PHOENIX

Electrical Power System's Shield against complex incidents and extensive cyber and privacy attacks

Deliverable D2.2
Secure and Persistent Communications Layer (Ver. 0)

Authors	Maryam Pahlevan, Pasi Lassila, Pekka Nikander (AALTO, D2.2 responsible partner); Wafa Ben Jaballah, Cyrille Piatte, Nicolas Peiffer, Vincent Thouvenot (TSG); Nikolaus Wirtz, Hendrik Flamme, Chijioke Eze, Charukeshi Joglekar, Ömer Sen (RWTH); Jose Martinez, Angel Palomares (ATOS); Dimitrios Skias, Sofia Tsekeridou (INTRA); Luigi Briguglio, Elena Sartini (CEL), Tommaso Bragatto, Francesca Santori (ASM), Konstantina Fotiadou, Artemis Voulkidis (SYN), Tomas Damjan, Corrado de Santis (BTC)
Nature	Report
Dissemination	Public
Version	1.2
Status	Deliverable
Delivery Date (DoA)	31.8.2020
Actual Delivery Date	18.9.2020

Keywords	Differential privacy, benchmark template, PRESS framework, cyber threat information, STIX, TAXII, blockchain, cloud technology, resilience by design, self-healing methods
Abstract	This deliverable provides a brief overview of differential privacy techniques for Electrical Power and Energy System (EPES). Furthermore, this document presents the benchmark template for examining the compliance of the Secure Persistent Communication (SPC) layer to the PRESS framework during different stages of development. This template will be adopted by other PHOENIX

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 832989

	components in order to comply with governance policies and data privacy rules defined in the PRESS framework. To enable data communication within Large Scale Pilot (LSP) premises, it is essential for SPC layer to have common understanding of data models of information that is exchanged between PHOENIX components and LSP entities. To achieve that, this deliverable first identifies data formats of data exchanges within LSP infrastructure and then considers STIX and TAXII as suitable protocols for modelling and transferring cyber threat information by SPC layer. Additionally, this document investigates viable blockchains and cloud technologies deployment scenarios within LSP1 and LSP3 infrastructures in order to enhance data persistency, traceability, availability, integrity and interoperability. To this end, this deliverable specifies the security, privacy, persistency and real-time requirements of each data flow within LSP infrastructure. Further, this document according to data flows specification recognizes FATE Cloud and permissioned private blockchains like QUORUM and Hyperledger Fabric as suitable solutions for data storage in SPC layer. This deliverable also discusses the concept of resilience by design for EPES infrastructures. Besides, it refines the non-functional requirements of SPC layer which were initially introduced in D2.1. Finally, this document describes the simulation set-up and outcomes which is aimed for evaluation of the proposed self-healing methodologies (i.e. double virtualization and service restoration).

DISCLAIMER

	Participant organisation name	Short	Country
01	Capgemini Technology Services	CTS	France
02	THALES SIX GTS FRANCE SAS	TSG	France
03	THALES Research & Technology S.A.	TRT	France
04	SingularLogic S.A.	SiLO	Greece
05	DNV-GL AS	DNV	Norway
06	INTRASOFT International S.A.	INTRA	Luxemburg
07	Iskraemeco	ISKRA	Slovenia
08	Atos SPAIN SA [Terminated]	ATOS	Spain
09	ASM Terni	ASM	Italy
10	Studio Tecnico BFP srl	BFP	Italy
11	Emotion s.r.l.	EMOT	Italy
12	Elektro-Ljubljana	ELLJ	Slovenia
13	BTC	BTC	Slovenia
14	Public Power Corporation S.A.	PPC	Greece
15	E.ON Solutions Gmbh [Terminated]	EON	Germany
16	Delgaz Grid SA	DEGR	Romania
17	Transelectrica S.A.	TRANS	Romania
18	Teletrans S.A.	TELE	Romania
19	Centro Romania Energy	CRE	Romania
20	CyberEthics Lab	CEL	Italy
21	GridHound GmbH [Terminated]	GRD	Germany
22	Synelixis Solutions S.A.	SYN	Greece
23	ComSensus	CS	Slovenia
24	AALTO-KORKEAKOULUSAATIO	AALTO	Finland
25	Rheinisch-Westfälische Technische Hochschule Aachen	RWTH	Germany
26	Capgemini Consulting [Terminated]	CAP	France

27	ATOS IT Solutions and Services Iberia SL	ATOS IT	Spain
28	DNV GL NETHERLANDS B.V.	DNV-NL	Netherlands

ACKNOWLEDGEMENT

This document is a deliverable of PHOENIX project. This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement Nº 832989.
The opinions expressed in this document reflect only the author's view and in no way reflect the European Commission's opinions. The European Commission is not responsible for any use that may be made of the information it contains.

Document History

Version	Date	Contributor(s)	Description
V0.0	11.6.2020	Aalto, TSG, CEL, RWTH	Initial ToC
V0.2	22.7.2020	Aalto, CEL, RWTH, IntraSoft	first iteration of D2.2 preparation
V0.3	5.8.2020	Aalto, CEL, RWTH, TSG, ASM, EMOT, BTC, SYN, IntraSoft	second iteration of D2.2 preparation
V0.4	19.8.2020	Aalto, CEL, RWTH, TSG, BTC, IntraSoft	third iteration of D2.2 preparation
V0.5	8.9.2020	All above	new version after partner updates following from review comments
V0.6	11.9.2020	All above	minor updates based on 2nd review by TSG and RWTH
V1.0	14.9.2020	All above	Last comments before submission
V1.1	18.9.2020	All above	Final version
V1.2	30.06.2021	SYN	Updates to address midterm review recommendations

Document Reviewers

Date	Reviewer's name	Affiliation

28/08/2020	Cyrille Piatte, Wafa Ben Jaballah	Thales SIX GTS France
28/80/2020	Nikolaus Wirtz, Ömer Sen, Joglekar Charukeshi	RWTH

Table of Contents

List of Figures

List of Tables

Definitions, Acronyms and Abbreviations

ASR	Attack Surface Reasoning
BDP	Battery-based Differential Privacy
CA	Certificate Authority
CISA	Cybersecurity and Infrastructure Security Agency
CMS	Configuration Maintenance Service
CPS	Cyber Physical System
CTI	Cyber Threat Intelligence
CVE	Common Vulnerabilities and Exposure
DER	Distributed Energy Resource
DHS	Department of Homeland Security
DIRE	Disturbance and Impact Resilience Evaluation
DLT	Distributed Ledger Technologies
DP	Differential Privacy
DV	Double Virtualization
EPES	Electrical Power and Energy System
EV	Electric Vehicle
FATE	Federated AI Technology Enabler
FLISR	Fault Localization Isolation and Service Restoration
HILP	High Impact Low Probability
HSM	Hardware Security Module
IDS	Intrusion Detection System
IED	Intelligence Electronic Device
IMEC	Incident Mitigation Enforcement Countermeasures
IoT	Internet of Thing
MBO	Malicious Behaviour Ontology

MISP	Malware Information Sharing Platform
MPC	Multi-Party Computing
MTA	Message Transfer Agent
MVCC	Multiple-Version Concurrency Control
NILM	Nonintrusive Load Monitoring
OBD	On-Board Diagnostic
PBFT	Proof Bayzantine Fault Tolerance
PMU	Phasor Measurement Unit
PoB	Proof of Burn
PoC	Proof of Capacity
PoET	Proof of Elapsed Time
PoS	Proof of Stake
PoW	Proof of Work
RBAC	Rule-based Access Control
SAPC	Situation Awareness, Perception and Comprehension
SCADA	Supervisory Control and Data Acquisition
SCC	Security Control Center
SDO	STIX Domain Object
SM	Smart Meter
SPC	Secure Persistent Communication
SR	Semantic Representation
SRO	STIX Relationship Object
STIX	Structured Threat Information Expression
TAXII	Trusted Automated Exchange Intelligence Information
USG	Universal Security Gateway
VPS	Virtual Private Server

Executive Summary

This document provides theoretical studies on data privacy, sovereignty and semantics as well as blockchains and cloud technologies. Furthermore, it discusses viable Distributed Ledger Technology (DLT) and cloud deployment scenarios for Large Scale Pilot 1 (LSP1) and LSP3 which can be served as a baseline for other LSPs in PHOENIX project. This deliverable also presents the initial simulation results which are obtained from so called “proactive by design” solutions.

Chapter 1 of this document briefly describes the objectives of D2.2 deliverable. Additionally, it explains the structure of this document in more details.

Chapter2 is built by taking care of the PRESS Framework and its recommendations. It identifies techniques and mechanisms to be adopted during the Secure and Persistent Communications (SPC) layer development, and specifically explains different privacy defence techniques including differential privacy for Cyber Physical System (CPS). It also presents the benchmark template for examining the compliance of SPC layer to PRESS framework.

Chapter 3 provides a brief overview of the renowned data representations for cyber threat information such Structured Threat Information Expression (STIX) and associated transport protocols (e.g. TAXII).

Chapter 4 explores plausible communication paradigms among different entities within an LSP infrastructure. The next chapter explains a generic framework comprising resilience enhancement approaches and a self-healing methodology which aim to provide resiliency and fault-tolerance by design for different LSPs.

Chapter 5 focuses on different resilience enhancement methods and self-healing methodologies for Electrical Power and Energy Systems (EPES) infrastructures.

Chapter 6 focuses on the non-function requirements of SPC layer including reliability, availability and performance.

Chapter 7 describes a lab set-up for simulation studies and further evaluates the resilience improvements attained by implementing the generic framework introduced in chapter 5.

1. Introduction

The main goal of PHOENIX project is to protect European Electrical Power and Energy System (EPES) assets and networks against cyber-attacks. To this end, it enables coordinated cybersecurity and privacy measures, guarantees the zero-time recovery and minimizes the cascading effects of cyber-attacks. This deliverable which is conducted as part of Work Package 2 (WP2) efforts to develop SPC layer of PHOENIX platform, will refine and expand the requirements of SPC layer that were initially introduced in D2.1 document [1].

More precisely, this document provides theoretical studies on data sovereignty and various differential privacy approaches along with the description of PRESS framework which has been introduced in D4.1 [2]. These studies are done in efforts to complete task 2.3 “Data sovereignty and semantic interoperability” which aims to evaluate the applicability of differential privacy methods and anonymization approaches for preventing privacy breaches. Additionally, this deliverable explores the concept of resiliency by design through replication and double virtualisation logic within the scope of task 2.4 “Resilience, Survivability & Self-healing by design”. This document also presents the simulation studies on double virtualisation methods which offer fast self-healing services and further evaluates the simulation outcomes based on different metrics such as cascading effects of an incident and severity.

This deliverable investigates practical secure cloud and DLT deployment scenarios for example LSPs (i.e. LSP1 and LSP3) as part of task 2.5 “Federated cloud/inter-ledger transactions platform”. The potential use cases are proposed based on the requirements of data exchanges across LSP infrastructures.

The rest of the deliverable is structured as follows:

Chapter 2 extends and identifies mechanisms and procedures to be implemented during the development of the SPC, to guarantee the compliance with the PRESS Framework. Specifically, it provides a brief overview of different privacy defence techniques such as differential privacy for CPS particularly smart grids. These mechanisms are used to protect a system against various privacy attacks, including data mining and correlation attacks. This chapter also explains the benchmarks that can be used during different phases of development to assess SPC layer against PRESS Framework, which defines compliance rules and governance policies for data exchange within an LSP infrastructure.

Next, Chapter 3 discusses the importance of having common understanding of data models in SPC layer in order to enable data exchange across an LSP infrastructure. Further, it provides a brief background on well-known cyber threat information ontologies. More precisely, this chapter describes Structure Threat Information Expression (STIX) which is one of the most comprehensive and widely used ontology for Cyber Threat Intelligence (CTI) data in greater detail. Additionally, chapter 3 explains Trusted Automated Exchange of Intelligence Information (TAXII) as a set of protocols and data formats which is aimed for transfer of cyber threat information. It also briefly discusses Malware Information Sharing Platform (MISP) and Common Vulnerabilities and Exposures (CVE).

The following chapter studies the potential deployment use cases for blockchains and cloud technologies within LSP1 and LSP3 infrastructures. To this end, Chapter 4 first identifies all data exchanges incurred within LSP premises, then specifies the requirements of each data flow in terms of security, privacy,

persistency and real-time and eventually proposes certain distributed ledger and cloud solutions considering the data flows specifications.

Chapter 5 focuses on different resilience enhancement methods and self-healing methodologies for EPES infrastructures. To be more specific, this chapter lists planning-based and operation-based strategies for improving system resiliency against any disturbances. Further it describes how resilience enhancement strategies are applied in conjunction with a self-healing methodology in order to guarantee service restoration and recovery after failures to EPES infrastructures.

Chapter 6 refines further the non-functional requirements of SPC layer which were discussed in D2.1 document [1]. To this end, Chapter 6 lists all data exchanges taking place among different components across PHOENIX platform. Furthermore, it studies each data flow from different perspectives (e.g. data format, security and privacy) and accordingly introduces multiple blockchains solutions that can be utilized in SPC layer to fulfil PHOENIX platform´s requirements.

Chapter 7 details a laboratory set-up which is used for the evaluation of the self-healing methodology proposed in the previous chapter. This chapter also presents simulation outcomes which are further utilized to assess the resilience and improvements brought to the system. Finally, Chapter 8 concludes the deliverable.

2. Data Privacy and Sovereignty

2.1. Data Sovereignty and the PRESS Framework

European Digital Transformation[1] is going to make a relevant revolution, and this expects to lead to innovation and benefits for both EU citizens and EU companies in general. After having removed the borders impeding the free movements of persons/workers, goods, services, and capital within the EU (the so called "free movements"), to complement and foster the Internal Market the European Commission released the "European Data Strategy"[2], according to which a single market for data is created where,

- data can flow within the EU and across sectors, for the benefit of all;
- European regulations, in particular privacy and data protection laws, as well as competition law, are fully implemented and respected; and,
- rules to access and use of data are fair, practical and clear.

Against this backdrop, as the European Commission pointed out, one of the ideas behind the European Data Strategy is to promote the adoption of a regulatory system able to defend and promote EU values and rights already part of the so called "*acquis communautaire*", as well as data sovereignty.

In light of the above, and considering the objectives of the PHOENIX Project, a key element for the Project itself is to implement technological solutions entailing data exchange among EU Member States, especially when the data flow concerns personal and/or sensitive information passing through/coming from a critical infrastructure (as the smart grid is), already embedding the capacity to maintain a form of control over the data exchanged and complying with EU rules and governance policies in the field of cybersecurity and privacy and data protection.

Following from this assumption, the Project delivered the PRESS Framework (i.e. deliverable D4.1), which has been designed to identify the relevant data protection, privacy, ethics and security framework (i.e. applicable laws and regulations) allowing the PHOENIX Consortium to contribute in identifying those requirements (derived by the applicable regulatory framework) that contribute also to support and foster the concept of data sovereignty.

Indeed, the PRESS Framework has identified and provided:

- IT requirements with respect to privacy and data protection, ethics and social, security aspects;
- potential concerns and/or threats able to impact the abovementioned requirements; and,
- recommendations for addressing (or in the worst-case mitigating) the occurrence of the identified potential concerns and/or threats.

Moreover, having in mind the structure of the Project, the recommendations identified and described within deliverable 4.1 are intended to be considered as key drivers for each activity in the Project, but specifically during the implementation of the PHOENIX Platform and its components and services.

[1] https://ec.europa.eu/digital-single-market/en/policies/building-european-data-economy
[2] https://ec.europa.eu/info/strategy/priorities-2019-2024/europe-fit-digital-age/european-data-strategy

Indeed, the rationale behind deliverable 4.1 was to design a useful instrument that might be easily understood also by IT Partners during the early stage of the development phase, in order to deliver the PHOENIX components in the most compliant way.

In consideration to the fact that SPC layer specification is the first to be released after the availability of the PRESS Framework, Chapter 2.3 later on aims to describe how the PRESS Framework is instantiated since the beginning of the development of SPC layer, i.e. at the design phase, and the tools and techniques adopted by the PHOENIX Consortium to monitor and assess the compliance of SPC layer. This represents a model that will be adopted for the whole PHOENIX Platform as well.

2.2. Differential Privacy Techniques for CPS and Smart Grids

In this section we focus on Privacy defense mechanisms for Cyber Physical Systems (CPS), in particular for Smart Grids. In [3], the authors propose a nice survey for this topic. The basic objective of a Privacy attack consists in leaking some private data that are shared in the network or in inferring some critical information from public dataset.

There are several potential privacy attacks in CPS. An attacker can achieve a disclosure attack. This is a traffic pattern analysis attack where the attacker tries to recognize a set of receivers with the observed traffic. In linking attack, an attacker uses external data and private data to infer sensitive information. Direct queries on a particular individual is generally blocked to avoid evident privacy breach. However, it is possible to perform multiple queries to deduce more information on one targeted individual. It is called a differencing attack. Finally, correlation attacks consist in using the correlation between different private dataset to infer information.

A first approach used to protect Privacy is encryption, where some public and private keys are assigned to transmitted data that are used to decrypt data. The original data are not lost, and they become hard to access for unauthorized users. However, it introduces a very high computational overhead and so reduces the system speed. Typical deployed field-devices in EPES are considered legacy equipment with limited computational performance. Another approach proposed to protect Privacy is anonymization. Anonymization consists in erasing identifying and quasi-identifying attributes. Generalization is one anonymization family where we search to regroup the similar individuals to force that nobody is unique or in a small class of individuals. K-Anonymization [4], L-Diversity [5] and T-Closeness [6] are in this anonymization family. This approach limits the disclosure risk and work with high dimensional data. However, we lost the original data, it is very difficult to guarantee the total privacy and with these approaches, we achieve a trade-off between protection and data utility, which makes the approaches very use case dependent. Differential Privacy (DP) has been proposed by [7] and allows defining mathematically what Privacy is. A Privacy algorithm which respects this definition ensures that whether someone participates in the dataset, it will not increase its Privacy risk. For this approach, some noise, e.g. Laplace or Gaussian noise, are added to the data to achieve the Differential Privacy definition. The noise quantity to add is a parameter of the algorithms, which achieves a trade-off between the protection and the data utility. With the Differential Privacy, we ensure that new things (e.g. a new available dataset) will not affect the protection, which is a major advantage (see [5], [6]).

2.2.1. Differential Privacy and Local Differential Privacy definitions

Differential Privacy

Differential Privacy (see [7], [8]) propose a mathematical definition of Privacy. It states that for two adjacent dataset x and y, a random algorithm A achieve the (ε, τ) Differential Privacy if $P(A(x) = o) \leq \exp(\epsilon)\, P(A(y) = o) + \tau$, for all $o \in Range(A)$, where $Range(A)$ refers to the co-domain of A.

Local Differential Privacy

Differential Privacy ensures Privacy at a global level. With Local Differential Privacy, we ensure the privacy at the local level. It states that for all pairs of users x and y, a random algorithm A achieves the (ε) Differential Privacy if $P(A(x) = o) \leq \exp(\epsilon)\, P(A(y) = o)$, for all $o \in Range(A)$.

2.2.2. Differential Privacy for Cyber Physical Systems

CPS produce a large amount of sensitive and personal data. In [9], the authors express that Differential Privacy does not degrade system's speed as compared to others privacy approach because of the light-weight nature of the DP algorithms. Due to its mathematical fundament, DP achieves guarantee both in terms of privacy level and in data quality (see [10]). Moreover, DP preserves some data statistical properties, which can be always used by analyst for the studies (see [11]).

Because of the composition theorem, if the list of queries is large on CPS datasets, the amount of noise to add is important. However, we can deploy DP directly on Machine Learning algorithms. One approach consists in adding the differential privacy during the optimization process. In [12], the authors propose a version of stochastic gradient descent which respects Differential Privacy for general optimization problems. In [13], the authors propose a differentially private stochastic gradient descent algorithm dedicated to multiparty classification. The work in [14] proposes a differential private stochastic gradient descent algorithm used to train Machine Learning model basically networks of mobile devices or Internet of Things applications. In [15], the objective was to propose a decentralized DP stochastic gradient descent algorithm. In [16], [17] and [18], the authors develop the optimization techniques used in tensorflow-privacy, a spin-off of tensorflow module, which is dedicated to Deep Learning, and which allows to achieve DP when we train a Deep Learning model. Some other approaches consist in perturbing the cost function and the output (see [19]). In [20], the authors show that these Private Machine Learning models can fit with the need of listings, perturbing and query evaluation in large datasets from CPS. DP can provide both edge and node privacy in social media by protecting individuals information and any specific relationship information (e.g. [21], [22]).

DP has already prove its interest for location based systems (see [23], for mobile device ([24], [25]), for Smart Grids (e.g. [26] [27], [28], [29]), Internet of things (see [30], [31], [32]) or for transportation system (see [33]). Local Differential Privacy has been used for Internet of Vehicles (see [34]) and for Internet of Things (see [35]).

2.2.3. Differential Privacy for Smart Grids

Privacy need description

Although very useful for a better management of the electricity demand and for the optimization of the power grid, the use of Smart Grids introduces new Privacy and Security risks for electrical grid. Leakage of real-time individual load can lead to serious threat for Privacy, as illustrated by [36]. Nonintrusive load monitoring (NILM) consists in the interpretation of power load signatures with the intent to obtain information about the appliances causing the load. Before Smart Grids, potential privacy threat had

already been exhibited when considering individual load demand. In [37], they could distinguish certain power events in the load signature and assign them to individual appliances. For example, they reported when the dryer ran, or the toaster was switched on. This effectively revealed that the residents were at home and some of their habits. In [38], the authors show that the generic isolation of single appliance loads from an aggregated load is possible. In [39], the objective is to propose some approaches to monitor the load. Such studies are possible if we gather data with a fine temporal granularity. With Smart Grids, we potentially gather load consumption with very small granularity (e.g. 10 minutes or 30 minutes). In fact, three dimensions mainly impact Privacy in Smart Grids: the sampling frequency, the ability to link a household with a load curve and the exactness of the load gathered. However, these three dimensions are important for the future use of the data. In particular, we do not need the same information for billing, where we need exact load consumption but with a low sampling frequency (e.g. monthly sampling) or load monitoring, for which exact load consumption is not mandatory. We need to have access to data with high frequency sampling. We need for instance to have access to infra daily data to study some short consumption peak.

Grids demands response

One of the objectives of Smart Grids is facilitate the energy monitoring to improve the efficiency of the consumption, according to the network safety and the optimization of the total consumption. Demand side management covers all aspects of demand response according to customer needs, and is important to reduce operational cost, to avoid a negative technical impact on the grid, and to reduce CO2 emission, and so on. For this purpose, in [40], the authors propose a Differential Privacy approach to protect the individual load demand based on the addition of a Laplace noise on the individual load consumption.

Smart Buildings

Smart Buildings refers to building that can perform measuring, controlling, monitoring and optimizing without any external support. A classical use case for such topics is adjustment of water heaters.

One important feature of Smart Building is that they produce real-time environmental data from sensors to make effective predictions and calculations. In [41], the authors present an algorithm for differentially private stream processing. Their algorithm supports a variety of stream processing tasks (count, sliding windows, event monitoring) over multiple resolution of the stream. In [30] and [41] the authors work on some real-world example for smart buildings.

Smart homes are currently in development. In particular, a lot of devices are connected to Internet. The work in [42] shows that this connection introduces privacy threat. Authors propose a privacy preserving traffic obfuscation framework to protect data in this context. They leverage the smart community network of wirelessly connected smart homes and intentionally direct each smart home's traffic to another home gateway before entering the Internet. The design jointly considers the network energy consumption and the resource constraints in IoT devices, while achieving strong differential privacy guarantee, with an exponential differential mechanism, to avoid that adversaries link any traffic flow to a specific smart home.

Smart meter load monitoring

Load monitoring is a major interest of Smart Grids. A classical way to protect the personal data of individual, is to balance the load demand by an external battery (e.g. [43]). Authors in [44] propose a differentially private meter reading report mechanism which narrows down the domain of the noise

distribution parameter, in order to decrease the possibility of violating the battery limits. It also combines a multi-armed bandit algorithm to further reduce the cost as much as possible. In addition, they propose a switch mechanism to prevent the meter from reporting its reading when the battery limitations might be violated. The work in [45] proposes a battery based differential privacy-preserving (BDP) scheme. They present two cost-friendly differential privacy-preserving schemes by extending BDP scheme. In [46], the authors analyze the challenges caused by the use of differential privacy combined with protection with external battery and proposes a stateless privacy protection scheme by exploring a boundary-changeable distribution for noise which satisfies differential privacy. To fit with the challenges they exhibit, they formalize the definition of a relaxed differential privacy and propose a protection scheme that satisfies this relaxed definition.

Another approach consists in directly adding noise in raw data. In [47], the authors propose a mechanism to allow to an electricity supplier to periodically collect data from smart meters and derive aggregated statistics without learning anything about the activities of individual households. This differential private mechanism is based on the addition of a noise using Gamma distribution and encryption for the aggregation step. The authors in [29] analyze the effect of differential privacy on real smart metering data, especially with respect to balancing utility and privacy requirements. They propose a mechanism based on point-wise differential privacy with Laplacian noise. In [48], the authors propose the mechanism Di-PriDA for appliance-level peak-time load balancing control in the smart grid. Di-PriDA achieves differential privacy, which provided indistinguishable application power consumption data to protect against eavesdroppers. The objective in [49] is to propose a computationally efficient and information-theoretic privacy engineering framework for datasets arising in the Smart Grids environments that robustly accounts for multi-attribute correlations while preserving data privacy in a provably optimal fashion. It uses a greedy algorithm and Laplace noise, and are under Markov assumptions. For intelligent electrical grid, the authors in [50] propose to combine K-Means with Differential Privacy, which enhances the selection of the initial center points and the distance calculation method from other points to center point.

Context	Use case	Year	Reference	Description
Smart Grids	Load monitoring	2011	[47]	Noise using Gamma distribution and encryption used for aggregation
		2015	[46]	Proposed relaxed differential privacy strategy by adjusting noise distribution along with battery capacity
		2017	[44]	DP used in conjunction with battery and multi-armed bandit algorithm
		2017	[45]	DP used in conjunction with battery and multi-armed bandit algorithm
		2017	[29]	Point-wise DP with Laplacian noise
		2018	[50]	K-Means clustering combined with DP
		2019	[48]	Laplacian noise
		2019	[49]	DP using greedy algorithm and Markov assumption

	Smart Buildings	2017	[41]	Perturbing, grouping and smoothing based on DP applied over sensors streaming
		2018	[30]	Based on [41]
		2018	[42]	Exponential DP
	Grids demand response	2016	[40]	Laplace noise
Mobile device	Release spatial temporal density with phone usage series	2014	[25]	Laplace or Gaussian noise adding on time series decomposition coefficients
	Mobile Crowd Coverage Maximization	2018	[24]	Based on Geographic differential privacy
IoT	Monitoring of building using a diverse set of sensors	2018	[30]	Based on [41]
		2018	[32]	Survey paper
	Simulating an edge-based IoT application	2020	[31]	Combination of Blockchain and edge computing
	Human Activity Recognition, census records from Brazil and Mexico	2020	[35]	Local differential privacy for Federated Learning
IoV		2019	[34]	Survey paper
Transportation system	Protection of Floating Car Data stored and processed in central Traffic Data Centers	2013	[33]	Laplace noise

Table 1: Summary of the main differential privacy mechanisms with respective to different contexts

2.3. Updated Data questionnaire

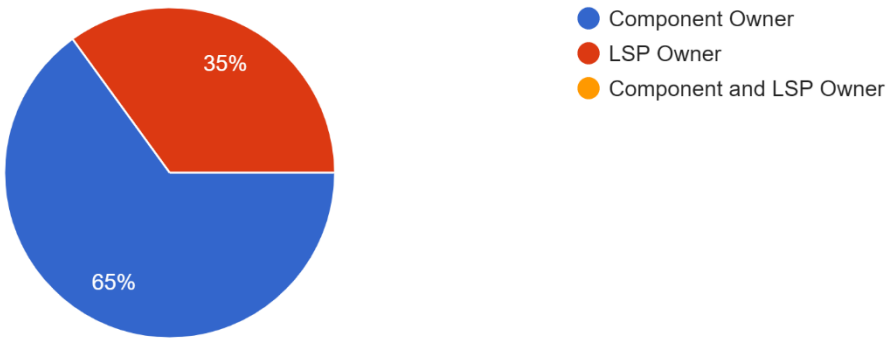
The identification of potential concerns impacting requirements defined in the PRESS Framework definitely requires a detailed knowledge of the dataset handled by the PHOENIX Platform, and for this reason, the Consortium uses the Project deliverable D7.1 (Data Management Plan) as a basis for creating the data knowledge base. Moreover, it has been decided to gather fresh and additional information on data handled by the component and LSP owners, and for this reason it has been submitted a new

questionnaire. This paragraph provides results of questionnaire and analyse them, with respect to previous information from D7.1.

Specifically, the updated "PHOENIX Data Management Plan Questionnaire" submitted in June 2020 to the PHOENIX Consortium focused on clarifying in 10 questions, with multiple choice answers to be selected, additional details available after the activities of WP1 and WP2 (i.e. "Identification of existing threats & data privacy requirements" released in D1.1, and "PHOENIX Platform Architecture Specification" released in D2.1). The submitted questions are:

- Q1 - Your role in PHOENIX is
- Q2a - What is the type of data you handle as output
- Q2b - What is the type of data you receive as input from other components
- Q3 - Is your data model defined?
- Q4 - Data model is/will be based on the following ontology
- Q5 - Data model includes personal data
- Q6 - Data model includes sensitive/classified data
- Q7 - Data format is/will be
- Q8 - Communication protocol
- Q9 - Data has specific time constraints for persistency, meaning how long data can be stored, e.g. is there a right for data to be forgotten? is there any obligation to keep them stored for a specific period?
- Q10 - Data samples are available/needed every

These questions have allowed to obtain additional details with respect to D7.1 in terms of ontology to be used, better understanding on concerns deriving from the handling of personal and/or sensitive/classified data, and potential communication protocol.



When dealing with data, PHOENIX is considering the two main players of Component Owners and LSP Owners, respectively representing the 65% and 35% of the participants to the questionnaire (see Figure 1).

Figure 1: PHOENIX roles from the Questionnaire

The questions Q2a and Q2b provide a wide agreement on two major types of data (see Figure 2), i.e. Cyber Threat Information (CTI) and Raw Data (i.e. all the raw data coming from Edge Network devices and SCADAs, such as metering data and data logs), respectively weighting for 45% and 40% of the answers. Such partners (5%) are considering to use both CTI and Raw Data. Federated ML models is considered as well (5%). Only a small representative group of the consortium is still working for defining the datatypes (5%).

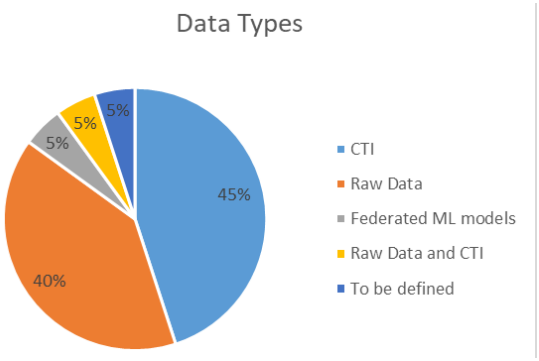


Figure 2: PHOENIX Data Types

This information will be sooner consolidated, due to the fact that only the 65% of the consortium has defined the data model for the components and the LSPs, as shown in Figure 3.

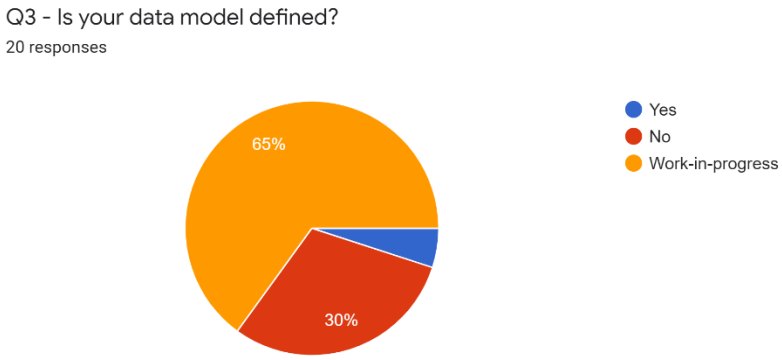


Figure 3: PHOENIX and Data Model definition

This type of questionnaire provides the consortium with additional tool for checking the status of development and driving its appropriate completion. Indeed, it encourage the partners to use the data modelling approach for defining the ongoing specifications (including the SPC layer specification of this document).

Due to the status of development, from question Q3, the rest of the questionnaire provides only a partial and still evolving information on the data management in PHOENIX.

For instance, the current most considered ontologies for the data model are STIX and OCPP (that provides both a protocol and the Device Model Data for Charger Stations), while other options (widely adopted standards) provided by the questionnaire have not been selected (i.e. W3C, WoT TD, W3C IoT-Lite, OMA OneM2M).

It is still interesting to highlight how the PHOENIX consortium has a less uncertainty to evaluate the presence of personal data (20%), see Figure 4, with respect to sensitive data (45%), see Figure 5.

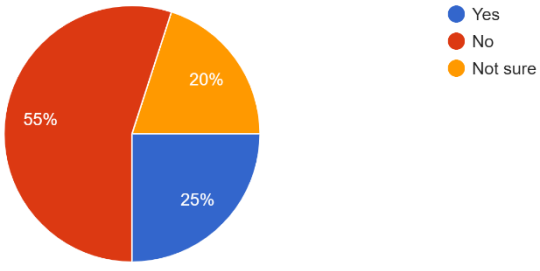


Figure 4: Data Model and personal data

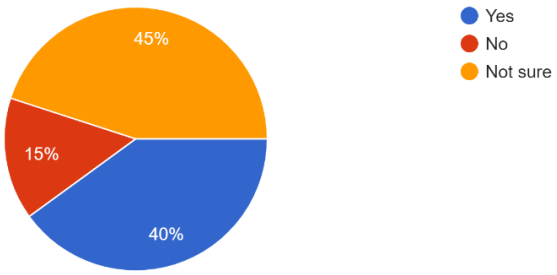


Figure 5: Data Model and sensitive data

The question Q7 on Data Format confirms the results reported in Data Management Plan (D7.1), i.e. relevance of JSON (50%) for the data format, as well as CSV (10%) and PCAP (5%).

Moreover, question Q8 on Communication Protocol shows that 45% of the partners are not yet defined this aspect, even if AMQP and MQTT look like the most promising solutions to be adopted.

Finally, the question Q9 remarks the need for further clarifications on the regulatory framework (i.e. storage keeping time), while question Q10 highlights a heterogeneous scenario with different data sampling rate (see Figure 6).

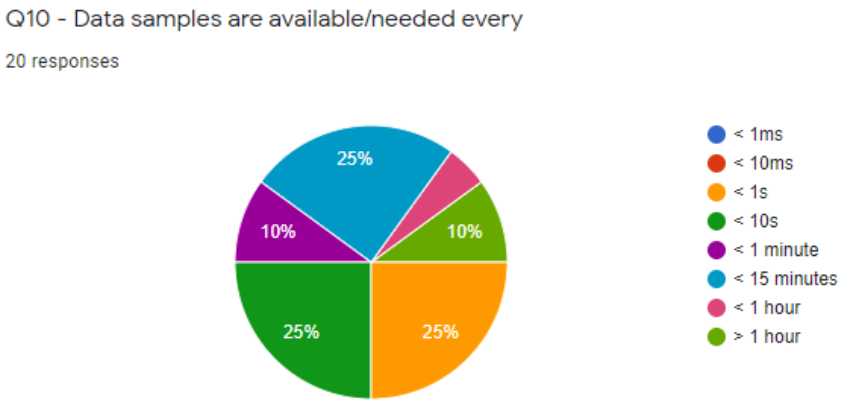


Figure 6: Data sampling rate

This analysis will be later used on the whole document for the specification of the SPC layer, as well as these results will be used for updating D7.2 and performing ethics monitoring.

2.4. PRESS-based Assessment of DevSecOps process

This paragraph provides the reader with a suggested template for test and validation report for the SPC components, in order to comply with PRESS Framework. SPC will be the first components to adopt PRESS, later all the PHOENIX components will adopt it as well.

The assessment process is based on the presence of different roles, i.e.:

- The component owner – the responsible of component development and delivery (based on specifications);
- The test operator – the responsible for component testing (based on specifications);
- The test validator – a third party from development team that repeats the tests together with the test operator for confirming the test report.

All these actors involved in the PRESS-based Assessment will use and adapt the following template for performing the specific checks, in order to satisfy both specifications and PRESS recommendations.

Acceptance constraints and rules, where not explicitly defined in this template (e.g. "Based on component specification."), will be defined during the component specification and its refinement. However, this checklist template is a tool for driving the specification of the components by taking care of the PRESS recommendations and the suggested mechanisms and techniques to address them.

PHOENIX PRESS TEST REPORT TEMPLATE		
Component		
Version		
Report Date		
Test Operator		
Test Validator		

#	Test Description	Results		Acceptance
1	Data Model is based on open/standard ontology	☐ Y	☐ N	Not mandatory, but open/standard ontology is preferred. Based on component specification
2	Specify the name of ontology			[STIX, OCPP, …]. Based on component specification.
3	Data Model requires personal data	☐ Y	☐ N	NO.
4	Personal data is strictly necessary to provide features and achieve objectives	☐ Y	☐ N	NO.
5	Real personal data is used for testing purpose	☐ Y	☐ N	NO
6	Data Subjects are informed about the processing of their personal data, and on the identity of the controller and/or processor	☐ Y	☐ N	Yes
7	Specify how the Data Subject is aware of the processing, and what is the lawful ground for the processing			(Consent, contract, etc). Based on component specification.
8	Data Subjects may always contact/send requests to data controllers/processors	☐ Y	☐ N	Yes
9	Data Subjects may always access their data	☐ Y	☐ N	Yes
10	Data Subjects may always exercise the right to rectification	☐ Y	☐ N	Yes
11	Data Subjects may always exercise the right to erasure	☐ Y	☐ N	Yes
12	Appropriate set of notifications have been developed depending on the addressee: for example, Data Subjects are always notified about the status of their requests and events involving on their data and incidents are notified to competent authorities	☐ Y	☐ N	Yes

13	Notifications include all the information necessary to ensure prompt response	☐ Y	☐ N	Based on component specification.
14	Data retention duration is fixed according to minimum required timeframe	☐ Y	☐ N	Yes
15	Appropriate levels of permissions have been developed to monitor operations on the data	☐ Y	☐ N	Yes
16	The traceability of events that occurs in the system is always ensured (regardless to the fact that the event involve personal data)	☐ Y	☐ N	Yes
17	Appropriate data access authentication roles have been developed	☐ Y	☐ N	Yes
18	Proportionate and adequate security measures have been developed to prevent intrusion or wrongdoing with data	☐ Y	☐ N	Yes

3. Semantics and Ontologies

It is essential for Secure Persistent Communication (SPC) layer to have common understanding of data models in order to exchange information among PHOENIX components and different LSP networks. To this end, first it is important to identify what types of information and data models exist in the system since each data model defines how the information is structured. Knowing data models, the SPC layer can process incoming information and potentially perform mapping between different data models. In other words, SPC layer enables interoperability between LSPs devices and PHOENIX components through defining common data representations.

It is noteworthy that in SPC layer, different data semantics are used for various purposes, i.e. internal and external data flows for LSP networks as well as internal communication between PHOENIX components. Therefore, it is essential for each component to specify the data representations that it is utilizing, otherwise other components in PHOENIX platform as well as different entities in LSP network are unable to process the data they will receive. For achieving this goal, the Semantic Representation module is defined in SPC layer and contains a wide range of data semantics and procedures for translation between data models. More specifically, the Semantic Representation module (SR) facilitates interoperability among different devices and entities inside the PHOENIX platform through defining common data models for information (e.g. cyber logs, anomalies report and SCADA metering information) which is exchanged across the PHOENIX platform. Each data model, in addition to data format, might also include other metadata such as data size and data frequency and can be used to parse data upon reception.

The essence of SR can be justified using LSP1 which comprises two separate segments. Each segment has its own SCADA server which is manufactured by different vendors. These vendors may implement different and even proprietary communication protocols. Therefore, data that is originated from these SCADA systems and sent to PHOENIX components could have different representation models. The SR translates these data semantics to the common data model that is understandable by PHOENIX components and enables these components to process versatile data.

It may be noted that the SR can comprise different data models and translation mechanisms for different LSPs since it is quite unlikely to define a common data representation which supports different types of data. The SR in SPC layer can be defined on basis of well-known ontologies for cyber security information such as Structure Threat Information Expression (STIX).

3.1. Background on ontologies, semantics and data formats

Ontologies in computer science are built upon semantics technologies which utilizes standardized data representations for determining the relationships among different concepts and subsequently enables computers to acquire knowledge from the represented information. More precisely, computers should be capable of interpreting data models.

According to Tom Gruber, «An ontology is an explicit specification of a conceptualization» [51] in which conceptualization is associated with abstract models independent from a certain language. Each ontology corresponds to a specific domain such information security and describes domain-specific elements, attributes and relationships. Thereby, ontologies allow sharing, consolidating and reusing

knowledge derived from different sources. To achieve this goal, the interpretation of ontologies must be unambiguous.

Cyber threat intelligence (CTI) which is considered part of cyber security, aims for early detection, prevention and mitigation of cyber-attacks through gathering, sharing and evaluation of threat information [52]. Since CTI is rather novel field and substantial percentage of data in this context are either unstructured or encoded in different formats, lately there have been remarkable efforts for modelling incidents, threat actors and attack patterns which lead to definition of several ontologies. These data representations are mostly use-case specific and facilitate sharing process of threat intelligence [53]. Incident Object Description Exchange Format (IODEF), Open Incident of Compromise (OpenIOC), Cyber Observable eXpression (CybOX), are examples of such ontologies [54].

IODEF is an IETF standard and mainly used to exchange incidents among Computer Emergency Response Teams (CERTs) and service providers. The IODEF data models are formatted in XML and thus can be easily transported across network. Additionally, these data models are generic and not tied to a specific implementation. Anti-Phishing Work Group (APWG), XMPP and Collective Intelligence Framework (CIF) are examples of communities that use IODEF standard. It is noteworthy that IODEF does not provide appropriate data models for packet sniffing, although it defines comprehensive data formats for incidents [55].

OpenIOC is another open platform for sharing CTI data among computers. Organizations utilizes this framework which was implemented by MANDIANT to disseminate the most recent Indicators of Compromise (IoCs) among themselves. This feature equipped organizations to the preventive defence against the latest security breaches. In OpenIOC, the pre-defined set of indicators are formatted using XML and can be easily extended for novel indicators. The key advantage of OpenIOC framework is to enable organizations to get a real-time access to the most recent IOCs across networks. Consequently, each organization has access to the threat intelligence which is built collaboratively [56].

The Cyber Observable eXpression (CybOX™) is a standardized framework which is used for modelling and sharing observables including dynamic events and stateful measures. Unlike many cyber intelligence ontologies, CybOX is not tailored for a particular cyber security scenario and supports a wide range of use cases from the operational instances of cyber observables to the potential patterns within cyber security realm. Therefore, CybOX facilitates automation of sharing, translating and assessment of cyber security events by defining standardized schemas. CybOX is mainly aimed for malware characterization, operational event management, logging, cyber situational awareness, incident response and threat assessment and characterization [57].

3.2. STIX Ontology

In cyber security systems, information sharing is of major importance. Information sharing involves either communication channels between components or between trusted parties. STIX ontology addresses the need for standardized, structured representations of this information to make it tractable. The Structured Threat Information eXpression (STIX™) is a community driven effort to represent structured threat information. STIX covers the full range of cyber threat information whilst at the same time maintains a human-readable format. In addition, STIX is actively adopted by a wide range of cyber threat related organizations.

STIX as a language that standardizes cyber threat information fits in a variety of high-level cyber security use cases that include:

- Analysing cyber threats
- Specifying indicator patterns for cyber threat
- Managing cyber threat response activities
- Sharing cyber threat information

In that sense, STIX provides a common mechanism for addressing structured cyber threat information across and among this full range of use cases improving consistency, efficiency, interoperability, and overall situational awareness.

At a high level the STIX language consists of 9 key constructs and the relationships between them:

- Observables describe what has been or might be seen in cyber.
- Indicators describe patterns for what might be seen and what they mean if they are.
- Incidents describe instances of specific adversary actions.
- Adversary Tactics, Techniques, and Procedures describe attack patterns, malware, exploits, kill chains, tools, infrastructure, victim targeting, and other methods used by the adversary.
- Exploit Targets describe vulnerabilities, weaknesses, or configurations that might be exploited.
- Courses of Action describe response actions that may be taken in response to an attack or as a preventative measure.
- Campaigns describe sets of incidents and/or TTPs with a shared intent.
- Threat Actors describe identification and/or characterization of the adversary.
- Reports collect related STIX content and give them shared context.

To enable such an aggregated solution to be practical for any single use case, STIX is both flexible and extensible. The latter, in the context of Phoenix project, promotes the adoption of STIX, as an effective structured cyber security related sharing language. Specific subsets of STIX capabilities can be defined; and agreed to beforehand in the form of profiles for use within sharing communities or by tools and components.

STIX 2.0 is based on the working foundation built by STIX 1.x, but with a few key distinctions. Most visibly

- STIX 2.0 uses JSON instead of XML. This matches common practice in development today.
- STIX 2.0 stresses simplicity and standardization. There are fewer options and more requirements. This makes it easier to implement, which is a requirement for broad industry adoption.
- STIX 2.0 is a graph-based model, where STIX Domain Objects, representing concepts in the cyber domain, are related to each other using STIX Relationship Objects.
- CybOX (Cyber Observable Expression) has been merged into the STIX specification. Now, STIX 2.0 is a multi-part specification with parts for STIX Core, STIX Objects, Cyber Observable Core, Cyber Observable Objects, and STIX Patterning.

STIX 2.0 Objects categorize each piece of information with specific attributes to be populated. Chaining multiple objects together through relationships allow for easy or complex representations of CTI. STIX defines 18 STIX Domain Objects (SDOs) which among others include Attack Pattern, Identity, Indicator, Infrastructure, Location and Intrusion Set.

Furthermore STIX 2.0, defines two STIX Relationship Objects (SROs) which are:

- **Relationship:** That is used to link together two SDOs or SCOs in order to describe how they are related to each other.
- **Sighting:** That denotes the belief that something in CTI (e.g., an indicator, malware, tool, threat actor, etc.) was seen.

Lastly, STIX 2.0 is transport-agnostic, i.e., the structures and serializations do not rely on any specific transport mechanism. A companion CTI specification, TAXII™, is designed specifically to transport STIX Objects. STIX provides a Bundle object[3] as a container for STIX Objects to allow for transportation of bulk STIX data, especially over non-TAXII communication mechanisms.

3.2.1. TAXII

Trusted Automated Exchange of Intelligence Information (TAXII™) v2.0, is a set of technical specifications and supporting documentation to enable sharing of actionable cyber threat information across organization and product/service boundaries. TAXII defines protocols and data formats for securely exchanging cyber threat information for the detection, prevention, and mitigation of cyber threats in real time.

TAXII is an application protocol for exchanging CTI (Cyber Threat Intelligence) over HTTPS. TAXII defines a RESTful API (a set of services and message exchanges) and a set of requirements for TAXII Clients and Servers. TAXII defines two primary services to support a variety of common sharing models:

- **Collection** - A Collection is an interface to a logical repository of CTI objects provided by a TAXII Server that allows a producer to host a set of CTI data that can be requested by consumers: TAXII Clients and Servers exchange information in a request-response model.
- **Channel** - Maintained by a TAXII Server, a Channel allows producers to push data to many consumers and consumers to receive data from many producers: TAXII Clients exchange information with other TAXII Clients in a publish-subscribe model. It is worth noting that TAXII v2.0 doesn't specify Channel services. Channels and their services will be defined in a later version of TAXII.

Collections and Channels can be organized in different ways. For example, they can be grouped to support the needs of a particular trust group.

[3] http://docs.oasis-open.org/cti/stix/v2.0/cs01/part1-stix-core/stix-v2.0-cs01-part1-stix-core.html#_rvtdrdkf1jdv

A TAXII server instance can support one or more API Roots. API Roots are logical groupings of TAXII Channels and Collections and can be thought of as instances of the TAXII API available at different URLs, where each API Root is the "root" URL of that particular instance of the TAXII API.

TAXII was specifically designed to support the exchange of CTI represented in STIX, and support for exchanging STIX 2.1 content is mandatory to implement. However, TAXII can also be used to share data in other formats. It is important to note that STIX and TAXII are independent standards: the structures and serializations of STIX do not rely on any specific transport mechanism, and TAXII can be used to transport non-STIX data.

TAXII design principles include minimizing operational changes needed for adoption; easy integration with existing sharing agreements, and support for all widely used threat sharing models: hub-and-spoke, peer-to-peer, source-subscriber.

3.3. OCPP Protocol

The Open Charge Point Protocol (OCPP) is an open protocol proposed by the Open Charge Alliance for communication between Electric vehicle (EV), charging stations and a central system across a charging station network. The primary goal of this protocol was to enable charging stations and central systems that are manufactured by different vendors to interact and exchange messages [58].

For better understanding, a central station can be assumed as a back-end web server while a charging station as an entity which is used for charging EV. To this end, a charging station first connects to a central system and further communicates relevant charge sessions and anomalies such as invalid charging profiles. A central management system holds the incoming information and responds with appropriate control commands if it is required. More accurately, a central system remotely supervises activities such as starting/stopping charge sessions, unlocking connectors and retrieving diagnostics reports.

However, there are three different drafts of OCPP, the OCPP 1.6 is used across charging station networks more than other versions of OCPP standard (i.e. OCPP 1.5 that is the older version and OCPP 2.0 which is a newer one).

OCPP 1.5 provides the initial version of charging protocol. This draft benefits from XML for encapsulating data and SOAP for transporting messages.

On the other hand, OCPP 1.6 uses JSON for formatting charging information and transports messages over WebSockets. The main difference between OCPP 1.6 and OCPP 1.5 is related to the newly added charging profiles and statuses. OCPP 1.6 defines six different profiles: core, firmware management, remote trigger, local auth list management, reservation and smart charging. The described profiles are implemented using a wide range of messages (e.g. Authorize.req, CancelReservation.conf) and data types (e.g. AuthorizationStatus, CancelReservationStatus).

Unlike slight differences between OCPP 1.5 and OCPP 1.6, OCPP 2.0 is a completely different protocol compared to older versions. More precisely, in OCPP 2.0, most of the message types are different and several operations are new. Additionally, OCPP 2.0 provides novel mechanisms for security enhancement, transaction handling, and smart charging [59].

3.4. Other Ontologies

3.4.1. MISP

Malware Information Sharing Platform - MISP, is an open-source threat information sharing platform, where users from various communities can share all kind of cyber-threats, indicators of compromise, and financial indicators among others. MISP is adopted by several organizations who run MISP instances. MISP allows organizations to share information about malware and their indicators. MISP users benefit from the collaborative knowledge about existing malware or threats. The aim of this trusted platform is to help improving the countermeasures used against targeted attacks and set-up preventive actions and detection.

Due to its peer-to-peer structure, multiple instances exchange information with each other, while its synchronization protocol is relied on three main criteria: efficiency, accuracy and scalability. The users can determine the granularity of the information they want to distribute in MISP, for instance with respect to the organization level, the community-level, or within their sharing groups. MISP is accessible from different interfaces like a web interface (for analysts or incident handlers) or via a REST API.

A shared piece of information in MISP is called an event. An event is composed of a list of attributes, including destination IP addresses and file hashes. An attribute is identified with the tuple: (i) category, (ii) type, (iii) value. Additionally, an event is linked with textual information, where it is available, such as date, threat level, description, organization and galaxies about threat actors, among others.

3.4.2. CVE

Common Vulnerabilities and Exposures (CVE) [60] is a dictionary that provides definitions for publicly disclosed cybersecurity vulnerabilities and exposures. The goal of CVE is to make it easier to share data across separate vulnerability capabilities (tools, databases, and services) with these definitions. CVE Entries are comprised of an identification number, a description, and at least one public reference. CVE does not contain information such as risk, impact, fix information, or detailed technical information. CVE only contains the standard identifier number with status indicator, a brief description, and references to related vulnerability reports and advisories. Through this standardized approach, CVE is designed to allow vulnerability databases and other capabilities to be linked together.

CVE although sponsored by the U.S Department of Homeland Security (DHS) and Cybersecurity and Infrastructure Security Agency (CISA), is copyrighted by MITRE thus it remains a free and open standard. Therefore, CVE is free to be used and publicly available to anyone interested in correlating data between different vulnerability or security tools, repositories, and services. This identifier that corresponds with a specific vulnerability provides for a quick and accurate investigation of the identified vulnerability on multiple information sources that are compatible with CVE schema. In the context of Phoenix, CVE provides a standardized identifier for a given vulnerability or exposure that can be communicated in an accurate manner towards Phoenix security authorities (e.g CIRTS, CERTS etc.)

3.4.3. Other Ontologies within the Cyber Security Domain

In the following, we provide an overview of the existing ontologies within the cyber security domain. In [61], the authors proposed an ontology for distributed anomaly based host Intrusion Detection System (IDS) sensors with the goal to contribute to a common knowledge-base and to identify quicker a possible attack. Following this proposal, in [62], the authors build a knowledge-base with reasoning capabilities

to take advantage of an extended various heterogeneous data sources. These data sources consider that data retrieved and included in the ontology is within the atomic indicator category of the CTI model.

In [63], the authors present an ontology to address the detection of modern complex malware families whose infections involve sets of multiple exploit methods. In order to achieve this, the authors designed a hierarchy of main behaviors. In particular, each behavior consists of a set of suspicious activities. Then they proposed an ontology that models the knowledge on malware behavior. The authors state that a program behaves maliciously if it presents one or more of the six events that are the attack launching, evasion, remote control, self-defense, stealing, and subversion. Hence, when a set of process actions with malicious behaviors appear, then the ontology can be inferred to see if an instance of suspicious execution is linked to a malware sample.

In [64], the authors describe their STUCCO ontology. This ontology includes data from 13 different structured data sources with different format. The data included in the STUCCO ontology fall into the categories of identity, tools, atomic indicators of the CTI model. In [65], the authors introduced the malicious behavior ontology (MBO). The MBO is able to detect modern complex malware families whose infections involve sets of multiple exploit methods, by applying Semantic Web Rule Language SWRL rules to the ontology for inferencing. The authors state that their model is able to detect known and unknown malicious programs by performing automatic inference of suspicious executions in monitored target systems.

In [66], the authors propose an ontology for quantifying attack surfaces. The Attack Surface Reasoning (ASR) explores the trade-offs between cost and security when deciding on composition of cyber defense. The created ontologies include those of attacks, defenses, metrics and systems. ASR is modeled after the Microsoft STRIDE threat classification framework, which categorizes attack steps into 6 categories and lacks details compared to CAPEC [67]. Following that, in [68], the authors propose a security metric ontology. Their ontology includes four sub-ontologies: vulnerability, attack, situations and defense mechanisms. In [69], the authors suggest the Unified Cybersecurity Ontology. This ontology serves to link cyber security and other relevant ontologies. There are mappings to aspects of STIX and references to CVE, CAPEC and STUCCO. Their model ontology was built by the same authors of STIX, which is currently the most used format for sharing threat intelligence.

4. Transaction and Communication Models

The SPC layer in PHOENIX platform offers trusted, traceable and persistent data communications through different components such as Universal Secure Gateway (USG), Interledger layer and 5G slice. Apart from communication security, which is provided by USG, SPC layer employs a a data centric approach based on federated cloud technologies and DLTs which result in enhancements of data persistency, traceability, availability, integrity and interoperability within EPES infrastructure.

This chapter is devoted to study the persistent aspects of SPC layer. To do that, first it provides a brief background on DLTs and federated cloud technologies and then discusses deployment of different data centric approaches in LSP1 and LSP2 considering specification of their data flows.

4.1. Technical background

Next we provide some technical background information no federated cloud technologies and DLT technologies.

4.1.1. Federated Cloud technologies

The understanding and definition of federated cloud or cloud federation depends on the context. For instance, [70] referred to the federated cloud as a system of systems comprising various cloud owners whose primary purpose is to provide flexible pricing options, maximize and distribute load utilization, and minimize energy use. There are several other definitions of the term with all pointing to the fact that to have a federated cloud, there is a need for collaboration between two or more cloud service providers. This could be a collaboration between cloud service providers within a single nation or could involve providers spanning several nations, integrating their resources including community, private, and public clouds to function as a single platform. Figure 7 depicts a hypothetical federated cloud consisting of various cloud providers ranging from Cloud A to Cloud F that are linked together via some connections. Essentially, each provider must have a link to one or several other providers in the federation through which its users can access services offered by the other providers when the need arises. The emergence of federated cloud has helped to solve some challenging information technology needs such as efficient resource (e.g. CPU) sharing and scalability issues. For instance, in the case of the energy systems applications, federated cloud platform can be established between various participants in the smart grid network, enabling smooth collaboration between the involved parties in terms of data and more efficient resource use. However, it also has its own drawbacks as outlined in the subsequent sections.

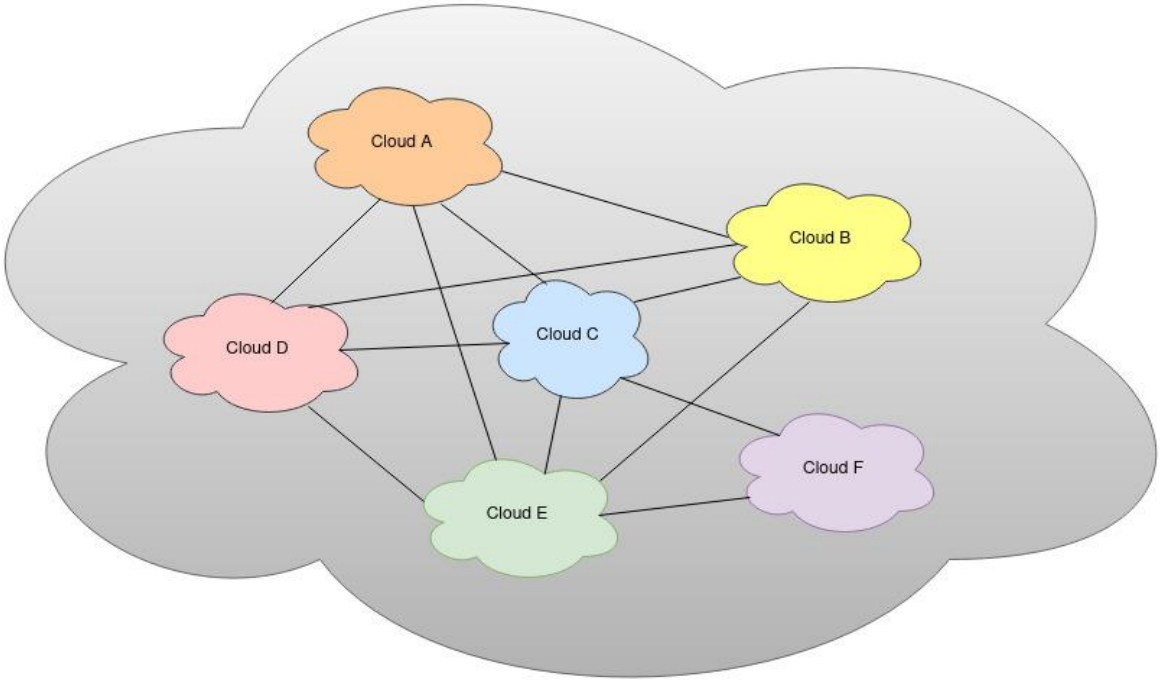


Figure 7: A Typical Federated Cloud

Driving factors for Cloud Federation

Several factors facilitated the emergence/need for federated cloud deployments. Herein, we present some of these factors (although not exhaustive).

- **Existence of several providers:** The proliferation of cloud service providers presents opportunity to users to make appropriate decision on the provider that suits their need. Reasons for the choice of provider could range from economic (lower prices) to issues such as security concerns.
- **Mix and match of services:** All business or personal needs may not be served effectively by a given cloud provider. In such a scenario, being able to seamlessly provision the services not covered by a primary provider becomes necessary. With federated cloud, such a need can easily be served without hitches.
- **Need to go local:** This essentially concerns security issues. In certain jurisdictions, the security policies in place may not be suitable for some clients. Hence, they may prefer their data to be stored within a particular jurisdiction instead of another one. For instance, EU regulation on user data is very comprehensive and may appeal to some particular clients of a cloud provider. In such a case, they may prefer to subscribe for storage resources hosted within the EU instead of ones located at other parts of the world.
- **Heterogeneity of platforms:** Presently, there is a plethora of cloud platforms located at various places, serving various categories of clients. Without collaboration, it will be almost impractical to make maximum use of resources offered by such platforms.
- **Scalability:** Federated cloud facilitates horizontal scaling of resources. By creating a virtual pool of resources, federated cloud enables automatic scaling of resources without loss of time. This is one of the greatest benefits of federated cloud. For instance, by using low cost devices, users can harness the power of several computing resources residing in various clouds to handle huge workloads. It should be noted

that based on agreement between the user and the respective provider(s), scaling can be done automatically even without explicit knowledge of the user.

Architecture of Federated Cloud

Although there are many federated cloud architectures in the literature, we will concentrate on the recent one proposed by [71] as it captures all the key components of a federated clouds. It is presented in Figure 8 below. As can be seen from the diagram, the main components of a federated cloud are: Federation Auditor, Federation Broker, Federation Carrier, Federation Manager, Federation Operator and Administrative Domain (which comprises the Cloud Service Provider (SP), Identity Provider (IdP) and Cloud Service Provider (CSC)). The Federation Auditor handles compliance related issues in the federation. The Federation Broker essentially connects the CSCs and the SPs for service discovery and provisioning. The Federation Carrier ensures connectivity and transport of services between SPs and CSCs. The Federation Manager and Federation Operator are the key management components of a cloud federation following this architecture. Finally, the IdP, SP and CSC form the Administrative Domain of the federation. Particularly, the IdP issues identity credentials to the CSCs while the SP validates consumer or user's credentials with the IdP before allowing the consumer access to requested service(s) or resource(s). Similarly, the CSC represents an individual or organisation using a particular service or resource in the federation. Refer to [71] for a more detailed explanation about this architecture.

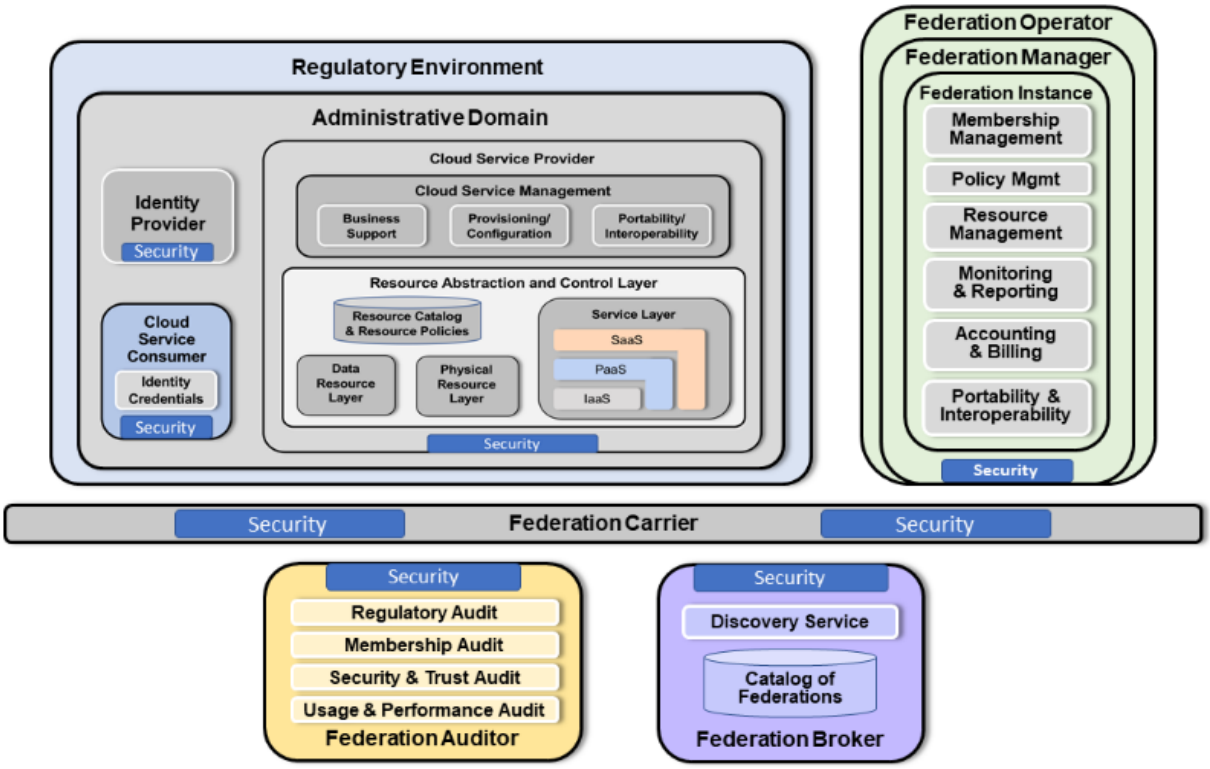


Figure 8: Reference Architecture of Federated Cloud [71]

Things to Consider When Implementing Phoenix Federated Cloud

As reported in [71], [72], [73], [74], many things need to be considered for an effective cloud federation implementation. Chief among these factors are:

- **Federation broker:** It is important to choose an appropriate broker when implementing the Phoenix federated cloud to ensure the frictionless exchange of data within the federation. The broker must manage how the component clouds in the federation will collaborate, discovery services, etc.
- **Identity Management:** It is necessary to select an appropriate technology for managing identity and access management (IAM) in the federation. OpenStack's Keystone can be used in this case. It has been proven to be effective in handling IAM in federated clouds.
- **Semantic Interoperability:** Because several components are going to exchange data within the federation, it becomes necessary to define common formats for data exchange in the federation. Using STIX and CTI as noted in Section 3 (Semantics and Ontologies) can realize this capability

4.1.2. DLT Technologies

This chapter is devoted to establish the base concepts related with blockchain development. In order to do so, it defines a brief introduction of the blockchain technologies defining the main concepts behind DLTs, a classification of the different types of blockchain technologies that can be applied and finally it defines the most used consensus algorithm available that can be used by the PHOENIX project.

Introduction

The use of decentralised systems or peer to peer systems is not new, Stuart Haber and W. Scott Stornetta coined the concept of a secured chain of blocks in 1991 but it was "Satoshi Nakamoto" [75] who provided a first implementation of the blockchain technology in 2008 when the blockchain concept started to be widely known. This implementation became the base of the broadly used cryptocurrency Bitcoin, popularizing the use of blockchain.

Blockchain is considered a series of time-stamped records or also known as information blocks, where each record is linked with the previous related block in the chain. In order to link a block with the previous record, cryptographic techniques are applied and, as result, the new block not only obtains a link to the chain but also a method to validate that the information linked is trusted and valid. Each time that a new record or block is incorporated in the chain, the system must verify using the cryptographic techniques that the new block is linked correctly but also the validity of the whole chain, auditing the content of the chain regularly. In addition, increasing the security and trustability of the data stored, these linked blocks or chain of blocks are stored in different redundant nodes, creating a decentralised network of nodes. How the different nodes of the decentralised network collaborate between them in order to manage the information stored is defined by a mechanism of consensus that is called consensus algorithm. As a result of applying all these different measurements and techniques the information stored has the following characteristics:

- **Immutable:** as a result of the cryptographic techniques, the system can verify mathematically that the content of the chain has never changed or altered.
- **Decentralized:** the chain of blocks is stored in each node of the network and therefore, in order to access any data of the chain, it can be performed accessing any of the nodes.

- **Traceability:** as each of the blocks are timestamped, the information of the whole system can be easily tracked and audited.
- **Consensus Driven:** as mentioned before, the collaboration between the different nodes and how the information is incorporated into the system is performed by a consensus mechanism in the decentralized network of nodes.
- **Transparent:** this feature is provided by different sides, from one side, all the information is stored in different nodes, and therefore accessible for any user with permissions. As the information is updated, a whole audit of the management of the data can be performed at any time. In addition, the cryptographic techniques assure the validity of the data store into the chain.

As a final consideration, the final product of applying this approach created a system that can be applied to several different domains and environments, not only for the management of cryptocurrencies, as it was originally designed for. From the point of view of the PHOENIX project, some of these characteristics suit to the stated requirements defined by the project developed by PHOENIX.

Blockchain Classification

As mentioned before a blockchain system is a combination of different techniques and protocols. Depending on the final implementation of the techniques or protocol applied will result in different types of blockchain systems, with the associated advantages and disadvantages. This chapter is devoted to define these possibilities detailing the pros and cons of each one, in order for the Consortium to be able to select the most suitable solutions for the project purposes. In this regard, this chapter uses two different criteria to establish the blockchain classification: the type of validation strategy and the type of end user access.

The *validation strategy* defines what nodes of the network can validate the information before it will be committed within the chain of blocks. In the case that any node will be able to validate and then commit the information within the ledger, it will be called **permissionless** strategy. However, if only a selected number of nodes can perform the validation of the information before it is included in the chain, we will be using a **permissioned** strategy.

On the other hand, the *end user access strategy* defines how the end users can interact with the ledger. In the case that the end users can access the information without any restriction, the strategy will be called **public**. But in the case that the system establishes a restricted access policy and therefore only the authorized end users can access the information, then the access will be denominated **private**.

The combination of these criteria will define different types of blockchains with different properties. For instance, a permissioned ledger will have more control over the information committed within the chain, as only a selected number of nodes can validate the operations and, therefore the consensus algorithm applied can be simpler and then faster than if we use a permissionless strategy. The following table summarizes the different possibilities of combining these two criteria.

Table 2 - Types of blockchain solutions

End-user access strategy	Public	Private

Validation strategy		
Permissionless	Public Permissionless (a.k.a. Public Blockchain)	Private Permissionless
Permissioned	Public Permissioned	Private Permissioned (a.k.a. Private Blockchain)

Public Permissionless: Anybody with access to the internet can incorporate a node to the network. Also, anyone can access the information contained within the ledger and can add information to it. Colloquially, it is also called public blockchain and therefore if somebody mentions a public ledger, without specifying the type of validation strategy chosen, he/she refers to a public permissionless ledger. Given the freedom to interact with this type of blockchain and the risks associated with that, for the ledger to remain secure, the actors involved must follow strict security and privacy measures and policies. Some examples of ledgers using these criteria are: Bitcoin, Ethereum, Litecoin. One of the main advantages of this type of ledger is that the cost of the infrastructure is lower, as new nodes from external actors can be incorporated to the decentralised network. However, the strict security and privacy policies and measurements necessary for this type of solution implies complex consensus algorithms and therefore sometimes the resulting systems need high processing requirements and they become slow.

Public Permissioned: In this case, the system allows all the end users with an internet connection to access the information contained in the ledger. However only a selected list of nodes can participate in the consensus mechanism and therefore only these selected nodes can validate the information and therefore commit information within the chain. As the validation process is limited to a controlled list of nodes, the consensus algorithm used on this type of blockchains can be less exhaustive and therefore faster. This type of solution used to be applied for those cases where the latency on writing in the blockchain needs to be low or where it is mandatory to control who can have write access to the ledger, for instance for identity management purposes. Some examples of ledger implementations following this model are: Ripple [76], private versions of Ethereum [77] and Hyperledger Indy [78]

Private Permissionless: These blockchain systems restrict the end users access to the information included within the ledger but allows external nodes to the system to perform the consensus mechanism to incorporate information to the chain. As main characteristics of this type of solution, a whole control of over the stored data is performed, so it can be applied when the access control over the data is critical. In addition, the consensus algorithm necessary for these blockchain systems must have a strict control over the validation of the information incorporate to the ledger. One example of solution following this specific combination is Exonum [79]

Private Permissioned: with this type of solutions both type of strategies restrict the access to the network. Thereby only the end users authorised can have access to the information contained with the system and only the restricted list of nodes can be part of the network. Similarly, to the previously mentioned public blockchain, this type of solution used to be named private blockchain colloquially. This type of solution provides a whole control over the data processed and therefore used to be selected for those cases where the control over the data used is critical (for instance using personal or sensitive data).

As the access control is performed beforehand and it is assumed that only the authorised actors can access the system, the consensus algorithm used in this type of solution can be easier, even straightforward, resulting in a system with a very low latency. Examples of blockchain implementations following this type of strategy are as follows: Rubix [80], Hyperledger Fabric [81].

Types of Consensus Algorithms

As described within the introduction subchapter, the network of nodes work collaborating between them. There are different alternatives to define the consensus mechanism and depending on the selection of a different consensus mechanism the final system has different properties. Therefore, the chosen consensus algorithm has a big impact on the final properties of the system implemented. This chapter is devoted to defining the broadest used consensus algorithms, detailing the advantages and disadvantages in order to select the most suitable solution for the project purposes.

The necessity of synchronization and to maintain a unique version of the information within distributed systems in not new. There is broad range of literature defining and suggesting a solution to the consensus problem. However, with the popularisation of the use of blockchain systems, new ad-hoc solutions have been defined to be specifically applied to the blockchain domain. Simplifying the problem, they are two main different types of consensus algorithm, the first one where it is a single node that carries on the validation of the information and finally take the decision to commit the information within the ledger (solution usually selected for the private blockchains) or when a group of nodes have to take the decision before a new data is incorporated to the chain, voting among all the nodes (normally used by public blockchains). Some of the possible solutions that can be applied are as follows:

Proof of Work (PoW) [82]**:**

This consensus algorithm defines a selected miner when a new block is incorporated to the system. Bitcoin is an example of an implementation using the PoW consensus algorithm. This solution is based on the resolution of a complex mathematical puzzle and as a result provide a solution to the consensus problem. A disadvantage associated with this algorithm is that the resolution of the mathematical puzzle has a high computational cost.

Practical Byzantine Fault Tolerance (PBFT) [83]**:**

Byzantine Fault Tolerance (BFT) main feature is that it proposes that the distributed network must reach a consensus, in other words the nodes must agree on the same value, in order to validate and finally incorporate the information into the chain. But this algorithm also contemplates the cases where some of the nodes fail to respond or even when they have a different result. This algorithm is derived from the classical computational problem Byzantine General's problem. Therefore, the main objective of this algorithm is about system failures, taking into consideration the result obtained from both correct and faulty nodes and then minimizing the impact of the faulty nodes.

Proof of Stake (PoS) [84]**:**

Proof of stake is the most common alternative to the original PoW described above. For instance, at the moment Ethereum is shifting from PoW to apply PoS consensus. This algorithm uses a completely different approach to the PoW, where the different validation nodes bet for what they estimate is the

best solution candidate, obtaining then proportional rewards to their bets once the solution is reached. As a result, it is not necessary to invest in expensive hardware to solve a complex puzzle like the PoW proposes.

Proof of Burn (PoB) [85]**:**

This algorithm takes a completely different approach. Instead of trying to find a solution for a mathematical puzzle, it is focused on discarding the non optimal results, "burning" them. In order to so, the validator nodes send the non-optimal records to an irretrievable address and by committing this change the nodes earns the privilege to mine on the system based on a random selection process. In other words, the main aim of the nodes is to burn non-optimal results in order to obtain more tickets to be selected to mine the next block. Therefore, with this method the algorithm encourages a long-term commitment on the integration of the nodes to the distributed network. Depending on the final implementation, the nodes can burn the native currency or an alternative ad-hoc one. Following this approach, the cost associated with expensive hardware devoted to compute complex mathematical problem is reduced drastically.

Proof of Capacity [86]**:**

In the Proof of Capacity consensus, also called Proof-of-space (PoC) or Proof-of-storage, the validators invest on hard drive resources instead of expensive hardware devoted for computational process. As a result, the more hard-drive space provided by a node, the more chances there are to be selected for mining the following block in the chain.

Proof of Elapsed Time [87]**:**

Proof of Elapsed-Time (PoET) is an algorithm designed to be applied on permissioned blockchains and therefore can't be applied for the other types. PoET is focus on obtaining a fair redistribution among all the nodes, hence the nodes waits a random amount of time to be selected and the winner is the node who can obtain a smaller time value in the proof part.

4.2. DLT deployment

Distributed Ledger Technologies (DLTs) provide a tamper-proof database where trust is by-product of the collaboration among a set of computers [88]. Thereby, PHOENIX platform can utilize DLTs in order to establish agreements and secure exchange data among different components. In DLTs, data records are immutable. This means once a block of data is written to a DLT, it cannot be altered because each data block contains cryptographic hash of its own content as well as the prior block. Consequently, if a block is modified for any reasons, its new hash would mismatch the successor data blocks and results in an invalid chain of blocks. Furthermore, within a DLT, several nodes retain the same copy of data records which leads to more resilience against failures. Due to aforementioned features, a DLT offers traceability, transparency and reliability as a data storage. Hence, DLTs are key enablers for the PHOENIX platform where the Cyber Threat Intelligence (CTI) data will be generated and exchanged among different components.

To explore potential deployment scenarios of DLTs across the PHOENIX platform (especially within the Secure and Persistent Communication layer (SPC)), two LSPs (i.e. LSP1 and LSP3) are considered in the following sections.

For each of mentioned LSPs, there is a permissioned DLT as a part of SPC layer which is used for storing CTI data generated by different PHOENIX components (e.g. USG and SAPC). Each of PHOENIX components can participate in the LSP-specific ledger network by including a full node. Subsequently, in LSP level, there is only one ledger type thus it does not need to communicate with other ledger technologies to fulfil its purpose. On the other hand, it is assumed that the I2SP component will be deployed outside the LSP premises and running its different modules on separate hardware. Additionally, I2SP owns one or more ledgers for recording CTI data. In order to detect attacks, select effective countermeasure and compute Machine Learning (ML) models, I2SP retrieves LSP-specific CTI data from Configuration and Maintenance Service (CMS) which is participating in LSP-specific ledger network. Therefore, this data transfer is occurred from CMS which runs a full node as part of LSP ledger network to I2SP ledgers via the Interledger layer. The integrity of communicated data can be verified through data records stored in LSP-specific DLT. On the other hand, I2SP can share the CTI data which is computed considering global view with CMS of each LSP. More specifically, for transparency and reliability purposes, I2SP shares the generated CTI data which is stored on its own DLTs with CMS via Interledger communication layer. The Interledger communication layer is defined as a part of SPC layer. The CMS runs a full node as part of LSP ledger network. Consequently, all other PHOENIX components which host a node in the LSP ledger network, will be received the copy of CTI data originated from I2SP.

To identify potential DLT and cloud deployment scenarios in above LSPs, the data communications within each LSP are specified using Table 3 . Furthermore, each data flow is studied from different perspective (such as security, persistence, format and real-time requirements) in order to facilitate selection of appropriate cloud and ledger technologies for purpose of data storage.

Component Name		<Name of Component>						
Data flows to / from the component (list all potential data communication between the component and any other components in LSP network or PHOENIX components)								
Data flow	Communication protocols	Ontology	Data format	Data persistency	Data Security	Data Privacy	Real time specs	Storage
		Data Model						local /DLT, cloud

Table 3 Template for data flow specification

4.2.1. LSP1 Scenario

PHOENIX LSP 1 comprises two segments: Segment A which will be implemented by ASM and EMOT and Segment B that will be developed by BFP. The main goal of LSP1 is to validate PHOENIX platform at DSO and prosumer level on a regional and cross-site information exchange at national level.

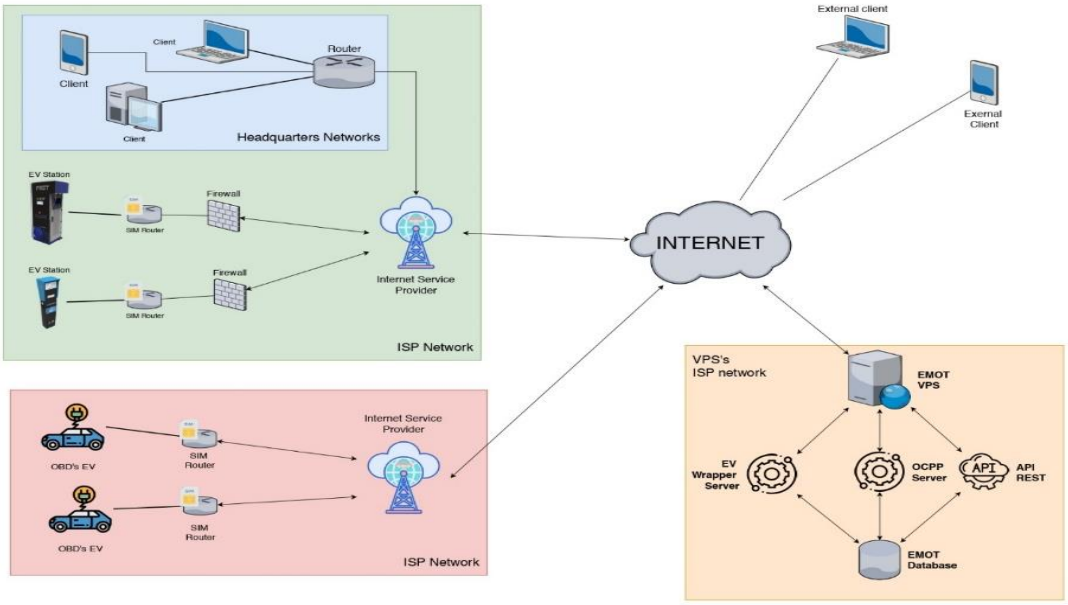


Figure 9 . Overall Structure of EMOT within LSP1 infrastructure [1]

Figure 9 depicts the overall structure of segment A where comprises three different networks: the first, top left, is the network to which the EMOT headquarters and charging stations are connected, the second, bottom left, is the network to which the electric vehicles are connected and the third, bottom right, is the OVH network where the EMOT Virtual Private Server (VPS) is hosted.

In order to benefit from PHOENIX platform services, the PHOENIX components might be deployed at the "Headquarters Networks". For LSP1, it is assumed a Universal Security Gateway (USG) is deployed very close to each charging station in order to acts as a gateway as well as Intrusion Detection System (IDS). The USG apart from setting up secure connections among the electric vehicle, the EV Station and the OCPP server, can detect abnormal messages originated from these entities.

Tables below list the data flows associated to segment A of LSP1.

Component name				Power quality analyser				
1. Power Quality analyser -> smart meter extension								
Data flow	Communication protocols	Ontology	Data format	Data persistency	Data Security	Data Privacy	Real time specs	Storage

1	http	Ontology are not defined	JSON	5 year	Power quality analysers are connected to the DMZ LAN of the company	Correspondence between meter ID and client personal data are managed offline according to GDPR		Stored locally in SMX

Table 4 Data flows associated with power quality analyser within ASM infrastructure

Component name	Smart Meter Extension (SME)
1. smart meter extension -> central broker	

Data flow	Communication protocols	Ontology	Data format	Data persistency	Data Security	Data Privacy	Real time specs	Storage
1	MQTT	Ontology are not implemented	JSON	5 years	User/password for MQTT subscription	mapping between meter ID and client personal data are managed offline according to GDPR		Local server (mongodb instance)

Table 5 Data flows associated with smart meter within ASM infrastructure

According to Table 4 and Table 5, there are two main data flows in ASM infrastructure: 1) Data which is generated by the power analyser in JSON format and further sent to Smart Meter Extension (SME) every 5 seconds. This data transfer is performed through ASM´s DMZ network thus it is protected from outside world. Besides, the collected data in smart meter is stored locally in the SMX. 2) Data which is collected by SME and sent to the central broker every 5 seconds. Like the former data flow, data is formatted in JSON and can only be accessed through authenticated MQTT subscription.

Figure 10 depicts the overall architecture of ASM infrastructure as well as the corresponding data communications among different entities within ASM premises.

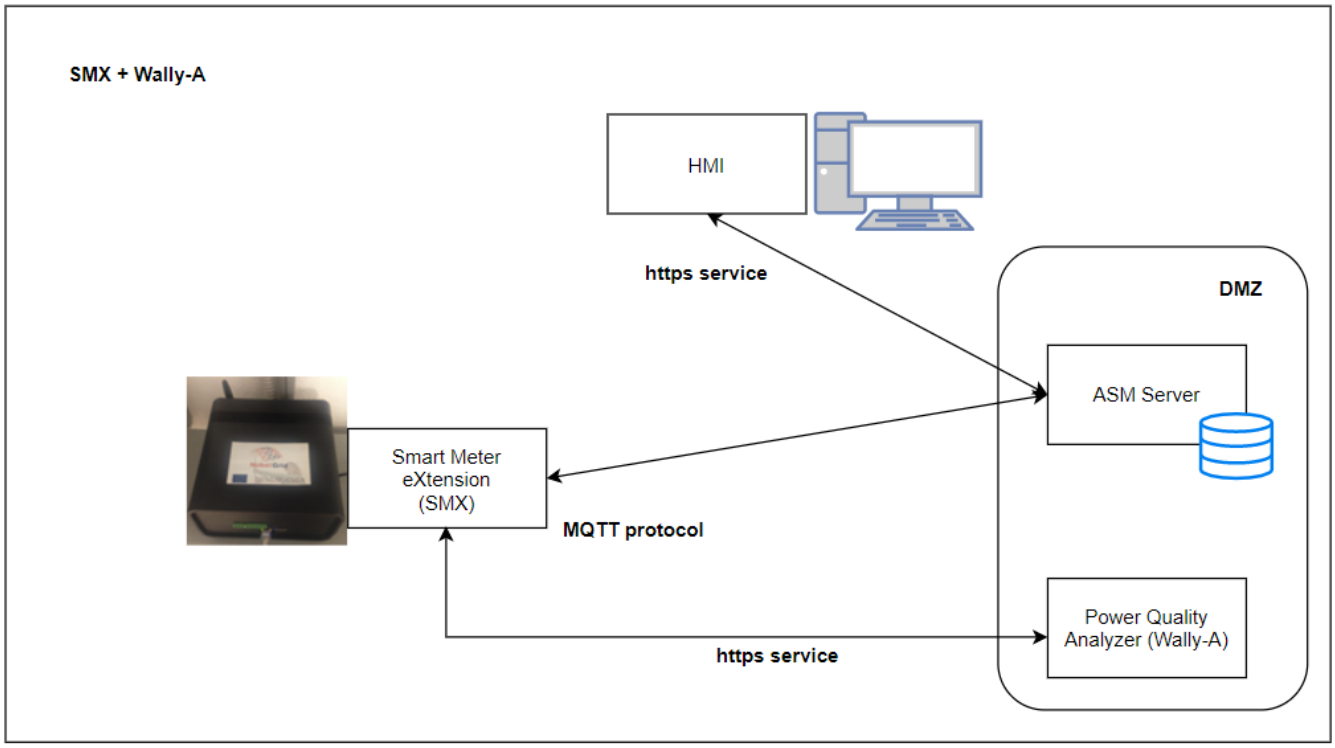


Figure 10 Overall architecture of ASM infrastructure and associated data exchanges

Table 6 and Table 7 list data flows that belong to EMOT infrastructure which comprises three main entities: charging stations, EMOT Virtual Private Server (VPS) and Electric Vehicles (EV). As stated in tables below, EMOT charging stations and electric vehicles send data periodically to EMOT VPS whose details are:

- CPU: 2 cores, clock speed 3.1 GHz;
- HDD: 50 GB;
- RAM: 4 GB;
- O.S.: Ubuntu 16.04 LTS.

EMOT VPS runs the EV Wrapper Server, OCPP server and API REST.

EMOT charging stations exchange data through a Teltonika RUT230 modem connected to a single-board computer, a Raspberry Pi 3, with a CPU of quad-core ARM Cortex A53 1.2 GHz, a SD of 16 GB, a RAM of 1 GB and a Raspbian Stretch 4.14O.S.; charging station protocols are OCPP (application protocol for communication between charging stations and EMOT central management system) and websocket (computer communications protocol, providing full-duplex communication channels over a single TCP connection). EMOT OCPP server accepts communications and data exchange only with the client program that is installed in the charging station computer. The client has an authentication key that is assigned to it by the EMOT management software when creating a new charging station (when a new record in the charging station table, in EMOT database, is created). The OCPP server accepts the

connection by the client only and exclusively if a valid authentication key is used at the time of the request. Charging station data format is JSON and the sampling rate is one second.

Regarding EV monitoring, EMOT use an on-board diagnostic (OBD) device to retrieve data from the EV; OBD is a IoT component, based on a Raspberry Pi 3 and Carberry; Carberry represents the link between car electronics and Raspberry Pi, which allows the development of end-user applications, such as media centers, vehicle diagnostics, data logging, fleet management, tracking, blackboxes, burglar alarms, carputing, internet, and much more. OBD communicate to a server using TCP/IP. The network connectivity of the OBD device is via data SIM (UMTS), thanks to a Raspberry module that works as a modem, and the server is a python software; OBD protocol is MQTT and the sampling rate is 5 seconds. The OBD connects to the diagnostic interface from which it can extract the information from the electric vehicle control unit using the CAN-bus protocol. The output data format of the OBD is an ASCII string; when the data is sent to the server, it is reorganized into a wrapper, thus obtaining a grouping of the data in JSON format.

It is noteworthy that both described data samples are stored on a cloud platform which belongs to EMOT.

Component name				Charging station				
1. charging stations -> EMOT VPS								
Data flow	Communication protocols	Ontology	Data format	Data persistency	Data Security	Data Privacy	Real time specs	Storage
1	OCPP 1.6	Not defined yet	JSON	5 years	Encrypted by websocket	Secured by user name and password	1 s	Cloud on EMOT VPS server

Table 6 Data flows associated with charging station in EMOT network

Component name				Electric vehicles				
1. electric vehicles -> EMOT VPS								
Data flow	Communication protocols	Ontology	Data format	Data persistency	Data Security	Data Privacy	Real time specs	Storage
	MQTT	Not defined yet	JSON	5 years	Not encrypted	Secured by user name and password	5 s	Cloud on EMOT VPS server

Table 7 Data flows associated with electric vehicles in EMOT network

In segment B of LSP1 which is associated to BFP infrastructure, there are two key data flows that are presented in Table 8. The BFP wind farm has two SCADA systems which are completely separated and independent.

As stated in Table 8, the first SCADA system manages wind turbines performance. To this end, for each wind turbine generator, it provides information in real time and records data on its own server for post analysis and for downloading data. To access the stored data, a user must first set a VPN connection in order to reach the website platform and then username and password to get in. The second one manages the whole wind farm performance; it provides information about energy exchange with the grid, voltage values, and alarms. A user needs username and password to get access to the collected data.

Component name				Wind Farm				
1. Data flows between the wind turbines and their SCADA system 2. Data flows between the electrical equipment the substation SCADA system								
Data flow	Communication protocols	Ontology	Data format	Data persistency	Data Security	Data Privacy	Real time specs	Storage
1	C30(Gamesa's protocol)	Not defined yet	CSV	8 years	VPN connection	secured by credentials	WGT performance in real time	local SCADA system
2	IEC 61850	Not defined yet	CSV	8 years	not encrypted	secured by credentials	wind farm performance in real time	local SCADA system

Table 8 Data flows associated with wind farm in BFP infrastructure

4.2.2. LSP3 Scenario

PHOENIX LSP3 is conducted as a joint effort by ELLJ and BTC and is aimed to demonstrate cyber threats mitigation and data privacy management in a decentralised EPES environment combining DSO, microgrid and Renewable Energy Source (RES) resources. It is important to expose, that in case of LSP3 we actually talk about two separate systems, one system considers the BTC company and the other one the distribution company of Elektro Ljubljana, the local DSO with its SCADA. The aim of this pilot is to connect these two systems for the future purpose of demand response. The BTC as a grid user shall act as a flexible load. This new role of the grid user shall result in helping the grid when the grid- the DSO would need this flexibility. This planned task is to be developed and it is challenging, because DSO's SCADA is because of its high security reasons currently a totally closed system and not interacting with any external systems. With doing this pilot, the main issue will be on how to connect two systems and in parallel to respect all necessary cyber security aspects and roles.

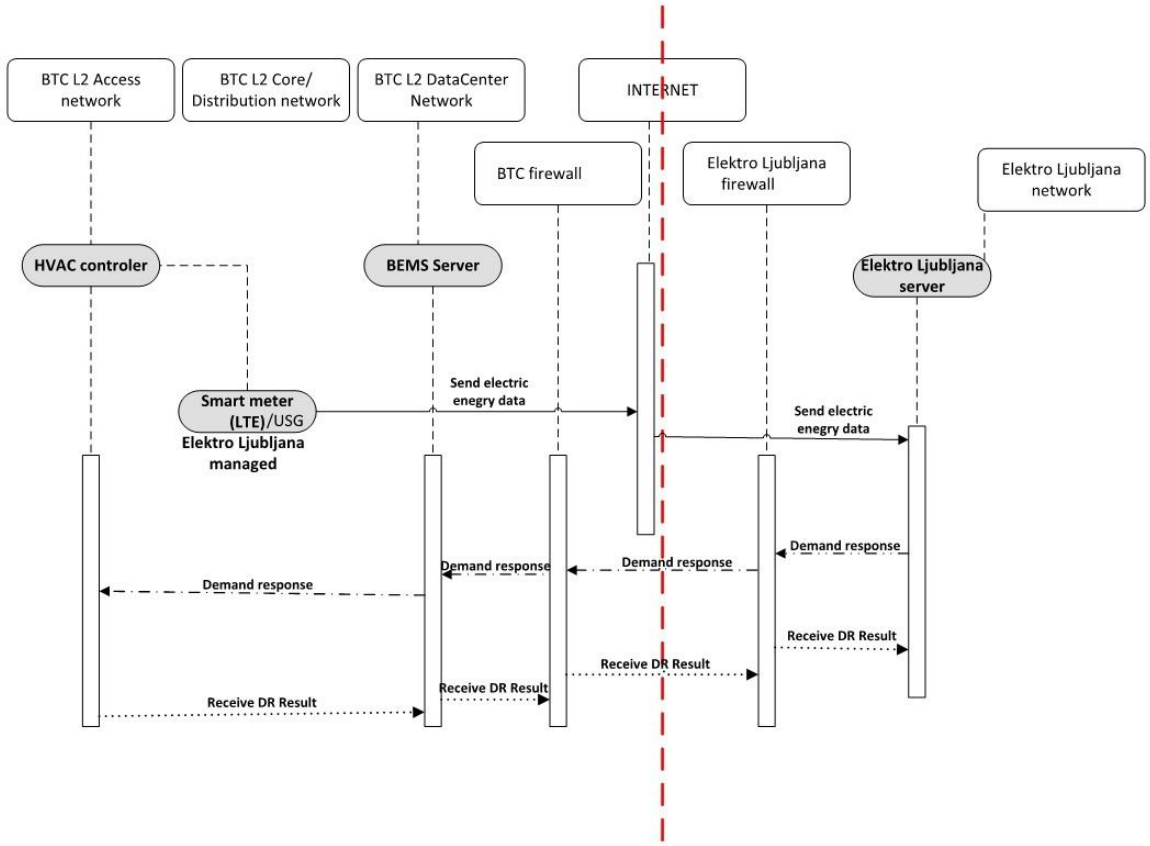


Figure 11 Sequence diagram of BTC infrastructure

As shown in Figure 11, the HVAC and the chiller controllers in LSP3 send measurement data (such as temperature and air quality) to the BTC SCADA server and in response they receive the control commands from the server in order to adjust the air quality and temperature of BTC accordingly. It may be noted the measurement data is exchanged once the value of the observed point changes at least for period of 30 second. The end-to-end communication latency between two devices which locate inside BTC LAN network is estimated between 2 to 3 milliseconds.

In LSP3, the authorized end-user (e.g. technical maintenance staff) is also capable of sending control commands directly to the SCADA server. As a following step, the server examines the validity of the user commands and if they are legitimate, the commands will be forwarded to either the HVAC controller or the chiller controller in order to take effect.

Table 9, Table 10 and Table 11 present the specification of each of described data flows separately. The metering data regardless of its origin (i.e. either HVAC controller or chiller controller) is formatted in either CSV or XML and is encrypted using 56-bit DES encryption mechanism. Additionally, the measurement data can be only accessed by entities who own SCADA credentials and will be stored locally on BMS server.

The user commands are formatted in XML and in a similar way to metering data, can only be accessed by entities who own SCADA credentials and are stored locally on BMS server. These commands are exchanged in a plain format and without any encryptions.

Component name	HVAC controller
1. HVAC controller to BMS (SCADA) server and back	

Data flow	Communication protocols	Ontology	Data format	Data persistency	Data Security	Data Privacy	Real time specs	Storage
1	BAC net/IP EN-ISO 16484-5		CSV, XML	1 year	56-bit DES	Secured by SCADA credential	2-3 ms	Local on BMS server

Table 9 Data flows associated with HVAC controller within BTC network

Component name	Chiller controller
1. Chiller controller to BMS (SCADA) server and back	

Data flow	Communication protocols	Ontology	Data format	Data persistency	Data Security	Data Privacy	Real time specs	Storage
1	Modbus TCP/IP		CSV, XML	1 year	56-bit DES	Secured by SCADA credential	2-3 ms	Local on BMS server

Table 10 Data flows associated with chiller controller within BTC network

Component name	BTC End user
1. End user to BMS (SCADA) server and back (user request command to SCADA, response is a confirmation of command)	

Data flow	Communication protocols	Ontology	Data format	Data persistency	Data Security	Data Privacy	Real time specs	Storage
	TCP/IP (HTTP)	Not defined yet	XML	1 year	Not encrypted	Secured by SCADA credentials	2-3 ms	Local on BMS server

Table 11 Data flows associated with end user within BTC network

4.3. Cloud deployment

Section 3 (Semantics and Ontologies) has already addressed the need to have common data models for effective exchange of information among various PHOENIX components and the LSPs. Additionally, a federated cloud enabling such interactions needs to be deployed using appropriate technologies in such a way that prevents unauthorized access to data or resources. LSP1 and LSP3 data flow discussed hereunder shows that any federated cloud approach to be used needs to ensure full control of data and other resources by the respective owners. There are many federated cloud deployment approaches

currently, however, we would propose an approach like the Federated AI Technology Enabler (FATE) Cloud [89]. FATE [90] is an open-source project initiated by Webank's AI Department to provide a secure computing framework for performing federated AI tasks. It implements secure computation protocols based on homomorphic encryption and multi-party computation (MPC).

FATE Cloud is an adaptation of FATE to provide an infrastructure for building and managing federated data collaboration network. FATE Cloud enables FATE to be managed in a multi-cloud environment, forming a secure federated data network, designed to provide secure and ensure compliant data cooperation solutions across or within organizations.

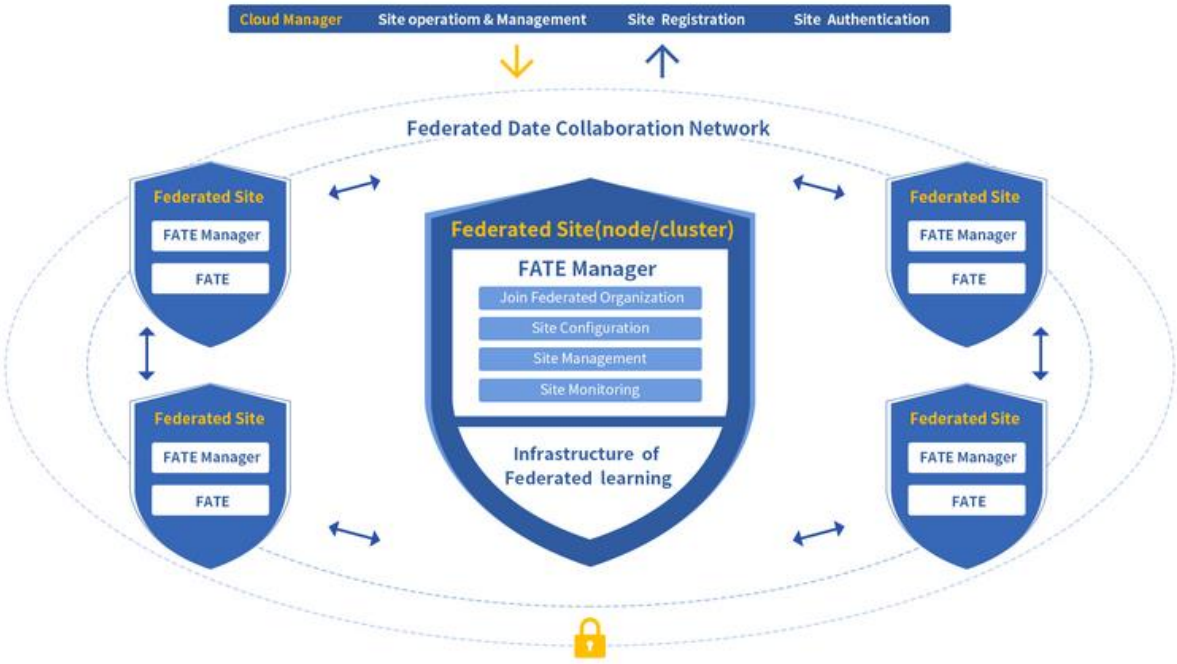


Figure 12 FATE Cloud

FATE Cloud provides standard federated infrastructure implementation capabilities, technical support capabilities, a unified technical framework for building federated data networks, and effectively handles distributed data processing and data authentication issues. As shown in Figure 12 above, FATE-Cloud comprises two types of roles: a neutral Federated Cloud (Cloud Manager) and local Federated Sites (FATE Manager). A brief explanation of these roles is presented hereunder.

- **Cloud Manager**

This is the management center of the federated network. It builds the entire federated network and performs site-wide operation and management, and provides site registration, authentication, cooperation management, etc. To achieve the security needs of SPC, the roles performed by this component can be realized by implementing smart contract to handle these tasks in the DLT layer.

- **Federated Site**

A Federated Site is an entity, institution, or organization that participates in the federation. Herein, any of the LSPs can be regarded as a federated site. The federated site consists of FATE and FATE manager.

Running FATE provides the sites with the ability to share data or resource within the federation. Similarly, FATE Manager provides the site with services such as joining federated organizations, site configuration, management, and monitoring.

- **FATE Cloud Deployment Notes**

At the moment, the cloud manager can be deployed as a separate service and does not depend on FATE to run. The only requirement is that the machine where it would be running needs to support the Java Development Kit and MySQL environments.

It is important to deploy the FATE system after completing FATE Manager deployment. The FATE manager service integrates into the FATE Board service of the FATE cluster as a plugin. Thus, if the FATE Board have already been deployed, then it is necessary to update it to the new version together with the FATE Manager. Refer to [91] for a more detailed information about FATE Cloud deployment.

- **Adding a new site to the Federation (see Figure 13)**

1. Cloud Manager adds a new site for the organization applying to join the federation, and assigns site identity information (PartyID, role) and key information (SecretKey).
2. FATE Manager starts the service and configures the local site network and submits the application.
3. Cloud Manager verifies the information submitted by the site. After the verification completed successfully, the site joins the federation.
4. Afterward, sharing of data and resources can be initiated between sites. The FATE Board can be launched to view current activities.

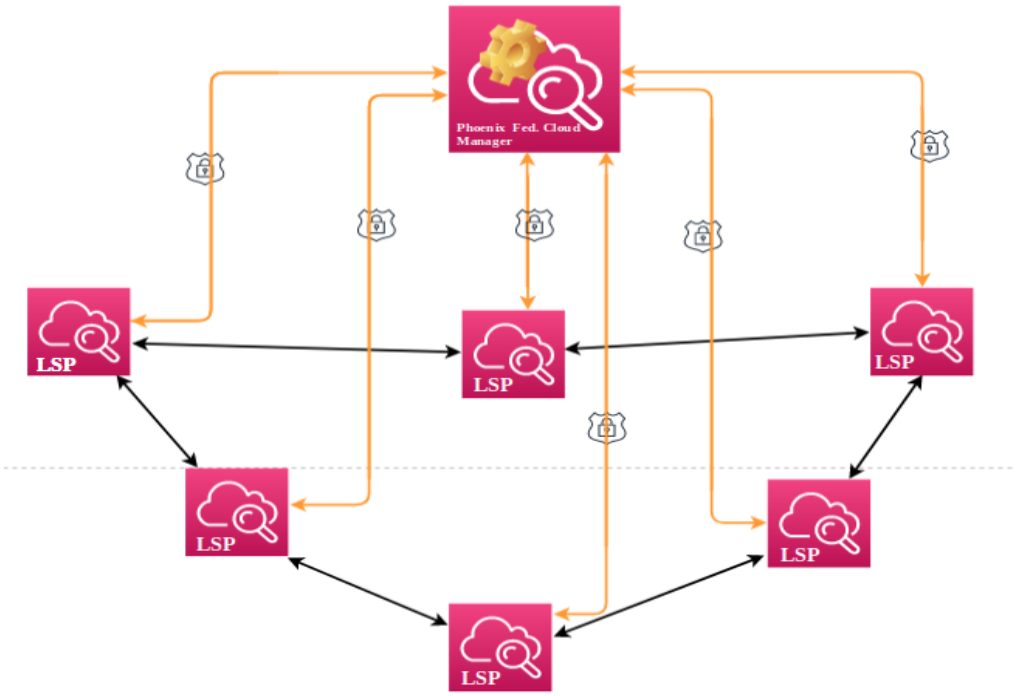


Figure 13 Schematic of Proposed Federated Cloud for the PHOENIX Use Cases

5. Resilience

Although there is not yet a standardized definition of resilience evaluation criteria and metrics, a common basic understanding is shared in different definitions [92]. Resilience in power systems expresses the systems capability to withstand High Impact Low Probability (HILP) events. Hence, it focuses on large and potentially catastrophic disturbances instead of single failures under normal operating conditions. Generally, resilience covers the preparation and prevention (before an event), the resistance of the system to degradation caused by the event and the recovery of the system, considering immediate response, restoration, and adaptation of the system.

The resilient behaviour of the system can be visualized through a Disturbance and Impact Resilience Evaluation (DIRE) curve (see Figure 14) [93] [94], which shows the performance of the system relative to pre-defined optimal and minimal performance levels.

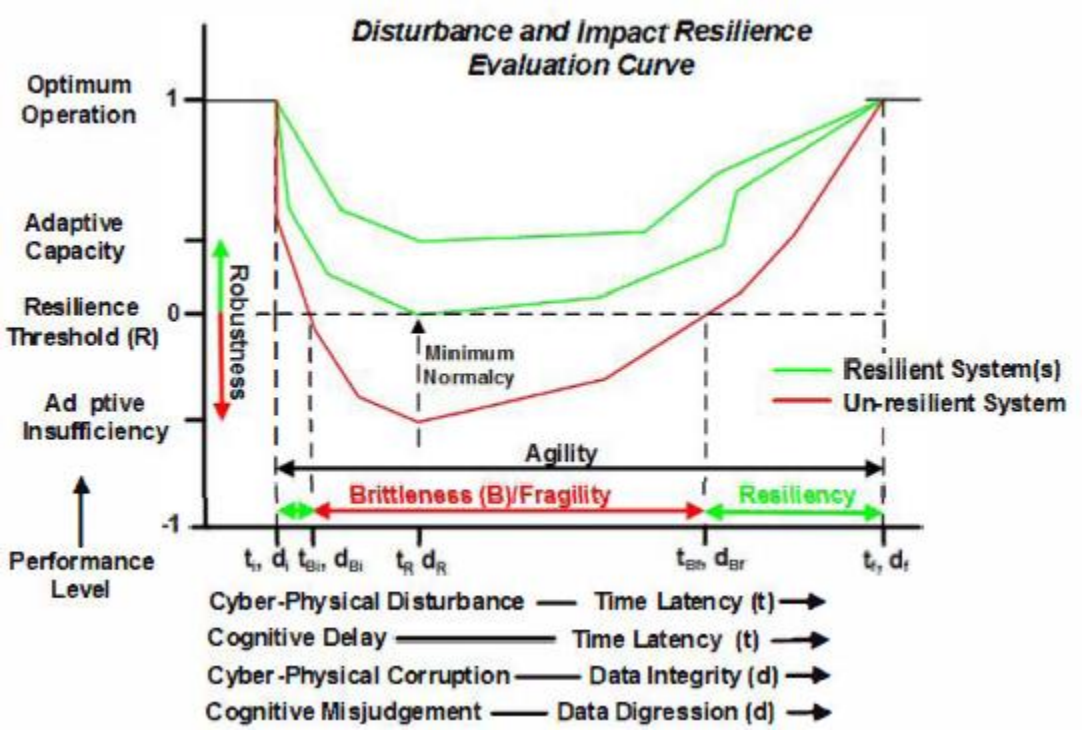


Figure 14: Resilience base metrics – DIRE [94]

To enhance the resilience of EPES infrastructures (i.e. the system's capability to withstand disturbances as natural disasters or cyber-attacks), both planning-based and operation-based strategies can be utilized. Several strategies are proposed in the scope of PHOENIX, as shown in Table 12.

Table 12: Proposed resilience enhancement methods

Enhancement class	Enhancement measure	Target domains
Planning-based strategy	Introduction of replication and Double Virtualization (DV) logic	Communication systems and EPES IT systems
Planning-based strategy	Utilization of blockchains and interledger-transactions	Communication systems and EPES IT systems
Operation-based strategy	Self-healing functions	Communication systems, EPES IT systems, and EPES electrical systems
Operation-based strategy	Survivability and recovery methods	EPES electrical systems

These strategies are to be combined in a resilience enhancement methodology, aiming to provide to provide self-healing functions for service restoration and recovery after failures to EPES infrastructures. The approaches to the different measures are outlined in the following sections, while the approach for a lab set-up and evaluation of the resilience improvements are described in Chapter 7.

5.1. Self-healing in EPES infrastructures

The self-healing functionality addresses the case of (partial) system degradation due to a successful attack or other causes such as technical failures or damage due to natural disasters. A set of initial failures of the electrical equipment may also lead to subsequent cascading effects, e.g. due to voltage or frequency instabilities or line overloading. Depending on the severity of the system degradation, different self-healing functions need to be provided.

In case of partial loss of load, e.g. due to single failures, Fault Localization, Isolation and Service Restoration (FLISR) algorithms can be applied to localize and isolate the fault and reconnect lost loads by reconfiguration of the electric grid, as described in [95].

In case of islanding of the local electrical system, existing resources need to be managed to allow for optimal supply of the local loads, considering prioritization of critical loads. Additionally, the local islanded microgrid may contribute to black start capability of the superordinate grid in case of a large-scale blackout.

5.2. Requirements to the fault tolerant resilience enhancement methodology

Even if advanced IDS solutions are deployed to protect EPES infrastructures from cyber-attacks, the risk of a successful attack cannot be completely eliminated, due to various reasons:

- Budgets for cyber security might be limited, offering only a certain level of protection
- Unknown threats may arise, new vulnerabilities might be detected and exploited by threat agents
- Human failure might prevent a quick and appropriate reaction to counteract cyber-attacks
- Human failure might enable threat agents to compromise protected systems

The enhancement of EPES infrastructures with fault tolerant self-healing functionalities is aiming to address this remaining risk of successful attacks and resulting failures. These can be both failures that are a direct result of an attack; but can also be consequences of implemented countermeasures to stop or contain an attack (e.g. separation of compromised electrical and communication networks to avoid cascading failures). The fault tolerant resilience enhancement methodology's targets are:

- Utilizing limited resources in the local infrastructure to enable continued operation, including provision of the most critical functionalities and supply of the highest prioritized assets or consumers
- Service restoration after failures by network reconfiguration
- Implementation in a fault tolerant way, avoiding single points of failure

5.2.1. EPES infrastructure and use cases

While DV can be applied in different contexts, the intended implementation to provide fault tolerance and self-healing to cyber-physical EPES is focusing on small, local energy communities in the first step. Considering PHOENIX LSPs, LSP3 and LSP4 are providing such environments in form of the BTC City and the Simris village, respectively. Both sites include:

- an active distribution grid with controllable loads and decentralized generation, storage units and/or household prosumers
- a local SCADA system and a communication infrastructure to control these assets.

The relevant threat scenarios for these LSPs as defined in D1.2 [96] include attacks on the SCADA systems, potentially leading to outages in the electrical grid. Hence, these LSPs are found suitable to provide the context of specifying use cases for the definition of the fault-tolerant resilience enhancement methodology and derive a laboratory set-up for testing and validation. A detailed specification is planned in cooperation with respective LSP owners and will be provided in upcoming deliverables D2.3 and D2.4.

5.2.2. Functional requirements of the resilience enhancement methodology

The functional requirements of the fault tolerant resilience enhancement methodology are defined in Table 13.

Table 13: Functional requirements

Requirement ID	Aspect	Requirement description
FR_DV_01	Double Virtualization	DV MUST provide virtualizing and mapping for control and monitoring functions
FR_DV_02	Double Virtualization	DV MUST provide virtualization of the interface device representation
FR_DV_03	Double Virtualization	DV MUST provide administration and management to track actively executed functions in different runtimes
FR_DV_04	Double Virtualization	DV MUST provide distributed administration and management to avoid single point of attack and/or failure

FR_DV_05	Double Virtualization	DV SHOULD provide virtualizing for databases
FR_DV_06	Double Virtualization	DV COULD provide distributed resource management which can be used in order to optimize response time during migration
FR_DV_07	Double Virtualization	DV COULD provide continuous reallocation or migration of virtualized functions and databases
FR_DLT	DLT	The utilization of DLT in the resilience enhancement methodology not yet defined. The functional requirements related to DLT will be provided in D2.3

An overview of functions that are contributing to self-healing EPES infrastructures and therefore would be suitable for implementation in the proposed methodology is presented in Table 14.

Table 14: Overview of functions contributing to resilience enhancement of EPES infrastructures

Aspect	Function	Description
FLISR	Fault detection and isolation	Fault detection and isolation provides information of the location of a fault, updating the network topology as a prerequisite for reconfiguration actions.
FLISR	Service Restoration	Service Restoration after a fault can be realized via network reconfiguration, where a new topology and the corresponding sequence of switching actions are computed. Network reconfiguration is leveraging on redundant paths provided in a meshed electrical grid.
Self-healing	Survival mode	Local microgrids might be able to operate in islanding mode, disconnected from the main grid and only supplied via local generation and storage. Islanding mode might be caused by incidents or attacks but could also be side effects of countermeasures (deliberate disconnection to prevent cascading outages). While operating in islanding mode, survival mode ensures prioritized operation of critical loads according to available energy in the islanded grid.
Self-healing	Black start	In case any grid-forming assets with black-start capability are available in the local EPES, the grid can be rebuilt after local or large-scale blackouts.
Self-healing	Rebuild grid from microgrids	In case of a large-scale blackout, local microgrids could participate in rebuilding the higher-level grid.

5.2.3. Non-functional requirements of the resilience enhancement methodology

The non-functional requirements of the fault tolerant resilience enhancement methodology are listed in Table 15. The final specification is depending on the actual use cases and is yet to be defined.

Table 15: Non-functional requirements

Requirement ID	Aspect	Requirement description
NFR_DV_01	DV	Performance (execution time)
NFR_DV_02	DV	Number of DV runtimes to be managed
NFR_DLT_01	DLT	Performance (execution time)
NFR_DLT_02	DLT	Number of nodes
NFR_SH_01	Functions for self-healing	Performance (execution time)

5.2.4. Implementation of the resilience enhancement methodology

A potential DV implementation of the FLISR functions is shown in Figure 15. Each of the assets is running a Node-RED runtime where various flows can be deployed. The term "flow" is also used to informally describe a single set of connected input-, output-, and processing-nodes. Thus, a flow (tab) can contain multiple flows (sets of connected nodes).

The Administration and Monitoring flow is required to run in each runtime, since it is providing the monitoring of DV assets and manages the (re-) deployment of flows. In addition, different flows containing the actual self-healing functions can be run. In case of failure or unavailability of an active DV asset, all flows can be redeployed in another available DV asset.

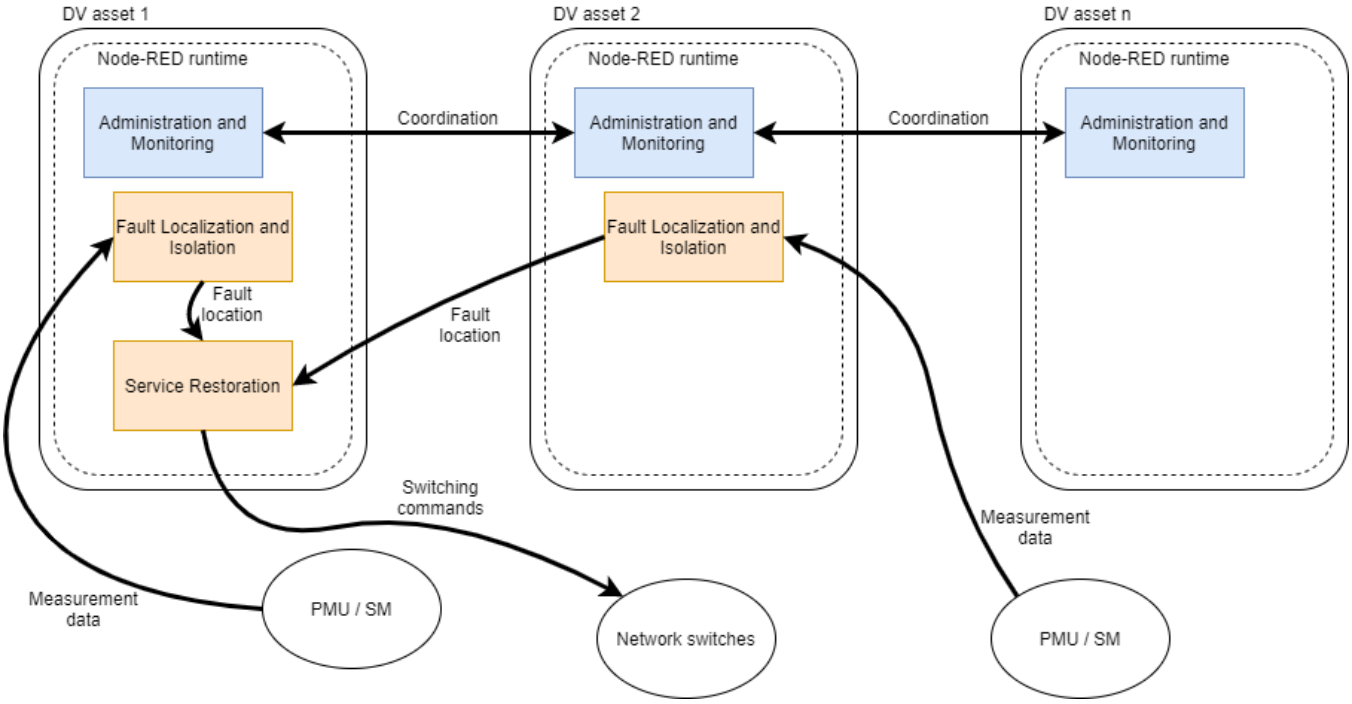


Figure 15: DV implementation of FLISR

Actions taken by the Administration and Monitoring (deployment of flows) and DV asset status as well as actions taken in the self-healing functions (switching commands) and electrical grid status and detected faults can be stored in a suitable DLT. However, the application and choice of a DLT depends on the use case that is to be implemented.

6. Non-functional Requirements of SPC Layer

To refine specification of the Secure and Persistent Communication (SPC) layer, this chapter lists data exchanges among different PHOENIX components where information is required to pass through SPC layer before delivering to destined entities. Additionally, each data flow is described from different perspectives including communication protocols, semantics, security, real-time and persistent requirements. This chapter only covers data exchanges which the SPC layer does participate in. For simplifying the whole concept in this effort, we assume an LSP network comprises two segments: an edge network, which provides operational data such as measurement information and a SCADA system which oversees issuing control commands for the edge network.

6.1. Non-functional requirements of SAPC data flows

The Situation Awareness, Perception & Comprehension (SAPC) component first gathers operational data such metering data from LSP devices and anomalies reports from USG. Further, it employs rather intensive and effective detection methods, with a global view of the various events on the networks which eventually results in providing situational awareness to LSP infrastructure as well as other PHOENIX components. To this end, SAPC identifies indicators of an attack, other incidents, and anomalies. Besides, it infers the probability of an attack occurrence according the current state of the system. SAPC also utilizes the privacy-preserving Federated Machine Learning (ML) and rule-based mechanisms in order to predict the potential attack evolution considering detected cyber-attacks and their potential ramifications. Table 16 presents characteristics of a data flow which is received by SAPC from USG. It is noteworthy that Table 16 only lists the data flow where SAPC exchanges messages to another PHOENIX component (i.e. USG) via SPC layer.

Component Name	SAPC
1. USG -> SAPC (Data pre-processing and normalization)	

Data flow	Communication protocols	Ontology	Data format	Data persistency	Data Security	Data Privacy	Real-time specs
1	Protocols such as REST and MQTT	Ontologies such as STIX	Data formats such as JSON	Between 6 months -1 year		no	<1s

Table 16: Data flows associated with SAPC component

As stated before, the data which is transferred from USG towards SAPC contains operational data, anomalies report and suspicious logs. Therefore, there are different data representations for operation data and CTI information. STIX 2.0 can be utilized to model CTI data within SPC layer since it is widely used in numerous cyber physical systems for representing different events and relations in the context of cyber threat intelligence. On the other hand, data models are used for operational data are highly dependent on underlying LSP infrastructures. For better understanding, let´s consider LSP1 where charging stations exchanges charging profiles with the central system. In this context, OCPP 1.6 protocol

is used to facilitate communication between these two entities. Additionally, charging profiles are encoded into different OCPP message formats using JSON. However, in LSP3 infrastructure, the data communication between HVAC controller and BMS server is formatted in CSV and XML and is transmitted via IP EN-ISO 16484-5 protocol.

According to Table 16, anomaly reports and threat logs generated by USG do not contain any personal data. Thereby, they can be easily stored in a DLT considering the immutable nature of data blocks in such technologies. This scenario was not feasible if the concerned data would contain any personal data with the right to be forgotten.

Upon detection of a threat at SAPC, the threat report, given the time constraints (i.e. less than 1 second), should be swiftly forwarded to IMEC and I2SP components via either a direct communication channel or an API call so that it can be promptly mitigated. In addition, SAPC hosts a full node which is part of the LSP ledger network. Thereby, the detected threat at SAPC can be written directly to the LSP-specific DLT within SPC layer for persistent storage. Further, the LSP-specific DLT may share the threat logs with other DLTs across PHOENIX platform (e.g. DLT of I2SP) via the Interledger layer. Regarding other SAPC outcomes, such as ML models, whose communication is not time-constrained, they might only be written to a DLT within SPC layer for persistent storage and further be communicated to DLTs of other LSPs and I2SP component through Interledger layer. It may be also noted that the countermeasure proposals are generated by SAPC and communicated to LSP infrastructure by IMEC.

6.2. Non-functional requirements of IMEC data flows

Upon reception of situational awareness of an LSP infrastructure and forecast of attack evolution from SAPC, Incident Mitigation and Enforcement Countermeasures (IMEC) component opts a set of countermeasures from a countermeasure repository. IMEC performs this selection in a way that the chosen countermeasures could minimize attack consequences at the optimal deployment cost. IMEC also provides countermeasure scores to SAPC in order to prioritize the most effective countermeasures considering a given situation. Besides, IMEC improves the countermeasure proposal from SAPC using the contextual information and further communicates the enhanced countermeasures to an LSP infrastructure where countermeasures can be applied either automatically or by an interference of a human operator. The characteristics of a data flow which is sent from IMEC towards an LSP infrastructure carrying countermeasures are the same as the ones shown in Table 16.

In a similar way to SAPC, IMEC uses STIX for modelling the proposed countermeasures. Given the time constraints (i.e. less than 1 second), the countermeasure proposals should be swiftly forwarded to LSP infrastructure via a direct communication channel so that it can be promptly mitigated. Furthermore, according to the table the countermeasures do not carry any personal data thus they can be stored persistently in a DLT without any concern about the right to be forgotten. To be more specific, IMEC runs a full node which is part of LSP ledger network and shares the countermeasures with other nodes in the ledger network.

It may be noted that SAPC and IMEC components use a message broker for all communication between themselves. The rationale behind this design choice is to separate the communication layer from the functional layer which results in reducing the friction and coordination effort. In other words, this communication paradigm helps separating the different components and tasks within SAPC and IMEC components, thus simplifying the development work.

6.3. Non-functional requirements of CMS data flows

The Configuration and Maintenance Service (CMS) provides configuration of the various networking assets. To this end, it carries out extensive testing on the network security level. Furthermore, it supports scheduled maintenance and periodic evaluation of software updates and fixes.

Table 17 lists data exchanges between CMS and other components across PHOENIX platform through SPC layer. It is good to mention that LSPs just share subsets of their local CTI data which is relevant for building the global view of EPES networks with I2SP. To this end, LSP operators specify a series of detected attacks, proposed countermeasures and ML models which can be shared with I2SP through Security Control Centre (SCC) dashboard. SCC is responsible for both visualizing the output of the PHOENIX EPES components (e.g. SAPC, IMEC) and allowing their (partial) customization, acting as the entry point of EPES operators to PHOENIX. After that, the desired information is retrieved from a node which runs on CMS and is sent later to I2SP DLTs via Interledger layer.

On I2SP side, the global view of EPES networks is built based on information transmitted by LSP infrastructures. Afterward, I2SP detects cyberattacks, identifies countermeasures and computes solid ML models for predicting potential attacks and associated consequences. The acquired CTI knowledge within I2SP is communicated to CMS via Interledger layer. More precisely, I2SP first stores the generated data on own consortium ledgers and further shares this information with every LSP DLT via Interledger layer.

Component name				configuration and maintenance service			
1. Incidents, Countermeasures detected/identified at LSP level (CMS -> I2SP) 2. Incidents, Countermeasures detected/identified at I2SP (I2SP -> CMS)							
Data flow	Communication protocols	Ontology	Data format	Data persistency	Data Security	Data Privacy	Real time specs
1	AMQP	STIX	JSON	>10 years	YES	No	1 minute
2	AMQP	STIX	JSON	>10 years	YES	No	1 minute

Table 17: Data flows associated with CMS

As described in Table 17, both data flows (either sent or received by CMS) are modelled using STIX. Besides, neither of data flows are carrying any personal data. Thereby, they are not protected by any privacy preserving methods. Due to absence of data privacy restrictions and long lifespan of data (i.e. more than 10 years), the described data can be stored in LSP-specific ledger as well as I2SP DLTs.

6.4. DLT requirements for SPC layer

Figure 16 indicates data flows which are exchanged across PHOENIX platform. Based on tables above, threat reports, proposed countermeasures and ML models regardless of their origins, do not contain any personal data. Consequently, they can be easily stored in ledger technologies considering the immutable nature of data records on DLTs and without violating GDRP compliance. Additionally, the CTI data generated in PHOENIX platform cannot be shared with everyone specially nodes outside the LSP premises. In a private blockchain, one organisation manages the access rights to data records thus only certain nodes are authorized to execute various transactions on blocks of data. While in a public blockchain, any node can join the blockchain network and further carry out different transactions such as read and write on data records. Considering the requirements of data flows in PHOENIX platform and characteristics of different types of blockchains which are described above, a private blockchain is seen as an appropriate data storage for secure and persistent communication layer. For prototyping purposes, Quorum and Hyperledger fabric which are both belonging to the category of private blockchains, will be tried out and eventually the blockchain which is better fitted the PHOENIX platform and at the same time simplifies the development process, will be selected for the final implementation.

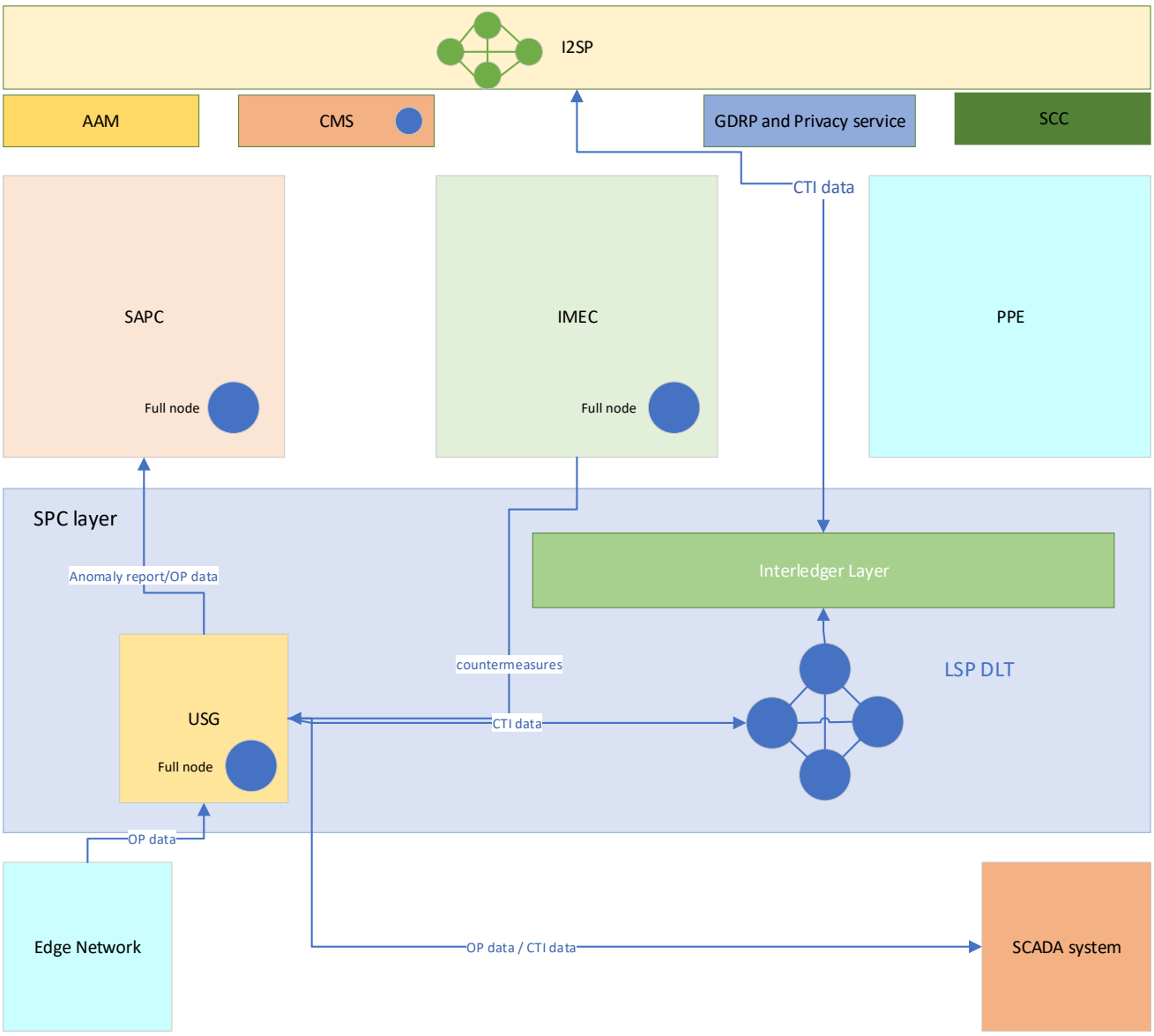


Figure 16: Simplified architecture of PHOENIX platform with associated data flows

6.4.1. Quorum

The main goal of Quorum platform is to enable businesses to benefit from public blockchains specifically Ethereum while their enterprise-driven needs are met. To this end, Quorum comprises a fork of the public Ethereum client 'geth' which is improved with several extensions in order to accommodate enterprise needs. More specifically, businesses can only utilize blockchain technologies where they fulfil the enterprise requirements such as privacy, performance and permissioning. To meet privacy needs, blockchains must guarantee the confidentiality of transactions while for performance, they need to offer the certain speed and scalability of the blockchain network. Besides, only authorized nodes can access the data records on blockchains.

Quorum, apart from Ethereum advantages such as transparency and immutability, offers business features like transaction privacy, multiple consensus methods, fine-gained access control management

and enhanced performance. It is noteworthy that Quorum supports public and private transactions. For public transactions, Quorum acts as a public Ethereum while it guarantees the confidentially of private transactions using a component called privacy manager. This module operates as an off-chain privacy mechanism which means outside the blockchain. Furthermore, the privacy manager interacts with Quorum entities via HTTPS and distributes transaction payloads among authorized entities. In other words, privacy manager is responsible for secure information exchange. To this end, it comprises transaction manager and Enclave [97]. The transaction manager which operates in a similar way to a network of Message Transfer Agents (MTA), is not specifically designed for blockchain technologies. Therefore, it can play a key role in any use cases where the secure communication is essential.

The transaction manager closely cooperates with Enclave to carry out the cryptography operations independently from other modules in Quorum. This feature leads to the notable privacy enhancement as well as performance improvement across the Quorum network. To this end, Enclave actively participates in symmetric key generation and data encryption/decryption. More precisely, Enclave first generates a random master key and a random nonce. Then it encrypts a transaction payload with the generated symmetric key. Additionally, Enclave calculates the hash of the encrypted payload. At the end, it encrypts the random master key with the public keys of the recipient nodes and sends them back along the encrypted payload to the transaction manager. It may be noted that Enclave only interacts with the corresponding transaction manager. The overall architecture of Quorum is presented in Figure 17.

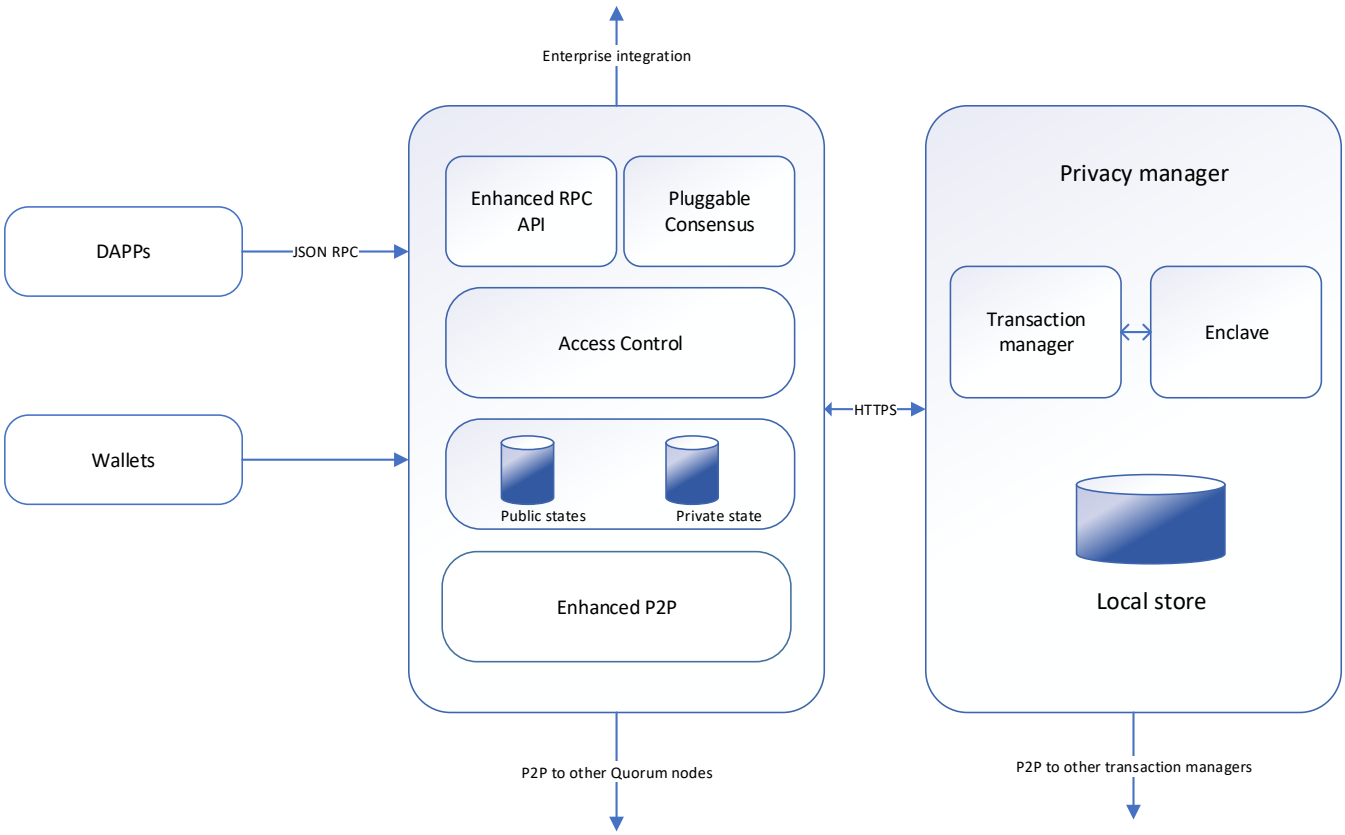


Figure 17: overall architecture of Quorum [98]

In Quorum, Tessera which is a java based stateless software, acts as a privacy manager and facilitates encryption, decryption, and distribution of private transactions. Tessera supports a wide range of mechanisms for key generation and data encryption and decryption. Besides, it can effortlessly integrate external Hardware Security Modules (HSMs) or cloud-hosted key management (e.g. Azure and AWS) which results in a more secure and flexible key management procedure. Quorum also allows deployment of various consensus methods across enterprise networks. This feature enables higher degree of transaction throughput for Quorum platform compared to the standard public Ethereum blockchain which uses a proof of work procedure as a consensus algorithm. For access control management, Quorum develops subset of Role Based Access Control (RBAC) method which is widely used in enterprise networks using smart contracts.

The permissioning mechanism in Quorum consists of two elements: policy management which issues an access control decision for each entity according to its own roles and enforcement logic which employs the decisions made by the policy management [98].

6.4.2. Hyperledger Fabric

In Hyperledger Fabric, different number of organizations interact with each other through elements called channels. Each channel is seen as a tunnel where a certain organization shares its own information and associated transactions with other organizations who are also participating in the same channel.

A node within Fabric network may take on any of the following roles:

- Peer is a node which joins one or more channels and records all transactions performed on corresponding channels. It is noteworthy that a peer stores the information associated with different channels separately.
- Orderer plays a key role in Fabric consensus mechanism. More accurately, an orderer orders transactions, creates a new block of ordered transactions and eventually disseminates a new block to all peers of a corresponding channel.
- Certificate Authority (CA) handles operations related to user certificates (e.g. user registration and revocation) since Hyperledger Fabric is a permissioned blockchain. For modeling roles and permissions, Hyperledger Fabric benefits from an X.509 standard certificate.
- Client is an application which enables interactions with Fabric network.

A peer´s ledger consists in blockchains and world state. Blockchain keeps track of all transactions executed on a specific channel while world state holds the current status of variables for a particular chaincode. It is good to mention that a smart contract concept in Ethereum network is called chaincode in Fabric. In Fabric, world state is implemented either by LevelDB or by CouchDB. LevelDB is a basic key-value database while CouchDB is a JSON-based database and unlike LevelDB is equipped with efficient querying mechanism.

A peer node may deploy one or more user-defined chaincodes. On the other hand, an orderer node instantiates system chaincodes that are responsible for gathering network, channel and configuration information. Fabric, before finalizing a block of transactions on ledgers, examines the permission, endorsement, data integrity and transaction order in several stages. For this purpose, Fabric executes a permissioned voting-based consensus where all nodes that belong to the same channel are partially trusted.

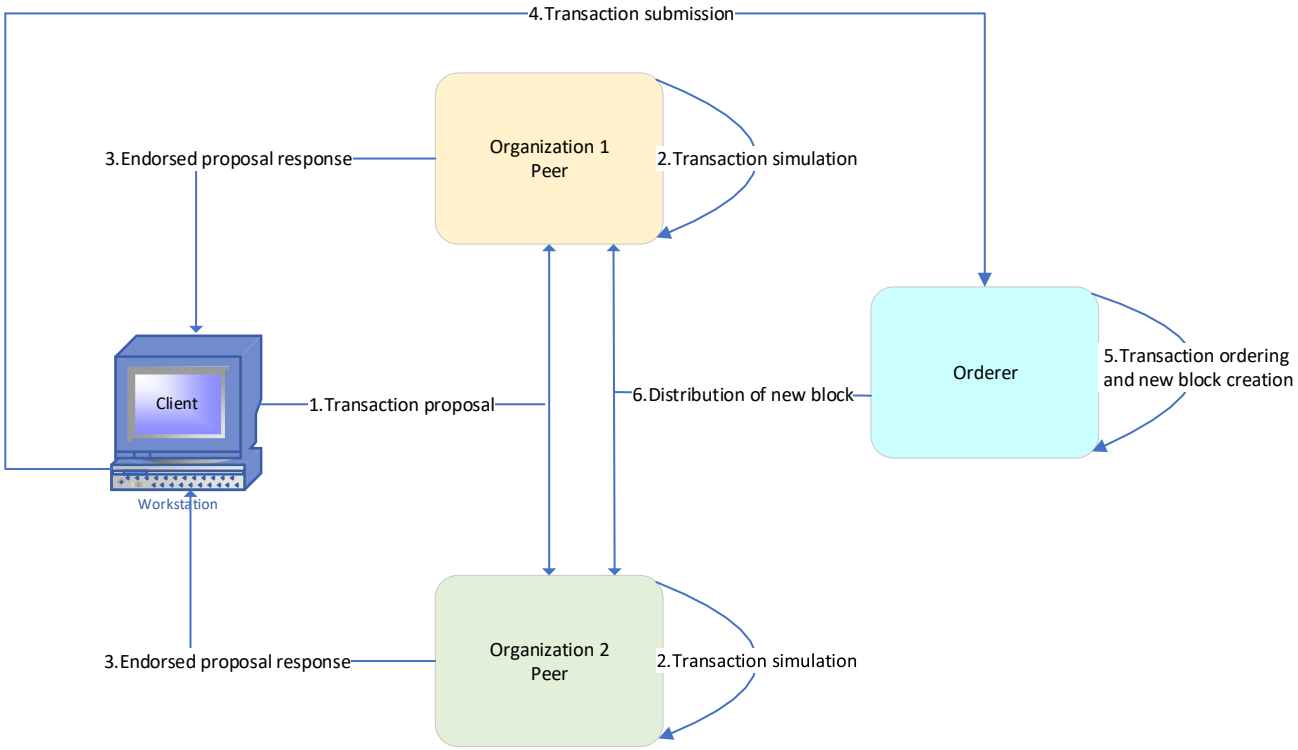


Figure 18: Workflow of Hyperledger Fabric [99]

As shown in Figure 18, to execute a transaction on Fabric network, the following steps need to be taken:

1. Client creates a transaction proposal, signs it by using their own certificate and eventually distributes the proposal on a relevant channel to pre-defined endorsing peers.
2. Every endorsing peer, upon reception of a transaction proposal and after validating the client certificate, executes the transaction and returns the results. Further, an endorsing peer advocates the generated response with its own certificate.
3. Client collects the endorsed proposal responses and examines them.
4. As a next step, client dispatches the transaction along the endorsed proposal responses to an orderer.
5. Orderer sorts the transactions, creates a new block out of them and signs the newly created block with own certificate.
6. At the final stage, the orderer distributes the newly created block to all peers communicating on a particular channel. Upon reception of a new block, each peer makes sure that the new block has been endorsed by enough number of endorsing peers. Further, the correctness of each transaction ordered in the new block is verified using the multi-version concurrency control (MVCC) check. After passing mentioned checks, the new block is added to the peer´s blockchain and peer´s world state is updated accordingly [99].

7. Simulation Studies

The resilience enhancement methodology described in Chapter 5 requires the implementation of functions that are directly controlling assets in the EPES infrastructure and can only be evaluated properly by triggering (partial) failures in the system. Since this is usually difficult or even impossible in EPES, a simulation set-up is foreseen for the evaluation of the resilience enhancement methodology. The simulation studies are complemented by an assessment of the resilience and improvements brought by implementing the resilience enhancement methodology. For the latter aspect, an initial assessment for a generic infrastructure is provided in Section 7.1. Section 7.2 presents the draft for a laboratory set-up.

7.1. Improvement of resilience by fault tolerant design approaches and self-healing functionality

The system investigated in this example scenario is based on a generic distribution grid segment [100], which has been extended by communication network and a measurement and control network as described in Figure 19. The power grid consists of multiple loads at the Low Voltage (LV) level, supplied by four different substations in a radial configuration (located at nodes 0, 5, 13 and 14). Since the grid is meshed, different paths are available to supply the loads under normal operating conditions. Figure 19 only shows the current radial configuration of the power grid, excluding the lines that are not utilized in this configuration. This redundancy is utilized for network reconfiguration in case of a fault.

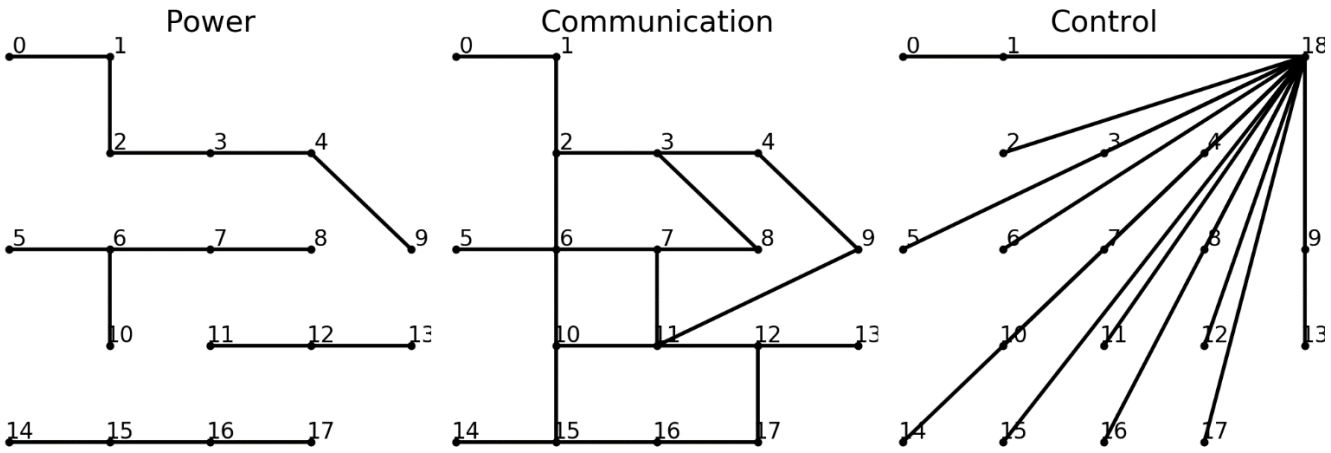


Figure 19: Different networks of the EPES of the example scenario

The following assumptions have been made for the communication and control networks and the interdependencies between the different networks that comprise the EPES analysed in the example scenario:

- An optical communication network is used to support measurement and control of the power grid. The optical cables are parallel to power lines, and there is a communication node at each load node of the power grid.
- The grid operation uses a measurement and control network with a central control node (the control centre node), which gathers measurement data, processes it, and provides control

functionality. This control centre node is connected to each of the measurement and control nodes, representing the nodes in the power grid that are equipped with measurement devices and switching equipment that can be remote-controlled.

- To model the interdependencies, it is assumed that each node of the power grid supplies the local communication node with power, which in turn supports one or multiple nodes of the measurement and control network. All the dependencies between nodes are represented in a node dependency graph.

In addition to the interdependencies, there are domain-specific intra-dependencies in each of the networks. For the power grid, it is assumed that each subgraph must include at least one load node as energy consumer and one substation node as energy provider. If this is not the case, all nodes of the subgraph fail. For the communication network, it is assumed that the isolated nodes fail, since they are no longer able to send or receive data and thus cannot fulfil their purpose. For the measurement and control network, it is assumed that each subgraph must include at least one control centre node as measurement data consumer and control command provider and one measurement node as measurement data provider and control command consumer.

7.1.1. Investigated configurations

In this example scenario, four different configurations of the system are evaluated and compared:

- A default configuration, where no by-design measures are implemented
- The “DV” configuration, where Double Virtualization is applied to virtualize part of the functionality of the measurement and control network and thus make it independent from the actual hardware. In this configuration, the control functionality is virtualized, and it is assumed that it can be hosted by any of the communication nodes at the power grid substations. This is represented in the model by adding dependencies, as control node 18 is now depending on communication nodes 5, 13 and 14 in addition to the dependence on node 0 (see Figure 19 for the default configuration)
- The “SR” (Service Restoration) configuration, where a network reconfiguration algorithm provides service restoration by design for the power grid, to resupply lost loads after a failure in the grid. This is represented in the model by adding edges to the power network, representing the lines with switches that are normally open and can be used to provide redundant paths. These additional lines are only used in the reconfiguration after a fault happened, therefore depending on a control command from the control centre. In case the control command cannot be received, the switches cannot close, and the line will not be put in operation.
- The “DV_SR” configuration, where both by-design measures are applied.

To evaluate the performance of the complete system after a fault has occurred, two performance indicators were chosen:

- The number of supplied loads, as a measure of the service level to energy consumers.
- The number of loads that are controllable, as a measure of the reliability of the final configuration after the cascading sequence has ended. For loads that are not controllable, measurements are not available, and targeted reconfiguration is not possible. If the number of controllable loads is smaller than the number of supplied loads, the capability to react to load changes or additional failures is impaired.

7.1.2. Results and conclusion

The results for a failure at load node 1 in the power subgraph for the default configuration are explained in detail to show how the failure is cascading through the system. Initially, as shown in Figure 20, load node 1 is failing, causing the edges to the neighbouring nodes to fail. The resulting cascading failures are shown in Figure 21:

- Since nodes 2, 3, 4, and 9 are no longer connected to a substation, they are unsupplied and fail.
- The respective communication nodes are no longer supplied and fail, too, isolating communication node 0 and causing it to fail.
- The cascade proceeds to the measurement and control network, causing the failure of some nodes (including the node that supports or provides control, respectively), then causing all links to fail, and finally also the isolated nodes.
- In the final state, a large part of the power grid is still supplied, but the controllability has been lost completely and the remaining grid can no longer be monitored.

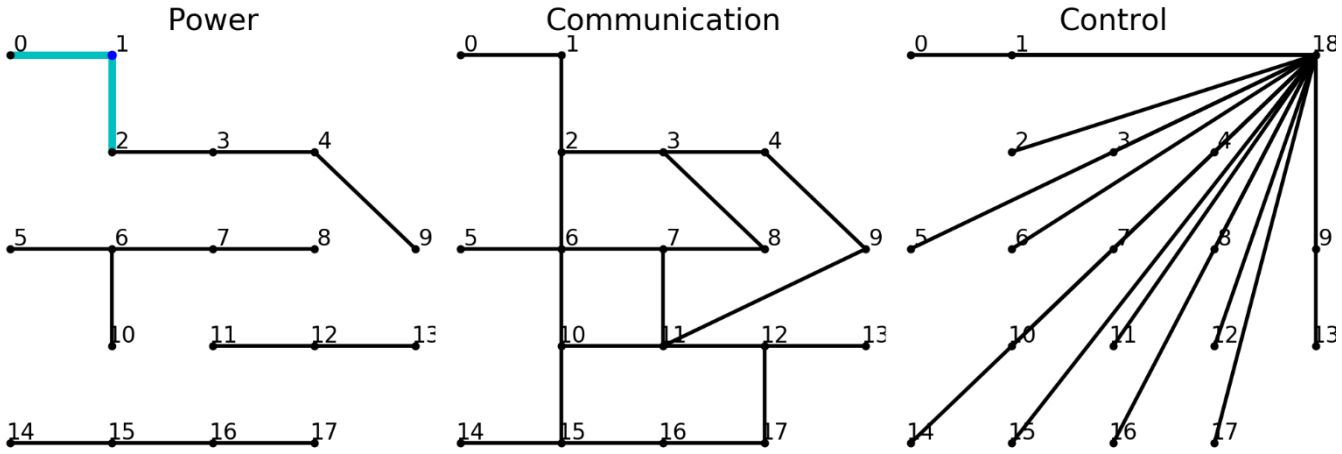


Figure 20: Initial failure of load node 1 in the default configuration

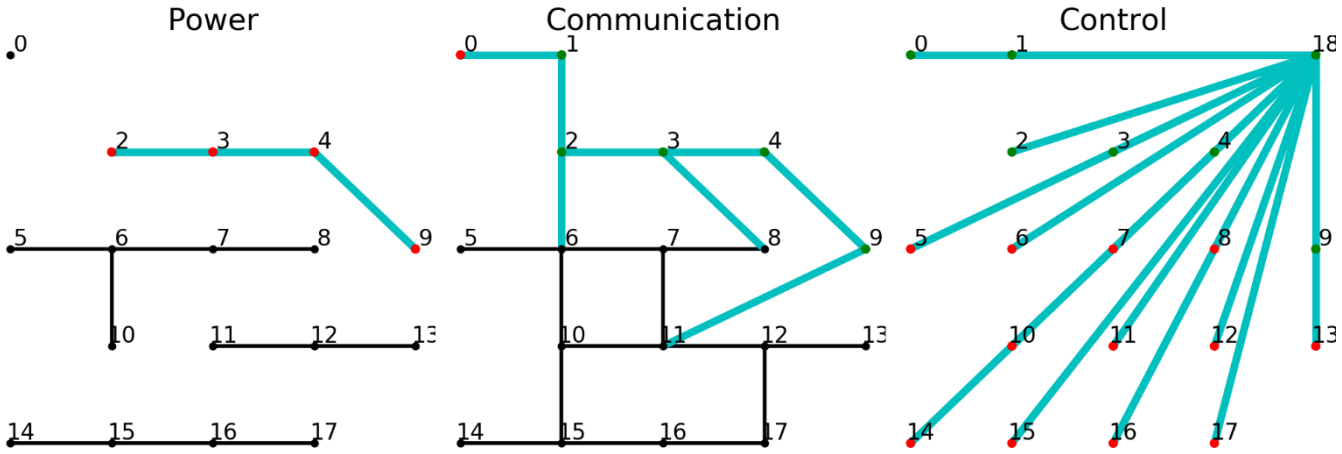


Figure 21: Cascading failures in the default configuration

For a comparison of the different by-design measures, the failure scenario described above has been repeated for each of the nodes in the power grid, representing a single fault happening in different parts of the grid. For each fault, the final state of all networks has been determined via the cascading analysis.

The result is shown for the different configurations as boxplots in Figure 22, where the box is marking the upper and lower quartile and the orange line marking the median, while the whiskers mark the minimum and maximum.

Even in the default configuration, most of the loads remained supplied no matter where the initial fault happened. However, as presented in the example, the capability to monitor and control the remaining part of the power grid is lost completely, if the substation hosting the control centre is affected by the initial fault. The DV configuration manages to solve this issue, enabling the other substations to provide redundancy for supporting the control centre functionality. In effect, the remaining part of the power grid remains controllable due to the DV application. Yet, it must be noted that DV does not improve the number of supplied loads. Since a reconfiguration of the power grid is not considered here, it is likely that the initial fault causes additional load nodes of the subgraph to fail. Due to the radial topology of the grid, the closer the fault is to the substation, the more load nodes are failing.

The Service Restoration configuration greatly improves this situation and enables more supplied loads in the final state. Due to the meshed topology of the grid that can be utilized for reconfiguration, if the initial fault occurs at a load node only the initially failed load is lost in the final state. If it occurs at a substation node, there may not be a load node failure at all. However, if the substation hosting the control centre functionality fails, the ability to reconfigure the grid is lost and the final state of the system is as in the default configuration.

Only the combination of both by-design measures provides complete containment of a fault, independent from its location. If a load node fails initially, only this node is failed in the final state. If a substation node fails initially, the power supply of all load nodes can be maintained. Finally, the capability of monitoring and control of the grid is secured.

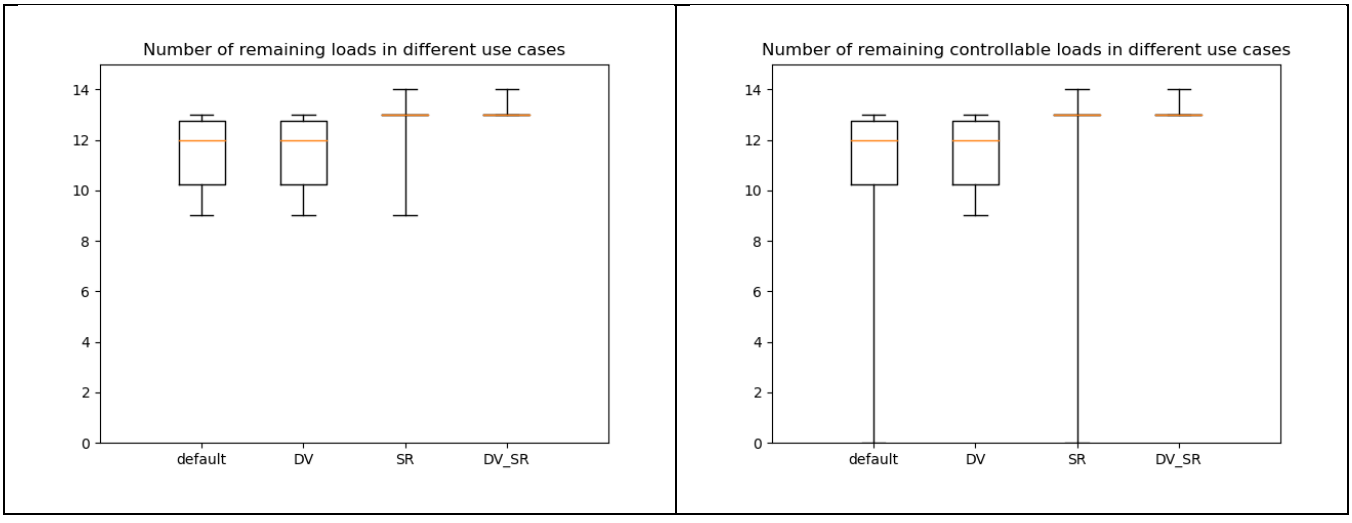


Figure 22: Remaining loads and controllable loads in different use cases

The above results show how the impact on the power supply can be minimized effectively by applying by-design measures. While each of the investigated measures improved the resilience of the grid, the combination of both measures provided additional synergies and can avoid the worst case of a failure at the substation hosting the control centre. Containing initial failures and reducing cascading to a minimum independent from the location of the initial failure is of increased importance in case of

targeted attacks on the most critical components of the system, to stop attackers exploiting vulnerabilities of the system.

7.2. Laboratory set-up to test and validate the fault tolerant resilience enhancement methodology

A first draft of the laboratory set-up to implement and evaluate the resilience enhancement methodology is depicted in Figure 23. The upper left side shows the power grid emulation, which is based on the Real-Time Digital Simulator (RTDS) of the RWTH lab. To replicate closely the real LSP environment, the simulation system will be modelled according to the real electrical systems of relevant LSPs, as described in Section 5.3.1. A model of the relevant communication infrastructures and the interdependencies between the different domains is included in the proposed set-up as shown on the lower left side. The DV infrastructure in the upper right side includes the different DV assets, which could be Raspberry PIs running a Node-RED runtime and the required administration and monitoring functions as well as the self-healing functions. A suitable DLT or blockchain infrastructure complements the set-up according to the defined use cases.

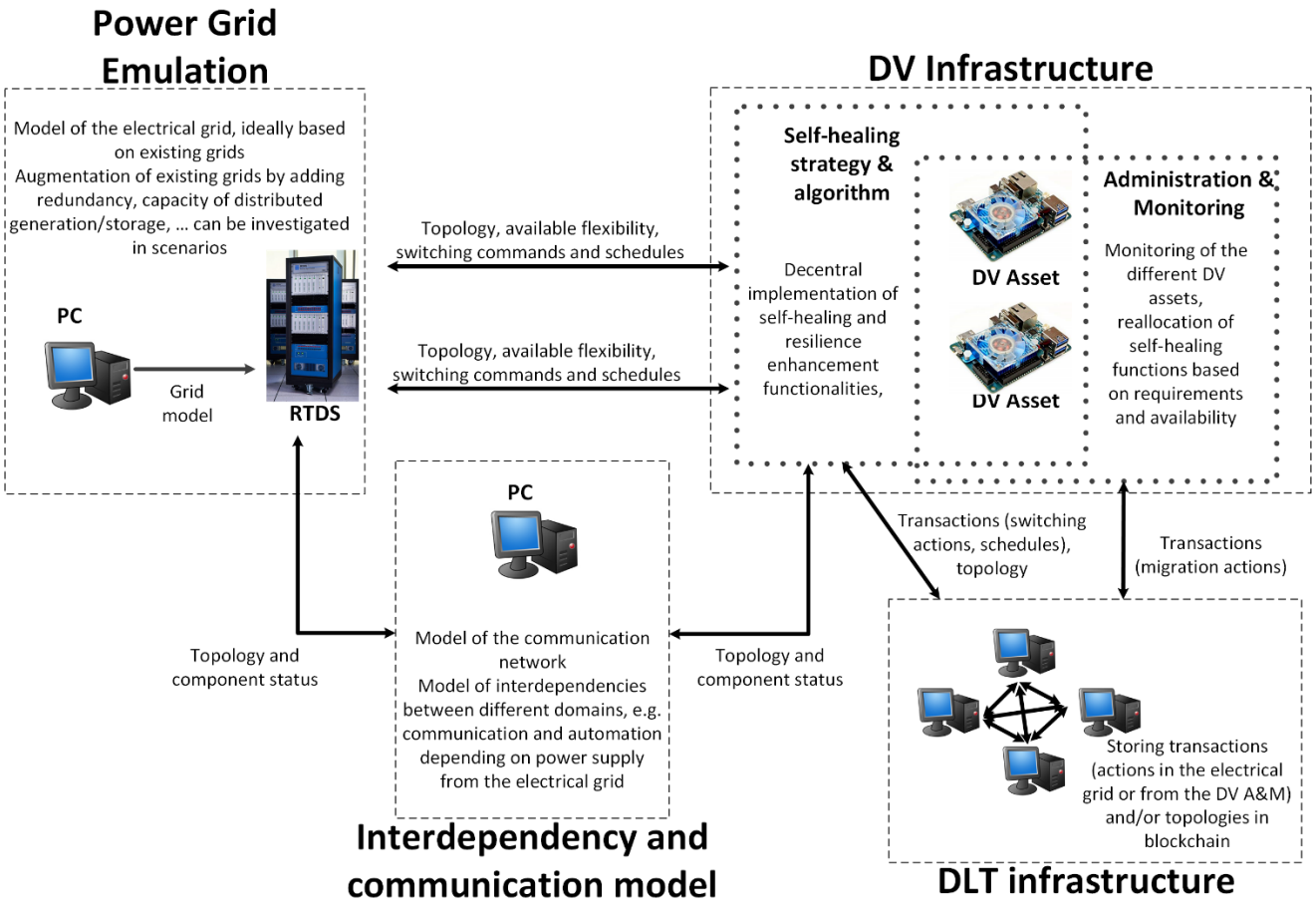


Figure 23: Draft of the laboratory set-up for evaluation of the resilience enhancement methodology

8. Conclusions

This deliverable provides a brief overview of data privacy preserving techniques such as differential privacy. Besides, it presents the benchmark template that is utilized to examine the compliance of the SPC layer to the PRESS framework throughout development phases. This template will be adopted by other PHOENIX components in order to comply with governance policies and data privacy rules defined in the PRESS framework.

This document first identifies data formats of data exchanges within LSP infrastructure as well as PHOENIX platform. Then, it recognizes STIX and TAXII as suitable protocols for modelling and transferring cyber threat information across LSP infrastructure. This deliverable also discusses the possibility of blockchains and cloud technologies deployment within LSP1 and LSP3 infrastructures as an effort to enhance data persistency, traceability, availability, integrity and interoperability. To do that, first it specifies the security, privacy, persistency and real-time requirements of each data flow within LSP premises. Later, considering data communication specifications it recognizes FATE Cloud and permissioned blockchains such as QUORUM and Hyperledger Fabric as suitable solutions for data storage.

In this deliverable, the concept of resilience by design for EPES infrastructures is explained thoroughly. More accurately, this document discusses planning-based and operation-based strategies as well as self-healing methodologies which aim to improve the overall system resilience. Additionally, this deliverable provides the refinements for the non-functional requirements of SPC layer which were initially introduced in D2.1.

Finally, this document presents the simulation set-up for evaluation of the proposed self-healing methodologies (i.e. double virtualization and service restoration).

9. References

[1] PHOENIX, “D2.1 PHOENIX Platform Architecture Specification,” H2020 - 832989 - PHOENIX Deliverable Report, 2020.

[2] PHOENIX, “D4.1 PRESS Framework Analysis,” H2020 - 832989 - PHOENIX Deliverable Report, 2020.

[3] Hassan, Rehmani and Chen, "Differential Privacy Techniques for Cyber Physical Systems: a Survey," *IEEE Communications Survey & Tutorials,* 2020.

[4] Sweeney, “K-anonimity: A model for protecting privacy.,” *Journal of Uncertainty, Fuzziness and Knowledge-Based Systems,* 2002.

[5] Machanavajjhala, Kifer, Gehrke and Venkitasubramaniam, “l-diversity: Privacy beyond k-anonymity,” *ICDE,* 2006.

[6] N. Li, T. Li and S. Venkatasubramanian, “t- closeness: Privacy beyond k-anonymity and l-diversity,” in *2007 IEEE 23rd International Conference on Data Engineering*, 2007.

[7] Dwork, “Differential Privacy,” 2006.

[8] Dwork and Roth, The Algorithmic Foundations of Differential Privacy, Now Foundations and Trends, 2014, p. 288.

[9] Begum and Nausheen, “A comparative analysis of differential privacy vs other privacy mechanisms for big data,” *In proceeding of IEEE ICISC,* 2018.

[10] Vadhan, “The complexity of differential privacy,” *Tutorials on the Foundations of Cryptography,* 2017.

[11] Ye, Liu, Wang, Li, Li and Li, “Secure and efficient outsourcing differential privacy data release scheme in cyber physical system,” *Future Generation Computer Systems,* 2020.

[12] S. Song, K. Chaudhuri and A. D. Sarwate, “Stochastic gradient descent with differentially private updates,” *IEEE Global Conference on Signal and Information Processing,* 2013.

[13] Rajkumar and Agarwal, “A Differentially Private Stochastic Gradient Descent Algorithm for Multiparty classification,” *In Proceedings of the 15th International Conference on Artificial Intelligence and Statistics ,* 2012.

[14] Hegedus and Jelasity, “Distributed Differentially Private Stochastic Gradient Descent: An Empirical Study,” *2016 24th Euromicro International Conference on Parallel, Distributed, and Network-Based Processing (PDP),* 2016.

[15] Jin, He and Dai, “Decentralized Differentially Private Without-Replacement Stochastic Gradient Descent,” *Cornell University - CoRR,* 2019.

[16] Bassily, Smith and Thakurta, “Private Empirical Risk Minimization: Efficient Algorithms and Tight Error Bounds,” *In: 2014 IEEE 55th Annual Symposium on Foundations of Computer Science,* 2014.

[17] McMahan, Ramage, Talwar and Zhang, “Learning Differentially Private Recurrent,” 2018.

[18] McMahan, Andrew, Erlingsson, Chien, Mironov, Papernot, Kairouz and Adding, “Differential Privacy to Iterative Training Procedure,” 2019.

[19] Jain and Thakurta, “(Near) Dimension Independent Risk Bounds for Differentially Private Learn,” *In: Proceedings of the 31st International Conference on Machine Learning,* 2014.

[20] Zhu, Xiong, Li, Zhou and Yu, “Differentially private model publishing in cyber physical systems,” *Future Generation Computer Systems,* 2020.

[21] Task and Clifton, “A Guide to Differential Privacy Theory in Social Network Analysis,” *IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining,* 2012.

[22] Task and Clifton, “What Should We Protect? Defining Differential Privacy for Social Network Analysis,” *State of the Art Applications of Social Network Analysis ,* 2014.

[23] Andrés, Bordenabe, Chatzikokolakis and Palamidessi, “Geo-Indistinguishability: Differential Privacy for Location-Based Systems,” in *CCS '13: Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security*, 2014.

[24] Wang, Qin, Yang, Han and Ma, “Geographic Differential Privacy for Mobile Crowd Coverage Maximization,” 2018.

[25] G. Acs and C. Castelluccia, “A Case Study: Privacy Preserving Release of Spatio-temporal Density in Paris,” in *KDD '14: Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2014.

[26] Lou, Tan, Yau and Cheng, “Cost of Differential Privacy in Demand Reporting for Smart Grid Economic Dispatch,” *IEEE Transactions on Network Science and Engineering,* 2020.

[27] U. Hassan, Rehmani, Kotagiri, Zhang and Chen, “Differential privacy for renewable energy resources based smart metering,” *Journal of Parallel and Distributed Computing,* 2019.

[28] Jonsson and Nelson, “Applied Differential Privacy in the Smart Grid,” Chalmers University of Technology, 2015.

[29] Eibl and Engel, “Differential privacy for real smart metering data,” *Computer Science - Research and Development volume ,* 2017.

[30] Ghayyur, Chen, Yus, Machanavajjhala, Hay, Miklau and Mehrotra, “IoT-Detective: Analyzing IoT Data Under Differential Privacy,” *In SIGMOD’18: 2018 International Conference on Management of Data,* 2018.

[31] K. Gai, Y. Wu, L. Zhu, Z. Zhang and M. Qiu, “Differential Privacy-Based Blockchain for Industrial Internet-of-Things,” *IEEE Transactions on Industrial Informatics,* vol. 16, no. 6, pp. 4156-4165, 2020.

[32] Li and Palanisamy, “Privacy in Internet of Things: from Principles to Technologies,” *IEEE Internet of Things Journal,* 2018.

[33] Kargl, Friedman and Boreli, “Differential privacy in Intelligent Transportation Systems,” *Proceedings of the 6th ACM Conference on Security and Privacy in Wireless and Mobile Networks,* 2013.

[34] Zhao, Zhang, Wan, Liu and Umer, “A survey of local differential privacy for securing internet of vehicles,” *The Journal of Supercomputing,* 2019.

[35] Zhao, Zhao, Yang, Wang, Wang, Lyu, Niyato and Lam, “Local Differential Privacy based Federated Learning for Internet of Things,” 2020.

[36] Finster and Baumgart, “Privacy-aware smart metering: a survey,” *IEEE Communications Surveys & Tutorials,* vol. 17, no. 2, pp. 1088-1101, 2015.

[37] Hart, “Nonintrusive appliance load monitoring,” *Proceedings of the IEEE,* vol. 80, no. 12, pp. 1870-1891, 1992.

[38] Sultanem, “Using appliance signatures for monitoring residential loads at meter panel level,” *IEEE Trans. Power Del,* 1991.

[39] Aladesanmi and Folly, “Overview of non intrusive load monitoring and identifications techniques,” *IFAC PapersOnLine,* 2015.

[40] Barbosa, Brito and Almeida, “A technique to provide differential privacy for applicance usage in smart metering,” *Information Sciences,* vol. 370–371, pp. 355-367, 2016.

[41] Chen, Machanavajjhala, Hay and Miklau, “PeGaSus: Data-Adaptive Differentially Private Stream Processing,” *In Proceeding of ACM SIGSAC Conf. Comput. Commun. Security,* 2017.

[42] Liu, Zhang and Fang, “EPIC: A differential privacy framework to defend smart homes against Internet traffic analysis,” *IEEE Internet Things J.,* 2018.

[43] Sun, Lampe and Wong, “EV-assisted Battery Load Hiding: A Markov Decision Process Approach,” *IEEE International Conference on Smart Grid Communications ,* 2016.

[44] Zhang, Cao, Qin, Zhu, Yu and Ren, “When privacy meets economics: enabling differentially-private battery-supported meter in smart grid,” *In Proceding of IEEE 25th Int. Symp. Qual. Service,* 2017.

[45] Zhang, Qin and Zhu, “Cost-Friendly Differential Privacy for Smart Meters: Exploiting the Dual Roles of the Noise,” *IEEE Transactions on Smart Grid,* 2017.

[46] Z. Zhang, Z. Qin, L. Zhu, W. Jiang, C. Xu and K. Ren, “Toward practical differential privacy in smart grid with capacity-limited rechargeable,” 2015.

[47] G. Ács and C. Castelluccia, “I have a DREAM! (Differentially private smart metering),” in *International Workshop on Information Hiding*, 2011.

[48] Liao and Formby, “Di-PriDA: Differentially Private Distributed Load Balancing Control for the Smart Grid,” *IEEE TRANSACTIONS ON DEPENDABLE AND SECURE COMPUTING,* 2019.

[49] Pal, Hui and Prasanna, “Privacy Engineering for the Smart Micro-Grid,” *IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING,* 2019.

[50] Xiong, Ren, Chen, Yao, Lin, Wu and Niu, “Enhancing privacy and availability for data clustering,” *IEEE Internet of Things Journal,* 2018.

[51] T. R. Gruber, “A translation approach to portable ontology,” *Knowledge Acquisition,* vol. 5, no. 2, p. 199–220, 1993.

[52] M. Grønberg, “ An Ontology for Cyber Threat Intelligence,” University of Oslo, 2019.

[53] F. Böhm, F. Menges and G. Pernul, “Graph-based visual analytics for cyber threat intelligence.,” *Cybersecurity 1,* vol. 16, 2018.

[54] C. Sauerwein, C. Sillaber, A. Mussmann and R. Breu, “Threat intelligence sharing platforms: An exploratory study of software vendors and research perspectives,” in *13th International Conference on Wirtschaftsinformatik*, 2017.

[55] E. W. Burger, M. D. Goodman, P. Kampanakis and K. A. Zhu , “Taxonomy Model for Cyber Threat Intelligence Information Exchange Technologies,” in *Proceedings of the 2014 ACM Workshop on Information Sharing & Collaborative Security*, 2014.

[56] Cyware, “What is Open Indicators of Compromise (OpenIOC) Framework?,” 17 June 2019. [Online]. Available: https://cyware.com/educational-guides/cyber-threat-intelligence/what-is-open-indicators-of-compromise-openioc-framework-ed9d. [Accessed 23 August 2020].

[57] The MITRE Corporation, “Cyber Observable eXpression,” 2014. [Online]. Available: https://cybox.mitre.org/about/. [Accessed 23 August 2020].

[58] Wikipedia, “Open Charge Point Protocol,” 22 January 2020. [Online]. Available: https://en.wikipedia.org/wiki/Open_Charge_Point_Protocol. [Accessed August 2020].

[59] Y. PIRILDAK, “An Overview of OCPP (Open Charge Point Protocol),” 4 March 2020. [Online]. [Accessed 23 August 2020].

[60] CVE, “Common Vulnerabilities and Exposures,” 2020. [Online]. Available: https://cve.mitre.org/cve/. [Accessed 15 September 2020].

[61] J. Undercoffer, A. Joshi and J. Pinkston, “Modeling Computer Attacks:An Ontology for Intrusion Detection,” in *International Workshop on Recent Advances in Intrusion Detection*, 2003.

[62] S. More, M. Matthews, A. Joshi and T. Finin, “A Knowledge-Based Approach to Intrusion Detection Modeling,” in *Security and Privacy Workshop*, 2012.

[63] A. Grégio, R. Bonacin, O. Nabuco, V. M. Afonso, P. L. De Geus and M. Jino, “Ontology for Malware Behavior: a Core Model Proposal,” in *In IEEE International WETICE Conference*, 2014.

[64] M. Iannacone and J. Goodall, “Developing an Ontology for Cyber Security Knowledge Graphs,” in *in Proceedings of the 10th Annual Cyber and Information Security Research Conference*, 2015.

[65] A. Grégio and P. Licio, “An Ontology of Suspicious Software Behavior,” *Applied Ontology,* vol. 11, no. 1, pp. 29-49, 2016.

[66] M. Atighetchi, B. Simidchieva, F. Yaman, T. Eskridge , M. Carvalho and C. N. Paltzer, “Using Ontologies to Quantify Attack Surfaces,” in *STIDS Proceedings*, 2016.

[67] MITRE, “Common Attack Pattern Enumeration and Classification,” [Online]. Available: https://capec.mitre.org/..

[68] M. Pendleton, R. Garcia-Lebron and S. Xu, “A Survey on Systems Security Metrics,,” *ACM Computing Surveys (CSUR),* vol. 49, no. 4, 2016.

[69] Z. Syed, A. Padia, T. Finin, L. Mathews and A. Joshi, “UCO: A Unified Cybersecurity Ontology,” in *Proceedings of the AAAI Workshop on Artificial Intelligence for Cyber Security*, 2016.

[70] Y. Biran, G. Collins, S. Azam and J. Dubow, “Federated Cloud computing as System of Systems,” in *2017 Int. Conf. Comput. Netw. Commun. ICNC 2017*, 2017.

[71] C. A. Lee, R. B. Bohn and M. Michel, “The NIST Cloud Federation Reference Architecture,” 2020.

[72] R. Shere, S. Srivastava and R. K. Pateriya, “A review of federated identity management of OpenStack cloud,” in *Int. Conf. Recent Innov. Signal Process. Embed. Syst. RISE 2017*, 2018.

[73] M. R. M. Assis, L. F. Bittencourt, R. Tolosana-Calasanz and C. A. Lee, “Cloud Federations: Requirements, Properties, and Architectures,” in *Developing Interoperable and Federated Cloud Architecture*, IGI Global, 2016, p. 1–41.

[74] P. Thakur and D. K. Shrivastava, “Interoperability Issues and Standard Architecture for Service Delivery in Federated Cloud: A Review,” in *Proc. - 2015 Int. Conf. Comput. Intell. Commun. Networks, CICN 2015*, 2016.

[75] “Bitcoin: A peer-to-peer electronic cash system,” academia.edu , 2008. [Online].

[76] Ripple. [Online]. Available: https://ripple.com .

[77] ethereum.org, “Ethereum,” [Online]. Available: https://ethereum.org/en/.

[78] Hyperledger Indy, [Online]. Available: https://www.hyperledger.org/projects/hyperledger-indy.

[79] Exonum, [Online]. Available: https://exonum.com/index .

[80] Rubix, [Online]. Available: https://rubix.io/.

[81] Hyperledger Fabric, [Online]. Available: https://www.hyperledger.org/projects/fabric.

[82] Wikipedia, “Proof of work,” [Online]. Available: https://en.wikipedia.org/wiki/Proof_of_work.

[83] Wikipedia, “Byzantine fault,” [Online]. Available: https://en.wikipedia.org/wiki/Byzantine_fault.

[84] Wikipedia, “Proof of stake,” [Online]. Available: https://en.wikipedia.org/wiki/Proof_of_stake.

[85] Wikipedia, “Proof of burn,” [Online]. Available: https://en.bitcoin.it/wiki/Proof_of_burn.

[86] Wikipedia, “Proof of space,” [Online]. Available: https://en.wikipedia.org/wiki/Proof_of_space.

[87] Cryptonomist, “Proof of Elapsed Time (PoET): the time-based consensus algorithm,” [Online]. Available: https://en.cryptonomist.ch/2019/06/15/proof-of-elapsed/.

[88] V. Siris, P. Nikander, S. Voulgaris, N. Fotiou, D. Lagutin and G. Polyzos, "Interledger Approaches," in IEEE Access, vol. 7, pp. 89948-89966, doi: 10.1109/ACCESS.2019.2926880., 2019.

[89] Github, “FATE Cloud on Github,” [Online]. Available: https://github.com/FederatedAI/FATE-Cloud.

[90] FATE community, “Fate cloud website,” [Online]. Available: https://fate.fedai.org/.

[91] Github, “FATE Cloud Deployment Guide,” [Online]. Available: https://github.com/FederatedAI/FATE-Cloud/blob/master/cloud-manager/deploy/doc/Cloud-Manager_Deploy_Guide.md.

[92] N. Bhusal, M. Abdelmalak, M. Kamruzzaman and M. Benidris, “Power System Resilience: Current Practices, Challenges, and Future Directions,” *IEEE Access, vol. 8, pp. 18064-18086,* 2020.

[93] L. M. F. Stevens, C. Rieger and W. C. Phoenix, “HTGR Resilient Control System Strategy,” Idaho National Lab. (INL), Idaho Falls, ID (United States), 2010.

[94] C. G. Rieger, “Resilient Control Systems - Practical Metrics Basis for Defining Mission Impact,” *2014 7th International Symposium on Resilient Control Systems (ISRCS),* August 2014.

[95] A. Dognini, A. Sadu, A. Angioni, F. Ponci and A. Monti, “Rule-Based Optimization Algorithm for Service Restoration of Active Distribution Grids,” in *(currently under revision)*.

[96] PHOENIX Consortium, “EPES threat modelling & analysis of new threats,” 2020.

[97] Anjuna Security, Inc., “What is a Secure Enclave?,” 2020. [Online]. Available: https://www.anjuna.io/what-is-a-secure-enclave. [Accessed 14 September 2020].

[98] Blockgeeks, “What Is Quorum Blockchain? A Platform for The Enterprise,” 2020. [Online]. Available: https://blockgeeks.com/guides/quorum-a-blockchain-platform-for-the-enterprise/#View_of_Party_B. [Accessed 2020].

[99] P. Thummavet, “Demystifying Hyperledger Fabric (1/3): Fabric Architecture,” 2 May 2019. [Online]. Available: https://medium.com/coinmonks/demystifying-hyperledger-fabric-1-3-fabric-architecture-a2fdb587f6cb. [Accessed 13 August 2020].

[100] N. R. M. Fontenele, L. S. Melo, R. P. S. Leao and R. F. Sampaio, “Application of multi-objective evolutionary algorithms in automatic restoration of radial power distribution systems.,” *2016 IEEE Conference on Evolving and Adaptive Intelligent Systems (EAIS),* p. 33–40, May 2016.