5G PPP
PUBLIC-PRIVATE PARTNERSHIP

DEDICAT 6G

DEDICAT 6G: Dynamic coverage Extension and Distributed Intelligence for human Centric Applications with assured security, privacy and Trust: from 5G to 6G

Deliverable D5.1
Specification of security framework and trust management platform

Project Details

Call	H2020-ICT-52-2020
Type of Action	RIA
Project start date	01/01/2021
Duration	36 months
GA No	101016499

Deliverable Details

Deliverable WP:	WP5
Deliverable Task:	Task T5.1and T5.2
Deliverable Identifier:	DEDICAT6G_D5.1
Deliverable Title:	Specification of security framework and trust management platform
Editor(s):	Srdjan Penjivrag (VLF)
Author(s):	D. Draskovic (Nokia), Dusan Borovcanin (Nokia), S. Penjivrag (VLF), F. Carrez (UoS), S. Abdelkareem (UoS), S. Roumpis (WINGS), V. Stavroulaki (WINGS), P. Demestichas (WINGS)
Reviewer(s):	(Nokia), (Airbus)
Contractual Date of Delivery:	31/12/2021
Submission Date:	23/12/2021
Dissemination Level:	PU
Status:	Final
Version:	1.0
File Name:	DEDICAT6G_D5.1 Specification of security framework and trust management platform_v1.0

Disclaimer
The information and views set out in this deliverable are those of the author(s) and do not necessarily reflect the official opinion of the European Union. Neither the European Union institutions and bodies nor any person acting on their behalf may be held responsible for the use which may be made of the information contained therein.

Deliverable History

Version	Date	Modification
V1.0	*23/12/2021*	*Final version, submitted to EC through SyGMa*

Table of Content

List of Acronyms and Abbreviations

Acronym/Abbreviation	Definition
AES	Advanced Encryption Standard
AGV	Automated Guided Vehicle
AI	Artificial Intelligence
AIDS	Anomaly (-based) Intrusion Detection System
API	Application Programming Interface
CA	Certificate Authority
CART	Classification And Regression Tree
COTS	Component Off the Shelf
CPU	Computer Process Unit
DES	Data Encryption Standard
DLT	Distributed Ledger Technology
DNN	Deep Neural Network
DRNN	Deep Recurrent Neural Network
DAE	Deep Auto Encoder
DH	Diffie Hellman
DL	Deep Learning
DoS	Denial of Service
DDoS	Distributed Denial of Service
FC	Functional Component
FL	Federated Learning
FTP	File Transfer Protocol
GW	Gateway
HIDS	Host (-based) Intrusion Detection System
HTTP	Hyper Text Transfer Protocol
HTTPS	Hyper Text Transfer Protocol Secured
IaaS	Infrastructure as a Service
IDS	Intrusion Detection System
IoT	Internet of Things
IIoT	Industrial Internet of Thing
KPI	Key Performance Indicator
KNN	K-Nearest Neighbours
LR	Logistic Regression

MAP	Mobile Access Point
MD5	Message Digest (algorithm) 5
ML	Machine Learning
MLP	Multi-Layer Perceptron
NDM	Nokia Data Marketplace
NIDS	Network (-based) Intrusion Detection System
NN	Neural Network
PKI	Public Key Infrastructure
PaaS	Platform as a Service
PCA	Principal Component Analysis
PST	Privacy Security (and) Trust
QoS	Quality of Service
REST	Representational State Transfer
RLC	Radio Link Control
RPC	Remote Procedure Call
RRC	Radio Resource Control
RSA	Rivest-Shamir-Adleman
RSS	RDF Site Summary
RSU	Road Side Unit
RU	Radio Unit
SC	Smart Contract
SHA	Secure Hash Algorithm
SLA	Service Level Agreement
SMF	Session Mobility Function
SNR	Signal Noise Ratio
SOTA	State Of The Art
SPP	Security and Privacy Protection
SSH	Secure Shell
SSL	Secured Socket Layer
TEE	Trusted Execution Environment
TTL	Time To Live
ToC	Table of Content
TSL	Transport Security Layer
U2R	User-to-Root (attack)

UAV	Unmanned Aerial Vehicle
UC	Use-Case
UDR	Unified Data Repository
UE	User Equipment (e.g., mobile phone)
UML	Unified Modelling Language
UPF	User Plane Function
V2V	Vehicle to Vehicle
V2X	Vehicle to X
VEC	Virtual Environment Control
VIM	Virtual Infrastructure Manager
VM	Virtual Machine
VN	Vehicular Node
VNF	Virtual Network Function
vRAN	Virtual Radio Access Network
VRU	Vulnerable Road User
WMS	Warehouse Management System
YAML	Yet Another Markup Language
ABAC	Attribute-based access control
RBAC	Role-based access control

List of Figures

List of Tables

Executive Summary

This deliverable reports the activity conducted in the Work Package 5 on Mechanisms for Security, Privacy, and trust. It will include initial specification of the DEDICAT 6G security and data protection framework and DEDICAT 6G trust management platform, architectures, tools, *Key Performance Indicators* (KPI) and set of standards and best practices to follow towards realization of the security and privacy protection tasks.

D5.1 will specify a federated learning mechanism for distributed training of machine learning models for threat detection and classification. System logs and key event information will be used for training local ML models within ecosystem of communication and computation networks formed with the project's framework. In this way highest level of privacy protection is ensured since data for model training are not transferred to a centralized entity.

Also, focus will be specification and implementation of trust management platform for DEDICAT 6G dynamic networking and computational distributed systems. This platform would be responsible to ensure the integrity of highly dynamic and distributed communication and computation systems by building certification mechanisms and compliance tests for all nodes, applications and services which comprise such systems. Validation mechanism will be detailed in the future deliverables.

The Key Performance Indicators measured in the WP 5 include:

- Adopted best practices and IoT domain standards for security and data protection to build ML models.
- Federated learning global models for security and data protection threat detection and classification
- Federated learning mechanisms implemented
- Implemented federated learning mechanisms for multiple system layers: federated learning local enabler for mobile platforms, microprocessor environments and lo-cal servers.
- Implemented blockchain data architecture/network for the project trust management platform (various configurations to be tested with ChainRider blockchain as a service solution)
- Configured and validated smart contract templates for trusted data exchange within project use cases and automated auditing of the edge computing system status

1 Introduction

The DEDICAT 6G *Artificial Intelligence* (AI) powered security management DEDICAT 6G framework will provide mechanisms for realizing threat detection, classification, and risk mitigation in the context of highly dynamic and distributed communication and computation networks.

The threat detection and classification mechanisms will be based on machine learning models which will be trained in federated learning manner. The ML models will run within a premise of locally established DEDICAT 6G communication and computation networks while a central orchestrator and aggregator maintains global models. This way the privacy protection is ensured since data remain within locally established systems while at the same time the ML models are adapted and trained on local conditions. The trust management platform based on private permissioned blockchain will provide an immutable record of key information required for ensuring compliance of all nodes participating in communication and computation networks.

Analytics for security, privacy and trust includes derivation of best practices for threat detection and classification, Federated learning approach for training AI elements of the security framework, Trust management with private permissioned blockchain networks and smart contracts for management of who, what and under what conditions writes into and reads from ledgers, Automated auditing during and after network operation, compliance checks for participating nodes and policy updates for trustworthy nodes will be implemented as part of the trust management functionality.

Federated Learning (FL) will be developed for training *Machine Learning* (ML) models (deterministic and probabilistic) capable of detecting and classifying security threats in communication and computation networks. Global ML models and FL strategy (local model update policy, global model update policy, ML model performance metrics, metadata, data models and interfaces) will be tailored to address the main challenges regarding security threat detection/classification and data privacy breach risks identification.

The security and data protection framework will integrate attribute-based access control and authorization for devices, services, and users. A trust management platform based on private permissioned blockchain, and a collection of smart contract templates will be implemented to facilitate trusted exchange of information and commands (including updates for local ML models) between nodes and systems participating in opportunistic communication and computation networks. This trust platform will include consensus mechanisms and set of rules indicating who, when, what and under which conditions can read/write to the immutable record. It will also include smart contracts supporting automated audits about performance of opportunistic communication and computation networks as well as automated compliance tests and certifications for candidate nodes and systems which are trusted to form and join Federated learning for threat identification and trust management platform based on private permissioned blockchain DEDICAT 6G networks.

This document aims to provide a first approach for the specification of the DEDICAT 6G security framework and trust management platform. The document starts from an overview of state of the art (section 2). Section 3 addresses privacy, security and trust protection plans and strategies. Section 4 focuses on the specification of the security and privacy protection framework. Section 5 presents threat identification and classification mechanisms. Section 6 provides a first specification of the trust management platform. Finally, section 7 concludes the document with a summary of the key points.

In compliance with the initial system architecture (D2.2), section 3 and section 4 from this document correspond to the specification of Audit FC, AuthN FC, AuthZ FC and Data marketplace FC. Section 5 is linked to the specification of the Threat Analysis FC, while section 6 is related to the specification of the Distributed Ledger FC, Trust Metrics FC, IdM FC and Logging FC.

2 State of the Art

This first section provides a short literature review focusing on the DEDICAT 6G research topics linked to Privacy Security and Trust. In Section 4 and 5 respectively, we then describe our plan and give initial specifications of the novel methods we intend to develop.

2.1 Intrusion Detection in IoT-traffic

The third industrial revolution is considered the *Internet of Things* (IoT) [1]. The IoT term refers to a new communication paradigm where the sensing of surrounding environments can be done with devices that have actuators and sensors that communicate with one another and exchange data through the internet [2]. The growth of the IoT market has been exponential as it started with 2 billion devices in 2006 and over 50 billion in the present day, which is expected to grow significantly in the next few years [3,4]. The use of IoT devices has been seen in several fields, including but not limited to agriculture, education, entertainment, finance, health, transportation, to name a few.

With the rapid commercialization of IoT, academia, individuals, and people in the industry are trying to address the safety and security concerns of IoT devices and networks. This is because IoT devices are exposed to many security vulnerabilities as they are connected to the global internet, which is not entirely safe. Intruders may exploit these vulnerabilities by injection of anomalies into the devices that may trigger wrong control decisions, leading to disastrous economic, life and property impacts [5,6].

2.1.1 Network Intrusion Detection in IoT

In detecting attacks in an IoT network, a network *Intrusion Detection System* (NIDS) is proposed to serve as additional line of defence after access control systems, antivirus, and firewalls for connected IoT devices [7]. A NIDS is an IDS system that capitalizes on network behaviour to work effectively by examining data exchanged within the network. A network-based IDS is deployed to detect intrusions in network data over network connections and protect all network nodes. Network-based intrusion detection systems that are most efficient can keep track and gather real-time system audits. It could also be scheduled, allowing for reduced computational resources used by the edge devices that connect the IoT devices to the network. Network intrusion detection can be classified into two categories: anomaly/ML-based and signature-based [8].

Anomaly-Based

This technique relies on finding unusual or deviating behaviour in the network [9]. The technique uses learned network features to determine if a behaviour is normal or an attack. The generality of this technique makes it extremely hard for intruders to avoid. Additionally, compared to the signature-based technique, the anomaly-based technique can detect new attacks that have not been previously known (day zero attacks). Nevertheless, the technique has the drawback of having high-false positive rates, which makes working in some cases difficult for it.

Signature Based

This technique uses a 'signature' that is linked with a specific intrusive exploit [10]. An antivirus program is the most common signature-based technique software, which has the task of scanning the signatures of all data downloaded or traversing through the network on a device. The alarm alert gets activated if the data that is being scanned is a known virus. Furthermore, unlike the anomaly-based technique, this technique hardly gives false alarms.

Hence it has a low false-positive rate. However, it can only detect known attacks. Additionally, regular updates and active subscriptions are essential to get the best out of this technique.

2.1.2 SOTA Approach for NID in IoT Network

Network intrusion detection techniques have been used and are still being studied in many recent studies. Many studies are still utilizing traditional datasets that have existed for over 15 years, specifically KDD99 [11] and NLS-KDD [12] datasets, which do not represent the present-day network features. In the study of Almseidin et al. [13], the focus was on evaluating various machine learning classifiers for false positive and false negative performance using the KDD99 dataset. However, the dataset is an outdated one. Similarly, Obeidat *et al.* [14], investigated the accuracy performance of some ML classifiers for attack detection on the dataset using the KDD99 dataset. In the work of Rajadurai and Gandhi [15], a stack ensemble learner was evaluated using the NLS-KDD dataset for anomaly detection. Although all the above-listed studies are recent, they all utilized datasets that do not have IoT network traces. They have been publicly available for many years before the IoT innovation. More recently, due to the non-representative and some other issues associated with the traditional datasets, researchers have created new datasets representing the present-day network settings.

Some of these datasets are ToN-IoT [16] and Bot-IoT [17] datasets used extensively to study the classification behaviour of ML classifiers. Some studies that have utilized them in their work are Gad et al. [18], that evaluated the detection performance of new network attacks by different machine learning classifiers using the ToN-IoT dataset. Also, Ferrag *et al.* [19] developed an ensemble learning classifier to classify network categories. In evaluating the classification performance of their classifier, the BoT-IoT network dataset was utilized in the study. Although conventional ML classifiers have been adopted in many studies, the need to improve the detection speed due to the significant IoT data volume being generated makes the technique's efficiency a concern.

To this end, the *Deep Learning* (DL) intrusion detection technique has been proposed to address these concerns seen with the conventional ML classifiers. DL expedites the analysis between fast and real data streams in extracting relevant information to predict the future of the IoT domain. This is because it is deemed more reliable due to its ability to extract generated dataset information for its classification task easily. Though this is the most recent approach researchers used to detect network attacks, its research study is not as pronounced in comparison to the ML techniques. Some recent studies that have used this approach are Popoola et al. [20] that evaluated the performance of a *Deep Recurrent Neural Network* (DRNN) using the Bot-IoT dataset for the classification of different attack types. In comparison, to other SOTA classifiers, theirs had a better classification performance in addition to being faster than most of the other approaches that used conventional ML classifiers, hence, addressing the shortcomings of the previous approach. Furthermore, in the work of Ferrag *et al.* [21] investigated the effectiveness of *Deep Auto Encoder* (DAE) and some other DL classifiers for botnet attack detection in IoT networks using a single hidden layer and different numbers of neurons. The DAE had the best detection speed and accuracy in comparison to other DL classifiers in the study. The research in the DL area is still an ongoing process, as researchers are looking to see how they can further optimize its intrusion detection performance to improve its efficiency using features selection techniques that remove redundant dataset features. Also, resampling the generated network data to enhance the training of the DL classifiers is a research area that is being studied actively.

2.1.3 IoT Network Traffic Generation

The generation of IoT dataset takes a different form compared to the traditional datasets that have been in existence over the years. In generating the KDD 99 [11] dataset, 41 features of an earlier created dataset network (DARPA) traces containing seven weeks of packet-based data records of an Air-Force base. There was no form of IoT traces in the dataset as it was created before the advent of IoT, and the network traffic was basically from host systems such as workstations and servers. However, the TON-IoT [16] dataset has IoT traces in it, as it was developed based on IoT and edge network of smart cities testbed architecture. The testbed design is based on interacting network and IoT/IIoT systems with three layers of edge, fog and cloud. There is a similarity in service delivery between edge and fog computing ad they both offer on-premises services, like cloud services, including *Software-as-a-Services* (SaaS), *Platform-as-a-Service* (PaaS), and *Infrastructure-as-a-Service* (IaaS).

The IoT devices that contributed to the dataset generation are Modbus, light bulb sensors, smartphones, and smart TVs. However, in generating the BoT-IoT [17] dataset, a testbed was created to generate data from a weather station, smart fridge, motion-activated light, remotely activated garage door, and smart thermostat. The note-red tool was used in simulating the IoT devices network behaviour. Node-red is a popular middleware used to connect IoT physical devices with their backend cloud server and applications, improving and speeding up communications between the various parts of an IoT deployment. The significant difference between traditional and IoT datasets is that the latter is generated using IoT devices while the previous is generated from host systems, mostly workstations and servers. Additionally, the attack types that the IoT dataset is exposed to are more diverse than the traditional dataset.

2.1.4 Future Directions

The future of network intrusion detection looks promising with the continuous research on generating modern representative network datasets. Also, existing approaches (ML and DL) used in detecting network attacks are being optimized to make them more effective and efficient. Additionally, new detection approaches are being proposed to diversify further the detection of intrusion, federated learning based on training ML classifiers on user data without transferring or collecting accumulated data from different repositories or servers is one of the new approaches. Furthermore, feature selection techniques on the different ML and DL detection approaches are being researched further. It helps make the classifiers run faster and smoother due to eliminating redundant dataset features, which further optimizes their performances.

2.2 Anomaly Intrusion Detection System (AIDS)

2.2.1 Introduction

Cyber-attacks are becoming more disruptive and sophisticated and therefore present an increasing challenge in accurately detecting malicious activities, break-ins, penetrations, and intrusions. *Intrusion Detection Systems* (IDS) dynamically monitor network logs, file systems, and real-time events occurring in a computer system or network and analyse them for signs of adversaries or attacks. [22]

IDSs are classified as *Host-based IDS* (HIDS), or *Network-based IDS* (NIDS). Host-based IDSs operate on information collected from within an individual computer system, such as audit trail information and system logs to detect malicious activities inside the system. Monitor the process activities and ensure security policies of system files, system logs, and registry keys. For example, repeated failed login attempts. On the other hand, NIDSs monitor incoming

and outgoing traffic to and from networked devices, collect raw network packets as the data source from the network, and analyse for signs of intrusions, malicious traffic on a network, attacks, or abnormal behaviour. [23]

NIDS systems can be broadly categorized into two groups: SIDS are based on pattern matching techniques to find a known attack. When an intrusion signature matches with the signature of a previous intrusion that already exists in the signature database, an alarm signal is triggered. Even though SIDS is simple and effective in detecting known attacks they have little understanding of states and protocols and it's hard to keep signatures/patterns up to date. Furthermore, they are ineffective to detect unknown attacks and variants of known attacks. *Anomaly-based Intrusion Detection Systems* (AIDS) have drawn interest from a lot of researchers as they can overcome the limitation of SIDS. In AIDS, a normal model of the behaviour of a computer system is created using machine learning methods. Any significant deviation between the observed behaviour and the model is regarded as an anomaly, which can be interpreted as an intrusion. The assumption for this group of techniques is that malicious behaviour differs from typical user behaviour. They are more effective to detect new and unforeseen vulnerabilities, but they have High False-positive alarms.[24]

AIDS methods can be categorized into three main groups: Statistics-based, knowledge-based, and machine learning-based. The statistics-based method collects and examines every data record in a set of items and builds a statistical model of normal user behaviour. On the other hand, knowledge-based tries to identify the requested actions from existing system data such as protocol specifications and network traffic instances, while machine-learning methods acquire complex pattern-matching capabilities from training data. [24]

2.2.2 AIDS based on machine learning techniques

Machine learning techniques are being widely used in AIDS. Several algorithms and techniques such as clustering, *Support Vector Machine* (SVM), naive Bayes, neural networks, decision trees, *K-Nearest Neighbor* (KNN), *neural network* (NN), *Deep Neural Network* (DNN), and random forest, have been applied for discovering the knowledge from intrusion datasets [25]. Furthermore, a convolutional neural network model was used to create a multiclass classification model [26].

Several researchers proposed a hybrid intelligent approach, applying feature selection methods to reduce the complexity of the data combined with machine learning techniques. Panda *et al.* [27] combined *Principal Component Analysis* (PCA) and random forest among other techniques. Bajaj *et al.* [28] proposed a technique for feature selection using *Information Gain* (IG) and Correlation Attribute evaluation combined with C4.5, naïve Bayes, NB-Tree, and *Multi-Layer Perceptron* (MLP). Based on the univariate statistical test result, N. Aboueata *et al.* [29] proposed a univariate chi-square test (ChiX2) that selects the best K features. Z. Chkirbene *et al.* [30] used the Random Forest algorithm for the feature selection to find the most important features combined with *Classification and Regression Trees* (CART).

2.2.3 AIDS Datasets

Researchers have created engineered benchmark NIDS datasets because of the complexity in obtaining labelled realistic network traffic. KDDCup 99 is one of the most commonly used publicly available datasets. It contains a series of TCP sessions starting and ending at well-defined times, between which data flows to and from a source IP address to a target IP address, which contains a large variety of attacks simulated in a military network environment. The dataset contains 41 features and 5 classes ('Normal', 'DoS', 'Probe', 'R2L', 'U2R').

NSL-KDD is the distilled version of KDDCup 99 intrusion data. Filters are used to remove redundant connection records in KDDCup 99. It can protect machine learning algorithms not to

be biased. Although both suffer from representing the real-time network traffic profile characteristics. [31]

The IXIA PerfectStorm tool in the Cyber Range Lab of UNSW Canberra created the raw network packets of the UNSW-NB15 dataset for generating a hybrid of real modern normal activities and synthetic contemporary attack behaviors. This dataset has nine types of attacks, Fuzzers, Analysis, Backdoors, *Denial of Service* (DoS), Exploits, Generic, Reconnaissance, Shellcode and Worms and 49 features with the class label. It is also considered as the new benchmark to evaluate intrusion detection systems.[32]

CICIDS2017 includes real-time background traffic. Using the B-profile system, benign background traffic was collected. This benign traffic contains the characteristics of 25 users based on HTTP, HTTPS, FTP, SSH, and email protocols. It contains attacks that have occurred recently. It also contains 49 features and 8 attack classes.[31]

2.2.4 Threat analysis

The primary DEDICAT 6G threat analysis was conducted in the context of the architecture work leading to deliverable D2.2. Being an integral part of the requirement engineering process, it was used to:

- Determining the vulnerabilities of the DEDICAT 6G system after all actors and physical system and entities pertaining to the DEDICAT 6G eco-system were clearly identified;
- Listing the threats that can potentially exploit those vulnerabilities;
- Assessing the likelihood of such threats to be indeed exploited (risk);
- Assessing the impact resulting from the exploitation of the vulnerabilities (through implementing the threats then).

This early activity is paramount when it comes to deciding about our platform Privacy, Security and Trust functional and non-functional requirements.

Those requirements were analysed, then strategies and tactics needed being implemented were devised (see in particular the Perspective Sections 6.1 (Privacy), 6.2 (Security) and 6.3 (Trust) in D2.2), leading to PST-specific functional components, part of the whole project functional decomposition.

However, implementing functional components comes with its own set of new vulnerabilities treats and risks, especially considering PST –related components.

In this section we therefore not only remind about the main results of the primary threat and risk analysis but also assess the vulnerabilities and threats that apply to the PST-related subsystem we are planning to implement, leading to some sort of secondary protection-schemes.

2.3 Primary Threat Analysis

This first section recalls the main outcomes of Section 3.1 in Deliverable D2.2 and focuses on the DEDICAT 6G specifics. The two following tables (integrally taken from D2.2) summarize the risks and impacts, the first one sets the ground with general concerns while the second one tackles the DEDICAT 6G project specific issues focusing in particular on all physical system (either baseline or scenario-specific).

Table 1: General risk definition and the assignment of evaluation values

Risks	Likelihood of occurrence	Potential impact
loss of integrity or confidentiality	High	High

data interception of signalling and user data	Low	Medium
Modifying data or code	Low	High
Hardware failure caused by cyber attack	Low	High
Data leaks	Low	Medium
loss of privacy	Medium	Medium
loss of availability of resources or service	Medium	High
Loss of trustworthiness to authorities	Low	Medium
Destruction of components	Medium	High
Installation of intentional malfunction, sabotage	Low	High

Table 2: Risk definition and the assignment of evaluation values to the Physical Systems

Physical system name	Risks	Likelihood of occurrence	Potential impact
Edge-terminal	• Loss of availability of resources or service • Destruction of components	• Low • Medium	• Medium • Medium
AGVs	• Loss of integrity or confidentiality • Hardware failure caused by cyber attack • Loss of availability of resources or service • Modifying data or code • Loss of trustworthiness • Installation of intentional malfunction, sabotage	• Low • Medium • Medium • Medium • Low • Low	• Medium • High • High • Medium • Low • High
Forklift/machine	• Hardware failure caused by cyber attack	• Medium	• High
SmartAccess360 controller	• Loss of integrity or confidentiality • Hardware failure caused by cyber attack • Loss of availability of resources or service • Modifying data or code • Loss of trustworthiness	• Low • Medium • Medium • Medium • Low	• Medium • High • High • Medium • Low
Warehouse personnel smartphone/mobile device	• Loss of availability of resources or service • Loss of integrity or confidentiality • Data leaks • Loss of privacy	• Medium • Low • Low • Medium	• High • Low • Medium • Medium
(B)5G Networking Equipment	• Loss of integrity or confidentiality • Hardware failure caused by cyber attack	• Low • Medium	• Medium • High

	• Loss of availability of resources or service • Data leaks • Installation of intentional malfunction, sabotage	• Medium • Low • Low	• High • Medium • High
Video streaming platform	• Loss of availability of resources or service • Modifying data or code • Loss of trustworthiness	• Medium • Medium • Low	• High • Medium • Low
Drones	• Loss of integrity or confidentiality • Hardware failure caused by cyber attack • Loss of availability of resources or service • Data leaks • Loss of trustworthiness • Installation of intentional malfunction, sabotage	• Low • Medium • Medium • Low • Low • Low	• Medium • High • High • Medium • Low • High
Smart phones	• Loss of availability of resources or service • Data leaks • Loss of privacy	• Medium • Low • Medium	• High • Medium • Medium
smartGlass	• Modifying data or code • Data leaks • Loss of privacy • Installation of intentional malfunction, sabotage	• Medium • Low • Medium • Low	• Medium • Medium • Medium • High
Connected Car (maybe different from UC1)	• Hardware failure caused by cyber attack • Loss of availability of resources or service	• Medium • Medium	• High • High
MCS mobile server	• Hardware failure caused by cyber attack • Loss of availability of resources or service • Modifying data or code • Data leaks	• Medium • Medium • Medium • Low	• High • High • Medium • Medium
Attendee smartphone	• Loss of integrity or confidentiality • Data leaks • Loss of privacy	• Low Low • Medium	• Low Medium • Medium
1st Responder smart phone	• Loss of availability of resources or service • Loss of integrity or confidentiality • Data leaks • Loss of privacy	• Medium • Low • Low • Medium	• High • Low • Medium • Medium
SmartGate	• Loss of integrity or confidentiality • Loss of trustworthiness • Installation of intentional malfunction, sabotage	• Low Low • Low	• Low • Low • High
Smart vehicle	• Hardware failure caused by cyber attack	• Medium	• High

	• Loss of availability of resources or service • Modifying data or code	• Medium • Medium	• High • Medium
Smarter vehicle (incl. tablet-like terminal)	• Loss of availability of resources or service • Modifying data or code	• Medium • Medium	• High • Medium
IoT Nodes	• Loss of availability of resources or service • Modifying data or code • Data leaks	• Medium • Medium • Low	• High • Medium • Medium
RSU	• Data leaks	• Low	• Medium

Following the previous table, a corresponding set of PST functionalities have been assigned to each physical system:

Table 3: PST functionalities

Physical system name	Security Requirement
Edge-terminal	• Access, Authentication, and Authorization Management • Network and data security • Audit logging and analysis
AGVs	• Access, Authentication, and Authorization Management • Audit logging and analysis • Network and data security • Code integrity
Forklift/machine	• Access, Authentication, and Authorization Management
SmartAccess360 controller	• Access, Authentication, and Authorization Management • Audit logging and analysis • Cryptography and key management • Network and data security • Code integrity • Data validation and sanitization
Warehouse personnel smartphone / mobile device	• Access, Authentication, and Authorization Management • Audit logging and analysis • Data validation and sanitization
(B)5G Networking Equipment	• Access, Authentication, and Authorization Management • Audit logging and analysis • Cryptography and key management • Network and data security • Code integrity
Video streaming platform	• Audit logging and analysis • Code integrity
Drones	• Access, Authentication, and Authorization Management • Audit logging and analysis • Network and data security • Code integrity
Smart phones	• Access, Authentication, and Authorization Management • Audit logging and analysis • Data validation and sanitization
SmartGlass	• Access, Authentication, and Authorization Management • Audit logging and analysis

Connected Car (maybe different from UC1)	• Access, Authentication, and Authorization Management • Audit logging and analysis
MCS mobile server	• Access, Authentication, and Authorization Management • Audit logging and analysis • Network and data security • Code integrity • Data validation and sanitization
Attendee smartphone	• Access, Authentication, and Authorization Management • Audit logging and analysis
1st Responder smart phone	• Access, Authentication, and Authorization Management • Audit logging and analysis
SmartGate	• Access, Authentication, and Authorization Management • Audit logging and analysis
Smart vehicle	• Access, Authentication, and Authorization Management • Audit logging and analysis
Smarter vehicle (incl. tablet-like terminal)	• Access, Authentication, and Authorization Management • Audit logging and analysis
IoT Nodes	• Access, Authentication, and Authorization Management • Audit logging and analysis • Network and data security • Code integrity • Data validation and sanitization
RSU	• Audit logging and analysis

2.4 Secondary Threat Analysis

In this section we identify a few vulnerabilities and threats applying to the baseline security solutions devised from the previous table:

- **Authentication:** password-only based authentication is considered as weak. Simple password can be easily attacked using a dictionary brute-force method. Token-based authentication can be also subject to replay-attacks. Different methods exist to deal with such attacks like strengthening the complexity and length of passwords, or using specific security protocols like challenge-response for example;
- **Cryptographic keys:** those can be stolen or tampered with or intercepted during the distribution process. They also need to be protected after reaching their destination;
- **Logs:** logs contain the history of almost everything occurred during the DEDICAT 6G run time. Logs are used in particular by the Audit functionality that aims at running specific algorithms off-line in order to identify threats. The efficiency and accuracy of the audit procedure depends in turn on the accuracy of the logs themselves. Logs need therefore to be protected against attacks like deletion, modification or even unauthorized insertion or log entries.

The measures to be provided to face those threats are the following:

- **Securing the cryptographic keys:** the storage of the various cryptographic keys after being generated and distributed must be secured in order to make sure they cannot be stolen and used for instance to perform masquerading or to jeopardize the digital signature process which would result in threatening data integrity;

- **Key distribution:** The process of key distribution between two parties must be secured for obvious reasons. DEDICAT 6G will use both symmetric and asymmetric encryption methods. Both distributions methods need being addressed. public key in order to generate a common secret;
- **Stronger authentication process:** different methods can be used to strengthen the authentication process, both for peer-to-peer authentication and Human-to-System authentication. They can be based on specific policies when choosing a password or on involving protocols or multiple authentication factors that cannot be forged.
- **Log protection:** Peer-to-peer authentication, access-control and data integrity must be provided;

DEDICAT 6G does not plan to conduct any research in order to improve the state of the art in authentication, access control and cryptography-related matters. We will therefore use COTS open-source security package in order to address the threats described above. We will elucidate further our different objectives and strategies, as far as utilizing those open-source features for our own sake is concerned, in the next Section.

2.5 SOTA Approach for Trust management

One of key challenges and strong points of DEDICAT 6G is the application of blockchain for enhancing the trustworthiness of the system. DEDICAT 6G will integrate and deploy advanced AI based security and privacy protection framework with blockchain based trust management platform for securing the distributed network ecosystem, building trust between parties, devices and sub-systems, as well as providing intelligence for detecting and preventing potential security, privacy, and trust issues.

Figure 1 provides an overview of Security, Privacy and Trust platforms:

For implementing and validating trust management platform for innovative system integrity and data protection management blockchain as a service platform ChainRider will be applied. Along with quick network configuration, the service supports networks with multiple organizations and channels allowing for innovative business models and use cases. The service supports scalable and custom-made topologies of the network that include an unlimited number of organizations, channels, physical machines, and peers, in order to build the network tailored to any business use case.
Smart Contract Generator service allows for the creation of Hyperledger Fabric smart contract to be built and deployed. All smart contracts are available in Node.js. Once the smart contract is deployed on a blockchain network user can immediately write/read to/from its data ledger. A smart contract, together with the ledger, form the heart of a Hyperledger Fabric blockchain system. Whereas a ledger holds facts about the current and historical state of a set of business objects, a smart contract defines the executable logic that generates new facts that are added to the ledger. A chaincode is typically used by administrators to group related smart contracts for deployment but can also be used for low level system programming of Fabric. In this topic, we'll focus on why both smart contracts and chaincode exist, and how and when to use them.

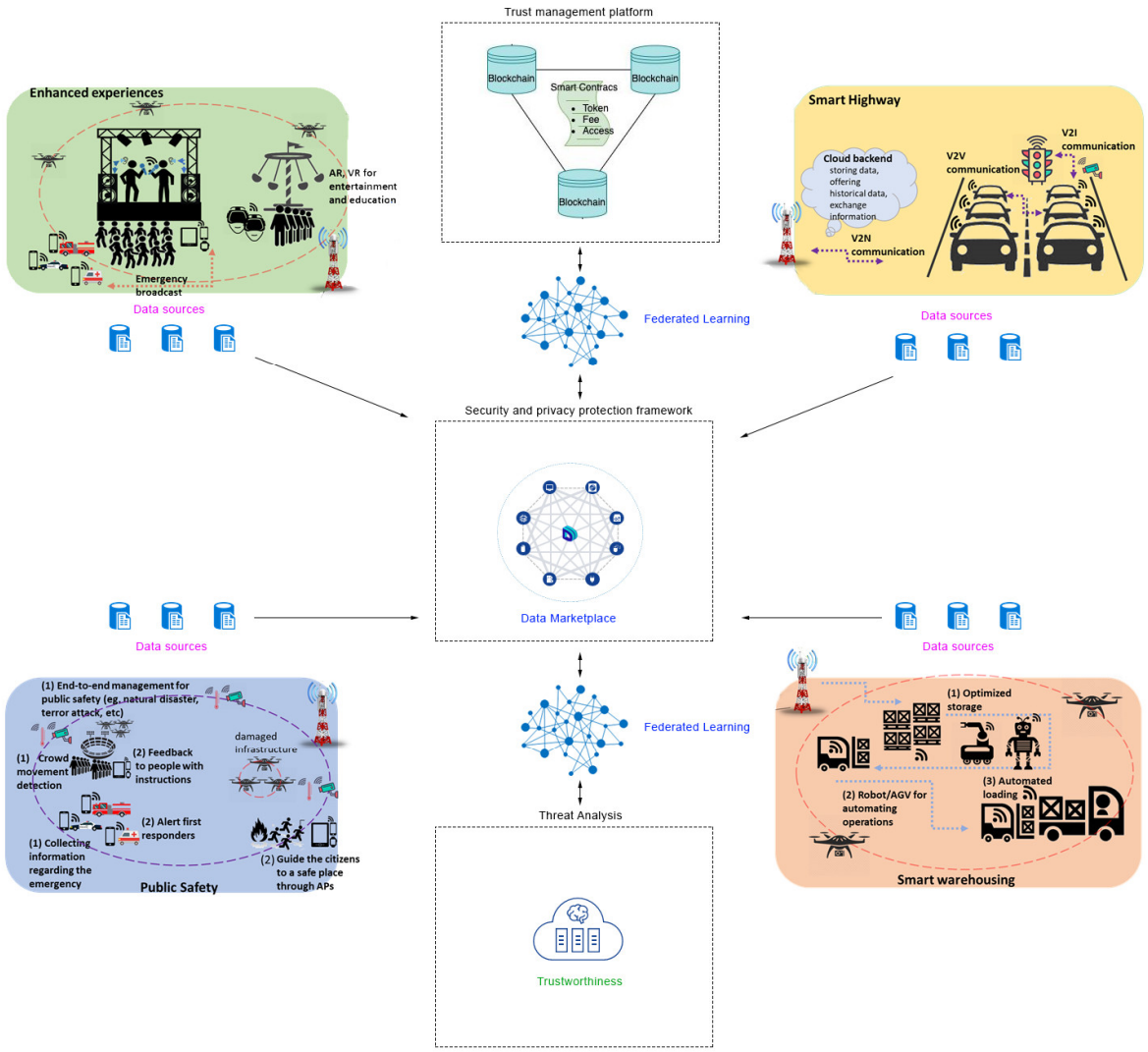

Figure 1: Overview of Security, Privacy and Trust platforms and DEDICAT 6G use cases

3 Privacy, Security and Trust protection plans and strategies

Considering the result of the initial Threat Analysis as depicted in Section 3.1 of D2.2 and its extension as show in the previous section a certain number of PST-related threats have been identified and need to be dealt with because of the level of risk they imply. The purpose of this section is to elucidate the different strategies we will implement in the DEDICAT 6G project in order to mitigate those threats, focusing initially on a cybersecurity management plan. It is also worth noting that both primary and secondary threat analysis are taken into account in this process.

In this section we recall the different solutions we are going to address in DEDICAT 6G for ensuring:

- **Strong authentication** of the Actors engaged in Human/platform interactions;
- **Strong Peer-to-peer authentication** of FCs in the context of FC-to-FC interactions (especially in a 0-Trust setting);
- **ID & Attribute-based access control;** where access means more specifically the ability to be granted access to resources or to invoke specific FCs methods;
- **Anonymity** based Identity management features;
- **Confidentiality:** data encryption is used to preserve the nature and content of data and to prevent that data to be accessed by individuals or entities who/which are not entitled to;
- **Data integrity:** Either in storage or on the move, data is vulnerable and can be tempered with or stolen. In order to ensure data integrity, techniques based on the use of hash-function can be used;
- **Code integrity:** Dynamic Intelligence Distribution is one of the three main innovation pillars of DEDICAT 6G. It implies that code can be dynamically migrated between execution environments depending on the context. As for data, code can be tempered with while be transferred through a communication channel, and therefore its integrity can also be ensured using hash-functions;
- **Non-repudiation:** this functionality involves signing, logging and some other mechanisms based on ledger and the blockchain (which would also provide integrity);
- **Intrusion Detection:** identifying those off-the-chart suspicious behaviors that can be classified as intrusion attempt, identifies originators and preventing them accessing the system further;
- **Auditing:** off-line overall security analysis though Audit;
- **Secured communication channels**.

As far as Authentication, Identity Management, Access control and Encryption are concerned, some *Components-Off-The-Shelf* (COTS) will be used as there is no plan to go beyond the SOTA for those functionalities; As for encryption we will be using both the *Public Key Infrastructure* (KPI) and therefore asymmetric encryption scheme (e.g. RSA) and symmetric encryption (e.g., DES, AES, triple-DES or Blowfish); each one being used in different contexts for different purposes.

Context in particular depends on the nature of the entity responsible of the ciphering/deciphering task and the computing power assigned to that task, as symmetric encryption is greedy in term of CPU. The key length (number of bits) can also be adjusted to mitigate such

issues or depending on the *Time to Live* (TTL) of the key itself, assuming compromising the key would take longer time that the very expected key TTL given a certain key length.
As far as the KPI is concerned, DEDICAT 6G is also expected to play the role of *Trusted Third Party* (TTP) responsible for key generation and compromised key-pairs revocation.
As for Integrity and non-repudiation some clear strategies will be elucidated in the Privacy and Trust sections.
This next paragraph elucidates potential strategies for dealing with the above cyber-security objectives. Decisions about those strategies will be ultimately agreed upon later during the course of the project, especially after some system use-cases have been thoroughly devised by the project partners involved in the architecture work:

- **Stronger Authentication:** Choosing strong (2 or more of the 3 following factors) authentication over simpler authentication scheme (e.g., password-based only) decrease the risk of masquerading and intrusion. Three different factors are usually considered, consisting of:
    - **Part of the person** authenticating (retina scan, fingerprint)
    - **What the person knows** (e.g., password): a secondary measure may involve password policy (password complexity, length, using non alphabetical/numerical characters and mix of upper/lower cases) in order to make brute force attacks difficult or even impossible. Restricting the number of attempts with increasing delay time between attempts e.g.
    - **What the person owns** (e.g., a token with one shot generated password or a mobile phone with dedicated app)
- **Confidentiality:** the confidentiality of data during storage relies on encryption, while its confidentiality during communication to a 3rd party relies on a secure communication channel, itself be based also on encryption. The security of streamed data can rely on specific symmetric encryption algorithms like RC4. It is worth noting that the confidentiality of data over the air (meaning during its communication via the 5G radio is encrypted by default following the 5G 3GPP standards);
- **Key distribution:**
    - Symmetric keys can be exchanged between two parties using *Diffie-Hellman* (DH) key-exchange protocol. In this protocol each party generates its own key pair (public and private) and exchange their private keys. Each party then "combines" its own secret key and other parties in order to generate a common secret which was never disclosed during the process. The "combination" is where the magics happens. This common secret is then used to generate a common secret key (symmetric encryption); It is worth noting that the TTL of the two RSA key-pairs generated in that process is restricted to the duration of the DH protocol; When a TTP generates RSA key-pairs they also need being distributed to the 3rd party requiring them. This can be achieved by using the DH protocol as defined above. Of course, public keys pertaining to a given party can be requested any time by a third party (for verifying a digital signature for instance), but the private keys need be securely stored at the TTP side.
- **Secured communication channels:**
    - **Asynchronous communication channel** (e.g., REST): the message pay-load can be encrypted using any method agreed between the two parties involved;
    - **Publish/subscribe communication channel:** this kind of communication is 1-to-N where N depends on who has subscribed to the topic of the message being published. Confidentiality of that channel can reasonably only rely on symmet-

ric encryption where the secret key is shared by all parties involved in the communication. Message being expected short in term of bytes, the CPU requested by the ciphering / deciphering process should not raise issues even for small devices at the far-edge. The topic shall not be encrypted.

- **Integrity/non-repudiation:** integrity of data and code is usually dealt with using hash function (e.g., MD-5 and those of the SHA family) and possibly digital signature (encryption of the digest with an owner or originator's private key). Blockchain provides an alternative way to deal with those features;
- **Intrusion detection:** this part is covered in detail in section 5;
- **Logging:** some aspects of logging are being covered by the Nokia *Data Marketplace* (NDM) and blockchain as explained in section 4;
- **Audit:** the auditing related to security threats is covered by the Threat Identification process

4 Security and privacy protection framework specification

In order to monitor security threats and establish secure and privacy-preserving AI/ML training and inference, a framework for secure data exchange must be set up.

Nokia have been working in recent years on a *Nokia Data Marketplace* and architectural blocks and elements of this product will be incorporated in DEDICAT 6G privacy-protection framework specification.

The DEDICAT 6G security and privacy protection framework will be based on a decentralized, blockchain powered data marketplace for secure, automated monetization, processing and exchange of IoT sensors and digital assets data with technical and policy-based data verification.

The framework's unique features for monetization and exchange of data between arbitrary interested parties are:

- Private, permissioned Blockchain technology which provides network security, data integrity, smart contract for fast automated transactions and micropayments with a token economy;
- Data verification, technical and policy-based through blockchain Smart Contracts and data hashing (anchoring).

A framework like this can be used to stream any data from any source, IoT devices, physical assets, autonomous cars, drones and many more. It enables integration of 3rd party data and exploitation through the same marketplace.

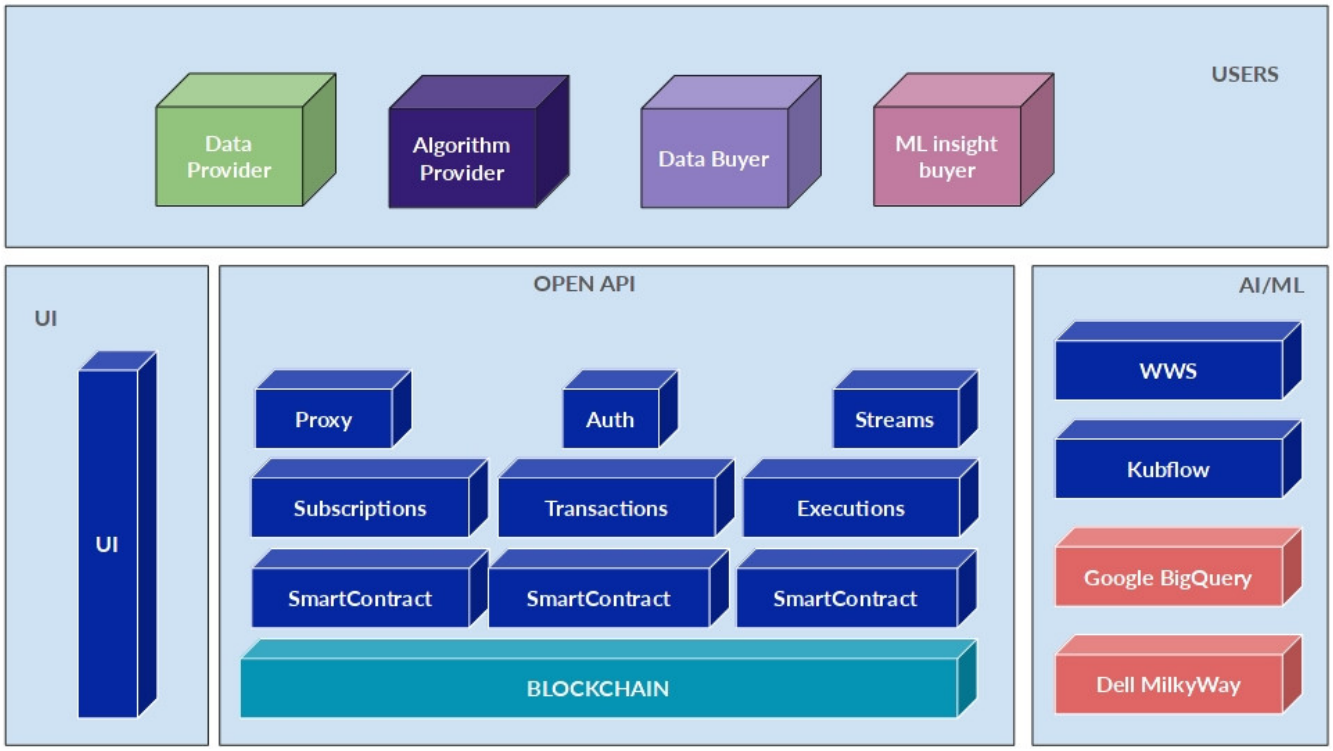

Figure 2: Security Framework components

The Security and privacy protection framework/Data marketplace is organized in a form of platform on top of which Machine Learning or Artificial intelligence orchestration will be

specified, so that AI/ML algorithms can be executed in a secure and privacy-preserving manner.

The platform uses a microservices architecture to provide performance and horizontal scalability.

Figure 2 shows the Security and privacy protection framework components:

Data and algorithm providers, as well as data buyers and consumers, can use web or mobile app clients to communicate with the framework using specified API, while framework components communicate internally using internal APIs and/or message brokers. Framework components include two domains:

- *Privacy preserving domain* that relies on blockchain
- *AI domain t*hat relies on tools for AI workflow management and workflow management in general.

All the internal components from both domains – services and its dependencies – are running in the private network and public entry points to them are strictly controlled by ingress controllers and reverse-proxies.

Inter-component communication and API won't be discussed in this document. It can be implemented using one of the following approaches or can combine them:

- HTTP API
- Remote procedure calls
- Message broker

We will focus on the interface exposed to the end clients and the visible part of the platform that can be reached from the outside. The framework exposes HTTPS RESTful API to the clients. Now we'll explain the main domain entities, what they represent and how they interact. The main two resources are **Asset** and **Subscription**.

4.1 Asset

Asset represents a digital asset. It can be anything from dataset and algorithm to picture, video, or NFT – anything that has URI. That's something users can sell in the Data Marketplace and something other users can buy or, in the context of the data marketplace, rent.

The API specification for Assets management is provided below:

4.1.1 Create a new asset (POST /assets)

Parameters

Type	Name	Description	Shema
Header	Authorization *required*	User's access token.	string
Body	asset *required*	JSON-formatted document describing the new asset.	asset

Responses

HTTP Code	Description	Schema
201	Asset created.	Asset ID

400	Failed due to malformed JSON.	No Content
403	Missing or invalid access token provided.	No Content
409	asset already exists.	No Content
500	Unexpected server-side error occurred.	No Content

4.1.2 Retrieve assets by query parameters (GET /assets)

Parameters

Type	Name	Description	Schema	Default
Header	Authorization *required*	User's access token.	string	
Query	**limit** *optional*	Number of items that will be taken starting from page parameter. The final result set is in range [page * limit, page * limit + limit).	number	20
Query	**maxPrice** *optional*	Upper bound of price range search. Entries with value equal to this will be included in result set.	number	
Query	**minPrice** *optional*	Lower bound of price range search. Entries with value equal to this will be included in result set.	number	
Query	**name** *optional*	Name of the asset to be found. Name search is case sensitive and uses partial matching. You can search by name using logical NOT to match all assets with the name other than the provided one. You can achieve this by adding "-" character at the beginning of the value. If the name starts with the "-" character, add one more "-" to the prefix to prevent "-" at the beginning being interpreted as logical NOT.	string	
Query	**owner** *optional*	ID of the owner of the asset. You can search by the owner using logical NOT to match assets with the owner other than the provided one. You can achieve this by adding "-" prefix to the value.	string	
Query	**page** *optional*	Start page from which items will be included in the result set.	number	0

Query	**type** *optional*	Type of the asset to be found. Type search is case sensitive and uses partial matching. You can search by type using logical NOT to match all assets with the type other than the provided one. You can achieve this by adding "-" character at the beginning of the value. If the type starts with the "-" character, add one more "-" to the prefix to prevent "-" at the beginning being interpreted as logical NOT.	string	
Query	**x0** *optional*	X value of the first search point. X values represent longitude here.	number	
Query	**x1** *optional*	X value of the second search point. X values represent longitude here.	number	
Query	**x2** *optional*	X value of the third search point. X values represent longitude here.	number	
Query	**x3** *optional*	X value of the fourth search point. X values represent longitude here.	number	
Query	**y0** *optional*	Y value of the first search point. Y values represent latitude here.	number	
Query	**y1** *optional*	Y value of the second search point. Y values represent latitude here.	number	
Query	**y2** *optional*	Y value of the third search point. Y values represent latitude here.	number	
Query	**y3** *optional*	Y value of the fourth search point. Y values represent latitude here.	number	

Responses

HTTP Code	Description	Schema
200	Data retrieved.	Asset page
403	Missing or invalid access token provided.	No Content
500	Unexpected server-side error occurred.	No Content

4.1.3 Add a bulk of new assets (POST /assets/bulk)

Parameters

Type	Name	Description	Schema
Header	**Authorization** *required*	User's access token.	string
FormData	**csv** *optional*	The csv file that contains a list of assets to be uploaded.	file

Responses

HTTP Code	Description	Schema
201	Asset created.	Asset ID
400	Failed due to malformed JSON.	No Content
403	Missing or invalid access token provided.	No Content
409	Some of the assets already exist in the database. This status is most likely a result of asset URL uniqueness violation. All valid assets are successfully added.	Error text
500	Unexpected server-side error occurred.	No Content

4.1.4 Retrieve asset info (GET /assets/{assetId})

Parameters

Type	Name	Description	Schema
Header	**Authorization** *required*	User's access token.	string
Path	**assetId** *required*	Unique asset identifier.	string

Responses

HTTP Code	Description	Schema
200	Data retrieved.	Asset
403	Missing or invalid access token provided.	No Content
404	asset does not exist.	No Content
500	Unexpected server-side error occurred.	No Content

4.1.5 Update asset1 (PUT /assets/{assetId})

Parameters

Type	Name	Description	Schema
Header	**Authorization** *required*	User's access token.	string
Path	**assetId** *required*	Unique asset identifier.	string
Body	**asset** *required*	JSON-formatted document describing the updated asset.	Asset

[1] Update is performed by replacing the current resource data with values provided in a request payload. Note that the asset's ID cannot be changed.

Responses

HTTP Code	Description	Schema
200	asset updated.	No Content
400	Failed due to malformed JSON.	No Content
403	Missing or invalid access token provided.	No Content
404	asset does not exist.	No Content
415	Missing or invalid content type.	No Content
500	Unexpected server-side error occurred.	No Content

4.1.6 Remove asset

(DELETE /assets/{assetId})

Parameters

Type	Name	Description	Schema
Header	**Authorization** *required*	User's access token.	string
Path	**assetId** *required*	Unique asset identifier.	string

Responses

HTTP Code	Description	Schema
204	Asset removed.	No Content
403	Missing or invalid access token provided.	No Content
500	Unexpected server-sed error	No Content

4.1.7 Definitions

Location

Name	Description	Schema
coordinates *required*	Location coordinates	< number > array
type *optional*	Location type	string

Page

Name	Description	Schema
content *optional*		Asset
limit *optional*	Preferred size of the page content. The size of returned set might be less than the limit but must not be more than the limit.	number
page *optional*	Number of pages returned by service. **Minimum value:** 0	number
total *optional*	Total number of elements satisfying query. **Minimum value:** 0	number

Asset

Name	Description	Schema
description *optional*	Description of the asset.	string
Id *required*	Unique asset identifier generated by the service.	string
Location *optional*		Location
name *optional*	Name of the asset.	string
owner *optional*	Id of the user who created the asset.	string
price *optional*	Price of the asset in tokens.	integer
type *optional*	Type of the asset.	string
url *optional*	URL of the asset.	string

4.2 Subscription

Subscription represents data info about renting the Asset. Subscriptions are created by renting the Asset. Subscriptions can't be deleted but can expire. The price is equal to the Asset unit price multiplied by the subscription period. Below you can find the API specification for subscriptions management.

API specification for Subscription management is provided below:

4.2.1 Create Subscription (**POST** /subscriptions)

Parameters

Type	Name	Description	Schema
Header	**Authorization** *required*	User access token.	string
Body	**subscription** *required*	JSON-formatted document describing the new subscription.	Subscription Request

Responses

HTTP Code	Description	Schema
201	Subscription created.	Subscription ID
400	Failed due to malformed JSON.	No Content
402	Subscription already taken.	No Content
500	Unexpected server-side error occurred.	No Content

4.2.2 Retrieve all the subscriptions for the user (GET /subscriptions/bought)

Parameters

Type	Name	Description	Schema
Header	**Authorization** *required*	User access token.	string

Responses

HTTP Code	Description	Schema
200	Subscription list retrieved.	Subscription Page
403	Missing or invalid access token provided.	No Content
404	User doesn't have any subscription.	No Content
404	Unexpected server-side error.	No Content

4.2.3 Retrieve subscriptions for the assets owned by the user (GET /subscriptions/sold)

Parameters

Type	Name	Description	Schema
Header	**Authorization** *required*	User access token.	string

Responses

HTTP Code	Description	Schema
200	Subscription list retrieved.	Subscription page
403	Missing or invalid access token provided.	No Content
404	User doesn't have any subscription.	No Content
500	Unexpected server-side error occurred.	No Content

4.2.4 Retrieve a Subscription by the given ID (GET /subscriptions/{id})

Parameters

Type	Name	Description	Schema
Header	**Authorization** *required*	User access token.	string
Path	**id** *required*	Unique Subscription identifier.	string

Responses

HTTP Code	Description	Schema
200	Subscription retrieved by ID.	Subscription

403	Missing or invalid access token provided.	No Content
404	User doesn't have any subscription.	No Content
500	Unexpected server-side error occurred.	No Content

4.2.5 Definitions

Page

Name	Description	Schema
content *optional*		Subscription array
limit *optional*	Preferred size of the page content. Size of returned set might be less than limit but must not be more then limit. **Maximum value:** 100	number
page *optional*	Number of pages returned by service. **Minimum value:** 0	number
total *optional*	Total number of elements satisfying query. **Minimum value**: 0	number

Subscription

Name	Description	Schema
end_date *optional*	Date to which subscription is active.	string (date-time)
hours *optional*	Number of hours subscription is valid. **Minimum value:** 1	integer
id *optional*	Unique subscription identifier generated by the service.	string
start_date *optional*	Date from which subscription is active.	string (date-time)
stream_id *optional*	Unique identifier of the stream subscription is related to.	string
stream_owner *optional*	Unique identifier of the owner of the stream subscription is related to.	string
stream_url *optional*	URL of the stream generated by the proxy server.	string
user_id *optional*	Unique identifier of the user which subscribed to the stream.	string

Subscription Request

Name	Description	Schema
hours *required*	Subscription duration.	integer
asset_id *required*	Asset ID.	string

4.3 Security and privacy

The main problem with data exchange today are complex legal procedures combined with trust issues and data protection. The platform aims to resolve those issues using blockchain and providing extension points for different use cases. By default, platform does not store the data, but only assets that link to the real data source. This way, we avoid all the potential security issues related to storing potentially sensitive users' data. This also means that platform provides not only data exchange, but also that limited-time subscriptions to continuous data streams (potentially of infinite size) are supported. This is particularly useful for IoT use-cases where data is streamed directly by the edge gateways. The platform is extensible to support secure data storing if that's required by the use-case. Third-party solutions for data-at-rest can be employed to improve data privacy and protection.

The platform provides user management and authorization and employs best practices for protecting sensitive user data. Attribute-based access control (ABAC) is used to enable fine-grained permissions. ABAC represents using the internal properties of entitles within the system for authorization. The most common version of ABAC is role-based access control (RBAC): using an artificial attribute(s) called *role* in the system to evaluate access control rights. The platform uses roles for determination of high-level access rights and attributes to provide more fine-grained control. All the passwords in the platform are encrypted using one-way hashing methods to protect users' privacy.

Tokens and **Transactions** are used to verify data transfer when creating data subscriptions. Tokens are digital assets stored in the blockchain and used for trading assets. The transaction is any exchange of tokens, including funding, defunding, and renting digital assets. Blockchain is used to store tokens in the form of a smart contract and each transaction results in a change of token balance for each participating party, platform included (platform can take a configurable fee from each transaction – also using fee smart contract). While this concept reminds us of the real marketplaces where the real money is exchanged for real goods, tokens are not necessarily used as a monetization tool, but simply the asset that proves the transaction that results in enabling and disabling access to the virtual resource. Two main benefits of using blockchain as the underlying technology are security and trust that come from its decentralized ledger nature. Blockchain acts as a source of trust between parties. Blockchain is also used to store users' terms and conditions of dataset/stream - that way platform can track eventual changes in data usage terms and condition and provide its integrity which is very important for the end-user.

4.4 ML/AI orchestration

Once data exchange problem is resolved, the question remains of utilizing those data. Data consumer will usually use access to the data to perform different analysis in purpose such as predictive maintenance, cost reduction, different kinds of recommendation and optimization... These analyses come under the domain of artificial intelligence or machine learning. The platform resolves some of the most important data-related issues for ML/AI, providing an integration with different platforms for ML/AI workflow control and federation.

The main issues data scientists face is:

- Lack of quality data sources
- Data leaks
- Poor support for edge computing and stream processing
- Data has to be disclosed

4.4.1 Lack of quality data sources

Access to quality data is crucial for creating good AI/ML models. While platform itself does not provide any mechanisms to guarantee data quality per se (since it is content-agnostic), it provides an easy and simple way of data exchange, as well as data descriptions and terms and conditions that help data scientists find and access datasets of interest. Even data samples/examples can be posted to simplify this process.

4.4.2 Data leaks

There are three states of data:

- data at rest
- data in transit
- data in use

Data leaks can happen in any of these states by either exposing unprotected data to malicious users or compromising privacy by cracking data protection mechanisms. The platform by default does not store data and relies on third-party solutions for use-cases that require storing data and uses TLS to secure data in transport. Data in use is an active issue and is related to data disclosure explained in chapter 4.4.4.

4.4.3 Poor support for edge computing and stream processing

As we already mentioned in 4.3, the platform supports data streams that can be utilized by limited-time subscriptions. Implying provided ML/AI workflow orchestration, it is possible to create long-lasting data acquisition and/or processing based on data streams.

4.4.4 Data has to be disclosed

This is a very active problem that gets a lot of attention from researchers recently. Processing data requires known data formats and types. Providing ML/AI algorithms that are going to be performed against datasets while not disclosing algorithms themselves is especially complex. There are two modern approaches to it:

- hardware enclaves
- homomorphic encryption

Hardware enclaves are used to execute algorithms in specially designed hardware components. Both data and algorithms are encrypted up until execution starts inside the enclave. Data and algorithms are decrypted only inside enclaves and the output is also encrypted, so data never leaves the enclave unprotected.

Homomorphic encryption is a mathematical model that enables computation on encrypted data.

4.5 5G Security

5G cyber security follows the design principles of defense in depth, zero-trust, and adaptive security, which collaboratively provide a systematic, dynamic, and adaptive security framework. Defense in depth provides multi-layer security measures to protect critical internal assets from external threats. Defense in depth prevents system breakdown caused by attacks and unauthorized access. Information is encrypted, so even if it is stolen, no information leakage will occur. Malicious tampering can be identified so that mitigation measures can be taken accordingly.

Zero-trust is a security model built on the principle that no user or network function can be trusted, whether internal or external to the network. Zero-trust focuses on protecting resources, including assets, services, workflows, and accounts instead of protecting network segments. Zero-trust architecture (ZTA) is built upon the principles of zero-trust to minimize access to resources, such as data, compute resources, applications, and services, to only those subjects and assets identified as needing access, as well as continually authenticating and authorizing the identity and security posture of each access request.

For securing 5G networks this will be applied:

- 5G networks with a ZTA that is complemented with perimeter security to provide protection from internal and external threats.
- Implementation of 3GPP 5G standalone network to benefit from security enhancements that support a zero-trust architecture and follow CSRIC VII recommendations.
- Industry best practices for secure cloud deployments, including secure CNF, orchestration, automation, APIs, and infrastructure models.

3GPP has standardized 5G [33][34] and is introducing the following security improvements:

- Subscriber authentication: secure mutual authentication using 5G Authentication and Key Agreement (5G-AKA), Extensible Authentication Protocol Authentication and Key Agreement Prime (EAP-AKA'), and Extensible Authentication Protocol – Transport Layer Security (EAP-TLS), Home Control of authentication for roaming devices, and non-SIM card-based authentication for IoT devices
- Subscriber privacy: Stronger False Base Station (FBS) protection and Subscription Permanent Identifier/Subscription Concealed Identifier (SUPI/SUCI) for encrypted long-term subscriber identifiers. CSRIC VII recommends that the SUCI feature is mandatory for U.S. deployments, except when the UE is requesting emergency services.
- Secure service-based architecture (SBA): TLS and OAuth 2.0 on all mandatory functions
- Secure roaming interconnects: introduction of the Security Edge Protection Proxy (SEPP) at the application layer
- Non-Public Networks (NPN): 5G Private networks to provide security and privacy on dedicated resources that are independently managed.
- Use case specific security enhancements for cellular IoT and URLLC services.

5 DEDICAT 6G threat identification and classification mechanisms

Machine Learning and Deep Learning techniques are being used to help in the detection and the classification of cyberattacks and thus to develop an Intrusion Detection System (IDS) to protect the network from those threats.

5.1 Anomaly Intrusion Detection System

The objective of this work is to classify novel attacks by examining the structures of normal behaviour in network traffic while trying to improve detection accuracy and reduce the false-positive rate.

AIDS, anomaly-based detection, is one type of IDS, where network traffic is analysed to establish the normal behaviour of the network and then a model is trained based on that behaviour. The model has information for bandwidth utilization, protocols, the port used, IP addresses, etc. Furthermore, to detect if there is an anomaly/intrusion the network traffic is compared with the trained model to detect any deviation or anomaly. Despite its various benefits, AIDS has a certain shortcoming, the high rate of false positives. For this reason, it is better to use Deep Learning mechanisms and not simple Machine Learning techniques to alleviate this deficiency. As a result, this work intends to use a convolutional neural network (CNN) that will try to improve the result of false positives that traditional AIDS has been carrying. Additionally, different techniques are used to process and analyse the data set that we use (UNSW-NB15) to train the convolutional neural network.

An effective benchmark dataset to help us compare and create different intrusion detection methods is the UNSW-NB 15 [32]. The IXIA PerfectStorm tool in the Cyber Range Lab of UNSW Canberra created the raw network packets of the UNSW-NB 15 dataset for generating a hybrid of real modern normal activities and synthetic contemporary attack behaviors. This dataset has nine types of attacks, Fuzzers, Analysis, Backdoors, DoS, Exploits, Generic, Reconnaissance, Shellcode, and Worms, and 49 features with the class label. Furthermore, Deep Learning models, such as CNN combined with the Random Forest algorithm and PCA will be explored for the development of a flexible and effective AIDS to detect and classify unforeseen and unpredictable cyberattacks and to improve the false positives rate.

A CNN architecture will be used to build both binary and multiclass classification models for AIDS. CNNs have shown great success in various studies, can understand complex structures. And have the power to leverage spatial or temporal correlation in data. They have also been used for both feature extraction and classification in IDS. Their complexity is lower compared to other deep learning architectures as they require fewer parameters. Different trials of the architecture will be run, and the optimal will be chosen. Then will try to fine-tune the hyperparameters on this model (learning rate, number of epochs, activation functions, dropout rate).

In addition, the data will be transformed into a 2D format to be suitable for the deep learning architecture. The non-numeric features to numeric features using one-hot encoding, and all features will be normalized by subtracting the mean and scaling to unit variance. Furthermore, instead of relying only on hand-crafted features a hybrid two-step pre-processing method that combines dimensionality reduction and feature engineering is proposed. For the dimensionality reduction, principal component analysis (PCA) will be used on the continuous features, such that 95% of the variance will be retained, and Random Forests will help in feature engineering for selecting the most relevant features for better model accuracy.

Finally, to evaluate the model's performance test data will be used to make predictions, and metrics such as accuracy score and confusion matrix, Precision, F1-Score, and Recall will be used for the evaluation. The final results will also be compared with similar deep learning approaches and state-of-the-art classification models.

5.2 Intrusion detection in IoT traffic

In extending the SOTA, the ensemble ML approach (stacking), which involves combining multiple weak classifiers to form a strong learner (classifier), will address the classification underperformance recorded with single machine learning classifiers. Notably, the research will focus on finding the right combination and tuning the weak learners that will make up the strong learner to form the best classification performance. In our preliminary experiments, we have so far used Naïve Bayes, Logistic Regression (LR) and Decision Tree classifiers in building the stack classifier. The first two are the base learners and the latter as the meta-learner. The first two classifiers were selected because of their speed of classifying the different network categories, while the latter is based on the branching concept that is associated with using the 'Gini Index', which is a measure of inequality in samples to determine the best split for classification.

Additionally, feature engineering will be introduced as an add-on to the ensemble classifier to remove redundant features from the dataset without compromising the classification performance while at the same time increasing the classifier's speed and reducing the computational resources required for the classification task. Specifically, we will apply the feature engineering to reduce the features of the dataset to the barest minimum to achieve the performance enhancement task. Furthermore, a new machine learning approach called federated learning developed with other conventional ML classifiers aside from NN will be explored. This is because many of the federated learning classifiers researched so far are based on NN classifiers. Additionally, the training time and computational resources needed for the NN based federated learning is quite much due to the hidden layers and number of neurons they contain.

On the other hand, the conventional ML approaches do not use as many resources as the NN based federated learning, so we intend to solve these problems by using this approach without compromising the data integrity and classification performance. In doing this, we adopt LR as the classifier used in building the federated learner. This is because very efficient for classification and does not require as many computational resources as the variant classifiers of NN. Additionally, tuning and scaling of the training dataset input features are not required when using LR, thereby saving more time in the pre-processing stage, which contributes towards detecting network anomalies more swiftly. The steps in the federated learning are as follows:

i. Global model sends the latest model parameters to the nodes;
ii. Data is collected at each node;
iii. Feature engineering of the dataset and training of each local model is done using the latest parameters;
iv. The parameters of the updated model are communicated back to the global model;
v. Combine update from each model and retrain the global model which is a new model;
vi. Repeat all the processes from step 1 until an optimal level of classification performance is reached.

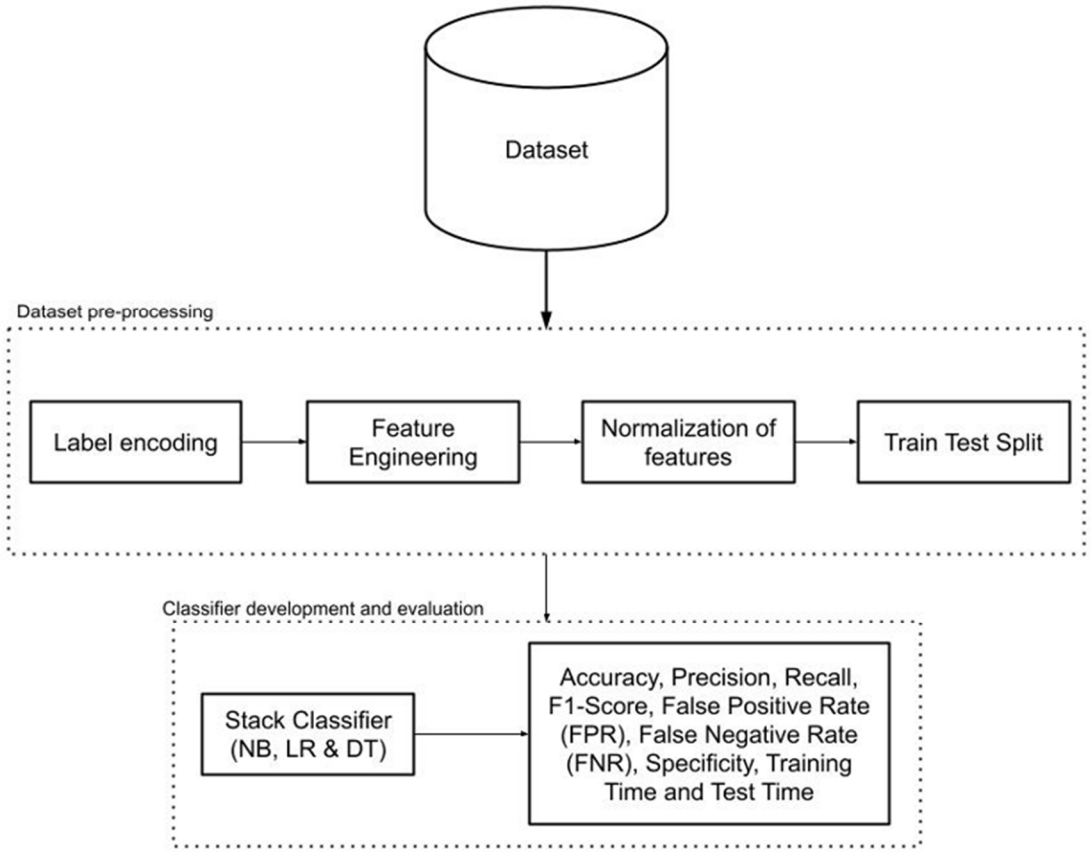

Figure 3: Stack

Figure 3 and Figure 4 show the graphical representation of the proposed stack ensemble learner approach and that of federated learning.

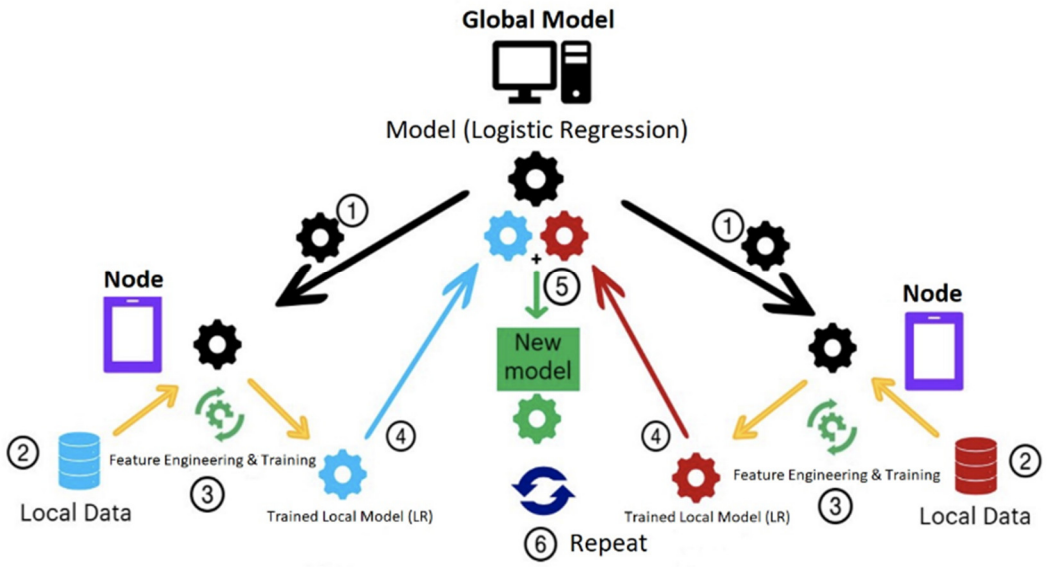

Figure 4: Federated Learning

5.3 Federated learning approach for threat analysis model

In order to increase security and privacy-preserving guarantees, the use of federated ML will be used in the proposed DEDICAT 6G framework. Federated ML allows to avoid data movement, thus not only does it contribute to increased security, but also to saving bandwidth and creating more efficient AI applications.

In order to achieve this, DEDICAT 6G specifies AI/ML orchestration on the top of the NDM-based data marketplace, following recommendations and best practices from Nokia research.

Federated AI/ML framework is based on a following architectural open-source components:

- Kubernetes;
- Kubeflow (https://www.kubeflow.org/);
- Argo Workflows (https://argoproj.github.io/argo-workflows/);
- KubeFATE (https://github.com/FederatedAI/KubeFATE);
- PySift (https://github.com/OpenMined/PySyft).

Additionally, federated ML approach tries to utilize Confidential Computing TEE enclaves whenever possible (i.e. on all nodes that support *Trusted Execution Environment* (TEE) hardware extensions).

Federated ML is done on the top of the decentralized marketplace. Data sets and algorithms that are advertised in the catalog can be used for execution of AI/ML computes. Data marketplace communicated directly to orchestration layer via APIs, providing configuration files with recipes that explain:

- Algorithm deployment strategies on the top of a Kubernetes cluster;
- Data set locations behind access control layer.

Orchestration layer is based on Kubernetes. It creates defined Docker containers and distributes them according to YAML configuration provided by the data marketplace via APIs. Docker containers contain AI/ML algorithms that will be sent towards the edge, where the data resides.

At the edge, Docker container will be set in that manner by the orchestration layer, that data will be mapped into it (via persistent volume mechanism, for example), based on the configuration provided by the marketplace. Data is mapped via a distributed and decentralized proxy, deployed in the same container. Proxy gets configuration directly from the orchestration layer, so it can prevent algorithm's access to the data at any time (if data was leased for example by time constraint via the data marketplace).

Very important feature of this framework is that the orchestration layer will bring back to data scientist only resulting trained model or inference results – never the raw data. This is because the Docker container in which computation (application of the algorithm on the mapped data) will be destroyed (data deleted and nulled), establishing in that way privacy-preservation and preventing raw data leakage.

5.4 DEDICAT 6G privacy and data protection approaches

All data exchange within the DEDICAT 6G system must be logged, and a framework based on an immutable blockchain database in the form of append-only log will be established. Framework specifies use of Hyperledger Fabric (https://www.hyperledger.org/use/fabric) blockchain released and maintained by Linux Foundation (https://www.linuxfoundation.org/) under open-source and patent-free Apache-2.0 license.

Logging in the blockchain is done through the:

- Blockchain transactions when quiring data access;
- Smart Contracts immutable (and thus un-hackable) code running in the blockchain itself (on all of the blockchain nodes simultaneously).

In order to establish strongly protected and formalized data access control, Nokia worked and contributed the architectural blueprint, proven in practice through several deployments of Nokia Data Marketplace product in industrial setups.

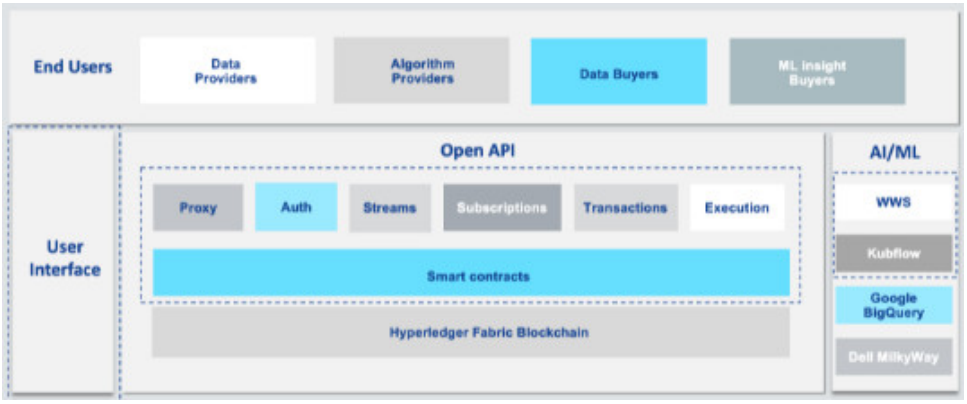

Figure 5: Nokia Data marketplace architecture diagram

NDM is based on a distributed microservice architecture, where several microservices can interface the blockchain layer to execute transactions. Main role of the framework is to allow data protection layer through APIs and blockchain Smart Contracts.

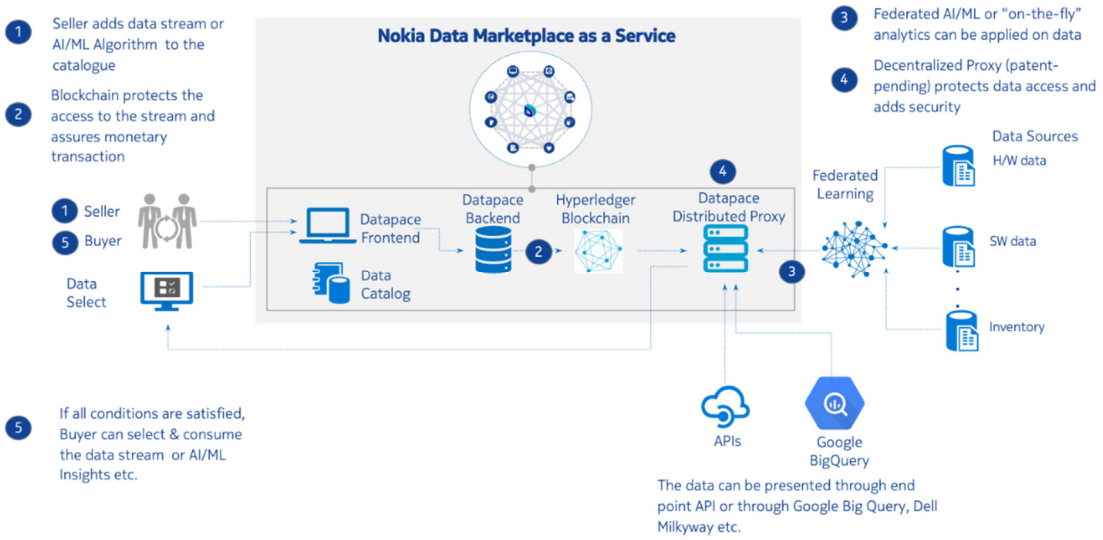

Figure 6: Nokia Data Marketplace functionality

All data sources are registered in the NDM catalog. They are described via various metadata, that describe either a data set or and algorithm. Data consumer browses a catalog and requests access to a selected data source. This request is translated into a blockchain transaction via APIs and stays recorded in a blockchain's immutable log. Upon a granted request (additional blockchain transaction), data consumer digitally signs Terms & Conditions contract, and the contents of the contracts are hashed into a subsequent blockchain block through an additional transaction.

Apart from pure data exchange (access control) function, DEDICAT 6G security framework will enable AI/ML orchestration on top, leveraging on open-source software for AI/ML orchestration and following recommendations and research done by Nokia and Bell Labs as an evolution of NDM product. This framework allows for execution of federated ML loads using Kubernetes-based orchestrator. Data sets and algorithms are accessed through data marketplace, assuring secure logging through blockchain transactions.

6 Trust management platform specification

One of key challenges and strong points of DEDICAT 6G is the application of blockchain for enhancing the trustworthiness of the system. DEDICAT 6G will integrate and deploy advanced AI based security and privacy protection framework with blockchain based trust management platform for securing the distributed network ecosystem, building trust between parties, devices and sub-systems, as well as providing intelligence for detecting and preventing potential security, privacy, and trust issues.

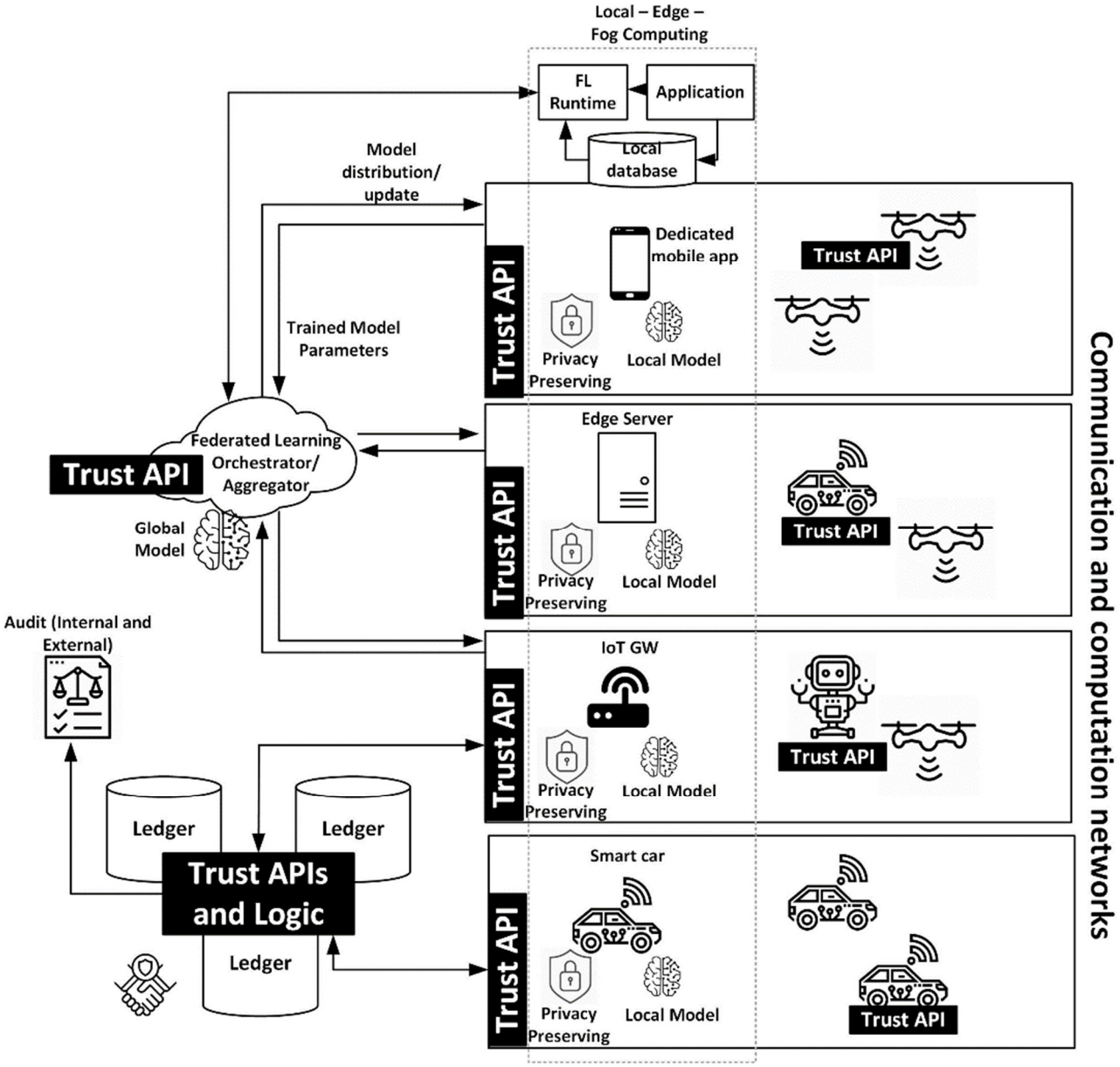

Figure 7: Federated learning for threat identification and trust management platform based on private permissioned blockchain

A trust management platform based on private permissioned blockchain, and a collection of smart contract templates will be implemented to facilitate trusted exchange of information and commands (including updates for local ML models) between nodes and systems participating in opportunistic communication and computation networks. This trust platform

will include consensus mechanisms and set of rules indicating who, when, what and under which conditions can read/write to the immutable record.

It will also include smart contracts supporting automated audits about performance of opportunistic communication and computation networks as well as automated compliance tests and certifications for candidate nodes and systems which are trusted to form and join federated learning for threat identification and trust management platform based on private permissioned blockchain.

6.1 Trustworthiness metrics

Trustworthiness metrics are calculated for edge nodes, processes, users and data streams. Trust metric value indicates if a node can join a local network, if process output can be further used, if a user can execute specific rule (figure 8). Trust metrics are implemented as ML models whose outputs are written on private permissioned blockchain through dedicated smart contracts. This way all stakeholders in DEDICAT 6G instance have access to immutable record of trust metrics calculated for all actors, resources and processes.

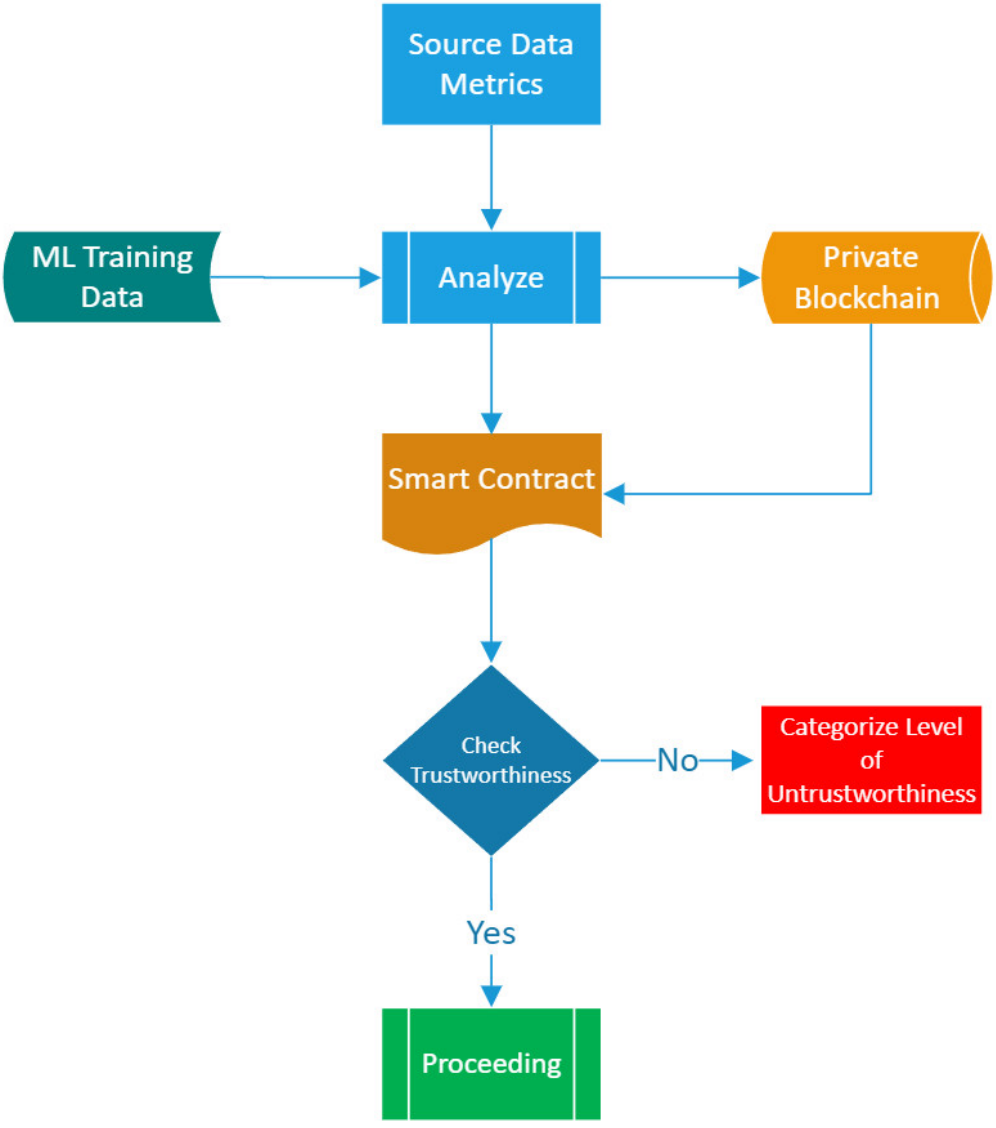

Figure 8: Flowchart of the model for trustworthiness metrics

Decision making processes of the DEDICAT 6G system must consult calculated trust metrics before proceeding with decision implementation or execution. The following trust metrics are envisioned for the DEDICAT 6G system:

1. *Device-based metrics* - capturing a device's state and feature set;
2. *Connection-based metrics* - define the connection types which are established in DEDICAT 6G systems;
3. *Behaviour-based metrics* - capturing the user's and device's behaviour within the observed network;
4. *Context-based metrics* - expected operations of a node in a known context. e.g. in case of failures;
5. *Composite metrics* - computed based on the weighted calculation of different security, reliability, safety and privacy metrics at the device, connection, behaviour, and application levels.

6.2 Trustworthiness levels

Different levels of device trustworthiness will be configured: relay node (only relaying encrypted information end extending range), bridge node (translating between systems and protocols), computation node (processing exchanged/collected data – including local federated learning entity), decision making node (acts on data analysis results) and orchestrator node (responsible for managing established networks). Each node type will have specific compliance test and will receive certificate which is updated based on performance of the node within networks.

The following trust levels will be established (with smart contracts):

- *Service-level trust: services within the DEDICAT 6G platform and its instances should be able to query reputation ratings for specific services;*
- *Data stream-level trust: reputation ratings for specific data stream (e.g. sensory reading);*
- *Device-level trust: services can query reputation rating of a device.*

The outcomes of the various analytics mechanisms are recorded by the Knowledge functional entity along with corresponding contexts and situations encountered (triggering those decisions), policies that were considered, the efficiency of the decisions and actions taken in terms of achieved power consumption, latency, QoS, cost, etc. Additional information stored by the Knowledge entity is related to security, privacy and trust issues identified and respective measures taken. Knowledge can be developed autonomously (e.g., by each *Mobile Access Point* (MAP) or edge node) and in a centralized, aggregated manner (in the "global" cloud).

6.3 DLT approach for trust management

Distributed Ledger Technology (DLT) and *Smart Contracts* (SC), due to its intrinsic properties of transparency, immutability, and underlying secure-by-design architecture, allows distributed, decentralized, automated workflows.
A Distributed Ledger is a synchronized, shared set of digital data replicated across multiple computers. Using a peer-to-peer network, cryptographic algorithms and consensus mechanisms, it guarantees decentralized data storage and control, and immutability.
The security and accuracy of assets stored in the ledger are maintained cryptographically using "keys" and signatures to control what can be done by whom within the shared ledger. Thus, any changes to the ledger are reflected in all copies in minutes, or in some cases, seconds.

DLTs generally integrate a number of innovations which include: database (ledger) entries that cannot be reversed or otherwise modified, the ability to grant granular permissions, automated data synchronization, rigorous privacy and security capabilities, process automation, and transparency, such that any attempts at changes to entries will notify others. Its primary disruptive attribute is that it is decentralized and therefore not dependent on a central controller or storer of the data.

6.4 Configuration of blockchain networks

Blockchain is a solution that forms a growing list (ledger) of immutable records (blocks) that are linked together to form a chain and propagate securely to participants. The participants are represented as peer nodes within a widely spanned network even across the Globe. This approach allows organizations to come to mutual agreement on a single, distributed source of truth. Every peer contains a copy of the ledger which is used to apply transactions in case they have been validated with consensus protocol. And each block is bound to the preceding one with a hash ensuring that the blockchain is resistant to its data modification.

Blockchain network is a technical infrastructure that provides ledger and smart contract (chaincode) services to applications. Primarily, smart contracts are used to generate transactions which are subsequently distributed to every peer node in the network where they are immutably recorded on their copy of the ledger. The users of applications might be end users using client applications or blockchain network administrators.

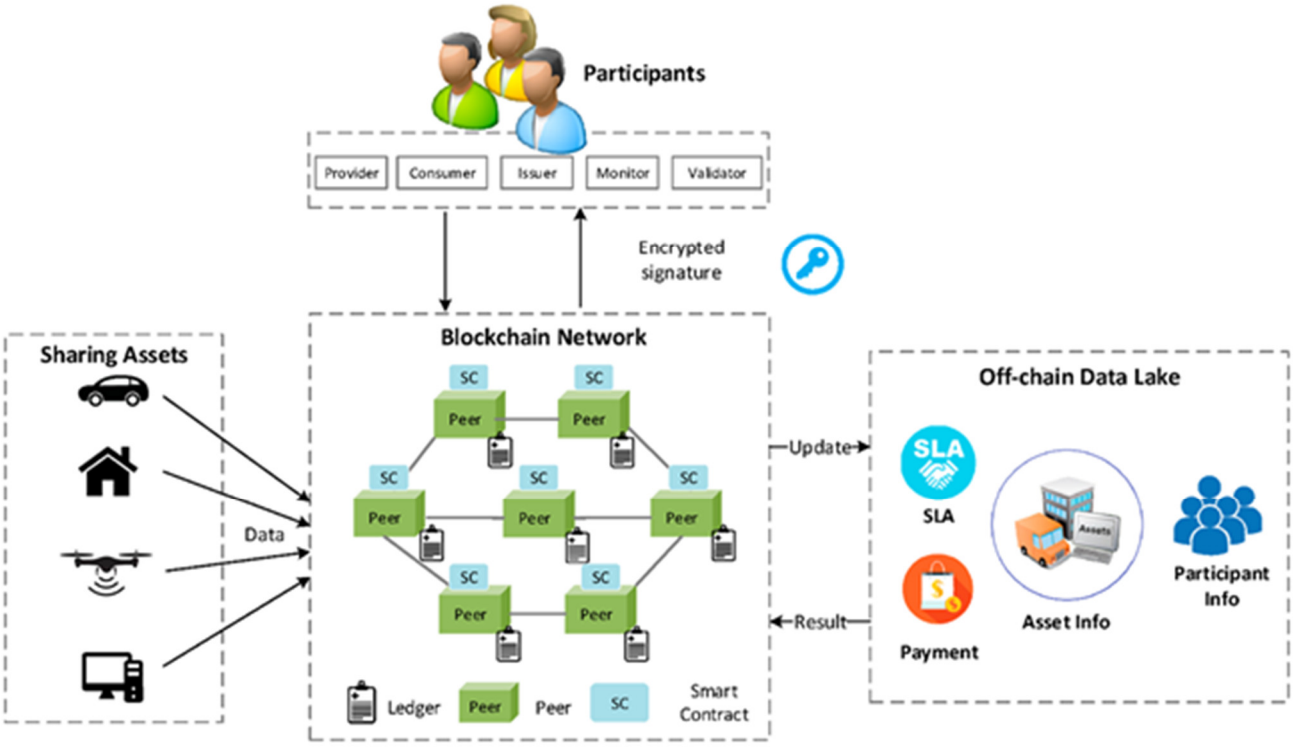

Figure 9: Blockchain network

The Hyperledger Architecture Working Group has singled out the following business components of Blockchain:

- Consensus Layer - Responsible for creating an order agreement and validating the set of transactions that make up the block;
- Smart Contract Layer - Responsible for processing transaction requests and determining the validity of transactions by applying business logic;

- Communication layer - Responsible for peer-to-peer transmission of messages between nodes participating in a common general ledger;
- Data Warehouse Abstraction - Allows different data warehouses to be used in other modules;
- Crypto Abstraction - Allows you to replace different crypto algorithms or modules without affecting other modules;
- Identity Services - Allows you to establish trust roots during Blockchain instance setup, enrol and register identities or system entities during network operation, and manage changes. They also provide authentication and authorization;
- Access Services - Responsible for managing various approaches specified in the system, such as the validation approach, the consensus approach, or the group management approach. They connect and depend on other modules to implement different approaches;
- Application User Interfaces - Allows clients and applications to connect to Blockchain.

Most important components of a Hyperledger Fabric blockchain are: the domain, the organizations, the peers, the orderers and the certificate authorities. Domain presents a top-level network and project name. The organizations are the containers for the peers and respective *Certificate Authorities* (CA). Each organization has its own CA and a list of peers. Usually, organizations are used for physical separation of the blockchain network where each organization who uses your product can set up their physical machines and join your network. The peers are nodes which are connected to clients and are responsible for committing transactions to the world state. Each peer has its own copy of transactions in a couchdb database. An organization can have more than one peer. The orderers are responsible for making sure that all the peers in the network have committed a transaction. When a transaction is proposed and committed by a peer, the orderer is informed about the new transaction and it forwards and commits this block to all adjacent peers. The certificate authority is responsible for creating users certificates. It is used for verifying ownership in the network. Each certificate authority is tied with an organization.

The VizLore federated Hyperledger Fabric system is deployed on two system levels - cloud/core level and edge/local level of a typical IoT system.

a) Core Hyperledger Fabric Blockchain Network - is the cloud level of the system holding core components for building and deploying distributed blockchain system on local IoT systems. Its' main components are:

- CouchDB datastore;
- Core Hyperledger Certificate Authority;
- Core Hyperledger Distributed Kafka Ordering Service;
- Hyperledger Fabric Native Binaries for Crypto material generation;
- Hyperledger Fabric core Peers;
- Core Blockchain channels.

The above-mentioned components are necessary infrastructural components for a fault tolerance functioning of the blockchain network. The below-mentioned components are additional components and are providing access and communication throughout the blockchain ecosystem.

- IoT Hyperledger Infrastructure configuration generator;
- Hyperledger Fabric Agent;
- Hyperledger Fabric chaincode generator.

b) Hyperledger Fabric Blockchain Networks deployed on local level – make the local level where actual psychical deployment of IoT systems and their gateway/controller devices takes place. Our platform supports multiple IoT system interconnection and enables their

communication with the Core Hyperledger Fabric Blockchain Network. Its' main components are:

1. CouchDB datastore;
2. Local Hyperledger Certificate Authority;
3. Hyperledger Fabric local Peers;
4. Local Blockchain channels.

At the core level (deployed on the Google Cloud Platform) system components are spreading over several virtual machines. The number of machines depends on the requirements of the network, number of IoT systems that are a part of it, sizes of the ledgers, etc. We are currently working on broadening the core Hyperledger network onto the Microsoft Azure platform instances. Cross platform hosting of the core Hyperledger nodes ensures that the core segment will be available in scenarios where one PaaS/IaaS provider experience issues.

The Core Hyperledger Certificate Authority, Core Hyperledger Distributed Kafka Ordering Service, Hyperledger Fabric core Peers and Hyperledger Fabric Native Binaries for Crypto material generation are necessary Hyperledger Fabric components that are needed for correct functioning of the blockchain network. Therefore, these modules are integral part of the VizLore IoT Hyperledger framework.

The Hyperledger framework introduces blockchain channels which represent logically separated blockchain applications with their own fabric, topology and blockchain access rules. One or more channels can be deployed on the core level. The channels contain deployed smart contracts (chaincodes in Hyperledger Fabric terminology), and that channel's ledger data is available through joining a channel. There can be more than one smart contract deployed on the same channel.

6.5 Smart contract templates

A smart contract, together with the ledger, form the heart of a Hyperledger Fabric blockchain system. Whereas a ledger holds facts about the current and historical state of a set of business objects, a smart contract defines the executable logic that generates new facts that are added to the ledger. A chaincode is typically used by administrators to group related smart contracts for deployment but can also be used for low level system programming of Fabric. In this topic, we'll focus on why both smart contracts and chaincode exist, and how and when to use them.
A smart contract defines the rules between different organizations in executable code. Applications invoke a smart contract to generate transactions that are recorded on the ledger. Hyperledger Fabric users often use the terms smart contract and chaincode interchangeably. In general, a smart contract defines the transaction logic that controls the lifecycle of a business object contained in the world state.
When a smart contract executes, it runs on a peer node owned by an organization in the blockchain network. Hyperledger Fabric allows an organization to simultaneously participate in multiple, separate blockchain networks via channels. By joining multiple channels, an organization can participate in a so-called network of networks. Channels provide an efficient sharing of infrastructure while maintaining data and communications privacy. They are independent enough to help organizations separate their work traffic with different counterparties but integrated enough to allow them to coordinate independent activities when necessary. Configured and validated smart contract templates for trusted data exchange within project use cases and automated auditing of the edge computing system status. Certification and automated compliance tests will be implemented and validated.
Figures 10. and 11. below are showing how smart contracts will be applied in the DEDICAT 6G environment:

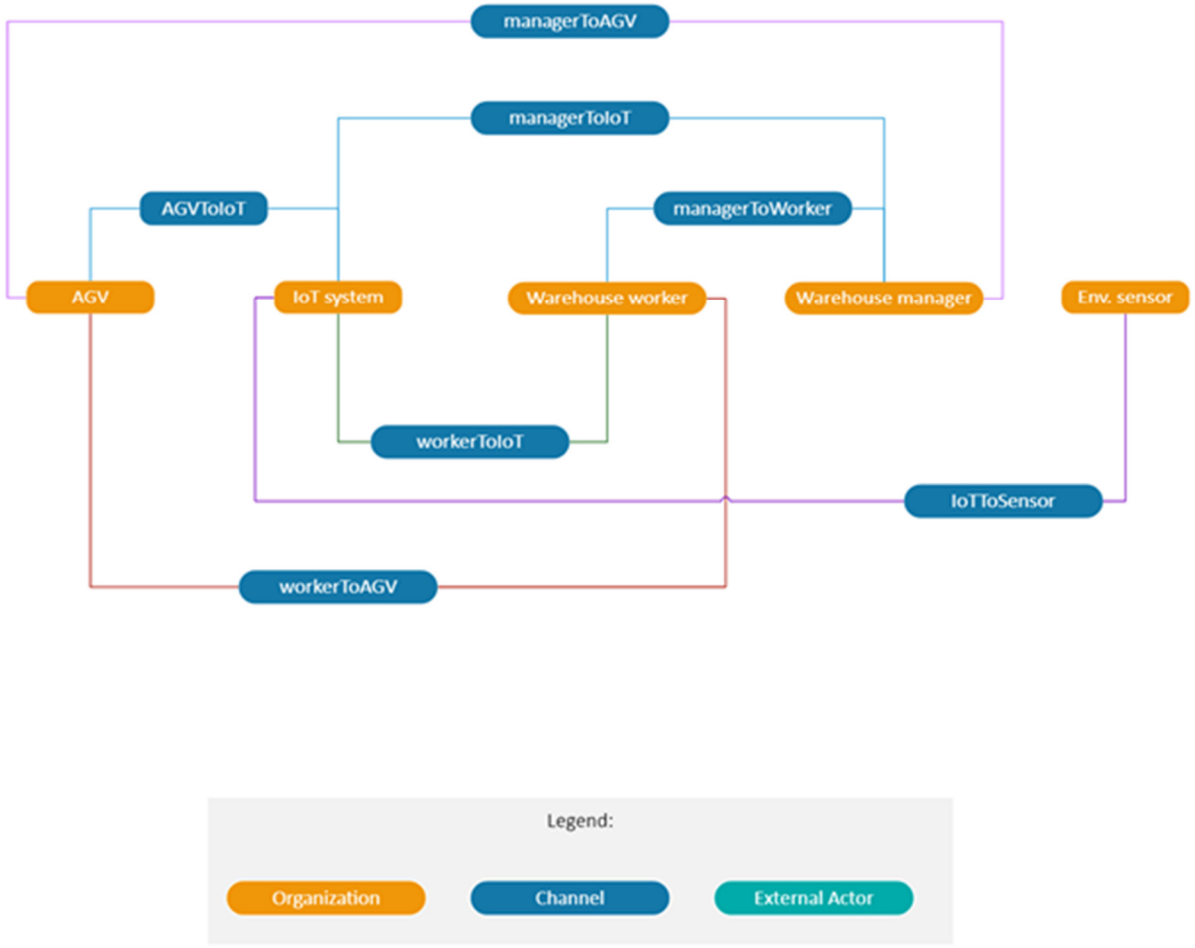

Figure 10: Smart Contracts (Organizations and Channels)

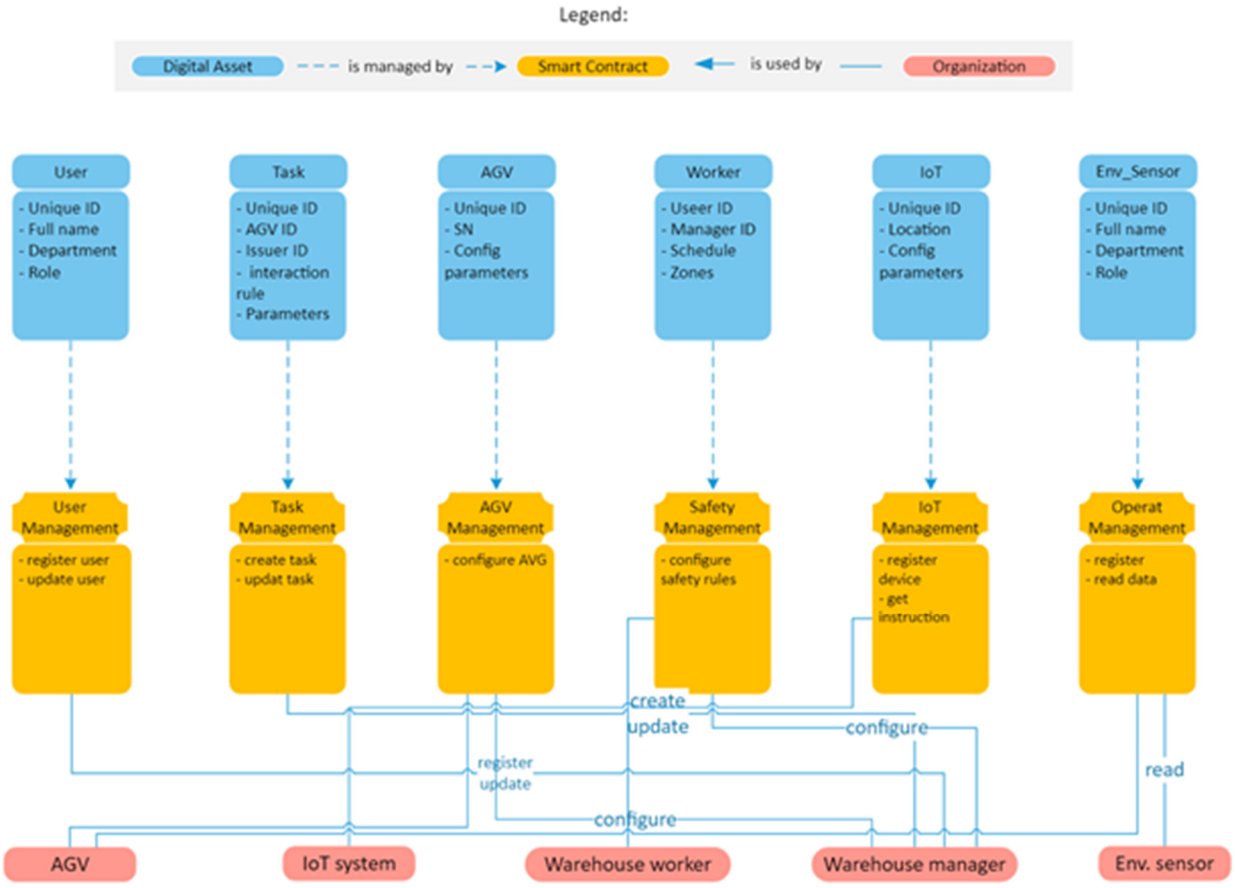

Figure 11: Smart Contracts

7 Conclusions

The documented provided an Initial specification of the DEDICAT 6G security and data protection framework and DEDICAT 6G trust management platform. The document also provided a short overview of the State-of-Art including anomaly intrusion detection system and threat analysis. Privacy, security and trust protection plans and strategies were addressed. Threat identification and classification mechanisms were described. Essentially the document is linked to the specification of the security, privacy and trust FCs of the DEDICAT 6G architecture, namely Audit FC, AuthN FC, AuthZ FC, Data marketplace FC, Threat Analysis FC, Distributed Ledger FC, Trust Metrics FC, IdM FC and Logging FC.

The next step is to implement the DEDICAT 6G security and data protection framework as well as blockchain based trust management platform and mechanisms, including federated learning and define strategy for integration and validation within project pilots.

References

[1] J.Rifkin, The Zero Marginal Cost Society: The Internet of Things the Collaborative Commons and the Eclipse of Capitalism, New York, NY, USA: Palgrave Macmillan, Apr. 2014.

[2] M. F. Elrawy, A. I. Awad, H. F. A. Hamed. Intrusion detection systems for IoT-based smart environments: A survey. J. Cloud Comput. **2018**, 7, 21.

[3] M. Almiani, A. AbuGhazleh, A. Al-Rahayfeh, S. Atiewi, and A. Razaque, Deep recurrent neural network for IoT intrusion detection system. Simul. Model. Pract. Theory **2020**, 101, 102031.

[4] S. J. Moore, C. D. Nugent, S. Zhang, and I. Cleland. IoT reliability: A review leading to 5 key research directions. CCF Trans. Pervasive Comput. Interact. **2020**, 2, 147–163.

[5] M. A. Ferrag, L. Shu, X. Yang, A. Derhab, and L. Maglaras. Security and Privacy for Green IoT-Based Agriculture: Review, Blockchain Solutions, and Challenges. IEEE Access **2020**, 8, 32031–32053.

[6] F.A. Ghaleb, M. A. Maarof, A. Zainal, M. Rassam, F. Saeed, and M. Alsaedi. Context-aware data-centric misbehaviour detection scheme for vehicular ad hoc networks using sequential analysis of the temporal and spatial correlation of the consistency between the cooperative awareness messages. Veh. Commun. **2019**, 20, 100186.

[7] M. AL-Hawawreh, N. Moustafa, and E. Sitnikova. Identification of Malicious Activities in Industrial Internet of Things Based on Deep Learning Models. J. Inf. Secur. Appl. **2018**, 41, 1–11.

[8] H. Liao, C. R. Lin, Y. Lin, K. Tung. "Intrusion detection system: A comprehensive review." Journal of Network and Computer Applications, 36, 16-24. 2013.

[9] M. Fahim, and A. Sillitti. Anomaly Detection, Analysis and Prediction Techniques in IoT Environment: A Systematic Literature Review. IEEE Access **2019**, 7, 81664–81681.

[10] Michalski, J. Carbonell, and T. Mitchell, "Machine learning: An artificial intelligence approach, tioga publ," Co., Palo Alto, CA, 1983.

[11] M. Almseidin, M. Alzubi, S. Kovacs, and M. Alkasassbeh, "Evaluation of machine learning algorithms for intrusion detection system," in 2017 IEEE 15th International Symposium on Intelligent Systems and Informatics (SISY). IEEE, 2017.

[12] W. Lee and S. J. Stolfo, "A framework for constructing features and models for intrusion detection systems," ACM transactions on Information and system security (TiSSEC), vol. 3, no. 4, pp. 227–261,2000

[13] M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, "A detailed analysis of the kdd cup 99 data set," in2009 IEEE symposium on computational intelligence for security and defense applications. IEEE, 2009, pp. 1–6.

[14] I. Obeidat, N. Hamadneh, M. Alkasassbeh, M. Almseidin, and M. AlZubi, "Intensive pre-processing of kdd cup 99 for network intrusion classification using machine learning techniques," 2019.

[15] H. Rajadurai and U. D. Gandhi, "A stacked ensemble learning model for intrusion detection in wireless network," NEURAL COMPUTING & APPLICATIONS, 2020.

[16] N. Moustafa, ToN-IoT Dataset, 2020, [online] Available: https://cloudstor.aarnet.edu.au/plus/s/ds5zW91vdgjEj9i.

[17] N. Koroniotis, N. Moustafa, E. Sitnikova, B. Turnbull, "Towards the development of realistic botnet dataset in the internet of things for network forensic analytics: Bot-iot dataset." Future Gener. Comput. Syst. 2019.

[18] A. R. Gad, A. A. Nashat and T. M. Barkat, "Intrusion Detection System Using Machine Learning for Vehicular Ad Hoc Networks Based on ToN-IoT Dataset," in *IEEE Access*, vol. 9, pp. 142206-142217, 2021.

[19] M. A. Ferrag, L. Maglaras, A. Ahmim, M. Derdour, and H. Janicke. RDTIDS: Rules and Decision Tree-Based Intrusion Detection System for Internet-of-Things Networks. Future Internet 2020, 12, 44.

[20] S. I. Popoola, B. Adebisi, R. Ande, M. Hammoudeh, K. Anoh, and A. A. Atayero," Smote-drnn: A deep learning algorithm detection in the internet-of-things networks," Sensors, vol. 21, no.9, p.2985, 2021.

[21] M. A. Ferrag, L. Maglaras, S. Moschoyiannis, and H. Janicke. Deep learning for cyber security intrusion detection: Approaches, datasets, and comparative study. J. Inf. Secur. Appl. 2020, 50, 102419.

[22] H. G. Kayacik, A. N. Zincir-Heywood, and M. I. Heywood, "Selecting features for intrusion detection: A feature relevance analysis on kdd 99 intrusion detection datasets," in Proceedings of the third annual conference on privacy, security and trust, 2005.

[23] Shijoe Jose et al 2018 J. Phys.: Conf. Ser. 1000 012049.

[24] Khraisat, A., Gondal, I., Vamplew, P. et al. Survey of intrusion detection systems: techniques datasets and challenges. Cybersecur 2, 20 (2019). https://doi.org/10.1186/s42400-019-0038-7

[25] L. Xiao, X. Wan, X. Lu, Y. Zhang and D. Wu, "IoT Security Techniques Based on Machine Learning: How Do IoT Devices Use AI to Enhance Security?," in IEEE Signal Processing Magazine, vol. 35, no. 5, pp. 41-49, Sept. 2018, doi: 10.1109/MSP.2018.2825478.

[26] Ullah and Q. H. Mahmoud, "Design and Development of a Deep Learning-Based Model for Anomaly Detection in IoT Networks," in IEEE Access, vol. 9, pp. 103906-103926, 2021, doi: 10.1109/ACCESS.2021.3094024.

[27] M. Panda, A. Abraham, and M. R. Patra, "A hybrid intelligent approach for network intrusion detection," Procedia Engineering, vol. 30, pp. 1–9, 2012.

[28] Bajaj K, Arora A (2013) Dimension reduction in intrusion detection features using discriminative machine learning approach. IJCSI International Journal of Computer Science Issues 10(4):324–328

[29] N. Aboueata, S. Alrasbi, A. Erbad, A. Kassler and D. Bhamare, "Supervised Machine Learning Techniques for Efficient Network Intrusion Detection," 2019 28th International Conference on Computer Communication and Networks (ICCCN), 2019, pp. 1-8, doi: 10.1109/ICCCN.2019.8847179.

[30] Z. Chkirbene, S. Eltanbouly, M. Bashendy, N. AlNaimi and A. Erbad, "Hybrid Machine Learning for Network Anomaly Intrusion Detection," 2020 IEEE International Conference on Informatics, IoT, and Enabling Technologies (ICIoT), 2020, pp. 163-170, doi: 10.1109/ICIoT48696.2020.9089575.

[31] R. Vinayakumar, M. Alazab, K. P. Soman, P. Poornachandran, A. Al-Nemrat and S. Venkatraman, "Deep Learning Approach for Intelligent Intrusion Detection System," in IEEE Access, vol. 7, pp. 41525-41550, 2019, doi: 10.1109/ACCESS.2019.2895334.

[32] https://research.unsw.edu.au/projects/unsw-nb15-dataset

[33] 3GPP, Specification 23.502 "Procedures for the 5G System (5GS)", Release 15, 3GPP, 2017.

[34] 3GPP TS 23.791 V1.0.0, "Architecture enhancements for 5G System (5GS) to support network data analytics services; (Release 16)", June 2019.