribution C_5** is placed in **Chapter VI** of this Thesis. The corrections on the X-axis should be the bar chart groups should include Chaincode API on the left and private data local management vs. private data collection on the right; the legend (bellow the X-axis) should include Put State on the left and Put Private State on the right maintaining the current position in the figure.

III. **Contribution C_6**: S. Figueroa-Lorenzo, J. Añorga, and S. Arrizabalaga, "Modbus access control system based on SSI over Hyperledger Fabric Blockchain," Sensors, MDPI, vol. 21, no. 16, p. 5438, 2021, doi: doi.org/10.3390/s21165438.

   a. The boxplots have been created using only maximum and minimum values of latencies. Therefore, the information provided by, e.g., quartiles cannot be used. The boxplots are in **Figure 6**, **Figure 7**, **Figure 8**, **Figure 9** and **Figure 10** of the contribution. The **Contribution C_6** is placed in **Chapter VII** of this Thesis.