NEXT GENERATION INTERNET

Grant Agreement No.: 957216
Call: H2020-ICT-2018-2020

Topic: ICT-56-2020
Type of action: RIA

D5.1 Key Technologies for IoT Data Management Benchmark

Revision: v1.0

Work package	WP5
Task	T5.1, T5.2, T5.3
Due date	30/06/2021
Submission date	30/06/2021
Deliverable lead	TEI
Version	1.0
Editors	Gino Ciccone (TEI), Stefan Köpsell (BI), Marek Bednarczyk (PJATK), Carlos Alcaide Pastrana (TID), Anssi Iappalainen (AWA); Alexandr Tardo (CNIT)
Authors	Gino Ciccone, Giuseppina Carpentieri, Cosimo Zotti (TEI), Alexandr Tardo (CNIT), Marek Bednarczyk, Tadeusz Puźniakowski, Paweł Czapiewski (PJATK), Stefan Köpsell, Kumar Sharad (BI), José Luis Cárcel, Joan Meseguer (FV), Ahmad Nimr, Ivo Bizon Franco de Almeida (TUD), Pietro Piscione, Erin Seder, Giacomo Bernini (NXW), Carlos Alcaide Pastrana, César Rodríguez Cerro (TID), Juan Jose Garrido Serrato, Christos Politis (SES), Jussi Poikonen (AWA)
Reviewers	Carsten Weinhold (BI), Nuria Molner, Salvador Garcia Jimenez (UPV), Anton Luca Robustelli, Raffaele De Santis (TEI),

Abstract	This document describes the approach of iNGENIOUS to develop an interoperable layer, aggregating data coming from different existing and forthcoming IoT technologies. The deliverable describes the state of the current technologies and the planned innovations applied to internet-of-things (IoT) data management and applications.
Keywords	Smart IoT GW, IoT, DVL, cross-DLTs, Smart application, Interoperable Layer

Document Revision History

Version	Date	Description of change	List of contributors(s)
V1.0	30/06/2021	EC Version	See Authors

Disclaimer

This iNGENIOUS D5.1 deliverable is not yet approved nor rejected, neither financially nor content-wise by the European Commission. The approval/rejection decision of work and resources will take place at the Mid-Term Review Meeting planned in June 2022, after the monitoring process involving experts has come to an end.

The information, documentation and figures available in this deliverable are written by the "Next-Generation IoT solutions for the universal supply chain" (iNGENIOUS) project's consortium under EC grant agreement 957216 and do not necessarily reflect the views of the European Commission.

The European Commission is not liable for any use that may be made of the information contained herein.

Copyright Notice

Project co-funded by the European Commission in the H2020 Programme		
Nature of the deliverable:	R*	
Dissemination Level		
PU	Public, fully open, e.g. web	✓
CL	Classified, information as referred to in Commission Decision 2001/844/EC	
CO	Confidential to iNGENIOUS project and Commission Services	

** R: Document, report (excluding the periodic and final reports)*

DEM: Demonstrator, pilot, prototype, plan designs

DEC: Websites, patents filing, press & media actions, videos, etc.

OTHER: Software, technical diagram, etc.

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

Executive Summary

This document describes the state of art of current technologies applied to Internet-of-Things (IoT) data management and applications, and proposes an innovative solution for the implementation of the interoperable and application layers foreseen in iNGENIOUS platform able to aggregate data coming from different existing and forthcoming IoT technologies. This layer exposes to the upper application the information needed to implement the service logics.

The document starts with the description of the technologies for IoT Data Management related to the M2M interoperability, data virtualization, DLT interoperability, security, privacy and smart IoT gateway.

The available supply chain IoT applications related to smart factories, warehouses, smart transportation and port are then addressed. The description of tactile IoT applications and AI based algorithms are also provided.

Finally, the deliverable gives an overview of the solutions to be adopted In INGENIOUS to develop the interoperable and application layers:

- the *smart IoT gateway*, integrating fixed, radio and satellite access technologies. The large amounts of data and information captured by the NG-IoT devices will be monitored, analysed, and exploited through the development and deployment of IoT platforms and applications, which will be properly interconnected by means of an interoperable philosophy;
- the *interoperable layer*, the DVL, will ensure that different M2M platforms (PI System OSIsoft, OneM2M, OM2M, Symphony, etc.) will be connected, providing the requested information to the dedicated cross-DLT layer for secure and provable recording of information across multiple DLTs solutions (Bitcoin, Ethereum, Hyperledger, IOTA);
- *next generation applications* related to situational awareness and machine learning -based predictive model applied to the supply chain processes and IoT tactile applications to be use in the industrial field.

The deliverable ends with a detailed analysis of the innovations introduced in the context of the use cases that will support their validation.

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

Table of Contents

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

List of Figures

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

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

List of Tables

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

Abbreviations

3GPP	3rd Generation Partnership Project
AES	Advanced Encryption Standard
AF	Asset Framework
AGV	Automatic Guided Vehicle
AI	Artificial Intelligence
AIS	Automatic Identification System
AMF	Authentication and Mobility management Function
AOL	America OnLine
API	Application Programming Interface
AR	Augmented Reality
CA	Certificate Authority
CCSS	Cryptocurrency Security Standards
CIA	Confidentiality Integrity Availability
CoAP	Constrained Application Protocol
CRC	Cyclic Redundancy Check
CVE	Common Vulnerabilities and Exposures
CVSS	Common Vulnerability Scoring System
DAG	Direct Acyclic Graph
DLT	Distributed Ledger Technology
DoS	Denial of Service attacks
DP	Differential Privacy
DVL	Data Virtualization Layer
E2E	End-to-End
eIDAS	electronic IDentification Authentication and Signature
eMTC	enhanced Machine Type Communication
EOA	Externally Owned Accounts
ERC20	Ethereum Request for Comment
ESCC	Endorsement System Chaincode
ETH	Ether
ETSI	European Telecommunications Standards Institute
EVM	Ethereum Virtual Machine
GDPR	European General Data Protection Regulation
GW	Gateway
HME	Homomorphic Encryption
HMI	Human machine interface
HSI	Human System Interface
HSM	Hardware Security Module
HTTPS	HyperText Transfer Protocol over Secure Socket Layer
I2C	Inter-integrated Circuit
ICT	Information and Communications Technology
IIOT	Industrial IOT
IOI	Item of Interest
IoT	Internet of Things
IP	Internet Protocol
IPC	Inter-Process Communication Gateway
ISM	Industrial, Scientific, and Medical
IPsec	IP security
JDBC	Java DataBase Connectivity
JSON	JavaScript Object Notation
KETI	Korea Electronics Technology Institute
LN	Lightning Network
LoRa	Long Range
LPWAN	Low-Power Wide-Area Network
LTE-M	Long Term Evolution for Machines

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

M2M	Machine-to-Machine
MAC	Message Authentication Code
MCMC	Markov Chain Monte Carlo
MEC	Multiaccess Edge Cloud
MIOTA	IOTA cryptocurrency
ML	Machine Learning
MLaaS	Machine Learning as a Service
MPC	Multi-Party Computation
MQTT	Message Queuing Telemetry Transport
MS	Milestone
MQTT-SN	MQTT for Sensor Networks
NIST	National Institute of Standards and Technology
NoSQL	Non-Structured Query language
NB / NB-IOT	Narrow Band / Narrow Band - IOT
NF	Network Function
NR	New Radio
NSA	National Security Agency
OCEAN	Open allianCE for IoT stANdards
ODBC	Open DataBase Connectivity
OSGi	Open Service Gateway initiative
OSI	Open Systems Interconnection
OWASP	Open Web Application Security Project
PANs	Personal Area Networks
PETs	Privacy-Enhancing Technologies
PGP	Pretty Good Privacy
PII	Personal Identifiable Information
PoW	Proof of Work
RAN	Radio Access Network
REST	Representational State Transfer
RNG	Random Number Generator
RSA	Rivest–Shamir–Adleman, asymmetric encryption algorithm
RSK	Rootstock technology
SBFT	Simplified Byzantine Fault Tolerance
SCF	Service Control Function
SDK	Software Development Kit
SDO	Standard Development Organization
SIM/eSIM	Subscriber Identity Module / evolved Subscriber Identity Module
SMC	Secure Multi-party Computation
SOAP	Simple Objects Access Protocol
SDK	Software Development Kit
SPI	Serial Peripheral Interface
SQL	Structured Query language
TCB	Trusted Code Base
TCP	Transmission Control Protocol
TEEs	Trusted Execution Environments
TETs	Transparency-Enhancing Tools
TLS	Transport Layer Security
TOR	The Onion Router
TTT	Truck Turnaround times
URLLC	Ultra-Reliable Low-Latency Communications
VHF	Very High Frequency
VPNs	Virtual Private Networks
VR	Virtual Reality
VSCC	Validation System Chaincode
WP	Work Package
WSDL	Web Services Description Language
XCMP	Cross-Chain Messaging Protocol

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

1 Introduction

1.1 Objective of the Document

This document aims at describing the state of the art and the first sketch of the proposed solution of the iNGENIOUS platform's interoperable layer and smart IoT applications, which will be addressed in the WP5 activities.

Available relevant solutions related to IoT data management technologies and supply chain IoT applications are reported, highlighting the implemented functionalities, characteristics, security and privacy aspects.

The document is considered part of the milestone MS5 and can be considered the input for the implementation and integration of the data management framework that will be deployed later on in tasks 5.1, 5.2 and 5.3 and delivered in D5.2 and D5.3.

1.2 Structure of the Document

This deliverable is organized in three main sections, briefly described below:

- *Section 2, Available Technologies for IoT Data Management*: this section describes the state of the art of IoT data management technologies available today, from M2M Interoperability to data virtualization mechanisms, over DLT interoperability to smart IoT gateway. Additionally, privacy and security aspects are deeply analyzed;
- *Section 3, Available Supply Chain IoT Application*: this section describes the state of the art of supply chain and tactile IoT applications. Then, AI algorithms are proposed and privacy and security aspects regarding their application is discussed;
- *Section 4, Ingenious Proposed Solution*: this section is the core of this document. It describes the new functionality provided and the key innovations proposed by iNGENIOUS for what concern IoT data management and IoT applications.

1.3 WP5 Scope in iNGENIOUS

Currently, the different supply chain platforms are often "point-to-point" interconnected, where the systems are put into communication through ad-hoc interfaces. Connections are not much scalable and are poorly integrated.

The Work Package (WP) 5, taking as input the overall architecture defined within WP2, focuses on IoT data management and data analytics in order to solve the poor integration among supply chain platforms. Specifically, WP5 will address the Smart IoT Gateway, the interoperability layer and the next-generation IoT applications of the INGENIOUS platform, allowing different fragmented and non-harmonised IoT data to be collected and accessed through one system only, taking also in account the relevant security and privacy aspects. The Smart IoT Gateway will support several IoT communication technologies and protocols, ensuring the routing towards different M2M platforms. The interoperability layer is the core of this WP. It is composed of a Data Virtualization Layer (DVL) and a cross-DLT layer. The DVL provides a virtual approach to accessing, managing and delivering data from multiple M2M platforms without replicating it in a physical repository. The cross-DLT layer abstracts the complexity of the underlying DLTs and acts as a standard interface between DLT networks and the higher layers. It will allow any M2M platform to interact with any DLT, enabling every supply chain actor to anchor its data to the preferred DLT without a need to understand the role of nodes, transactions, blocks and smart contracts.

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

The last commitment of WP5 will be the delivery of End-user applications and an analytic layer required by next-generation supply chain domains. This new layer will use APIs to access data received from heterogeneous sources. It will use several predictive models to obtain a holistic view or situational awareness of events and processes related to selected application areas in the logistic chain. Such information will be the input to develop specific analytics used by the end-user applications.

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

2 Available Technologies for IoT Data Management

2.1 M2M Interoperability

Machine-to-Machine (M2M) communications are the main facilitators of the Internet-of-Things (IoT) paradigm. A common complaint is the lack of accepted standards and the huge fragmentation of the IoT market due to extreme level of heterogeneity in terms of device protocols, controllers, network connectivity methods, application protocols, data formats, etc. The lack of interoperability in IoT and M2M domains is mainly caused by absence of standardization. Vendors are defining different IoT and M2M platforms, proprietary protocols and interfaces which can be incompatible with other solutions. This leads vendors to create different verticals and mostly closed ecosystems (namely "silos") implying technological lock-in of end users. From this perspective, the lack of interoperability means that end users are bound to the IoT device and/or software offered by a single provider and must stick with it, which may bring a potential risk of higher operation cost.

Based on the definition from IEEE [1] and ISO/IEC [2], the interoperability can be seen from different perspectives:

Device Interoperability: devices that want to exchange information can use different communication technologies and protocols (e.g. ZigBee, BLE, LoRa, LTE Rel.13, NB-IoT, Bluetooth, LoWPAN, 3GPP Rel.8, LP-WiFi, etc.) which requires interoperability between different types of heterogeneous devices that coexist in the M2M and IoT ecosystem. Device interoperability is then concerned with the exchange of information between heterogeneous devices and heterogeneous communication protocols as well as the ability to integrate new devices into any M2M and/or IoT platform.

Networking Interoperability: network layer interoperability deals with seamless message exchange between systems, through different networks, for end-to-end communication. Because of the dynamic and heterogeneous network environment in M2M and IoT domain, the network interoperability level should handle issues such as addressing, routing, resource optimization, security, quality of service, and mobility support.

Syntactical Interoperability: it refers to interoperation of the format as well as the data structure used for any exchanged information between heterogeneous M2M systems. An interface needs to be defined for each resource, exposing some structure according to a schema (e.g., WSDL or REST APIs). Moreover, the content of the exchanged messages needs to be serialized according to a given data format (e.g., XML, CSV, JSON, etc.). Usually the message sender/receiver encodes/decodes data using a syntactic rule, in a specified grammar. Syntactic interoperability problems arise when the sender's encoding rules are incompatible with the receiver's decoding rules, which leads to mismatching message parse trees.

Semantic Interoperability: data generated by smart objects may even have a defined and common data format, but the data models and schemas used are usually dissimilar and not always compatible. This semantic incompatibility between information models results in IoT and M2M systems not being able to dynamically and automatically inter-operate as they have different descriptions or understandings of resources and operational procedures. From this perspective, semantic interoperability ensures a common meaning of the data or command to be interpreted or executed in a proper way.

Platform Interoperability: this interoperability level arises because of the availability of diverse operating systems, programming languages, data structures and access mechanisms for smart-objects and data. This non-uniformity causes developers obtaining extensive knowledge of the platform specific APIs and information models of each different platform to be able to adapt their applications from one platform to another. A cross-platform approach is usually adopted

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

so that applications can access different M2M and IoT platforms and integrate data from various sources: cross-domain interoperability allows to federate M2M and IoT platforms operating in different domains enabling the development of new and innovative applications. On one hand these aspects lead to the definition of new requirements that future M2M-IoT platforms and architectures must meet in terms of resource control, energy awareness, quality of service, interference management and security. On the other hand, there are still relevant open challenges that need to be properly addressed: interoperability, scalability, flexibility, energy efficiency, mobility management and security. This leads to consider that the standardization is still the key to achieve universally accepted specifications and protocols for interoperability between devices, platforms and applications based on new requirements from IoT domain. Although M2M and IoT interoperability issues still persist, different H2020 initiatives have already tried to address this topic from different perspectives. In order to complete the state of the art of M2M and IoT interoperability as well as different approaches for data management, we mention some of them below (further details can be found in reference section):

- BIG IoT - Bridging the Interoperability Gap of the Internet of Things [3];
- symbIoTe - Symbiosis of smart objects across IoT environments [4];
- InterIoT - Interoperability of Heterogeneous IoT Platforms [5];
- AEOLIX - Architecture for EurOpean Logistics Information eXchange [6].

In this context, iNGENIOUS project expects to contribute to M2M and IoT interoperability topic by exploring solutions based on unified interfaces for data access and data management, also guaranteeing data integrity/immutability through DLTs native capabilities. The proposed solution will be presented and further discussed in Section 4.

2.1.1 PI OSIsoft System Platform

One of the most common tools for data processing and data streaming in heavy industrial environments is PI software from OSIsoft. In fact, this platform is the one being used in the Port of Valencia as the industrial IoT system for the port. The PI System is composed of a number of software modules that are used for data collection, time-series historicizing, finding, analysing, delivering, and visualizing. It is marketed as an enterprise infrastructure for management of real-time data and events. Data can be automatically collected from many sources, such as control systems (e.g. SCADA), industrial computers (e.g. PLCs), dedicated hardware, laboratory equipment and many others. It also allows the automatic insertion of calculations made on data and the manual entry of information. Most data are gathered using one of the many OSIsoft and third-party PI Interfaces. Users can then access this information using a common set of tools (such as Microsoft Excel, web browser, or external tools) and look for correlations.

Components of PI System

The OSIsoft PI products include the following functional components:

- *PI Data Archive*: the PI Data Archive is a time-series database of the PI System. It receives and stores the data collected by PI Interfaces, PI Connectors and PI Integrators from data sources. This module also organizes data providing an information infrastructure. It lays on the PI Server which includes tools for analytics, alerts, and auditing. The PI Server can be connected to almost any existing automation, lab, or information system. Operators, engineers, managers, and other plant personnel can use client applications to connect to the PI Server to view data stored in the PI Server or in external data archive systems;
- *PI Asset Framework (AF)*: PI AF allows the contextualization of raw data streams by adding metadata and attributes to collected information. In this module, PI users can organize

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

assets and signals following complex hierarchical structures. In fact, this module is basically used to create digital twins of sensors and objects;

- *PI Web API*: the PI Web API is the access layer that provides a RESTful interface to the PI system. The RESTful interface allows client applications read and write access to their PI Asset Framework and PI Data Archive over HTTPS programmatically;
- *PI Vision*: a simple and intuitive web-based tool for quick ad-hoc displays. Users can create powerful dashboards in minutes with PI Vision's drag-and-drop utility. With its management panel, the user can easily search and add signals to the displays;
- *PI Connectors and PI Interfaces*: PI Connectors and PI Interfaces are the components used to read or receive data from a data source, convert to a PI readable format, and send to the PI Data Archive to be stored. Unlike Interfaces, Connectors can detect all parameters and signals automatically. The PI connection points are created on the fly by the PI connectors. They also create a default structure within the AF. There are more than 400 connectors and interfaces, including: OPC, Modbus, RDBMS, and Universal File Loader (UFL);
- *PI Integrators*: PI Integrator boosts businesses' operational insight by formatting PI Server data for fast, easy, native integration into cloud platforms, advanced analytics, relational databases, geographic information systems and business intelligence tools. The PI Integrator cleanses, augments, shapes, and transmits data from PI into common business intelligence and data warehouse tools, such as Power BI, Tableau, Hadoop, Microsoft Azure, and Amazon Web Services.

Figure 1 shows the PI System deployment at the Port of Valencia. The system is deployed in four different servers, one for each of the components, in order to guarantee the scalability of the implementation.

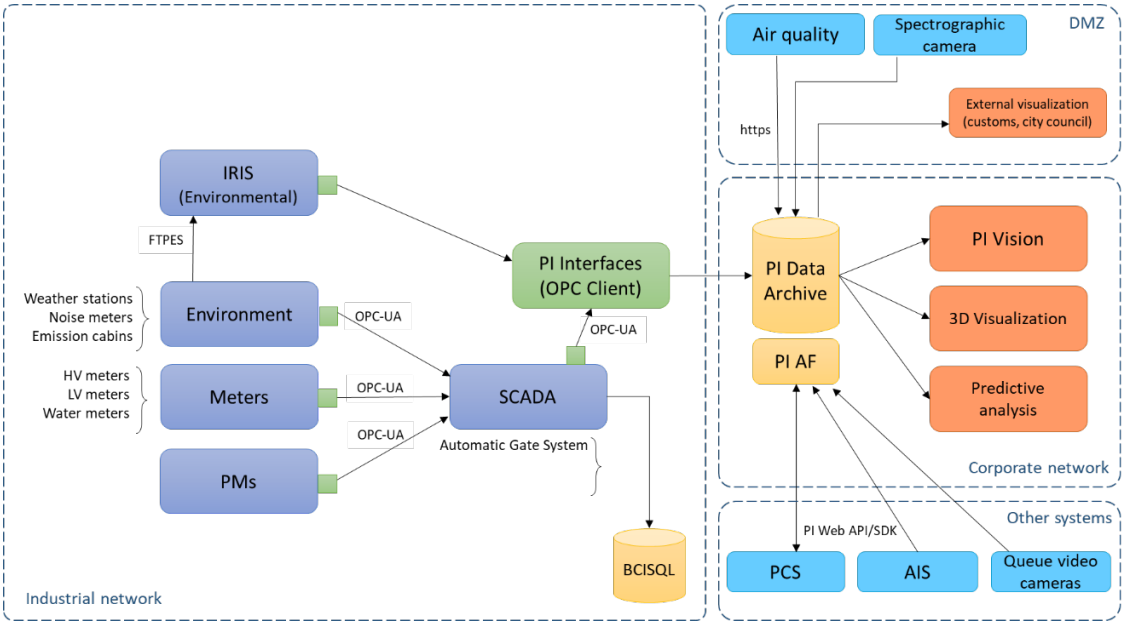

FIGURE 1: VALENCIA PORT PI DEPLOYMENT.

For the connectivity with the different ports' data sources, the PI Interfaces component is deployed in a specific server in the industrial network, as most of the data sources come from this network. There are currently some sensors and systems connected to the industrial platform for monitoring environmental data, the gate access system, electric smart meters and AIS system. On the other hand, the other 3 components of PI (I.e. PI AF, PI DA and PI Vision) are in the corporate network, where the data will be analysed and exploited. The storage layer is basically formed by the PI Data Archive, where all the data is saved optimizing the storage. For

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

the digital twin creation and real-time data processing, a virtual object is created for each of the sensors, devices, or systems and data is merged from different sources. For that the PI AF component is used. Finally, for the application layer the PI Vision component is used, which allows to create several dashboards easily including data from PI DA and PI AF. The dashboards can be customized with maps figures, machinery or even the infrastructure plan. Furthermore, the system is replicated. There is a production deployment for real use of the system and the data, and a test or pre-production deployment to introduce new functionalities, data sources, etc.

2.1.2 Mobius OneM2M Platform

The oneM2M [7] global initiative is an international partnership project established in June 2009 by the most important SDOs and various alliances and industries (e.g. ETSI, TTC, TTA, TIA, ATIS, etc.). The main aim was to define a globally agreed M2M service platform by consolidating currently isolated M2M service layer standards activities.

The oneM2M standard supports a resource-oriented architecture and adopts a layered model with different domains as illustrated in the picture below. Common service functions typically include: registration, discovery and announcement, security, groups management, data management and repository, subscription and notification, device management, application and service management, communication management, network service exposure, location, service charging and accounting, semantics and interworking:

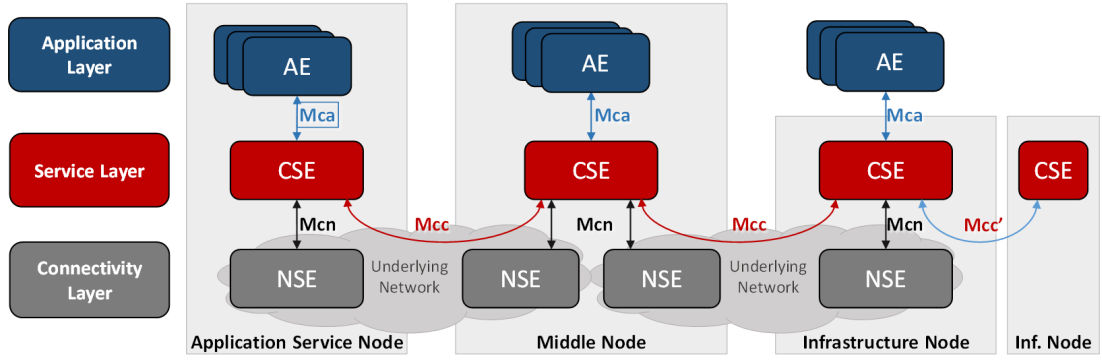

FIGURE 2: ONEM2M LAYERED MODEL [8].

Multiple protocol bindings (e.g. HTTP, CoAP, MQTT or Web Sockets) are also supported over Mca, Mcc and Mcc' interfaces.

In order to exchange resources between M2M entities, the ETSI oneM2M standard specifies procedures based on a RESTful architecture style. The RESTful architecture has indeed a set of basic methods which act on resources, referred as CRUD methods:

- *CREATE* method creates a resource;
- *RETRIEVE* method reads a resource's content;
- *UPDATE* method writes the resource's content;
- *DELETE* method deletes a resource.

Additionally, the standard also specifies two extra methods, NOTIFY and EXECUTE. The first one is used to notify any changes on a resource, and it's mapped to RESTful architecture as an UPDATE method. The second one is used to execute a management command and it is also mapped to a UPDATE method, however the request doesn't contain a payload.

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

Mobius is an open source server platform implementing oneM2M standards available in the OCEAN (Open allianCE for IoT stANdards), an open source-based global partnership project for IoT (CNIT is a developer partner). One advantage of using the public IoT server (from KETI - Korea Electronics Technology Institute) is that it is possible to use a web-based oneM2M resource monitoring application that makes it easy to monitor sensing values and actuation commands for the IoT devices in real-time. The Port Authority of Livorno in collaboration with CNIT, has employed the public Mobius-based IoT server provided by KETI in order to perform smart-objects and IoT devices management at seaport. Currently, the Mobius platform allows to manage and interact with meteorological stations, parking sensors, seaside surveillance and bathymetric services by means of a web-based interface. Moreover, the usage of Mobius platform based on ETSI oneM2M standard has been widely consolidated through R&D activities carried out by CNIT during the last years. Some good examples are H2020 projects such as *AUTOPILOT - Automated driving Progressed by Internet of Things* [9] and *COREALIS - Capacity with a pOsitive enviRonmEntal and societAL footprInt: portS in the future era* [10] where oneM2M standard platform has been exploited in the C-ITS and logistics/maritime domains respectively.

2.1.3 Eclipse OM2M Platform

The Eclipse OM2M project [11], initiated by LAAS-CNRS, is an open-source implementation of oneM2M and SmartM2M standard. It provides a horizontal M2M service platform for developing services independently of the underlying network, with the aim to facilitate the deployment of vertical applications and heterogeneous devices. The Smart IoT GW relies on the Eclipse OM2M implementation of the oneM2M standards. Essentially, OM2M implements components and applications for each of the standards described in oneM2M. Among its main features are:

- Not only compliant with oneM2M but also with SmartM2M [12];
- Open Service Gateway initiative (OSGi)-based architecture extensible via plugins;
- Implements a restful API with a generic set of service capabilities;
- Provides oneM2M services such as machine registration, application deployment, container management, resource discovery, access right, authorization, subscription/notification, group management and non-blocking requests.

It integrates with the Smart IoT GW flows and processes via its plugins and reduces the complexity of the architecture. Another oneM2M framework to be used in the implementation phase is openMTC [13]. It provides the same common interface, but it has a very convenient deployment system as containers, being aligned with the overall deployment strategy for the Smart IoT GW architecture.

2.1.4 Symphony Platform

Symphony is a comprehensive cloud-based software suite of building management tools to create a complete Building Management System (BMS). The Symphony M2M platform can communicate with any automation controls, both standard protocols and proprietary systems. It allows the monitoring and control of diverse building automation systems, by integrating different protocols under a coordinated, unified management level with an open and modular approach. The main functionalities of the Symphony platform include:

- Support for major automation standards (e.g. KNX, BACnet, LonWorks, OPC-UA, Modbus), allowing seamless interconnection of heterogeneous systems for cross-technology operation;

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

- Physical objects abstractions (e.g. a virtual sensor that is a composition of physical ones);
- Groups of (heterogeneous) objects accepting the same commands;
- Semantic information model (e.g. for lighting, curtains, lifters, HVAC, automation control) aligned with OGC SensorThings, ETSI oneM2M and OPC-UA models;
- User defined scenarios;
- Pervasive environment sensing;
- Energy monitoring and power management with customizable user defined policies;
- Security, Access Control and Anti-intrusion;
- Rule-based event management;
- Notification engine;
- Cloud platform for remote management and control;
- Data collection and storage to enable analytics;
- Hooks to attach external business intelligence services (e.g. to optimize production workflows).

Symphony control logics and system behaviors can be defined both at a local level (single system) as well as at a global level (multiple systems controlled by cloud-based services). Incorporation into the DVL allows communication with other M2M platforms and provides means of information transfer for use in the cross DLT layer for a secure exchange of information. The generic development framework of Symphony can define reactions to events which are fed to potentially complex processing rules ranging from simple algorithms to more complex decision systems, which result in actions performed by the system. Events are generated by objects (e.g. motion/presence detectors, open/close contacts), simple threshold comparators (e.g. lux sensor, temperature sensor) or processing rules themselves. Actions include specific operations on (groups of) objects, notifications, activation of scenarios, etc. A simplified graphical interface can be used as an alternative to scripting languages (e.g. LUA or Python) which are also available. An action scheduler allows programming functions with fine-grained timings or simply in a daily, weekly, monthly or seasonal fashion. The basic building blocks of the Symphony BMS are depicted in Figure 3. They include:

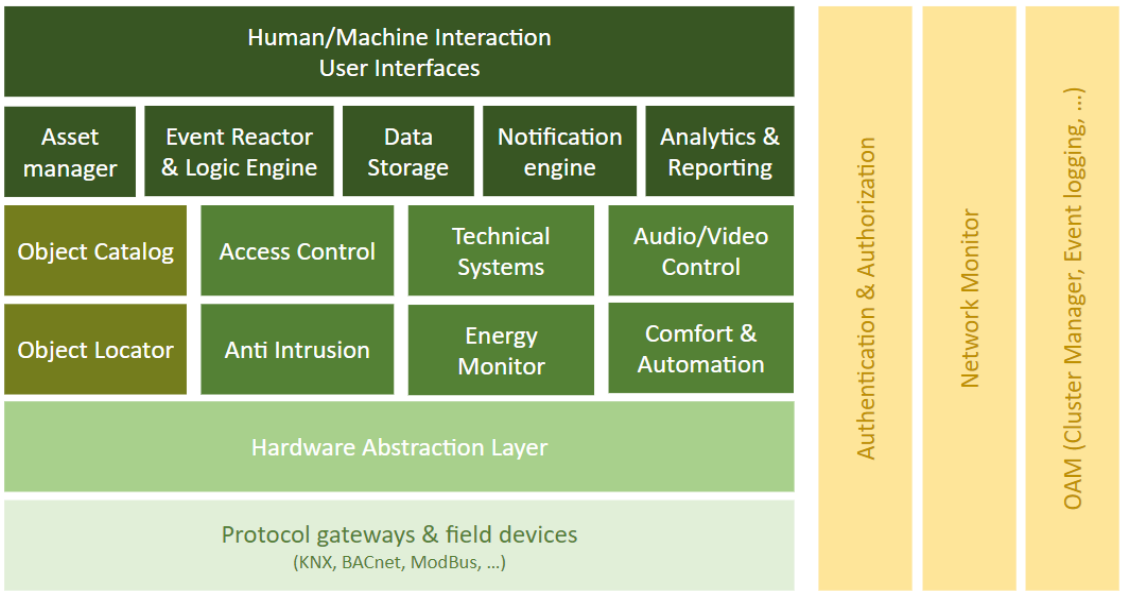

FIGURE 3: SYMPHONY BMS BUILDING BLOCKS.

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

- Low level fieldbus controller (Hardware Abstraction Layer);
- Specialized functional modules;
- Comfort automation system;
- Technical monitoring system;
- Energy manager;
- Access control / anti-intrusion system;
- Video surveillance system;
- Communications.
- Network controller;
- Event Reactor;
- Multi-tier storage system;
- Analytics & reporting engine;
- Authentication and authorization manager;
- Event reactor & logic engine to develop processes and workflows;
- Cluster manager (high availability);
- Visualization App;
- Cloud gateway;
- Cloud reflector.

These building blocks provide a complete functional stack covering the automation chain from the field bus level up to the user interface level.

2.1.5 NB-IOT Platform

NB-IOT is radio technology defined in 3GPP to provide connectivity for low power consumption devices. NB-IOT is optimized to utilize mobile signalling messages to provide data from devices, where data transfer is limited. NB-IOT is cellular connectivity enabler for higher level M2M platform that can utilize other radio technologies such as WiFi, Fixed networks, etc.

The NB-IOT platform consists of the mobile packet core components that will provide the connectivity with the NB-IOT capable Radio Access Network (RAN) and data networks where services and application servers will receive the data collected from the NB-IOT devices for the processing. The NB-IOT platform includes the mandatory network functions (NF) required for the device authentication and authorization but also to provide IP connectivity. Thus, the NB-IOT device provisioned with SIM or eSIM would be connecting through the RAN to the NB-IOT platform and deliver data over IP network to the required services processing the incoming data. Figure 4 shows the NF required to deploy a 4G or 5G mobile network infrastructure and the specific NF needed to collect the NB-IOT received in the signalling messages by the Authentication and Mobility management Function (AMF) that are delivered to the SCF which decapsulates the data and sends it to the IOT application servers.

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

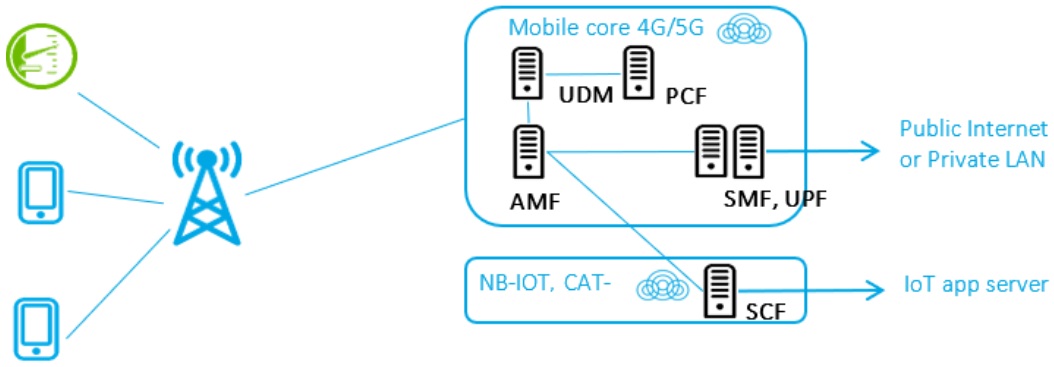

FIGURE 4: DEPLOYMENT OF A 4G/5G MOBILE NETWORK.

2.2 Data Virtualization Approach

Data Virtualization is an advanced approach to data integration. Data Virtualization can be considered as an easier way to integrate, federate and transform data coming from multiple data sources into a single and unified environment in real-time. It is not about collecting data from different data sources but connecting them and leveraging data warehouses, data lakes or different data infrastructures already in place. As opposed to traditional style of data integration (Extract, Transform and Load - ETL), the data virtualization approach allows to interact with data sources without physically centralizing them. According to Data Management Book of Knowledge [14], "data virtualization enables distributed databases, as well as multiple heterogeneous data stores, to be accessed and viewed as a single database. Rather than physically performing ETL on data with transformation engines, data virtualization servers perform data extract, transform and integrate virtually." This approach does not require users to physically move their own data and it is useful for quickly accessing new data sources for different purposes. Moreover, users do not need to know any technical details about the data, such as how it is formatted or where it is physically located. As a result, organizations are able to restrict data access to authorized users only to ensure security and meet data governance requirements. Data virtualization is performed by means of data virtualization tools that can be open-source or commercial though they all operate on the same principles:

- *Load metadata from the source*: data virtualization tools work with metadata rather than the data itself. This metadata includes details of the source data such as table structures and information related to data lineage;
- *Load views from the source*: if present, data virtualization tool can also load views from the data source without requiring an understanding of the data source structure;
- *Apply business rules*: by using metadata and physical views from data source it is possible to create logical views. Some data virtualization tools work with a query language like SQL, while other tools use a drag-and-drop interface. This way it is possible to create logical views that cover multiple data sources;
- *Make virtual data source available*: the output of the logical views can be used as data source to feed applications running on top. Data remains unchanged on the source database and the data virtualization platform outputs a virtual representation of that data (NOTE: virtual views are dependent on the original data source and if that source is lost or changed, data are removed from the virtual view accordingly).

From this perspective, a typical full featured data virtualization platform usually includes different mandatory components such as:

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

- *Data Ingestion Layer*: includes connectors to enable access to data from data sources such as big data systems (e.g. Hadoop, NoSQL, etc.), enterprise service bus, enterprise applications (e.g. ERPs and CMRs), data warehouses, cloud data systems and many others;
- *Security Layer*: provides authorization and authentication mechanisms to ensure data security, usually based on predefined roles for users;
- *Query Engine Layer*: accepts incoming queries and creates an efficient execution plan by breaking the incoming query into multiple subqueries which can retrieve data from data sources via the data ingestion layer. In order to improve the performance of the incoming queries, a caching and optimization mechanism is also supported;
- *Data Distribution Layer*: exposes the data in response to the queries. This layer usually publishes data supporting different interfaces such as Web Services (e.g. REST and SOAP), JDBC, ODBC, OData, message queues, etc.

In terms of operational scenarios and use cases, the data virtualization approach suits fine for a wide range of activities such as:

- *Analytics*: data virtualization is an easy way to bring data sources together, which makes it ideal for data analytics. Analysts have full control over the way data sources are combined, and they can adjust the structure of virtualized data without compromising the integrity of any data repositories;
- *Real-time Reporting*: since data virtualization is fast, it can offer quicker results than most data pipelines;
- *Data Exploration*: the data virtualization approach can provide a quick view of the available data allowing for speedy exploration;
- *Testing*: virtual data sources are easy to set up, hence it is useful for testing. The flexibility of data virtualization is especially helpful in high-speed testing environments.

Although the use of this approach for data management must be evaluated on a case-by-case basis, it leads to a common set of benefits that can be summarized as follows:

- *No Replication*: integrated views of data draw from multiple sources without moving or replicating it, bypassing redundant copies and reducing storage footprints;
- *Abstraction*: data is accessed without the knowledge of its location or format;
- *Real-Time Access*: the latest version of data is immediately available once it has been published on the relative data source;
- *Agility*: DVL is in principle universally semantic for the applications; it also bridges the semantic understanding of unstructured data with a schema-based approach;
- *Data Governance*: all data is discoverable through a single virtual layer which provides a unified data governance capability, supporting also data security aspects.

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

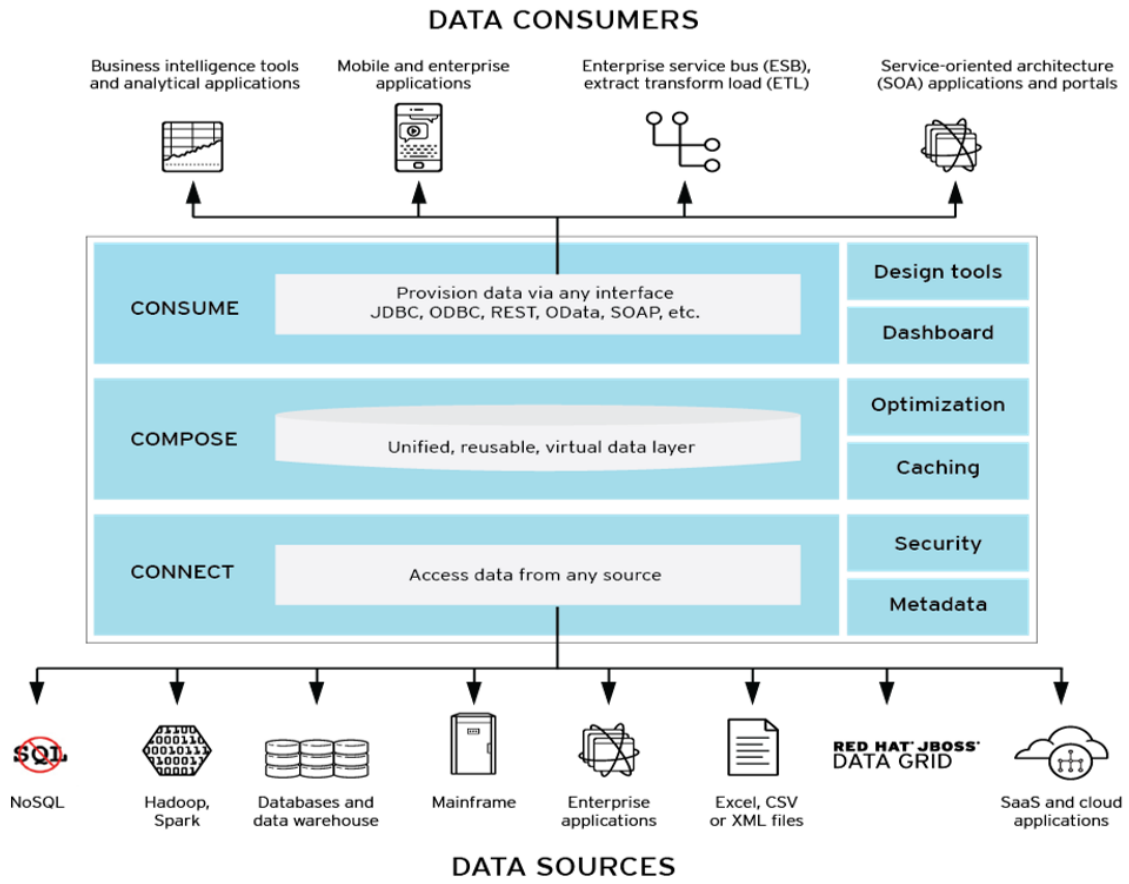

FIGURE 5: TYPICAL DATA VIRTUALIZATION FUNCTIONAL COMPONENTS [15].

Data virtualization market provides already several commercial solutions implementing tools with data virtualization capabilities, and it is expected to grow between 2022 and 2026. The current solutions come from key market players such as IBM, SAP SE, Informatica, Denodo Technologies, Oracle Corporation, TIBCO Software, Microsoft Corporation, Red Hat, SAS Institute, etc. On the other hand, also open-source solutions exist though with less and/or limited capabilities.

In iNGENIOUS project, we relied on a custom solution based on open-source data virtualization alternative, namely Teiid [16]. This solution has been already tested in maritime domain at the Port of Livorno and it is part of the overall ICT of the seaport. Teiid was used in order to retrieve bathymetric data from Port of Livorno seabed analysis. The following Figure 6 provides details in terms of data flows as well as in terms of involved architectural components:

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

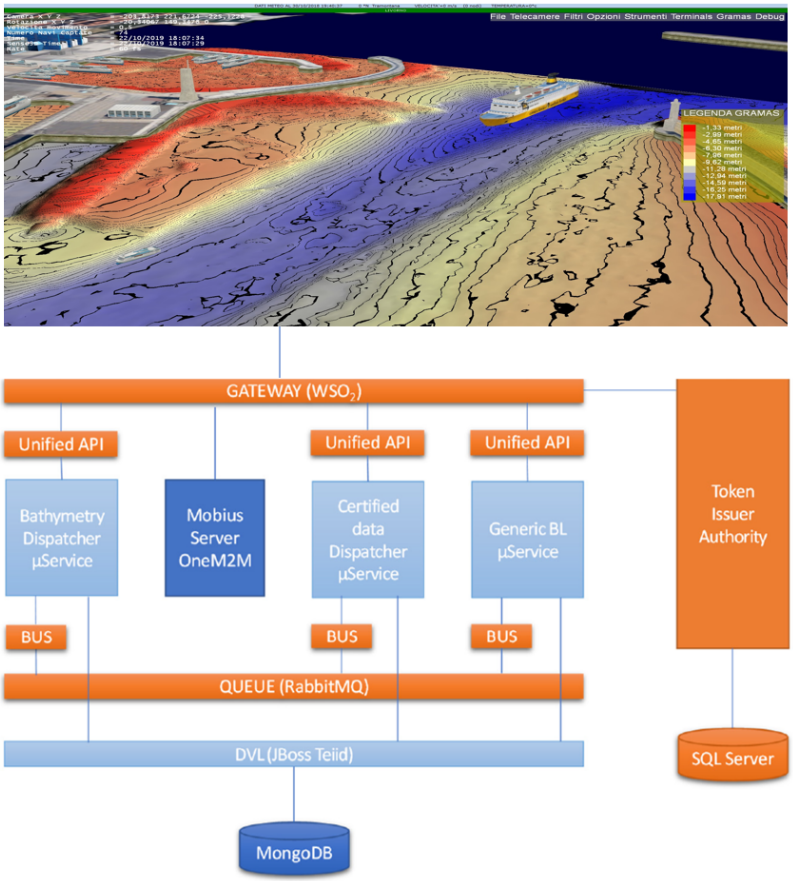

FIGURE 6: EXAMPLE OF DVL APPLICATION AT PORT OF LIVORNO.

In this context, a DVL was successfully used to store and retrieve bathymetric campaigns data for their graphical representation by means of the Port Monitoring Platform. Moreover, the seabed monitoring application addressed some relevant challenges such as i) dredging needed to accommodate new type of traffic, ii) relocation of sediments sets environmental issues and iii) fast planning needed to minimize economic and environmental impact. In iNGENIOUS project, DVL capabilities are expected to be further improved with additional functionalities such as personal data pseudonymization and network parameters analysis. Further details are provided in Section 4.

2.3 DLTs Interoperabilities

2.3.1 State of the art: DLTs

The Distributed Ledger Technologies are a new technology in which there are a set of databases or information storage instances distributed among several participants. This shared and agreed state of the whole DLT is managed by a so-called “consensus algorithm”. A consensus algorithm is the computer implementation of the rules for the system that also enforces them by rejecting any updates and data that do not follow the consensus. The general rule is that the databases are identical to each other and may be distributed both geographically and computationally. The consensus algorithm executed before any update to the ledgers is a key characteristic of each DLT. This algorithm specifies how the network is governed and defines the performance and scalability of it and its features. The main objective of this algorithm is allowing a group of network participants, that may or may not trust or even know each other, to agree on the state of the system. The original idea is to protect the system from fraud, duplicated information or malicious actions by some powerful actor. There are many different consensus algorithms, but the two most relevant are the following:

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

- *Proof of Work*: the blocks are validated using computational power. The process has the feature that in order to provide an alternative version of the block (fraud), probabilistically the same amount of energy should be consumed. This model of security assumes that a hypothetical attacker needs to have more resources than the honest participants. The blockchain-based data structures involved ensure that, the older the data, the more work that must be committed to change it (blocks are built one on another generating dependency);
- *Proof of Stake*: this method is based on the idea of "stake" - the share of the cryptocurrency associated with the DLT, that is locked as a proof of stake, and the keys associated with it are used for validation of blocks/updates to the system. The security model is based on the idea, that participants with high stake are not willing to cooperate with each other in order to redo the history or forge transactions.

2.3.1.1 Bitcoin

Bitcoin was invented in 2008 in the aftermath of the global financial crisis. Its theoretical foundations were presented in "Bitcoin: Peer-to-peer electronic cash system" by pseudonymous author Satoshi Nakamoto [18]. The Bitcoin network started its operations in January 2009. The goal of the creator was to establish an electronic cash system that would operate without the help of any financial institution. This is a daunting task to achieve the level of security comparable (or higher) to the existing financial solutions without involving intermediary trusted parties. The main assumptions and the working principles of the Bitcoin network are the following:

- There is a limited supply of the currency hardwired in the software implementing the consensus – this makes Bitcoin akin to gold;
- The information about transactions performed by users is broadcasted to every node constituting the Bitcoin network;
- A transaction in Bitcoin may have many recipients and does not contain any information about the identity of the sender. Instead, transaction refers to a non-empty set of *inputs* and a non-empty set of outputs, see FIGURE 7*;*
- Each input and output have associated to it some value in Bitcoins. The sum of values associated to inputs must be greater or equal to the sum of values associated to outputs. The difference is the *transaction fee;*
- Each input is a reference to some earlier transaction output;
- Confirmation of a transaction requires a *Proof-of-Work* and is done simultaneously for many transactions collected in a *block* at once;
- Blocks are created by *miners* who try to make the block acceptable to the network by guessing a value of a *nonce* which depends on the difficulty of the network and is also a part of the block, see FIgure 8. This process is called *mining;*
- The effort of a miner is rewarded by the transaction fees inside the block and by *block subsidy* related to minting new Bitcoins;
- A confirmed block is added at the end of the list, or *ledger,* of other blocks confirmed earlier called *blockchain,* see FIgure 8, and the miner broadcast this fact to the network;
- Each block contains also the hash of its predecessor. Since the effort to confirm new blocks is done by many miners at the same time, forks of the blockchain happen. The branch with the highest proof-of-work effort is considered as the valid one;

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

- The Bitcoin security is based on public key cryptography (elliptic curves), cryptographic hashes (sha256) and the *proof-of-work*;
- Anyone can run a *full node* on its own. A *full node* is a Bitcoin network node that validates and stores the blockchain. It allows individuals to verify the state of the network and personal balance without trusting third parties.

There are multiple aspects that should be addressed to understand the technical part of the solution. First, every output from the previous transaction must be fully spent – it is impossible to reduce the value from an already recorded transaction.

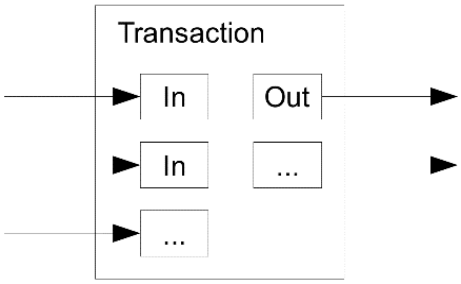

FIGURE 7: SCHEME OF BITCOIN TRANSACTION (FROM [18]).

A user can spend the funds using a private key to prove that he can move the funds. A cryptocurrency wallet (in the context of transactions) is a software that manages multiple private keys and tracks transactions with unspent outputs that can be unlocked by these keys (belong to the owner). Valid transactions are broadcast to the known peers, that verify its correctness, and send them as broadcast messages, so every node in the P2P network can see the transaction. This unconfirmed transaction is added to the set of other unconfirmed transactions stored in the memory pool (mempool). Miners pick unconfirmed transactions from mempool and try to find the block in the process called mining. Once the block is found, it is broadcasted, checked by every node that sees it, and if correct, added to the blockchain. Modification of any transaction in the block would require updating each subsequent block in the ledger because they refer to the hash of the previous block. This renders any attempt to cheat the system by double spending extremely expensive, as one node needs to "prove more work" than the rest of the network, which is mostly unlikely.

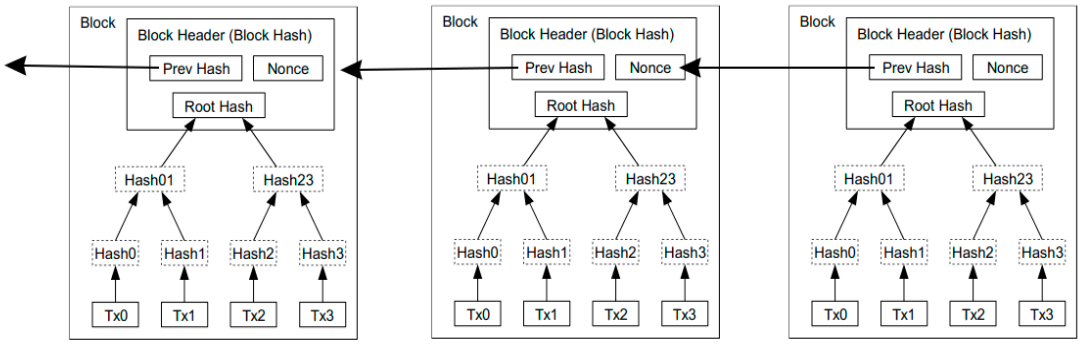

FIGURE 8: STRUCTURE OF BITCOIN BLOCKCHAIN (FROM [18]).

Mining

Immutability of the blockchain is secured by the Proof-of-Work. The process, in the case of Bitcoin, is performed by generating SHA256 hashes of the block's transactions. These hashes should meet a specific criterion, defined by the difficulty of the network at that time. It is similar to the process of inversing the hash: the brute force guessing. The difficulty of the process is adjusted to reach an average of 10 minutes per confirmation of block by the whole network.

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

The consensus is that the valid blockchain is the one that is the longest in terms of the cumulative proof-of-work done. This way, if most miners are honest, the blockchain can grow. From the perspective of iNGENIOUS, even if the entire world stops mining, the content of the Bitcoin blockchain would contain an immutable history of events. The miner that mined a block broadcasts it to the network. Each node in the P2P network checks if the block is valid (no double spends, sufficient difficulty, and so on) and if it is, attaches it at the end of the blockchain. The miner can claim transaction fees and block reward. The block reward is divided by 2 every 4 years (or more precisely every 210000 blocks), as depicted in FIgure 9. This event is known as halving.

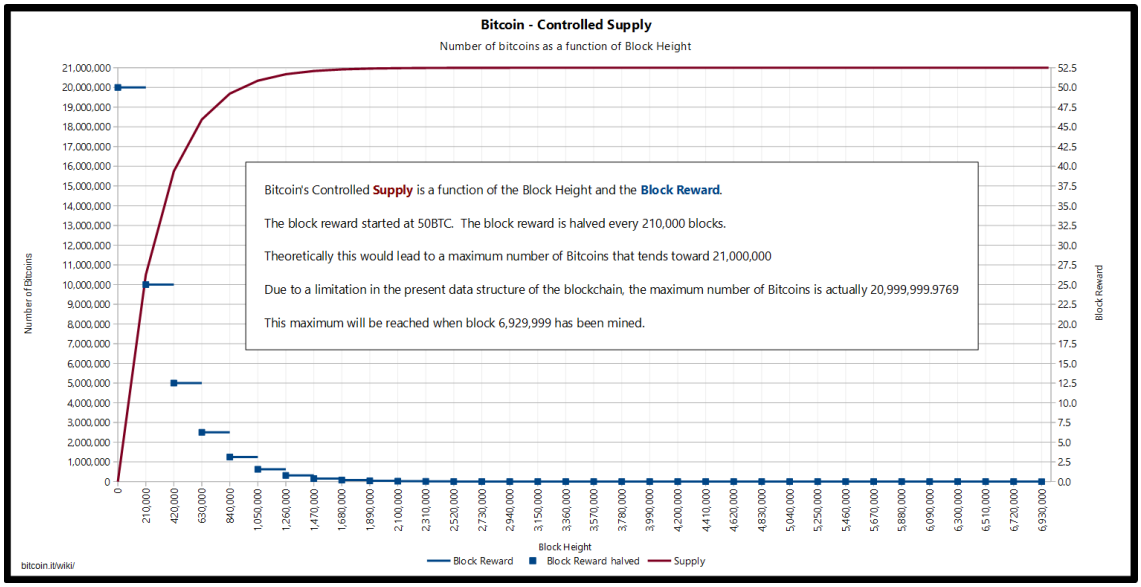

FIGURE 9: BITCOIN SUPPLY SCHEDULE.

The Bitcoin privacy model assumes that the transactions do not contain any personal data, they are determined by their inputs and outputs, see FIgure 10. The transactions are public. Unless the owner of the transaction reveals itself, there is no direct connection between the person and the funds. Nevertheless, there are situations in which privacy may be compromised.

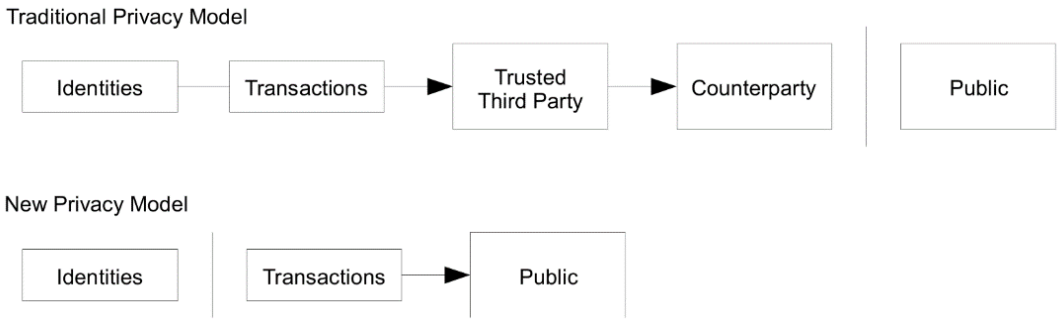

FIGURE 10: COMPARISION OF PRIVACY MODELS (FROM [18]).

Smart Contracts

Bitcoin supports smart contracts. It uses so-called Script [23] that is based on postfix notation (Reverse Polish Notation) and operates on stack. Script combines two parts:

- *scriptSig*: the first part of the script that must be provided in the spending transaction;
- *scriptPubKey*: the second part of the script that locks the funds on the transaction.

This construction of transactions allows for some smart contracts. They are limited, because there are no loops, or variables available, but robust enough to allow many real-life smart

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

contract scenarios like oracle, multisignature, or escrow, see [22]. A more expressive smart contract language called Solidity is now available on a sidechain to Bitcoin called RSK.

Security

Blocking transactions from appearing on the blockchain is theoretically possible, but unlikely in practice – it is more profitable to collect *transaction fees*. Transaction reversal is also hard. To rewrite the history of transactions the attacker would have to have more computing power than the rest of the network. The higher the price, the more tempting it is to keep mining to get the block reward. Currently the block subsidy is 6.25BTC, that is around 334 000 EUR (as of 2021-04-14). There are *transaction fees* that also go to miners. As a result, the global hashing rate keeps growing and is estimated to be 152.467 EH/s (according to the blockchain.com). The mining process is the element that connects the inner mechanics of the Bitcoin network with the real world. Energy is transmuted into security. The permissionless system forces some tradeoffs, like the size of block, that is limited in order to allow more participants to verify the consensus rules. In order to increase transactional throughput, there are already working solutions based on sidechain or offchain transactions. One of the most notable developments is the Lightning Network. It has almost no upper limit on transaction throughput. The idea was first described in [19], and right now it is a working solution incorporated by multiple exchanges, wallets, and payment processors. For comprehensive explanation, see [20]. The Lightning Network is not itself a blockchain, nor does it produce a blockchain. It is a network that relies on an existing external blockchain for its security. Typically, a Lightning Network user will run a Lightning node and a Bitcoin node. It provides extremely fast, permissionless, secure and private transactions that don't involve the blockchain. It is based on the idea of payment channels that can be described as a shared wallet with two balances. Opening a channel involves onchain commitment transaction. Sending funds is moving from one side of the channel to the other. The Lightning Network consists of multiple payment channels and a cryptographic protection against fraud in the case of multi-hop payments. The payment channel can be closed in three ways – mutual close, force close, and protocol breach (prevent fraud). When the channel is closed, an on-chain transaction takes place returning every part the remaining on their bill.

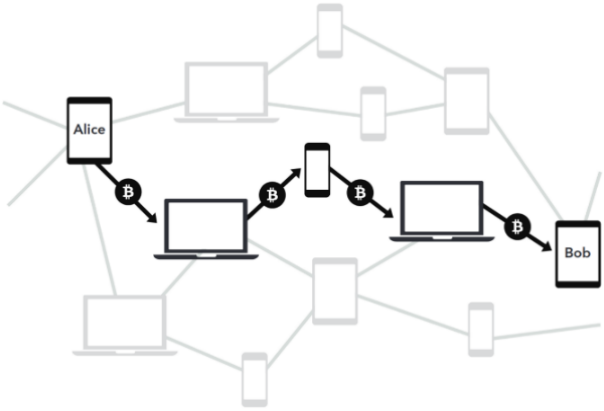

FIGURE 11: LIGHTNING NETWORK [21].

Current state of the Bitcoin ecosystem

The known mining pools mostly operate in China, but miners operate in a more decentralized manner. There are metrics that try to gather information about mining pools and their hashing performance. Historically there was one time that one mining pool reached more than 51% of total Bitcoin hashing power (GHash.io). Right now, the hashing power is more distributed, and there is an increasing share of unknown and other miners/mining pools. However, this is the legitimate concern. Individual miners can migrate between different mining pools. Full nodes are in different countries. Mostly in the North hemisphere. According to [215] there are 9573

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

listening nodes as of March 2021. The number of users of Bitcoin can only be estimated. The report provided by Crypto.com [210] estimates that there are more than 100M users worldwide that use cryptocurrencies, and around 71M Bitcoin users. The number is estimated using mostly the data from cryptocurrency exchanges and analyzing on-chain transactions.

2.3.1.2 Hyperledger Fabric

Hyperledger is an open source collaborative effort created in 2015 for the development of DLT technologies in order to advance cross-industry blockchain technologies. This collaboration, organized by the Linux Foundation, includes leaders in banking, finance, Internet of Things, manufacturing, supply chains, and technology.

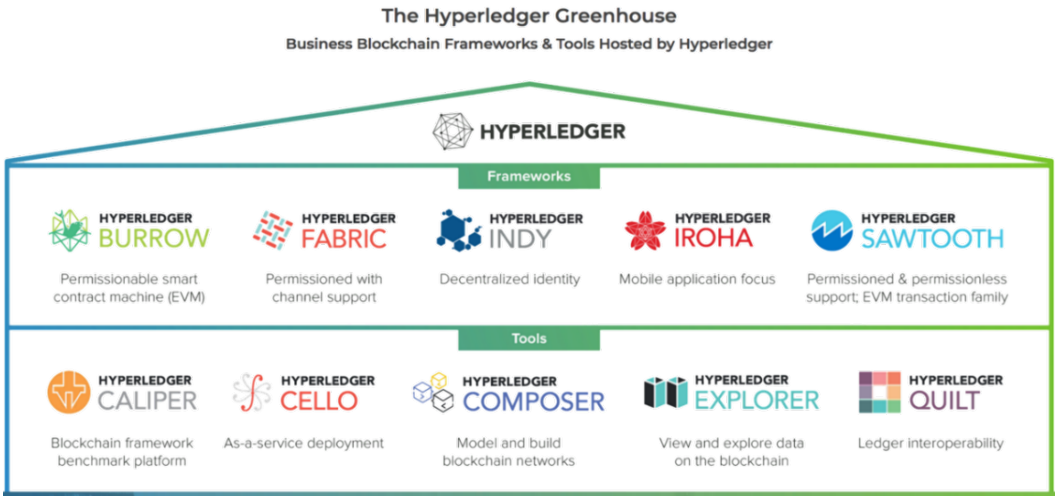

FIGURE 12: HYPERLEDGER BUSINESS FRAMEWORK.

As shown in FIGURE 12, the Hyperledger approach is structured in two types of initiatives: (i) frameworks, which are differentiated approaches used to build enterprise blockchains for a consortium of organizations; and (ii) tools, which help to design, prototype, and extend blockchain networks for a specific framework. Among the existing frameworks, Hyperledger Fabric is considered as one of the most suitable platforms for building a modular, scalable, and resilient solution for a broad set of industry use cases.

Hyperledger Fabric is an enterprise-grade, distributed ledger platform that offers modularity and versatility through the use of different plug and play components, such as consensus, privacy and membership services. The first Hyperledger Fabric version v1.0.0 was released in July 2017 and since then multiple updates have been performed in order to enhance Hyperledger Fabric performance. Currently, the last version available is version 2.3.2. The key technical features of Hyperledger Fabric are:

- *Permissioned architecture*, where the participation is selective depending on the identity of the users, which is used for establishing certain degrees of trust. In these conditions, users generate agreements through consensus, which in Fabric can be reached according to various protocols such as Kafka or RAFT. In these conditions, Fabric operates under a shared governance model where the members of the network enrol through a trusted Membership Service Provider;
- *Highly modular architecture* with pluggable ordering services for establishing consensus, pluggable membership service providers, pluggable endorsement and validation policy services, and optional peer-to-peer gossip services. Additionally, in Fabric, ledgers can be stored in multiple formats according to different databases like LevelDB or CouchDB;
- *Privacy depending on the needs of the network*, Fabric enables the use of channels allowing a group of participants to create a separate ledger of transactions. This option is especially

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

important for networks where some participants might be competitors and do not want every transaction they make known to every participant;

- *Smart Contracts* are used to execute business logic that generates new facts to be added to the ledger. In Hyperledger Fabric, smart contracts are written in chaincodes and are invoked by an application external to the blockchain when that application needs to interact with the ledger. Chaincodes can be implemented in several programming languages like Go, Node.js, and Java;
- *Execute-order-validate architecture* is used instead of the traditional order-execute approach within the lifecycle of a transaction. Thanks to this approach, Fabric lets transactions execute before the blockchain reaches consensus on their place in the chain. As a consequence, Fabric adds scalability and allows flexible trust assumptions, avoiding the sequential execution of transactions and the deterministic nature of the order-execute approach.

Regarding the reference architecture of Fabric networks, the main components of the network are the ones shown in FIGURE 13 and explained below:

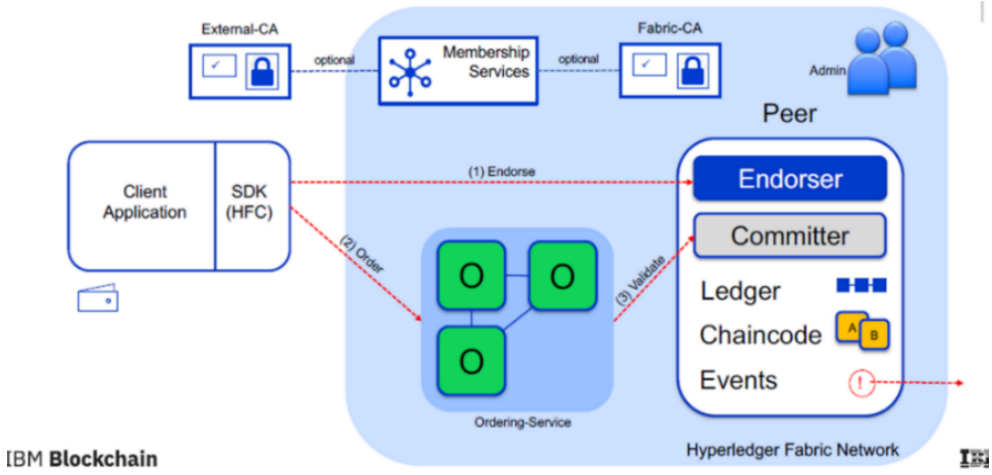

FIGURE 13: HYPERLEDGER FABRIC ARCHITECTURE [211].

- *Ledger*: book or database that contains the state of a business as a journal of transactions. Hyperledger Fabric ledger consists of two related parts: (i) the world state, which represents the current values of all ledger states, and (ii) the blockchain, which contains the transaction log structured as interlinked blocks where each block contains a sequence of transactions. Each transaction represents a query or update to the world state;

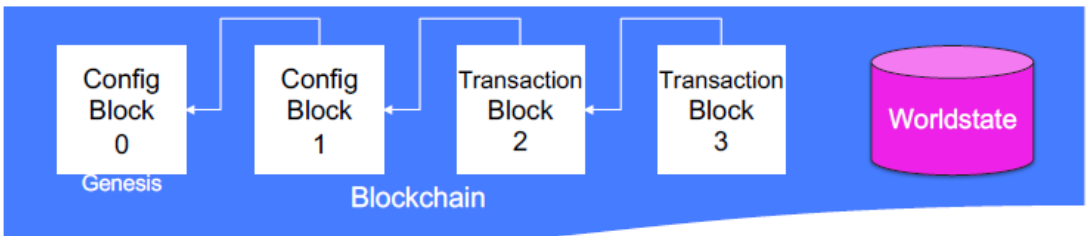

FIGURE 14: HYPERLEDGER FABRIC LEDGER [211].

- *Smart Contracts*: as explained above, smart contracts are a piece of code that defines the executable logic of the business and that generates modifications to the set of key-value pairs in the world state via transactions after being invoked by a client application. In Fabric, smart contracts are called chaincodes, and there are two types: (i) system chaincodes, which handle the system-related transactions such as lifecycle management and policy configurations, and (ii) application chaincodes, which manage different application states on the ledger;

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

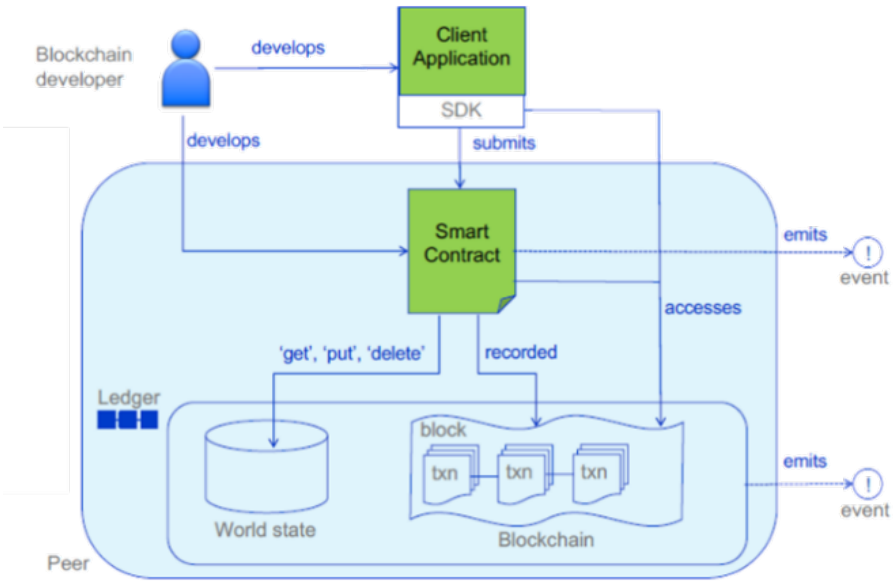

FIGURE 15: HYPERLEDGER FABRIC SMART CONTRACTS [211].

- *Client Applications* and *Fabric SDK* are used by clients to generate transactions, receive events from peers, and for connecting over different channels to one or more peers and orderer nodes. They can be written in different languages such as Node.js, Go, Java or Python;
- *Membership Service Providers* are needed to manage the set of identities within the Fabric network. MSPs provide identity, authentication, validation, signing and issuance to peers, orderer nodes, client applications and administrators. Identities can be issued by the Fabric Certificate Authority (Fabric-CA) or by an external CA, and a network can include one or multiple MSPs (typically there is one per organization);

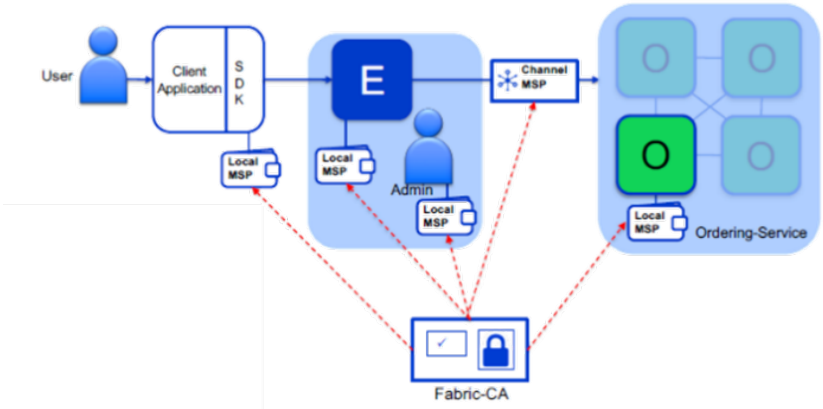

FIGURE 16: HYPERLEDGER FABRIC MEMBERSHIP SERVICE PROVIDER [211].

- *Nodes* are the main component of the network. They can be created, started, stopped, reconfigured, and deleted. Unlike a traditional blockchain node, they must have an administrator for proper management. The nodes are provided by the organizations belonging to the consortium that operates the deployed Hyperledger system and are identified with digital certificates. The nodes can be of two main types:
    - *Ordering nodes* are used to approve the inclusion of transaction blocks into the ledger and communicate with committers and endorsing peer nodes. Basically, they order the transactions, group them into blocks and spread them. Ordering nodes are a modular service that can support different consensus protocols;

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

- *Peer nodes* are used to store and maintain one or more ledgers (participating in several channels), and to manage smart contracts for status modification. The different roles that peers can play are:
  - *Committing peer*: All nodes of a channel are committing peers since they receive transaction blocks and validate them before incorporating them into their copy of ledger;
  - *Endorsing peer*: Nodes that have one or more chaincodes installed and that receive a transaction proposal for endorsement, responding by granting or denying the endorsement;
  - *Leader peer*: Node in charge of distributing the transactions of the channel to the other members when there are multiple nodes in the organization;
  - *Anchor peer*: When a node has the need to communicate with another node in another organization, it can use one of the channel's anchor peers that are configured for that organization. This role is optional.

It is important to highlight that the same ordering service node is used by all channels while peers and clients are different for each channel.

- *Endorsement Policy* is used to describe the conditions by which a transaction can be endorsed. A transaction can only be considered valid if it has been endorsed according to its policy. Endorsement policy is composed of the Endorsement System Chaincode (ESCC), which signs the proposal response on the endorsing peer, and the Validation System Chaincode (VSCC), which validates the endorsement.

After describing the different components, it is important to understand how transactions and consensus take place in Fabric. Consensus is the process comprising the entire transactional flow that serves to generate an agreement on the order for confirming the correctness of a set of transactions constituting a block. As previously mentioned, Fabric offers several consensus protocols like Kafka or SBFT. Based on consensus, the transaction process takes place exploiting the execute-order-validate approach. In particular, the following steps are usually taken when processing transactions:

- Client applications propose transactions for a specific chaincode and endorsing peers receive this proposal;
- Endorsing peers execute the propose transaction capturing the set of Read and Written data (RW sets);
- RW sets are signed by each endorser and returned to the client application;
- The client application responds with a transaction to be ordered;
- Ordering nodes receive transactions and run the ordering service, which turns transactions into blocks for distribution to committing peers;
- Committing peers receive and validate transactions against their endorsement policy and check that RW sets are still valid for the current state. Transactions are written to the ledger and update the caching DBs with validated transactions;
- Client applications are notified by each peer when transactions succeed and when blocks are added to the ledger.

2.3.1.3 Ethereum

Ethereum is a public blockchain with Smart Contracts functionality originally thought for extending the capabilities of other public blockchains like Bitcoin and Mastercoin. Its native cryptocurrency is the Ether (ETH). It was designed in 2013 by Vitalik Buterin and Gavin Wood as

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

a general purpose blockchain, so any developer could develop his specific idea on top of it. Some years later, on July 30 of 2015, Ethereum's first block was mined. Ethereum development was planned over a series of stages, maybe the most famous of them is the DAO (decentralized autonomous organization) event, when a hacker exploited the vulnerabilities of a set of Smart Contract and stole US $50M. At that stage there was a hard fork of the chain evolving it into two different networks, Ethereum and Ethereum Classic, in order to reappropriate the affected funds of the users. It is a distributed Turing complete state machine that tracks the state of the memory instead of tracking of a currency. That storage is used to store both data and code. The Smart Contracts are code stored on the chain, they are programs developed in a specific programming language that will run in the network nodes and interact with them, being capable of altering the memory state and access it. Ethereum works as other public blockchains like Bitcoin, basically it is a set of nodes (Ethereum clients running on a computer with Internet connection) working as a network following specific rules to reach a consensus about the state of the chain. Every node has a copy of the chain and every change is transmitted to the nodes after reaching the consensus. This chain and this consensus are maintained by some nodes called "miners" that incorporate new transactions and blocks to the network and receive a reward for that in Ethers. Currently, the consensus algorithm is Proof of Work, but it is planned to move to Proof of Stake in order to improve the scalability of the system. This change to Proof of Stake will come with Ethereum 2.0. As in some other blockchains, in order to interact with an Ethereum blockchain, an identity (digital signature) is required to operate on accounts. There are two kind of accounts: externally owned accounts (EOA) that are secured by the use of a private key, and smart contracts accounts that are controlled directly from the contract itself. Both of them have an account balance with the number of ethers in possession of the owner to interact with the network. Every account has the following fields:

- *Nonce*: in the case of an EOA this number is the number of transactions sent. If the account is a smart contract account, it is the number of contracts created by the account;
- *Balance*: the amount of ether the account has in weis (10^{-18});
- *Storage root*: root of the Patricia Merkle Tree of the transactions.

Smart Contracts were first described as a set of promises, digitally specified, that are kept automatically thanks to the implementation of rules and protocols. It is a program composed of variables, methods and events that has its own memory instance. Once the contract is deployed it receives a contract account, this identifies the contract and allows the users or other contracts to interact with it. There are different programming languages for implementing a Smart Contract, but the most popular and used is Solidity. Regarding the network itself, there is a main net for Ethereum where the production projects are deployed, and the transactions have a real economic value. However, there are other networks that have no real economic value and are only meant for testing and developing purposes:

- *Kovan*: works under a Proof of Authority consensus algorithm and it is only compatible with Parity clients;
- *Rinkeby*: a blockchain based on Proof of Authority algorithm, as Kovan, and only compatible with Geth clients;
- *Ropsten*: Proof of Work consensus algorithm blockchain and compatible with Geth and Parity;
- *Görli*: Proof of Authority-based testnet, compatible with most of the clients.

These are the public networks implementing Ethereum, but a private network can be implemented and deployed too where there is needed permission for accessing it. As mentioned before, there are many clients implementing Ethereum nodes. These clients are necessary in order to interact with Ethereum networks. A client allows to perform all the operations related

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

to the network like: sending transactions, compiling and deploying a contract, storing the account's balance, block mining, etc. The most popular clients are:

- *Geth*: implemented in Golang by the Ethereum Team;
- *Parity*: implemented in Rust by Ethcore;
- *Pyethapp*: implemented in Python also by the Ethereum Team.

2.3.2.4 IOTA

The term IOTA was introduced in 2015 by David Sønstebø, Sergey Ivancheglo, Dominik Schiener, and M. Sergueï Popov. It refers both to the cryptocurrency (MIOTA) and the underlying protocol, including consensus mechanism and transaction generation [24]. The main goal in launching IOTA was to create a crypto-based solution to address IoT requirements and use cases due to a huge amount of smart and connected devices that require cryptographic capabilities.

IOTA Network – Tangle

From architectural point of view, IOTA uses a novel invention called Tangle which is based on Direct Acyclic Graph (DAG), where each node is a single transaction. The Tangle is a distributed ledger architecture though it differs from the standard blockchain solutions. Although the Tangle is a blockchain without blocks and the chain, it still has the same underlying principles as a blockchain: it is still a distributed database, it is still a P2P network and it still relies on a consensus and validation mechanism. If we refer to the blockchain in terms of graph, every vertex is a block with a lot of transactions and edges go from the current vertex to the previous one. Instead, in the Tangle each vertex is a single transaction referenced by two past transactions. This is considered as an approval since with our transaction we attest that those two transactions, as well as all the transactions referenced directly or indirectly by them, are valid and conform to the protocol's rules. In other words, the edge set of the Tangle is obtained in the following way: when a new transaction arrives, it must approve two previous transactions. This is the reason why IOTA does not need miners. To issue a transaction, users must work to approve other transactions and therefore contribute to the network security. For this reason, IOTA transactions are feeless. The actual fee is the work that each user must do to approve other transactions during issuing process. There is also the so called "genesis" transaction which is approved either directly or indirectly by all other transactions. Of course, we can also be sure that nodes check if the approved transactions are not in conflict with each other. If a specific node realizes that a transaction is in conflict with the Tangle historical status, the node will not approve the conflicting transaction in either a direct or indirect manner. Moreover, when a transaction receives approvals, it is accepted by the system with a higher level of confidence: it will be difficult to make the system accept a double-spending transaction. By using a standard nomenclature, the transaction that does not have a successor yet is called "tip" (a kind of leaf in a graph). In order to attach a new transaction to the Tangle, a user has to randomly choose two tangle tips validating them (based on Markov Chain Monte Carlo - MCMC selection algorithm). This validation means that the user should:

- check the tip's signature;
- check the tip's Proof-of-Work;
- make sure the tip is not in conflict with previous transactions.

Tangle transactions occur asynchronously, and the nodes do not see the set of all transactions. This implies, as already mentioned, the possibility of having conflicting transactions. Currently, the main rule used to decide between conflicting transactions is to execute the tip selection algorithm several times so that it is possible to identify which one is more likely to be indirectly approved by the chosen tip (the branch with higher probability is chosen while the other one is

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

abandoned). We can summarize the process to create a transaction in terms of record in the IOTA ledger as follows:

- The transaction is signed with the private key of the requester;
- The transaction is linked to two previous transactions in order to be validated. This procedure is known as tip selection and it is based on Markov Chain Monte Carlo (MCMC) algorithm. During this selection procedure the node's weight and depth are considered;
- After the nodes selection, the requester (who emits a transaction) needs to perform the Proof-of-Work in order to be able to put his transaction in the list of pending ones;
- Finally, the transaction is broadcasted in the network to be validated.

IOTA Network - Nodes

IOTA network consists of different nodes that interact with each other allowing to perform specific functions across the network:

- *Full Node*: it is a standard participant in the IOTA network that distributes transactions. This node does not store the full transaction history of the Tangle but only the current state. All full nodes also help the network by accepting transactions from other full nodes, validating those transactions, and then relaying them to further full nodes. From this perspective, the full node contains part of the Tangle graph and it is able to provide tips to other nodes such as Light Nodes. When a snapshot occurs, most of the graph can be deleted from the node;
- *Perma Node*: it is responsible for keeping track of the whole transaction history of the Tangle. For this reason, it requires a huge amount of memory space (the IOTA foundation runs this kind of node);
- *Light Node*: this node does not store transactions or propagates them, it just gets the information needed from either the Full or the Perma Node. This node does not contain any Tangle graph. They just perform the Proof-of-Work for their own transactions but ask tips to a full node;
- *Coordinator*: this node is run by the IOTA foundation to protect the network. This node generates so called "milestones" which are just normal transactions. If a transaction already stored in the tangle graph is indirectly approved by the last "milestone", we can consider it as confirmed.

IOTA Network - Miners

Within the IOTA network there are no miners. If a node wants to send a transaction to the Tangle, it has to perform the Proof-of-Work on its own. In comparison to Bitcoin technology, the Proof-of-Work in IOTA is far less difficult than the one computed by miners in Bitcoin just because it is more similar to hashcash. The Proof-of-Work difficulty can be freely managed by the issuer of the transaction. Transactions with higher Proof-of-Work have a lower average confirmation time: all transactions that the node wants to broadcast to its neighbors are put in a priority queue and this priority is given according to the transactions' Proof-of-Work. The only cost that a node has to pay to issue the transaction is that of electricity, therefore it is not relevant.

IOTA Network - Smart Contracts

Although IOTA does not natively support Smart Contracts, the release of the IOTA Smart Contract Protocol (ISCP) derived from Qubic project [25] is allowing to explore the usage of Smart Contracts in a form of the Quorum-based Computations. The protocol is based on UTXO Digital Assets model [26]. In order to create a Smart Contract, a Digital Asset needs to be created and then sent to the address of the above-mentioned Smart Contract. The generated transaction is then considered as the origin transaction (Genesis) of the Smart Contract. This process does

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

not require any additional fees to be charged. The Digital Asset remains active for the whole Smart Contract duration, remaining at the address of the considered Smart Contract. In order to create a Smart Contract, we need at least two IOTA tokens (MIOTA). We also need to incentivize nodes to execute the code in a Smart Contract and this makes charges necessary. Of course, it is not necessary to execute, from network perspective, every Smart Contract code as for the case of ETH but only a limited group of nodes can be selected to execute a specific Smart Contract and eventually be rewarded accordingly.

IOTA Network – Scalability

In the Tangle-based architecture, each transaction can be represented by a vertex. As mentioned, each transaction has to approve at least two other transactions. This means that every vertex has to be connected by directing edges to at least two other vertices. A transaction is said to be indirectly approved by another transaction if there is a connection between those two transactions with at least on transaction between them. Instead of storing transactions in blocks with a limited size (as for the case of blockchains), each transaction lives on its own and has to approve other transactions. This way, the number of transactions that can be handled in a certain amount of time increases with the number of transactions: there is no waiting time between the issuing of one transaction and that of another one. Since consensus runs in parallel (not done in sequential intervals of batches as in blockchains), the network is able to grow and scale dynamically with the number of transactions. Every transaction confirms two other transactions, therefore the more transactions and nodes are in the network, the quicker transactions are verified. That is why IOTA distributed ledger technology suits fine for the IoT domain if compared to other available blockchain solutions.

2.3.2 State of the art: cross-DLT

Cross-DLT interoperability is mostly based on the asset exchange between different DLT solutions. There are multiple methods that are generic and well described, For example [Herlihy, M.: Atomic cross-chain swaps. In: Proceedings of the 2018 ACM Symposium on Principles of Distributed Computing, pp. 245–254. ACM (2018)]. The atomic swap allows for two parties that don't trust each other to exchange one cryptocurrency into another using hash and time locks. There are solutions that allow for blocking funds in one DLT in order to unlock it in another. We will not discuss these techniques in detail, but we will focus on two selected solutions that implement the cross-DLT interaction.

Polkadot

Polkadot is a network protocol to exchange data between two or more blockchain networks allowing the existence of multi-chain applications. It is designed to support from the classical public permissionless blockchains like Ethereum or Bitcoin, to private blockchains. Polkadot is based on the following stack of technologies:

- Substrate. A framework developed by Parity technologies for building blockchains;
- WASM (WebAssemply). Polkadot's state machine is compiled to WebAssemply as the runtime;
- libp2p. A peer-2-peer network framework that enables peers communications and peers discovery;
- As runtime environment, Polkadot is coded in Rust, C++ and Go.

Polkadot uses a sharded model which improves scalability and performance but having security as a drawback. Its Polkadot shard, named parachain, allow its own sequential transactions. Having more than one parachains allows to have parallel transactions. Each parachain has its state transition function and its network maintainer receives the name of collator. It will maintain the full-node, all the necessary information and the blocks sending process to the main

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

chain. This main chain is called Relay Chain. Blocks are proposed by the parachains' collators to the Relay Chain validators, then the blocks go under availability and validity checks to finally being added to the main chain. With the purpose of allowing other blockchain networks to interact with Polkadot network protocol, like Bitcoin or Ethereum, it has the parachains bridges. This component enables the transactions flow between different parachains. In order to achieve this, a specific protocol named XCMP (Cross-Chain Messaging Protocol) is used so the different parachains can interact with each other but with the security and trust that the Relay Chain offers. There are three possible ways to implement a bridge depending on the characteristics of the blockchain where it will be used. If the blockchain is a substrate native chain, the Substrate Pallet must be used. An example of a bridge communication using this approach is between Kusama and Polkadot. The second way to implement a bridge is using a Smart Contract, this can be used only on blockchains that support the Smart Contract functionality like Ethereum or Tezos. Bridges based on Smart Contract are composed of a set of two contracts, one on each chain, which allows the transfer of value. The last option to implement a Polkadot's bridge is using an external High-Order protocol just in case the other two option does not fit in the chain, the case of Bitcoin.

Hyperledger Cactus

Hyperledger Cactus is an incubation project inside the Hyperledger ecosystem and aims to achieve a decentralized, secure and adaptable integration between blockchain platforms. It is a framework that tries to solve the fragmentation problem which is present in blockchain technologies. Cactus has an extensible plugin-based architecture in order to connect as many blockchain networks as possible regardless of technology restrictions. Its developers call it the SDK of SDKs. New functionalities can be added by implementing new plugins. It provides a set of libraries and SDKs to assist the development of integrated solutions. The main advantages of Hyperledger Cactus are:

- Allows the connection and transactions across different blockchain networks with a global approach;
- Provides a standardized interface which enables secure transactions across different protocols;
- Abstracts the app layer from the technology complexity and used protocols promoting the replacement of the underlying blockchain platform;
- No tokens are required to send transactions;
- Good scalability by combining the transactions of the different networks.

Currently it provides connectors with Hyperledger Fabric, Hyperledger Besu, Quorum and Corda. The main supported tasks are read/write ledger, sign transaction and verify transaction. Its use is indicated when combining data from different blockchain networks generates value. It is possible to issue a transaction that is verified by both ledgers. Broadly speaking, the transaction flow is the following:

- The transaction is proposed and sent to the two plugins;
- It is checked that the execution of the transaction is valid in both nodes of the two different ledgers;
- The transaction is executed in both nodes;
- It is checked that there have been no changes between steps 2 and 3. If any, a rollback is performed on both nodes.

Another advantage of Cactus is that the protocol supports transactions between permissioned and public networks and batch validations when possible. Cactus provides a set of validator

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

nodes which ensure the proofs of the state of the connected ledger. This group of validators executes its own consensus algorithm which is independent of the consensus algorithm run by subjacent blockchain networks. This way, the state of the underlying blockchain ledger is evaluated throughout the network. An implementation of validator nodes for specific networks is necessary to ensure the state of the ledger. The main advantage of this aspect is that the communication between blockchains does not need to deal with the different nodes of the different platforms but is done through the signatures of the Cactus validators.

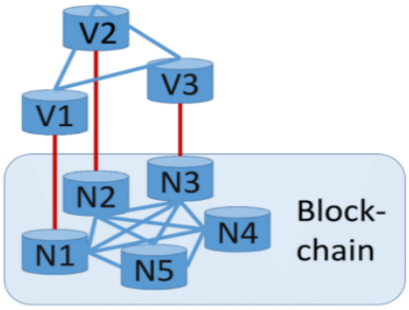

FIGURE 17: CACTUS VALIDATOR NODES.

The following interoperability implementations are supported by Cactus:

- *Ledger transfer*: It is not possible to have two active representations of the same asset at the same time on different blockchains, so it is necessary to burn or block the asset on one blockchain and then release it on the other blockchain;
- *Atomic swap*: Write operations are executed independently on each blockchain but both are committed atomically as well.
- *Ledger interaction*: This type of transaction occurs when an action on one blockchain generates another action on another blockchain. This means that it causes changes in the state of the other ledger;
- *Ledger entry point coordination*: Is the authorization of an end-user's wallet for the purpose of sending read and write transactions to different ledgers from a single-entry point.

Cactus provides the integration of different ledgers through transactions that are controlled by the module of Hyperledger Cactus Business Logic plugin. When a new platform integration is needed, a new plugin must be implemented. Once a user request is sent to the Cactus framework, its module analyses to which ledger and what kind of operations should be executed. The system architecture in Figure 18.

Cactus exposes also a set of APIs for Service Applications to interact with network plugins:

- *Cactus Service API*: this API is used by Application users and allows to request for initializing a business logic which is implemented at Business Logic Plugin;
- *Ledger plugin API*: is implemented for allowing Cactus Business Logic Plugin to manage specific ledgers behind the components of Verifier and Validator. Verifier receives requests from Business Logic Plugin and emits events asynchronously. Validator exposes a set of functions that enable the communication between ledger and Verifier.

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

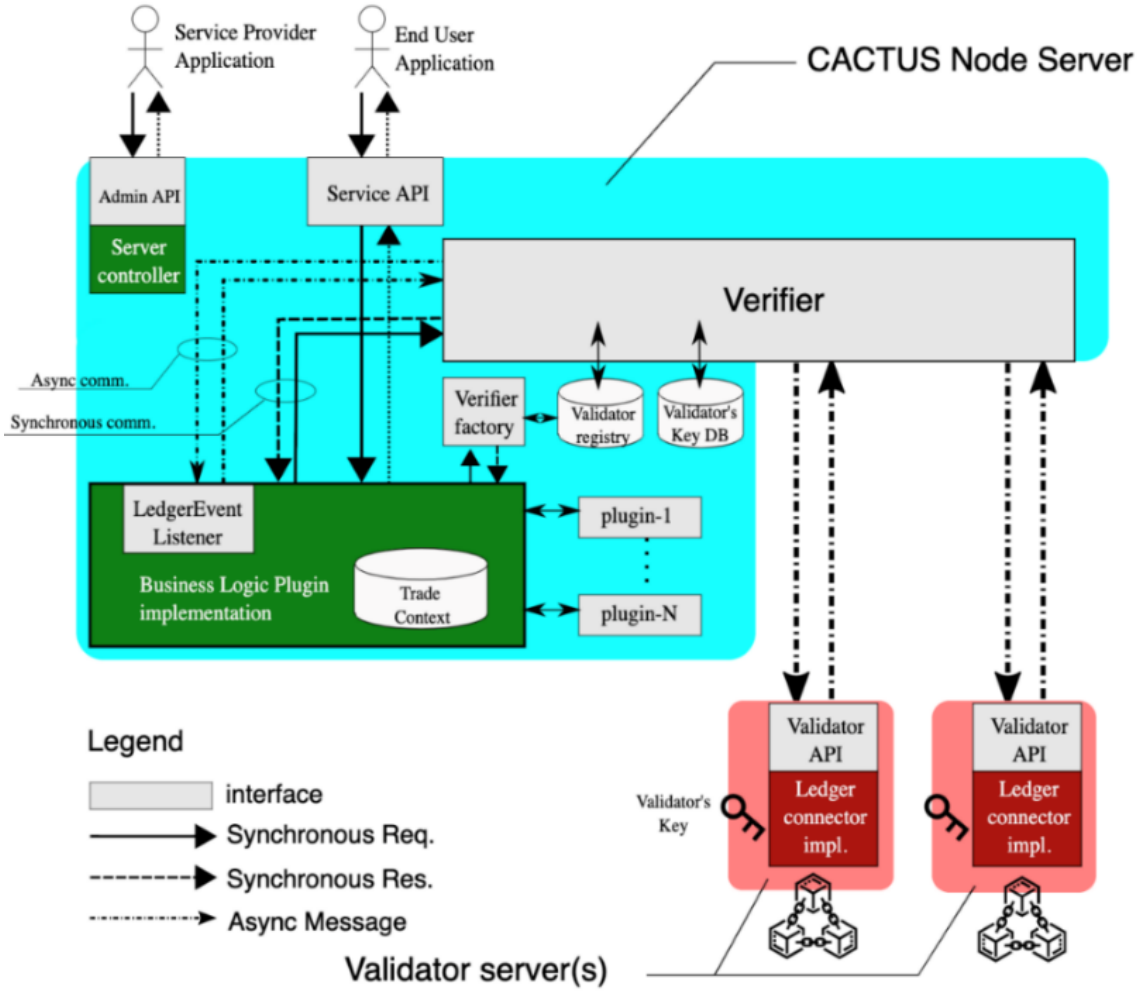

FIGURE 18: CACTUS SYSTEM ARCHITECTURE.

2.3.3. About Wallets

Cryptocurrencies are very valuable assets. Their use within the project is inextricably linked with the high risk of losing it. This can happen as a result of theft or loss of access to the accumulated funds. For this reason, the protection of wallets with accumulated cryptocurrencies should be taken very seriously by all involved participants. The development and promotion of an appropriate safety culture can provide an impetus for companies that are still afraid to use tools based on public DLTs. There are some major groups of wallets:

- *Cloud Wallets*, cryptographic keys for funds are stored by third party that runs dedicated online service, usually categorized as hot wallets as they are not offline and are meant for a frequent use;
- *Software Wallets*, this category includes the software like desktop and mobile applications that store and manage access keys and addresses necessary to administer the funds held by user. This software can also provide the additional layer of security and is often very helpful in creation of new addresses and keys;
- *Hardware Wallets*, specialized computing devices that store addresses and keys. Most notable examples are Trezor or KeepKey. One of the biggest advantages of these wallets is its physical separation from cell phones or computers, because transactions are signed on the wallet device itself, and only the signed transaction is given in return. Thus, the cryptographic keys are not accessible by a third party, e.g., via Internet connection if the host device is compromised;

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

- *Paper Wallets*, the cryptographic key and address that is stored on a piece of paper or other kind of surface. Typically, it consists of two pieces of information – an address, and a private key. The address is presented for payee and the private key is kept secret. It can provide high level of security if generated on the offline device and if the private key part is stored securely.

Cryptocurrency wallets that are stored offline are referred to as cold storage, the most common ones are hardware wallets but the coldest kind of wallet would be a paper wallet, which is completely analogue and offline. Their great advantage is keeping private keys away from security threats that come from computer networks and malicious software.

2.3.3.1 Wallet Backup

The basic role of wallets is to keep safe the private keys which are necessary to spend the funds. Unlike the regular bank accounts, loss of private key equals to irreversible loss of cryptocurrency. Because of that the role of backups is even more critical for wallets than any other information-based assets. We can identify many potential threats to making the owned currency unable to be spent like serious physical damage to the device, steal or loss. DLTs can also implement its own specific controls that can prevent the potential loss. For example, one of the basic security features that makes easier to avoid the loss of Bitcoin wallet are mnemonic seeds introduced with BIP 39 document. It allows to recreate private keys by 12 to 24 special English words memorized or written down previously. Another important feature of Bitcoin wallet security are passphrases. This mechanism allows user to select the password for accessing encrypted wallet data on the device. This also allows to hide Bitcoins behind multiple passphrases so the adversary part cannot take out all bitcoins with knowledge only about one of all passphrases, It cannot verify if the entered passphrase is valid in other way than checking available funds for each tested passphrase.

Regular audits play a crucial role in maintaining wallet security. Location and condition of backup carriers should be checked, passphrases should be recalled and tested, and the restoration procedures need to be exercised.

2.3.3.2 Wallet Entropy

Another important role of modern cryptocurrency wallet is securely generating the new private keys. This cannot be done without establishing secure seed – initial value for creating the private keys. Because people are bad generators of random numbers, this step should be chosen by good wallet software or hardware. Sophisticated procedures for selecting the seed can be more complex and require more steps.

2.3.3.3 Multi-Signature Addresses

A Multi-Signature Address can be regarded as the specific type of a bank account shared by several partners. Spending funds stored on such an account requires several valid signatures. For example, a multi-signature may require to be signed with 2 of 3 private keys. This can make new opportunities for securing the wallets:

- *Multi-Signature multiple factor access*, one owner has many devices and carriers which act like different wallets containing one of private keys required by multi-signature, so the user cannot spend cryptocurrency without confirmation by 2 or more wallets;
- *Creating escrows*, some amount of cryptocurrency can be transferred only by cooperation of multiple parties.

Any device and data carrier that stores information that allows access to cryptocurrency assets should be treated with caution. Plastic and paper cards with mnemonic words or private keys should be stored in a place that prevents unauthorized physical access like a safe or vault. This

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

should be also safe from water and fire damage, for example by lamination for making them waterproof. Its good practice to keep printed backup data in opaque and sealed envelopes that cannot be opened without leaving the visible trace. Also, any forms of permanent authorization data carriers should stay in geographical diversity. Many copies of the data should be kept at the distance, this applies to the data that provides different layers of security too, for example passphrase and mnemonics for Bitcoin or many private keys securing the same transaction.

2.3.3.4 Cryptocurrency Security Standards

The community built up around the cryptocurrencies has set up the document called Cryptocurrency Security Standards (CCSS, [27]). The main intention is to demonstrate the best industry practices and to help businesses to easily implement them. All standards are grouped into 10 core areas: Key/Seed Generation; Wallet Creation; Key Storage; Key Usage; Key Compromise Policy; Keyholder Grant/Revoke Policies & Procedures; Third-Party Security Audits; Data Sanitization Policy; Proof of Reserve; Audit Logs.

CCSS were designed as a separate set of recommendations that are applied overtop standard security practices in other domains. Because of that, it does not cover common standards and practices for increasing the cybersecurity of an information system in general. The recommended minimal level of security for the proof of concept is the Level I that includes only basic security policy.

2.4 Security

The overall goal of security and privacy is to protect the data and services with respect to intentional attacks by malicious entities. Security and privacy are therefore important non-functional properties of every IT system. Thereby security protects the data and services itself, i.e., ensures so-called protection goals like confidentiality, integrity, accountability or availably with respect to data and services. Privacy instead comes into play, if data related to a human being (so-called personal identifiable information (PII) is involved. Thereby privacy protects the human being with respect to misuse of his personal data against his interests. Ensuring the enforcement of protection goals is achieved by the help of so-called security and privacy controls like cryptographic mechanisms, physical protection, access control, etc. Thereby it is obvious that no absolute security and privacy guarantees can be achieved. The question is “only”: How many resources does the attacker need to invest to break the security and privacy policies. The goal is therefore to raise the bar to a level, that makes the probability of successful attacks negligible. Nevertheless, an overall risk assessment should consider even this negligible probability of attack success and should make an informed decision how to deal with this. While designing the overall security concept and choosing the right security and privacy controls, one has to find the right balance/trade-off between the “costs” induced by the security and privacy measures and the benefit gained (e.g., in terms of risk reduction). The reason is, that applying security and privacy controls usually does not “come for free” but introduces some costs. Thereby these “costs” can be of different nature: monetary costs, e.g. for paying the involved security experts or increased operational monetary costs for operating the additional security and privacy controls; costs in terms of decreased usability or degradation of system parameters like increased latency, bandwidth overhead, computational effort, energy consumption, etc. Especially some privacy controls might lead to a decrease in utility of the system functionality. In order to end up with a consistent security and privacy architecture one has to reason about all data processing activities and consider data at rest, data at transit and data at processing. In the following sections the fundamental concepts and measures related to security and privacy will be described. Due to the complexity of the topic this will only be a short introduction and overview. The interested reader is advised to have a look at the many existing textbooks covering security and privacy protection for further explanations.

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

2.4.1 Protection Goals

Protection goals define what needs to be protected. There are many different security and privacy related protection goals depending on which aspect is important regarding protection. Below are the definitions for some security related protection goals. The definition of privacy related protection goals can be found in Section 2.5.

- *Confidentiality*: Only authorized users get the data;
- *Integrity*: Data are correct, complete, and current or this is detectably not the case;
- *Availability*: Data and resources are accessible where and when the authorized user needs them;
- *Accountability*: Accountability ensures that sender and recipients of information cannot successfully deny having sent or received the information. This means that communication takes place in a provable way.

Note that with respect to integrity, although the broader goal would be, that no manipulations on the data by an attacker happens, such a goal would be hardly achievable with realistic effort (just think about Internet-based communication and all the possibilities for an attacker to tamper with the data, i.e. if the attacker can control one of the many involved network elements, like switches, router etc.). Therefore, we relax this goal and say that integrity is also achieved, if we are able to detect manipulations. Obviously, the system needs to react accordingly (e.g., reject the manipulated data). Note also that the term “data” used in the definitions above should be interpreted in a broad sense. It subsumes data, programs, and hardware structures, etc. Additionally, the protection goals do not reflect a strict classification, i.e., one cannot always clearly say, which protection goal is involved for achieving a specific security target. Nevertheless, it turns out, that certain definitions of protection goals are useful from a pragmatic point of view. Consider the case of availability, which could be attacked by manipulating the software. In this case one can either take the viewpoint that this is an attack on availability (with respect to the service involved) or an attack on integrity (with respect to the manipulated program). Moreover, availability implies integrity, since it is not only important that a given service outputs “some” data but the correct one (otherwise there is hardly any reason to operate the service at all). The definitions of protection goals as given above induce some inherent properties regarding the protection goals as well as security in general. The term “authorized user” implies that for a given use case or scenario one has to define who is authorized to do what. Regarding confidentiality it should be mentioned that successful attacks are hard (or even impossible) to detect – since data is not “stolen” but just copied; they cannot be reversed (deletion of all possible copies cannot be enforced); but security controls can prevent successful attacks. Regarding integrity it is somehow the opposite: manipulations cannot be prevented; but they can be detected, and they can be reversed. One difficulty from an engineering point of view is that protection goals have certain dependencies among each other – especially the “weakens” relation is of importance here, since it implies that one needs to find the right trade-off. One such pair of protection goals which weakens each other is “anonymity” vs. “accountability”. Another one is “confidentiality” vs. “availability”.

2.4.2 Attacker Model & Stakeholder Trust Relations

An attacker model describes the maximal strength/power an attacker can have so that the system in question is still secure. In other words: if the attacker is in fact more powerful than described in the attacker model, he might be able to break the security policy. Reasoning about the attacker model and thus limiting the power of the attacker is necessary for at least two reasons: the fundamental reason is that there is no protection against an all-powerful (omnipotent) attacker. The pragmatic reason is that defending against a more powerful attacker

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

usually induces higher costs. There exist many different properties, which can be used to describe the considered attacker. Common ones are the following:

- *Role of the Attacker*: is he an outsider or an insider? Is he user, administrator, operator, maintainer, designer, producer, etc. of the system?
- *(Physical) Distribution and Access*: to which parts of the system has the attacker physical access? How distributed is the attacker in general, e.g., is he attacking at a local, regional or global scale?
- *Resources*: How many resources has the attacker? Resources cover especially computational power and storage space; time; money. Note that you can trade one for the other, e.g., you can buy more computational power now by investing more money or just wait until one you can buy more computational power for the same amount of money (thanks to Moore's law);
- *Behavior*: Is the attacker active or passive? Note that in the literature these terms are usual used as follows: "active" means, that the attacker is (actively) interacting with the system (e.g., by manipulating messages sent over the line); "passive" means that he is just (passively) listening to communication. Another way of describing the behavior of the attacker can be done with the words "observing" vs. "modifying". Thereby these words refer to the fact, if the attacker follows the rules within the parts of the system which are not under his physical control (observing attacker) or if he is willing to break the rules (modifying attacker). An observing attacker has the advantage that he and his attacks are much harder to detect;
- *Intention*: What is the intention of the attacker? Does he want to learn some secrets? Does he just want to disturb or destroy the system?

Related to the description of the attacker model is a comprehensive stakeholder analysis. The goal is to identify the (human) entities involved in the system and especially their roles, intentions, and goals. Additionally, one needs to identify or specify who trust whom with respect to what — i.e., analyze the trust relations among the involved entities. The rationale behind this is the concept/paradigm of multilateral security, a concept which is also known as "security with minimal trust assumptions" [28]. The usual conception regarding security is a kind of fight of "good against evil". But this black-and-white categorization of entities into friends and enemies oversimplifies the complex real-world trust relations. Therefore, multilateral security starts with the observation, that each party has its particular goals from which its protection goals can be derived and formulated. This variety of goals and interests will lead to security conflicts which needs to be identified and compromises needs to be negotiated. The overall security architecture shall be designed in a way which allows each party to enforce its protection goals within the agreed compromise. As far as limitations of this cannot be avoided, they shall equally apply to all parties.

2.4.3 Overview of Security Controls

Security controls are the actual means to achieve the defined protection goals with respect to the defined attacker model. Although the ultimate goal is to prevent any violation of the defined security policy, in reality it turned out to be impossible to achieve. Therefore, one has to apply a meaningful mixture of measures to prevent security violations, to detect security violations, and to react accordingly. Many of the security controls in IT systems are from the domain of cryptography, although there are also many non-cryptographic security controls like physical security or access control. Regarding the three important protection goals confidentiality, integrity, and availability, the first two are often achieved with the help of cryptography – but with respect to availability, cryptography is only of little help. To achieve availability with respect to physical attacks (or forces of nature) it is important to physically distribute the IT system in question (i.e., place parts of the system on different geographic locations). Besides distribution,

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

two other concepts are of importance: redundancy and diversity. The latter refers to using different technologies or different technology providers while building the overall IT-system. Note also that distribution does not only refer to physical aspects, but also to logical ones, i.e., one should apply distributed control and implementation structures. The goal is to avoid any single point of failure or power (e.g., an administrator, who has control over all the parts of the physically distributed system).

Kerckhoff's principle states that the security of the overall system should not depend on the secrecy of the design of the system nor the secrecy of the involved (cryptographic) algorithms but should only depend on the secrecy of certain well-defined secrets known only by well-defined entities. These secrets are called (secret) keys. Key exchange is therefore critical to the security of the overall system, and it needs to be an integral part of its design.

Cryptographic algorithms can be "classified" or grouped according to different properties:

- *The number/kind of keys involved*: There are so-called symmetric algorithms, where all operations (e.g., encryption and decryption) use the same key and there are so-called asymmetric algorithms, where different kinds of keys are involved. Here we have a public key and a private key which together form the asymmetric key pair;
- *The purpose*: cryptographic algorithms are designed to achieve a (set of) well-defined protection goals. There are e.g., encryption algorithms to protect confidentiality or authentication algorithms to protect integrity. Note that using a cryptographic algorithm for a purpose it was not designed for will usually lead to an insecure solution, e.g., encryption does by no means ensure integrity;
- *The security level*: cryptographic algorithms provide some inherent "level of security" based on the design principles they are built upon. The most secure level is called "information theoretic secure". It means, that the attacker cannot derive any information (besides the length) regarding the plaintext from the ciphertext even with unlimited computational power. A related security level is called "semantic security". The difference is that the attacker is restricted with respect to his computational resources. Another security level is "cryptographically strong", which refers to algorithms which are provably as hard to break as solving a believed to be hard mathematical problem (e.g., factoring large numbers or calculating the discrete logarithm). The last one to mention is "well-analyzed". This refers to cryptographic algorithms, for which no mathematical proof exists with respect to their security – but for which many mathematicians and cryptographers have tried to break them without any success.

Many of the "classic" asymmetric cryptography algorithms, based on the mathematical problems, are considered to be insecure if quantum computers come into existence. Therefore, "post-quantum" algorithms are being developed, which remain secure even with respect to the new computational possibilities offered by quantum computers.

2.4.4 Security Controls for Confidentiality

Two important security controls to protect confidentiality are access & admission control and encryption. The model behind access & admission control is as follows: a so-called reference monitor controls the access to the protected resources (e.g., a file). After authentication of a given subject (e.g., a user or another machine) the reference monitor decides based on defined rules (policies), if that subject is allowed to access [28] the requested object. As mentioned above, encryption can be further sub-dived into symmetric and asymmetric encryption. In case of symmetric encryption, a single key is used for encryption and decryption. The symmetric key is just a random bit string, which for many symmetric algorithms – including the often used and NIST standardized Advanced Encryption Standard (AES) – should be at least 128 bits long. One notable exception is the so-called One Time Pad – a symmetric, information

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

theoretic secure encryption algorithm. In order to achieve information theoretic security, the key has to be really random, needs to be as long as the message and can be used only once. These properties of the secret key make the practical application of the One Time Pad hard in many use cases due to the necessary secure key exchange.

In case of asymmetric encryption, the recipient holds a key pair consisting of a private and a public key. The public key is used by the sender to encrypt the message. The resulting ciphertext can be decrypted only with the help of the private key, which is solely in the recipient's possession. There are no information theoretic secure asymmetric algorithms – the best we can achieve is semantic security. Although the public key can be (usually) known to anybody (i.e., does not need to be kept secret) this does not imply, that there are no security requirements with respect to exchanging public keys. Most notably the integrity/authenticity of the key needs to be protected. Otherwise, an attacker can execute a so-called man-in-the-middle attack, i.e., exchange the true key of the recipient with its own public key. Therefore, the attacker would be able to decrypt an intercepted ciphertext which was generated by the sender using the wrong key. Additionally, the attacker could re-encrypt the message with the true key of the recipient and forward the resulting ciphertext to the recipient making the attacker more difficult to detect. One way of achieving the necessary authenticity of the public key is utilizing a so-called public key infrastructure (PKI). In this case special entities (so-called certification authorities) certify that a given public key belongs to a given entity with the help of issuing so-called (public key) certificates.

One famous asymmetric algorithm is called RSA, where the size of the private key should be at least 2048 bit. Another class of asymmetric algorithms is based on a special mathematical structure called elliptic curves. One advantage of asymmetric cryptography based on elliptic curves is the short key size (at least 256 bits) which allows for more efficient calculations compared to RSA. Nevertheless, asymmetric algorithms are usually at least three order of magnitude slower compared to symmetric algorithms. Therefore, asymmetric and symmetric encryption are often combined in practice. The resulting so-called hybrid encryption works as follows: an ephemeral session/message symmetric key is created and used for encryption of the plaintext. This symmetric key is then encrypted using the public key of an asymmetric algorithm. The encrypted key is sent together with the ciphertext to the recipient, who first decrypts the key to be able to decrypt the ciphertext. Hybrid encryption therefore combines the efficiency of symmetric algorithms with the easier key exchange in case of asymmetric encryption. Finally, it should be mentioned that encryption can be achieved with no or negligible overhead in terms of message size, i.e., no length expansion happens meaning that the ciphertext is no longer than the plaintext.

2.4.5 Security Controls for Integrity & Authentication

Similar to encryption there are symmetric and asymmetric authentication algorithms for integrity protection and authentication. In case of symmetric authentication, the procedure is as follows: The sender uses the authentication algorithm to create as so-called message authentication code (MAC) from a given plaintext and secret (symmetric) key. He sends this MAC together with the plaintext to the recipient. Using the same authentication algorithm and based on the received plaintext and the secret key, the recipient can also calculate a MAC. The recipient compares the MAC he has calculated with the received one. If both are equal, no manipulation of the message happened (with very high probability). If they are not equal, some manipulation happened (either of the message or the MAC). Typical sizes of secure MACs are 256 bits. This could introduce a severe overhead for certain use cases especially in the IoT domain, e.g., if a sensor transmits measured values using messages of just a few bits. To overcome this drawback special kinds of MACs can be used, which "accumulate" security over a set of messages, meaning that the attacker might be able to undetectably tamper with the first message of a set of messages (e.g., a series of measured values) – but he would not be able to

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

undetectably manipulate the whole set of messages, i.e., the manipulation could be detected after a certain number of messages was received.

In case of asymmetric authentication, the sender creates an asymmetric key pair and keeps the private key secret. He uses his private key to create a (digital) signature for a given message. He sends this signature together with the plaintext message to the recipient. The recipient can use the public key of the sender, to verify, if message and signature match with respect to the public key. One notable difference between symmetric and asymmetric authentication regarding achievable protection goals is that asymmetric authentication can be used to achieve accountability, which cannot be achieved using symmetric algorithms. The reason is that in case of asymmetric authentication only the sender knows the private key, whereas in symmetric authentication sender and recipient know the secret key. Therefore, in case of asymmetric authentication even a third party can verify that a given message has in fact been sent by the sender – the same is not possible in case of symmetric authentication, even if the secret key would be provided to the third party (to allow verification). This difference in achievable protection goals is the reason why asymmetric authentication algorithms are usually called (digital) signature algorithms. A class of helper functions which are often used in combination with asymmetric or symmetric authentication (or more generally speaking: related to integrity protection) are the so-called (cryptographic) hash functions. A hash function is a compression function, e.g., a function which takes arbitrary long inputs and produces fixed sized outputs (called hash values or hashes for short) having the following cryptographic properties:

- *Collision resistance*: it should be hard to find two (different) inputs which have the same output;
- *Weak collision resistance/second pre-image resistance*: given a pair of input/output it should be hard to find a second input with the same output;
- *Pre-image resistance/one-way function*: given an output it should be hard to find any input that would lead to the given output.

Collision resistance and weak collision resistance are sometime also referred to as “binding” properties, since they “bind” an input to a given output. Pre-image resistance is sometime called “secrecy”, because (at least to a certain extent) the input is kept confidential given only the output.

Note that hash functions alone do not provide integrity since they do not involve any secret. Therefore, the attacker can always easily generate the corresponding hash value for any message of his choice.

2.4.6 Security Assessment

In order to prevent application security defects and vulnerabilities one has to perform the security assessment. It is used in order to make the decision regarding resource allocation, tooling, and implementation. Often it is used to provide the mapping between assets, associated threats, risk, impact, and mitigating actions.

The goal of the security assessment includes identifying the assets which require protection, a related risk, the impact of the asset on the business operations, apply mitigation controls for each asset. Typically, the security risk assessment includes the following steps:

- Identification of all critical assets and associate risks;
- Assessment of identified security risks for critical assets, determine effective and efficient risk mitigation actions, how to effectively and efficiently allocate time and resources towards risk mitigation;
- Mitigation approach and enforce security controls for each risk;

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

- Prevention, by pre-emptive utilization of tools and processes to minimize threats and vulnerabilities.

The good practices assume that the risk is assessed in regular time intervals. The continued assessments help to provide an up-to-date picture of threats and risks. The frequency of audits can be adopted according to assets criticality or impact of associated risks. During the assessment, the following information should be investigated [28]:

- Creating an application portfolio for all current applications, tools, and utilities;
- Documenting security requirements, policies, and procedures;
- Establishing a collection of system architectures, network diagrams, data stored or transmitted by systems, and interactions with external services or vendors;
- Developing an asset inventory of physical assets;
- Maintaining information on operating systems;
- Information about data repositories;
- Current security controls, baseline operations, and security requirements;
- Assets, threats, and vulnerabilities (including their impacts and likelihood);
- Mapping of mitigating controls for each risk identified for an asset.

There is a set of many reliable and complete frameworks and standards to provide the guidelines in the security assessment and risk management. The most prominent include the following:

- ISO 31000 [29], Organization for International Standardization put international standards which gather the risk management principles and guidelines;
- ISO 27000 [30] series, for Information Security Management System. Including the ISO 27005, which specifically looks at information technology-related risk;
- NIST Publication, including 800-39 [31], Managing Information Security Risk, and related to that, others such as 800-30, which specifically looks at risk assessment.

Understanding the attacks, threats, and vulnerabilities companies face is a critical skill required in securing a company's valuable assets. A threat is anything that may harm the assets owned by an application by exploiting a vulnerability [32]. An important tool to use during an assessment of risks is threat modelling, which allows one to understand what the threats are and how those threats could exploit weaknesses within systems. During the assessment of the vulnerability it is useful to follow one of the following models:

- *Common Vulnerabilities and Exposures (CVE)* [33], a list containing an identification number, a description, and at least one public reference for publicly known cybersecurity vulnerabilities. It is maintained by the Mitre Corporation;
- *Common Vulnerability Scoring System (CVSS)* [34], an open framework for communicating the characteristics and the severity of software vulnerabilities. provides a way to capture the principal characteristics of a vulnerability and produce a numerical score reflecting its severity. So, it's basically a quick way to assign value and then assign prioritization to those vulnerabilities;
- *OSWAP Ranking*, the ranging regular update with most serious threats, Top 10 Ranking [35].

2.4.7 Roles and Responsibilities

The important issue about security is to answer the question of who is responsible for it. In a security-related project, one often defines a set of several roles and clearly defined

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

responsibilities. Those roles could be covered by individuals or could be multiple entities. Typically, the following roles are defined:

- *Head of the Agency*, a person who's been legally designated to make decisions and lead the agency;
- *Chief Information Officer*, who we know is responsible for all the IT assets within the organization;
- *Authorizing Official*, responsible for formerly authorizing systems to be in operation;
- *Common Control Providers*, give common controls to the organization such as physical or administrative controls that are inherited by the system;
- *Control Assessor*, assess the effectiveness of the security controls you have placed in effect for the system;
- *An Official for Privacy*, responsible for overseeing privacy, ensuring that privacy requirements are met, and that rules and regulations are followed;
- *Agency Information Security Officer*, overall responsible for information security. Often there is also *a senior accountable official* for risk management;
- *Security or Privacy Architect* designs the mechanisms and the processes that protect security and privacy;
- *System Administrator*, a person who day to day administers the system;
- *Security or Privacy Engineer* designs these mechanisms and implements these mechanisms along with the security or privacy architect.

An extended list of the roles and responsibilities can be also found in the NIST publication dedicated to 800-12 publications [36].

2.4.8 Security Testing

After the security assessment, appropriate testing activities should be performed. Besides testing the system itself, the testing program should also check the people, process, and utilized technologies.

The various techniques should be combined in the security testing including Manual Inspections, Threat Modelling, Code Review, Penetration Testing. OWASP [37] proposed the following rule of the thumb to select proportion for the testing activities.

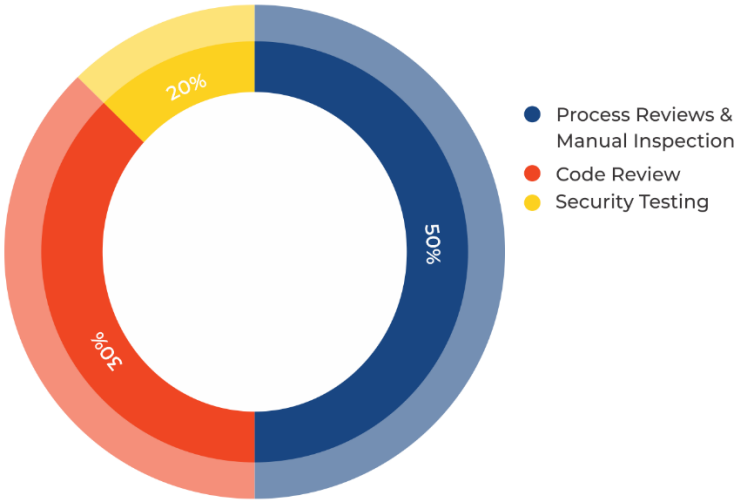

FIGURE 19: THE PROPORTION OF TEST EFFORT ACCORDING TO TEST TECHNIQUE.

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

Enabling the assessments can be done through audit hooks, which will extract data at certain times during a process, allowing us to see what the status of that data is at that point in time. Important guidelines on security testing can be found in the Open Web Application Security Project [38]. OWASP provides resources for security professionals including comprehensive guidelines. More details can be found on link [39]. The goal of security testing is to evaluate the security of a system by methodically validating and verifying the effectiveness of application security controls.

2.4.9 Timestamping

Data integrity can be accomplished for instance by means of timestamping. Here, the idea of using Bitcoin to implement timestamping is explained.

A timestamp ("proof-of-existence") is a proof that some data existed prior to a specific point in time. Timestamping is useful for verification of record integrity of the data stored on a server. If there are trustworthy timestamps for the data kept at a secure location, it is possible to check if the data on the server was modified by an intruder. Timestamping can be used in PGP and software signing. For instance, in the case of revocation of a certificate used to sign software the timestamp can be used for verifying the date of revocation to decide if the software is OK, like in case it has been timestamped before revocation.

The timestamps before Bitcoin were created by trusted authorities or internally by the organization. It usually involves recording a hash of data or signed hash of data and storing it using some archive method. In 2012, Peter Todd started working on OpenTimestamps for Bitcoin [40]. The first version of the software was presented and described in 2016 and it solves the problem of trusting one single entity for providing timestamps.

The method for creating timestamps using an immutable ledger (originally Bitcoin, but this applies to any DLT) involves recording the hash of the data and the date to be timestamped in the Merkle Tree. The tree root is recorded on the blockchain, hence stored forever (assuming the underlying blockchain will not fail). In case of Bitcoin, this kind of timestamps have been available without downtimes since 2013. The timestamping on the DLT can be accompanied with the calendar server that receives timestamps in real time, for storing combined sets of them on the underlying DLT. This way it is possible to create timestamps instantly. If the calendar server is trusted, then there is high accuracy of timestamps. If the calendar is compromised, then the resolution of the timestamps can be assumed to be the same as the storage frequency of the underlying DLT. Right now, there are multiple public calendar servers for timestamping on Bitcoin, and there are other solutions available for different DLTs. See, for example [41], [42] and [43].

2.4.10 Physical Anchoring: TEEs & Remote Attestation

Another fundamental fact related to cryptography is that every cryptographic algorithm needs physical anchoring, i.e., there must be some trusted part of the system to which the attacker has no access. The reason is that otherwise the device or system in question would have no way to store or use the secret cryptographic keys in a secure way (so that the attacker will not learn them). Note that implementing these also called tamper-resistant devices or microchips is not an easy task. One has to combine measures for shielding (preventing easy access to the secrets), detection (of attempts to tamper with the device), delaying, and deletion (of all secrets in case of detected tampering attempts).

Trusted Execution Environments (TEEs) are one way of implementing such physical anchoring. The main idea is to provide a secure compute environment even in case that the user has no physical control over the machine or device used. Such scenarios become especially relevant for IoT-systems combined with (edge) clouds. Based on the highly distributed nature of many IoT-

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

systems one cannot assume that the attacker has no physical access to the involved IoT-devices. The same holds true for edge cloud devices – or even for machines which are operated by a non-trustworthy cloud operator. A more recent term related to TEEs is therefore “confidential computing”.

One recent development, which makes the use of TEEs more “mainstream” was the introduction of Intel Software Guard Extensions into Intel processors. Intel SGX consists of a set of new processor instructions and support for transparent memory encryption. In a nutshell, Intel SGX allows to execute a piece of software within a so-called enclave. The main idea is that all data and program code inside the enclave is running protected from the operating system and hypervisor. Furthermore, all memory assigned to such an enclave is encrypted while stored in main memory. It gets only decrypted when being inside the CPU caches during execution. Therefore, neither an attacker with physical access to the machine trying to spy e.g. on the buses nor the hypervisor or the operating system can get access to plaintext data inside the enclave’s memory or manipulate the code that is executed in it.

It has to be noted that – although TEEs are a valuable and important building block to allow secure computations even in untrusted environments – current implementations of TEEs such as Intel SGX have some limitations with respect to the achievable security. In particular, it has been shown that SGX is vulnerable to side-channel attacks such as variants of Spectre [212], Foreshadow [213], and others (see [214] for an overview). In iNGENIOUS deliverable “D3.1 Limitations and improvement axis for the communication of IoT devices“, a more resilient hardware/software co-design for constructing TEEs is outlined that will be developed within WP3 of the project.

Another concept which is highly related to TEEs is Remote Attestation. Remote attestation refers to means to remotely verify that a given system of hard- and software components is in a given, well-defined state, e.g. that a specific (non-manipulated) piece of software is in fact executed within a well-defined trusted execution environment. Therefore, the combination of remote attestation and trusted execution environments allow to achieve a level of trust and security for a remote machine comparable to a local machine which is under the full (physical) control of the user. Thus, remote attestation is a technology for enabling mutual trust between cooperating entities in a distributed system. In the IoT, this could for example be an edge device and a cloud server.

2.4.11 Network Security

Besides protecting data at rest and during processing, another important aspect is to protect data during transport. Note that communication usually involves many different entities with whom no special trust relations exists. This covers e.g. all the telecommunications operators which provide the communication services. Therefore, communication opens up a large attack surface and thus needs to be especially secured. Fortunately, there are a broad variety of mechanisms, protocols and technologies which could be used to prevent and detect attacks on communication. Note that besides the obvious protection goals confidentiality and integrity, availability is also of special importance, since many of the attacks on availability (so-called Denial of Service (DoS) attacks) involve or are based on communication.

An important property regarding the design of the security mechanisms is the question, if the solution offers so-called End-to-End protection (E2E) or just link encryption. End-to-End protection refers to the fact, that no entity besides the involved true end points of the communication (user device, machine, etc.) can tamper with the transmitted data or get access to the related plaintext messages. In case of link encryption, the protection only covers certain communication link(s) on the path between the two end points, e.g. the communication is only protected between the two gateways of the end point networks. In practice usually a

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

combination of end-to-end and link encryption is applied, since both technologies have their advantages and disadvantages.

Another distinction can be made regarding the network layer at which a given security technology is applied. The main difference is, if a given security technology is application agnostic or if it is tightly coupled with the application, or something in between. Application agnostic solutions usually happen at the IP or transport layer. One main class to mention here are Virtual Private Networks (VPNs). A well-known, standardized implementation is IPsec, which protects communication at the IP layer. Although IPsec is considered to be secure, it is quite complicated to set up correctly and thus prone to security problems induced by faulty configurations. A recently developed, more lightweight and easier to configure alternative is called Wireguard.

One important representative of the "somehow application agnostic" type of solutions is Transport Layer Security (TLS). Coming from the need of securing Web traffic (or more precisely: securing the HTTP protocol), TLS is now a widely adopted security protocol which is utilized in many application domains. There exist even VPN solutions based on the TLS protocol. Although TLS is application agnostic, it cannot (by itself) transport arbitrary IP or transport layer data but has to be embedded into the application design or the overall security architecture.

Another aspect to consider is, if the communication service which should be secured is a reliable one or not. Regarding securing unreliable communication one has to consider, that the security context stored at the involved end points of the communication might become "out of sync" due to packet loss or changes in packet ordering. One usual way of dealing with this challenge is to include certain state information into every data packet. Related to integrity, it is important to define that exact integrity with respect to unreliable communication – and here especially regarding packet loss – means. Applying the strict definition of integrity as given in Section 2.4.2 Attacker Model & Stakeholder Trust Relations would essentially transform unreliable communication into a reliable one. Relaxing the definition (e.g., consider integrity to be achieved, even if some packets get lost or arrive in the wrong order) would allow the attacker to tamper with the communication. It is therefore very important to make application specific decisions if, which and how much data can get lost without any severe negative influence on the application.

Regarding availability, the obvious consideration is that the attacker could try to "destroy" the communication service or at least decrease the provided quality of service. Since most communication involves public networks, the principal ability to tamper with the communication service is usually given to the attacker. Besides these general considerations and important aspects while designing or selecting protocols for a given use case, these protocols have to be designed in a way which does not "support" attacks on availability, i.e., denial of service attacks. Imagine e.g., a typical communication setting where clients connect to a server to use a certain service. The service is offered using a public cloud, so that in principle everybody can communicate with the service. Therefore, for security reasons the protocol used involves authentication meaning that the client has to digitally sign its messages. The point is that verifying these digital signatures requires certain computational effort, while an attacking client can just send random bit strings with close to no effort. This asymmetry in computational effort can lead to an overload even of powerful service by comparably weak attackers.

2.4.12 Summary & Conclusion

Based on a comprehensive requirements analysis of protection worthy assets, related protection goals, and the considered attacker model, an overall security architecture needs to be designed which combines measures to prevent unintended things, detect security violations and react accordingly. Therefore, plenty of security controls exists which have to be composed and orchestrated carefully. Nevertheless, even the best system will not be able to offer "100%

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

security". Thus, an overall risk assessment needs to consider this achievable level of security. Additionally, "security is a process, not a product" (Bruce Schneier [44]) meaning that the overall security architecture and the derived and implemented mechanisms and security controls need to be constantly monitored and adopted based on newly discovered attacks, advances in cryptography and technology.

2.5 Privacy

Personally Identifiable Information (PII) – The European General Data Protection Regulation [180] (GDPR) defines PII as "*any information relating to an identified or identifiable natural person ('data subject'); an identifiable natural person is one who can be identified, directly or indirectly, in particular by reference to an identifier such as a name, an identification number, location data, an online identifier or to one or more factors specific to the physical, physiological, genetic, mental, economic, cultural or social identity of that natural person*".

In general, PII and personal data are terms often used in privacy regulations and laws to summarize personal entities which need to be protected to safeguard the privacy of an individual. As a rule of thumb, if PII is not involved, then it cannot lead to any privacy harm as far as privacy regulations are concerned.

Protection Goals – Several protection goals have been proposed by researchers to ensure the privacy of personal data. We use the definitions proposed by Pfitzmann and Hansen in [168], which provides a more detailed description. The core privacy protection goals are:

- *Anonymity* of a subject means that the subject is not identifiable within a set of subjects, the anonymity set;
- *Unlinkability* of two or more items of interest (IOIs, e.g., subjects, messages, actions, ...) from an attacker's perspective means that within the system (comprising these and possibly other items), the attacker cannot sufficiently distinguish whether these IOIs are related or not;
- *Undetectability* of an item of interest (IOI) from an attacker's perspective means that the attacker cannot sufficiently distinguish whether it exists or not;
- *Unobservability* of an item of interest (IOI) combines undetectability of the IOI against all subjects uninvolved in it and anonymity of the subject(s) involved in the IOI even against the other subject(s) involved in that IOI;
- *Pseudonymity* is the use of pseudonyms as identifiers. A pseudonym is defined as an identifier of a subject other than one of the subject's real names;
- *Transparency* aims at an adequate level of clarity of who processes which personal data for which purposes. This information has to be available before, during and after the processing takes place. Thus, is it meant to empower the party whose data is being processed and stored to have full view over its treatment [169];
- *Intervenability* – aims at the possibility for parties involved in any privacy-relevant data processing to interfere with the ongoing or planned data processing. The objective of intervenability is the application of corrective measures and counterbalances where necessary.

The security controls – also called privacy controls or privacy-enhancing technologies (PETs) – which can be used to achieve the privacy related protection goals mentioned above can be grouped to be either opacity or transparency-enhancing tools (TETs). The general principle behind opacity tools is to suppress personal data or prevent access to personal data. Well-known mechanisms to achieve this are e.g. anonymization or pseudonymization. The general challenge with respect to opacity tools is to find the right trade-off between privacy and utility.

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

Transparency-enhancing tools are a means of dealing with this challenge: if the collection and processing of personal data cannot be avoided due to functional requirements, then the data subject should be at least informed (protection goal: transparency) and should have the possibility to decline the data collection and processing (protection goal: intervenability). A related legal term from the European data protection regulation is “informed consent”. It is important to understand, that transparency not just refers to the fact, that the user is somehow told which data is collected and how the data is processed e.g. by providing lengthy legal text documents – but transparency inherently includes usability, i.e. the information regarding data collection and processing needs to be easily accessible and understandable so that the data subject has the correct mental model regarding the collection and processing of his personal data.

2.5.1 Anonymity & Pseudonymity

Anonymous Communication

Protection of data alone is not sufficient to safeguard privacy and often significant amount of sensitive information can be gleaned by the nature of communication and their participants, or more generally known as metadata. Anonymous communication is the branch of privacy dedicated to safeguarding communication metadata. Popular techniques such as mixnets have been proposed to hide the participants of a communication. Mixnets were first proposed by Chaum [170] and since then it has seen significant development [171]. The core idea behind mix networks is to use a chain of proxy servers or mixes which collect messages from multiple senders and shuffle them after which the messages are forwarded in random order to the next destination which may be another mix node. The mixing is done to break the direct communication link between participants of a conversation, the source and destination. Additionally, using several mixes at least some of which are non-colluding ensures that sender and receiver remain decoupled [172]. The idea of mix networks has been widely used in anonymous communication e.g. as foundation of the popular low-latency mix network The Onion Router (TOR) [178].

Anonymity in datasets

As availability and aggregation of data has become more convenient the abundance of rich datasets with sensitive information has increased. Often researchers and analysts want to study such datasets, which motivate the data holders to release them. Doing so without due diligence poses a serious privacy threat to the members of the dataset [131][132][133].

The most popular mechanism is stripping the data of all identifiers, such mechanisms give an illusion of privacy without providing guarantees of any sort. Research suggests that anonymizing high-dimensional (which is the nature of most feature rich and hence desirable datasets these days) data while preserving utility is a very hard problem [134][135][136][137][138] which may not be achievable [139]. Some techniques go further than just removing identifiers and employing further techniques such as:

- *Data Generalization:* this makes data homogeneous and harder to identify;
- *Data Suppression:* suppresses outliers which are easier to attack;
- *Data Perturbation:* adding noise to data records which prevent straight forward re-identification attacks.

These techniques might make re-identification attacks a bit more complicated but none of them are sufficient to protect privacy while preserving utility and very often such techniques destroy utility without protecting privacy [140][141][142].

There has been some recent work in the direction of generating synthetic data with properties similar to original data using machine learning techniques. The challenge there remains that the

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

ML model which is used to generate such data might still learn the original data and insert its artifacts in the generated data. So far, research indicates such techniques do not provide good privacy properties [143] but this might change in future.

Formal definitions of privacy pertaining to datasets have been proposed which use the same idea as disappearing in the crowd by providing a lower-bound on the anonymity set.

- *k-anonymity* was proposed by Latanya Sweeney [88] to anonymize relational databases. A dataset is said to be k-anonymous, if the information of each individual contained in the dataset cannot be distinguished from at least k-1 other individuals. The notion of k-anonymity is vulnerable to the homogeneity attack where a set of individuals might be k-anonymous yet share the same value of a sensitive attribute;
- *i-diversity* [89] addresses the gaps in protection provided by k-anonymity. The authors propose the notion of an equivalence class which is said to be l-diverse if there are at least l well-represented values for the sensitive attribute. If all the equivalence classes of the dataset are l-diverse then the dataset is considered to be l-diverse;
- *t-closeness* [90] further introduces a threshold t, the distance between the distribution of the sensitive attribute in the equivalence class and the distribution of the attribute in the whole dataset. This provides a much tighter definition.

Differential Privacy (DP)

The basic idea of DP [130] is rather than releasing the whole dataset for analysis, provide a tightly controlled query-based access to private data which provides theoretical privacy guarantees. DP promises that nothing about an individual in the dataset would be learned (via the query-based access) that could have been learned without this individual in the dataset. To achieve this goal the queries are answered with added noise which is calibrated to provide good utility. Since, given a sufficient number of queries any dataset can be reconstructed a good balance must be found between the noise which controls the privacy vs. utility trade-off. Despite its nice properties and strong guarantees, DP has found only limited adoption in practical applications. That is because it is not trivial to deploy and hard to ascertain a good privacy vs. utility balance which affects the noise added to query responses. Additionally, DP is a live system which requires constant monitoring as compared to the previous approach of releasing pseudonymous data and then forgetting about it. DP is also challenging to apply to datasets of disparate nature and varied applications, especially if the nature of computations is not known.

2.5.2 Anonymous Credentials

Anonymous attribute-based credentials (or anonymous credentials for short) [173][174][175][176] are a specific type of cryptographic credentials allowing the credential holder to prove towards third parties certain attributes about him or herself. Yet anonymous credentials allow to do this in a privacy-preserving way. An anonymous credential would allow several things:

- Make the demonstration of the credential unlinkable, meaning that different services cannot decide, if a credential belongs to the same holder or not. Even if the same credential is shown multiple times to the same service, that service cannot decide, if the credentials belong to the same holder or not;
- Support control of the granularity of the revealed information: the credential holder can decide at which level of granularity information should be revealed while showing the credential, e.g., just the information “I am older than 18” or the information “I was born between x and y”; “I was born in February” and so on.

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

Additionally, multiple credentials (from the same or different issuing authorities) can be combined with the help of logical expressions. Moreover, also the requesting entity (verifier) can formulate so-called presentation policies using logical expressions. These presentation policies describe, which credentials the user has to present allowing for some flexibility on the user side, e.g., the verifier could formulate a presentation policy like: "show that you are over 18 or that you have a driving license". Moreover, credentials can be revoked by the issuing authority. Finally, some systems provide the possibility to reveal the identity of the credential holder (revoke anonymity) in case of abuse of the credentials.

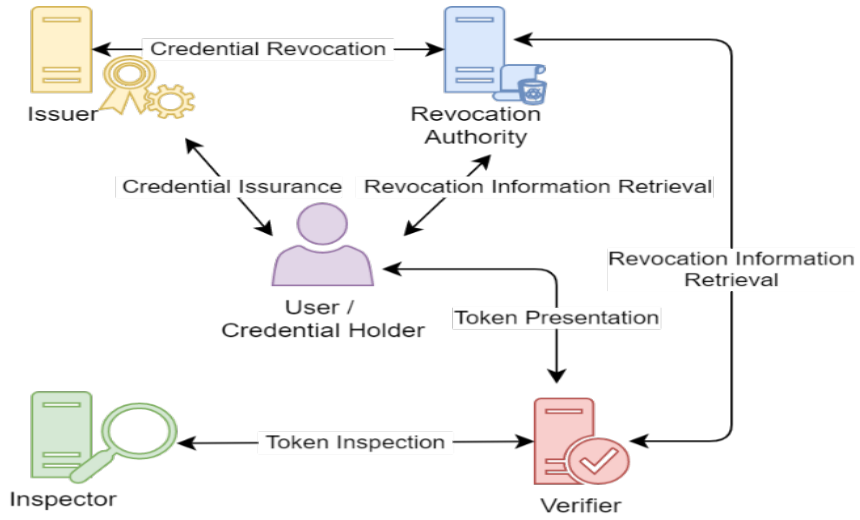

FIGURE 20: ARCHITECTURE OF AN ANONYMOUS CREDENTIAL SYSTEM: ENTITIES AND INTERACTIONS.

The different entities and their relations are depicted in Figure 20 [144]. The figure is inspired by work created by the European ABC4Trust project [181]. Besides this research project, the technology behind anonymous credentials was previously developed within the European PRIME [182] and PrimeLife [183] projects. Moreover, there are certain (industry) implementations like IBM's Identity Mixer [184], Microsoft's U-Prove [185] or the more recent IRMA system [186] (which is based on the same concept the Identity Mixer is based on).

A related concept is that of "self-sovereign identity". Here the fundamental idea is, that the user should be in control of his digital identity – whereby the digital identity is established by the credentials the user presents. Thus, the user has to be in control, how his credentials are used including that the credentials of a given user can only be used with the consent of that user. Anonymous credentials can be seen as one means for implementing the concept of self-sovereign identity. The European Self Sovereign Identity Framework [187] is another approach. It provides an eIDAS compliant identity framework based on blockchain technologies.

2.5.3 Pseudonymization, Guidance & Best Practice

Pseudonymization has an important role in GDPR as a security measure (art. 32 GDPR), as well as in the context of data protection by design (art. 25 GDPR). The most obvious benefit of pseudonymization is to hide the identity of the data subjects from any third party (i.e. other than the Pseudonymization Entity, i.e. the entity responsible for pseudonymization) in the context of a specific data processing operation. Still, pseudonymization can go beyond hiding real identities into supporting the data protection goal of unlinkability, i.e. reducing the risk that privacy-relevant data can be linked across different data processing domains. Furthermore, pseudonymization (being itself a data minimization technique) can contribute towards the principle of data minimization under GDPR, for example in cases where the data controller does not need to have access to the real identities of data subjects but only to their pseudonyms.

There exist different pseudonymization scenarios that can be found in practice considering the various actors and specific use cases. Before showing them let's define terms and actors:

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

- *Pseudonymization* is the processing of personal data in such a manner that the personal data can no longer be attributed to a specific data subject without the use of additional information (GDPR, art. 4(5)), [188];
- *Anonymization* is a process by which personal data is irreversibly altered in such a way that a data subject can no longer be identified directly or indirectly, either by the data controller alone or in collaboration with any other party (ISO/TS 25237:2017) [188];
- *Pseudonym* is a piece of information associated to an identifier of an individual or any other kind of personal data (e.g. location data). Pseudonyms may have different degrees of linkability (to the original identifiers):
    - *Public pseudonym*: the linking between a public pseudonym and its holder may be publicly known even from the very beginning;
    - *Initially non-public pseudonym*: the linkage is (at least initially) not know publicly, e.g. the bank account number;
    - *Initially unlinked pseudonym*: the linkage is (at least initially) not know to anybody (possibly including the pseudonym holder). An example would be the DNA.

  Considering the linkability across different contexts we have the following pseudonym types:
    - *Person pseudonym: the pseudonym has a long lifetime and is used by a human being across many different contexts and can be seen as a substitute for the real name of the pseudonym holder. Social security number, tax number or mobile phone number are some examples;*
    - *Role pseudonym: the pseudonym is used by the pseudonym holder if the pseudonym holder* acts in a certain role;
    - *Relationship pseudonym*: here the pseudonym depends on the communication partner.
    - *Role-relationship pseudonym*: this type of pseudonym is essentially a combination of role and relationship pseudonyms;
    - *Transaction pseudonym*: transaction pseudonyms provide the strongest anonymity. Transaction pseudonyms are only valid with respect to a single transaction – transaction pseudonyms of different transactions are unlinkable, even if they belong to the same holder.

Follow definitions regarding data processing:

- *Identifier* is a value that identifies an element within an identification scheme [189]. A unique identifier is associated to only one element;
- *Data controller* or *controller* is the natural or legal person, public authority, agency or other body which, alone or jointly with others, determines the purposes and means of the processing of personal data (GDPR, art. 4(7)). The data controller is responsible for the data processing;
- *Data processor or processor* is the natural or legal person, public authority, agency or other body which processes personal data on behalf of the controller (GDPR, art. 4(8));
- *Pseudonymisation entity* is the entity responsible of processing identifiers into pseudonyms using the pseudonymisation function. It can be a data controller, a data processor (performing pseudonymisation on behalf of a controller), a trusted third party or a data subject, depending on the pseudonymisation scenario.

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

2.5.3.1 Scenarios

In literature [188] the following scenarios are relevant to understand the roles of data processing actors previously defined:

- *Pseudonymization for internal use*, when data are collected directly from the data subjects and pseudonymised by the data controller, for subsequent internal processing;
- *Processor involved in pseudonymization*, where a data processor is also involved in the process by obtaining the identifiers from the data subjects (on behalf of the controller). However, the pseudonymisation is still performed by the controller;
- *Sending pseudonymised data to a processor*, in this scenario the data controller again performs the pseudonymisation but this time the processor is not involved in the process but only receives the pseudonymised data from the controller;
- *Processor as pseudonymization entity*, in this scenario the task of pseudonymisation is assigned by the controller to a data processor. In this scenario, only the pseudonymised data are stored on the controller's side. A variation of this scenario could be a case where several different processors are involved in the pseudonymisation process as a sequence of pseudonymisation entities (chain of processors);
- *Third party as pseudonymization entity*: the pseudonymisation is performed by a third party (not a processor) who subsequently forwards the data to the controller. Contrary to the Scenario 4, the controller in this scenario does not have access to the data subjects' identifiers (as the third party is not under the control of the data controller);
- *Data subject as pseudonymization entity*: the pseudonyms are created by the data subjects themselves as part of the overall pseudonymisation process. The goal of pseudonymisation in such case is that the controller does not learn the identifiers of the data subjects and the data subjects can be in control of the pseudonymisation process.

2.5.3.2 Goal of Attack on Pseudonymisation

Attacks are performed by adversaries that are cataloged on the type of permissions or knowledge they have:

- *Insider Adversaries*, he is an adversary with specific knowledge, capabilities, or permissions (with regard to the target of the adversary) [190]. As example, considering the previous scenarios, an insider could be an employee working for the controller;
- *External Adversaries*, he is an adversary that does not have direct access to the pseudonymisation secret or other relevant information. However, this type of adversary may have access to a pseudonymised dataset and may also be able to perform the task of pseudonymisation to arbitrary input data values chosen by the adversary.

An adversary can focus on discovering the **pseudonymisation secret** (when used), re-identifying any pseudonym in the dataset (complete re-identification or discrimination). When adversary wishes to achieve the linking of one or more pseudonyms back to the identity of the pseudonym holders, the goal is known as **complete re-identification.** The most severe complete re-identification attack consists of the re-identification of all pseudonyms.

2.5.3.3 Main Attacking Techniques

- *Brute Force attack* is based on the adversary's ability to compute the pseudonymisation function (i.e., there is no pseudonymisation secret) or his/her access to a "black box" implementation of the pseudonymisation function;

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

- *Dictionary search* the adversary has to deal with a large number of pseudonyms to carry out complete reidentification or discrimination. Therefore, he or she precomputes a (huge) set of pseudonyms and saves the result in a dictionary. Each entry in the dictionary contains a pseudonym and the corresponding identifier or information. Each time the adversary needs to re-identify a pseudonym, he/she is going to search into the dictionary. This search has a pre-computation cost of an exhaustive search and stores the result in large memory;
- *Guesswork* utilizes some background knowledge (such as probability distribution or any other side information), which the adversary may have on some (or all) of the pseudonym holders, the pseudonymisation function, or the dataset. Some identifiers may be more frequent than others. Exploiting the statistical characteristics of the identifiers is known as guesswork and is widely applied in the password-cracking community.

2.5.3.4 Pseudonymization Techniques

In principle, a pseudonymisation function maps identifier to pseudonyms. There is one fundamental requirement for a pseudonymisation function. Considering identifiers $Id1$ and $Id2$ and their corresponding pseudonyms $pseudo1$ and $pseudo2$, a pseudonymisation function must ensure that $pseudo1$ is different than $pseudo2$. Otherwise, the recovery of the identifier could be ambiguous: the pseudonymisation entity cannot determine if $pseudo1$ corresponds to $Id1$ or $Id2$. However, a single identifier Id can be associated to multiple pseudonyms ($pseudo1$, $pseudo2$…) as long as it is possible for the pseudonymisation entity to invert this operation.

- *Counter***,** the identifiers are substituted by a number chosen by a monotonic counter. First, a seed s is set to 0 (for instance) and then it is incremented. It is critical that the values produced by the counter never repeat to prevent any ambiguity;
- *Random number generator (RNG)*, aka Tokenization, is a mechanism that produces values, or characters, in a set that have an equal probability of being selected from the total population of possibilities and, hence, are unpredictable. This approach is similar to the counter with the difference that a random number is assigned to the identifier;
- *Cryptographic hash function* takes input strings of arbitrary length and maps them to fixed length outputs. It satisfies the following properties:
    - *One-way*: it is computationally infeasible to find any input that maps to any pre-specified output;
    - *Collision free*: it is computationally infeasible to find any two distinct inputs that map to the same output.

  A cryptographic hash function is directly applied to the identifier to obtain the corresponding pseudonym: $Pseudo = H(Id)$. The domain of the pseudonym depends on the length of the digest produced by the function.

  Apart from plain hashing of data, more advanced structures like:

    - *Merkle trees* utilize hashes of sets of hashes, e.g. h3=hash(h1, h2), to achieve structured pseudonyms that can be uncovered only partially instead of completely;
    - *Hash chains* rely on repeatedly hashing the hash values of hash values.
- *Message authentication code (MAC)*, this primitive can be seen as a keyed-hash function. It is very similar to the previous solution except that a secret key is introduced to generate the pseudonym. Without the knowledge of this key, it is not possible to map the identifiers and the pseudonyms;
- *Encryption*, according to the Article 32(1) GDPR, pseudonymisation – as well as encryption - may be an appropriate technical and organizational measure towards ensuring an appropriate level of security (security of processing).

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

It is clear that encryption and pseudonymization, both referenced in GDPR as security measures, have different scope. The main focus of pseudonymization is to protect the identities of individuals (for anyone without access to additional information). Yet, pseudonymised data provides some legible information and, thus, a third party (i.e. other than the controller or processor) may still understand the semantic (structure) of the data, despite the fact that these data cannot be associated with an individual.

On the other hand, encryption aims at ensuring – via appropriately utilizing mathematical techniques - that the whole dataset that is being encrypted is unintelligible to anyone but specifically authorized users, who are allowed to reverse this unintelligibility (i.e. to decrypt). To this end, encryption is a main instrument to achieve confidentiality of personal data by hiding the whole dataset and making it unintelligible to any unauthorized party.

*Symmetric encryption***,** the original identifier of a data subject can be encrypted through a symmetric encryption algorithm (e.g. the AES, being the encryption standard [FIPS, 2001]), thus providing a ciphertext that is to be used as a pseudonym; the same secret key is needed for the decryption.

Asymmetric encryption (i.e. public key) the main characteristic of asymmetric encryption is that each entity participating in the scheme has a pair of keys, i.e. the public and the private key. The public key of an entity can be used by anyone to encrypt data but only the specific entity can decrypt these data with the use of its private key. To provide the so-called **ciphertext indistinguishability property**, the public key algorithms may be appropriately implemented in a probabilistic form by introducing randomness in the encryption process. This means that randomly chosen values are being used at each encryption cycle. In this way, if the same message is encrypted twice with the same public key each time, the corresponding two ciphertexts will be different, without affecting the decryption capability for the holder of the decryption key.

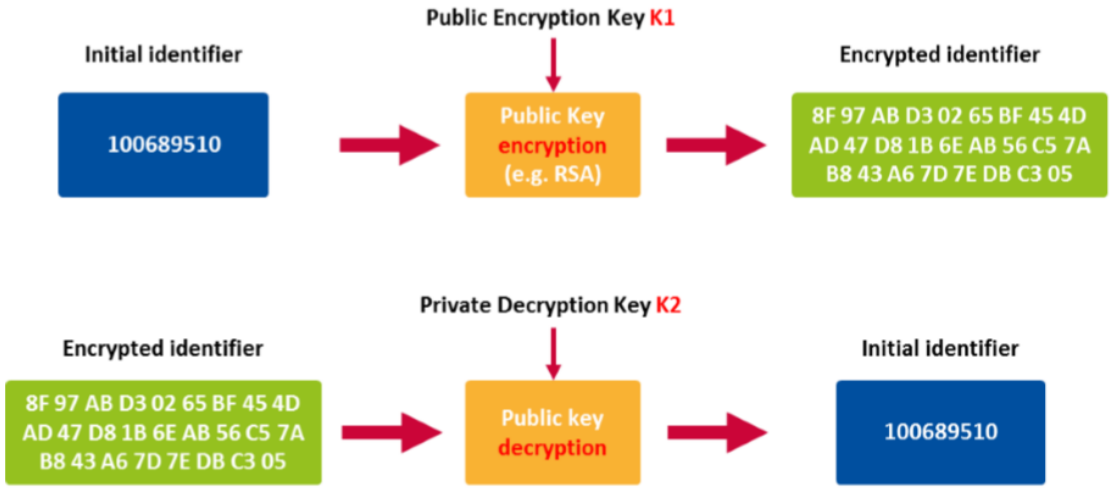

FIGURE 21: ASYMMETRIC ENCRYPTION ALGORITHM.

All techniques that can effectively be utilized, to increase anonymization can also be useful for pseudonymisation, such as the common techniques for **k-anonymity** or **differential privacy** and beyond.

2.5.3.5 Pseudonymization Policies

While the choice of the pseudonymisation technique is essential, the policy (or mode) of implementation of pseudonymisation is equally important to its practical application.

Three types of policies can be considered:

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

- *Deterministic pseudonymization*: in all the databases and each time it appears, the identifier is always replaced by the same pseudonym;
- *Document-randomized pseudonymisation*: Each time the identifier Id appears in a database, it is substituted with a different pseudonym ($pseudo1$, $pseudo2$, ...). However, Id is always mapped to the same collection of ($pseudo1$, $pseudo2$) in the dataset A and B;
- *Fully randomized pseudonymisation*: for any occurrences of Id within a database A or B, Id is replaced by a different pseudonym ($pseudo1$, $pseudo2$). This case is fully randomized pseudonymisation.

The choice of a pseudonymisation technique and policy depends on different parameters, primarily the data protection level and the utility of the pseudonymised dataset. In terms of protection, as discussed in the previous sections, RNG, message authentication codes and encryption are stronger techniques as they thwart by design exhaustive search, dictionary search and guesswork.

2.5.3.6 Recovery

The pseudonymisation entity must implement a recovery mechanism. This mechanism can be more or less complex depending on the pseudonymisation function. In general, they consist of the use of a pseudonym $pseudo$ and a pseudonymisation secret S to recover the corresponding identifier Id.

The recovery mechanism could be also necessary in order to allow for the exercise of data subjects rights (under articles 12-21 GDPR).

Most methods described previously require the pseudonymisation entity to keep the mapping table between the identifiers and the pseudonyms to perform identifier recovery with the exception of encryption. Indeed, decryption can be directly applied on the identifier.

2.5.3.7 Protection of the Pseudonymization Secret

The pseudonymisation entity must always protect the pseudonymisation secret by proper technical and organizational measures.

Firstly, the pseudonymisation secret must be isolated from the dataset, i.e. the pseudonymisation secret and the dataset must never be handled in the same file (otherwise, it will be too easy for an adversary to recover the identifiers). Secondly, the pseudonymisation secret must be securely deleted from any insecure media (memory storage and systems). Thirdly, strong access control policies must ensure that only authorized entities have access to this secret. A secure logging system must keep track of all the access requests made to the secret. Finally, the pseudonymisation secret must be encrypted if it is stored on a computer, which in turn necessitates a proper key management and storage for this encryption.

2.6 Smart IoT Gateway

Currently there are several different emerging technologies on the Internet of Things (IoT) communication and interoperation fields. These technologies are tightly related to the evolution found in the IoT devices. Smaller and more power efficient devices make possible new scenarios where a significant number of physically separated devices are able to transmit data in a myriad of protocols, data rates, formats, or encapsulation streams.

In order to make a taxonomy of the current industry trends in IoT technologies, we make a first segregation based on the Open Systems Interconnection (OSI) model [45]. More specifically, we will be focusing just on the layers that have a special relevance as these IoT technologies introduced significant contributions to allow the implementation of these new evolved scenarios.

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

2.6.1 Physical and Data Link Layer

This is probably the most significant area for IoT, as the devices most be ubiquitous, and they must be as connected as possible. To achieve this, IoT devices should be able to transmit and receive messages using different mediums in different environments. IoT devices designed for industrial applications do not always share the same communication links as domestic or scientific IoT devices.

There are generic common interfaces, which are found in many different types of devices. They are not optimized for IoT, but they are well known and deployed:

- Institute of Electrical and Electronics Engineers (IEEE) 802.15.1 & Bluetooth [46];
- IEEE 802.11 (Wi-Fi) [47];
- IEEE 802.3 (Ethernet) [48];
- 5G New Radio (NR) [208].

There are industrial wired interfaces, for devices that do not support wireless communication. Among these, the 2 dominant standard interfaces are:

- Serial Peripheral Interface (SPI) [49];
- Inter-integrated Circuit (I2C) [50].

There are IoT specific which are focused to be used in small devices with limited processing power:

- Narrowband (NB)-IoT (part of the 3rd Generation Partnership Project (3GPP) Low-Power Wide-Area Network (LPWAN) standards) [51];
- Long Term Evolution for Machines (LTE-M)/enhanced Machine Type Communication (eMTC) which are part of 3GPP LPWAN standards [52];
- Long Range (LoRa): A proprietary low power and long-range communications [53];
- IEEE 802.11ah (Wi-Fi HaLow): A derivative from 802.11 focused on networks of sensors [54];
- SigFox: A popular proprietary that uses the Industrial, Scientific, and Medical (ISM) band [55];
- IEEE 802.15.4 (ZigBee): A protocol that allows small power devices to create Personal Area Networks (PANs) [56].

2.6.2 Network, Transport Layer and Application Layer

There are protocols and transport frameworks that support different physical interfaces and provide an envelope to encapsulate IoT messages efficiently depending on the scenario.

- *Transmission Control Protocol (TCP)/Internet Protocol (IP)*: Provides maximum integration with existing networks at a cost of resources;
- *Constrained Application Protocol (CoAP)*: A specialized protocol designed to have an easy interface with TCP/IP and to be translated into Hyper Text Mark-up Language (HTML) [57];
- *Message Queuing Telemetry Transport (MQTT)*: Along with its extensions (MQTT for Sensor Networks (MQTT-SN), formerly known as S-MQTT), is popular and well documented protocol based in publisher/subscriber pattern [58];
- *OneM2M*: A set of standards (that covers open interfaces and protocols) to allow easy interoperability between the device space and the Machine to Machine (M2M) space [59].

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

2.6.3 Presentation Layer

There are supporting applications although not directly related in the presentation layer but tightly associated to the model itself. In this group we include software inherent to the nature of the IoT space data:

- Non-Structured Query language (NoSQL) databases, especially document NoSQL databases, to store the information in a format close to the source [60];
- Time series databases, as almost all data generated by the IoT devices are time stamped [61];
- Data visualization tools that are able to support the technologies listed in the previous points.

2.6.4 Bridging Between Device Space and the Cloud

In a typical IoT setup, and due to the different nature of each element, components do not form a homogenous system. IoT devices are deployed in numbers of several orders of magnitude higher than the machines that will interface them. There are also different protocols, and the communication is not usually symmetric (sensors transmit way more data than the other way around).

Essentially, there are 2 big spaces when describing IoT scenarios:

- *Device space*: where sensors, actuators or any other small device is deployed;
- *M2M space*: where bigger structures such as enterprise systems are deployed.

For this reason, the industry gathered to provide useful standards that could simplify the communication between machines (the M2M space). This point was especially relevant at that time. The variety of devices and manufacturers appearing in the IoT sector, each one with its own implementation and protocols, was cluttering the scene. M2M interfaces needed to constrain themselves to a specific vendor or become complex parts of the architecture to accommodate implementations for the different types of vendors they want to support.

This situation changed with the introduction of oneM2M. This extensive set of protocols covers all M2M communication, including testing, interoperability with known IoT protocols or reference use cases. As these standards are open, different implementations are available, being some of them open source.

OneM2M is a cross sector framework as it covers the needs of different scenarios, not matters if it is automotive, medical, industrial, or home. It abstracts the device space providing a common interface for different devices. It also defines logical containers and nodes, that will allow scalability from small scenarios to multi device networks including multiple gateways and application servers.

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

3 Available Supply Chain IoT Application

3.1 Supply Chain IoT Applications

Supply chains are one of the most complex parts of business operations since they require synchronization and collaboration between different business segments and actors. In this context, IoT and data analytics applications have an important role since they may contribute to optimize operations, resolve issues, and identify potential bottlenecks across different segments like factories, warehouses, transportation and logistics, or maritime ports. In the following, several examples of IoT applications in supply chains are presented.

3.1.1 IoT Applications in Smart Factories and Warehouses

Nowadays a large percentage of the companies involved in the industrial manufacturing sector are carrying out smart factory initiatives where systems and devices are expected to become fully interconnected and where streams of data will be turned into valuable insights. In this context, Next Generation IoT (NG-IoT) technologies will not only enable optimization of the industrial operations but also will affect product, development, storage, and delivery processes thanks to an efficient use of the data.

The use of NG-IoT applications to connect sensors and establish machine-to-machine communication protocols will help factories and warehouses to get real-time data at every stage of the supply chain with high levels of reliability and availability (up to 99.9999% for control and up to 99.9% sensing), low latencies (≤ 10ms), and positioning (≤ 0.5m). Additionally, the combination of IoT applications with cloud-based management systems and Artificial Intelligence modules will improve machinery operation and asset management procedures, allowing companies to know the location of machinery and goods while predicting risky events like machinery failures or low-stock events.

Other innovations like Automated Guided Vehicles (AGVs) will be able to calculate the shortest route for product delivery, reducing the amount of time needed to complete the operation together with fuel costs. At the same time, by connecting IoT platforms and enabling the exchange of data between supply chain players through Distributed Ledged Technology (DLT) solutions, factories and warehouses will be able to track the different events that take place when products are manufactured and when orders are delivered out of the facilities.

Some examples of IoT applications in smart factory and warehouse scenarios are:

- *MindSphere Industrial IoT Solution* developed by Siemens is an Industrial IoT as a service solution that uses advanced analytics and AI to power IoT solutions from the edge to the cloud. Thanks to MindSphere, factories and warehouses can ingest and visualize immediate real-time data and analytic results in one centralized location with no development required. For that purpose, it includes different components such as an Asset Manager, Fleet Manager, Usage Transparency or an Operator Cockpit;
- *Amazon Web IoT Services for Industrial Applications* developed by Amazon is an Industrial IoT solution that brings machines, cloud computing and analytics together to improve the performance and productivity of industrial processes. Thanks to this AWS module, factories and warehouses can cover different use cases such as predictive quality and predictive maintenance or asset condition monitoring. A detailed set of the different components together with their role within the AWS Industrial IoT (IIoT) architecture is shown in FIGURE 22.

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

FIGURE 22: AWS IoT ARCHITECTURE – SOURCE: AMAZON.

3.1.2 IoT Applications for Smart Transportation and Ports

NG-IoT technologies for transportation encompasses the set of devices, services, applications and platforms that gather and transmit data about transport and logistics operations in terrestrial, maritime and aerial segments. Within these areas, NG-IoT solutions are called to help companies to provide an accurate and updated real-time flow of data for scheduling and planning transportation flows while improving communications. At the same time, transport agencies are adopting these technologies for tracking assets and connecting vehicles to other vehicles and infrastructures. Complementing operational aspects, IoT applications will also help to improve safety and reduce the environmental impact of transportation by tracking transportation location, speeds and emissions.

Nevertheless, the use of NG-IoT applications is not only expected to affect transportation but also the logistics segment, covering terrestrial hubs and maritime ports and terminals. Within ports, the integration of IoT technology is poised to enhance the interconnection of all kinds of objects and sensors with the vehicles and equipment used at ports, facilitating smart loading, unloading, and transportation operations. Essentially, the Industrial Internet of Things is used to perform four basic operations within ports: sensorization, positioning, interconnection between devices, and monitoring of events thanks to the interconnection with DLT technologies.

Some examples of IoT application for smart transportation and logistics are:

- The cargo tracking platform developed by Traxens relies on a merger of IoT, Big Data and data pipelines used to massively connect containers everywhere and in real-time, enabling the delivery of information to all supply chain actors. Traxens technology is based on 3 building blocks or pillars: (i) Traxens-Box, a context-aware cost-effective versatile IoT device with an embedded processor, several sensors (temperature, vibration and shock, door opening detection) and various communication methods prioritized in a dynamic way to select the least energy consuming communication mode; (ii) Traxens-Net, an ultralow power wireless communication protocol designed to meet the container transportation requirements by self-organizing a mesh network able to communicate between different Traxens-Boxes and make them share the power required for access to existing global networks; and (iii) Traxens-Hub, an online cloud-based platform that compiles all data gathered by the box and net components through the use of APIs. The Traxens-Hub plays

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

an important role in providing full visibility of transport and data governance and role-based services access [62];

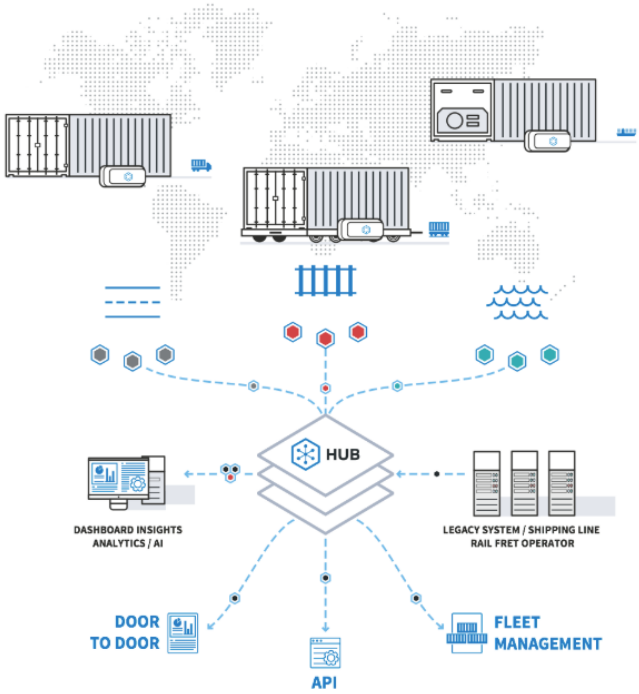

FIGURE 23: TRAXENS SOLUTION ARCHITECTURE – SOURCE TRAXENS.

- TradeLens is a digital platform developed by IBM and Maersk for enabling global shipping through permissioned data sharing across the global supply chain ecosystem, including different players like shippers, freight forwarders, ports and terminals, ocean carriers, intermodal operators, government authorities, customs brokers, and more. The platform is powered by IBM Cloud and IBM Blockchain solutions (Hyperledger Fabric based) in order to enable permissioned data sharing between participants while maintaining authorised and safe access to the different flows of data depending on the identity and role of the entity.

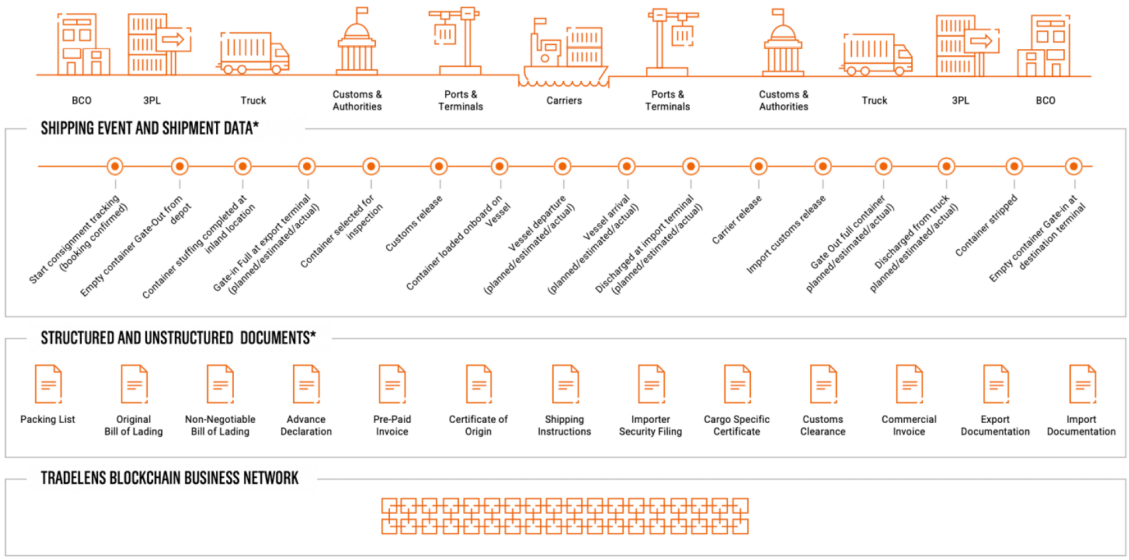

FIGURE 24: TRADELENS DATA FLOWS AND ACTORS.

The TradeLens solution is made of three key components: (i) the ecosystem, which is integrated by the different supply chain players participating in the network, (ii) the digital platform, which is accessible from an open API and a web interface and is powered with a Hyperledger Fabric based solution for the safe exchange of information; and (iii) the

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

marketplace, that allows the publication of applications and services developed for enhancing the digital platform, enabling innovation and the creation of added value [62];

- Big Schedules platform is an intelligent sailing schedule search engine developed by CargoSmart.ai that delivers up-to-date vessel schedules from multiple shipping lines and terminal operators. Big Schedules accesses multiple data sources, including Internet of Things (IoT) data, monitoring up to 7,000 vessels' live movements, covering 90% of the world's container capacity and over 800 global container ports.

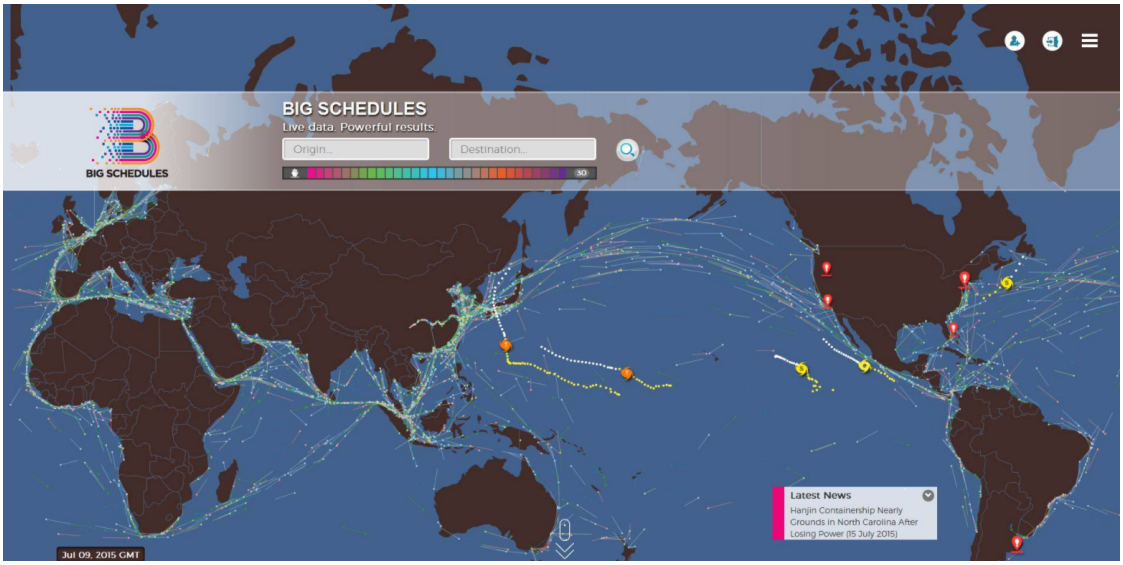

FIGURE 25: BIG SCHEDULES PLATFORM.

To integrate data flows between different supply chain entities, this search engine exploits AI, IoT and blockchain technologies enabling hauliers, terminals, railway operators and other players a safe and dynamic data access [63]. Some of the features included in the platform are: (i) event monitoring for informing of vessel and port incidents that may impact the sailing schedules of interest; (ii) schedule comparison for selecting routes by comparing up to three schedules along with their arrival accuracy; (iii) vessel tracking for tracking the current position and past routes of over 7,000 container vessels on an interactive map; (iv) watch list for receiving alerts about sailing schedule changes and vessel status and (v) carrier performance analytics for gaining market intelligence on carrier performance and customize analysis options.

3.2 Tactile IoT Applications

Tactile internet is an evolution of IoT characterized by extremely low latency in combination with high availability, reliability, and security. Tactile IoT applications are designed to perform certain tasks by monitoring input data and altering output data accordingly. In tactile IoT applications, the input data are the outcome of sensing and the output data are translated to actions performed by actuators. The main task is realized by means of one or more processes that define the relations between a set of inputs and set of outputs. The outputs of some process might be used as input to others. An exemplary procedure could be: if a bell rings, check the identity of the person, open a door if the person is found in a whitelist. In this case, the input data is the bell signal, and the process to check the identity of the person, which can be accomplished by sending a command to a camera to take a picture. This leads to a secondary procedure where the input is now the image, the process is face detection, and the action is to open the door or decline entry.

IoT is commonly associated with massive deployment of light devices such as sensors and switches with relaxed timing requirements. Tactile internet IoT includes a wide range of applications that are technically distinguished by stringent end-to-end latency requirements.

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

3.2.1 Available Supply Chain IoT Applications

Tactile applications are based on the remote interactions between objects, which can be humans, physical or virtual machines, while preserving similar perception as when the objects are directly connected [173]. The interactions involve remote accessing, perceiving, manipulating, or controlling of real or virtual objects or processes, and are distinguished by the requirements of URLLC to achieve perceived real-time response. The exact definition of real-time response depends on the application. This category of applications is specifically driven by the challenges of remotely conveying the sensing of haptic touch and kinaesthetic muscular movement for human, in addition for the timing requirements of closed-loop control systems. Accordingly, two main scenarios are encountered:

- *Human-in-loop*: in which the humans should be able to remotely interact with real or virtual objects and perceive different auditory, visual and haptic feedback with the same experience when directly dealing with the physical objects. This requires an HSI, such as haptic gloves, VR headset, to translate the human actions to machine type commands, and the machine feedback to human perceived signals;
- *Machine-in-loop*: this corresponds to remote connection of machines, such as sensors, actuators, robots, control process. In this case, the different interaction should lead to the same performance as when the different identities are directly or closely connected.

Tactile applications have practical use in supply chain operations. For example, teleoperation and automation can improve working conditions and increase productivity in logistics. In addition, autonomous as applications-e.g. in transportation and warehouse management. These applications are outlined in more detail below.

3.2.1.1 Teleoperation

This application allows a human user to operate a device or machine located in a remote area [174]. Teleoperation enables performing task in hazardous or inaccessible environments to ensure workers' safety, and it can also be employed to provide comfortable working conditions. In contrast to conventional remote control, the tactile version, depicted in FIGURE 26: TACTILE REMOTE OPERATION., offers a realistic experience as the user feels like they are operating a physical device. For that, the master operator, and the slave teleoperator device exchange haptic signals, such as forces, position, velocity, vibration and torques, in addition to video and audio signals by means of an HSI. The HSI encodes the human actions to commands understood by the teleoperator and translates the feedback from the teleoperator to signals perceived by the human.

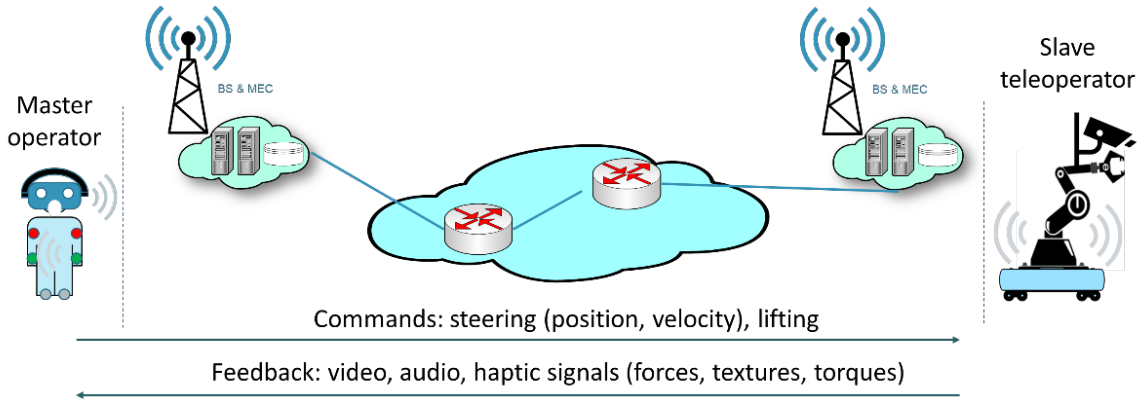

FIGURE 26: TACTILE REMOTE OPERATION.

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

3.2.1.2 Autonomous Driving

Mobility is essential for supply chains both for transporting goods and for handling raw materials and products in production lines. Autonomous driving enables smarter, more ecological, and safer movement of people and goods. Self-driving requires processing multiple types of information such as optical images, radar, etc., generated by sensors installed as part of the surrounding infrastructure or on vehicles. The sensed data are conveyed to a controller, which needs to compute and forward driving commands such as steering, breaking, and acceleration within a latency constraint. Autonomous driving may also involve platooning, where it is required to control the speed of a line of vehicles traveling in the same direction.

3.2.1.3 Industrial Automation

Industrial closed-control loop has URLLC requirements similar to those of tactile internet [175]. Thus, the first application is to replace the wired industrial network by a wireless one, as illustrated in FIGURE 27, which leads to greater flexibility and reduced cost of installation and maintenance, especially for connecting moving devices. This flexibility has inspired new industrial applications that connect people, objects, and systems. The conventional Human Machine Interface (HMI), which typically consists of a display, input terminal, and software for gathering data and altering control parameters, can be replaced by alternative Augmented Reality (AR) or VR interfaces. Mobile platforms such as AGVs and mobile robots can be programmed to perform various scenario-dependent tasks instead of fixed robot arms dedicated for certain tasks. This allows more efficient use of resources and forms the core of next-generation industrial applications, as will be discussed in Section 4.3 Next Gen IoT Applications.

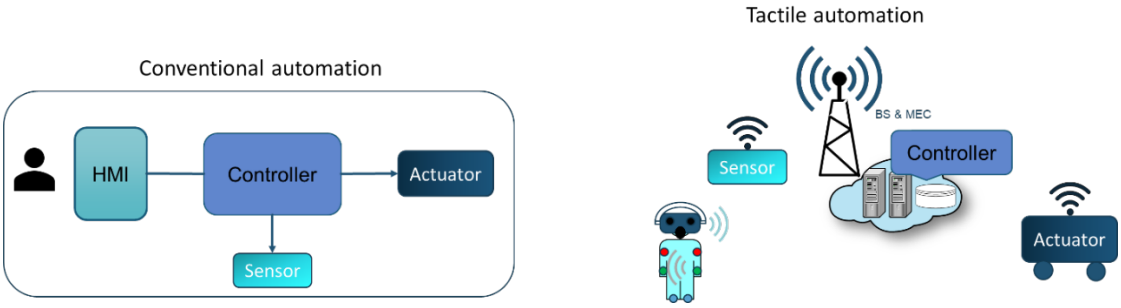

FIGURE 27: FROM FIXED CONVENTIONAL AUTOMATION TO FLEXIBLE TACTILE AUTOMATION.

3.3 AI Based Algorithms

3.3.1 Security in Machine Learning

As the use of Machine Learning (ML) in security critical applications has increased, properties such as security, privacy, robustness, explainability, and fairness have become desirable. Consequently, a fair amount of research has gone into critically analyzing these properties from the perspective of training data, learned models and model behavior. While some of the properties can be achieved in limited scenarios, much work remains to be done to guarantee them in general conditions. Deep learning has seen quite a steep ascent in popularity in the last few years, and consequently it has attracted a lot of attention from researchers who have found it to be particularly vulnerable to security and privacy attacks [124]. In the next two sections we take a brief look at the security and privacy landscape and state-of-the-art in attacks and defenses (non-exhaustive).

Threat model: We define the adversary in the context of attacking an ML system in terms of the attack capabilities it may have, such as:

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

- *Background Knowledge:* the adversary could either have full knowledge of the ML system it is attacking, referred to as a White-box adversary, or is limited to only observing the inputs and outputs, referred to as a Black-box adversary. An adversary with an access level between these two ends of the spectrum is referred as a Gray-box adversary;
- *Active vs. Passive:* the adversary can choose to operate in either an active or passive mode. Active attacks involve deliberate actions on the part of the adversary to undermine the system for its benefits while in the passive mode it only observes the system without tampering with it;
- *Attack Phase:* the adversary can also decide during which phase of the ML pipeline it mounts the attack based on its capabilities and motivation to best optimize the outcome. Attacks might take place while the model is being trained or after the training phase during inference when the model is operational.

Machine Learning Setting: The ML setting is an important consideration for the attacker as it has an impact on the nature of attacks possible:

- *Centralized Learning: t*raditionally ML models have been trained by a single party with all the data pooled together. This setting is often referred to as Centralized Learning;
- *Decentralized (Collaborative, Federated) Learning:* recently Decentralized Learning techniques such as Federated Learning [97] have been proposed where multiple participants train a central model jointly. However, instead of pooling their individual data they train a local model and periodically exchange model parameters, parameter updates, or partially trained models with other participants. Decentralized Learning was motivated by the goal to keep data private, but even sharing just computations on private data in the form of updates could be sufficient to cause a privacy leak.

An overview of the attacks: In this section we survey the prominent attacks in ML considering the threat model and machine learning setting discussed above.

- *Model Evasion*: in this attack the adversary tries to misclassify an input by introducing minor perturbations under some constraint at inference time. Model Evasion has been a very popular attack for vision models where the attacker tries to create an adversarial sample by perturbing the pixels of an image so that it gets misclassified while appearing visually indistinguishable from the original image to the human eye - for example, fooling the classifier to classify a house as a car [106][124][125][126][127]. Similar attacks have been used to modify parts of an image in a readily apparent manner to deceive the classifier, known as adversarial patches or malicious stickers. Such attacks are very pertinent to the domain of autonomous vehicles where a tampered road sign that appears as having been vandalized could lead the ML system to behave adversely [128]. Other applications such as adversarial glasses [129] have also been proposed which can be used for impersonation or dodging a face recognition model;
- *Data Poisoning*: the data poisoning attack is carried by the adversary at training time, where its goal is to contaminate the training dataset in a manner which would help it to subvert the ML system during the inference phase. Such attacks are particularly relevant when labeled data is contributed by numerous entities such as in a collaborative learning setting. Early attacks showed how data poisoning can be used to fool PCA-based malicious traffic detectors and spam filters [91][92]. In [93] has been later demonstrated how to manipulate latent Dirichlet allocation to select relevant topics favoring the adversary. [94] shows that data poisoning is also useful in attacking the robustness of feature selection methods which are a central part of many ML algorithms and are important to ensure computational efficiency and generalization of results;

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

- *Model Poisoning*: [99] presents an attack to backdoor Federated Learning via model poisoning. The attack can be crafted by a malicious participant who introduces updates into the system in a manner which allows it to perform model replacement. Once trained such a model performs as expected in the normal course of action but performs predictively on inputs carefully chosen by the adversary. The attack exploits the secure aggregation mechanism of Federated Learning systems which prevents any analysis of updates. The results were later confirmed in [98] which also showed that under a standard Federated Learning setting if the adversary is aware of defenses deployed, it can still evade them;
- *Jamming Neural networks*: [95] presents a new class of attacks on the availability of ML models at inference time by presenting techniques to craft adversarial samples which could have a significant impact on neural networks where energy consumption or decision latency are critical goals. The authors craft adversarial samples which are designed to maximize energy consumption and latency and are successful in increasing the energy consumption by a factor of 10 to 200 in vision and language-based models. Such attacks can have an adverse effect on the robustness of autonomous vehicles where real-time decision making is critical and on the energy efficiency of mobile devices which often use pre-trained models;
- *Adversarial reprogramming neural networks*: [96] presents attacks to re-purpose neural networks for a task chosen by the attacker by introducing a chosen perturbation to inputs at inference time. The perturbation is same for all the inputs and does not require the attacker to specify or compute the desired output for each test-time input. The attacks work even when the model was not trained to perform the task chosen by the adversary.

An overview of the defenses: We now take a look at how the defenses have evolved with attacks. We note that it has been observed very often in the rapidly evolving field of security and privacy in ML that proposed defenses are quickly broken. In general, defending against adaptive adversaries who are aware of the defense employed is still an open problem.

- *Adversarial training:* one of the key reasons behind the success of adversarial attacks is that the adversary can exploit the uncertainty around the decision boundary in high-dimensional space and nudge samples across the decision boundary to fool the ML model. Researchers have proposed adversarial training as way to provide robustness against such attacks. The key idea is to use attacks to generate adversarial samples targeting the ML system and then training on those samples. Techniques such as retraining the classifier after generating adversarial samples [100], clustering adversarial samples and regular samples [101], generic training against all adversarial samples [102] have been proposed. While such techniques have been shown to be feasible [103][104], such a defense requires generation of a large number of adversarial samples which is computationally expensive and still vulnerable to unknown attacks;
- *Detecting adversarial samples*: defenses have tried to exploit the statistical properties of adversarial samples to detect them [107][100]. Such methods are computationally expensive and require a large amount of data for reliable statistical results. Another approach in this line of work is to train a detector that aims to learn the distribution of adversarial samples [108];
- *Defensive distillation:* [105] proposed defensive distillation as a technique to make the ML model generalize better to samples outside its training set and consequently to adversarial samples. In essence distillation transfers knowledge from a trained Deep Neural Network (DNN) to train another DNN thus improving generalization. This defense was broken [106] and a substitute models was retrained with a substitute model and then attack the substitute model to generate adversarial samples which transfer across models;
- *Randomness-based defenses:* adding randomness to the ML system has been explored as a defense to thwart adversarial perturbations. [109] proposes randomizing the weights of

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

neural networks and selecting a model at random from a set of models during inference time to stop transferability-based attacks. [110] proposes randomizing the input images in vision-based models at inference time by resizing and padding them. [111] introduce region-based classification in which an input data point is randomized to produce points within a hypercube, these points are then classified, and a final decision is made taking a majority vote among the set of outputs. [112] proposed adding a randomization layer to deterministic defenses to defend against adaptive attackers. Their work demonstrates that while randomness makes it more expensive to compute adversarial samples they cannot be fully stopped;

- *Limiting the attack surface: i*nput transformation has been proposed to limit the attack surface available to the adversary and make the model more robust to small perturbations. [113] proposes dimensionality reduction, which limits the attack space available to the adversary. [114] uses the same idea to propose feature squeezing in the domain of images. Some other works have also explored similar strategies [115][116]. Adding random noise and transforming inputs have also been suggested as techniques with similar motivations [108]. Most of these defenses cannot protect against an adaptive attacker and have been shown to be broken [106]. Another approach in this direction is that of *Certified defenses*, where the motivation is to ensure robustness against all attacks. Such defenses aim to provide provable guarantees against bounded adversarial perturbations. Given a classifier and a test input, a certificate of robustness guarantees that no perturbation within the bounded threshold can lead to the test input being transformed into an adversarial sample [117][118][119][120][121]. Although promising, certifiable defenses are bounded to a specific test set [122][123] and do not scale well to large datasets. Additionally, they can be bypassed using alternate definitions of distance which still resemble the original input.

In summary, we find that the cat-and-mouse game of attack and defense is far from over and much work remains to be done to achieve robustness and security in ML models.

3.3.2 Privacy in Machine Learning

The privacy of ML algorithms becomes an important issue if the model or the training data is being shared. In the scenario where both data and model belong to the same party privacy is not a serious concern. However, privacy of the ML algorithms and models needs to be considered under the following scenarios:

- *Machine Learning as a Service (MLaaS):* MLaaS is commonly referred to the setting where one party holds the data and utilizes the services of the other party to train a model on their data. Another popular setting is when the model is trained using the data owned by the same commercial entity and subsequently a query-based interface is provided to the paying customers. Such cases have become very common with big tech companies like Google, Amazon, Microsoft, IBM, etc. offering their services to customers such as app developers or small businesses to train models on their data or to access models trained on rich datasets owned by the model provider. MLaaS helps customers outsource complex ML tasks for which they lack expertise or compute infrastructure, such as facial recognition, spam filtering, or visualization and analysis of their own data. Such a scenario poses several privacy implications:
    - A query-based access to the MLaaS trained model should not reveal information about the data which might compromise its privacy;
    - The model trained by the MLaaS provider is their intellectual property and is often not shared. Providing a query-based access to the model should not reveal details used for training the model;

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

- It might also be desirable in scenarios such as medical use cases that the data used to query the MLaaS model be kept hidden from the service provider;
- It is somewhat counter-intuitive that while the MLaaS model must be widely available as the business model depends upon monetizing pay-per-query access, the model's parameters and training data must remain private. One could argue that with sufficient queries any model can be reconstructed.

- *Decentralized (Collaborative, Federated) Learning*: as described in the previous section such a setting poses unique privacy challenges.

An overview of the attacks: Here we present an overview of attacks exploiting privacy leakage in ML models, [67].

- *Statistical disclosure: i*n 1977 Dalenius proposed a desideratum for controlling statistical disclosure in datasets [64]. The desideratum stated that nothing about an individual should be learnable from the database that cannot be learned without access to the database. When extended to ML models, it states that a model should not reveal any information about an input to which it is applied than what was not already known before applying the model. Such a requirement is too strict and as shown cannot be guaranteed by any useful model [65][66];
- *Model inversion: i*n this attack the adversary's objective to discern sensitive properties of the inputs used to train the model by observing the prediction output of the model. [79][80] introduced this attack by demonstrating how an attacker can use model classification outputs along with a white-box access to the model and some demographic information of the patient who is a part of the training dataset to extract sensitive information about the patient. We note that the inference of sensitive information applies to the entire class and not necessarily to the training dataset, especially in cases where the model has learned a generalized representation of the training data, and not an overfitting to training data outliers;
- *Inferring class representatives: t*his is a general case of the model inversion attack where the attacker having learned sensitive information tries to reconstruct representatives of the class. Such attacks have been demonstrated in the context of Decentralized Learning [81], where an adversary uses generative adversarial networks to recover class representatives. [82] shows that similar attacks can also be extend to privacy-preserving collaborative deep learning protocol proposed [83]. Further work in this direction was done by Song et al. [84] who design ML models that allow extraction of memorized training data via a black-box access without affecting the accuracy of the model significantly. Similar attacks have also been presented in [85][86];
- *Membership inference: t*he adversary tries to infer whether a given data point was used to train a given model. This can be quite privacy invasive depending upon the training task that the data is being used for. Existence of membership leakage is a sign of poor privacy and indicates that further privacy attacks may also be feasible [67]. On the positive side such attacks can be used to establish data provenance [68] and misuse of data for training a model for tasks not explicitly agreed to. [65] exploited the disparity in the model's prediction confidence values on data points included in the training set compared to ones those that were not, to mount a membership inference attack. The attack was later improved [69] by making it more generic. Other works have also proposed similar attacks [70][71]. [72] shows that such attacks are not just applicable to discriminative models but can also be extended to generative models. The attack was further extended to Federated Learning, [73], where the adversary is a participant in the collaborative learning protocol;
- *Property inference: t*he adversary tries to infer properties that apply to a subset of the training data but not to the whole class of a dataset. [73] presents such attacks in the context

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

of collaborative or federated learning, while [87] demonstrates these on fully connected neural networks where the adversary has a white-box access;

- *Model extraction:* such an attack is particularly relevant for MLaaS, here the adversary's goal is to reconstruct a close approximation of the MLaaS model by querying the model and observing its predictions on chosen inputs. [75] presented an attack to steal the model parameters in the black-box adversarial setting. Subsequently, attacks have also been presented to steal model hyperparameters (training configuration) [76] and deduce model architecture [77];
- *Functionality stealing: t*his attack is similar to model extraction, however, in this setting the attacker's goal is more modest; to construct a knock-off model purely based on query responses [74][78].

Privacy Enhancing Technology for Machine Learning – Privacy preserving solutions in the context of ML are still in nascent stages. The solutions proposed can be broadly categorized into three approaches:

- *Cryptography based approaches: c*ryptography can provide some desirable properties for an ML system, especially in the scenario of MLaaS where the server wants to keep its model confidential, and the client wants to keep its input data private. Secure multi-party computation (SMC) and fully homomorphic encryption based approaches have been employed to provide confidentiality in trained ML models such as matrix factorization [101], linear classifiers [145][146], decision trees [147][148], linear regressors [149], and neural networks [150][151][152][153]. SMC has also been widely used to provide confidentiality in Federated Learning. [153] proposes using SMC to train a model on encrypted client data by distributing it to two non-colluding servers. [150] uses secure aggregation techniques based on SMC to train a central model in the context of Federated Learning: the model updates are encrypted and can only be decrypted by the central server upon aggregation. [154] proposes using multi-party lattice-based cryptography to preserve the confidentiality of the training data, the model, and the evaluation data in the Federated Learning setting. We note that solutions that guarantee confidentiality can still leak privacy;
- *Differential Privacy: a*chieving DP is the gold standard in providing access to information without leaking privacy. ML systems have been proposed to satisfy DP to preserve privacy. In the context of neural networks, noisy stochastic gradient descent has been proposed to provide privacy guarantees [155][156]. Another approach in a similar vein is known as Private Aggregation of Teacher Ensembles (PATE) [157][158]. PATE trains a student model using noisy aggregation of predictions from an ensemble of teacher models which allows providing strong privacy guarantees. DP based approaches have also been proposed in a decentralized learning setting: [83] proposes collaboratively training a ML model where independent entities share subsets of noisy model parameters during training. There are also proposals for training Federated Learning models with DP for language models [159] and vision models [159];
- *Trusted Execution Environments: a*nother strategy to provide privacy in ML is by using Trusted Execution Environments (TEEs), such as Intel SGX or ARM TrustZone. These approaches are especially pertinent in the domain of MLaaS. Some ideas that the research in this direction have explored are training ML algorithms in the SGX [161][162][163], getting privacy as well as performance benefits by carefully dividing neural network computations between trusted and untrusted devices [164], and guarded offline deployment of MLaaS where the models are run on the client device [165]. TEEs provide a significant speedup compared to purely cryptographic solutions for privacy.

In summary, assuring privacy in ML is still an open problem, and significant community effort is needed to surmount it. Moreover, privacy is not a property that is just desirable in isolation and

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

must co-exist with other properties. Researchers have discovered that securing models against adversarial samples makes them vulnerable to privacy risks. For example, robustness against perturbation of individual data samples makes the model vulnerable to inference attacks [166]. Bagdasaryan et al. [167] found that ML models trained with DP show a drop in accuracy for underrepresented classes and subgroups, thus it seems that the cost of privacy is borne disproportionately by some members. This tension between the desirable properties of ML such as Security, Privacy, Robustness, Explainability, and Fairness, needs to be resolved to design a holistic system.

3.3.3 Predictions and Estimates

Artificial Intelligence (AI) and Machine Learning (ML) -based predictive analytics are increasingly used in supply chain applications, often taking advantage of various sensor and IoT technologies. Below we outline current examples of such applications.

As outlined in the previous sections, there are many applications for automated vehicles in industrial and supply chain scenarios. These range from the use of AGVs, robots, and other automated machinery in smart factories and warehouses, to use of automated vehicles in multimodal transport. Machine learning methods are one of the key enablers of automated vehicles operating in complex environments without continuous human supervision or control. In many scenarios, ML models are a necessary component in autonomous vehicle sensor fusion systems, which need to maintain situational awareness, or a model of the surrounding environment, to enable planning of navigation and other interaction with the environment. Key operations enabled by modern ML models include detecting and classifying relevant objects, such as pedestrians or traffic signs, from various sensor data.

Computer vision methods used in autonomous vehicles are reviewed in detail [191] , as well as the role of machine learning in sensor fusion and navigation systems of maritime autonomous surface ships [192].

Other applications of machine learning and predictive analytics in autonomous vehicles include navigational planning, identifying and predicting actions by other entities in the environment, and optimizing maintenance schedules and energy consumption based on continuous sensing and monitoring the operation of the vehicle. These last applications, predictive maintenance and predictive energy management, are applicable in supply chains also outside autonomous vehicles, and are commercially viable for existing land, sea, and air transportation.

Predictive maintenance is a key application area for combining IoT sensing for continuous data collection with predictive analytics. In many industrial applications this provides potential benefits in increased reliability of equipment and optimization of maintenance schedules. Supply chain examples are considered. for aviation in [193], and for maritime systems in [194]. Commercial examples are offered by multiple companies providing supply chain equipment such as trucks [195], and cranes [196].

Predictive energy management is another example of predictive analytics combined with IoT data, enabling the optimization of vehicle operation in multimodal supply chains. This is in many cases considered as an optimization component for autonomous and electric or hybrid vehicles. As an example, [197] considers predictive energy management for autonomously operating hybrid trucks. Similarly, [198] considers predictive control methods for energy management in hybrid vessels.

Regarding more generic examples of predictive models in supply chain applications, a recent survey on the use of predictive big data analytics in supply chain demand forecasting is presented in [199]. There the authors outline sources of information and data types required for predicting demand, also including the role of IoT data in such prediction. Reviewed analytical approaches to demand forecasting include time-series forecasting, clustering, K-nearest-

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

neighbours, neural networks, regression analysis, support vector machines, and support vector regression.

Although not focusing directly on the IoT aspect, Nvidia in [200] describes an interesting supply chain optimization application using machine learning. This focuses on how Zalando used neural network-based machine learning to optimize the picker routing in their warehouse operations. The basic principle considered is to learn the output of a computationally demanding optimization algorithm using neural networks to implement the optimization more efficiently.

As outlined by the examples above, there are many supply chain applications which focus on use of predictive analytics and ML methods with IoT data from individual systems or sensors. A possible evolution of such use cases is a System of Systems approach, where multiple intelligent systems are combined in a holistic manner. For example, systems for warehouse operations optimization, machinery automation, and vehicle schedule optimization could be integrated (e.g. combining maritime, port, and truck IoT data) for predicting logistics process flows and optimizing overall resource usage and scheduling over the multimodal logistics chain. Especially in multimodal logistics, currently siloed organisations and information systems hinder the formation of holistic information and related optimization of operations. Data is not collected in one place to enable situational awareness across transport modes and is not shared efficiently between supply chain actors. This results in suboptimal planning. Currently for example autonomous vehicles could not be used optimally in logistics without broader systems integration, situational awareness, and predictive models covering the entire supply chain to some degree.

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

4 Ingenious Proposed Solution

In this chapter the proposed architecture to be implemented within iNGENIOUS project is presented and discussed. All the architectural components involved are treated separately in relation to the main objectives of the project as well as to the state of the art discussed in the Chapter 2. Although the proposed architectural components are listed below, not all of them are related to the demonstration and validation of the Use Case *DVL/Cross-DLT*. Instead, they are related to the implementation activities expected to be carried out in the WP5.

4.1 Smart IoT Gateway

As already exposed in previous sections, the proliferation of small interconnected devices has caused the appearance of multiple new technologies related to the IoT world. For this reason, it makes sense to introduce a new device capable of ease of the interoperability between different groupings of IoT devices (i.e. in the form of independent meshes or networks) and logical higher-level systems. The Smart IoT Gateway (GW) is a physical element with multiple interfaces that allows end-to-end M2M communication, having as main responsibility to ensure correct and secure routing of messages between the sensors or any other deployed IoT devices, and M2M interfaces to interoperable layer. To achieve this goal other sub-goals - not obvious at the first glance - are necessarily needed to be achieved for a correct implementation:

- *Communication integrity*: to ensure no data has been lost or modified during the message transmission or reception;
- *Message translation*: between different formats and protocols;
- *Secure link*: to provide protection to message tampering or electronic eavesdropping;
- *Resilience*: to mitigate connectivity outages or any other non-nominal behaviors;
- *Flexibility*: to allow the interconnection of multiple physical interfaces.

To achieve this, the Smart IoT GW has been divided into functional blocks with an implementation that isolates them as independent containers, following a philosophy quite similar to micro-services architecture. The blocks are presented in Figure 28 and covers the points exposed earlier.

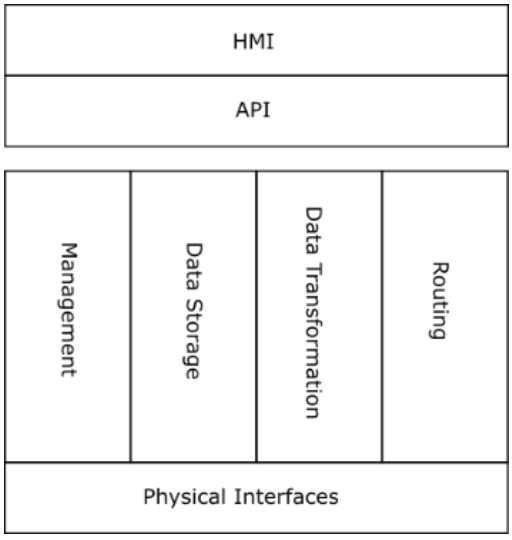

FIGURE 28: SMART IO GW BLOCK DECOMPOSITION.

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

Connected to the different physical interfaces the following services are available in the Smart IoT GW:

- *Management*: management of the GW includes the current status of the device, internal configuration, local logs and firmware update. It will include also some limited control over the subsystems;
- *Data storage*: databases (time series and NoSQL). All data coming from the sensor space is time stamped and stored in a time series database that has the capacity of performing some operations such as data aggregation, statistical calculations, and time consolidation (i.e. decimate values and down-sample the data);
- *Data transformation*: message translation between formats and protocols. This does not include just direct or indirect mapping of fields, but also this service will be responsible of encapsulating related messages that came from independent sources or to perform compression when possible;
- *Routing*: incoming message routing through the gateway to a different interface, following predefined rules, respecting priorities and in a secure way.

An Application Programming Interface (API) is available for external interoperability and automation. A simple Human-Machine Interface (HMI) is provided for local data monitoring.

4.1.1 Smart IoT Management

The Smart IoT GW will be able to provide endpoints for local configuration and management. An administrator will be able to control and configure the local operations in the gateway such as i) Stopping and restarting services; ii) Change configuration for each service; iii) Enable or disable specific operations or interfaces; iv) Update firmware; v) Add or delete routing, priority, or security rules. The Smart IoT Management service will be able to monitor and show data related to the execution and performance of running services. This includes:

- Metrics associated to routing, security, transmission errors and statistics;
- Health status of THE VARIOUS SERVICES, INCLUDING LOGS AND AUDITING.

4.1.2 Smart IoT Data Storage Services

The Smart IoT GW will have a limited storage capacity to buffer information and to store data to avoid information loss in case of communications outage. This storage will also have basic configuration data.

- NoSQL document-type persistent database that will contain decomposed messages;
- Time Series database that will store time stamped data covering a configurable time span.

4.1.3 Smart IoT Data Transformation Services

This component will provide flow-based, sequential execution of processes that will retrieve messages stored in an incoming queue, process them, and put it in an outgoing queue. The available message operations include:

- Message translation (data mapping between standards);
- Implementation of multiprotocol Interoperability (generation of specific protocol elements like headers, Cyclic Redundancy Check (CRC) or any other element not related to the payload data);
- Compression (when needed or available).

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

For convenience, the IoT GW will use internally the MQTT format. This will allow simple transformations between IoT formats. Externally, the following physical interfaces will be available: Wireless: Bluetooth, Wi-Fi, LoRa; and Wired: Ethernet, SPI, I2C.

The following protocols are supported in the sensor space:

- Plain TCP/IP (in the form of network sockets);
- MQTT (over TCP/IP);
- Hypertext Transfer Protocol (Secure) (HTTP(S)) including Representational State transfer (REST) or Simple Object Access Protocol (SOAP).

In the M2M space, the IoT GW will support oneM2M standard, integrating one of the well-known implementations (oneMTC or Eclipse OM2M).

4.1.4 Smart IoT Message Routing

The Smart IoT GW will have a process that will route messages based on priorities and on availability of recipient's endpoints. This will include:

- Multiple interface routing from an arbitrary number of incoming or outgoing interfaces;
- Secured endpoints.

4.1.5 Smart IoT API

The Smart IoT GW will have as a transversal element to the presented services an API that will allow external systems to interact and operate the GW allowing them to perform: remote MONITORING; and automation.

4.1.6 Local HMI

The simple HMI is provided to allow local interaction of the device. This include: local monitoring and control.

4.1.7 OneM2M implementation

The Smart IoT GW relies in Eclipse OM2M [201] implementation of the oneM2M standards. Essentially, OM2M implements components and applications for each of the standards described in oneM2M. Among its main features are:

- Not only compliant with oneM2M but also with SmartM2M [202];
- Open Service Gateway initiative (OSGi)-based architecture extensible via plugins;
- Implements a restful API with a generic set of service capabilities;
- Provides oneM2M services such as machine registration, application deployment, container management, resource discovery, access right, authorization, subscription/notification, group management and non-blocking requests.

It integrates with the Smart IoT GW flows and processes via its plugins and reduces the complexity of the architecture.

Another oneM2M framework to be used in the design is openMTC [203] It provides the same common interface, but it has a very convenient deployment system as containers, being aligned with the overall deployment strategy for the Smart IoT GW architecture.

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

4.2 Interoperable Layer

In this section the main components of the interoperability layer are presented and discussed in relation to the main objectives of the iNGENIOUS project. The main aim of the interoperability layer is to cover two different interoperability aspects foreseen by the project. On one side the integration of different M2M platforms for machine-to-machine communications and on the other side the integration of different DLT solutions for cross-DLT interoperability. In order to cover the above-mentioned interoperability aspects, two core components of the interoperability layer are going to be described in the next sections:

- *Data Virtualization Layer*: to guarantee M2M interoperability as well as integration of data coming from different data sources (e.g. MANO platform, Port Community Systems, Awake.AI Smart Port platform);
- *Cross-DLT layer*: to guarantee the integration and interaction with different available DLT solutions ensuring secure data storage and management (e.g. Ethereum, Bitcoin, IOTA and HyperLedger Fabric).

4.2.1 Data Virtualization Layer

In order to achieve M2M interoperability, Data Virtualization has been chosen as the main approach to be used within the project. For this purpose, the DVL will act as an intermediate layer between data sources and Cross-DLT layer in order to collect and aggregate data coming from different M2M platforms according to a given data model that will be defined during the implementation phase of the project. The main aim of data aggregation procedure performed by the DVL is to build up different type of events relevant for the maritime domain such as Gate-In, Gate-Out, Vessel Arrival and Vessel Departure (other events could be also considered during the life time of the project such as loading onto a vessel and unloading from the vessel). Within iNGENIOUS project, M2M platforms and data sources like Port Community Systems will be used to feed this architectural component with a sufficient set of data for the events definition. The DVL will collect these data, aggregate and then share them with the Cross-DLT layer (based on Trust-OS solution) as well as with MANO and Awake.AI platforms for network performance and predictive models' analysis. For a better comprehension of the functionalities provided by this layer, the following functional schema to be used in the project is proposed and discussed below, see Figure 29.

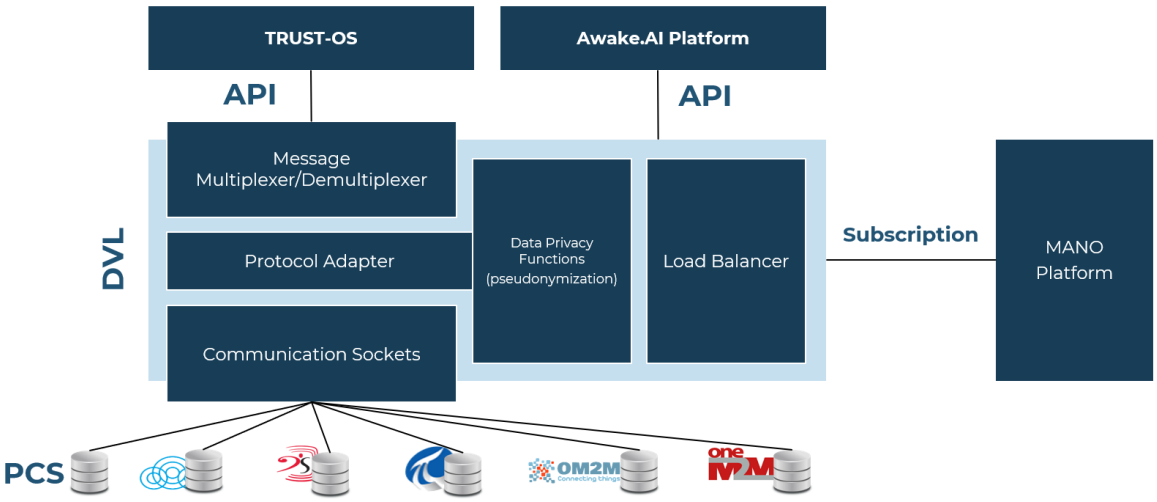

FIGURE 29: DATA VIRTUALIZATION LAYER - FUNCTIONAL SCHEMA.

As depicted in the picture above, we can identify the following components that will communicate with DVL:

M2M Platforms and other data sources

Different M2M platforms, in particular those from the consortium, are expected to be used to feed the DVL with relevant raw data coming from the underlying device layer. More specifically, the following M2M platforms are considered: Mobius OneM2M from CNIT, Eclipse OM2M from SES, Symphony from NXW, PI System OSIsoft from FV and NB-IoT Platform from CMC. Since the main role of the DVL is to provide a set of concise events to be shared with the Trust-OS platform for their distribution across different DLT solutions, data coming from above mentioned M2M platforms will need to be also combined with data coming from other data sources, such as Port Community Systems, so that events can be completely defined. In order to guarantee the interaction with M2M platforms by retrieving relevant data, the DVL will need to have the following basic functionalities:

- *Communication Sockets*: will allow to establish bidirectional communication between DVL and M2M platforms (e.g. stream sockets, datagram sockets, raw sockets, sequenced packet sockets, etc.);
- *Protocol Adapter*: will allow to receive data from the underlying sockets and translate them into a common format according to events' data model by using a common protocol (each M2M platform is using its own communication protocol);
- *Data Privacy Functions*: will allow to split collected data in Personal data and Non-Personal Data subsets so that personal data can be either pseudonymized or anonymized (when needed) in order to guarantee compliance with GDPR regulations; an example of such data is the truck plate number during Gate-In and Gate-Out operations;
- *Message Multiplexer/Demultiplexer*: after data discrimination, this functionality allows the multiplexing/demultiplexing of the messages in order to deliver the information to Cross-DLT layer;
- *Load Balancer*: will allow preventing bandwidth saturation to mitigate situations with huge amount of data from a given M2M platform.

During the implementation phase, two operative environments are going to be considered: Port of Livorno and Port of Valencia. Although the above mentioned M2M platforms are currently used in different operative domains, within iNGENIOUS project these platforms are expected to be instantiated in maritime domain so that only data useful for the events definition can be collected.

MANO Platform

iNGENIOUS MANO platform is conceived as a cross-layer network slice orchestration framework with three main management functions that provide a complete integration of 5G NR, NG-IoT and edge computing technologies as part of end-to-end network slices. MANO layer is expected to be integrated with DVL in order to enable the management and orchestration of network slices to support supply chain scenarios. The main aim of this integration is to be able to foresee the network requirements at both core and edge sides of the underlying network infrastructure ensuring SLAs for covering use cases to be implemented in Livorno and Valencia seaports. For this purpose, MANO platform will expose AI/ML functionalities to consume heterogeneous monitoring data from different sources, including the network slice orchestration layers themselves as well as IoT and M2M data retrieved from DVL. To this end, MANO platform will use a subscription mechanism supported by DVL so that data related to network utilization can be received and the network resources usage optimization can be performed.

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

Awake.AI Platform

On one side the DVL will interact with the Cross-DLT layer in order to provide information based on identified events in the maritime domain (e.g. Gate-In, Gate-Out, Vessel Arrival and Vessel Departure). On the other side the DVL will also share the information coming from the Port Community Systems and M2M platforms with the Awake.AI platform, allowing the ML-based predictive models to enhance the situational awareness in logistics processes (e.g. trucks schedules) as well as to identify potential bottlenecks in the supply chain. In order to allow this communication, the DVL will be exposed as an endpoint so that Awake.AI platform can retrieve and process data of interest by means of SQL requests (based on the defined data model). In this case historical data will also include personal data such as truck plate number and for this reason a proper pseudonymization technique will be necessary.

Cross-DLT Layer (Trust-OS)

The Cross-DLT layer (implemented by means of Trust-OS solution) is an intermediate layer between DVL and different DLT solutions brought in the project (e.g. Bitcoin, Ethereum, IOTA and HyperLedger Fabric). The main aim of this integration is to provide defined events to the Cross-DLT layer based on data collected from M2M platforms (including other sources such as the Port Community Systems) as well as to distribute them over different DLTs for secure data storage according to their own capabilities (e.g. hash and data storage) so that End Users from the Supply Chain can get access and control only over their own data. By means of this, End Users can benefit from native cryptographic capabilities provided by available technologies (e.g. Proof-of-Existence). Although each DLT has different capabilities for secure data management (e.g., consensus mechanisms, transactions, wallets and smart contracts management, data storage over ledgers, etc.), as a minimum requirement the storage of a data hash is expected to be guaranteed for all available DLTs.

In order to fulfill this interaction, the DVL is expected to be exposed as an endpoint. It will be responsible for data retrieval from different data sources such as M2M platforms. Trust-OS will then perform specific SQL requests to the DVL in order to obtain the needed data according to a previously defined data model. Further details about the Cross-DLT layer are provided in the following sections.

Data aggregation

Data aggregation operation will be performed at DVL level. As mentioned within previous chapters, DVL will collect data coming from different data sources (e.g. M2M platforms and external systems such as PCS), extract and aggregate them according to a given maritime events data model. On one side these data will be used to feed platforms such as MANO platform and Awake AI platform. On the other side, these data will allow to have a complete dataset for the events definition in the maritime domain so that they can be shared with Trust-OS platform and distributed across different DLTs for secure storage. In order to produce consistent events from the collected data, it has been decided to rely on events data model defined and used by the TradeLens platform 00. For the time being, we are going to consider the following events in the maritime domain, further discussed in iNGENIOUS deliverables D5.2 and D5.3:

- *Gate-In*: defines the actual time of arrival of a truck carrying the transport goods at a gate both on export and import side and for full and empty transport goods;
- *Gate-Out*: defines the actual time of departure of a truck carrying the transport equipment both on export and import side and for full and empty transport goods;
- *Vessel Arrival*: defines the time (expected or actual) of arrival of the vessel in seaport (at berth);
- *Vessel Departure*: defines the time (expected or actual) of departure of the vessel from the seaport (from berth).

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

In iNGENIOUS project, the main aim of data aggregation procedure performed by DVL is to extract only data (from the underlying data sources including M2M platforms) that are relevant and sufficient for the complete definition of the considered events. Moreover, based on different use cases (e.g. UC *Ship* and UC *Port Entrance*), during the project lifetime it will be also assessed the possibility to extend the list of considered events, by including also other maritime events such as Loading and Unloading of the container from the vessel. Further details about discussed events can be found by consulting the official TradeLens documentation 0.

Privacy management

The DVL will act as pseudonymization entity introducing a function that protects personal data coming from different M2M platforms. This function will be configurable to adopt different pseudonymization techniques that could change during the life of iNGENIOUS platform. One specific technology which can be applied is differential privacy as described in Section 2.5.2 Anonymous Credentials. A deep analysis will be performed on foreseen managed personal data to understand the best technique and policy to adopt, considering the protection versus usability of pseudonymised dataset. Furthermore, an analysis of an encrypted structure to be used to protect the conversion table retained by DVL and strong mechanisms of key protection, such as an HSM, will be used.

4.2.2 Cross DLT

Trust-OS

Trust-OS is both an SDK and a powerful, easy-to-use set of decentralized network modules deployed in several blockchain networks. Thus, it is a decentralized software layer based on Smart Contracts. Trust-OS exposes basic trust functionalities through a series of JSON and API libraries including traceability, token issuance, agreement resolution, certificate issuance and decentralized identity that flow and interact seamlessly with existing platforms, networks and ecosystems. In short, Trust-OS acts as a trusted operating system and connects business applications directly with the functionality they require from the blockchain. Trust-OS software can be deployed on any Hyperledger Fabric based blockchain network. Once deployed, an HTTP API is published so that it can be used from any programming language. The API hides much of the complexity of working with a Blockchain, but its direct use still requires low-level knowledge. For that reason, Trust-OS also includes some libraries to be consumed by applications and other Smart Contracts. Therefore, Trust-OS commands can be invoked from outside the blockchain through its APIs or from other Smart Contracts inside the blockchain itself. Trust-OS APIs do not speak the blockchain language (send, validate or verify a transaction, query a block, provision or obtain gas at an address, compile, deploy or execute contracts, manage and safeguard cryptographic material, etc.) but the language a customer understands (create an asset and update its status, add account activity, verify an identity, create or transfer a token, etc.).

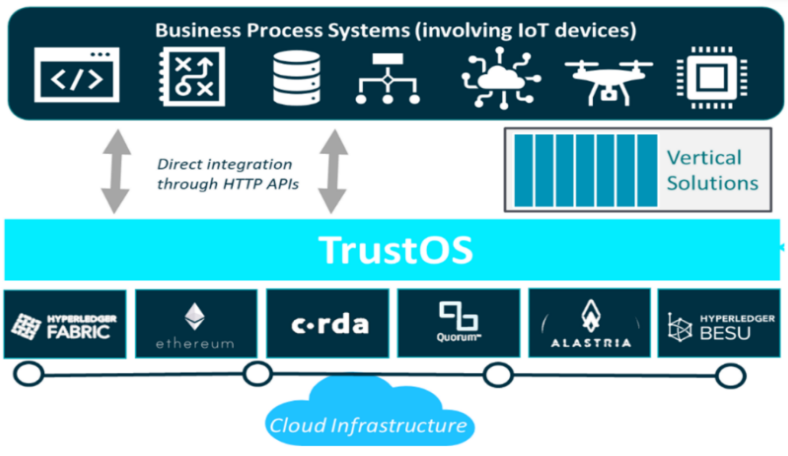

FIGURE 30: TRUST-OS ARCHITECTURE.

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

As depicted in Figure 30, Trust-OS consists of the following modules:

- *Track module* to create and follow the life cycle of the assets in the Blockchain. An asset is a digital representation of a real object in the business world. Using this module, you can record immutable information that uniquely characterizes the asset and update its attributes inherently linked to the instant it was created/updated;
- *Token module* allows to easily create transferable value on the network. Applications can issue tokens, transfer them or lock them by keeping at any time an absolute and certain control of the property or rights of use over it. This is an adaptation of the Ethereum ERC20 standard;
- *Settle module* generates a general structure for reconciling activity records or aggregation operations based on a set of conditions defined in a smart contract;
- *Trust* is a horizontal module that allows to make snapshots of the assets generated in the other modules and to register them in the public blockchain;
- *Cert module* is used to create, sign and verify digital certificates on the blockchain;
- *TrustID* is a standalone identity module for Trust-OS. It Is a solution for Decentralized Identity Management where users and services are identified through a decentralized identifiers (DIDs).

The decision to implement these modules was due to the fact that these functionalities are the most demanded in the blockchain technology environment. However, in Trust-OS it is possible to deploy any chaincode that offers a different logic. The TrustOS-based solution offers the following advantages:

- It is DLT agnostic, it is not necessary to have low-level knowledge about DLT to take advantage of its characteristics;
- APIs are exposed via HTTPS, so they can be used by any programming language;
- It is possible to define rules or alarms on certain asset values. If a rule is broken when a value is updated, an event is generated (at blockchain level) and issued to notify of what has happened.

On the other hand, there could be the following disadvantages:

- Support for other DLTs platforms-based networks such as CORDA are not yet available but planned in the long term.

Track module

Through this module it is possible to define digital assets of different types that correspond to the assets of the business process. It allows to create and follow the complete life cycle of the assets in the Blockchain. The flexibility in the asset management is provided since this module handles JSON objects so the asset structure will be defined according to the business requirements. The main advantage is that a complete view of the traceability of the asset is provided. It is a very simple module to use and the only requirement is to model the real asset with the necessary properties to generate its digital twin within Trust-OS. The main element is the asset. Its structure is shown below:

```
{
"assetId": "1",
"data":{},
"metadata":{}
}
```

- *Asset identifier*: Each asset is differentiated by a unique id;

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

- *Data*: This property identifies the immutable data of the asset process;
- *Metadata*: Records all changing information throughout the asset's life cycle.

Information is never overwritten, when a new update arrives to the platform, a completely new object is generated allowing full traceability at the end of the process. The following is an example structure representing a device-type asset:

```
{
  "assetId":"Device001",
  "data":{
    "company":"ACME Robotics S.A",
    "model":"X1.0"
  },
  "metadata":{
    "position":{
      "x":23.34,
      "y":-24.22
    },
    "temperature":"20.0"
  }
}
```

Main methods of the module are:

- Create a digital asset;
- Get asset from the blockchain identified by assetId;
- Update the mutable property of an asset;
- Get all transactions for the whole life cycle of the asset;
- Transfer the ownership of the asset. The user has to be the owner;
- Authorize user access for an asset;
- Unauthorize user access for an asset;
- Add rules to monitor asset parameters;
- List all the assets of a user.

Trust Points

Once the raw data that arrived to Trust-OS is stored, it is possible to generate a proof of its integrity in order to store it in other DLTs and for a future validation that the data was not modified, proving its integrity through the time. The tool that is going to be used for this purpose of generating a proof of integrity is called Trust Points and it is part of the Trust-OS software. Trust Points are part of the Trust module described before. A Trust Point is an object that contains information necessary to export the integrity of a set of transactions in the private blockchain, such as the one Trust-OS is built on, to other blockchain either public or private like Ethereum or Bitcoin, adding this way an extra layer of security and trust over a private blockchain. The Trust Points uses the Merkle Tree concept and are nested among them in the same way as in a blockchain, pointing to the previous object. The creation of a Trust Point is described in Figure 31. To create a Trust Point, first, a set of transactions have to be defined. This is done by inserting an initial time and an end time. These make a time range for selecting the transactions that are going to be part of the Trust Point, becoming each one a leaf of the Merkle Tree. The transactions are the ones corresponding to a specific asset created with Track API and its updates. With all these the tree is built and returns a Merkle Tree root hash, that will be used in the future for validating the integrity.

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

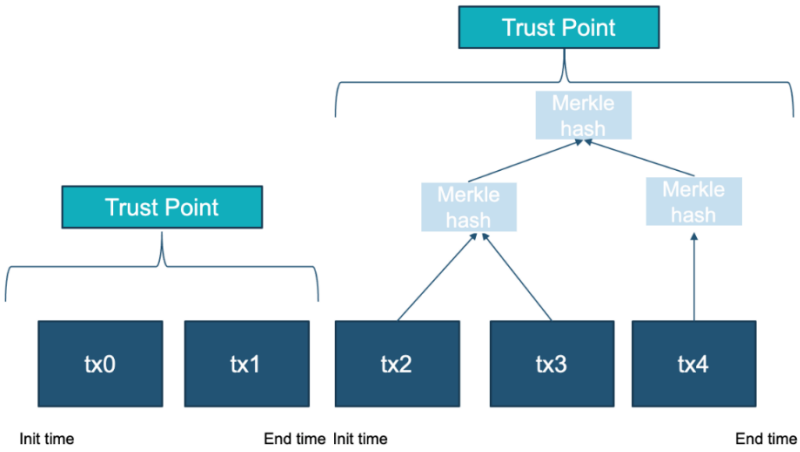

FIGURE 31: TRUST POINT DIAGRAM.

But a Trust Point is not only composed of the root hash of the Merkle Tree, it has other fields containing information relevant for future validations:

- *Asset ID*: identifier of the asset for which the Trust Point is going to be created;
- *Init*: timestamp of the previous Trust Point or zero, if this is the first Trust Point;
- *End*: timestamp at which the Trust Point is made;
- Merkle Tree root hash of all transactions in the interval;
- Current trust point transaction in Hyperledger Fabric;
- Previous trust point transaction in Hyperledger Fabric;
- Hash of the previous Trust Point;
- Hash of the Trust Point to verify the integrity;
- Other DLTs information relevant to identify and verify the Trust Point according to what has been stored on them.

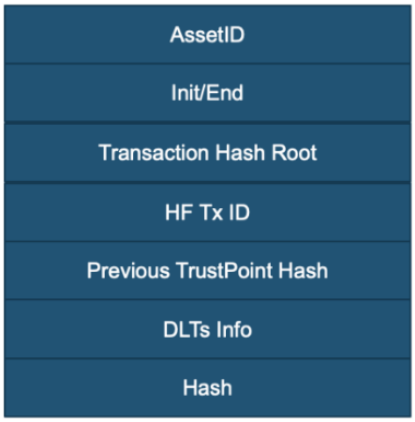

FIGURE 32: TRUST POINT STRUCTURE.

As mentioned before, each Trust Point has a field that contains the hash of the previous Trust Point. This is intended to be used also for integrity verification of the Trust Points. As each Trust Point references the previous one, they are building a validation structure similar to the one that blockchain has, having to modify all the previous ones (already stored in different blockchains) in order to modify one. This makes it really difficult to alter the data integrity proofs, having then a tamper proof system for data integrity verification by itself.

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

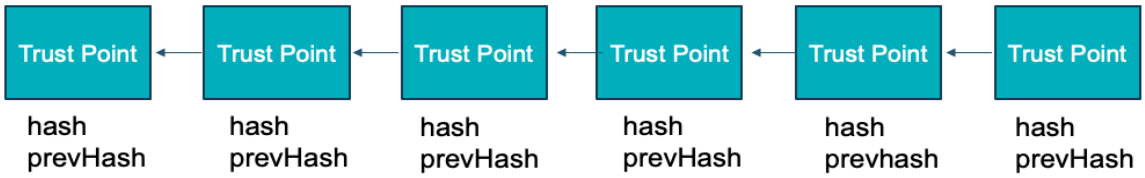

FIGURE 33: TRUST POINTS REFERENCES EACH OTHER WITH THE HASH.

Connection with the APIs

In this part of the document, a flow of the information from its creation, as an asset with Track API, to the storage of its verification proof in the different DLTs, is presented. For this purpose, an example workflow is described. The example consists of a device that periodically sends location and temperature parameters that has been modelled as a digital asset.

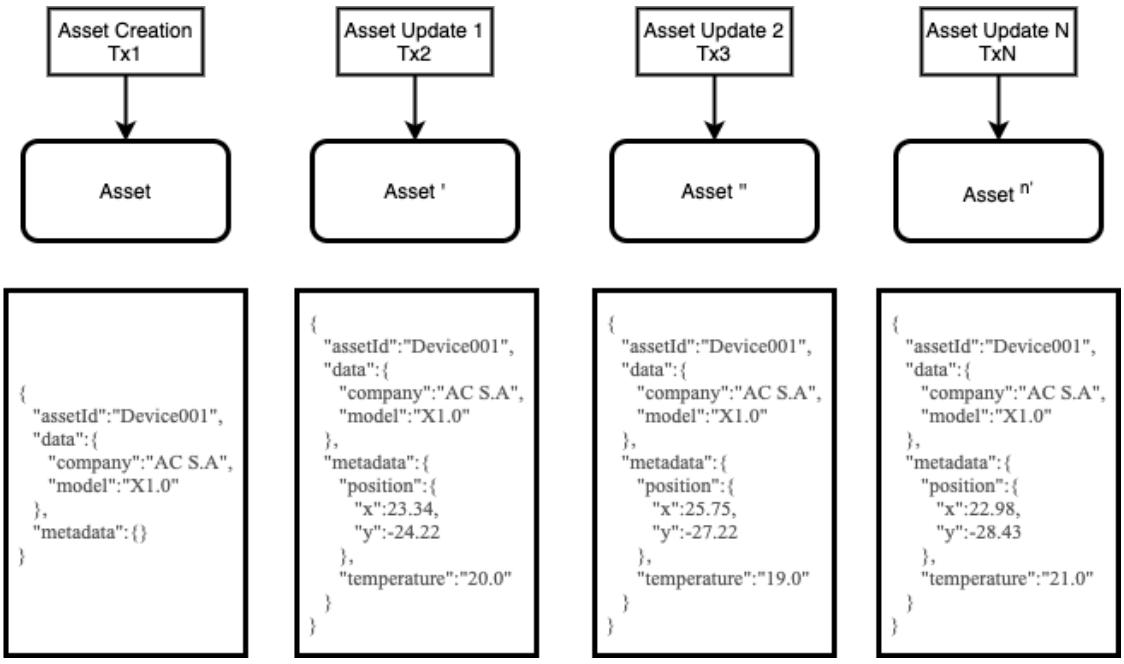

FIGURE 34: ASSET CREATION AND UPDATE.

In the first part of Figure 34 the asset is modelled and updated following these steps:

- First of all, it is necessary to create the digital asset using “create” function;
- Each parameters update is added to the metadata property through “update” function;
- At the end of the process, the transaction history constitutes the complete traceability of the asset. This feature is provided by “transactions” function.

Once all the information (the raw data coming from the DVL) has been stored in Trust-OS as a digital asset, a Trust Point can be created as shown in the next picture. It is the same process explained before, each transaction (the asset creation and the asset updates) becomes a leaf of a Merkle tree, the field that keeps a proof of the data integrity.

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

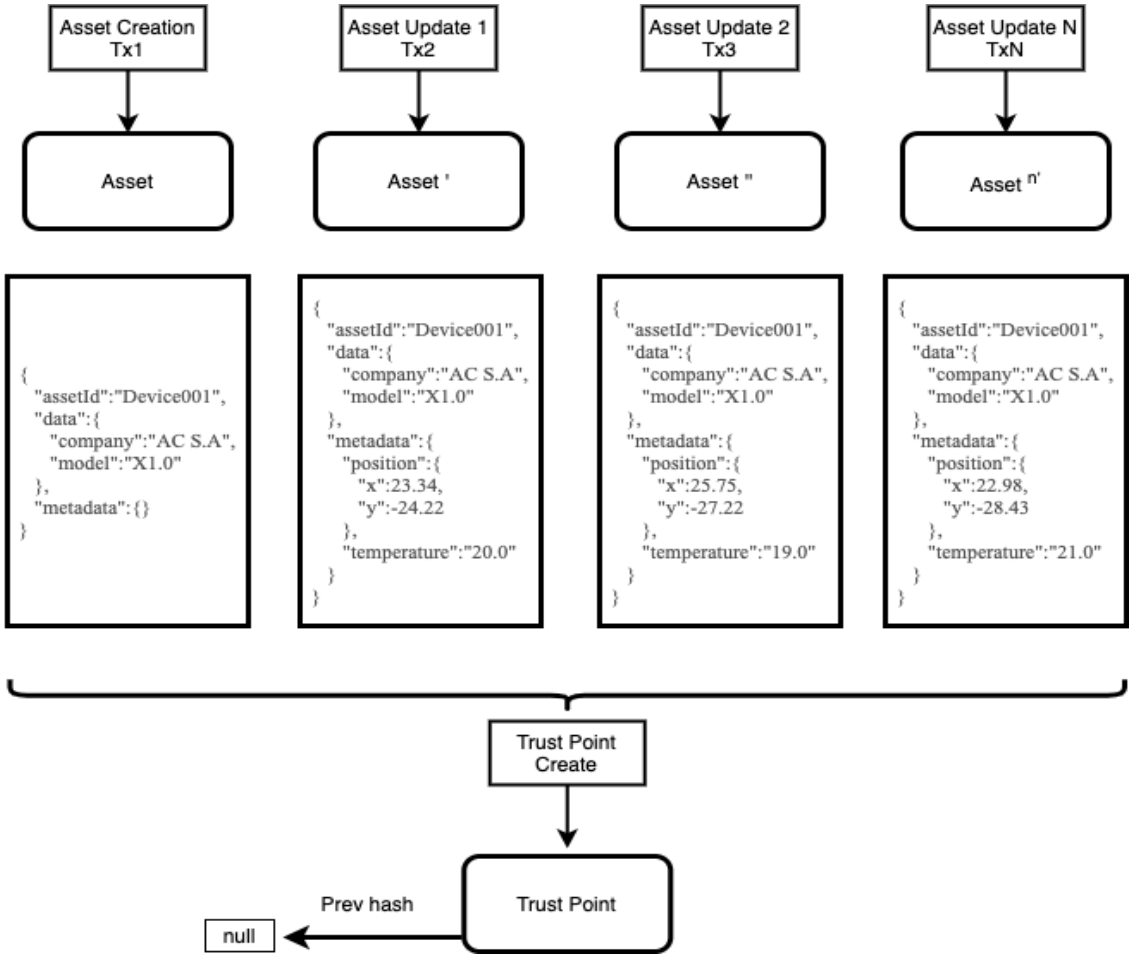

FIGURE 35: TRUST POINT CREATION.

Now there is a Trust Point containing a proof of the data stored on each transaction regarding the specific asset within a range of time. A particular point about this first Trust Point is that as it is the first one the previous hash field points to null because there is no more Trust Points associated to this asset. When more updates are done over the asset metadata, another Trust Points may be created and so on. These are going to reference to their previous ones, building the Trust Points chain mentioned before that adds an extra layer of trust over the already trusted blockchain network. This is depicted in the following diagram:

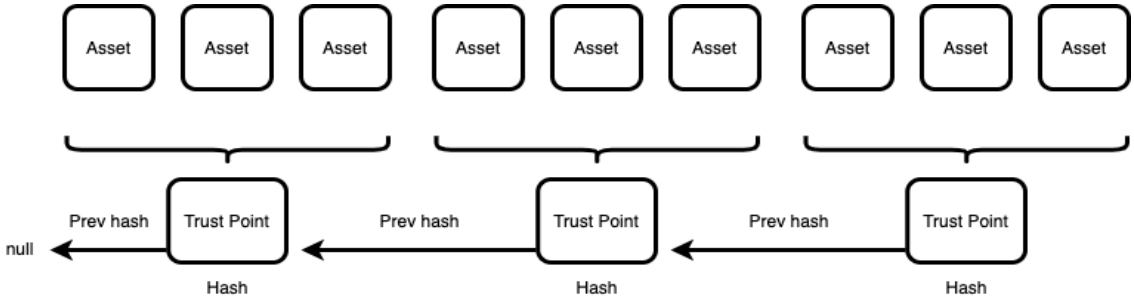

FIGURE 366: TRUST POINTS CHAIN.

All these data, assets and Trust Points are stored on the private blockchain on which Trust-OS is based on, Hyperledger Fabric, but this may be improved by storing the integrity verification proof on different DLTs, such as Bitcoin, Ethereum, IOTA, or even another network of Hyperledger Fabric managed by another actor. In order to achieve this, there is a function inside the Trust module of Trust-OS that will store the selected Trust Point into another blockchain.

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

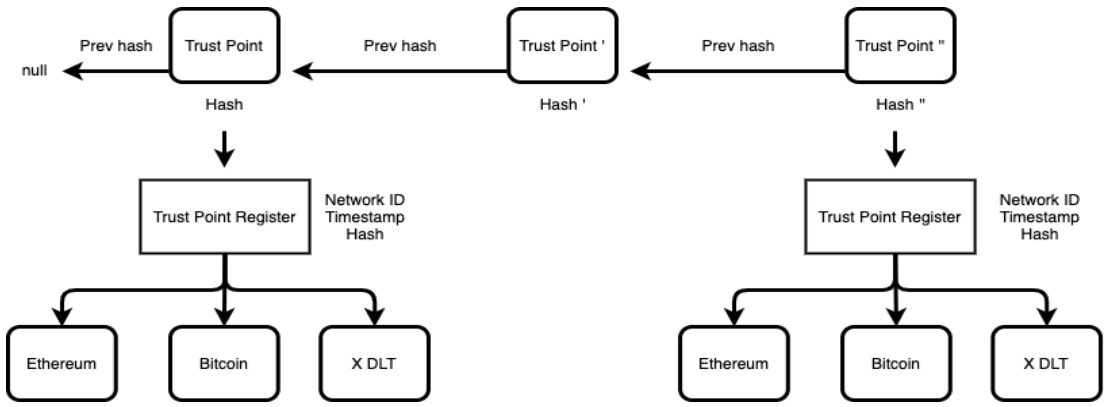

FIGURE 377: STORING RELEVANT INFORMATION.

The current functionality of Trust-OS only encompasses registering the Trust Points in the Ethereum network, but for the purpose of achieving interoperability among networks, the register function will be modified. So, when it is called, the networks into which the user wants to store the integrity proof, can be specified. In the previous picture, it can be seen how the necessary information like a timestamp and the hash of the Trust Point are stored in the different networks, specified by its network's identifiers.

Internally, Trust-OS will call the different APIs developed for each DLT by the different partners, so it eases the access to that network and exposes some key functionalities. How this is done, and which functionalities should these APIs expose are explained in further sections.

4.2.3 Federated DLTs

4.2.3.1 Bitcoin

Bitcoin ledger, as one of the federated iNGENIOUS cross-DLT solutions, will be used to store hashes and the timestamps of data provided by Trust-OS. Bitcoin has the longest history of immutability and stability, but it comes with the price of lower throughput on the base layer and more expensive transactions than on other DLTs. With the rest of the iNGENIOUS project, Bitcoin DLT will be integrated by means of a connector. To this end, an API will be defined to allow for interaction between Trust-OS and the Bitcoin network. The Bitcoin connector will require:

- Bitcoin full node;
- Testnet or Mainnet bitcoins in order to execute transactions on Bitcoin DLT;
- Internet connection.

The connector will support operations involving small packets of data (hashes, timestamps, etc.). It will also handle the security aspects of the wallets and funds involved in the process. The connector will allow:

- Storing a piece of data and providing transaction ID for future quick search of it directly on Bitcoin DLT;
- Finding specific data using the data itself;
- Finding specific data using ID;
- Providing information about confirmation status for the data;
- Configuration of fees for the transactions;

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

- Error handling;
- Financial reports.

There is also a research aspect that involves investigating the possibility to use the Lightning Network for receiving micropayments that will allow for funding the operation of the Bitcoin DLT. To achieve the above goals, the connector will be implemented, and a Bitcoin network full node will be established on a workstation with Linux operating system.

4.2.3.2 Hyperledger Fabric

Hyperledger Fabric will be one of the federated DLT solutions used in iNGENIOUS to exchange data associated to maritime port events, e.g. gate-in and gate-out events, with different supply chain stakeholders under high levels of trustiness, immutability and integrity. Within iNGENIOUS, supply chain stakeholders will use both Hyperledger Fabric and other DLT solutions considered in the project. To ensure an efficient interoperability between Fabric and other DLTs, the Hyperledger Fabric solution will be connected to Trust-OS, which will act as Cross-DLT layer as described in the Use Case on Supply Chain Ecosystem integration described in D2.1 [209]. The connection between Hyperledger Fabric and Trust-OS will be carried out at API level through the development of reading and writing APIs able to interact with Trust-OS for exchanging and retrieving information coming from other DLTs. Additionally, another instance of Hyperledger Fabric could be deployed in order to have another network to interact with, but this depends on the outcome of the development of the Cross-DTL layer.

4.2.3.3 Ethereum

With the aim of providing greater transparency of the transactions generated within the Trust-OS platform in a simple way, Telefonica decided to implement a dedicated connector to Ethereum. This connector is wrapped in a JavaScript driver that allows to communicate and interact with the Ethereum network. Main requirements for a service to be able to send transactions to Ethereum are:

- Have an Ethereum client node that is the gateway to the network;
- A deployed smart contract which enables the recording of information through the implementation of dedicated functions;
- A collection of libraries to interact with a local or remote Ethereum node using HTTP, IPC or WebSocket.

The functionalities for a smart contract that the Trust-OS Ethereum Service offers are set out below:

- *Compile*: This function generates contract Application Binary Interface (ABI) and Ethereum Virtual Machine (EVM) bytecode from the source code of the smart contract;
- *Deploy*: Allows to deploy a smart contract using the ABI and EVM bytecode generated in the compile process;
- *Get request*: Sends a reading request to the smart contract that does not require any gas fees;
- *Set request*: Sends a writing request to the smart contract that requires a transaction fee. A trust point is sent to be stored, consisting of a snapshot used to export the integrity of the transactions over an asset Trust-OS to public blockchains such as Ethereum, in order to provide an extra layer of transparency. Broadly speaking, the registration of a trust point consists of the storage of a timestamp and a hash that uniquely identifies all transactions that have been included in it.

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

4.2.3.4 IOTA

Within the scope of iNGENIOUS project, the interaction between the Cross-DLT layer and IOTA will be implemented by means of a middleware called InteroperaChain [204]. InteroperaChain has been developed by CNIT in order to track specific events during logistics operations at the Port of Livorno (e.g. empty/full containers) providing data immutability capabilities by interacting with IOTA network. It has been implemented by using the OpenAPI standard for the input/output interfaces definition that can be then easily integrated abstracting the complexity of the underlying DLTs. As a matter of fact, OpenAPI allows to automatically generate client-side code needed for the integration activities by supporting 33 different programming languages. Moreover, InteroperaChain guarantees a high service availability as well as a robust tolerance to faults, avoiding technological lock-in from the user perspective. Within the project, InteroperaChain will expose communication interfaces for reading and writing operations over the IOTA network (to be instantiated) so that the Cross-DLT layer can bidirectionally interact with it by exchanging relevant information in the context of maritime events. InteroperaChain will be able to interact with the underlying IOTA network by performing data-hash storage over ledgers (including events data when needed). In order to perform this kind of operations, InteroperaChain will interact directly with the IRI-Node (IOTA Reference Implementation) [205], an open-source software for the interaction with IOTA protocol. By means of this node, it will be possible to perform different operations over IOTA network, such as transactions validation as well as their storage in a ledger (in case of valid transactions). During the implementation phase of the project the possibility to use both a private and public IOTA networks for the validation of the use cases (e.g. Mainnet and Devnet) will be assessed, although a private tangle seems to be a more feasible solution since it allows to control only known nodes within the network. Moreover, the usage of IOTA Light Wallet will be also exploited as a user interface for the sending/receiving transactions operations on the network.

In order to achieve the interaction between Trust-OS and IOTA private network, the following setup is presented.

Writing Operation

As depicted in the Figure 38, the write operation starts from a general entity named "Writer" that can be a real user or a machine-generated write event. In our case, the writer is the Trust-OS layer.

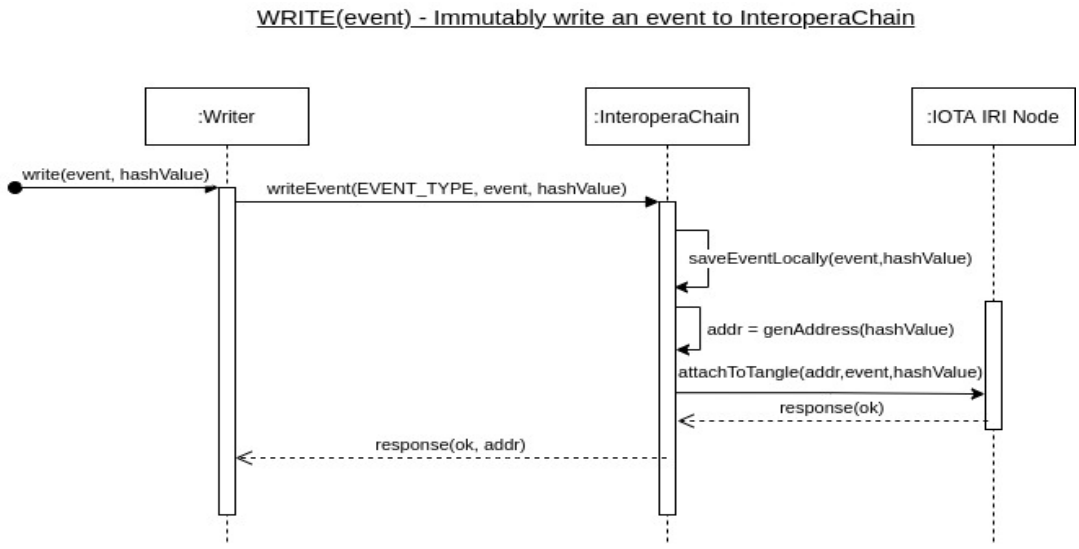

FIGURE 38: WRITING OPERATION SEQUENCE DIAGRAM.

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

To effectively perform the write operation, the InteroperaChain system needs the following inputs:

- *EVENT_TYPE*: string: (required) is the event type that corresponds to the *event* descriptor to be saved in the InteroperaChain system;
- *event: JSON*: is the event descriptor containing the event's metadata to be immutably saved in the InteroperaChain system;
- *hashValue*: string: (required): is the hashValue calculated on the event descriptor corresponding to the unique representation of the event to be immutably saved in the InteroperaChain system.

The event descriptor is not mandatory, but it can be provided to InteroperaChain in order to immutably save also the content of the event. Otherwise, if only hashValue is provided as input to the write operation, the InteroperaChain system will immutably save only hashValue. Upon receiving the above inputs, the InteroperaChain system first saves the information locally: this step is fundamental to be tolerant to any issue that may arise while saving the provided information to the underlying Distributed Ledger Technology (DLT) (i.e. IOTA), and is important also to implement a simple caching method to let the user exploit InteroperaChain as a proxy to the underlying DLT. Once the provided event is correctly saved locally, the InteroperaChain generates the IOTA address corresponding to the unique representation of the event (i.e. *hashValue*). To this end, the InteroperaChain takes advantage of the current IOTA address standard format [206][206] to encode *hashValue* in a valid *trite* IOTA address format. In this way, InteroperaChain leverages IOTA standard transactions to implement immutable storage of the provided events. Finally, if the transaction attachment (i.e. *attachToTangle* operation) succeeds, a positive response containing the calculated IOTA address (i.e. addr) is sent back to the Writer entity to let it save this information. Anyway, it is worth mentioning that saving the *addr* is not needed since InteroperaChain calculates it based on the provided *hashValue*, so this is the only needed information to access the provided event with a read operation.

Reading Operation

There exist two supported ways to read an event from the InteroperaChain system: the first one exploits the specific InteroperaChain supported read operations; the second one leverages the direct access to the underlying DLT (i.e. IOTA Tangle) via the available IOTA IRI Node, provided that the "Reader" entity knows the address of the needed event. The first option is depicted in Figure 39: reading operation sequence diagram to read events from InteroperaChain.. The Reader entity provides the *hashValue* of the needed event to the InteroperaChain system, together with the EVENT_TYPE that the event is expected to be part of. The InteroperaChain system recovers the local copy of the object that was saved during the write operation and returns it to the Reader entity.

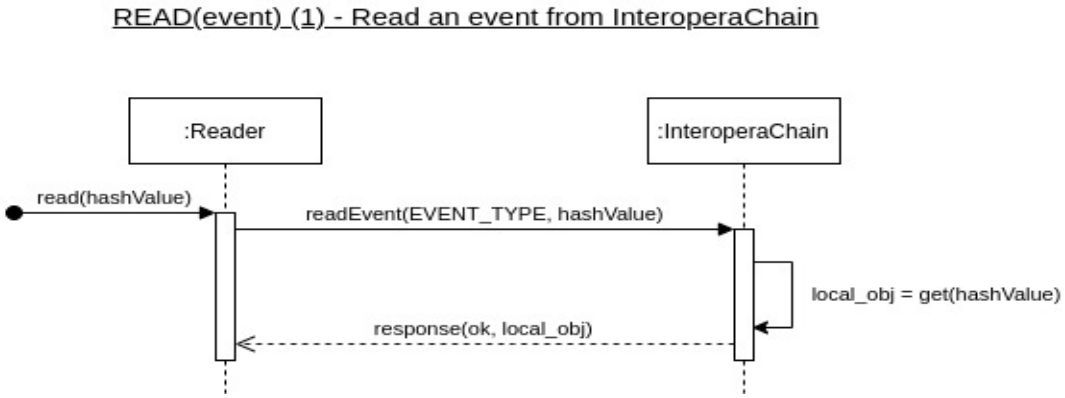

FIGURE 39: READING OPERATION SEQUENCE DIAGRAM TO READ EVENTS FROM INTEROPERACHAIN.

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

The second option is depicted in Figure 40. The Reader entity directly accesses the IOTA Tangle to access the content of the calculated IOTA Address, corresponding to the unique representation of the event that is intended to be read. It is worth mentioning that in this case the Reader entity needs to calculate the IOTA address on its own starting from the *hashValue* of the event. This confirms once again that hashValue is the only information that is needed to access each event.

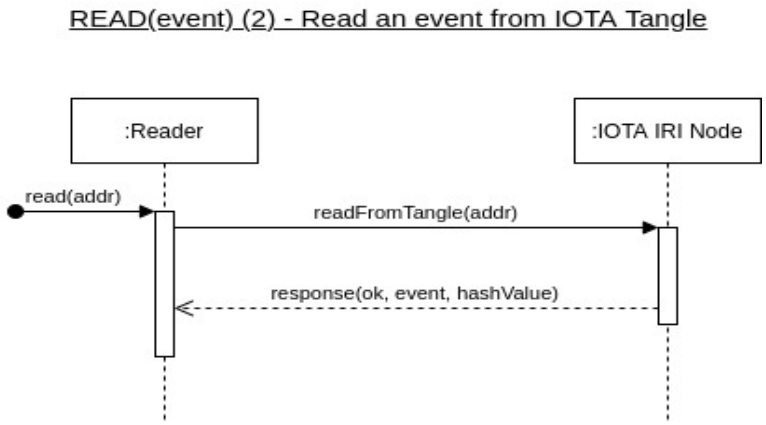

FIGURE 40: READING OPERATION SEQUENCE DIAGRAM TO READ EVENTS DIRECTLY FROM IOTA TANGLE.

4.2.4 APIs

Each DLT will have their own interface in charge of accessing the DLT network using the appropriate methods. These APIs must share a common interface as much as possible with common functionalities that exposes an access to the different DLTs. There are two main functionalities that the APIs must cover in order to be compatible with the explained above: storing and reading information to and from the different DLTs required in the project. The first one, storing capabilities, will be used to store the data coming from the different Trust Points created throughout the life of an asset. As the DTLs required have different capabilities, for example there are some with Smart Contracts capabilities and others not, the data to be stored is not fixed and may vary. At least, every DLT must store the hash or even the Merkle Tree root of the Trust Point. The second functionality is reading. This will be used to read the integrity verification data stored in the different DLTs so that the Trust module can verify the data integrity having multiple sources of truth. This functionality is simpler than the other and the only requirement is the capability of reading the desired data directly from the blockchain and not from an intermediary storing instance like a database.

4.2.5 General Considerations towards Security and Privacy in IoT Data Management

The aim of the project is to ensure a high degree of security, preventing unauthorized entities to access HW, SW, Network and Application layers by means of anonymization/de-identification techniques as well as by implementing a private blockchain network that exposes the access to data (e.g. smart contracts and data encryption).

From a hardware point of view, it will be considered how features of recent hardware security modules (HSMs) can provide means to protect secrets (encryption keys) while supporting all cryptographic operations within the boundaries of a secure hardware. A set of high-level interfaces based on common security standards and protocols will be provided to get access to these functions as a service in order to facilitate its access from other components in the layers that need to use it.

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

The project will reach the required security and privacy levels, aligned with the principles and practices of network neutrality. The project will investigate new attestation techniques to securely identify remote systems by adopting appropriate trusted hardware components, certificates, and cryptographic protocols.

The important part of security models involves protecting the privacy of the individuals whose data is processed. iNGENIOUS will investigate which data exchanged between partners in the supply chain need to be protected and implement various privacy preserving techniques. The comparison of the k-anonymity, l-diversity and t-closeness and pseudonymization techniques for enhanced privacy protection will be performed. The feasibility of various anonymization and pseudo anonymization techniques will also be investigated.

Interoperability, scalability and security are three of the most essential features of IoT ecosystems, which are not fully addressed in current architectures. iNGENIOUS platform will address these ever-increasing challenges by developing sophisticated security and inter-operability solutions to protect and guarantee the successful deployment of next generation IoT infrastructures on the universal supply chain. Therefore, next generation supply chain platforms should consider security when procuring both products and services throughout the entire operation in an IoT ecosystem from single devices to communications. IoT device manu-facturers, from product makers to semiconductor companies, should focus on building security in chip from the start. Security defences in hardware and software are often decoupled and not co-designed. Today, there are microkernel-based operating systems, which enable to run these system services as isolated user-space programs, and thus provide application isolation at the software level. iNGENIOUS will design a componentized microkernel-based operating system with a tile-based hardware architecture including communication control. This co-design approach is based on security by design and isolation by default. iNGENIOUS will reduce threat possibilities, i.e. will reduce the trusted computing base.

Data collected by IoT sensors is usually very fine grained due to constant sensing and high-dimensional due to a diversity of sensors being used. Such feature-rich data is of great use to learn more about the environment and extract valuable insights through analysis. At the same time, it poses a severe security and privacy risks if not handled with care. The key is to build in privacy-preserving secure design right into the entire data-analysis lifecycle starting from data collection, data storage, data transmission, data analysis to finally decommissioning the data when it has served its purpose. It is of paramount importance that each of these stages are designed in harmony that come together to provide a holistic solution. Starting from data collection caution must be exercised and only data with explicit need should be collected. Sweeping data collection makes ensuring security and privacy both expensive and complicated. The collected data should be carefully categorized as sensitive and non-sensitive and should be treated differently where possible. Combining different streams of data may have adverse effects on privacy which need to be considered meticulously. Pseudo-anonymization techniques should be utilized where possible to anonymize the data before storage. However, it is critical to note that pseudo-anonymization only acts as a deterrence against honest-but-curious passive adversaries and should not be seen as a concrete privacy protection mechanism against a motivated and resourceful adversary. As discussed in Section 2.5 anonymizing high-dimensional data is a very hard problem and perhaps impossible while preserving utility which is important for processing data. Hence, pseudo-anonymization of data should be supported with other mechanisms such as limiting access to sensitive data and tightly controlling parties who can access the sensitive data.

Nevertheless, access control and authorization are not only part of the solution but also part of the challenge, since the data collected during authorization and access control could violate the privacy of the involved humans, e.g. by allowing different collaborating services to link activities of a given user. Therefore, the feasibility of using anonymous attribute-based credentials for implementing privacy-respecting access control and authorization will be elaborated.

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

Data can be protected during transit using Transport Layer Security (TLS) which works at the network layer and is application agnostic. For more sensitive cases End-to-End protection (E2E) can be explored in conjunction with TLS which only provides link security. E2E secures the data at the end points and limits the access only to authorized entities.

Security and privacy risks during data processing could be mitigated by distributing the storage of data to avoid a single point of failure. Secure Multi-Party Computation (MPC) based protocols can be used on distributed data for analysis and processing. The confidentiality of data during processing could be further enhanced by using Homomorphic Encryption (HME) techniques which allows computation on encrypted data. Though these techniques do introduce additional computational overhead, judicious and careful integration of privacy friendly data processing mechanisms could help us achieve the best of both worlds. Lightweight cryptography can be greatly useful in the case of IoT devices as it focuses on energy efficiency and has a low computational demand. Trusted Execution Environment (TEE) provides a much efficient alternative to secure critical parts of code by executing it in a tightly controlled tamper resistant environment.

To provide concrete privacy guarantees it is important that once the data has served its purpose it should be securely deleted and all mechanisms to recover it be dismantled.

The final solution would depend on the specific use-case at hand. A one-size-fits-all approach may not be wise to pursue specially in an ecosystem populated by a wide variety of devices with varying constraints. The choice of tools would also be influenced by the end goal and system priorities.

4.3 Next Gen IoT Applications

4.3.1 AWAKE Application

In iNGENIOUS, Awake.AI will develop and deploy their software platform for providing human decision-makers coordinating the supply chain operations with situational awareness, i.e. shared visibility to what is happening in various parts of the supply chain. This will also incorporate machine learning -based predictive models for component processes such as vehicle schedules to enable more informed planning of resource allocations. Sharing of situational awareness is especially important with respect to logistics processes in ports, as these are often significant bottlenecks in the total supply chain, and ports today are typically not able to provide real-time situational awareness or reliable schedule predictions to other actors in the supply chain, such as cargo owners, shipping companies, or road haulers.

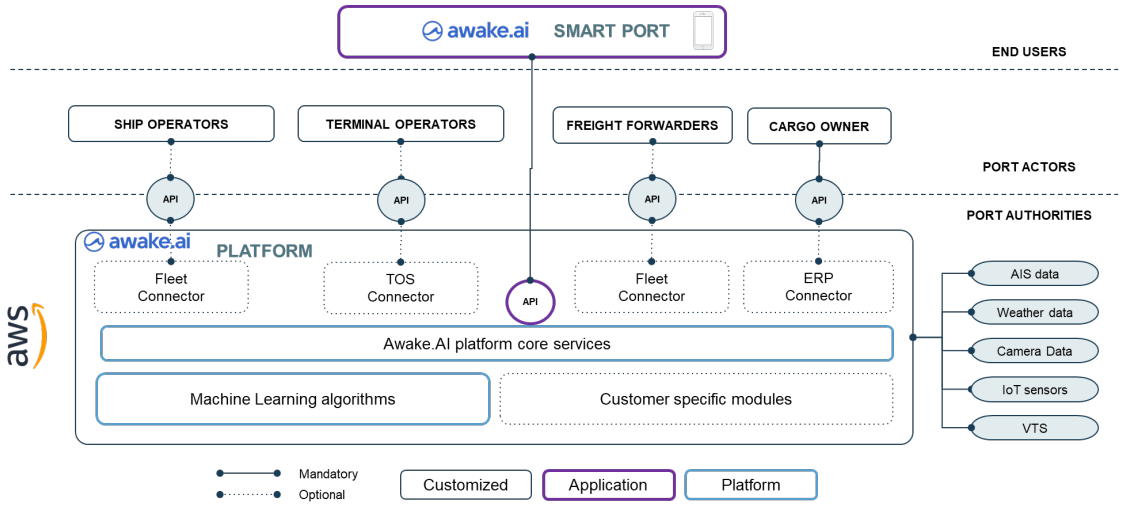

FIGURE 41: OVERVIEW OF THE AWAKE.AI SMART PORT PLATFORM.

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

Thanks to the availability of supply chain data extracted from IoT networks, data analytics and AI technologies will provide the ability to support optimization of process automation, asset management, and resource allocation at many stages of the supply chain. Predictive models, combining momentary IoT sensor data with machine-learned understanding of historical data, are significant for improving the efficiencies of multimodal logistics processes, as these require fitting together the future schedules of distinct and geographically distributed transport resources. In UC *Ship*, the focus will be on modelling, predicting, and communicating truck congestion levels and turnaround times in ports of Valencia and Livorno. For this, information from multiple global and port-specific data sources will be ingested to the Awake.AI platform for analysis. Large historical datasets will be used for model development, while corresponding live data feeds will be used for producing current situational awareness and inferences using predictive models. Model development for UC *Ship* focuses broadly on two main areas: predicting future vessel port call schedules as accurately as possible and modelling the correlation between truck traffic activity (including congestion and increased turnaround times) and vessel port calls. This requires combining data from multiple sources to obtain information on vessel locations, port call plans, cargo volumes to be exchanged, hinterland traffic volumes etc., and sufficient metadata to combine the information from various sources.

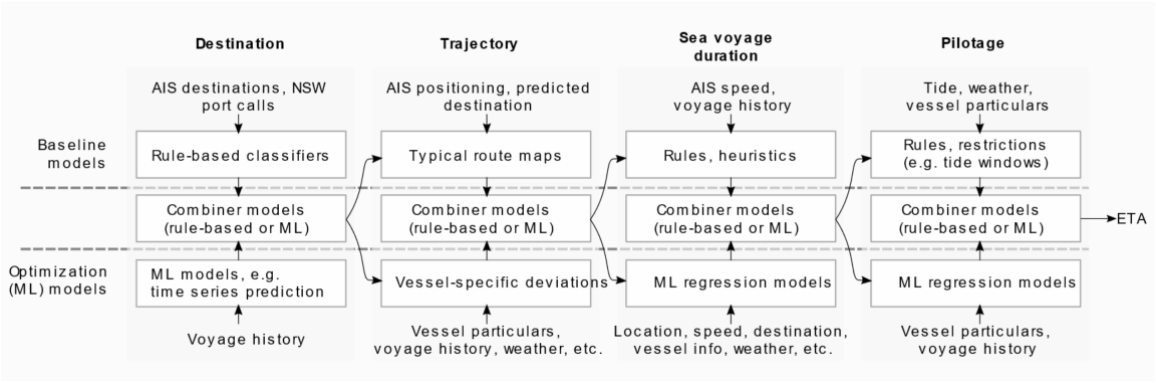

FIGURE 422: HIGH-LEVEL OVERVIEW OF THE MODEL ARCHITECTURE USED FOR VESSEL SCHEDULE PREDICTION.

This uses primarily globally available information sources such as Automatic Identification System (AIS) broadcasted by vessels over VHF frequencies, weather data, vessel particulars, etc. The main steps associated with predicting the course of the ongoing voyage of a vessel include classification of the current destination of the vessel, predicting the trajectory or route that the vessel will travel to the destination, predicting the effective speed for the remainder of the voyage, and predicting port-specific pilotage processes to the destination as needed.

Figure 43 shows an example of vessel destination, route, and schedule prediction as presented in the Awake.AI web application. This application also supports visualization of cargo and truck traffic statistics from port IoT and edge systems, as illustrated in Figure 44. Modelling and visualization of future truck traffic activity by combining models for vessel traffic and resulting cargo flow and truck traffic levels will be developed during the project in Use Case Ship.

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

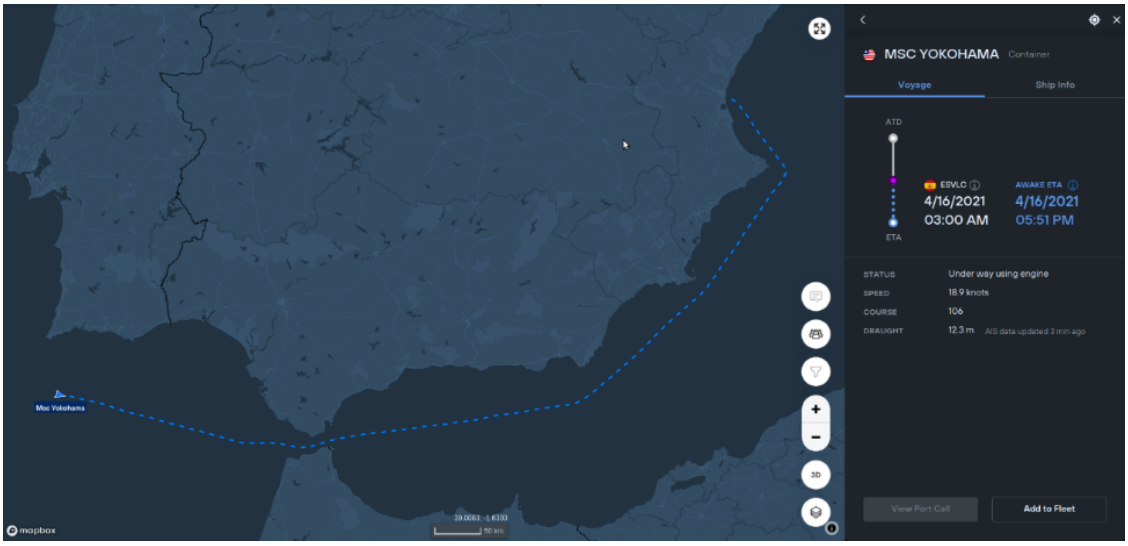

FIGURE 43: VESSEL DESTINATION, ROUTE, AND ESTIMATED TIME OF ARRIVAL (ETA) PREDICTIONS AS VISUALIZED IN THE AWAKE.AI WEB APPLICATION.

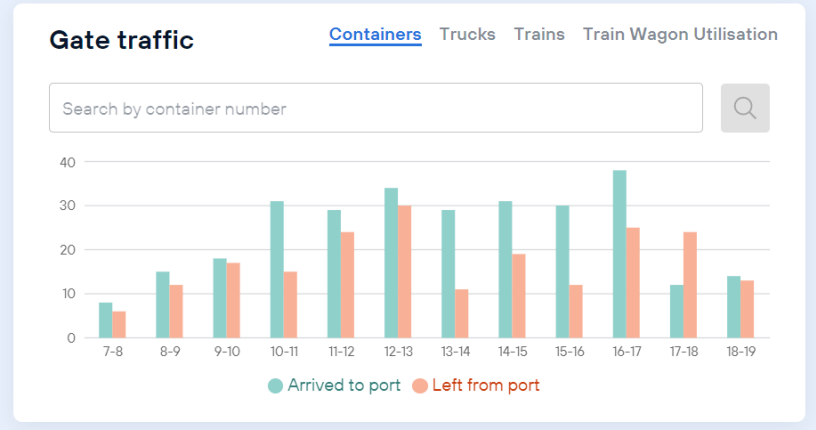

FIGURE 44: EXAMPLE VISUALIZATION IN AWAKE.AI WEB APPLICATION OF CARGO AND HINTERLAND TRAFFIC LEVELS BASED ON PORT IOT AND EDGE SYSTEMS.

4.3.2 Industrial & Tactile Application

In the context of the emerging industry 4.0 framework [207][207], the next generation factories are expected to be efficient, flexible, dynamic, self-organized, and able to produce customized products rather than massive products as in conventional factories. This requires flexible deployment and reconfiguration of production tools, in addition to a dynamic network to fulfill different requirements for connecting people, physical and virtual machines in real and virtual environment. Accordingly, the factory infrastructure, which includes different types of sensors, actuators, processing units, network resources, will be considered as a ubiquitous computing platform that is ready to execute customized end-user applications. These applications are composed of multiple tactile and non-tactile processes. The tactile processes comprise available tactile applications as discussed in Section 0, such as remote operations, autonomous driving, and automation. The other processes involve non-time critical missions such as data collection for monitoring, surveillance, predictive maintenance, and reporting. Aligned with the concept of tactile internet, which focuses on providing a seamless experience when interacting with remote objects as dealing with direct object, the tactile applications will provide similarity in programming remote distributed terminals as they are on a single computer (see Figure 45):

- The processing and storage capabilities act like multi core processers and memories, that are used in the execution of specific tasks;

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

- The I/O domain consists of devices, which can be simple inputs (sensors), outputs (actuators), or a stand-alone device with local inputs and outputs, such as tactile HIS;
- The network provides the connection bus system to interconnect the I/O with the processors and memories;
- The network management and orchestration play the role of operating system and expose different APIs that abstracts the hardware and provide the programming tools that are used by software developers like system calls as they program a single device, such as a smart phone;
- The end-user application runs on top of the network operating system.

As in any computer architecture, a compiler is required to translate the programming code to the hardware language and to ensures at least that some constraints are respected at the compile time. In the network scenario, the compiler is responsible of provisioning the required network and processing resources to fulfill different process requirements in term of latency, reliability, and data rate, and to consider dynamic conditions such as mobility and changing environment. In addition, run-time error handling should be thoroughly considered to avoid any malfunctioning of the operating system. For instance, to check if a device is already in use by other applications, the communications KPIs are fulfilled, and to properly release unused resources.

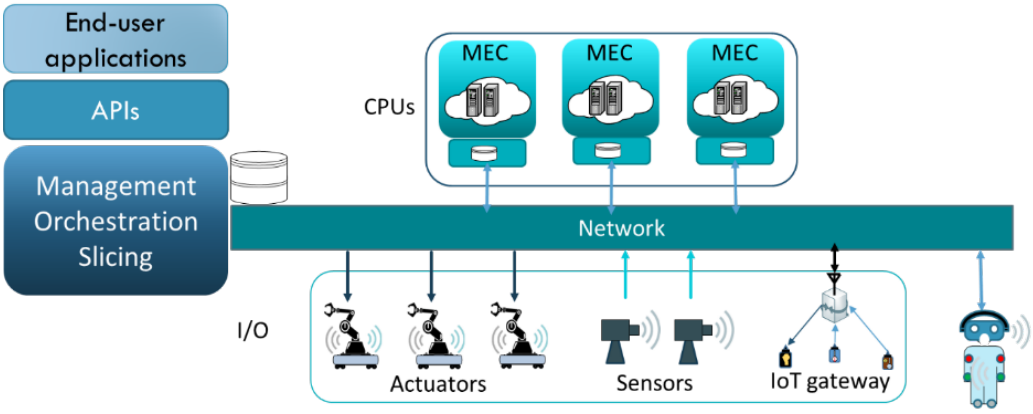

FIGURE 45: NETWORK COMPUTER ARCHITECTURE.

Connecting wireless terminals for computer systems is already implemented in everyday life, such as monitors, keyboards, mice, headsets, cameras, remote controls. However, when considering stringent timing requirements in terms of latency and synchronization between many devices, existing operating system and connectivity approaches are not sufficient.

4.4 Relation to UCs and Expected Innovation

This chapter presents the relation between the use cases *Ship*, *Port Entrance* and *DVL/Cross-DLT* and the expected innovations introduced by WP5. Most of the technologies presented in this document are, of course, relevant for the other use cases of iNGENIOUS project (*Factory*, *Transport* and *AGV*) from an architectural point of view, but we do not consider DVL, cross-DLT, and the other technologies introduced, in the demos of them.

4.4.1 Inter-modal Asset Tracking via IoT and Satellite - Ship UC

The UC on Inter-modal Asset Tracking via IoT and Satellite In addition targets to develop a Smart IoT Gateway which will ensure the connectivity for a vast number of heterogeneous IoT devices, by harmonizing different IoT technologies and application protocols and formatting the data to be transferred across the network.

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

In terms of the Table 1, the contributions of this UC are the following:

- *Cross-DLT Interoperability*: Not Relevant. The Smart IoT GW will not provide any service related to Distributed Ledger Technologies. In this UC, data collected from sensors are processed and transmitted to higher level components, who will store the data and make it available for end users' applications.
- *Cross-M2Ms Interoperability/DVL*: Innovation Demo. The Smart IoT GW architecture allows the implementation of any potential M2M endpoints. In practice, an oneM2M suitable interface will be demonstrated in this UC that will allow seamless communication with compatible systems such as Eclipse OM2M, OpenMTC or Mobius. On the other side, the Smart IoT GW will provide interoperability of multiple device protocols and the routing capability between them and the M2M interfaces.
- *Configurable Pseudonymization Function*: Not Relevant. The information gathered in this UC is tagged with IDs belonging to containers and sensors, which are needed for routing purposes. After that, it is sent to higher level systems via M2M interfaces. The data consolidation and information aggregation happen outside of the scope of this UC.
- *Development of AI-based Algorithms*: Not Relevant. Transformation, routing and optimization will be based on rules.
- *AWAKE Application*: Not Relevant. No direct interface with the Awake.AI platform.

4.4.2 Situational Understanding and Predictive Models in Smart Logistics Scenarios - Port Entrance UC

The UC on Situational Understanding and Predictive Models in Smart Logistics Scenarios targets the development of AI/ML-based predictive models to estimate and optimize truck turnaround times for enhancing the operational access and reduce the waiting times of vehicles at the port accesses, leading to corresponding savings on direct costs for carriers. This will enhance the situational understanding of events in maritime ports and terminals by collecting and aggregating data processing. By analyzing the available data sources, the use case will optimize and predict processes for reducing the time that trucks spend inside the port and terminal facilities, i.e., truck turnaround times (TTT).

The contribution of WP5 in this UC, reported in Table 1, consists of the following innovations:

- Cross-M2M Interoperability/DVL & Configurable Pseudonymization Function: Innovation Demo. Ingestion and pseudonymization of different data sources coming from M2M platforms through the DVL. (Section 2.5 and 4.2.1 Data Virtualization Layer);
- Development of AI-based algorithms for estimating and optimizing truck turnaround times (Section 3.3);
- Deployment of models and visualization of prediction and optimization outcomes at AWAKE application. (Section 4.3 Next Gen IoT Applications).

4.4.3 Supply Chain Ecosystem Integration – DVL/Cross-DLT UC

The UC on Supply Chain Ecosystem Integration addresses the interoperability aspects both at M2M platforms and DLTs level. The considered UC proposes an interoperable layer in order to overcome interoperability issues by means of Data Virtualization approach (based on Teiid open source solution) and cross-DLT layer (Trust-OS framework). For a better understanding of the main contribution of the WP5 to this UC in terms of innovation, summarized in Table 1, the following mapping is considered:

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

- *Cross-DLTs Interoperability*: Innovation Concept. This layer is expected to be used for the interaction with different DLTs according to identified maritime events and data coming from Data Virtualization Layer. By means of this layer, aggregated data will be first stored in a dedicated DBMS and then distributed (in a form of hash) across different DLTs. The main data that will be used for these events will come from the UC *Port Entrance*. Within the UC *DVL/Cross-DLT*, the Cross-DLT layer implementation will take place while its validation will be performed by means of UC *Port Entrance*.
- *Cross-M2M Interoperability*: Innovation Concept. This layer will be developed to be used as an intermediate layer between the Cross-DLT layer and underlying data sources such as M2M platforms. The layer will be implemented by means of Data Virtualization approach within the UC *DVL/Cross-DLT,* but its real validation will take place according to UC *Ship* (interaction with OM2M platforms) and UC *Port Entrance* (interaction with PISystem M2M platform, MANO platform and Awake Platform).
- *Configurable Pseudonymization Function*: Innovation Concept. According to data coming from M2M platforms and other data sources, a pseudonymization function will be developed at Data Virtualization Layer level within the scope of UC *DVL/Cross-DLT*. This will allow to automatically detect any personal data such as truck plate number and/or its localization. The real validation of this function will take place in UC *Port Entrance* according to data that are expected to be collected.
- *Development of AI-based Algorithms*: Not Relevant. AI-based algorithms are not part of this use case.
- *AWAKE Application*: Not Relevant. Although the integration between Data Virtualization Layer and AWAKE Platform is expected to be performed according to UC *Port Entrance* workflow (for historical data analysis), this activity is not considered as part of the UC *DVL/Cross-DLT* itself. As a matter of fact, this integration will be performed in the scope of UC *Port Entrance*.

Functionality	Factory UC	Transport UC	AGV UC	Ship (Container) UC	Port Entrance UC	DVL/Cross-DLT UC
Cross-DLTs interoperability	-	-	-	-	-	Innovation Concept
cross-M2Ms interoperability / DVL	-	-	-	Innovation Demo	Innovation Demo	Innovation Concept
Configurable Pseudonymization function	-	-	-	-	Innovation Demo	Innovation Concept
Development of AI-based algorithms	-	-	-	-	Innovation Demo	-
AWAKE Application	-	-	-	-	Innovation Demo	-

TABLE 1: NEXT-GEN IOT APPLICATIONS MAPPED TO UCS

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

5 Conclusion

This document has described the state of the art and innovations introduced by iNGENIOUS with respect to an Interoperable Layer and smart IoT applications. The deliverable has been structured in three main parts, namely 'IoT data management technologies', 'supply chain IoT applications' and 'INGENIOUS proposed solutions'.

With respect to the IoT data management technologies, the M2M Interoperability, Data Virtualization and DLT Interoperability's main available solutions have been described. Furthermore, a comprehensive description of security and privacy aspects has been provided.

Regarding the supply chain IoT applications, the available applications and the tactile applications of smart factories and warehouses, as well as smart transportation and ports have been addressed. This part also provided examples of AI algorithms supporting commercial supply chain applications.

Finally, a first description of the Interoperable Layer architecture implementation is provided, highlighting the innovations addressed in the DVL and cross-DLT layers compared to the functionalities requested for the UCs implementation. Also, in this final part supply chain applications to be developed in the project have been presented.

The development iterations will allow to further elaborate the innovation aspects of the proposed solutions, starting from the initial and valuable considerations addressed in this deliverable.

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

6 References

[1] http://www.mit.jyu.fi/ope/kurssit/TIES462/Materiaalit/IEEE_SoftwareEngGlossary.pdf
[2] https://www.iso.org/obp/ui/#iso:std:iso-iec:2382:-1:ed-3:v1:en
[3] https://cordis.europa.eu/project/id/688038
[4] https://cordis.europa.eu/article/id/300721-cuttingedge-middleware-for-interoperable-iot-services-new-revenue-streams-and-lower-market-en
[5] https://cordis.europa.eu/project/id/687283/results
[6] https://cordis.europa.eu/project/id/690797
[7] https://www.onem2m.org/
[8] https://www.onem2m.org/getting-started/onem2m-overview/introduction/functional-architecture
[9] https://autopilot-project.eu/
[10] https://www.corealis.eu/
[11] Eclipse Foundation, "Eclipse OM2M," https://www.eclipse.org/om2m/, Accessed: March 2021.
[12] ETSI, "Technical Committee (TC) Smart Machine-to-Machine Communications (SmartM2M)," https://www.etsi.org/committee/smartm2m, Accessed: March 2021.
[13] OpenMTC, "OpenMTC in a nutshell," https://www.openmtc.org/, Accessed: March 2021.
[14] https://www.dama.org/cpages/home
[15] https://developers.redhat.com/products/datavirt/overview
[16] https://teiid.io/
[17] Bitcoin Core source code, https://github.com/bitcoin/bitcoin
[18] "Bitcoin: Peer-to-peer Electronic Cash System", Satoshi Nakamoto, 2009 https://en.bitcoin.it/wiki/Controlled_supply, 2021
[19] "The Bitcoin Lightning Network: Scalable Off-Chain Instant Payments", Joseph Poon and Thaddeus Dryja, 2016
[20] "Mastering Lightning Network", Andreas M. Antonopoulos, Olaoluwa Osuntokun, Rene Pickhardt, working draft, 2021
[21] "The Bitcoin Lightning Network", http://lightning.network/
[22] https://en.bitcoin.it/wiki/Contract
[23] https://en.bitcoin.it/wiki/Script
[24] https://www.iota.org/
[25] https://blog.iota.org/the-state-of-qubic-63ffb097da3f/
[26] https://blog.iota.org/tokenization-on-the-tangle-iota-digital-assets-framework/
[27] https://cryptoconsortium.github.io/CCSS/
[28] https://www.synopsys.com/glossary/what-is-security-risk-assessment.html
[29] https://www.iso.org/standard/72140.html
[30] https://www.iso.org/obp/ui/#iso:std:iso-iec:27000:ed-5:v1:en
[31] https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-39.pdf
[32] https://github.com/OWASP/wstg/tree/master/document/4-Web_Application_Security_Testing/00-Introduction_and_Objectives
[33] https://cve.mitre.org/
[34] https://www.first.org/cvss/
[35] https://owasp.org/www-project-top-ten/
[36] https://csrc.nist.rip/publications/nistpubs/800-12/800-12-html/chapter3.html
[37] wstg/document/2-Introduction at master · OWASP/wstg · GitHub
[38] https://owasp.org/
[39] https://github.com/OWASP/wstg
[40] https://github.com/opentimestamps/opentimestamps-server/commit/6e2519d0bb8af2c6ec006e7849d626b20af99a77

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

[41] https://petertodd.org/2016/opentimestamps-announcement
[42] https://www.sciencedirect.com/science/article/abs/pii/S0306457320309602
[43] https://github.com/GabrielScheibler/iota-timestamping
[44] https://www.schneier.com/essays/archives/2000/04/the_process_of_secur.html
[45] International Telecommunication Union (ITU), "Data networks and open system communications. Open system interconnection - model and notation."
[46] Bluetooth Special Interest Group, "Bluetooth specifications," https://www.bluetooth.com/specifications/, Accessed: March 2021.
[47] IEEE 802.11 WLAN Working Group, "IEEE 802.11 WIRELESS LOCAL AREA NETWORKS," https://www.ieee802.org/11/, Accessed: March 2021.
[48] IEEE 802.3 Ethernet Working Group, "IEEE 802.3 ETHERNET," https://www.ieee802.org/3/, Accessed: March 2021.
[49] P. Dhaker, "Introduction to SPI Interface," https://www.analog.com/en/analog-dialogue/articles/introduction-to-spi-interface.html#, Accessed: March 2021.
[50] NXP Semiconductors, "UM10204 I2C bus specification and user manual," April 2014, https://www.nxp.com/docs/en/user-guide/UM10204.pdf, Accessed: March 2021/
[51] 3GPP, "Standardization of NB-IoT completed," https://www.3gpp.org/news-events/1785-nb_iot_complete, Accessed: March 2021.
[52] 3GPP, "Standards for the IoT," https://www.3gpp.org/news-events/1805-iot_r14, Accessed: March 2021.
[53] LoRa Alliance Technical Committee, "LoRaWAN 1.1 Specification," October 2017, https://lora-alliance.org/resource_hub/lorawan-specification-v1-1/, Accessed: March 2021.
[54] "IEEE Standard for Information technology--Telecommunications and information exchange between systems - Local and metropolitan area networks--Specific requirements - Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications Amendment 2: Sub 1 GHz License Exempt Operation," in IEEE Std 802.11ah-2016 (Amendment to IEEE Std 802.11-2016, as amended by IEEE Std 802.11ai-2016) , vol., no., pp.1-594, May 2017, doi: 10.1109/IEEESTD.2017.7920364.
[55] SigFox, "Sigfox Certification Handbook Rev. V3.2.3.," February 2021, https://support.sigfox.com/docs/sigfox-certification-handbook, Accessed: March 2021.
[56] ZigBee Alliance, "ZigBee Specification," August 2015, https://zigbeealliance.org/wp-content/uploads/2019/11/docs-05-3474-21-0csg-zigbee-specification.pdf, Accessed: March 2021.
[57] CoRE Working Group, IETF, "Constrained Application Protocol (CoAP) RFC7252," https://tools.ietf.org/html/rfc7252, Accessed: March 2021.
[58] Oasis Open, "MQTT Version 5.0 Oasis Standard," March 2019, https://docs.oasis-open.org/mqtt/mqtt/v5.0/os/mqtt-v5.0-os.html, Accessed: March 2021.
[59] oneM2M organization, "Standards for M2M and the Internet of Things," https://www.onem2m.org/technical/partner-transpositions, Accessed: March 2021.
[60] "What Is NoSQL," https://www.mongodb.com/nosql-explained, Accessed: March 2021.
[61] Influxdata, "Time series database (TSDB) explained," https://www.influxdata.com/time-series-database/, Accessed: March 2021.
[62] https://www.traxens.com/
[63] https://www.cargosmart.com/en/solutions/mobile.htm
[64] T. Dalenius. “Towards a methodology for statistical disclosure control”. Statistik Tidskrift, 15(429-444), 1977.
[65] R. Shokri, M. Stronati, C. Song, and V. Shmatikov. Membership inference attacks against machine learning models. In IEEE S&P, 2017.
[66] C. Dwork and M. Naor. On the difficulties of disclosure prevention in statistical databases or the case for differential privacy. Journal of Privacy and Confidentiality, 2(1), 2010.
[67] E. De Cristofaro. An Overview of Privacy in Machine Learning, 2020.

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

[68] C. Song, V. Shmatikov. Auditing Data Provenance in Text-Generation Models. In KDD, 2019.
[69] A. Salem, Y. Zhang, M. Humbert, P. Berrang, M. Fritz, and M. Backes. ML-Leaks: Model and data independent membership inference attacks and defenses on machine learning models. In NDSS, 2019.
[70] S. Truex, L. Liu, M. E. Gursoy, L. Yu, and W. Wei. Towards demystifying membership inference attacks. IEEE Transactions on Services Computing. IEEE. 2019.
[71] S. Yeom, I. Giacomelli, M. Fredrikson, and S. Jha. Privacy Risk in Machine Learning: Analyzing the Connection to Overfitting. In IEEE CSF, 2018.
[72] J. Hayes, L. Melis, G. Danezis, and E. De Cristofaro. Logan: Membership inference attacks against generative models. Proceedings on Privacy Enhancing Technologies, 2019(1).
[73] L. Melis, C. Song, E. De Cristofaro, and V. Shmatikov. Exploiting unintended feature leakage in collaborative learning. In IEEE Symposium on Security and Privacy, 2019
[74] T. Orekondy, B. Schiele, and M. Fritz. Knockoff nets: Stealing functionality of black-box models. In IEEE CVPR, 2019.
[75] F. Tramer, F. Zhang, A. Juels, M. K. Reiter, and T. Ristenpart. Stealing machine learning models via prediction APIs. In USENIX Security, 2016.
[76] B. Wang and N. Z. Gong. Stealing hyperparameters in machine learning. In IEEE Symposium on Security and Privacy, 2018.
[77] S. J. Oh, M. Augustin, M. Fritz, and B. Schiele. Towards reverse-engineering black-box neural networks. In ICLR, 2018.
[78] N. Papernot, P. McDaniel, I. Goodfellow, S. Jha, Z. B. Celik, and A. Swami. Practical Black-Box Attacks against Machine Learning. In AsiaCCS, 2017.
[79] M. Fredrikson, S. Jha, and T. Ristenpart. Model inversion attacks that exploit confidence information and basic countermeasures. In ACM CCS, 2015.
[80] M. Fredrikson, E. Lantz, S. Jha, S. Lin, D. Page, T. Ristenpart. Privacy in pharmacogenetics: An end-to-end case study of personalized warfarin dosing. In USENIX Security, 2014.
[81] B. Hitaj, G. Ateniese, and F. Pérez-Cruz. Deep models under the GAN: information leakage from collaborative deep learning. In ACM CCS, 2017.
[82] Y. Aono, T. Hayashi, L. Wang, S. Moriai, et al. Privacy-preserving deep learning: Revisited and Enhanced. In ATIS, 2017.
[83] R. Shokri and V. Shmatikov. Privacy-preserving deep learning. In ACM CCS, 2015.
[84] C. Song, T. Ristenpart, and V. Shmatikov. Machine learning models that remember too much. In ACM CCS, 2017.
[85] N. Carlini, C. Liu, Ú. Erlingsson, J. Kos, D. Song. The Secret Sharer: Evaluating and Testing Unintended Memorization in Neural Networks. In USENIX Security, 2019.
[86] N. Carlini, F. Tramèr, E. Wallace, M. Jagielski, A. Herbert-Voss, K. Lee, A. Roberts, T. Brown, D. Song, Ú. Erlingsson, A. Oprea, C. Raffel. Extracting Training Data from Large Language Models. arXiv:2012.07805, 2020.
[87] K. Ganju, Q.Wang, W. Yang, C. A. Gunter, N. Borisov. Property Inference Attacks on Fully Connected Neural Networks using Permutation Invariant Representations. In CCS, 2018.
[88] L. Sweeney. K-anonymity: A model for protecting privacy. In International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems, 2002.
[89] A. Machanavajjhala, D. Kifer, J. Gehrke, and M. Venkitasubramaniam. L-diversity: Privacy beyond k-anonymity. In TKDD, 2007.
[90] N. Li, T. Li, and S. Venkatasubramanian. t-closeness: Privacy beyond k-anonymity and l-diversity. In ICDE 2007.
[91] B. I. P. Rubinstein, B. Nelson, L. Huang, A. D. Joseph, S. Lau, S. Rao, N. Taft, J. D. Tygar. ANTIDOTE: understanding and defending against poisoning of anomaly detectors. IMC, 2009.
[92] B. Nelson, M. Barreno, F. J. Chi, A. D. Joseph, B. I. P. Rubinstein, U. Saini, C. Sutton, J. D. Tygar, K. Xia. Exploiting Machine Learning to Subvert Your Spam Filter. LEET, 2008.

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

[93] S. Mei, X. Zhu. The Security of Latent Dirichlet Allocation. AISTATS, 2015.
[94] H. Xiao, B. Biggio, G. Brown, G. Fumera, C. Eckert, F. Roli. Is Feature Selection Secure against Training Data Poisoning? ICML, 2015.
[95] I. Shumailov, Y. Zhao, D. Bates, N. Papernot, R. Mullins, R. Anderson. Sponge Examples: Energy-Latency Attacks on Neural Networks. EuroS&P, 2021.
[96] G. F. Elsayed, I. J. Goodfellow, J. Sohl-Dickstein. Adversarial Reprogramming of Neural Networks. ICLR, 2019.
[97] B. McMahan, E. Moore, D. Ramage, S. Hampson, B. Agüera y Arcas. Communication-Efficient Learning of Deep Networks from Decentralized Data. AISTATS, 2017.
[98] A. Bhagoji, S. Chakraborty, P. Mittal, S. Calo. Analyzing Federated Learning through an Adversarial Lens. ICML, 2019.
[99] E. Bagdasaryan, A. Veit, Y. Hua, D. Estrin, V. Shmatikov. How To Backdoor Federated Learning. AISTATS, 2020.
[100] K. Grosse, P. Manoharan, N. Papernot, M. Backes, P. D. McDaniel. On the (Statistical) Detection of Adversarial Examples. arXiv:1702.06280, 2017.
[101] Z. Gong, W. Wang, W. Ku. Adversarial and Clean Data Are Not Twins, 2017.
[102] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, A. Vladu. Towards Deep Learning Models Resistant to Adversarial Attacks. ICLR, 2018.
[103] U. Shaham, Y. Yamada, S. Negahban. Understanding adversarial training: Increasing local stability of supervised models through robust optimization. In Neurocomputing, 2018.
[104] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus. Intriguing properties of neural networks. In ICLR, 2014.
[105] N. Papernot, P. D. McDaniel, X. Wu, S. Jha, and A. Swami. Distillation as a defense to adversarial perturbations against deep neural networks. In IEEE 2016.
[106] N. Carlini and D. A. Wagner. Towards evaluating the robustness of neural networks. In IEEE S&P, 2017.
[107] R. Feinman, R. R. Curtin, S. Shintre, and A. B. Gardner. Detecting Adversarial Samples from Artifacts. arXiv:1703.00410, 2017.
[108] D. Meng and H. Chen. Magnet: a two-pronged defense against adversarial examples. In CCS, 2017.
[109] Y. Zhou, M. Kantarcioglu, and B. Xi. Breaking transferability of adversarial samples with randomness. arXiv:1805.04613, 2018.
[110] C. Xie, J. Wang, Z. Zhang, Z. Ren, and A. Yuille. Mitigating adversarial effects through randomization. In ICLR, 2018.
[111] X. Cao and N. Gong. Mitigating evasion attacks to deep neural networks via region-based classification. In ACSAC, 2017.
[112] K. Sharad, G. Marson, H. Truong, and G. Karame. On the Security of Randomized Defenses Against Adversarial Samples. In ASIACCS, 2020.
[113] A. Bhagoji, D. Cullina, C. Sitawarin, and P. Mittal. Enhancing Robustness of Machine Learning Systems via Data Transformations. In CISS, 2018.
[114] W. Xu, D. Evans, Y. Qi. Feature Squeezing: Detecting Adversarial Examples in Deep Neural Networks. In NDSS, 2018.
[115] K. Grosse, N. Papernot, P. Manoharan, M. Backes, and P. McDaniel. Adversarial perturbations against deep neural networks for malware classification. arXiv:1606.04435, 2016.
[116] F. Zhang, P. Chan, B. Biggio, D. Yeung, and F. Roli. Adversarial feature selection against evasion attacks. In IEEE Trans. Cybernetics, 2016.
[117] K. Dvijotham, S. Gowal, R. Stanforth, R. Arandjelovic, B. O'Donoghue, J. Uesato, and P. Kohli. Training verified learners with learned verifiers. arXiv:1805.10265, 2018.
[118] M. Hein and M. Andriushchenko. Formal Guarantees on the Robustness of a Classifier against Adversarial Manipulation. In NIPS, 2017.
[119] A. Raghunathan, J. Steinhardt, and P. Liang. Certified Defenses against Adversarial Examples. In ICLR, 2018.

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

[120] A. Raghunathan, J. Steinhardt, and P. Liang. Semidefinite relaxations for certifying robustness to adversarial examples. In NeurIPS, 2018.
[121] E. Wong and J. Kolter. Provable Defenses against Adversarial Examples via the Convex Outer Adversarial Polytope. In ICML, 2018.
[122] N. Carlini, A. Athalye, N. Papernot, W. Brendel, J. Rauber, D. Tsipras, I. Goodfellow, A. Madry, and A. Kurakin. On Evaluating Adversarial Robustness. arXiv:1902.06705, 2019.
[123] A. Ghiasi, A. Shafahi, and T. Goldstein. Breaking Certified Defenses: Semantic Adversarial Examples with Spoofed Robustness Certificates. In ICLR, 2020.
[124] N. Papernot, P. McDaniel, S. Jha, M. Fredrikson, Z. Celik, A. Swami. The Limitations of Deep Learning in Adversarial Settings. In IEEE S&P, 2016.
[125] I. Goodfellow, J. Shlens, and C. Szegedy. Explaining and harnessing adversarial examples. In ICLR, 2015.
[126] A. Kurakin, I. Goodfellow, and S. Bengio. Adversarial examples in the physical world. In ICLR, 2017.
[127] S. Moosavi-Dezfooli, A. Fawzi, and P. Frossard. Deepfool: A simple and accurate method to fool deep neural networks. In CVPR, 2016.
[128] T. Brown, D. Mané, A. Roy, M. Abadi, J. Gilmer. Adversarial Patch., 2017.
[129] M. Sharif, S. Bhagavatula, L. Bauer, M. Reiter. "A General Framework for Adversarial Examples with Objectives". In ACM Transactions on Privacy and Security, 2019.
[130] C. Dwork, F. McSherry, K. Nissim, A. Smith. Calibrating Noise to Sensitivity in Private Data Analysis. In TCC, 2006.
[131] A. Narayanan and V. Shmatikov. Robust de-anonymization of large sparse datasets. In IEEE S&P, 2008.
[132] A. Narayanan, E. Shi, and B. Rubinstein. Link prediction by de-anonymization: How we won the kaggle social network challenge. In IJCNN, 2011.
[133] A. Narayanan and V. Shmatikov. De-anonymizing social networks. In IEEE S&P,2009.
[134] G. Danezis and C. Troncoso. Vida: How to use bayesian inference to de-anonymize persistent communications. In PETS. 2009.
[135] G. Danezis and C. Troncoso. You cannot hide for long: de-anonymization of real-world dynamic behaviour. In WPES, 2013.
[136] A. Biryukov, I. Pustogarov, and R. Weinmann. Trawling for tor hidden services: Detection, measurement, deanonymization. In IEEE S&P, 2013.
[137] B. Malin and L. Sweeney. How (not) to protect genomic data privacy in a distributed network: using trail re-identification to evaluate and design anonymity protection systems. In Journal of Biomedical Informatics, 2004.
[138] J. Calandrino, A. Kilzer, A. Narayanan, E. Felten, and V. Shmatikov. "you might also like: " privacy risks of collaborative filtering. In IEEE S&P, 2011.
[139] C. Aggarwal. On k-anonymity and the curse of dimensionality. In VLDB, 2005.
[140] K. Sharad. True Friends Let You Down: Benchmarking Social Graph Anonymization Schemes. In AISec, 2016.
[141] K. Sharad. Change of Guard: The Next Generation of Social Graph De-anonymization Attacks. In AISec, 2016.
[142] K. Sharad and G. Danezis. An Automated Social Graph De-anonymization Technique. In WPES, 2014.
[143] B. Oprisanu, G. Ganev, E. De Cristofaro. Measuring Utility and Privacy of Synthetic Genomic Data. arXiv:2102.03314, 2021.
[144] https://protect2.fireeye.com/v1/url?k=b0b3392a-ef28002f-b0b379b1-86959e472243-2c3a0e453fec592b&q=1&e=750e1802-9cd6-45ee-8cb1-5043d4e1521d&u=http%3A%2F%2Fabc4trust.eu%2Fdownload%2FDeliverable_D2.2.pdf
[145] J. W. Bos, K. Lauter, and M. Naehrig. Private Predictive Analysis on Encrypted Medical Data. Journal of Biomedical Informatics, 50:234–243, 2014.

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

[146] T. Graepel, K. Lauter, and M. Naehrig. Ml confidential: Machine learning on encrypted data. In ICISC, 2013.
[147] R. Bost, R. A. Popa, S. Tu, and S. Goldwasser. Machine learning classification over encrypted data. Technical report, Cryptology ePrint Archive Report 2014/331, 2014.
[148] Y. Lindell and B. Pinkas. Privacy preserving data mining. In CRYPTO, 2000.
[149] W. Du, Y. S. Han, and S. Chen. Privacy-preserving multivariate statistical analysis: Linear regression and classification. In IEEE ICDM, 2004.
[150] K. Bonawitz, V. Ivanov, B. Kreuter, A. Marcedone, H. B. McMahan, S. Patel, D. Ramage, A. Segal, and K. Seth. Practical secure aggregation for privacy-preserving machine learning. In ACM CCS, 2017.
[151] N. Dowlin, R. Gilad-Bachrach, K. Laine, K. Lauter, M. Naehrig, and J. Wernsing. Cryptonets: Applying neural networks to encrypted data with high throughput and accuracy. In ICML, 2016.
[152] J. Liu, M. Juuti, Y. Lu, and N. Asokan. Oblivious neural network predictions via minionn transformations. In ACM CCS, 2017.
[153] P. Mohassel and Y. Zhang. Secureml: A system for scalable privacy-preserving machine learning. In IEEE S&P, 2017.
[154] S. Sav, A. Pyrgelis, J. R. Troncoso-Pastoriza, D. Froelicher, J. Bossuat, J. Sa Sousa, J. Hubaux. POSEIDON: Privacy-Preserving Federated Neural Network Learning. In NDSS, 2021.
[155] M. Abadi, A. Chu, I. Goodfellow, H. B. McMahan, I. Mironov, K. Talwar, and L. Zhang. “Deep learning with differential privacy”. In ACM CCS, 2016.
[156] M. Abadi, U. Erlingsson, I. Goodfellow, H. B. McMahan, I. Mironov, N. Papernot, K. Talwar, L. Zhang. “On the protection of private information in machine learning systems: Two recent approaches”. In IEEE CSF, 2017.
[157] N. Papernot, M. Abadi, U. Erlingsson, I. Goodfellow, and K. Talwar. Semi-supervised knowledge transfer for deep learning from private training data. In ICLR, 2017.
[158] N. Papernot, S. Song, I. Mironov, A. Raghunathan, K. Talwar, U. Erlingsson. “Scalable private learning with pate”. In ICLR, 2018.
[159] H. B. McMahan, D. Ramage, K. Talwar, L. Zhang. “Learning differentially private language models without losing accuracy”. In ICLR, 2018.
[160] R. C. Geyer, T. Klein, and M. Nabi. “Differentially private federated learning: A client level perspectiveW. 2017.
[161] O. Ohrimenko, F. Schuster, C. Fournet, A. Mehta, S. Nowozin, K. Vaswani, and M. Costa, “Oblivious multiparty machine learning on trusted processors”, USENIX Security, 2016.
[162] T. Hunt, C. Song, R. Shokri, V. Shmatikov, and E. Witchel. “Chiron: Privacy-preserving machine learning as a service”, 2018.
[163] N. Hynes, R. Cheng, D. Song. “Efficient deep learning on multi-source private data”, 2018.
[164] F. Tramer and D. Boneh. “SLALOM: Fast, verifiable and private execution of neural networks in trusted hardware”. In ICLR, 2019.
[165] L. Hanzlik, Y. Zhang, K. Grosse, A. Salem, M. Augustin, M. Backes, M. Fritz. “Mlcapsule: Guarded offline deployment of machine learning as a service”, 2018.
[166] L. Song, R. Shokri, P. Mittal. Privacy Risks of Securing Machine Learning Models against Adversarial Examples. In CCS, 2019.
[167] E. Bagdasaryan, O. Poursaeed, V. Shmatikov. “Differential Privacy Has Disparate Impact on Model Accuracy”. In NeurIPS, 2019.
[168] A. Pfitzmann, M. Hansen. “A terminology for talking about privacy by data minimization: Anonymity, Unlinkability, Undetectability, Unobservability, Pseudonymity, and Identity Management”. 2010.
[169] M. Hansen. “Top 10 mistakes in system design from a privacy perspective and privacy protection goals”. In Privacy and Identity Management for Life, 2012.

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

[170] D. Chaum. “Untraceable Electronic Mail, Return Addresses and Digital Pseudonyms”. In Secure Electronic Voting, 2003.
[171] G. Danezis, R. Dingledine, N. Mathewson, “Mixminion: Design of a Type III Anonymous Remailer Protocol”. In IEEE S&P, 2013.
[172] G. Danezis. Mix-Networks with Restricted Routes. In PETS, 2003.
[173] R. Soltani, D. Goeckel, D. Towsley, A. Houmansadr, “Towards provably invisible network flow fingerprints”, In ACSSC, 2017.
[174] D. Chaum, “Security without identification: Transaction systems to make big brother obsolete”, Communication of the ACM, 28(10):1030–1044, 1985.
[175] S. Brands. “Untraceable off-line cash in wallet with observers”. In Advances in Cryptology, CRYPTO’93, pages 302–318. Springer, 1994.
[176] J. Camenisch, A. Lysyanskaya. “An efficient system for non-transferable anonymous credentials with optional anonymity revocation”. In Advances in Cryptology – EUROCRYPT 2001, pages 93–118. Springer, 2001.
[177] J. Camenisch, A. Lysyanskaya. “Signature schemes and anonymous credentials from bilinear maps. In Advances in Cryptology”, CRYPTO 2004, pages 56–72. Springer, 2004.
[178] https://www.torproject.org/
[179] https://www.osec.doc.gov/opog/privacy/PII_BII.html
[180] https://gdpr.eu/eu-gdpr-personal-data/
[181] https://abc4trust.eu/download/Deliverable_D2.2.pdf
[182] https://web.archive.org/web/20161026044309/https://www.prime-project.eu/
[183] http://primelife.ercim.eu/
[184] https://hyperledger-fabric.readthedocs.io/en/release-1.3/idemix.html
[185] https://www.microsoft.com/en-us/research/project/u-prove/
[186] https://privacybydesign.foundation/
[187] https://decentralized-id.com/government/europe/eSSIF/
[188] https://www.anonos.com/enisa-guidelines-chapter-3-pseudonymisation-scenarios
[189] The Article 29 Data Protection Working Party refers to identifiers as pieces of information, holding a particularly privileged and close relationship with an individual, which allows for identification. The extent to which certain identifiers are sufficient to achieve identification is dependent on the context of the specific personal data processing. Hence, identifiers may be single pieces of information (e.g. name, email address, social security number, etc.) but also more complex data.
[190] According to the CERT Insider Threat Center at Carnegie Mellon University's Software Engineering Institute (SEI), an insider threat is being defined as the potential for an individual who has or had authorized access to an organisation's assets to use their access, either maliciously or unintentionally, to act in a way that could negatively affect the organisation, https://insights.sei.cmu.edu/insider-threat/2017/03/cert-definition-of-insider-threat---updated.html
[191] Janai, J., F. Güney, A. Behl and Andreas Geiger. “Computer Vision for Autonomous Vehicles: Problems, Datasets and State-of-the-Art.” ArXiv abs/1704.05519 (2020)
[192] Ringbom, H., Viljanen, M., Poikonen, J. and Ilvessalo, S. “Charting Regulatory Frameworks for Maritime Autonomous Surface Ship Testing, Pilots, and Commercial Deployments”, Publications of the Ministry of Transport and Communications, 2020
[193] Daily J., Peterson J., “Predictive Maintenance: How Big Data Analysis Can Improve Maintenance”. In: Supply Chain Integration Challenges in Commercial Aerospace. Springer, 2017. https://doi.org/10.1007/978-3-319-46155-7_18
[194] Tinga, T., et al. "Predictive maintenance of maritime systems: models and challenges." European Safety and Reliability Conference (ESREL). 2017.
[195] https://www.scania.com/content/dam/scanianoe/market/au/products-and-services/services/Scania-Connected-Uptime-Brochure.pdf, mining equipment https://new.abb.com/mining/services/advanced-digital-services/predictive-maintenance-mining

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

[196] https://www.konecranes.com/service/predictive-maintenance-and-remote-monitoring/predictive-maintenance
[197] L. Johannesson, N. Murgovski, E. Jonasson, J. Hellgren, B. Egardt, "Predictive energy management of hybrid long-haul trucks", Control Engineering Practice, Volume 41, 2015, Pages 83-97, ISSN 0967-0661.
[198] A. Haseltalab, R. R. Negenborn, G. Lodewijks, "Multi-Level Predictive Control for Energy Management of Hybrid Ships in the Presence of Uncertainty and Environmental Disturbances", IFAC-PapersOnLine, Vol. 49, Issue 3, 2016, Pages 90-95, ISSN 2405-8963.
[199] Seyedan, M., Mafakheri, F. "Predictive big data analytics for supply chain demand forecasting: methods, applications, and research opportunities". J Big Data 7, 53 (2020).
[200] https://developer.nvidia.com/blog/optimizing-warehouse-operations-machine-learning-gpus/
[201] Eclipse Foundation, "Eclipse OM2M," https://www.eclipse.org/om2m/, Accessed: March 2021.
[202] ETSI, "Technical Committee (TC) Smart Machine-to-Machine Communications (SmartM2M)," https://www.etsi.org/committee/smartm2m, Accessed: March 2021.
[203] OpenMTC, "OpenMTC in a nutshell," https://www.openmtc.org/, Accessed: March 2021.
https://platform-sandbox.TradeLens.com/documentation/swagger/?urls.primaryName=Event%20Publish%20API
[204] https://interoperachain.labtlclivorno.it/ui/#/
[205] https://docs.iota.org/docs/node-software/0.1/iri/how-to-guides/install-iri
[206] IOTA Addresses documentation - https://docs.iota.org/docs/getting-started/0.1/clients/addresses
[207] R. Drath and A. Horch, "Industrie 4.0: Hit or hype?," IEEE Ind. Electron. Mag., vol. 8, no. 2, pp. 56–58, Jun. 2014.
[208] The 3rd Generation Partnership Project (3GPP), "3GPP 5G NR (Rel-15)".
[209] D2.1 iNGENIOUS project, https://zenodo.org/record/4683717/#.YHWtwHUzZEY
[210] K. Wang, "Measuring Global Crypto Users A Study to Measure Market Size Using On-Chain Metrics", February 2021
[211] www.hyperledger.org
[212] Chen, Guoxing; Chen, Sanchuan; Xiao, Yuan; Zhang, Yinqian; Lin, Zhiqiang; Lai, Ten H. SgxPectre: Stealing Intel Secrets from SGX Enclaves Via Speculative Execution 2019 IEEE European Symposium on Security and Privacy (EuroS&P) pp. 142-157, 2019
[213] Van Bulck, Jo; Minkin, Marina; Weisse, Ofir; Genkin, Daniel; Kasikci, Baris; Piessens, Frank; Silberstein, Mark; Wenisch, Thomas F.; Yarom, Yuval; Strackx, Raoul - Foreshadow: Extracting the Keys to the Intel SGX Kingdom with Transient Out-of-Order Execution 27th USENIX Security Symposium, 2018
[214] Nilsson, Alexander; Bideh, Pegah Nikbakht; Brorsson, Joakim – A Survey of Published Attacks on Intel SGX, https://arxiv.org/abs/2006.13598, 2020
[215] https://bitnodes.io/

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