Grant Agreement N°: 101020259

Topic: SU-DS02-2020

ARCADIAN-IoT

Autonomous Trust, Security and Privacy Management Framework for IoT

D5.2: Integrated ARCADIAN-IoT framework – final version

Revision: v.1.0

Work package	WP 5
Task	Task 5.1
Due date	31/01/2024
Submission date	08/02/2024
Deliverable lead	IPN
Version	1.0
Partner(s) / Author(s)	IPN: Paulo Silva (co-editor), Sérgio Figueiredo (co-editor), Rúben Leal, Fernando Bastos UWS: Jose M. Alcaraz Calero, Qi Wang, Antonio Matencio Escolar, Ignacio Martinez-Alpiste, Pablo Benlloch Caballero, Julio Diez-Tomillo 1GLOBAL: João Casal, Afonso Paredes, Ivo Vilas Boas XLAB: Benjamin Benčina, Jan Antić, Nejc Bat UC: Bruno Sousa, Luis Paquete, João Nunes MARTEL: Giacomo Inches, Gabriele Cerfoglio BOX2M: Alexandru Gliga, Ovidiu Diaconescu, Marian Macoveanu RISE: Alfonso Iacovazzi, Han Wang ATOS: Ross Little RGB: Ricardo Ruiz LOAD: Pedro Colarejo

Abstract

This report documents deliverable D5.2 of ARCADIAN-IoT, a Horizon 2020 project with the **grant agreement number 101020259**, under the topic **SU-DS02-2020**. The main purpose of the current report is to present the **final prototype (P2)** of ARCADIAN-IoT framework, which demonstrates **ARCADIAN-IoT functionalities -** resulting from the **integration** between the components of the framework's Horizontal and Vertical planes – and its readiness for enabling the project's use cases.

Keywords: ARCADIAN-IoT Framework; Implementation; Integration, Identity, Trust, Privacy, Security, Recovery; secure IoT services

Document Revision History

Version	Description of change	List of contributors
V0.1	Table of Contents, highlighting of sections requiring inputs, abstract and executive summary	IPN
V0.2	Inputs for technical sections (component and domain integration)	All partners
V0.3	New chapter on architecture overview and split from integration approach	IPN
V0.4	Inputs provided for all sections. Pending updates for figures, tables and minor edits on section	All partners
V0.5	Consortium expert review	ATOS
V0.6	Updates addressing comments from consortium expert review	IPN, all
V0.7	Figure updates, resolving comments, setting headers and numbering	IPN, all
V0.8	Re-structuring deployment view based on comments, added missing descriptions on component integration	IPN
V1.0	Final Document	IPN

Project co-funded by the European Commission under SU-DS02-2020		
Nature of the deliverable:	OTHER*	
Dissemination Level		
PU	Public, fully open, e.g., web	√
CI	Classified, information as referred to in Commission Decision 2001/844/EC	
CO	Confidential to ARCADIAN-IoT project and Commission Services	

* *R: Document, report (excluding the periodic and final reports)*
DEM: Demonstrator, pilot, prototype, plan designs
DEC: Websites, patents filing, press & media actions, videos, etc.
OTHER: Software, technical diagram, etcba

EXECUTIVE SUMMARY

ARCADIAN-IoT framework proposes an integrated approach for managing identity, trust, privacy, security and recovery, across IoT devices, persons and services, relying on specialised components distributed across Vertical and Horizontal planes. The vertical planes cover Identity, Trust and Recovery management, while the horizontal planes – on which the vertical planes rely – are responsible for managing Privacy, Security and other functionalities (i.e., encryption and Blockchain mechanisms).

This report documents the final version of ARCADIAN-IoT framework's pilot: **P2 – Prototype 2**. P2 provides a complete and functional version of **ARCADIAN-IoT functionalities** and builds on features developed and provided by each component from its Horizontal and Vertical planes, their intra- and inter-plane integration – Component x Component integration work -, and the necessary tailoring and testing for enabling their intended functionality within the targeted use cases – Component x Domain integration work.

The document presents the adopted integration approach and plan, split in the two integration dimensions (component x component and component x domain), and the final integration status, including performed tests and integration validation experiments. The report on the use cases end-to-end validation and legal compliance focusing on P2 functionalities will be delivered in M36.

The final prototype P2 characterization delivered in this document is the main outcome of **Task 5.1 (Integration of ARCADIAN-IoT framework)**. For integration activities previously performed for prototype P1 (provided in D5.1), it provides revised or extended descriptions**.** The delivered material additionally builds both on the concluded research and implementation of each ARCADIAN-IoT component (addressed in WP3 and WP4) and the preparation and implementation of the use cases (Tasks 5.2, 5.3 and 5.4). To ensure a comprehensive and thorough approach, all technical and domain owners were involved in this iterative process of integrating ARCADIAN-IoT framework for ensuring the successful integration in the domains.

LIST OF FIGURES

LIST OF TABLES

ABBREVIATIONS

3PP	3rd Party Platform
ABE	Attribute Based Encryption
AI	Artificial Intelligence
AIDS	Anomaly Intrusion Detection System
BLE	Bluetooth Low Energy
CBOR	Concise Binary Object Representation
CoT	Chain of Trust
CTI	Cyber Threat Intelligence
DB	Database
DDoS	Distributed Daniel of Service
DID	Decentralized Identifiers
DLT	Distributed Ledger Technologies
eSIM	Embedded Subscriber Identity Module
eUICC	Embedded Universal Integrated Circuit Card
FE	Functional Encryption
FL	Federated Learning
GSMA	Global System for Mobile communications Association
HIDS	Host Intrusion Detection Systems
HE	Hardened Encryption
IDPS	Intrusion Detection and Prevention System
ICS	Industrial Control Systems
IDS	Intrusion Detection System
IETF	Internet Engineering Task Force
IoC	Indicator of Compromise
IoT	Internet of Things
IPR	Intellectual Property Rights
IT	Information Technologies
KPI	Key Performance Indicator
MIoT	Medical IoT
MISP	Malware Information Sharing Platform
ML	Machine Learning
NFM	Network Flow Monitoring
NSH	Network Self-Healing

NSP	Network Self-Protection
OTA	Over-the-Air
OWASP	Open Web Application Security Project
PCA	Protection Control Agent
PII	Personally Identifiable Information
R&I	Research & Innovation
RATS	Remote Attestation Procedures
RIA	Resource Inventory Agent
RoT	Root of Trust
RSP	Remote SIM Provisioning
SE	Secure Element
SHDM	Self-Healing Decision Manager
SIDS	Signature Intrusion Detection System
SIM	Subscriber Identity Module
TCP	Transmission Control Protocol
VDR	Verifiable Data Registry
W3C	World Wide Web Consortium

1 INTRODUCTION

This section lays out the objectives and assumptions for this deliverable and related outcome for the final Prototype P2, provides key background information on the ARCADIAN-IoT framework, and presents the overall document structure and organization.

1.1 Objectives and assumptions

The overall goal for final prototype P2, described in this document, has been to enable the complete availability and high-granularity integration and testing of each of the ARCADIAN-IoT components functionalities in experimental environments corresponding to the needs of diverse IoT applications.
As such, P2 builds on the preliminary functionalities integrated and made available in P1, consolidating them (e.g. Trust Plane, Identity Plane), and adding a significant number of relevant features (e.g. those relating to the Recovery plane).

This deliverable builds upon several different inputs, across three Work Packages, and incorporating the collaboration amongst all the consortium partners. First and foremost, it stems from the use cases specification and planning (reported in D2.2 [1]), the elicitation of the ARCADIAN-IoT framework requirements (reported in 1and the specification of the architecture of ARCADIAN-IoT framework (documented in D2.4 [2]), all performed within WP2. Additionally, P2 builds on the functionalities implemented by each component taking part in the Horizontal and Vertical planes responsible for providing Privacy, Security, Identity, Trust and Recovery management capabilities (detailed in D3.3 [3] and D4.3 [4], from WP3 and WP4 respectively). Finally, P2 is also enabled via the applications, hardware and other artefacts provided (i.e. produced from scratch or adapted) by each of the domains, which interact or integrate with ARCADIAN-IoT framework (to be reported in D5.5 by M36).

1.2 Integration planning methodology

To ensure the success of the integration activities, which require functional prototypes of the technical components, there was a considerable effort in dividing the target features with a subset within the first version of the ARCADIAN-IoT framework (P1), already reported in the previous deliverable D5.1, and the remaining integration activities that were executed for the final version (P2), in the context of the current deliverable. The associated effort took place within the integration task (T5.1) and has involved all consortium partners, i.e., technical partners, domain owners and legal experts. The complete integration planning process methodology was essentially the following:

- A plan regarding the availability of **ARCADIAN-IoT functionalities and interfaces** made available by each component (resulting from the planned results within WP3/WP4) was defined
- Based on the above, a plan regarding the required **ARCADIAN-IoT intra- and inter-plane integration** was established; such plan was performed based on the agreement and coordination between partners, partner-specific priorities - especially for partners responsible for multiple components, each with multiple interfaces with other ARCADIAN-

IoT components – and project-wide priorities, such as integrating first components / features with presence in P1 Use Cases, and then with components / features with presence in P2 Use Cases.

- Similarly, a **Domain integration readiness** plan was established. Such plan included two dimensions. The first dimension was the expected “Domain artifact” plan for integrating ARCADIAN-IoT, reflecting Domain owner’s expected readiness for the availability of artifacts (e.g. Applications, hardware and others) for each use case. This greatly depended on the domain owners’ alignment / coordination with technical partners responsible for components involved in their respective domain, mostly for obtaining sufficient awareness to ARCADIAN-IoT impact (i.e., spanning both benefits, resources overhead, integration requirements). The second dimension, based on the “Domain artifact” plan, was the expected integration readiness for each use case from the point of view of the technical partners responsible for ARCADIAN-IoT features. Such readiness was overall relative to ARCADIAN-IoT components’ tailoring and testing for enabling all P1 and P2 use cases defined within the project, equally dependent on alignment between component’s owners and domain owners.

The different elaborated integration plans were continuously monitored against the actual integration status during frequent (at least monthly) T5.1 meetings, which was necessary for the identification and resolution of integration issues, prioritization of integration activities, and the adjustment of the integration timings.

1.3 Document structure

The remainder of this document is presented as follows:

Section 2 provides background information regarding ARCADIAN-IoT architecture, the overview of features planned for P1 vs P2, as well as a characterization of the Chain of Trust for IoT enabled by the framework.

Section 3 details the ARCADIAN-IoT framework integration approach, which is organized according to component-to-component and component-to-domain points of view. The supporting tools are also presented in this section.

Section 4 presents the component integration status. It demonstrates the integration status of the components in the horizontal planes (Privacy, Security and Common) and vertical planes (Identity, Trust and Recovery).

Section 5 documents the current readiness of each of the domains’ use cases, and the status of ARCADIAN-IoT framework integration towards enabling them.

Section 6 concludes the document with a summary of the main conclusions of the results achieved so far and provides an overview of the next activities, that include overall ARCADIAN-IoT framework integration and validation.

2 ARCHITECTURE AND PROTOTYPE OVERVIEW

This section recaps the ARCADIAN-IoT framework in which P2 builds upon, provides a high-level overview of P1 vs P2 functionalities, and describes the Chain of Trust provided by the ARCADIAN-IoT Framework.

2.1 Recap on ARCADIAN-IoT architecture

The overall architecture, including conceptual and deployment views, was initially defined in D2.5 [2]. It has been iteratively improved and adjusted as a result of the progress in the research, development and integration activities, visible in some modifications in the components' interfaces and interactions. Figure 1 depicts the updated ARCADIAN IoT architecture.

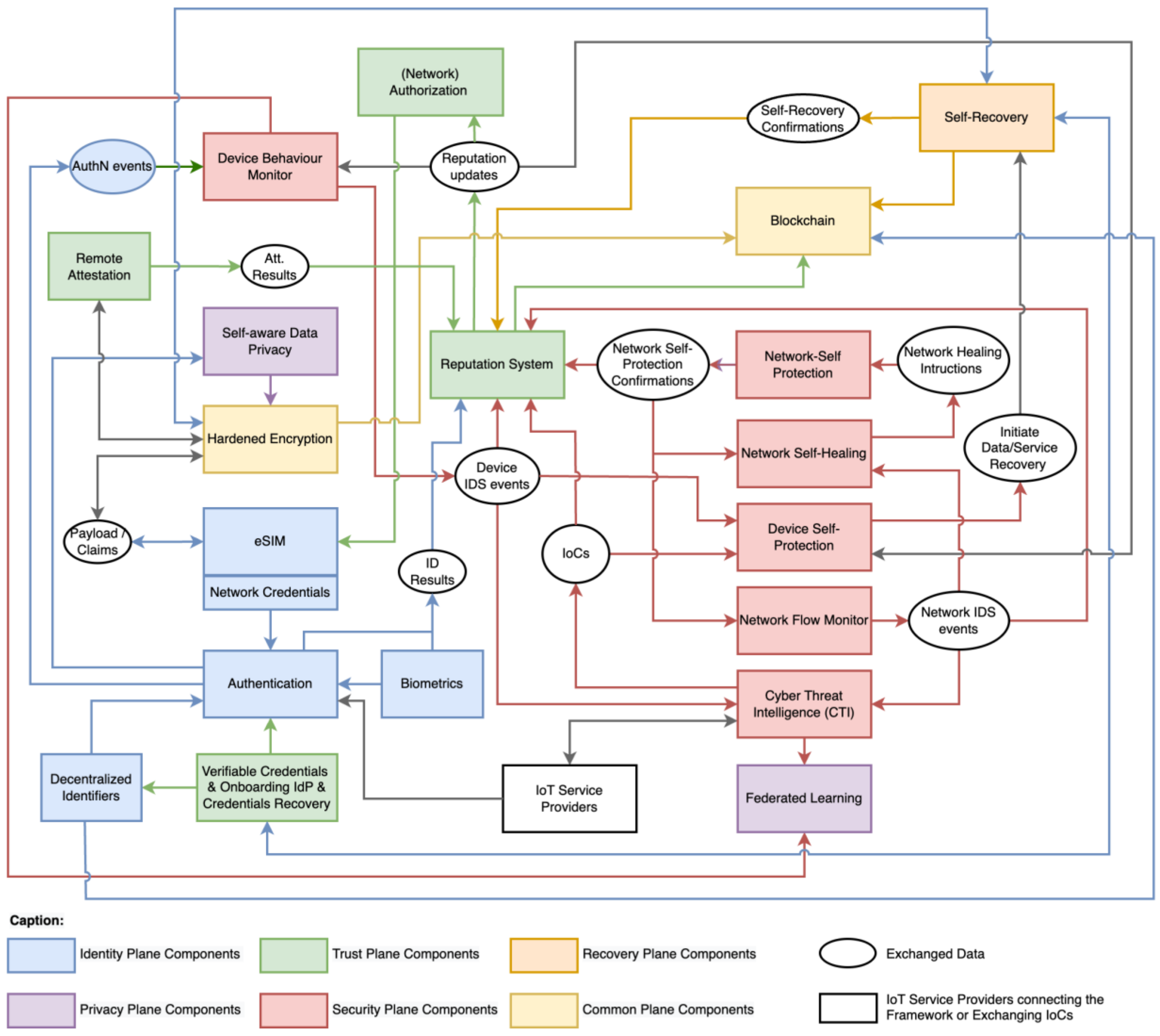


Figure 1 - ARCADIAN-IoT functional architecture

As observed in Figure 1, ARCADIAN-IoT comprises a significant number of interactions between different systems or components. Information such as intrusion detection events, indicators of compromise, claims and others, are exchanged bilaterally between components. This information is mostly exchanged via the frameworks' message bus – an instance of RabbitMQ that provides communication support for the components and agnosticism to the application domain (described in Section 5).

The updated deployment view is also presented, being depicted in Figure 2. A highlight of the main changes is reflected in blue coloured interfaces.

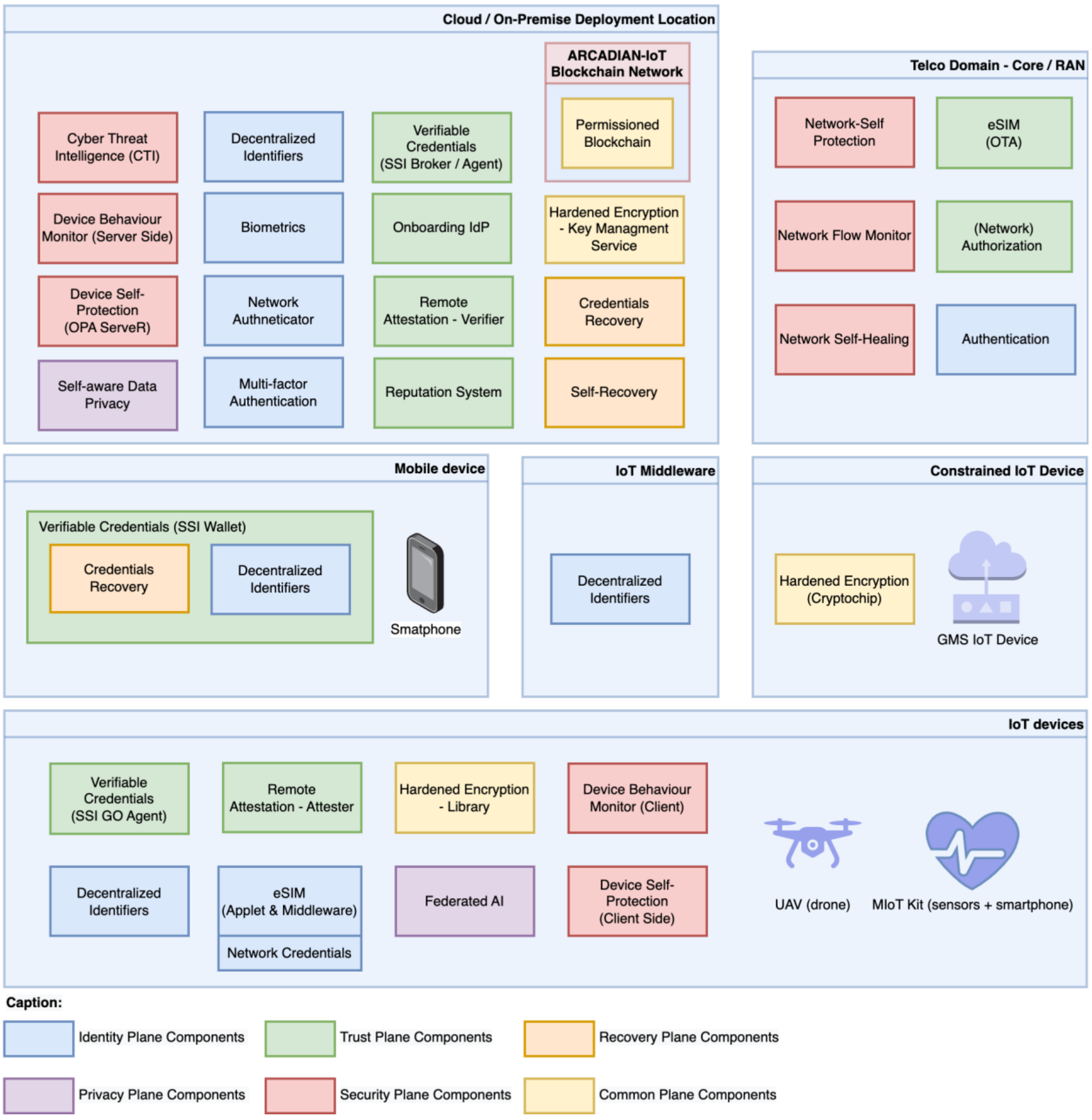

Cloud / On-Premise Deployment Location
ARCADIAN-IoT Blockchain Network
Permissioned Blockchain
Cyber Threat Intelligence (CTI)
Decentralized Identifiers
Verifiable Credentials (SSI Broker / Agent)
Device Behaviour Monitor (Server Side)
Biometrics
Onboarding IdP
Hardened Encryption - Key Managment Service
Device Self-Protection (OPA ServeR)
Network Authnethicator
Remote Attestation - Verifier
Credentials Recovery
Self-aware Data Privacy
Multi-factor Authentication
Reputation System
Self-Recovery
Telco Domain - Core / RAN
Network-Self Protection
eSIM (OTA)
Network Flow Monitor
(Network) Authorization
Network Self-Healing
Authentication
Mobile device
Verifiable Credentials (SSI Wallet)
Credentials Recovery
Decentralized Identifiers
Smatphone
IoT Middleware
Decentralized Identifiers
Constrained IoT Device
Hardened Encryption (Cryptochip)
GMS IoT Device
IoT devices
Verifiable Credentials (SSI GO Agent)
Remote Attestation - Attester
Hardened Encryption - Library
Device Behaviour Monitor (Client)
Decentralized Identifiers
eSIM (Applet & Middleware)
Network Credentials
Federated AI
Device Self-Protection (Client Side)
UAV (drone)
MIoT Kit (sensors + smartphone)
Caption:
Identity Plane Components
Trust Plane Components
Recovery Plane Components
Privacy Plane Components
Security Plane Components
Common Plane Components

Figure 2 - ARCADIAN-IoT deployment view

The ARCADIAN-IoT components (the ones without IP restrictions) have their code hosted in the project repository[1]. The following subsections provide additional detail on the interfaces and functionalities that were a target of integration within both P1 and P2.

2.2 Prototype functionality overview

A high-level overview of the main functionalities and outcomes that were delivered in P2 from the point of view of each component is presented in the tables presented next (**Error! Reference source not found.** and **Error! Reference source not found.**).

Horizontal ARCADIAN-IoT Planes

Table 1 – Summary of P2 functionalities and outcomes regarding components from ARCADIAN-IoT horizontal planes

Plane	Component	P2 functionalities and outcomes
Privacy Plane	Self-aware Data Privacy	• Access management refactor and integration with authentication component • Policy description and retrieval • Policy enforcement • Integration with Hardened Encryption • Recommender System prototype
	Federated AI	• Data rebalancing • Communication-efficient and robustness functions for model resizing and sharing
Security Plane	Network Flow Monitoring	• Prototype of DDoS known attacks detection with 5GS capabilities • Accurate detection of the attacks • 5G System network metrics recorded by the component • Aggregation of metrics • Integration with Network Self-Healing, Network Self-Protection components
	Device Behaviour Monitoring	• Data Extraction and log parsing from devices (Linux, smartphone / Android) • Data preparation (transforming log info into ML model inputs) • Event classification • Implementation of central model aggregator • Integration of data rebalancer (from Federated AI) • Generation of CTI events • Federated training
	CTI	• Internal automated functions • Event parsing and formatting • IoC aggregation • ML model manager
	Network Self-Healing	• Topology discovery • Prescriptive analytics • Integration with Network Flow Monitoring, Network Self-Protection

[1] https://gitlab.com/arcadian_iot

		components
	Network Self-Protection	• Enforce security mitigation rules • Abstract different mitigation technologies • Integration with Network Flow Monitoring, Network-Self-Healing components
	IoT device Self-Protection	• Integration with Message bus and Device Behaviour Monitoring • Self-protection policies (definition and storage) • Self-protection policies to device manager (Android devices) • Enforcement of policies on device emulator (Linux) • Packaging for Android-based devices • Assessment of self-protection policies based on Indicators of Compromise and Device Reputation
Common Plane	Hardened Encryption (w/ eSIM)	• eSIM security applet for Root of Trust digital signature of encrypted payloads • Over the air management of the SIM applets, including for mobile app (Android) identification and authorization for interaction with SIM. • Middleware for integration of the HE with eSIM security applet in Python and Android implementation of it • HE library for encrypting data with ABE that can be used in Go, Python, C or Java • Public keys integrated with Blockchain • Decentralized key management system integrated with Blockchain and Authentication
	Hardened Encryption (w/ Cryptochip)	• Hardware and firmware implementation (integration of IoT device motherboard, extension boards for grid interfacing, communication boards) • Encryption / decryption middleware baseline solution, including encryption/decryption • Integration with Device Behaviour Monitoring, Remote Attestation, DID, Self-Aware Data Privacy,
	Permissioned Blockchain	• DID:ELEM method trust anchor • ABE Key Management • Reputation scores • DID:PRIV • Trusted Organization Register[2]

Vertical ARCADIAN-IoT Planes

Table 2 - Summary of P2 functionalities and outcomes regarding components from ARCADIAN-IoT vertical planes

Plane	**Component**	**P2 functionalities and outcomes**

[2] Trusted Organization Register was later added to facilitate the trusted registration of Persons, IoT Devices & Services only by organizations that were previously provisioned in the Trusted Organization Register.

Identity Plane	Decentralized Identifiers	• Private DID:PRIV method implemented on ARCADIAN-IoT Permissioned Blockchain. • DID:KEY on mobile wallet • DID:WEB on SSI Agent • DID:WEB with Authentication based on Signed Challenge / Response for constrained devices • Creation of Service Providers DIDs for the ARCADIAN-IoT Trusted Organization Registry
	Verifiable Credentials / Onboarding IdP	• SSI Issuer to Issue Person, Organization Member & Device Verifiable Credentials • SSI Wallet for a person and Organization Member to manage their Verifiable Credentials on their mobile device • Single SSI Verifier to verify the presentation of Person, Organization Member & Device VCs on behalf of the ARCADIAN-IoT framework • Integration with the MFA to return a token to the Service Provider for the verified entity i.e. Person, Organization Member, IoT Device and constrained IoT Device • Onboarding IdP frontend and backend to support registration and authentication to the ARCADIAN-IoT framework for Persons, Organization Members, IoT Devices and constrained IoT Devices • SSI Agent developed in GO to be deployed on an IoT Device to manage its DID and Verifiable Credentials
	SIM/eSIM (Network-based authentication)	• Zero-touch authentication of devices in third-party services leveraging cellular networks credentials
	Biometrics	• Pre-processing of video stream • Face detection model • Face recognition model • User registration • Identification results report • CRUD operations and extension of functionalities. • Improvement in terms of speed and accuracy • Encryption of images and messages
	Authentication (Multi-Factor Authentication)	• Multi-Factor Authenticator (MFA) – person authentication - integrating with Biometrics, Network-based Authentication and Verifiable Credentials • Provisioning of Authentication results to infer threats (e.g. processed by Device Behaviour Monitoring) • Device MFA integrating Network-based Authentication with Decentralized Identifiers and Verifiable Credentials.
Trust Plane	Network Authorization	• Distribution of trustworthiness information to the eSIM for self-protection and self-recovery actions • Integration of the Network-based Authorization, running in a testbed of a Core Network (4G and 5G), with production networks, controlling the communication of real devices according to trust information provided by the Reputation System.

	Reputation System	• RabbitMQ communication with components sending event information • Reputation score based on Alpha-Beta model (for events received) • Map reputation score with policies • Component to allow user to specific policies (via a REST API) • Reputation score calculation • Support 3 reputation models: Alpha-Beta, Alpha-Beta with severity and Dominance Models • Process Attestation Results for obtaining reputation scores (Remote Attestation integration) • Support entities' reputation storage in Blockchain
	Remote Attestation	• Ability of Attester to request encryption & signing of claims (via Hardened Encryption libraries), generate and format Evidence (CBOR) and transmit it to Verifier • Ability of Verifier to process encryption keys, receive dummy reference values and evidence • Claims collection in real devices (Android, Linux OS's) • Appraisal of evidence for generation of Attestation Results and transmission to Reputation System
Recovery Plane	Self-Recovery	• Prototype for SIM/eSIM-related operational recovery process • Data backup and recovery client + server part • Backup encryption via integration with HE
	Credentials Recovery	• SSI Wallet backup and recovery of DID and Verifiable Credentials based on user identity and secret • IoT device public DID recovery with key rotation and Verifiable Credentials restoration. • eSIM credential rotation

2.3 Chain of Trust

One of the main objectives of ARCADIAN-IoT is to "*provide distributed and autonomous models for trust, security and privacy, enabling a Chain of Trust (CoT)"*. According to NIST, a Chain of Trust is a "*A **method for maintaining valid trust boundaries by applying a principle of transitive trust**, where each software module in a system boot process is required to measure the next module before transitioning control*"[3]. CoTs depend on Roots of Trust (RoT), or more precisely, "*every **CoT starts with a RoT** module*". The same source refers RoTs as "*highly reliable hardware, firmware, and software components that perform specific, critical security functions*"[4]*, and "because roots of trust are inherently trusted, they must be **secure by design**. Roots of trust provide a firm foundation from which to build security and trust*."

NIST also defines Chain of Trust from a supply chain point of view as "*a certain **level of trust in supply chain interactions such that each participant in the consumer-provider relationship***

[3] NISTIR 8320 – Hardware-enabled Security: Enabling a Layered approach to platform security
[4] NIST SP 800-172 - Enhanced Security Requirements for Protecting Controlled Unclassified Information: A Supplement to NIST Special Publication 800-171

provides adequate protection for its component products, systems, and services"[5].

ARCADIAN-IoT implements multiple technologies which, building on both hardware-based / software-based Roots of Trust, provide integrity assurance for IoT devices and associated services. On the other hand, a subset of ARCADIAN-IoT security-related components, while contributing to building trust for IoT devices, IoT services and persons accessing the services, are not part of the Chain of Trust due to the lack of underlying foundational Root of Trust. The detailed contribution of each ARCADIAN-IoT component involved in the CoT is described below.

2.3.1 Decentralized Identifiers

Role: Decentralized Identifiers (DIDs) enable user-centric, secure, and interoperable digital identities, reduce reliance on central authorities and enhance privacy and data control.

Contribution to Chain of Trust: Decentralized Identifiers are a foundational component in the chain of trust in that they provide a secure self-sovereign identity and provide Verifiable Credentials with a cryptographic secure means to attest their claims.

Root of Trust: The root of trust for DIDs is established through decentralized, cryptographically secure key pairs and a globally recognized and resolvable identifier structure. This ensures trust without relying on a specific technology, such as blockchain. That said, Distributed Ledger Technologies (DLT) technologies are well recognized as enabling a secure scalable decentralized ledger for public DIDs for certain use cases e.g. Trusted Issuers ledger.

2.3.2 SIM

Role: The SIM technologies developed in ARCADIAN-IoT are intended to be used as Root of Trust[6] for other components (e.g. Hardened Encryption and Authentication) and for IoT services. Some of these developments departed from GSMA IoT SAFE, which employs the SIM as hardware Root of Trust.

Contribution to Chain of Trust: The SIM will provide digital signatures to information leaving the device, ensuring the identification of its sender and that the information was not tampered with after leaving the device. The secrets to perform the digital signature are generated in the hardware secure element, and there is no method to access them from outside the SIM. A public key for verifying the hardware-based digital signatures is provided by the SIM when the secrets are generated. SIM network credentials will also be used in a Network-based Authentication component, which will be part of ARCADIAN-IoT Multifactor Authentication.

Root of Trust: SIM hardware, proven and well-accepted as secure.

2.3.3 Verifiable Credentials

Role: Verifiable Credentials (VCs) provide a standard and cryptographically secure way to attest claims or assertions about a person, entity, or thing.

Contribution to Chain of Trust: Verifiable Credentials together with Decentralized Identifiers,

[5] NIST SP 800-37 Rev.2 - Risk Management Framework for Information Systems and Organizations: A System Life Cycle Approach for Security and Privacy

[6] SIM technologies developed apply to any widely used SIM form, namely SIM, eSIM and iSIM)

establish a reliable and auditable chain of trust, enabling trusted parties to issue and verify credentials, fostering transparency and trustworthiness in their digital interactions. This combination ensures end-to-end trust in the issuance, presentation, and verification of credentials, making it an integral part of modern identity and trust systems.

Root of Trust: The root of trust for Verifiable Credentials is established through the cryptographic signing and verification of these credentials, based on DIDs and their associated public keys, as well as the VCs being issued by a trusted issuer DID e.g., listed in a trusted registry.

2.3.4 Multifactor Authentication (MFA)

Role: Provide robust identification mechanisms to devices and persons.

Contribution to Chain of Trust: Based on SIM (hardware-based) credentials, biometrics and DIDs, it will support the trust of other components in the identity of devices and persons. Based on the successful verification of the credentials provided, it will issue a protected and signed ID token that devices and people (mobile applications) can use to perform actions in the IoT service. By the successful validation of the ID token, other components can trust the identity of the entity.

Root of Trust: SIM-based / cellular network-based credentials and DIDs

2.3.5 Remote Attestation

Role: Remote Attestation in ARCADIAN-IoT is aimed at assessing the conformity of IoT devices (i.e. their configurations, software versions, etc) to expected (acceptable according to predefined policies) state.

Contribution to Chain of Trust: Upon analysis of Evidence sent by each IoT device, the Verifier component generates Attestation Results which are appraised / evaluated by Reputation System, directly contributing to the IoT device (attester) trust modelling / reputation.

Root of Trust: Remote Attestation builds on a HW-based RoT such as eSIM or cryptochip for the measurement of the device's claims (measurements or readings corresponding to current state).

2.3.6 Reputation System

Role: Reputation System is ARCADIAN-IoT is aimed to assess the reputation of different entities, including the IoT devices, the persons using specific services, and the services that are enabled.

Contribution to Chain of Trust: The Reputation System analyses the information received from diverse components to determine the reputation score mapping to a trust level, where 0 is not trusted and 1.0 is fully trusted. Initially, the Reputation system initializes the reputation of an entity, after the confirmation of its registration, that is it was assigned an identifier. The initial reputation value is set for 0.5, neither being trustworthy or not trustworthy. For instance, the Reputation System uses the information from the Remote Attestation component to update the reputation score of a device, upon the results of the attestation. The information of the Multifactor Authentication is also used to update the trust level information of devices and persons upon the successful or unsuccessful authentication. The results of the biometrics component are also considered to update the reputation information of persons. The trust information is also kept on permissioned blockchain to ensure transparency and traceability of the reputation information.

Root of Trust: The reputation system builds on the information collected from other ARCADIAN-IoT components to build trust, not having itself a direct link to a RoT. For instance, it relies on the DIDs & VCs to securely and uniquely identify the different entities. It also relies on attestation results (whose RoT as either the eSIM or cryptochip) for updating the entities' reputation scores.

As observed through the above descriptions, ARCADIAN-IoT Chain of Trust is constituted as well with components from the Horizontal Planes such as Privacy (Self-aware Data Privacy) and Common Plane (the eSIM- or cryptochip based Hardened Encryption approaches, and the Blockchain). Their detailed involvement in the CoT is presented from this point on.

2.3.7 Self-Aware Data Privacy

Role: The Self-Aware Data Privacy component introduces the definition and enforcement of data policies, which include attribute based hardened encryption and anonymisation techniques to be applied on domain specific user data.

Contribution to Chain of Trust: Self-Aware Data Privacy ensures that sensitive data is protected, ensuring that only trusted entities can access information such as a patient's medical records.

Root of Trust: Hardened encryption techniques (eSIM- or cryptochip-based) ensure that only a specific set of entities are capable of accessing encrypted data, meaning that not only is a decryption key required, but also for the entity to belong to a previously specified group for example (e.g. a user must be a doctor to access a patient's full medical records, a nurse may only access certain attributes).

2.3.8 eSIM-based Hardened Encryption

Role: The eSIM-based Hardened encryption ensures reliable and secured end-to-end communication between the base system and mobile device. This enables over-the-air management of the SIM applets, including for mobile app (Android) as well as identification and authorization for interaction with SIM. The Hardened Encryption component provides functionalities to secure data at rest in ARCADIAN-IoT. It provides encryption libraries allowing to encrypt/decrypt data using Attribute Based Encryption (ABE) and hardens the security by signing the encrypted payload with eSIM based signatures. This includes also the final version of encryption library and key management that is now integrated with and public keys and with Blockchain and Authentication.

Contribution to Chain of Trust: In the case of loss or tampering with the eSIM this can be marked as un-trusted and removed/banned from the system and eventually replaced with a new device, thus ensuring the integrity and security of the whole system.
Root of Trust: the eSIM in itself is a Root of Trust for this system. Additionally, the root of trust is ensured by the reputable and verified vendor of the HW, as well as the complete own development of the middleware. Thus, ensuring that the system is fully controlled by the Manufacturers.

2.3.9 Cryptochip-based Hardened Encryption

Role: The Crypto chip-based Hardened Encryption is used to provide reliable security for existing or new industrial IoT systems, powered by microcontrollers, and with limited computing resources as consequence. The encryption logic stands on 2 components: a cryptochip embedded (from hardware and firmware perspective) into the IoT device, and a middleware software application running the keys provisioned into crypto chip (more details in D3.3 [3]) and relaying the authentication and traffic performed by devices to/from the IoT platform.

Contribution to Chain of Trust: Crypto chip-based Hardened Encryption ensures that the whole data transmitted / received by IoT field devices is protected permanently, across device lifecycle, and across communication stages (registration, authentication, traffic, events & incidents). Only trusted users and entities are allowed to operate and access data collected from field sensors or sent to actuation field components, and system design permits, due to the Root of Trust mechanism, a prompt and solid control over encryption and decryption keys management, minimizing potential hacking, data alteration or other sabotage attempts - including physical attacks on field devices.

The RoT ensures that any single message, at any stage of communication (DIDs, attestation – involving Remote Attestation, traffic – involving Self-Aware Data Privacy, or regular IoT platform sensors data communication) get conditioned by a successfully hardware encryption *a priori* / by default.

Root of Trust: provided by the encryption chipset, the specialised encryption chipset firmware and provisioning software provided by a high reputation vendor (Infineon Technologies in this case).

2.3.10 Permissioned Blockchain

Role: Blockchain is used to secure and maintain a decentralized ledger of transactions and data in ARCADIAN-IoT, so to provide a tamper-resistant, transparent, and auditable record.

Contribution to Chain of Trust: The whole concept of blockchain technology is to provide a secure store for transactions and data that is virtually impossible to corrupt. To this effect, ARCADIAN-IoT employs the blockchain to record DIDs, public keys, Reputation, and trusted organizations in a permissioned based trusted ecosystem. In this way it helps ensure the integrity of published information by the trusted entities given permission to the blockchain and serves as a highly available trust anchor for the trusted organizations registered in the ecosystem and verifying the authenticity of all data published on it by the ecosystems' organizations.

Root of Trust: The permissioned blockchain´s root of trust is based on the blockchain´s inherent tamper resistant architecture and also its authorized access to the blockchain.

2.3.11 Chain of Trust Summary

The different Chains of Trust provided by ARCADIAN-IoT can be organized as follows:

- Chain of Trust for personal devices
- Chain of Trust for IoT devices
- Chain of Trust for computationally constrained devices.

ARCADIAN-IoT provides a solid CoT for personal and IoT devices building from the SIM[7] secure element RoT role for supporting essential security services such as hardened encryption or remote attestation. Secure identification of personal devices is grounded in both eSIM's secure element and DID's ability to cryptographically issue and attest Verifiable Credentials' claims.
Trust establishment, in the form of Reputation, is calculated based on complementary information both deriving from RoT-backed security services (e.g. device attestation, device authentication), as well as those services lacking RoT (e.g., behaviour anomaly identification). Permissioned Blockchain is used to record DIDs, public keys, Reputation, and trusted organizations, further

[7] The tech developed works for any SIM form (SIM, eSIM, iSIM)

ensuring their integrity. Finally, Self-Aware Data Privacy enables the definition and enforcement of user data policies using techniques such as hardened encryption or anonymisation according to the privacy requirements.
ARCADIAN-IoT CoT for constrained IoT devices is built upon the cryptochip secure properties as RoT. Trust establishment leverages the cryptochip as RoT for attestation at device provisioning or (re)authentication stages, with Permissioned Blockchain and Self-aware Data Privacy components performing the same role as for regular IoT devices.
In this case, DIDs are implemented on the middleware on behalf of the IoT Devices for authentication requests, so do not actually provide CoT on the devices themselves for constrained IoT devices.

With respect to other ARCADIAN-IoT components not taking part in the CoT, some notes are now provided. From the Vertical planes' perspective, and particularly Recovery plane, both Self-recovery (for data recovery) and Credentials recovery build on RoTs (DIDs, eSIM, etc.,) for their security and integrity-enhancing role. However, they do not directly contribute to ensuring the integrity of the system's components, secure boot or cryptographic verification, thus do not take part in the CoT. Also, the Biometrics component, part of the Identity Plane, is used to enhance security by offering an additional factor of authentication for individuals, but not building on a foundational and always trusted component (RoT), it is not part of the CoT.

From the Horizontal planes' perspective, the Security plane components are absent from the CoT due to the lack of a foundational Root of Trust in the collected or observed data (e.g., device or network traffic). The same applies for the Federated AI component, which can be seen as contributing to AI model trustworthiness over the training time, but depends on the assumption of the absence of stealthy activities during the communication of the federated training process.

3 FRAMEWORK INTEGRATION APPROACH

This section addresses ARCADIAN-IoT framework integration. It starts with a description of the approach taken to integrate technical components with each other and it is follows with a description of the integration approach taken with domains' uses cases. An overview of the supporting tools is provided at the end of this section.

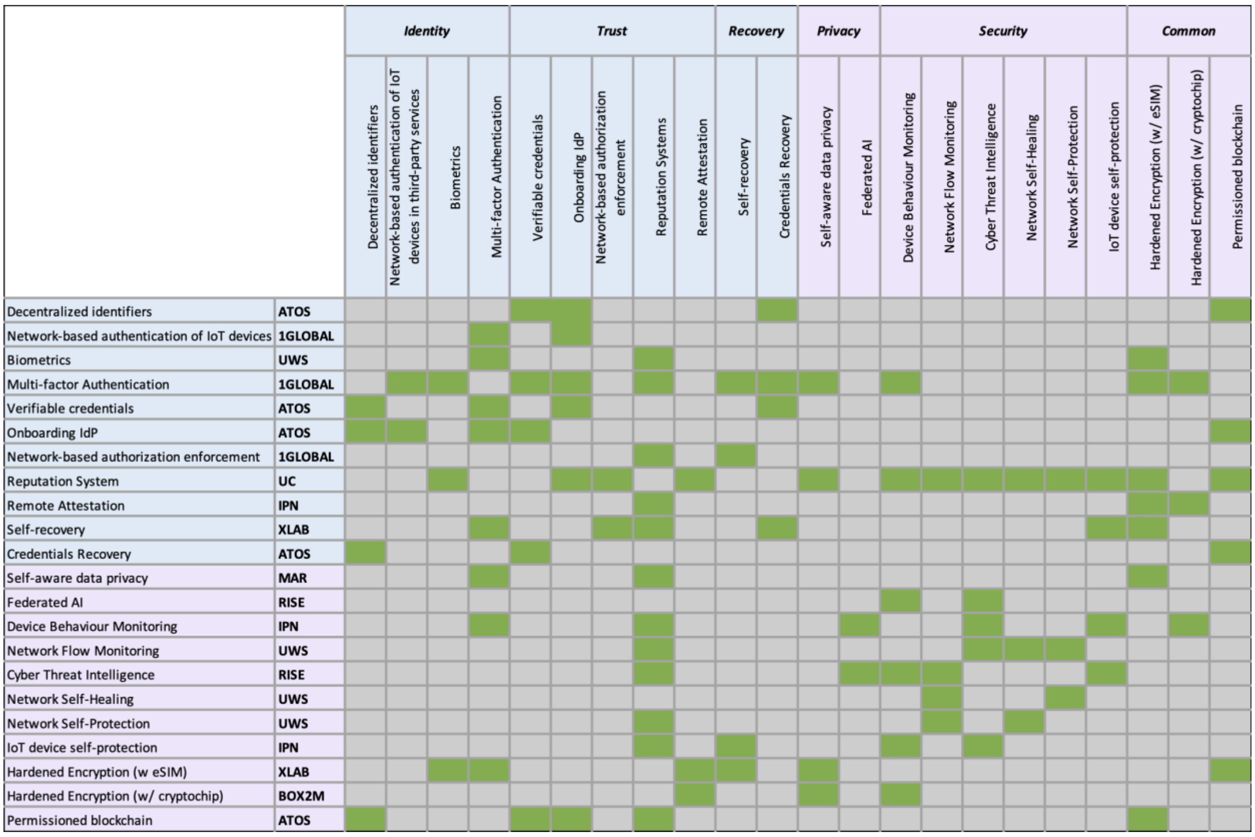
3.1 Component to component approach

There are 22 main components in ARCADIAN-IoT framework, spread across 3 horizontal planes and 3 vertical planes. The horizontal planes comprise the components that provide privacy, security and common aspects, while the vertical planes comprise the components that provide identity, trust and recovery actions. As depicted in the previous section (deployment view), each component may itself be composed by multiple sub-components.

As soon as the ARCADIAN-IoT framework architecture was delivered (in M12, as stated and depicted in the previous section), the integration activities have been initiated (M13). During project execution it was defined a two phased approach designed as Prototype 1 (P1) and Prototype 2 (P2), which included the agreement on the interfaces to be used and the set of functionalities that had to be integrated during P1 (already reported in D5.1) and during P2 (this document).

3.1.1 Component integration overview

In addition to the functionalities defined to be ready for P1 and P2, it was also necessary to define which interfaces need to be implemented between ARCADIAN-IoT components. Figure 3 depicts the final integration that each component has with other ARCADIAN-IoT components (in green integrations that were implemented).



		Identity				Trust					Recovery		Privacy		Security						Common		
		Decentralized identifiers	Network-based authentication of IoT devices in third-party services	Biometrics	Multi-factor Authentication	Verifiable credentials	Onboarding IdP	Network-based authorization enforcement	Reputation Systems	Remote Attestation	Self-recovery	Credentials Recovery	Self-aware data privacy	Federated AI	Device Behaviour Monitoring	Network Flow Monitoring	Cyber Threat Intelligence	Network Self-Healing	Network Self-Protection	IoT device self-protection	Hardened Encryption (w/ eSIM)	Hardened Encryption (w/ cryptochip)	Permissioned blockchain
Decentralized identifiers	**ATOS**					🟩	🟩					🟩											🟩
Network-based authentication of IoT devices	**1GLOBAL**				🟩		🟩																
Biometrics	**UWS**				🟩				🟩												🟩		
Multi-factor Authentication	**1GLOBAL**		🟩	🟩		🟩	🟩		🟩		🟩	🟩	🟩		🟩						🟩	🟩	
Verifiable credentials	**ATOS**	🟩			🟩		🟩					🟩											
Onboarding IdP	**ATOS**	🟩	🟩		🟩	🟩																	🟩
Network-based authorization enforcement	**1GLOBAL**								🟩		🟩												
Reputation System	**UC**			🟩			🟩	🟩		🟩			🟩		🟩	🟩	🟩	🟩	🟩	🟩	🟩		🟩
Remote Attestation	**IPN**								🟩												🟩	🟩	
Self-recovery	**XLAB**				🟩			🟩	🟩			🟩								🟩	🟩		
Credentials Recovery	**ATOS**	🟩				🟩																	🟩
Self-aware data privacy	**MAR**				🟩				🟩												🟩		
Federated AI	**RISE**														🟩		🟩						
Device Behaviour Monitoring	**IPN**				🟩				🟩					🟩			🟩			🟩		🟩	
Network Flow Monitoring	**UWS**								🟩								🟩	🟩	🟩				
Cyber Threat Intelligence	**RISE**								🟩					🟩	🟩	🟩				🟩			
Network Self-Healing	**UWS**															🟩			🟩				
Network Self-Protection	**UWS**								🟩							🟩		🟩					
IoT device self-protection	**IPN**								🟩		🟩				🟩		🟩						
Hardened Encryption (w eSIM)	**XLAB**			🟩	🟩					🟩	🟩		🟩										🟩
Hardened Encryption (w/ cryptochip)	**BOX2M**									🟩			🟩		🟩								
Permissioned blockchain	**ATOS**	🟩				🟩	🟩		🟩												🟩		

Figure 3 - Component to component integration mapping

The consortium has achieved a component integration as depicted in Figure 3, overcoming the inherent complexity derived not only from having a large number of components in ARCADIAN-IoT framework and the research required to develop the technical components, but also from having a short integration period.

3.2 Component to domain approach

Besides the integration of technical components, there is also effort associated to the integration of the technical components with the specific use cases (initially specified in D2.2 [1]) of the three distinct ARCADIAN-IoT domains: Emergency and Vigilance (Domain A), Grid Infrastructure Monitoring (Domain B) and Medical IoT (Domain C). This integration refers to the tailoring and adaptation of components to the specifics of an application domain, such as enabling a given function or service to support and fully cope with the target execution environment (e.g., hardware, operating system, application artifacts). As such, it includes any necessary setup or configuration for smoothly running and executing the expected role in the use case.

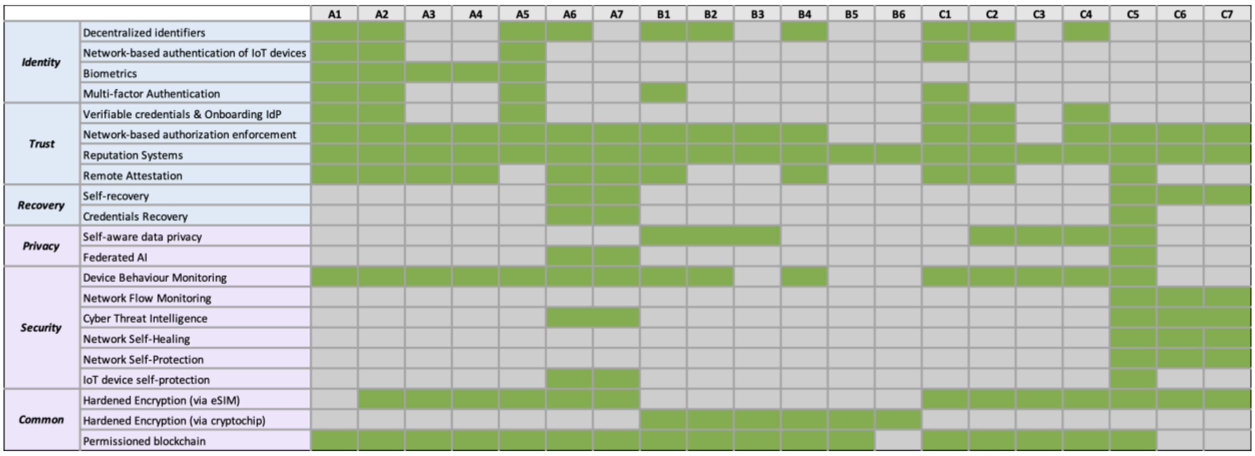
A subset of the use cases was initially addressed as part of P1 (reported in D5.1), while P2 covers the complete functionality required by the whole list of use cases (i.e. both those targeted in P1 and the remaining ones). Table 3 depicts the use case coverage per P1 and P2.

Table 3 - Use cases scoped in each prototype: blue background for use cases addressed in P1 and evolved in P2; green background for use cases only addressed in P2.

Responsible	Use case ID	Use case name
LOAD	A1	Person registration at DGA service
	A2	Person authentication at the DGA service
	A3	Person retrieving and editing personal databiome
	A4	Person requesting a DGA service
	A5	DGA service
	A6	Drone security or privacy incident
	A7	Personal device security or privacy incident
BOX2M	B1	New device registration
	B2	GMS IoT device data gathering and transmission process
	B3	Service request from third-party IoT monitoring platforms
	B4	GMS IoT device security or privacy incident
	B5	GMS middleware security or privacy incident
	B6	External data audit to grid infrastructure
RGB	C1	MIoT kit delivery - Patient registration and authentication
	C2	MIoT Capturing and sending vital signs and perceived health status
	C3	Personal data processing towards health alarm triggering
	C4	Monitor a patient and update a patient monitoring protocol
	C5	Patient MIoT devices security or privacy incident
	C6	MIoT Cloud services security or privacy incident
	C7	Medical 3rd party security or privacy incident

3.2.1 Use case integration overview

Figure 4 represents the activities around the integration of technical components with the domains' use cases (in green the components that participate in each Use-Case and in grey the cases that are not-applicable).



		A1	A2	A3	A4	A5	A6	A7	B1	B2	B3	B4	B5	B6	C1	C2	C3	C4	C5	C6	C7
Identity	Decentralized identifiers	🟩	🟩			🟩	🟩		🟩	🟩		🟩			🟩	🟩		🟩			
	Network-based authentication of IoT devices	🟩	🟩			🟩									🟩						
	Biometrics	🟩	🟩	🟩	🟩	🟩															
	Multi-factor Authentication	🟩	🟩			🟩			🟩						🟩						
Trust	Verifiable credentials & Onboarding IdP	🟩	🟩			🟩									🟩	🟩		🟩			
	Network-based authorization enforcement	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩			🟩	🟩		🟩	🟩	🟩	🟩
	Reputation Systems	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩
	Remote Attestation	🟩	🟩	🟩	🟩		🟩	🟩	🟩			🟩			🟩	🟩			🟩		
Recovery	Self-recovery						🟩	🟩											🟩	🟩	🟩
	Credentials Recovery						🟩	🟩											🟩		
Privacy	Self-aware data privacy								🟩	🟩	🟩					🟩	🟩	🟩	🟩		
	Federated AI						🟩	🟩											🟩		
Security	Device Behaviour Monitoring	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩		🟩			🟩	🟩	🟩	🟩	🟩		
	Network Flow Monitoring																		🟩	🟩	🟩
	Cyber Threat Intelligence						🟩	🟩											🟩	🟩	🟩
	Network Self-Healing																		🟩	🟩	🟩
	Network Self-Protection																		🟩	🟩	🟩
	IoT device self-protection						🟩	🟩											🟩		
Common	Hardened Encryption (via eSIM)		🟩	🟩	🟩	🟩	🟩	🟩							🟩	🟩	🟩	🟩	🟩	🟩	🟩
	Hardened Encryption (via cryptochip)								🟩	🟩	🟩	🟩	🟩	🟩							
	Permissioned blockchain	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩		🟩	🟩	🟩	🟩	🟩		

Figure 4 - Use Case Integration overview

The integration into ARCADIAN-IoT domains mostly depended on the readiness of the use cases implementation. As the consortium follows an iterative development and integration approach and as each domain has approximately 7 use cases, considerable efforts have been made in parallel and iteratively.

The information presented, intends to grant the reader with the necessary information about the integration approach of ARCADIAN-IoT framework. Additional details will be provided in Section 4 (Component Integration) and Section 5 (Domain Integration).

3.3 Supporting communication tools

The ARCADIAN-IoT framework is comprised of 22 technical components, which, in cases of a fully-fledged deployment, may translate into the associated physical deployment locations spanning resources as diverse as IoT devices, mobile devices, IoT GWs, on-premises or private networks, or public Clouds, thus traversing different administrative domains. As such, ARCADIAN-IoT requires a reliable and efficient communication mechanism for both intra-component and ARCADIAN-IoT – third party services communications.

As documented in deliverable 2.5 (ARCADIAN-IoT architecture), **RabbitMQ** was the preferred tool for supporting ARCADIAN-IoT's component communication. By deploying an ARCADIAN-IoT managed message bus, the consortium can assure the necessary functional and operational requirements are met. While several ARCADIAN-IoT components consider the usage of publish/subscribe (pub/sub) mechanisms via a common message bus, other communication mechanisms are supported. For instance, Self-Aware Data Privacy relies on **REST APIs**, and SSI (with Decentralized Identifiers) supports **DID Comm** messaging. Other components comprise heterogeneous communications, leveraging for instance both REST APIs and message bus (e.g., Authentication component).

With respect to the RabbitMQ message bus, ARCADIAN-IoT Framework currently considers: (i) an exchange for every producer component (i.e., a component the produces and sends information to other components) and (ii) queues for every consumer component (i.e., a component the receives information for other components).

The types of currently defined exchanges are topic and fanout - the latter being used for instance by the Reputation System for broadcasting reputation updates to all subscribing components. The routing schema that was defined was based on keys that identify the <domain>, <component>, <location>, <entity_id>, and <service>. The current format is flexible enough to allow components to use a more fine-grained routing, when and if necessary.

Secure communications are a must-have, even at early stages. Therefore, P2 requires authentication (RabbitMQ PLAIN authentication mode) for all components connected to RabbitMQ's instance (deployed at IPN's premises and available to partners for demonstration of the project's prototype). Non-authenticated communication attempts are not allowed and requests are refused.

The setup allows the setting of other aspects such as SSL certificates, time to live, dead letter exchange or dead letter routing. Other communication approaches (via REST APIs and DID Comm) equally employ appropriate security considerations.

4 COMPONENT INTEGRATION

The functional integrations of the final prototype (P2) of ARCADIAN-IoT Framework are captured according to a) components' technical interfaces and functional integration flows (e.g., component A to component B), and b) use case or domain support (e.g., component "A" external system integration for supporting use case "Y") - the latter is addressed in Section 5.

Thus, this section describes the outcomes of the integration between ARCADIAN-IoT components: the integration from the point of view from horizontal and vertical planes are presented and followed by a visual representation of the integration status of ARCADIAN-IoT components with respect to the framework's architecture.

4.1 Integration of horizontal planes

This section presents the integration of the components that are part of the privacy, security and common planes, providing a view regarding the components services made available to other components.

4.1.1 Privacy Plane

Self-Aware Data Privacy

The Self-Aware Data Privacy component aims at allowing the definition of user-defined privacy policies for data, and by suggesting policies on similar data. It contains two main modules: a policy management module which enforces data privacy via the definition of privacy policies and a recommender module.

The policy management module relates specifically to attributed-based data encryption and/or anonymization and leverages synergies with the Hardened Encryption component to employ their Go libraries for demonstrating encryption in transit and at rest.

Policy Management and Hardened Encryption

The component, called Amon, integrates the hardened encryption library as an option for data processing, as defined in its data policies. When data is sent from a data provider (like a device making some measurements) to the service where the data is needed, the data is intercepted by an Envoy proxy, alongside the security token allowing to authenticate the request (containing a device ID for instance). The data is forwarded to the Amon processing middleware, which in turn will retrieve the data policies stored in Amon's API. Those policies contain the details on how the data is meant to be processed, which include the option for hardened encryption. An example policy would be:

{
id: 123
resource: patient1
resource_type: patient
mode: inTransit
technique: hardenedEncryption
attributes: [blood_type, blood_pressure]

}

If such a policy exists for the data being received, the middleware can then retrieve the keys needed for the encryption, retrieve the specific encryption policies for the data, and encrypt the data accordingly. Amon data policies are attribute based, so individual attributes can be encrypted separately, or a different type of processing can be performed (e.g. anonymise A, encrypt B, etc…). The processed data is then sent from the proxy to the destination service, now encrypted as specified in the policies. From the Data Provider's perspective, it had sent the data directly to the Service, where the proxy is effectively transparent to it.

Integration with reputation system

Amon can inform the reputation system when data is sent through the envoy proxy. The processing middleware can create a message, containing the information about the request and the identifiers contained in the security token, and publish it to a RabbitMQ queue. This message can then be consumed afterwards by any consumer connected to the same queue.

Integration with Multi Factor Authentication

Since the component acts as a proxy, it receives the security tokens sent by the source of the data being processed, and retrieves the information they contain, as well as verifying their signature.

Recommender system

The recommender system is deployed as a separate web service which provides a single endpoint which, when queried with a set of attributes and their relevant domain, will provide a recommendation for which techniques to apply to each attribute. For instance, the attributes of a medical patient could be the patient's name, age, blood type, and current blood pressure measurements. The recommender, given the attributes *name*, *age*, *blood type* and *blood pressure* might recommend anonymisation for the former three, but for the last one will recommend encryption.

The recommender can be used when creating new policies to make more informed decisions on how each individual attribute should be handled. Currently, a web UI interacting with both the data policies API and the recommender is available to do so.

Federated AI

The federated AI component offers a set of libraries that can be deployed by those components which integrate ML models and may benefit from a distributed training approach but also require that (i) the privacy of raw data and local model updates is preserved and (ii) no malicious participant is involved in the process.

Integration with Device Behaviour Monitoring

Since the Device Behaviour Monitoring component performs, in each end device (both typical IoT devices and smartphones), ML-based intrusion detection (namely a MLP based Neural Network), it deploys Federated AI libraries in the model training phase so that it updates the local client models not only with recent and local data, but also with synthetic data generated by the data rebalancer library, as a way to balance out the two classes (benign and malign) before performing the collaborative and federated re-training phase of the models among IoT devices that are not willing to share their local training datasets.

Integration with CTI

The CTI component includes an “event classification module” which uses ML models to classify the type of threats, such as malware, phishing, or DDoS attacks. ML-based event classification deploys Federated AI libraries in the model training phase so that local client models can be updated not only with recent and local data, but also with remote data owned by other parties. The CTI exploits the data rebalancing techniques to balance out the event classes before performing the collaborative and federated training of the models.

4.1.2 Security Plane

Device Behaviour Monitoring

The Device Behaviour Monitoring (deployed directly on the devices or at any different hosting environment when devices are not able to support installation) – monitors the devices activity and detects anomalies when the device behaviour deviates from normality. It forwards anomaly detection alerts (potential threats) to other components.

Integration with IoT device self-protection

The implementation of the DBM component considers activities on device level, and thus, the component runs passively on the background of the device, monitoring its behaviour based on system calls. The component outputs results obtained by the Machine Learning models, identifying the input vector as benign or malign.

In case the component identifies a malicious event, the Device Behaviour Monitoring component sends the information to other components, in this case, to the Device Self Protection component. Since the output of the model indicates a binary result (0 for normal, 1 for anomaly), additional information about the detected events is packaged, such as the PID and Service/Program name associated to the process to indicate the possible origin of the intrusion for the Device Self Protection component. Other information such as the time when attack was first detected, timestamp, confidence level is also included, which is packaged in JSON format and submitted to the RabbitMQ exchange (dbm_exchange).

Integration with Reputation System

As in the integration with IoT device self-protection, in case the component identifies a malicious event, the DBM component also submits the information to the ARCADIAN’s Message Bus, which the Reputation System component will consume. Since the output of the model indicates a binary result (0 for normal, 1 for anomaly), the information about the detected events is packaged, such as the PID and Service/Program name associated to the process to indicate the possible origin of the intrusion. Other information such as the time when attack was first detected, timestamp, confidence level is also included, which is packaged in JSON format and submitted to the RabbitMQ exchange (dbm_exchange).

Additionally, the DBM is constantly receiving and analysing reputations updates of the device, sent by the Reputation System, to detect sudden changes and alert the relevant security components.

Integration with Cyber-Threat Intelligence

Whenever an anomaly is detected, the same alerts sent via the message bus to the Reputation System and Device Self-Protection, are also intended to inform the Cyber-Threat Intelligence component.

Integration with Federated AI

The DBM component relies on Machine Learning models to perform the intrusion detection, namely a MLP based Neural Network. This means that overtime, the models need to be updated with recent data, collected from the devices, to increase the accuracy of event classification, and thus, Federated Learning is implemented using TensorFlow open-source library and applied in the model updates to guarantee the privacy of the devices that integrate this component. The process implements the Weighted Federated Average algorithm, which takes all the model parameters from the participating clients, and outputs an aggregated scaled mean of the Neural Network weights.

The DBM component implements a local training phase that updates the local client model not only with recent data, but also with synthetic data generated by RISE's Federated AI component, namely the data rebalancer library, which is incorporated on the client side, as a way to balance out the two classes (benign and malign) before performing the re-training phase of the models.

Integration with Multi-Factor Authentication

The DBM receives authentication events from the Multi-Factor Authentication system, transmitted via a RabbitMQ exchange, whenever there is an authentication attempt on the corresponding device. It collects and analyses failed attempts, initiating analysis after accumulating a minimum such attempts (10 by default). To detect brute force attacks, it employs an Isolation Forest algorithm to examine these traces. If an anomaly is detected in the authentication process, an alert is sent to the relevant security components.

Network Flow Monitoring

The Network Flow Monitoring (NFM) component is the horizontal plane component of the ARCADIAN-IoT framework responsible for the detection of known cyber threats alongside the network infrastructure. Whenever the NFM detects a potentially malicious flow in the network, it will trigger an alert to the Network IDS Event exchange. Such an alert is generated in a JSON format (see details in D3.3 [3]) and transmitted using AMQP over a RabbitMQ message bus. The NFM will be deployed centralised in a mirrored network where its traffic will be analysed.

Integration with Network Self-Healing

Whenever a malicious network flow, considered a DDoS attack, is detected by the NFM, this component will trigger a new NFM Event, as an alert, in a JSON format to the RabbitMQ exchange. Thus, the Network Self-Healing (NSH) will consume that message in its RabbitMQ queue. Then, the processes of the NSH to perform the analysis of the threat will start.

Integration with Network Self-Protection

As soon as the NSH publishes the set of decisions made to the corresponding RabbitMQ exchange, the Network Self-Protection (NSP) will consume these healing instructions. Therefore, this component executes the set of instructions in the provided order, with the specified network technology and topology location. Afterwards, the module will publish a confirmation message. This confirmation contains information about the mitigated flow and that it has been dropped. Hence, the NFM immediately changes the state of the flow from ACTIVE to DROP.

Integration with Reputation System

Whenever a malicious network flow, considered a DDoS attack, is detected by the NFM, this component will trigger a new NFM Event, as an alert, in a JSON format to the RabbitMQ exchange. Thus, the Reputation System (RS) will consume that message in its RabbitMQ queue. Then, the RS processes the new NFM Event received.

Integration with Cyber Threat Intelligence

Whenever a malicious network flow, considered a DDoS attack, is detected by the NFM, this component will trigger a new NFM Event, as an alert, in a JSON format to the RabbitMQ exchange. Thus, the Integration with Cyber Threat Intelligence (CTI) will consume that message in its RabbitMQ queue. Then, the CTI processes the new NFM Event received.

Cyber Threat Intelligence

Integration with Network Flow Monitoring

The network flow monitoring component provides an enhanced system to detect attacks along an entire IoT infrastructure. Whenever a new attack is detected in the infrastructure an alert (network IDS event) is generated in a JSON based format (details of the fields of the alert are provided in Deliverable D3.3 [3]) and transmitted over RabbitMQ exchange. The CTI component receives all the alerts which are parsed and processed in order to generate MIPS-based Indicator of Compromises (IoCs) that will be further analysed by the ML-based subcomponents. Once the IoCs are generated, they are ready to be shared.

Integration with Device Behaviour monitoring

The Behaviour monitoring component monitors the devices activity and detects anomalies when the device behaviour deviates from normality. It then forwards alerts (potential threats) to CTI. The CTI component receives all the alerts which are parsed and processed in order to generate MIPS-based Indicator of Compromises (IoCs) that will be further analysed by the ML-based subcomponents. Once the IoCs are generated, they are ready to be shared.

Integration with Federated AI

The CTI component includes three modules with ML-based capabilities. In particular the "event classification module" uses Machine Learning (ML) models to classify the type of threats, such as malware, phishing, or DDoS attacks. ML-based event classification deploys Federated AI libraries in the model training phase so that local client models can be updated not only with recent and local data, but also with remote data owned by other parties. The CTI exploits the data rebalancing techniques to balance out the event classes before performing the collaborative and federated training of the models.

Integration with IoT Device Self-protection

The CTI component processes and complements IoCs gathered from local and external sources. After the processing, the CTI component publishes the processed IoCs. They are received and used also by the IoT Self-protection component to determine self-protection policies.

Integration with Reputation System

The CTI component processes and complements IoCs gathered from local and external sources. After the processing, the CTI component publishes the processed IoCs. They are received and used also by the Reputation System to accordingly modify reputation levels.

Network Self-Healing

The Network Self-Healing (NSH) is the horizontal plane component of the ARCARDIAN-IoT framework that receives information about detected cyber threats from the Network Flow Monitoring (NFM) component. It deploys multiple instances to discover the network topology. It performs Predictive Analytics to create a decision that contains WHAT, WHERE, WHEN and for HOW LONG the attack should be stopped. Such a decision is sent through the Network Healing Instruction exchange of the RabbitMQ message bus to the specific instance of the Network Self-Protection component that will enforce it in the data plane.

Integration with Network Flow Monitoring

When an attack is detected by the Network Flow Monitoring, it sends a message through the Network IDS Event exchange where the Network Self-Healing is listening and consumes the specified message by the message bus queue. This interface is already integrated and successfully tested between both components. It uses AMQP to exchange messages in JSON format over a RabbitMQ message bus provided by IPN. Then, the NSH performs prescriptive analytics (a deeper explanation can be found in D3.3 [3]) to prepare a set of decisions that will be published to the Network Healing Instructions exchange.

Integration with Network Self-Protection

When the Network Self-Healing creates the decision, it sends a message through the Network Healing Instruction exchange where the Network Self-Protection is listening; and this module consumes the specified message by the message bus queue. This interface is already integrated and successfully tested between both components. It uses AMQP to exchange messages in JSON format over a RabbitMQ message bus provided by IPN.

Network Self-Protection

The Network Self-Protection (NSP) is the horizontal plane component of the ARCARDIAN-IoT framework that stops illegitimate traffic from attacks against the network. It is a distributed component that is deployed at every single point within the data plane where attacks can be stopped. NSP uses the decisions created by the Network Self-Healing component to stop the attack in the data plane. An extended set of these decisions can be found in D3.3 [3].

Integration with Network Flow Monitoring

Whenever the Network Self-Protection (NSP) enforces the mitigation rule in the data plane to stop the incoming attack detected by the NFM, the module will publish a confirmation message. This confirmation contains information about the mitigated flow and that it has been dropped. Hence, the NFM immediately changes the state of the flow from ACTIVE to DROP.

Integration with Network Self-Healing

When the Network Self-Healing creates the decision, it sends a message through the Network Healing Instruction exchange where the Network Self-Protection is listening; and this module consumes the specified message from the message bus queue. This interface is already integrated and successfully tested between both components. It uses AMQP to exchange messages in JSON format over a RabbitMQ message bus provided by IPN.

Integration with Reputation System

Whenever the Network Self-Protection (NSP) enforces the mitigation rule in the data plane to stop the incoming attack detected by the NFM, the module will publish a confirmation message. Thus, the Reputation System (RS) will consume that message in its RabbitMQ queue. Then, the RS processes the new confirmation message received.

IoT Device Self-Protection

The IoT Device Self Protection (deployed directly on the devices or at any different hosting environment when devices are not able to support installation) - receives information about threat detection. The threat information is sent from the Device Behaviour Monitoring.

Integration with Device Behaviour Monitoring

The threat information comes from the Device Behaviour Monitoring component, which runs at device level, or elsewhere when the devices are not able to support local deployment. When both components are running on the same device, the Device Self-Protection runs in a thread of the Device Behaviour Monitor, and the communication is performed inter-thread. Otherwise, both components communication via RabbitMQ exchange (dbm_exchange).

The alert messages are sent in JSON format from the Device Behaviour Monitor, containing data related to an IoT device's threat. Data such as the attack start date, device ID, cause and process ID enable further possibilities for IoT Device Self-Protection to select the ideal policy that mitigates or minimizes the impact of the detected threats.

Integration with Self Recovery

Upon receiving the anomaly detection alert from the Device Behaviour Monitor and identifying the appropriate policy to apply, the Device Self-Protection component sends the suggested policy to an exchange that serves the Self-Recovery component.

Integration with Reputation System

The Device Self-Protection receives reputation updates from the Reputation System. Each message contains the current and previous reputation score. This component uses this information to verify if protective policies must be applied. Additionally, as is the case for Self-Recovery, suggested policies are sent to the same exchange to be received by the Reputation System.

Integration with Cyber-Threat Intelligence

The Device Self-Protection also determines policies according with Indicators of Compromised received from the Cyber-Threat Intelligence component, via message bus.

4.1.3 Common Plane

Hardened Encryption with SIM as RoT

Hardened encryption component provides functionalities to secure data at rest in the ARCADIAN-IoT framework. It provides encryption libraries allowing to encrypt/decrypt data using Attribute Based Encryption (ABE) paradigm and hardens the security by signing the encrypted payload with eSIM based signatures.

Integration with Self-recovery

Hardened encryption component provides functionalities to secure data at rest in the ARCADIAN-IoT framework. It provides encryption libraries allowing to encrypt/decrypt data using Attribute Based Encryption (ABE) paradigm and hardens the security by signing the encrypted payload with eSIM based signatures.

An ABE based encryption library was implemented in Go that can be cross compiled for multiple types of devices and platforms. Furthermore, bindings to use the library in Python and Java have been provided. The use of the encryption capabilities has been successfully integrated in the Self-recovery component, to protect the data stored for the recovery. Special keys can be delegated, allowing only those who are entitled to access the data. Currently the keys that are provided over the permissioned blockchain are shared through an API provided by the key management subcomponent.

Integration with Remote Attestation

Similarly, as with Self-recovery, the Hardened encryption has been successfully used by the Remote Attestation component for encryption (at the Attester) and decryption (at the Verifier) of evidence (see Section 3.2 Remote Attestation).

Integration with SIM/eSIM

The integration between the encryption technology and the SIM/eSIM security applet (Root of Trust) has been implemented and tested with several devices holding several SIM chipsets. Beyond the applet, it is provided a *device middleware* that allows the encryption component to interact with the security applet for, e.g., requesting the digital signature of the hash of the encrypted payload. It was made available *device middleware's* for Linux based IoT devices (Python) and for Android devices. In regard to Android devices, an authorization SIM applet has been integrated as well, to identify the mobile apps that are entitled/authorized to interact with the SIM.

Integration with MFA

A basic key management subcomponent of Hardened encryption has been implemented and further integrated with Authentication component, so that the appropriate cryptographic keys are delegated only to authenticated devices.

Integration with Permissioned blockchain

To enhance the trust in the system, public cryptographic keys were published on a blockchain. Currently the key management subcomponent of HE holds this information, hence needs to be trusted. Clients can request keys in all interfaces.

Integration with Biometrics

Similarly, as with Self-recovery and Remote Attestation, the Hardened encryption has been successfully used by the Biometrics for decryption of data (i.e., images encrypted by 3rd party providers).

Hardened Encryption with crypto chip system

A dedicated API orchestrator (part of the Middleware) was designed and can set up secured (TLS enabled) communications with external systems. The logic of design considered the capability for rapid generation of multiple API channels (based on different criteria), used in the same time for different purposes, correspondent with the integrated system.

For regular sensors data management and actuators command & control, by using an IoT platform as destination endpoint (for monitoring) and as originating endpoint (for commands launch), a dedicated API channel is used by Middleware Hardened Encryption to relay the traffic messages received from IoT devices through TLS to IoT platform, or vice versa.

Integration with other ARCADIAN-IoT components

For integration with other security systems, used across the device lifecycle (registration, authentication, authorisation, traffic), the Middleware provides a dedicated API channel to be used by Middleware to communicate.

The Middleware is deployed to support both API communication and RabbitMQ communication according to the ARCADIAN-IoT component (security service). Only one of the two modes is supported for a given ARCADIAN-IoT component, in order to avoid vulnerabilities brought by data redundancy. By default, RabbitMQ is the first option, with the API being the backup option.

The API has been tested with Self-aware Data Privacy (refer to section for domain B status description in Section 5 for further details). Integration with Self-aware Data Privacy was previously validated by consuming properly and in real time messages from Rabbit MQ originated by Middleware.

The integration with the message bus for consumption of events from the Device Behaviour Monitoring component was successfully tested.

The integration with the Remote Attestation system was done using the API channel mentioned above (refer to section 4.2.2. (Remote Attestation) for further details).

Permissioned Blockchain

Integration with Hardened Encryption

To enhance the trust in the system, public cryptographic keys have been published on the permissioned blockchain so that only the trusted organizations that need access to the keys are able to retrieve them.

Integration with Decentralized Identifier

The Decentralized Identifier (DID) component makes use of a dedicated DID smart contract in the final P2 prototype to publish public Decentralized Identifiers (DID:PRIV method) Documents to the Hyperledger Fabric blockchain, and so implements a decentralized Verifiable Data Registry. This makes available public key information to organizations in the ecosystem for verifying IoT Devices Verifiable Credentials. This supersedes the previous Sidetree Node integration in P1.

Integration with Verifiable Credentials

The Verifiable Credentials component resolves the DID:PRIV to retrieve its DID Document from the Verifiable Data Registry that resides on the Permissioned Blockchain.

Integration with Reputation System

The Reputation System publishes reputation information on the permissioned blockchain using the API that is provided by this component. The reputation system stores the reputation scores in the blockchain in a JSON object. This is able to be consumed by all parties that are permitted to run the permissioned blockchain for the reputation channel where it is published.

Integration with Onboarding IdP

A Trusted Organization register was seen to be needed for only allowing trusted organizations to be able to register entities (Persons, IoT Devices & Services) in the ARCADIAN-IoT Framework. Therefore, this was added during P2 integration making use of the publisher smart contract, so that only trusted organizations identified by their public DID can make use of the ARCADIAN-IoT framework.

4.2 Integration of vertical planes

This section presents the integration of the components that are part of the identity, trust and recovery planes.

4.2.1 Identity Plane

Decentralized Identifiers

Integration with Permissioned Blockchain

For P1, public Decentralized Identifiers were provided by a Sidetree implementation integrated with a Private Ethereum Blockchain. However, for P2 this was replaced by providing a DID method only resolvable to trusted organizations in the ARCADIAN-IoT framework over the Hyperledger Fabric blockchain called DID:PRIV.

Integration with Verifiable Credentials

The ARCADIAN-IOT Framework makes use of the Ledger uSelf SSI Broker / Agent solution in the Verifiable Credential´s component since P1. As regards, Decentralize identifiers:

(1) the SSI Broker is integrated with the DID:WEB method that was facilitated by an operator provisioning a public/private key pair to the SSI Broker. The DID along with its private key is used by the SSI Agent to create connections with wallets and sign Verifiable Credentials it issues.
(2) The SSI Wallet make use of Peer DIDs and makes a connection with the ARCADIAN-IoT Framework SSI Broker / Agent to issue Person and Organization VCs to SSI Wallet signed by the private key associated to the frameworks DID:WEB.
(3) The IoT Devices SSI GO Agent makes use of the public DID:PRIV method.
(4) Constrained IOT Devices support DIDs through the IoT GW middleware acting on their behalf with DID:WEB and a simple challenge response DID authentication protocol.
(5) DID:WEBs are also supported for the piloting Service Providers to publish their public keys for their registering to the framework´s Trusted Organization Registry.

Integration with Onboarding IdP

The onboarding IdP supports the registration of constrained Persons, IoT Devices, and Services to the ARCADIAN-IoT Framework through use of DID Authentication by HTTP Signatures.

Integration with Credential Recovery

For DID:PRIV DID Docs have integrated a key rotation function as part of recovering control of the DID for an IoT Device as ordered by the business application on the IoT Device. For SSI Wallets the Peer DID is not needed to be recovered se new DID based connections are made with each entity that the wallet connects too.

Network/Hardware-based authentication – SIM-based

Integration with MFA

As one of the identification factors for devices and persons (through their personal device), this component depends on the integration with ARCADIAN-IoT MFA to provide its functionality to the project. This integration has been performed and successfully tested within 2 Solution Providers technologies.

Integration with Onboarding IdP

A protected Network ID Token is onboarded by the Onboarding IdP with an ARCADIAN-IoT ID (aiotID) that is created at registration of the entity being onboarded (person or IoT device). For this purpose, an integration of the Onboarding IdP with the Network/Hardware-based authentication component was developed and successfully tested.

Biometrics

The Biometrics component is responsible of verifying people faces from a smartphone camera and from a video streamed from an Unmanned Aerial Vehicle (UAV).

Integration with MFA

- REST interface is tested and it will provide the AMQP information to ask for biometrics authentication.
- AMQP first prototype interface is working. MFA shares an image to perform face verification and the biometrics component replies if the identity of the person is correct.

Although the first prototype was already implemented including every of the previous interfaces, prototype 2 was further improve by adding the capability of updating/deletion of people’s faces and images and message encryption. Besides, most information is shared with MFA in order to provide more details about the accuracy of the verified user.

Integration with Hardened Encryption

The integration with Hardened Encryption is used to apply a new layer of security when transmitting sensitive information such as the user face. This integration was prepared for use cases A1, A2 and A3. The bash client-side scripts are integrated with Hardened Encryption, providing decryption capabilities in Biometrics allowing this component to decrypt the images encrypted by the Third-Party Provided.

Integration with Reputation System

The description regarding the integration with Reputation System & Policy Manager is described in “Reputation System & Policy Manager (UC)”.

Multi-factor Authentication – MFA

Integration with Network/Hardware-based Identification (SIM-based)

ARCADIAN-IoT MFA component receives, among other credentials, a network-based identity token associated with an ARCADIAN-IoT ID, to ascertain if there is a match. This match is verified by a subcomponent of the Network/Hardware-based Identification, which is integrated with the MFA for that purpose. Upon the receival of the confirmation of a match of the network identifiers with the ARCADIAN-IoT ID, if the same happens with the other identification factors used in that authentication process, the MFA crafts an ARCADIAN-IoT ID token to be used by the requesting entity to prove its successful authentication. This integration has been successfully achieved.

Integration with Self-Sovereign Identity – SSI (Verifiable Credentials – VC)

Like with the previous authentication factor, for the SSI the MFA receives a DID (only for constrained IoT Devices) or a VC and requests the related integrated component to confirm if it matches a given ARCADIAN-IoT ID. If so, and if the results are positive for the other identity credentials used, the MFA crafts the ARCADIAN-IoT ID token to be used by the requesting entity to prove its successful authentication. This integration of the MFA with DIDs and VC has been successfully achieved.

Integration with Onboarding IdP

The entity is onboarded by the Onboarding IdP with an ARCADIAN-IoT ID and makes use of the MFA to handle the authentication requests as part of the registration of the entity being onboarded (person, IoT device).

Integration with Biometrics

As like with the other credentials, in the case of Biometrics, there is an integration with the MFA that allows to infer if the biometrics credentials match a requesting entity that claims to have a given ARCADIAN-IoT ID. If the biometrics used match the ARCADIAN-IoT ID biometrics credentials registered, the result will be successful. In this case, and if the results are also positive for the other identity credentials used, the MFA crafts the ARCADIAN-IoT ID token to be used by the requesting entity to prove its successful authentication. This integration has been successfully achieved.

Integration with Device Self-Protection and Reputation System

The MFA integration with the Device Self-Protection and Reputation System targets having a security action with reduced human intervention, triggered by Authentication events, e.g. of repeated authentication attempts failed in short periods, which can indicate an attack tentative (e.g. Brute Force attack). This integration has been successfully achieved.

Integration with other framework components

The MFA component provides to any framework component that is entitled to verify ARCADIAN-IoT ID tokens – verify if an entity was successfully authenticated using the MFA – means to do so. This should happen with the standard verification of a JWT (JSON Web Token), including the verification of the MFA signature and of other common fields like expiration. The Self-Aware Data Privacy, Hardened Encryption, Credential Recovery, and Self-recovery components are the components that verify if an entity was successfully authenticated with this procedure.

4.2.2 Trust Plane

Verifiable Credentials

Integration with Decentralized Identifiers

The ARCADIAN-IoT´s SSI Broker/Agent is integrated with the Decentralized Identifiers component to be issued with a public DID:WEB.

It supports the issuing and verification of Person and Organization Member Verifiable Credentials to end user´s mobile SSI Wallets signed by the ARCADIAN-IoT´s public Decentralized Identifier (DID).

It supports the issuing and verification of Device Verifiable Credentials to IoT Devices signed by the ARCADIAN-IoT´s public Decentralized Identifier (DID).

It supports the authentication of constrained IoT Devices through the DID Challenge / Response authentication based on the DID:WEB method.

Integration with Onboarding IdP

The onboarding IdP supports the issuing of Verifiable Credentials to end user´s mobile SSI Wallets.

Integration with MFA

The Onboarding IdP and Broker/Agent is integrated for the persons and organization member registration and authentication flows with the Multi-Factor Authentication component.

The MFA additionally supports the issuing of a signed JWT token for IoT Devices registration and authentication flows.

Integration with Credential Recovery

SSI Wallet integrated with Credential Recovery dedicated back-up server for storing and retrieval of encrypted SSI Wallet containing its previously issued verifiable credentials.

SSI GO Agent integrated with SSI Broker to support re-issuing of Device VC for a specific Device with key rotation on its public DID.

Onboarding IdP

The onboarding IdP supports the issuing of Verifiable Credentials to end user´s mobile SSI Wallets and the registration of persons, IoT Devices, and Services to the ARCADIAN-IoT Framework. This is provided by an IdP (FE/BE) module and integrated with the SSI Broker to request connections, issuing and verification of Person and Organization Verifiable Credentials.

A registered entity is created with an ARCADIAN-IoT ID that is published to the framework´s message bus implemented on RabbitMQ.

Decentralized Identifiers

The onboarding IdP supports the registration of constrained Persons, IoT Devices, and Services to the ARCADIAN-IoT Framework through use of DID Authentication by HTTP Signatures.

Verifiable Credentials

The onboarding IdP supports the issuing of Verifiable Credentials to end user´s mobile SSI Wallets.

Integration with MFA

The entity is onboarded by the Onboarding IdP with an ARCADIAN-IoT ID and makes use of the MFA to handle the authentication requests as part of the registration of the entity being onboarded (person, IoT Device).

Integration with Network/Hardware authentication

A protected Network ID Token is onboarded by the Onboarding IdP with an ARCADIAN-IoT ID (aiotID) that is created at registration of the entity being onboarded (person, IoT device).

Permissioned Blockchain

Service Providers are registered to an Organization Trust Registry via the Onboarding IdP frontend.

Integration with other framework components

The Onboarding IdP component publishes registered entities with its ARCADIAN-IoT ID and entity type and other identity attributes to the framework´s message bus so that other components may initialise instances for the new registered entity e.g. Reputation would create a new record for the entity.

Network-based Authorization Enforcement

Integration with Reputation System

The Network-based Authorization Enforcement integrates with the project' Reputation System through ARCADIAN-IoT's RabbitMQ infrastructure. The component receives and enforces reputation-based actions (allow or deny incoming or outgoing communication) according to an entity's reputation score. The final prototype is functional for real devices which are directed to a network core testbed. For this purpose, the devices used for demonstration have a private APN configured in the production networks of 1GLOBAL. The component itself lives in the network testbed (4G and 5G) previously mentioned. The integration with the Reputation System has been successfully tested.

Integration with SIM Security Applet (Trust Information Distribution)

This component also integrates with ARCADIAN-IoT's SIM/eSIM security applet for informing it of devices' trustworthiness. This integration is performed using GSM over-the-air (OTA) services. With the devices' trustworthiness, the security applet is expected to enforce actions of self-protection and self-recovery - e.g. stop signing the hash of encrypted payloads if the device is found to be compromised and recover the normal behaviour of the device is recovered. This integration has been tested with devices from 2 Solution Providers.

Reputation System & Policy Manager

The Reputation System operates in tandem with its Policy Manager, a component for managing the policies for authorization and attestation. These policies are used by the reputation system when updating the reputation score. The policy manager has a REST API to allow the CRUD operations of policies and informs the policies to the reputation system through the *policies_queue* queue.

The reputation system when receiving information of policies maintains information of the policies in memory (e.g., through Redis for fault tolerance) to map the reputation scores into the policies to apply.

Integration with Authorization

The Reputation System when providing the reputation scores, also provides information regarding the policy that is applicable to that specific entity and reputation score. The information is sent to the *reputation_update* queue.

Integration with Device Behaviour Monitoring

The Reputation System also consumes the information provided by the Device Behaviour Monitoring to determine reputation scores of devices. This exchange occurs in the *dbm_exchange*.

Integration with IoT Device Self-Protection

The interactions with this component allow the reputation system to manage the reputation of devices, which is exchanged in the *dsp_exchange*.

Integration with Network Flow Monitoring

This exchange is performed via the *nfm_exchange*. It should be noticed that the network flow monitoring does not contain information of entities' identifiers (i.e., AIoT IDs), as such the information is not considered for the reputation score calculation. The Reputation system is able to process this information but it is not used since no identifiers exist in the message.

Integration with Network Self Protection

The Reputation System uses the information provided by Network Self Protection in the *nsp_exhange* via JSON format to determine reputation. It should be noticed that the Network Self-Protection does not contain information of entities' identifiers (i.e., AIoT IDs), as such the information is not considered for the reputation score calculation. Despite not using, the reputation system is able to parse the information sent by the NSP component.

Integration with Remote Attestation

The Reputation System uses the information provided by the Remote Attestation components through the *ra_exhange* to determine reputation.

Besides receiving the information, the reputation system can also trigger a request for attestation for instance to confirm the reputation score given to a specific device. This interaction is performed via a RabbitMQ exchange using the *attest_cue* topic.

Integration with Network Authorization

The Reputation System uses the information provided by the Network Authorization component through the *naz_exhange* to determine reputation.

Integration with MultiFactor Authentication

The Reputation System uses the information provided by the MultiFactor Authentication component through the *mfa_exhange* to determine reputation.

Integration with Onboarding IdP

This integration is very important since it is through this registration that the reputation system knows the entities that have been registered and initializes their reputation values accordingly. The information from this component is crucial for the reputation system, since it allows to recognize a new entity and to initialize the values of the reputation accordingly.

Integration with Biometrics

The Reputation System uses the information provided by the Biometrics component through the *bio_exhange* to determine reputation associated with the biometrics events.

Integration with Self-Aware Data Privacy

The Reputation System uses the information provided by the Self-Aware Data Privacy component through the *sadp_exhange* to determine reputation.

Integration with Permissioned Blockchain

The integration with the Hardened Encryption has not been concluded at the date of this deliverable. The message format to be exchanged has been already documented and agreed between parties. It is expected to be concluded before the end of the project.

The Reputation System stores information in the Permissioned Blockchain using an API that is provided by this component. The reputation system stores the reputation scores in the blockchain in a JSON object.

Integration with Self Recovery

The integration was defined but fully implemented due to delays in the specification. The involved partners will aim to finalise and demonstrate this integration by the time that Deliverable 5.5 (use cases technical and legal validation) will be submitted.

Remote Attestation

The development of ARCADIAN's Remote Attestation (RA2IoT) solution progressed in two phases. Initially, the mbedTLS library was used to handle the cryptographic operations. In the second phase, the solution evolved by replacing the mbedTLS-based cryptographic functions with those of Hardened Encryption, followed by testing and validation.

The communication between Attester and Verifier is established through Remote Procedure Calls (RPC). Here, the Verifier uses RPC to send the Attestation Request and to receive the corresponding response from the Attester.

Integration with Hardened Encryption

To facilitate an efficient and smooth integration process with minimal setbacks, the demo python implementation provided by XLAB was reused, as it already contained all essential functions. The current implementation of RA2IoT utilizes (and has been tested with) Hardened Encryption for encrypting, decrypting, and signing data exchanged between Attester and Verifier. When available, the signing and verification of signatures is facilitated by eSIM/SIM, a result of the collaboration between XLAB and 1GLOBAL.

Specifically, during the attestation process, the Attester uses the HE library to encrypt and sign the data as requested by the Verifier. Conversely, the Verifier uses the same library to decrypt and verify the signature of this data.

Integration with crypto chip-based Hardened Encryption

The integration with BOX2M's devices for supporting Remote Attestation is achieved using a specialized Verifier which integrates with IoT Middleware (BOX2M) using its API. The IoT middleware's API enables the Verifier to dispatch Attestation requests through HTTPS (by TLS) requests, accompanied by a nonce. In response, the IoT Middleware returns the measurements of claims it gathered from the IoT devices

The Attester functionality is in this case embedded in the crypto chip and performed by its firmware module, managed by BOX2M master firmware running the IoT device. The claims are signed and encrypted by the crypto chip module, decrypted by the Middleware using the correspondent key, and relayed via TLS through the API. The claims format and selection were established by BOX2M, with the claim selection being consistent between attestations.

Integration with Reputation System

The RA2IoT, in particular the Verifier, is fully integrated with the Reputation System, which acts as the Relying Party. The results of each attestation are provided to the Reputation System through a RabbitMQ exchange, influencing the computation of the reputation for the attested device or service. These results include the proportion of correct claims and the integrity level indicating the trustworthiness of the device that provided those claims. Furthermore, the Reputation System possesses the capability to request the Verifier to initiate the attestation process when necessary.

4.2.3 Recovery Plane

Self-recovery

The Self-recovery component is developed primarily as stand-alone, data backups and retrieval are enabled via a REST API, clustered storage backend (CEPH) and client-side recovery scripts written in Bash. Data about recovery operations, such as frequency and consistency of backups and the success rate of recoveries are communicated to the Reputation System via RabbitMQ. The client-side portion of the Self-recovery component is expanded with the IoT Device Self-protection component, which provides event messages that inform Self-recovery when a recovery process should be triggered.

Integration with Hardened Encryption

The bash client-side scripts are integrated with Hardened Encryption, providing encryption and decryption capabilities in the recovery process, available already in P1. During P2, the client-side scripts were ported to Python and also integrated with Hardened Encryption.

Integration with Multi Factor Authorization

The server-side component of Self-recovery interfaces with the Authorization module, relying on ID tokens from the Service Provider domain to identify persons or devices attempting to perform a backup or recovery operation and validate the permissions for such actions.

Credentials Recovery

Integration with Verifiable Credentials

SSI Wallet integrated with Credential Recovery dedicated back-up server for storing and retrieval of encrypted SSI Wallet containing its previously issued verifiable credentials.

SSI GO Agent integrated with SSI Broker to support re-issuing of Device VC for a specific Device with key rotation on its public DID.

Integration with Network/Hardware-based authentication

The Onboarding IdP supports an update of the network ID token for existent ARCADIAN-IoT entities, after the successful recovery of the SSI.

5 DOMAIN INTEGRATION

This section provides an overview of the integration of the ARCADIAN-IoT framework for each use case previously defined in D2.2 [1] spanning all three ARCADIAN-IoT domains, for the final ARCADIAN-IoT framework prototype - P2. The integration description covers the perspectives from both:

- **domain responsible** (as IoT service provider): depicts the status over both the service-specific enablers and artefacts, as well as the support that each domain provides for the integration with ARCADIAN-IoT
- **ARCADIAN-IoT technical component(s) responsible**: reports the tailoring and adaptation performed to allow execution in the domain-specific environments

5.1 Domain A - Emergency and vigilance using drones and IoT

Regarding domain A use-cases, several tasks were focused on concluding all specifications regarding interactions between domain A and all the components of the ARCADIAN-IoT framework with which domain A directly interacts. To achieve this, several actions were made to align between domain leader (LOAD), and all involved technical partners (IPN, ATOS, MAR, RISE, 1GLOBAL, UWS, XLAB). The targeted ARCADIAN-IoT involvement in the domain is depicted via the Figure 5 below:

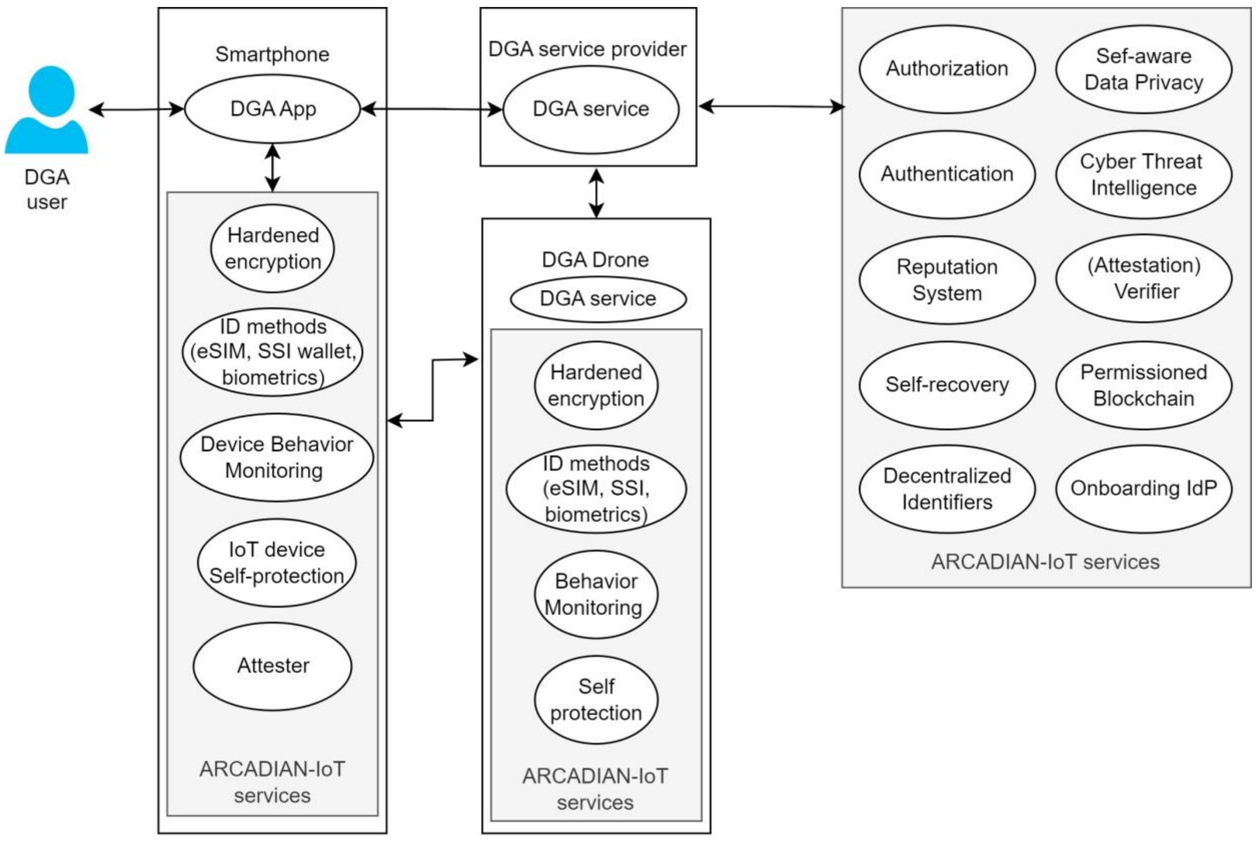


Figure 5 - High-level view of ARCADIAN-IoT involvement in Domain A

The use cases addressed in prototype P1 were A1, A2 A3. The remaining use cases (A4, A5, A6, A7) were addressed during P2. The work done regarding each one of these use cases is described in the following subsections.

5.1.1 Use case A1 - Person registration at DGA service

Overview

The work done to achieve the goals of use case A1 was related to the features allowing a user to register in the DGA service by using a mobile app. This work involved the following activities:

<u>App Development</u>

1 Elaboration of the UX/UI study and layouts of the onboarding process of the user and a very user-friendly flow to support the user on positioning his face correctly to take snapshots in each of the 5 positions used for face recognition.

2 A mobile App was created, allowing a process of registration of the user, collecting username and password, as well as minimal profile date information (name, address, contacts) essential to the operation of all domain A features.

3 User's identification data is collected and integrated with the Verifiable Credential (VC) wallet installed in the smartphone and VC SSI Agent in the backend for issuing and verification of credentials.

4 The flow of screens being presented to collect 5 photos of the user's face in 5 different positions was validated

5 After user's confirmation, these photos are encrypted with Hardened Encryption and sent to the Biometrics components, where they are validated as useful images and stored to be used later as authentication factor (described in use-case A2).

6 In each step of this flow user-friendly and helpful feedback is given to the App's user, not requiring any previous knowledge or training of the user regarding the actions to be done.

7 The end user is registered in the Onboarding IdP with an ARCADIAN-IoT ID with event raised on the message bus.

Backend Development

1 Creating server infrastructure and initial services in LOAD's development server

2 Component to support hard encryption of user's face images and sending them to Biometrics

3 Component to support video streaming for the authentication via face recognition ongoing

Component integration

DIDs

- Peer DIDs are supported and working in the end user mobile SSI Wallet.
- Public DID:WEB is working for the ARCADIAN-IoT Decentralized Identity integrated with the Ledger uSelf SSI Broker / Agent for issuing and verifying Verifiable Credentials for persons.

Biometrics

The request to register a new person in the Biometrics component is carried out via the message bus. A consumer program was implemented to receive the ARCADIAN-IoT identifier and face images of the person's face from different positions from LOAD.

Once registered, the biometrics component will decrypt each image based on the keys received by the Hardened Encryption. Then, the Biometrics component will respond with the status of this registration. This integration has been successfully achieved. AMQP interface for registration has been validated with LOAD.

The message received and the reply are composed of a JSON file as follows:

Request JSON file - Register	**Reply - Register**

{
    "encrypted": boolean,
    "img_user_b64_front": "b64_image",
    "img_user_b64_left": "b64_image"
    "img_user_b64_right": "b64_image"
    "img_user_b64_up": "b64_image"
    "img_user_b64_down": "b64_image"
}
	{
	    "message": "Human friendly description of the code"
	    "code": "…"
	}
	
	CODES:
	0: User registered correctly
	1: Error processing image - Face not detected

	2: Error processing image - More than one face detected 3: Error processing image - Other error 4: ID already in the Database 10: Other error

Verifiable Credentials & Onboarding IdP

- Registration request from the end user app results in the MFA requesting a verification presentation of the Person VC from the end user´s SSI Wallet over the framework’s SSI Broker/Agent. The Onboarding IdP will subsequently generate an ARCADIAN-IoT ID (aiotID) and provision this to the Network Authenticator and finally publish it over RabbitMQ for all concerned framework components (such as Reputation System) to be informed of a new registered entity. The registration API supports added security so that only registered organizations in the Organization Trust Register can register entities to the framework based on the private key (associated to the Organization´s public DID) signing the Register API.

Reputation System and Policy Manager

The sharing of person registration information is received by the reputation system, which then sends and Attestation Cue with the aiotID to initialize the attestation process and to inform Remote Attestation of the new aiotID. The attestation result contributes to the initialization the reputation of the registered entity.

The information that is received on the registration of an entity is exemplified below.

Received JSON	**Result of processing**

{
"aiotID": "entity 1",
"type": "Service",
"info": {
"contactEmail": "test@mail.com",
"did": "did:test", "domainURL": https://test.com
}
}
	{
	“AIoTID”: “entity1”,
	“currentScore”: 0.5,
	“previousScore”: NULL,
	“reputation_model”: “Alpha-Beta”,
	“alpha”: 0.0,
	“beta”: 0.0,
	“action”: 3
	}
	Action values:
	▪ Stop communications
	▪ Slowdown communications
	▪ Allow communications

The result of the processing is performed in the RabbitMQ and also on the permissioned blockchain, where the results is stored.

In order to know which policies apply the policy manager shares policies with the reputation system in the following format:

Received JSON

{
"minReputation": 0.0,
"maxReputation": 0.5,
“description“: “Example policy”,
“action”: 1
"actionRatio": 50%,
“AIoTIDS”:[“id1”, “id2”],
“InformationToSim”: “none”,
“ Direction”: “both”
}
	Values of **action**:
	• Stop communications
	• Slowdown communications
	• Allow communications
	ActionRatio – percentage of slowdown (when action=2)
	
	Values of **InformationToSim**: None, Self-Protection, Self-Recovery
	
	Values of **Direction**: both, uplink, downlink

Remote Attestation

As a necessary requirement for the use case, the Verifier receives Reference Values and corresponding appraisal policies for both types of devices (end user’s smartphones/Android and DGA’s drones/Linux OS) from the Domain Owner via a RabbitMQ exchange.

Upon completion of device registration, the Verifier receives an Attestation Cue from the Reputation System, which triggers the attestation procedure. The previously received Reference Values and its policies are used to appraise the claims sent by the device, whose results are sent to the Reputation System to help determine the device's initial reputation score.

Device Behaviour Monitor

The Device Behaviour Monitor operates in the background of the smartphone, continuously scanning for anomalies related to authentication and reputation events sent by the Reputation and Authentication components via RabbitMQ exchange. Its involvement in the presence of anomalies associated to the smartphone is described in use case A7.

Network-based Authentication – SIM-based

In the person/user registration flow, a network-based ID token issued by a subcomponent living at the network core to the personal device (smartphone), is used to provide unique information to be associated with the new ARCADIAN-IoT ID. This association will be stored in a subcomponent that will be responsible for the verification of the network-based credentials in the authentication flow. The unique information mentioned includes an encrypted network identifier (e.g. encrypted IMSI) of the of the network subscriber.

This component will also trigger the registration of the new ARCADIAN-IoT entity in the Network-based Authorization component, by providing to it the encrypted identifier and the newly created ARCADIAN-IoT ID.

This component was successfully integrated and tested in the context of this use case.

Network-based Authorization

In the registration flow, this component will receive the encrypted network identifier and the newly created ARCADIAN-IoT ID from the Network-based Authentication component, as previously described above. Living in the network core, this component is the only one with the keys to decrypt the network identifier (IMSI) and, at registration, associate it with the newly created ARCADIAN-IoT ID. This match will be critical for this component to enforce policies in the network (policies for a given network subscriber with a given IMSI) accordingly to the received trust-based actions to enforce provided by the Reputation System (who identify entities with the ARCADIAN-IoT ID).

This component was successfully integrated and tested in the context of this use case.

Hardened Encryption

Hardened Encryption is used to secure the backups and provide layered access policies to different level of users, for example, device owners are able to decrypt all backups, while system administrators are able to decrypt system logs only. The preferred location of data encryption is on the device itself, though resource constrains may render the encryption process unfeasible. For addressing this issue, an encryption proxy is used to receive plain data, encrypt it and either return it to the device, or forward it to the storage module of the recovery component.

Permissioned Blockchain

Supports the Trusted Organization Registry, Reputation Scores and Hardened Encryption public keys.

5.1.2 Use case A2 - Person authentication at the DGA service

Overview

The work done to achieve the goals of use case A2 was related to the features allowing a user to authenticate/login in the system with the previous registered authentication credentials, by using a mobile app. This work involved the following activities:

<u>App Development</u>

- Elaboration of the UX/UI study and layouts of the authentication process of the user when logging in to the app with user credentials
- An authentication module was added to the DGA Guard mobile App, which will present to the user a screen with login and password fields.
- After this first step, a screen using the smartphone camera is presented to the user to capture the user’s face in a front position.
- The data collected and the photo are sent to the Multi-Factor Authentication component, which will handle all the validation actions against the stored information of the user.
- In each step of this flow, user-friendly and helpful feedback is given to the App’s user, not requiring any previous knowledge or training of the user regarding the actions to be done. Once the flow described above is completed and validated, the necessary feedback is provided to the user.
-

<u>Backend Development</u>

1. Creating server infrastructure and initial services in LOAD's development server
2. Component to support face images sending to Biometrics (in common with A1 use case)

Component integration

DIDs

Same integration as mentioned in A1 (section 5.1.1)

Verifiable Credentials

Verifiable Credentials are presented from the user´s mobile SSI Wallet after a request from the MFA request to uSelf Broker which requested them from the wallet. The verified credential identity is returned to the MFA.

Remote Attestation

Remote Attestation procedures for attesting the smartphone (Android OS) are performing via watchdog-based remote attestations conducted periodically to attest the device with the previously (A1) received reference values provided by the domain owner. The evidence collected from the device is successfully appraised against the associated policies previously provided by the DGA service provider to the Verifier.

Reputation System and Policy Manager

The reputation system receives information regarding the authentication of the user, and with this information it determines the reputation score.

After the initialization in A1, the Reputation System receives information from the MFA component regarding the authentication, as follows.

Received JSON	**Result of processing**

{
"aiotID": "entity 1",
"entityType": "device",
"timestamp": “2024-01-30 10:30:00”,
“authResult”: true,
“resultExplanation”: “SSI Successful”
}
	{
	“AIoTID”: “entity1”,
	“currentScore”: 0.55,
	“previousScore”: 0.5,
	“reputation_model”: “Alpha-Beta”,
	“alpha”: 0.5,
	“beta”: 0.0,
	“action”: 3
	}
	Note: The value of CurrentScore is just do demonstrate that the reputation increases on a successful authentication, as expected

By receiving this information, the reputation of the entity is modified and updated in the exchange *reputation_updates* and also updates this information on the permissioned blockchain.

Biometrics

The multifactor authentication component will send an authentication request through the API REST created. An encrypted image is received by the Biometrics component. The component will decrypt the image and process the image in order to verify the identity of the person who requested the authentication. The results of the face verification algorithm, when an authentication is received, are provided by the same REST API. The messages exchanged can be seen in Table 7.

Device Behaviour Monitor

The Device Behaviour Monitor operates in the background of the smartphone, continuously scanning for anomalies related to authentication and reputation events sent by the Reputation and Authentication components via RabbitMQ exchange. Its involvement in the presence of anomalies associated to the smartphone is described in use case A7.

Network-based Authentication

Deployment and integrated testing of this technology in this use case has been done successfully.

A subcomponent in the core network issues a Network ID token to a given subscriber with a given ARCADIAN-IoT ID, and those credentials match is verified by a second subcomponent, integrated with the MFA.

Multi-factor Authentication (MFA)

This component was successfully integrated and tested in the context of this use case – integration with the solution provider and with the 3 identification components (Network-based, VCs and Biometrics).

Network-based Authorization

This component is ready to operate in this use case, particularly to distribute trust information to the devices secure element (SIM/eSIM), according to information received from the Reputation System. Considering that the device is a smartphone (with several mobile applications) no device communication enforcement policy should apply – only the trust information distribution to the secure element, which will trigger SIM-based self-protection and self-recovery within the context of the app (e.g. refuse to provide digital signatures when the device is found to be compromised).

Permissioned Blockchain
Supports the Reputation information updates and Hardened Encryption public keys.

5.1.3 Use Case A3 - Person retrieving and editing personal data

Overview

The work done to achieve the goals of use case A3 was related with updating user profile information, that was previously provisioned in the registration flow of use case A1. This work involved the following activities:

App Development

- Elaboration of the UX/UI study and layouts of the user profile areas and the process of retrieving and editing personal data
- The user opens the App with the intention of changing its personal data and/or face image.
- After the authentication procedure (as described in use-case A2), the user can access the profile screen, when he/she can edit its personal data
- The user will also be able to change its face image, by going through the image capturing procedure, as he did when he registered the first time.
- The data collected is sent to the DGA Guard backend and the photo set is then sent to the Biometrics component to update its records.
- In each step of this flow a user-friendly and helpful feedback is given to the App's user, not requiring any previous knowledge or training of the user regarding the actions to be done. Once the flow described above is completed and validated, the necessary feedback is done to the user.

Backend Development

- Creating server infrastructure and initial services in LOAD's development server
- Component to support sending updated face images to Biometrics component

Component integration

Remote Attestation

Same as for A2.

Hardened Encryption

Hardened Encryption is used to secure the backups and also provide layered access policies to different level users, for example, device owners are able to decrypt all backups, while system administrators are able to decrypt system logs only. The preferred location of data encryption is on the device itself, though resource constrains may render the encryption process unfeasible. For addressing this issue, an encryption proxy is used to receive plain data, encrypt it and either return it to the device, or forward it to the storage module of the recovery component.

The SIM ecosystem for acting as RoT in the HE process was fully integrated within this use case. It provides the expected digital signatures of the hash of the encrypted payloads, and is ready to receive trust-based information from the Network-based Authorization component. The SIM also informs the device operating system of the apps allowed to interact with it, identifying these apps within an authorization mechanism embedded in it.

Reputation System and Policy Manager

The reputation system receives information of this event from the self-aware data privacy component, regarding the operation of person, editing or retrieving information in the service. The reputation is updated according to the information received.

From the self-aware data privacy component the information is shared as follows.

Received JSON from self-aware data privacy	Result of processing

{
"aiotID": "entity 1",
"URL": "https://DGA.service",
"method": "POST",
"User": "tokenOfUser",
"OperationPerformed": "encrypt",
"source": "device1",
"destination": "DGA service"
}
	{
	"AIoTID": "entity1",
	"currentScore": 0.56,
	"previousScore": 0.55,
	"reputation_model": "Alpha-Beta",
	"alpha": 0.55,
	"beta": 0.0,
	"action": 3
	}

In addition, the reputation system also receives information from the DGA, as follows.

Received JSON from DGA	Result of processing

{
"aiotID": "entity 1",
"typeOfOperation": 2,
"timeOperation": "2024-01-30 10:30:00",
"statusOperation": 1,
"OperationPerformed": "encrypt",
"source": "device1",
"destination": "DGA service"
}
Values of Type of Operation: 1-edit, 2-read
Values of Status of Operation: 1-success, 0-not successful
	{
	"AIoTID": "entity1",
	"currentScore": 0.56,
	"previousScore": 0.55,
	"reputation_model": "Alpha-Beta",
	"alpha": 0.55,
	"beta": 0.0,
	"action": 3
	}

Biometrics Component

The biometrics component will receive a request to update or delete the image stored in the registration step (A1). The biometric component will update or delete the image and provide the result of this operation one completed by by RabbitMQ, with an exchange named "bio_exchange". This interface communicates the instructions of Registration, Update and Delete user information.

The message received and the reply for the update instruction are composed of a JSON file as follows:

Request JSON file - Update	**Reply - Update**

{
    "encrypted": boolean,
    "img_user_b64_front": "b64_image",
    "img_user_b64_left": "b64_image"
    "img_user_b64_right": "b64_image"
    "img_user_b64_up": "b64_image"
    "img_user_b64_down": "b64_image"
}
	{
	    "message": "Human friendly description of the code"
	    "code": "..."
	}
	
	CODES:
	0: User updated correctly
	1: Error processing image - Face not detected
	2: Error processing image - More than one face detected
	3: Error processing image - Other error
	4: Error updating user – No user with that ID
	5: User updated correctly - No changes made
	10: Other error

Device Behaviour Monitor

The Device Behaviour Monitor operates in the background of the smartphone, continuously scanning for anomalies related to authentication and reputation events sent by the Reputation and Authentication components via RabbitMQ exchange. Its involvement in the presence of anomalies associated to the smartphone is described in use case A7.

Network-based Authorization

Same as A2.

Permissioned Blockchain

Supports the Reputation information updates and Hardened Encryption public keys.

5.1.4 Use Case A4 - Person Requesting a DGA Service

Overview

The work done to achieve the goals of use case A4, related to the requesting of drone service via the mobile App to follow him/her from its location to its destination. This request is evaluated in terms of cybersecurity, and a drone (also evaluated by the system) is selected to perform the service.

App Development

- The user requests a drone to a desired service location, sending the necessary personal data encrypted with RoT information (location and destination).

Backend Development

- After verifying the user's identity in Use Case A3, its location, and the requesting app and device trustworthiness (via the reputation system), a drone is selected from the available ones – the drone itself is selected considering that IoT device reputation information.
- The selected drone identity integrity data is retrieved from its hardware (RoT) and attested, assuring the device trustworthiness (no impersonation) before granting it access to the user's personal data.
- The device, if found trustworthy, is able to decrypt the data with RoT information, allowing it to meet the person and attest its identity. The person is informed that a trustworthy drone has its location and identification for performing the service (self-aware data privacy).

Drone

- Software environment of the drone (Nvidia Jetson and VM based on QEMU) regarding person tracking and physical threat detection ongoing, and also support IPN to access several kernel functions
- Mechanical building of two drones
- Electronic assembly of two drones

Component integration

Device Behaviour Monitor

The Device Behaviour Monitor operates in the background of the smartphone, continuously scanning for anomalies related to authentication and reputation events sent by the Reputation and Authentication components via RabbitMQ exchange. Its involvement in the presence of anomalies associated to the smartphone is described in use case A7.

Remote Attestation

Remote Attestation procedures for attesting Linux-based drones are supported. This involves watchdog-based remote attestations conducted periodically to attest the compliance of the device with the reference values provided by the domain owner. The evidence collected from the drone is successfully appraised against the associated policies previously (A1) provided by the DGA service provider to the Verifier.

Reputation System and Policy Manager

The reputation system receives information of this event and updates the reputation score accordingly. The information is provided by the DGA service, using the information the same information fields as presented in A2.

Network-based Authorization

Same as A2.

Hardened Encryption

Same as A3.

Permissioned Blockchain

Supports the Reputation information updates and Hardened Encryption public keys.

5.1.5 Use Case A5 – DGA Service

Overview

The work done to achieve the goals of use case A5, when, after execution of use cases A2 and A4, and being granted with a service and having the necessary data, a drone needs to meet and identify the person that requested it and proceed with the vigilance service.

Drone

1. The drone moves to the location of the requested service.
2. Arrived at the given position, the drone informs ARCADIAN-IoT DGA services that it has arrived the location of the service.
3. ARCADIAN-IoT DGA services inform the person that the drone has arrived, what is the device security and privacy reputation, and that he/she should get closer to it and allow biometric identification.
4. A process of IoT devices mutual authentication (person personal device and drone) with data that included location validation between drone and smartphone, and video streaming to Biometrics component, mediated by ARCADIAN-IoT DGA services. All the private data is securely exchanged in this process is encrypted with RoT information.
5. Drone receives the user identity confirmation and the tracking frame to allow the user tracking.
6. When the process of identification is successful, the drone follows the user and, if something abnormal is detected, data about the event is collected, encrypted with RoT information, and sent to DGA services. In DGA services, the event data is decrypted for being analysed by an operator, who performs the follow-up measures needed.

App Development

- Using the App, the person receives the drone's information that it has arrived, and the person moves to the service location to meet the drone and allowing it to identify her/him using the several simultaneous mechanisms described below.
- If the DGA facial recognition service identifies the user, the App received a confirmation message, to inform the user that can then start walking.
- The service ends when the user informs, using the app, DGA services that it has arrived the desired location and that the drone service is no longer needed.

Backend Development

- Video is streamed and hard encrypted by the drone to the biometric component to confirm the identity of the user.
- If the DGA facial recognition service identifies the user, a confirmation message will be sent to the drone and to the user. The user can then start walking.
- The service ends when the user informs, using the app, DGA services that it has arrived the desired location and that the drone service is no longer needed. At the end of the service, all the user personal data is deleted from the drone, and any data needed in DGA services about the user or the service is kept encrypted.

Component integration

DIDs

- Public DID:WEB is working for the ARCADIAN-IoT Decentralized Identity integrated with the Ledger uSelf SSI Broker / Agent for issuing and verifying Device Verifiable Credentials for IoT Devices for an onboarding request from the business logic. .

- IoT Devices onboarding integrated with DID:PRIV.

Verifiable Credentials & Onboarding IdP

- Integrated onboarding Request to uSelf GO Agent on the drone IoT Device results in it being issued with a DID:PRIV published on the Permissioned Blockchain and being Issued with a Device Verifiable Credential.
- Integrated registration request from the IoT Device business logic to generate an ARCADIAN-IoT Identity (aiotID) in the Onboarding IdP.
- Integrated request from the IoT Device business logic to verify its Verifiable Presentation and return a token from the MFA.

Biometrics

The Biometrics component which is attached to a RabbitMQ queue will receive a verification request. This request contains a JSON file with the URL of the video that is streamed from the drone operated by LOAD. This JSON file is encrypted by DGA and decrypted by the Biometrics component thanks to the integration between Hardened Encryption, DGA and Biometrics. The JSON file is composed as follows:

```
{
    "encrypted": boolean,
    “URLVideoDrone”:”protocol://IP:Port/” (full URL - String)
}
```

The video is formatted in H264 with a resolution of 1280x720 at 25 frames per second.

Once verified the person, the results will be shared with the DGA service. The results are composed in the following form:

```
{
    "verified": true,
    "accuracy": acc,
    "coordinates": [x, y, w, h],
    "image_size": [W, H]
}
```

Device Behaviour Monitoring

The Device Behaviour Monitor operates in the background of the smartphone, continuously scanning for anomalies related to authentication and reputation events sent by the Reputation and Authentication components via RabbitMQ exchange. Its involvement in the presence of anomalies associated to the smartphone is described in use case A7.

Reputation System and Policy Manager

The reputation system receives information of this event, which includes the status of the devices (e.g., mission Started, or mission completed) and updates the reputation score accordingly.

The reputation system also receives information from the DGA, as follows.

Received JSON from DGA	Result of processing
{ "aiotID": "entity 1", "statusDevice": 1,	{ “AIoTID”: “entity1”, “currentScore”: 0.57,

"timeOperation": "2024-01-30 10:30:00", } Values of Status of Device: 1-missionStarted, 2-missionComplete	"previousScore": 0.56, "reputation_model": "Alpha-Beta", "alpha": 0.56, "beta": 0.0, "action": 3 }

Network-based Authentication

This component is ready to operate in this use case. A subcomponent in the core network issues a Network ID token to a given subscriber (drone) with a given ARCADIAN-IoT ID, and those credentials match is verified by a second subcomponent, integrated with the MFA. Particularly and specifically to this use case, this component is ready to cope with the IoT device MQTT communication.

Multi-factor Authentication (MFA)

This component is ready to operate in this use case, integrating with the solution provider and with the 2 identification components used for IoT devices (Network-based authentication and the Decentralized Identifiers).

Network-based Authorization

This component is ready to operate in this use case, particularly to enforce trust-based communication policies in the cellular network and to distribute trust information to the devices' secure element (SIM/eSIM). This operation is according to information received from the Reputation System. As in A2, the trust information distribution to the secure element will trigger SIM-based self-protection and self-recovery (e.g. refuse to provide digital signatures when the device is found to be compromised).

Self-recovery

The Self-recovery component is composed of a storage server, that exposes a REST API via HTTP/S and client-side (on-device) scripts, that allow devices interface with the storage server and store and retrieve backups. The types of data that are stored vary from device to device, ranging from configurations that are required for the device to operate normally, system logs, on-device application data and if necessary, data gathered by sensors. In case of a device failure, the configuration is applied to a new device based on the backup.

Hardened Encryption

Hardened Encryption is used to secure the backups and also provide layered access policies to different level users, for example, device owners are able to decrypt all backups, while system administrators are able to decrypt system logs only. The preferred location of data encryption is on the device itself, though resource constrains may render the encryption process unfeasible. For addressing this issue, an encryption proxy is used to receive plain data, encrypt it and either return it to the device, or forward it to the storage module of the recovery component.

Permissioned Blockchain

Supports the DID:PRIV method for IoT Devices, Trusted Organization Registry, Reputation and Hardened Encryption.

5.1.6 Use Case A6 – Drone security or privacy incident

Overview

In A6, when under normal drone operation the device is subject to a security or privacy incident, there are several possible causes, such as:

- Privilege escalation
- Device loss or theft
- Distributed Denial of Service
- NMAP scans
- Attempted Encryption of Large Number of Files (Ransomware)

Backend Development

The backend supporting this domain have been adapted to integrate with security or privacy components of the ARCADIAN-IoT Framework (e.g., Device Behaviour Monitoring, Device Self-Protection CTI). This allows for the components to monitor the drone and issue warnings or suggest protective measures.

Component integration

Device Behaviour Monitor

The Device Behaviour Monitor operates in the background of the drone, continuously scanning for anomalies related to system call sequences collected on the device, as well as anomalies related to authentication and reputation events sent by the Reputation and Authentication components via RabbitMQ exchange. When an anomaly is detected, the Device Behaviour Monitor alerts the Reputation System and Cyber-Threat Intelligence components via the message bus about the detected anomaly. In parallel, it also sends this anomaly data to the Device Self-Protection component to determine the appropriate policies.

Cyber Threat Intelligence

Upon receiving threat-related information from Device Behaviour Monitor, The CTI component parses and processes it to generate MIPS-based Indicator of Compromises (IoCs). The processed IoCs are then further analysed by the ML-based subcomponents by adding complementary details (e.g., threat level). Once a complete IoC is generated, it is locally stored to update IoC database, and also forwarded to output interface.

Device Self-Protection

Upon receiving threat-related information from the Device Behaviour Monitor, Reputation System, or Cyber-Threat Intelligence, the Device Self-Protection component identifies and applies the most suitable security policies to the device. At the same time, it forwards these suggested policies to Domain Owners to decide on further actions, if any. Additionally, these policy suggestions are also sent to the Self-Recovery component, to evaluate them, decide the actions to be taken (apply or not), and report the decisions back to the Device Self-protection. Finally, the Reputation System receives the confirmation from Device Self-protection of whether or not the policies were applied by itself and self-recovery.

Remote Attestation

Remote Attestation attests the drone and its data both periodically and on-demand, as requested by the Reputation System. The Attestation Result comprises two fields, with values between 0 and 1, where the first field reflects the proportion of claims adhering to the Verifier's appraisal policies. If a response is received with a nonce differing from the one sent in the Attestation Request, or if there are issues with the cryptographic procedures on the Verifier's side, both fields are set to 0. In case of an invalid response from the Attester, both fields are set to –1. In all scenarios, the Attestation Result is sent to the Reputation System for further actions.

Reputation System and Policy Manager

The reputation system receives information from DGA service, regarding devices (e.g., if it is lost, or if it is being recovered) and updates the reputation score accordingly.

The reputation system also receives information from the DGA, as follows.

Received JSON from DGA	**Result of processing**

{
"aiotID": "entity 1",
"statusDevice": 1,
"timeOperation": "2024-01-30 10:30:00",
}
Values of Status of Device: 1-lost, 2-recovered
	{
	"AIoTID": "entity1",
	"currentScore": 0.56,
	"previousScore": 0.57,
	"reputation_model": "Alpha-Beta",
	"alpha": 0.57,
	"beta": 0.0,
	"action": 3
	}
	Note: assuming a lost event, the reputation score decreases.

Network-based Authorization

This component is ready to operate in this use case, particularly to enforce trust-based communication enforcement policies in the cellular network and to distribute trust information to the devices' secure element (SIM/eSIM). This operation is according to information received from the Reputation System. The communication enforcement based on trust information can automatically stop network related incidents like private information leakage or DDoS. As in other previously described use cases, the trust information distribution to the secure element will trigger SIM-based self-protection and self-recovery (e.g. refuse to provide digital signatures when the device is found to be compromised).

Verifiable Credentials & Credential Recovery

The SSI GO Agent on the IoT Device supports a re-onboarding for a specified DID as requested from the business logic on the IoT Device. Once requested it will perform a key rotation for that DID on the blockchain and request to be re-issued with its device Verifiable Credential based on the device´s specific fingerprint.

DIDs

Public DID:PRIV supports an update so that the SSI GO Agent will generate a new key pair and the DID Doc stored on the permissioned blockchain is updated with the new public key.

Permissioned Blockchain

Supports the DID:PRIV method for IoT Devices, and also updates for the IoT Device´s Reputation information.

Self-recovery

Same as A5.

5.1.7 Use Case A7 – Personal device security or privacy incident

Overview

In A7, when under normal personal device operation the device is subject to a security or privacy incident, there are several possible causes, such as:

- Privilege escalation
- Device loss or theft
- Distributed Denial of Service
- NMAP scans
- Attempted Encryption of Large Number of Files (Ransomware)

<u>Backend Development</u>

The backend supporting this domain have been adapted to integrate with security and privacy components of the ARCADIAN-IoT Framework (e.g., Device Behaviour Monitoring, Device Self-Protection CTI). This allows for the components to monitor the drone and issue warnings or suggest protective measures.

Component integration

Device Behaviour Monitor

The Device Behaviour Monitor operates in the background of the smartphone, continuously scanning for anomalies related to authentication and reputation events sent by the Reputation and Authentication components via RabbitMQ exchange. When an anomaly is detected, the Device Behaviour Monitor alerts the Reputation System and Cyber-Threat Intelligence components via the message bus about the detected anomaly. In parallel, it also sends this anomaly data to the Device Self-Protection component to determine the appropriate policies.

Cyber Threat Intelligence

Upon receiving threat-related information from Device Behaviour Monitor, The CTI component parses and processes it to generate MIPS-based Indicator of Compromises (IoCs). The processed IoCs are then further analysed by the ML-based subcomponents by adding complementary details (e.g., threat level). Once a complete IoC is generated, it is locally stored to update IoC database, and also forwarded to output interface.

Device Self-Protection

Upon receiving threat-related information from the Device Behaviour Monitor, Reputation System, or Cyber-Threat Intelligence, the Device Self-Protection component determines the most suitable policies for application. Since these policies cannot be directly applied to Android devices, the component forwards the recommended policies to Domain Owners, who will then decide on the necessary actions. Additionally, these policy suggestions are also sent to the Self-Recovery component, to evaluate them, decide the actions to be taken (apply or not), and report the decisions back to the Device Self-protection. Finally, the Reputation System receives the confirmation from Device Self-protection of whether or not the policies were applied by itself (which is negative in this use case) and self-recovery.

Remote Attestation

Remote Attestation attests the smartphone and its data both periodically and on-demand, as requested by the Reputation System. The Attestation Result comprises two fields, with values between 0 and 1, where the first field reflects the proportion of claims adhering to the Verifier's appraisal policies. If a response is received with a nonce differing from the one sent in the Attestation Request, or if there are issues with the cryptographic procedures on the Verifier's side, both fields are set to 0. In case of an invalid response from the Attester, both fields are set to –1. In all scenarios, the Attestation Result is sent to the Reputation System for further actions.

Reputation System and Policy Manager

Same as A6.

Network-based Authorization

This component is ready to operate in this use case, particularly to distribute trust information to the devices' secure element (SIM/eSIM). This operation is according to information received from the Reputation System. As in other previously described use cases, the trust information distribution to the secure element will trigger SIM-based self-protection and self-recovery (e.g., refuse to provide digital signatures when the device is found to be compromised).

Self-recovery

The Self-recovery component is composed of a storage server, that exposes a REST API via HTTP/S and client-side (on-device) scripts, that allow devices interface with the storage server and store and retrieve backups. In case of a device failure, the configuration is applied to a new device based on the backup.

Hardened Encryption

Hardened Encryption is used to secure the backups and also provide layered access policies to different level users, for example, device owners are able to decrypt all backups, while system administrators are able to decrypt system logs only. The preferred location of data encryption is on the device itself, though resource constrains may render the encryption process unfeasible. For addressing this issue, an encryption proxy is used to receive plain data, encrypt it and either return it to the device, or forward it to the storage module of the recovery component.

Verifiable Credentials & Credential Recovery

The SSI Wallet integrates with a dedicated Credential Recovery back-up server to store encrypted wallet credentials and recover them when needed. Supports a dedicated backup server for storing user´s backups based on their user identity.

Permissioned Blockchain

Supports the Reputation information updates and Hardened Encryption public keys. Additionally, supports key rotation for the DID:PRIV method.

5.2 Domain B - Secured early monitoring of grid infrastructures

To conclude the integration of the smart grid service with relevant components of the ARCADIAN-IoT framework, several actions were made. Namely, there were several design discussions and decisions (e.g. regarding DID support) involving domain leader (BOX2M) and involved technical partners / component owners (i.e. IPN, ATOS, MAR, 1GLOBAL, UC, XLAB), which led to the consolidation of the use cases. The targeted ARCADIAN-IoT involvement in the domain is depicted via the Figure 6 below:

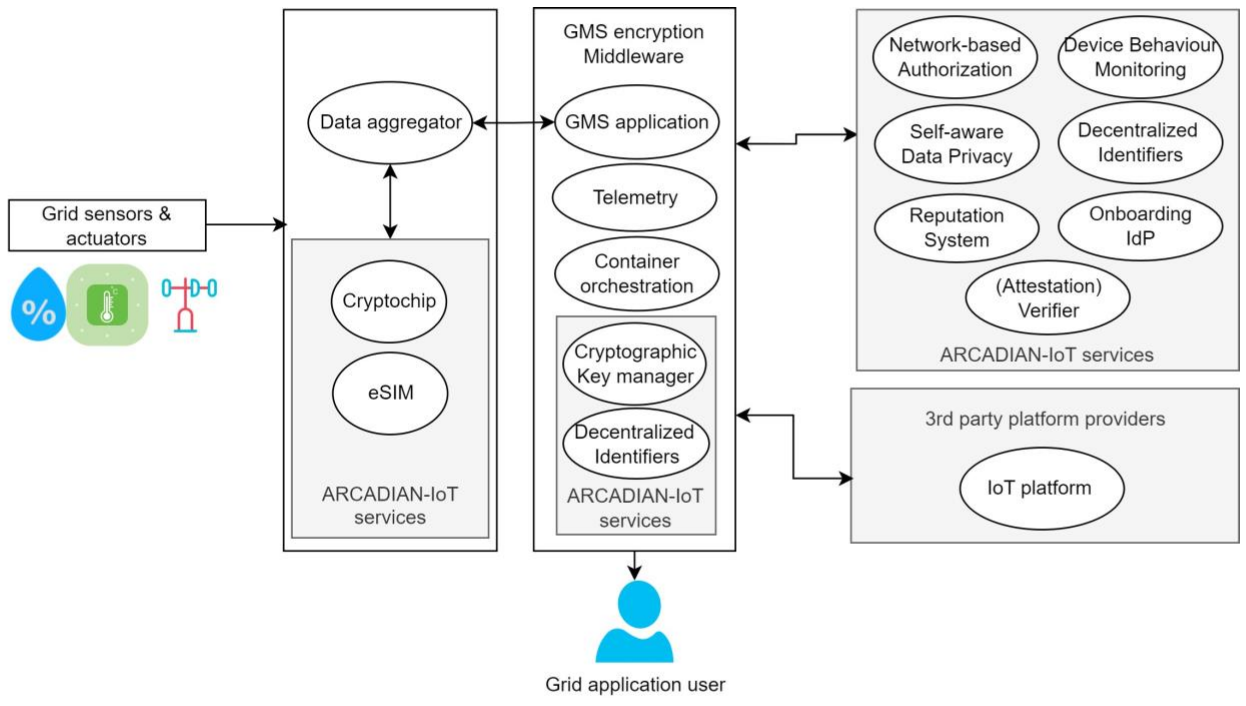


Figure 6 - High-level view of ARCADIAN-IoT involvement in Domain B

5.2.1 Use case B1 - New device registration

Overview

The work done to achieve the goals of use case B1 was related to the features allowing an IoT device to register in the encryption/ decryption middleware system. Registration takes place in 2 scenarios: 1) when a new IoT device is provisioned; 2) when an existing device has the crypto chip replaced or reprovisioned with new key pair due to operational lifecycle changes (e.g. broken crypto chip, stolen & recovered device, sabotage, hacking or cloning susceptions). Industrial IoT device registration is a dual-side activity, both being performed physically on the device set-up, and into the middleware. The process is analogous to that of hardware token service in the banking sector.

Moreover, to make use of – and be identified by - the ARCADIAN-IoT framework, IoT devices must be registered in it, which demands new associated ARCADIAN-IoT Identities (aiotID). Likewise, IoT devices must be issued with a Decentralized Identifier to perform DID authentication. Both activities were not able to be directly supported by the IoT device itself but enabled by using Middleware as providing this component capability on behalf of the IoT Device[8].

This work involved the following activities:

<u>Hardware Development</u>

- Design and deployment of device motherboard, powered by microcontroller, equipped with hardware crypto chip and with configuration port, according to initial established HLD (high-level design)
- Design and development of communication and sensors interfacing plugin units, according to initial established HLD (high-level design)
- Test of device boards for hardware functionalities validation (power supplies, short circuits, power budget, signals & interfaces functionalities, sudden operations events)
- Design and implementation of a customised laboratory set-up, for running all current and further device exploitation tests

<u>Firmware Development</u>

- Design and deployment of firmware structure and agents, including main firmware engine, communication firmware agent, edge computing firmware agent, system firmware engine for encryption and decryption
- Firmware tests for individual agents and holistically (there were used GSM and ETH telecommunication modules, and 2 types of grid sensors interfacing boards)
- Selection of ST Electronics firmware as encryption and decryption system firmware engine - integrated / called by device microcontroller main firmware engine
- Provisioning of credentials (user, pass) for device registration and device unique authorization encryption key; test of the support for multiple devices (to simulate a fleet).

<u>Backend Development</u>

- Set up the cloud computing infrastructure on Microsoft Azure
- Set up the Docker infrastructure of Middleware (the cloud side encryption and decryption system component)
- Middleware code development (provisioning, monitoring of operations performed, encryption and decryption system – including keys database, API)
- Middleware backend tests:
    - Test the encryption and decryption in correlation with device's provisioned keys
    - Test the API with BOX2M IoT platform as external interface
    - Test the support for DID:WEB endpoints, integrating client support for DID
- Provisioning of credentials (user, pass) for device registration into telemetry protocol broker, and provisioning of device unique authorization encryption keys into encryption and decryption system database; test with multiple devices to simulate a fleet

<u>Frontend Development</u>

[8] The support of DIDs on the MCU with DIDCOMM specification was trialled for the first prototype P1, however the crypto operations were found to consume too many resources. As such, no integration was achieved with the MCU for Decentralized Identifiers in P1.

- Middleware frontend code development (provisioning web pages for keys, API, monitoring process)
- Middleware frontend tests (creation, removal, modifications, view)

Component Integration

DIDs

Public DID:WEB is supported on the IoT Device GW middleware with DID Challenge / Response authentication to the Onboarding IdP backend.

Onboarding IdP

Registration of the constrained IoT device and generation of ARCADIAN-IoT identity (aiotID) is supported. Authentication of the IoT Device is also supported based on the DID Challenge / Response authentication which will return a self-describing ID token (obtained from the MFA) that can be used as part of the authorization of IoT device to upload its connected sensors data.

Supports securing Registration of constrained IoT Devices with HTTP Signature with the Organizations DID (store in Trusted Organization Registry).

Hardened Encryption using crypto chip system

The registration of multiple industrial IoT devices was designed, implemented and successfully tested. Additional tests were done with BOX2M live IoT platform, connected via TLS to middleware, in a real deployment scenario.

A set of scenarios simulating malicious attacks during device registration process were executed to test the behaviour of the encryption & decryption system. Both the specific cases of a wrong, correct or missing any of device ID, credentials and keys, were performed sequentially, both in the device side and in the middleware side.

Self-aware data privacy

Previously, dedicated experiments were done testing the self-aware data privacy component's integration with BOX2M live IoT platform, by connecting via secure TLS connection to the encryption / decryption middleware via BOX2M's dedicated API.

At this point the data processing component is capable of consuming messages from a queue on the RabbitMQ instance containing data sent from IoT devices. These messages are then encrypted through Hardened Encryption and pushed to a different queue. Currently both queues are durable, ensuring compatibility with components that assume encryption as well as those who do not.

Device Behaviour Monitoring

The Device Behaviour Monitoring component operates remotely, processing updates on reputation, authentication, and IoT device events from the Middleware. These updates are received through a RabbitMQ exchange, as the component cannot run directly on the devices. A correct registration will be consumed via message bus by the Device Behaviour Monitoring.

Reputation System and Policy Manager

The reputation system receives information regarding the device that was registered. This information is provided by the middleware and consumed via RabbitMQ (the backup channel being the API). This process triggers the transmission of an Attestation Cue towards the Remote Attestation's Verifier and consequently the receival of the associated Attestation Result.

The information of the new device registration is received through the SSI exchange, as demonstrated in the A1 use case.

Remote Attestation

The processing of an Attestation Cue from the Reputation System, containing the target device's aioID, successfully initiates the Remote Attestation procedure. Based on the previous collection of Reference Values[9] associated to the device ID (pre-condition), the Verifier sends an attestation request containing both the predefined claim selection (mentioned in section 4.2.2) and a nonce via the REST API provided by the IoT Middleware. The evidence is appraised against policies also previously provided by the domain owner to the Verifier.

Network-based Authorization

This component is ready to operate in this use case. In the registration flow, it will receive the encrypted network identifier and the newly created ARCADIAN-IoT ID, similarly to what was described in A1. Living at the network core, this component will associate the network identifier with the newly created ARCADIAN-IoT ID. This match will be critical for the component to enforce policies in the network (policies for a given network subscriber with a given IMSI) accordingly to the received trust-based actions to enforce provided by the Reputation System (who identify entities with the ARCADIAN-IoT ID).

Multi-factor Authentication (MFA)

This component supports the generation of the ARCADIAN-IoT ID token at the end of a successful authentication which, in the case of this domain, uses the DID and the crypto chip data (as a hardware-based ID).

Permissioned Blockchain

Supports the publishing of Reputation information.

5.2.2 Use case B2 - GMS IoT device data gathering and transmission process

Overview

The work done to achieve the goals of use case B2 was related to the features allowing an IoT device to transmit the gathered and encrypted data from sensors. Encryption of data takes place every time a device runs such operations. Encryption is always a dual side and serialised activity: 1) in the device by the encryption agent and 2) in the middleware by relaying the traffic via API, by TLS encryption, to an external IoT platform.

This work involved the same activities (readiness level) as for B1, given that both B1 and B2 are serialized, correlated and tested together.

Component Integration

Hardened Encryption using crypto chip system

The registration of multiple industrial IoT devices was designed, implemented and successfully tested. Additional tests were done with BOX2M live IoT platform, connected via TLS to

[9] BOX2M's crypto chip system uses the following fields as claims: 'catalogue id', 'device id', 'equipping class', 'firmware version', 'vendor name'

middleware, in a real deployment scenario.

A set of scenarios simulating regular device traffic process (sending sensors data path) were executed to test the behaviour of the encryption & decryption system. Both the specific cases of correct keys or HTTP certificates, settings and provisioning, were performed sequentially, both in the device side and in the middleware side.

Device Behaviour Monitoring

As the Device Behaviour Monitoring component cannot run directly on this kind of constrained devices due to their nature, the component operates remotely on any desired location (currently at IPN's premises for the prototype demonstration), processing updates on reputation, authentication, and IoT device events from the Middleware. These updates are received through a RabbitMQ exchange.

Self-aware data privacy

The self-aware data privacy component integrates with Middleware by listening in on a RabbitMQ queue (receiving data from BOX2M's live IoT platform), and encrypt the data in the messages via hardened encryption, pushing them to another queue afterwards. Messages aren't consumed from the source queue, this ensuring that the data can be consumed by either, depending on whether hardened encryption is expected or not.

Reputation System and Policy Manager

Reputation system is integrated with the Middleware by RabbitMQ as main communication path and consumes events relating to the IoT device Behaviour produced by Middleware. API channel could be a backup. Such information is used to update the reputation score of the device.

BOX2M designed a specific policy for abnormal situations regarding its devices lifecycle which leads to upgraded / downgraded device reputation. The policy covers all stages (registration, authentication, traffic). In this use case, the relevant criteria, assessed during traffic stage, includes the correctness of a) keys used by a device and b) HTTPS certificates used by Middleware to send specific messages to the IoT platform. Integration was tested and validated.

The infomation that is received from the middleware is exemplified below.

Received JSON from middleware	**Result of processing**

{“message”:
{"ClientId":"l42nebnokl",
"Username":"l42nebnokl",
"Topic":"device/l42nebnokl",
"Payload":"{\r\n \u0022vbat\u0022: 3.06,\r\n \u0022ts\u0022: 1678729239871,\r\n \u0022values\u0022: {\r\n \u0022o6ge80yxyr\u0022: {}\r\n }\r\n}","TimestampUTC":"2023-03-14T23:05:10.5014637Z"},
"ArcadianId":"api.box2m.io:b666ca65-0faa-4e8b-ane}
	{
	“AIoTID”: “api.box2m.io:b666ca65-0faa-4e8b-ane”,
	“currentScore”: 0.55,
	“previousScore”: 0.50,
	“reputation_model”: “Alpha-Beta”,
	“alpha”: 0.0,
	“beta”: 0.0,
	“action”: 3
	}

Network-based Authorization

This component is ready to operate in this use case, particularly to enforce trust-based communication enforcement policies. This operation is according to information received from the

Reputation System. The communication policy enforcement based on trust information can automatically stop network related incidents like private information leakage, unauthorized control of the device, or DDoS.

Permissioned Blockchain

Supports the publishing of Reputation information.

5.2.3 Use Case B3 – Service request from third-party IoT monitoring platforms

Overview

The work done to achieve the goals of use case B3 was related to the features allowing the IoT platform to operate the actuators connected by IoT devices, to dispatch commands to them.

Commands may relate to the automation of the movement of contactors, separators, valves and other direct action in field elements of grid & utility industry, with different impacts according to the exact application domain. As such, timing, data integrity and data accuracy are necessary criteria for commands to be executed properly.

A command is either generated by allowed human operators or automatically preconfigured and generated by IoT platform itself, being then encrypted by TLS and sent to Middleware via a dedicated API channel. The Middleware decrypts the command message, and internally relays it to its Hardware Encryption System. This system is responsible for identifying the proper encryption key from the database (i.e. associated to the traffic stage category and to the Device unique ID), encrypt the command and build the telemetry message operated by MQTT message broker appending the encrypted command as content. The Device then receives the message and applies the correspondent traffic decryption key; for this, its master firmware calls the encryption firmware module which operates the keys provisioned in the crypto chip. Once the message is unencrypted, the master firmware calls the edge firmware module, which executes the command accordingly by triggering the proper actuation element.

Each of the operations described above has a "local control feedback loop", which involves a command confirmation message from the IoT Device to the IoT platform - this message is treated just as another message from B2 use case. The "local control feedback loop", a familiar term for telematics, is like any other "sensor data" for the Hardware Encryption System.

This work involved the following activities:

<u>Hardware Development</u>

- Design and development of specialised actuation board, with multiple command ports, plug-in motherboard, able to operate with other plugin boards (for monitoring, communication) according to initial established HLD (high-level design)
- Tests of the actuation board for hardware functionalities validation (power supplies, short circuits, power budget, signals & interfaces functionalities, sudden operations events)
- Design and build a customised laboratory set-up, for running all current and further board tests, including the "local control feedback loop" described above

<u>Firmware Development</u>

- Design and deployment of edge computing firmware agent, enhanced with functions for executing local commands

- Integrated tests of firmware, involving different telecommunication boards (GSM, LTE and ETH), multiple network operators (1 regular network operator architecture and 2 MVNO network architectures), and 2 types of field actuators connected to actuation board. Both field actuators are usual components chosen by grid & utilities companies for setting up field operations

<u>Backend Development</u>

- Middleware backend tests: test the correlation between encryption and decryption with device provisioned keys; test of API with BOX2M IoT platform as external interface. All tests aimed for a complete operation (command sent -> feedback regarding command execution) from the usability perspective
- Multiple devices were tested (to simulate a fleet) by forging the IoT devices simulator

<u>IoT platform used into demo – frontend and backend development</u>

- Dedicated menu and frontend page
- Backend command engine development
- Backend development for correlation of commands with feedback received
- Backend development (upgrade or reporting & events modules)

<u>IoT devices simulator – designed and built for fasting the integration with other security systems</u>

- Configuration to act on actuation triggers
- Provision and tests

Component integration

Hardened Encryption using crypto chip system

The registration and traffic performed by multiple industrial IoT devices were successfully tested. Additional tests were done with BOX2M live IoT platform, connected via TLS to middleware, in a real deployment scenario.

A set of scenarios simulating regular device traffic (getting commands path) process were executed to test the behaviour of the encryption & decryption system. Multiple specific cases of correct keys, HTTPS certificates, settings and provisioning, were performed sequentially, both in the device side and in the middleware side.

Self-aware data privacy

The self-aware data privacy component integrates with Middleware by listening in on a RabbitMQ queue (receiving data from BOX2M's live IoT platform), and encrypt the data in the messages via hardened encryption, pushing them to another queue afterwards. Messages aren't consumed from the source queue, this ensuring that the data can be consumed by either, depending on whether hardened encryption is expected or not.

Reputation System and Policy Manager

The same description as provided for B2 applies.

Network-based Authorization

This component is ready to operate in this use case, similarly to what is described in B2.

Permissioned Blockchain

Supports the publishing of Reputation information.

5.2.4 Use Case B4 – GMS IoT device security or privacy incident

Overview

The work done to achieve the goals of use case B4 was related to the features allowing the Hardened Encryption System to be aware and act on security incidents occurred on IoT device.

These could be due to, for example, failed encryption at the device (due to lack of correlation with the information provisioned into Middleware). B4 is applicable for both ways of communication involving IoT devices (sending sensors data, getting commands).

The following specific incidents can take place at stage 1 (authentication):

- the Device ID, credentials or key for the correspondent device are not recognised (e.g., for failed authentication attempts or wrong Device ID) by the Middleware or the Device Behaviour Monitoring. In such case, the IoT device will not be allowed to authenticate.

During stage 2 (traffic), the following incidents may be considered:

- a key for the correspondent device is not recognised by Middleware. In this case, the IoT Device will not be allowed to perform any type of traffic, independently of the type of carried messages (Attestation Evidence / Device Attestation Certificate, or regular sensors data).

- a command sent by the IoT platform and encrypted by the Middleware with a proper key for that stage and that device, but not executed by the IoT device.

In any of the incidents, the security systems involved in the Security Chain of Trust (e.g. Reputation System) are notified.

This work involved the following activities:

Firmware Development

- Firmware tests by wrongly provisioning credentials, Device IDs, keys

Backend Development

- Backend tests by wrongly provisioning credentials, Device IDs, keys

Frontend Development

- Frontend by wrongly provisioning credentials, Device IDs, keys

IoT devices simulator – designed and built for fasting the integration with other security systems

- Provisioned and tested

Component integration

DIDs

In the case that the private key was compromised a new key pair can be generated and update the public key in the DID:WEB.

Hardened Encryption using crypto chip system

The registration and traffic performed by multiple industrial IoT devices were successfully tested. Additional tests were done with BOX2M live IoT platform, connected via TLS to middleware, in a real deployment scenario.

A set of scenarios simulating malicious attacks during device traffic process (sending sensors data path) were executed to test the behaviour of the encryption & decryption system. Multiple specific cases of wrong or missing keys, HTTPS certificates, settings and provisioning, were performed sequentially, both in the device side and in the middleware side.

Device Behaviour Monitoring

The Device Behaviour Monitoring, operating remotely, process relevant reputation updates, authentication events, and IoT devices' events from the middleware. Whenever it detects an anomaly in any of these, it notifies the Reputation System and Cyber-Threat Intelligence components via message bus.

Reputation System and Policy Manager

The reputation system receives information regarding the device security or privacy incident which is used to update the reputation score of the device. The updates are done in the *reputation_updates* exchange, as well as in the permissioned blockchain.

Reputation system is integrated with Middleware by RabbitMQ as main communication path, and consumes security events messages produced by Middleware, based on IoT device behaviour. API channel could be a backup.

BOX2M designed a specific policy for its devices lifecycle abnormal situations, which is in scope of upgraded / downgraded reputation. Policy covers all stages (registration, authentication, traffic). In this use case, there are relevant the criteria about potential wrong keys used by a device. Integration was tested and validated with Reputation System.

The information that is received from the middleware is exemplified below.

Received JSON from middleware	**Result of processing**

{“message”:
{"ClientId":"l42nebnokl",
“Reason”: “Invalid decryption key”,
,"TimestampUTC":"2023-03-14T23:05:10.5014637Z",
“Reason”: “NotAuthorized”},
"ArcadianId":"api.box2m.io:b666ca65-0faa-4e8b-ane}
	{
	“AIoTID”: “api.box2m.io:b666ca65-0faa-4e8b-ane”,
	“currentScore”: 0.50,
	“previousScore”: 0.55,
	“reputation_model”: “Alpha-Beta”,
	“alpha”: 0.55,
	“beta”: 0.1,
	“action”: 3
	}
	Note: in this case the reputation decreases since the device does not have the correct key for decryption.

Network-based Authorization

This component is ready to operate in this use case, similarly to what is described in B2.

Remote Attestation

In case of the need to re-authenticate the IoT device, the Remote Attestation's Verifier is able to receive the Attestation Cue from the Reputation System. The Verifier uses the middleware's API to collect the evidence (Device Attestation Certificate) and appraise it; in this case, at least one of the claims (‘catalogue id’, 'device id', 'equipping class', 'firmware version', 'vendor name')) will not correspond to the reference values, which will be reflected in the Attestation Result. The Attestation Result comprises the proportion of device claims adhering to the Verifier’s appraisal policies and the trustworthiness of the device whose claims are being appraised. In case of an invalid response from the middleware, both fields are set to –1, indicating an exception. In any case, the Attestation result is sent to the Reputation System.

Permissioned Blockchain

Supports the publishing of Reputation information.

5.2.5 Use Case B5 – GMS middleware security or privacy incident

Overview

The work done to achieve the goals of use case B5 was related to the features allowing the Hardened Encryption System to be aware and act on security incidents that have occurred on Middleware software platform. Incidents could be due to failed encryption run by the Middleware (due to lack of correlation with information provisioned into the IoT device or the IoT platform). B5 is applicable for both ways of communication involving the Middleware (sending sensors data, getting commands), and to both encryption technologies used by Middleware (TLS certificates and hardware encryption keys).

The following incidents taking place during stage 2 (traffic) are in scope of this use case:

- a command sent by IoT platform, TLS-encrypted and sent by the dedicated API to the Middleware, but not decrypted due to the lack of proper TLS certificate, will not be relayed forward for hardware encryption, and consequently will not be sent to the target IoT device.
- a command sent by IoT platform and encrypted by Middleware with an unrecognized key or not provisioned into Device crypto chip repository will not be decrypted and consequently not executed.

In both cases, the correspondent integrated security system involved in the Security Chain of Trust should be notified.

This work involved the following activities:

<u>Firmware Development</u>

- Firmware tests by wrongly provisioning credentials, Device IDs, keys

<u>Backend Development</u>

- Backend tests, by wrongly provisioning credentials, Device IDs, keys

<u>Frontend Development</u>

- Frontend tests, by wrongly provisioning credentials, Device IDs, keys

<u>IoT devices simulator – designed and built for fasting the integration with other security systems</u>

- Provisioned and tested

Component integration

Hardened Encryption using crypto chip system

The registration and traffic performed by multiple industrial IoT devices was successfully tested. Additional tests were done with BOX2M live IoT platform, connected via TLS to middleware, in a real deployment scenario.

A set of scenarios simulating malicious attacks during device traffic (getting commands path) process were executed to test the behaviour of the encryption & decryption system. Multiple specific cases of wrong or missing keys, HTTPS certificates, settings and provisioning were performed sequentially, both in the device side and in the middleware side.

Reputation System and Policy Manager

Same description as for B4.

In this use case, the relevant criteria include potential wrong keys or wrong HTTPS certificates used by Middleware. Integration was tested and validated with Reputation System.

Permissioned Blockchain

Supports the publishing of Reputation information.

5.2.6 Use Case B6 – External data audit to grid infrastructure

Overview

The work done to achieve the goals of use case B6 was related to the features allowing an authorised operator (belonging to a state authority or an empowered entity) to recover encrypted data from a recovered device. IoT device was supposed to get attacked, disappeared from a while or get exposed under uncertain conditions.

As its unique ID, credentials and crypto keys may be compromised, and sensors data buffered during attack should be recovered, even if it is encrypted and stored locally on device EPROM, the vendor must facilitate a way for such data recovery.

The authorised operator will connect locally to device, by using a specialised hardware gateway and local software application, will login into device firmware CLI, with credentials provided by vendor, will copy locally on its computer drive the encrypted stored file.

The authorised operator will use a CLI type software application designed and deployed by vendor for such situation, will apply an encryption key, unique for that stage (recovery) and Device ID, and will decrypt the file.

Once operation is done, all credentials, IDs and keys belonging to device will be erased and never used again, in both device master firmware, device crypto chip repository and Middleware.

<u>Hardware Development</u>

- Testing of motherboard EPROM and of interfacing kit / gateway for local connection, for hardware functionalities validation (power supplies, short circuits, power budget, signals & interfaces functionalities, sudden operations events)
- It was designed and built a customised laboratory set-up, for running all current and further board exploitation tests, including a mobile interfacing kit
- This scenario cannot be run by IoT simulator, and it is very sensible for electronic

components (could damage whole mother board); antenna disconnection may generate small (micro-Amps) short circuits, which could damage some sensitive electronics (as crypto chip)

- Communication interruption was simulated from SIM orchestrator of a MVNO

Firmware Development

- Designing and deployment of a CLI software application for device master firmware
- Testing of firmware, holistically, by executing all CLI defined commands
- Testing of firmware for local data buffering, in case of communication interruption

Backend Development

- Middleware backend testing (encryption and decryption keys for data buffering event)

Recovery software application

- a functional CLI tool was prototyped

Component integration

Hardened Encryption using crypto chip system

The registration and traffic performed by multiple industrial IoT devices were successfully tested. Additional tests were done with BOX2M live IoT platform, connected via TLS to middleware, in a real deployment scenario.

A set of scenarios simulating malicious attacks during device data recovery process were executed to test the behaviour of the encryption & decryption system. Multiple specific cases of wrong or missing keys, credentials settings and provisioning, were performed sequentially, both in the device side and in the middleware side.

Reputation System and Policy Manager

Same description as for B4.

5.3 Domain C - Medical IoT

Regarding domain C use-cases, several tasks were focused on concluding all specifications regarding interactions between domain C and all the components of the ARCADIAN-IoT framework with which domain C directly interacts. To achieve this, several actions were made to align between domain leader (RGB), and all involved technical partners (IPN, ATOS, MAR, UC, XLAB, UWS, 1GLOBAL). Use cases C2, C3 and C4 were addressed within the scope of P1. Integration activities related to use cases C1, C5, C6 and C7 were addressed within during P2. The targeted ARCADIAN-IoT involvement in the domain is depicted via the Figure 7 below.

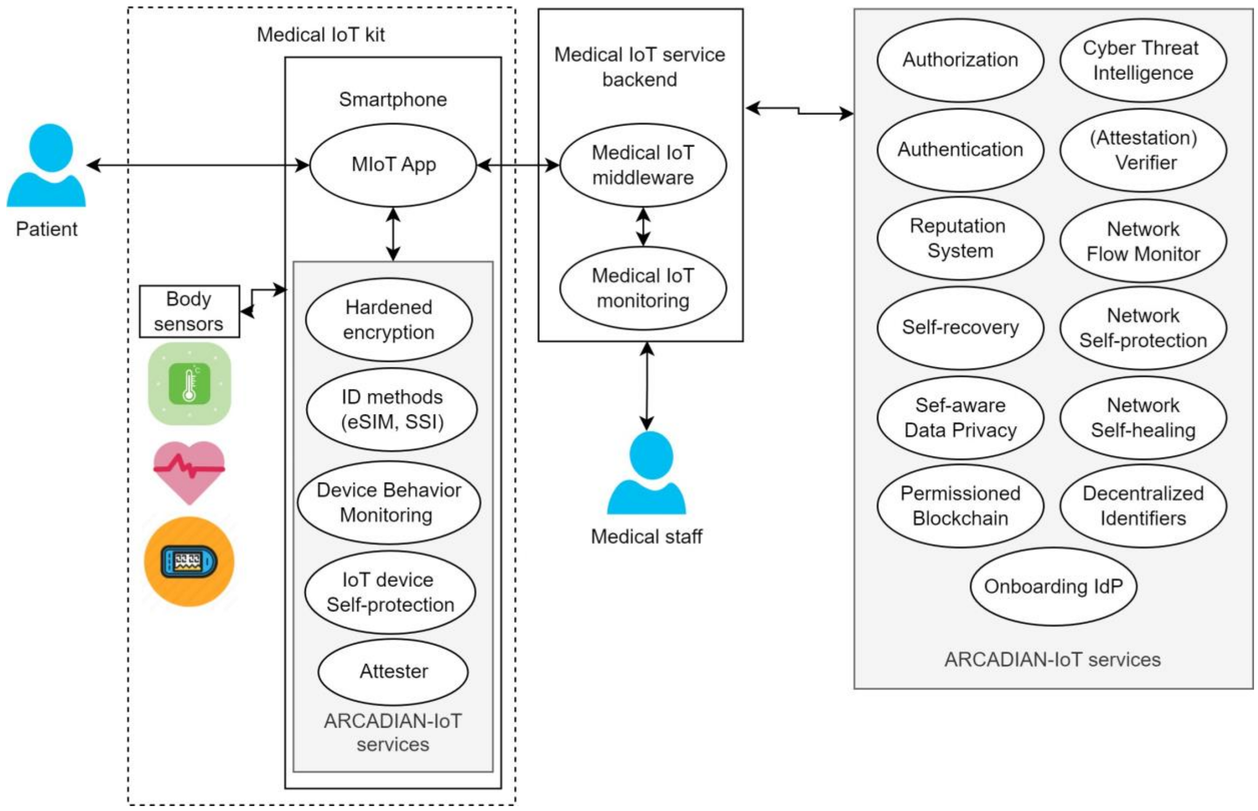


Figure 7 - High-level view of ARCADIAN-IoT involvement in Domain C

5.3.1 Use Case C1 – MIoT kit delivery - Patient registration and authentication

Overview

To achieve the goals of use case C1, we have successfully integrated the Device Behaviour Monitor and Device Self-Protection to continuously assess and respond to security threats. A Remote Attestation mechanism, using RabbitMQ for secure communication, ensures device integrity and aids in reputation scoring. Key integrations include patient authentication with ATOS, and a Hardened encryption module developed with XLAB and 1Global for secure medical information. The system is designed to be dynamic, responsive, and secure, with ongoing enhancements like the planned modification to Martel's proxy for optimal performance.

Component integration

DIDs

- Peer DIDs are supported and working in the end user mobile SSI Wallet.

Public DID:WEB is working for the ARCADIAN-IoT Decentralized Identity integrated with the Ledger uSelf SSI Broker / Agent for issuing and verifying Verifiable Credentials for persons.

Verifiable Credentials and Onboarding IdP

Registration request from the end user app results in MFA requesting presentation request to the SSI wallet and subsequent verification of Person VC from the end user´s SSI Wallet is supported by the frameworks SSI Broker/Agent to verify the Person VC. The Onboarding IdP will subsequently generate an ARCADIAN-IoT ID (aiotID) and provision this to the Network

Authenticator and finally publish it over RabbitMQ for all concerned framework components such as Reputation to informed of a new registered entity.

Authentication request from the end user app results in MFA requesting presentation request to the SSI wallet and subsequent verification of Person VC and from the end user´s SSI Wallet is supported by the frameworks SSI Broker/Agent to verify the Person VC.

Device Behaviour Monitor

The Device Behaviour Monitor operates in the background of the smartphone, continuously scanning for anomalies related to authentication and reputation events sent by the Reputation and Authentication components via RabbitMQ exchange. Its involvement in the presence of anomalies associated to the smartphone is described in use case C5.

Remote Attestation

As a necessary requirement for the use case, the Verifier receives Reference Values and corresponding appraisal policies for android devices from the Domain Owner via a RabbitMQ exchange. Upon completion of device registration, the Verifier receives an Attestation Cue from the Reputation System, which triggers the attestation procedure. The previously received Reference Values and its policies are used to appraise the claims sent by the device, whose results are sent to the Reputation System to help determine the device's initial reputation score.

Reputation System and Policy Manager

The reputation system receives information regarding the device that was registered and initializes the reputation score.

The information that is used, was exemplified in the A1 use case.

Network-based Authentication – SIM-based

In what regards the new person/user registration flow, a network-based ID token issued by a subcomponent living at the network core to the personal device (smartphone), will be used to provide unique information to be associated with the new ARCADIAN-IoT ID. This association will be stored in a subcomponent that will be responsible for the verification of the network-based credentials in the authentication flow. The unique information mentioned includes an encrypted network identifier (e.g. encrypted IMSI) of the of the network subscriber. This component will also trigger the registration of the new ARCADIAN-IoT entity in the Network-based Authorization component, by providing to it the encrypted identifier and the newly created ARCADIAN-IoT ID.

In the authentication flow, the subcomponent in the core network issues a Network ID token to a given subscriber with a given ARCADIAN-IoT ID, and those credentials match is verified by the second subcomponent, integrated with ARCADIAN-IoT MFA.

This component was successfully integrated and tested in the context of this use case.

Network-based Authorization

In what regards the registration flow, this component will receive the encrypted network identifier and the newly created ARCADIAN-IoT ID from the Network-based Authentication component previously described. Living at the network core, this component is the only one with the keys to decrypt the network identifier (IMSI) and, at registration, associate it with the newly created ARCADIAN-IoT ID. This match will be critical for this component to enforce policies in the network (policies for a given network subscriber with a given IMSI) accordingly to the received trust-based actions to enforce provided by the Reputation System (which identifies entities with the ARCADIAN-IoT ID).

After the registration, this component is ready to operate. Particularly it enforces trust-based communication policies and distributes trust information to the devices secure element (SIM/eSIM), according to information received from the Reputation System. Despite the targeted device being a smartphone, as its only purpose is serving as proxy to the medical data coming from the body sensors, the cellular communication enforcement polices can apply (in opposition to what is expected to happen with the smartphones in Domain A, where several apps are expected to co-exist). The trust information distribution to the secure element will trigger SIM-based self-protection and self-recovery within the context of the app (e.g. refuse to provide digital signatures when the device is found to be compromised).

This component was successfully integrated and tested in the context of this use case.

Multi-factor Authentication (MFA)

This component was successfully integrated and tested in the context of this use case – integration with the solution provider and with the 2 identification components (Network-based Authentication and Verifiable Credentials).

Permissioned Blockchain

Supports the Organization Trust Registry, publishing of Reputation information and HE public keys.

5.3.2 Use case C2 - MIoT Capturing and sending vital signs and perceived health status

Overview

The main effort has been on finalizing a robust HW/SW system for the telemedicine modules, Android app to collect the patient health data and to send it to the Medical IoT web service. Additionally, effort has been put in identifying and completing necessary adaptations for integrating the Medical IoT service with ARCADIAN-IoT.

Component Integration

Remote Attestation

Remote Attestation procedures for attesting the smartphone (Android OS) are supported. This involves watchdog-based remote attestations conducted periodically. The evidence collected from the device is successfully appraised against the associated policies previously provided by the service provider to the Verifier.

Hardened Encryption (via eSIM)

The encryption library based on the attribute-based encryption paradigm is available to be used in Go, Java and Python. A key management service is currently being prototyped.

The SIM ecosystem for acting as RoT in the HE process was fully integrated within this use case. It provides the expected digital signatures of the hash of the encrypted payloads, and is ready to receive trust-based information from the Network-based Authorization component. The SIM also informs the device operating system of the apps allowed to interact with it, identifying these apps within an authorization mechanism embedded in it.

Network-based Authorization

This component is ready to operate in this use case, particularly to enforce trust-based communication policies in the cellular network and to distribute trust information to the devices' secure element (SIM/eSIM). This operation is according to information received from the Reputation System. As in every flow after the registration, the trust information distribution to the secure element will trigger SIM-based self-protection and self-recovery (e.g. refuse to provide digital signatures when the device is found to be compromised).

Reputation System and Policy Manager

The reputation system receives information regarding the sending of vital signs and updates the reputation score, previously initialized in use case C1.

The information that is received from the medical IoT is exemplified below.

Received JSON from medical IoT Service	Result of processing

{
"AIoTID": "atos.net:4701f4e6-79f2-4330-9a98-25497255bccf",
"eventDescription": "Personal data processing towards health alarm triggering",
"result": "started",
"timestamp": 1702471567,
"severity": ""
}
 	
Note: The **eventDescription** can have multiple values:
• "Personal data processing towards health alarm triggering"
• "Monitoring a patient and updates to the patient monitoring protocol"
The result can include the following values: "started", "success", "error"
	{
	"AIoTID": "atos.net:4701f4e6-79f2-4330-9a98-25497255bccf",
	"currentScore": 0.55,
	"previousScore": 0.50,
	"reputation_model": "Alpha-Beta",
	"alpha": 0.50,
	"beta": 0.0,
	"action": 3
	}

Self-aware Data Privacy

The component can receive data from a device monitoring medical information, decrypting the data in its entirety, and encrypting the data again following the data transformation and encryption policies provided by the other components i.e. Hardened Encryption. This includes retrieving the device's ID from a security token, retrieving encryption keys and policies from a specified service, and interacting with the reputation system via sending messages to a RabbitMQ queue. In addition, the component can act as a consumer for a specified queue, consuming messages and encrypting them, pushing them to a different queue if need be. The component is also capable of performing anonymisation on the data instead of encryption, if policies specify this to be the preferred approach to a given set of data attributes.

Device Behaviour Monitor

The Device Behaviour Monitor operates in the background of the smartphone, continuously scanning for anomalies related to authentication and reputation events sent by the Reputation and Authentication components via RabbitMQ exchange. Its involvement in the presence of anomalies associated to the smartphone is described in use case C5.

Permissioned Blockchain

Supports the publishing of Reputation information.

5.3.3 Use case C3 - Personal data processing towards health alarm triggering

Overview

For supporting this use case, an Android application has been integrated with the telemonitoring web service and the ARCADIAN-IoT framework. Such integration aims at enabling that the information collected by the telemedicine modules – as well as personal data - can be safely accessed as soon as possible by the doctor and nurses, for a prompt diagnosis.

Towards this goal, secure data transmission, by sending encrypted events and medical data (with ARCADIAN-IoT's Hardened Encryption) to Medical IoT service, has been implemented.

Component Integration

Self-aware Data Privacy

The component can receive data from a device monitoring medical information, decrypting the data in its entirety, and encrypting the data again following the data transformation and encryption policies provided by the other components i.e. Hardened Encryption. This includes retrieving the device's ID from a security token, retrieving encryption keys and policies from a specified service, and interacting with the reputation system via sending messages to a RabbitMQ queue. In addition, the component can act as a consumer for a specified queue, consuming messages and encrypting them, pushing them to a different queue if need be. The component is also capable of performing anonymisation on the data instead of encryption, if policies specify this to be the preferred approach to a given set of data attributes.

Hardened Encryption (via eSIM)

Similarly, as in Use case C2, HE component currently enables encryption and key management based on attribute-based encryption. In what regards the use of SIM as RoT, it is operational in this use case in a similar way as C2.

Device Behaviour Monitor

The Device Behaviour Monitor operates in the background of the smartphone, continuously scanning for anomalies related to authentication and reputation events sent by the Reputation and Authentication components via RabbitMQ exchange. Its involvement in the presence of anomalies associated to the smartphone is described in use case C5.

Reputation System and Policy Manager

The reputation system receives information from the self-aware data privacy component and updates the reputation score. Using the same information as in C2 from the medical IoT service.

Permissioned Blockchain

Supports the publishing of Reputation information.

5.3.4 Use case C4 - Monitor a patient and update a patient monitoring protocol

Overview

The focus has been on improving the communication protocol of the modules to avoid failures, as well as improving the integration of the mobile phone with the telemedicine module, aimed at more efficient and reliable collection of patient medical data.

The protection of the information through ARCADIAN-IoT's encryption and authentication components during the process of sending information to the web platform is completed. Medical professionals are authenticated within the system by using their SSI wallets to store and present verifiable credentials. The SSI Broker/Agent, along with other framework components, ensures the security and integrity of the authentication process, while the use of messaging systems like RabbitMQ allows for communication and notification of new registrations across the framework. This authentication system leverages the principles of SSI, where users have control over their identity credentials and can securely prove their professional status without relying on a central authority.

Component Integration

DIDs

- Peer DIDs are supported and working in the end user mobile SSI Wallet.

Public DID:WEB is working for the ARCADIAN-IoT Decentralized Identity integrated with the Ledger uSelf SSI Broker / Agent for issuing and verifying Organization Member Verifiable Credentials for medical professionals.

Verifiable Credentials and Onboarding IdP

Registration request from the medical professional app results in MFA requesting presentation request to the SSI wallet and subsequent verification of Organization Member VC from the medical professional ´s SSI Wallet is supported by the frameworks SSI Broker/Agent. The Onboarding IdP will subsequently generate an ARCADIAN-IoT ID (aiotID) and provision this to the Network Authenticator and finally publish it over RabbitMQ for all concerned framework components such as Reputation to informed of a new registered entity.

Authentication request from the end user app results in MFA requesting presentation request to the SSI wallet and subsequent verification of Organization Member VC and from the end user´s SSI Wallet is supported by the frameworks SSI Broker/Agent to verify the Person VC.

Hardened Encryption (via eSIM)

Please refer to use case C2.

Device Behaviour Monitor

The Device Behaviour Monitor operates in the background of the smartphone, continuously scanning for anomalies related to authentication and reputation events sent by the Reputation and Authentication components via RabbitMQ exchange. Its involvement in the presence of anomalies associated to the smartphone is described in use case C5.

Network-based Authorization

Same as C2.

Self-aware Data Privacy

The component can receive data from a device monitoring medical information, decrypting the data in its entirety, and encrypting the data again following the data transformation and encryption policies provided by the other components i.e. Hardened Encryption. This includes retrieving the device's ID from a security token, retrieving encryption keys and policies from a specified service, and interacting with the reputation system via sending messages to a RabbitMQ queue. In addition, the component can act as a consumer for a specified queue, consuming messages and encrypting them, pushing them to a different queue if need be. The component is also capable of performing anonymisation on the data instead of encryption, if policies specify this to be the preferred approach to a given set of data attributes.

Reputation System and Policy Manager

The reputation system receives information regarding the monitoring of the patient and updates the reputation score, accordingly. Refer to use case C2 for further information.

Permissioned Blockchain

Supports Reputation Scores and Hardened Encryption encryption Keys.

5.3.5 Use Case C5 – Patient MIoT devices security or privacy incident

Overview

In C5, when under normal usage of the Patient MIoT device, it is subject to a security or privacy incident, there are several possible causes, such as:

- Privilege escalation
- Device loss or theft
- Distributed Denial of Service Attacks
- NMAP scans
- Attempted Encryption of Large Number of Files (Ransomware)
- Attempted tampering with security token attached to HTTP requests

Backend Development

The work done to achieve the goals of use case C5 includes integration with Device Behaviour Monitor and Self-Protection to monitor and mitigate threats, and a Remote Attestation mechanism that uses RabbitMQ for securing communications and maintaining device integrity. It integrates patient and doctor credential recovery through ATOS, works with 1Global for reputation management, and employs a robust encryption module co-developed by XLAB and 1Global for protected medical communications. The system is built to be adaptable, proactive, and secure, with continuous improvements such as latest Martel proxy update to improve performance. UWS has created a 5G security system with three parts: monitoring for threats, analysing and planning a response, and then implementing protection measures, all communicated through RabbitMQ, ensuring quick and coordinated defence against cyber threats like DDoS attacks.

Component integration

Device Behaviour Monitor

When an anomaly is detected, the Device Behaviour Monitor alerts the Reputation System and Cyber-Threat Intelligence components via the message bus about the detected anomaly. In parallel, it also sends this anomaly data to the Device Self-Protection component to determine the appropriate policies.

Cyber Threat Intelligence

Upon receiving threat-related information from Device Behaviour Monitor, The CTI component parses and processes it to generate MIPS-based Indicator of Compromises (IoCs). The processed IoCs are then further analysed by the ML-based subcomponents by adding complementary details (e.g., threat level). Once a complete IoC is generated, it is locally stored to update IoC database, and also forwarded to output interface.

Device Self-Protection

Upon receiving threat-related information from the Device Behaviour Monitor, Reputation System, or Cyber-Threat Intelligence, the Device Self-Protection component determines the most suitable policies for application. Since these policies cannot be directly applied to Android devices, the component forwards the recommended policies to Domain Owners, who will then decide on the necessary actions. Additionally, these suggestions are sent to the Self-Recovery component, to evaluate them, decide the actions to be taken (apply or not), and report the decisions back to the Device Self-protection. Finally, the Reputation System receives the confirmation from Device Self-protection of whether or not the policies were applied by itself (which is negative in this use case) and self-recovery.

Remote Attestation

Remote Attestation attests the smartphone and its data both periodically and on-demand, as requested by the Reputation System. The Attestation Result comprises two fields, with values between 0 and 1, where the first field reflects the proportion of claims adhering to the Verifier’s appraisal policies. If a response is received with a nonce differing from the one sent in the Attestation Request, or if there are issues with the cryptographic procedures on the Verifier’s side, both fields are set to 0. In case of an invalid response from the Attester, both fields are set to –1. In all scenarios, the Attestation Result is sent to the Reputation System for further actions.

Self-aware Data Privacy

When data sent from a device pass through the Envoy proxy, the signature of the token that is sent alongside the data is verified, to ensure the token is valid. Requests with an invalid token are dropped and logged.

Reputation System and Policy Manager

The reputation system receives information regarding the possible incident and updates the reputation score, accordingly.

Network-based Authorization

This component is ready to operate in this use case, particularly to enforce trust-based communication enforcement policies in the cellular network and to distribute trust information to the devices’ secure element (SIM/eSIM). This operation is according to information received from the Reputation System. The communication enforcement based on trust information can automatically stop network related incidents like private information leakage or DDoS. As in other previously described use cases, the trust information distribution to the secure element will trigger SIM-based self-protection and self-recovery (e.g. refuse to provide digital signatures when the device is found to be compromised).

Network Flow Monitoring

The Network Flow Monitoring (NFM) component consists of a set of subcomponents that monitor and analyse the 5G network traffic. This component detects when a malicious 5G flow is passing through the network infrastructure. Hence, raises an alert with the specific information of the detected malicious flow to its RabbitMQ exchange. Therefore, the rest of the framework components subscribed to these alerts (i.e. Network Self-Healing and Cyber Threat Intelligence) will be able to consume them and perform their analysis of the detected threat. . This component is ready to operate in this use case, similarly to C6.

Network Self-Healing

The Network Self-Healing (NSH) component works for the analysis of the detected threats of the NFM and oversees creating a set of healing instructions by the prescriptive analysis which provides insights about WHAT, WHERE, WHEN and for HOW LONG the attack should be stopped. These healing instructions are submitted to the particular RabbitMQ exchange for this component, where the following actor of the self-protection loop will consume (i.e. Network Self Protection). This component is ready to operate in this use case, similarly to C6.

Network Self-Protection

The Network Self-Protection component provides the network topology with mitigation rules and actuators that enforce the multiple healing instructions orchestrated by the NSH. Hence protecting the network data plane topology. These healing instructions are consumed by the NSP by the RabbitMQ queue where it is subscribed and where the NSH is publishing them. Right after the rules are enforced on the data plane, the NSP publishes a confirmation message to its particular message bus exchange where the rest of components subscribed to it will consume (i.e. Network Flow Monitoring and Reputation System). This component is ready to operate in this use case, similarly to C6.

Verifiable Credentials & Credential Recovery

The SSI Wallet integrates with a dedicated Credential Recovery back-up server to store encrypted wallet credentials and recover them when needed. Credential Recovery supports a dedicated backup server for storing user´s backups based on their user identity.

Hardened Encryption

Please refer to use case C2.

5.3.6 Use Case C6 – MIoT Cloud services security or privacy incident

Overview

In C6, when under normal operation of the MIoT Cloud services they are subject to a security or privacy incident, there are several possible causes, such as:

- Privilege escalation.
- Distributed Denial of Service Attacks.

<u>Backend Development</u>

Use Case C6 involves addressing a security incident related to MIoT cloud services, specifically in the context of the medical application domain and network incidences. The focus of the work done in C6 is on mitigating security threats and ensuring the protection of cloud infrastructure for medical applications. This includes measures to prevent and respond to attacks on MIoT cloud services, ultimately aiming to uphold the security and integrity of the medical data and devices involved. This specific use case focuses on the mitigation of a DDoS attack, orchestrated by the combination of multiple infected MIoT devices and others non-MIoT. The target of the attack is the MIoT cloud service, and the malicious network flows are happening through the 5G network infrastructure before reaching their target.

Component integration

Network-based Authorization

This component is ready to operate in this use case, particularly to enforce trust-based communication enforcement policies in the cellular network and to distribute trust information to the devices' secure element (SIM/eSIM). This operation is according to information received from the Reputation System. In this particular use case, this component action may be relevant if the incident is triggered by compromised medical IoT devices.

Network Flow Monitoring

This component continuously monitors the specified 5G network and whenever a new attack is detected, it raises a new Network IDS Event to the message bus exchange with information gathered from the malicious network flow. This information is recovered by multiple components in the ARCADIAN-IoT framework (i.e. Network Self-Healing, Reputation System and Cyber Threat Intelligence). However, for this use case validation, the Network Self-Healing is the consequent actor in the protection loop.

Cyber Threat Intelligence

Upon receiving threat-related information from Network Flow Monitoring, The CTI component parses and processes it to generate MIPS-based Indicator of Compromises (IoCs). The processed IoCs are then further analysed by the ML-based subcomponents by adding complementary details (e.g., threat level). Once a complete IoC is generated, it is locally stored to update IoC database, and also forwarded to output interface.

Network Self-Healing

This component receives the Network IDS Event published by the NFM in the previous steps of the self-protection loop. After receiving the message, a set of prescriptive analytics is performed to analyse and create a new set of instructions to mitigate the incoming threat. This set of instructions is published to the Healing Instructions message bus exchange, where the following actor of the self-protection loop (the Network Self-Protection) takes place.

Network Self-Protection

This component, deployed alongside the network topology, receives the set of instructions created by the analysis of the NSH and performs the enforcement of the mitigation rules on the data plane. Hence, a new protection rule is applied in the closest topology interface to the source of the attack. This rule drops all the malicious specific 5G network flows in the particular interface with the provided technology in the healing instructions. Afterwards, the Network Self-Protection component raises a new confirmation message to the message bus exchange.

5.3.7 Use Case C7 – Medical 3rd party security or privacy incident

Overview

In C7, when under normal operation 3rd party medical services, they are subject to a security or privacy incident, there are several possible causes, such as:

- Privilege escalation.
- Distributed Denial of Service Attacks.
- Attempted tampering with security token attached to HTTP requests.

Backend Development

Use Case C7 involves addressing a security incident related to third-party medical services in the context of the demonstration for Domain C (medical application). The work done in C7 focuses on mitigating security threats and ensuring the protection of third-party services that are integral to the medical application domain. This may include measures to prevent and respond to security incidents involving services such as the web browser used by the doctor/nurse while viewing patient medical data over 5G connections.

Component integration

Network-based Authorization

This component is ready to operate in this use case, similarly to C6.

Network Flow Monitoring

The Network Flow Monitoring (NFM) component consists of a set of subcomponents that monitor and analyse the 5G network traffic in search of any possible known DDoS attack. This component is ready to operate in this use case, similarly to C6.

Cyber Threat Intelligence

Upon receiving threat-related information from Network Flow Monitoring, The CTI component parses and processes it to generate MIPS-based Indicator of Compromises (IoCs). The processed IoCs are then further analysed by the ML-based subcomponents by adding complementary details (e.g., threat level). Once a complete IoC is generated, it is locally stored to update IoC database, and also forwarded to output interface.

Network Self-Healing

The Network Self-Healing (NSH) component works for the analysis of the detected threats of the NFM and oversees creating a set of healing instructions by the prescriptive analysis which provides insights about WHAT, WHERE, WHEN and for HOW LONG the attack should be stopped. This component is ready to operate in this use case, similarly to C6.

Network Self-Protection

The Network Self-Protection component provides the network topology with mitigation rules and actuators that enforce the multiple healing instructions orchestrated by the (NSH). Hence protecting the network data plane topology. This component is ready to operate in this use case, similarly to C6.

5.4 Validation considerations

This document provided information regarding the integration activities performed for building P2 from both ARCADIAN-IoT framework perspective (i.e., availability of ARCADIAN-IoT for addressing the requirements of each targeted use case) and IoT application / domain perspective (i.e., adaptations required from the IoT service side to leverage ARCADIAN-IoT functionalities). However, the complete and end-to-end validation of ARCADIAN-IoT framework in the three domains is outside the scope of this report. This will be addressed by D5.5 – "ARCADIAN-IoT use cases validation and legal compliance – Final Version", to be delivered by M36.

From the time of submission of this report (D5.2) until M36, the consortium will execute the remaining validation scenarios, bind the appropriate metrics and KPIs to each scenario, produce the complete sequence diagrams to document the interactions between components, use cases and supporting tools. With respect to the legal validation, the consortium – guided by its legal partner (Elex) – will assess to what extent ARCADIAN-IoT framework and its use cases comply with existing cybersecurity regulations.

6 CONCLUSIONS

This document details the main integration outcomes of the final Prototype of ARCADIAN-IoT framework, which considered a first release in M24 (P1) and a final release in M33 (P2). These outcomes are part of Task 5.1 (Integration of ARCADIAN-IoT framework) and are associated to Milestone 3 (MS3) and Milestone 4 (MS4), due on M20 and M33 of the project. The outcomes presented in this document build upon the research and implementation of each ARCADIAN-IoT component (addressed in WP3 and WP4) as well as the preparation and implementation of ARCADIAN-IoT use cases (Task 5.2, 5.3 and 5.4) – hence documenting the overall component to component integration and support levels provided by the use cases to enable component integration.

All technical partners were involved in an ongoing iterative integration process of the ARCADIAN-IoT framework. At the same time, domain owners have followed the integration of technical components, providing insight on the readiness levels of the use cases and contributing to the specifications and readiness of the final Prototype.

In this document, the final architecture, the adopted integration approach and execution plan were presented in the initial sections. The deliverable partitioned the contents in two main views: (1) the component-to-component integration, where an inter-component integration map was provided, and the (2) component-to-domain support, where an integration map reflecting the ARCADIAN-IoT support for the identified use cases was shown.

Overall, from a total of 22 technical components of the ARCADIAN-IoT framework, all has been integrated successfully both for the Component x Component scope and for the complete Use Case x Component scope, enabling 20 use cases with ARCADIAN-IoT security-related functionalities (detailed in Figure 4). The report on the use cases validation and legal compliance focusing on both P1 and P2 functionalities will be delivered in M36.

As final take away, the integration work performed during the integration task (5.1) proved that a distributed and well partitioned security architecture (with its components in different security planes providing different security controls and establishment of a clear Chain of Trust as outlined in section 2) can be added successfully to different Technology Domains, enabling the holistic management of Trust, Identity, Privacy, Security and Recovery aspects.

With a highly extensible architecture enabling the design and implementation of novel or distinct functionalities in the different architectural planes, additional components (beyond the actual project's scope) can be added to the framework (e.g., privacy risk assessment, AI robustness assessment components). The framework's interfaces can equally be evolved to cope with additional IoT application domains - and corresponding needs - that could be supported by the ARCADIAN-IoT ecosystem with low impact on the current ARCADIAN-IoT architecture (e.g., law enforcement). Thus, making it easier for organizations to enhance their security stance and perceived trust in their participation in the IoT ecosystem.

REFERENCES

[1] ARCADIAN-IoT, “D2.2: Use case specification,” 2021.

[2] ARCADIAN-IoT, “D2.5: ARCADIAN-IoT architecture,” 2022.

[3] ARCADIAN-IoT, “D3.3: ARCADIAN-IoT Horizontal Planes - final version,” 2024.

[4] ARCADIAN-IoT, “D4.3: ARCADIAN-IoT Vertical Planes - final version,” 2024.

[5] ARCADIAN-IoT, “D2.4: ARCADIAN-IoT framework requirements,” 2021.