TRANSACT

Transform safety-critical Cyber Physical Systems into distributed solutions for end-users and partners

D17 (D3.4)

Solutions for end-to-end security and privacy for distributed CPS v1

This project has received funding from the ECSEL Joint Undertaking (JU) under grant agreement No 101007260. The JU receives support from the European Union’s Horizon 2020 research and innovation programme and Netherlands, Finland, Germany, Poland, Austria, Spain, Belgium, Denmark, Norway.

ECSEL Joint Undertaking
Electronic Components and Systems for European Leadership

Document Information

Project	TRANSACT
Grant Agreement No.	101007260
Work Package No.	W3
Task No.	T3.4
Deliverable No.	D17
Deliverable No. in WP	D3.4
Deliverable Title	Solutions for end-to-end security and privacy for distributed CPS v1
Nature	Report
Dissemination Level	Public
Document Version	v1.0
Date	01/12/2022
Contact	Alexey Kirichenko
Organization	WithSecure
Phone	+358 40 517 4548
E-Mail	alki@withsecure.com

Authors Table

Name	Company	E-Mail
Ralph Weissnegger	CISC	r.weissnegger@cisc.at
Sivam Pasupathipillai	WithSecure	Sivam.Pasupathipillai@withsecure.com
Pilvi Tunturi	WithSecure	pilvi.tunturi@withsecure.com
Heikki Partanen	WithSecure	heikki.partanen@withsecure.com
Perttu Ranta-aho	WithSecure	perttu.ranta-aho@withsecure.com
Andrew Patel	WithSecure	andrew.patel@withsecure.com
Alexey Kirichenko	WithSecure	alki@withsecure.com
Nicola Dragoni	DTU	ndra@dtu.dk
Gaurav Choudhary	DTU	gauch@dtu.dk
Paul Pol	DTU	paupo@dtu.dk
Mateusz Groth	GUT	Mateusz.groth@pg.edu.pl
Peter Mortier	FEops	Peter.mortier@feops.com
Abel Gómez	UOC	agomezlla@uoc.edu
Iván David Alfonso	UOC	ialfonsod@uoc.edu
Wolfram Ratzke	AVL	Wolfram.Ratzke@avl.com
Claus-Henning Friederichs	AVL	Claus-Henning.Friederichs@avl.com
Krzysztof Oborzyński	PMS	krzysztof.oborzynski@philips.com
Juhani Latvakoski	VTT	Juhani.Latvakoski@vtt.fi
Paweł Madej	DAC	pawel.madej@dac.digital
Egbert Jaspers	VIN	egbert.jaspers@vinotion.nl
Ignacio Martínez	SNG	ignacio.martinez@singlarinnovacion.com

Reviewers Table

Version	Date	Reviewer
0.4	18.11.2022	Mateusz Groth (GUT)
0.4	17.11.2022	Wolfram Ratzke (AVL)
0.5	29.11.2022	Sasa Marinkovic (PMS)

Change History

Version	Date	Reason for Change	Affected pages
0.1	09.03.2022	Initial Draft	All
0.4	09.11.2022	Document sent for review	All
0.5	29.11.2022	Applied feedback of external reviewers	All
1.0	30.11.2022	Final version for submission	n/a

Table of Contents

List of Figures

List of Tables

1 Glossary

Term	Definition
2FA	Two Factor Authentication
AES	Advanced Encryption Standard
AI	Artificial Intelligence
API	Application Programming Interface
AWS	Amazon Web Services
BDaaS	Big Data as a Service
BLE	Bluetooth Low Energy
BMS	Battery Management System
BPEL	Business Process Execution Language
CAD	Computer-Aided Diagnosis
CPS	Cyber-physical System
CSMS	CyberSecurity Management System
CSP	Cloud Service Provider
CSPM	Cloud Security Posture Management
CT	Computed Tomography
DAG	Directed Acyclic Graph
DDoS	Distributed Denial-of-Service
DICOM	Digital Imaging and Communications in Medicine
DID	Decentralized Identifier
DLT	Distributed Ledger Technology
DOP	Data-Oriented Programming
DPIA	Data Protection Impact Assessment
DSL	Domain Specific Language
E2E	End-to-End
ECC	Elliptic Curve Cryptography
EEPROM	Electrically Erasable Programmable Read-Only Memory
ESPAR	Electronically Steerable Parasitic Array Radiator
EVF	Electric Vehicle Fleet
FP	False Positive

Term	Definition
GDPR	General Data Protection Regulation
GPS	Global Positioning System
HIPAA	Health Insurance Portability and Accountability Act
HMAC	keyed Hash Message Authentication Code
HSM	Hardware Security Module
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
IAM	Identity and Access Management
IoT	Internet of Things
KPI	Key Performance Indicator
M2M	Model to model (transformation)
M2T	Model to text (transformation)
MAC	Message Authentication Code
MCU	Microcontroller Unit
MDE	Model-driven Engineering
MQTT	Message Queueing Telemetry Transport
MRI	Magnetic Resonance Imaging
NFC	Near-Field Communication
OTA	Over-The-Air
PACS	Picture Archiving and Communication System
PHI	Protected Health Information
PII	Personally Identifiable Information
PKI	Public Key Infrastructure
QoS	Quality of Service
RAM	Random Access Memory
RBAC	Role-based Access Control
REST	Representational State Transfer
RNG	Random Number Generator
ROP	Return Oriented Programming
RoT	Root of Trust
RUL	Remaining Useful Life

Term	Definition
S×C	Security-by-contract
SIEM	Security Information and Event Management
SQL	Structured Query Language
SSRF	Server-side request forgery
STK	Short-Term-Keys
SUMS	Software Update Management System
T2M	Text to model (transformation)
TBLSI	Ticket Based Lightweight Security Integration
TLS	Transport Layer Security
TPM	Trusted Platform Module
UEBA	User and Entity Behavior Analytics
URL	Uniform Resource Locator
USB	Universal Serial Bus
VPN	Virtual Private Network
WWTP	Wastewater Treatment Plant
XML	Extensible Markup Language

Table 1: Terms, Abbreviations and Definitions

2 Introduction

This report is a result of Task T3.4 of the TRANSACT project. This task is to develop solutions to monitor, enforce and manage end-to-end security and privacy of distributed applications, including solutions for (re)qualification. The task investigates solutions and techniques for preserving security and privacy in edge/cloud computing environments when deploying and running distributed applications for safety-critical CPS (Cyber Physical System), taking into account the concepts developed in Task T3.2.

2.1 Role of the deliverable

This document has the following major purposes:

- Documentation of selected solutions for end-to-end security and privacy to be applicable across the TRANSACT domains and use cases.
- Documentation of selected solutions to monitor and manage end-to-end security and privacy in run-time operation.
- Documentation of selected security concepts and risk assessment methods to validate the design of end-to-end security and privacy solutions.
- Documentation of selected methods and controls for monitoring security and privacy, introducing baseline of normal behaviour and detection of abnormal activity based on network traffic or user and entity behaviour analytics (UEBA).
- Documentation of selected solutions for securing communication in distributed architectures, including solutions for robust and secure communication among CPS devices and edge servers within a trusted group of devices to preserve privacy and security.
- Documentation of selected platforms and methods to accomplish AI-based real-time monitoring for security and privacy.
- Documentation of selected AI techniques for detecting anomalies in traffic patterns, such as positional anomalies and missing/incomplete trajectory data, and techniques for generating alerts/alarms indicating safety, security and privacy violations.

2.2 Relationship to other TRANSACT documents

This report has aligned the presented content and results with the ‘sister’ deliverable D3.3 (from Task T3.3), which targets end-to-end solutions for safety and performance for distributed CPS, in order to ensure that this task's outcomes fit in the TRANSACT reference architectures as described in Deliverable D2.1. More precisely, each of the solutions presented in this document has a clear connection to the TRANSACT reference architecture and a description that tells how it is related to or can be used in various use cases of the TRANSACT project.

This document relates to the following TRANSACT deliverables:

- *D1.1 Use case descriptions, end user requirements, state of the art and KPI's (M10)*
  The selected solutions for end-to-end security and privacy for distributed safety-critical CPS are aligned with the needs of the use cases as documented in D1.1 and we will discuss how they can help or be applied to different use cases of the project.
- *D2.1 Reference architectures for SCDCPS v1 (M12)*
  The developed solutions for end-to-end security and privacy for distributed safety-critical CPS are aligned, and ensured to be consistent, with the TRANSACT reference architecture as documented in D2.1.

- *D3.2 Selection of concepts for end-to-end security and privacy for distributed CPS solutions (M12)* The solutions presented in this deliverable are realization of the concepts for end-to-end security and privacy for distributed safety-critical CPS that have been introduced in Deliverable D3.2.
- *D3.1 Selection of concepts for end-to-end safety and performance for distributed CPS solutions (M12) and D3.3 Solutions for end-to-end safety and performance for distributed CPS (M18)* The solutions developed for end-to-end security and privacy for distributed safety-critical CPS are harmonised with the complementary concepts and solutions for end-to-end safety and performance for distributed CPS solutions as documented in D3.1 and D3.3.

2.3 Structure of this deliverable

Task T3.4 has two deliverables: D3.4 (due in M18) and D3.6 (due in M33). The purpose of this breakdown is to allow solutions that require further investigation or are tightly coupled with the use cases enough time to mature. Some of those solutions will appear only in D3.6, while the others will have two versions: the first version will be presented in D3.4, and an extended version will be presented in D3.6.

In this report, after recollecting the TRANSACT reference architecture, use cases, and technical security requirements in Section 3, the solutions are presented as assigned to nine major categories as follows:

1. Secure Communication based Solutions (Section 4)
2. Identity Access and Services (Section 5)
3. Safety, Security, and performance monitoring Services (Section 6)
4. Cloud Security (Section 7)
5. Data Communication Security (Section 8)
6. AI/ML based Solutions (Section 9)
7. Risk Analysis (Section 10)
8. Hardware based security (Section 11)
9. Security Solutions for New Services (Section 12)

We discuss the interplay between the solutions for safety, performance, security, and privacy in Section 13 and we conclude in Section 14.

3 TRANSACT reference architecture, use cases, and technical security requirements

3.1 The TRANSACT reference architecture

The TRANSACT project has adopted a three-tier, device-edge-cloud, architecture concept. Based on this concept, the project has proposed the first reference architecture in deliverable D2.1 (see Figure 1). In the deliverable D2.1, a full description is given of the TRANSACT reference architecture; here a summary is included for positioning the selected end-to-end security and privacy concepts into this reference architecture.

The domain-specific functions or components are depicted in red, yellow, and green depending on their criticality. Domain-specific functions may be offloaded from the device to other tiers. Core TRANSACT components, available to every use case, are depicted in grey. Finally, blue components refer to potential Value-Added functions that may be included depending on the use case.

The TRANSACT reference architecture defines the safety and mission critical functions, the core services and functions, and further value-added services and functions (see Figure 1).

The safety and mission critical functions are key in the safety-critical CPS. The distributed safety-critical CPS solutions will be deployed over three-tier (device-edge-cloud) architecture continuum. Each tier in the architecture provides a specific quality of service level especially with respect to performance aspects, such as response times and data transfer guarantees, ranging from best effort to reliable and time-deterministic data transfers. Safety critical functions often have hard real-time related constraints, whereas the mission critical functions may have soft real-time constraints (which may degrade the system's quality of service when missed, but do not necessarily lead to failures). In the cloud, also Big Data as a Service (BDaaS) services may be deployed.

TRANSACT aims at improving over monolithic CPS by offloading functions to the edge or cloud tier. A few use cases will offload safety-critical functions to the edge tier, more use cases will offload mission-critical functions to edge and cloud. Such offloading gives numerous advantages such as: improved reliability and performance of the device (as fewer services are running on the device), improved efficiency of the offloaded functions due to the use of better hardware in the edge or cloud, improved innovation speed of the distributed CPS as new or upgraded functions can be deployed easier in the edge and cloud.

However, when considering offloading functions from the device, it is critical to ensure CPS system end-to-end safety, performance, security, and privacy. Therefore, several dedicated core services are introduced to cooperatively realize that objective. The safety, performance and security monitoring services are responsible for monitoring, detecting, and preventing safety, security, and performance failures. In addition, they track the devices' KPIs (such as latency, throughput, accuracy, availability) that are used by the operational mode manager (running on the device) and the operational mode coordinator (running at the edge/cloud tier) to decide at runtime whether a device's function can be executed remotely or not.

Another area that the TRANSACT reference architecture addresses is updating the system. To achieve safe and predictable updates to the system, the following core services have been identified: the remote update client (running on the device) and the update coordinator (running at the edge/cloud). Those services cooperate across tiers to perform remote automatic updates of different device services in a secure and safe way. The updates ensure uniform software versions on the tiers and keep the system services up-to-date with the latest functionality. In addition, the automatic updates allow rolling-out a new functionality or introduce new value-added services minimizing system downtime. Each update activity is coordinated with the operational mode coordinator service to keep the system in the safe state at any time.

Further core services address additional security and privacy concerns. The secure access to the system functionality is managed by the identity and access services, which are responsible for granting/denying access to the system resources based on the policies defining who has what access (in which role) to which resources. Other core services contributing to the system security are the auditing services. These services collect information about accessing and using the system to help detect security policy violations when the system is accessed by unauthorized users or in an unauthorized way. The security aspects are also addressed by the (federated) data services and comms services helping in efficient and secured data handling, both in transit and at rest.

In this project, each use case will experiment with the TRANSACT reference architecture, its components, and the selected concepts presented in this deliverable with the aim to capture the overarching results across the various use cases. This allows TRANSACT to validate the approach and refine the proposed reference architecture over the course of the project.

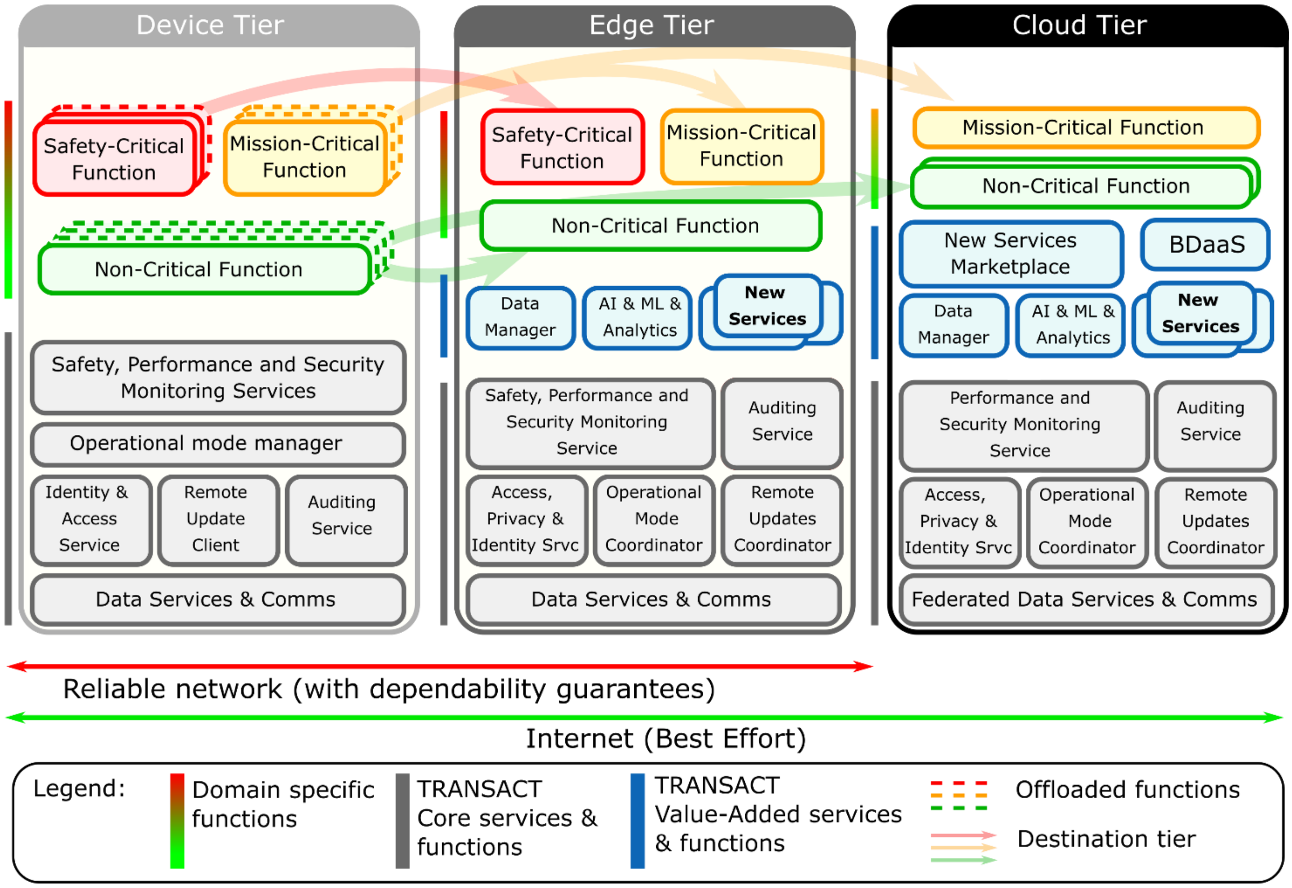

Device Tier
Edge Tier
Cloud Tier
Safety-Critical Function
Mission-Critical Function
Non-Critical Function
Safety, Performance and Security Monitoring Services
Operational mode manager
Identity & Access Service
Remote Update Client
Auditing Service
Data Services & Comms
Safety-Critical Function
Mission-Critical Function
Non-Critical Function
Data Manager
AI & ML & Analytics
New Services
Safety, Performance and Security Monitoring Service
Auditing Service
Access, Privacy & Identity Srvc
Operational Mode Coordinator
Remote Updates Coordinator
Data Services & Comms
Mission-Critical Function
Non-Critical Function
New Services Marketplace
BDaaS
Data Manager
AI & ML & Analytics
New Services
Performance and Security Monitoring Service
Auditing Service
Access, Privacy & Identity Srvc
Operational Mode Coordinator
Remote Updates Coordinator
Federated Data Services & Comms
Reliable network (with dependability guarantees)
Internet (Best Effort)
Legend:
Domain specific functions
TRANSACT Core services & functions
TRANSACT Value-Added services & functions
Offloaded functions
Destination tier

Figure 1: TRANSACT reference architecture.

3.2 Overview of Use cases

3.2.1 Use Case 1 – Remote Operations of Autonomous Vehicles for Navigating in Urban Context (UC1)

In this use case, Fleetonomy and partners will develop a solution for remote control of (semi-) automated vehicles for navigating in urban environments (see Figure 2). The solution will allow vehicles to be moved from one location to another even without a driver, but with a remote operator. The operator will receive continuous feedback on vehicle state and environment, allowing him/her to assist the vehicle to navigate through urban traffic. The vehicle will have autonomy provided by current state-of-the-art automated driving solutions taking care of normal driving, and capable of detecting and reacting to arising hazardous situations.

During the TRANSACT project, the use case team will enhance the capability of the vehicle to understand its surroundings, react to pedestrian and other road user behaviours and make local decisions. The interaction and cooperation of vehicles and human operators in remote operating centre will be seamless and enhanced through visualisation and communication of the vehicle understanding and intent in augmented reality camera view and user interfaces with 3D data model of the driving environment. This allows the remote operator to understand the vehicle's independent capability to manage safe driving in a complex environment including people in different roles. The remote operator provides supervision and additional safety as well as the intelligence to resolve arising exceptional traffic situations. Hand-over of control between operator and vehicle is performed in smart way.

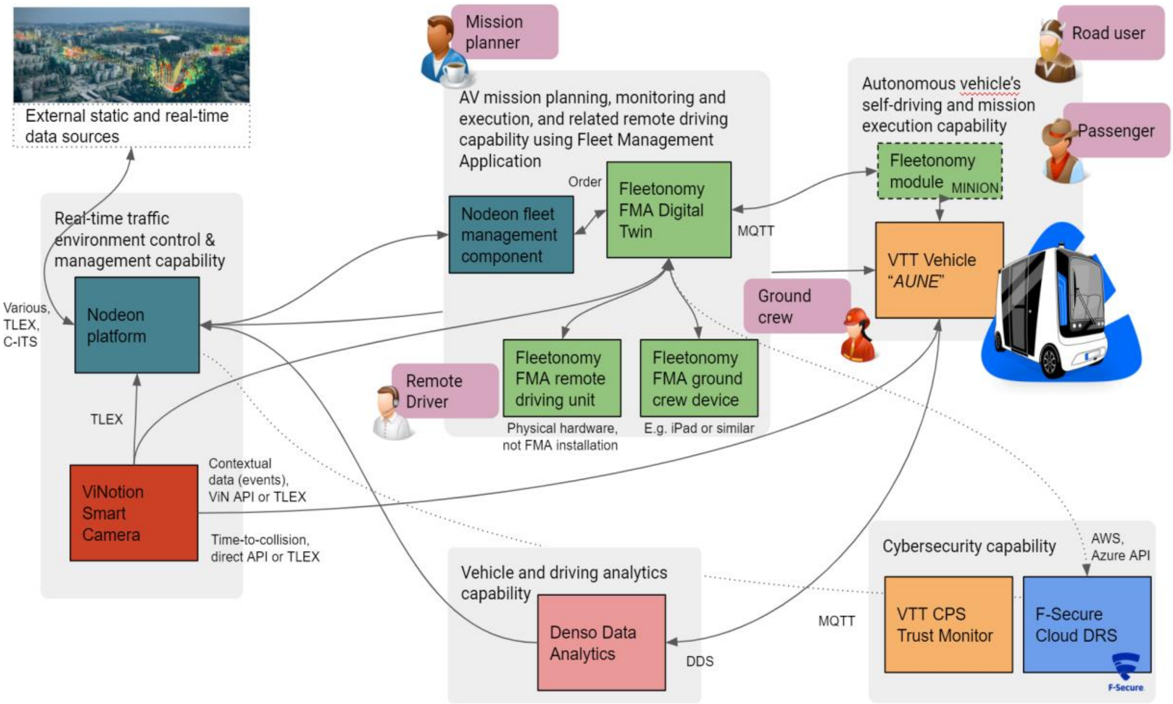

External static and real-time data sources
Mission planner
AV mission planning, monitoring and execution, and related remote driving capability using Fleet Management Application
Road user
Autonomous vehicle's self-driving and mission execution capability
Passenger
Real-time traffic environment control & management capability
Order
Nodeon fleet management component
Fleetonomy FMA Digital Twin
MQTT
Fleetonomy module
MINION
VTT Vehicle "AUNE"
Various, TLEX, C-ITS
Nodeon platform
Ground crew
Remote Driver
Fleetonomy FMA remote driving unit
Fleetonomy FMA ground crew device
Physical hardware, not FMA installation
E.g. iPad or similar
TLEX
ViNotion Smart Camera
Contextual data (events), ViN API or TLEX
Time-to-collision, direct API or TLEX
Cybersecurity capability
AWS, Azure API
Vehicle and driving analytics capability
Denso Data Analytics
MQTT
DDS
VTT CPS Trust Monitor
F-Secure Cloud DRS
F-Secure

Figure 2: The remote operations use case cloud-edge-device continuum

TRANSACT security challenges: The architecture should be able to negotiate the Confidence Level with Vehicle automatically. Communication channel between data exchange hub and end-user must be secure and safe, end-to-end protected.

3.2.2 Use Case 2 - Critical Maritime Decision Support Enhanced by Distributed, AI Enhanced Edge and Cloud Solution (UC2)

The maritime use case will demonstrate advancements in safe and efficient maritime navigation made possible by enhancing the existing basic edge/cloud technologies in the NAVTOR e-Navigation Suite to the TRANSACT architecture. This will allow for integration of traditional advisory services, AI-based advisory services, and data-analytics services into the device-edge-cloud continuum to improve safety, efficiency, and security, as will be demonstrated for automated High Sea vessels and an autonomous harbour-based support vessel. In the Figure 3, NAVTOR's pre-TRANSACT e-Navigation suite is illustrated. In the Figure 4, the planned device-edge-cloud services are detailed, building a holistic AI-based monitoring and decision support service for safe and efficient navigation.

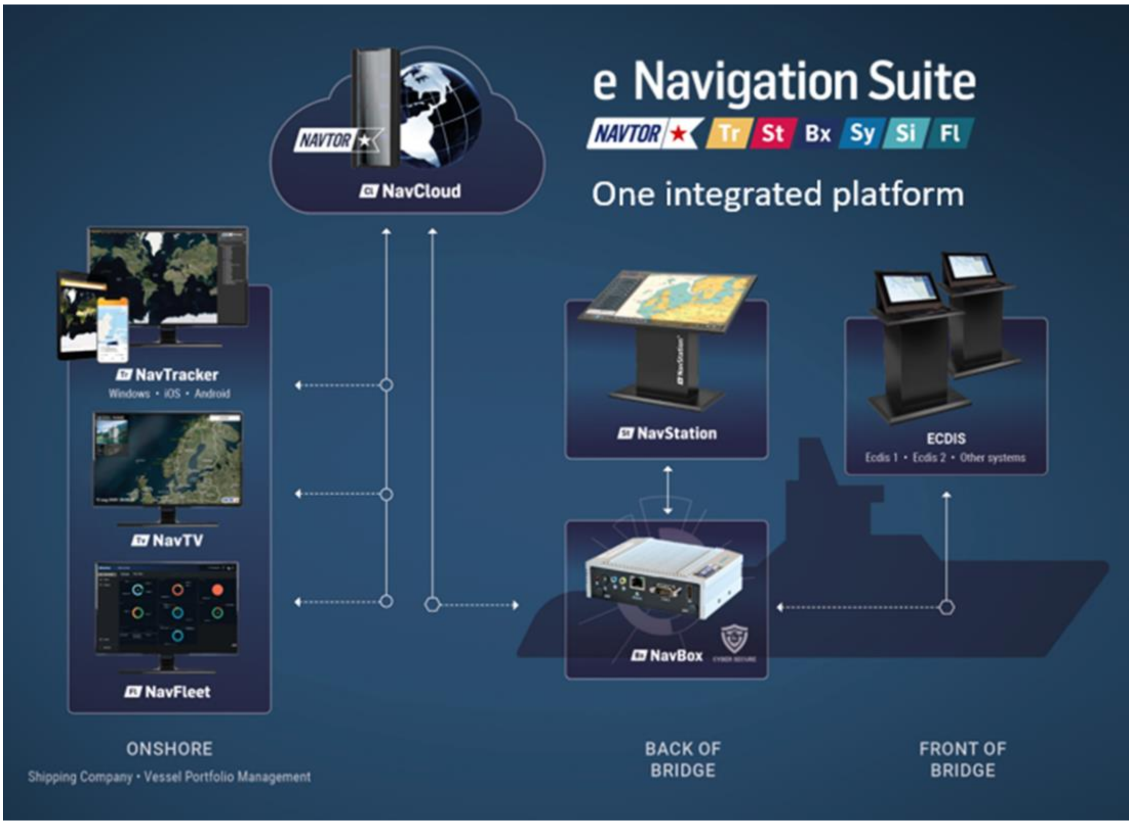


Figure 3: NAVTOR's pre-TRANSACT e-Navigation suite

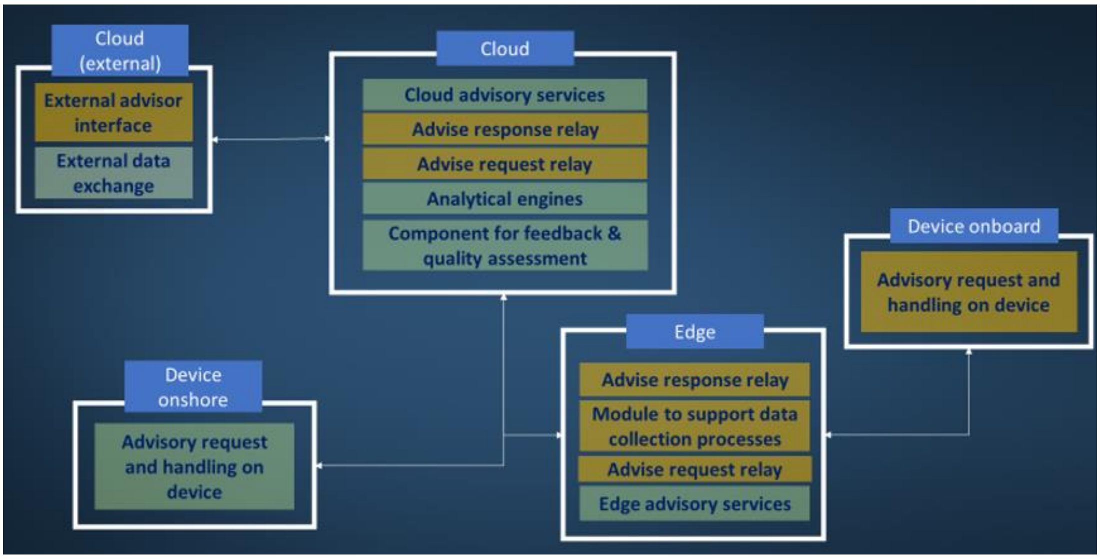


Figure 4: NAVTOR's e-Navigation suite built on TRANSACT architecture; yellow boxes are pre-TRANSACT, green boxes are by TRANSACT project, and will be demonstrated by UC2.

TRANSACT Security challenges: The security challenges for the High Sea demonstrator are mainly related to the SatCom-connection between Vessel and Shore, in addition to the "connected Bridge" setting up a secure connection between Front of Bridge and Back of Bridge strictly when required. Normally, the real-time navigation system (ECDIS) is for security purposes a stand-alone system. In TRANSACT, connection between vessel and shore is given, and security and performance issues must be handled by secure communication between cloud-based advisory services and the vessel-based edge advisory services, utilizing decision data structure updated from the Cloud. The safety and security critical communication between the device (ECDIS Front of Bridge) and the edge (NavBox Back on Bridge) and a new security structure, including new APIs, has to be developed to take advantage of the AI-based Cloud advisory services. Due to vessels' satcom-related challenges, a distributed PKI system will be investigated.

In the demonstrator related to an unmanned surface vessel in port, near-real-time secure communication is a must, and security mitigation actions will be investigated to enhance the security level of the wireless communication.

List of the main security-related requirements includes: Distributed PKI (as vessel is off-line at certain times); remove data sharing by USB-sticks; encryption mechanisms on all messages; detection of false sensor or data injection; detection of spamming/jamming of signals.

3.2.3 Use Case 3 - Cloud-featured battery management systems (UC3)

Vehicle battery data is collected and transmitted using an advanced and secure data logger and transferred encrypted to a data broker cluster; the data is stored in an optimized database. All of this is happening while the Electric Vehicle Fleet (EVF) is driving. In the backend, the data is analysed and used for the improvement of functionality (e.g., time left to charge), safety or autonomous driving (e.g., fail-operation in Battery/BMS). Such improvements are sent back to the EVF, where the infrastructure is used in the opposite direction. The vehicles in the EVF are now consumers and consume the software update. All of this still happens in an encrypted way, ensuring the integrity of the software update. A further topic is the handling of the impact of low state-of-charge (keyword: "safe energy supply") on autonomous functionalities. Necessary updates or system decisions can be done over-the-air; safety-relevant warnings can be communicated to the driver. The generated data leads to a better estimation of battery remaining useful life (RUL), battery failure prediction and error management.

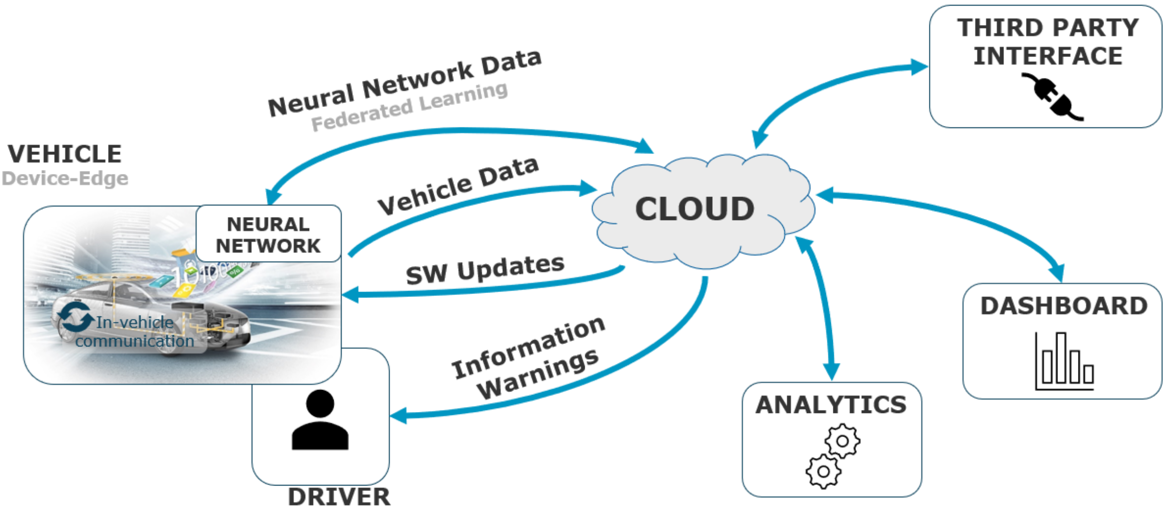


Figure 5: Involved components and communication paths of the cloud-featured battery managements use case.

TRANSACT security challenges: So far, the battery management system has been a closed system. The security is inherently guaranteed by the lack of possibilities to connect to the hardware. Dedicated hardware and special knowledge are required to access and change the software. Now, the system will be transformed into a distributed system which has a permanent accessible connection. With the increase of attack surface, the challenges arise.

Data are no longer processed locally but exchanged with the cloud backend. In addition to technical telemetric data, personal data are of interest as well. Together with state-of-the-art encryption, further methods are investigated to establish a secure-by-design transmission channel. That means data are pre-processed and abstracted before they are transmitted and stored.

Another challenge arises with the new possibility to perform software updates over the air. By design of the electrical-electronic architecture, the LTE gateway will have access to any control unit within the sub-system. It must be guaranteed that only an authorised person can access the gateway while, e.g., roles limit the control of the functions. Since changes to the software can be performed easier, mechanisms must be established to confirm the integrity.

3.2.4 Use Case 4 - Edge-cloud-based clinical applications platform for Image Guided Therapy and diagnostic imaging systems (UC4)

Use Case 4 “Edge-cloud-based clinical applications platform for Image Guided Therapy and diagnostic imaging systems” is a healthcare use case, aiming to improve the workflow and interoperability in hospitals. In particular, the use case addresses image-based minimally invasive clinical procedures which are typically performed in Cathlabs or Operating Rooms (see Figure 6).

In the currently deployed system, the security and privacy of data is guaranteed on a number of levels, i.e., by restricting access to the system in Cathlab (physical security), using the user access management to restrict system functionality and data access as needed per user roles, and storing all sensitive data in the system encrypted. Moreover, when healthcare data need to leave hospital environment, it is anonymised to ensure data privacy.

Figure 6: Typical workflow setting during image guide therapy with physicians utilizing medical imaging equipment for the minimally invasive treatment of patients

TRANSACT security challenges: Changing the architecture of the healthcare diagnostic imaging systems from the centralized, on-device solution toward the distributed, cloud-based architecture significantly increases the attack surface of the new solution by making it more vulnerable for security threats. Also, the data privacy concerns are growing significantly in such an architecture as the healthcare data is highly sensitive and requires special care not to be exposed due to being transferred over a public network or due to security attacks and software vulnerabilities.

The new edge/cloud-based architecture of the healthcare diagnostic imaging systems should ensure that the risks of security breaches and privacy violations are minimized. Therefore, one of the biggest challenges in the healthcare systems is to ensure the end-to-end security and privacy, i.e., the system design and deployment need to apply security mechanisms ensuring proper safeguards to comply with the regulatory requirements and to prevent disclosure, compromise, or misuse of the stored and processed healthcare data. The new edge-cloud-based components implementing security- and privacy-related functionality need to be designed with security and privacy in-depth approaches to ensure adequate quality and protection of the processed healthcare data.

Moreover, as the clinical procedures are typically very complex and involve a team of healthcare professionals (with a variety of expertise in multiple disciplines, who are located inside and outside the hospital) their effective collaboration is paramount to ensure the best treatment outcome for the patients. Therefore, ensuring the security and privacy of the shared healthcare data is critical for more efficient collaboration of healthcare specialists within and outside of the hospital's Cathlab.

3.2.5 Use Case 5 – Critical wastewater treatment decision support enhanced by distributed, AI enhanced edge and cloud solutions (UC5)

Use Case 5 "Critical wastewater treatment decision support enhanced by distributed, AI enhanced edge and cloud solutions" is an industrial use case, which addresses three problems: the detection upon industrial discharges, the need for better strategies for equipment maintenance, and the need for a more efficient cross-WWTP operation.

Wastewater treatment plants (WWTP) goal is cleaning sewage and water coming from citizen consumption, drainage and rainwater with the propose of returning these wastewater streams safely to the environment. Sometimes, the environmental areas where the treated water is discharged are sensitive or protected areas and, therefore, the correct water depuration has a strong impact on the environment, population welfare and agriculture in the surrounding zones. Therefore, disruptions and dysfunctions in the management of the main processes related to the achievement of proper water quality may lead to high risks to the society, the environment and the local economies. The most extended kind of WWTPs involve physic-chemical treatment and biological treatment in different stages for removing solids and pollutants, breaking down organic matter and restoring the oxygen content of treated water (see Figure 7).

Unfortunately, disruptions in the depuration processes usually happen, especially in industrial areas, where the WWTP are severely affected when the toxic spills reach the facilities, leading to an interruption in the operation of the critical biological reactors and failures in appropriate water depuration. Therefore, these toxic discharges have impact on the environment and could seriously affect the protected natural area (e.g., fish kills). The re-establishment of each biological reactor may involve around 20k - 25k euros, aside from the heavy penalties for the plant managers.

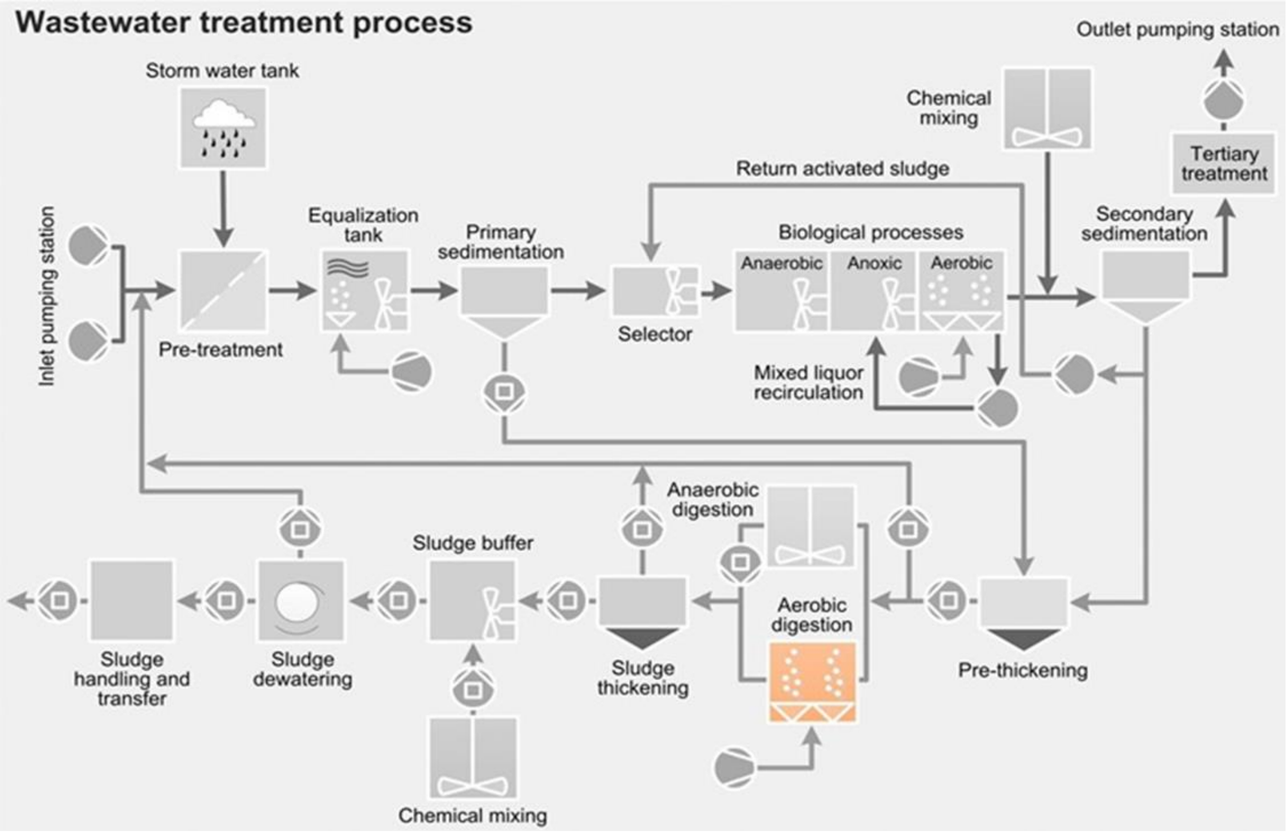


Figure 7: General scheme of a typical wastewater treatment plan process

TRANSACT security challenges: The security challenges in this use case are mainly related to the Authentication and Authorization Attacks. The system must implement an artificial intelligence algorithm capable of detecting anomalies in the behaviour of each machine. The system must identify rare elements, events, or observations of the parameters of the machine that raise suspicion by significantly differing from the usual or daily behaviour. The system should be protected against attacks on edge computing infrastructures and cloud services.

4 Secure Communication based Solutions

4.1 Secure transmission of data from device/edge to cloud

4.1.1 Overview, motivation

With the transformation from an isolated and disconnected to an open and connected solution, new security risks emerge. Especially the wireless communication between the cloud service, edge and the device exposes additional attacking points, but even wired connections may be exposed to attacks if a malicious device is added to the network or an existing device is manipulated to show malicious behaviour.

To authenticate between two or more electronic units, cryptography is used to sign a message and ensure integrity of transferred data. This requires a proper handling and storage of cryptographic keys in all the instances taking part in the communication.

Hence, it is crucial to establish secure communication channels which are based at least on the state-of-the-art security standards and preferably offer the ability to install upgrades to provide crypto agility.

4.1.2 State of the art

To establish a connection between a cloud service and devices (e.g., cars), different brand-specific systems are present in the market. These systems are integrated services which offer automotive specific functions for the car user to get access via smartphone or the use of online services provided by the in-vehicle infotainment system. Those systems are for instance 'VW-AC (Volkswagen Automotive Cloud) / Volkswagen We Connect', 'GM Onstar / OPELConnect', 'Mercedes me connect'.

Depending on the brand-specific design, these systems are based on OMA DM (Open Mobile Alliance – Device Management), or the open network protocol MQTT (Message Queueing Telemetry Transport) or other custom specific messaging solutions.

MQTT, for instance, gathers data from lightweight electronic units (devices) to a central broker (cloud service) and system orders can be distributed by the broker to the electronic devices (see Figure 8). As an option, MQTT offers the use of Transport Layer Security (TLS) as cryptographic service to encrypt the transferred data to prevent them from manipulation and eavesdropping. Other systems may use certificates in a PKI which relies on a trusted instance built up around the communication system.

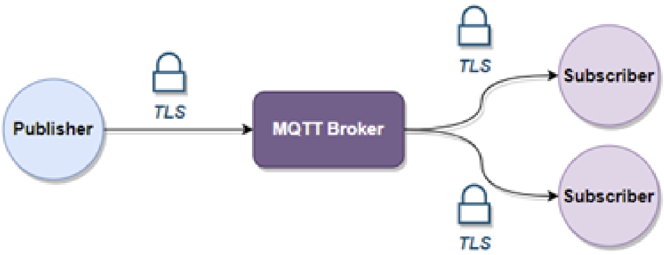


Figure 8: Example of using MQTT with TLS.

OMA DM is a similar system to exchange XML data based on a request-response protocol including an authentication process to switch over to communication (see Figure 9). Once the communication is established between the server and client, a sequence of messages might be exchanged to complete a given device management task. Either server or client might issue notifications or alerts, which are messages that can occur out of sequence to establish and terminate connections or trigger error handling.

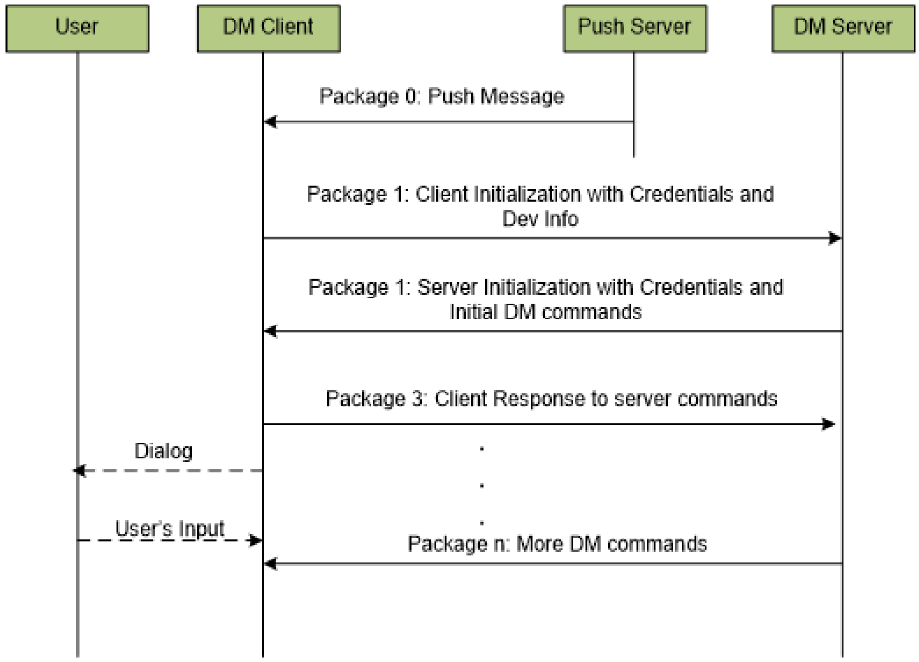

User
DM Client
Push Server
DM Server
Package 0: Push Message
Package 1: Client Initialization with Credentials and Dev Info
Package 1: Server Initialization with Credentials and Initial DM commands
Package 3: Client Response to server commands
Dialog
User's Input
Package n: More DM commands

Figure 9: Data exchange phases in OMA DM

Today's most popular cryptographic primitives in embedded systems are AES (Advanced Encryption Standard - symmetric) and ECC (Elliptic Curve Cryptography - asymmetric) as well as DH (Diffie-Hellman) for the negotiation of keys to build up a secure channel.

As stated before, strong cryptographic protocols rely on the proper handling of the cryptographic keys. In the existing security architectures, this is mostly solved by using a PKI (Public Key Infrastructure) managing asymmetric keys and certificates for authentication of communication partners involved. Public (asymmetric) keys to encrypt data can be authenticated using certificates, but this can be done also by a signed key exchange.

4.1.3 Innovation step

To organize a security environment for lightweight electronic units, TBLSI (Ticket Based Lightweight Security Integration) is proposed to be used for handling the security issues of the use case 'BMS' (see Figure 10). This protocol is designed for a typical automotive workshop-diagnostic structure consisting of a diagnostic tester (coordination service - agent) to perform diagnostic operations, a cloud service (policy administration - server) to manage policies and credentials, and electronic units (telemetric unit - device) to be administrated in a vehicle.

TBLSI organizes (gives credentials for) the access to diagnostic functions of every electronic unit involved in this security environment. Furthermore, an access profile (eligibility to perform a set of operations), session keys to establish secure channels and SOC (Secure Onboard Communication) keys for communication to other units in operation mode can be transferred into the unit to ensure authenticated and encrypted data exchange.

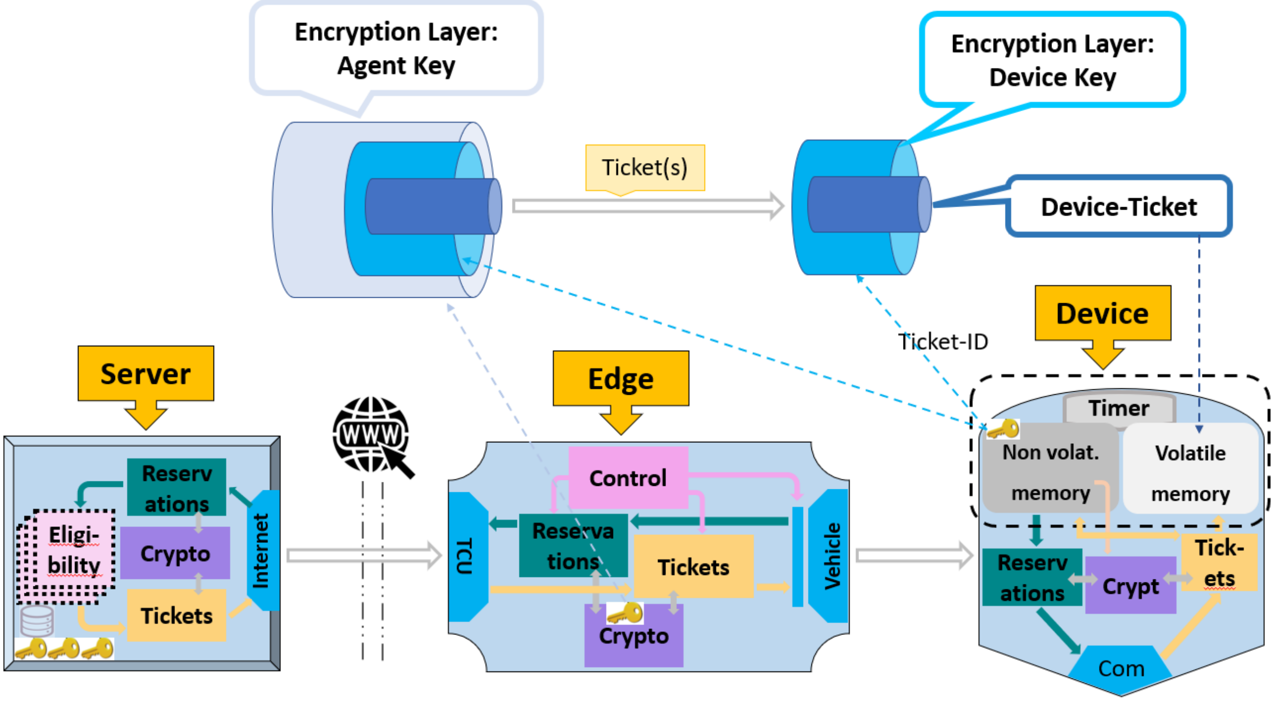


Figure 10: Entities of TBLSI and nested encryption layers from server – agent - device

Up to now, TBLSI is implemented in a demonstrator setup consisting of five Raspberry Pis to showcase the functionality of the protocol. TBLSI is used to open administrative functions on the electronic units to set up configurations, read diagnostic data and prepare settings, e.g., cryptographic keys (for SOC), for running in the operation mode. TBLSI does not support the operational part of security like SecOC, MACsec, IPsec etc., this has to be implemented separately.

4.1.4 Application to use cases

In the TRANSACT's use case "Cloud Featured Battery Management System" (UC3), this protocol can be adapted to the BMS functionality to secure the data transmission of telemetric data, remote procedure calls or updates. This can be done in two ways:

1 – In an initial setup operation, cryptographic keys are set to establish permanent secure channels to exchange data between device/edge and server.

2 – For each data exchange, 'tickets' are issued by the server to install temporary session keys for secure channels.

The chosen BMS system is not capable of installing additional software for the security functionality and does not allow self-installation of software, so an additional component on the device side has to be added to the TRANSACT demonstrator setup which performs the security functionality and device control. This can be an LTE Gateway which has to be declared as a solution for demonstrating what functionality should be integrated into the BMS components in the future.

TBLSI ensures that all administrative communication between telemetric unit, coordination service and policy administration in the cloud backend is always signed (authenticated and integrity protected) and encrypted (hides secrets and prevents eavesdropping).

4.2 Security and integrity for over-the-air updates

4.2.1 Overview, motivation

The TRANSACT architecture enables easy data installation for configuration setting and software updating of individual components. However, as the update procedure may be a remote one, i.e., over the air, one must focus on the security and integrity of updates and the installed software. Otherwise, it would be easy to deploy malicious software which could infect the whole system. Moreover, there may be a large number of devices which are recipients of software updates and are prone to systematic attacks. Hence, software updates need to be transmitted in a secure manner and the integrity of software update packages must be guaranteed.

4.2.2 State of the art

Software updates in cars in the market are usually performed in a workshop using a workshop tester (workstation for SW installation) plugged in the OBD connector. In computers, smartphones and IoT devices, software updates are usually performed OTA (over-the-air). Modern cars increasingly support OTA updates initiated by a central cloud service to keep the systems up-to-date.

Especially in road vehicles, safe and secure software updating is strictly recommended to keep the systems in a safe operational mode. Today, there are different architectures in the market to offer updates to a subset of electronic units to gain experience in the field:

1 – Software packages can be downloaded via WiFi over home internet at a chosen time.

2 – Software packages can be downloaded to a smartphone and transferred to the car via Bluetooth.

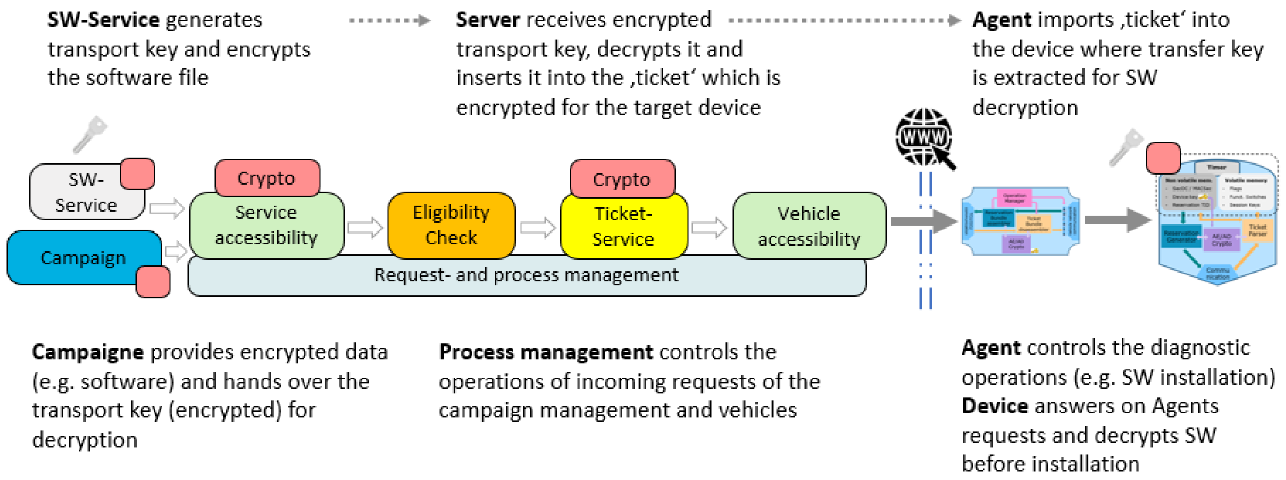
3 – The vehicle downloads software packages over cellular network and offers installation at a chosen time.

To ensure proper installation of software, the installation process has to be designed ‘fail safe’, what means the success of the operation has to be proven accurately and failures have to be handled to lead to a safe state of the vehicle. This requires an update manager with capabilities to perform or roll back an update process together with the ability to lock and restart coherent functionality.

In addition, the integrity of the software has to be ensured using digital signatures which are verified inside the target unit. A modern approach is to deliver the cryptographic key for verification in a certificate which is verified against the public root key in the PKI of the security environment. To prevent eavesdropping of software content, the data can be transferred encrypted and revealed short before writing it to program memory. Therefore, a device may generate a key pair and export the public key for encryption of the delivered software. To perform such operations, a strong security architecture with reliable security controls is required.

4.2.3 Innovation step

As described in Section 4.1, the proposed TBLSI protocol is capable of inserting a temporary symmetric session key, secured using a device-specific key, into the target device, which sets the endpoint of a secure channel to transfer the software file into the device for writing it to the memory (see Figure 11).



SW-Service generates transport key and encrypts the software file

Server receives encrypted transport key, decrypts it and inserts it into the ‚ticket‘ which is encrypted for the target device

Agent imports ‚ticket‘ into the device where transfer key is extracted for SW decryption

Campaigne provides encrypted data (e.g. software) and hands over the transport key (encrypted) for decryption

Process management controls the operations of incoming requests of the campaign management and vehicles

Agent controls the diagnostic operations (e.g. SW installation) **Device** answers on Agents requests and decrypts SW before installation

Figure 11: Transport of encryption key for SW update via ‘ticket’ from server via agent to device.

TBLSI has to be integrated to a SW delivery architecture to ensure secure transport to the target system which needs to be a system capable of performing the update, including the flashing procedure of the control units. The update manager should be implemented with respect to UNECE R 155/156 to ensure proper handling of the update operation and to meet future legal regulations.

4.2.4 Application to use cases

The UNECE R 155/156 is the future regulation for cybersecurity in road vehicles to ensure resilience of vehicle systems against operation failures and cyberattacks, especially in the context with connection to a cloud system.

The core requirement of UNECE R 155 is to establish a CSMS (cybersecurity management system), which means a “systematic risk-based approach to define organisational processes, responsibilities and governance for risks in context with cyber threats for vehicles”.

In the future, a CSMS system is regarded as demanded for homologation of road vehicles for use on public roads with these characteristics:

a. Gathering and approval of information regarding this regulation for the whole delivery chain.
b. Documentation and assessment of risk mitigation measures in a context with construction-specific information.
c. Implementation of suitable cybersecurity measures in the conception phase of the vehicle.
d. Concept for detection of and reaction to possible cyberattacks.
e. Recording of data to support the detection of cyberattacks and provision of forensic data.

The core demand of UNECE R 156 is to establish a SUMS (software update management system) to ensure cybersecurity and protection against manipulation which means a “systematic approach to define organisational processes to meet the requirements of this regulation for software updates”. The manufacturers shall be able to detect and to fix security gaps and vulnerabilities remotely. Furthermore, it shall be made obvious for drivers, vehicle owners and responsible authorities what effects the software updates have on homologation parameters to reconstruct the concession and observance of governance regulations.

One of the requirements to a SUMS is to develop a process for documentation and preserving of all relevant information around the update process to be able to provide it to appropriate authorities on demand. In addition, a process to identify target vehicles and document the history of update operations shall be established.

Regarding the TRANSACT architecture, it should be verified if and how these regulations can be met in an AI-based parameterisation and software correction of a safety-critical system.

4.3 Wireless end-to-end security channels using PKI (BLE, NFC, LTE/5G)

4.3.1 Overview, motivation

When exchanging sensitive data between different devices, it must be protected. In this sense, security paradigms such as authentication, data confidentiality, and integrity are important tools. One way to increase security is through the use of secure elements (SE). The following subsections examine how to include a secure element to a communication module used for battery monitoring and infrastructure communication.

Key objectives related to Security/Safety/Privacy, Interoperability, Life Cycle Management amongst others, need to be taken into account through a deep use case analysis in order to propose appropriate solutions.

The requirements are:

- The personalization process needs to support methods to securely exchange data between all involved stakeholders.
- Secret keys stored on the device are the base for a secure and tap-proof communication channel between sensors, gateways and a backend cloud system.
- To ensure secure storage and execution of cryptographic operations, the use of Secure Elements is suggested for a root of trust.
- To ensure valid product configurations, a verification process is required which validates the configuration against the requirements.
- Furthermore, methods are investigated to securely import stakeholder generated chip-individual data into the personalization process. To ensure the confidentiality of the data throughout all the process stages, secure hardware solutions are investigated for a secure Key Management System.

For the threat model, it is assumed that an attacker can potentially listen / record all received / sent messages between the sensor node and the gateway. Further, she/he has access to the sensor node and may manipulate the memory content of the (flash) memory of the microcontroller. The attacker, however, may not be able to get access to keys / functionality that is run on the Secure Element since it is a certified cryptomodule with tamper-resistant measures.

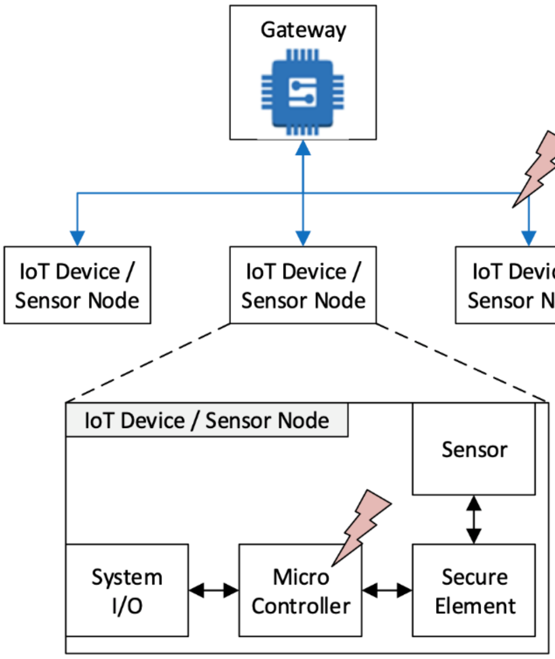


Figure 12: Application of Secure Elements

The main objective is to provide means on gateway and cloud layer for assuring the integrity of the chain of trust, which starts from data collection on smart sensor devices and ends on the service provider side. This main goal can be achieved by taking into account the following requirements:

- Confidentiality: In order to meet the high-level security demands towards the end-to-end security paradigm, the gateway is extended with a secure element, providing the necessary means for secure handling of cryptographic keys and cryptographic primitives.
- Privacy: When data, especially device-related data needs to be passed between different parties for enabling collaboration, privacy aspects become more important. Therefore, privacy-critical data is tokenized and managed throughout its lifecycle (creation, storage, and transfer).
- Integrity and Authentication: A custom public key infrastructure (PKI) forms the basis of the platform's security and ensures the identity of all communication participants (defined in D3.2, Chapter 6.7). Furthermore, by relying on asymmetric and symmetric cryptographic primitives, the integrity and authenticity of the transferred data can be achieved.

The overall framework (Figure 12: Application of Secure Elements) shall provide secure authentication between the following components:

- Cloud to Cloud: Data shall be passed between service marketplace (federated service engine) to other service provider, while preserving the end-to-end security paradigm.
- Sensor to gateway: Encrypted data is passed between smart sensor nodes and the gateway.
- Gateway to Cloud: Data is collected offline and uploaded to the cloud via a secure channel.
- Authorization: The process of getting access to specific datasets shall be secured taking into account the server-to-server as well as sensor-to-gateway scenarios. In this sense, only authorized entities shall access the data in question.

4.3.2 State of the art

The data collection system elaborated within the project is oriented towards the system architecture of common IoT systems (Figure 13, Figure 14).

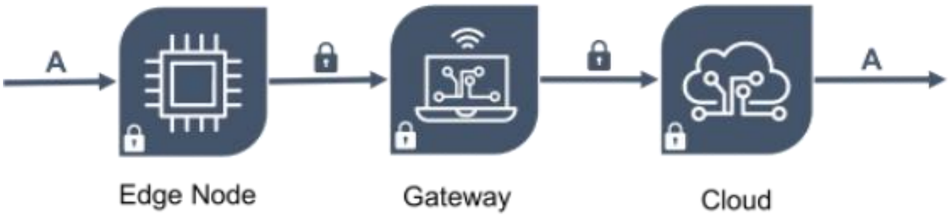


Figure 13: Data collection system in the project uses common IoT system architecture.

Common Edge Node – Data Collector. A power and computational resources constrained device. It collects data via sensors from the local environment at constant intervals. It encrypts all the data it retrieves before forwarding it to other devices.

Gateway – Device Connector. The gateway acts as a connecting unit between edge devices and the virtual kiosk. It receives data from locally available edge devices, optionally extends the data bundle by adding additional information, and forwards it to the cloud. The gateway is a trusted communication participant but cannot read the received encrypted data.

Server – Data Repository. This server is the direct point of contact for the gateway device and receives the secure data package. The server can decrypt the received data.

As an edge node, a Raspberry Pi 3 Model B device with integrated sensors was used. The secure element SE050 was attached to the edge node. It works as an HSM (Hardware Security Module) and increases the security of the host controller concerning cryptographic key and certificate management and distribution. The SE050 connects and communicates to the host MCU (Microcontroller Unit) using the chip's I2C interface. The host MCU runs the application logic and controls all cryptographic operations (encrypting, signing, hashing, etc.). The latter was achieved through software integration of the SE050 support package. Regarding wireless connectivity, the Raspberry Pi's integrated BLE 4.2 chip was utilized to establish local communication to the gateway. The gateway itself is a Raspberry Pi 4 Model B controller that supports BLE and has an integrated Wi-Fi module and an Ethernet port that are used for HTTPS calls to a server. The server was realized as an Amazon EC2 instance and extended with the AWS Cloud-HSM. The server's application connects to the HSM using mutually authenticated TLS channels established by the HSM client software.

4.3.3 Innovation step

Root of Trust:

The A71CH Plug and Trust Secure Element of NXP provides a solution for secure provisioning. This is achieved by serving as a root of trust for the user. The root of trust is given by a pre-provisioned token. This token is provisioned while manufacturing the chip and enables the user to build a secure connection to the internet of things.

Using this secure connection, the user is able to securely provision his/her device with other tokens and credentials or even provision other hardware parts with desired information or intellectual property. The chip serves as a secure element and secure memory and is able to generate additional secure keys. These attributes guarantee a secure peer-to-peer connection as well as a secure connection to the cloud.

The chip is also outfitted with security measures preventing many physical and logical attacks and offers plug-and-play capabilities to ensure zero-touch secure provisioning. All of the advantages of the chip mark it as a good solution for smart-home, smart-industry or even smart-cities.

The integration of a secure element offers several benefits regarding use cases related to security and privacy. As the secure element comes with a pre-configured asymmetric key pair, the private key does not need to be stored or transmitted outside the secure element. The public key can be distributed freely and is also supplied by the manufacturer together with the fresh and untouched hardware. This circumstance enables multiple security features:

- Encrypted throughout lifetime: All connections with the gateway are encrypted during its complete lifetime. The private key, securely stored in the device, and the public key, known by all stakeholders, can be used to securely exchange session keys for each session and use common technologies like Transport Layer Security (TLS) protocols for secure data transmission.
- Secure Commissioning: The initial settings sent to the device can be cryptographically verified using the key pair from the secure element. This ensures that any updates, software, settings or any other data, are safe from manipulation during the download to the gateway device.
- Secure Boot: Even if no data is transmitted, devices in the field are prone to manipulations by attackers who have physical access to them. A secure element, which is designed to be hard to manipulate, can be used to verify the overall system's state and especially the main application during the boot phase and detect unauthorized manipulations to it.

The security features of the protocol comprise multiple steps:

Initial distribution of certificates: The gateway is shipped with a configured key pair. The public key is known (supplied by manufacturer) to a cloud backend which subsequently can issue a certificate proving the authenticity of the gateway's public key. The client, which is connected to the cloud backend and knows its identity, trusts certificates issued by and gets its own certificate from the cloud backend.

Authentication handshake: Both sides now have asymmetric key pairs and can prove authenticity of their public keys. When a communication channel is established, both parties exchange initial handshake messages which prove their identity and prepare each side for the further process/protocol.

Payload messages: Depending on the agreed protocol in step 2, the gateway and the connected consumer device will exchange an arbitrary number of payload messages. The content of these messages can be defined freely by higher level application protocols.

Finalization: After finalization of a (possible) higher level application, a standardized finalization message is exchanged between the gateway and the consumer device and the gateway stores the summarized outcome for later traceability.

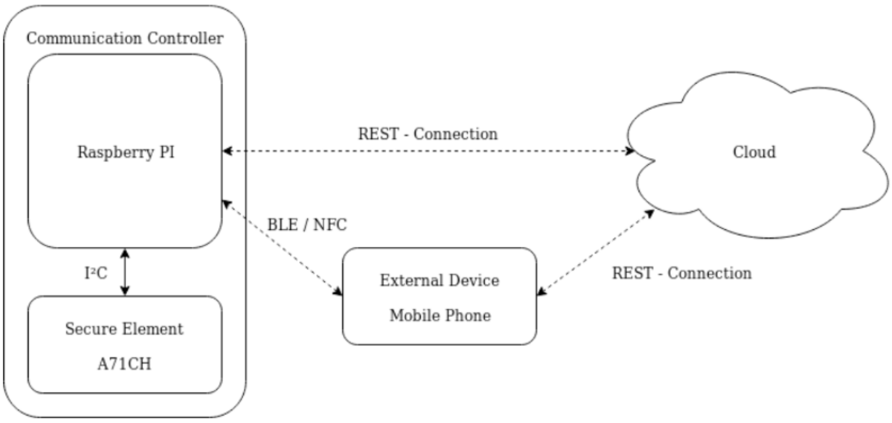


Figure 14: Architecture of secure cloud connectivity

4.3.4 Application to use cases

The proposed solution will be applied to UC3 – cloud-featured battery management system. Together with AVL and TU Graz, the secure wireless end-to-end communication is applied to the setup prepared by the partners. CISC's solution will provide security and privacy (concerning user data) protection from the sensor to the cloud.

4.4 Wireless Communication Security

4.4.1 Overview, motivation

Wireless communication is a subject to various types of interference: either unintentional (such as other communication signals in the range) or intentional (which aim is to deliberately corrupt the communication). From a security point of view, both types should be considered a potential threat to the security, though intentional interference is a bigger risk.

In order to prevent the risk, a modification in the physical layer of communication can be implemented. For that purpose, the use of a reconfigurable antenna, which allows to adjust the radiation pattern, can minimize the impact of the jamming attack on the quality of the received signal. An illustration of this solution is presented in Figure 15 below. In the case of an attack involving the transmission of a signal, the received signal is very noisy and impossible to read. The use of a reconfigurable antenna with the selected optimal shape of the radiation beam allows to "filter out" the jamming by reducing the share of its power in the received signal. As a result, it is possible to correctly receive the transmitted message.

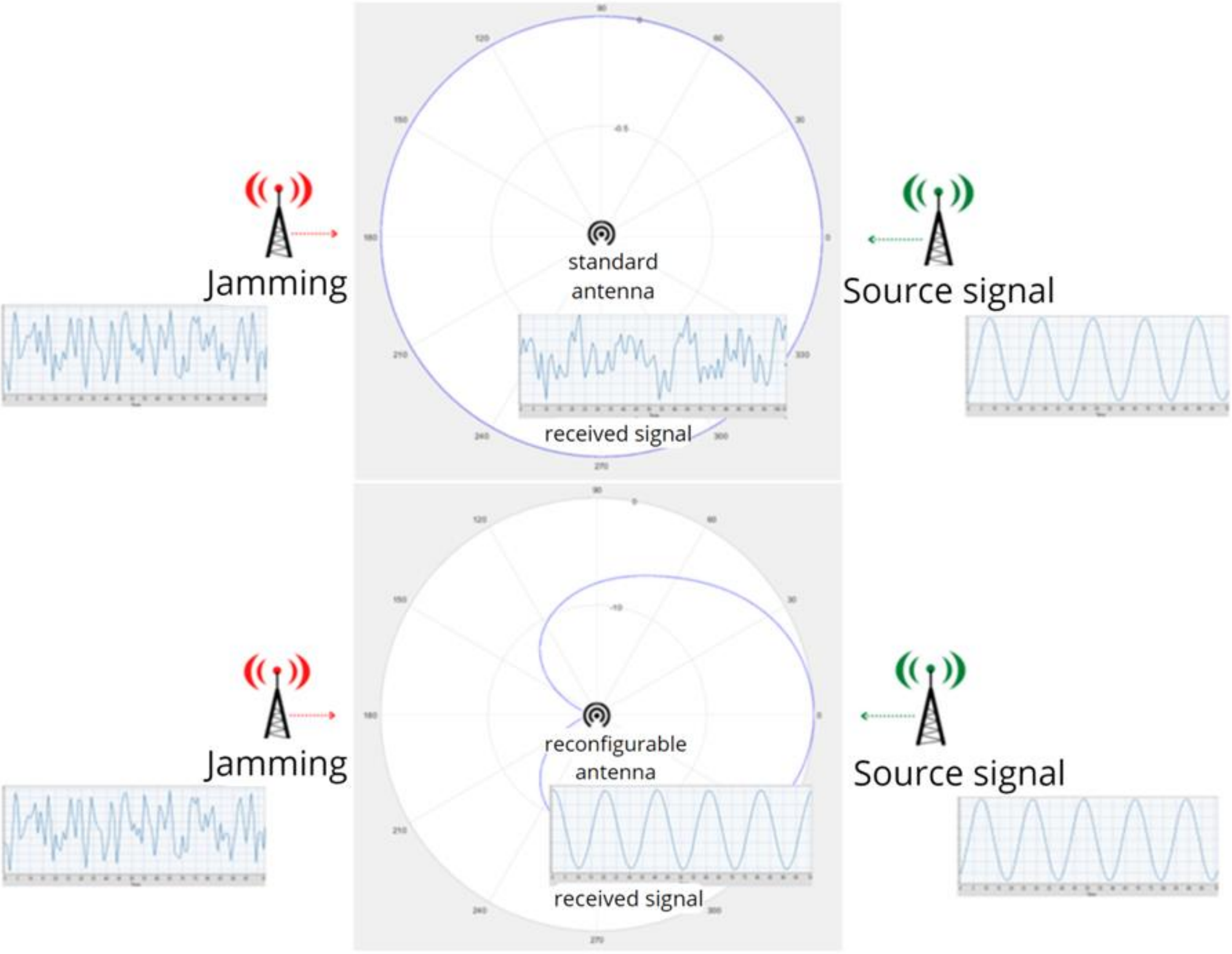


Figure 15: Use of a reconfigurable antenna

The use of a switched beam antenna with reconfigurable radiation pattern will increase the ratio between the desired signal and the interference in the receiver. Additionally, a quick control algorithm should be implemented that allows the solution to operate in real-time systems.

4.4.2 State of the art

Current solutions in this field focus on the use of arrays of radiators to shape the antenna beam together with circuits that allow for digital or analogue beamforming. However, they use expensive systems that consume much more power than in the case of single transmitting and receiving systems. Such systems are characterized by a very high energy consumption and a high price, which excludes them from mobile applications.

Systems for counteracting jamming can be found in the literature. One of the available solutions (Chen Sun, 2004) uses the ESPAR (Electronically Steerable Parasitic Array Radiator) antenna with continuous impedance regulation. The algorithm used to drive the antenna allows for fast beamforming in noisy environments. However, the implementation of the algorithm requires the determination of a loss function as cross-correlation coefficient between the received and expected signals, which can be considered an absolute quality indicator of received signal. The available solution requires to send a reference signal from the transmitter to receiver via cable. This misses the point of wireless communication and cannot be used in practical applications. However, the results of the experiments show that thanks to the use of the ESPAR antenna to counteract jamming attacks, the signal to jamming power ratio in the receiver can be improved. This shows a big potential of this approach.

4.4.3 Innovation step

The proposed solution is the practical use of the switched beam antenna to counteract jamming attacks on wireless communication. The use of an ESPAR antenna (Chen Sun, 2004) with switched impedances for this task, characterized by very low power consumption and low production costs, will be innovative. The antenna, located in the physical layer of wireless communication, will allow to counteract jamming attacks already at the stage of receiving the signal.

The use of this type of antenna requires the adaptation of the algorithm to work with key impedances. The use of other optimization algorithms or implementation of dedicated beamforming algorithms operating in real-time systems will also be considered. This will allow the system to react quickly in the event of potential jamming attack.

In order to make the loss function independent of the cross-correlation coefficient between the received signal and the transmitted signal (removing the wired connection of the receiver with the transmitter, which is redundant in the previous solutions), a new implementation will be prepared. The new function will be based on the signal quality determinants available in wireless communication modules.

Additionally, in the absence of a jamming attack, this approach will allow the antenna radiation pattern to be adapted to changing environmental conditions, improving the overall quality of wireless communication.

It is planned to implement dedicated beamforming algorithms operating in real-time systems. This will allow the system to react quickly in the event of an attack. Additionally, in the absence of a jamming attack, this approach will allow the antenna radiation pattern to be adapted to changing environmental conditions, improving the quality of wireless communication.

4.4.4 Application to use cases

The proposed solution will be applied in UC2, as a component for secure and reliable wireless communication between the shore and an unmanned surface vessel. In unmanned operations, it is crucial to assure the connectivity of the communication for at least the most important sensor readouts as well as commands transmitted from the operator on the shore. The proposed solution will reduce the risk of losing or interrupting the communication between the vessel and the shore. Additionally, it can be used to reduce the possibility of eavesdropping or impersonation by adjusting the requirements to the algorithm.

Furthermore, the described solution can be identified as domain independent, as it is applicable not only in maritime, but also in other areas where reliable communication is needed.

5 Identity Access and Services

5.1 Authentication Management

5.1.1 Overview, motivation

The seamless integration of smart devices into our daily lives is becoming increasingly ubiquitous. Different methods for accessing infrastructure and mobility services as well as connected information systems are part of today's smart cities. The problem is, however, that due to environmental and use case restrictions, no standard authentication and authorization scheme for accessing heterogeneous services can be applied. Nevertheless, secure authentication and authorization to different services are highly sought after.

5.1.2 State of the art

A service from a general point of view is always mapped to a specific resource and sets rules on how to access it. The resource can be accessed by an entity as long as it is authorized or entitled to do so. This entitlement can be redeemed at corresponding validation units to utilize the service and to get access to the resource. Resources can either be available online but also locally. Examples are, e.g., getting access to a vehicle or specific data (e.g., battery data), entering a parking lot, be entitled to charge a vehicle at a charging station or getting access to a specific data stream.

Regarding the topic of how to authorize systems and their users to access different infrastructural and cloud-based services, we distinguish between local and online redemption methodologies. Depending on the use case and the underlying application, different sets of tokens are created. They are managed (creation, storage, and transfer) by a trusted cloud-layer among all participating entities. The cloud-based layer acts as central authentication and authorization unit, responsible for issuing and managing different digital tokens, while locally available devices are used for obtaining and redeeming these tokens at dedicated kiosk units. A client is a personal device owning digital tokens and exchanging them for resources, while the kiosk entity acts as a validation authority and service redemption unit. It receives and verifies digital tokens for checking if a specific user is entitled to access different services.

5.1.3 Innovation step

Token Types for Authentication and Authorization

The high-level objective of our solution is to provide different methods for accessing heterogeneous services. We distinguish between local and online redemption methodologies for authorisation systems and their users to access infrastructure and cloud-based services. Depending on the application and the use case, different sets of tokens for authentication and authorization can be created and are managed by a central trusted cloud-layer. A custom public key infrastructure (PKI) in combination with local authentication and authorization tokens form the basis of the platform’s offline security capabilities and ensure the identity of all communication participants. OAuth 2.0 is integrated into the framework for distributed online authentication and authorization.

Tokens are digital objects that act as unique tickets for authenticating devices as well as entitling them to consume services of a different kind. They are derived by the core server, signed by it, and bound to a specific device (for reference see D3.2 Chapter 6.7).

A-Token

An A-Token is device-bound and responsible for securely authenticating a device when communicating to another device over a local channel, countering also eavesdropping and replay attacks. It can be used in offline mode without an active internet connection.

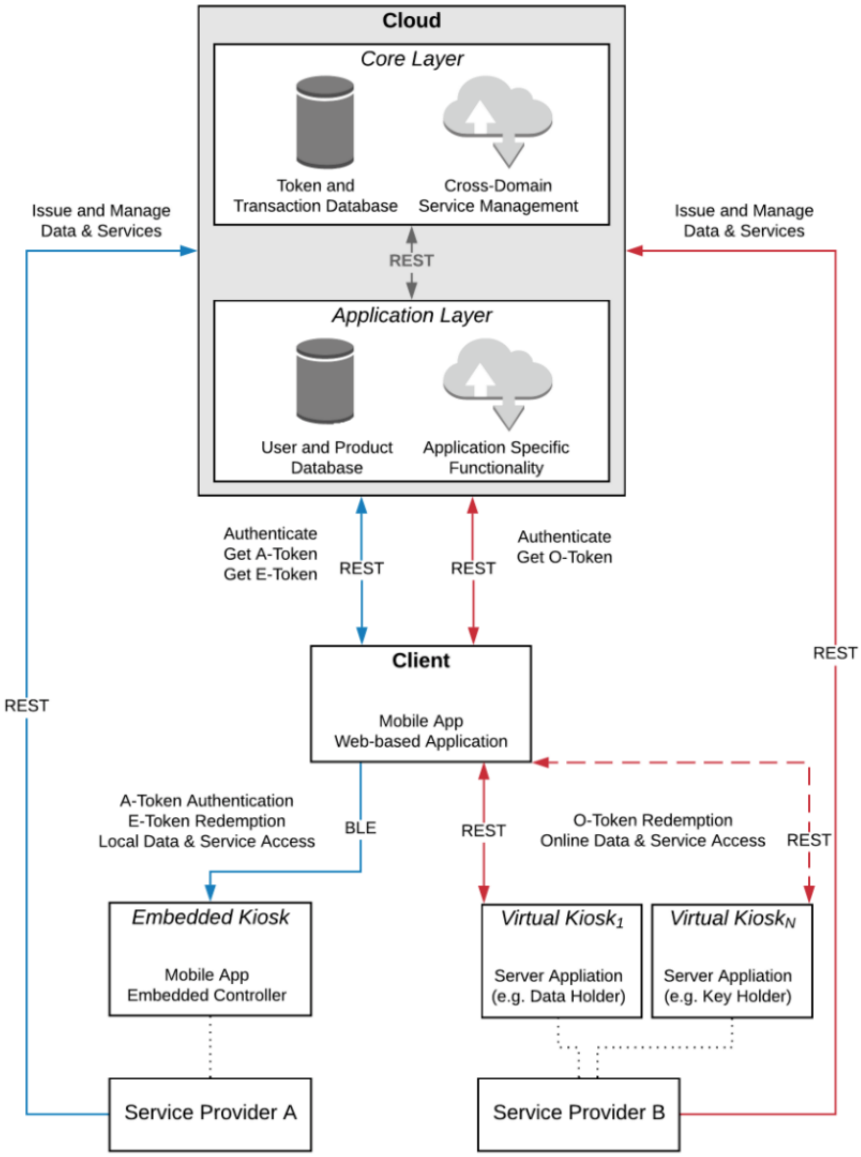


Figure 16: Overview of the distributed platform subdivided into a cloud part as well as client and kiosk devices interacting with each other via BLE and a RESTful interface.

First, an application-specific user-login on the application layer is required. Subsequently, a one-time-ticket is fetched by the application layer and passed to the device who requested it in the first place. It can be redeemed with the device's public key at the core-layer. Finally, the device receives an A-Token as well as an API key for accessing the core layer's REST-based interface, additionally establishing a trusted link between the local device and the cloud layers. Regarding the structure of the A-Token object it is composed of the following elements:

- Unique device-bound identifier
- The public key of the device that requested the token
- Signature of the core server
- Additional attributes for storing application-related datasets
- Validity period

E-Token

An entitlement-token (E-token) represents a service and is issued and signed by the core server. It consists of a service id and an application id to uniquely identify the corresponding service and is linked to an authentication-token

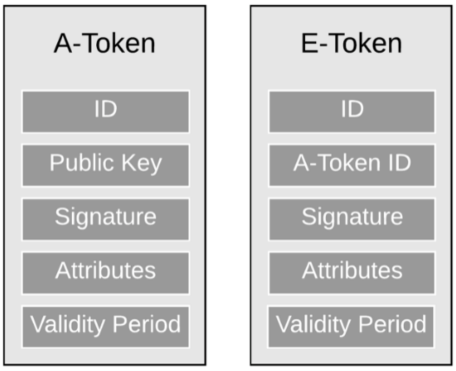


Figure 17: Overview of an authentication-token and an entitlement token for local authentication and authorization

O-Token

Next to the previously mentioned tokens that enable offline communication, the O-Token is used between different online-available parties to exchange data, while a central unit is responsible for global authorization management. The cloud environment was extended with an OAuth 2.0 part for managing the redemption procedure of online services. In this context, the cloud creates a valid O-Token whose purpose is to entitle a specific application to access certain datasets of another entity. The request for receiving the O-Token specifies the grant type that determines which of the OAuth 2.0 authentication flows is used. In our case, the authentication relies on the client credentials flow. Therefore, an additional client-ID, as well as a client-secret, have to be submitted. Last but not least, also an accessTokenUri (specifies where the O-token is being requested from) and the scope (defines which resources are being requested) are included in the request. If the authentication mechanism was successful, the O-token is forwarded from the core layer to the client. Summarised, the structure of the O-Token is defined as follows:

- Scope: Determines the data the receiver of the token is allowed to access
- Validity period
- Value: Functions as the token identifier
- Token Type: Specifies how the token will be used to access the resource. In this case, it is a bearer type token, meaning that access to the resource is only given to the bearer of the token.

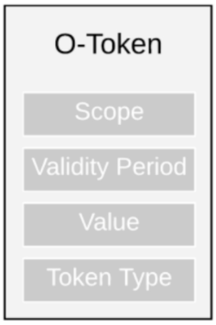


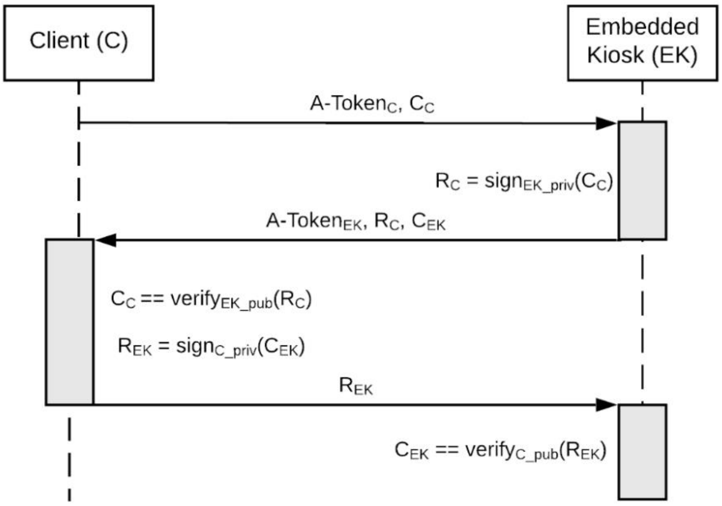
Figure 18: O-token attributes

Local Service Redemption

In case of local service redemption, embedded kiosks are responsible for interacting with client devices for authenticating them, verifying their entitlements with additional interaction with the core server, and in further consequence, authorizing them to access local services. The embedded kiosk is an application running on a mobile device (e.g., supervised parking use case; attendant checks parking permit) or a stationary device embedded into the infrastructure (e.g., parking gate; embedded device checks parking permit). S-Tokens can be distributed directly to client devices and enable local service redemption. Additionally, local redemption always foresees different checks to guarantee the physical presence of a client device. Depending on the application and the use case a connection to the kiosk can be established via BLE or a dedicated HTTP-channel. In both cases, local BLE and HTTP redemption the S-Token is transmitted to the server after passing the authentication step. Consequently, the validity of the token is checked and its signature is verified. If all checks are passed, the redemption of the S-Token is acknowledged by the server and the token, as well as the corresponding entitlement-object, are going to be redeemed. Two different local redemption procedures are going to be mentioned in the following.

Local BLE Redemption

The communication between the client and the redemption unit is established via BLE. Secure authentication is provided with a dedicated challenge-response mechanism. The following figure shows details about the mutual authentication procedure between the client and embedded kiosk.



Client (C)
Embedded Kiosk (EK)
A-Token$_C$, C$_C$
$R_C = sign_{EK\_priv}(C_C)$
A-Token$_{EK}$, R$_C$, C$_{EK}$
$C_C == verify_{EK\_pub}(R_C)$
$R_{EK} = sign_{C\_priv}(C_{EK})$
R$_{EK}$
$C_{EK} == verify_{C\_pub}(R_{EK})$

Figure 19: Mutual challenge response protocol between a client and an embedded kiosk

The redemption of an S-Token involves the interaction of an embedded kiosk. During this event, the client's A-Token and the S-Token that should be redeemed are transferred to the embedded kiosk. For checking the authenticity of the two communication participants a challenge-response protocol is applied between them. First, both A-Tokens (in this example A-TokenC and A-TokenEK) are exchanged, checked for their validity, and verified with the server's public key. Next, the client generates a random number (challenge CC) and challenges the embedded kiosk to sign it with its private key before it is sent back to the client. Furthermore, the received signature is verified against the random number CC. This is done by using the kiosk's public key that is embedded into the previously exchanged A-TokenEK. If this verification was successful, the ownership of A-TokenEK was proven. Before exchanging data, the same method is also applied for A-TokenC. If the mutual challenge-response protocol is passed, the S-token is sent from the client to the embedded kiosk. The kiosk checks its signature and validity period. Last but not least, the redemption of the S-Tokens is triggered by the embedded kiosk by involving the core server via a REST interface. Since this challenge-response protocol is applied to the application layer, it can be used independently of the underlying link-layer protocol (BLE, NFC, etc.), enabling a higher level of modularity and reuse.

Local HTTP Redemption

The local HTTP redemption procedure includes again a client acting as an entitlement holder and a kiosk device acting as a service redemption unit. However, depending on different technology and use case requirements not always a direct communication between these devices is desired or even possible. Therefore, the part of the local kiosk is extended by a dedicated application server. In case the client wants to redeem a specific location-dependent service (e.g., accessing a car) it communicates directly to the application server that is responsible for authenticating the client in the first place. Next, the client submits his S-Token to the server where its signature and validity is checked. Additionally, the client's position data (x and y coordinates) have to be submitted and are checked online against the possible redemption range of the S-Token. If all tests are passed, the server sends a signal to the locally available kiosk device which completes the service redemption by providing the service and the associated resource (e.g., reading car data, opening a gate, unlocking a vehicle, etc.). Compared to the local BLE redemption case, the client has to be online.

5.1.4 Application to use cases

The proposed solution will be applied to UC3 – cloud-features battery management system. Together with AVL and TU Graz the secure wireless end-to-end communication is applied to the setup prepared by the partners. CISC’s solution will provide security and privacy (concerning user data) protection from the sensor to the cloud.

5.2 Role-based Access Control Rules and Security

5.2.1 Overview, motivation

To ensure the confidentiality, integrity and availability of a system's information and services, it is necessary to limit user access and the actions they can perform. However, if the number of users is high and dynamic, the authorization granting and revocation operations can grow, making it difficult to manage. For CPS systems adopting the TRANSACT reference architecture, managing user permissions is a concern due to the large number of services and components deployed at all three tiers (appliance, edge, and cloud).

One of the security strategies to address this concern is Role Based Access Control (RBAC). RBAC allows users to be restricted in the use of the system according to their role. Each role can have a set of permissions or authorizations to access, modify, or manage resources and services. RBAC is one of the key mechanisms widely implemented in cloud environments (Li, 2015), but little explored in the management of permissions for distributed architectures of CPS systems such as the TRANSACT architecture.

Our objective is to define a metamodel and implement a Domain Specific Language (DSL) covering the main concepts enabling the modelling of RBAC for TRANSACT architecture. This DSL enables the specification of business/design level policies to grant/deny access to system resources.

5.2.2 State of the art

Several studies have proposed generic metamodels for RBAC (Salvador Martínez, 2018) (Antonia M. Reina Quintero, 2022) (Nguyen, 2013) (Mouelhiv, 2008) that address the basic concepts of this security strategy. For example, the metamodel proposed by Martinez et al. (Salvador Martínez, 2018) includes the definition of key concepts such as *Policy* which is composed of *Roles* and *Rules*. Basically, it enables the specification of rules that associate roles with permissions (or prohibitions) to perform actions on the elements. However, these generic metamodels should be extended to address the concepts of CPSs based on the TRANSACT architecture and be applied to more specific domains such as UC5. In addition, these generic metamodels commonly address only CRUD operations to define a permission on a resource, but other types of actions must be addressed and included in the definition of the abstract syntax (metamodel).

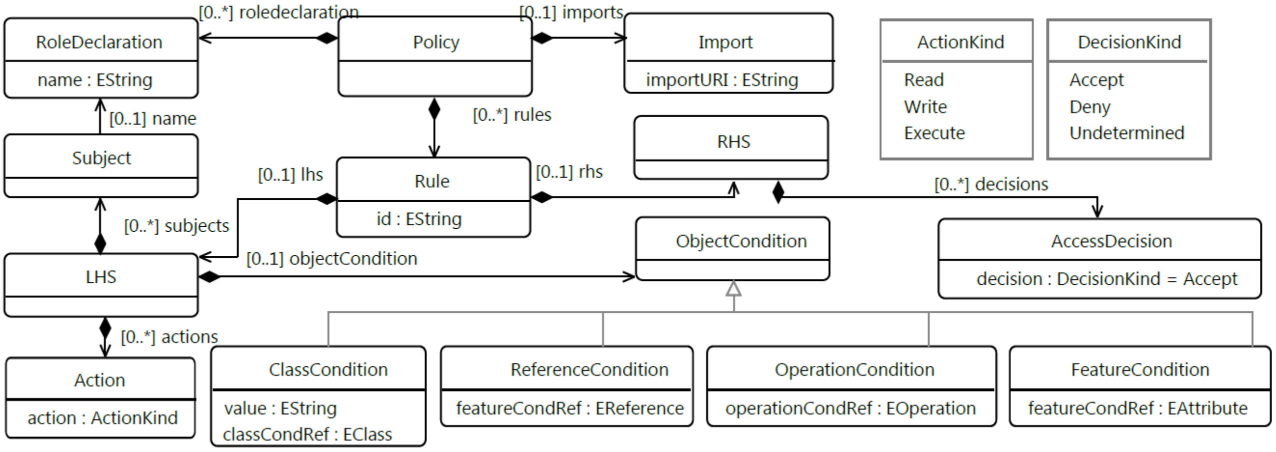


Figure 20: RBAC meta-model (Salvador Martínez, 2018)

Other studies have been focused on designing metamodels and solutions for RBAC applied to different domains and contexts. For instance, Mendling et al. (Mendling, 2004) proposed a metamodel that integrates RBAC and Business Process Execution Language (BPEL) concepts. In this study, BPEL elements are mapped to RBAC elements to build the metamodel that integrates both (RBAC and BPEL). For example, BPEL activities can be mapped to RBAC operations. Guo et al. (Guo, 2010) propose an access control integration framework (based on RBAC) for legacy systems. This framework transforms the control policies of each legacy system into a unified RBAC structure. Zhang and Tian (Zhang, 2010) propose an extended RBAC model for managing access to services that control devices in an IoT system. However, these studies do not address the definition of access control policies involving the services and resources of distributed architectures.

To summarize, RBAC modeling for CPSs based on distributed architectures (such as TRANSACT's architecture) has not been addressed. However, the generic metamodels found in the literature could be extended to define the concrete syntax of a DSL that enables the modeling of role-based policies to manage operations on system resources.

5.2.3 Innovation step

The design of a DSL involves three key aspects: the abstract syntax commonly represented by a metamodel that abstracts domain concepts such as roles, permissions, actions, and system resources; the concrete syntax that defines the notation of the language as graphical, textual, or hybrid; and the semantics of the language to correct the models defined with the DSL.

Our main contributions in this field consist of: (1) the design of a metamodel that defines the RBAC concepts for CPS systems based on the TRANSACT multi-tier architecture, (2) the definition of a hybrid notation that offers textual and graphical forms for the creation of the model that describes the system and the RBAC, and (3) the definition of the semantics that analyzes the correctness of the model created by the user.

Two types of access control rules could be configured with the DSL:

- Business rules these rules involve *Actions* (e.g., querying) on business information such as data collected by system sensors, actuator status and system alarms.
- System infrastructure management rules: these rules involve *Actions* (such as create, update, delete) of infrastructure components such as nodes, software containers, and services.

5.2.4 Application to use cases

The RBAC implementation will be applied to UC5 (Critical wastewater treatment decision support enhanced by distributed, AI enhanced edge and cloud solutions). This industrial use case involves CPSs to perform monitoring and control of wastewater treatment processes in each plant. Various types of sensors are deployed to continuously monitor variables such as pH, temperature, oxygen, conductivity, and other characteristics of the liquids involved in the physicochemical treatments. These collected data allow unexpected events (such as spills) to be identified in real time, and even predicted by AI algorithms. However, several users and roles should have limited access to this sensitive information (sensor data) or even to system management functions (such as starting/stopping a new AI algorithm).

Through the DSL, access control rules can be configured to define permissions on users who fulfill different roles. For example, users who are operators of the biological reactors could manage information from the oxygen and conductivity monitoring probes, but not the infrastructure's computational resource consumption information. In UC5 it is planned to specify such rules involving different types of users interacting with the CPS system deployed in the WWTP.

5.3 Generation of customized views of security-related information

5.3.1 Overview, motivation

The architecture of a CPS involves sensors, monitors, and different types of data sources that produce large volumes of information including monitoring data about the context or physical entities and data about the current state of the system (e.g., infrastructure metrics and QoS). This information collected during CPS operation should be presented to system users according to their role. For example, the information authorized and displayed to a system infrastructure administrator is different than for a user who wants to know the status of a specific process controlled by the CPS.

According to (Bendre, 2016), offering support tools (such as visualizations) for data analysis allows to improve manufacturing processes, production control, increase profits, and improve customer service at lower costs. Visualization is a way to efficiently organize and present information to the user. Using graphs, tables, maps, and other visual elements to display data, the user is enabled to analyze information in real time, understand trends, identify outliers and patterns in the data. Ideally, the visualizations built for users depend on permissions to query system information. Information that is not authorized for a user should not be displayed to them. This allows users to focus on the information of interest to them and ensures the confidentiality of the information. One of the strategies to define these permissions is RBAC (Role Based Access Control) as discussed in Section 5.2.

Therefore, the main objective of this topic is to define a code generator to automatically produce visualizations for users, based on the RBAC model built with the DSL proposed in Section 5.2. These visualizations will show information using graphs about the data collected by the CPS sensors, and data about resource consumption.

5.3.2 State of the art

Data visualization is a technique that consists of transforming and presenting information in a visual context, using graphs or maps to facilitate its understanding. Several tools have been focused on the design and customization of visualizations, but most of them require a high technical knowledge. Other studies based on model-driven engineering (MDE) have focused on reducing that complexity designing languages and generating visualizations automatically.

An evaluation of custom visualization generation capabilities of Security Information and Event Management Systems (SIEM) is presented in (Sönmez, 2018). For each SIEM system evaluated (such as Splunk, Rapid7, and Alien Vault), several aspects are evaluated such as the ability to load custom data files, the ability to form custom searches, and built-in visualization capabilities to display the selected data results. Although these SIEM systems offer several customizable features for displaying data, none of them address the automatic generation of visualizations based on the permissions defined through RBAC.

MDE have been applied in some works to define and generate visualizations in different domains. For example, Hernández et al. (Hernández, 2021) propose a model-based approach to develop data visualizations in the field of structural health monitoring, particularly for bridges. This approach consists of the design of a metamodel, a DSL, and a set of model transformations to generate data visualizations. However, real-time visualizations of monitored data are not supported. Other similar works also focused on the design of DSLs for the generation of visualizations to analyze data are (Ledur, 2015) (Smeltzer, 2018) (Brambilla M. a., 2017). Although the definition and customization of visualizations using different types of charts (such as pie chart, bar chart, and histogram) is covered by some of these DSLs, RBAC concepts for the metamodel or concrete syntax design are not addressed.

5.3.3 Innovation step

Code generation is described as the vertical transition or transformation from high-level abstraction models to low-level artifacts (Brambilla M. a., 2017). A model transformation is a mapping that takes a source model and generates an objective model following transformation rules. There are three types of model transformations: model-to-model (M2M), text-to-model (T2M), and model-to-text (M2T). Commonly a model transformation chain results in the generation of system artifacts (e.g., source code and configuration files).

The main contribution is the design of a code generator to produce the source code for creating custom visualizations. The visualizations built are based on the permissions assigned to each role. The visualizations could be different for each role according to the reading permissions assigned on the system data. For example, a visualization that shows real-time data from a machine's sensors will not be produced for a user who is not authorized. In this way, the confidentiality of the information is guaranteed by preventing unauthorized users from consulting it. Figure 21 shows an overview of the code generator and the artifacts involved. The RBAC model describing user permissions on system resources is the input to the code generator. Transformations (e.g., M2T) are applied to this model to finally produce the code that will display the visualizations. This code must be executed by a data visualization technology or platform such as Grafana.

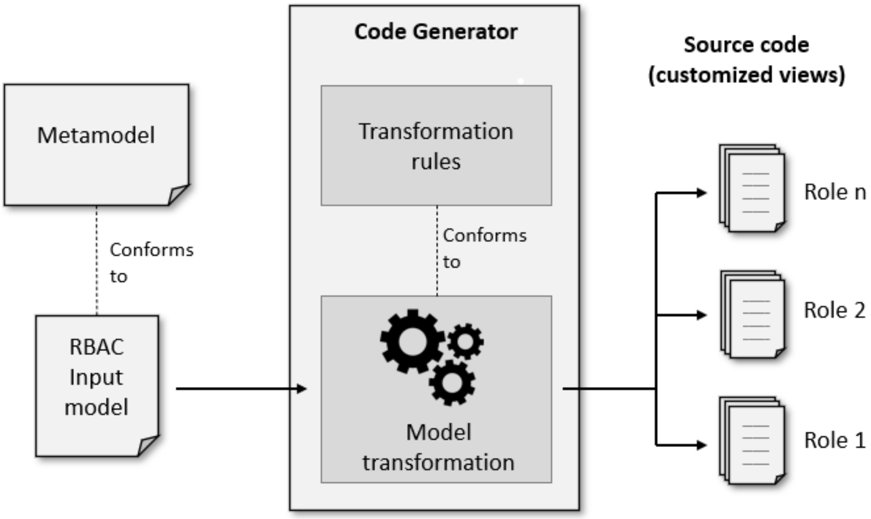


Figure 21: Code generation approach

5.3.4 Application to use cases

The generation of customized views will be applied to UC5. The CPSs implemented in the WWTPs are operated by users with various roles that query and analyze information on different aspects. For example, some technicians might be authorized to query only the physical variables (such as temperature, acidity, conductivity, etc.) in biological reactor A, while other users may be authorized only to query information in the de-sanding and de-greasing station.

In addition to enabling queries of data collected by the CPS sensors, the visualizations can also be configured to query information about resource consumption and system infrastructure status. For example, users with the role of CPS infrastructure administrators in the WWTP could also obtain visualizations to analyze CPU and RAM consumption on edge or cloud nodes.

6 Safety, Security, and performance monitoring Services

6.1 Remote attestation-based Security

6.1.1 Overview, motivation

IoT enabled CPS devices are exposed to a wide variety of rapidly evolving attacks, in particular, runtime attacks, where attackers exploit buffer overflow vulnerabilities or leverage Return Oriented Programming (ROP) to hijack the execution flow of a running program without injecting a new malicious code in the device. This has led researchers to propose different dynamic Remote Attestation approaches to detect compromised devices. Such dynamic Remote Attestation approaches use a complex algorithm to trace the runtime execution flow of IoT devices, introducing high overhead or external hardware to the Prover. However, some attacks still remain undetected. For instance, Data-Oriented Programming (DOP) attacks can compromise variables without deviating the control-flow execution of the running software.

Additionally, existing Remote Attestation schemes do not cover the entire memory of IoT devices, exposing the devices to mobile attacks that can relocate themselves during attestation and exposing devices to be purposefully misconfigured, compromising their integrity without detection. This is especially crucial for multi-service devices. Indeed, the integrity of such devices depends also on the integrity of any attached external peripheral devices. For example, if an adversary successfully alters the configuration of a peripheral temperature sensor to provide an inaccurate representation of the temperature, any internal process that relies on this data, as well as any other system to which this inaccurate data is propagated, may behave in an unexpected way. As a result, to accurately verify the device, all attached peripheral devices must also be verified.

6.1.2 State of the art

Remote Attestation approaches are generally classified into three main categories: software-based, hardware-based and hybrid approaches. Software-based schemes (e.g., SWATT (Seshadri, Perrig, van Doorn, & Khosla, 2004), Pioneer (Seshadri, et al., 2005)) do not make any hardware assumptions and purely rely on the strict execution time of the Remote Attestation protocol.

Despite their advantages, software-based Remote Attestation schemes do not provide strong security guarantees. Hardware-based schemes use a tamper-resistance hardware module as a Trusted Execution Environment (TEE). While hardware-based designs offer strong security guarantees, they are unsuitable for low-cost resource-constrained IoT devices. To provide lightweight secure Remote Attestation protocols, hybrid designs (e.g., SMART (Eldefrawy, Perito, & Tsudik, 2012), TrustLite (Koeberl, Schulz, Varadharajan, & Sadeghi, 2014), TyTan (Brasser, El Mahjoub, Sadeghi, Wachsmann, & Koeberl, 2015)) rely on minimal hardware changes to ensure that the Remote Attestation protocol and associated authentication keys cannot be tampered with.

All these schemes perform attestation on a single device. Collective attestation schemes (e.g., SEDA (Asokan, et al., 2015), SANA, SHeLA (Ambrosin, et al., 2016), PADS (Ambrosin, Conti, Lazzeretti, Rabbani, & Ranise, 2018), PERMANENT (Ankergård & Dragoni, 2021)) aim to provide scalable RA solutions that attest efficiently large-scale IoT networks.

While the Remote Attestation schemes perform only static attestation, dynamic Remote Attestation schemes aim to attest dynamic data memory. C-FLAT (Abera, et al., 2016) is the first dynamic Remote Attestation protocol for resource-constrained devices, focusing on detecting control-flow attacks. C-FLAT relies on software instrumentation to trace the execution of a running software and generates an accumulative single hash value for each execution flow. At the verification phase, the Verifier compares the generated hash value with a set of expected legitimate values to determine whether the device is trustworthy or not. C-FLAT is implemented in a TEE such as TrustZone. However, C-FLAT introduces a high overhead because at runtime each instrumented code instruction is intercepted and redirected to the TrustZone secure world. LO-FAT (Dessouky, et al., 2017) enhances C-FLAT by replacing software instrumentation with a hardware module, implemented on an external FPGA, which intercepts the executed instructions at runtime.

Some recent Remote Attestation protocols in the literature consider the attestation of IoT devices that contain one or more services (also called modules). DIAT (Abera, et al., 2019) aims to perform the attestation of modules in the embedded devices of an autonomous collaborative system. For each pair of interacting IoT modules, DIAT performs control-flow attestation and authenticates the exchanged data between each pair. In this way, DIAT ensures that the data sent from one module to another has not been maliciously changed.

RADIS (Conti, Dushku, & Mancini, 2019) attests a group of interacting services that compose a distributed IoT service. To detect malicious services that impact the behaviour of other legitimate services in the network, RADIS performs the control-flow attestation of the entire distributed service. SARA (Dushku, Rabbani, Conti, Mancini, & Ranise, 2020) aims to attest distributed IoT service communicating by a publish/subscribe scheme. By using logical vector clocks, SARA allows the verifier to construct a historical graph of the occurrence of service interactions and identify the maliciously influenced provers.

Besides Remote Attestation protocols, some works within the field of offloading are of interest. In particular, CloneCloud (Chun, Ihm, Maniatis, Naik, & Patti, 2011) allows a resource-constrained mobile device to offload its execution threads to a clone of itself operating in a virtual machine with more computational capabilities. In the context of remote attestation, it will be more beneficial to gather an accurate clone of the device memory, on which the memory forensics can be performed, rather than replicating the device's functionality. Additionally, certain security guarantees not considered by offloading techniques must be provided by RA designs, as they are intended to be used on potentially malware-infected platforms. Due to these differences in security requirements and their purpose, the works within the field of offloading are not directly applicable.

Table 2: State of the art-work Summary.

Scheme	Static Memory	RAM	Peripheral	Verification	Attestation
SWAT, Pioneer	Yes	No	No	Program checksum	On-demand
SMART, TrustLite, TyTan	Yes	No	No	Program checksum	On-demand
C-FLAT, LO-FAT	No	Yes*	No	Control flow integrity (CFI)	On-demand
DIAT	Yes	Yes*	No	Program checksum & CFI	On-demand
RADIS	No	Yes*	No	Program checksum & CFI	On-demand
CloneCloud	No	Yes	No	_	_

6.1.3 Innovation step

ERAMO protocol (Edlira Dushku, 2022) consists of three main phases: (1) Setup phase, (2) Attestation phase, and (3) Verification phase.

1. Setup Phase: A network operator guarantees the secure bootstrap of the software deployed on each Prover. Considering the limited capabilities of Provers, the Verifier and the Prover establish a shared symmetric attestation Message Authentication Code (MAC) key k. To prevent untrusted parties from using Prover's key, the shared attestation key k is stored in a

hardware-protected memory. Alternatively, a Prover can establish a secure communication channel with the Verifier by possessing an asymmetric key-pair (pk, sk) and knowing the Verifier's public key. Note that the key management details are out of the scope of this text. The protocol description is independent of the key management; thus, the symmetric key usage can be easily replaced by an asymmetric key pair. For simplicity, preserving our work's generality, it is assumed that the Prover and the Verifier share a symmetric key k.

2. Attestation Phase: Figure 22 illustrates the protocol. To initiate the attestation, the Verifier generates a nonce N and sends it to the Prover (Step 1). The Prover then relinquishes control to the Remote Attestation protocol residing in the hardware-protected component. The Prover's Remote Attestation protocol reads the device memory contents m (Step 2) and computes a hash h = hash(m). Next, the Prover concatenates the computed hash h with the received nonce N and authenticates it by computing a keyed Hash Message Authentication Code (HMAC) over the obtained result s = HMAC(k, (h||N)). Finally, the memory m and HMAC s are transmitted to the Verifier (Step 3), which checks whether it corresponds to the transmitted data. The transmission may be split into smaller chunks, e.g., by authenticating individually memory blocks or regions. In that case, integrity, authenticity, and temporal freshness must be ensured for each transmitted memory chunk, e.g., by adding a unique extra byte for each chunk or securely generating a pseudo-random number inside the Prover.

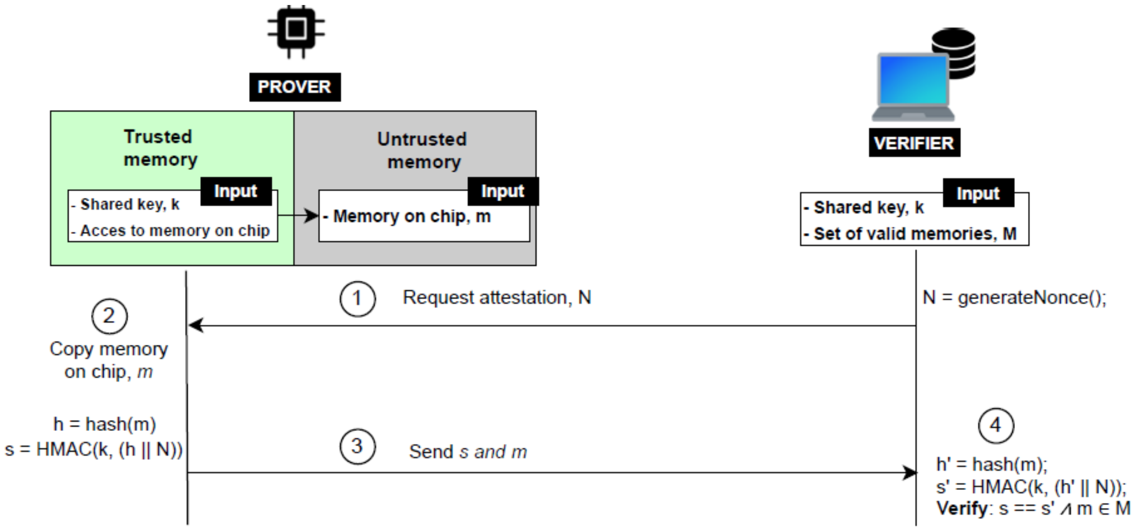


Figure 22: ERAMO protocol.

3. Verification Phase: The verification phase starts when the Verifier receives an attestation response from the Prover. By using the shared attestation key k, the Verifier checks the authenticity and integrity of the attestation result (Step 4). Assuming that the Verifier knows all valid combinations of memory M, the Verifier has the ability to determine whether a given memory m is in the set M. A powerful Verifier which is able to perform advanced memory forensics analysis can use the offloaded dynamic memory contents to provide a detailed attestation and precisely determine the Prover's integrity.

4. Attested Device Memory: Figure 23 shows the attested memory regions verified by ERAMO protocol for a device with a flash memory and a memory-mapped peripheral region. A certain portion of the flash region allocates data memory, whereas the memory-mapped peripheral

region contains both readable and write-only registers. All readable memory can be attested apart from the secure memory allocated to the trusted component performing attestation.

Furthermore, if a region of the flash/EEPROM is used for data, such as calibration values or network information, this region may also be verified through offloading. This data may change during runtime and may depend on the electrical characteristics of the specific device, and thus may not be verifiable through hashing. Assuming that the Verifier has some notion of what differentiates legitimate values of this region, the integrity verification of this region is possible through offloading.

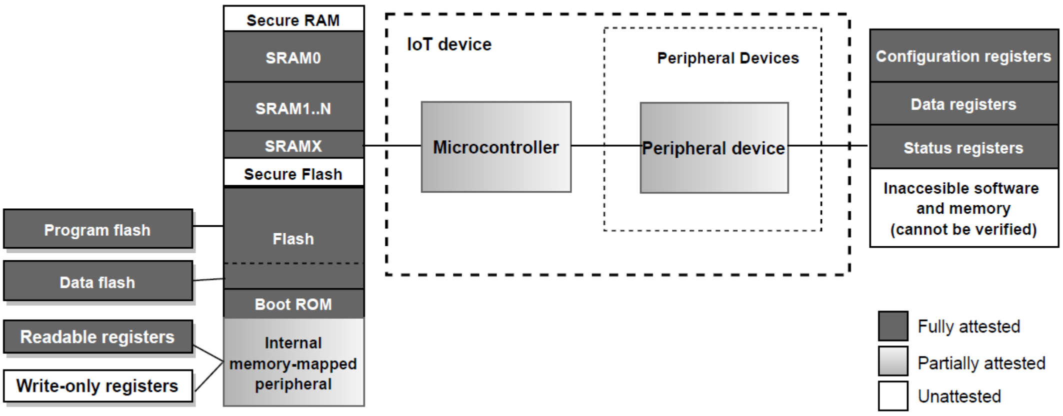


Figure 23 : Memory regions attested by the ERAMO protocol

6.1.4 Validation results

The efficiency of ERAMO highly depends on the choice of hardware. Memory transmissions rely on the choice of communication and its transmission speed. The time required for authentication depends on Prover's computational capabilities and available hardware to assist with the process.

The runtime measurements of the procedure were measured on the LPC55S69 running at 150 MHz. To simplify the connection to the Verifier, a serial connection was established using the on-chip UART configured to a baud rate of 806,400. The LPC55S69 hash engine was used to compute the necessary authentication using SHA-256 for hashing and the HMAC. The procedure was tested on different memory sizes, increasing in steps of 1 KB. The memory offloaded was the 240 KB of non-secure RAM associated with the IoT application.

The time used for the offloading procedure is proportional to the offloaded memory size, as shown in Figure 24. The Time used to authenticate memory is shown in Figure 25.

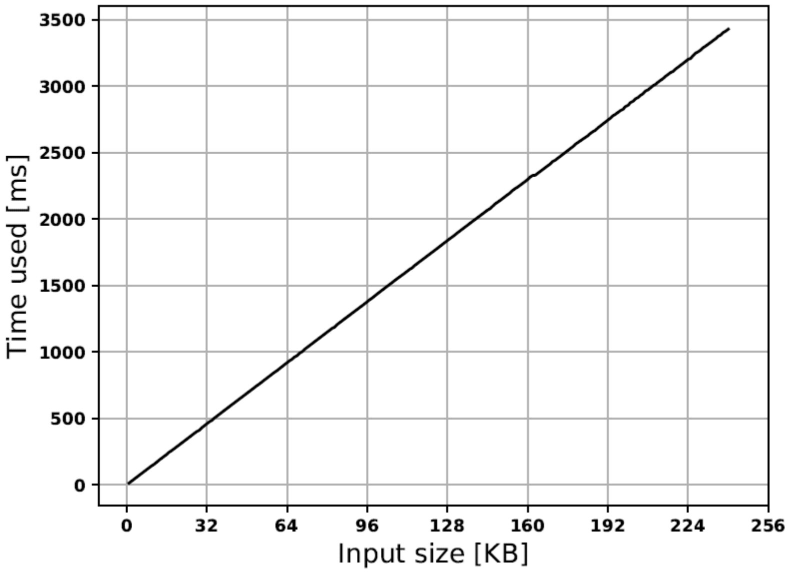


Figure 24 Time used to transmit memory

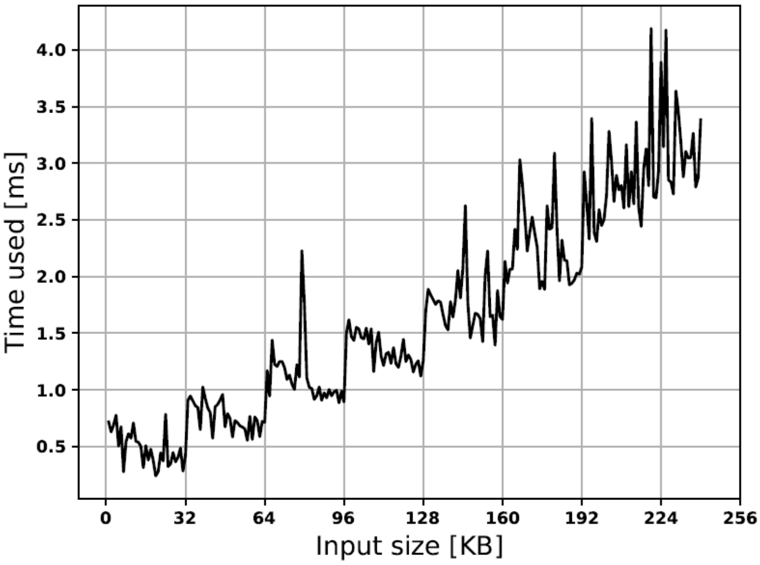


Figure 25 Time used to authenticate memory.

6.1.5 Application to use cases

ERAMO is a novel Remote Attestation protocol that relies on a memory offloading approach to verify the Prover’s integrity. ERAMO verifies more dynamic memory areas (such as the internal and external peripherals) that are not covered by existing Remote Attestation schemes. This protocol uses memory offloading to shift the attestation from low-end devices to nearby devices with more powerful computational capabilities. This approach is aligned with and leverages the emerging Edge computing paradigm, which extends the Cloud by bringing computational resources next to IoT devices. UC5, “Critical wastewater treatment decision support enhanced by distributed, AI-enhanced edge and cloud solutions,” is an industrial use case. ERAMO can help identify rare elements, events, or observations of the parameters of the sensors that arouse suspicion by significantly differing from the usual or daily behaviour. This solution can be applicable in various use case of TRANSACT Project, where an attacker that discovers and exploits a program vulnerability such as a buffer overflow. By leveraging the Return-Oriented Programming (ROP) technique, the attacker alters at runtime the execution flow of legitimate code already loaded on the device's memory to produce a malicious operation. Additionally, the attacker can use the Data-Oriented Programming (DOP) technique to compromise variables' values and manipulate data pointers. Such attacks are common in IoT as resource-constrained IoT devices are exposed to many well-known vulnerabilities, e.g., format string and integer overflow.

6.2 Security, privacy and trust related solutions for remote driving operation

6.2.1 Overview, motivation

The analysis clarified in D3.2, Section 9.1, revealed serious risks, problems and challenges for security, privacy and trust in the remote driving operation. There are huge number of potential threats against the safety of the operation, and therefore a number of security, privacy and trust requirements have been identified (D3.2). The referred challenges and requirements, the targeted concepts for security, privacy and trust are quite much related to architectural patterns, e.g., to accountability, identity and access control, and data confidentiality/integrity/availability. However, the complexity and dynamicity of the urban mobility environment set some specific challenges which are investigated here. The focused research challenge can be described as follows: when system has unexpected events during remote driving operation in urban traffic context, it is very essential to know what the situation was just before such events. For example, who has been in charge of the remote driving, what interactions have happened between the vehicle and remote driver, what other vehicles and road users have been nearby, what information has been provided by traffic infrastructures (e.g., traffic lights, traffic cameras), what were the positions of the entities, etc.

Here, the research challenge has been divided to two parts: identification and access control, and traceability. The identification refers here to the secure identification of the physical entities, service providers, users and owners in the remote driving ecosystem. Access control refers to the capabilities of the owners to control the use of their resources by giving access rights to the other users. The traceability refers to the capabilities to monitor events/data from multiple resources owned by different stakeholders in reliable way.

6.2.2 State of the art

The traditional perimeter-based network security model has serious risks to the assets of an enterprise, because an attacker may in one way or another gain access to the enterprise system. The likelihood of attackers getting access to the systems has increased, as remote work seems to increase the risks for security threats, phishing of credentials, and therefore the likelihood of a malicious user being able to access the resources in the enterprise systems. The ZeroTrust security models have been developed to contribute towards solving these problems and focus on resource protection and the premise that trust is never granted implicitly but must be continually evaluated (Rose S., 2020). When speaking about safety sensitive cases, like remote driving of autonomous vehicles, it is obvious that the traditional perimeter-based network security model is not enough, but a ZeroTrust type of security model needs to be applied instead. The traditional schemes are not enough for the remote driving case, because of the need to ensure trust relationships of multiple persons, organisations and physical assets simultaneously and control access to the related monitoring and controlling data streams.

The World Wide Web Consortium (W3C) has created the Verifiable Credentials Data Model 1.0 specification, which is approved as a full W3C standard in Sep 2019. The specification applies self-sovereign identities, also called decentralized identifiers, as the basis for the solution ((W3C), World Wide Web Consortium, 2021), (Sovrin, 2020), (Kronfellner B., 2021).

The system works so that a *holder* (person, item, service, etc.) obtains a decentralized identifier (DID) together with its public key from a reliable provider, who also stores it to some type of verifiable data registry, which can be for example, blockchain/distributed ledger, a distributed database or any other sufficiently trusted publicly accessible utility. After that, the holder requests *verifiable credentials* from various *issuers* who, after determining that the credentials can be granted, use their private key to digitally sign the credential (and any other cryptographic material needed to verify the issuer's credentials), and issues it to the holder to store in his/her/its digital wallet. Note that to preserve privacy, this issuance process does not need to involve any interaction with a verifiable data registry—in other words, no personal data needs to be written to a blockchain or third-party data repository. The process can be fully confidential between the issuer and holder. Later, when the holder needs to gain access to some resource controlled by a *verifier*, the verifier requests digital proof of one or more credentials from the holder. If the holder consents, the holder's wallet generates and returns the proofs to the verifier. Since the proofs contain the issuer's DID, the verifier can use it to read the issuer's public key and other cryptographic data from the verifiable data registry. In the final step, the verifier uses the issuer's public key to verify that the proofs are valid and that the digital credential has not been tampered with (Trust over IP Foundation, 2021). Because of the safety sensitive nature of remote driving, the confidence of the involved stakeholders and users could benefit from such a digital trust ecosystem so that the control concept could be acceptable in the public urban traffic system.

The other challenge is related to the ability to monitor events and data from multiple resources owned by different stakeholders of the traffic ecosystem in a reliable way. Because of the remote driving happens in an urban traffic environment, it is obvious that trust in the monitored trace is very important, including from the point of view of authorities. When applying the W3C approach for digital trust, application of blockchain/distributed ledger technologies also for tracing provides a possible approach, and these are here applied for the remote driving case. Distributed ledger technology (DLT) refers to storage, distribution and exchange (share) data between the users of private or public distributed computer networks located in multiple sites (Liu X, 2020). One example of DLT is blockchain, which is the underlying technology of Bitcoin (Nakamoto, 2009). Blockchain is a linked-list type of data structure, which is updateable only via consensus among a majority of the existing peers in the network and thus there is not a single CA controlling the ledger. Each block contains a set of transactions and their hash, with link to the previous block hash. Only after successful consensus, a new block can be added into the chain. Another type of DLT is, e.g., the directed acyclic graph (DAG), where each transaction is represented as a node that is linked to one or several other transactions. The links are directed so that they point from earlier transactions to newer ones without allowing loops (Liu X, 2020). The transactions provide validation for each other, but a transaction cannot validate itself. A new transaction has to validate one or more previous transactions to join the DAG. Every new transaction refers to its parent transactions, signs their hashes, and includes the hashes in the new transaction. One essential difference compared with blockchain is that DAG does not need miners, which makes it cheaper (no mining fee), more rapid and scalable. This makes DAG quite interesting technology for the CPS, which has a large number of transactions that need to be almost free to be realistic. An example of DAG application is IOTA (Liu X, 2020), who call their distributed ledger the Tangle. In IOTA, newer transactions validate one or more earlier transactions, sign their hashes, and include the hashes in the new transaction. The tangle uses Winternitz signatures, which are much faster than elliptic curve cryptography (ECC) applied in Blockchain (IOTA Foundation, 2022). Tangle applies Kerl realizing SHA-3, based on ternary operations, which is more secure than the crypto technologies applied in Blockchains (Green, 2018). However, current realizations of IOTA also support ECC-based signatures and binary (vs. ternary) operations (Cech, 2020).

6.2.3 Innovation step

The key elements of the security, privacy and trust related solutions for remote driving operation are depicted in Figure 26by dividing them conceptually to Trust, Credentials, Control Data, and Trust Storage levels. The Trust level is related to the relationships of people with the organizations and physical assets (trust entities). For example, in the remote driving case, several trust relationships between stakeholders are needed, such as between the autonomous vehicle owner and the autonomous vehicle, between the autonomous vehicle owner and the remote driving company, between the remote driving company and the remote driver, and between the remote driver and authorities (e.g., drivers' license). The Credentials level is related to the digital identities of trust entities, means to give credentials from the issuers to holders, storing the credentials to wallets, and checking the credential proofs with verifier(s). The Control Data level is related to exchanging control data between the entities in an end-to-end manner in a secure way. The control data can be, e.g., credentials, security keys or other cryptographical material, or meta information on the data stream related to the real data flow between the trust entities required to be known by the other parties of communication. The Trust Storage level is related to storing the transactions related to, e.g., critical trust relationships between trust entities, smart contracts, verifiable credentials, and other security, privacy and safety related critical events (traces monitored from the system) so that they cannot be changed after they have been verified and added to the distributed ledger.

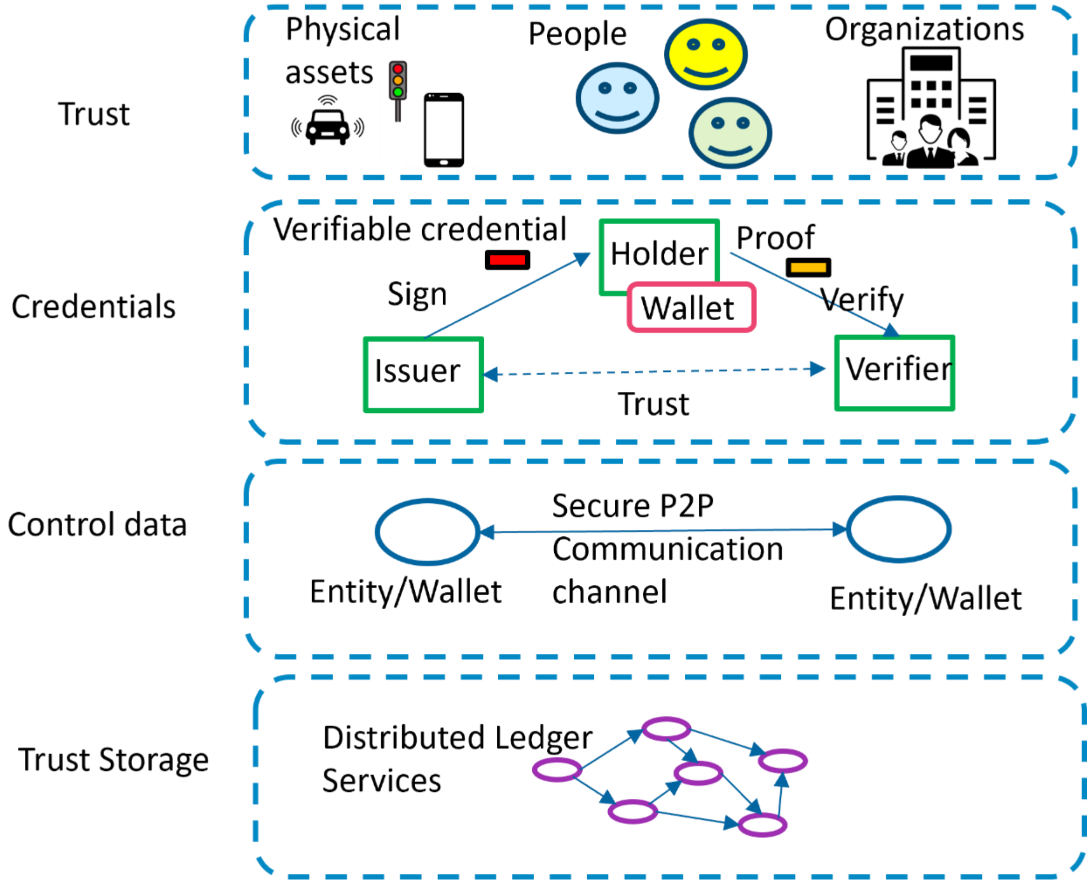


Figure 26. The security, privacy and trust related conceptual solutions (SPT concept) for the remote driving case.

In this research, the referred conceptual solutions have been experimentally developed by relying on the verifiable credentials and digital identities. In the solution the PKI is combined with a decentralized approach using decentralized identifiers (DIDs) and verifiable credentials (VCs). IOTA-based distributed ledger is applied for traceability. *VTT has developed a component called a Trust monitor, also used here in the role of supervisor (CPS Trust@vtt), which applies referred technologies in order to study the operation of the SPT concept.*

First, the credentials for the remote driving endpoints and stakeholders are checked. If all the credentials are ok, then permission for remote driving is given for the autonomous vehicle (e.g., AUNE@vtt) and the specific remote driver (e.g., remote-driver-A@fleetonomy) to start the remote driving session execution in end-to-end manner. The security parameters of the E2E remote driving session are exchanged via a secure IOTA channel. When some critical events happen, e.g., the driving mode changes from autonomous to remote-operated, or a safety button is pressed in the vehicle, or any credentials have changed (including startup), these events (with time and locations) are traced to the trust monitor (CPSHub Trust@vtt) which stores the events to the IOTA Tangle. When some unexpected situations happen in urban traffic, then the IOTA tangle can be applied to study the preceding situations related to the remote driving operation.

6.2.4 Validation results

The 1st validation of the SPT concept solutions is carried out with simulated remote controllable toy vehicle. The procedure for the startup, remote driving session and critical events are executed by means of simulation. The results of the validation will be described in the final deliverable.

6.2.5 Application to use cases

It is estimated that the developed solutions could be applied in the remote driving operation (UC1). However, it is seen that some development steps are needed, e.g., to adjust the interfaces so that the real vehicle and remote driving center could apply the solutions.

7 Cloud Security

7.1 Secure cloud-based infrastructure

7.1.1 Overview, motivation

Cloud applications are frequently targeted by criminals as these attacks can be easily automated. However, security of cloud-based applications is more than protection against attackers, and also covers potential accidental loss of data, unintentional misuse, etc. It always comes down to protection of information of all kinds, such as personal information, IP, or financial information. In the medical use case (i.e., UC4), this could be personal health information (PHI), such as medical images of patients, which should of course be considered as highly sensitive information.

Within the TRANSACT project, security of the cloud applications is of utmost importance, as we are dealing with safety-critical systems. An alteration of data could result in significant harm. An example, again from the medical use case, would the alteration of data from a patient, or the mix-up of data from different patients. It is easy to understand that this could result in a dangerous situation, as physicians could take the wrong decision for a specific patient.

There are numerous risks for cloud-based applications, related to technical vulnerabilities. The Open Web Application Security Project (OWASP) is a worldwide non-profit organization that periodically publishes a list of the top ten most critical web application security risks. This is important and helps people to focus their attention and efforts. The figure below shows the top ten security risks that was published in 2017 and 2021.

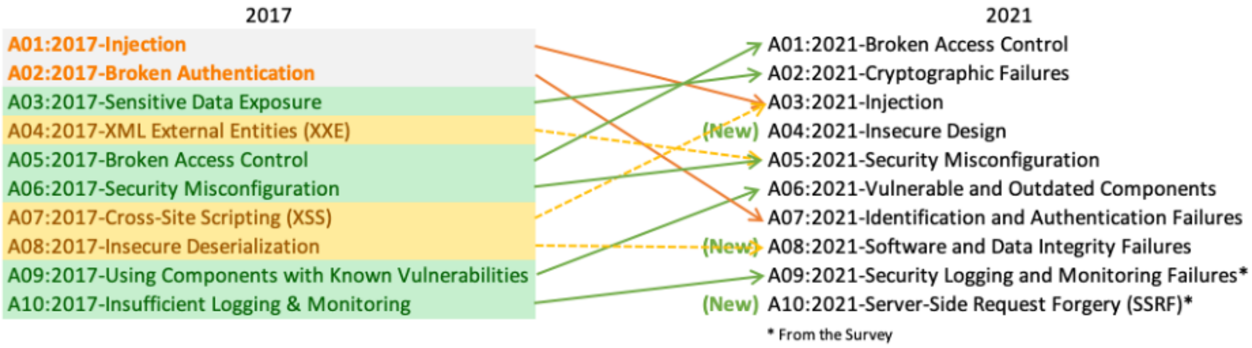

2017
A01:2017-Injection
A02:2017-Broken Authentication
A03:2017-Sensitive Data Exposure
A04:2017-XML External Entities (XXE)
A05:2017-Broken Access Control
A06:2017-Security Misconfiguration
A07:2017-Cross-Site Scripting (XSS)
A08:2017-Insecure Deserialization
A09:2017-Using Components with Known Vulnerabilities
A10:2017-Insufficient Logging & Monitoring
2021
A01:2021-Broken Access Control
A02:2021-Cryptographic Failures
A03:2021-Injection
(New) A04:2021-Insecure Design
A05:2021-Security Misconfiguration
A06:2021-Vulnerable and Outdated Components
A07:2021-Identification and Authentication Failures
(New) A08:2021-Software and Data Integrity Failures
A09:2021-Security Logging and Monitoring Failures*
(New) A10:2021-Server-Side Request Forgery (SSRF)*
* From the Survey

Figure 27: Top ten security risks for web applications in 2017 and 2021 according to the OWASP (https://owasp.org/www-project-top-ten/).

Here is a brief overview of the top ten security risks:

Broken access control: Access control are important to prevent that user can act outside of their intended permissions. Failures typically lead to unauthorized information disclosure, modification, or destruction of all data.

Cryptographic failures: Often, there is a lack of data encryption for sensitive or personal information. Alternatively, the cryptographic algorithms or protocols that are used might be old or weak.

Injection: Injection flaws allow an attacker to “inject” data into a system, which can then enable the attacker to execute commands or access data without proper authorization. SQL is commonly targeted by such injections.

Insecure design: It is important to differentiate between design flaws and implementation defects for a reason, they have different root causes and remediation. A secure design can still have implementation defects leading to vulnerabilities that may be exploited. An insecure design cannot be fixed by a perfect implementation as by definition, needed security controls were never created to defend against specific attacks.

Security misconfigurations: Many server-side issues are due to misconfigurations. These can vary from default accounts being left unchanged through to unprotected files and directories.

Vulnerable and outdated components: Using components that are outdated, unsupported or vulnerable is an important risk. Attackers may detect unpatched components, and target these.

Identification and authentication failures: These failures refer to web applications that use default, weak or well-known accounts/passwords, or that do not prevent attackers from performing brute-force attacks (e.g., dictionary attacks).

Software and data integrity failures: Software and data integrity failures relate to web applications that do not protect against integrity violations. An example of this is where an application relies upon plugins or modules from untrusted sources. An insecure pipeline can introduce the potential for unauthorized access, malicious code, or system compromise.

Security logging and monitoring failures: Sufficient monitoring and logging is fundamental to detect security breaches, and to react to these in an adequate manner.

Server-side request forgery (SSRF): SSRF flaws occur whenever a web application is fetching a remote resource without validating the user-supplied URL. It allows an attacker to coerce the application to send a crafted request to an unexpected destination.

This overview shows that there are many aspects to take into account when developing secure cloud applications. There will always be residual risks, but those should be reduced to the minimum. Typically approaches to do this are described in the next section.

7.1.2 State of the art

There are many ways to prevent and mitigate the above-mentioned security risks. Some state-of-the art concepts for (the development of) secure cloud-based applications are highlighted below:

Security-by-design: As already mentioned, security should already be taken into account during the design phase. For example, account lockout after a number of failed login attempts could be implemented to increase an application’s resilience against attacks.

Security-by-default: The default configuration settings in a product should be the most secure.

Security testing: It is important to test cloud-based applications from security point-of-view, in addition to functional tests. This could be static or dynamic code analysis. Static code review refers to analysing the source code for vulnerabilities without running the code, while dynamic analyses are performed while executing the code. Related to this, external penetration testing could be considered to detect vulnerabilities.

Encryption: Data encryption with strong and up-to-date algorithms should be used to protect important data, both “in transit” and “at rest”.

Access control + role management: Effective account management principles such as strong password requirements and 2FA are very important. In addition, every user should be given as little privileges as possible for them to get what they need from the system (i.e., minimal privileges principle).

Logging / monitoring: Logging is important, not only for recording that suspicious activity is taking place, but also to analyse possible incidents or data breaches.

7.1.3 Innovation step

A typical scenario to use cloud applications is to offload devices. For example, by sending demanding algorithms or AI tasks to the cloud, one can reduce the requirements on the device level. This involves sending data from the device to the cloud, and depending on the use case, this could be (sensitive) personal information. Pseudonymisation techniques could be applied (on the device or edge) to limit the associated risks, and this pseudonymisation could be reverted when the results are sent from the cloud to the edge/device. This workflow is depicted in the figure below.

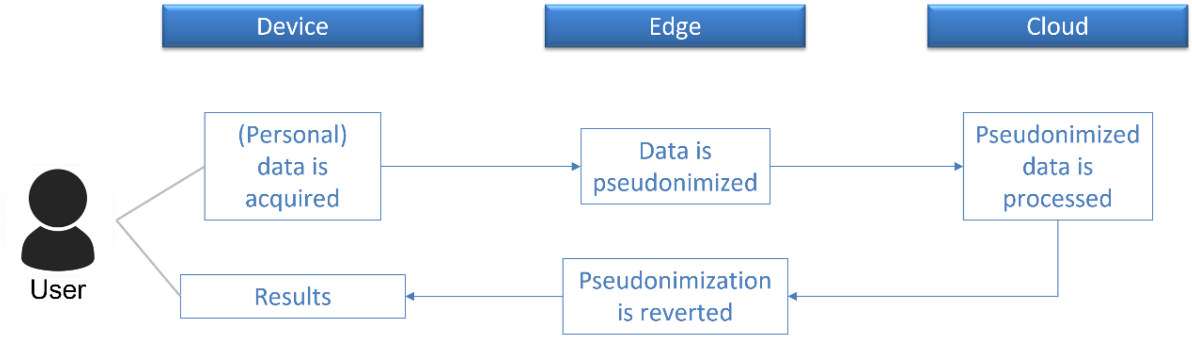

Device
Edge
Cloud
User
(Personal) data is acquired
Data is pseudonimized
Pseudonimized data is processed
Pseudonimization is reverted
Results

Figure 28: Diagram illustrating the use of pseudonymisation when sending data to the cloud

There is however another scenario, where the cloud is not only used to automatically process data, and to send the results back to the edge/device. Some cloud applications have a front-end that allows the user to analyse the data and/or the results. Pseudonymisation of personal data could also be applied in this second scenario, but this results in an important new challenge. It is often important to know which data belongs to which person, as is the case in the medical use case. Let's take the example of a web-based application that analyses patient data, and that has a front-end to show the results to a physician. The data could be pseudonymized when it leaves the hospital, and certain parameters such as the patient's age and/or gender could be maintained, to help the physician in linking the provided results to the correct patient. Age and gender may however not be sufficient, and more parameters may be needed to correctly identify a specific patient. Adding more parameters however results in a weaker pseudonymisation and increased privacy risks.

The solution that we propose to balance the two conflicting risks (data mix-up vs privacy/security risk) is depicted in the diagram below (Figure 29: Diagram illustrating decryption of data in the client's web browser). The personal data gets pseudonymised when sent to the cloud, and highly sensitive information such as the patient’s name gets encrypted. The user could add the encryption key locally to his browser (not to the database of the cloud application), and the highly sensitive information will be decrypted only on the client side. In this way, the risk of not being able to associate the data with the correct patient is eliminated, while the data stored in the cloud is encrypted/pseudonymised.

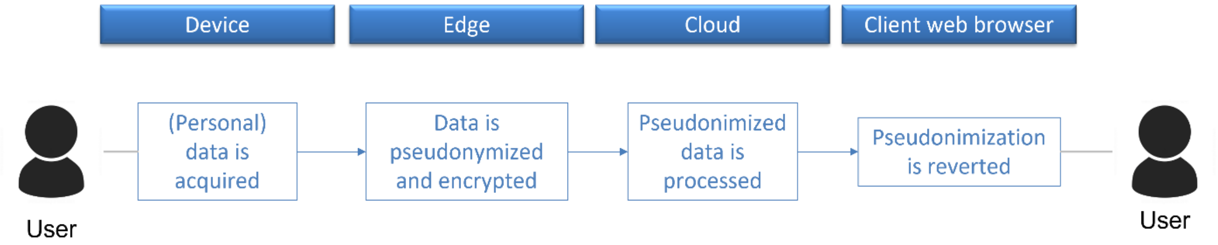

Device
Edge
Cloud
Client web browser
User
(Personal) data is acquired
Data is pseudonymized and encrypted
Pseudonimized data is processed
Pseudonimization is reverted
User

Figure 29: Diagram illustrating decryption of data in the client's web browser

7.1.4 Application to use cases

The concept describe above is highly relevant to the medical use case (UC4). The processing and analysis of medical images, using for example AI algorithms, could be efficiently performed in the cloud. Dedicated cloud-based applications exist in which the results of such analysis are visualised to the user. The FEops HEARTguide platform is such an example, and this platform supports physicians to plan structural heart interventions based on the CT images. The CT images contain several tags in which personal information is included, such as the patient's name. These tags can be easily pseudonymised during the data transfer process, but as mentioned above, it is of course important that the physician links the images and the analysis results in the web application to the correct patient.

7.2 Cloud security posture management tool

7.2.1 Overview, motivation

Over 90% of enterprises utilize a hybrid, multi-cloud strategy in their operations. This is because cloud environments provide benefits over on-premise hosting that include flexibility, a reduced need for scarce resources, improved support, and in some respects, better security. However, the responsibility for providing security in these environments is shared between the cloud provider and the entity provisioning its resources. Of all security concerns related to cloud usage, misconfiguration is the leading cause of data breaches and from our research, the most common source of major cloud security incidents. Gartner predicts that “through 2025, 90% of the organizations that fail to control public cloud use will inappropriately share sensitive data.” Cloud vendors provide tools that are capable of spotting misconfigurations, but to be effective, they must be configured and managed by a skilled expert. The scarcity of cloud security skills makes products hard to maintain and can lead to difficulty in interpreting their outputs. Added pressure also comes from regulators who often request evidence that security controls governing data in the cloud are in-place and sufficient. Cloud security risk is often managed by regular audits.

7.2.2 State of the art

Figure 30 depicts a multi-cloud concept for a Cloud Security Posture Management (CSPM) solution which utilizes both machine learning and heuristic rules to identify weaknesses such as configuration mistakes and security best practice anti-patterns in the deployed cloud infrastructure.

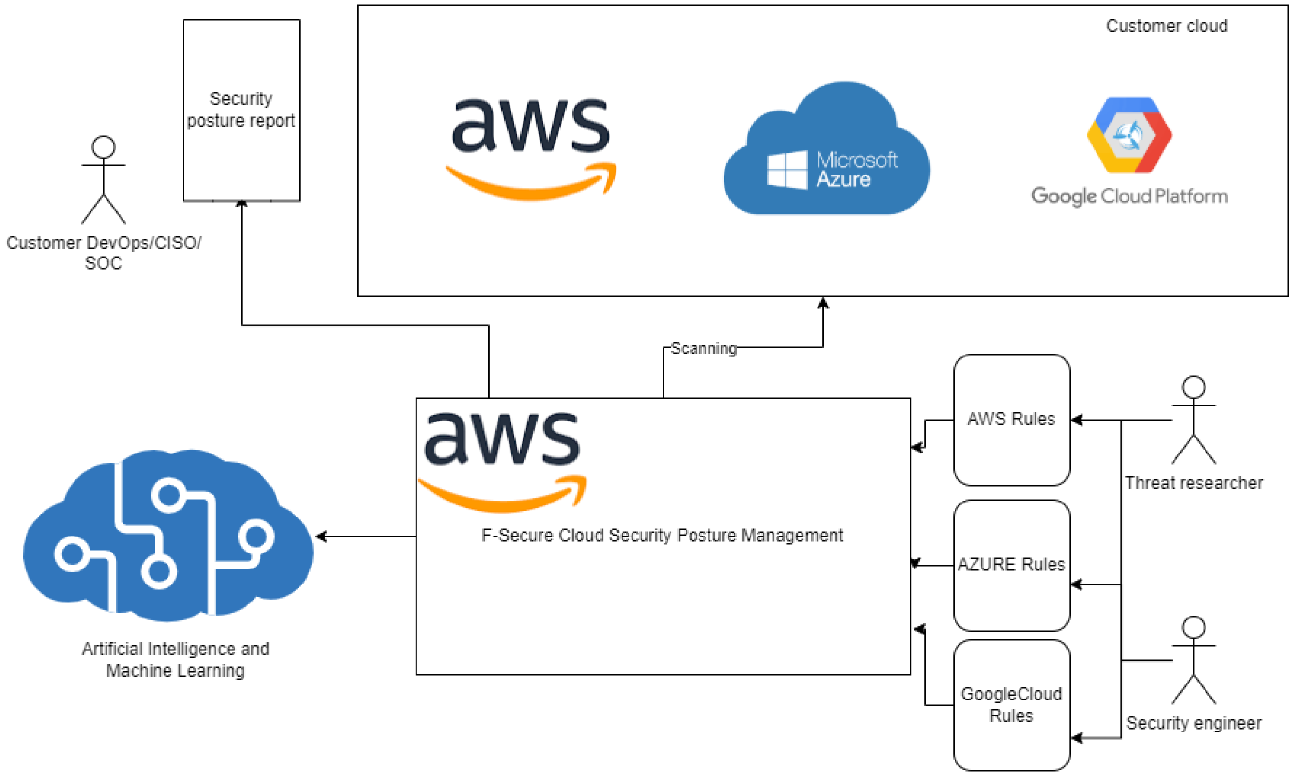


Figure 30: A multi-cloud concept for a Cloud Security Posture Management (CSPM) solution

Without CSPM, the users of cloud services may introduce misconfiguration vulnerabilities, and may fail to fulfil compliance requirements. Misconfigurations can result in major failures including loss of data and service downtime, which in turn may lead to direct business impact and reputation loss. Note that there are also risks associated with CSPM, since it has privileged access to cloud infrastructure and could potentially be used by an adversary to access sensitive data.

CSPM is an industry standard that is available as both commercial and open-source solutions. Cloud providers issue native security tools that can be integrated by cloud builders. An example of this is AWS GuardDuty that can be used to detect both compliance violations and suspicious API activity, the latter being relevant to cloud detection and response capabilities.

7.2.3 Innovation step

WithSecure created a CSPM tool to (i) enable security consultants to help customers understand their security posture, and (ii) assist analysis operations during the aftermath of a security incident. The original tool was standalone and designed to be run directly from a consultant's laptop. The tool itself contained heuristic rules that were updated with information gathered during the analysis of security breaches and critical issues. Early versions of this tool required a lot of manual work to use – they needed to be installed and run from the command line and managing data and creating reports were also completely manual tasks. This original tool was not designed to scale to enterprise use that can have hundreds of cloud accounts, multi-cloud environment and thousands of findings.

Thus, a CSPM tool cloudification project was started, the aim of which was to preserve the rules written by security consultants while enabling scalability for use in partner and customer environments. The design principles were as follows:

- Enable security experts and consultants to create cloud security posture rules easily and flexibly while still being able to run the tool standalone for consultancy engagements
- Create a cloud native infrastructure for the tool in Amazon public cloud
- Enable connectors and rules creation on multiple public cloud environments
- Automate account configuration and storage of access keys using least privilege principles
- Create an easily understandable user interface that enables customers to understand and prioritise findings
- Enable the future addition of machine learning models by collecting metadata required to train relevant models

7.2.4 Validation results

Our initial hypothesis was that CSPM could be productized as a service offering for both cloud consultancy and peacetime value needs. The idea here was that the service would be run immediately upon first installation and then on a monthly basis for continuing subscribers. Customer feedback from the first phase focused on security outcomes instead of usability or configuration options. From this feedback, it was determined that the coverage of the AWS rules was adequate, but the Azure rules required improvements, as their detection power hindered service adoption (we note that Google Cloud rules are in the plans).

The tool's capability of enumerating misconfigurations in cloud infrastructure was tested within WithSecure's test and production environments (supported by the CISO and CIO offices). That brought several important findings, e.g., a misconfiguration of an SQS queue (the resource policy allowed anyone knowing the queue URL to push messages to the queue) and S3 buckets configured to be publicly accessible. Moreover, the findings of this study illustrated how vast WithSecure's in-house cloud service adoption was. New needs were identified during this phase of experimentation – since all DevOps teams were running their own cloud services independently, finding a single team to fix all raised issues was not possible. It was thus determined that it was no longer sufficient to leave overall cloud security in the hands of centralized team – each team or even individual developer were deemed responsible for securing their own environments.

Finally, it was necessary to develop an understanding of how to productize the CSPM tool. Pilot workshops with key partners illustrated that willingness to adopt such a tool required more thinking on how to bring new customers in, what level of self-service configuration was required, and surprisingly, how rule generation might address multiple industry benchmarks such as CIS or HIPAA.

7.2.5 Application to use cases

The following features of the system under development are relevant for TRANSACT use cases:

Security of distributed systems and cloud estates against misconfigurations

Relevant use cases: UC1, UC2, UC3, UC4, UC5

Rationale: all TRANSACT use cases rely on a distributed processing platform. It is critical to secure this platform against misconfigurations. Continuous CSPM monitoring can contribute to securing cloud platforms by alerting on misconfigurations in both AWS and Azure environments.

It is in the plan to generate CSPM reports for several cloud environments of the TRANSACT partners.

7.3 Cloud Detection and Response capabilities

7.3.1 Overview, motivation

Most companies are in the process of migrating systems and services to the cloud. Since many non-cloud security mechanisms are not suited to this environment, companies are faced with the need to design or install new, cloud-specific security measures. The main requirement for such solutions is that they can detect attacks against cloud infrastructure.

Cloud attacks are a serious threat. In our survey of 3072 participants, 47% said they had experienced one or more cloud-targeted attacks. The Ermetic IDC Survey Report suggests that 98% of all companies using the cloud are susceptible to data breaches. As such, securing cloud infrastructure and understanding the risks associated with cloud usage has never been more important. In our survey, only 21% of respondents reported having a security solution for Cloud Workloads, leaving a wide-open opportunity for malicious actors to exploit vulnerabilities and misconfigurations. Attackers are reportedly already using stolen credentials to gain access to cloud systems for the purposes of data theft and crypto-currency mining.

Given the fact that most companies deploy a large number of cloud instances, securing each platform and data source separately is not feasible. The need to be able to secure multiple cloud instances from a central location is also important when considering that contextuality and data fusion are crucial to recognizing future criminal activities.

Following attackers with Cloud Detection & Response (Cloud DR) orchestration for multiple cloud instances and environments represents a next-generation, high-level concept for detecting security-related activities in cloud environments and digital platforms. Cloud DR extends User Entity and Behaviour Analytics (UEBA) to cover different cloud systems and extends the detection and response paradigm.

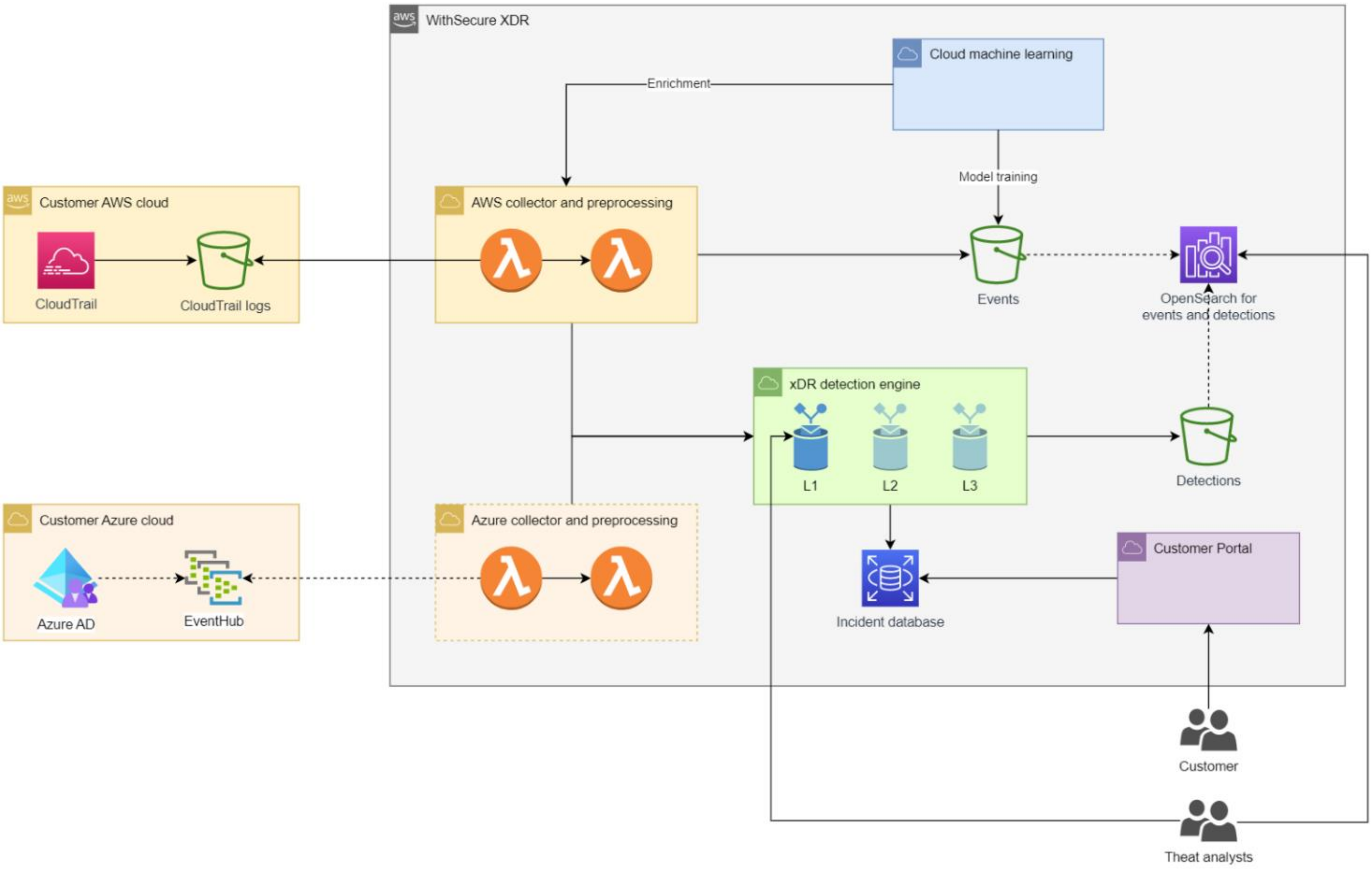


Figure 31: An architecture for collecting log data from customer AWS infrastructure and processing events, detections, and incidents through an analysis pipeline.

Figure 31: An architecture for collecting log data from customer AWS infrastructure and processing events, detections, and incidents through an analysis pipeline. introduces the overall architecture. The next step is to extend this architecture to cover multiple clouds by introducing new data sources to our Cloud Sensor components.

7.3.2 State of the art

Cloud infrastructure security monitoring tools can be divided into two categories – compliance-based tools that inspect the current state of the infrastructure against rulesets (such as Cloud Security Posture Management), and real-time tools that monitor the log flow or control plane activity of the infrastructure. All cloud providers provide native security tools that often offer capabilities from both categories. For example, AWS GuardDuty by Amazon can detect both compliance violations and suspicious API activity. In addition, some commercial products already contain both rule-based and machine learning-based detection capabilities for cloud event sources.

7.3.3 Innovation step

WithSecure's Cloud Detection and Response service extends the real-time protection capabilities of native tooling by offering a single solution for multi-cloud environments and integrating cloud detection capabilities with endpoint data. The service uses cloud provider-specific interfaces to tap into control plane activity event streams of each data source.

When events are ingested into the service, a processing pipeline wraps them into a generic cloud event schema. Components that handle these events can then be built either a) in a generic manner, providing common functionality such as estimating the prevalence of an event, or b) specific to certain event type, such as generating a detection based on suspicious activity on a certain API.

Detections can be triggered from a near-real-time event stream by either a multi-layered rule-based detection engine or behavioural models that compare events against a known baseline of entity activities.

As an input for the required detection capabilities, we constructed an attack matrix based on the knowledge of cloud domain security experts. This matrix has enabled us to create capabilities to detect relevant real-world threats that are not covered by existing native cloud security solutions. Using a shared event processing pipeline and detection engine for both cloud events and events from traditional EDR sensors, it is possible to create a broader view of possible malicious activity within a customer’s whole estate. For example, if a detection event from an endpoint device signals a possible unauthorized access, a seemingly normal cloud event triggered by the owner of that endpoint device can be linked to the same incident, providing visibility to the attacker’s actions.

As part of TRANSACT, a scalable pipeline for collecting AWS CloudTrail data and validating the architecture with internal data sources was first created. The resulting generic cloud event schema was then validated to be fit for Azure AD log events. Furthermore, a proof-of-concept mechanism for event ingestion from that log source was also created and validated.

One caveat with such detection and response technologies is the fact that the solution itself has permissions to access potentially confidential data. This is a generic security requirement for any security technology that has permissions to operate inside a customer’s cloud infrastructure. Unfortunately, this means that an adversary could potentially hijack the security system to perform criminal activities. We designed and implemented a secure way of storing the cloud account access using AWS Parameter store. Customers created a read only IAM role and deployed our custom cloudformation template to form the connection with our CSPM scanning tool.

7.3.4 Validation results

An initial set of detection rules was able to detect (simulated, in the red-teaming fashion) attacks identified in our attack matrix. Subsequent efforts will go into reducing false positives via statistical methods, machine learning models, and by utilizing the broader context collected by the multi-layer detection engine.

7.3.5 Application to use cases

The following features of the system under development are relevant for TRANSACT use cases:

Security of distributed systems and cloud estates against malicious intrusions (intrusion detection).

Relevant use cases: UC1, UC2, UC3, UC4, UC5

Rationale: all TRANSACT use cases rely on a distributed processing platform. It is critical to secure this platform against malicious actors. The Cloud Detection and Response system described can contribute to securing cloud platforms by alerting on malicious activity.

Monitoring and near real-time alerting for distributed systems in the cloud (fault detection).

Relevant use cases: UC1, UC2, UC3, UC4, UC5

Rationale: all TRANSACT use cases include a cyber physical component. Physical infrastructure is subject to faults and failures. The Cloud Detection and Response system described can contribute to protecting cloud platforms by alerting on potential faults and malfunction.

8 Data Communication Security

8.1 Privacy-by-design solutions for traffic monitoring in public spaces

8.1.1 Overview, motivation

Monitoring traffic for safety, maintaining the mobility or to reduce missions for sustainable transportation are a legitimate purpose but does not require personal data. Hence, the collection of personal data should be avoided or adequate measures should be taken to avoid a data leak or the publication of personal data by means of risk mitigation. In short, you should erase or anonymize personal data as soon as possible.

Only where essential for the functioning of a system – for example in prioritising public transport – are data used and exchanged that (in combination with information from outside the system) could be derived to an individual person. It is never the aim of the process to obtain information about a specific individual, but this is in theory possible by combining data sources. In these cases, measures have been taken to guarantee the privacy of road users, including agreements about which parties are authorised to use the information.

An example of personal data are vehicle license plates that reveals the identity of the owner once it is combined with the registration database. Video footage where a human being can be recognized is personal data. Also, less obvious data can reveal the identity of a person when combined with other information sources. For example, GIS-location messages that are sent by a traffic participant or a mobile phone can reveal the identity of a person once they are combined with an ANPR system or the schedule of a bus driver.

Road authorities therefore have to arrange a number of measures both technically and administratively. An important part of this is a processing agreement with the various stakeholders and a Data Protection Impact Assessment (DPIA).

This section focusses on the technical measures to avoid the distribution of privacy sensitive data.

8.1.2 State of the art

To mitigate the privacy threats /risks, we consider the privacy paradigm of prof. Jaap-Henk Hoepman (see Deliverable D3.2). For UC1, where traffic monitoring in public spaces is part of the fleet management, we consider the optical sensors that are used as a device that could jeopardize the privacy. People in the video footage may be recognizable and license plates of vehicles may be readable. The state-of-the-art implementation comprises camera sensors and one or more edge devices that read the video streams and extract the necessary information toward the cloud. The connection between the camera sensor and the edge device can be realized with different interfaces: compressed video via RTSP of raw video via GigE Vision, CameraLink, USB3, CoaXpress, or SDI.

We address the state-of-art per strategy that is formulated by the privacy-by-design paradigm:

Minimize. We only store metadata that is used for analytics. Video data that is received from the camera (tier) by the edge device is analysed and not stored. With respect to detected objects, only the following data is stored: counting results per 30 seconds per object class (for example: car, pedestrian, bicycle); trajectory/path per object; classification per object (into car, pedestrian, bicycle etc.); speed per object; density of the counting area per 30 seconds.

Hide. Analysis data is stored on a local storage. When connected to the edge Tier, the user has to provide authentication (username, password) to obtain rights to retrieve the stored data or to access the web UI. Data is sent to clients using HTTPS or web socket connections over VPN. This data does not involve personal data.

Separate. The data mentioned in point 1 is stored in separate storages on the same device. For example, object trajectories are stored separately from their classifications, counting results are stored separately from the camera configuration. Databases containing object information are only linked together using unique identifiers (UUIDs) but require access to the different containers. These UUIDs are not traceable to individual persons.

Aggregate. Personal information in the form of video images containing natural persons is discarded directly after usage and any visual data about detected objects is discarded after analysis. The only information available is classification into a type (for example: person, car), location (with respect to camera, and GPS) and speed. However, this information cannot be traced to individuals. This information is required for correct analytics for mobility research and crowd management.

Inform: According to the GDPR, data subjects, in this case the observed people, need to be informed when video surveillance is present, and/or when personal information is processed (such as unique MAC addressed from Bluetooth/WiFi sniffers). However, since the video feed of the camera is not stored or used for surveillance, and no personal information is detected or stored, informing the public is not strictly required. However, we do strongly recommend that system integrators and customers are correctly informed about the people that they are being monitored and ensure them that their privacy is respected.

Control: Since no personal data is stored nor exposed outside the edge device, the right of access, rectification, erasure, and restriction does not apply. Nevertheless, everyone is allowed to contact the Data Protection Officer of the contractor that is responsible for the processing.

Enforce: A privacy document describes the technical safeguards that the TRANSACT product uses to protect the privacy of the data subjects. This is enforced by the appointed Data Protection Officer of the supplier and reviewed at every TRANSACT release. In case of systems that process personal data, a Data Protection Impact Assessment (DPIA), should be performed. This enables both customers and data subjects to read about our privacy policy and data processing methods.

8.1.3 Innovation step

We consider the above mitigation as adequate, but still see room for improvement. The camera sensor and the edge computer where the images are analysed is typically hosted in a separate device. Although the output of the edge device does not contain personal data, the output of the camera sensors does. So, the interconnection is a vulnerability, especially when it is IP based.

Besides the traffic data that does not contain personal data, we do recognize a need to generate a video output. For example, for fleet management, the best situational awareness for the remote driver is to visualize the viewpoint from the vehicle as if the operator is in the vehicle. Besides this feature, it is also desirable to store video data to validate the traffic data from the edge device.

We propose the following innovations

- The camera sensor and the edge computer are embedded in a single device and connected with point-2-point connection using e.g., a MIPI CSI-2. As a result, there is no physical external cable that can be tapped to access personal data.
- As part of the video processing and analysis, processing technique can be applied to anonymize the video before it is stored or streamed. For example, a deep fake can be used to synthesize human faces and replace the original once. Faces, license plates and other sensitive information that is not relevant can be blurred.

8.1.4 Validation results

The validation will be done via a field lab in Tampere where road-side cameras will be installed for the fleet management application.

The validation of the above-described innovation is straightforward. After the project, once we developed our first prototype of a product, we will expose the prototype to a penetration test to see if access to personal data could be possible via unauthorized access.

8.1.5 Application to use cases

This concept will be applied to UC1 where traffic monitoring in public spaces is part of the fleet management. This will be demonstrated in a field lab. The concept of privacy-by-design is generic, and the innovations could be applied more broadly for use cases where personal data is not required. For use cases like medical imaging, the processing of personal data is the objective and therefore, the personal data cannot be removed directly after the sensor. Hence, the innovative steps are not applicable in these cases. Nevertheless, the privacy paradigm of prof. Jaap-Henk Hoepman still applies for all use cases.

8.2 Security and Privacy of medical/healthcare DICOM Solutions

8.2.1 Overview, motivation

So far, the imaging modalities (such as MRI, CT, X-ray, and others) have been installed as standalone devices with secured connectivity to the local hospital infrastructure. Moving the safety-critical imaging modalities architecture from the centralized, on-device solution toward the distributed, cloud-based architecture significantly increases the new solution's attack surface. Also, the data privacy concerns are growing considerably in such architecture as the user data, especially in the healthcare domain, is highly sensitive and requires special care not to be exposed due to being transferred to/from the cloud or due to security attacks and software vulnerabilities (e.g., cloud services using the health data). Therefore, the successful edge/cloud-based imaging modality healthcare solution must address end-to-end security and privacy. Specifically, it needs to apply the security mechanisms to safeguard the regulatory requirements and prevent disclosure, compromise, or misuse of the processed (DICOM (DICOM, 2022)) imaging healthcare data.

8.2.2 State of the practice

Current medical devices are fulfilling the safety, security, and privacy regulatory requirements addressing them in the context of their local, on-premises, hospital deployments. For example, in Philips devices (like interventional X-ray devices) the defence in-depth principles are employed to ensure proper security posture—the main defense security layers are:

- Operating system hardening: disabling all unnecessary operating system services and functions not required by the device that may become vulnerable over time.
- Malware protection: using anti-virus software or whitelisting (allowing only trusted applications and libraries to execute).
- Access controls and audit trail: allowing access to the system functionality and data only by authorized users.
- Secure patient data handling: data encryption in rest (the system drives are encrypted) and in transit (transferring DICOM data out of the system to, e.g., PACS[1] systems) in accordance with the DICOM transport security or selective encryption of DICOM headers protocols. The de-identification is typically used on the user request when exporting the patient data to the configured network nodes, printers, or removable media (e.g., DVD, USB drive).

[1] PACS—Picture Archiving And Communication System

- Network segmentation and firewall: minimize connectivity to the hospital network and block all unnecessary ports inhibiting communication with unauthorized computers to prevent network intrusion.
- Physical security: ensuring that system devices are located in secure areas safeguarded from unauthorized access.

The above security measures are not sufficient anymore when trying to use cloud-based services, especially to ensure healthcare data security in privacy compliant way. The majority of the current healthcare cloud-based solutions focus on helping hospital IT infrastructure cope with the big volume of data storage (in the PACS imaging archive systems) and securing its availability according to the regulatory (see Figure 32).

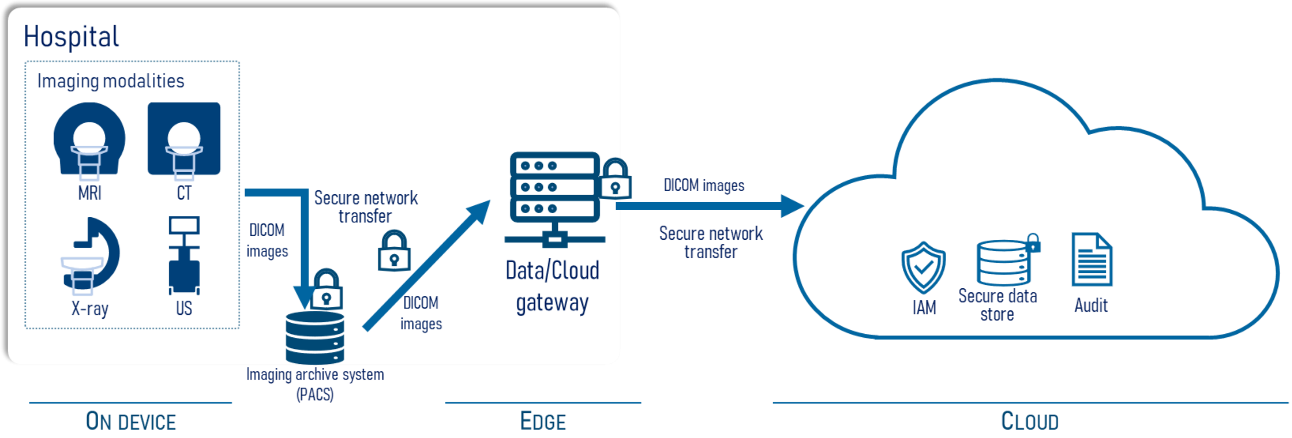


Figure 32: Imaging modalities storing DICOM images in device-edge-cloud deployment

However, for the interventional X-ray systems, cloud services are not only relevant for data storage but also for using CAD services (computer-aided diagnosis) helping to improve diagnostics of patient conditions during the intervention. Using cloud services as part of the interventional X-ray system architecture increases the attack surface of the new solution. Therefore, the security and privacy aspects need special attention to ensure patient data handling according to all regulations while ensuring the interventional system can still provide all required functionality safely and with needed performance.

8.2.3 Innovation step

Connecting the medical device (like an interventional X-ray device) to the cloud services impacts the end-to-end security and privacy of the new solution. To address those concerns and improve innovation speed, the new solution can use well-established healthcare domain solutions where the healthcare security and privacy aspects are already addressed. As presented in deliverable D2.1 (D2.1, 2022), the Philips HealthSuite Cloud and Philips HealthSuite Edge combined proposition is a healthcare-focused platform that helps building the innovative medical solutions. Specifically:

- The *HealthSuite Cloud* is a cloud-based platform comprised of health care -focused services (such as DicomStore, Auditing, IAM, etc.), capabilities, and tools that are optimized for health innovation solutions. It supports data collection, integration, and analysis from multiple data sources, such as medical devices, imaging modalities, genomics, digital pathology, patient monitors, etc. The data confidentiality and integrity are assured due to the multilayer security approach and centralized identity and access management. Security and privacy are critical components of the HealthSuite Cloud platform, all embedded into each aspect of the platform development and operational lifecycle, which is confirmed by a broad set of external compliance certifications and attestations such as

ISO27001/18 (ISO27001_18), SOC2 (AICPA, 2022), HITRUST (HITRUST Alliance, n.d.). In addition, HealthSuite Cloud meets rigorous local, national and global *regulatory* standards, such as HIPAA (U.S. Department of Health and Human Services, 1996), HDS (ASIP SANTÉ, 2022), GDPR (GDPR), FDA 21 CFR Part 11 (FDA-CFR-Title-21, 2022), enabling rapid development of medical solutions using infrastructure satisfying the health-domain regulatory and compliance requirements.

- The *HealthSuite Edge* is designed to provide a safe and easy to maintain virtual private network and lightweight application hosting environment that can be centrally managed from the HealthSuite Cloud. In addition, it establishes a secure connection (acts as a gateway) between on-premises hospital devices and the HealthSuite Cloud services.

The cloud-based interventional X-ray system solution will use *HealthSuite Cloud* and *HealthSuite Edge* (see Figure 33) to connect to the cloud services, such as IAM[2], CAD[3], etc. As a result, it is desirable to evaluate how the security and privacy aspects of the new solution impact the imaging device design, i.e., what security and privacy aspects need to be addressed to securely connect to the cloud services and share data. Since the interventional X-ray device has very strong safety and performance requirements it is interesting to understand if there are particular security and privacy measures that need to be used in order to ensure proper performance of the new solution so the feedback from CAD services is available in time.

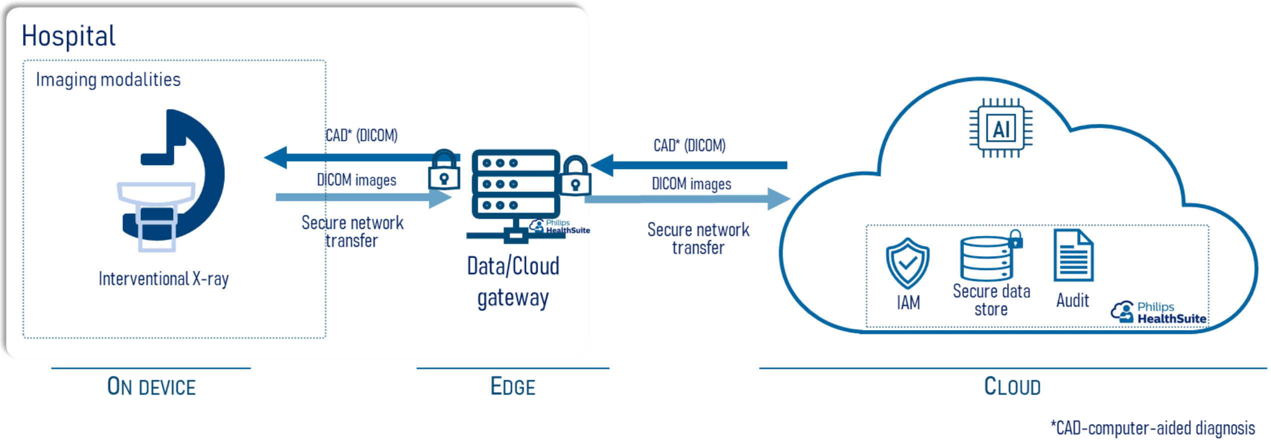


Figure 33: Interventional Xray-modality deployed on the HealthSuite Cloud/Edge platform

8.2.4 Application to use cases

Since the solution presented in this section focuses on security and privacy aspects around processing the DICOM imaging healthcare data, it is the most applicable to *Use Case 4 - Edge-cloud-based clinical applications platform for Image Guided Therapy and diagnostic* imaging systems (see Section 3.2.4).

[2] IAM—Identity and Access Management

[3] CAD-Computer Aided Diagnostics

9 AI/ML based Solutions

9.1 Privacy Preservation by leveraging the concept of federated learning

9.1.1 Overview, motivation

Probably, the most common problem in Machine Learning is the lack of data for training. Centrally collecting data from different sources including private ones, however, will most likely conflict with the European General Data Protection Regulation. The regulation is based on two pillars: data protection and data privacy. The articles of the regulation describe aspects of: Transparency and communication, right of access, accuracy, right to erasure, right to restrict processing, data portability and right to object. The GDPR was a reaction on the lousy self-understanding of companies when it comes to respecting the privacy, ownership and usage of users' data. However, still non-compliants of the regulations become public.

Apparently, privacy and data handling for machine learning seem to be competing. Instead of developing systems which are compliant with every given and future regulation, an alternative approach is privacy-by-design, e.g., Federated Learning. In this case, private data are never transmitted and only stored locally.

9.1.2 State of the art

The common approach to use distributed data for machine learning is based on transmitting all data from the different clients to a central cloud backend, where data is stored and processed. See step (1) in Figure 34. In this way, the scalability in both directions (horizontal and vertical) of the cloud can be fully leveraged. Then, (2) the aggregated data are used for a training for the neural network but also for performing the inference. The result is then transmitted to the clients (3).

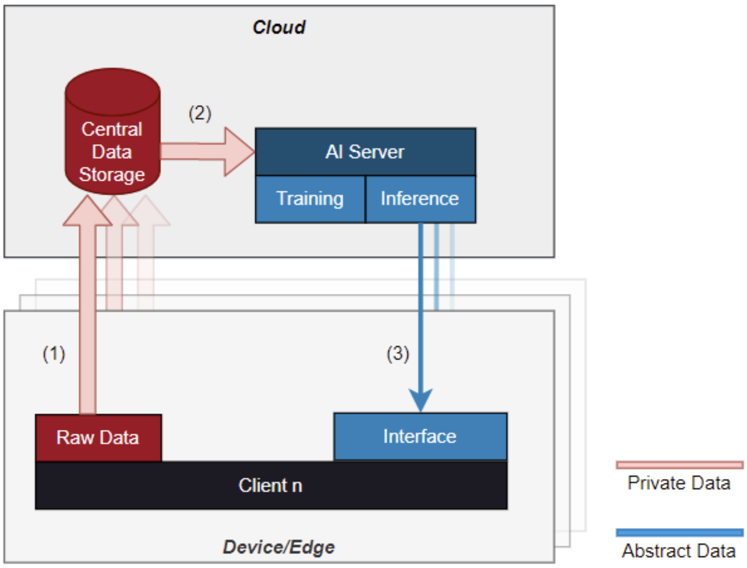


Figure 34: Data handling and processing for centralized learning of distributed data

Indicated in Figure 34: (1) Relevant raw data are transmitted from all clients to a central data storage. (2) The data are aggregated and used for performing the training and the inference. (3) The result is transmitted to the individual client.

The main benefits are:

1) Scalability of the cloud performance.

2) High flexibility and maintainability as the majority of the source code runs in the cloud backend.
3) Availability of all data.

On the downside of this concept – beside of privacy - the latency of transmitting the data and especially the non-dependable edge-cloud roundtrip communication limits the usage to non-time-critical and mostly to non-safety-critical functions, accordingly.

9.1.3 Innovation step

Instead of transmitting the private data to the cloud backend for training, the training process runs locally on the device or edge resulting in a local update of the pre-deployed neural network. See Figure 35, step (1). Then the new and abstract weights are transmitted to the cloud backend (2), where they are collected from different clients and averaged. There is no transmission of raw data anymore and any issues regarding privacy has been side-stepped. The new weights are then re-deployed as a global update to all clients and the inference (3). In this way, the private data remain on the device/edge which is owned by the person which produces the data – a privacy shield is built up.

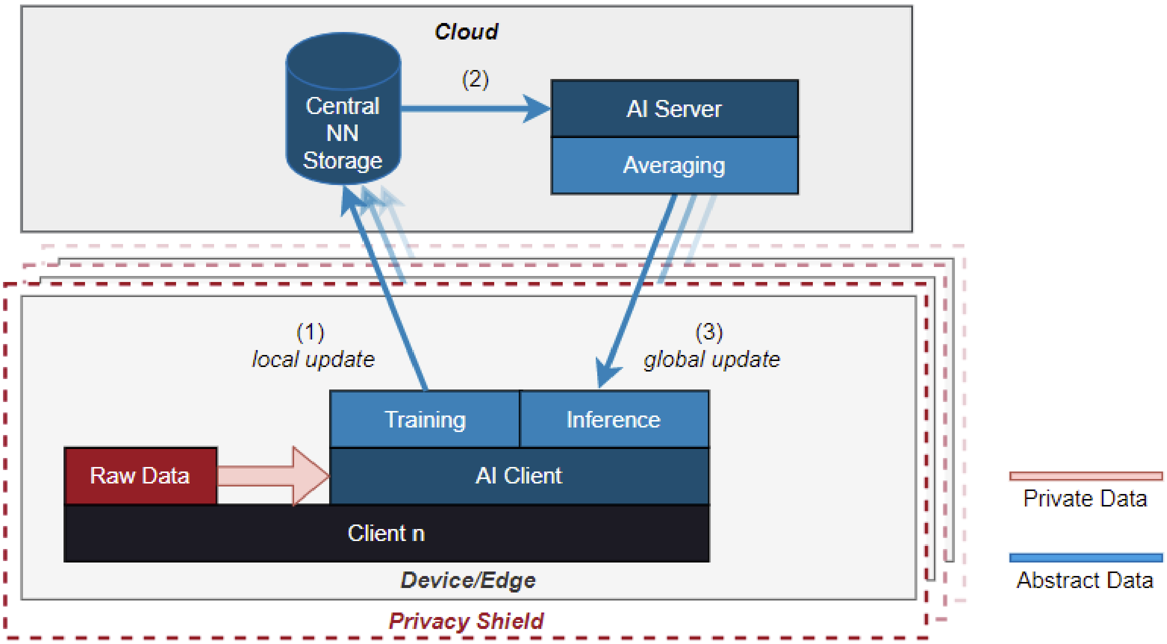


Figure 35: Privacy-by-design solution for training decentralized data.

(1) Raw data are only used locally for the inference but are never transmitted to an instance outside the vehicle. They are also used for a training which is performed locally only. (2) The local update of the neural network weights is then transmitted to the central storage where they are averaged with the weights of all other participating clients. (4) The global update is then deployed at all clients.

With this concept, new requirements for the device/edge component emerge:

1) Pre-processing of the data to select and clean up the relevant data.
2) Capability of storing data in a non-volatile storage for the whole lifetime.
3) Storage for the neural network.
4) Computational power to perform the training and inference locally.

9.1.4 Application to use cases

In UC3 (Cloud-Featured Battery Management), the remaining useful lifetime (RUL) of the battery is predicted using a machine learning approach. In the “classical” situation of centralized learning, all BMS and vehicle data would have been sent to a single cloud backend. However, these data may include sensitive data which are valuable to improve the quality of the neural network’s prediction. Instead of transmitting the raw data, they are used to perform the training of the neural network directly on the vehicle but only the result is exchanged with the central cloud instance. The procedure looks as follows: An initial neural network is trained by using lab data which is deployed on the vehicle from the very beginning. During operation, every vehicle submits local data which are processed by the Gateway to further improve the neural network. After a defined step, e.g., time, data points, the Gateways transmit the new local updates to a central cloud where they are averaged to create a new global update based on all local updates. Next, the global update is sent to all clients and replace the existing model. In this way, all vehicles exchange their knowledge and benefit from each other albeit they have not exposed their raw data (see Figure 36).

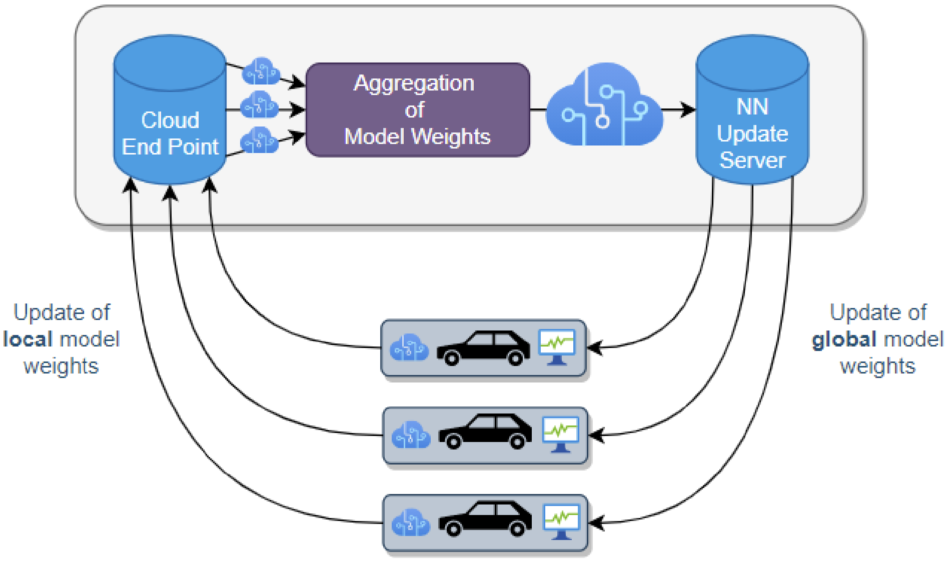

Cloud End Point
Aggregation of Model Weights
NN Update Server
Update of local model weights
Update of global model weights

Figure 36: Exchange of model weights of a neural network which is used to predict the remaining useful lifetime of the battery.

9.2 User behavioral based ML models for anomalous activities in cloud environments

9.2.1 Overview, motivation

Detecting anomalous user behaviour is of utmost importance for preserving the security of multi-tenant systems. This task is particularly challenging given the wide variety of user behavioural patterns that might exist.

Solutions based on rules or heuristics lack both the flexibility and accuracy necessary to capture the large number of possible user behaviours. Additionally, these solutions are not scalable, since manually creating and maintaining a set of rules for each user requires huge effort.

Behavioural machine learning models can model the behaviour of a large user base using a compact representation, referred to as a user profile. Given a sample of user-generated activity, these models can learn how to discriminate between anomalous and normal user behaviour. Note that these profiles are not restricted to representing users. In fact, the concept of a behavioural profile can be extended to any relevant entity in the system under analysis, such as processes, endpoints, and IoT devices. These behavioural models are usually categorized under the term User and Entity Behavioural Analytics (UEBA).

Cloud environments are becoming increasingly critical for the well-being of organizations. Cloud adoption has gained popularity for reducing infrastructure costs and achieving scalability, independently from the size of the organization. This highlights the need for managing the cloud resources of an organization in a secure and responsible way.

All cloud environments are built on the concept of strong authentication. This is natural given the fact that cloud services are provided through publicly accessible endpoints exposed to the Internet. Therefore, each request must be strongly authenticated from the cloud provider for security, monitoring, and billing purposes.

Considering the anomalous activity detection task, cloud environments present unique challenges that need to be addressed. Firstly, the concept of 'user' needs to be revisited in favour of a more general term. In a cloud environment, each action (or request) is assigned to an entity. In some cases, these entities might directly represent a physical user of the system, but more often they represent abstract entities such as processes, services, or accounts, which cannot be directly linked to a physical entity. This introduces the challenge of entity identification which is required before entity behavioural analysis can be performed.

Another challenge for UEBA in cloud environments is their dynamicity. The flexibility of cloud resource provisioning means that many cloud resources are frequently provisioned and then discarded across relatively short timeframes. With these premises, it becomes even more challenging to reliably monitor the behavioural patterns of any given entity.

9.2.2 State of the art

UEBA is used by market analysists as an umbrella term to categorize security systems featuring behavioural entity analysis capabilities. Considering that such capabilities can be provided by leveraging a large variety of machine learning techniques, it is somewhat arbitrary to define what the current UEBA state of the art is.

All cloud platforms rely on network communication to function properly. For this reason, most of the anomaly detection literature for cloud platforms focuses on network logs. These logs represent communication flows between hosts within the cloud estate. The analysis of this data using anomaly detection algorithms can provide insight into malicious misuse of the cloud platform, service faults and other interesting events.

One limitation of analysing the network as a whole is that entity-level anomalies might go unnoticed. Additionally, if different entities in the system have very different behaviour it is probable that the system would classify that behaviour as anomalous, even though the behaviour of each entity, considered individually, does not change.

UEBA techniques try to overcome this limitation by considering behavioural pattern at the entity level. To provide this feature UEBA techniques must implement at least the following phases:

- P1 – Profile generation (model training)
- P2 – Anomaly detection & alerting (inference)
- P3 – Model lifecycle management (concept drift & retraining)

9.2.2.1 Profile generation (model training)

UEBA techniques rely on a representation of historical (normal) behaviour which is then used as a reference to detect anomalies. This representation needs to be learned from historical data using a procedure commonly known as model training. Several challenges exist when training models for UEBA systems.

Naturally, the training data set must be limited in size so that training can be performed cost-effectively. However, the selection of the training set might introduce biases and it is one of the most important choices affecting the trained model accuracy. The training set should contain data representative of normal behaviour. This introduces two conflicting requirements:

The training set needs to be large enough to be representative of the system under consideration.

The training set must not include anomalous activity.

Another important choice when designing the training procedure is the selection of the size of the training set. A small training set, e.g., one week, might not include seasonal behaviour that happens infrequently, such as monthly maintenance tasks. If that is the case, and a proper false positive handling procedure has not been designed, the model might report some activities to be anomalous only because they happen infrequently.

A wide range of machine learning algorithms are suitable for UEBA systems. One-class classifiers are suitable for discriminating between expected behaviour and anomalies. Unsupervised approaches, such as clustering and nearest-neighbours searches can also be used to devise an anomaly detection scheme to identify suspicious activity. Most of these approaches implicitly rely on a vectorization scheme for the input data. The choice of the vectorization scheme can also significantly affect model accuracy.

Apart from the temporal horizon to consider, a key question when designing a training procedure for UEBA systems is entity granularity definition. In practice, the same system can be analysed at various levels of granularity such as at the network level, at the endpoint level, or at the process level.

Choosing the right entities to model is an important design choice. A more holistic choice would be to design UEBA systems for different levels of granularity and combine their outputs using an ensemble scheme. Naturally, this approach is conditioned on available project resources.

The output of the training procedure is some form of artifact, usually a model object, which compactly represents the training data and can be used to identify anomalous activity on previously unseen data.

9.2.2.2 Anomaly detection & alerting (inference)

The anomaly detection phase consists of using the artifacts produced during a training phase to analyse new data and detect behavioural anomalies.

Most UEBA models provide either a binary classification or a score representing the anomalousness of an activity. A score can be converted into a binary classification by introducing a numerical threshold.

One important thing to consider is that behavioural anomalies might not be directly linked to dangerous or malicious behaviour. This consideration is the key to understanding one of the main issues with UEBA techniques, namely the control of false positives (FPs). False positives are incorrectly classified data samples such as when a normal maintenance activity is categorized as anomalous by the system.

Models trained on historical data are designed to correctly classify the activity in the training set as normal. However, given the training limitations discussed previously, the training set cannot possibly include all examples of normal behaviour. Due to this limitation, a certain number of FPs is unavoidable. Therefore, a proper UEBA system should be designed with an FP handling strategy.

Implementing an FP handling strategy requires careful trade-offs between model accuracy and usability. A model reporting a lot of FPs alongside true positives is unusable, since the attention of the user will be wasted on the FPs. On the other hand, a model not reporting anything is clearly useless.

Deploying a model on live data might introduce significant latency depending on the characteristics of the pipeline. The throughput of the data sources under consideration is the most critical aspect to consider. When the data volume or velocity is very high it might be unfeasible to provide a classification for each data point independently. In this case, batch or on-demand analysis techniques can be used to mitigate the latency issues.

9.2.2.3 Model lifecycle management (concept drift & retraining)

The trade-off between model reliability and dynamic data source is unavoidable. A model cannot access data generated after its training has been executed. Additionally, most data sources follow some form of periodicity in how data is generated. This is particularly true when considering user-generated data. This phenomenon is referred to as concept drift.

Concept drift introduces a conundrum between having a stable model and having a model that correctly represents the status of the system, which varies over time. To mitigate this issue several strategies are available:

Periodic model retraining – model can be periodically retrained to prevent excessive concept drift. The retraining period should be set based on domain knowledge and data source characteristics.

Streaming algorithms – some machine learning algorithms can be implemented to work directly on data streams using one of many windowing operators, for example, a sliding window. This allows for stale data to be dynamically forgotten while new data is incorporated into the active training set.

9.2.3 Innovation step

In the context of the project, we developed a detection and response system with UEBA capabilities for anomalous activity detection in cloud environments.

The system consists of a data processing pipeline ingesting near real-time data from cloud service providers (CSPs). The processing pipeline can ingest data from multiple CSPs.

All CSPs provide monitoring services for cloud estates. These services collect logs on each activity performed on the estate, such as resource provisioning, API calls, etc. By building our processing pipeline on top of these services, we can provide monitoring services over the whole cloud estate. The data collected by the pipeline is both analysed in near-real-time and stored for later consumption. Logs collected by the pipeline are converted into event objects.

The pipeline is designed to integrate with arbitrary machine learning algorithms, which can provide insight on the event flow, e.g., detect anomalies. The algorithms are trained on historical event data, and they provide inference results in near-real-time. We implemented our UEBA capabilities on top of this processing layer.

UEBA models are built as a collaboration between machine learning experts and security experts to capture the most relevant information needed to detect security threats. As previously mentioned, UEBA capabilities can be developed at different levels of granularity, depending on which entities are being considered. As a first iteration, we built a UEBA model to detect anomalous activity at the organization level. In this context, entities are services, accounts, and users performing actions within the cloud estate.

The set of features to be considered in model training was designed iteratively in collaboration with security experts. The feature set includes the API call identifier, the entity identifier, and the authentication mode used, among others. These features provide a high-level baseline of activity within the cloud estate, and they can be used to detect anomalies.

The model is trained on historical data collected by the pipeline. An investigation is ongoing to find a reasonable value for the training window to consider. Techniques to handle possible outliers in the training set are also under investigation. The training algorithm used in this first iteration is a histogram-based technique to model the event data distribution and detect outliers. The data is grouped by feature set as well as by other characteristics such as time of day, errors, etc. The probability of each group in the training set is converted to a real-valued score between zero and one hundred. This transformation is helpful to analysts for quickly evaluating the abnormality of an event.

At inference time, each event is considered with respect to the feature group to which it belongs. The event is then assigned a score representing whether it is considered suspicious by the model.

To handle false positives, model results are not directly used to generate alerts. Instead, the model results are one component of a decision-making system which relies on several data sources. The ultimate responsibility of raising an alert is given to the security researchers managing the decision-making system.

The integration with the processing pipeline was designed to make model development and deployment seamless. We are ultimately planning on developing specialized UEBA models to detect misuse and anomalies at different granularity levels.

9.2.4 Validation results

We are currently analyzing the first outcomes of the model. This task is being carried out by security experts with the goal of providing feedback to iteratively improve the model results until an acceptable level of accuracy is achieved. The evaluation data is generated by both realistic system usage and handcrafted simulations. This approach is used to evaluate the model results on corner cases that might happen very infrequently on a cloud estate, such as malicious intrusions. For these cases, the model is evaluated on simulation data.

9.2.5 Application to use cases

The following features of the system under development are relevant for TRANSACT use cases:

Security of distributed systems and cloud estates against malicious intrusions (intrusion detection).

Relevant use cases: UC1, UC2, UC3, UC4, UC5

Rationale: all TRANSACT use cases rely on a distributed processing platform. It is critical to secure this platform against malicious actors. The UEBA detection system described can contribute to securing cloud platforms by alerting on anomalous behaviour from entities in the system.

Monitoring and near-real-time alerting on potential equipment failures (fault detection).

Relevant use cases: UC1, UC2, UC3, UC4, UC5

Rationale: all TRANSACT use cases include a cyber physical component. Physical infrastructure is subject to faults and failures. The proposed UEBA detection system can be easily adapted (via selecting appropriate entities and their features) to detect workflow anomalies within the device fleet, and to prevent catastrophic failures before they happen.

It is in the plan to use the UEBA tool for training a few models for specific TRANSACT partners' use cases, which will require historical data for 1 - 2 months of the normal operating behavior of the corresponding systems.

10 Risk Analysis

10.1 Solutions to minimize security and privacy risks of medical/healthcare components/apps deployed in the edge/cloud

10.1.1 Overview, motivation

The security and safekeeping of protected health information (PHI), medical records, and personally identifiable information (PII) is critical topic for healthcare solutions. Data breaches within the healthcare industry are caused by vulnerabilities in used services and applications, inadequate system configurations, and disclosure of patient data as a human error or via lost, stolen or wrongly disposed devices. Understanding the common security and privacy threats helps to focus on *proactive* approaches to prevent security incidents from happening in the first place. One of the key proactive approaches to ensure secure and privacy compliant medical solutions is to use adequate security and privacy *development practices* to create secure healthcare products.

Current medical devices already adhere to stringent regulatory requirements ensuring security and privacy are part of their product lifecycle (see also Section 8.2.2). However, by extending the medical device functionality to use the cloud services brings additional security and privacy concerns not seen earlier when creating on-premise-based solutions. To address those concerns proactively, the security and privacy development practices require further attention during the early phases of the product lifecycle, starting from the product requirements through the product design, development, verification, and release. Each phase should employ practices helping to ensure that the patient data is protected at all times according to healthcare data regulations and laws in the released product.

The Philips HealthSuite Cloud platform provides services, technical tools and resources optimized for the co-creation, rapid development and deployment of healthcare applications. The provided platform cloud services are built to be used in healthcare solutions adhering to healthcare security and privacy requirements. Those services are developed following the industry security and privacy practices during each phase of the product lifecycle. Leveraging those practices, while rearchitecting a medical device solution (like interventional X-ray) to be deployed across the device/cloud/edge continuum, can help to improve its security and privacy posture.

10.1.2 State of the practice

When building healthcare-cloud targeted solution the security and privacy of the cloud platform have to be considered. The Philips HealthSuite Cloud platform is built on top of AWS taking advantage of its ”*security of the Cloud”* solutions and enhances them by the healthcare requirements: ranging from adapting a least-permission security model (where clients/operations/administrators only get a minimal number of roles/privileges required to perform their tasks) to additional strict pre-conditions measures to get elevated permissions in order to access restricted system functionality (e.g., by using multi-factor authentication). The HealthSuite platform services are built primarily for the healthcare domain following rigorous security and privacy development practices to ensure the highest level of security and compliance with healthcare regulations. The HealthSuite platform’s product development lifecycle practices are based on the industry frameworks such as Microsoft Security Development Lifecycle (SDL) (Microsoft SDL, 2022) and ISO27034 (ISO27034). The relevant security and privacy practices are outlined in Figure 37.

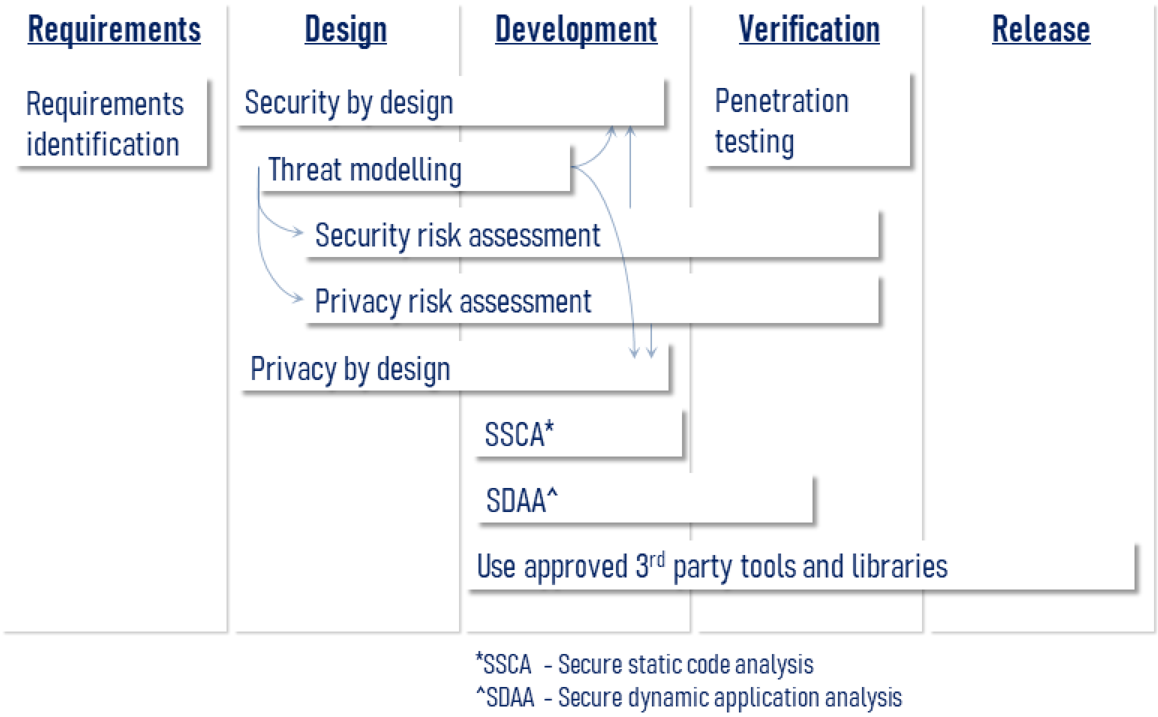


Figure 37: Overview of the security and privacy development practices

10.1.2.1 Requirements identification [Requirements]

The security and privacy requirements analysis should be performed as soon as it is clear the scope for the developed product. It should include specification of the security requirements for the product as it is designed to run in its planned operational environment. The privacy requirements should cover the processed data related aspects as per healthcare regulation and law needs. Deliverable D3.2 (D3.2, 2022) provides a comprehensive list of security and privacy requirements that should be considered.

10.1.2.2 Security by design [Design/Development]

The security by design approach ensures that the security controls are designed into the product instead of relying on security auditing done retroactively. It enables a “defense-in-depth” approach which places security controls at various levels—application, computing, data, or network–and administrative and operational safeguards. They cover different areas including authorization, audit controls, emergency access, data integrity and authenticity, secure storage (encryption ‘*at-rest*’), secure communication (encryption ‘*in-transit*’), secure deployments, secure key management, etc. They typically map to well established security frameworks and standards such as ISO27001/27002/27017/27018 and NIST SP 800-53 (NIST-SP800-53, 2022).

10.1.2.3 Security risk assessment [Design/Development/Verification]

The product security risk assessment helps to determine the security weakness of the products at the early stages of the product development process. It helps to identify, communicate and understand product/solution threats and vulnerabilities and identify additional countermeasures and operational controls to be implemented during the design and development phases. Typically, the security risk assessment has the following activities:

- Identifying the scope for the assessment: collect information about product assets (such as the hardware, software, data assets, and services) that could be compromised.
- Conduct the risk assessment by following the steps:
    - Identify risks: use threat modeling, asset/impact assessment, a vulnerability assessment, or any combination thereof, to identify the relevant threats sources, corresponding threats events, and the product vulnerabilities exploitable by the threat sources through specific threat events.
    - Determine the likelihood of an identified risk: estimate the likelihood of the threat and the likelihood of the vulnerability.
      Typically, the likelihood is expressed in 5 levels: *very high* (likely to happen and there are no good mitigations possible), *high*, *medium*, *low*, and *very low* (low possibility of occurrence due to available controls in place).
    - Determine the impact of an identified risk: when considering the intended use of the product, estimate how significant is the impact of the successful exploitation of the risk on system/function confidentiality, integrity, and/or availability.
      Typically the impact is expressed in 5 levels: *very high*, *high*, *medium*, *low*, and *very low*.
    - Calculate the risk value: it is a combination of the likelihood of an identified risk and its impact
      Typically the risk value is expressed in 5 levels: *very high*, *high*, *medium*, *low*, and *very low* based on the matrix with likelihood and impact values (NIST-SP800-37, 2022):

LIKELIHOOD	**IMPACT**				
	Very low	**Low**	**Medium**	**High**	**Very high**
Very high	*Very low*	*Low*	*Medium*	*High*	*Very high*
High	*Very low*	*Low*	*Medium*	*High*	*Very high*
Medium	*Very low*	*Low*	*Medium*	*Medium*	*High*
Low	*Very low*	*Low*	*Low*	*Low*	*Medium*
Very low	*Very low*	*Very low*	*Very low*	*Low*	*Low*

- Identify the risk mitigations: based on the risk assessment results identify the risk mitigation that should be implemented in the product or relevant processes.
- Manage residual risks: for all the not-mitigated high risks values a risk/benefit analysis should be performed on how to monitor or cope with each risk.

All the risks need to be documented and managed so their resolutions are acceptable at the release time of the product.

The security risk assessment for cloud-based services is a crucial methodology to proactively address security weaknesses. However, to have the best outcome of the security risk assessment it should be executed by the cross-functional team consisting primally of (in the case of healthcare products): the product/solution architect and designers, clinical experts, safety experts, operational IT staff, service engineers, legal representative, quality, and regulatory representative.

The USA NIST SP800-37 (NIST-SP800-37, 2022) is an example of the complete process description of how to conduct the security risk assessment.

10.1.2.4 Privacy by design [Design/Development]

The GDPR introduces a legal requirement on privacy by design for any party processing personal data. The concept of *privacy by design* aims to embed privacy and data protection controls throughout the entire data lifecycle, from the early product design stage to deployment, then during data collection, use, and ultimate data disposal. Privacy by design is outlined in (PrivacyByDesign, 2022)], some of the principles are:

- Prevention: anticipates and prevents privacy invasive events before they happen.
- Privacy as the default setting: no action is required on the part of the individual to protect their privacy—it is built into the system, by default.
- Privacy should be embedded into the design and architecture of the systems and business practices.
- Data end-to-end security: privacy must be protected across the domain and throughout the life cycle of the data.
- User-centric approach—requires architects and operators to keep the uppermost interests of the individual by offering such measures as strong privacy defaults, appropriate notice, and empowering user-friendly options.

Some of the techniques covering the privacy by design aspects are:

- Ensure protection of personal data by adequate data access techniques (authentication, role-based authorization, attribute-based credentials),
- Data minimization,
- Data de-identification,
- Design for data retention and deletion (e.g., ensure the data deletion information is propagated among involved services)
- Ensure mechanisms for data accuracy, availability, and integrity,
- Encrypt data where possible,
- Segregated data stores for different customers or propositions,
- Design a mechanism to support data breach notification.

10.1.2.5 Privacy risk assessment [Design/Development/Verification]

The privacy risk assessment may follow the same steps as the security risk assessment presented in Section 10.1.2.2. In essence, the privacy threats can be identified by leveraging the threat model created during the security risk assessment, including the way of scoring the risks' likelihood, impact, and overall risk end value. Similarly, the mitigation and residual risks can be addressed.

10.1.2.6 Threat modelling [Design/Development]

Threat modeling is a proactive strategy for evaluating cybersecurity threats. It involves identifying potential threats, attacks, vulnerabilities, and developing solutions to respond to those threats. Understanding how threats may impact services/applications/data helps to build appropriate countermeasures into the system. This way threat modeling allows analysis of security implications of chosen designs in the context of their planned operational environment.

Examples of the threat modeling approaches are Microsoft's STRIDE model (Microsoft_STRIDE, 2022), or NIST's CVSS threat scoring system (NIST_CVSS, 2022)

10.1.2.7 Use approved 3rd party tools, components, and libraries [Development-Operation]

Today, the vast majority of software products are built using third-party components (both commercial and open source). Therefore, when selecting 3rd party components to use, it's essential to understand the impact of a security vulnerability in them that could impact the security of the built services and applications. These risks may be evaluated by leveraging dedicated utilities to ensure that vulnerabilities are identified and remediated in time. An example of such a tool is BlackDuck (BlackDuck, 2022) which not only checks the potential risks in 3rd party code, but also reports security risks related to licenses and outdated versions of 3rd party libraries.

Another aspect of external libraries is having a proper *software bill of materials* that documents the tools used to build the system's applications and services and identifies precisely which 3rd-party components are included. This helps security organizations respond quickly and precisely to potential risks.

10.1.2.8 Secure static code analysis [Development]

Secure static code analysis helps to identify security vulnerabilities during the development phase. Analyzing the source code is a highly scalable method of security code review and helps ensure that secure coding policies are followed. Those tools are typically embedded into the SW development environment, or they are part of the delivery pipeline.

Example of such a tool is HP' Fortify (Fortify, 2022), Coverity (Coverity, 2022), Klockwork (Klockwork, 2022).

10.1.2.9 Secure dynamic application analysis [Development,Verification]

Secure dynamic application analysis tools perform run-time verification of the running integrated application. This is typically achieved using a tool that specifically monitors application behavior for memory corruption, user privilege issues, and other critical security problems. Those tools are typically used during the testing phase of the delivery pipeline.

Examples of such tools are WebInspect (WebInspect, 2022), Burp Suite (BurpSuite, 2022) and ZAP (ZAP, 2022).

10.1.2.10 Penetration testing [Verification]

Penetration testing is a technique for analyzing the security of a software system and it is performed by security professionals who simulate the actions of a hacker. The objective of penetration testing is to uncover potential vulnerabilities resulting from coding errors, system configuration faults, or other operational deployment weaknesses. Such a testing technique typically finds the broadest variety of vulnerabilities (PenTesting Microsoft, 2022) (PenTesting, 2022).

Independent third-party organizations can also perform that kind of testing as part of compliance or on-demand risk management efforts.

10.1.3 Innovation step

The on-premises healthcare devices are already built with good security and privacy techniques. However, it is critical to assess how the security and privacy aspects are affected by connecting the healthcare device with the cloud/edge-based services (or moving some of its services to the edge or cloud). Using the Philips HealthSuite Platform's cloud and edge services (fulfilling already healthcare security and privacy requirements) and its operations capabilities as part of the new solution addresses a significant portion of the security and privacy aspects.

Innovation focuses on using and assessing the security and privacy development practices (presented in Section 10.1.2) while transforming the interventional X-ray imaging modality from a device-only solution towards a solution based on the HealthCare Cloud/Edge platform. Especially, the *security and privacy risks assessments* practicies are critical to ensure building the end-to-end device/edge/cloud continuum deployment solutions that have adequate end-to-end security controls and comply with the privacy regulations and laws.

10.1.4 Application to use cases

The secure development practices presented in this section can be applied to most of the TRANSACT use-cases. However, they are the most relevant to the *Use Case 4 - Edge-cloud-based clinical applications platform for Image Guided Therapy and diagnostic* imaging systems (see Section 3.2.4), as this use-case covers system that processes the healthcare data so it requires to fulfil the healthcare regulations such GDPR (in Europe) or HIPAA (in the United States).

10.2 Risk analysis and SIRENA tool to real time monitoring

10.2.1 Overview, motivation

As information and communication technologies are growing in their importance in the society, it also increases the risks derived from their use. So, it is necessary for companies to analyse risks and potential threats, and how they would affect assets and business operations.

For that matter, it is important to carry out risk assessments, where you evaluate different security dimensions, such as:

Availability: Have access to assets when they are needed [UNE 71504:2008].

Integrity: Assets must have not been modified [ISO/IEC 13335-1:2004].

Confidentiality/ Non-disclosure: Asset's information should not be available to unauthorised parties [UNE-ISO/IEC 27001:2007].

Authenticity: An entity or asset is what they claim to be [UNE 71504:2008].

Accountability: The activity of the entity or asset can be monitored [UNE 71504:2008].

A risk assessment allows governing bodies to make decisions taking into account the risks derived from the use of information technologies.

For that matter, companies use tools that allow conducting risk assessment and monitoring in real-time for identifying threats or malfunctioning in company networks. SIRENA allows us to quickly identify all these issues by identifying not with an IP but with the name given. The risk assessment is based on two different methodologies: ISO/IEC 31000:2018 and MAGERIT. These methodologies allow consultants and companies to control Complex Management Systems. It monitors and centralises all the information in an optimal way, facilitating the location of reports, records, and indicators.

Thanks to these synergies between the risk assessment and the monitoring it is possible to control the risk, and therefore, to elaborate a risk management plan. One of the motivations, is to connect the business information from a client can provide us new semantic security functionalities since we can link this business information and IT information with the goal to monitoring threats with a set of assets obtained in the risk analysis.

10.2.2 State of the art

Risk analysis and risk management tools have grown in popularity in many applications to handle all the relevant information in a user-friendly way and to improve risk analysis services, including online, in-cloud, and real-time monitoring scenarios.

We think it is necessary for security and safety to ensure that the IT service industry follows the management standards for regulation compliance and governance in companies and uses the latest techniques in industrial network security to provide rapid response to possible abnormal behaviour. So, it is important to have tools aggregating and presenting all the relevant information via the same interface, ensuring smooth communication between the areas of AARR and Network IT.

10.2.3 Innovation step

GConsulting uses the MAGERIT methodology to conduct high-profile risk assessments, evaluating if a threat can actually materialise, its probability, and how it could damage assets. GConsulting also goes beyond the threats identified by MAGERIT, including certificate-based security solutions. It can monitor and centralise all the information related to the risk assessment, which ensures centrally monitored access control and function activation to provide secure diagnostics for companies.

After a risk assessment procedure is complete, SIRENA is used for threat prevention. In the SIRENA tool, we make an inter-connection with GConsulting to track all the assets, including IT or OT devices, with specific tags. Manually, we create active rules for devices and business assets.

10.2.4 Application to use cases

The need for performance management can be illustrated by the use case scenario “Transformation of the monolithic critical system in wastewater treatment plants to the distributed system supported in the cloud: management of the biological reactor” of UC5. This scenario is concerned with controlling the process and improving water quality. Furthermore, a wastewater treatment plants analysis will enhance the analysis and obtention of insights by the operator and lead to newer more advanced applications (predictive maintenance) that will result in a reduction of downtime, costs, and better service.

Through monitoring system, any abnormal behaviour, such as a leak, or low water pressure will be detected by SIRENA and alert the staff. The following picture shows the SIRENA tool GUI.

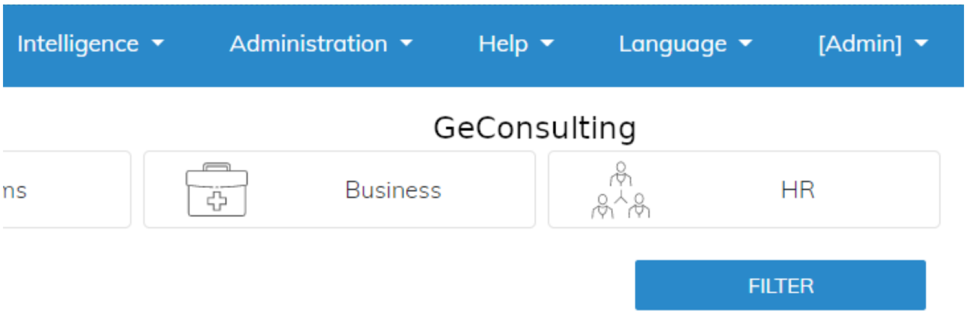
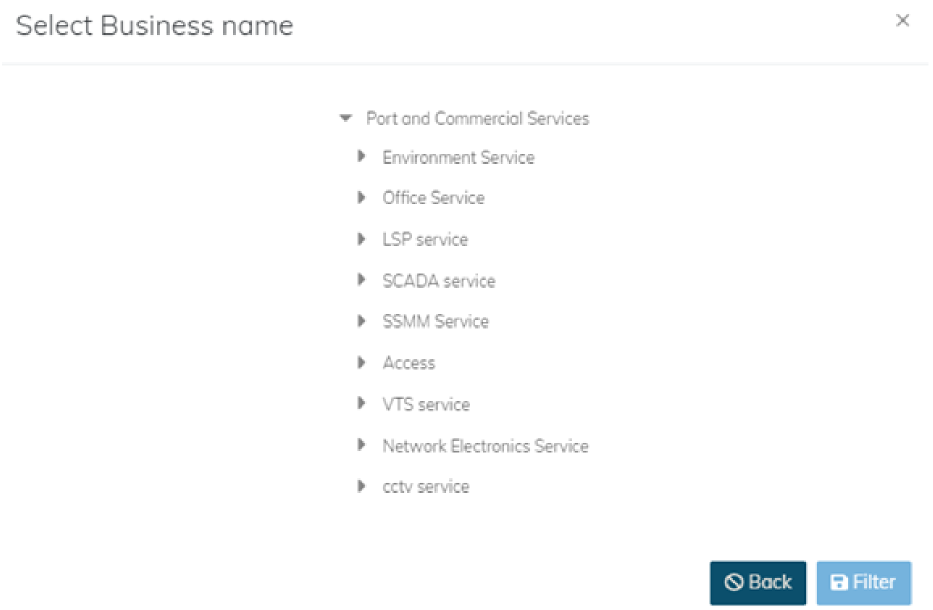


Figure 38: SIRENA tool

When creating a rule, it is possible to add a specific business tag obtained from earlier Risk Analysis procedures. The following picture shows the tags from a Risk Analisys in the Port Industry. Depending on the type of a company or the market, we can adapt this specification.

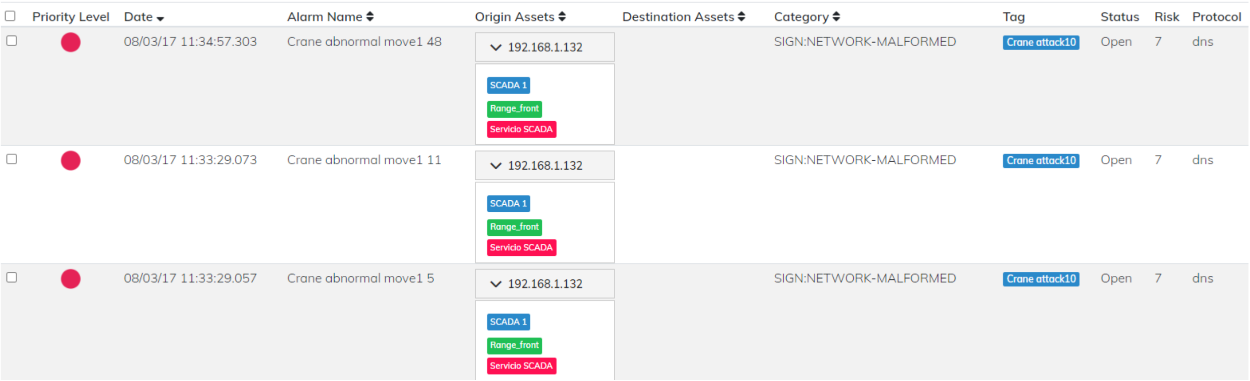


Select Business name ×

- Port and Commercial Services
  - Environment Service
  - Office Service
  - LSP service
  - SCADA service
  - SSMM Service
  - Access
  - VTS service
  - Network Electronics Service
  - cctv service

Figure 39: List of business tags from AARR in SIRENA

There we will link with the corresponding device, and when an alarm occurs (based on specific behaviour), tags from AARR are displayed in red colour. The below picture shows a list of alarms:



□	Priority Level	Date ▾	Alarm Name ⬍	Origin Assets ⬍	Destination Assets ⬍	Category ⬍	Tag	Status	Risk	Protocol
□	●	08/03/17 11:34:57.303	Crane abnormal move1 48	192.168.1.132 SCADA 1 Range_front Servicio SCADA		SIGN:NETWORK-MALFORMED	Crane attack10	Open	7	dns
□	●	08/03/17 11:33:29.073	Crane abnormal move1 11	192.168.1.132 SCADA 1 Range_front Servicio SCADA		SIGN:NETWORK-MALFORMED	Crane attack10	Open	7	dns
□	●	08/03/17 11:33:29.057	Crane abnormal move1 5	192.168.1.132 SCADA 1 Range_front Servicio SCADA		SIGN:NETWORK-MALFORMED	Crane attack10	Open	7	dns

Figure 40: List of alarms from SIRENA

11 Hardware based security

11.1 Root of Trust based Security

11.1.1 Overview, motivation

The IoT sector involves many linked devices that are increasing in their number each year. Those devices, as being exposed to the outside via Internet connection, are vulnerable to a number of different types of attacks. The lack of or insufficient security provides easy access to the critical parts of the system infrastructures. A Root of Trust (RoT) is the essential basic element of a secure IoT device. It can be perceived as a component providing a set of operations ensuring security that the rest of the system architecture can base on to perform further functionalities. The RoT implements credible functions including secure boot, attestation and cryptography that the other parts of the architecture can depend on. As this element is intrinsically trusted, it must be secure by design.

While implementing RoT processes, an elemental component must by all means work and behave in the expected manner. Using RoT techniques in a system allows for keeping private cryptographic keys confidential at all times. They are protected on the hardware side and kept isolated from the easier-to-hack software of the system. Moreover, the RoT modules must also be secured physically, at least for modest types of known attacks. But in more complex cases, perimeter detection systems can also be employed. Cryptographic operations performed by the RoT elements have the ability to handle, manage those keys and perform device authentication from the very beginning of a whole workflow as well as conduct encryption and decryption of the transferred data. In conclusion, using the RoT approach in a system requires executing some initial crypto-related functions before running the software that provides the rest of the service, which can be used to assure the authenticity of used hardware in a project.

11.1.2 State of the art

The RoT capabilities can be created software-side. The approach however is prone to errors and unexpected misbehavior. It requires exceptional control of the whole software, but its security level degrades over the years. Developing software-only Root of Trust introduces high complexity of the system’s code making a room for unnoticed, or appearing only after a wider span of time, undetected previously flaws.

Hence, the only reliable approach for this moment is the usage of hardware-based RoT modules. The electronic appliances that can perform such tasks include Trusted Platform Modules. They are ready-to-use (in comparison to crafting such devices from the very beginning or developing software RoT models) solutions that are able to perform secure functions while providing a defense from various types of attacks. TPM modules provision such functionalities as encryption, signing, attestation and secure boot. The system’s architecture can build on them to ensure its integrity and validity. One of those devices is Infineon Iridium SLM 9670 TPM2.0 module.

In the perfect scenario, RoT is a starting point of software that begins to run on the target platform. After launching, it checks the integrity of the software and analyses if nothing has been tampered with. If the validation sequence meets all the requirements, other parts of the system can boot chain by chain and start performing their tasks.

11.1.3 Innovation step

Connecting the vehicle’s hardware to the cloud services impacts the privacy and security of the integrated solution with the main focus on authenticating the device. To address those concerns and improve the system's innovation, as described in deliverable D9 (D3.2), DBox with a TPM module shall be introduced to enhance privacy aspects.

11.1.4 Application to use cases

The battery management system used in the TRANSACT project must be secure by design. Only the privileged hardware present in the vehicle can get access to the cloud connection. In order to achieve that, the Root of Trust chain must be introduced. That way, it is possible to guarantee that the device sending data to the cloud to process it further is not any malicious impersonating unit or spoofing device.

11.2 Hardware security: HSM to store keys on edge device

11.2.1 Overview, motivation

Data managed and communicated by IoT devices is often sensitive and needs to be protected from malicious third parties. When compromised, large fleets of devices can pose a significant threat to businesses and even the society. The target of this work package was to study and develop new hardware building blocks for the connected objects needed for IoT solutions (such as Smart Nodes and Gateways). These hardware building blocks focused on low power and secure communication. To ensure the confidentiality and integrity of the data throughout all process stages, secure hardware solutions were investigated for a secure key management system. End to end security concepts, with holistic product personalization processes and a related secure key management system, have their base in Secure Elements, to allow secure personalization from the very beginning in chip manufacturing up to custom configuration data provisioning.

11.2.2 State of the art

We repeat here part of the text in section 4.3.3 (Root of Trust) for convenience.

The A71CH Plug and Trust Secure Element of NXP provides a solution for secure provisioning. This is achieved by serving as a root of trust for the user. The root of trust is given by a pre-provisioned token. This token is provisioned while manufacturing the chip and enables the user to build a secure connection to the internet of things.

Using this secure connection, the user is able to securely provision his device with other tokens and credentials securely or even provision other hardware parts with desired information or intellectual property. The chip serves as a secure element and secure memory and is able to generate more secure keys. These attributes guarantee a safe peer-to-peer connection as well as a safe connection to the cloud.

The chip is also outfitted with security measures preventing many physical and logical attacks and offers plug and play capabilities to ensure zero-touch secure provisioning. All of the advantages of the chip mark it as a good solution for smart-home, smart-industry or even smart-cities.

The integration of a secure element offers several benefits regarding use cases related to security and privacy. As the secure element comes with a pre-configured asymmetric key pair, the private key new needs to be stored or transmitted outside the secure element. The public key can be distributed freely and is also supplied by the manufacturer together with the fresh and untouched hardware. This circumstance enables multiple security features:

Encrypted throughout lifetime: All connections with the gateway are encrypted during its complete lifetime. The private key, securely stored in the device, and the public key, known by all shareholders, can be used to securely exchange session keys for each session and use common technologies like Transport Layer Security (TLS) protocols for secure data transmission.

Secure Commissioning: The initial settings sent to the device can be cryptographically verified using the key pair from the secure element. This ensures that any updates, software, settings or any other data, are safe from manipulation during the download to the gateway device.

Secure Boot: Even if no data is transmitted, devices in the field are prone to manipulations by attackers who have physical access to them. A secure element, which is designed to be hard to manipulate, can be used to verify the overall system's state and especially the main application during the boot phase and detect unauthorized manipulations to it.

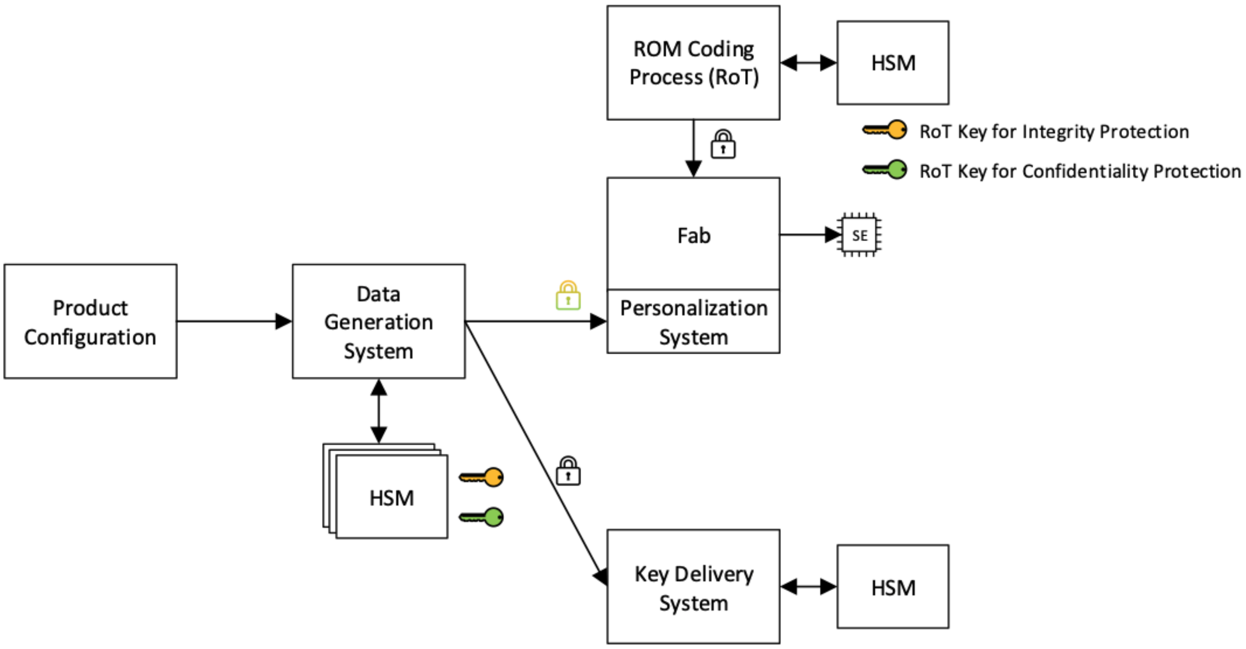


Figure 41: HSM to store keys on edge device

11.2.3 Innovation step

A secure system is only as strong as the weakest link in the overall system. Consequently, the process how confidential data (like keys) is transferred to the Secure Element is as critical as the design of the overall hardware element. This essentially consists of two aspects: Protection of initial data programmed during the manufacturing of the devices, and the protection of data or services used in the field. In the context of this work, the focus is on the personalization process for the initial device data during manufacturing.

The key aspects in the secure personalization process are:

- Confidentiality- and integrity protection of firmware images downloaded to the devices during manufacturing.
- Confidentiality- and integrity protection of custom configuration data (like keys) downloaded to the devices during manufacturing.

If these aspects are not covered during manufacturing, an attacker could e.g., manipulate firmware images and could include backdoor functionality that allows the export of any arbitrary data stored on the device. In addition, if the communication channel during manufacturing is not secured, an attacker could simply eavesdrop the communication to gain access to custom configuration data. As such, a manufacturing root-of-trust (mRoT) on hardware level needs to be established. This mRoT is usually created via complex processes where confidential data is split into multiple shares which are hidden in hardware units (so called mask coded bits or multiplexers) which are used in combination with functions implemented in hardware (encryption & scrambling, and partially physically unclonable functions). As such, mRoT keys can be used to protect the confidentiality and the integrity of data downloaded to devices during manufacturing. These mRoT keys must be kept private to ensure the overall system security. Thus, these keys are usually only accessible by the device itself and HSMs with which those keys are shared securely. A high-level personalization system making use of the mRoT keys is highlighted in Figure 41: HSM to store keys on edge device.

A Secure Element can be considered a Hardware Security Module (HSM) which protects the contents stored on the device against a wide variety of attacks. The protection mechanisms include measures against hardware attacks (like attacks involving etching) and software attacks (side-channel attacks; like differential power analysis).

After the finalization of the hardware requirements, CISC build the hardware prototype for the secure sensor node. For this the EdgeLock SE050 development board was used as well as the very versatile controller, Raspberry Pi. In the present case, the external I2C connector interface was utilized to enable local communication to the Raspberry Pi device. Next to the hardware integration of SE050, we integrated NXP's support package for managing the interaction between host controllers and secure elements in software. While the host-MCU runs the application logic and controls all cryptographic operations, the secure element executes them.

The sensor nodes form the first end of the end-to-end secured communication. They meet important security requirements, such as data confidentiality, integrity, as well as authentication and authorization management. Figure 42: gives an overview of the Sensor Node and its integrated hardware building blocks. Summarized, the sensor node is equipped with temperature and humidity sensors and dedicated Wi-Fi and BLE modules. It collects the raw sensor data at regular intervals, applies filters, and encrypts it. Last but not least, the data is forwarded to a nearby gateway device. There are two proposed types of sensors attachable to the Sensor Node itself, both use different interfaces to connect to the Node.

1. The Sensor of type 1 uses the General Purpose Input-Out (GPIO) ports. In order to guarantee accurate sensor readings, multiple reading iteration have to be executed.
2. The sensor module of type 2 uses the I2C connection to communicate to the sensor node. It not only consists of different sensors but also of actuators (e.g., LED-matrix) and input-methods (e.g., Joystick).

Regarding the software integration of the secure element, the implementation of CISC and TUG foresees the execution of the following tasks:

- Store Long-Term-Key: This is part of the RSA key provisioning method
- Store Key Pair: An additional key pair that is required for using a BLE security feature
- Random-Number-Generator (RNG): A true random number generator enables secure creation of different key material. Due to the implementation in hardware, lower execution times can be expected.
- Generate and Store Short-Term-Keys (STK): The creation of STKs enables a more efficient end-to-end data transfer via symmetric cryptographic approaches.

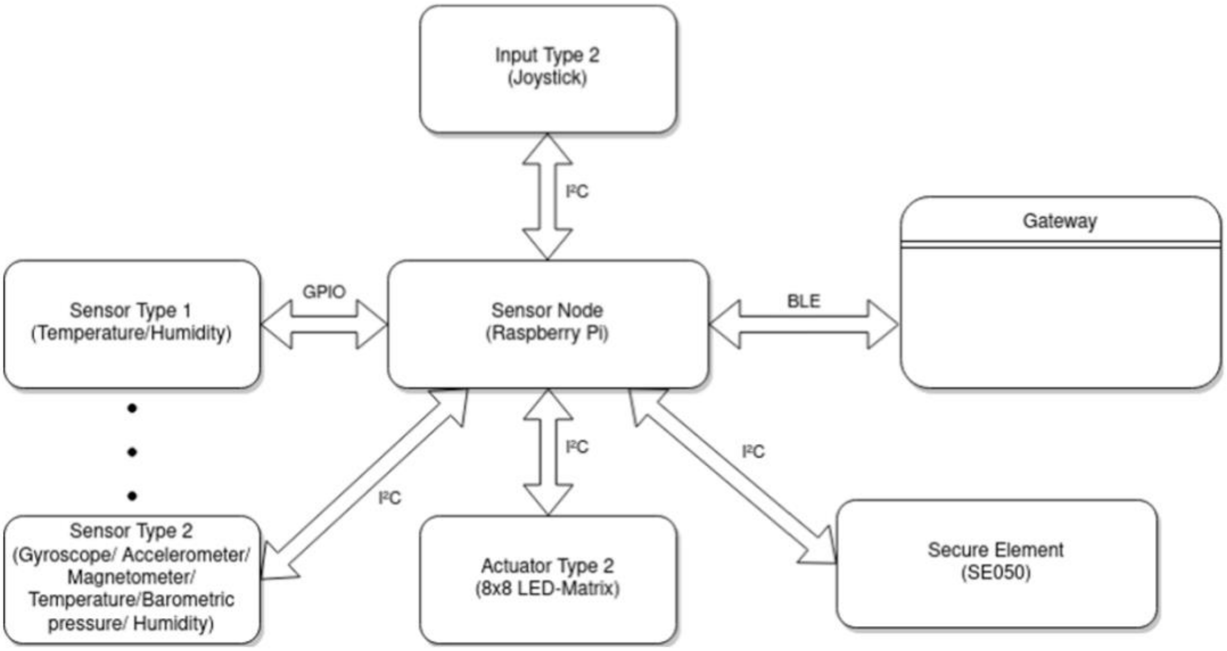


Figure 42: Sensor node and integrated building blcoks

Taking a closer look at the gateway device, its hardware building blocks are depicted by the following figure. Due to the properties of the BLE specification and the powerful processing unit of the gateway, multiple connections with sensor nodes can be maintained. Additionally, an 8x8-LED-Matrix can be used to signal state changes of the device, thus attracting attention in case of errors or warnings. On the gateway device the secure element has been used to enable secure communication to the sensor nodes and different server instances, i.e.:

- Inject & retrieve certificates and confidential data: This is important for enabling mutual authentication on transport layer when communicating online to the Secured Business Logic Layer. The Mutual Transport Layer Security (TLS) protocol is applied in this case. Furthermore, the SE050 is used for enabling a secure online channel to the update/commissioning service and storing confidential data packages.
- Random Number Generator (RNG): A true random number generator enables secure creation of different key material.

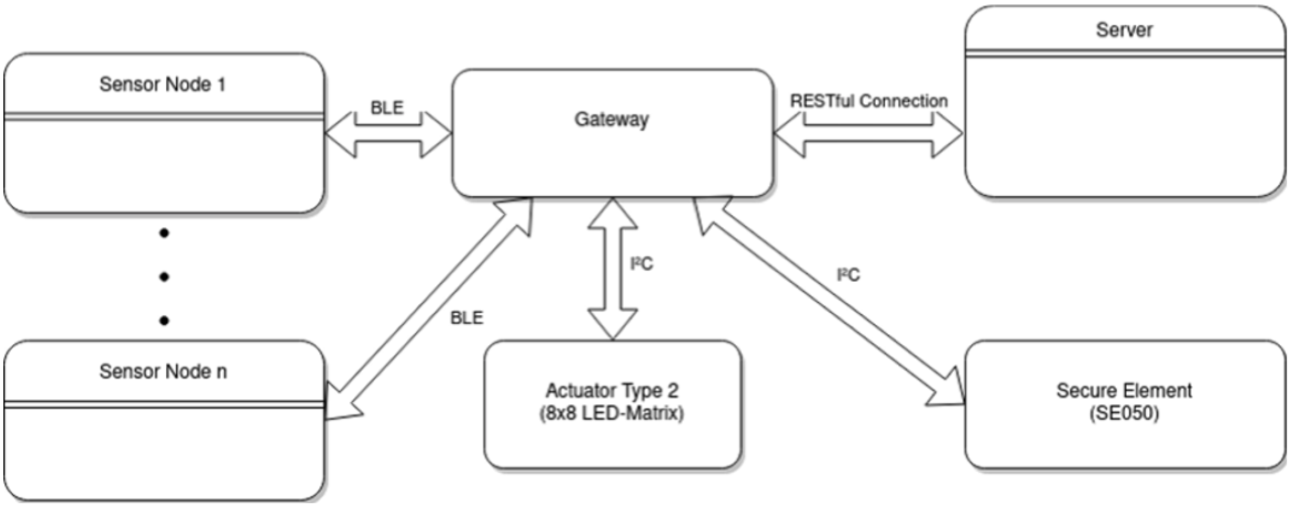


Figure 43: Gateway interaction and interfaces

After the integration process, CISC defined and developed the commissioning methodology as well as the maintenance access and personalized process methods. The presented approach includes the SE050 module as a secure key management unit, with a fully automated just-in-time commissioning of devices using the Amazon Webservices (AWS) IoT toolkit. Moreover, an MQTT message broker was used to facilitate application-specific communication. While authentication is achieved via X.509 certificates, traffic was sent securely over Transport Layer Security (TLS). To prevent man-in-the-middle attacks, a central authority certificate is checked upon connection to the MQTT broker. AWS, with its IoT Core, was the vendor of choice for the broker implementation. It provides state of the art security as well as a convenient interface to manage fleets of IoT devices.

11.2.4 Application to use cases

The proposed solution will be applied to UC3 – cloud-features battery management system. Together with AVL and TU Graz the secure wireless end-to-end communication (including the secure element) is applied to the setup prepared by the partners. CISC's solution will provide security and privacy (concerning user data) protection from the sensor to the cloud.

11.3 TPM2.0 based E2E security

11.3.1 Overview, motivation

End-to-end (E2E) security is a mechanism that ensures guarded and secure communication between two endpoints preventing any third parties from accessing the exchanged data, usually being critical data. All the data protection is being implemented directly on the end device. There is no additional medium interfering in the whole process. The data that is meant to be protected gets encrypted on the sender's side. Only the recipient of those messages has the ability to decrypt the encoded data stream and read the information. The messages, while being transferred between those two entities cannot be intercepted, and later read or tampered with by anyone else.

11.3.2 State of the art

In most applications involving data transmission, a secure transmission channel must be developed. Most cases are based on generating key pairs (public and private) and storing the private keys on the server's side during the negotiations of the communication. That way, the sender encrypts the message and sends it to the server. The server uses the sender's private key to make it readable and forwards it further to the recipient. This approach however allows the server to know the context of the message making it insecure in case of a direct attack towards the server. Some third parties that get access to the server may intercept them and read their contents. Given that the data is the most vulnerable when stored on a proxy server, hacking techniques are focused on gaining access to them as they are the weakest link in this approach.

End-to-end encryption uses an asymmetric keys approach. All the cryptographic keys used in encrypting and decrypting procedures are being stored on both communicating devices, not the server. The public key, which can be shared, is used to encrypt a message. The encrypted data can be decrypted only by the private key saved on the receiver side. No other key has the ability to make the data readable. The E2E encryption protects against reading messages that are in transit. Only the sender and receiver have the keys necessary for cryptographic operations. Even if the message that is being sent is visible in the network on its way, it can't be read directly. Additionally, using E2E encryption prevents from tampering and altering the data before it reaches the recipient. Any changes in the data stream may make it corrupted.

The most popular use of E2E encryption is implemented in the most popular messaging platforms. It allows their users for secure communication in their private chats. The keys used to encrypt and decrypt the messages, and hence the plain readable messages are being stored only on the devices. Platform's servers do not know what users are sending between each other, only the fact of active transmission can be noticed without detailed information regarding the context of those messages or any details.

To ensure the security on the highest level the most up-to-date algorithms must be used. That way, the attempts of calculating the secret keys will not succeed due to the huge computational resources required, as well as enormous time needed for performing these calculations. Encryption and decryption procedures are being handled only on the endpoints, not on any proxy server. Using the Diffie-Hellman key exchange method a common private shared key can also be generated and stored only on both the endpoints which can be used further to perform symmetric cryptographic operations. That way, the public key used for encrypting the data, which is public by default in asymmetric cryptography, is no longer used.

11.3.3 Innovation step

Connecting the vehicle's hardware to the cloud services impacts the privacy and security of transmitted data. All sent information shall be protected during the communication between both endpoints. To address those concerns and improve the system's innovation, as described in deliverable D9 (D3.2), DBox with TPM module shall be introduced to enhance the privacy of the transmitted data.

11.3.4 Application to use cases

The battery management system used in the TRANSACT project is meant to be developed and carried over into a distributed system. This system has a permanent accessible remote connection to the cloud. Exposing this solution outside creates room for possible attack surfaces. In order to protect the transmitted data, a secure solution managing the communication must be introduced. Both endpoints involved in data transmission, that is a vehicle and the cloud, need to establish a protected point-to-point medium. Vehicle data including telemetric statistics and personal information cannot be accessible by any third parties. Pre-processing of the data in order to keep confidentiality during the transmission period can be achieved by implementing end-to-end encryption into the final solution.

12 Security Solutions for New Services

12.1 Security-by-Contract Frameworks

12.1.1 Overview, motivation

The Internet of Things (IoT) and Cyber Physical System (CPS) revolutionised how devices and human beings cooperate and interact. The interconnectivity and mobility brought by IoT devices led to extremely variable networks, as well as unpredictable information flows. In turn, security proved to be a serious issue for the IoT, far more serious than it has been in the past for other technologies. IoT devices need detailed descriptions of their behaviour to achieve secure default configurations, sufficient security configurability, and self-configurability. The insecure default configuration problem states that IoT devices are routinely shipped with poor configurations concerning cybersecurity best practices. The insufficient security configurability problem states that current IoT devices do not offer enough tools for configuring them according to users' necessities.

IoT devices do not provide tools for defining and announcing their behaviour within a network, which prevents from regulating complex interactions between actors, whether they are human beings or devices. For example, an administrator might want to allow access to an IoT camera live-stream only to devices that do not communicate over the Internet. To achieve self-configurability, IoT devices need to be aware of the surrounding environment in terms of devices and services. However, the answer cannot come from IoT devices' direct observation of network traffic. Direct observation entails data extrapolation and data analysis, and it requires a considerable amount of time and computing power, unavailable to most IoT devices. Therefore, behavioural descriptions are of paramount importance for allowing IoT devices to self-configure and cooperate. From this point of view, this issue is strongly related both to the insufficient security configurability problem and the insecure default configuration one.

12.1.2 State of the art

Matheu et al. (Sara N. Matheu-García, 2019) highlighted that manufacturers should be included in the loop for creating more resilient devices. The authors proposed a certification methodology that delivers a measurable evaluation of IoT devices security, as well as an automatic security assessment. Moreover, they noted the lack of an IoT vulnerability database, which would enable better automatic security tests. Once these problems are amended, manufacturers could include in the contracts useful information about compliance with security standards. As an example, a contract could include the last time the device software was verified against the vulnerability database, or the security level assigned by the automatic evaluation tool.

Kuusijärvi et al. (Jarkko Kuusijärvi, 2017) proposed to strengthen IoT security through a network edge device (NED), a secure device that stores the user-defined policies and enforces them on resource-constraint IoT devices. MUD (Weis, 2016) is an IETF specification (RFC8520) in which manufacturers specify which hosts and ports their devices need to operate correctly. Matheu et al. (Sara N. Matheu, 2019) proposed an extension to the MUD model, going beyond communication restrictions at the network level and considering other factors, such as cryptographic algorithms and keys size. Hamza et al. (Ayyoob Hamza, 2018) tried to undertake the problem of enforcing policies using a combination with a software-defined network (SDN). The state-of-the art summary of existing security by contracts is shown in Table 3.

	MUD	W3C TD	Kuusijärvi et al.	Matheu et al	BACnet	Hamza et al.
Feasibility	Existing implementations	Relevant effort for producer	Relevant effort for producer	Based on original MUD feasibility	Already deployed for HVACs	Based on original MUD feasibility
Admin experience	Intuitive semantics	Complex semantics	Relevant effort From end users	Intuitive semantics	Somewhat intuitive semantics	Intuitive semantics
End-user experience	Transparent	Transparent	Transparent	Transparent	Transparent	Transparent
Compatibility with legacy devices	✗	✗	✓	✗	✗	✓
Human readability	JSON	JSON	✗	JSON	Clear text specification	JSON
Behavioural description	MUD file	TD file	✗	MUD file	BACnet file	MUD file
Detection of privacy leaks	End-to-end focus	End-to-end focus	End-to-end focus	End-to-end focus	End-to-end focus	End-to-end focus
Detailed security properties	✗	✓	✗	✓	✗	✗
Reasoning on sets of devices	✗	✗	✗	✗	✗	✗

Table 3: S×C Framework Comparison with existing Related Works.

12.1.3 Innovation step

The S×C (Security-by-contract) framework is based on two fundamental concepts, the security contract and the security policy. A security contract (or simply, a contract) specifies an IoT device's behaviour for what concerns relevant security actions. Every S×C-compliant device stores a contract and exhibits it to the network before being allowed to participate. The manufacturers create the S×C contracts for their IoT devices (Stage 2A in Figure 44: Security by contract main phases overview.).

Similarly, a security policy (or simply, a policy) specifies the acceptable behaviour of the IoT devices concerning their relevant security actions. A trustworthy device stores a policy within a network, such as an edge node, to verify that IoT devices' behaviour complies with the security policy. We refer to the process of verifying a contract against a policy as contract/policy matching. Table 3 shows S×C4IoT Framework, main phases overview. Box A encompasses the basic components necessary for an S×C framework and includes Stages 2A, 3, 4, and 5. Box B, composed of Stage 2B and 2C, describes two techniques that can be implemented for allowing S×C-noncompliant devices to be included in S×C4IoT. These techniques are not critical for the base functionalities of S×C4IoT, and different techniques for identifying IoT devices could be similarly used. Last, Box C corresponds to Stage 6 and details the algorithms for managing IoT device dynamic evolution.

For providing the core features presented in Figure 44, an S×C4IoT framework should provide certain minimum services. For example, the framework must be able to assess a triplet <Software, Contract, PoC> validity and verify a device contract against a policy. Moreover, the framework should provide every necessary service for managing foreseen dynamic variations.

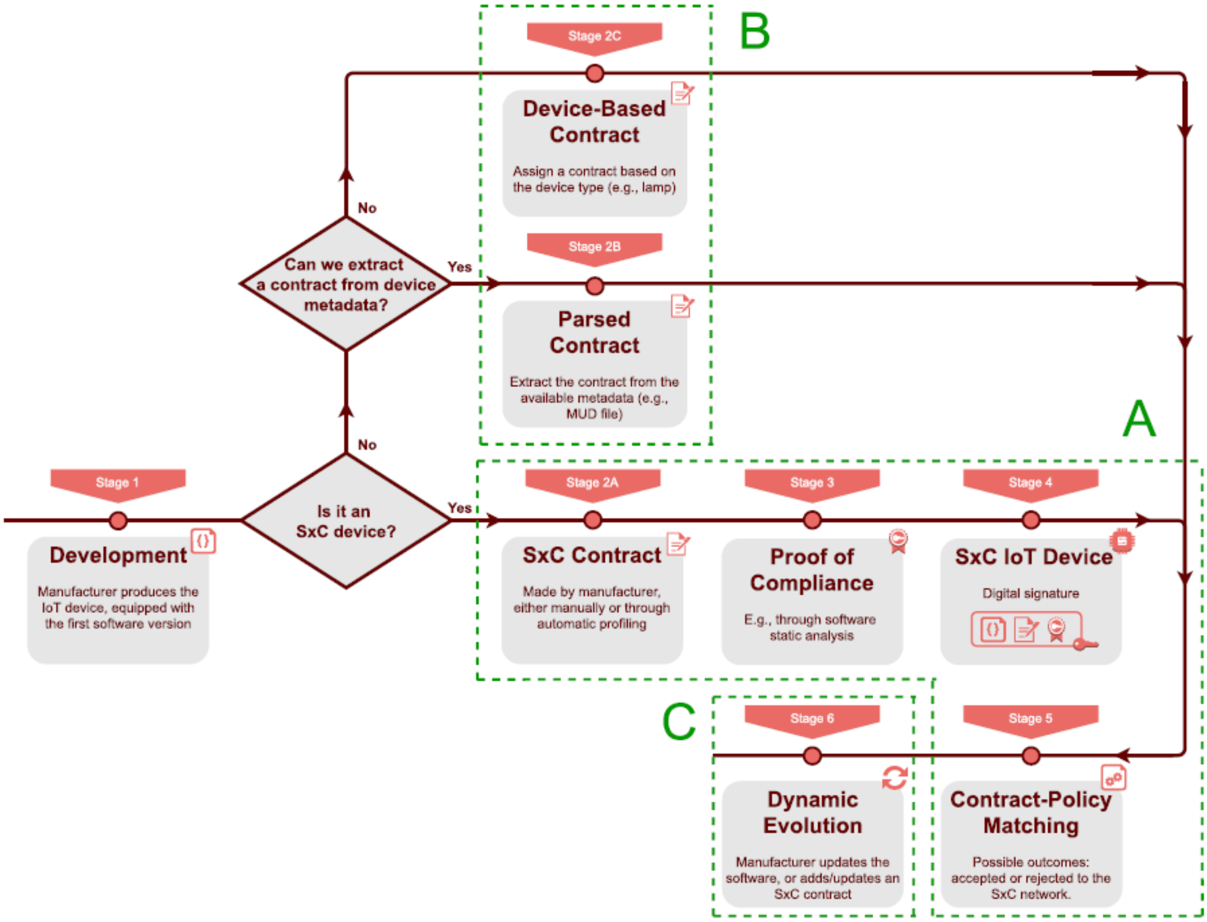

B
Stage 2C
Device-Based Contract
Assign a contract based on the device type (e.g., lamp)
No
Can we extract a contract from device metadata?
Yes
Stage 2B
Parsed Contract
Extract the contract from the available metadata (e.g., MUD file)
A
No
Stage 1
Development
Manufacturer produces the IoT device, equipped with the first software version
Is it an SxC device?
Yes
Stage 2A
SxC Contract
Made by manufacturer, either manually or through automatic profiling
Stage 3
Proof of Compliance
E.g., through software static analysis
Stage 4
SxC IoT Device
Digital signature
C
Stage 6
Dynamic Evolution
Manufacturer updates the software, or adds/updates an SxC contract
Stage 5
Contract-Policy Matching
Possible outcomes: accepted or rejected to the SxC network.

Figure 44: Security by contract main phases overview.

12.1.4 Validation results

Figure 45 represents the time overhead from applying S×C. The overhead grows with the number of rules per contract. This overhead does not happen because of larger size packets. The overhead is mostly since more security rules require more comparisons to establish whether a device complies with a network policy.

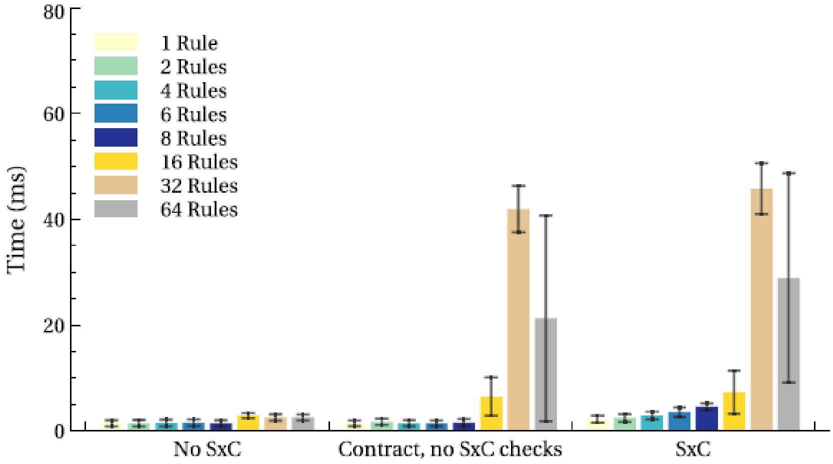


Figure 45: Average time overhead with growing number of rules per contract.

12.1.5 Application to use cases

The TRANSACT reference architecture addresses updating the system. Default insecure configurations and insufficient security configurability are significant challenges. Those services cooperate across tiers to securely perform remote automatic updates of the different device services. The updates ensure uniform software versions on the tiers and keep the system services up-to-date with the latest functionality. To achieve safe and predictable system updates, the following security by contract solution can be helpful.

13 Relation/interaction between solutions for safety, performance, security, and privacy

In this section, we explicitly focus on the positive relations and potential tensions between the solutions designed for improving/guaranteeing performance and safety and the technical requirements and relevant concepts for security and privacy.

Deliverable D3.2 (due in M12) has already provided an extensive analysis on the relevant technical requirements for security and privacy, also called TSRs (in Table 8 of D3.2), which is summarized here in Table 5. We start with the solutions presented in D3.3.

Solution	Title
S1	Mode change management on the device
S2	Mode change coordination
S3	Solutions for scalable applications
S4	Solutions for AI-monitoring
S5	Solutions for ensuring data integrity
S6	Performance observability and monitoring
S7	AI-based performance modeling and prediction
S8	Simulation-based performance analysis
S9	Performance analysis using formal methods
S10	Workflow simulation
S11	Scenario-based performance management and reconfiguration
S12	Risk management planning/monitoring
S13	Real-time machine-learning based solutions for detecting safety, security, and privacy anomalies
S14	Mapping and scheduling techniques across device, edge, and cloud
S15	Service continuity monitoring
S16	Solutions for dependable wireless communication
S17	Solutions for scalable platforms, and run-time scaling strategies

Table 4: List of solution items presented by the Deliverable D3.3

All the security- and privacy-related TSRs have been covered by the security and privacy concepts identified in Deliverable D3.2.

TSR No	Description	Relevant security concepts identified in D3.2 (see Table 6)
TSR 1	The TRANSACT system architecture should include protection and recovery mechanisms for data and centers for cloud services, and continuously protect data involved in transfers or transmissions	C2, C6-C15
TSR 2	The architecture should be protected against most attacks on edge computing infrastructures. This protection is mainly supposed to be against the following four categories: DDoS attacks, side-channel attacks, malware injection attacks, and authentication and authorization attacks.	C1, C5, C7, C14
TSR 3	The architecture should be protected against DDoS attacks.	C7, C14
TSR 4	The architecture should include effective solutions against flooding attacks and support the technique of detection and filtering.	C2, C7, C14, C15
TSR 5	The architecture should support packet-based detection aims to detect flooding-based attacks.	C2, C14
TSR 6	The architecture will support statistics-based approaches to detect DDoS attacks	C14
TSR 7	The architecture will be protected against zero-day attacks.	C14
TSR 8	The architecture should be protected against side-channel attacks.	C7, C14
TSR 9	The architecture will include components of a defence protection mechanisms suitable for data perturbation and differential privacy.	C5, C6, C12, C13, C14, C15
TSR 10	The architecture should be protected against malware Injection attacks.	C3, C4, C14
TSR 11	To counter the server-side injection attacks, the architecture will include detect-and-filter technique.	C14
TSR 12	The architecture will include components for defence against Device-Side Injections.	C3, C4
TSR 13	The architecture should be protected against Authentication and Authorization Attacks.	C1, C5, C6, C12, C15
TSR 14	The architecture will be protected against threats to Membership Inference Attacks.	C1, C4, C5, C6, C12
TSR 15	The architecture will be protected against Data Poisoning.	C1, C4, C5, C6, C12
TSR 16	The architecture will include components for defence against evasion attacks.	C1, C4, C6, C7, C12
TSR 17	The architecture will ensure the following security requirements: the confidentiality of permanently stored elements, executed-code authenticity, and run-time state integrity. The security architecture consists of four security mechanisms: security by separation, secure boot, secure key storage, and secure interdomain communication.	C4, C15

TSR 18	The cloud systems when used by the architecture should provide the details of how Use Case data will be handled, what types of security they already apply to the cloud infrastructure, what happens in case the system was compromised, if and how they will participate in the investigation and prosecution.	C5, C6, C9, C10, C11, C12, C15
TSR 19	The cloud systems used by the architecture should ensure that the data from the Use Cases is not shared with any third party.	C6, C8, C9, C10, C11, C12, C15
TSR 20	The cloud systems and their provider when used by the architecture should establish trust in the service offered to the Use Cases.	C8, C9, C10, C11, C12, C15
TSR 22	If used in the Use Case, an edge device in the architecture, will be secured on the basis of two factors: (1) root of trust (RoT), in which the edge device is unclonable in addition to the integrity, nonrepudiation, and authenticity of the running software at edge devices; and (2) chain of trust (CoT), in which the edge device is designed to boot up only if cryptographically signed software by a trusted entity is first executed using public-key cryptography. In addition, the keys are stored in specialized secure hardware; this hardware is also responsible for verification and RoT processes.	C1, C4, C5

Table 5: Technical security requirements (from Table 8 in D3.2 and Section 6.9.2 of D1.2)

Concepts referred to in Table 5 are presented in Table 6.

Concept No	**Concept description**	**Applicable to**		
		Device	**Edge**	**Cloud**
C1	PKI Infrastructure	●	●	●
C2	Concept for risk analysis and management	●	●	●
C3	Runtime verification	●	●	●
C4	TPM2.0-based edge and device security	●	●	●
C5	Role-based access control rules at the business/design level	●	●	●
C6	Anonymization: prevent personal data leak		●	
C7	Security and privacy concepts for communication	●	●	
C8	Centralized Machine Learning with Decentralized Data	●	●	●
C9	Multi-cloud concept for cloud security posture management (CSPM)			●
C10	User and entity behavioural analytics (UEBA) concept for cloud security			●
C11	Cloud detection & response (Cloud DR) orchestration for multiple clouds			●
C12	Security and privacy concepts for cloud-based applications			●

C13	Safety and privacy of off-the-shelf components, including MQTT, non-doubled 4G networks, open IP networks, Unity3D			●
C14	Security and privacy concepts for secure remote driving operation	●	●	●
C15	Security and privacy requirements and patterns for the healthcare DICOM data and applications	●	●	●

Table 6: Relevant concepts for security and privacy (from Tables 6 and 7 of Deliverable D3.2)

In the rest of this section, we discuss potential contributions of the solutions developed in D3.3 to the technical security/privacy requirements and then investigate potential tensions between the concepts developed to achieve security/privacy and the performance and safety requirements.

13.1 Contributions of solutions for performance and safety to security/privacy requirements

Solutions for runtime monitoring. Detecting security threats and anomalies (e.g., via signatures of DDoS attacks) requires monitoring the system, in particular, when the system is distributed across edge and cloud continuum. D3.3, in particular, focuses on investigating monitoring tools/techniques that are capable of gathering a wide range of metrics from the system/application on the cloud and edge (S6). It also considers monitoring techniques for service continuity (availability) via S15. Thus, both S6 and S15 can be used for TSR3 to TSR6 (see Table 5).

Solutions for anomaly detection. Detecting several security attacks (such as DDoS and flooding attacks) requires analyzing output traces of the monitoring tools, as these attacks leave an obvious footprint on the end-to-end response-time of the applications. Therefore, runtime solutions that use data-driven methods to predict the end-to-end response time can also be used to distinguish normal patterns from abnormal ones. Namely, S7 and S13 can directly contribute to achieving TSR3 to TSR6.

Some attacks involve data manipulation (for example, those addressed by TSR15 or parts of TSR11 and TSR12 that relate to data-injection attacks). S4, which provides a solution to monitor the health and performance of AI-based applications, could be used to raise alarms when there is a significant change (deviation from the expected outcome/quality) of the AI applications because that could be a sign of data-injection attacks.

Solutions for providing isolation. Guaranteeing timing and performance requirements typically involves the use of isolation techniques (for example, to separate non-real-time applications from real-time applications when they execute on the edge or cloud). Solutions S11 and S14 provide facilities to apply such isolation when mapping different applications to the computing resources on the cloud and edge platforms. Later in D3.5 (due in M33), more resource-management policies will be investigated, which in turn will improve the diversity of solutions that provide isolation. Isolation can diminish the impact of DDoS attacks on certain services/applications (hence addressing TSR2, TSR3, TSR4, TSR5, and TSR6), reduce the risk of malware injection, side-channel attacks, and authentication/authorization attacks (hence addressing TSR2, TSR8, TSR10, TSR11, TSR12, TSR13, TSR15, and TSR17).

Solutions for ensuring data integrity. Clearly, solutions in S5 for ensuring data integrity directly contribute to TSR2, TSR8, TSR12, TSR15, which are all about integrity protection.

Solutions for a combined risk analysis for safety and security risks. S12, which provides a risk-analysis framework using MARGERIT methodology, directly focuses on the assessment of the impact of security and safety risks in a use case. Therefore, it relates to all the security requirements, though only some of them might be present in (or needed for) a certain use case.

The above is summarized in Table 7.

Solutions for performance and safety	**Technical security/privacy requirements**
S6 (performance observability and monitoring)	TSR3, TSR4, TSR5, TSR6 (detecting DDoS and flooding attacks)
S15 (service continuity monitoring)	TSR3, TSR4, TSR5, TSR6 (detecting DDoS and flooding attacks)
S7 (AI-based performance modeling and prediction)	TSR3, TSR4, TSR5, TSR6 (detecting DDoS and flooding attacks)
S13 (real-time machine-learning based solutions for detecting safety, security, and privacy anomalies)	TSR3, TSR4, TSR5, TSR6 (detecting DDoS and flooding attacks)
S4 (solutions for AI-monitoring)	TSR15, TSR11, TSR12 (detecting data-injection attacks)
S5 (solutions for ensuring data integrity)	TSR2, TSR8, TSR12, TSR15 (ensuring data integrity)
S11 (scenario-based performance management and reconfiguration)	TSR2, TSR3, TSR4, TSR5, TSR6, TSR8, TSR10, TSR11, TSR12, TSR13, TSR15, and TSR17 (isolating services/applications from each other)
S14 (mapping and scheduling techniques across device, edge, and cloud)	TSR2, TSR3, TSR4, TSR5, TSR6, TSR8, TSR10, TSR11, TSR12, TSR13, TSR15, and TSR17 (isolating services/applications from each other)
S12 (risk management planning/monitoring)	Relates to all security requirements

Table 7: Solutions for performance/safety that contribute to technical security/privacy requirements

13.2 Contributions of solutions/concepts for security and privacy to performance and safety requirements

Some of the concepts/solutions proposed in D3.2 (and realised in D3.4) may not only be useful to address security requirements but also performance and safety requirements.

Concepts/solutions for runtime verification. Typically, the goal of the runtime verification (Concept C3 in Table 6) is to ensure that the safety and security requirements of a system are met. If such a mechanism is in place, it will directly contribute to the detection of safety anomalies and triggering safeguards and mode changes, resulting in guaranteeing safety and performance/timing requirements. Consequently, C3 could also be seen as an alternative (or complementary) way to achieve the same goal as S11, S12, S13, and S15.

Similarly, C3 can be used to detect timing anomalies (e.g., when the response-time of an application becomes much larger than expected), hence can be used along with S7 and S13 (for detecting timing anomalies) and with S11, S12, S14, S17 (to trigger resource scaling strategies to meet timing requirements).

The above is summarized in Table 8.

Concepts/solutions for security and privacy	**Concepts/solutions for Safety/performance**
C3 (runtime verification)	S11, S12, S13, S15 (detecting safety anomalies and triggering mode changes or safeguards)
	S7 and S13 (to detect timing anomalies)
	S11, S12, S14, S17 (to trigger resource scaling strategies to meet timing requirements)

Table 8: Concepts/solutions for security/privacy that contribute to performance/safety requirements

13.3 Tensions between solutions/concepts for security/privacy and performance/safety

Next, we will look at the tensions between the concepts/solutions for security/privacy and performance/safety . These tensions are summarized in Table 9:

- They may add (timing) overheads to the system (e.g., to the resource-manager component, platform, network, or application), and therefore jeopardize performance criteria such as end-to-end response time due to the overheads.
- They may increase resource consumption (due to the execution of security/privacy solutions on the same computing resources).
- They may jeopardize safety or quality of service of an application (e.g., due to the use of anonymized or perturbed data, which may reduce the accuracy of the application, or may require a more complex application to be implemented).

Similarly, solutions to improve/guarantee performance and safety may have the following negative impacts on the technical security/privacy requirements (this is also summarized in Table 9):

- They may reduce isolation between trusted and not-trusted (or safety-critical and non-safety-critical) applications and hence increase the risk of side-channel attacks (for example, if the runtime resource management strategies are now security-aware).
- They may increase the attack surface for DDoS attacks (for example, in a case where the attacker can exploit the fact that in order to “degrade performance”, it is enough to trigger the fallback mechanism of the resource-management strategy, hence, instead of a full-fledged easy-to-detect DDoS attack, the attacker can design a stealthier attack with a small footprint).

Type of threats	**Negative impact (or tension)**	**Concepts or solutions that may cause the impact**
Threats to performance/safety	Add timing overhead and hence jeopardize timing requirements	C3, C5, C6, C7, C8, C9, C11, C12, C13, C15 (due to the overhead of runtime security/privacy enforcement mechanisms)
	Increase resource consumption (due to the execution of security/privacy solutions on the same computing resources)	C3, C5, C6, C7, C8, C11, C14, C15 (due to the execution of security/privacy enforcement mechanisms on the same platforms as the system)
	Jeopardize safety by impacting QoS (or functionality) of the system	C5, C6, C12, C13, C14, C15 (because of data perturbation and anonymization) C14 (because of the detect-and-filter technique)
Threats to security/privacy	Increase the risk of side-channel attacks	S11, S14, S17 (due to the lack of security awareness in resource management and mapping)
	Increase the attack surface for DDoS attacks	S1, S2, S11, S14, S17 (due to providing alarms or triggering mechanisms that may result in the activation of a degraded mode)

Table 9: Tensions between some of the security/privacy concepts and solutions for performance and safety

14 Conclusion

In this deliverable D3.4, we presented the first results of the TRANACT efforts on implementing the concepts and plans for security and privacy defined in D3.2. We also considered (i) connections and contributions of the solutions for performance and safety to the security and privacy requirements, (ii) contributions of the concepts for security and privacy to the performance and safety requirements, and (iii) potential conflicts and "tensions" between the solutions and concepts for security/privacy and performance/safety.

The final results of the TRANSACT efforts covered in this document (and a few additional lines of the work on security/privacy which are in early stages at the moment) will be presented in D3.6.

15 References

(W3C), World Wide Web Consortium. (2021, August 03). Decentralized Identifiers (DIDs) v1.0 – Core architecture, data model, and representations. W3C proposed recommendation. Retrieved April 28, 2022, from https://www.w3.org/TR/did-core/

Abera, T., Asokan, N., Davi, L., Ekberg, J., Nyman, T., Paverd, A., . . . Tsudik, G. (2016). C-FLAT: Control-FLow ATtestation for Embedded Systems Software. *In Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security.* Vienna, Austria,.

Abera, T., Bahmani, R., Brasser, F., Ibrahim, A., Sadeghi, A., & Schunter, M. (2019). DIAT: Data Integrity Attestation for Resilient Collaboration of Autonomous System. *In Proceedings of the 26th Annual Network & Distributed System Security Symposium.* San Diego, CA, USA.

AICPA. (2022, 10 1). *SOC for Service Organizations*. Retrieved from SOC for Service Organizations: https://us.aicpa.org/interestareas/frc/assuranceadvisoryservices/socforserviceorganizations

Ambrosin, M., Conti, M., Ibrahim, A., Neven, G., Sadeghi, A., & Schunter, M. (2016). SANA: Secure and Scalable Aggregate Network Attestation. *In Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security CCS '16.* Vienna.

Ambrosin, M., Conti, M., Lazzeretti, R., Rabbani, M., & Ranise, S. (2018). PADS: Practical Attestation for Highly Dynamic Swarm. *In Proceedings of the 2018 InternationalWorkshop on Secure Internet of Things (SIoT).* Barcelona, Spain.

Ankergård, S., & Dragoni, N. (2021). PERMANENT: Publicly Verifiable Remote Attestation for Internet of Things through Blockchain. *In Proceedings of the 14th International Symposium on Foundations & Practice of Security.* Paris, France.

Antonia M. Reina Quintero, S. M.-V. (2022). A domain-specific language for the specification of UCON policies. *Journal of Information Security and Applications.*

ASIP SANTÉ. (2022, 09 30). *HDS - Health Data Hosting Certification*. Retrieved from HDS - Health Data Hosting Certification: https://industriels.esante.gouv.fr/en/products-services/hds-health-data-hosting-certification

Asokan, N., Brasser, F., Ibrahim, A., Sadeghi, A., Schunter, M., Tsudik, G., & Wachsmann, C. (2015). SEDA: Scalable Embedded Device Attestation. *In Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security (CCS).* CO, USA,.

Ayyoob Hamza, H. H. (2018). Combining MUD policies with SDN for IoT intrusion detection. *Workshop on IoT Security and Privacy* (pp. 1--7). ACM.

Bendre, M. R. (2016). Analytics, challenges and applications in big data environment: a survey. *Journal of Management Analytics*, 206--239.

BlackDuck. (2022, 10 1). *Black Duck Software Composition Analysis*. Retrieved from Black Duck Software Composition Analysis: https://www.synopsys.com/software-integrity/security-testing/software-composition-analysis.html

Brambilla, M. a. (2017). Model-driven development of user interfaces for IoT systems via domain-specific components and patterns. *Journal of Internet Services and Applications*, 1--21.

Brambilla, M. a. (2017). *Model-driven software engineering in practice.* Morgan \& Claypool Publishers.

Brasser, F., El Mahjoub, B., Sadeghi, A., Wachsmann, C., & Koeberl, P. (2015). TyTAN: Tiny trust anchor for tiny devices. *In Proceedings of the 2015 52nd ACM/EDAC/IEEE Design Automation Conference (DAC),* (pp. 1-6). San Francisco, CA, USA: ACM.

BurpSuite. (2022, 10 1). *BurpSuite*. Retrieved from BurpSuite: https://portswigger.net/burp/enterprise

Cech, J. (2020). Chrysalis (IOTA 1.5) Phase 2. Update and next steps. Retrieved November 23, 2022, from https://blog.iota.org/chrysalis-iota-1-5-phase-2-update-and-next-steps-eecabe55d7bd/amp/

Chen Sun, A. H. (2004). Fast beamforming of electronically steerable parasitic array radiator antennas: theory and experiment. *IEEE Transactions on Antennas and Propagation, vol. 52, no. 7*, 1819-1832. doi:10.1109/TAP.2004.831314

Chun, B., Ihm, S., Maniatis, P., Naik, M., & Patti, A. (2011). CloneCloud: Elastic Execution between Mobile Device and Cloud. *In Proceedings of the Sixth European conference on Computer systems (EuroSys '11).* Salzburg.

Conti, M., Dushku, E., & Mancini, L. (2019). RADIS: Remote Attestation of Distributed IoT Services. *In Proceedings of the 6th IEEE International Conference on Software Defined Systems (SDS 2019).* Rome, Italy,.

Coverity. (2022, 10 1). *Coverity Static Application Security Testing*. Retrieved from Coverity Static Application Security Testing: https://www.synopsys.com/software-integrity/security-testing/static-analysis-sast.html

D2.1. (2022). *Reference architectures for distributed safety-critical distributed cyber-physical systems v1.* TRASACT.

D3.2, D. (2022). *D9 (D3.2) Selection of concepts for end-to-end security and privacy for distributed CPS solutions.* TRANSACT.

Dessouky, G., Zeitouni, S., Nyman, T., Paverd, A., Davi, L., Koeberl, P., . . . Sadeghi, A. (2017). LO-FAT: Low-Overhead Control Flow ATtestation in Hardware. *In Proceedings of the 54th Annual Design Automation Conference (DAC).* Austin, TX, USA.

DICOM. (2022, 10 1). *DICOM*. Retrieved from DICOM: https://www.dicomstandard.org/

Dushku, E., Rabbani, M., Conti, M., Mancini, L., & Ranise, S. (2020). SARA: Secure Asynchronous Remote Attestation for IoT Systems. *IEEE Trans. Inf. Forensics Secur.*, 3123–3136.

Edlira Dushku, J. H. (2022). Memory Offloading for Remote Attestation of Multi-Service IoT Devices. *Sensors*, 1-28.

Eldefrawy, K., Perito, D., & Tsudik, G. (2012). SMART: Secure and Minimal Architecture for (Establishing a Dynamic) Root of Trust. *In Proceedings of the Network and Distributed System Security Symposium (NDSS), ,* (pp. 1-15). San Diego, CA, USA.

FDA-CFR-Title-21. (2022, 09 30). *CFR - Code of Federal Regulations Title 21*. Retrieved from CFR - Code of Federal Regulations Title 21: https://www.accessdata.fda.gov/scripts/cdrh/cfdocs/cfcfr/CFRSearch.cfm?CFRPart=11&showFR=1

Fortify. (2022, 10 1). *Fortify-Application security*. Retrieved from Fortify-Application security: https://www.microfocus.com/en-us/cyberres/application-security

Green, M. (2018). Hash-based signatures: An illustrated primer. Retrieved November 23, 2022, from https://blog.cryptographyengineering.com/2018/04/07/hash-based-signatures-an-illustrated-primer/

Guo, H. a. (2010). RBAC-based access control integration framework for legacy system. *International Conference on Web Information Systems and Mining* (pp. 194--201). Springer.

Hernández, B. Q. (2021). Automatic Code Generation of Data Visualization for Structural Health Monitoring. *IEEE Latin America Transactions*, 1041--1050.

HITRUST Alliance. (n.d.). *HITRUST CSF*. (HITRUST Alliance) Retrieved from https://hitrustalliance.net/csf-license-agreement/

IOTA Foundation. (2022). IOTA overview. Retrieved November 23, 2022, from https://wiki.iota.org/learn/about-iota/an-introduction-to-iota

Jarkko Kuusijärvi, R. S. (2017). Mitigating IoT security threats with a trusted Network element. *11th International Conference for Internet Technology and Secured Transactions (ICITST)* (pp. 260–265). Barcelona, Spain: IEEE.

Klockwork. (2022, 10 1). *Klocwork: Best Static Code Analyzer for Developer Productivity, SAST, and DevOps/DevSecOps*. Retrieved from Klocwork: Best Static Code Analyzer for Developer Productivity, SAST, and DevOps/DevSecOps: https://www.perforce.com/products/klocwork

Koeberl, P., Schulz, S., Varadharajan, V., & Sadeghi, A. (2014). TrustLite: A Security Architecture for Tiny Embedded Devices. *In Proceedings of the Ninth European Conference on Computer Systems (EuroSys), .* Amsterdam, The Netherlands.

Kronfellner B., M. T. (2021). Me, myself and (SS)I. Retrieved April 28, 2022, from https://web-assets.bcg.com/6b/6d/84e00cad4c939c870d833b96321c/white-paper-me-myself-ssi.pdf

Ledur, C. a. (2015). Towards a domain-specific language for geospatial data visualization maps with big data sets. *2015 IEEE/ACS 12th International Conference of Computer Systems and Applications (AICCSA)* (pp. 1--8). IEEE.

Li, H. a. (2015). A survey of extended role-based access control in cloud computing. *Proceedings of the 4th International Conference on Computer Engineering and Networks* (pp. 821--831). Springer.

Liu X, F. B. (2020). Distributed ledger technology. In K. C. Farshad Firouzi, *Intelligent Internet of Things* (pp. 393-431). Springer. doi:10.1007/978-3-030-30367-9

Mendling, J. a. (2004). An approach to extract RBAC models from BPEL4WS processes. *13th IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises* (pp. 81--86). IEEE.

Microsoft SDL. (2022, 10 1). *Microsoft Security Development Lifecycle*. Retrieved from https://www.microsoft.com/en-us/securityengineering/sdl

Microsoft_STRIDE. (2022, 09 30). *STRIDE*. Retrieved from https://en.wikipedia.org/wiki/STRIDE_(security)

Mouelhiv, T. a. (2008). A generic metamodel for security policies mutation. *2008 IEEE International Conference on Software Testing Verification and Validation Workshop* (pp. 278--286). IEEE.

Nakamoto, S. (2009). Bitcoin: A Peer-to-Peer Electronic Cash System. Retrieved November 22, 2022, from https://bitcoin.org/bitcoin.pdf

Nguyen, P. H. (2013). Model-driven adaptive delegation. *Proceedings of the 12th annual international conference on Aspect-oriented software development*, (pp. 61--72).

NIST_CVSS. (2022, 09 30). *The Common Vulnerability Scoring System (CVSS) and Its Applicability to Federal Agency Systems*. Retrieved from The Common Vulnerability Scoring System (CVSS) and Its Applicability to Federal Agency Systems: https://nvlpubs.nist.gov/nistpubs/Legacy/IR/nistir7435.pdf

NIST-SP800-37. (2022, 09 30). *Risk Management Framework for Information Systems and Organizations: A System Life Cycle Approach for Security and Privacy*. Retrieved from Risk Management Framework for Information Systems and Organizations: A System Life Cycle Approach for Security and Privacy: https://csrc.nist.gov/publications/detail/sp/800-37/rev-2/final

NIST-SP800-53. (2022, 10 1). *Security and Privacy Controls for Information Systems and Organizations v5*. Retrieved from Security and Privacy Controls for Information Systems and Organizations: https://csrc.nist.gov/publications/detail/sp/800-53/rev-5/final

PenTesting. (2022, 10 1). *Penetration test*. Retrieved from Penetration test: https://en.wikipedia.org/wiki/Penetration_test

PenTesting Microsoft. (2022, 10 1). *Perform Penetration Testing*. Retrieved from Perform Penetration Testing: https://www.microsoft.com/en-us/securityengineering/sdl/practices#practice11

PrivacyByDesign. (2022, 10 01). *Privacy by Design*. Retrieved from Privacy by Design: https://www.ipc.on.ca/wp-content/uploads/Resources/7foundationalprinciples.pdf

Rose S., B. O. (2020). Zero Trust Architecture. NIST special publication 800-207. doi:10.6028/NIST.SP.800-207

Salvador Martínez, A. F. (2018). Automatic generation of security compliant (virtual) model views. *International Conference on Conceptual Modeling* (pp. 109-117). Springer.

Sara N. Matheu, J. L.-R. (2019). Extending MUD profiles through an automated IoT security testing methodology. *IEEE Access*, 149444–149463.

Sara N. Matheu-García, J. L.-R. (2019). Risk-based automated assessment and testing for the cybersecurity certification and labelling of IoT devices. *Comput. Stand. Interf.*, 64–83.

Seshadri, A., Perrig, A., Luk, M., Van Doom, L., Shi, E., & Khosla, P. (2005). Pioneer: Verifying code integrity and enforcing untampered. *Oper. Syst. Rev*, 1-16.

Seshadri, A., Perrig, A., van Doorn, L., & Khosla, P. (2004). SWATT: SoftWare-based attestation for embedded devices. *In Proceedings of the IEEE Symposium on Security and Privacy.* Berkeley, CA, USA: IEEE.

Smeltzer, K. a. (2018). A domain-specific language for exploratory data visualization. *Proceedings of the 17th ACM SIGPLAN International Conference on Generative Programming: Concepts and Experiences*, (pp. 1--13).

Sönmez, F. Ö. (2018). Evaluation of security information and event management systems for custom security visualization generation. *2018 International Congress on Big Data, Deep Learning and Fighting Cyber Terrorism (IBIGDELFT)* (pp. 38--44). IEEE.

Sovrin. (2020, August). Self-sovereign Identity and IoT. Sovrin foundation SSI in IoT task force. Retrieved April 28, 2022, from https://sovrin.org/wp-content/uploads/SSI-in-IoT-whitepaper_Sovrin-design.pdf

Trust over IP Foundation. (2021, November 17). Introduction to Trust Over IP. Version 2.0. Retrieved November 22, 2022, from https://trustoverip.org/wp-content/uploads/Introduction-to-ToIP-V2.0-2021-11-17.pdf

U.S. Department of Health and Human Services. (1996). *Health Insurance Portability and Accountability Act of 1996*. Retrieved from https://aspe.hhs.gov/reports/health-insurance-portability-accountability-act-1996

WebInspect. (2022, 10 1). *Fortify WebInspect*. Retrieved from Fortify WebInspect: https://www.microfocus.com/en-us/cyberres/application-security/webinspect

Weis, E. L. (2016). Slinging MUD: Manufacturer usage descriptions: How the network can protect things. *In International Conference on Selected Topics in Mobile Wireless Networking (MoWNeT).* (pp. 1--6). Cairo, Egypt: IEEE.

ZAP. (2022, 10 1). *OWASP Zed Attack Proxy (ZAP)*. Retrieved from OWASP Zed Attack Proxy (ZAP): https://www.zaproxy.org/

Zhang, G. a. (2010). An extended role based access control model for the Internet of Things. *2010 International conference on information, networking and automation (ICINA)* (pp. V1--319). IEEE.