NEXT GENERATION INTERNET

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

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

D5.3 Final iNGENIOUS data management platform

Revision: v1.0

Work package	WP5
Task	T5.1, T5.2, T5.3
Due date	31/01/2023
Submission date	31/01/2023
Deliverable lead	TEI
Version	1.0
Authors	Gino Ciccone (TEI), Giuseppina Carpentieri (TEI), Carsten Weinhold (BI), Marek Bednarczyk (PJATK), Tadeusz Puźniakowski (PJATK), Cesar Rodriguez Cerro (TIOTBD), Jussi Poikonen (AWA), Alexandr Tardo (CNIT), Ivo Bizon (TUD), Rania Rojbi (TUD), Christos Politis (SES); Ignacio Benito Frontelo (NOK), Enrique Sierra Garcia (ASTI), Giacomo Bernini (NXW), Erin Seder (NXW)
Reviewers	Gino Ciccone (TEI), Giuseppina Carpentieri (TEI), Carsten Weinhold (BI), Yevhen Zolotavkin (BI), Tadeusz Puźniakowski (PJATK), Cesar Rodriguez Cerro (TIOTBD), Jussi Poikonen (AWA), Alexandr Tardo (CNIT), Mariano Falcitelli (CNIT), Ivo Bizon (TUD), Rania Rojbi (TUD), Christos Politis (SES), Francisco Javier Curieses Sanz (UPV), Nuria Molner (UPV), Rodrigo Martinez (ASTI), Giacomo Bernini (NXW)
Abstract	This document describes the final solution of iNGENIOUS interoperable layer and gives a detailed technical description of how components coming from different existing and forthcoming IoT technologies are integrated.
Keywords	Smart IoT GW, IoT, DVL, Cross-DLTs, Smart application, Interoperable Layer, TrustOS, Predictive Analytics, Industrial Tactile applications, Secure Communication

Document Revision History

Version	Date	Description of change	List of contributor(s)
V1.0	31/01/2023	EC version	See author list

Disclaimer

This iNGENIOUS D5.3 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 Final Review Meeting planned in July 2023, 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

The coexistence between fixed, radio and satellite access technologies, the interaction of different heterogeneous M2M systems and the large amounts of data and information produced by the NG-IoT devices create complex new scenarios which require the overcoming of compatibility and data harmonization issues, arising also from the EU directives about data privacy and security.

The IoT interoperability enables the federation of different IoT platforms within heterogeneous domains, overcoming the compatibility issues between both standard and non-standard, proprietary, and custom M2M solutions

This deliverable describes the final solution of iNGENIOUS interoperable layer and gives a detailed technical description of how components coming from different existing and forthcoming IoT technologies (described in D5.2 [7]) are integrated with an emphasis on addressed innovations.

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

Abbreviations

5CMM	FiveComm
AGV	Automatic Guided Vehicle
AI	Artificial Intelligence
AIS	Automatic Identification System
AMQP	Advanced Message Queuing Protocol
API	Application Programming Interface
BI	Barkhausen Institut
CDMA	Code-division multiple access
DAG	Direct Acyclic Graph
DLT	Distributed Ledger Technology
DVL	Data Virtualization Layer
E2E	End-to-End
ETA	Estimated time of arrival
EVM	Ethereum Virtual Machine
FDMA	Frequency-division multiple access
FPE	Format Preserving Encryption
GDPR	General Data Protection Regulation
GPS	Global Positioning System
GW	Gateway
GUI	Graphical User Interface
HAL	Hardware Abstraction Layer
HWK	Hash Without Key
HTTPS	HyperText Transfer Protocol over Secure Socket Layer
ICT	Information and Communications Technology
IN-CSE	Infrastructure Node Common Service Entity
IoT	Internet of Things
IP	Internet Protocol
JDBC	Java DataBase Connectivity
JSON	JavaScript Object Notation
M2M	Machine-to-Machine
MAC	Media Access Control
MEC	Multi-access edge computing
MN-CSE	Middleware Node Common Service Entity
MQTT	Message Queuing Telemetry Transport
OSI	Open Systems Interconnection
PoS	Proof of Stake
PoW	Proof of Work
PF	Pseudonymization Function
QoS	Quality of Service
RAN	Radio Access Network
RATLS	Remotely Attested Transport Layer Security
REST	Representational State Transfer
SDMA	Space-division multiple access
SDR	Software-defined radio
TDMA	Time-division multiple access
TLS	Transport Layer Security
TTT	Truck Turnaround Times
VPN	Virtual Private Network
VR	Virtual Reality

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

1 Introduction

This document concludes the WP5 work, which consists of a trilogy of deliverables:

- D5.1 [3], where the state of the art and the first sketch of the proposed solution of the iNGENIOUS Data Management Platform has been described.
- D5.2 [7], where the iNGENIOUS Data Management Platform has been designed, providing and describing technical details of its first release.
- D5.3, this document, where the iNGENIOUS Data Management Platform is finalized.

1.1 Objective of this Deliverable

The main goal of this deliverable is to describe the final iNGENIOUS Data Management Platform, including the Applications and Data Analytics layer and the Data Management layer, which manipulates and stores data received from heterogeneous machine-to-machine (M2M) platforms that serve the different stakeholders' supply chain, as shown in Figure 1.

Moreover, the employment of the different architecture components in the realization of the iNGENIOUS use cases is presented. The business potential and exploitation plans are summarized at the end of each chapter, and further detailed in deliverable D7.3, [9].

1.2 Role of WP5 in iNGENIOUS Architecture

WP5 work concentrates on the two top layers of the INGENIOUS architecture as shown in Figure 1Figure 1. An overview in the form of a description of individual components and component groups can be found in iNGENIOUS deliverables D2.2 [1] and D2.4 [2], respectively. We summarize the WP5 role within the architecture below.

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

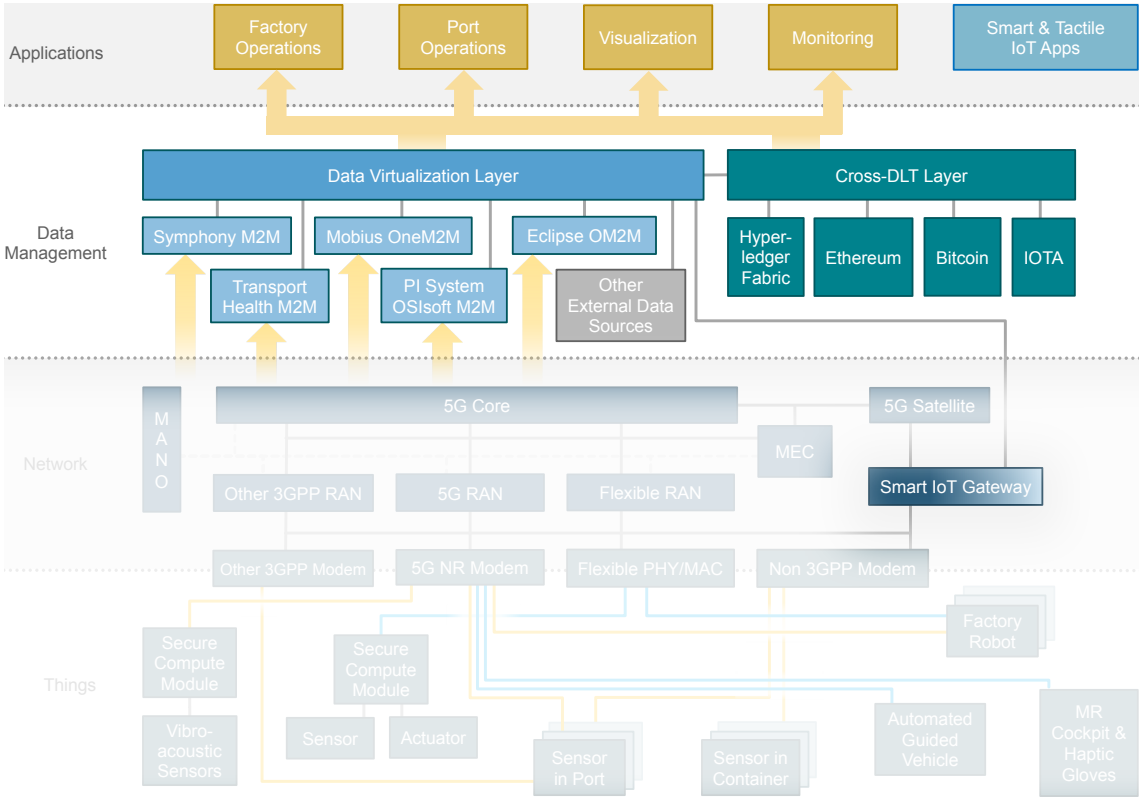

FIGURE 1: WP5 COMPONENTS WITHIN THE INGENIOUS CROSS-LAYER ARCHITECTURE

WP5 is about the applications that implement the logistics and supply chain use cases, as well as the data management components that enable those applications. The three key iNGENIOUS innovations in the data management layer are 1) the Data Virtualization Layer (DVL), which enables uniform access to many otherwise incompatible machine-to-machine (M2M) platforms, 2) the Cross-DLT Layer, which enables integrity-protected data storage of supply chain events across multiple distributed ledger technology (DLT) networks, and 3) a smart Internet of Things (IoT) gateway, which enables integration of IoT sensors using various connectivity technologies and which also performs data transformation and processing.

1.3 Role of WP5 in the Use Cases

The role of WP5 is to develop the interoperability layer with data management capabilities as well as to enable smart and tactile IoT applications on top of it, ensuring that security and privacy aspects are properly treated by identifying risk profiles and protection schemes for iNGENIOUS assets. Several iNGENIOUS use cases are covered by the development activities performed in this work package according to the table below:

Use case definition	Use case abbreviation	WP5 component
Automated Robots with Heterogeneous Networks	**Factory**	IoT Application Layer
Transportation Platform Health Monitoring	**Transport**	N/A

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

Situational Understanding and Predictive Models in Smart Logistics	**Port Entrance**	IoT Application Layer, Data Virtualization Layer, Distributed Ledger Technologies Layer
Improve Driver's Safety with MR and Haptic Solutions	**AGV**	IoT Application Layer
Inter-Model Asset Tracking Via IoT and Satellite	**Ship**	Smart IoT Gateway, Data Virtualization Layer, Distributed Ledger Technologies Layer
Supply Chain Ecosystem Integration	**DVL/DLT**	IoT Application Layer, Data Virtualization Layer, Distributed Ledger Technologies Layer

TABLE 1: RELATION BETWEEN WP5 COMPONENTS AND INGENIOUS USE CASES

Further details of the relation with iNGENIOUS use cases are provided in Subsections 2.1.1, 2.2.1 and 2.3.1 as well as in Sections 3.2, 4.3 and 5.3 of this document respectively.

1.4 Structure of the Document

This deliverable is organized into four main sections, briefly described below:

- Section 2 - *IoT Application layer:* outlines the IoT applications implemented in the project use cases, focusing mainly on port operations. The main application areas are AI algorithms for predicting traffic rates and congestion at the port based on heterogeneous data sources along the maritime supply chain, and a platform for remotely operating automated guided vehicles in the port area using a variety of sensors and actuators in the vehicle and remote operation station.
- Section 3 - *Smart IoT Gateway*: focuses on the Smart IoT Gateway, a physical element with multiple interfaces that allows end-to-end M2M communication, describing the use of the industry standard M2M protocol Message Queuing Telemetry Transport (MQTT) to complement the oneM2M based solution.
- Section 4 - *Data Virtualization Layer*: focuses mainly on the integration activities with the supported external data source platforms (OneM2M, Symphony M2M, Eclipse OM2M, PISystem M2M, and Tuscan Port Community System). The section also describes the finalization of the Pseudonymization Function, a DVL module that guarantees the pseudonymization of the personal data handled by the layer.
- Section 5 - Distributed Ledger Technologies: this section introduces the Cross-DLT layer, a software component connected to multiple Distributed Ledger Technologies (Bitcoin, Ethereum, HyperLedger Fabric, and IOTA). In this last deliverable, integration activities are described between TrustOS and the connected DLTs and between TrustOS and DVL via the introduced Integration Bridge, the role of TrustOS in Cross-DLT is

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

described in section 5.1 Integration activities. Furthermore, a graphical user interface is described which shows in a simple and visual way the information stored in TrustOS and in the different DLTs.

Each section also describes the relationship with the use cases involved, the innovations introduced and finally a summary of the business potential and exploitations.

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

2 IoT Application Layer

In D5.2 [7] we identified the main functionalities, data requirements, and system components for IoT applications in the project use cases. These are divided on the high level into two main topics: predictive analytics and industrial & tactile IoT applications. For predictive analytics, we described the model components to be developed, historical data required for the development, and initial estimations on the continuous data integrations required to deploy these models as an online service. For industrial & tactile applications, the system components and data sources planned for developing an end-to-end (E2E) platform for remotely controlling automated guided vehicles (AGVs) in the port area were described.

In the following, we describe the final implementation of these applications with respect to the project use cases, their requirements, and KPIs. Furthermore, we consider the innovation aspects and exploitation plans of the applications.

2.1 Predictive Analytics

In short, the predictive analytics application focuses on using data from heterogeneous sources in the port environment to train and apply predictive models providing estimates on future traffic conditions to actors interested e.g., in better coping with expected congestion. Figure 2 shows an outline of the relevant data sources and model components. As further details on the data flow and application components of the predictive analytics application, Figure 3 illustrates the data sources, necessary data elements or features, component models, and their role in the final predictive simulation model. Model development and performance evaluation are discussed in more detail in D6.3 [12].

Implementation of the final predictive analytics application includes two interdependent phases: model training and online predictions. Model training involves the preparation of historical datasets to represent the expected inputs and outputs of the predictive application, and selection and optimization of models and model parameters to produce the expected outputs from the available input data. Furthermore, the model training process involves testing and validation of the training process itself to ensure that the obtained models are valid also for data not present in the historical datasets used.

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

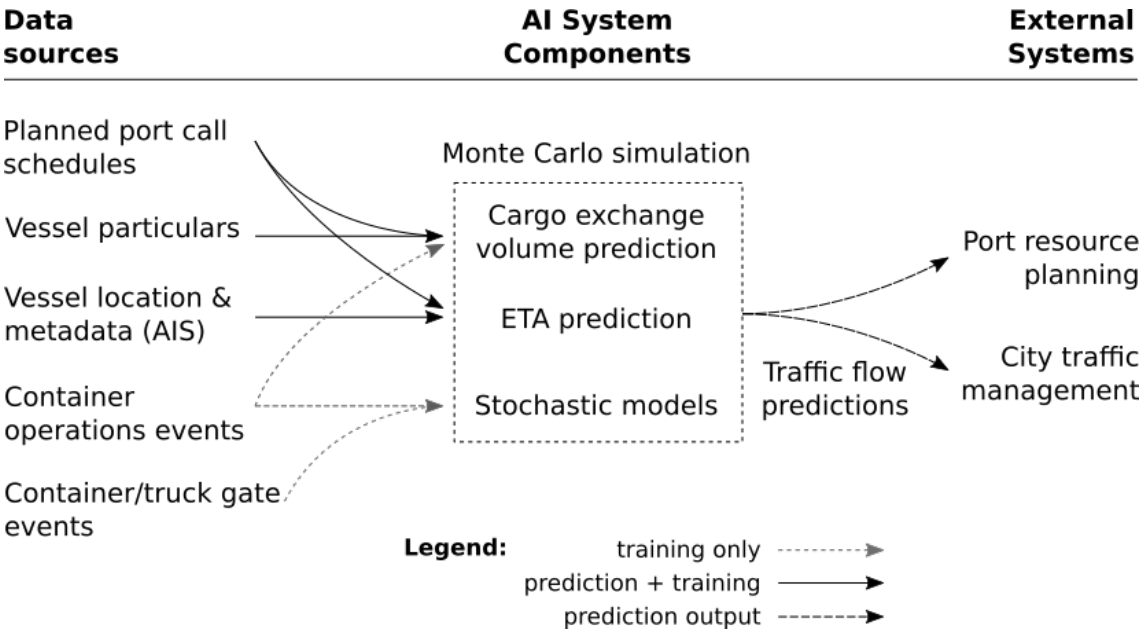

FIGURE 2: OVERVIEW OF THE DATA FLOW IN THE PREDICTIVE ANALYTICS APPLICATION

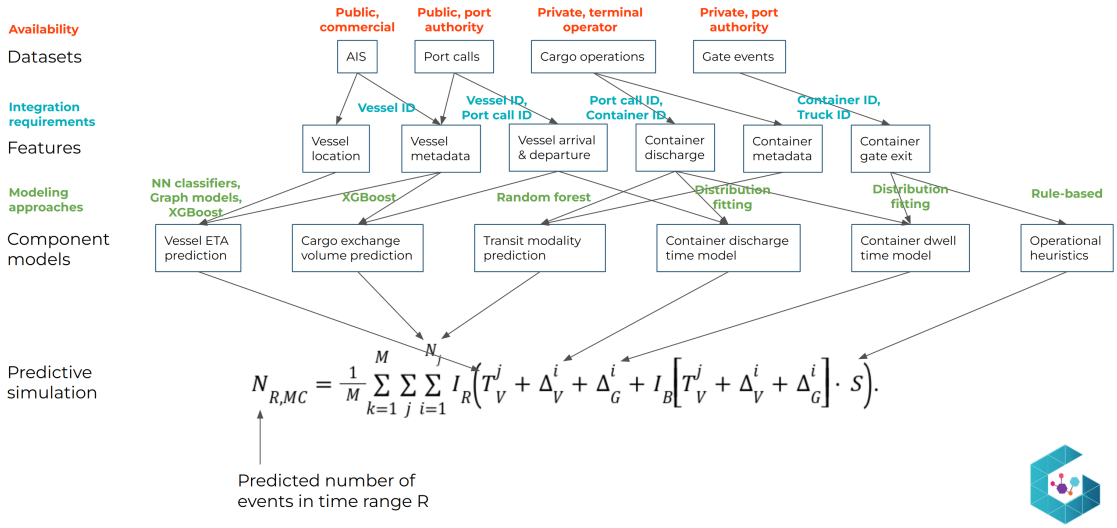

FIGURE 3: DATA FLOW AND MODEL COMPONENTS OF THE PREDICTIVE ANALYTICS APPLICATION

To perform online predictions using the developed models, cloud-based services are developed for model deployment, ingestion of up-to-date input data from other systems (such as the iNGENIOUS data virtualization layer), and providing generated prediction data to other systems and users through application programming interfaces (APIs) and web-based user interfaces.

2.1.1 Relation to the Use Cases

With respect to the Port Entrance use case as specified in D2.1 [11], the main application requirements are fulfilled by implementing a web service providing access to a dashboard visualizing the relevant analytics information as illustrated in Figure 4 below. In this context, the main objective is to

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

visualize the operation of the developed prediction services and provide easy visual access to the predictions, but in commercial exploitation of the system, the outputs can be connected via APIs to external systems to provide supporting information for port and hinterland operations planning. The original Use Case requirements focus mainly on the truck turnaround times (TTT), but in the final application we consider as an additional main output the prediction of the total truck traffic rate (truck/container exits at the port gate) over time. This provides another metric reflecting the congestion at the port and also the effect on the overall road traffic in the surrounding city area.

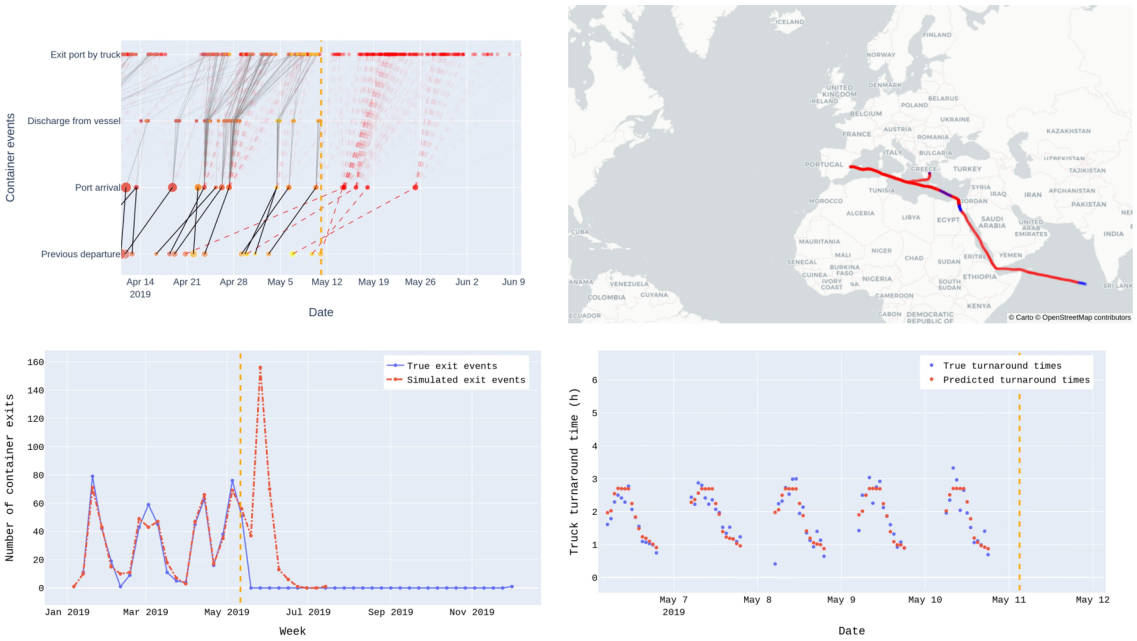

FIGURE 4: WEB SERVICE DASHBOARD COMPONENTS FOR THE PREDICTIVE ANALYTICS APPLICATION. TOP RIGHT: PREDICTED VESSEL ROUTES TO THE DESTINATION PORT. TOP LEFT: TIMES OF SIGNIFICANT CONTAINER EVENTS RELATED TO HISTORICAL AND PREDICTED PORT CALLS. BOTTOM LEFT: AGGREGAT

Some aspects of the planned Use Case cannot be implemented fully at this point due to the lack of access to up-to-date data. Particularly, this involves the visualization of all historical traffic rates and events and related prediction performance measurements. Since continuous access to the truck traffic events is currently not available in either the port of Valencia or the port of Livorno, these can only be presented using historical data which is updated manually on demand.

Regarding the planned system requirements for the application, increasing the port and terminal performance is an objective that requires the application to be integrated into production use in port/terminal operations planning, with actions taken in the operations based on the provided information. This cannot be fully tested within the project, and falls under business potential and exploitation, as discussed in the relevant section below. Otherwise, the planned system requirements can be met with the planned final implementation of the application.

Regarding the performance of the developed application, the planned target of 10 % reduction in truck turnaround/idling times is also related to further exploitation of the work as discussed above. The final prediction accuracy of the implemented models remains to be evaluated with comprehensive data sets, but we see the objective of 90 % relative prediction accuracy as achievable when required input data is available. Regarding data source

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

sufficiency, the planned historical data is available for model and service development, but full access to required real-time data sources is not available in either of the target ports. The effects of missing data elements on performance targets are discussed in more detail in D6.3 [12], along with other requirements and KPIs achieved. There, data quality is also considered from the perspective of whether the available historical data are of sufficient quality to train accurate models for the task, and whether the data quality available for online services is sufficient to reach estimated performance targets. Thus, regarding the predictive modeling objectives, data quality can be evaluated in a quantitative manner from the perspective of the data quality characteristics (accuracy, completeness, consistency, credibility, currentness) and properties specified in ISO/IEC 25012 and ISO/IEC 25024, as applicable.

2.1.2 Innovation

As described in D2.4 [2], the main innovation component in predictive analytics is building predictive models for port logistics processes and traffic patterns using heterogeneous data sources provided by the iNGENIOUS architecture. The developed solution enables actors in the maritime supply chain to share data for prediction model training in a way that does not expose potentially sensitive information to external parties. One of the research objectives in the project has been to study what parts of the data needed for the technical model development contain sensitive information, and how privacy can be ensured. Sensitive information refers here to both operational business information which is not related to personal privacy (such as cargo operations efficiency in a given port terminal) and data which may contain personally identifiable information (such as vehicle license plate data). From the user perspective, privacy is then maintained in that the prediction service outputs contain only variables which are generally obtainable from public sources (such as vessel schedules and truck traffic rates) but are here produced in a predictive manner. These models can also be applied with public non-sensitive data sources to provide information which can benefit other actors in the supply chain such as port resource planners of city traffic management, if taken into account in operational planning. This kind of data sharing is not common in the maritime supply chain, where the actors are typically operating in silos due to the perceived sensitivity of the type of data elements described above; the developed system is a practical innovation to facilitate increased data sharing in the industry through abstraction of the sensitive elements of the data to machine learning and stochastic models.

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

2.1.3 Business Potential and Exploitation

Component Group: Predictive Analytics

- **AWA**: Several business cases exist for both the individual components of the developed predictive analytics application and the application as a whole. The vessel estimated time of arrival (ETA) prediction has already been commercialized by AWA during the project and is offered globally as a cloud-based service. Cargo operations duration prediction, which is another subcomponent of the application, is of commercial interest e.g. for terminal operators and cargo owners, and discussions on related commercial pilots are currently ongoing. Regarding the predictive analytics application as a whole, commercial interest has been expressed by major city ports and multimodal logistic operators, who would benefit from foresight into overall truck traffic rates and congestion at the port. Here also, discussions on commercial pilots are ongoing.

  For the ports and port-related research partners participating in the development (FV, CNIT, AdSPMTS), the exploitation potential of the results relates to the increased understanding of how data sharing within the port community and with new stakeholders and service providers can enable producing beneficial information for all parties. Furthermore, incorporating the developed models and application into operational planning by the port or city traffic management offers possibility of improving operational efficiency by offering foresight into upcoming congestion periods.

  Further details about the innovation, exploitation and impacts are reported in D7.3 [9] as well as in D2.5 [10].

2.2 Industrial & Tactile IoT applications

With the rapid advancement of wireless communication technologies, numerous emerging solutions and applications in industrial sites have been developed to ensure flexible, dynamic, self-organized, and product-customized factories. In such a scenario, a large number of sensors and actuators are deployed within the industrial floor, where the outcomes of the sensors are translated to actions performed by actuators in order to execute a user-defined (customized) task. If the information exchange required to perform the tasks is carried out via wireless links, the application is here named a tactile IoT application.

Tactile IoT applications refer to a communication network that provides real-time control, and sensor/action information through a sufficiently reliable, and intelligent connection, within industrial systems. It is designed to offer a promising opportunity to reshape industrial communication and transform the operation of many existing industrial sites. The main task of such an application is realized by means of one or more processes that define the relations between a set of inputs and a set of outputs. The outputs of some processes might be used as inputs for others.

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

In other words, industrial and tactile IoT applications are made up of a set of operations that allow the application developer to get data into and out of the system in a unified manner. Within this context, industrial IoT applications require high availability, reliability, and security.

As an example, efficient automated material transportation with the factory plant is key for enabling higher levels of automation, and downtime in the transport system impacts productivity significantly. With this in mind, the employment of automated guided vehicles (AGVs) for such tasks is an attractive option. However, in the current technology, the major part of sensor and command data processing happens at the device. This operation mode hinders a flexible and dynamic transportation system, where multiple AGVs can be controlled and monitored from a central command unit. In order to accomplish this task, a common data exchange structure has to be agreed upon among the components, i.e., AGVs, controlling units, and sensors. Therefore, an API that allows this common ground for communication appears as an attractive way to implement such technology.

To demonstrate this concept within iNGENIOUS, the integration between a non-3GPP compliant radio access network (RAN), i.e., the Flexible physical layer (PHY), and management and orchestration (MANO) is done using JavaScript Object Notation (JSON) format, which consists of an open standard file format for data exchange. A JSON file stores various data types in key-value pairs in a human-readable format, with the keys serving as names and the values containing the related data. An example is illustrated in Figure 5. It is commonly used for APIs because of its lightweight to send back and forth as it has a small file size. And it is also easy to manipulate (read/write) compared to other data formats, as JSON files are structured and written in an organized and clean way.

In our case, the MANO sends a JSON file containing the resource allocation for each application, and the PHY extracts this information and distributes the resources accordingly.

```
{
"Robot": {
        "ID":"AGV",
        "NavigationRange":{
                "max":"0m",
                "min":"20m",
                 },
        "MaxSpeed": "1.2m/s",
        "MaxLoad":"35kg",
        "BatteryLife":"12h",
        "Weight":"100kg",
        "Camera":"Yes"
        "RequiredDataRate":"5Mbps"
        }
}
```

FIGURE 5: EXAMPLE OF A JSON FILE CONTAINING INFORMATION ABOUT AN AGV DEVICE. THIS INFORMATION IS SHARED WITH THE IOT APPLICATION DEVELOPER

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

2.2.1 Relation to the Use Cases

The concept of the Industrial and Tactile API is employed in the Factory and Transport use cases since it allows the employment of multi-access edge computing (MEC), which in turn allows the creation of industrial processes that require local computation for attaining latency constraints. The API was essential for allowing the integration between the flexible PHY/MAC network with the MANO, as well as the integration among the AGVs provided by ASTI and the cockpit from 5CMM.

Requirements and KPIs achieved, are reported and discussed in D6.3 [12] accordingly.

The concept of the Industrial and Tactile API is employed in the AGV use case (Improved Drivers' Safety with MR and Haptic Solutions) since it allows the employment of multi-access edge computing (MEC), which in turn allows the creation of industrial processes that require local computation for attaining latency constraints. The API was essential for allowing the integration between the flexible PHY/MAC network with the MANO, as well as the integration among the AGVs provided by ASTI and the cockpit from 5CMM.

There is a constant flow of KPIs and statistical real-time information flowing from the AGV from and to the cockpit components in the use case. MEC instances are in charge not only of enabling the scalability and low latency for the network connectivity in the industrial area but for registering and hosting the KPI database.

A Grafana installation is hosted in the edge in which latency and network statistics can be retrieved either from real-time events but also for past experiments and use case tests and demonstration.

All the data is stored in InfluxDB database, and the data is being postprocessed and redirected using Telegraf agents.

One example are the specific latency and Speed stats for the AGV as seen in Grafana dashboard in **Error! Reference source not found.**

FIGURE 6: GRAFANA DASHBOARD WITH LATENCY KPIS FOR RD

The importance of low-.latency communications is a must and we should be able to constantly measure it during all the tests as depicted in FIGURE 7.This is achieved by deploying a MEC system which has direct connectivity with radio network in isolated VLANs connected to optimized linux appliances that process all the data and act as gateways for transmitting the data payloads

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

from the AGVs and asynchronously register the KPI information in other external server to the MEC infrastructure.

FIGURE 7: GRAFANA LATENCY INFORMATION AND BANDWIDTH USAGE

The user is able to have real-time information on the data flow including the latency RTT statistics both in the command-line interface output running in the cockpit component (see FIGURE 8) as well as inside the VR application with color-scaled representations of such latency in milliseconds, FIGURE 9.

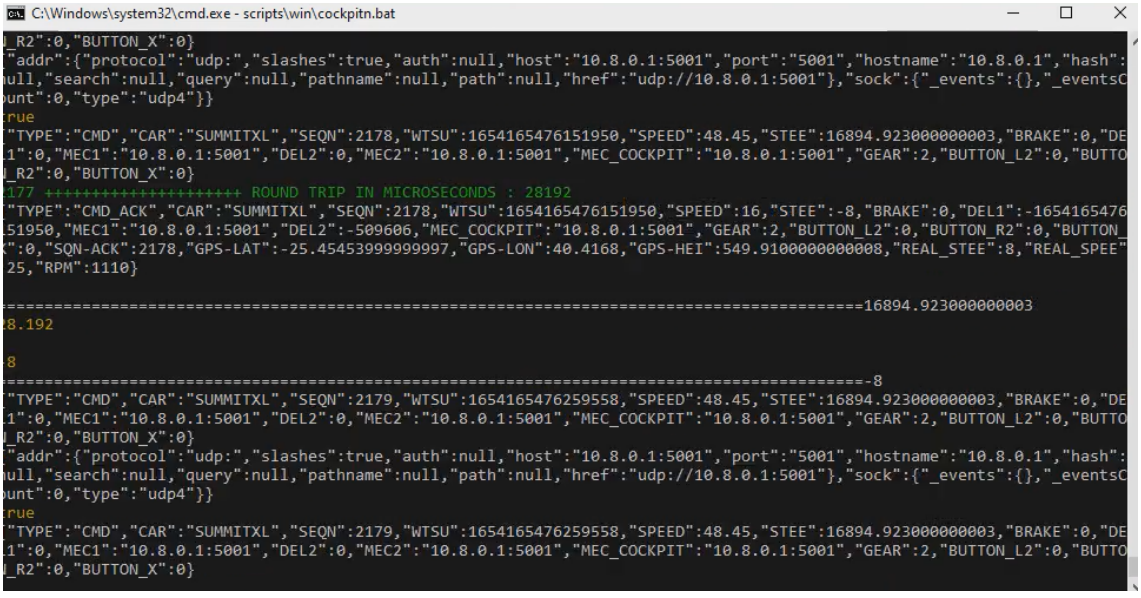

FIGURE 8: COMMAND LINE JSON REAL-TIME STATS

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

FIGURE 9: COCKPIT VR LATENCY KPI

For this use case we measure the RTT for both the video streams and the control commands issued and acknowledged from the AGV. Another related KPI is the number of frames missed or dropped during the transmission which are also stored in Grafana and have their own entries in the dashboards and influxDB databases.

Requirements and KPIs achieved, are reported and discussed in D6.3 [12] accordingly.

2.2.2 Innovation

The developed API enables resource allocation from the MANO layer on non-3GPP RAN. Furthermore, the implemented flexible MAC design includes the multiple access schemes based on time-division multiple access (TDMA), but in general other schemes such as frequency-division multiple access (FDMA), space-division multiple access (SDMA), code-division multiple access (CDMA). And any combination of them is possible. Finally, the Integration with the 5G core is achievable through the MANO layer.

2.2.3 Business Potential and Exploitation

Component Group: Industrial & Tactile IoT application
• **TUD**: The Flexible PHY/MAC implementation in software-defined radio (SDR), and its integration with MANO through the industrial and tactile API allows for

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

a unique testing platform where end-to-end performance metrics can be gathered from the integration of techniques such as non-3GPP RAN and 5G core. This will further develop the industrial wireless IoT network landscape.

- **ASTI**: The Tactile API allows the control of ASTI AGVs wirelessly through the 5G modem of 5CMM. The control data is encapsulated and can be transferred over a fully 5G compliant network. Thus, experiments and development of applications for industry 4.0 can be easily carried out due to the simpler integration among components which use different structures for data exchange.

  Further details about the innovation, exploitation and impacts are reported in D7.3 [9] as well as in D2.5 [10].

2.3 Secure Communication and Cooperation

In the last section of this chapter, we discuss the importance of network and operating system-level support for building secure IoT applications that consist of multiple individual components. Generally, all IoT use cases require secure communication. Many of them can also benefit from additionally securing the endpoint devices that communicate with each other. In iNGENIOUS, innovations regarding this second aspect are evaluated within the Transport use case.

IoT systems are distributed systems, where cooperating software components run in multiple physically separate locations. Individual programs run on IoT devices at the edge, on servers in an edge cloud or remote data centre, and on stationary or mobile end-user devices such as office PCs, laptops, and hand-held devices. The software components on all these devices must work together to implement the functionality of an IoT application or use case. To enable this cooperation, the software components must communicate over networks to bridge the physical distance between them. Thus, it is critically important that the communication channels between the software components are secure, such that the distributed IoT applications are resilient against attacks on network connections. However, the security of the endpoints is equally important, because software components running on different machines must trust each other to work correctly. This means it must be ensured that all of them do their part of the joint work as required to implement the overall use case.

We briefly discuss the state of the art for secure communication. This summary allows us to clearly point out the iNGENIOUS innovation.

State of the Art: Transport Layer Security (TLS) is the state-of-the-art protocol for securing communication channels between two programs on separate computers. It encrypts all information transmitted over the communication channel and protects its integrity using message authentication algorithms. In addition to guaranteeing confidentiality and integrity, TLS ensures the authentication of communication partners. This assurance relies on the possession of cryptographic keys, which must be kept secret by the peer who wishes to authenticate itself to its communication partner. In practice, this authentication method requires that users trust the other party to operate their remote computer securely. If Alice wants to exchange data via TLS with a computer operated by Bob, she has to make two assumptions: 1) Bob keeps

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

secret the cryptographic keys required for TLS authentication and 2) the software running on Bob's computer does what he claims (e.g., not leak data received from Alice).

Unfortunately, TLS alone cannot provide verifiable evidence that assumptions 1) and 2) hold. However, in certain use cases, including many IoT scenarios, such proof is desirable. For example, lives could be at stake if an attacker managed to manipulate the safety features in the firmware of an IoT actuator device, which then may misbehave and harm human workers nearby. Likewise, an IoT device should only accept commands sent by the "right" control computer. Both IoT devices in the field and edge servers outside highly guarded data centers have a higher risk of attackers compromising TLS keys (assumption 1) or the integrity of software (assumption 2) on the device or server. Therefore, technical measures are needed to detect, if the integrity of an IoT device has been compromised.

Remote Attestation is a cryptographic protocol that can supplement TLS by solving the two trust issues described above. First, using hardware support, it ensures better protection of cryptographic keys. Second, using these better-protected keys, it can provide a computer, the *challenger*, verifiable proof that the software running on another computer, the *attester*, is in the correct state. In summary, remote attestation works as follows:

1. **Identifiability:** The hardware of the attester includes a root-of-trust that contains a unique cryptographic identity key that cannot be forged[1]. By verifying signatures created by this key, the challenger can identify the attester device and therefore make assumptions about its capabilities and security properties.
2. **Integrity:** Using its identity key and a verified startup process, the root-of-trust of the attester can create a digital signature over the software on the device. This signature allows the challenger to know what software currently runs on the attestor and, thus, if it can be trusted to behave as required.

A root-of trust is designed and built to be much more difficult to compromise than pure software solutions, because hardware needs to be attacked. In Chapter 3 of iNGENIOUS deliverable D3.3 [4], the design of a root-of-trust for BI's M³ hardware/software co-design platform is described, including the system-level components to use it (part of Task T3.2 of WP3 in the project). In the following, we give an overview of the application-layer functionality built on top of roots-of-trust.

Integration of Remote Attestation into TLS: INGENIOUS partner Barkhausen Institut (BI) combined remote attestation with TLS, thereby creating a combined protocol called *RATLS* (Remotely-Attested Transport Layer Security). This combined protocol provides secure communication based on the widely-used TLS standard, but with the additional security guarantees provided by remote attestation.

The point where both protocols are joined is the connection-establishment phase, which is called *handshake*. After the handshake, when a cryptographically-secured connection has been established between the two peers, RATLS behaves exactly as TLS except for one detail.

[1] Under an attacker model that typically assumes that hardware attacks to extract the identity key are not feasible or too difficult.

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

Standard TLS supports mutual authentication, where each of the two peers authenticates to the other. Mutual remote attestation is similar in the sense that after the protocol is completed, each peer attested itself (in the role of an attester) and verified the identity and integrity of the other peer (in the role of a verifier). One-way authentication (e.g., when a web server proves its identity to a web browser) is a common special case and RATLS supports it, too. However, in this report, we consider only the general case of mutual authentication. Both standard TLS and remote attestation are challenge-response protocols that use a two-way handshake as shown in Figure 10.

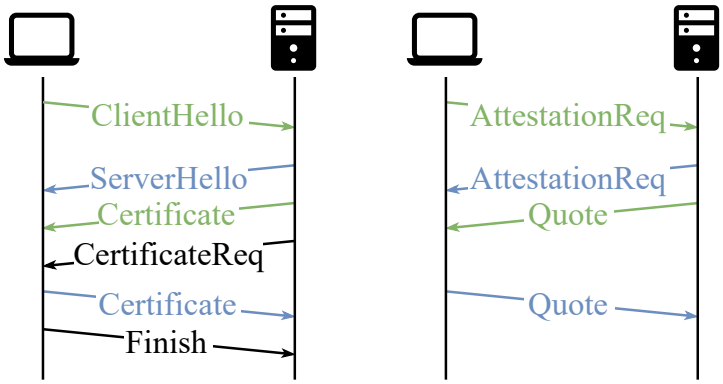

FIGURE 10: TWO-WAY HANDSHAKE OF TLS (LEFT) AND REMOTE ATTESTATION (RIGHT)

The structure of both handshakes is similar. The colors highlight how conceptually similar requests and responses of the TLS and attestation protocols are exchanged via messages between the two peers. Where TLS sends a (client or server) *Hello* message that is answered by a *Certificate* message, a verifier sends an *Attestation* request, which is answered by the attester with a *Quote*. In both protocols, multiple messages traveling in the same direction at the same time (e.g., *ServerHello* and *Certificate* for TLS) are merged into one combined message at the network level. Thus, the number of network roundtrips is the same in both protocols.

In RATLS, we exploit the structural similarity and the ability to merge messages at the network level by piggy-backing remote-attestation messages onto TLS protocol messages. To this end, RATLS builds on a feature of the TLS v1.3 standard [5], where applications can attach custom extensions to TLS handshake messages. With this mechanism, RATLS can transmit messages related to attestation on top of TLS handshake messages. The RATLS approach can be applied to any TLS implementation that supports handshake extensions.

BI developed a proof-of-concept implementation of RATLS as an accompanying library for the widely used OpenSSL [14] implementation. This implementation targets OpenSSL's extension API, but the design of RATLS is independent of the specific TLS library. RATLS is also agnostic to the underlying hardware root-of-trust. The details of the RATLS design and our implementation are described in a publication [6]. In this report, we summarize the benefits of the RATLS approach.

The key information that is exchanged between the RATLS peers is the *Quote*, which is an attestation report (or evidence) of the attesting peer's identity and software integrity. These quotes are generated by the roots-of-trust of the

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

participating peers. The content of the Quote is opaque to the RATLS implementation, as it is not examined by RATLS itself but only transmitted via handshake messages. Instead, RATLS relies on attestation provider plugins that utilize the platform-specific root-of-trust of the computers that represent the RATLS endpoints. It is the responsibility of the attestation provider to create a Quote for its own software state and to verify the reported evidence created by the other machine's root-of-trust. As a result of this design, there are several noteworthy properties of the RATLS design and implementation:

- **"Don't Roll Your Own Crypto":** The TLS standard and its implementations have been subject to extensive review by a huge number of experts in the fields of computer security and distributed-systems engineering. RATLS avoids changes to the TLS protocol and, in the case of our proof-of-concept based on OpenSSL, does not require modification of the TLS implementation. By avoiding modifications to the TLS protocol and implementation, we minimize the risk of RATLS introducing new security vulnerabilities and keep maintenance overhead low.
- **Minimal Number of Handshake Messages:** The number of messages that are exchanged (and must be waited for) at the network level is the same for pure TLS and RATLS. There is, however, cryptographic overhead to the generations and verification of quotes.
- **Low-Barrier Adoption:** The RATLS API is identical to the TLS API of OpenSSL, except for library initialization, which involves activation of the RALS companion library. This simplicity will make it easy for application developers to upgrade their communication from traditional TLS to TLS with remote attestation.
- **Separation of Concept and Realization:** RATLS extends TLS with remote-attestation support in such a way that the concept of remote attestation is agnostic to the root-of-trust of the underlying platform of the peer. Specific roots-of-trust and attestation reports (evidence encoded in Quote formats) are separated into platform-specific attestation provider plugins. At the time of writing, a plugin for industry-standard Trusted Platform Modules (TPMs) exists and support for other roots-of-trust like Intel SGX and BI's hardware/software co-design platform are work in progress.
- **Session Resumption:** TLS supports an optimization, where a client can resume a recently-closed session instead of performing the complete TLS handshake with the server. This feature, called *session resumption*, reduces the cost of establishing a secure communication channel, if two peers must open and close connections frequently. RATLS supports a variant of TLS session resumption that minimizes the cryptography-related costs for creating and validating quotes.

2.3.1 Relation to the Use Cases

The trust problem between software components cooperating across multiple computers is a vertical (cross-layer) and horizontal (within layer) problem. It affects all distributed applications, including IoT applications. BI

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

addresses this challenge by innovating on *Remote Attestation*, which is a technical solution to this problem.

Within the project, BI contributes to the Transport use case, where RATLS is used as replacement for standard TLS to report train carriage health information securely to the train operator's control center. But remote attestation could help secure cooperation and communication between networked devices and software components of all other use case cases, where the distributed components must communicate and cooperate securely.

Requirements and KPIs achieved, are reported and discussed in D6.3 [12].

2.3.2 Innovation

Remote attestation is a known technique and multiple implementations exist. But it is difficult for application developers to deploy it, because the available implementations do not easily integrate with system services that applications depend on. By combining remote attestation with TLS, BI created a combined protocol called *RATLS*. This combined protocol can be used as a drop-in replacement for the widely-used TLS, with minimal effort by application developers. The gain for this effort is additional security guarantees provided by remote attestation, namely the verification of the identity and integrity of software running on a remote IoT device or cloud server.

In summary, RATLS is an innovative and universal building block for securing distributed systems with low effort for developers. It can help building a more secure and trustworthy IoT.

2.3.3 Business Potential and Exploitation

Component Group: Secure Communication and Cooperation

- **BI**: In iNGENIOUS, the trust and security benefits of securing communication channels between IoT devices and cloud servers are evaluated in the Transport use case. In this use case, the market potential of smart edge sensors such as those developed by iNGENIOUS partner NeuroControls (NCG) is enlarged. However, the security advantage of remote attestation and the ease-of-use of an implementation like RATLS can benefit all usage scenarios, where networked devices need to cooperate across physical and organizational distances, including cases where different stakeholders operate the devices.
  As an academic research institute, BI will exploit and further develop RATLS in future research projects and teaching students, who will bring a security-focused mindset into future employment in the industry. The current state is published and it is the intention of BI to make future developments available as open source for easier adoption by academics and industry.

  Further details about the innovation, exploitation and impacts are reported in D7.3 [9] as well as in D2.5 [10].

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

3 Smart IoT Gateway

In the deliverable D5.2 [7], we presented the development activities in order to create a workflow allowing to receive (sensor) data from different sources via different means of communication and transform, interpret and route this data in a unified way via different routes to a central cloud M2M infrastructure. The main challenge to overcome was how to implement a system that handles network traffic via multiple routes, that can be manipulated on the application layer, while still keeping it transparent to the data flow of the system and meeting the requirements of security and confidentiality. This was achieved by using an oneM2M compliant system in combination with encrypted VPN connections and layered HTTP reverse proxies. oneM2M defines a standardized, HTTP(S) based communication framework, which is routed through configured VPN connections to reach the cloud M2M server. The VPN connections add layers of security on top of the oneM2M communication and abstract from complex network topology between the Smart IoT Gateway and the cloud infrastructure. Having reverse proxies at each end of the VPN connections gives the possibility to select the desired route, based on decisions taken by other components of the system (i.e., the rule engine).

However, although it worked, we faced several issues and difficulties to achieve the desired functionality [7] of the Smart IoT gateway (GW) by using the oneM2M standard, which led us to investigate alternative approaches. The decision was made to complement the oneM2M based solution with the industry standard M2M protocol MQTT. In the following sections, the overall scope, and requirements as well as the reasons behind this decision and the final design are described.

3.1 Introduction of MQTT in oneM2M cloud interface

There are two distinct use cases of our project scope, in which an M2M communication technology is to be used. The primary use case is the communication between the Smart IoT Gateway and the SES Cloud Server via different types of physical network routes. The secondary use case is the integration with external 3rd party systems, in which external software systems request data from the SES Cloud Server through a standardized application interface. In the upcoming sections, the focus will remain on the primary use case, however, all considerations for the final design also took the secondary use-case into account.

It should also be noted that in order to abstract from the physical nature of the individual network routes, all communication between the IoT gateway and the cloud server is tunneled through one of the three VPN tunnels, each of which is routed through its dedicated physical network interface, representing the Satellite, Terrestrial or ISP route.

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

3.1.1 Primary Requirements

We identified the following primary requirements, which must be met by the selected M2M communication technology, either directly or indirectly via workarounds:

- The selected M2M protocol shall be an accepted and well-defined industry standard protocol.
- It shall be robust and resilient to connection outages and interruptions during transmissions.
- It shall allow secure communication – e.g., by enabling TLS encryption.
- It shall be able to support link abstraction. The selection of the target route is done on the application layer. This selection and the nature of the network link must be transparent to the communication partners (IoT Gateway and Cloud server) and must not affect their connection.

3.1.2 Secondary Requirements

The following secondary requirements are considered nice-to-have and hence are optional:

- The selected technology should be easy to integrate into the system.
- The communication protocol should be lightweight with low overhead.
- The technology should allow for the straightforward implementation of bi-directional communication in the given scope.
- The technology should be flexible to changes in namespaces and should be able to handle discrepancies easily.

3.1.3 Eclipse OM2M

As already explained, the industry standard M2M communication framework oneM2M was selected for the M2M communication in the scope of the Smart IoT Gateway. Individual implementations of this framework specification are available, and the choice was made to use Eclipse OM2M, which is based on the Eclipse OSGi framework, written in Java. Since it uses HTTP as its communication protocol, it already complies with three of the four primary requirements – Industry standard; robustness and resilience due to TCP; secure TLS communication using HyperText Transfer Protocol over Secure Socket Layer (HTTPS). The last remaining requirement of link abstraction could be achieved by the specific architecture described below. Furthermore, it was fairly straightforward to integrate the system, since there are open-source node packages available for Node-RED, which only required minimal additional implementation to be useful in our system.

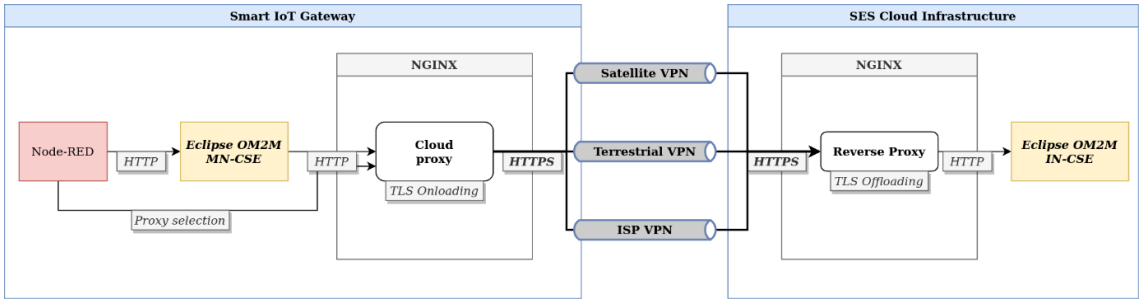

FIGURE 11: THE COMPONENT LAYOUT WITH ECLIPSE OM2M

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

The Eclipse OM2M architecture featured the core server component Infrastructure Node Common Service Entity (IN-CSE) in the cloud infrastructure and a middleware component MN-CSE on the Smart IoT Gateway itself. The first difficulty to overcome was enabling HTTPS communication between the MN-CSE and IN-CSE, because of a bug in the software. As a workaround solution, we used the already existing NGINX proxy on the gateway for TLS Onloading and the cloud-side NGINX for TLS-Offloading. This way, the CSEs could be configured to use plain HTTP, while the TLS encryption is added between the NGINX proxies only for the actual outgoing traffic.

The Link abstraction requirement could be met by using the NGINX JavaScript scripting language NJS to dynamically remap the proxy target of the gateway-side cloud proxy to a different (VPN) IP address. For this purpose, an internal POST endpoint was added that allowed the routing engine on Node-RED to explicitly set the proxy target IP, based on the selected route the traffic shall take.

While this setup works from a functional perspective, it comes with several downsides:

1. Very high complexity. Besides the need for a middleware component on the gateway, the design requires an outgoing cloud proxy instance as well as an additional scripting mechanism for route selection.
2. It is not thread-safe. A mutex (Single thread semaphore) had to be introduced to prevent multiple concurrent changes to the proxy IP, since this might interfere with the transmission which is currently in progress.
3. HTTP has a large protocol overhead.
4. Bi-directional communication is not straightforward to be implemented since it would require complex handling of gateway hostnames.
5. While HTTP is an industry standard protocol, oneM2M, which is a well-defined framework too, it is not very well received by the industry when compared to other technologies, especially in the IoT world:
    - The documentation of Eclipse OM2M (and other implementations) is inadequate.
    - The state of the Eclipse OM2M implementation is not mature.
    - Very low activity in the OM2M repositories and forums.
    - Not many applications and projects could be found using OM2M, making the research and development regarding OM2M difficult.

3.1.4 MQTT

We decided to use an MQTT-based implementation as It could offer similar functionality with oneM2M in a less complex and much more mature way:

- MQTT is simple and lightweight and with a smaller protocol overhead compared to HTTP.
- It features a simple publish/subscribe model and a fully dynamic topic hierarchy:
    - In contrast, oneM2M also allows for a publish/subscribe model, however, it is wrapped into the request/response model of HTTP.
- Changes in the topic namespace of MQTT do not require explicit creation of the whole topic hierarchy as is the case with oneM2M.

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

- Since MQTT (+ TLS) can simply be used via IP addresses, there is no need for an intermediate Cloud proxy. The three routes to the same cloud broker can be held in parallel using individual client IDs.
- No middleware is required on the gateway itself.
- Bi-directional communication can easily be achieved by subscribing to dedicated topics for downstream commands towards the Smart IoT Gateway.

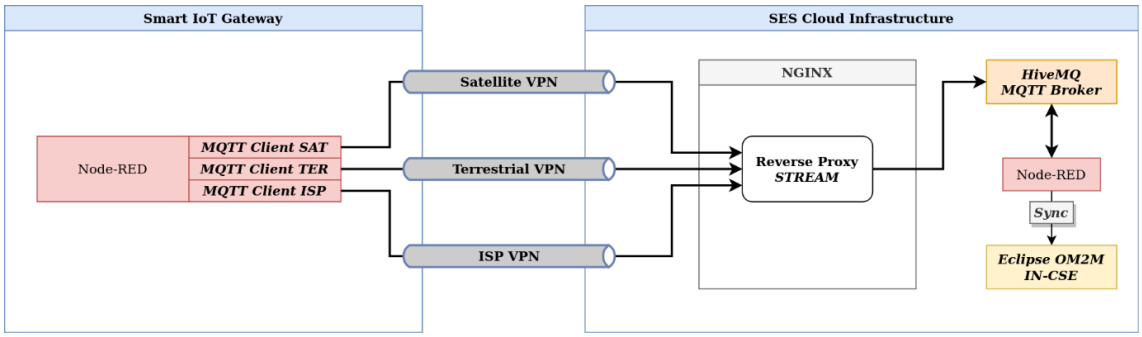

FIGURE 12: THE COMPONENT LAYOUT WITH MQTT

The finally agreed design features a HiveMQ MQTT broker on the cloud side and MQTT client nodes available for Node-RED on the gateway side. Each gateway uses three clients with individual client IDs to allow multiple consecutive connections to the same broker from the same gateway via the three routes. The cloud-side NGINX reverse proxy uses streaming endpoints to perform the proxying task. Since the dynamic proxy on the gateway is not required anymore, there is also no need for the mutex control, allowing multiple concurrent transmissions to be performed without interference.

Since the integration with external 3rd party systems as part of the secondary use case still requires a oneM2M interface, we decided to preserve the IN-CSE instance in the cloud and introduce a synchronization mechanism, which subscribes to the MQTT broker and publishes the received data to the OM2M platform. This allows external partners to select the desired interface based on their preferences. Connecting directly to the MQTT broker provides access to real-time data, while the data in the OM2M platform may only be near-real-time since the synchronization is not guaranteed to be instantaneous.

3.2 Relation to the Use Cases

The Smart IoT GW was mainly used in the Ship use case. For this use case, the Smart IoT GW provides a single unified MQTT broker, to which all types of sensors must connect in order to have their data processed. In addition, the Smart IoT Gateway interfaces with the M2M cloud space through a local MN-CSE instance (Middleware Node Common Service Entity) implemented as *Eclipse OM2M* in compliance with the oneM2M standard. Requirements and KPIs achieved, are reported and discussed in D6.3 [12] accordingly.

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

3.3 Innovation

The Smart IoT Gateway ensures connectivity for a vast number of heterogeneous IoT devices, by harmonizing different IoT technologies and application protocols and formatting data to be transferred across the network, either terrestrial or satellite. Therefore, IoT interoperability enables the federation of different IoT platforms within heterogeneous domains, overcoming the compatibility issues between both standard and non-standard, proprietary and custom M2M solutions. Furthermore, it enables the better exploitation of data in optimization and prediction, which provides the greatest business value. In Figure 13, you can see the interaction of the Smart IoT GW in the iNGENIOUS architecture.

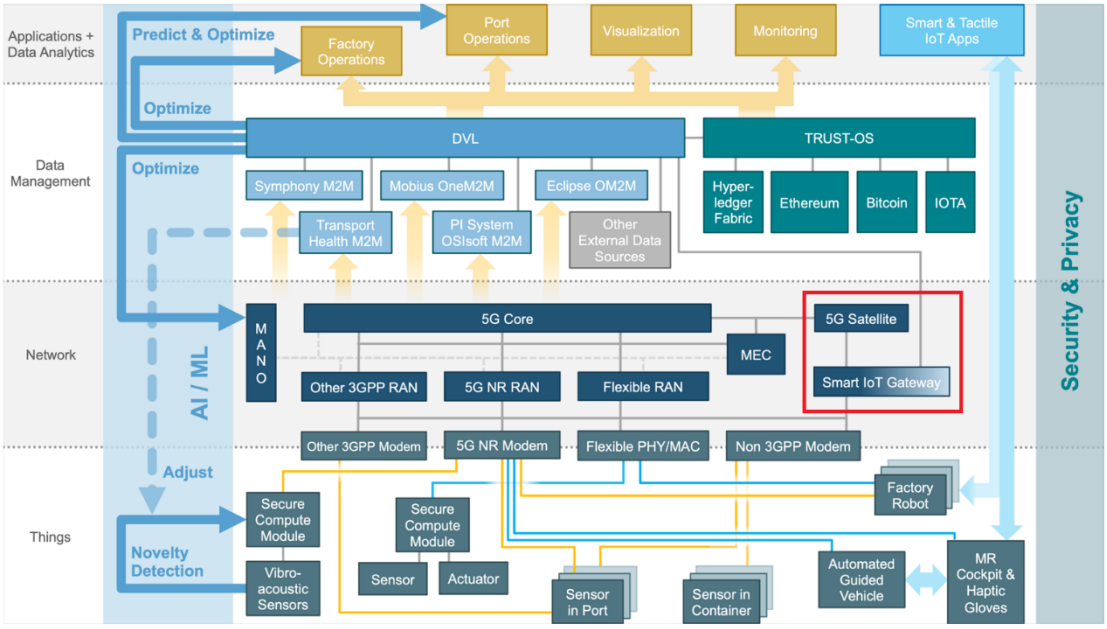

FIGURE 13: INTERACTION OF THE SMART IOT GATEWAY WITH THE OTHER COMPONENTS OF THE INGENIOUS ARCHITECTURE

3.4 Business Potential and Exploitation

Component Group: Smart IoT Gateway

- **SES**: The Smart IoT GW can be used in several commercially attractive mMTC use cases and for multiple market verticals, such as Fixed Data, Aero, Maritime, Energy, Government, Agriculture, Cloud, and Video.
  For example, the Smart IoT Gateway can play an important role in solutions of disaster response where we need to re-establish communication (Internet, phone) after a disaster, to support the coordination efforts of humanitarian organisations in the field and contribute to saving lives during humanitarian emergencies. The Smart IoT GW can help in asset monitoring for equipment/sites. The edge-computing capabilities of the Smart IoT GW can be used to monitor equipment which is deployed on the sites and can provide an alerting interface for the on-site personnel by sending notifications to the

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

smartphones or other smart devices, in case there are issues with monitored devices.
Furthermore, the Smart IoT GW can provide useful services to agriculture. The vines are especially exposed to the effects of climate change and to the spread of diseases. Providing the winegrowers with near-real time information about their vineyards will offer them the possibility to adapt their management according to their needs. The availability of this information allows them to a more time and cost-efficient planning. The data from the sensors deployed in the vineyards can be processed by the Smart IoT Gateway that supports several IoT communications technologies and protocols, and intelligently routes the data to satellite or terrestrial backend networks.

Further details about the innovation, exploitation and impacts are reported in D7.3 [9] as well as in D2.5 [10].

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

4 Data Virtualization Layer

Data Virtualization Layer (DVL) is part of the interoperability layer, together with TrustOS (presented in Chapter 5), as described in D2.4 [2].

In D5.2 [7], the Data Virtualization Layer (DVL) component has been described in terms of its native functionalities as well as in terms of the role it plays in the iNGENIOUS project (as a mean of federation of the M2M platforms and data sources). In this respect, we identified and described the main M2M platforms and data sources which are currently used to feed DVL by providing specific data sets needed for the definition of maritime events, based on a data model that has been previously defined. Such events were also described and linked to the use cases of the iNGENIOUS project, and a preliminary integration approach with the underlying data sources was presented. Moreover, a set of applications (maritime events consumers) were also identified on top of the DVL component. These applications are able to interact with DVL by retrieving data included within a given maritime event, according to their role in the iNGENIOUS project (e.g., data pseudonymization, data analytics, data visualization, and data integrity).

In this chapter, the integration with the above-mentioned data sources is described and linked to the iNGENIOUS use cases accordingly.

4.1 Integration activities

As described in D5.2 [7], the DVL component is integrated with different M2M platforms and external data sources of the iNGENIOUS project in order to gather, aggregate, and expose data to be used by different data consumers as depicted in the Figure 14 below (for further details see D2.4 [2]):

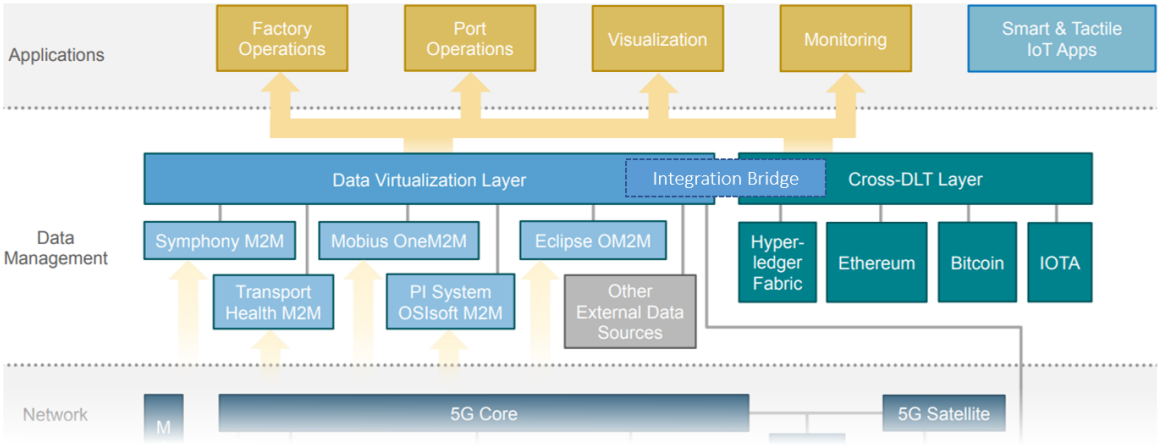

FIGURE 14: INGENIOUS COMPONENTS OF THE INTEROPERABILITY LAYER

Here we describe how this integration was performed for each M2M platform as well as the way data are aggregated and exposed through developed interfaces. Further technical details are included in the final version of the deliverable D6.2 [8].

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

4.1.1 Mobius OneM2M Platform

The Mobius OneM2M platform has been deployed in the Port of Livorno, and it is used in the context of the Port Entrance use case by collecting meteorological data within the seaport area. From the DVL perspective, this integration activity considers the following aspects:

- **Data Source**: Mobius OneM2M platform;
- **Data Consumer**: Awake.AI platform.

In order to expose meteorological data, two interfaces have been implemented and integrated with DVL: MeteoData1() and MeteoData2() which are referred to two different sensors installed within the seaport area for meteorological conditions monitoring. The access to the main interfaces is supported by Java Database Connectivity (JDBC) protocol and it requires a specific library which implements this protocol (on back-end side). Further technical details of this platform are included in *D5.2* [7].

4.1.2 Symphony M2M Platform

The Symphony M2M platform has been enhanced and deployed in the Port of Livorno, and it is used in the context of Port Entrance and DVL/DLT use cases. From the DVL perspective, this integration activity considers the following aspects:

- **Data Source**: Symphony M2M platform;
- **Data Consumer**: Web Application for the visualization.

This platform is used to integrate the IoT tracking sensors onboard trucks within the port area. The aim is to collect and store positioning data and events from the IoT tracking devices and expose them to the DVL. Finally, a RESTful interface implemented at the DVL level allows a web application to consume such data by providing a graphical user interface for visualization. To support the integration with the Micktrack MT821 IoT tracking sensors deployed in the Port of Livorno, a set of enhancements to the baseline Symphony platform (and specifically to the hardware abstraction layer module) have been developed in iNGENIOUS.

First, a dedicated southbound plugin for proper hardware abstraction has been implemented, allowing to interface with the IoT tracking sensors' custom communication protocol and thus be able to collect the related data. Moreover, for the exposure of the collected data towards the DVL, the selected approach for the Hardware Abstraction Layer (HAL) northbound plugin is based on a Message Queuing Telemetry Transport (MQTT) interface. This approach allows seamless integration with the native Symphony Data Storage and therefore storing the data collected from the IoT tracking sensors through the use of a message bus based on RabbitMQ. This approach is motivated by the fact that the DVL does not store itself any data, therefore the iNGENIOUS Symphony deployment was required to provide data storage functionalities, that have been integrated with the message bus through an Advanced Message Queuing Protocol (AMQP) connector. On top of this data

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

storage, dedicated RESTful APIs have been implemented to expose data towards the DVL. This approach allows the DVL to retrieve historical data with filtering options related to a specific time window. This API exposes three main GET operations, as follows:

- Method 1: retrieves the full list of available digital object time-series within the Symphony Data Storage. This operation returns information related to the total number of available time-series;
- Method 2: retrieves the time-series data for the digital object identified with a given identifier. The returned JSON message includes the full list of data collected by Symphony;
- Method 3: retrieves the time-series data for the digital object identified with a given identifier, limited to a time window.

Further technical details about the Symphony M2M platform enhancement, APIs and data model can be found in D5.2 [7].

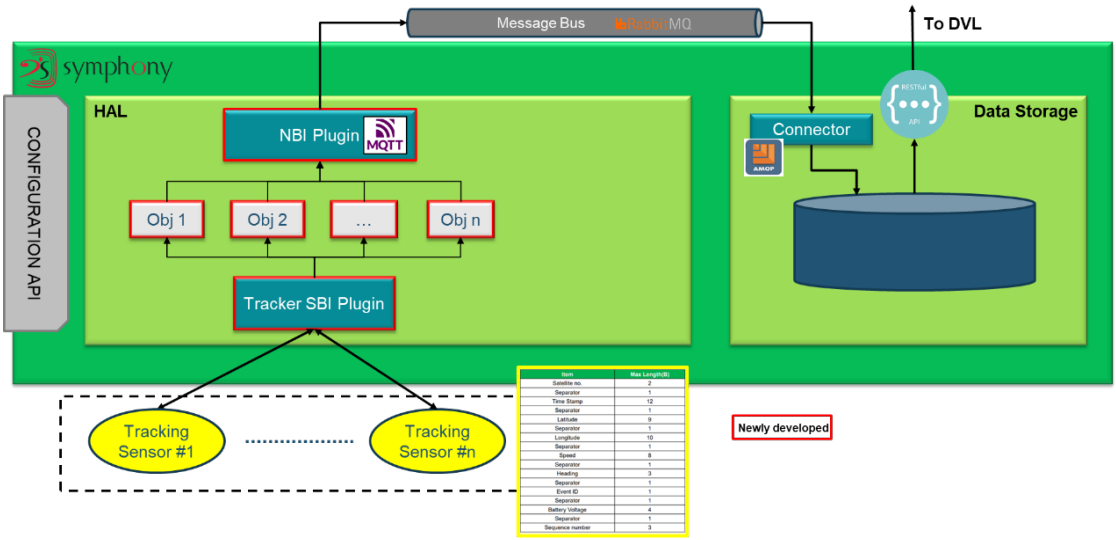

FIGURE 15: OVERVIEW OF SYMPHONY HAL ENHANCEMENTS.

Finally, the DVL implements a RESTful connector/wrapper for the interaction with Symphony M2M platform DBMS so that the stored data can be consumed through a RESTful interface (named FetchTracker), as described in D6.2 [8].

4.1.3 Eclipse OM2M Platform

The Eclipse OM2M platform is used in the context of Ship and DVL/DLT use cases. From the DVL perspective, this integration activity considers the following aspects:

- **Data Source**: Eclipse OM2M platform.
- **Data Consumer**: TrustOS.

Eclipse OM2M Platform is part of the Smart IoT Gateway component, as described in Chapter 3 of this deliverable. In the context of the Ship use case, smart IoT devices are installed on the container loaded on the COSCO ship for real-time monitoring purposes. One of these sensors is used for the

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

container's door status monitoring. All data are published and stored on the Eclipse OM2M platform (M2M Cloud Space) by means of the Smart IoT Gateway. External components (DVL in this case) which need to gather data from the M2M cloud space, are able to connect to the IN-CSE through a private IP endpoint, accessible via a secured VPN tunnel. The sensor data in the IN-CSE are organized according to a given resource tree and several subtrees. All subtrees include several content-instances with a given set of attributes, as described in D6.2 [8]. In order to support the *sealRemoved* event, which occurs when the IoT sensor is removed from the iNGENIOUS container, the following data model was adopted and implemented at DVL level:

Attribute	Type	Source
originatorName	Static	Port of Valencia
originatorId	Static	VLC
equipmentNumber	Static	ZIMU1381282
eventOccurrenceTime 8601	Dynamic	SENSOR_DATA/signal_state/.../ct
smdgTerminal	Static	CSP Iberian Valencia Terminal
latitude	Dynamic	SENSOR_DATA/GPS/.../con/latitude
longitude	Dynamic	SENSOR_DATA/GPS/.../con/longitude
sealType	Static	Carrier
sealNumber	Dynamic	CONTAINER_INFO/.../con/dev_eui
signalState	Dynamic	SENSOR_DATA/signal_state/.../con/signal_state

FIGURE 16: DATA MODEL FOR THE SEALREMOVED EVENT.

Finally, a RESTful interface was implemented (named sealRemoved(), as described in D6.2 [8]) and it allows to expose such data externally through the DVL component.

4.1.4 PISystem M2M Platform

The PISystem OSIsoft M2M platform is used in the context of DVL/DLT and Port Entrance use cases. From the DVL perspective, this integration activity considers the following aspects:

- **Data Source**: PISystem M2M platform and its modules.
- **Data Consumer**: TrustOS.

The M2M platform is deployed in a staging environment in Valencia seaport. The platform allows to collect and store data for the GateIn and GateOut events related to vehicles' transit in Valencia. The extraction of GateIn and GateOut data from PISystem is performed through the PIWeb API/SDK interface, which provides external systems or client applications reading and writing access to their PI Asset Framework and PI Data Archive over the HTTPS protocol. This interface is integrated with DVL and an additional RESTful interface is implemented allowing TrustOS to retrieve GateIn and GateOut events from the DVL accordingly. Further technical details are described both in D6.2 [8] and D5.2 [7].

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

4.1.5 Tuscan Port Community system (TPCS)

The Tuscan Port Community System (TPCS) is used in the context of DVL/DLT and Port Entrance use cases. From the DVL perspective, this integration activity considers the following aspects:

- **Data Source**: TPCS;
- **Data Consumer**: TrustOS, pseudonymization function (PF) module and Awake.AI platform.

The TPCS is the main Port Community System used in Livorno seaport for GateIn, GateOut, VesselArrival and VesselDeparture data provisioning. An instance of this platform is deployed in a staging environment in Livorno seaport. The DVL implements a RESTful connector to interact with its underlying DBMS and aggregates data needed for the definition of before-mentioned events as per the defined data model (refer to D6.2 [8] and to D5.2 [7]). Moreover, a RESTful interface was implemented at the DVL level so that such events can be exposed and consumed by different data consumers as described in Figure 17:

Interface	Data Consumer
LatestGateInEvent()	TrustOS
LatestGateOutEvent()	TrustOS
HistoricalGateInEvent()	PF module
HistoricalGateOutEvent()	PF module
LatestVesselArrivalEvent()	TrustOS,
LatestVesselDepartureEvent()	TrustOS,
HistoricalVesselDepartureEvent()	Awake.AI platform
HistoricalVesselArrivalEvent()	Awake.AI platform
fetchGateEvents()	Awake.AI platform

FIGURE 17: MAPPING BETWEEN THE INTERFACES AND DATA CONSUMERS BASED ON THE INTEGRATION BETWEEN DVL AND TPCS.

The above mentioned interfaces are remotely accessible via RESTful paradigm based on OData protocol (an open protocol to allow the creation and consumption of queryable and interoperable RESTful APIs).

4.2 PF module implementation

The Pseudonymization Function (PF) module, integrated into the DVL as shown in Figure 18, allows to pseudonymize the personal data handled by the data management layer, generating pseudonyms to be used instead of personal data. In this way, applications that need to use data from M2M platforms will not have direct access to sensitive data.

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

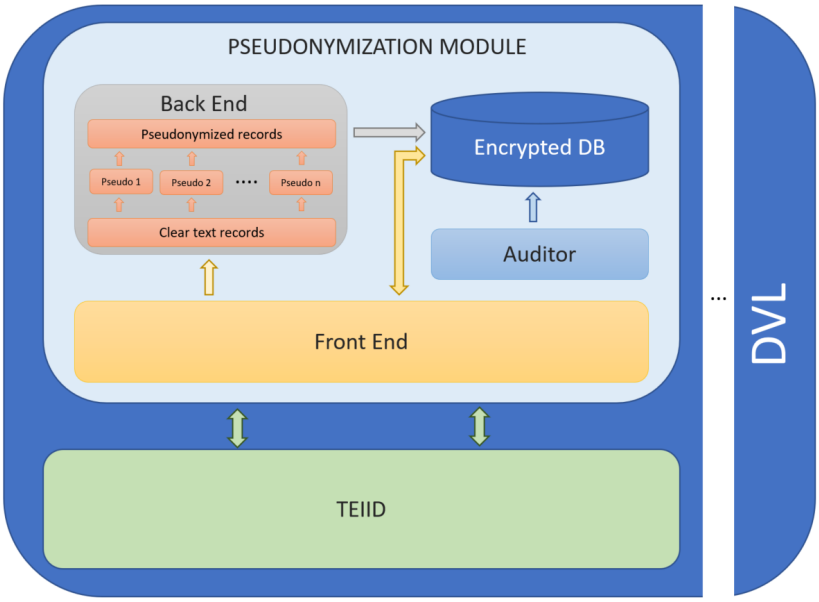

FIGURE 18: PSEUDONYMIZATION FUNCTION MODULE

The deliverable D5.3 finalizes the PF module introduced in D5.2 [7], introducing a new pseudonymization technique, an auditor function and a delete data on request function.

4.2.1 Front End

The new Rest API *DELETEDATA* has been added to the already delivered APIs:

- configTemplate, used to set the pseudonymization technique to apply.
- HistoricalGateInEvent, used by PF to retrieve historical GateIn events
- HistoricalGateOutEvent, used by PF to retrieve historical GateOut events
- FetchGateEvents: used by DVL to retrieve historical pseudonymized GateIn/GateOut events
- deleteData used by DVL to delete pseudonyms

The *deleteData* API permits deletion of the pseudonym of a specific vehicle (the only personal data handled by the DVL is the truck plate, aka vehicle) from the encrypted DB handled by the PF module. In this way it will be no more possible to retrieve the vehicle in clear text format from its pseudonym:

https://xx.xx.xx.xx:8081/deleteData/<pseudonym>

4.2.2 Back End

The new pseudonymization technique *HASHING WITH KEY* has been added to the already delivered ones:

- Format Preserving Encryption (FPE)
- Blurring
- Hashing without key (HWK)

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

- Hashing with key

The new technique permits to increase the randomness of the generated pseudonyms.

4.2.3 Auditor

According to GDPR, it is important to keep personal data for a limited time (retention period). The PF introduced the Auditor function that every night removes (clearing field) all the pseudonyms in each pseudonymized event in which the retention period has expired. The default retention period is set to 5 years, but it can be modified at instantiation.

4.3 Relation to the Use Cases

The DVL is part of the DVL/DLT, Port Entrance and Ship use cases and it is used in four different scenarios, as already described in D2.4 [2]:

- **Scenario 1:** This scenario is linked to the DVL/DLT use case where both Port of Livorno and Port of Valencia are involved. Here, the DVL is responsible for providing GateIn, GateOut, VesselArrival and VesselDeparture events for both seaports. Such events are retrieved both from the Tuscan Port Community System (TPCS) and PISystem M2M platform.

- **Scenario 2:** This scenario is linked to the Ship use case where the Port of Valencia is involved. The DVL is responsible here for interacting with Eclipse OM2M platform in order to retrieve data coming from the smart sensors installed on the transported container.

- **Scenario 3:** This scenario is linked to Port Entrance use case where both Port of Valencia and Port of Livorno are involved. The DVL is responsible for interacting with the Symphony M2M platform in order to retrieve GPS data coming from tracking devices installed on trucks in the Livorno seaport.

- **Scenario 4**: This scenario is also related to Port Entrance use case implemented in both Port of Valencia and Port of Livorno. Two different components (integrated with DVL) are involved: Mobius OneM2M platform and the Pseudonymisation Function module. The M2M platform is responsible for collecting meteorological data at the Livorno seaport. Then, a Pseudonymization Module component is used to process GateIn and GateOut events in order to identify personal data and pseudonymize it accordingly (trucks' plate numbers).

Requirements and KPIs achieved, are reported and discussed in D6.3 [12] accordingly.

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

4.4 Innovation

The DVL is used as an intermediate component between the underlying data sources (e.g., M2M platforms) and applications (e.g., TrustOS, Awake.AI, etc.) by abstracting their complexity and by providing an effective tool for data access and aggregation. The most innovative aspects of such approach are listed below accordingly:

- Abstraction of the complexity of the underlying data sources (e.g., M2M platforms and PCS) by implementing a common layer for data access and aggregation among heterogeneous platforms.
- In addition, the Pseudonymization Function allows to extend DVL's capabilities by providing a pseudonymization functionality for personal data, as described in D2.4 [2].

4.5 Business Potential and Exploitation

Component Group: Data Virtualization Layer

- **CNIT**: DVL goes beyond the capabilities of traditional data integration approaches based on extract-transform-load (ETL) paradigm or data warehouse software. It enables data consumers to access all data through a single view in real-time instead of moving a huge amount of information which may lead to further data redundancies. It does not replicate data itself or store it anywhere, and it does not require developers to approach the underlying Database Management System (DBMS). For all these reasons, DVL can bring the following benefits:
  - ***Efficiency***: it enhances the utilization of server and storage resources (no data replication);
  - ***Time-to-solution acceleration***: it allows spending less time in development and integration activities if compared to standard approaches. It is easy to use and it enables doing more in less time;
  - ***Scalability***: it provisions lightweight database copies in minutes through a user interface or API enabling to scale an agile development;
  - ***Profitability***: it provides seamless access to data, enabling companies to make informed and profitable decisions.

  Within the iNGENIOUS project, the DVL solution has been also enhanced by integrating data pseudonymization techniques to meet data privacy requirements according to the EU regulations (namely GDPR). This makes the DVL an efficient tool in big data and predictive analytics scenarios where datasets come from heterogeneous data sources and include personal/confidential data. In this sense, the DVL makes it easier for a data consumer to access diverse datasets from a single platform and use it to perform analytics. In iNGENIOUS project, the use of this tool was limited to the maritime context (e.g., DVL/DLT, Port Entrance and Ship use cases) but it can be also applied in other IoT verticals such as Smart Cities, Smart Agriculture, Industrial IoT, Transport & Logistics and other Smart Systems.

  Further details about the innovation, exploitation and impacts are reported in D7.3 [9] as well as in D2.5 [10].

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

5 Distributed Ledger Technologies Layer

In the deliverable D5.2 [7], we described APIs' development and integration activities between different sets of available DLTs and the Cross-DLT layer. Initially, the main fundamentals of Distributed Ledger Technology were detailed, allowing us to have a clear context of the value that this technology brings to the projects in which it is included [3]. The main characteristics of the technology were explained, differentiating between two types of DLTs: Blockchain and Direct Acyclic Graph (DAG). Reference was also made to the different types of DLT systems depending on the access rights perspective. To conclude the introduction to DLT, the most used consensus mechanisms today (PoW and PoS) were explained, emphasising the differences that mark the use of one and the other.

The main challenge for the Cross-DLT Layer to overcome is the ability to store the events (what we have called Trustpoint) coming from the Data Virtualization Layer in any of the different DLTs regardless of the technology they use.

The use case is about interoperability between DLT solutions. The main aim of this use case is to provide an interoperable layer in order to abstract the complexity of the underlying DLT solutions, guaranteeing at the same time data privacy and security by means of encoding and anonymization techniques. The additional benefit of using multiple DLT solutions is increased level of immutability and accountability – if in unlikely event one of the DLTs fails, the proof will be available on the other.

In addition, it is necessary to provide some user interface through which the events and related Trustpoints that have been stored in the different DLTs can be easily visualised and verified by end users.

5.1 Integration activities

In this context, TrustOS [13] acts as the orchestrator of the Cross-DLT Layer as it receives the information coming from the DVL, stores it in the form of assets and generates Trustpoints based on these assets to send them to the different DLTs.

This layer involves different integration activities to meet all requirements (for further requirements details see D6.3 [12]). At a high level, 3 different lines of activities can be distinguished.

- DLT Common API: Common interface to be implemented by the different DLT connectors in order to receive Trustpoint registration requests from TrustOS.
- DVL Integration Bridge: Intermediate component to enable communication between two REST APIs.
- Graphical User Interface: Graphical interface that allows the end user to visualize and verify the events and their Trustpoints.

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

5.1.1 DLT Common API, an interface for accessing DLTs within TrustOS (Bitcoin, IOTA and Ethereum).

In order to have a common way of calling APIs regardless of the DLT technology, a common API has been defined to fulfil all the necessary requirements to store the Trustpoints in any Distributed Ledger.

The different connectors implemented for the Cross-DLT layer implement this API, and then the TrustOS platform will act as a bridge and distribute the information of the Trustpoints among the different ledgers making a unique request to the DLT connectors compliant with the common API.

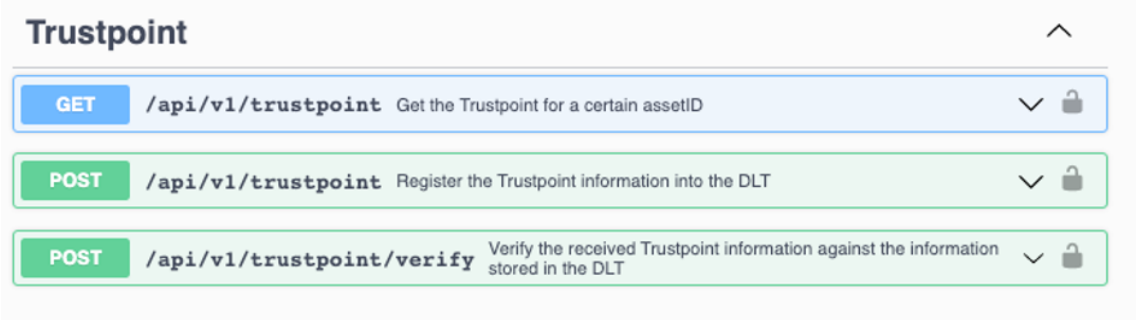

FIGURE 19: THE DLT COMMON INTERFACE FUNCTIONS

Initially, TrustOS offered a native connection to Ethereum. To extend the potential of TrustOS, work has been done on the integration of the Polygon network and it is now possible to register Trustpoints there as well, thus extending the list of DLT connectors. In short, TrustOS offers native integration with all Ethereum-based platforms and others of a different nature.

- Ethereum (deployed in Telefonica facilities).
- Polygon (deployed in Telefonica facilities)
- IOTA (deployed in CNIT facilities).
- Bitcoin (deployed in PJATK facilities).
- Hyperledger Fabric (deployed in FV and Telefonica facilities).

By providing context, Polygon is a decentralized Ethereum scaling platform with increased scalability, performance and lower fees. It is full Ethereum Virtual Machine (EVM) compatibility which means it is possible to deploy smart contracts directly on the Polygon chain.

For the management of TrustPoints or evidence, TrustOS offers three functions.

- Create an asset Trustpoint in a public Network. This function is in charge of making the corresponding request to the DLT Common API to register a Trustpoint in the corresponding DLT.
- Get a specific asset Trustpoint from a public Network. This function is in charge of making the corresponding request to the DLT Common API to retrieve the information of a Trustpoint in the corresponding DLT.
- Get all asset Trustpoints from public Networks. Initially will not make requests to the DLT common API.

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

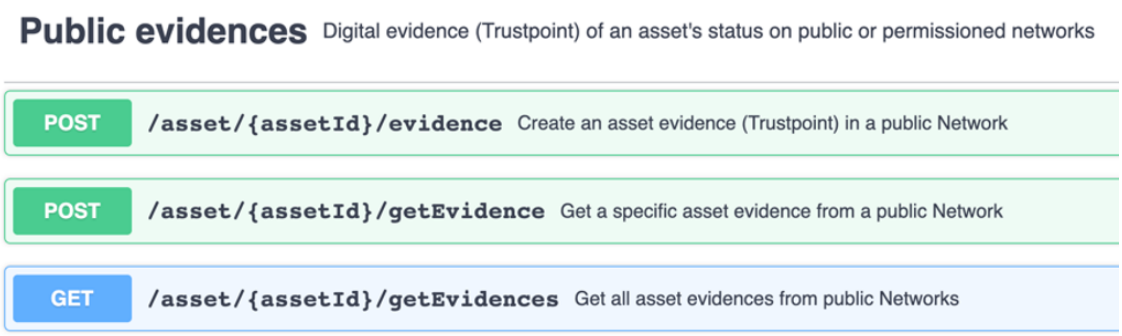

FIGURE 20: TRUSTOS FUNCTIONS RELATED TO PUBLIC EVIDENCE OF TRUSTPOINTS

The integration that has been done in TrustOS to include the different DLTs used in the project is shown below.

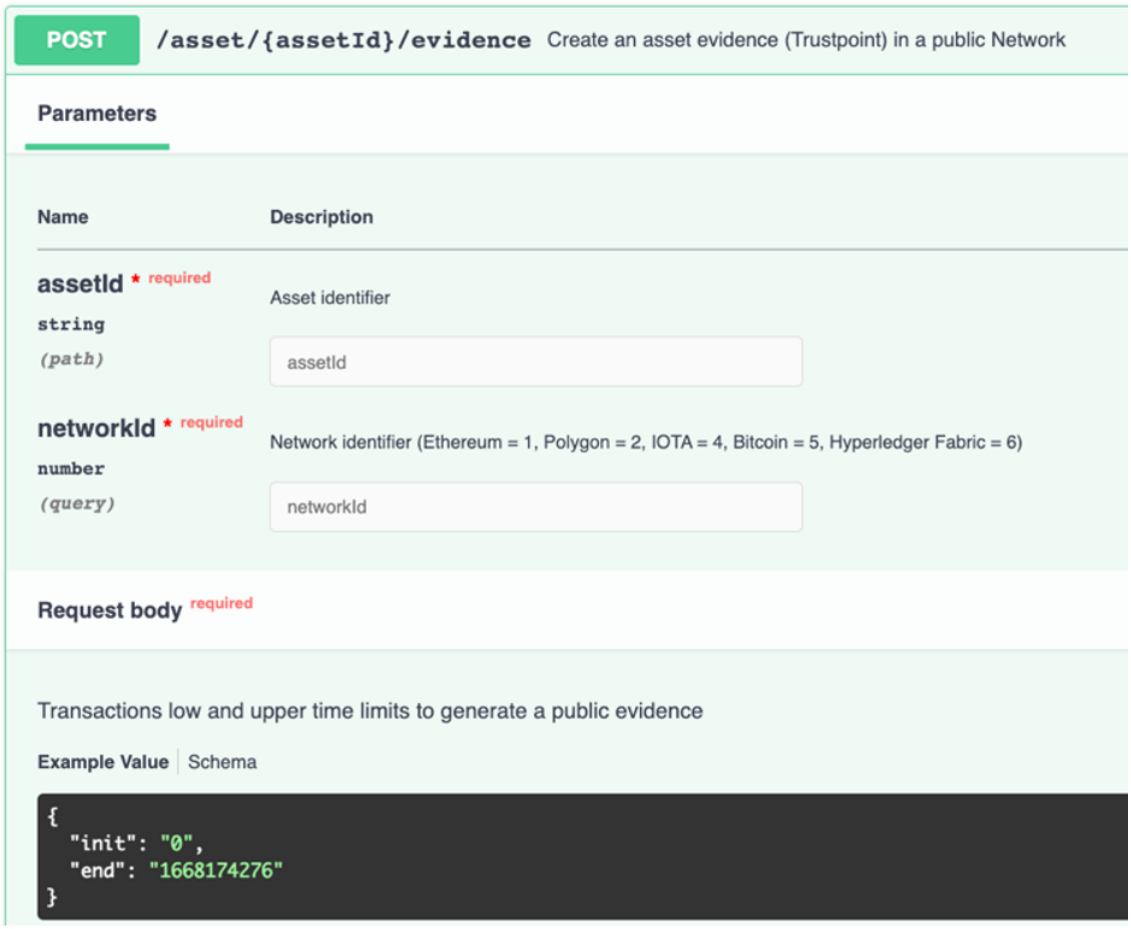

FIGURE 21: TRUSTOS FUNCTION TO REGISTER A TRUSTPOINT ON A SPECIFIC DLT

As shown in the image above, it is only necessary to specify the identifier of the event asset and the identifier of the DLT in which the Trustpoint will be stored. This request triggers another request to the DLT common API with the information corresponding to the Trustpoints for registration.

The following is the record of a Trustpoint generated from a "Vessel Arrival Event" from the DVL.

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

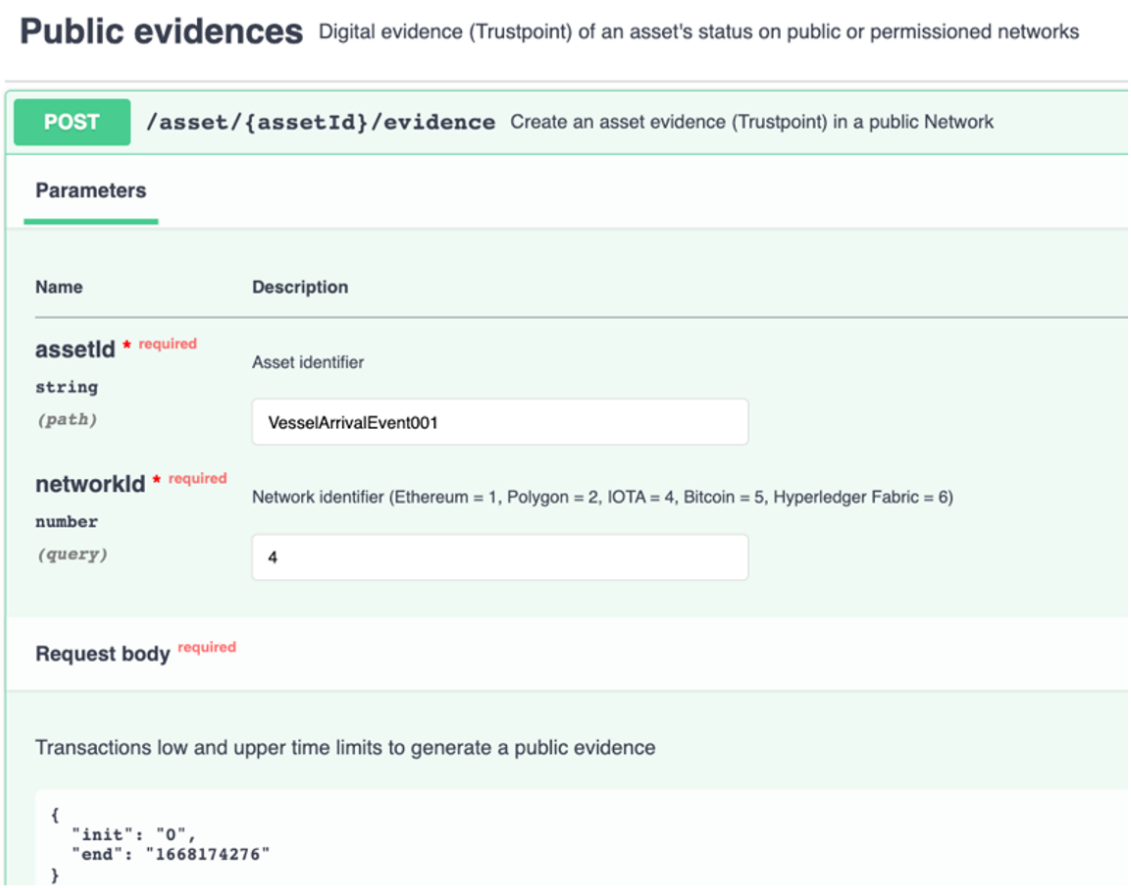

FIGURE 22: TRUSTPOINT STORAGE ON IOTA DLT

As shown below the registration is successful after calling the corresponding IOTA endpoint.

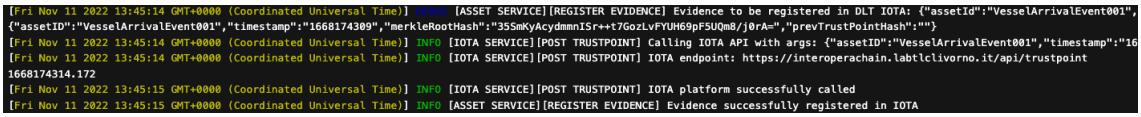

FIGURE 23: TRUSTOS LOG TO TEST THE SUCCESS OF TRUSTPOINT REGISTRATION

For the rest of the DLTs the process is completely analogous.

5.1.2 Integration Bridge

In the phase of analysis of the automation of the communication between the DVL and the Cross-DLT Layer, a problem to be solved was detected. TrustOS and DVL are passive elements, specifically REST APIs. Therefore, we need a mechanism that queries data in the DVL and stores it in TrustOS in an active way. This is the main reason for the implementation of the Integration Bridge. The three main goals of the bridge are:

- Act as an intermediate component between DVL and TrustOS.
- Use a long polling mechanism because it is one of the simplest ways of getting new information from the server with a certain frequency.
- Model the event information coming from the DVL in the structure that TrustOS understands. TrustOS offers complete flexibility to represent any type of information, which is an advantage for the future if, for example, other types of events need to be stored.

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

The bridge asks the DVL frequently and if there are new events. If the information belongs to a new event, this event asset is created in TrustOS. If any information has already been registered for that event and it is new, an update is made on the asset.

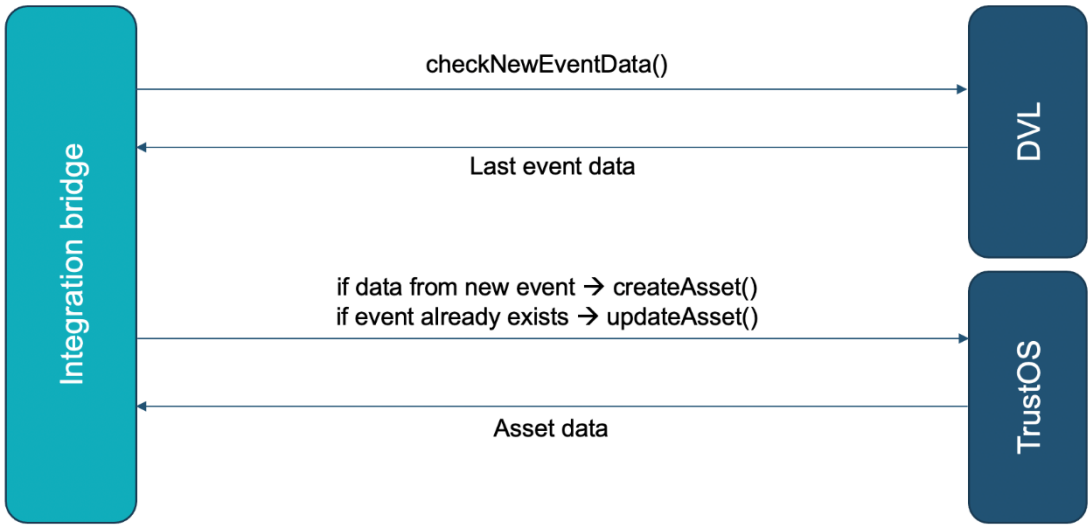

FIGURE 24: INTEGRATION BRIDGE FLOW DIAGRAM

The Integration Bridge has been implemented as a microservice hosted in Telefonica facilities and it is already working constantly checking for new data for the defined events.

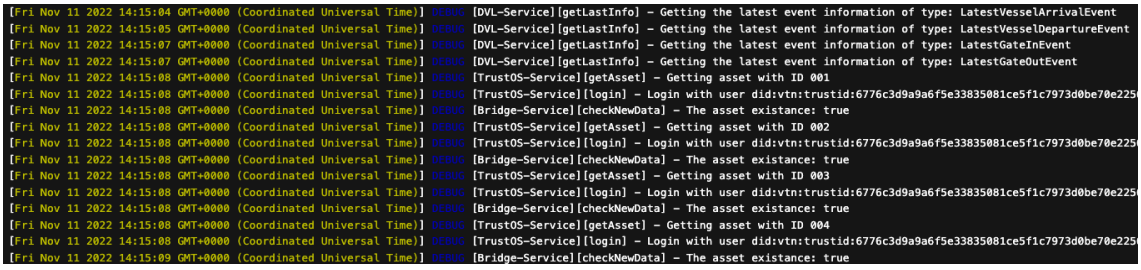

FIGURE 25: INTEGRATION BRIDGE LOG FOR CHECKING NEW DATA FROM DVL

5.2 Development of GUI for end-users to interact with TrustOS and a given DLT

In order to show in a simple and visual way the information stored in TrustOS and in the different DLTs, it is necessary to implement a Graphical User Interface (GUI). Through this interface, the end user will be able to visualize the different events recorded, as well as to verify the Trustpoints stored in each of the corresponding DLTs. Thus, the GUI provides an easy way to verify the information stored in the DLTs. Access to information (except for public information such as Trustpoints) is role-based.

At the technical level, this interface will be implemented through Front-End tools. Integration with the core element of the Cross-DLT Layer, TrustOS, is obviously necessary. In this sense, the interface must communicate with TrustOS to obtain the information on both the assets and the Trustpoints

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

stored in each of the DLTs. To achieve this goal, the interface must make the corresponding requests to TrustOS to obtain the desired information. We will call this interface DLT Events Visualizer.

5.2.1 DLT Events Visualizer design aspects

In terms of design, two different views are proposed:

- Asset view: Displays asset information representing the different events stored in TrustOS such as vessel arrival, vessel departure, gate-in, and gate-out. The objective is to show the asset identifier, its permanent information and the different transactions (status updates) it is experiencing.
- Trustpoint view: Displays information of a Trustpoint stored in a specific DLT. To verify its validity, both the information stored in TrustOS and the information stored in the corresponding DLT are shown, as both must coincide.

A high-level schematic for the DLT Events Visualizer is shown below.

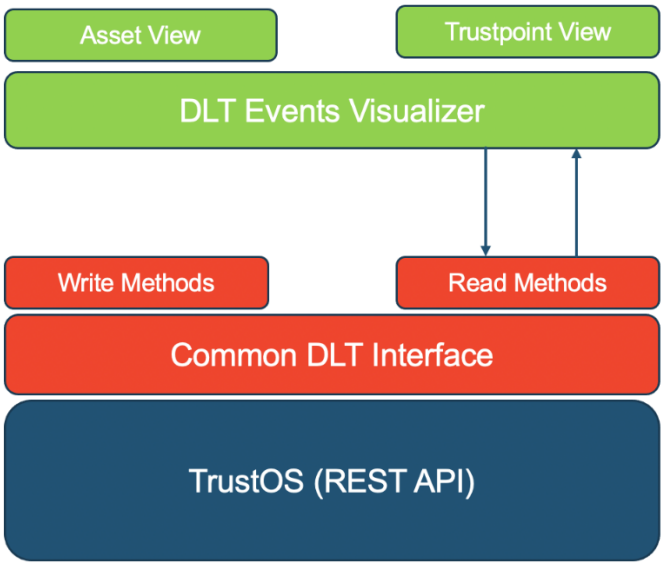

FIGURE 26: HIGH LEVEL DIAGRAM OF THE DLT EVENTS VISUALIZER

As an initial approximation, the basic functions that the GUI should include are:

- Get event current status
- Get all event history
- Get event Trustpoint
- Verify event Trustpoint

The DLT Events Visualizer is currently in the design phase. As a starting point (take this as a simple sketch), the asset view should look like Figure 27.

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

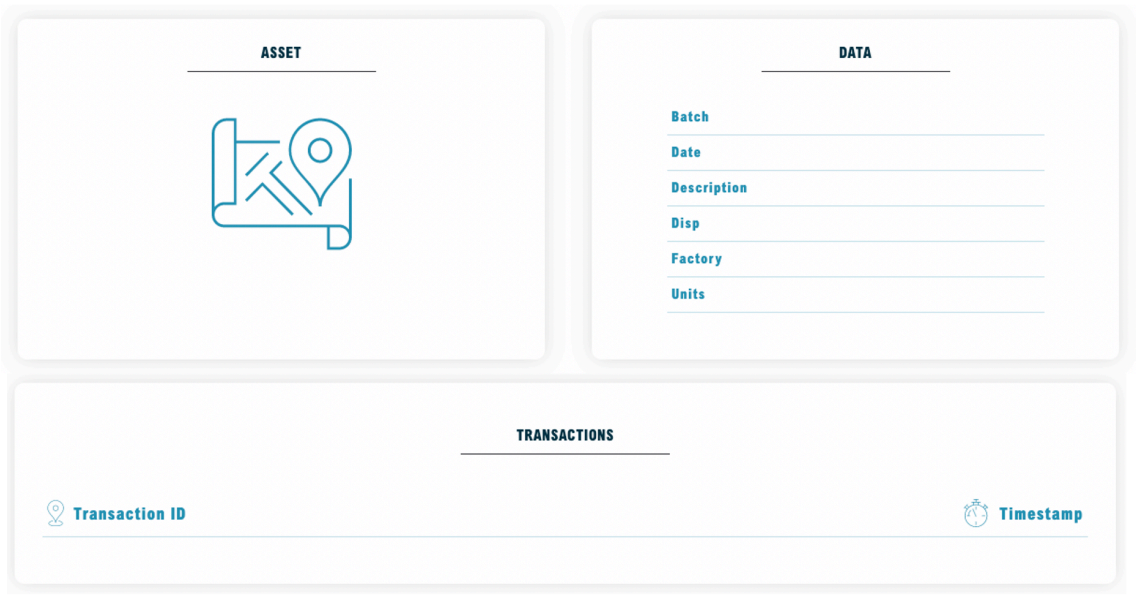

FIGURE 27: SKETCH PROPOSAL FOR THE GUI ASSET VIEW

This mock-up was used for a proof of concept where it was necessary to show details about the information of the assets stored in DLT networks. To generate an interface with a good user experience, Telefónica carried out an analysis to apply the most common UX design methods. Specifically, methodologies such as a study of the value proposition, brainstorming, user interviews and storytelling, stakeholders' interviews, and usability testing were used.

5.3 Relation to the Use Cases

Cross-DLT Layer enables the exchange of data between private and public DLTs. In the DVL/DLT use case, the traceability, transparency and interoperability are necessary requirements. Cross-DLT Solution Layer allows to store the original data and events collected from the M2M layer and then made available to authorized actors using the desired DLT share and store the information in a secure manner. One of the main advantages of this solution is the ability to store events in different DLTs on demand. In other words, events can be recorded in real-time (considering the information validation times of the public platforms) in one or several DLTs depending on the specific needs of each one of them. This makes the solution more flexible and appeals to users who prefer some DLT networks over others. Besides, public DLTs, like Bitcoin, Ethereum and so on, allow for international cooperation even if different nation states don't trust each other. The information can't be practically erased/altered, so the proof is very solid.

The DLT layer includes the Integration Bridge that enables the flow of event information once captured and processed by the DVL to TrustOS in an automated way, so that whenever a new event is generated, it will be automatically registered both in TrustOS and in the corresponding DLT.

In addition, thanks to the graphical user interface called DLT Events Visualizer, all the information corresponding to the different events can be displayed. Thus, the solution provides a simple and effective way of verifying the entire lifecycle of events by the different roles.

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

In addition, the underlying infrastructure allows for high availability of information as well as good performance and processing of concurrent requests to the different networks, always maintaining the confidentiality of personal data. Even if Public, DLTs record only the evidence of the data, not the data itself.

Requirements and KPIs achieved, are reported and discussed in D6.3 [12] accordingly.

5.4 Innovation

Cross-DLT Layer brings innovation to iNGENIOUS from different perspectives. If we focus on TrustOS, it is a piece of software that removes the underlying complexity of DLTs. It offers the ability for any project to record business process information quickly and without the need for technical expertise. It offers native integration with Ethereum and all Ethereum-based platforms such as Polygon. This provides interoperability between different platforms because it is not necessary to choose a single one, but the data can be stored in several DLTs and the end-user makes the decision. Thanks to the DLT Common API, the solution offers ease of integration by any DLT based on the OpenAPI standard, this means that if new DLT platforms are to be included in the future, they only need to implement such an interface.

In short, Cross-DLT Layer allows to increase trust and extend the reach of your solution. The combination of multiple DLTs makes the solution attractive to a wider group of users as some will prefer some DLTs over others. It also provides a way to ensure and verify the integrity of information coming from different systems.

5.5 Business Potential and Exploitation

Component Group: Distributed Ledger Technologies Layer

- **TIOTBD**: From the perspective of exploitation and potential business the integration between TrustOS and different DLT connectors such as Ethereum, Polygon, Bitcoin, IOTA/Tradelens and Hyperledger Fabric could be exploited to provide interoperability between DLTs in projects that need data immutability, trust, and security.
  Integration with DVL through Integration Bridge can be exploited as a joint solution to secure information from different sources such as IoT devices.
  Also, the integration between TrustOS and Hyperledger Fabric, and the DVL and PI System OSI soft could be exploited in future projects to enhance the interoperability of the IoT systems of the port of Valencia and other external systems.

  Further details about the innovation, exploitation and impacts are reported in D7.3 [9] as well as in D2.5 [10].

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

6 Conclusions

This document reported on the finalization of the technical work previously provided with deliverables D5.1 [3] and D5.2 [7], and described the final solution of iNGENIOUS interoperable layer, focusing on a detailed technical description on how components were integrated with emphasis on addressed innovations and achievements. Furthermore, the relationship of the developed components with the project Use Cases and their future exploitation possibilities were discussed.

The final implementation of the iNGENIOUS IoT application layer was described including the implemented functionalities, data architecture, and system components for the two main development topics of predictive analytics and industrial & tactile IoT applications. The IoT applications technical functionalities were developed according to the development and integration activities specified in D6.1, while verifying the larger-scale benefits of the solutions' deployment in a commercial setting remains as part of future exploitation of the work.

With the aim of integrating the Smart IoT devices with the M2M space, the Smart IoT GW made available new interfaces towards the M2M platform. The oneM2M based solution was complemented with the industry standard M2M protocol MQTT. The entire network infrastructure for the M2M platform was implemented.

The final version of the Data Management framework was presented and described in terms of the available set of M2M platforms and data sources. The way such platforms are integrated with the DVL component, for data aggregation in the context of the different iNGENIOUS use cases, was also outlined. As part of the overall framework, the DVL was discussed with a focus on developed interfaces for maritime events and the main interaction with the Cross-DLT layer for data evidence and immutability proofs. The DVL was also described in relation to data consumers (namely Cross-DLT layer, pseudonymization module, Awake.AI platform, and dashboard for trucks' tracking) and the main data flows were explained accordingly.

Finally, the cross-DLT Layer provided iNGENIOUS with a new way of storing evidence of key points in the supply chain. This recording of information was orchestrated by TrustOS, which made it possible to store evidence of different events in several DLT networks simultaneously and regardless of the specific DLT technology used. TrustOS has the necessary methods to add new DLT providers in a simple way, as it has already integrated with several of them (Ethereum-based, IOTA, Bitcoin, Hyperledger Fabric, Besu, Polygon, etc). Thus, the project achieved a very high degree of DLT interoperability. It is currently possible to store evidence or trustpoints in four different DLTs and subsequently verify the information corresponding to the events coming from the DVL through each of them.
This feature, together with the automation of the entire information flow from the DVL and the end-user visualization interface, makes iNGENIOUS a pioneering solution in the new supply chain model based on distributed ledger technologies.

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

7 References

[1] iNGENIOUS Consortium, "D2.2 System and architecture integration (Initial)" 2022
[2] iNGENIOUS Consortium, "D2.4 System and architecture integration (Final)" 2022
[3] iNGENIOUS Consortium, "D5.1 Key Technologies for IoT data management benchmark" 2022
[4] iNGENIOUS Consortium, "D3.3 Secure, private and more efficient HW solutions for IoT devices" 2022
[5] Internet Engineering Task Force (IETF), Request for Comments 8446, The Transport Layer Security (TLS) Protocol Version 1.3, https://www.rfc-editor.org/rfc/rfc8446
[6] Robert Walther, Carsten Weinhold, Michael Roitzsch, RATLS: Integrating Transport Layer Security with Remote Attestation, 4th Workshop on Cloud Security and Privacy (Cloud S&P), 2022
[7] iNGENIOUS Consortium, "D5.2 Baseline iNGENIOUS data management framework" 2022
[8] iNGENIOUS Consortium, "D6.2 PoC Development, Platform and Test-bed Integration" 2022
[9] iNGENIOUS Consortium, "D7.3 Final Dissemination, Standardization and Exploitation" 2022
[10] iNGENIOUS Consortium, "D2.5 Regulatory Framework and Business Models" 2022
[11] iNGENIOUS Consortium, "D2.1 Use cases, KPIs and requirements" 2022
[12] iNGENIOUS Consortium, "D6.3 Final Evaluation and Validation" 2022
[13] TrustOS Documentation, https://TrustOS.readthedocs.io/en/latest/
[14] https://www.openssl.org

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