fGrant Agreement No.: 952697
Call: H2020-SU-ICT-2018-2020
Topic: SU-ICT-02-2020
Type of action: RIA

D1.4 REPORT ON SECURITY, PRIVACY & ACCOUNTABILITY MODELS FOR DYNAMIC TRUSTED CONSENT & DATA SHARING

Revision: v.1.0

Work package	WP 1
Task	Task 1.5
Due date	31/08/2021
Submission date	15/09/2021
Deliverable lead	TUDE
Version	1.0
Authors	Kaitai Liang (TUDE)
Reviewers	Liqun Chen (SURREY) Sotiris Koussouris (SUITE5)
Abstract	Deliverable D1.4 focuses on the definition of the security, privacy, accountability and trust requirements of ASSURED towards providing secure data sharing between all involved stakeholders that comprise the complex ecosystem of the next-generation smart-connectivity “Systems-of-Systems”. Detailed data sharing behaviors, capturing all the layers of the supply chain, and data flow models are outlined and instantiated in the context of the envisioned use cases, for all type of data encountered (operational, security and threat intelligence). State of the Art analysis of consent management, use of distributed ledgers and smart contracts for data management are also provided coupled with preliminary insights on ASSURED value propositions and envisioned approaches towards trustworthy data sharing.
Keywords	Data Sharing, Blockchain, Data Flow Profiles, Security, Privacy and Trust

Document Revision History

Version	Date	Description of change	List of contributors
V0.1	01.05.2021	ToC and Questionnaire circulated to use case partners regarding data sharing requirements	Kaitai Liang (TUDE)
V0.2	19.05.2021	Data sharing requirements and data flow models (Chapter 3) in the context of the use cases – 1st Draft	Kaitai Liang (TUDE), Thanassis Giannetsos (UBITECH), Karthik Shenoy, Shantanoo Desai (BIBA), Stelios Basagiannis, Riccardo Orizio (UTRCI), Nikos Drosos (SPH), Dimitra Tsakanika, Ilia Christantoni (DAEM)
V0.3	31.05.2021	Definition of security, privacy, accountability and trust requirements and ASSURED value propositions towards trustworthy data sharing (Chapter 2)	Kaitai Liang (TUDE), Thanassis Giannetsos, Dimitris Papamartzivanos (UBITECH), Edlira Dushku (DTU)
V0.4	18.06.2021	Definition of data sharing models (based on questionnaire) and mapping to concrete security and privacy requirements (Section 3.2)	Kaitai Liang (TUDE), Thanassis Giannetsos (UBITECH), Karthik Shenoy, Shantanoo Desai (BIBA), Stelios Basagiannis, Riccardo Orizio (UTRCI), Nikos Drosos (SPH), Dimitra Tsakanika, Ilia Christantoni (DAEM)
V0.5	30.06.2021	SotA analysis of trusted consent management approaches (Chapter 4)	Liqun Chen, Nada El Kassem (SURREY), Jingru Wang (TUDA), Kaitai Liang (TUDE), Ilias Aliferis (UNISYSTEMS)
V0.6	16.07.2021	ASSURED mechanisms, based on the use of trusted computing, towards trusted consent management and secure data sharing (Section 4.2)	Liqun Chen, Nada El Kassem (SURREY), Jingru Wang (TUDA), Sotiris Koussouris, Stefanos Venios (S5), Dimitris Papamartzivanos, Thanassis Giannetsos (UBITECH), Ilias Aliferis (UNISYSTEMS), Kaitai Liang (TUDE)
V0.7	18.08.2021	Final description of ASSURED fit-in to the envisioned use cases based on the use of policy-compliant Blockchain structures and cryptographic primitives for the secure on- and off-chain data management (Chapter 5)	Kaitai Liang (TUDE), Thanassis Giannetsos (UBITECH), Jingru Wang (TUDE), Edlira Dushku (DTU)
V0.8	27.08.2021	Final description of data sharing models and preliminary insights on how ASSURED Blockchain and DLT technologies can fit in the use cases	Kaitai Liang (TUDE), Dimitris Papamartzivanos, Thanassis Giannetsos (UBITECH), Edlira Dushku (DTU)
V0.9	07.09.2021	Review the document	Liqun Chen (SURREY), Sotiris Koussouris (SUITE5)
V1.0	13/09.2021	Finalization of the document	Kaitai Liang (TUDE), Jean-Baptiste Milon (MARTEL)

DISCLAIMER

The information, documentation and figures available in this deliverable are written by the "Future Proofing of ICT Trust Chains: Sustainable Operational Assurance and Verification Remote Guards for Systems-of-Systems Security and Privacy" (ASSURED) project's consortium under EC grant agreement 952697 and do not necessarily reflect the views of the European Commission.

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

COPYRIGHT NOTICE

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

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

DEM: Demonstrator, pilot, prototype, plan designs

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

OTHER: Software, technical diagram, etc.

EXECUTIVE SUMMARY

Following the compilation of Deliverable D1.2 [99] which focuses on the definition of the general ASSURED architecture, the ASSURED consortium has identified the core technical components and how these components work and interact with each other in the context of "Systems-of-Systems" for enhanced **operational assurance and (operational and security-related) data sharing**. Understanding the functionalities and requirements for the general technical framework is able to help the consortium to narrow down the focus towards safeguarding the data flow and data sharing among the envisioned use cases.

Towards this direction, deliverable D1.4 **specifies and models (threat intelligence and operational) data sharing behaviors** among all ASSURED parties and stakeholders based on defined **trusted consent activities** between them, to be enforced through the ASSURED Blockchain infrastructure and trust anchors. It covers both: (i) **operational data** originating from the deployed Cyber-Physical Systems-of-systems (CPSoS) that have strict *trustworthiness* requirements, and (ii) **threat intelligence data/evidence** based on the attestation policies to be enforced. This set of data sharing behaviors are also mapped to the envisioned ASSURED use cases (*Smart Manufacturing, Smart Cities, Smart Aerospace and Smart Satellites*) that will serve as the basis for the extraction of the complete set of **security, privacy and trust requirements** that need to be achieved by the provided functionalities throughout the entire data lifecycle; from the **trust on agreement on registration and data sharing/collection to storage and use of data**. These requirements will help to guide the path towards the concrete design of the ASSURED Blockchain infrastructure and data sharing related components, as defined in D1.2 [99]; i.e., ASSURED Distributed Ledger Technology (DLT) Engine, Trusted Platform Module (TPM)-based Wallet, Smart Contract Composer and Data Storage Engine. Essentially, this deliverable forms the basis for the further modelling and implementation of the modeled data sharing behaviors via the use of smart contracts (to be defined in WP4) for capturing the [105]: (i) **enforcement of attestation policies through their conversion into smart contract logic** (performed by the ASSURED Security Context Broker [99]) and their further deployment/sharing, to the CPSoS, through the distributed ledgers, (ii) monitoring of the corresponding attestation output and its auditable recording to attestation history chains on the ledger [109], and (iii) sharing of both operational and threat intelligence data with other data collectors [110].

The **related parties and necessary data type and structure definitions** are first introduced from the four use cases. Based on those, a general data flow framework following by the specific descriptions for the use cases are well defined in the deliverable. Clear data flows, following a layered architecture, across edge devices to cloud-based backend, are captured. Besides, the internal/external data sharing, threat information sharing behaviors and models are also explained in detail. From the above, one can clearly see how the data sharing behaviors in the use cases perform and how these behaviors interact with the ASSURED components. Beyond that, the security and privacy requirements, for each use case, are concretely defined.

The overview of the conceptual architecture and workflow of actions that need to take place during a data sharing transaction, are outlined, including the data flow envisioned within ASSURED between the participating entities, combined with Smart Contracts. The initial description of the security and crypto primitives for secure data management is also put forth. This sets the scene for the concrete functionalities and algorithms of ASSURED on- and off-chain data functionalities towards enhanced data privacy as will be investigated in the context of WP4. These mechanisms rely on the:

Funded by the Horizon 2020
Framework Programme of the European Union

I. project's value proposition with the exact functionalities to be provided (when it comes to data privacy and anonymization) including also the conceptual architecture and workflow of actions;

II. **security and trust bundles** for data privacy and conveyance of data, namely **Attributed-based Encryption (ABE)** and **Direct Anonymous Attestation (DAA)**, as well as

III. the integration of such security and trust bundles on top of the ASSURED DLTs.

The overall purpose of this deliverable is to provide a reference document on the security and privacy-preserving trust anchors that have been selected by the consortium for integration in the overall ASSURED platform towards achieving the main vision of secure operational and security-related data sharing services. This will be used as input to the platform's detailed Blockchain architecture definition (D4.1), the functionality of platform's security sub-components and the further investigation, design and development of the core ASSURED security, privacy and trust bundles.

TABLE OF CONTENTS

Funded by the Horizon 2020
Framework Programme of the European Union

LIST OF FIGURES

Funded by the Horizon 2020 Framework Programme of the European Union

LIST OF TABLES

Funded by the Horizon 2020
Framework Programme of the European Union

1. INTRODUCTION

1.1 SCOPE AND PURPOSE

The main focus of this deliverable is to provide a comprehensive overview of the **data flows and data sharing behaviors and models**, encountered in supply chain ecosystems comprising heterogeneous CPSoS, between both **internal and external (to the supply chain) stakeholders**. In this context, the large amount of data generated by the executed mixed-criticality services increases the risk to **data security and privacy** – especially considering the type of data managed within such environments that may cover both **operational data but also threat intelligence information**. For instance, as described in D1.3, one of the core value propositions of ASSURED, with respect to safety and security, is that components must be enabled to make and prove statements about their state and actions so that other components can align their actions appropriately and an overall system state can be assessed and security policies can be evaluated and enforced. This requires the execution and **sharing of security attestation related data so that trust-aware service graph chains can be achieved** where any stakeholder can verify the correct state of a specific device prior to establishing a communication path.

While the creation of such data sharing ecosystems is currently been proposed through the integration of Blockchain technology into IoT applications [107], [108], most of the existing systems use the Blockchain to simply store access control policies, thus, underutilizing the power of Blockchain technology. There is an urgent need for the creation of **digital data semantic marketplaces where all interested stakeholders can securely interact with each other towards leveraging and learning from the unprecedented amount of data available**. Doing so will heavily contribute to the improvement of all business processes that are part of the entire supply chain. But to materialize such enhanced data sharing, there is one crucial challenge (overarching all others) and that is **lack of trust**. *Most people believe that information is a valuable commodity but is of no use if we cannot trust the source or organize it in a meaningful way*.

ASSURED, as described in D1.1 [98], meets these requirements by providing **secure, trusted and auditable data sharing environments for a new generation of policy-compliant Blockchain structures enhanced with advanced on- and off-chain data and knowledge management services** through the specification of novel TPM-based security and privacy-preserving protocols. The vision is to enable data confidentiality, integrity and multi-level access control (**security by design**), data ownership safeguarding (**privacy by design**), data provenance and sovereignty checking and trusted consent management, while respecting prevailing GDPR legislation.

Towards this direction, D1.4 **specifies and models (threat intelligence and operational) data sharing behaviors** among all ASSURED parties and stakeholders based on defined **trusted consent activities** between them, to be enforced through the ASSURED Blockchain infrastructure and trust anchors. It covers both: (i) **operational data** originating from the deployed CPSoS that have strict *trustworthiness* requirements, and (ii) **threat intelligence data/evidence** based on the attestation policies to be enforced. This set of data sharing behaviors are mapped to the envisioned use cases (*Smart Manufacturing*, *Smart Cities*, *Smart Aerospace and Smart Satellites*) that will serve as the basis for the extraction of the complete set of **security, privacy and trust requirements** that need to be achieved by the provided functionalities throughout the entire data lifecycle; from the **trust on agreement on registration and data sharing/collection to storage and use of data**. These requirements will help to guide the path towards the concrete design of the ASSURED Blockchain infrastructure and data sharing related components, as defined in D1.2 [99]; i.e., ASSURED DLT Engine, TPM-based Wallet, Smart Contract Composer and Data Storage Engine.

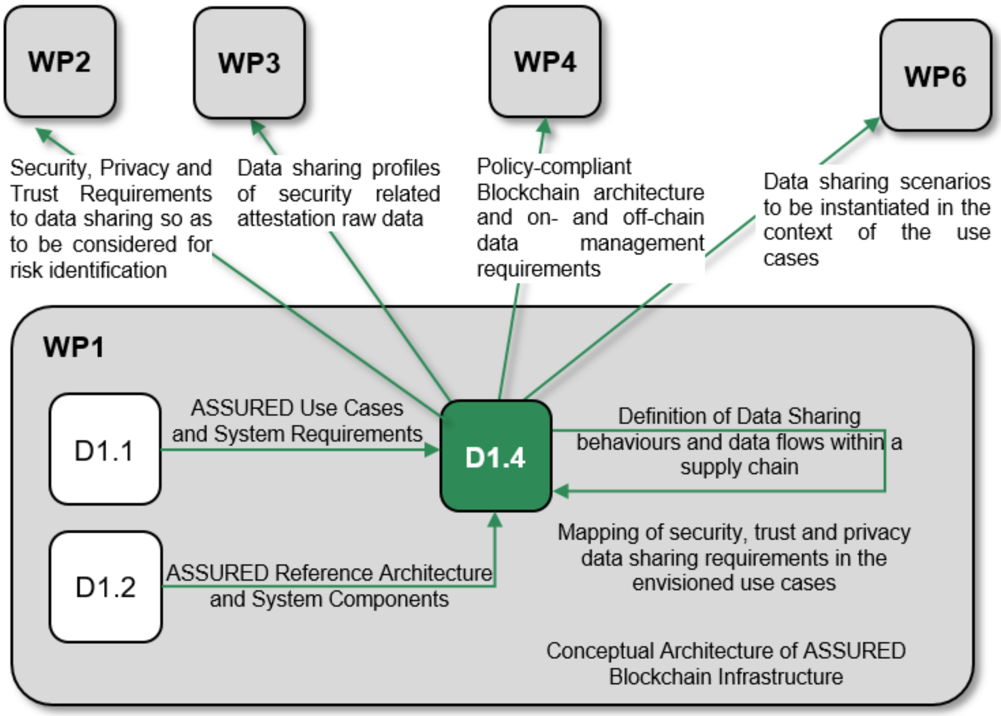

WP2
WP3
WP4
WP6
Security, Privacy and Trust Requirements to data sharing so as to be considered for risk identification
Data sharing profiles of security related attestation raw data
Policy-compliant Blockchain architecture and on- and off-chain data management requirements
Data sharing scenarios to be instantiated in the context of the use cases
WP1
D1.1
ASSURED Use Cases and System Requirements
D1.4
Definition of Data Sharing behaviours and data flows within a supply chain
D1.2
ASSURED Reference Architecture and System Components
Mapping of security, trust and privacy data sharing requirements in the envisioned use cases
Conceptual Architecture of ASSURED Blockchain Infrastructure

FIGURE 1: *RELATION OF D1.4 WITH OTHER WPS AND DELIVERABLES*

Essentially, this deliverable forms the basis for the further modelling and implementation of the modelled data sharing behaviors via the use of smart contracts (to be defined in WP4) for capturing the [105]: (i) **enforcement of attestation policies through their conversion into smart contract logic** (performed by the ASSURED Security Context Broker [99]) and their further deployment/sharing, to the CPSoS, through the distributed ledgers, (ii) monitoring of the corresponding attestation output and its auditable recording to attestation history chains on the ledger [109], and (iii) sharing of both operational and threat intelligence data with other data collectors [110].

Deliverable D1.4 also presents an investigation on the technical-level literature review on trust and Blockchain based technologies used for data access and knowledge management. This overview will provide technical options for ASSURED system components development. The deliverable further includes an initial vision of the combination of advanced techniques within ASSURED framework to capture the aforementioned security, privacy and trust requirements in the defined data sharing models. In summary, starting at the definition of parties and roles related to data sharing, the deliverable captures precise descriptions of data sharing models for each ASSURED use case, and further maps the models into security requirements. Beyond that, it exams potential and possible solutions for the requirements and outputs a general conceptual vision within the ASSURED context for secure data sharing. *We note that some concrete details of the vision and technical viewpoints have not been fully decided and confirmed in this stage and will be further elaborated in D4.1 where the final version of the ASSURED Blockchain conceptual architecture will be described*.

Funded by the Horizon 2020 Framework Programme of the European Union

1.2 RELATION TO OTHER WPS AND DELIVERABLES

With the definition of the security, privacy and trust requirements in the data sharing context of each one of the envisioned use cases, along with the state-of-the-art analysis of the most prominent trust and Blockchain-based data sharing technologies, this deliverable (D1.4) contributes a meaningful and useful technical roadmap and guidance for the development of the ASSURED Blockchain-based control services for the secure on- and off-chain data management and knowledge extraction. Figure 1 depicts the direct and indirect relationship of the D1.4 to the other Work Packages (WPs). The definitions of the overall high-level Blockchain-based architecture of ASSURED that needs to be achieved for enabling the envisioned data sharing behaviors, will drive the technical work of WP2 and WP4 towards the correct definition and deployment of smart contracts and the design and provision of the appropriate trusted Blockchain control services, respectively.

Within WP1, the current document directly gets input from D1.2 (with the clear definition of ASSURED conceptual and technical architecture), and further produces inputs to WP2 and WP3 (towards the definition and creation of the appropriate smart contract that can reflect the attestation enforcement business logic and threat intelligence sharing), WP4 (Blockchain-based control services & security context broker) and WP6 that undertakes the demonstration of the ASSURED Blockchain infrastructure in the context of the pilots.

More specifically, D1.4 acts as a starting point of technical reference for the later technical WPs: It provides technical options for policy enforcement into smart contract for data sharing which is in line with the upcoming developments of WP2. WP3 combines the design in trusted hardware and attestation which will be used to cover the attestation and trust auditing on data sharing. D1.4 proposes a general technical vision for Blockchain-based data sharing that will provide a step-stone for the Blockchain-enabled components and security context broker development in WP4. At last, the data sharing behaviors based on defined trusted consent activities will be enforced by the ASSURED SIADE component in WP5.

1.1 DELIVERABLE STRUCTURE

This deliverable is structured as follows. In Chapter 2, we first define the key parties and roles which are encountered in the ASSURED ecosystem and the types of data that will be considered for secure and privacy-preserving sharing. Furthermore, its puts forth the full list of security, privacy and trust requirements that need to be considered for achieving the value proposition of ASSURED towards the creation of a secure digital data semantic marketplace. We then proceed with Chapter 3 where the focus is on detailing the concrete data sharing behaviors and models for the envisioned use cases, so as to capture the security and privacy requirements and understand how we should further interact with the ASSURE components securely within the data sharing context. In Chapter 4, we review the key technologies that will become the foundational stones on the ASSURED technical development for data sharing. We provide an overview of the current state-of-the-art in trust, security and privacy technical solutions via the use of trusted hardware and Blockchain to safeguard data access, management and sharing in the “Systems-of-Systems”-enabled supply chains. In Chapter 5, we outline the general vision of how the ASSURED technical components could be fit-in for all the use cases in data sharing environment so as to achieve all the security and privacy requirements given by the use cases’ partners. Finally, Chapter 6 concludes the report and provides a summary of the presented work.

Funded by the Horizon 2020
Framework Programme of the European Union

2 ENHANCED DATA SHARING IN "SYSTEMS-OF-SYSTEMS" SUPPLY CHAIN

As aforementioned, one of the main visions of ASSURED is to provide a **secure, trusted, auditable and privacy-preserving platform** for operational and security-related **data sharing** functionalities. This is achieved through the design and implementation of policy-compliant Blockchain structures to be enhanced with advanced on- and off-chain data and knowledge management services through the specification of appropriate security services including access control, smart contract composition (reflecting the appropriate data sharing configurations), trusted consent management, membership authentication, trusted ledger and identify management (based on the use of trust anchors) as well as privacy-preserving services.

In this context, ASSURED will leverage a combination of private and public distributed ledgers (**Error! Reference source not found.**) as the Blockchain-powered infrastructure that will facilitate the sealing of (attestation) smart contracts on the side of the edge devices. All parties will be putting information and data, as transactions whereas external stakeholders will record any data sharing, and further record them on the ledgers to achieve information sharing with all nodes that will be granted access rights.

In what follows, we will give a high-level overview of the ASSURED Blockchain conceptual architecture – more details can be found in Chapter 5. As described previously, the goal is to leverage this initial architecture for then further specifying advanced protocol interfaces [105] towards: (i) **Integrity and verification of block data** for guaranteeing that stored data has not been tampered with, (ii) **Mining validation** for ensuring that a block mined by a user/device is valid, (iii) **Trusted Consent Management and Consensus agreement** for allowing a majority or all network users to reach an agreement on block or ledger validation, (iv) **Membership authentication** for providing access control mechanisms (read & write privileges) to authenticated users of the ledgers, (v) **Undeniable actions commitment** for guaranteeing indisputable user operations over the ledgers, and (vi) **Customized block data security** for enabling users to put various levels of encrypted metadata onto the ledgers.

Towards this direction, however, it is important to first define the roles and parties which are involved into the trust consent and data sharing behaviours in all the use cases. Thus, we proceed with presenting the **data flow, data sharing and threat information sharing models**.

2.1 ASSURED VALUE PROPOSITION AND DATA SHARING

As has been described in the context of WP1 deliverables [98], [99], the formation of secure CPSoS-enabled supply chains, based on the use of Blockchain Distributed Ledgers for enhanced data sharing and security policy enforcement, is considered one of the main value proposition of ASSURED. However, besides only security, **privacy** is also considered one of the core requirements that must be managed efficiently - especially in the context of the data value chains. Taking into consideration that ASSURED data value chains will form pipelines of operational and security-related (attestation) sensitive data (Section 2.2.1), it becomes clear that various data security and user privacy implications come into play and it is imperative to build new **on- and off-chain data management models and services** of privacy and data protection and the technologies that encode them.

In this direction, ASSURED enables enhanced **data privacy,** ownership safeguarding and **data provenance and sovereignty** checking mechanisms (more details on the comprised

Funded by the Horizon 2020
Framework Programme of the European Union

technical components can be found in Chapter 5). The platform uses Blockchain-based distributed ledgers for offering enhanced data and transaction security.

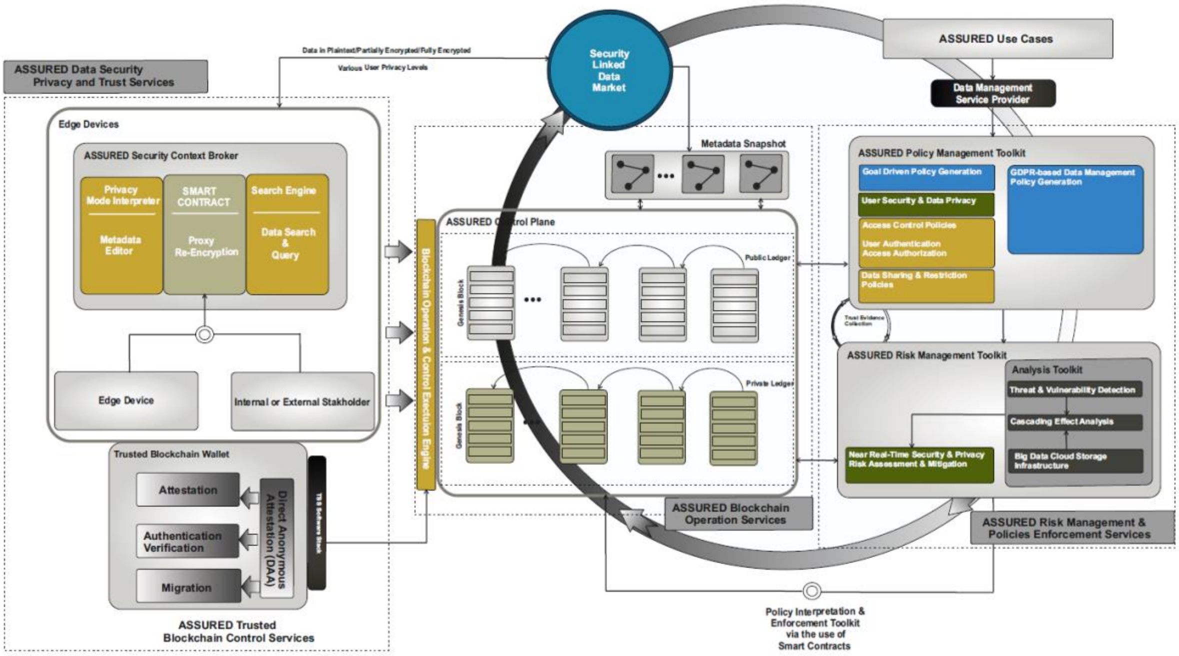

ASSURED Data Security Privacy and Trust Services
Data in Plaintext/Partially Encrypted/Fully Encrypted
Various User Privacy Levels
Security Linked Data Market
ASSURED Use Cases
Data Management Service Provider
Edge Devices
ASSURED Security Context Broker
Privacy Mode Interpreter
Metadata Editor
SMART CONTRACT
Proxy Re-Encryption
Search Engine
Data Search & Query
Edge Device
Internal or External Stakholder
Trusted Blockchain Wallet
Attestation
Authentication Verification
Migration
Direct Anonymous Attestation (DAA)
TSS Software Stack
ASSURED Trusted Blockchain Control Services
Blockchain Operation & Control Execution Engine
Metadata Snapshot
ASSURED Control Plane
Genesis Block
Public Ledger
Genesis Block
Private Ledger
ASSURED Blockchain Operation Services
ASSURED Policy Management Toolkit
Goal Driven Policy Generation
User Security & Data Privacy
Access Control Policies
User Authentication Access Authorization
Data Sharing & Restriction Policies
GDPR-based Data Management Policy Generation
Trust Evidence Collection
ASSURED Risk Management Toolkit
Analysis Toolkit
Threat & Vulnerability Detection
Cascading Effect Analysis
Big Data Cloud Storage Infrastructure
Near Real-Time Security & Privacy Risk Assessment & Mitigation
ASSURED Risk Management & Policies Enforcement Services
Policy Interpretation & Enforcement Toolkit via the use of Smart Contracts

***FIGURE 2:** ASSURED BLOCKCHAIN-BASED CONCEPTUAL ARCHITECTURE*

Blockchain is one of the most disruptive technologies related to data security today, but beyond the inherently sensitive nature of various personal and commercial data are the persistent challenges of interoperability, data matching, and data information processing, sharing and exchange. To this end, ASSURED protects data and resources against leak or improper modifications, while at the same time ensures data availability to the engaged entities of the data value chains. Internal storage and ledger infrastructures, handling operational and threat intelligence data, can track its provenance and are regularly audited to comply with specified **security and privacy policies**. This way data sources are in control of their own privacy, applications and services. For the former, data sources will be able to define the specification of privacy-related policies, afterwards translated in the appropriate smart contracts, following the principle of user privacy empowerment. More specifically, **privacy enhancement is achieved through the use of trusted computing technologies** (i.e., TPMs) as a central building block towards the provision of privacy-preserving signature schemes, such as Direct Anonymous Attestation (DAA). **By assuring auditable, security and privacy policy compliant actions, ASSURED also guarantees that application ecosystems, where such policies have been technically enforced, are highlighted.**

ASSURED will leverage two general types of ledger infrastructure, namely a private ledger which is responsible for the **creation and validation of contracts between the ASSURED Platform and the internal components of the use case deployments**, based on the details of the data sharing, and a public ledger for **recording the fine-grained details of extracted threat intelligence and business data towards efficient and secure data sharing with external stakeholders** (Figure 2).

Reflecting on ASSURED's work and data flow and how provided data security, privacy, sharing and management services can be engrained into the policy-compliant ASSURED structure, the envisaged conceptual architecture (Figure 2) captures the following set of provided on- and off-chain control functionalities and services based on the use of hardware trust anchors for privacy-preserving data sharing services:

Funded by the Horizon 2020 Framework Programme of the European Union

ASSURED Trusted Blockchain Control Services: Trusted Platform Modules (TPMs) are a central building block of ASSURED's privacy-preserving mechanisms and form the basis for enhanced security, privacy and reliability guarantees for ledger management and maintenance. The smart integration of the TPM technology will allow ASSURED to develop new Blockchain verification methods and significantly advance the state-of-the-art of Blockchain operation services: (i) **secure storage**: a user can store any secrets (keys, passwords or other sensitive data) associated with a TPM, and, when authorized by the user, the TPM allows access to the user's secrets, and (ii) **cryptographic primitives:** it provides a wide and solid basis of cryptographic primitive in order to enable the realisation of cryptographic abstractions, such as Attribute-based Encryption, to complement the ASSURED Blockchain-based offering with advanced authentication and access control.

Trusted Blockchain Wallet: In the ASSURED framework, TPMs are also the basis for trusted Blockchain wallets. They will be used to: (i) provide strong user authentication and to securely store the devices credentials based on the TPM's secure key storage, (ii) control and authorize access to *private* or *public* ledger channels based on the authentication process (e.g., to authorize access to or operations on different ledgers), and (iii) securely and efficiently verify Blockchain updates. In this way, ASSURED will significantly advance the state-of-the-art of Blockchain verification methods: Unlike current mechanisms that often rely on computationally costly and wasteful proofs of work or biased proofs of stake**, ASSURED will use TPMs as central building block to build a very resource-efficient and trustful two-staged Blockchain verification mechanism, which will be even suitable for resource-constrained devices** (such as smart devices - equipped with a TPM).

Trusted Blockchain Attestation: In order to guarantee that only trusted and uncompromised devices can participate in a supply chain, all involved devices will use the TPM secure boot mechanism, and their trust level will be continuously attested and assessed. To this end, all signatures on operational and security-related data (e.g., transactions, smart contracts) will include the respective platform's integrity state (which is the hash value held by the device's PCRs at the end of the secure boot process), which will allow any other party to check whether the data stems or was acknowledged by a trusted device. Depending on the selected privacy level, a conventional or a privacy-preserving signature scheme may be employed. In the former case, a plain digital signature scheme supported by the TPM (e.g. ECDSA) will be selected, whereas in the latter case the TPM-provided DAA scheme can be used as strong privacy-preserving signature scheme. DAA [110] can provide anonymous authentication, attestation and date integrity services. Several DAA schemes and their applications are specified in ISO/IEC 20008 [112] and ISO/IEC 20009 **Error! Reference source not found.**, respectively.

Trusted Authentication: To secure communication and prevent impersonation and man-in-the-middle attacks, peer authentication is of extreme significance. ASSURED will offer **multi-tier secure authentication** based on the aforementioned hardware root-of-trust: (i) trusted identity authentication between devices and the Blockchain, (ii) trusted membership authentication for read and write on ledger, (iii) trusted access authentication for cloud-cased storage system, and (iv) trusted actioner authentication for data search and sharing. ASSURED guarantees that a user or a device claims what it is that is exactly what it is, which means that trust can be delivered inside the physical level – providing trustworthiness for the edge device.

2.2 ASSURED DATA LIFECYCLE, USERS AND STAKEHOLDERS

As aforementioned, ASSURED aims to facilitate the **establishment of secure supply chains** comprising multiple, heterogeneous Cyber-Physical Systems-of-Systems (CPSoS) designed, implemented, operated, and owned by **multiple tenants with different security goals, requirements and priorities**. Therefore, with respect to the security and safety profile of the entire ecosystem, system components must be enabled to make and prove statements about their state and actions so that other CPSoS can align their actions appropriately and an overall system state can be assessed and security policies can be evaluated and enforced. This goes substantially beyond simple authorization schemes telling who may access what data and/or interact with which component but requires **understanding of the semantics of the various types of data that may be exchanged between different actors and for what purpose so as to decide on the security, privacy and integrity requirements of the sharing process**.

Towards this direction and in order to enable ASSURED to provide **enhanced information protection and secure data management over the entire data lifecycle**, in what follows, we first define the **type of data that can be exchanged/shared** (Section 2.2.1) within the environments encountered in ASSURED and outline the concrete data structures managed in the envisioned application domains, namely *Smart Manufacturing, Smart Cities, Smart Aerospace and Smart Satellites*, as put forth by the use case providers. This will then allow the definition of **detailed data sharing models and data flows** capturing the overall concept of ASSURED towards creating threat intelligence data sharing markets (Section 3.3), where (security) information can flow internally within a “Systems-of-Systems” (SoS) ecosystem (i.e., manufacturing floor of a smart factory) or *externally* from one market to another which creates a web of secure information exchange between all tenants and stakeholders that comprise the entire supply chain (e.g., police force or other public authority bodies wishing to get to access to publish safety data).

Furthermore, we also depict the **lifecycle of the data flows** (both operational and security as well as threat intelligence related data) that need to be managed by ASSURED, ranging from data generation, collection and storage to data search and querying, for all users and stakeholders (Section 2.2.2), based on the high-level architecture described in Section 2.1. The amount of IoT data, the velocity of change, and variety of sources implies new challenges on how to securely process and inter-operate between such heterogeneous data sources. Thus, based on this workflow of actions, the endmost goal is to define the **set of security, privacy and trust requirements that need to be met for achieving the secure data sharing requirements of the next-generation SoS** (Section 0).

2.2.1 Types of Data in ASSURED

Data Sources: We consider that a data source is any device or entity (overall, characterized as an *asset*) from which we obtain data, within the target network, either in real-time or upon request (stored data). This would already allow a first division to be made about the nature of the sources.

In a first group would be any device capable of producing new data and offering it on demand: sensors, sensor networks, cameras, satellites, etc. and that can also be divided into three subgroups:

- Sources that do not actively provide the data and should be queried;
- Sources that actively and periodically communicate the data;
- Sources that allow an exchange of information and adaptation to the needs of the requester.

The second group includes all the data repositories that store the data such as databases, repositories, etc. This group is fed with data from sources of the first group where there is a need for further data sharing with external stakeholders – outside *assets* that first need to acquire the appropriate access privileges.

Types of Data: We consider a combination of **operational and threat intelligence data** for maximizing the overall value of the target "Systems-of-Systems". Such a data network is an ecosystem of connected *production equipment* (e.g. Motion Capturing Systems, Industrial PC, CubeSat, etc. as detailed in Table 4), *supply chain inventory, quality assurance, maintenance, and business operations*. Connected, these systems can send, receive and collate such high-value operational data that when processed can be used to drive the optimized decision flows and/or equipment performance necessary for maximizing the business value and/or safety of the target application domain. For instance, the blending of knowledge and data stemming from the real-time location systems, motion capturing systems and Industrial PCs (IPCs) can ensure the correct operation of safety procedures, during the movement of the workers, so as to avoid fatal accidents during the operation of the equipment on a manufacturing floor (Table 2). Such knowledge is invaluable to be shared, thus, there needs to **exist secure processes for making it available as part of the overall pool of operational data**.

Therefore, the vision is to enable data confidentiality, integrity and multi-level access control (**security by design**), data ownership safeguarding (**privacy by design**), data provenance and sovereignty checking and trusted consent management, while respecting prevailing GDPR legislation (as mentioned in the Deliverable D1.1) [98]. In ASSURED, by "security- and privacy-by-design" we understand all methods, techniques and tools aiming at enforcing security and privacy properties at both network and system (software) level from their conception while guaranteeing validity in parallel [100]. As described in D1.2 [99], we make use of advanced property-based attestation and verification methods [101], [103] with the aim of allowing intelligent (unverified) system components and controllers to perform with a predetermined envelope of acceptable behavior and a risk management approach extending it to a larger SoS [102]. Since the required security and privacy properties depend on the system and application domain, understanding and evaluating – in real-time – the existing risks for the entire SoS is of paramount importance. This is where the exchange of threat intelligence information comes into play: **Threat intelligence data comes from the integration of security and analytics services**, either running at the edge (e.g., attestation, system execution introspection, etc.) or at the cloud-based backend (e.g., attack emulation [99]), and include **knowledge on any existing risks or zero-day exploits against an enhanced threat model** [100] including network operation and availability attacks, low-level system attacks and data privacy risks. This way, organizations and supply chain stakeholders not only can they be assured of increased operational correctness within their application domain but can also ensure the creation of a threat intelligence data market with enhanced knowledge sharing of the operational threat intelligence in ICT systems.

Based on the above, we establish that we can have the following types of data sets, as depicted in Table 1.

TABLE 1: *DATA STRUCTURES MANAGED IN ASSURED ECOSYSTEM*

Types	Definition/Description
Raw Data	This is the data collected from the deployed devices and assets (sources) and can either include **operational** or **security related data**. For the former, this depends on the type of application and business logic and can include stream data, text data, images, etc. Table 2 outlines all the type of raw data managed in the context of the envisioned use cases. Security raw data are mainly related to the system (software) data, traced individually per device (through the ASSURED Runtime Tracer [99]), needed for extracting the control flow, system state and

	any other validation properties (as defined in D1.3 [100]) that need to be attested for checking the correct state of the proving device. These are mainly the result of introspection mechanisms such as memory forensics, fuzzy hashing, program analysis, etc.
Threat Intelligence Data	This is the result of analysing raw security data to look for valuable information on what is a possible attack path exploited in case of a device compromise, i.e., identified through a failed attestation report. Analysis can be performed on a single device, for possible identifying a zero-day exploit, or a set of interconnected devices. In both cases, data need to be shared with the backend attack simulation and validation component so further investigation and processing can take place that can further uncover risks that may be a product of cascading effects through asset chains (as one asset can lead to the exploitation of another related asset). The result of this analysis labels as threat intelligence data that can then be further shared internally in the network (through enforced security policies) or externally with other interested stakeholders.
Identification Data	This is a sensitive type of data set aimed at operational (business data) that may include identifiable information. For instance, the stream data collected from the CCTV cameras in the context of the Smart Cities use case, towards enhancing public safety, will contain images of recorded persons. Such data while need to be shared, they adhere to much more strict privacy (and possibly anonymization) requirements. Data confidentiality and privacy protection need to be achieved by granting access to only those set of actors and stakeholders with the appropriate privileges (that can translate to the existence of specific attributes), with different levels of access and information granularity for specific targets groups of stakeholders.
Contractual Data	This is a special case which is aimed at regulating the relations in between the three former types, and between their owners and their users. They mainly contain several aspects such as: • who is the owner of a dataset; • who can analyse a dataset; • who is the owner of the result of the analysis; • the type of analysis it can be performed; • what kind of information can be included in the result of the analysis; • who can use the result of data analysis; • how data owners can execute their rights regarding GDPR • etc. These datasets might include identification of parts signing the contract.

Format: Another aspect to consider is the format of the data since it affects the **space required for communicating** (especially in the context of the security (attestation) raw data) and **storage needs**. Recall that, as depicted in **Error! Reference source not found.**, in ASSURED there are two main data categories: **one is stored on ledger – onchain data, and the other is the offchain data**. The former refers to pointers which are stored on the Blockchain and refer to the actual data which are stored off the chain. *The offchain data are the operational data of interest or the threat intelligence data (attestation raw data) that may be shared among the stakeholder to each infrastructure*. For the offchain data, ASSURED will provide a cloud-based database to form a data pool for those full and “hard” copies of data. To access those data, ASSURED will design database access control mechanism, for example, using whitelist/blacklist, and database access authentication.

Three major groups of formats can be distinguished:

- **Unstructured Data:** This refers to any dataset without a reliable structure from which we can extract other data of our interest. Operational (business) data such images, texts, etc. belong to this group. These are usually dependent on the application domain and the type of services running within the target ecosystem of devices and, thus, can vary greatly. For instance, in the context of Smart Cities, images and stream data from the smoke detecting sensors will be stored; in the context of Smart Manufacturing, contextual data in the form of numerical values will be stored depicting the location and movement of the workers with respect to the machinery per manufacturing floor, etc.

- **Structured Data:** This is the data that has fixed and well-known format and structure that allows relationships to be established. This essentially reflects the security (attestation) raw data that follow a well-defined model based on the type of memory introspection and tracing techniques leveraged [104]. It also depicts the case of relational databases with threat intelligence data that include knowledge extracted from analysing security raw data; e.g., zero-day exploits and other identified risks.

- **Semi-structured Data:** This is the data that has a fixed format but with a non-strict organization, this is the case of mark-up languages such as Extensible Markup Language (XML) and JavaScript Object Notation (JSON). These formats arose from the need to send data between systems in a versatile way that would serve in all contexts.

We can find various standards that use these data formats and that are dedicated to specific fields, for example in health. HL (https://www.hl7.org/implement/standards/) is a set of standards dedicated to the exchange of clinical and administrative data between different health service providers. In other fields such as energy, we find initiatives such as the CIM standard for the exchange of information in electrical networks, or the Energy@home data model which aims to create a standard to connect smart energy devices in home to the Smart Grid.

2.1.1.1 Data Structures in the context of the Use Cases

Based on the above, Table 2 outlines the types of data that is communicated and shared between the different actors (put forth in Section 2.2.2) encountered in the context of each one of the envisioned use cases, namely *Smart Manufacturing, Smart Cities, Smart Aerospace* and *Smart Satellites*. This constitutes the basis for then fleshing out the data sharing models and data flows, as described in Chapter 3.

__TABLE 2:__ DATA STRUCTURES IN THE ENVISIONED USE CASES

Item	Use case	Type	Definition/Description
GENERAL DATA TYPE: STREAM DATA, TEXT, NUMERIC DATA AND IMAGES			
1.	**Smart Cities**	Stream data, text data (e.g., system/incident report, log), sensors data (e.g. smoke, temperature).	The stream data is collected from the CCTV camera, while the sensor data is detected from the deployed edge sensor devices in physical locations. The text data may be related to system, incident report and data analysis generated by the backend operational centre (threat intelligence and public safety related data).
2.	**Smart Manufacturing**	Text (meta-data description) and Stream Numeric Data (actual values of sub-systems e.g. robots, personnel location)	The stream numeric data are the real-time data used to detect location and machine working statuses. Text data, in this case, is used to generate description data for the system report, data analysis results.
3.	**Smart Aerospace**	Text and numeric data	The numeric (operational) data are collected from sensors deployed within airplane representing status of the equipment (for maintenance purposes), performance, etc. This can be used for checking whether any specific maintenance is needed either on hardware level (supply chain inventory) or software (software update). In both

Funded by the Horizon 2020 Framework Programme of the European Union

			cases, this data needs to also be coupled with text data depicting the security (attestation) raw data so as to guarantee the integrity of the operational information knowledge.
4.	**Smart Satellites**	Images, meta-data of the files, status information about the camera, telemetry data.	The text data in this case are used for system deployment, update and status check-up. Images are collected from satellite camera.

2.2.2 ASSURED Users and Stakeholders

Having outlined the overall ASSURED Blockchain-based architecture in Section 2.1 (more details can be found in D4.1 [105]) and the workflow of actions to enable secure data sharing, in what follows, we will list who the users and stakeholders are. We have to highlight that since ASSURED aspires to provide a **Blockchain decentralized market that allows enhanced knowledge sharing of increased operational threat intelligence**, in supply chain ecosystems, by supporting them towards the **accountable reporting of newly discovered Advanced Persistent threats (APTs)**, it is mandatory that the definitions of actors involved in the use cases are aligned to the data protection and regulation primitives (GDPR).

The **GDPR in its 4th article** [106] lists a group of "Definitions" among which we can find the most significant **actors related to the data protection and regulation**. Table 3 also outlines the specific roles identified in the context of the envisioned use cases and provides a mapping with the following GDPR-aligned definitions of data actors.

- **Data Subject**: This role reflects the identified asset (Personnel/Worker or deployed edge device/sensor) who is providing operational and security raw data for further processing. The processing of the data must be lawful, fair, and transparent to the data subject and according to the legitimate purposes specified explicitly to the data subject when collected.

- **Data Controller:** The person or (software) asset (e.g., System Administrator, Public Authority, Agency, Database or other cloud-based decision support system) which alone or jointly with other can determine the purpose and means of the processing of the collected operational raw data. The data controller establishes the purpose for which extracted knowledge (business and/or threat intelligence information) can be used and what privacy protection should be implemented. Each controller shall maintain a record of processing activities that shall contain information on the type of analysis conducted and the provenance of processed data.

- **Data Processor:** A natural person, public authority, agency or other body party that processes operational and/or security related data on behalf of a data controller. That processing is described in the 28th article of the GDPR and shall be governed by a *contract that is binding on the processor with regard to the controller and that sets out the subject-matter and duration of the processing, the nature and purpose of the processing, the type of data and categories of data subjects and the obligations and rights of the controller*. For instance, consider the case of a Certification Body that might request access to the security attestation evidence for a given CPS environment, from the System Administrator (as the Data Controller), in order to certify/audit the correct state and operation of the entire supply chain.

- **Data Recipient:** A natural person, public authority, agency or another body (internal or external to current network), to which the operational and security related data are disclosed, whether a third party or not. Public authorities (e.g., police force as in the case

Funded by the Horizon 2020 Framework Programme of the European Union

of the Smart Cities use case) can be an exception when receiving operational data with personally identifiable information but in any case, the processing of the data should follow the GDPR.

- **Third Party:** A natural person, public authority, agency or body (external to the supply chain) other than the data subject, controller, processor and persons who are authorized to process personal data. Sometimes third parties can act as processors, but usually are vendors and other outside stakeholders which if performing any processing of personal data, it shall be governed by a binding contract.

The ASSURED project, and each individual demonstrator, has already determined who is who in each scenario. In a high-level vision and being aware of the necessity of a deep analysis of the scenarios, currently performed under D6.1, a first approach of the main roles in the different demonstrator can be as indicated in Table 3 whereas Table 4 puts forth the specific roles per demonstrator.

TABLE 3: *GENERIC USER ROLES AND STAKEHOLDERS*

Use case	Roles/parties	Data Subject Category	Definition/Description
GENERAL ROLE/PATY			
ALL	**Personnel/Worker (human asset or deployed edge device)**	Data Subject	This role refers to all of the assets that have been deployed within the target ecosystem and are the actual data sources. These data sources can interact with Internal Operators and/or System Administrators.
ALL	**System Administrator**	Data Controller, Data Processor	This role is to deploy the system and perform the necessary configurations in the environment of the use case owner. It also participates into the access policy definition for later data sharing.
ALL	**External Partners/Stakeholders**	Data Recipients	These external partners are outside the use case environments, and they are the potential data sharing entities. But they don't mainly participate into the internal system operations.
ALL	**Database**	Data Processor	A cloud-based backend system for file and data storage, recording all necessary system parameters, user data, operational data and logs.
ALL	**Threat Handler**	Data Controller, Data Processor	This role is designed to handle the system threat incident. And it could be an internal (within system) or external (e.g., stakeholders) party. After being notified the threat, this role will handle the corresponding incident and return the action results.

TABLE 4: *SPECIFIC ROLES AND PARTIES IN THE ENVISIONED USE CASES*

Use case	Roles/parties	Data Subject Category	Definition/Description

Funded by the Horizon 2020 Framework Programme of the European Union

Smart Cities	**CIO (Chief Information Officer)**	Data Processor, Data Recipient	It seeks security attestation evidence for a given CPS (Cyber-Physical System) environment (e.g. CCTV cameras and some detection sensors in the Athens Serafeio Complex).
Smart Cities	**Internal Operator**	Data Controller, Data Processor	This role reflects the system operator of the entire network topology of the Smart Cities deployment (i.e., CCTV cameras, smoke detection sensors, Cloud-based Analytics, Face Detection, Face Recognition, Object Detection systems) for coordinating the collection of raw data and the type of analysis to be performed. He/she operates before and after the deployment of the ASSURED security services to also collect security raw data for further threat analysis. Thus, he/she interacts directly with all deployed CPSoS as well as the ASSURED agents to perform the necessary operations; i.e., monitor extraction of raw operational and security data, definition of access and data sharing policies, etc.
Smart Cities	**External Member /Partner**	Data Recipient, Third Party	This role belongs to an external entity and receives information collected inside the deployment environment and shared through the Blockchain infrastructure provided ASSURED, for example, they could be the first responders, and Law Enforcement Agencies (LEAs).
Smart Manufacturing	**Real Time Location System**	Data Subject	Software asset, running at the machinery in a manufacturing floor, that generates text and numeric values of 3D Cartesian. Co-ordinate system based on Ultra-Wide Band wireless radio tags that provides the necessary operational data for further processing; i.e., identification of the position of a worker so as to avoid any possible fatal accidents.
Smart Manufacturing	**Motion Capturing System**	Data Subject	Software asset, running both in the deployed devices and the backend cloud-based analysis engine [99], that generates text and Numeric values of Motion Capturing, e.g., accelerometers based on Ultra-Wide Band wireless radio tags.
Smart Manufacturing	**Aggregator**	Data Subject, Data Processor	Ultra-Wide Band Transceiver nodes that collect the wirelessly transmitted information from the Real Time Location System and Motion Capturing System tags. They send it to the backend cloud-based analytics engine for further processing based on the sharing configuration defined by the IoT Gateway.
Smart Manufacturing	**IoT Gateway**	Data Processor, Data Controller	A Single-Board Computer that has networking capabilities and an Embedded Operating System able to communicate to the Aggregator in order to obtain information from the Real Time Location System and Motion Capturing data subjects for further processing as well as communication capabilities to take to the Industrial PC connected to Robotic Arms. This asset also enforces the data

Funded by the Horizon 2020 Framework Programme of the European Union

			sharing configurations (as defined by the administrator) of the type of data to be shared between the different endpoints as well as the security and trust requirements for the data in transit. T
Smart Manufacturing	**Industrial PC (IPC)**	Data Processor, Data Recipient	IPC that communicates with Programmable Logic Controls (PLCs) responsible for the control mechanisms of the robotic arms. The IPC also provides a communication server which can be used by the IoT Gateway for data acquisition of the robotic arm movement.
Smart Manufacturing	**Central Database**	Data Recipient	This is a private cloud database, either key-value storage or time-series database to store the information from location system, robotic motion, and decision-making system.
Smart Aerospace	**Analytics Cloud Server (ACS)**	Data Controller, Data Processor	Cloud-based software asset that is responsible for collecting and analysing real-time monitored data from the on-board units of an aircraft. Such raw data might depict information regarding maintenance, routine checks or if some anomalies (representing a possible exploit or malfunction of an on-board unit). Data sharing between the on-board units and the ACS takes place after the landing of the aircraft and needs to adhere to strict security and integrity requirements.
Smart Aerospace	**Secure Server Router (SSR)**	Data Subject, Data Processor	It is the central point of the use-case. It works as a hub for all the devices identified in the deployment topology [99] - facilitating communications inside and outside of aircraft. Its functionality resembles a communication gateway (i.e., radio, SATCom, LTE, Wi-Fi, Ethernet) that can provide information exchange between aircraft and ground stations, between avionics computer and cockpit as well as offering internet access (Wi-Fi) on aircraft.
Smart Aerospace	**Communicator**	Data Subject, Data Controller	This is a device used to connect flying airplanes and ground station via radio communication. For specific types of operational data that mainly have to do with the correct behavior of a sensor, there is a real-time communication between the aircraft and the ground station. Thus, the Communicator as a device holds the specific sharing policies for specific data sets.
Smart Aerospace	**OEM (Original Equipment Manufacturer)**	Data Subject, Data Controller	Manufacturer of the on-board units and sensors deployed in the aircraft that are responsible for providing trustworthy versions of any software/firmware updates that need to be securely pushed to the devices. Sharing of such trusted software updates are used as reference values (for software updates) to be triggered by the Firmware Server.
Smart Aerospace	**Firmware Server**	Data Processor, Data Recipient	This server is responsible for executing the software/firmware update policies, in the case of a compromised, malfunctioning or out-of-

Funded by the Horizon 2020
Framework Programme of the European Union

			data software (aircraft) sensor. Whenever a device requires an update, an authenticated software update version will be pushed to the device leveraging the ASSURED Blockchain infrastructure. Essentially, the device – through the Connector – will retrieve the software binary update from the ledger, verify its integrity and then perform the necessary update operations. The Firmware Server is then also responsible for collecting the necessary attestation evidence, for the target device, that the update was executed correctly.
Smart Aerospace	**Ground Station Server (GSS)**	Data Controller	This server is used when the aircraft is on ground and serves two purposes: (i) safe data transfer- all the data collected by the SSR during flight will be safely transferred and stored on the GSS, and (ii) remote firmware update - the authenticated server pushes a firmware update on the Firmware Server to then be put on the ASSURED Blockchain.
Smart Satellites	**CubeSat**	Data Subject	CubeSats are low earth orbit satellites, which are usually deployed on constellation. They receive and process commands from the Ground Station and send commands to its subsystems. Data are also exchanged with the Ground Station. Demonstrator within ASSURED will be done with the use of part of CubeSat (an on-board computer).
Smart Satellites	**KUB-OS**	Data Subject	Software asset that runs within CubeSat and provides communication services between CubeSat and Ground Station.
Smart Satellites	**Ground Station**	Data Controller, Data Processor	It commands and maintains the operational status of the CubeSats, distributes new software versions, stores data collection and handles sharing with external parties.
Smart Satellites	**Consultant for Safety Assessment**	Third Party	Impartial third-party operator that seeks attestation evidence in order to verify/certify the fulfilment of security/safety requirements

2.3 ASSURED DATA SHARING REQUIREMENTS – SECURITY, PRIVACY AND TRUST REQUIREMENTS

The following table (Table 5) sets the security, privacy and trust requirements for the design, development and validation of the ASSURED secure and accountable data sharing functionalities through the use of Blockchain and DLT technologies. As aforementioned, ASSURED meets these requirements by providing **secure, trusted and auditable data sharing environments for a new generation of policy-compliant Blockchain structures** enhanced with advanced **on- and off-chain data and knowledge management services through the specification of novel TPM-based security and privacy-preserving protocols** (these protocols will be documented in the respective WP4 deliverables). The endmost goal is to enable data confidentiality, integrity and multi-level access control (**security by design**), data ownership safeguarding (**privacy by design**), data provenance and sovereignty checking and trusted consent management, while respecting prevailing GDPR legislation in D1.1 [98].

Such security, privacy and trust requirements have been elicited adopting a systematic approach, driven by the Fairness & Privacy-by-Design-and-by-Default enriched with the Protection Goals method. Such elicitation relies on the analysis of the specific application domain data sharing regulatory landscape and the factual analysis of the privacy-relevant properties of all collected raw data (both operational and security-related), processing and sharing in each service, including details on the data categories, data sources and purposes of processing.

TABLE 5: *DATA SHARING SECURITY, PRIVACY AND TRUST REQUIREMENTS*

#	Requirements	Descriptions
SR1	Data Confidentiality and Integrity	Data must be protected with appropriate controls to ensure their integrity, confidentiality and availability throughout its entire life cycle (Mandatory)
SR2	Authorization and Access Control	The participating users, devices and stakeholders should act according to the security and privacy policies as dictated by the data sharing preferences (**deployed via smart contracts**) based on the sensitivity of the operational and/or security related data to be exchanged between either *internal* or *external* members of the target supply chain ecosystem. Thus, a specific **dataset can only be read by users matching such pre-defined access policies** (preferences) based on their shown **(verified) credentials** – to be protected from eavesdropping/leakage. In case such policies need to be updated, during runtime (e.g., specification of different attributes for accessing specific system raw (attestation) data), this should be reflected through the deployment of new smart contracts (Mandatory)
SR3	Cryptography	Having strong cryptographic primitives is a fundamental requirement of any security-oriented system. What is needed towards this direction is a good source of entropy that will be utilized in a secure pseudo-random number generator (PRNG) so that the keys generated by the system are secure. To make good use of this source of entropy, we also must ensure that the cryptographic primitives deployed in a root of trust and related systems are fit for purpose (Mandatory)
SR4	User/Device and Data Privacy	One key requirement, when sharing such sensitive type of data (especially in applications such as the ones envisioned in the context of Smart Cities for ensuring the public safety and require the exchange of personally identifiable information) is the **privacy guarantees on the both the data subjects and on the data themselves**. This includes: (i) the protection of **user's data confidentiality** during both data transmission and storage, (ii) the protection of **sensitive information extracted by edge devices**; i.e., data that may collect "personally identifiable information" (e.g., collected from CCTV cameras or other real-time data sources in the context of Smart Cities) should be further **"masked or anonymized"** before being ready for transmission by leveraging **encryption mechanisms or through the creation of virtual replicas** (digital representation) of the actual data based on the use of k-anonymity or l-diversity techniques, (iii) it should be hard for an adversary to learn the secret information required for any action (e.g., authentication, encryption, etc.), and (iii) credentials should be stored on user device and must be protect from eavesdropping/leakage (Mandatory)
SR5	User-controlled Anonymity	When anonymization is desired by the users (thus, empowering user controlled-anonymity), users (their devices and their actions) should not be identifiable without breaching the non-repudiation requirement of their actions (SR133). Observers should not be able to infer private information and whether a user performed or will perform a specific action. Moreover, no observer should be able to link an action to the user or infer if two (or more) actions were performed by the same user (device). Non-repudiation should be checked and verified by the Trusted Component (TC) hosted by each user device (Mandatory)

Funded by the Horizon 2020
Framework Programme of the European Union

SR6	Conditional Anonymity	Users should be anonymous within a set of potential participants. In case a user deviates from system policies, the corresponding credentials should be retrieved and revoked (Desirable)
SR7	User-controlled Unlinkability	According the users' preferences, in order to achieve unlinkability, no action or transaction should be able to be directly linked back to the original initiator without breaching the non-repudiations requirement of their transactions (SR13). Non-repudiation should be checked and verified by the Trusted Component (TC) hosted by each user device (Mandatory)
SR8	Forward and Backward Privacy	In case of a key compromise or identity compromise, of a user/device, this should not affect the privacy of other messages signed by the same user/device. In the same context, in case of a user/device misbehavior, the **revocation of thus user's/device's credential should not affect the unlinkability of previously signed data messages** (Desirable)
SR9	Data Encryption	A party is allowed to use an encryption algorithm, e.g., **Attribute-based Encryption (ABE)**, to encrypt a piece of information data under various attributes and policies (as depicted by the data sharing preferences – cf. SR2) to output a ciphertext, so that only **data seekers exhibiting valid attributes and credentials can decrypt and reveal the underlying plaintext** (Mandatory)
SR10	Trustworthiness of Exchanged data	The data sender/receiver/endpoint are authenticated with secure identities check based on (verifiable) credentials linked to valid attributed required for accessing specific information data (cf. SR4). Furthermore, data's confidentiality is protected via encryption (Mandatory)
SR11	Entity Authentication	In network communication (e.g., Wi-Fi, Ethernet, cellular network), **entity identities should be verified by providing their (verifiable) credentials** (issued by either a valid Issuer or by a certified Root-of-Trust attached to the host) and **access rights should be validated** before setting up secure conversations and data transfer (Mandatory)
SR12	Operational assurance	Evidence on the correct state of a deployed device – depending on the mixed-criticality nature of the processes running in a device we may need to have different levels of trust assurance which corresponds to different levels of attestation variants to be deployed (Mandatory)
SR13	Non-repudiation and Accountability of Actions	Actions should be non-repudiable and all system entities should be held accountable of their actions. For instance, a data subject cannot refuse the authorship of an attestation report that has been shared on the ledger for verification from other users/devices (Mandatory)
SR14	Data Error Recovery	Error correction and original data recovery from encrypted data should be considered in the case where the communication channels may have noise to affect the quality of transferred data (Desirable)
SR15	Honest Incentive for CPSoS	There may be a mechanism that ranks the behaviors of deployed edge devices, when interacting with the DLT for any data transactions, that can be considered as further evidence on their correct state (**reputation score**). Such reputation scores may be explored for triggering specific actions against devices: i.e., credential revocation in case their score goes below a threshold or selection of a set of devices for acting as "jury" in order to resolve an attestation dispute (prover and verifier are providing contradicting attestation evidence) (Desirable)
SR16	Ledger Security	(i) **Integrity of block data** - no one can tamper with the data stored in ledger; (ii) **Verification of block data** - the information stored in the block is valid and verified; (iii) **Mining validation** - a block mined by a user is valid; (iv) **Agreement on validation** – a majority or all network users to reach an agreement on validation; (v) **Membership authentication** - provide access control over ledger (read & write rights) for authenticated users; (vi) **Guarantee of actions** - deliver a mechanism that a "promised" action will be performed successfully; (vii) **Customized block data security** - enable authenticated user to put various encrypted levels of data on ledger (Mandatory)

Funded by the Horizon 2020 Framework Programme of the European Union

3 DATA SHARING PROFILES & INFORMATION EXCHANGE

In this chapter, we elaborate on the **data sharing profiles and the information exchanged with regards to all the use cases of the project**. More specifically, we identify the **data flows** formed through the use case deployments considering their business objectives and the ASSURED security functionalities. We further proceed to a clear categorization of the documented data flows in order to later on elaborate on the data sharing and threat information sharing models.

More specifically, the identified data flows are divided in the following categories:

- **Main business data flow:** Data which are transmitted in order to support the main business operations of the deployment;

- **Attestation and secure device on-boarding data flow:** Data transmitted in the context of the attestation processes for the validation of the operational assurance of the CPSoS of the deployment.

- **Risk assessment:** Data generated and transmitted throughout the use case devices and the ASSURED components in order to support the risk assessment operations and the attestation policies deployment.

- **Handling emergency/threat:** Data flows that refer to the exchange of operational and threat intelligence data with the external stakeholders of each use case domain.

In addition, in the context of the data sharing models of Section 3.2.1, we divide the models into ***Internal*** **and** ***External*** ones. The former refers to data sharing behaviors limited within the frame of a deployment and the ASSURED components, while the latter refers to the data sharing behaviors with external entities through the ASSURED public DLT.

Overall, the data types which are exchanged among the pilots' and ASSURED technical components, entries, and stakeholders are (as defined also in Section 2.2.1):

- **Operational data:** That refer to the actual data, e.g., among others, commands, configurations, files, exchanged for the operational purposes of the demonstrator and the ASSURED framework.

- **Security service data:** Data stemming from the attestation protocols and processes.

- **Threat intelligence data:** Data reflecting the vulnerabilities, threats and risks which are shared among actors, as those has been documented in D.1.1 [98] and Section 2.2.2.

All the data flows, when applicable, are transmitted through or supported by the ASSURED Blockchain infrastructure. ASSURED aims to provide strong **integrity, confidentiality and privacy-preservation on the formed data flows using Blockchain technology, strong cryptographic primitives and crypto abstractions to empower advanced authentication and authorization mechanisms.**

3.1 OVERVIEW OF DATA FLOW MODEL

To clearly understand the data sharing behaviors among the parties and roles in the use cases, we will need to investigate the data flow model beforehand. By data flow, we mean a data pipeline starting from an entity and ending to another. We will first introduce a general data flow model for all use cases, and further describe the specific data flow per use case. In the following chapters, we will define some terms for the data flow. A data flow may include the following data: (1) data collected from front-line devices; (2) data and the corresponding reports after data processing and analysis; (3) data and logs related to software/hardware update and operations; (4) attestation data; (5) threat intelligence information. Hereafter, we refer to the data in (1), that are just collected from the devices and before data analysis to as "raw data", and the data in (2) and (3) as operational data. Based on the definitions, one may mainly see how the data sharing, threat intelligence information exchange, and attestation data flows work in general and in the context of the use cases' models later.

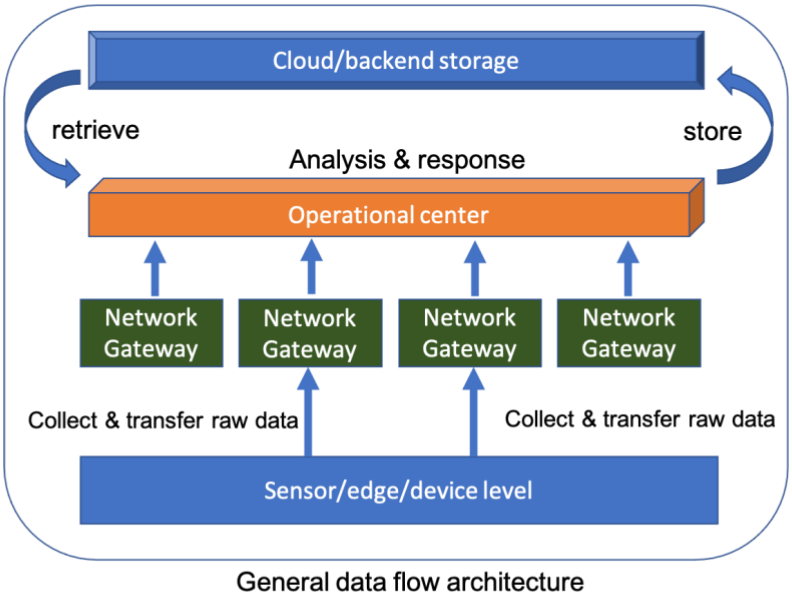


FIGURE 3: *A GENERAL DATA FLOW ARCHITECUTRE FOR USE CASES*

It is of extreme necessity to achieve privacy-preserving data sharing in the distributed and collaborative SoS supply chain. In this context, supply chain stakeholders should be allowed to only share meaningful and useful data with their specified users. For example, a material order and the corresponding payment between material provider and manufacturer, should be kept transparent and securely isolated from the view of market clients and customers. Similarly, the edge devices' data collected from a region or location, could be safely stored in a local data center, and further be able to securely be shared with external stakeholders who satisfy the pre-defined data access policies. A careless and unprotected data management, accessing and sharing will lead to a great threat to personal data and business secrets and can also lead to unforeseen financial loss. In order to protect the valuable data chain of SoS, following the definition of data flows and data sharing behaviors and models among all the ASSURED use cases, we will present the security, privacy and trust requirements that need to be considered.

3.1.1 A General Data Flow Diagram & Description

In this section, we illustrate a generic data flow diagram to summarize the common features for the required data flows in all the use cases. Note that in this general data flow description, we only focus on the data flow "actions" rather than the privacy and security requirement that need to be met. As for those requirements, Section 3.2.1 will provide a thorough documentation for each one of the envisioned use cases. From Figure 3, there are four different data flow layers, namely **sensor level (i.e., the front line for devices), network gateway, operational center level and cloud-based backend storage**. The raw data is originally collected from the front-line devices, e.g., sensors, edge devices, and further packed and sent to network

Funded by the Horizon 2020 Framework Programme of the European Union

gateway. The network gateway then organizes and reforms the data and forwards them to the upper layer, i.e., to operational center. In the center, the data is processed and analysed so that the corresponding data-driven results will be given to some action parties for further operations, for example, a first responder is notified by a fire detection alarm. All the aforementioned data (including raw data, analysis results, action reports and logs) will be finally stored and recorded on a cloud-based storage system. Later, the system enables the operational center for further data accessing and interactions. We note here that we do not specifically show the Blockchain layer, in between the operational and the cloud-based layers. This is because the Blockchain platform can be regarded as a “distributed cloud” component. Showing it or not will not affect the general data flow for all the use cases. Therefore, based on the previous descriptions, we have the general data flow architecture in Figure 3.

After the general descriptions, we present the privacy and security considerations for each layer based on the requirements already defined in Table 5**:** data sharing security, privacy and trust requirements. The summary can be seen in the following Table 6.

TABLE 6: *SUMMARY FOR PRIVACY AND SECURITY CONSIDERATIONS*

Layer	Information direction & involved actors	Considerations
Sensor level	This is the lowest level within the general architecture. In this level, data is collected directly from the real-time and active sensors and edge devices. The data will be transferred to an upper layer – network gateway level, which holds data receiver, gateway.	In this layer, the main security and privacy considerations are described as: (1) data can be securely transferred to the upper layer; (2) during and after data transfer, data integrity can be guaranteed; (3) authentication can be done between the actors in the two layers, i.e. sensors/devices and network gateways; (4) when required, the upper layer should not be able to link the uploaded data to a specific sensor or edge device – this can be seen as sensor’s level privacy that can be required in cases engaging sensitive data processing. (5) An operational assurance check-up should be considered so as to check if any device in this layer has issues or threat incidents. The device and the threat shall be identified, and its status shall be traced and monitored.
Network gateway	This layer is used to commute between they sensor layer and operational center layer. The network gateways here should collect the data from sensor layer, and then forward them to the operational entities at the upper layer.	In this layer, the privacy and security considerations mainly rely on the data confidentiality – meaning the data transferred from this layer to the upper layer should be protected, also considering data integrity. Data error correction should be also considered just in case there is noise in the communication channel. Further, authentication is needed so that the operation center can be convinced that the data is from an authenticated gateway. The gateway privacy may not be needed here. Operational monitoring for the gateway status could be required. Besides, a gateway’s data transmission actions should not be denied later.
Operation center	This layer is mainly used for data processing and analysis. The data is basically collected from the network gateway layer, and after data arriving, the layer feeds the data into analysis algorithms and components. The data along with analysis results will be later uploaded to cloud-based storage backend.	The main considerations in this layer are (1) data should be protected and will not be leaked out from this layer during analysis; (2) the received data is trustworthy, e.g., the data is from a trusted and authenticated source; (3) data error correction could be needed before the data analysis.

Funded by the Horizon 2020 Framework Programme of the European Union

Storage level	This is the top layer within the architecture – cloud-based storage backend. This layer will store the hard copies of all the data, including raw, analysed data and the related data logs. We note that this final layer should support a distributed, trust and secure way for data sharing with internal and external parties.	The crucial considerations for this layer are data's secrecy and integrity during data storage and sharing (access and retrieval). Secure authentication, secure data access policy control, and sufficient data sharing monitoring and record are also required.

3.1.2 Specific Data Flow per Use Case

In what follows, we define the data flow per use case and we show how these data flows work and interact with our ASSURED framework. For each use case, a general description is given and the data flow follows.

3.1.2.1 Use Case #1- Secure Collaboration of "Platforms-Of-Platforms" For Enhanced Public Safety

The Athens testbed consists of edge-devices, gateway infrastructure and data flows among the basic data storage and external stakeholders. The edge devices of the pilot-site include cameras and smoke detector sensors used in the context of the public safety in order to support the decision makers in the operation centre. The collected data (video-streams, sensors etc.) are shared in a network of switches and routers/gateways, stored in a cloud infrastructure and shared with the external stakeholders' ecosystem, such as first responders in case of incidents via the cloud-based backend. In this context, the main data flows of the enhanced public safety use case are as follows:

- ***Main business data flow:*** The deployed cameras, sensors and edge devices collect real-time video stream and sensor data from a specific location (e.g., in a building). The data is further transferred to an access point (which will perform data aggregation actions) and it forwards the data to network switch/gateway. Note that a gateway is assigned to a specific location to handle the data collection from all the devices, e.g., each building has its own gateway. The gateway (taking care of assigning incoming data package to the corresponding data center, based on the pre-defined access and data forwarding policy list that is determined by system administrators) sends the received data to data analysis center (i.e., the operation center) so as to perform risk event analysis. Eventually, data will be stored on the cloud-based storage by the DAEM system administrator. The aforementioned data transfer in the DAEM context takes place via WiFi. We note that the main business data flow should be protected by the ASSURED privacy and security enablers during the data transfer.

- ***Attestation and secure device on-boarding data flow:*** In case a new device is added in the demonstrator's environment, there are two potential cases that generate the respective data flows: (i) a new edge device is added (e.g., a sensor or camera) or (ii) a new PC is connected to the operation center in order to support new services or to support new members in the operation center team. In both cases an identification procedure will be followed in order to ensure the reliability and operational assurance of the new device through the ASSURED attestation. The edge component has to be verified in order to be connected to the gateway by DAEM system administrators as well as the new PC in order to be connected to the operation center environment. This verification process through attestation generates the respective attestation data flows as a result of the attestation interplay between the Verifier and the Prover. In both cases an authorization mechanism will take place to ensure trusted access control for secure device on-boarding. In fact, the information exchanged in the context of the authorization mechanism is considered part of the attestation and secure device on-boarding data flows.

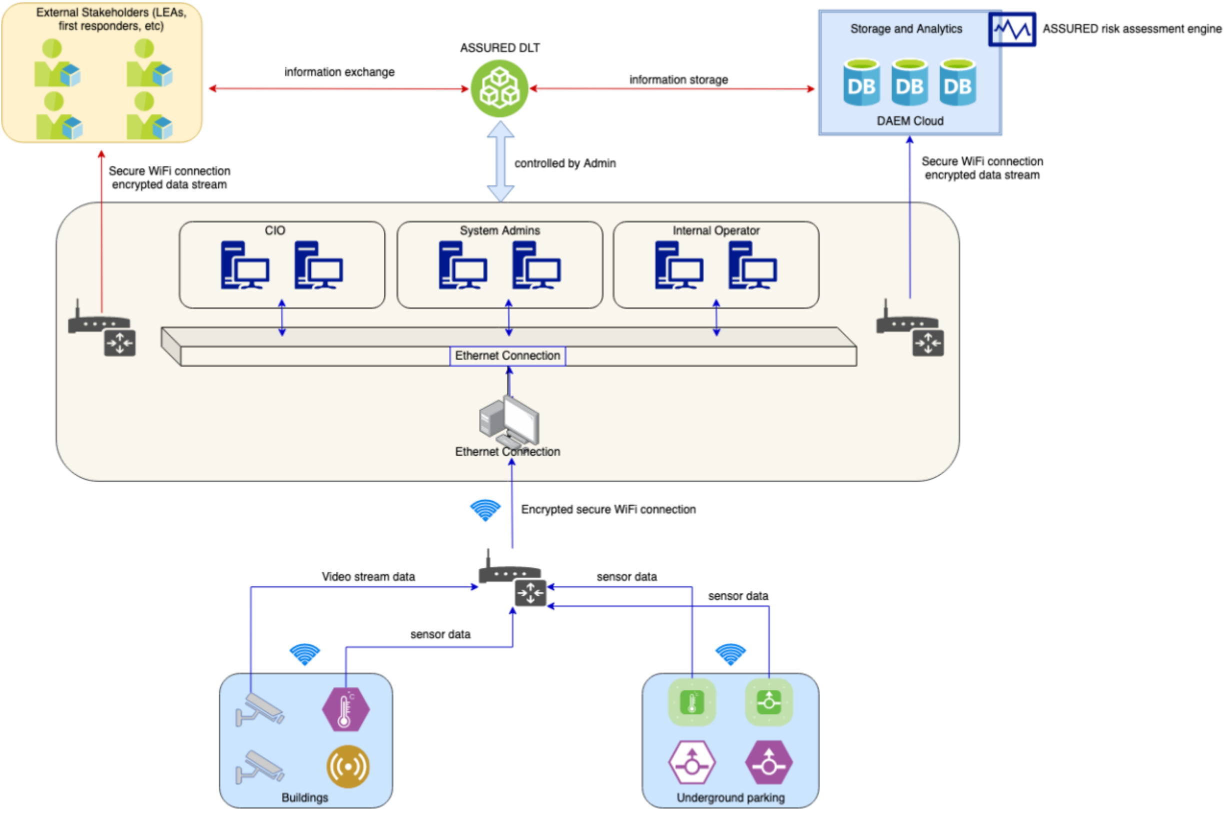


FIGURE 4: SMART CITIES UC MAIN DATA FLOW

- ***Risk assessment:*** The system administrator will make use of the attack validation component of ASSURED to collect transferred data from front-line devices and further analyze the data with the support of the Risk Assessment platform in order to check if there exist any attacks, faults and risks. In addition, the risk assessment data flows refer to the generation and enforcement of the necessary attestation policies which are defined by the policy manager of the risk assessment framework and enforced by the Security Context Broker of the ASSURED framework. These policies are transmitted to the devices under the form of smart contracts to be managed by the secure Blockchain wallets. Overall, the aim of the risk assessment process is to ensure that data confidentiality and integrity requirements are met through the ASSURED solution. The assessment process generates alerts which will be sent to the administrator if a cyber risk is identified.

- ***Handling emergency/threat:*** The collected data are sent for analysis via the use of ASSURED threat assessment engine that supports the operation center. If there is an emergency, an alert is generated in the public safety system of the operation center and the decision makers initiate a response process, e.g., by informing the first responders, other external members or internal related municipality agencies, such as the municipal police. Upon an incident, all the aforementioned entities can acquire through a secure, auditable and privacy-preserving manner threat intelligence and business data related to the incidents. The data access to these entities is based on a users' secure management and authentication prior to enrolling accounts and access rights. If a cyber-attack is identified in the system's endpoints, the system admins intervene. In case of a physical attack, city officials are notified for intervention. DAEM and city officials will exploit the results. Both the results and the final action reports will be stored in the cloud storage and further recorded on ASSURED DLT for distributed data accessing (with internal/external parties). Overall, the threat intelligence information shared suggests strong confidentiality

Funded by the Horizon 2020 Framework Programme of the European Union

and integrity requirements, while sharing results of operational data from the surveillance cameras and face detection systems pose strong privacy requirements.

In the above data flow (depicted in Figure 4), there are several types of data: (1) original real-time data, e.g., being taken by the CCTV camera, which are stream data; (2) network data of stream data, among access point, network switch and gateway; (3) analysis result data (could be in textual format) sent to operation center; (4) data completed by the operation center and related to the operations, e.g., operation results – fire was distinguished, sector was safe (could be in textual format); and (5) final public safety assessment data (e.g., incident report, all analysis results), stored in the cloud.

3.1.2.2 Use Case #2 - Safe Human Robot Interaction (HRI) in Automated Assembly Lines

The smart manufacturing demonstrator for safe HRI showcases a workplace environment which comprises of industrial robotic arms alongside a space where personnel can walk within this environment. The personnel carry an Ultra-Wide Band (UWB) wireless tag that transmits 3D Cartesian co-ordinates to Ultra-Wide Band anchors mounted in the workspace. The location information as well as the robotic movements are made available to the Industrial PC. The IoTGateway is an edge processing unit that consumes the above-mentioned data streams and tries to predict and avoid any sort of hazard in the workspace through custom algorithms that can control the robotic arm movements. From ASSURED perspective the demonstrator’s IoTGateway will benefit from: (i) authentication of firmware updates on the Gateway as well as assurance of no malicious code injection, and (ii) protection of trustworthiness of the workplace by avoiding unauthorized code injection and software manipulation of the custom algorithms running on the IoTGateway. In this context, the main data flows of the HRI use case are as follows:

- ***Main business data flow:*** As a worker with a mobile tag is moving in a sensing area, the motion capturing system monitors the worker’s position in a real-time manner. The Ultra-Wide Band positioning tags provide real-time location of workers as stream data of 3D Cartesian co-ordinate system wirelessly to aggregators. Aggregator provides noise filtering and some basic signal processing on the incoming wireless data from the tags and then publishes this filtered information towards an MQTT Broker on the same wired network. The IoTGateway is connected to the same network via Ethernet through a network switch. With the position data sent by aggregator, the IoT gateway runs Decision Making Services for Collision Avoidance / Prediction. If a dangerous pattern is matched, the gateway will immediately send a Stop Signal to the IPC, so that the IPC can deliver a “stop” command to the working device (robotic arms). The decision logs can be in textual format or JSON information which is uploaded to the central database. The transferred data and commands in the context of the HRI imply strong integrity requirements in order to ensure that safety critical decisions are taken based on credible information and sensor data.

- ***Attestation and secure device on-boarding data flow:*** When system administrator requires the IoT gateway to be attested, an attestation request will be sent to the gateway. The gateway will interact with the ASSURED attestation component. The component helps the gateway to generate attestation proof with the corresponding real-time status given by the gateway, so that the proof can be sent to the administrator for internal verification. In addition to the attestation data flows generated for the verification of the IoTGateways, attestation flows are also generated for the verification of the core processes of the edge devices. More specifically, we refer to robotic (edge) devices that run safety critical processes supporting positioning services of workers (to avoid collision) or those responsible to handle “stop” commands that immediately shut down the device operation. In those cases, the IoTGateways are responsible of attesting the edge devices acting as the verifiers in the attestation interplay. Hence, attestation is performed in multiple levels of the use case's deployment. This implies several attestation data flows spanning throughout the ASSURED-enabled infrastructure. Last but not least, new devices can be added in the manufacturing floor implying the generation of device on-boarding data flows that must be handled by the asset management services of the pilot and safeguarded by the ASSURED offerings for secure communication.

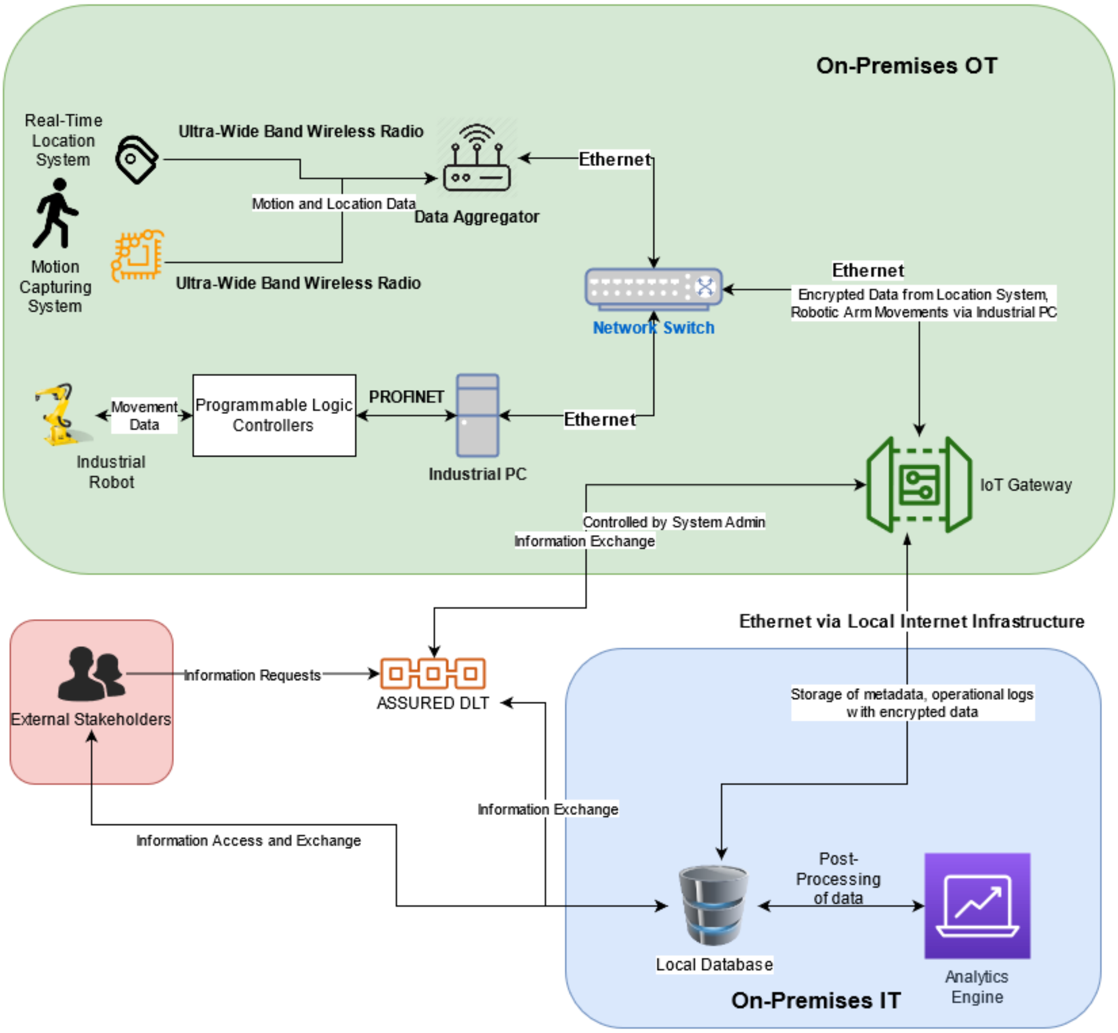

On-Premises OT
Real-Time Location System
Ultra-Wide Band Wireless Radio
Ethernet
Motion and Location Data
Data Aggregator
Motion Capturing System
Ultra-Wide Band Wireless Radio
Ethernet
Encrypted Data from Location System, Robotic Arm Movements via Industrial PC
Network Switch
Movement Data
Programmable Logic Controllers
PROFINET
Ethernet
Industrial Robot
Industrial PC
IoT Gateway
Controlled by System Admin
Information Exchange
Ethernet via Local Internet Infrastructure
External Stakeholders
Information Requests
ASSURED DLT
Storage of metadata, operational logs with encrypted data
Information Exchange
Information Access and Exchange
Post-Processing of data
Local Database
Analytics Engine
On-Premises IT

FIGURE 5: SMART MANUFACTURING UC MAIN DATA FLOW

- ***Risk assessment:*** The system administrator will make use of the attack validation component of ASSURED to collect data from the robotic devices, aggregators, and IoTGateways in order to further analyse the data with the support of the Risk Assessment platform. More specifically, the attack validation component will aim to verify the status of the PLCs used to manage the critical operations of the manufacturing floor. In this way, the admin will detect attacks, faults and risks in the process. The alerts generated and sent to the administrator belong in the risk assessment data flows. These flows refers also to the generation and enforcement of the necessary attestation policies which are defined by the policy manager of the risk assessment framework and enforced by the Security Context Broker of the ASSURED framework. Strong data confidentiality and integrity requirements must be considered to safeguard the risk assessment data flows.

- ***Handling emergency/threat:*** The collected data are stored on the databases residing at the backend systems of the demonstrator. Based on that data, a data analytics engine is fed in order to generate useful statistics reflecting the operational status of the deployment. Those analytics along with safety incidents are shared through the ASSURED Blockchain

Funded by the Horizon 2020 Framework Programme of the European Union

data value chains. Several stakeholders are interested to this information. More specifically, certification bodies that need to ensure the correct and according to the relevant standards operation of the manufacturing environment, other manufacturers and factories in the supply chain who want to audit the security/safety assurance of the deployment, and finally, safety auditors interested in checking the safety condition on the manufacturing floor. All the aforementioned entities can acquire through a secure, auditable and privacy-preserving manner threat intelligence and business data related to the safety critical processes. The data access to these entities is based on a users' secure management and authentication prior to enrolling accounts and access rights.

In the above data flows (as depicted in Figure 5), we can see several types of data: (1) Original stream data from the robotic arms. (2) Original stream data from the location system. (3) Network data of stream data from location system wirelessly sent from UWB tags to aggregators available for IoTGateway. (4) Network data of stream data from robotic arms sent via profinet to the IPC via wired network available for IoTGateway. (5) Analysis data from IoTGateway sent to the Central Database (JSON or text format). (6) Operations network data sent via IoTGateway to the IPC to send "Stop" command. (7) Final data stored in Central Database (JSON or Text).

3.1.2.3 Use Case #3: Secure and Safe Aircraft Upgradability & Maintenance

The secure and safe aircraft use case is focused on the ecosystem designed around the aircraft. A complex system composed of several on-board services, available to both the on-board personnel and the ground engineers, communicating with the ground station server. The main focus of the use case is a Secure Server Router (SSR), a real-time embedded device that is the centre of many functionalities offered on the aircraft. The main operations related to the SSR with respect to the ASSURED project are: (i) data collection from the aircraft sensors while flying and the following secure transmission to a ground station server when on the ground, (ii) the remote authenticated maintenance and firmware updates, (iii) possible attestation on devices on airplane, and (iv) data storage and sharing via ASSURED DLT. More introduction details with respect to the use case can be found in the Chapter 4.4 of Deliverable D1.1 [98].

- ***Main business data flow:*** The data flow of this use case mainly consists of two parts: one is within the airplane and the other is outside the airplane. Both parts are strongly linked to the SSR.

  - *Inner airplane:* The devices, temperature sensors, smoke sensors, fuel sensors, engine sensors, ice on wings sensors – all these inner sensors installed within the airplane are first authenticated with the SSR via the ASSURED secure communication enablers. These devices send the collected real-time data (e.g., temperature, smoke, fuel, etc.) to the SSR. The data will only be stored on the SSR on the flight and downloaded to the Ground Station Server (GSS) when the aircraft is on the ground (meaning the flight is over). Sensors' data are not shared outside the SSR with any inner airplane components and unauthorized outsiders. The SSR, within airplane, also provides Wi-Fi signal to the smart phones used by the passengers. Only the necessary information (including identity credential for authentication in the context of wireless connection) required to provide internet access to the passengers.

  - *Outside airplane:* Once the aircraft is set on ground, it starts the point-to-point communication with the GSS so that all the data collected by the SSR can be transferred to the server. Before the data uploading, mutual authentication is required to verify both identities. In case further analysis are required to be performed on the data, the data stored on the GSS must be securely transferred to an Analytics Cloud

Server (ACS). This would allow the data to be processed and identify anomalies, e.g., the performance of engine suffers from some problems, as part of the user story UTRC.US.2.

Whenever the SSR requires to have its firmware updated, due to proprietary upgrades made on the software, the airplane has to be on ground and a secure communication needs to be established between the SSR and the firmware server. Both parties need to be authenticated and the software itself should be attested to avoid software tampering during the whole process. (If needed, a firmware update for SSR will be remotely completed by the GSS as described in the context of US.1 in D1.1.)

- ***Attestation and secure device on-boarding data flow:*** When system admins require an (on-ground) airplane device to be attested, the device should refer to the smart contract (which defines the type of properties that need to be attested) stored on the ASSURED DLT ledger. Then, the device will connect to the ledger and execute the attestation contract and share the attestation result on the ledgers. The component helps the device to generate attestation proof, and further store the proof on ASSURED DLT (as well as GSS) so that the admins (as well as further auditors) can check the proof and the results. In addition, the ASSURED swarm attestation approach will be leveraged for the attestation of multiple devices to real-time check their statuses and integrity. For new devices and elements, we consider the following two cases: a new GSS is installed in the network and a new sensor in installed on the airplane. The new GSS must be authenticated and attested by the admins based on known behaviors based on other servers. The attestation has to be done remotely since the communication between these two elements is only via Wi-Fi. When a new sensor is installed on the airplane, the sensor is connected to the SSR via Ethernet. The data transfer has to be attested through data-flow attestation mechanisms to respect the nominal data transfer between sensors and the SSR. This attestation is done via the ASSURED attestation component.

- ***Risk assessment:*** In the context of the secure aviation use case, the risk assessment framework of ASSURED aims to contribute to the detection of cyber risks as a result of vulnerabilities and deviations of the normal behavioral profile of core services. In this context, threat intelligence data and attestation results are fed in the risk assessment engine of ASSURED to enable the administrator to evaluate the operational assurance of the aircrafts. Based on the assessment results, attestation polices are deployed in order to regulate the operation of safety critical processes. Thus, the risk assessment data flows refer to the threat intelligence data and the attestation reports and polices transferred between the aircraft components and the ASSURED components. Those data are also logged on the Blockchain of ASSURED.

- ***Handling emergency/threat:*** Once threat intelligence and operational data have been securely transferred to the GSS, other stakeholders, such as the ACS or external companies, might have the possibility to request access to parts of the data for internal analysis or for external audits (through ASSURED DLT). These interactions require: (i) authentication for all parties involved, (ii) authorization, access control and data integrity on the specific data required for the analysis. In fact, the sharing of threat intelligence and operational data through the ASSURED DLT, will give the opportunity to external certification bodies to audit the correct operational state of the safety critical systems or to certify that aircraft systems are up to date. Currently, this process requires the physical engagement of auditors in the airplane’s systems. Using the ASSURED DLT, this process can become more flexible and certification processes can occur remotely in order to validate that the maintenance processes occur as expected and according to the certification standards. Overall, this will be made possible by the ASSURED Blockchain capabilities which will not only allow to satisfy the security requirements but would also offer non-repudiation over the actions made by both parties.

Funded by the Horizon 2020 Framework Programme of the European Union

To summarize the above contents, we show the simplified data flow descriptions as follows.

1. Sensors → SSR: numerical real-time data [via Ethernet]
2. SSR ←→ Passenger: network data [via Wi-Fi]
3. SSR → GSS: numerical/textual data collected from the sensors, stored on SSR to GSS [via Wi-Fi]
4. GSS → SSR: binary data representing the potential firmware update for SSR [via Wi-Fi]
5. SSR → ACS: numerical/textual data collected from the sensors, stored on SSR are sent to ACS for data analysis. [via Wi-Fi]
6. Admins ←→ Devices: attestation data flow [via Wi-Fi]

The aforementioned data flows are visualized in the figure below (Figure 6) with respect to the communication type used for their transmission.

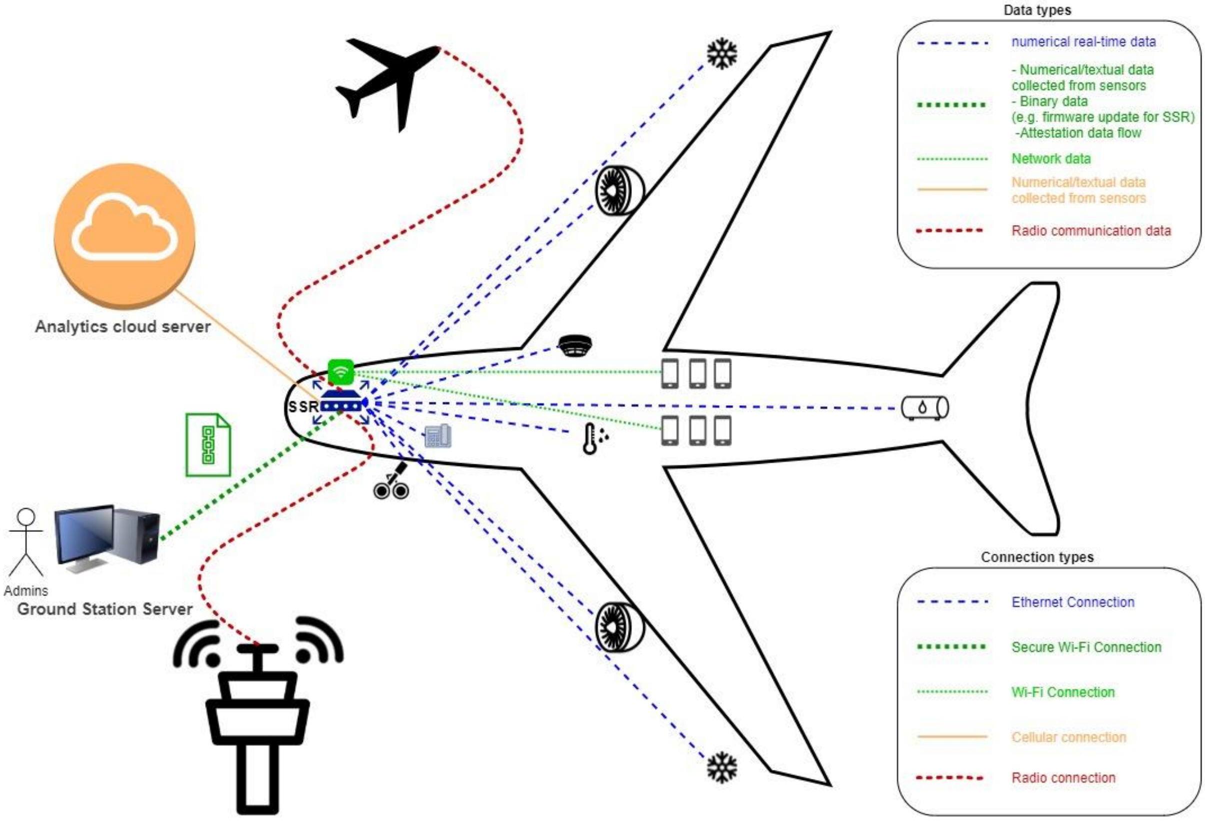


FIGURE 6: *SMART AEROSPACE INNER AND OUTSIDE AEROPLANE DATA FLOW*

3.1.2.4 Use Case #4: Digital Security of Smart Satellites

The digital security of smart satellites use case testbed consists of CubeSats operating and cooperating to execute specific mission(s) and Ground Station, which monitors, maintains, and controls their operation. Given the communication among them, there is a need for ASSURED to confirm the integrity of all modules cooperating to execute mission critical functions, enhance confidentiality and integrity and provide resilience of the software components (OS and Software modules) against specific attacks. The data flow in this use case is mainly among CubeSat and the ground station (GS).

Funded by the Horizon 2020 Framework Programme of the European Union

The following data flows conclude the day-to-day operations (business flow scenarios) of CubeSats (nominal operation and error recovery). Note that data encryption and checks for data integrity are of special emphasis for the CubeSat domain and all the communication among CubeSats and the Ground Station should be encrypted. In the data flows the calls of Security Attestation Mechanisms are included. The sharing of threat intelligence information is included in the outgoing data flow forwarding the necessary alerts to CubeSat Security Officer and to external stakeholders (in case it is needed). In this context, the main data flows of the smart satellite use case are as follows:

- ***Main business data flow:*** Command data flows are formed between the GS and the CubeSats. The GS operator is authenticated by the CubeSat and sends commands (from GS) to CubeSat (in which the commands are in the text format). This is mainly used for error recovery. For example, this can be used when the CubeSat experiences an error or problem that the automatic recovery methods cannot handle and therefore manual diagnosis might be necessary. In addition, trigger-driven mission control data flows are formed. More specifically, CubeSat operator controls the execution of Mission Application (coordinating various services) to perform on-board satellite actions (e.g., triggering a mission application which orients an imaging device to the requested coordinates and takes a picture). This data flow can be seen as a self-circling data flow within CubeSat. Usually this is an on demand triggered process by the CubeSat operator in order to perform a specific task. Furthermore, assuming a valid mutual authentication, GS can download Payload Files, such as the captured images. These data can be files that include images and text-based metadata. This process normally it is done automatically (periodically according to a pre-defined time window) and can also be triggered manually by the CubeSat operator. Finally, in the context of mission updates, GS will distribute new version of software to CubeSat through the file transfer service of KUBOS (in the CubeSat). This is mainly used for error recovery purposes.

- ***Attestation data flow:*** These flows refer to status monitoring through attestation, Telemetry & Logs checking (in which the status info and logs are in the form of text). More specifically, these flows may be formed, (i) automatically, within a specific time window, when CubeSat reports the status of the data to GS; (ii) when GS operator sends query request to visit the Telemetry Database in the CubeSat, and after the authentication by CubeSat, the access is granted; or (iii) when there is a need to verify that the CubeSat correctly and securely performs some missions but also maintains correct status. Then, GS will first send a request information to the CubeSat for attestation. The CubeSat then forwards verifiable evidence, a piece of current status, to the ASSURED Attestation Server. The server will generate a proof and result which are sent back to the CubeSat, so that the CubeSat will forward the result to the GS. The communications, in the context of the attestation data flows, occur via secure channels.

- ***Handling emergency/threat:*** GS will share the collected data with external stakeholders through ASSURED DLT component, and it will interact with ASSURED services components (e.g., risk assessment engine) for data analysis and attestation. Note that external stakeholders can include the CubeSat operator Security officer (sharing some threat intelligence). Some other examples of external stakeholders can include Regulatory Authorities (in case regulatory authorities require data to confirm compliance with specific regulations) or Communication Service Providers or Integrators (if for example CubeSats are used for telecommunication purposes and received data should become available to them).

Note we assume that the CubeSat is pre-set and pre-installed into the system, which will not be considered as a real-time new device/component deployment. In this case, there will be no data flow for installing new device/component.

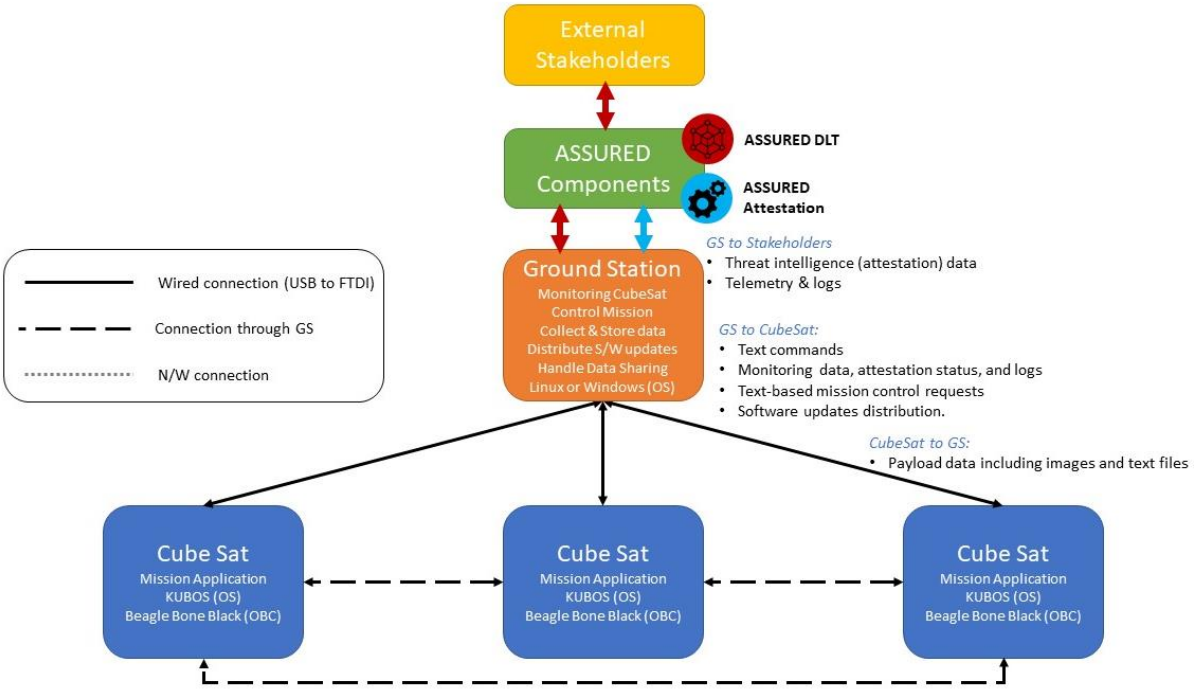


FIGURE 7: *SMART SATELLITES UC MAIN DATA FLOW*

In the above data flow (as depicted in Figure 7), several types of data can be summarized: (1) from GS to the CubeSat: (i) text commands; (ii) monitoring data, attestation status, and logs; (iii) text-based mission control requests; (iv) software updates distribution. (2) from the CubeSat to GS: (i) payload data including images and text files. (3) from GS to outside: data stored (such as threat intelligence (attestation) data, telemetry & logs) in the GS shared with outside components and stakeholders.

3.2 TRUST CONSENT AND DATA SHARING MODEL

The focal point of the previous section was the identification of the data flows and the data types which are formed in the context of the demonstration environments, not only as a result of the business and operational objectives of the demonstrators, but also, as a result of the integration of ASSURED offerings that will enhance their securing posture.

Based on the definition of the data flows and the data types that take place in each of the demonstrator's ecosystem, in what follows we extract and define the data sharing behaviors that capture these data flows. The goal of this section is to provide this high-level abstraction of such data sharing profiles, including the interactions of engaged entities that need to be managed by the ASSURED Blockchain technology for supporting the secure and privacy-preserving data sharing among all actors in a CPSoS-enabled supply chain. Note, that in this section we introduce a high-level definition of the data sharing models and a more detailed documentation will be offered in the deliverables of WP4. Essential, we define the data sharing behaviors that the ASSURED Blockchain technologies and the crypto operations need to be able to manage.

3.2.1 Data Sharing

This section will mainly introduce the data sharing model for all use cases. A general data sharing mechanism and the detailed models will be introduced below. In addition, for each of

Funded by the Horizon 2020 Framework Programme of the European Union

the ASSURED use cases, we first document the security and privacy requirements that need to be considered in order to safeguard the introduced data flows.

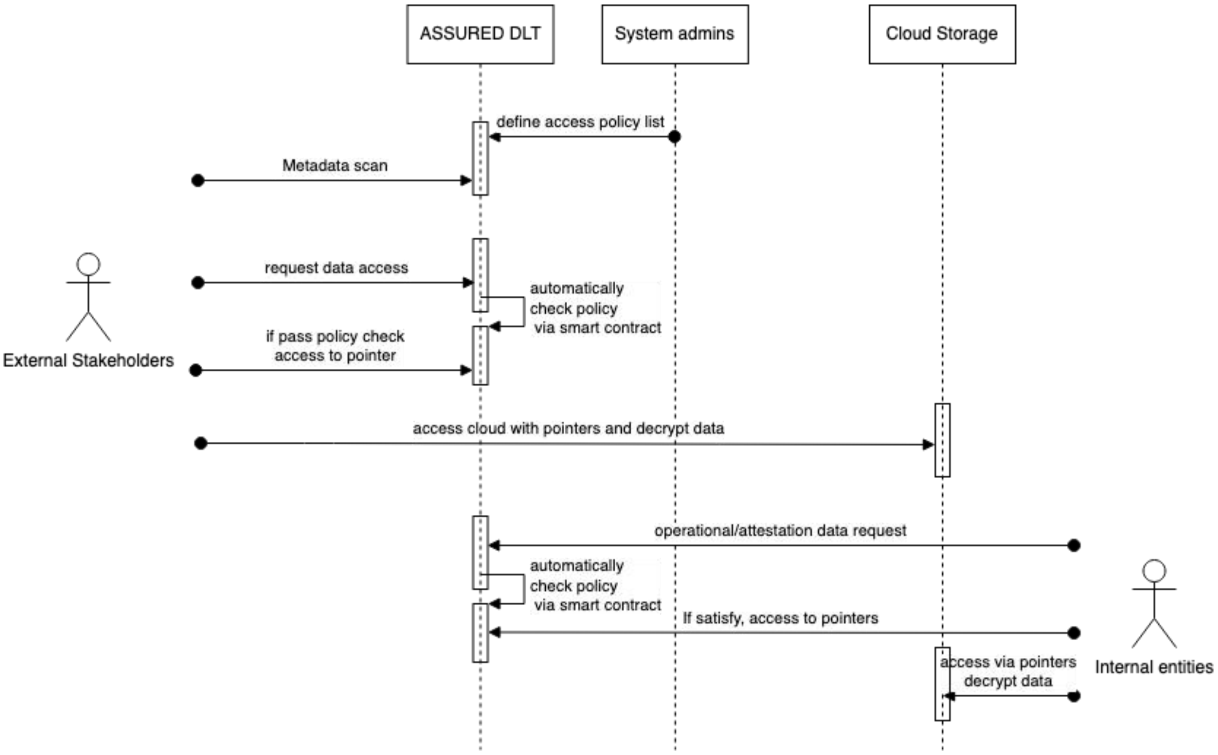

ASSURED DLT
System admins
Cloud Storage
define access policy list
Metadata scan
request data access
automatically check policy via smart contract
External Stakeholders
if pass policy check access to pointer
access cloud with pointers and decrypt data
operational/attestation data request
automatically check policy via smart contract
If satisfy, access to pointers
access via pointers decrypt data
Internal entities

FIGURE 8: *THE GENERAL DATA SHARING MODEL*

3.2.1.1 General Data Sharing Mechanism

The general data sharing model is described in the following descriptions and illustrated in Figure 8. The data sharing includes internal and external modes.

- ***Internal mode.*** As mentioned previously, the attestation data is stored in the ASSURED ledger and cloud-based backend, in which a real-time and live version of the attestation metadata resides in the private ledger and the hard copy is stored on backend. When internal parties, e.g., a device in some region, request access to the attestation data, they will need to authenticate and get authorization from the ledger. As for the sharing of operational data among the internal entities, we need to use access control policy list related to the pointers stored on ASSURED private ledger. This is enhanced by attribute-based encryption (ABE), so that an internal entity can decrypt the pointers as long as their attributes match the access control policy. The pointers enable one to direct himself on the data stored on the cloud backend, in which the data could be also encrypted, e.g., via ABE. The policy list is pre-defined by system admins and further merged into the smart contract for automatically policy checking. If the entities are in the list, then they will be allowed to access to the corresponding (encrypted) pointers. Here the decryption abilities of the internal parties are guaranteed by the trusted Blockchain wallet which manages the decryption keys. Note that the "double" protection here for the pointers is twofold: on the one hand, the protection is applied for access control in order to define who can access the data and, on the other, is for the protection of on-chain data via ABE – meaning that even if insiders or outsiders evade the access control mechanism on ASSURED private ledger, the on-chain data will remain secure.

- ***External mode.*** In this mode, operational and attestation data may be shared with external stakeholders of the supply chain. To enable the external stakeholders to search

Funded by the Horizon 2020 Framework Programme of the European Union

and identify the data of interests, ASSURED will present searchable metadata on the public ledger. These metadata could be seen as a summary of the features of the original data without revealing any sensitive information and without violating secrecy. The stakeholders will trigger a data sharing request to the private ledger, right after the scan of the metadata on the public ledger. If the stakeholders satisfy the access control list, they can access the pointers stored on the private ledger. Then, they will refer the pointers back to the cloud storage to acquire the related data (after correct and valid decryption). Here the corresponding decryption abilities used to recover the encrypted pointers and the data on private ledger and backend will be granted via the trusted hardware based Blockchain wallet. This wallet could combine the use of cryptographic tools, like ABE, searchable encryption and proxy re-encryption. We note that this will be further explored and investigated in WP4.

3.2.1.2 Use Case #1: Enhanced Public Safety Data Sharing model

Before documenting the data sharing model for the public safety use case, we need to document the security and privacy requirements that need to be considered for secure data sharing. Hence, the next section elaborates on the aforementioned requirements that need to be consider in the public safety use case data sharing, and then, the internal and external data sharing models are detailed.

3.2.1.2.1 Security and Privacy requirements

The security and privacy requirements that must be satisfied in the context of this use case are described as follows:

1. ***Data confidentiality.*** Throughout the architecture for the use case of the DAEM demonstrator, we have identified in Section 3.1.2.1 several data flows spanning from front line edge devices to all the way back to ASSURED ledger and the ICT and cloud-based infrastructures. That is, ASSURED needs to guarantee that the confidentiality of the data flow will be protected from and within different entities. More specifically, the streaming and sensor data should be protected within the edge devices through ASSURED attestation, and when they are sent to network gateway, they should be safeguarded from system outsiders, e.g., network eavesdroppers. Similarly, the operational layer's data, including the data handled by CIO, internal operator and admins, should be protected and accessed only by authenticated users under pre-defined policies. Finally, any additional data flow towards the DAEM cloud server, ASSURED DLT and external stakeholders should be encrypted as well.

2. ***Mutual Authentication.*** The use case also requires identity authentication for all involved entities. That is, the communication among all the engaged entities should be securely authenticated and data transfer can take place only if identity verification has been performed. Crucially, devices and gateways should be mutually authenticated, the gateways and the CIO, admins and operators should authenticate each other. Mutual authentication will be performed in a twofold manner. In case of internal actors, authentication will be achieved through attestation, whereas for external actors through the Blockchain. Similarly, data access from DLT and DAEM cloud should be granted via identity check.

3. ***Access control and data encryption.*** Data encryption is a topmost requirement and need to be complemented by specified policies and access control mechanisms (e.g., ABE) in order to guarantee that only valid data retrievers can decrypt them. In addition, when data is required to be transferred, it will be encrypted by the current data holder, e.g., the gateway. For those front-line edge devices, which cannot support heavy data encryption techniques, we will either consider using lightweight encryption techniques and

Funded by the Horizon 2020 Framework Programme of the European Union

primitives, e.g., AES256, or employing TLS/SSL connection between devices and gateway for secure data transfer.

4. ***Data encryption at rest.*** All data will be encrypted and stored in DAEM server (encryption at rest), but they will be decrypted in order to perform the necessary data analysis. Since the data is stored in encrypted format, we will need to have a fast but secure way to perform data query and search over the "encrypted" database.

5. ***Secure data sharing with external stakeholders.*** The current data flow structure suggests interactions between external stakeholders and DAEM internal system. We will need to guarantee and provide secure data sharing and the corresponding monitoring for the sharing within DAEM framework.

6. ***Data anonymization and privacy-preservation.*** The anonymization and privacy protection features are desirable as well. However, in the DAEM use case, we will only need to consider the anonymity in data collection for external stakeholders, i.e., guarantees that the external stakeholders will not be able to infer the origin of the collected data (e.g., the stream data was captured by a device with ID382).

7. ***Other desirable requirements.*** The last-but-not-least goal is to make sure that the operational status of devices can be verified and monitored. Note that this goal will be met through the deployment of the attestation mechanism and the corresponding data flow.

3.2.1.2.2 Data Sharing Models

Considering the security and privacy requirements and following the general data sharing structure defined in a previous section, we elaborate below on the internal and external data sharing models of the public safety use case.

Internal data sharing: The internal data sharing is quite naturally shown as the data flow: the lower-level entities share the collected data to the higher-level entities. In the DAEM use case, front-line devices directly collect and send data to gateways which later send the data to operational level. It has to be noted that, in the public safety case, the devices will not directly share data with each other and with those devices from different domains, e.g., in different buildings, or in the same building. The internal data sharing follows the following steps, which are presented also in Figure 9.

I. At the operational level, there are three main entities: CIO, admins, and internal operators (having access to lightweight edge analysis engine). Thus, the internal data sharing will consider the data sharing among them. Hence, from different locations, in which the DAEM admins are monitoring and managing the data sharing events, a data sharing request can be made.

II. The admins have pre-defined a policy access list and this list will be merged on smart contract to auto-check if different domains'/areas' operational level's entities could share the operational data with each other. In this setting, the internal data sharing will be granted permission by either access token or auto-policy-check, and the valid entities (either passing the authentication or satisfying the policy check) can retrieve data from the DAEM cloud if needed.

III. Based on the requests sent by one or more operational entities, if the auto-check is successful, the entities can obtain the pointers for accessing other areas' data.

IV. With an access token given via valid authentication, an internal party (acting as attestation auditor or verifier) can directly obtain a specified local entity's attestation data. The system can control the attestation data sharing via access token.

Funded by the Horizon 2020 Framework Programme of the European Union

Note that internal data sharing will not consider **revocation and sharing-time limit** at this stage, which means that the DAEM enables internal parties to access data anytime as long as the parties are authenticated themselves. This is mainly due to the fact that enforcing time limits on the usage of data – from Data Seekers – requires also strong traceability and sovereign data provenance mechanisms that can be depicted through additional types of smart contracts. However, this is not a requirement in the context of the envisioned use cases and it is left as an open research question that ASSURED will explore in WP4 for further enhancing the off-chain data management functionalities.

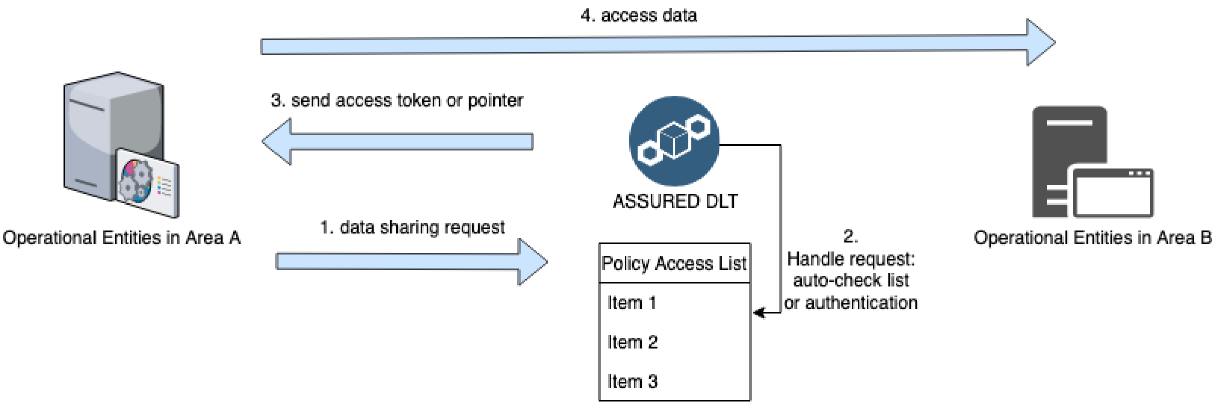


FIGURE 9: *INTERNAL DATA SHARING IN SMART CITIES*

External data sharing: The initial step for the external data sharing is for an external entity to search over the metadata on the ASSURED public ledger. If this entity wants to proceed to collect the detailed data but does not appear in the previous access control list, the entity sends a one-to-one data sharing request in order to access the private ledger. This means that an access control related smart contract will be generated (e.g., asking for a recorded CCTV stream data for a specified date and time) which is currently not available. By one-to-one we mean that only one party will request data sharing to DAEM per time, i.e., this smart contract will refer to the specific external entity and only. Thus, it will not be used to manage access to data for other external entities. In this context, the following steps are taken and illustrated in Figure 10.

I. The DAEM admins accept the request and create a smart contract on data sharing with the external stakeholder by building a data sharing record, that is added in the data sharing policy. From now on, the data request can be automatically handled by the defined smart contract.

II. In the contract, the time slot of data sharing, what are going to be shared, data request party information, and other information, will be clearly written. In this case, the externally data sharing consent can be given by DAEM, so it will be one-to-one consent (e.g., DAEM to Greek Police, DAEM to Greek Fire Department).

III. The contract enables the stakeholder to locate the pointers stored on private ledger.

IV. The stakeholder can then gain access to the cloud to obtain the requested data. Once the data is taken by the stakeholder, the contract will record a “finish” status on this data sharing event and close the event.

Note that the data sharing should be restricted to a specific time slot, e.g., for a week or a month, and the sharing rights should be further revoked by the contract. In addition, the data sharing among external stakeholders is out of consideration in this project.

Funded by the Horizon 2020
Framework Programme of the European Union

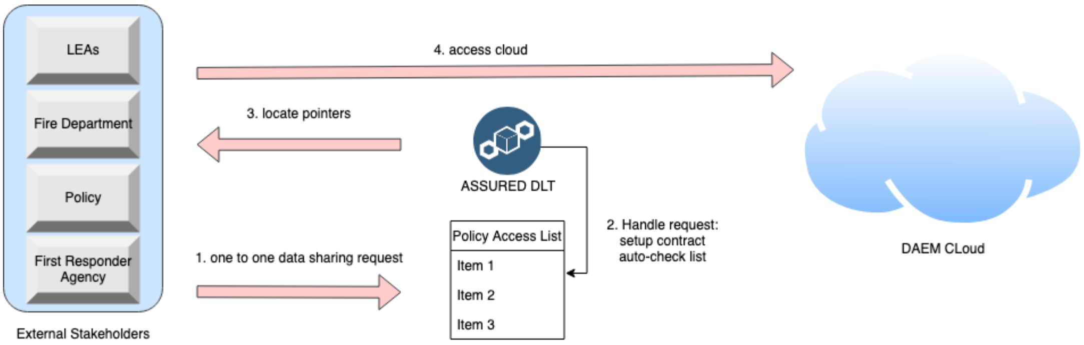


FIGURE 10: EXTERNAL DATA SHARING IN SMART CITIES

3.2.1.3 Use Case #2: Smart Manufacturing Data Sharing Model

This section documents the security and privacy requirements that need to be considered for secure data sharing in the smart manufacturing demonstrator of ASSURED. In addition, the internal and external data sharing models are detailed.

3.2.1.3.1 Security and Privacy Requirements

Before documenting the data sharing models for the smart manufacturing use case, we need to document the security and privacy requirements that need to be considered for secure data sharing. The requirements that must be satisfied in the context of this use case are described as follows:

1. ***Attestation for edge devices and IoT Gateway.*** The custom software and algorithms on the IoT Gateway should be attested by system admins via the ASSURED attestation mechanisms. In this context, the edge devices should also be attested in order to provide evidence for their operational integrity to the IoT Gateway. In this way, an attestation chain is formed spanning from the higher levels of the use case architecture to the front-line edge devices.

2. ***ASSURED DLT-related requirements.*** Identity management and authentication of the data as well as access control to the internal database is to be provided from the ASSURED DLT for external parties who wish to access information about operational logs or analytics information. To do so, the external entities need to obtain a cryptographic hash and use the hash token to query the internal databases for secure data access. In this case, the systems of the smart manufacturing demonstrator will also employ an automatic data access policy check for data requestor, so that the data accessing and sharing could be done with minimal involvement by the administrators.

3. ***Encrypted data transfer.*** The data flows from the IoT Gateway towards the on-premises IT infrastructure should be encrypted either by using standard best-practices like TLS/SSL mechanisms or furthermore enhanced by ASSURED offerings for secure communication. In addition, the data transferred among real-time location systems, motion capturing systems, data aggregators, IPC, IoT Gateway should be also protected.

4. ***Encrypted data storage.*** The system data, including logs, real-time edge operational and attestation data, IoT Gateway attestation data, and the analysis data from the analytic engine should be stored in encrypted format in ASSURED ledger and local databases.

5. ***Authentication among entities.*** Within the smart manufacturing ICT systems, authentication is of high importance. Entities and devices, which communicate with each other, should be authenticated. This should be applied also for accessing operational and attestation data, as well as for accessing databases and the ASSURED ledger. For those entities that fail to authenticate, access and communication should be denied.

Funded by the Horizon 2020 Framework Programme of the European Union

6. ***Data integrity consideration.*** *T*he data flows, especially those before reaching IoT Gateway, should maintain their integrity and confidentiality, as well as to be fault and error tolerant, so that the IoT Gateway and the analytics engine can process valid data from the front line.

7. ***Other desirable requirements.*** In the context of real-time risk assessment and system management, smart manufacturing environment should support full time event logging in order to enable an auditor to track what happens within the ICT systems. In addition, it is desirable to adopt mechanisms that guarantee non-repudiation so that the system entities cannot deny performed actions, e.g., sending a “stop” command, or a warning, reporting a “close” position, and external entities cannot deny that they’ve already accessed data.

3.2.1.3.2 Data Sharing Models

Considering the security and privacy requirements and following the general data sharing structure defined in Section 3.2.1.1, we elaborate below on the internal and external data sharing models of the smart manufacturing use case (Figure 11).

Internal data sharing: In this use case, as mentioned previously, the data is stored on IoT gateways, and the data archive/snapshot/reports are recorded at the BIBA central database. Different IoT gateways and a central database will commit data sharing behaviors. The system administrator, being responsible for deployment and IT administration, will monitor the data access among the gateways and the database, and meanwhile, they will create and hold the access policy control list for internal data access. In this context the internal data sharing behavior is described in the following steps:

I. The admins are in charge of the deployment of new IoT gateways. After the deployment, a data access token will be generated for each gateway for the purpose of storing information into central database of the IT infrastructure. The access policy list will be filled and maintained for these access tokens and gateways identities in the identity & access management component.

II. After the IoT Gateway instantiation, the gateway can connect to the infrastructure in order to support the data gathering and processing from the robotic and movement systems. During this process, the IoT Gateways will be requested to provide attestation data through the ASSURED attestation mechanisms. The attestation data is stored into the ASSURED DLT and cloud in order to support external / internal audits of operational logs and data access. If the attestation is completed successfully, data from location and robotic movement systems will start being transmitted from the HRC workspace though the gateway in order to be stored in the backend IT infrastructure. In case the attestation fails, the device onboarding process is terminated. Note that the attestation process may be triggered also as a standalone process and not necessarily in the context of other data sharing processes. In addition, the attestation process applies also to other devices apart from the IoT gateways. Here, for simplicity reasons, we refer only to the attestation.

III. In addition to the database access ability setting, the Blockchain DLT in ASSURED will provide information whether a gateway could be able to access other gateways’ data. This will be captured by the definition of access policy list as well. The only initial check conducted by system admins is the physical introduction of new devices into the workspace and initial functionality of the new device within the workspace as described in step I. When a gateway A requires access to the data of another gateway B, the ASSURED contract will be used to check the policy list: if the access from A to B is enabled, then A will be given an access token to access B; otherwise, the request will be turned down. Again, the policy checking, and enforcement will be supported by the ASSURED DLT and the underlined smart contracts, through the identity & access

Funded by the Horizon 2020 Framework Programme of the European Union

management component. In this context, the system admins will consider data access revocation and renewal for each IoT gateway. This will be also reflected on the smart contract, e.g., terminating the ledger access rights for the IoT gateway. The access rights of an IoT gateway – accessing to other gateways and central database - will be terminated at some point, and a new request and access evaluation will be re-examined after the access-valid period.

IV. If a valid access token is delivered to the gateway based on the policy validation process of step III, the gateway can be connected to the IT infrastructure to acquire the necessary data. Upon the completion of the process, the data sharing event will be logged to the ASSURED DLT.

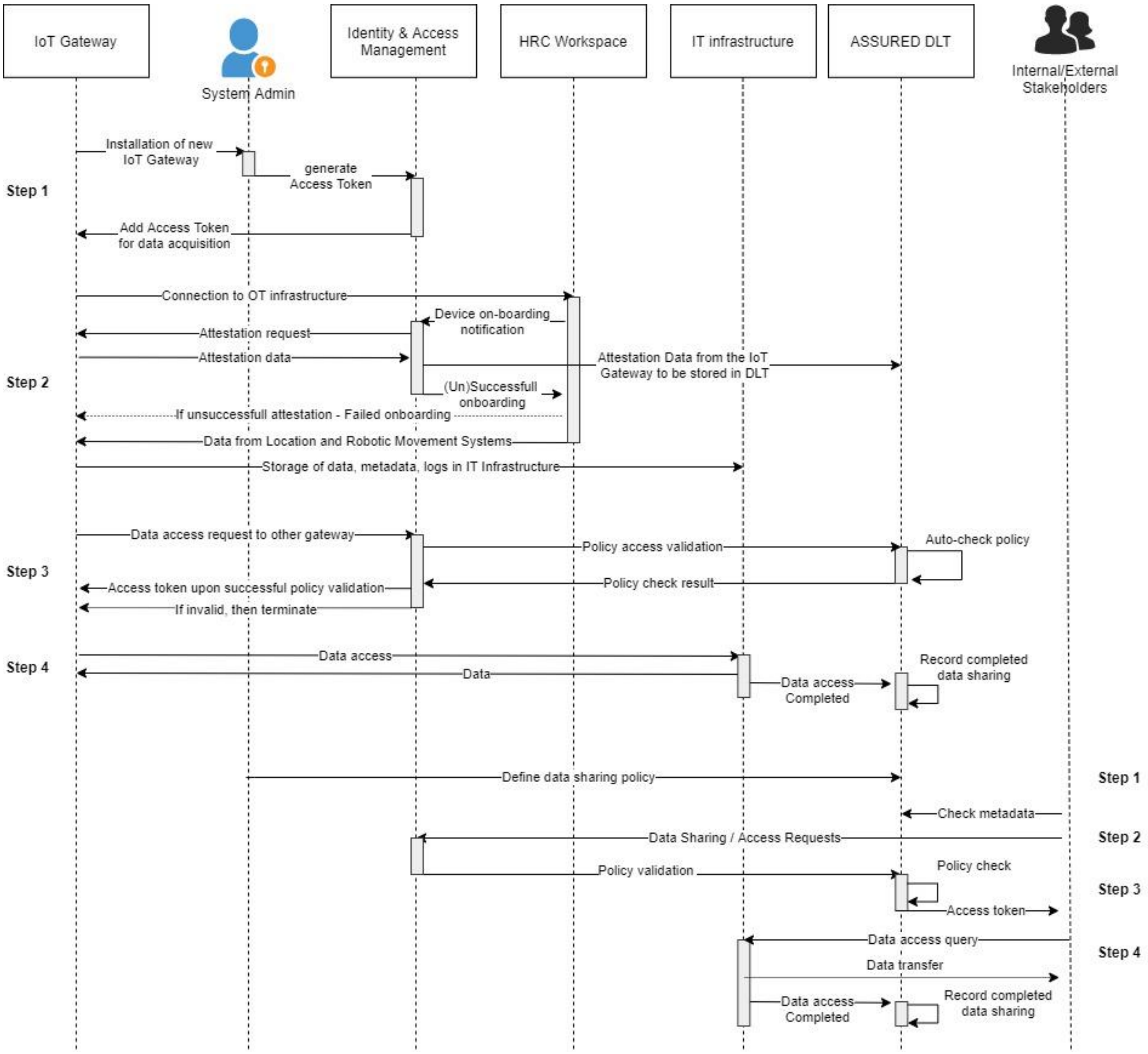

IoT Gateway
System Admin
Identity & Access Management
HRC Workspace
IT infrastructure
ASSURED DLT
Internal/External Stakeholders
Step 1
Installation of new IoT Gateway
generate Access Token
Add Access Token for data acquisition
Step 2
Connection to OT infrastructure
Device on-boarding notification
Attestation request
Attestation data
Attestation Data from the IoT Gateway to be stored in DLT
(Un)Successfull onboarding
If unsuccessfull attestation - Failed onboarding
Data from Location and Robotic Movement Systems
Storage of data, metadata, logs in IT Infrastructure
Step 3
Data access request to other gateway
Policy access validation
Auto-check policy
Policy check result
Access token upon successful policy validation
If invalid, then terminate
Step 4
Data access
Data
Data access Completed
Record completed data sharing
Define data sharing policy
Step 1
Check metadata
Data Sharing / Access Requests
Step 2
Policy validation
Policy check
Step 3
Access token
Data access query
Step 4
Data transfer
Data access Completed
Record completed data sharing

FIGURE 11: *INTERNAL AND EXTERNAL DATA SHARING IN SMART MANUFACTURING*

External data sharing: In this use case, as mentioned previously, the operational and attestation data collected data need to be shared via the ASSURED DLT to the external stakeholder of the supply chain.

I. The external data sharing behaviour is regulated by data sharing policies. These policies are defined by the system administrator and reside in the ASSURED DLT in the form of

Funded by the Horizon 2020 Framework Programme of the European Union

smart contracts. The predefined contracts should be used to check the access policy list to validate if a stakeholder, that requests access to data, is in the list.

II. External stakeholders will first view metadata stored on ASSURED public ledger and if they locate some data they prefer to acquire, they will send a data access request (along with the hash of that data stored on ledger – just the hash value but not the data itself) to the BIBA IT infrastructure for accessing data and logs stored in central database.

III. The request data will be shared with the verified and authenticated external parties via JSON Web access tokens (through API). If the parties acquire the token, they can query the API and further obtain the data. Note that, the data structure and integrity can be verified via the cryptographic hash which can be verified via the ledger data to guarantee that the received data is not tampered within the infrastructure. This implies that tokens can be only generated on-premises after internal confirmations (manually). The access tokens generation are time-bound which implies the tokens will be invalid if the authorization of the data requesters expires.

IV. By having the access token, the external stakeholder can perform data access and launch queries. The smart contracts will record the data sharing and the final status of the process on the ASSURED DLT.

3.2.1.4 Use Case #3: Smart Aerospace Data Sharing Model

This section documents the security and privacy requirements that need to be considered for secure data sharing in the smart aerospace demonstrator of ASSURED. In addition, the internal and external data sharing models are detailed.

3.2.1.4.1 Security and Privacy Requirements

The following requirements should be guaranteed in this use case:

1. ***Authentication between inner airplane devices and SSR.*** The devices and sensors installed inside and on the airplane should be securely communicating with the SSR for data transfer. To this end, authentication is required. In addition, it could be possible to require passengers to perform the smartphone authentication before they connect to the Wi-Fi provided by SSR.

2. ***Authentication between SSR and GSS, SSR and ACS.*** The authentication between the aforementioned entities is required in order to build secure communication among parties.

3. ***Data integrity and confidentiality.*** These properties are required for the data flows among different entities, including edge devices onboarding to SSR, SSR to GSS, to ACS. That is, the data transferred within airplane – between SSR and other on-plane devices, need to be protected. The same applies to the data transfer outside the airplane – between SSR and GSS, SSR and ACS, which need to be protected. Secure communication channel, e.g., TLS/SSL, encryption and crypto-enabled access control technologies should be considered so that only authenticated parties can receive and decrypt the data based on specified access policies and attributes. Since the data flow will finally end at the GSS, the stored data should be encrypted so as to maintain a long-term data security. Besides, data integrity check is needed to make sure the transferred data is not tempered and can be usable and credible.

4. ***Attestation between onboard devices - SSR, and GSS - System admins.*** The attestation should be done remotely, and the response and results should be stored on ledger for further data sharing.

Funded by the Horizon 2020
Framework Programme of the European Union

5. ***<u>Possible and flexible data sharing with external parties using Blockchain framework.</u>*** Operational and attestation data should be reflected on ASSURED DLT for the purpose of data sharing. The data sharing could be done via the DLT, in order to fulfil the fast and smart data sharing, by handling automatically data sharing requests and responses. The data sharing should be recorded on ledger for auditing purposes.

3.2.1.4.2 Data Sharing Models

Considering the security and privacy requirements and following the general data sharing structure defined in section 3.2.1.1, we elaborate below on the internal and external data sharing models of the smart aerospace use case.

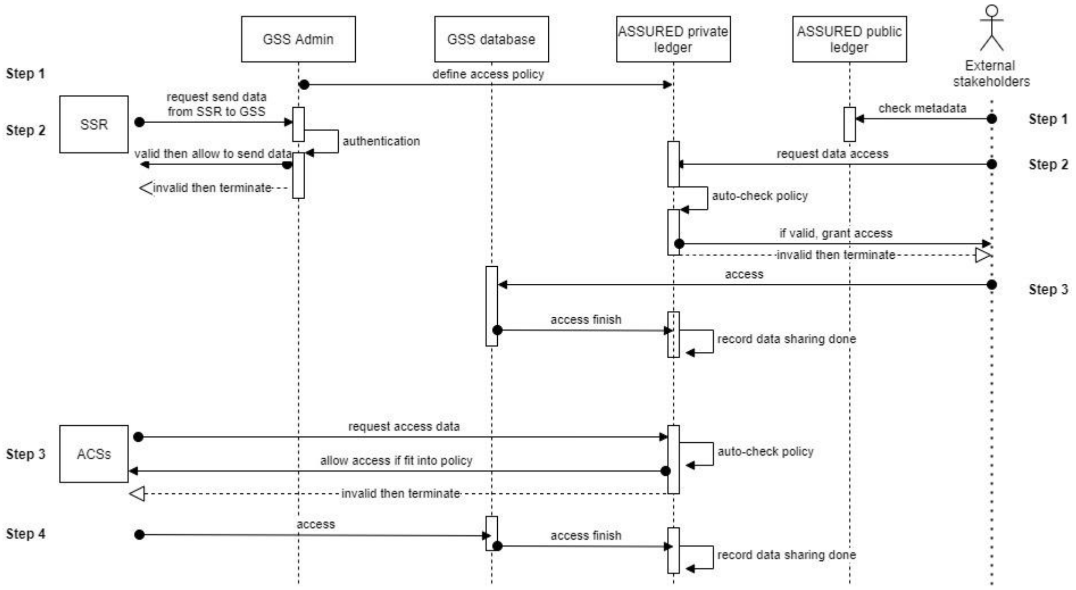
Internal data sharing: The internal data sharing of the smart aerospace use case will follow the main data flow illustrated in Figure 12.

I. The sensors within airplane are able to share data with the SSR after passing a successful on-boarding process through the ASSURED authentication mechanism. To do so, the SSR admin needs first to define the access policy to the ASSURED private ledger. If the authentication is successful and the access policy can be satisfied, the SSR can share its data with a GSS, and the GSS can share data with ACSs.

II. For the first case, the SSR first requests to share its data to a GSS admin. The admin first verifies the SSR identity, by using the authentication services offered by the ASSURED framework and checks if it is in the list of SSR and GSS pair. If so, the data can be transferred to the GSS; otherwise, turns down the request.

III. For the second case, one or many ACSs request data and the request is handled by the ASSURED DLT smart contract. If the pre-defined access policy, written on the contract, does not allow the data sharing, the contract will terminate the event; otherwise, data can be shared among the ACSs. In both cases, a set of predefined access policies is set, as described in step I.

IV. If the ACS gets a positive response for data sharing, an access request will be sent from the ACS to the GSS. The data access event will generate an internal data sharing event and the corresponding status will be logged to the ASSURED ledger.



Funded by the Horizon 2020
Framework Programme of the European Union

FIGURE 12: *INTERNAL AND EXTERNAL DATA SHARING IN SMART AEROSPACE*

External data sharing: The external data sharing focusing on sharing the demonstrator's data with external stakeholders if needed. For example, a scenario in which this could be possible would be if the data is requested by the authorities, such as insurance companies, for investigations. Such a data sharing process follows the following steps. The right part of Figure 6 illustrates the following interactions.

I. An external stakeholder will first look into the ASSURED public ledger's metadata to check if the GSS contains the data of their interest, such as the airplane full status throughout a given flight. If so, a data access request will be sent to the private ledger where the data is stored.

II. The data access request contains the necessary credentials to validate whether the stakeholder is in position to access the data. Based on this, a policy check is initiated, and the corresponding smart contract will be activated to validate the external stakeholder's access to the specific data based on the pre-defined policy. If the policy is satisfied, access to data is granted to the stakeholder.

III. If the access policy is satisfied, the stakeholder accesses the data residing in the GSS database. The logs of the procedure and the data exchange will be logged on the ASSURED private ledger.

3.2.1.5 Use Case #4: Smart Satellites Data Sharing Model

This section documents the security and privacy requirements that need to be considered for secure data sharing in the smart satellites' demonstrator of ASSURED. In addition, the internal and external data sharing models are detailed.

3.2.1.5.1 Security and Privacy Requirements

The following requirements should be guaranteed in this use case:

- ***Secure and trustworthy data communication between GS and CubeSat.*** To achieve this goal, ASSURED shall meet the following requirements: (1) secure authentication between GS and CubeSat; (2) secure data transfer between the parties, preferably using data encryption; (3) the communication channel may have noise, so that we need error correction or data recovery technique to back up the transmitted data; (4) data integrity is required to maintain a usable and meaningful transferred information.

- ***Secure data storage on GS.*** The sensitive data, e.g., passwords, stored in the GS should be protected via encryption. To this end, TPM will enable secure storage as a user or service can store any secrets (keys, passwords, or other sensitive data) associated with a TPM, and, when authorized, the TPM allows access to the user's secrets.

- ***Secure data access and sharing.*** The GS is allowed to share threat intelligence data, telemetry and logs with external stakeholders. To do so, these data should be put on a flexible and policy compliant DLT, to enable a fast and automatic data access. The data access policy should take the form of smart contracts so that access policies can be checked during data sharing.

- ***Assistant based attestation from CubeSat to GS.*** ASSURED attestation mechanisms should enable CubeSat to perform attestation to the GS. The attestation results can be verified and recorded on an accessible platform for internal entities. For external entities, ASSURED DLT will enable them to read the attestation data. In this case, access control policy enforcement is required.

- ***Possible extra and desirable requirements.*** The demonstrator would prefer to enable event recording to log operational events along with the corresponding data/status of the CubeSat and GS, so that all these recorded events can be stored in a bulletin board securely, e.g., via ASSURED DLT. CubeSat, GS, or any auditor can reference to the data on this "board" while examining the copies stored on CubeSat and GS. In this case, if CubeSat and/or GS make a change on its own data logs stored locally, this will be noticed via using the data stored on the "board". Of course, the board should not be published publicly and it should have some degree of data protection and data integrity guarantee.

3.2.1.5.2 Data Sharing Models

Digital security of smart satellites use case aims to enhance the security of CubeSats operation and communication in space. These, low earth orbit satellites that enable resource-limited organizations to operate in space deployed usually in constellations and many satellites are collaborating to perform a specific task. The main challenges involve protecting and ensuring availability, but also integrity and confidentiality of data exchanged. In this context, the following internal and external data sharing models are emerged to ensure the security of data sharing between CubeSats, Ground Station and external stakeholders.

Internal data sharing: The internal sharing is among CubeSat, GS and ASSURED components, and the operator of GS manages the process by monitoring the data sharing. More specifically, the following steps form the internal data sharing behaviour:

I. As mentioned in the data flow, the operator of GS sends data request to the CubeSat in order to directly transfer data (e.g., log, files download) via secure channel to the GS. GS and CubeSats have pre-set and pre-deployed identifier for mutual trust based on authentication via the ASSURED TPM based trust enabler. This pre-installed secret should be checked by ASSURED services in order for the secure communication channel to be created for further communication.

II. The data access request and the data sharing are logged on the ASSURED DLT. Note that for the direct data sharing between CubeSats and GS, no pre-defined policies and smart contracts need to be in place, due to the TPM-based instantiation of the secure and authenticated communication channel and the need of simplified operation of the CubeSats constellations.

III. As for the data sharing among ASSURED components, those will send a sharing request to the ASSURED DLT, and then, the smart contract embedded on the private ledger will check the access policy list to see if the components are allowed to access data.

IV. The data will be transferred to the components via secure channel if the access is granted. The internal sharing full request and response will be recorded and logged by the contract at ASSURED ledger for auditing purposes.

External data sharing: External data sharing includes sharing with external parties (e.g., Regulatory Authorities, Communication Service Providers or Integrators). Indicative scenarios for the need to share data with external stakeholders are in case regulatory authorities require data to confirm compliance with specific regulations. Another example could be the use of CubeSats for telecommunication purposes the data should be forwarded to communication service providers or communication service integrators. In this context, the following step are formed:

I. The sharing is handled and monitored by the pre-defined smart contract which should be designed by GS admin. In fact, the GS admin makes the preparatory work to create accounts and define a specific access and data sharing policy for each one of the external users.

II. The admin then takes advantage of the identity management system so as to create pre-defined accounts along with access rights token for the data requesters. This information is depicted to ASSURED private ledger. The access rights of each one of the external stakeholders are defined on smart contracts and enable them to be able to use their accounts credentials to authenticate themselves to the system.

III. Upon data request, a policy validation step takes place based on the defined smart contracts. If the policy if satisfied, the access token ii generated in order to request data stored on GS. This token is designed to be valid only for a specific period, meaning the data sharing is time limited.

IV. Once data are accessed, the corresponding sharing request and response will be logged on the ASSURED ledger at the end.

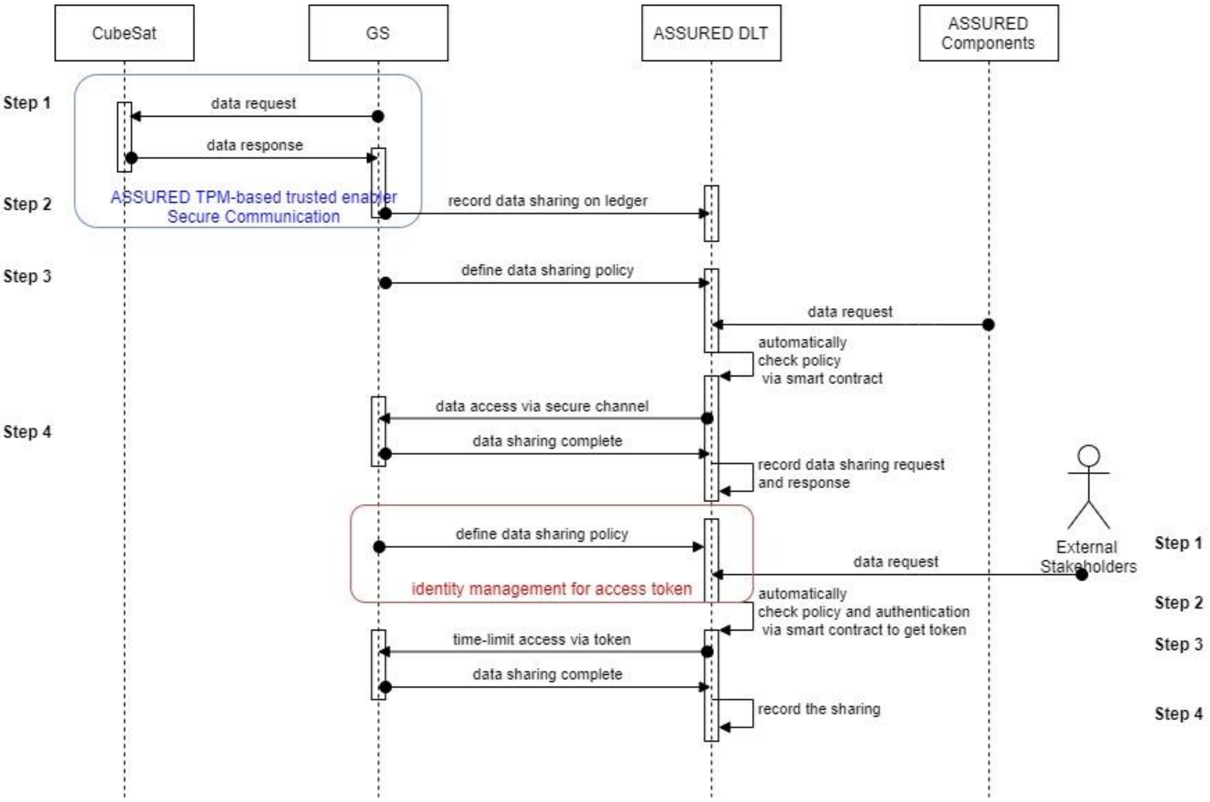

CubeSat
GS
ASSURED DLT
ASSURED Components
Step 1
data request
data response
Step 2
ASSURED TPM-based trusted enabler
Secure Communication
record data sharing on ledger
Step 3
define data sharing policy
data request
automatically check policy via smart contract
Step 4
data access via secure channel
data sharing complete
record data sharing request and response
define data sharing policy
identity management for access token
data request
External Stakeholders
Step 1
automatically check policy and authentication via smart contract to get token
Step 2
time-limit access via token
Step 3
data sharing complete
record the sharing
Step 4

FIGURE 13: *INTERNAL AND EXTERNAL DATA SHARING IN SMART SATELLITES*

3.3 THREAT INTELLIGENCE INFORMATION SHARING

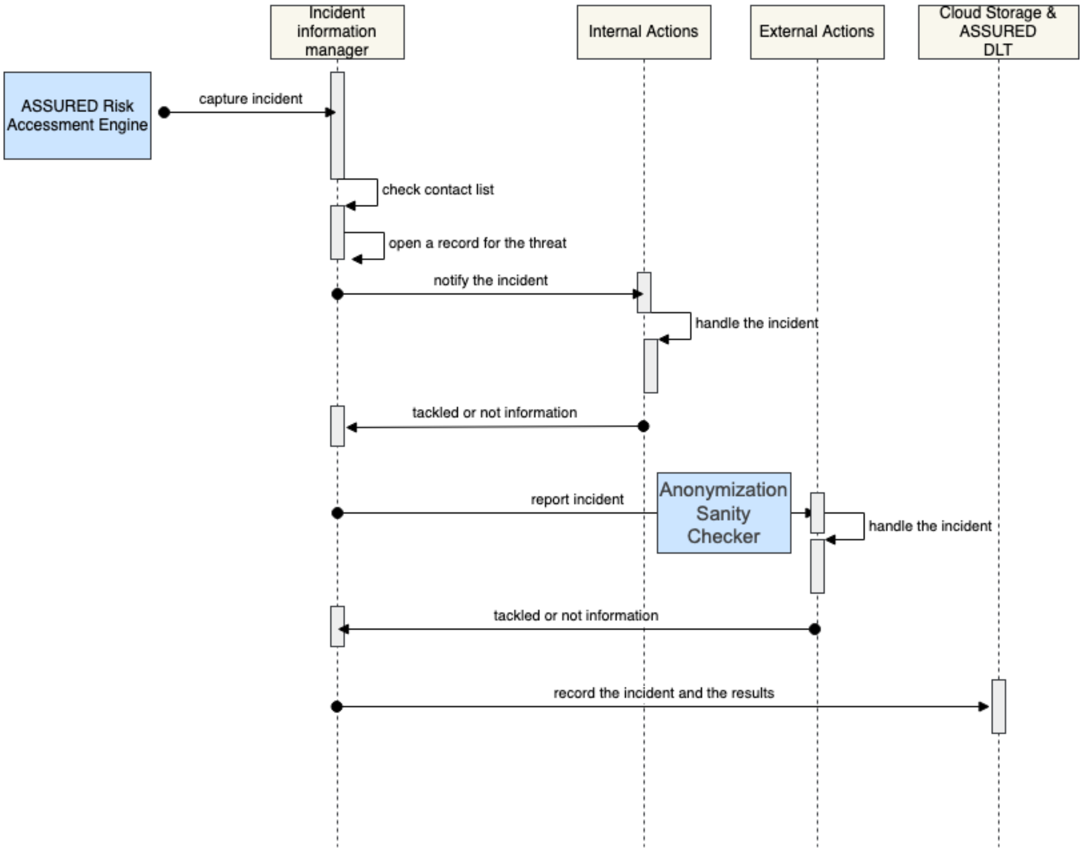
Besides the operational data that can be shared between the participating actors in the supply chain, another crucial piece of information to be securely shared is the security related data and mainly the attestation related data that depict the correct state of the attested device. As described in Chapter 2, the overarching goal is to enable components to make and prove statements about their state and actions so that other components can align their actions appropriately and an overall system state can be assessed, and security policies can be evaluated and enforced. By sharing such attestation data between all interested stakeholders, we can provide both the creation of trust-aware service graph chains where an actor can – at

Funded by the Horizon 2020 Framework Programme of the European Union

any point in time – certify/audit/verify the correctness of a software or hardware asset, but we can also fulfil the vision of a Blockchain decentralized market that allows enhanced knowledge sharing of increased operational threat intelligence, in supply chains, by supporting them towards the accountable reporting of newly discovered Advanced Persistent Threats (APTs).

In the context of ASSURED, such threat intelligence information (based on the result of the attestation and risk assessment processes) will be shared among necessary internal and external threat handlers and meanwhile, the reports and results of the actions will be recorded on ASSURED DLT and cloud-based backend for later auditing purposes. Since all use cases share similar threat information sharing behaviours, in what follows we summarize them (Figure 14).

Internal Threat Information Sharing. Threat reports will be created directly via the Risk Assessment engine that based on the system topology of the target SoS, the mixed-criticality services running and the type of hardware and software assets to be protected, it will output a risk assessment graph containing all types of risks and vulnerabilities (based on the adversarial model defined in D1.3 [100]) that can be exploited for attacking any of the safety-critical assets. Based on these reports, they will then be passed to the Policy Recommendation engine where an optimized set of security (attestation) policies will be provided for tackling such risks by attesting the required system validation properties [100]. These policies will be finally expressed as smart contracts for further enforcement to the edge devices. Furthermore, such identified threat reports will also be shared with the Attack Validation Component so as to be instantiated and further information can be extracted on the exact possible attack path to be exploited by an attacker. This type of transaction will also be recorded in the ASSURED DLT where threat reports and possible attack paths will be merged to threat records and their results (along with related information, e.g., time, date, locations, events, handlers, and final status) will be recorded on the ASSURED DLT. In this case, system administrator will always be able to monitor any incidents reported for specific types of threat records.


Incident information manager
Internal Actions
External Actions
Cloud Storage & ASSURED DLT
ASSURED Risk Accessment Engine
capture incident
check contact list
open a record for the threat
notify the incident
handle the incident
tackled or not information
report incident
Anonymization Sanity Checker
handle the incident
tackled or not information
record the incident and the results

Funded by the Horizon 2020 Framework Programme of the European Union

FIGURE 14: *GENERAL MODEL FOR THREAT INTELLIGENCE INFORMATION AND ATTESTATION DATA SHARING*

- ***External Threat Information Sharing.*** This is similar to the case of internal sharing, however, external stakeholders and third parties might need to get access to such threat intelligence information and security attestation related data. The main difference is that before any threat information is shared with external parties, it will have to be appropriately *anonymized* (Anonymization Sanity Checker in Figure 14 by employing either secure cryptographic primitives (e.g., group signatures, pseudonyms, etc.) or "blending" mechanisms such as k-anonymity and l-diversity) so as to reduce the leak of any sensitive information which could be linked to system devices and infrastructure. For instance, control-flow attestation might leak information about the type of operating system running in a device or the order of mixed-criticality functions be invoked – such information can lead to further exploits such as key extraction [111]. Such external parties can leverage such security related data to audit or access the logged threat records for certifying the correct state of the overall system topology.

Let us take an example in the context of the Smart Cities use case where such information might be exchanged for verifying the integrity of the deployed CCTV cameras. If there exists a threat incident, e.g., an intrusion attack to a CCTV camera on a building, or a fire incident detected from a region sensor, the ASSURED Risk Assessment engine, based on the received attestation report, will produce a threat incident report. The DAEM system admin will be notified about the threat and be able to read the report. And then it will check who are the internal and external threat handlers for this incident, e.g., an internal handler could be a network security engineer, and an external stakeholder could be a local fire department. After checking contact list, the DAEM admin will notify the handlers – for external sharing, the fire department notification message is filtered via sanitization and, thus, fire department has no information about the data sources and how the internal sensor layer layout looks like. Meanwhile, the admin will open an incident file and keep monitoring the status of the threat. Then the results of the actions (e.g., "threat is tackled") will be sent to the admin who will store the incident file with its results (as final report) on the DAEM cloud and ASSURED DLT.

By assuring auditable, security and privacy policy compliant actions, ASSURED guarantees that threat intelligence information sharing can enable the more efficient mitigation of such threats without, however, breaching the privacy of any of the involved actors and/or devices.

Funded by the Horizon 2020 Framework Programme of the European Union

4 SECURE DATA SHARING BASED ON BLOCKCHAIN TECHNOLOGY

In Chapter 2, we defined the security, privacy, and accountability requirements whereas in Chapter 3, we presented a detailed description of the data sharing profiles and information exchange for all use cases. In what follows, we are going to review the state-of-the-art in secure tools and platforms which can be used in the ASSURED context to achieve the requirements while guaranteeing consent provision (trusted consent) and management by all participating entities for sharing such sensitive information.

4.1 SYSTEM REQUIREMENTS AND ASSUMPTIONS

Before proceeding to review the state-of-the-art in techniques that can be used to capture trust consent and Blockchain-based data sharing, we first need to consider some requirements and assumptions for these technical building blocks. Note that mainly focus on the trusted hardware and Blockchain techniques. From Table 7, one can clearly see that there exist some assumptions and pre-conditions about using these techniques. For instance, we do not **consider some existing attacks on trusted hardware**, e.g., TPM, that an attacker can intrude into the TPM and compromise the stored keys, and we also assume that once a **cryptographic algorithm is working with or combining with the use of TPM, the TPM will help the algorithm to store and organize its key**. Likewise, in the Blockchain platform context, we do not consider the **existing attacks to the Blockchain and smart contract, e.g., Sybil and Eclipse attacks**, but only focus on the technical and functional supports given by the Blockchain.

TABLE 7: *REQUIREMENTS AND ASSUMPTIONS FOR TRUSTED CONSENT MANAGEMENT*

ITEMS	BUILDING BLOCKS	REQUIREMENTS/ASSUMPTIONS
1.	**TPM**	We assume that TPM is a fully trusted element that cannot be compromised. We also assume that the ASSURED framework will use the existing TPM2.0 version not a post-quantum TPM which is still under research. When a host machine starts to boot, its TPM is not controlled by any external entity, but it starts automatically in parallel to the CPU. The TPM can create attestations about the state of the host platform, e.g., certifying the boot sequence running by the host. Once the BIOS is loaded, the TPM takes control and measures the integrity of the OS Loader. The integrity measurements at each stage are made by creating a SHA-256 digest of the code to be loaded. This digest is stored in one of the PCR registers, which are initialised to zero. When access is requested to an Edge network, a signature of the PCRs, called the attestation, can be sent to the Edge device who forwards it to an external verifier that can verify the state of the platform. We further assume that there exists a simple and direct User Interface (UI) for the host to connect with the TPM, and meanwhile the UI can help system admin for deployment. The deployment and setting of the TPM is secure. We do not consider any attacks and vulnerabilities on the TPM in this stage.

Funded by the Horizon 2020 Framework Programme of the European Union

2.	**TEE**	We assume that both software and hardware Trusted Computing Base (TCB) are reliable and not compromised. And the coding executing in TEE is trusted, i.e., formally verified, and securely booted. Also, the isolated execution environment provided by TEE is protected against unauthorized accesses, such as accessed by privileged components, or side-channel attacks. We do not consider potential attacks and vulnerabilities on TEE.
3.	**TRUSTED HARDWARE WITH CRYPTOGRPAHIC TOOLS**	Since this part is related to the trusted hardware, e.g., TPM and TEE, we recognize the previously made assumptions in 1 and 2. And then, we suppose that the cryptographic tools can be supported by the trusted hardware. By “supported” we mean that trusted hardware can securely store and manage the keys of these tools, and when a tool user requests a key, the key can be retrieved safely from trusted hardware. Besides, the cryptographic tools, e.g., hash function, signature, and symmetric and asymmetric encryption, could be embedded within trusted hardware, in this case, the trusted hardware will act as a secure blackbox for the tools.
4.	**BLOCKCHAIN PLATFORM**	We assume that there exists a distributed, decentralized, and deployable Blockchain network (with nodes/peers) and platform for our project and the platform could be extended from some existing ones, e.g., Ethereum, Hyperledger Fabric. Within the Blockchain platform, we assume it should have common and necessary functional components, providing hash function, Merkle tree, digital signature (e.g., Elliptic Curve digital signature algorithm), block (with a fixed storage size, e.g., 1MB), wallet, mining and validation mechanism, transactions and transaction model, consensus algorithm, incentive, supports for smart contract (e.g., deployment and execution), etc. We further suppose that the Blockchain platform can provide public and private ledgers, but also a channel for private conversations among a group of ledger users (forming private ledger). There should exist Blockchain admins to define and set up access control policy (e.g., identity management, access control list) for ledger access and block data operation, e.g., if a user is able to reach a ledger and some blocks on that ledger. Further, a cloud-based backend is linked to the Blockchain platform. The connection may be various, but we assume that there exist pointers stored on ledger, pointing back to the data storage on the backend, which means that one, given a pointer, can gain access to the data recorded on the backend. And besides, the hash of the data could be stored on ledger for integrity check later. We think about the off-chain storage that could be implemented via the cloud-based backend in this stage. We don’t consider a “fully” public (but permissioned, e.g., via the use of membership mechanism for access) Blockchain platform in the project, which means that the Blockchain platform will not be accessed by all Internet users in an open network. We also don’t consider the use of cryptocurrencies. We don’t yet consider efficiency, throughput, scalability, resource consumption, fault tolerance, and usability for the Blockchain platform in this stage; this would be further analysed in the context of D4.1 [105]. We also ignore the possibility of Blockchain forking. The above could be properly handled after we

Funded by the Horizon 2020
Framework Programme of the European Union

		decide which Blockchain platform we will focus on for development.
5.	**BLOCKCHAIN WITH TRUSTED HARDWARE**	The combination of Blockchain platform and trusted hardware will follow both previous assumptions (stated in 1, 2 and 4). And besides, we assume Blockchain users (e.g., peers) can be equipped with trusted hardware, and the hardware can provide secure execution for some on-chain operations, e.g., signature, attestation. We here don't consider that Blockchain users may work together to harm or break trusted hardware's security and trust features.
6.	**BLOCKCHAIN WITH CRYPTOGRAPHIC TOOLS**	We assume some cryptographic tools (e.g., related to encryption, authentication, hash function) will be used within Blockchain platform (by users). The purposes of using these tools are to maintain the corresponding data confidentiality, secure authentication, data integrity and other requirements. We assume that the Blockchain platform users (e.g., peers, devices) can execute the tools correctly and safely, and the corresponding setup and key distribution for the tools are done securely for the users. And the execution results of these tools may be recorded on the Blockchain ledger, e.g., an encrypted pointer is stored on private ledger. We don't consider that Blockchain users may collude together to break the cryptographic tools' security in this stage.
7.	**SMART CONTRACT**	We assume that there is a conversion approach to convert business/action logic, e.g., data sharing behaviors, attestation policy and actions, into smart contract. And smart contract (expressed into programming languages, e.g., JavaScript, Golang) can be further deployed on Blockchain ledger correctly and executed to fulfil the corresponding logic. The input and the output of smart contract may be stored on Blockchain ledger for auditing purpose.
8.	**BLOCKCHAIN AND SMART CONTRACT SECURITY AND PRIVACY ASPECTS**	In the current stage, we do not consider the attacks and the potential vulnerabilities (as well as the corresponding countermeasures), e.g., sybil attacks, re-entrancy attacks, on Blockchain ledger and smart contract, in terms of network, software and deployment/execution levels. We note that this type of consideration could be put in the later system integration stage. But we consider some aspects related to the data confidentiality and data integrity for the data stored on ASSURED DLT ledgers. Besides, we also consider the use of trusted hardware (as the way of enhancing trust), e.g., TPM, to safeguard the authentication, usage of smart contract, and the supports for on-chain cryptographic operations, e.g., attribute-based encryption. We may consider the anonymity of DLT users' identity. And we further assume that some classic features, including, consistency, tamper-resistance, integrity, transparency, etc., can be guaranteed since we say that the Blockchain platform is able to provide common and necessary (security) components, like signature, hash function and consensus provided by Hyperledger Fabric platform.
9.	**OTHER CRYPTOGRAPHIC TOOLS**	As for the cryptographic tools, e.g., ABE, SE, we first do not consider the corresponding feasible attacks on them, e.g., post-quantum computer attacks, side channels and other physical level attacks. We then assume that these tools can be securely deployed and executed (by users), and their key generation and assignment are protected.

Funded by the Horizon 2020 Framework Programme of the European Union

10.	INTERFACES FOR BUILDING BLOCKS	In the ASSURED project, we will make use of many types of technical components as part of the overall Blockchain infrastructure. The endmost goal is the provision of appropriate interfaces that will enable the interoperability of such components and provide a visible and handy UI for system admins, users, and developers.

4.2 CURRENT ENFORCEMENT STRATEGIES

4.2.1 Trusted Hardware for Consent Management and Data Control

The use of cloud computing infrastructure has increased the need for new methods that allow data owners to share their data with others securely taking into account the needs of multiple stakeholders. For instance, in Blockchain applications the data owner should be able to share confidential data while delegating much of the burden of access control management to the cloud and trusted enterprises.

As discussed in [1], a Trusted Platform Module (TPM) allows stronger privacy of data compared to software-based security protocols. It provides stronger protection and management of cryptographic keys that allow data owners to securely store data on untrusted cloud services.

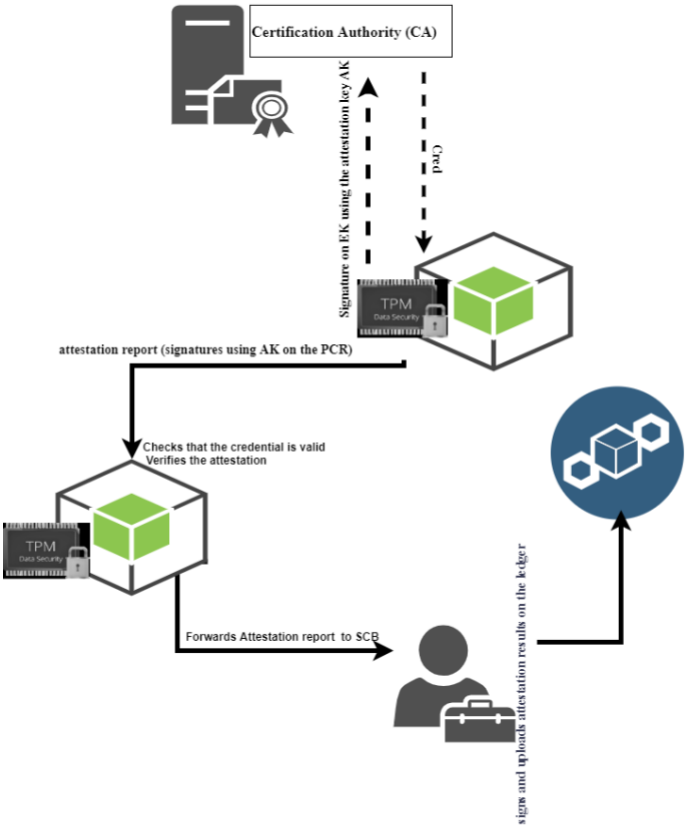

Certification Authority (CA)
Signature on EK using the attestation key AK
Cred
TPM
Data Security
attestation report (signatures using AK on the PCR)
Checks that the credential is valid
Verifies the attestation
TPM
Data Security
Forwards Attestation report to SCB
signs and uploads attestation results on the ledger

***FIGURE 15:** COMBINING TPMS WITH BLOCKCHAIN TECHNOLOGY*

Funded by the Horizon 2020
Framework Programme of the European Union

The TPM is used as a Root-of-Trust-for-Measurement by providing an attestation service on behalf of its host platform, and it also supports general cryptographic functionalities, such as symmetric and asymmetric encryption, digital signatures, key exchange, random number generation, hash functions and message authentication codes, for many other applications. Following its design goal, the TPM attestation service will eventually make the whole platform trustworthy from an external entity's point of view, when this entity remotely communicates with the platform and verifies the attestation information reported by the TPM. The attestation service is based on digital signatures: the TPM measures the platform software/firmware state, then provides a digitally signed software/firmware measurement report to the external entity (also called the external verifier). The signing and verification key pair used for this service is named as an Attestation Key (AK). In ASSURED framework, the attestation service can be much broader than just reporting the software/firmware state, e.g., it can be used to reliably provide a time stamp, a Blockchain cryptographic key or a key certificate. The TPM can also supports consent managements through the cryptographic mechanisms presented below.

4.2.1.1 TPM-based CA Credentials

When attesting new devices with embedded TPMs, a verifier needs to verify that the given attestation report was created by a genuine TPM even if it is unidentified. To meet this requirement, an attestation Certificate Authority (CA) (also called a credential provider) is involved to authenticate that the AK holder is a genuine TPM and then to issue a credential to the AK. This was presented in [2] and has been specified in the TPM specifications. The CA credential (Cred) is needed whenever a new edge device wants to get access to a Blockchain ledger and executes a smart contract as shown in Figure 15. Any administrator who can be the security context broker verifies that the TPM has a valid CA credential on the TPM public key. If this verification is successful, then any Blockchain user will be able to authenticate and attest the new edge device (**Secure Device on-boarding**). The attestation report is then forwarded to the security context broker who verifies the reported attestation result. Upon successful verification, the broker provides the new edge device with Blockchain keys that allow the device to access the ledger, securely download and execute the smart contract and upload its attestation results on the Blockchain ledger.

A TPM AK credential issuing scheme involves three entities: **a set of TPMs, a set of hosts and a credential provider**. The credential provider has a public and secret key pair (cpk; csk), which is used for a signature scheme. Each TPM has a public and secret Endorsement Key (EK) pair (epk; esk), which is used for an asymmetric encryption scheme. The EK is usually certified by the TPM manufacturer. The credential provider has access to an authentic copy of the public endorsement key and its certificate. The TPM also has a public and secret AK pair, which is used for a signature scheme (either a conventional signature scheme or a Direct Anonymous Attestation (DAA) signature scheme).

For the TPM to use its attestation key to create signature, it should have a valid credential from a trusted CA. The CA creates a credential on the TPM attestation key by using its secret signing key csk to sign the public part of the TPMs AK that originates from the TPM with a certified EK. The CA then uses “make credential” function to create a credential blob which typically contains an information used to decrypt the actual credential from the credential provider. The CA then sends an encrypted credential to the host who decrypts the credential with the help of the TPM. Basically, the ciphertext (encrypted credential) is delivered to the host that forwards it to the TPM who unwraps the credential and returns a decryption key to the host to decrypt the credential. This TPM operation is called “activate credential”. The host decrypts the credential using the decryption key provided by the TPM, then verifies the credential i.e. verifying that the signature provided on the TPM public attestation key under the CA public key is valid. Finally, the host stores the credential and loads it whenever the platform (TPM + Host) needs to generate signatures.

4.2.1.2 Attribute-based Signatures & TPMs

In addition to checking if a TPM has valid CA credential, sometimes a verifier may also verify that the TPM possess specific attributes needed to access safety-critical applications and information. For example, a TPM is only able to access or upload some sensitive data on the block chain ledger if it has some identified attributes. Thus, the TPM must convince a verifier that it possesses some attributes even without revealing its identity nor the attributes. This is needed in ASSURED privacy-preserving context. **Attributes may be embedded in the TPM, others may be defined by an external entity. Thus, in ASSURED framework, we need to consider the cases where the TPM supports attribute-based signature and attribute-based encryption.**

In attribute-based signatures (ABS) presented in [3], verifiers are convinced that the signer owns a set of attributes satisfying a so-called signing policy, however, they do not learn the signer's identity, nor the set of attributes used, and thus provide signer's anonymity within a set of users holding policy-conforming attributes. In ABS, users cannot forge signatures with attributes they do not possess even through colluding. In ASSURED framework, for each attribute granted/embedded in an edge device, we may correspond a secret attribute key stored inside the TPM and originally generated by some trusted authority using its master secret key SK as shown in Figure 16. To prove that a device possesses a certain attribute, we may let the TPM sign the attribute using the corresponding attribute key stored in the TPM.

TPM's attribute-based signature is a form of zero knowledge proof generated by the TPM using the corresponding Attribute Key <j>. The signature is then sent to an external verifier who verifies the signature under the master authority public key PK and verifies that the device has certain set of attributes but without knowing the exact attributes for privacy preserving purpose. In the ASSURED framework, a TPM may also support Direct Anonymous Attestation with attributes as presented in [5].

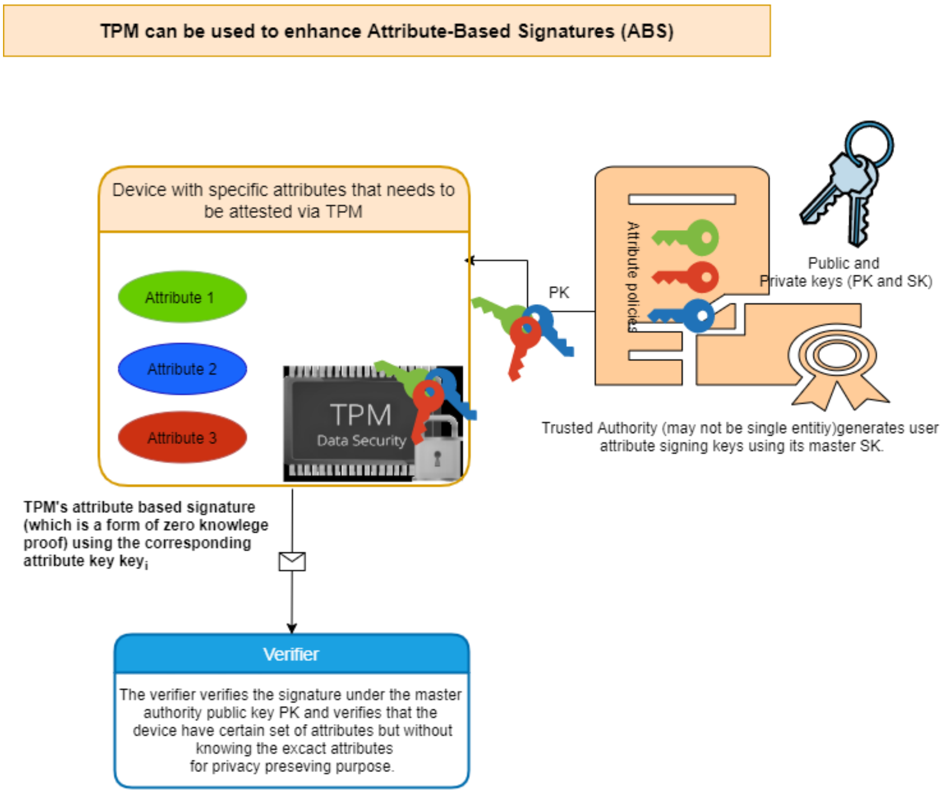

TPM can be used to enhance Attribute-Based Signatures (ABS)
Device with specific attributes that needs to be attested via TPM
Attribute 1
Attribute 2
Attribute 3
TPM
Data Security
PK
Attribute policies
Public and Private keys (PK and SK)
Trusted Authority (may not be single entitiy)generates user attribute signing keys using its master SK.
TPM's attribute based signature (which is a form of zero knowlege proof) using the corresponding attribute key key$_i$
Verifier
The verifier verifies the signature under the master authority public key PK and verifies that the device have certain set of attributes but without knowing the excact attributes for privacy preseving purpose.

FIGURE 16: *TPM PROVIDED ATTRIBUTE-BASED SIGNATURES*

Funded by the Horizon 2020 Framework Programme of the European Union

4.2.1.3 Attribute-based Encryption & TPMs

In ASSURED framework, a TPM may be used to support Attribute-based Encryption (ABE) where the secret key of a Blockchain user and the ciphertext are dependent on the user attributes. In such a system, the decryption of a ciphertext is possible only if the set of attributes of the user key matches the attributes of the ciphertext. The Blockchain user (being authenticated to a ledger) can only have decryption rights to the encrypted data stored on the ledger only if the user possesses some attributes that make correct decryption as defined in [4]. This allows data owners to share data safely with the designated users rather than a third party or other users.

The TPM supports ABE that is mainly used in order to ensure legitimate attribute-based access control to sensitive encrypted data. A trusted authority which can be many entities, generates the user attribute keys using the authority master key SK as shown in Figure 17. The trusted authority then sends the encrypted attribute symmetric keys together with the attribute policies to the Blockchain edge device which contains an embedded TPM. **The TPM stores the encrypted attribute symmetric keys and, using its own asymmetric decryption keys, decrypts the corresponding attribute symmetric keys and outputs them (from inside TPM) to the edge device/host when requested in order to decrypt the cyphertext stored on the ledger**.

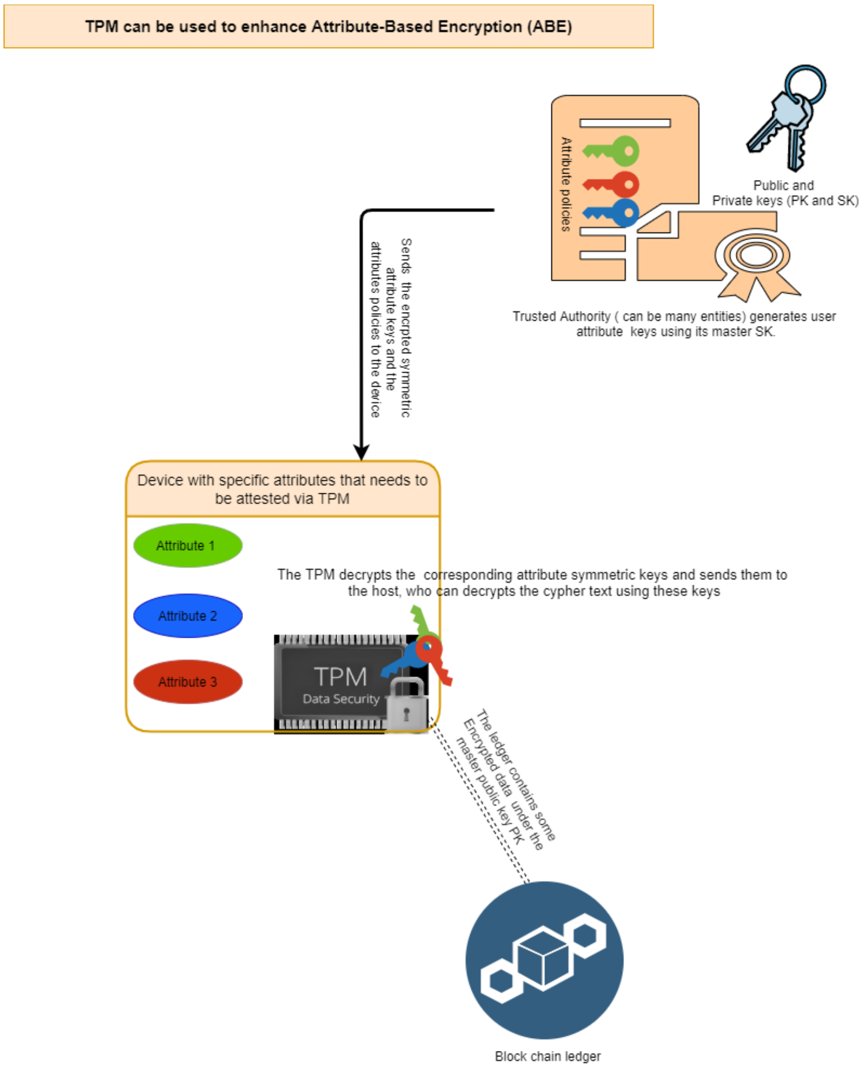

TPM can be used to enhance Attribute-Based Encryption (ABE)
Attribute policies
Public and Private keys (PK and SK)
Trusted Authority (can be many entities) generates user attribute keys using its master SK.
Sends the encrypted symmetric attribute keys and the attributes policies to the device
Device with specific attributes that needs to be attested via TPM
Attribute 1
Attribute 2
Attribute 3
The TPM decrypts the corresponding attribute symmetric keys and sends them to the host, who can decrypts the cypher text using these keys
TPM
Data Security
The ledger contains some Encrypted data under the master public key PK
Block chain ledger

FIGURE 17: TPM PROVIDED ATTRIBUTED-BASED ENCRYPTION

Funded by the Horizon 2020 Framework Programme of the European Union

4.2.1.4 Trusted Execution Environments

A Trusted Execution Environment (TEE) is a tamper-resistant, isolated processing environment in which applications can be securely executed. TEE is a new approach to realize trusted computing, i.e., secure computation, privacy and data protection. Other than TPM that relies on a crypto processor offering cryptographic keys and services such as encryption and signing, TEE provides a confined environment that allows the execution of authorized code only and protects its running states and stored assets, e.g., CPU registers, memory, and sensitive I/O, from observation or tampering by untrusted parties [6].

Despite a variety of TEE solutions from both industrial and academic worlds, there are several common security primitives considered as core properties of TEE, i.e., security boot, isolated execution environment, secure storage, remote attestation. Take ARM TrustZone as an example. Figure 18 shows the high-level architecture of TrsutZone based TEE. TEE measures the trustworthiness of code to be loaded and executed in TEE through secure boot (❶). During the runtime, TEE executes the code in an isolated computing environment with protected memory, so that untrusted components, including privileged software like OS and hypervisor, cannot directly access the assets or intervene the processes within TEE. The inter-communication between normal world and secure world takes place through pre-defined, sanitized interfaces. Besides, TEE leverages remote attestation to prove its trustworthy status, not only static status but also dynamic status, to third parties. The attestation report is signed by trust anchor, e.g., TPM, so that remote verifier can check the integrity and authenticity of attestation report (❷).

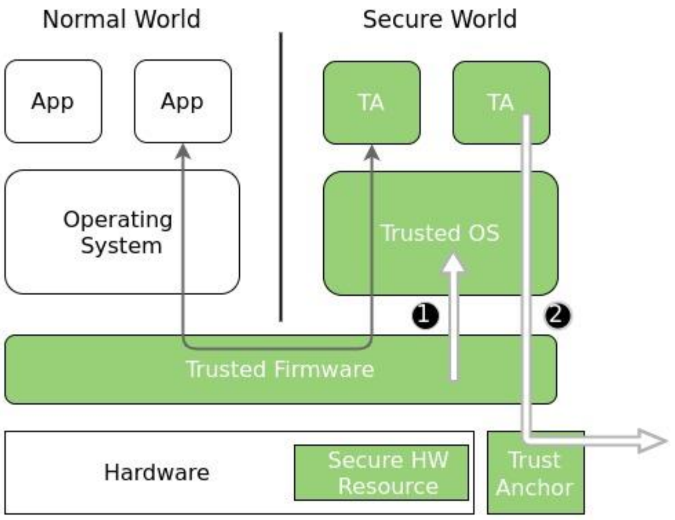


FIGURE 18: *GENERAL TECHNICAL ROADMAP OF TEES INTEGRATION IN DLT*

Several TEE designs have been proposed for different computing platforms, from embedded systems to high-performance computer systems. They leverage software or hardware features to build isolated execution environments called enclaves in different execution levels, such as user space enclave and kernel space enclave. For demonstration purpose, we introduce two mainstream industrial TEE solutions, namely Intel SGX, and ARM TrustZone.

Intel SGX [7] realizes the isolated execution environments by microcode-level implementation. It can create multiple instances of user space enclaves but not support kernel level code. SGX verifies the enclave's signature signed by the enclave author before initializing it to ensure the authenticity and integrity of enclave image. Each enclave is associated with a signature structure SIGSTRUCT consisting of author's public key and other metadata, which is signed by the author. Besides the static trust, SGX supports local and remote attestation to prove the dynamic status to third party, i.e., the enclave is running in a legitimate SGX platform, and the enclave execution has not been tampered with.

In contrast, ARM TrustZone divides the whole system into two domains named secure world and normal world and implements hardware isolation between these two domains. To do so, a TrustZone enabled processor runs in either a secure or a non-secure state, and a part of memory is reserved as secure memory that can only be accessed by secure world. Unlike Intel SGX, TrustZone provides a single TEE supporting both use-space applications and privileged code. Note that TrustZone itself only ensures an isolated execution environment, but not inherently trusted, because of the lack of trust anchor. This problem is usually solved by incorporating with a hardware trust anchor e.g., a TPM or secureROM that contains unique device keys to establish trust by attestation.

TEE allows services that process sensitive data or perform critical tasks to securely execute in an isolated container and attest the trust state to third parties. The nature of TEE initiates new lines of research that integrate TEE into diverse domains to tackle the performance or security limitation. One example is the combination of TEE and cryptographic techniques, such as Homomorhpic Encryption (HE), and Searchable encryption (SE). HE schemes enable untrusted entities to carry out operations on encrypted data but suffer from high performance overhead and unverifiable conditional variables issue. [8], [9], [10] proposed TEE-based HE schemes that perform some sensitive HE primitives inside isolated and remote attested enclaves in order to deduce performance overhead but keep a high assurance of data privacy. Another cryptographic technique called SE allows data search and query over encrypted data. Recently several TEE-based SE schemes [11], [12] have been proposed to address the security limitation of existing software-based SE schemes, namely search pattern leakage and query range leakage. The initial idea is to use the isolated environment provided by TEE to perform search processes, so that unauthorized entities cannot obtain any information about search pattern and query range.

TEE also owns properties that are complementary and appealing to Blockchain. A major drawback of current Blockchain systems is the lack of transactional privacy, as all data is replicated on all nodes in Blockchain network. Many TEE-based Blockchain schemes [13], [14], [15], [16], [17] have been proposed that offloads the execution of Blockchain applications and smart contracts into TEE to ensure the confidentiality of transaction data. Besides, TEE is also used to improve the existing Blockchain consensus mechanism. Milutinovic et al. [18] proposed a consensus primitive, called proof of luck, that leverages Intel SGX’s random number generation function to choose a consensus leader, which enables low-latency transaction validation and incurs negligible energy. Another application of TEE is Blockchain wallet based on TrustZone [19], [20] that protects sensitive information stored in wallet, such as private key and wallet addresses, from being accesses by unauthorized entities.

In ASSURED framework, we may consider using the combination of TEE and Blockchain techniques for different purpose. ASSURED leverages the attestation service provided by TEE to measure the trustworthiness of each edge device but also uses the Blockchain network to share attestation reports and threat intelligence information with other entities inside the supply chain ecosystem. Blockchain offers a promising solution to share and manage data in a decentralized way. Several data sharing schemes based on Blockchain and TEE have been proposed. The idea behind is to make use of TEE's data sealing service to protect the confidentiality of sensitive data and TEE's attestation service to assure the integrity of smart contract execution (Figure 19).

Ayoade et al. [21] proposed a decentralized data sharing system for IoT devices based on Blockchain and TEE. IoT service providers store the raw data to be shared with untrusted users in secure storage platform using TEE, and also store the hash of the data in the Blockchain for data access management and access history audit. Data access rules are specified by service providers and are realized in form of smart contracts. TEE is responsible for rechecking the access permission before sending raw data to users. Zhang et al. [22] proposed a different design called PrivacyGuard to tackle IoT data sharing problem. PrivacyGuard is also based on the combination of Blockchain and TEE. But they utilize TEE to build a trust entity called iDataAgent who is responsible for cryptographic key management, IoT data encryption, and remote attestation of smart contract execution environment and function to be executed on IoT data. Hu et al. [23] leveraged TEE to ensure the integrity of sensor data that is broadcasted among connected vehicles. They isolated all codes related to sensor data collection and transmission into enclaves and thus prevent compromised vehicles from sharing falsified data. [24] proposed a TEE-based approach called TITAN to address the trust concern of Blockchain-based threat intelligence sharing architecture. TITAN contains a TI quality assessment framework based on TEE measuring the trust of TI and shares the rate of TI with other peers through Blockchain.

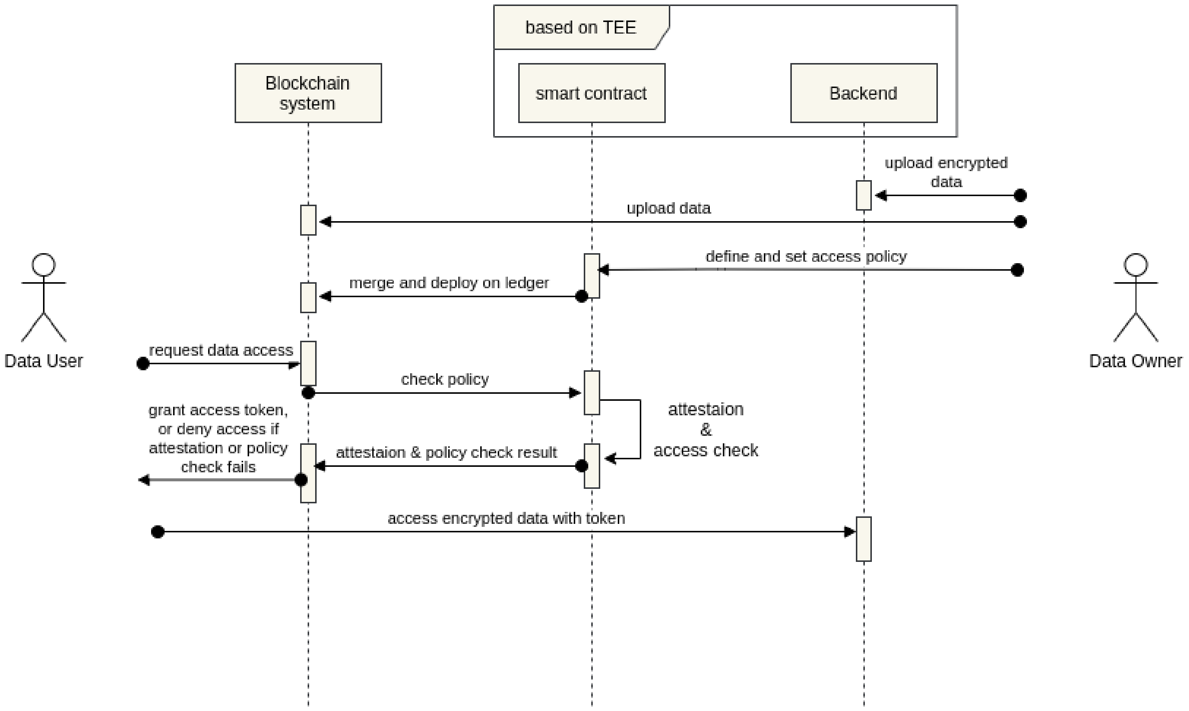

based on TEE
Blockchain system
smart contract
Backend
upload encrypted data
upload data
define and set access policy
merge and deploy on ledger
Data User
Data Owner
request data access
check policy
grant access token, or deny access if attestation or policy check fails
attestaion & policy check result
attestaion & access check
access encrypted data with token

FIGURE 19: *TEE INTERACTIONS WITH DLTS AND SMART CONTRACTS*

4.2.2 Blockchain-based Data Sharing

There exist several research works that have been proposed to use Blockchain techniques to achieve distributed data sharing. These works are applied to many real-world contexts, e.g., medical data access, vehicle networks, and IoT. Most of them have shared similar technical roadmap and physiology that can be summarized as follows. In the very beginning, a Blockchain-based data sharing system requires an original data source layer where data is collected from. To store the data, the system may use a backend data storage that works together with a distributed Blockchain platform. In the backend, original hard copies of data are stored, while the “soft copies” of the data are recorded on the Blockchain, e.g., Inter Planetary File System (IPFS) [25]. By the soft copies, we mean the “special type” of data that may not be but strongly related to the original data. For example, it could be a hash value of the original data stored on the Blockchain, or it could be a trust/reputation value, an access token, e.g., a secret key for decryption or a pointer, for the original data in the backend. These hard and soft copies could be stored in an encrypted format. If so, the data sharing later should require the share of secret key(s). After this set up, the system may consider how to implement the data sharing consent mechanism. To this end, it may employ an access control layer,

Funded by the Horizon 2020 Framework Programme of the European Union

provided by Blockchain itself, to manage data access. This layer usually offers access authentication. For example, in Hyperledger Fabric, the membership service provider component is used to control who is able to “enter” which channel, in which different channels have their respective ledgers. In this case, only valid authenticated users can reach the ledger data belonging to that private channel. Beyond that, the system may adopt smart contract technique to perform automatic policy, attribute check/confirm for data access. A data provider/owner is allowed to design a concrete data sharing policy for its data, e.g., whom could be granted access rights, which types of data could be shared, when and how the sharing could be, etc. Sometimes, system admins will act as a single point to pre-set the smart contract for all users to reach a general data sharing policy match the platform’s policy requirements. From this, data access permission can be maintained and monitored via the smart contract. The contract of data access/sharing policy is stored and merged on ledger (in the form of chaincode). If a data request is sent to the Blockchain, the smart contract will be triggered, and it automatically verify if the requestor can access the data via the pre-defined policy. Once the requester passes the check, it will receive a token or key (e.g., directly from the contract or admins) which can be used to reach the corresponding data. In some systems, revocation of data access is considered so that the access only is valid for some period, e.g., within the data subscription period. Further, the Blockchain-based system can utilize incentive mechanisms to motivate “good” behaviours during data sharing. The above process can be briefly illustrated in Figure 20.

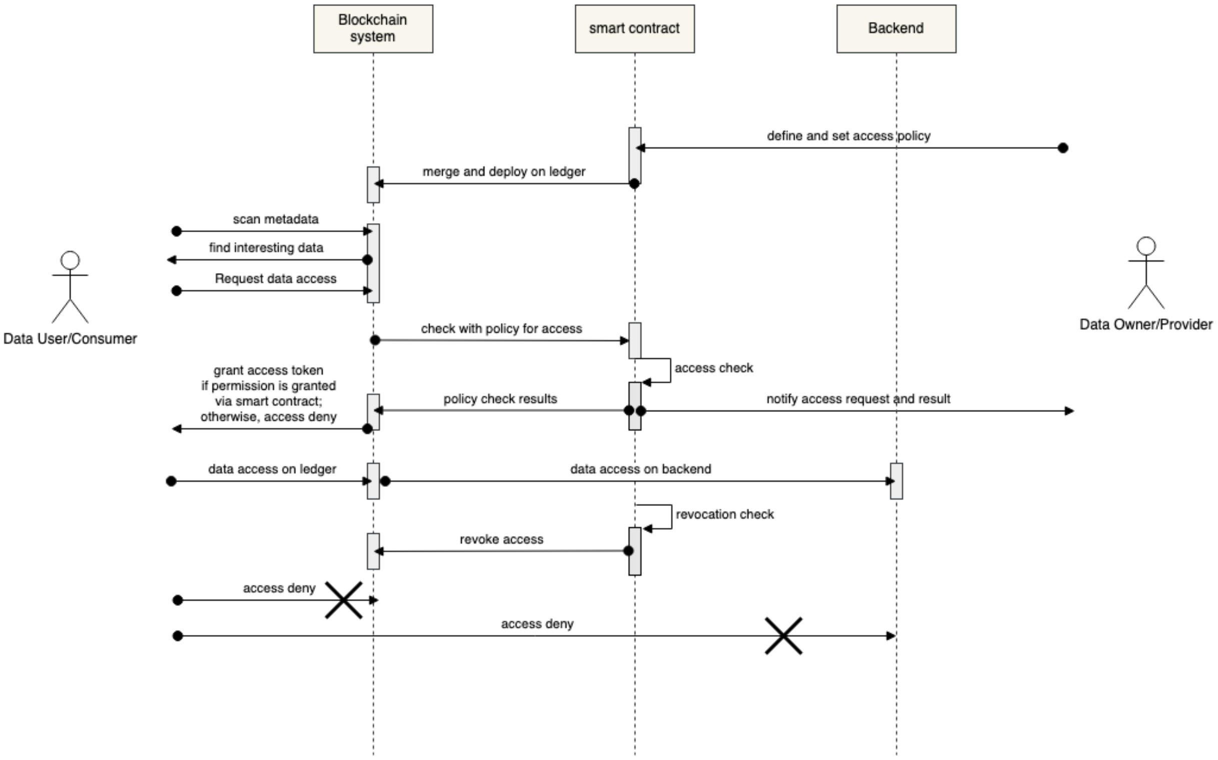


FIGURE 20: *BLOCKCHAIN-BASED DATA SHARING OVERVIEW*

MedRec [26] is developed to manage patients’ medical records via the Blockchain technology. The medical stakeholders (e.g., public health authorities) are designed to be the Blockchain miners to collect records and put them into Blockchain blocks. These miners will be rewarded from their honest behaviours and data aggregation and anonymization operations. Xia et al. [27] proposed a Blockchain-based data-sharing model for cloud service providers. They use smart contracts with access control policy to trace data owner’s behaviours and data sharing and enable one to revoke data access in case of policy violation. Liu et al. [28] introduced a Blockchain-based model for sharing medical records to preserve the privacy of patients. Attribute-based access control mechanism and the content extraction signature schemes are

Funded by the Horizon 2020 Framework Programme of the European Union

used for privacy preservation in data-sharing. Furthermore, smart contracts are defined to set access permissions and to ensure data access security.

Zyskind et al. [29] invented a decentralized personal data management system to enable users to manage data via using the Blockchain as an automated access-control manager. They proposed a scheme named Enigma based on multi-party computation [30], which theoretically resolves the issues on access control. Molina-Jimenez et al. [31] proposed a hybrid Blockchain-based data sharing model. They put contractual operation on Blockchain and assign centralized operation operations to a trusted third party as a manual admin help performing actions on the data access permission. Shrestha et al. [32] leveraged Blockchain and smart contract to execute an incentive mechanism to users for data sharing. The design enables users to track data sharing parties, time and means of sharing, based on a verifiable tactic, in a permissioned Blockchain network, MultiChain.

IPFS introduced a scheme that is proposed to incentivize network peers with Filecoin for using hard drive space as ability of mining rather than the computational power. In IPFS, miners can store files in a distributed fashion, and they should prove to verifier that they do create different copies of the files within the network, using the consensus model called proof of the replication. The Siacoin and Storjcoin make use of similar distributed data storage mechanisms by shredding the user-uploaded file, encrypting each segment and spreading the file ciphertext to the participating nodes across the currencies network. A Blockchain-based marketplace platform for vehicle data was introduced in [33], providing a data-owner-based attribute-based encryption to protect data stored on the cloud or IPFS system. After data user sends a data request and pay for data subscription, data owner can share the secret key for decryption. Ding et al. [34] designed a new attribute-based access control mechanism with Blockchain for IoT. A Blockchain-based data sharing mechanism, while using fine-grained access control and Artificial Intelligence was introduced in [35]. Two Blockchain ledgers are proposed for the data-sharing mechanism - Data chain and Behavior chain. The proposed system is based on hyper ledger Fabric. Javed et al. [36] used IPFS to store vehicular data for decentralized review data sharing.

Xia et al. [37] invented a Blockchain-based solution to organize and manage users to gain access to a pool of shared sensitive data. Niavis et al. [38] introduced a decentralized data sharing infrastructure using Fabric, Indy and IPFS. Fabric and IPFS are used to interactively storage pointer and original data, while Indy is used to manage devices' identity. The framework, DEON, is designed various API interfaces to merge the different ledgers and cloud storage components. Chi et al. [39] introduced a secure and efficient data-sharing scheme based on Blockchain and community detection that considers the relevance and sharing scope of the data to be shared. Its data-sharing is based on Hyperledger Fabric, enabling one to upload a large amount of data obtained from sensors and pre-pares them for sharing through the Blockchain network. Clients are divided into multiple communities according to the correlation and similarity of the collected data, and the data are only shared based on the decisions generated from the community detection algorithm (via identity), meaning that a specific domain users can only access that domain's data.

A reward-based Blockchain solution [40] was designed for distributed P2P networks to motivate honest and correct data sharing from one to others. If one successfully delivers the data to requesters, it will get rewards.

4.2.2.1 Combination of Cryptographic Tools and Blockchain-based Data Sharing

Symmetric searchable encryption technique has been used in the Blockchain-based data system in [41]. A piece of data is encrypted and stored in IPFS and a data user can be granted a keyword trapdoor by data owner to perform related data search in smart contract. Based on similar searchable idea, Zhang et al. [42] deployed searchable public key encryption to e-health system so that a health record is encrypted with keywords, and if data user is given a correct keyword token from the patient, the encrypted data can be located and retrieved. Proxy re-encryption is another technique deployed in Blockchain applications. In [43], data owner can design the so-called re-encryption keys to enable others to share decryption rights. In [44], a system administrator plays as a proxy to do re-encryption for data owner during data sharing; and in [45], a smart card of data owner can be used generated re-encryption key and a smart contract is used to add the key on the Blockchain. A distributed proxy re-encryption for protecting data nodes in the Ethereum Blockchain was introduced in [46]. ABE is also used in Blockchain data protection. Pournaghiet al. [47] used ABE to control data access. ABE also combines with other techniques like signatures [48], [49], and SE [50] while being used in Blockchain data protection. Wang et al. [51] proposed data-sharing model by combining the IPFS, Ethereum Blockchain and ABE technologies. The main purpose of the model is to provide privacy and fine-grained access control of data. Paillier encryption is used to encrypt blocks of ledger [52]. Nebula [53] merges Exonum Blockchain framework [54] with homomorphic encryption to protect the query of genomic data. Considering using cryptographic tools as extra security and privacy enhancement for the Blockchain data, ASSURED will attempt to use ABE, SE and potential PRE to securely wrap up Blockchain data and cloud-based backend data to ensure data confidentiality and fine-grained Blockchain-based data access control.

Speaking of cryptographic tools, we should mention an important tool supported by Blockchain platform, that is Blockchain wallet. A Blockchain wallet is a digital “pocket” that enables one to store and manage its cryptocurrencies. And the core of the management relies on a pair of public and private key for this user. Usually, the public key is used as an “address” of this wallet that is used to receive transactions, e.g., a payment via 10 Ether. As for the private key, it is a digital key allows the user to authorize and then access their cryptocurrencies. When there is a transaction, the wallet API will create a digital signature by processing the transaction using the private key. And the signature can be seen as a form of a signing for the ownership of this transaction or asset. It makes others, users within the same Blockchain network, believe that this transaction comes from the particular user but also the integrity of the transaction, e.g., amount, can be guaranteed. The Blockchain wallet is designed to store the private key for the user. But if the protection fails, the key will be compromised by network attackers; and further the “lost” key will definitely lead to serious financial loss for the user. To protect the key, many existing software and API make good use of different techniques, e.g., mnemonic seed, passwords, offline storage. And some turn their focuses on using trusted hardware. From this perspective, one can inject its key into a trusted hardware, e.g., TPM, so that the hardware is able to organize and store its private key, and meanwhile, some cryptographic functions, e.g., digital signature, the TPM may come to help the user to perform (also within the TPM). In the ASSURED framework, we may consider using the similar philosophy, in order to enable system devices to leverage trusted hardware to hold those keys for authentication, authorization and operations on ASSURED DLT ledgers.

4.2.2.2 Further Trust Considerations on Blockchain

We previously introduced useful trusted hardware, cryptographic tools and necessary components which can be used to support trust in the Blockchain context. For example, the TPM and TEE can guarantee a trusted software and hardware platform for running Blockchain transactions. And the related Blockchain-based data sharing mechanisms, managing the data access via consent strategies, use ledger storage to record data sharing policy and event for traceability, transparency and auditing features, and meanwhile the smart contract embedded data sharing consent/policy enhance the automation on data access and sharing. The Blockchain provides a natural setting for data sharing consent design, storage and further check the consent before retrieving data. We here highlight further details for the trust related to Blockchain which could be deployed in the ASSURED project.

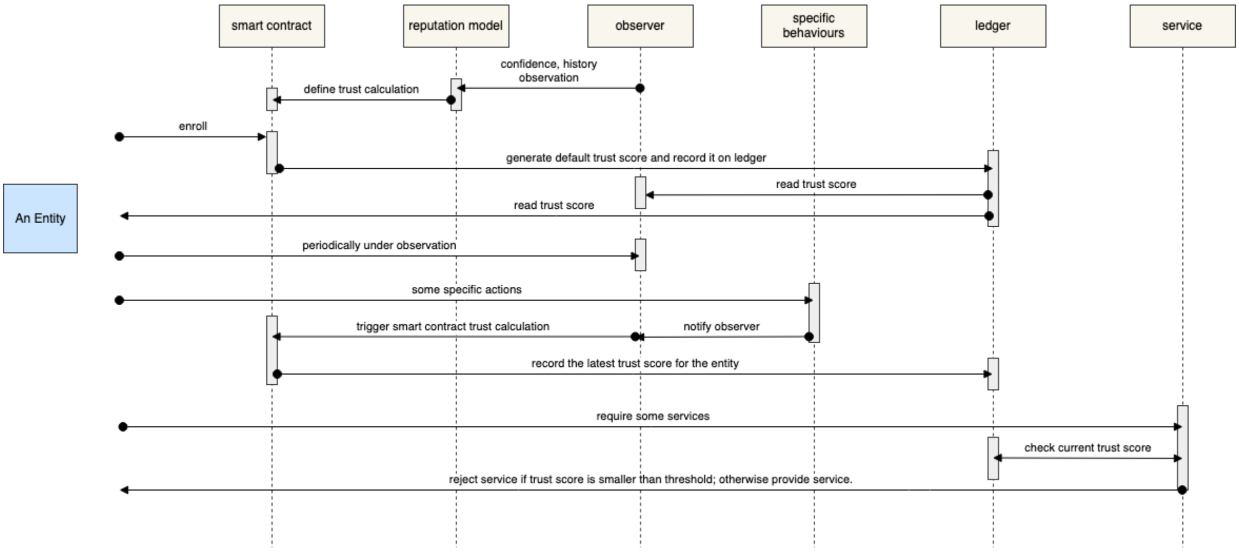
Achieving trust is very necessary in a Blockchain application. One may need to ensure that users' behaviours follow instructions, and they act as being expected. For instance, Synaptic Health Alliance (https://www.synaptichealthalliance.com/) uses Quorum to deliver trust consent on healthcare data usage. It generates a list of keys (identifiers for users) to let valid users to have permission to view heath data stored in transaction. The system also uses permissioned ledger to manage the authentication for users from different range and with various access rights – i.e., providing authentication. The offline storage is implemented via IPFS to generate storage pointers, and the pointers are stored on the online ledger. If the transaction can be read, a user will be able to access to the IPFS for accessing data. This is a classic Blockchain-based data sharing consent as we mentioned above. And of course, one may use smart contract to capture access control, as in [55], [56], [57], [58].

Trust among untrusted peers via consensus. In a Blockchain platform, there must be some operational peers that act as validators, endorsers, and other roles, so that transactions can be validated. No matter in permissionless or permissioned Blockchain, these operational nodes do not need to fully share trust to each other, but they should be able to reach a common decision for a given transaction. To make a trusted and reliable conclusion among these nodes, one may need to use consensus algorithms. That is what we can call – the consensus algorithms for trust. A consensus algorithm enables the untrusted network nodes to make agreement on a given fact/statement related to a current state of Blockchain ledger. And this type of consensus must achieve distributed.

Many consensus algorithms have been proposed in the literature, focusing on different features, e.g., depending on computational power, storage ability, etc. In the context of public Blockchains, all nodes are supposed to be untrusted, and complex and expensive computational-cost consensus algorithms should be used in such an open network, e.g., proof of work [59] – based on computational power, and proof of stake [60] – based on owned stake. On the contrary, a permissioned Blockchain supports identity authentication (e.g., membership mechanism) for nodes so that the nodes should have higher trust than those in public Blockchains. In this way, some lightweight protocols, e.g., Byzantine-Fault Tolerant (for untrusted misbehaviours), could be sufficient to capture the security and trust on transactions. Further, consensus algorithms can combine with trusted hardware to enhance trust. A typical example is the Hyperledger Sawtooth that uses Intel SGX to perform proof of waiting time to select a block winner – this algorithm is called proof-of-elapsed-time [61]. This mechanism can run a node in trusted and secure environment without using expensive resources. Trust can be also considered merging with the endorsement. In Hyperledger Fabric, trust assumptions are reflected on the transaction endorsement policy and smart contract execution. A group of peers are required to confirm transactions based on the trust level in smart contract. Some interesting works that make use of trust evaluation and reward for peers, in such a way that only high trust value peers can be selected to become a block winner, and if the block is successfully validated, the peer will be rewarded and punished otherwise. This type of trust-based consensus protocols can be seen in [62], [63], [64], [65], [66], in which the trusted evaluation considers several aspects, e.g., service rating, integrity checking. Note that we here only present a brief introduction on the trust supported by consensus algorithms and more details (for example, how to use TPM or SGX to enhance trust on consensus, like the proof-of-elapsed-time) related to the algorithms and the design will be presented in the D 4.1.

Relate trust to reputation. There exist distributed systems that consider integrating trust and reputation management into Blockchain. Andersen et al. [57] introduced a delegation of trust concept to enable a resource owner to have a corresponding user to consume the resource. An obligation chain is designed to store signed obligations between service providers and consumers in [67]. This work uses a reputation mechanism to allow providers to accept or decline the obligations, in which reputation scores are stored on ledger and calculated based on obligations. Putra et al. [68] designed a trust and reputation system (TRS) to monitor the behaviours of the users and nodes. Both trust and reputation scores are transparently calculated by smart contracts in a public Blockchain. Kouicem et al. [69] introduced a trustworthiness recommendation service. It manages the trustworthiness value of the service providers on Blockchain, and when a user is requesting a recommendation, the system outputs the evaluation. A blocktrust framework [70] is designed to allow service requesters to locate trustworthy service providers, force service providers and requesters to be trustworthy, and discourage dishonest participation. Dedeoglu et al. [71] proposed a layered trust architecture for Blockchain-based IoT applications. They use long-term reputation collected from the so-called system observers, and the data trust value is calculated based on the evidence from observation, reputation of the source and confidence of the observation. Rouhani et al. [72] introduced a concept of data trust framework in which a dataset has its own "trustworthiness" value calculated by data owner and other users' confidence, reputation and endorsement status. The calculation and summary of the value are performed by the smart contract automatically. Tang et al. [73] proposed a Blockchain-based trust framework for collaborative IoT scenarios. The trust management is to define and maintain the "trust" relationship among the IoT entities based on their credits. They use Trust-Oriented Credits (TOC) to provide dynamic trust management used in entity access control policies. TOC will indicate the reliability of trustworthiness, and it is presented as credit policies in a so-called credit management contract. A TrustChain for IoT supply chain is proposed, using smart contract to calculate trust and reputation scores based on the off-chain stored data (their hash value), e.g., trade event, regulatory endorsement. A distributed trust management scheme [74] is developed to calculate the credibility of exchanged messages based on the reputation value of observers in Blockchain. Blockchain-based anonymous reputation system in [75] uses direct historical logs and other indirect factors to set up trusted communication.

Most of the aforementioned systems share the following technical roadmap: (1) observe entities' behaviors and define the trust via a function of reputation/credit related to predefined factors, e.g., service response, service rating, peer integrity check; (2) make use of smart contract to automatically calculate the trust score; (3) reflect the trust report and values on Blockchain ledger; (4) incentive mechanism to enhance trusted behaviors. The corresponding sequence diagram is depicted in Figure 21, in which some cases could not require the existence of the observer and meanwhile, an accomplishment of a specific behavior, e.g., correctly mining a block, could trigger a reward/penalty on the trust score directly via smart contract-based calculation. This interesting trust enhance mechanism may be further deployed in the ASSURED DLT context.


smart contract
reputation model
observer
specific behaviours
ledger
service
confidence, history observation
define trust calculation
enroll
generate default trust score and record it on ledger
read trust score
An Entity
read trust score
periodically under observation
some specific actions
trigger smart contract trust calculation
notify observer
record the latest trust score for the entity
require some services
check current trust score
reject service if trust score is smaller than threshold; otherwise provide service.

Funded by the Horizon 2020 Framework Programme of the European Union

***FIGURE 21:** TRUSTED DATA STORAGE IN BLOCKCHAIN*

4.2.2.3 Error Correction Techniques

Recall that in the first three layers of the general data flow architecture we need to use encryption technology to protect data confidentiality. And the transmission of encrypted data usually requires the use of error correction codes to recover information efficiently and reliably during the decryption process. Traditionally, error correction and encryption are addressed independently, but there have been several successful attempts to combine encoding and cryptography operations into one.

McEliece proposed the first public-key cryptosystem based on algebraic coding theory. The idea behind this scheme is because the decoding problem of an arbitrary linear code is an NP-hard problem. A private-key cryptosystem based on burst error correcting codes was proposed in [76], where the ciphertext is obtained by encoding the message XORed with a predetermined burst sequence and permuting the result using the permutation matrix (kept secret between the sender and the receiver). Hwang and Rao proposed two secret error-correcting code (SECC) schemes [77] that use a nonlinear channel code (Preparata codes) as its starting point. However, the true error correcting capacity of these scheme is significantly reduced. In [78], Kak proposed an approach to joint encryption and error-correction, which is based on decimal expansions of D-sequences. It was shown that the encoding operation is equivalent to that of exponentiation in finite field, which is like encryption in public key ciphers.

The functions of error correction and security were truly integrated in works like the Godoy and Pereira Scheme [79]. The idea behind this scheme is to derive new generator matrices from existing generator matrices by row permutations. The security of the system relies on the change and secrecy of the generator matrices, which makes the scheme vulnerable to brute force attacks on the generator matrix. Cryptocoding [80] is another proposed technique for joint error correction and encryption. This technique is based on quasigroup (Latin square) string transformation. Every message is padded with a bunch of zeros before the encoding/encryption operation. Although this technique achieves both security and error correction, the decoding procedure is extremely complicated and cannot be used in a resource constrained environment. In [81], the authors defined a new class of codes, called High Diffusion (HD) codes, that possess optimal diffusion along with being maximum distance separable (MDS) to achieve joint error correction and encryption. A High Diffusion (HD) cipher was then proposed in [82] based on the HD codes, which was shown secure as the AES cipher and outperforms the traditional mechanism both in terms of security and error correction. Recently, Moldovyan et al. [83] proposed two novel modes for using block ciphers that provides to perform error correction in the case of sending data via a noisy channel. In one of the modes, data encryption is combined with error correction coding. In the second proposed mode, pseudo-probabilistic block encryption is combined with error correction coding in the single crypto scheme. This existing error-correct-and-encryption merging technologies give us a feasibility to provide data confidentiality but also data recovery in our data flow model.

4.2.3 Smart Contract Conversion for Trust and Data Sharing

One key technology which is part of DLTs is that of smart contracts. The term smart contract describes in a sense a series of actions which go into effect automatically, based on some specific conditions, and in essence resembles contracts, which are automatically executed based on a programmable logic.

As such, we regard smart contracts are computer programs, or snippets of code, which are stored on a Blockchain and are executed when predetermined conditions are met. Their main goal is to automate transactions between different parties. There exist many research works and studies which very well describe how smart contacts are defined and operated such as

[84], which also identifies a set of challenges that is witnessed in many systems and has to do with both technological or legislative issues. Such issues include legal issues that have to do with how data is handled relevant to GDPR rules, how these "Contracts" comply with the legislation of countries [85], what is their interplay and dependency and optimal strategies when dealing with off-chain resources [86], how can they guarantee immutability, how can they scale in an efficient and cost-effective manner, or what types of consensus mechanisms one should use.

One of the main operations where such contracts are useful and widely used, are for enabling data sharing between different parties. As identified in [87], [88]. In the recent years many approaches have emerged relevant to data-sharing, especially between enterprises, though several barriers are there, such as privacy, as indicated in [89]. To mitigate such issues, various designs have been proposed, which try to combine the power of smart contracts with those of user control, as the platform proposed in [32] where distributed ledgers and smart contracts are used to offer user-controlled privacy and define data-sharing policies which are encoded in smart contracts.

What is also quite important is the power of Blockchain technology, as provided by smart contract to enforce the execution of such agreements, as for example illustrated by [90] where smart contracts are generated based on parameters extracted from on legal data sharing agreements and put these terms in immediate effect once a contract is executed.

As seen above, with data sharing there is the issue of access control and access policy enforcement. As DLTs are based on the notion of distribution and information sharing, mechanisms need to be put in place to safeguard access to data and resources. Various schemes and research works have been performed in the area of policy design and enforcement for access in such networks as well for accessing different contents of the network, based on the rights of each actor. Luckily, the emerging the Blockchain and smart contract technology has attracted significant scientific interest in research areas like authentication and access control processes. This is demonstrated in [91] where smart contracts are used to resolve and execute in a completely distributed manner Access Control Policies or in [92].

The area of IoT has been one of the more researched ones, as it offers a quite interesting case; multiple devices exchanging large amounts of data, and there is always the need to control how access to such data is provided. In [93] a system is described which is based on a trust and authentication framework for access control such environment. Similarly, [94] discuss how with Blockchain technology and smart contracts can be used for access control judgment in IoT infrastructures, [95] discusses the implementation of a Capability-Based Access Control scheme in a local Ethereum network and [96] provides insights on how a multi-authority attribute-based access control scheme can work, where smart contracts are used define the interactions between data owner, data user, and multiple attribute authorities. At the same time, also zero-knowledge proof concepts in combination with Blockchain technologies are emerging, such as the concept proposed in [97] where a model using zero-knowledge proof and smart contracts is presented to improve the access control security in IoT networks.

As seen from the above placed high-level review, it is evident that smart contracts can, and are already playing a significant role when it comes to actions that have to do with trust and data sharing management. In ASSURED the aim is to leverage the power offered by smart contracts and DLTs to strengthen those operations by building an infrastructure that can automatically facilitate the needs of different entities that work together, in order to create rigorous and undisputable relationships of trust between entities, back-up by smart contracts that convert the logic that surrounds such actions into automatically executed actions that enforce and guarantee the execution of policies, triggered by different events in the network that have certain, well described and mutually agreed outcomes (such as for example what to share in terms of data upon a failed attestation events, how to enforce the execution of an attestation policy between two different entities, etc).

As shown in Figure 22, smart contracts can be used to enforce logic, such as for example an access policy, as they are able to be used as a direct representation of a set of logic conventions in the IT world, where the reasoning is done by machines. As such, a block of logic (or an access policy, etc) can be translated into one or many smart contracts (depending on how complex the logic is and what is the optimum way of managing this in certain blocks of executable code), where the actors are actually the users who are involved in the smart contract, the sphere of application becomes the deployment point of the smart contract (e.g. where is it executed and which (sub-)network it concerns), and the actual logic, inputs and outputs are the application code, the inputs that are passed into the contract and the final output/result of the code are yielded from the contract. The converted smart contract is written into a type of script programming language, e.g., JavaScript, and further is merged on Blockchain ledger turning into a so-called chaincode. When it needs to be executed, it can be called locally (via download the copy) or remotely, intaking input and yielding output. The output (sometimes along with input) later will be recorded on ledger. For example, given an attestation and verification algorithms (e.g., source code of a remote attestation from TPM2.0), we may reflect the logic, the algorithm code into the “core” of smart contract, and further, we enable this script-based contract to be sent out as a transaction with the Blockchain network, so that a network peer is able to merge the contract into a chaincode which is stored on the ledger. Later, if a device is required to perform an attestation on its status, it may download and execute the smart contract code locally, and then send the results to the Blockchain peer who will record the results on ledger. In this case, any auditor or verifier can check the attestation results from ledger (assuming that auditor/verifier is within the same Blockchain network with appropriate authentication). Similarly, the ASSURED project will consider converting data sharing policy into smart contract. In this type of contract, we may transform

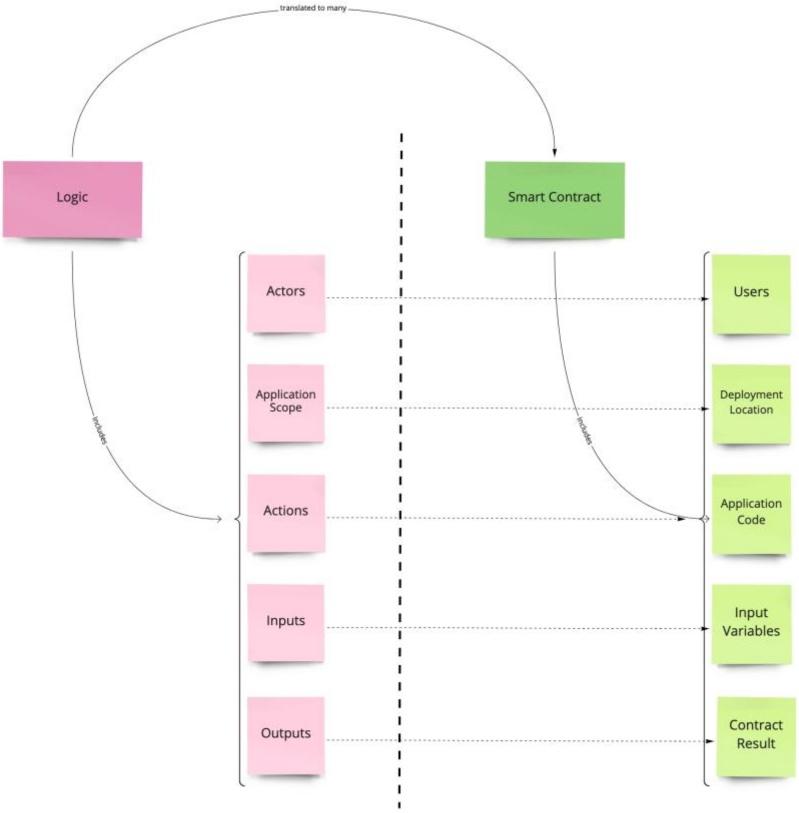


FIGURE 22: *SMART CONTRACT BUSINESS LOGIC*

the policy into data sharing condition, and then the data sharing behaviours or operation is converted into a programmable algorithm that control the release of access token for ledger and the ledger's block data. Once the sharing condition is satisfied, e.g., a user's attributes are allowed to access some ledger data, the smart contract will automatically return a token to the user. The data sharing smart contract may be also designed in a time restrict way, i.e., a data sharing token could be revoked or use-limit within a pre-set time frame, via the use of time counter. Based on the same philosophy, we may use smart contract to trigger event recording based on the input condition, and further, we may convert a reputation and trust score equation/model into a calculation algorithm within the contract, so that a user's behaviours can be also reflected into the reputation score, and the score can be further recorded on ASSURED Blockchain ledger.

Nevertheless, as our study has shown, there exists various obstacles in the way, that should be tackled for delivering the anticipated services that should cover the needs of the ASSURED framework, such as how to offer smart contracts which are essentially based on nested events (or the output of other contracts), how to build a mechanism to allow for dynamic, on-the-fly deployable smart contracts, how and where to deploy smart contracts in networks where we need to attest the various nodes making sure at the same time that the smart contracts operate on well-performing entities, and how to be support "updating" smart contracts since they are immutable. The above questions will be tackled in the later WP4 activities and will be documented in the respective deliverables [105], [109], [110].

5 ASSURED BLOCKCHAIN-BASED DATA SHARING MODELS FIT-IN TO USE CASES

In the previous chapters we have introduced the definitions and the related trust consent and data sharing models as well as the security, privacy, and trust requirements for all envisioned use cases. Furthermore, a details analysis of Blockchain technologies was also put forth with mechanisms (leveraging either SW- or HW-based trust anchors) capable of providing data sharing capabilities coupled with such strict requirements.

In what follows, we are going to present a general technical vision and roadmap for an ASSURED fit-in framework (in the context of the use cases) so that all defined data sharing behaviors and can achieve the required secure and privacy-preserving data flows. The framework will capture the **ASSURED policy-compliant Blockchain-based conceptual architecture** (as defined in Section 2.1 and further elaborated in D4.1 [105]) and the corresponding **technical components related to trust consent and data management**, which will be useful for the future development of the *ASSURED Security Context Broker, Blockchain-based Data Control Services and TPM-based Wallet*. Please note that the goal is to highlight the general vision of the framework as the more detailed designs, cryptographic models and developments will be considered in the context of WP4.

5.1 ASSURED BLOCKCHAIN-BASED CONCEPTUAL ARCHITECTURE

In Figure 23**:** assured general technical vision on data sharing models, we put forth a more technical description of the underpinnings and interactions of the general ASSURED Blockchain framework described in Section 2.1. As aforementioned, the goal is to be able to identify the **type of technical components and mechanisms to be further integrated in ASSURED** (based on the SOTA performed in Chapter 4) and describe their **mode of operation in order to achieve the security, privacy and trust requirements, identified in Chapter 2, for all of the defined data sharing behaviours (Chapter 3)**. We note that this whole architecture is to present a general technical baseline for our later design in particular in WP4. We here want to make a high-level connection between the components that will be leveraged for enabling the described data sharing behaviours. We do not explore more technical details related to the development and implementation of the protocols and techniques at this stage; for instance, as is aforementioned, ASSURED will explore the use of ABE mechanisms for offering multi-level access control (accessing different levels of data granularity), however, detailed models on a new HW-enabled ABE scheme (supported by the underlying Root-of-Trust) will be given in the Deliverable D4.1 [105].

The framework adheres to the following workflow of actions: From down-to-top operational layers (Figure 3), we have the sensor, gateway, operational center and cloud-based storage layers. We assume that the **devices and gateways**, in the respective layers, **could be securely pre-equipped with a trusted hardware, e.g., TPM, and the cryptographic tools used in the framework have been safely deployed alongside their keys** (e.g., the Attribute-related Keys – detailed models on these operations will be defined in [109]). In the upper layers, namely, operational and cloud-based backend (including ASSURED DLT ledgers), we assume users, e.g., CIO, system admins, security context broker, are securely assigned cryptographic keys as well.

There are several main technical components used in the general framework. The trusted hardware component, named TH component, is the core technical building block. It takes charge of the following functionalities: (1) secure cryptographic key storage and management (for cryptographic tools); (2) secure authentication for two communication parties, e.g., via

TPM credential mechanism, and authorization for ledger access; (3) (anonymous) digital signature, e.g., ABS or DAA; (4) other possible embedded secure operations (within the trusted hardware), e.g., TPM based hash function for integrity check, TPM based encryption RSA, AES256 for data confidentiality; (5) support trust and secure smart contract based attestation – attesting statuses of devices. We state that the TH component can be seen as the initial form of trusted hardware-based wallet.

The ABE component is mainly used to protect confidentiality via the use of attributes and policies. In this case, an encryption of transferred data under a policy can be only decrypted and recovered by those entities with matching attributes. This component is used with the error correction component. The error correction technique is used to help recover data from encrypted data with communication channel's noise. In this way, even when the encrypted IoT data is mixed with noise during transferring, the underlying real data can be still correctly recovered. As for the Searchable Encryption (SE), this is used to provide secure encrypted data search, enabling entities to search over (ABE) encrypted pointers and (ABE) encrypted data stored on private ledger and cloud-based backend, respectively. Besides, we also use an integrity component which can be designed with the use of digital signature and hash function (e.g., TPM based HMAC).

To alleviate the user and data privacy concerns, we also consider using an **anonymous component that mainly considers protecting the identities of sensor layer's devices**. For instance, a group of edge devices can encrypt and sign their data for its IoT gateway, but the gateway may not be able to link the data to its data source. To this end, we plan to make use of advanced cryptographic tools such as Direct Anonymous Attestation (DAA) or ABS or other anonymous signatures, e.g., group signature, to hide the signers' IDs. *We note that in the ASSURED context, we plan to make trusted hardware support this anonymous component*.

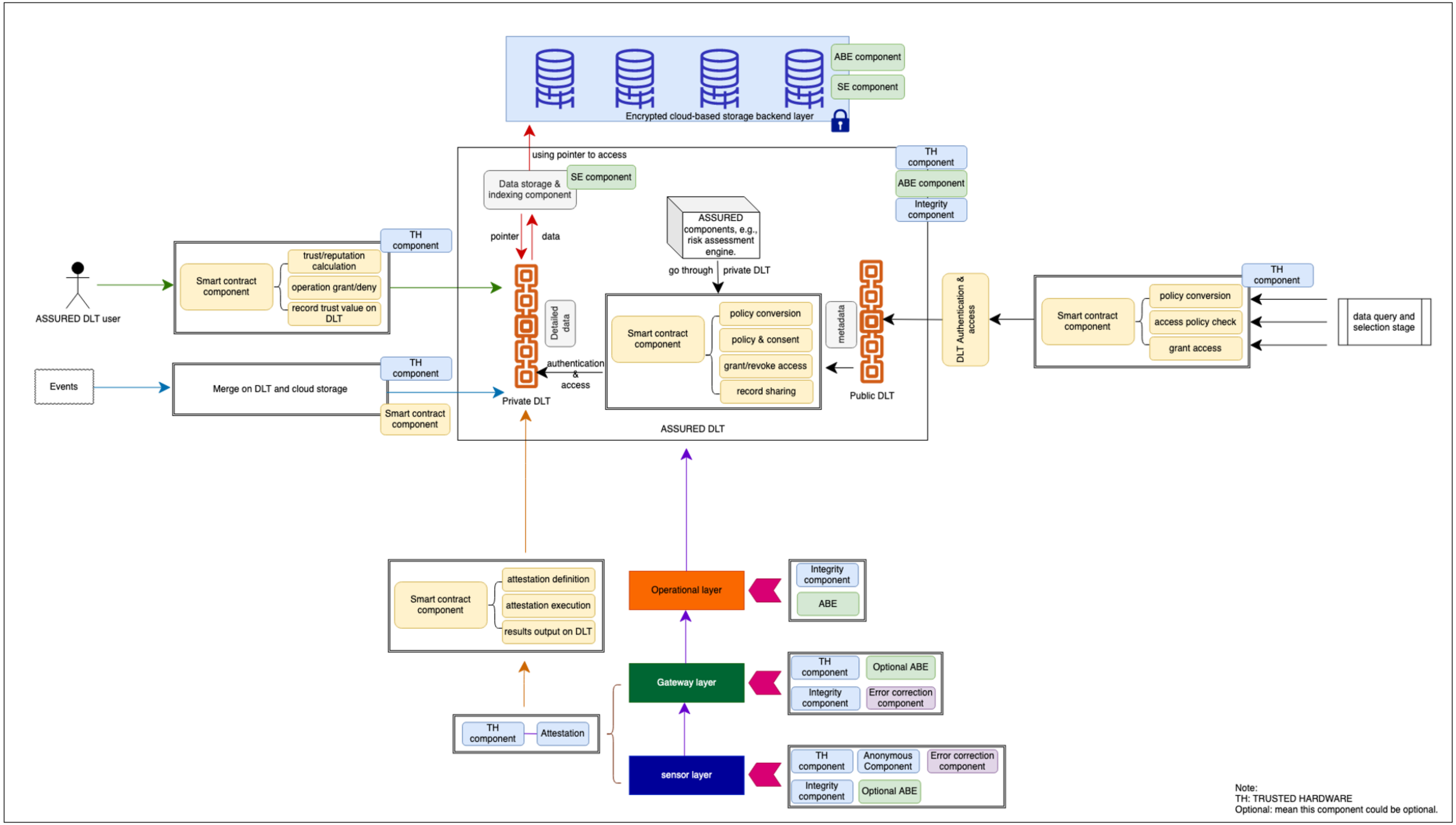
Furthermore, we require the existence of a smart contract component to be able to interact with the TH component, ASSURED DLT ledgers, the SE component, Security Context Broker, internal and external entities, and other ASSURED components for supporting the secure enforcement of attestation policies and the subsequent (operational and threat intelligence) data sharing. As mentioned previously, we will convert the **data sharing policies and attestation enhancement into smart contract programmable codes which are embedded on ASSURED DLT ledgers as chaincode**. This component will mainly provide: (1) attestation: policy/attestation conversion, attestation execution and verification, record attestation results; (2) data accessing/sharing: policy consent/check, grant/reject access, record data sharing request and response; and (3) trust score define and calculation, operation grant/deny based on trust score, and record trust score on ledgers. This component can be further used for events recording. We note that we've introduced and reviewed secure components within a blockchain platform in the Deliverable D1.2, e.g., consensus algorithm, mining, Merkle tree, digital signature etc. Here we do not review them again, but we state that those blockchain "self-embedded" components (e.g., in Ethereum and Hyperledger Fabric) will be enhanced by using smart contract, ABE, and TH components; and the concrete details will be given in the Deliverable D4.1.

In the overall framework there are also some other components, including Data Storage & Indexing component, and the so-called ASSURED components [99]. For the former, it is used to support **searchable encryption – indexing the data using keywords/tags, so that to link the keywords/tags to encrypted stored data**. For the latter, they could be the ASSURED runtime Risk Assessment component, Collective Threat Intelligence and Forecasting Engine, which can be referred to the Deliverable D1.2 [99]. We state that these components will need to go through the smart contract component if they would like to share/access data from ASSURED DLT.

5.1.1 General Fit-In Framework Functionalities Description

The sensor layers should provide **data confidentiality, data integrity, error correction, device privacy and authentication**. Here we put the "optional" tag because the devices and gateways may leverage secure communication – authenticated and secure channel, e.g., TLS/SSL, – to deliver data. If that channel is used, ABE may not be needed. And each device is able to perform (local/remote) attestation (for their current inner data flow and status) via the use of smart contract, and the attestation results are stored on the ASSURED private ledger (as well as cloud-based backend), in which the hash of the results (achieving integrity) could be merged on ledger and final copy could be on the backend. We say that this will form the initial technical vision on the ASSURED attestation component – trusted hardware based (e.g., TPM) smart contract attestation, offering secure, distributed (batch-wide via swarm approach), trust and automatic attestation.

Funded by the Horizon 2020 Framework Programme of the European Union



Note:
TH: TRUSTED HARDWARE
Optional: mean this component could be optional.

FIGURE 23: *ASSURED GENERAL TECHNICAL VISION ON DATA SHARING MODELS*

Funded by the Horizon 2020
Framework Programme of the European Union

Similarly, we have the above technical components in the gateway layer except the anonymous component. This is because there is no need to protect the anonymity that which gateways send the data to the backend. **We note that only the attestation data could need to be linked to specific devices, but of course, if anonymity is required, then the trusted hardware, e.g., TPM, could be used to support DAA to anonymize the attestation**. And besides, like in the sensor layer, we may provide TH component for the gateways for attesting their inner status.

As for the operational center layer, we only require ABE and integrity components. Most of the data in this layer is taken to analysis and processing. And at the end, the data and the analysed results will be reflected and stored on ASSURED DLT private ledger and cloud-based backend, for example, the hash of the data analysis results, and the corresponding ABE encrypted pointer is recorded on the ledger, while the full results copy is encrypted under ABE and stored on backend. In this case, the pointer and full data confidentiality and the data integrity are captured via ABE and hash operations. We note that in this layer, we do not consider using the error correction component. This is because we assume there will exist a stable and reliable network quality (connection) between operational center and backend (as well as ASSURED DLT). In practical deployment, use case partners may regard this component as optional depending on specific network conditions.

The ASSURED DLT acts as a middleware for the cloud-based backend. It consists of two levels: public ledger (metadata) and private ledger (for detailed data). And it requires the TH, ABE and integrity components to maintain trust, data confidential and integrity. The TH component is a trust anchor in this layer, and it will provide necessary secure operations, e.g., authentication, trust enhancement on smart contract and ledger, and trust-based attestation. The crucial component in this middleware is smart contract component, and it has four main supports:

1. If an external stakeholder requires to search interested data on ASSURED DLT public ledger, its request will go through a smart contract (deployed on public ledger). The contract will check the predefined access policy (defined by security context broker), if the stakeholder matches the policy, then it is granted the access and deny the access, otherwise. After searching the metadata on the public ledger (and identifying a further data access), the stakeholder proceeds a request to get into the private ledger. This request is handled by another smart contract (deployed on private ledger). The contract, similarly, will check policy, grant/revoke/deny access, and further record this data sharing, e.g., (“stakeholder”, “access”, “medical file A”, “Monday 1pm”). Besides, in the contract, we may inject the anonymization sanity checker so that the access data’s real identity (i.e. its source, where it is from) can be “fuzzy”. As mentioned in the Deliverable D1.2 [98], we will put metadata and detailed data on ASSURED public and private ledgers, respectively. And in the private ledger, we use data storage and indexing component to relate the search index and storage pointer between the ledger and cloud-based backend. On the ledger, a (ABE encrypted) pointer is stored pointing back to the backend, enabling the stakeholder to locate the (ABE) encrypted file; and the SE component will help the stakeholder quickly and precisely to locate the pointer (corresponding to privacy-preserving queries) without revealing other contents stored on the ledger. We further note that the SE component can also help cloud-based backend admin/manager quickly and securely search encrypted data via some specific keywords as well. The above smart contract-based data sharing may be also applicable to internal entities and ASSURED components data access. Similarly, the contract also needs to check policy list before granting/denying access. And further, the data sharing (request and response) will be recorded via the contract to the private ledger; and meanwhile, the sharing could be time-limit if we design a time counter within the contract – so that once the count down, say 1 month, is finished, the sharing can be revoked. We note that the logic, algorithm, data

Funded by the Horizon 2020
Framework Programme of the European Union

sharing policy, and attestation definition and pre-set, e.g., who can access what type of data via which token or pointer, are captured and done by the security context broker.

2. A device performs attestation via the use of smart contract. In this case, the attestation behaviors will be reflected into smart contract and executed by running the smart contract via the help of TH component. The results will be merged on ASSURED DLT private ledger and a summary of the results (metadata) may be put on public ledger. In the attestation case, attestation and verification algorithms, supported by the TH component, may be converted into the smart contract. A device can easily download the contract locally to run the attestation and further forward the results to a verifier. The verifier thus is able to run the verification algorithm from the smart contract intaking the results to see if the attestation is valid. The attestation results and output of verifier will be recorded on the private ledger for later auditing.

3. A trust behavior evaluation ranking via the use of smart contract. The smart contract will be used to evaluate a DLT user's reputation/trust value so as to enhance the trust among users. And the evaluation and update model (e.g., by assigning a series of actions with specific points and weights, the sum of the points could be the trust score for these actions) for the reputation will be pre-designed by the security context broker and then further, the model will be implemented by the smart contract, so that the calculation and results will be auto-executed and stored on ASSURED private ledger when needed. The contract will be also designed to have an algorithm to control system users' operation, e.g., denying or granting action, based on reputation/trust scores.

4. Event log handler via the use of smart contract. Similar to the policy conversion, the contract may first define "which events should be recorded and which not", it then further checks the condition, and records those satisfied on ASSURED DLT private ledger and cloud-based backend, in which the hash of the event could be stored on the ledger and the original encrypted version is on backend.

Within the ASSURED DLT, we assume the platform we identify to develop on (note this practice will be done in WP4) will be able to provide secure blockchain supportive components, e.g., hash function – like SHA256, consensus algorithm – like Byzantine Fault Tolerance, Merkle tree, digital signature and authentication – like membership service providers/access control list. With these tools, we will be able to guarantee the integrity of block data via the use of hash function and Merkle tree, authentication via the membership management, guarantee of action and ownership via the use of digital signature, mining validation and agreement on validation through consensus algorithm. Beyond these, trust, data confidential and automatic data access and sharing can be done via the above ABE, smart contract and TH components. And we will embed SE component to enhance secure search on ASSURED DLT.

The top layer is for the cloud-based backend. We will provide ABE component and SE component here. That means encryption – capturing data confidentiality – is required in this layer, but also a secure data search on cloud backend can be provided as well.

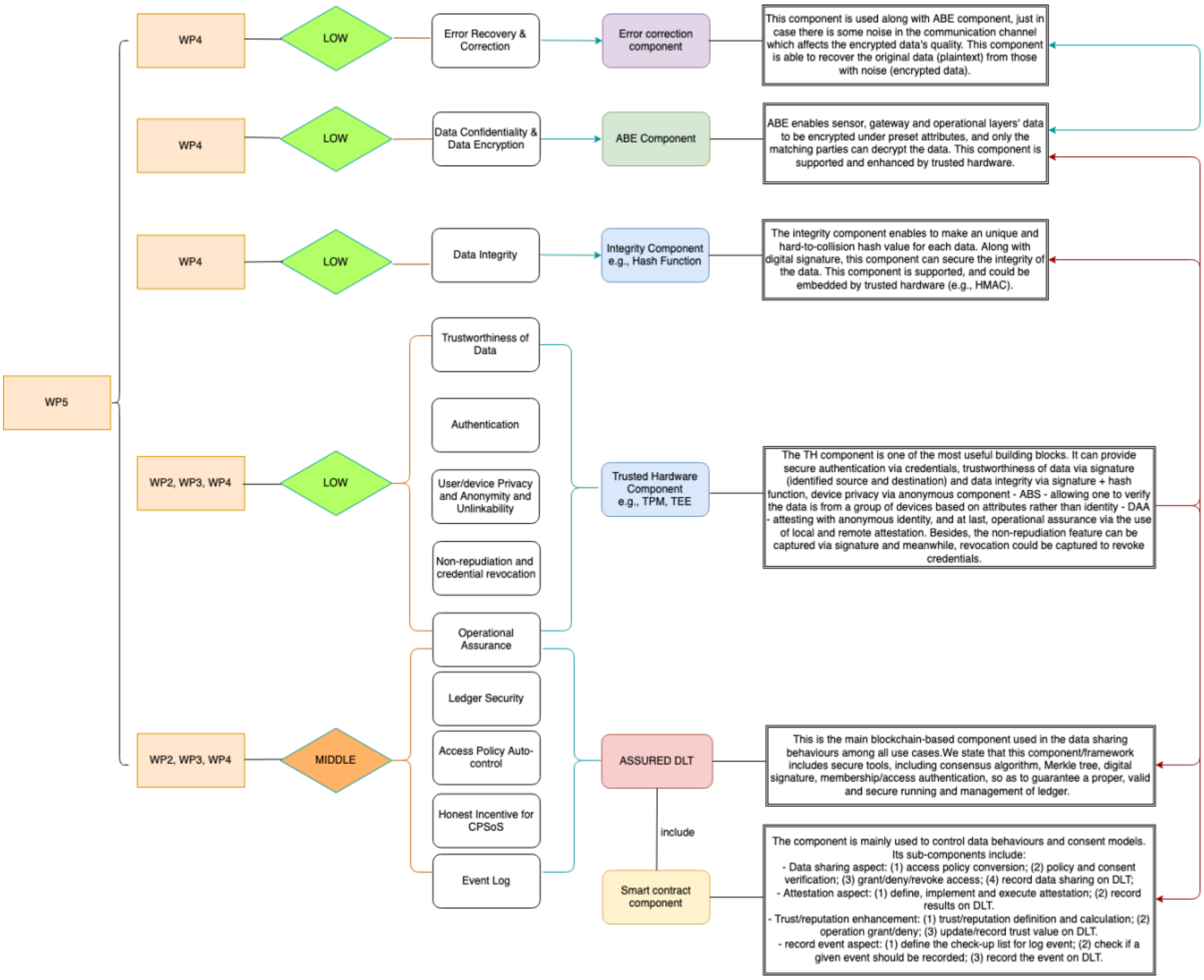
From the above general technical vision framework, in what follows, we also give the corresponding adjust to each use case to show that all use cases can fit in the framework with slight revision.

In the context of Smart Cities, the technical vision perfectly matches the main data flow. Specifically, the data is transferred from sensor layer all the way to cloud-based backend, as depicted in Figure 4. And the data transfer and communication provide security and privacy guarantee via the use of ABE, integrity, error correction and TH components. Sensors and CCTV devices can authenticate themselves to the IoT gateway and encrypt the data via ABE with error correction component before sending, while the gateway can do similar to CIO,

Funded by the Horizon 2020
Framework Programme of the European Union

admins and internal operator. As for the attestation part, the admins can require the CCTV and sensors to perform smart contract-based attestation to check runtime status. And the data and information sharing behaviours of DAEM internal, say between CIO and internal operator, external, say with LEAs, can be captured in the framework. Specifically, data sharing is protected by access policy automatic control via the use of smart contract backed up by TH component. In this case, DAEM's system admins play the role of security context broker who is able to design and inject data sharing policy and attestation (along with event recording condition, trust credit calculation) into the smart contracts.

In the context of Smart manufacturing use case, the data flow is clearly captured from down to top in the vision framework. Data confidentiality, data integrity and error correction are captured in the lower levels, from those below the IoT Gateway (Figure 5). The internal and external data sharing is also done via the use of smart contracts to check access policy list (designed by system admins). But BIBA enables system admins to generate access token for external parties for data accessing. Based on this, the vision framework may be revised to use the access tokens instead of encrypted pointers, and further these tokens will be stored on the private ledger accordingly. With the tokens, parties can access the data. For the attestation, the IoT Gateway can request the robotic devices to perform smart contract-based attestation, while itself could be the verifier for the proof.


WP5
WP4
LOW
Error Recovery & Correction
Error correction component
WP4
LOW
Data Confidentiality & Data Encryption
ABE Component
WP4
LOW
Data Integrity
Integrity Component e.g., Hash Function
WP2, WP3, WP4
LOW
Trustworthiness of Data
Authentication
User/device Privacy and Anonymity and Unlinkability
Non-repudiation and credential revocation
Operational Assurance
Trusted Hardware Component e.g., TPM, TEE
WP2, WP3, WP4
MIDDLE
Ledger Security
Access Policy Auto-control
Honest Incentive for CPSoS
Event Log
ASSURED DLT
include
Smart contract component

This component is used along with ABE component, just in case there is some noise in the communication channel which affects the encrypted data's quality. This component is able to recover the original data (plaintext) from those with noise (encrypted data).

ABE enables sensor, gateway and operational layers' data to be encrypted under preset attributes, and only the matching parties can decrypt the data. This component is supported and enhanced by trusted hardware.

The integrity component enables to make an unique and hard-to-collision hash value for each data. Along with digital signature, this component can secure the integrity of the data. This component is supported, and could be embedded by trusted hardware (e.g., HMAC).

The TH component is one of the most useful building blocks. It can provide secure authentication via credentials, trustworthiness of data via signature (identified source and destination) and data integrity via signature + hash function, device privacy via anonymous component - ABS - allowing one to verify the data is from a group of devices based on attributes rather than identity - DAA - attesting with anonymous identity, and at last, operational assurance via the use of local and remote attestation. Besides, the non-repudiation feature can be captured via signature and meanwhile, revocation could be captured to revoke credentials.

This is the main blockchain-based component used in the data sharing behaviours among all use cases.We state that this component/framework includes secure tools, including consensus algorithm, Merkle tree, digital signature, membership/access authentication, so as to guarantee a proper, valid and secure running and management of ledger.

The component is mainly used to control data behaviours and consent models. Its sub-components include:
- Data sharing aspect: (1) access policy conversion; (2) policy and consent verification; (3) grant/deny/revoke access; (4) record data sharing on DLT;
- Attestation aspect: (1) define, implement and execute attestation; (2) record results on DLT.
- Trust/reputation enhancement: (1) trust/reputation definition and calculation; (2) operation grant/deny; (3) update/record trust value on DLT.
- record event aspect: (1) define the check-up list for log event; (2) check if a given event should be recorded; (3) record the event on DLT.

FIGURE 24: *ASSURED BLOCKCHAIN COMPONENTS AND REQUIREMENTS RELATIONSHIP*

In the context of Smart Aerospace use case, the flow of actions is somewhat similar to that of Smart Cities in terms of the overall data flow architecture. More concretely, all airplanes inner devices are regarded as the core components comprising the sensor layer, SSR is the network

Funded by the Horizon 2020 Framework Programme of the European Union

gateway, GS may be in the operational layer and its GSS is the cloud-based backend. Much like the above use cases, data confidentiality and the corresponding authentication and security requirements (for data flow) in the lower layers are captured by the secure components, namely TH, ABE, integrity and error correction components. As for the data sharing: internal sharing can be done via the private ledger and while the external data sharing is guaranteed via "public ledger → private ledger → smart contract control → access" mode, in which contract's policy is defined previously by GS's admins. All onboard devices are connected to the SSR via a "heads unit" that can also act as a (intermediate) gateway. It's the "heads unit" that can be equipped with a trusted anchor for attesting the correct state of all other devices.

In the context of Smart Satellites, as illustrated in Figure 7, the CubeSat is set as the combination of sensor, gateway and operational layers, while the ground station is the mix of operational center and cloud-based backend layers. The data communication between the ground station and CubeSat can be protected by the secure and trust components (e.g., TH component ensures authentication via key exchange). The data sharing, especially between ground station and external stakeholders, is provided via the ASSURED DLT and smart contract component for automatic policy checking (defined by GS's admins – the role of security context broker), and the access grant/deny. To prove the status, CubeSat may input evidence to the smart contract and run locally, and then output the result to the GS (the attestation requester) for verification. The result and check will be stored on ASSURED DLT for auditing purpose.

For all use cases, system events can be fully monitored and tacked – via the use of smart contract and could be triggered by system admins, DLT users' reputation and trust score can be calculated and recorded to encourage honest DLT behaviours – where the reputation model may be pre-designed and triggered by admins, attestation can be also enhanced via the combination of TH and smart contract components. We summarize all the aforementioned main components, their descriptions, and relationships with the requirements in Figure 24.

5.2 RISK ASSESSMENT OF THE ASSURED BLOCKCHAIN TECHNICAL COMPONENTS

Below we are going to generally describe the risk assessment on the aforementioned technical components for trust consent and data sharing models in the ASSURED use cases. We note that the purpose of this part is not going to give a detailed assessment but to give a general awareness for the future work packages to understand which components should be carefully designed and developed in the use cases.

__TABLE 8:__ RISK ASSESSMENT OF THE ASSURED BLOCKCHAIN TECHNICAL COMPONENTS

Items	Component	Risk Level	Descriptions
1.	**Error correction**	LOW	This component is listed at a low risk level. This is because there has been well-studied works and implementation in the literature for this technique. It is not difficult to reproduce it in algorithm and software levels. But one thing should be carefully handled that is the technical interface with trusted hardware supported ABE – how to securely and function-harmlessly combine the ABE seamlessly and correctly with error correction techniques would be an outstanding point.
2.	**ABE component**	LOW	The component should be supported by trusted hardware, e.g., TPM. It should be designed to issue

			decryption keys based on entity's attributes. The current TPM can be used to support key management and decryption key release. The tricky part here is to make use of the TPM to safely control the key storage and release and associate the key related to attributes, so that a valid authenticated user (matching attributes) can use the key for decryption.
3.	**Integrity component**	LOW	This can be satisfied via the use of hash function and digital signature, which can be supported by the trusted hardware, e.g., TPM. We may need to consider if we should provide a fully trust environment for the execution of both cryptographic operations – hash and signature (full protected), or just – either hash or signature is done by trusted hardware.
4.	**Trusted hardware component**	LOW	This component is required to present to support various operations, e.g., hash, signature, attestation, authentication. It has sufficient theoretical and practical/implementation-level knowledge and cores for its development. We may pay attention to how to enable this component to securely support other cryptographic operations, e.g., decryption, and searchable encryption. And we also need to consider its interfaces with smart contract and Blockchain platform via the implementation of TC-based trusted wallet, which could be mainly related to management of credential and keys.
5.	**Smart contract and DLT component**	MIDDLE	Smart contract is one of the most important building blocks in the fit-in framework. It is required to support several functions. Although smart contract currently has been well developed in many Blockchain platforms, e.g., Ethereum, Hyperledger Fabric, a few points should be carefully handled: (1) policy conversion and enforcement – reflecting policy into logic of smart contract; (2) securely execution of the attestation via smart contract – reflecting TPM attestation execution on smart contract; (3) security and privacy consideration on the input/output of smart contract; (4) secure related functional extensions from existing smart contract applications, e.g., we may use smart contract to record events and trust reputation calculation, how to extend these new functions into existing/identified blockchain platform is outstanding. Besides, based on current DLT techniques, we should consider how to build up TPM interfaces within the DLT framework. Luckily, Hyperledger Fabric can provide the interface, but we still need to consider the interface extension to support secure cryptographic operations over ledger with TPM. This may be a challenging point.

5.3 FUTURE RESEARCH

We finally summarize some open problems from our components and framework for the future work packages.

__TABLE 9:__ TECHNICAL COMPONENTS AND OPEN RESEARCH QUESTIONS

Item	Component	Descriptions	Work Package
1.	Error correction component	Identify a concrete algorithm or re-design an existing but most-related tool for error correction. This algorithm must be compatible with the trusted hardware-based	WP4

Funded by the Horizon 2020 Framework Programme of the European Union

		ABE, e.g., TPM-based ABE, which means that it should support somewhat ABE format's error correction.	
2.	ABE component	Identify ABE tools that work with TPM. If cannot, design a new component that enables using TPM to manage decryption key of the ABE.	WP4
3.	Integrity component	Identify hash and signature algorithms from the current TPM algorithms.	WP4
4.	Trusted Hardware component	Design and implement a wallet capable of integrating hardware-based keys (from the attached TPM) and securely downloading and executing the attestation policies via smart contracts – policy enforcement, attestation, and the way it supports cryptographic operations – especially supporting the key management; consider deploying trusted hardware into which entities on which layer.	WP2, WP3 and WP4
5.	Concrete DLT framework	Identify a potential blockchain platform, e.g., Hyperledger Fabric, for ASSURED DLT develop foundation. Make sure the platform can provide flexible and extensive interfaces for smart contract, trusted hardware and secure cryptographic tools. And its self-embedded components, e.g., consensus algorithm, hash, digital signature etc, should be identified and developed their extension for trusted hardware.	WP4
6.	Smart contract component	Choose a foundation Blockchain platform for smart contract (captured in the above item); define types of smart contract, e.g., supporting data sharing, attestation, event logs; identify the way of converting policy enforcement and access policy into smart contract; develop attestation smart-contract-based execution; develop trust/reputation mechanism.	WP2, WP3, WP4
7.	Integration & test	A system wide components integration and test to see if the above main (secure) technical components can be interactive with each other, provide defined functions, and be compatible with the whole system.	WP5

Funded by the Horizon 2020 Framework Programme of the European Union

6 SUMMARY AND CONCLUSION

This section will conclude the deliverable and summarize its findings. The main mission of this deliverable was to collect the **data flow, data and threat information sharing profile from all use cases**, define the corresponding **security, privacy and trust requirements** and further review the most prominent **Blockchain and trust-based technologies**. This then allowed us to also give a **general technical vision within ASSURED framework for secure data sharing and trust consent** by providing secure, trusted and auditable data sharing environments for a new generation of policy-compliant Blockchain structures enhanced with advanced on- and off-chain data and knowledge management services through the specification of novel TPM-based security and privacy-preserving protocols. The vision is to enable data confidentiality, integrity and multi-level access control (**security by design**), data ownership safeguarding (**privacy by design**), data provenance and sovereignty checking and trusted consent management, while respecting prevailing GDPR legislation [98]. In ASSURED, by “security- and privacy-by-design” we understand all methods, techniques and tools aiming at enforcing security and privacy properties at both network and system (software) level from their conception while guaranteeing validity in parallel [100].

Overall, taking the technical requirements and the conceptual architecture of ASSURED from deliverable D1.2 [99], this deliverable takes a step further on the security, privacy and trust vision on trust consent and data sharing models.

ASSURED is based on a **hybrid Blockchain-powered infrastructure** (integrating the use of both **private and public ledgers**) that will facilitate sealing of (attestation) smart contracts on the side of the edge devices, as well as their secure sharing between both internal and external stakeholders. The **secure data storage, publish and sharing** will follow the latest trends in DLTs to rely on trust anchors of different types, each being important in terms of some dimension of policy, technology, data, security, assurance and more. ASSURED relies on a combination of advanced set of cryptographic trust anchors towards binding entities and attributes to data subjects and data principals, as well as to actors within the system that operate the ASSURED trust framework. Different from current Blockchain functions, ASSURED will consider **secure on chain data searching so as to provide a privacy-preserving way for all stakeholders to search preferred information without leaking sensitive information of the data (on private ledger) before being granted read rights**.

In summary, deliverable D1.4 **specifies and models (threat intelligence and operational) data sharing behaviors** among all ASSURED parties and stakeholders based on defined **trusted consent activities** between them, to be enforced through the ASSURED Blockchain infrastructure and trust anchors. It covers both: (i) **operational data** originating from the deployed CPSoS that have strict *trustworthiness* requirements, and (ii) **threat intelligence data/evidence** based on the attestation policies to be enforced. This set of data sharing behaviors are also mapped to the envisioned ASSURED use cases (*Smart Manufacturing, Smart Cities, Smart Aerospace and Smart Satellites*) that will serve as the basis for the extraction of the complete set of **security, privacy and trust requirements** that need to be achieved by the provided functionalities throughout the entire data lifecycle; from the **trust on agreement on registration and data sharing/collection to storage and use of data**. These requirements will help to guide the path towards the concrete design of the ASSURED Blockchain infrastructure and data sharing related components, as defined in D1.2 [99]; i.e., ASSURED DLT Engine, TPM-based Wallet, Smart Contract Composer and Data Storage Engine. Essentially, this deliverable forms the basis for the further modelling and implementation of the modelled data sharing behaviors via the use of smart contracts (to be defined in WP4) for capturing the [105]: (i) **enforcement of attestation policies through their conversion into smart contract logic** (performed by the ASSURED Security Context Broker [99]) and their further deployment/sharing, to the CPSoS, through the distributed ledgers, (ii)

Funded by the Horizon 2020 Framework Programme of the European Union

monitoring of the corresponding attestation output and its auditable recording to attestation history chains on the ledger [109], and (iii) sharing of both operational and threat intelligence data with other data collectors [110].

7 LIST OF ABBREVIATIONS

ABE	Attribute-based Encryption
ABS	Attribute-based Signature
ACS	Analytics Cloud Server
AES	Advanced Encryption Standard
API	Application Programming Interface
AK	Attestation Key
CA	Certificate Authority
CFA	Control Flow Attestation
CPS	Cyber Physical System
CPSOS	Cyber-Physical Systems-of-Systems
CIO	Chief Information Officer
CCTV	Closed-Circuit Television
CTI	Collective Threat Intelligence & Forecasting Engine
DAA	Direct Anonymous Attestation
DLT	Distributed Ledger Technology
DRTM	Dynamic Root of Trust for Measurement
ECC	Elliptic-Curve Cryptography
ECDSA	Elliptic Curve Digital Signature Algorithm
GDPR	General Data Protection Regulation
GS	Ground Station
GSS	Ground Station Server
GDPR	General Data Protection Regulation
HE	Homomorphic Encryption
HD	High Diffusion
HRI	Human Robot Interaction
IOT	Internet of Things

Funded by the Horizon 2020
Framework Programme of the European Union

IPC	Industrial PC
IPFS	Inter Planetary File System
JSON	JavaScript Object Notation
LEA	Law Enforcement Agency
MDS	Maximum Distance Separable
OS	Operating System
SE	Searchable Encryption
SHA	Secure Hash Algorithm
SOS	Systems-of-Systems
SSL	Secure Socket Layer
SSR	Secure Server Router
SECC	Secret Error Correcting Code
TCB	Trusted Computing Base
TCBW	TC-based Blockchain Wallet
TEE	Trusted Execution Environment
TLS	Transport Layer Security
TOC	Trust-Oriented Credits
TPM	Trusted Platform Module
TRS	Trust and Reputation System
UWB	Ultra-wide Band
WP	Work Package

Funded by the Horizon 2020
Framework Programme of the European Union

REFERENCES

[1] Thilakanathan, D., Chen, S., Nepal, S., Calvo, R. A., Liu, D., & Zic, J. (2014, June). Secure multiparty data sharing in the cloud using hardware-based TPM devices. In *2014 IEEE 7th International Conference on Cloud Computing* (pp. 224-231). IEEE.

[2] Chen, L., & Warinschi, B. (2010, December). Security of the TCG privacy-CA solution. In *2010 IEEE/IFIP International Conference on Embedded and Ubiquitous Computing* (pp. 609-616). IEEE.

[3] Maji, H. K., Prabhakaran, M., & Rosulek, M. (2011, February). Attribute-based signatures. In *Cryptographers' track at the RSA conference* (pp. 376-392). Springer, Berlin, Heidelberg.

[4] Lewko, A., & Waters, B. (2011, May). Decentralizing attribute-based encryption. In *Annual international conference on the theory and applications of cryptographic techniques* (pp. 568-588). Springer, Berlin, Heidelberg.

[5] Chen, L., & Urian, R. (2015, August). DAA-A: Direct anonymous attestation with attributes. In *International Conference on Trust and Trustworthy Computing* (pp. 228-245). Springer, Cham.

[6] M. Sabt, M. Achemlal and A. Bouabdallah, "Trusted Execution Environment: What It is, and What It is Not," 2015 IEEE Trustcom/BigDataSE/ISPA, 2015, pp. 57-64, doi: 10.1109/Trustcom.2015.357.

[7] Intel. Software Guard Extensions Programming Reference. https://software.intel. com/sites/default/files/329298-001.pdf

[8] Drucker, N., & Gueron, S. (2017, October). Combining Homomorphic Encryption with Trusted Execution Environment: A Demonstration with Paillier Encryption and SGX. In *Proceedings of the 2017 International Workshop on Managing Insider Security Threats* (pp. 85-88).

[9] Wang, W., Jiang, Y., Shen, Q., Huang, W., Chen, H., Wang, S., & Lin, D. (2019). Toward scalable fully homomorphic encryption through light trusted computing assistance. *arXiv preprint arXiv:1905.07766*.

[10] L. Coppolino, S. D'Antonio, V. Formicola, G. Mazzeo and L. Romano, "VISE: Combining Intel SGX and Homomorphic Encryption for Cloud Industrial Control Systems," in IEEE Transactions on Computers, vol. 70, no. 5, pp. 711-724, 1 May 2021, doi: 10.1109/TC.2020.2995638.

[11] FUHRY, Benny, et al. HardIDX: Practical and secure index with SGX. In: IFIP Annual Conference on Data and Applications Security and Privacy. Springer, Cham, 2017. p. 386-408.

[12] AMJAD, Ghous; KAMARA, Seny; MOATAZ, Tarik. Forward and backward private searchable encryption with SGX. In: Proceedings of the 12th European Workshop on Systems Security. 2019. p. 1-6.

[13] Microsoft, "The Coco Framework," 2017, whitepaper, https://github. com/Azure/coco-framework.

[14] Bowman, M., Miele, A., Steiner, M., & Vavala, B. (2018). Private data objects: an overview. arXiv preprint arXiv:1807.05686.

[15] Cheng, Raymond, et al. "Ekiden: A platform for confidentiality-preserving, trustworthy, and performant smart contracts." 2019 IEEE European Symposium on Security and Privacy (EuroS&P). IEEE, 2019.

[16] Wang, Y., Li, J., Zhao, S., & Yu, F. (2020). Hybridchain: A Novel Architecture for

Confidentiality-Preserving and Performant Permissioned Blockchain Using Trusted Execution Environment. *IEEE Access*, *8*, 190652-190662.

[17] M. Brandenburger, C. Cachin, R. Kapitza, and A. Sorniotti, ‘‘Trusted computing meets Blockchain: Rollback attacks and a solution for hyperledger fabric,’’ in Proc. 38th Symp. Reliable Distrib. Syst. (SRDS), Oct. 2019, pp. 324–32409.

[18] Milutinovic, M., He, W., Wu, H., & Kanwal, M. (2016, December). Proof of luck: An efficient Blockchain consensus protocol. In proceedings of the 1st Workshop on System Software for Trusted Execution (pp. 1-6).

[19] Dai, Weiqi, et al. "SBLWT: A secure Blockchain lightweight wallet based on Trust zone." IEEE Access 6 (2018): 40638-40648.

[20] Gentilal, Miraje, Paulo Martins, and Leonel Sousa. "TrustZone-backed bitcoin wallet." Proceedings of the Fourth Workshop on Cryptography and Security in Computing Systems. 2017.

[21] Ayoade, G., Karande, V., Khan, L., & Hamlen, K. (2018, July). Decentralized IoT data management using Blockchain and trusted execution environment. In 2018 IEEE International Conference on Information Reuse and Integration (IRI) (pp. 15-22). IEEE.

[22] Zhang, N., Li, J., Lou, W., & Hou, Y. T. (2018). PrivacyGuard: Enforcing private data usage with Blockchain and attested execution. In Data Privacy Management, Cryptocurrencies and Blockchain Technology (pp. 345-353). Springer, Cham.

[23] Hu, S., Chen, Q. A., Joung, J., Carlak, C., Feng, Y., Mao, Z. M., & Liu, H. X. (2020, March). Cvshield: Guarding sensor data in connected vehicle with trusted execution environment. In *Proceedings of the Second ACM Workshop on Automotive and Aerial Vehicle Security* (pp. 1-4).

[24] Wu, Y., Qiao, Y., Ye, Y., & Lee, B. (2019, October). Towards improved trust in threat intelligence sharing using Blockchain and trusted computing. In *2019 Sixth International Conference on Internet of Things: Systems, Management and Security (IOTSMS)* (pp. 474-481). IEEE.

[25] Benet, J. (2015). IPFS-Content Addressed, Versioned, P2P File System (DRAFT 3). Available online at: https://ipfs.io/ipfs/QmV9tSDx9UiPeWExXEeH6aoDvmihvx6jD5eLb4jbTaKGps

[26] Azaria, A., Ekblaw, A., Vieira, T., and Lippman, A. (2016). “MedRec: using Blockchain for medical data access and permission management,” in 2016 2nd International Conference on Open and Big Data (OBD) (Vienna: IEEE), 25–30.

[27] Xia, E. B. Sifah, K. O. Asamoah, J. Gao, X. Du, and M. Guizani, ‘‘MeD-Share: Trust-less medical data sharing among cloud service providers via Blockchain,’’ IEEE Access, vol. 5, pp. 14757–14767, 2017.

[28] J. Liu, X. Li, L. Ye, H. Zhang, X. Du, and M. Guizani, ‘‘BPDS: A Blockchain based privacy-preserving data sharing for electronic medical records,’’ in Proc. IEEE Global Commun. Conf. (GLOBECOM),Dec. 2018, pp. 1–6

[29] Zyskind, G., Nathan, O., and Pentland, A. S. “Decentralizing privacy: using Blockchain to protect personal data,” in Proceedings - 2015 IEEE Security and Privacy Workshops, SPW 2015 (San Jose, CA), 180–184.

[30] Zyskind, G., Nathan, O., and Pentland, A. Enigma: decentralized computation platform with guaranteed privacy. ArXiv:1506.03471, 1–14.

[31] Molina-Jimenez, C., Solaiman, E., Sfyrakis, I., Ng, I., and Crowcroft, J. (2019). On and Off-Blockchain Enforcement of Smart Contracts. Cham: Springer, 342–354.

[32] Ajay Kumar Shrestha, Julita Vassileva and Ralph Deters (2020). A Blockchain Platform

for User Data Sharing Ensuring User Control and Incentives. Frontiers in Blockchain. 22 October 2020. https://doi.org/10.3389/fbloc.2020.497985.

[33] Jeong BG, Youn TY, Jho NS, Shin SU. Blockchain-Based Data Sharing and Trading Model for the Connected Car. Sensors (Basel). 2020;20(11):3141. Published 2020 Jun 2. doi:10.3390/s20113141.

[34] Ding, S.; Cao, J.; Li, C.; Fan, K.; Li, H. A Novel Attribute-Based Access Control Scheme Using Blockchain for IoT. IEEE Access2019,7, 38431–38441

[35] Zhang, G.; Li, T.; Li, Y.; Hui, P.; Jin, D. Blockchain-based data sharing system for ai-powered network operations. J. Commun. Inf. Netw.2018,3, 1–8

[36] Javed, M.U.; Rehman, M.; Javaid, N.; Aldegheishem, A.; Alrajeh, N.; Tahir, M. Blockchain-Based Secure Data Storage for Distributed Vehicular Networks. Appl. Sci. 2020, 10, 2011. https://doi.org/10.3390/app10062011.

[37] Xia, E. Sifah, A. Smahi, S. Amofa, and X. Zhang, ‘‘BBDS: Blockchain-based data sharing for electronic medical records in cloud environments,’’Information, vol. 8, no. 2, p. 44, Apr. 2017.

[38] Harris Niavis, Nikolaos Papadis, Venu Reddy, Hanumantha Rao, Leandros Tassiulas: A Blockchain-based Decentralized Data Sharing Infrastructure for Off-grid Networking. IEEE ICBC 2020: 1-5.

[39] Chi J., Li Y., Huang J., Liu J., Jin Y., Chen C., Qiu T. A secure and efficient data sharing scheme based on Blockchain in industrial internet of things. J. Netw. Comput. Appl., 167 (2020), Article 102710-102720.

[40] Shrestha, Ajay Kumar, and Julita Vassileva, ‘‘Blockchain-based research data sharing framework for incentivizing the data owners,’’ inProc. Int.Conf. Blockchain. Cham, Switzerland: Springer, 2018, pp. 259–266.

[41] L. Chen, W. Lee, C. Chang, K. R. Choo, and N. Zhang, “Blockchain based searchable encryption for electronic health record sharing,” Future Gener. Comput. Syst., vol. 95, pp. 420–429, 2019.

[42] A. Zhang and X. Lin, “Towards secure and privacy-preserving data sharing in e-health systems via consortium Blockchain,” J. Medical Syst., vol. 42, no. 8, pp. 140:1–140:18, 2018.

[43] T. T. Thwin and S. Vasupongayya, “Blockchain based secret-data sharing model for personal health record system,” in 2018 5th International Conference on Advanced Informatics: Concept Theory and Applications (ICAICTA). IEEE, 2018, pp. 196–201.

[44] X. Liu, Z. Wang, C. Jin, F. Li, and G. Li, “A Blockchain-based medical data sharing and protection scheme,” IEEE Access, vol. 7, pp. 118 943–118 953, 2019.

[45] B. Sharma, R. Halder, and J. Singh, “Blockchain-based interoperable healthcare using zero-knowledge proofs and proxy re-encryption,” in COMSNETS. IEEE, 2020, pp. 1–6.

[46] G. G. Dagher, J. Mohler, M. Milojkovic, and P. B. Marella, “Ancile: Privacy-preserving framework for access control and interoperability of electronic health records using Blockchain technology,” Sustainable cities and society, vol. 39, pp. 283–297, 2018.

[47] S. M. Pournaghi, M. Bayat, and Y. Farjami, “Medsba: a novel and secure scheme to share medical data based on Blockchain technology and attribute-based encryption,” J. Ambient Intell. Humaniz. Comput., vol. 11, no. 11, pp. 4613–4641, 2020.

[48] H. Yang and B. Yang, “A Blockchain-based approach to the secure sharing of healthcare data,” Nisk Journal, pp. 100–111, 2017.

[49] H. Wang and Y. Song, “Secure cloud-based EHR system using attribute-based

cryptosystem and Blockchain," J. Medical Syst., vol. 42, no. 8, pp. 152:1–152:9, 2018.

[50] Y. Xiaodong, L. Ting, L. Rui, and W. Meiding, "Blockchain-based secure and searchable ehr sharing scheme," in 2019 4th International Conference on Mechanical, Control and Computer Engineering (ICMCCE). IEEE, 2019, pp. 822–8223.

[51] S. Wang, Y. Zhang, and Y. Zhang, "A Blockchain-based framework for data sharing with fine-grained access control in decentralized storage systems," IEEE Access, vol. 6, pp. 38 437–38 450, 2018.

[52] S. Yaji, K. Bangera, and B. Neelima, "Privacy preserving in Blockchain based on partial homomorphic encryption system for ai applications," in 2018 IEEE 25th International Conference on High Performance Computing Workshops (HiPCW). IEEE, 2018, pp. 81–85.

[53] D. Grishin, K. Obbad, P. Estep, K. Quinn, S. W. Zaranek, A. W. Zaranek, W. Vandewege, T. Clegg, N. César, M. Cifric et al., "Accelerating genomic data generation and facilitating genomic data access using decentralization, privacy-preserving technologies and equitable compensation," Blockchain in Healthcare Today, 2018.

[54] "Exonum," 2020, https://exonum.com/.

[55] O. Novo, "Blockchain meets IoT: An architecture for scalable access management in IoT," IEEE Internet Things J., vol. 5, no. 2, pp. 1184–1195, Apr. 2018.

[56] Y. Zhang, S. Kasahara, Y. Shen, X. Jiang, and J. Wan, "Smart contract-based access control for the Internet of Things," IEEE Internet Things J., vol. 6, no. 2, pp. 1594–1605, Apr. 2019.

[57] M. P. Andersen, J. Kolb, K. Chen, G. Fierro, D. E. Culler, and R. A. Popa, "WAVE: A decentralized authorization system for IoT via Blockchain smart contracts," Dept. Elect. Eng. Comput. Sci., Univ. California, Berkeley, CA, USA, Rep. UCB/EECS-2017-234, 2017.

[58] S. Pal, T. Rabehaja, A. Hill, M. Hitchens, and V. Varadharajan, "On the integration of Blockchain to the Internet of Things for enabling access right delegation," IEEE Internet Things J., vol. 7, no. 4, pp. 2630–2639, Apr. 2020.

[59] M. Vukolic, "The quest for scalable Blockchain fabric: Proof-of-Work vs. BFT replication," in International Workshop on Open Problems in Network Security. Springer, 2015, pp. 112-125.

[60] G. Wood, "Ethereum: A secure decentralised generalised transaction ledger," Ethereum Project Yellow Paper, vol. 151, 2014.], or Proof-of-Elapsed-Time (PoET) [Intel: Sawtooth Lake (2017). https://intelledger.github.io/.

[61] Mic Bowman, Debajyoti Das, Avradip Mandal, Hart Montgomery: On Elapsed Time Consensus Protocols. IACR Cryptol. ePrint Arch. 2021: 86 (2021).

[62] J. Zou, B. Ye, L. Qu, Y. Wang, M. Orgun, L. Li, A proof-of-trust consensus protocol for enhancing accountability in crowdsourcing services, IEEE Trans. Serv. Comput. (2018). https://doi.org/10.1109/TSC.2018.2823705.

[63] L. Bahri, S. Girdzijauskas, When trust saves energy: a reference framework for proof of trust (PoT) Blockchains, in: Web Conference (WWW'18), 2018, pp. 1165–1169. https://doi.org/10.1145/3184558.3191553.

[64] The Trust Chain Consensus, COTI: a Decentralized, High Performance Cryptocurrency Ecosystem Optimized for Creating Digital Payment Networks and Stable Coins, White Paper, (2018), available from: https://coti.io/files/COTI- technical-whitepaper.pdf.

[65] I-CASH, A Smart Contract Origination and Settlement Platform Leveraging the Proof

of Trust Protocol, White Paper, available from: https://cdn2.hubspot.net/hubfs/4276960/ICASH%20whitepaper.pdf?t=1524150610703.

[66] Besfort Shala, Ulrich Trick, Armin Lehmann, Bogdan V. Ghita, Stavros Shiaeles: Novel trust consensus protocol and Blockchain-based trust evaluation system for M2M application services. Internet Things 7 (2019).

[67] R. Di Pietro, X. Salleras, M. Signorini, and E. Waisbard, ‘‘A Blockchain- based trust system for the Internet of Things,’’ in Proc. 23nd ACM Symp. Access Control Models Technol., Jun. 2018, pp. 77–83, doi: 10.1145/ 3205977.3205993.

[68] Guntur Dharma Putra, Volkan Dedeoglu, Salil S. Kanhere, Raja Jurdak, Aleksandar Ignjatovic: Trust-Based Blockchain Authorization for IoT. IEEE Trans. Netw. Serv. Manag. 18(2): 1646-1658 (2021).

[69] D. E. Kouicem, Y. Imine, A. Bouabdallah, and H. Lakhlef, “Decentralized Blockchain-based trust management protocol for the Internet of Things,” IEEE Trans. Dependable Secure Comput., early access, Jun. 18, 2020, doi: 10.1109/TDSC.2020.3003232.

[70] Vatankhah Barenji, R. A Blockchain technology based trust system for cloud manufacturing. J Intell Manuf (2021). https://doi.org/10.1007/s10845-020-01735-2.

[71] V. Dedeoglu, R. Jurdak, G. D. Putra, A. Dorri, and S. S. Kanhere, “A trust architecture for Blockchain in IoT,” in Proc. 16th EAI Int. Conf. Mobile Ubiquitous Syst. Comput. Netw. Serv., 2019, pp. 190–199.

[72] Sara Rouhani, Ralph Deters: Data Trust Framework Using Blockchain Technology and Adaptive Transaction Validation. IEEE Access 9: 90379-90391 (2021).

[73] Bo Tang, Hongjuan Kang, Jingwen Fan, Qi Li, Ravi S. Sandhu: IoT Passport: A Blockchain-Based Trust Framework for Collaborative Internet-of-Things. SACMAT 2019: 83-92.

[74] A. Kchaou, R. Abassi, and S. Guemara, "Toward a distributed trust management scheme for vanet," Proceedings of the 13th International Conference on Availability, Reliability and Security, 2018, pp.53:1-53:6.

[75] Z. Lu, W. Liu, Q. Wang, G. Qu and Z. Liu, "A Privacy-Preserving Trust Model Based on Blockchain for VANETs," in IEEE Access, vol. 6, pp. 45655-45664, 2018.

[76] Hung-Min Sun: Private-key cryptosystem based on burst-error-correcting codes. Electronics Letters 33(24): 2035-2036 (1997).

[77] Tzonelih Hwang, T.R.N. Rao: Secret Error-Correcting Codes (SECC). CRYPTO 1988: 540-563.

[78] Subhash C. Kak: Joint Encryption and Error-Correction Coding. IEEE Symposium on Security and Privacy 1983: 55-60.

[79] Walter Godoy Jr., Dyson Pereira Jr.: A proposal of a cryptography algorithm with techniques of error correction. Comput. Commun. 20(15): 1374-1380 (1997).

[80] Danilo Gligoroski, Svein J. Knapskog, Suzana Andova: Cryptcoding - Encryption and Error-Correction Coding in a Single Step. Security and Management 2006: 145-151.

[81] Chetan Nanjunda Mathur, Karthik Narayan, K. P. Subbalakshmi: High diffusion codes: a class of maximum distance separable codes for error resilient block ciphers. 2nd IEEE International Workshop on Adaptive Wireless Networks (AWiN) 2005.

[82] Chetan Nanjunda Mathur, Karthik Narayan, K. P. Subbalakshmi: High Diffusion Cipher: Encryption and Error Correction in a Single Cryptographic Primitive. ACNS 2006: 309-324.

[83] Dmitriy Nikolaevich Moldovyan, Nikolay Andreevich Moldovyan, Sy Tan Ho, Quang

Minh Le, Long Giang Nguyen: New Modes of Using Block Ciphers: Error Correction and Pseudo-probabilistic Encryption. Frontiers in Intelligent Computing: Theory and Applications 2020: 57-68.

[84] Khan, S.N., Loukil, F., Ghedira-Guegan, C. et al. Blockchain smart contracts: Applications, challenges, and future trends. Peer-to-Peer Netw. Appl. (2021). https://doi.org/10.1007/s12083-021-01127-0

[85] Reyna A, Martín C, Chen J, Soler E, Díaz M (2018) On Blockchain and its integration with IoT. challenges and opportunities. Future Gener Comput Syst 88:173–190.

[86] Levi SD, Lipton AB (2018) An introduction to smart contracts and their potential and inherent limitations. In: Harvard law school forum on corporate governance & financial regulation.

[87] Shrestha, A. K., and Vassileva, J. (2016). “Towards decentralized data storage in general cloud platform for meta-products,”. in Proceedings of the International Conference on Big Data and Advanced Wireless Technologies – BDAW (Blagoevgrad), 1–7.

[88] Wang, Z.; Zheng, Z.; Jiang, W.; Tang, S. (2021) Blockchain-Enabled Data Sharing in Supply Chains: Model, Operationalization, and Tutorial. In Production and Operations Management.

[89] Tenopir, C., Palmer, C. L., Metzer, L., van der Hoeven, J., and Malone, J. (2011). “Sharing data: practices, barriers, and incentives,” in Proceedings of the American Society for Information Science and Technology, 48, 1–4. doi: 10.1002/meet.2011.14504801026.

[90] Liu. K.; Desai, H.; Kagal, L.; Kantarcioglu, M. (2018). Enforceable Data Sharing Agreements Using Smart Contracts. arXiv, 1804.10645.

[91] Di Francesco Maesa, D.; Mori, P.; Ricci, L. (2018). Blockchain Based Access Control Services. 2018 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData), 2018, pp. 1379-1386, doi: 10.1109/Cybermatics_2018.2018.00237

[92] Ghaffari, F.; Bertin, E.; Hatin, J.; Crespi, N. (2020). Authentication and access control based on distributed ledger technology: A survey. BRAINS 2020: 2nd conference on Blockchain Research & Applications for Innovative Networks and Services, Sep 2020, Paris (online), France. pp.79-86] which provides a survey of such approaches.

[93] Sultana, T.; Almogren, A.; Akbar, M.; Zuair, M.; Ullah, I.; Javaid, N. (2020). Data Sharing System Integrating Access Control Mechanism using Blockchain-Based Smart Contracts for IoT Devices. Appl. Sci. 2020, 10, 488. https://doi.org/10.3390/app10020488

[94] Songa, L.; Lia, M.; Zhua, Z.; Yuana, P.; Hea Y. (2020). Attribute-Based Access Control Using Smart Contracts for the Internet of Things. 2019 International Conference on Identification, Information and Knowledge in the Internet of Things (IIKI2019). In: Procedia Computer Science 174, pp. 231–242

[95] Nakamura, Y.; Zhang,Y.; Sasabe, M.; Kasahara, S. (2020). Exploiting Smart Contracts for Capability-Based Access Control in the Internet of Things. Sensors (Basel). 2020 Mar; 20(6): 1793

[96] Guo. H.; Meamari, E.; Chien-Chung Shen, C. (2019). Multi-Authority Attribute-Based Access Control with Smart Contract. In Proceedings of 2019 International Conference on Blockchain Technology (ICBCT 2019) (ICBCT 2019). ACM, New York, USA,. https://doi.org/10.1145/3320154. 3320164

Funded by the Horizon 2020
Framework Programme of the European Union

[97] Song, L.; Ju, X.; Zhu, Z; et al. (2021). An access control model for the Internet of Things based on zero-knowledge token and Blockchain. J Wireless Com Network, 105. https://doi.org/10.1186/s13638-021-01986-4.

[98] The ASSURED Consortium, "D1.1 – ASSURED Use Cases and System Requirements", 2021.

[99] The ASSURED Consortium, "D1.2 – ASSURED Reference Architecture", 2021.

[100] The ASSURED Consortium, "D1.3 – Operational SoS Process Models & Specification of Properties", 2021.

[101] N. Koutroumpouchos, C. Ntantogian, S. Menesidou, K. Liang, P. Gouvas, C. Xenakis, and T. Giannetsos, "Secure edge computing with lightweight control-flow property-based attestation," in 2019 IEEE.

[102] D. Papamartzivanos, S. A. Menesidou, P. Gouvas, and T. Giannetsos, "A perfect match: Converging and automating privacy and security impact assessment on-the-fly," Future Internet, vol. 13, no. 2, 2021. [Online]. Available: https://www.mdpi.com/1999-5903/13/2/30.

[103] B. Larsen, H. B. Debes, and T. Giannetsos, "Cloudvaults: Integrating trust extensions into system integrity verification for cloud-based environments," in European Symposium on Research in Computer Security. Springer, 2020, pp. 197–220.

[104] D. Papamartzivanos, S. Menesidou, P. Gouvas, T. Giannetsos, "Towards Efficient Control-Flow Attestation with Software-Assisted Multi-Level Tracing", In IEEE MeditCom 2021.

[105] The ASSURED Consortium, "D4.1 – ASSURED Blockchain Architecture", 2021.

[106] E. Union, «GDPR.eu,» 2020. [Online]. Available: https://gdpr.eu/article-4-definitions/. [Consultato il giorno February 2021].

[107] H. T. T. Truon, M. Almeida, G. Karame, C. Soriente, "Towards Secure and Decentralized Sharing of IoT Data", In CoRR, August 2019.

[108] C. Pouyioukka, T. Giannetsos, W. Meng, "CrowdLED: Towards Crowd-Empowered & Privacy-Preserving Data Sharing using Smart Contracts", In IFIPTM 2019.

[109] The ASSURED Consortium, "D4.2 – ASSURED Secure Distributed Ledger Maintenance & Data Management", 2021.

[110] The ASSURED Consortium, "D4.5 – ASSURED TC-based Functionalities", 2022.

[111] H. B. Debes, T. Giannetsos, "Segregating Keys from noncense: Timely Exfil of Ephemeral keys from Embedded Systems", In DCOSS 2021.

[112] ISO/IEC 20008 (all parts) Information technology — Security techniques — Anonymous digital signatures.

[113] ISO/IEC 20009 (all parts) Information technology — Security techniques — Anonymous entity authentication