Grant Agreement N°: 101020259

Topic: SU-DS02-2020

ARCADIAN-IoT

Autonomous Trust, Security and Privacy Management Framework for IoT

D5.1: Integration of ARCADIAN-IoT framework – 1st version

Revision: v.1.0

Work package	WP 5
Task	Task 5.1
Due date	31/12/2022
Submission date	23/01/2023
Deliverable lead	IPN
Version	1.0
Partner(s) / Author(s)	IPN: Paulo Silva (co-editor), Sérgio Figueiredo (co-editor), João Rainho, Vitalina Holubenko, Ruben Leal UWS: Jose M. Alcaraz Calero, Qi Wang, Antonio Matencio Escolar, Ignacio Sanchez Navarro, Pablo Benlloch Caballero, Ignacio Martinez Alpiste, Gelayol Golcarenarenji, Julio Diez-Tomillo TRU: João Casal, José Rosa, Tomás Silva, Ivo Vilas Boas, Carlos Morgado XLAB: Tilen Marc, Benjamin Benčina, Jan Antić UC: Bruno Sousa MARTEL: Giacomo Inches BOX2M: Alexandru Gliga RISE: Alfonso Iacovazzi, Han Wang ATOS: Ross Little, Miguel Angel Mateo Montero RGB: Ricardo Ruiz LOAD: Pedro Colarejo

Abstract

This report documents deliverable D5.1 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 **first prototype (P1)** of ARCADIAN-IoT framework, which demonstrates **preliminary ARCADIAN-IoT functionalities -** resulting from **partial 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	IPN
V0.2	Insertion of aspects to cover in each section	IPN
V0.3	Abstract, introduction and executive summary	IPN
V0.4	Integration approach	IPN
V0.5	Component integration status	All Technical Partners
V0.6	Use case support description	Doman Leaders and Technical Partners
V0.7	Conclusions	IPN
V0.8	Internal review	ATOS
V0.9	Address review comments and feedback	ALL
V1.0	Document cleaning and final quality check	IPN

Disclaimer

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, etc

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 status of ARCADIAN-IoT framework's pilot **P1 – Prototype 1**. P1 provides **preliminary ARCADIAN-IoT functionalities** and builds on currently available features provided by each component from the Horizontal and Vertical planes, their partial intra- and inter-plane integration, and the necessary tailoring and testing for enabling subsets of the use cases defined within the project.

The document presents the adopted integration approach and plan, split in two views (component-to-component and component-to-domain integration), and the current integration status, including performed tests and integration validation experiments.

The report on the use cases validation and legal compliance focusing on P1 functionalities will be delivered in M24, while the remaining ARCADIAN-IoT components' functionalities and domains' use cases artifacts will delivered in prototype 2 (P2), due in M30.

The material presented in this document is the main outcome of **Task 5.1 (Integration of ARCADIAN-IoT framework)** and builds both on the 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). All technical partners and domain owners were involved in the iterative process of integrating ARCADIAN-IoT framework.

EXECUTIVE SUMMARY **6**
LIST OF FIGURES **8**
LIST OF TABLES **9**
ABBREVIATIONS **10**
1 INTRODUCTION **12**
1.1 Objectives and assumptions 12
1.2 Background on P1 planning process 12
1.3 Document structure 13
2 INTEGRATION APPROACH **13**
2.1 Recap on ARCADIAN-IoT architecture 14
2.2 Component to component approach 17
2.3 Component to domain approach 22
2.4 Supporting communication tools 24
3 COMPONENT INTEGRATION STATUS **25**
3.1 Integration status for horizontal planes 26
3.2 Integration status for vertical planes 31
3.3 Overall integration status 36
4 DOMAIN INTEGRATION STATUS **37**
4.1 Domain A - Emergency and vigilance using drones and IoT 38
4.2 Domain B - Secured early monitoring of grid infrastructures 42
4.3 Domain C - Medical IoT 46
4.4 Validation considerations 49
5 CONCLUSIONS **50**
REFERENCES **51**

LIST OF FIGURES

Figure 1 - ARCADIAN-IoT functional architecture 14

Figure 2 - ARCADIAN-IoT deployment view 15

Figure 3 - Component to component integration mapping 21

Figure 4 - Use Case Integration Plan 24

Figure 5 - Overview of integration status with respect to the functional architecture 37

Figure 6 - High-level view of ARCADIAN-IoT involvement in Domain A 38

Figure 7 - High-level view of ARCADIAN-IoT involvement in Domain B 43

Figure 8 - High-level view of ARCADIAN-IoT involvement in Domain C 46

LIST OF TABLES

Table 1 – List of updates to ARCADIAN-IoT functional view 14

Table 2 - List of updates to ARCADIAN-IoT deployment view 15

Table 3 - Prototype functionality roadmap for components belonging to ARCADIAN-IoT horizontal planes 17

Table 4 - Prototype functionality roadmap for components belonging to ARCADIAN-IoT vertical planes 19

Table 5 - Use cases addressed in P1 23

Table 6 - Use cases planned for P2 23

Table 7 - Example of Biometrics authentication request and reply for facial recognition 32

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
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
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 Identification Module
TCP	Transmission Control Protocol
VDR	Verifiable Data Registry
W3C	World Wide Web Consortium

1 INTRODUCTION

This section starts by laying out the objectives and assumptions for this deliverable and related outcome: P1 – Prototype 1. It is followed by a presentation of key background information on the ARCADIAN-IoT framework developed for P1 (i.e., technical components, domains and use cases). Finally, this section concludes with a presentation of the overall document structure and organization.

1.1 Objectives and assumptions

The main purpose of this report is to document **P1**, which as a first pilot provides **preliminary functionalities ARCADIAN-IoT framework**, and is the sum of the following:

A. The available **functionalities from each component** taking part in the Horizontal and Vertical planes responsible for providing Privacy, Security, Identity, Trust and Recovery management capabilities (devised in WP3 and WP4);
B. ARCADIAN-IoT partial **intra- and inter-plane integration**;
C. ARCADIAN-IoT components' **tailoring and testing** for enabling subsets of the use cases defined within the project, both resulting from the first year of work in T5.1;
D. The available **applications, hardware and other artefacts** provided (produced from scratch or adapted) by each of the domains (emergency and vigilance, smart grids, medical IoT), particularly those directly interacting or integrating with ARCADIAN-IoT framework - resulting from the first year of activities within T5.2, T5.3 and T5.4.

As will be shown, the first prototype of ARCADIAN-IoT framework (P1) considers multi-fold integration actions involving the different components. As expected, and either driven from their research scope or initial maturity, not all components display the same maturity level, either from the point of view of functionality availability, interface readiness or both.

The overall goal of this first version is to, based on the currently available components, partial integrations and tailoring for the execution environments, pave the path for a set of functionalities to be validated and demonstrated in a **disaggregated way** - and not in an end-to-end one - and under controlled conditions. Such technical and legal compliance validation are in the scope of Task 5.5 and will be delivered in D5.4 (by M24).

1.2 Background on P1 planning process

This deliverable builds upon several different inputs, across three Work Packages, and stemming from the involvement among all the consortium partners. The use cases specification and planning (T2.1), the elicitation of the ARCADIAN-IoT framework requirements (T2.2) and the specification of the architecture of ARCADIAN-IoT framework (T2.3), all produced within WP2 were a foundational basis for P1. The considered outputs can be found in deliverables D2.2 [1], D2.4 [2] and D2.5 [3], respectively.

To ensure the success of the integration activities, which require functional prototypes of the technical components, there was a considerable effort in establishing the target features set for the first version of the ARCADIAN-IoT framework (P1) – which directly affected the features set expectable only in time for the final version (P2). 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. Each of the A, B and C items, mentioned in Section 1.1, have

themselves had an impact on the resulting plan, as follows:

- Item A, the ARCADIAN-IoT functionalities and interfaces expected from each component for P1 were directly dependent on the estimated component's maturity and the partner's research priorities and actually obtained results (within WP3/WP4);
- Item B, i.e., ARCADIAN-IoT intra- and inter-plane integration, depended on agreement and coordination between partners, as well as the establishment of 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 with a central role or with wide presence across multiple use cases;
- Item C, ARCADIAN-IoT components' tailoring and testing for enabling subsets of the use cases defined within the project, depended on alignment between component's owners and domain owners;
- the **applications, hardware and other artefacts** readiness and adequateness for demonstrating ARCADIAN-IoT (e.g., invoking its APIs or using its message bus), 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 research and development outcomes of these activities are documented in deliverables D3.2 [4] and D4.2 [5]. Section 2 provides a description of functionalities available under each component both on the horizontal and the vertical planes.

1.3 Document structure

The remainder of this document is presented as follows:

Section 2 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 3 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 4 documents the current readiness of each of the domains' use cases, and the status of ARCADIAN-IoT framework integration towards enabling them.

Section 5 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 INTEGRATION APPROACH

This section starts by recapping the ARCADIAN-IoT framework in which the integration builds upon. It is followed by a description of the approach taken to not only integrate technical components with each other, but also the integration with domains' uses cases. An overview of

the supporting tools is provided at the end of this section.

2.1 Recap on ARCADIAN-IoT architecture

As mentioned in the previous section, Task 2.3 has produced the overall ARCADIAN-IoT architecture (available in Deliverable 2.5 [3]), both in its conceptual and deployment models. The architecture was defined on M12 of the project when the research and development activities of the technical components were in their initial stages. As a result, from M12 to M20 there were iterative adjustments to the architecture with respect to some of the components' interfaces and interactions, as the result of low-level design discussion and decisions. Figure 1 depicts the updated ARCADIAN IoT architecture; the list of main changes from the version documented in D2.5 are listed in Table 1:

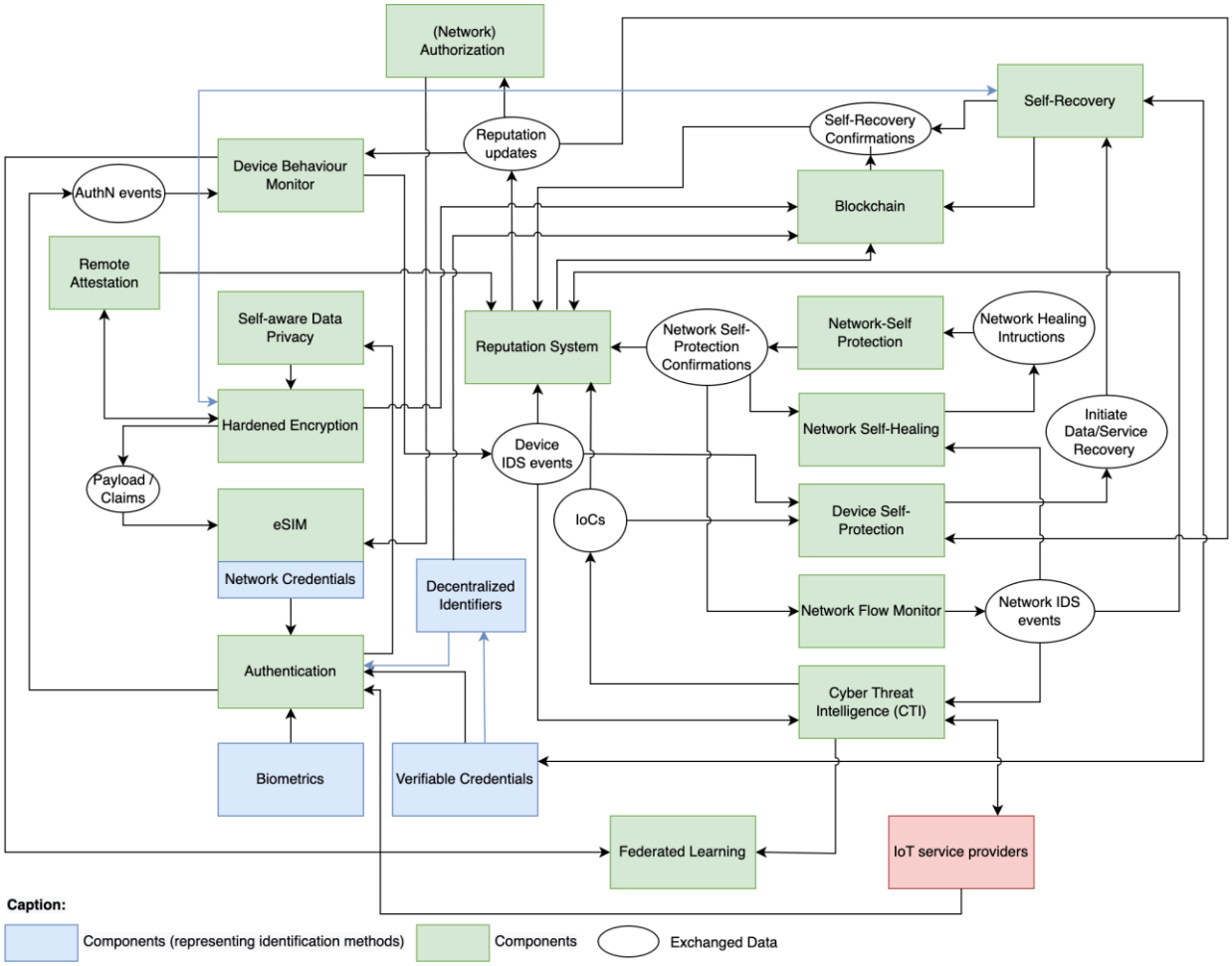

Figure 1 - ARCADIAN-IoT functional architecture

Table 1 – List of updates to ARCADIAN-IoT functional view

Modification	**Justification**
Representation of **IoT service providers**	The interaction with **SSI** for service authentication or **CTI** for receiving CTI events was missing.
SSI was replaced, now being split between **VCs** and **DIDs**	SSI was overly-simplifying and hiding two important components of ARCADIAN-IoT framework.

Link between **HE** & **Authentication**	Devices will need to authenticate to get keys. In D2.5 the logic was that this connection goes through Self-Aware Data Privacy, but the development went in the way that the connection is direct (since HE can be used without Self-Aware Data Privacy).
New link between **Self-recovery** – **HE**	**Self-recovery** also uses **HE** to encrypt data that is needed for recovery.
Removed link between **Authorization** and **Self-recovery**	Weak link, not worth representation.

As observed from the figure, 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 4).

The updated deployment view is also presented, being depicted in Figure 2. The overview of main changes is presented in Table 2 and is reflected in blue coloured interfaces in Figure 2.

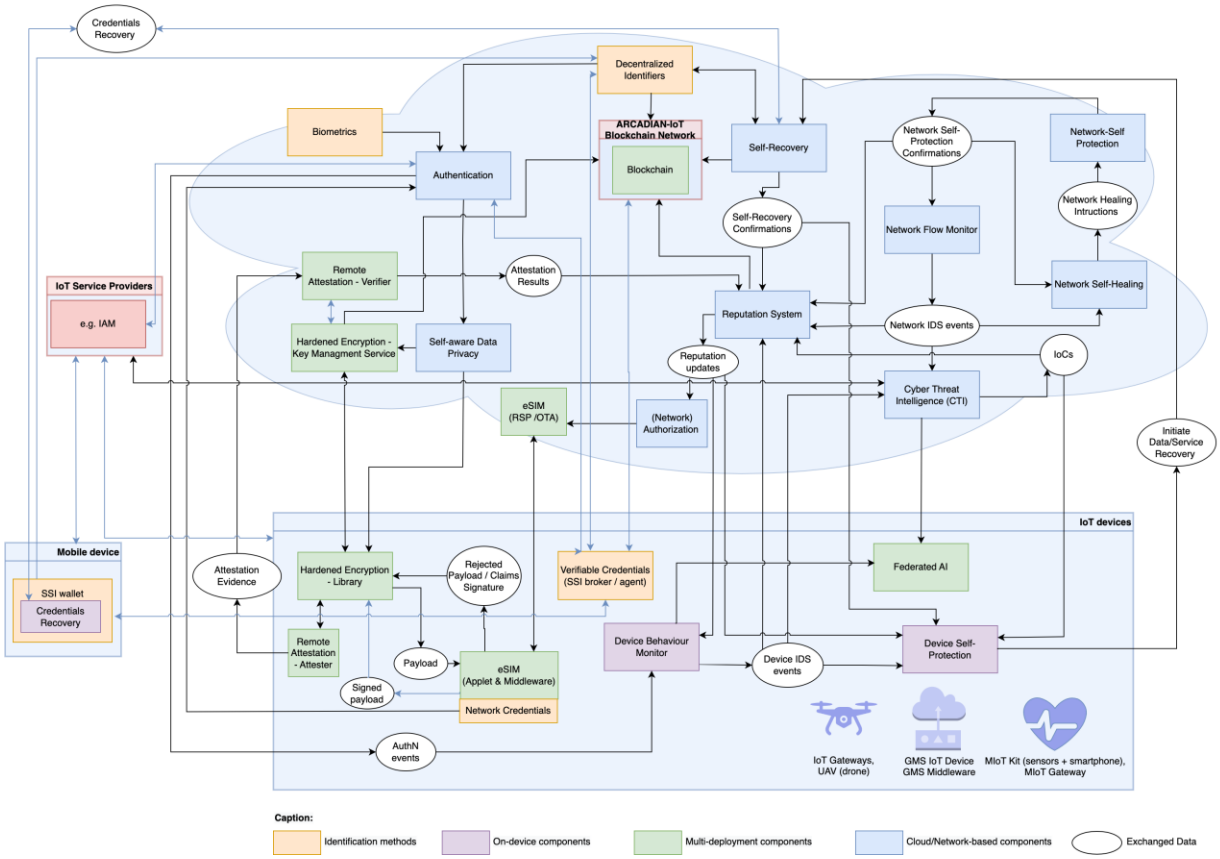

Figure 2 - ARCADIAN-IoT deployment view

Table 2 - List of updates to ARCADIAN-IoT deployment view

Modification	Justification
Merged **Policy Manager** into **Reputation System**.	Deployment view intends to provide a high-level view of distribution between different deployment locations (not a complete sub-component view).
Representation of **IoT service providers**	Important to enable depict how ARCADIAN-IoT framework is accessed by 3rd parties for ensuring their services security, privacy and trust compliance.
Added link between **IoT service providers** & **CTI**	3rd parties generalized (previously only referred to external organizations exchanging CTI).
TBC (raised by ATOS): new link between **IAM @IoT Service Providers** and **Authentication**	Representation of IoT service providers.
Inclusion of **Mobile device** (split from IoT device)	Only Mobile Devices (not IoT devices) can hold SSI wallet.
SSI was replaced, now being represented as **VCs** (in the IoT device, including SSI broker / agent) and **DIDs** (@network/Cloud, replacing previous SSI module)	SSI was overly-simplifying and hiding two important components of ARCADIAN-IoT framework.
VCs link with **Authentication**, **Blockchain**, **SSI wallet** @mobile device, and **DIDs** @ network	Result of split between DID and VCs.
TBC (raised by ATOS) link between **Authentication – DIDs** (previously SSI)	Result of split between DID and VCs.
Representation of **Blockchain** as part of new domain **Blockchain network**	
New link between **blockchain** and **VCs**	Result of split between DID and VCs.
Remote Attestation representation now explicitly depicts **RA - Attester** @device & renaming Remote Attestation to **RA – Verifier** @network. Moreover, representation of signed payload going from **eSIM** to **HE Library**	Improved representation accuracy.
New link between **Hardened encryption - Key Management** and **Authentication**	Devices will need to authenticate to get keys. In D2.5 the logic was that this connection goes through Self-Aware Data Privacy, but the development went in the way that the connection is direct (since HE can be used without Self-Aware Data Privacy).
Included **Credentials Recovery** within **SSI Wallet**, interacting with Self-Recovery	Reflects the support of wallet credential recovery via reading QR Code with recovery key to fetch backup file in Self-Recovery.
Added link between IoT service provider & Mobile device. IoT service provider & IoT device	Authentication processes are initiated in the device, passing through the IoT service providers, before reaching Authentication component.
Removed Biometrics - Hardened Encryption (libraries) link	Incompatibility with target use cases, unclear added-value of integration. To be further assessed before P2 delivery.

The following subsections provide additional detail on the interfaces and functionalities that were a target of integration within P1.

2.2 Component to component approach

There are 20 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). One of first action points was the definition Prototype 1 (P1) characteristics and assumptions, which included the agreement on the interfaces to be used and the set of functionalities that must be integrated by P1.

2.2.1 Planned prototype functionality roadmap

A high-level overview of the plan for which functionalities are delivered at each prototype (P1 or P2) from the point of view of each component is presented in the tables presented next (Table 3 and Table 4).

Horizontal ARCADIAN-IoT Planes

Table 3 - Prototype functionality roadmap for components belonging to ARCADIAN-IoT horizontal planes

Plane	Component	Main P1 functionalities	Functionalities expected for P2
Privacy Plane	Self-aware Data Privacy	Access management refactor and integration with authentication component	Final prototype of P1 features
		Policy description and retrieval	Recommender System prototype
		Policy enforcement	
		Integration with Hardened Encryption	
	Federated AI	Data rebalancing	Communication-efficient and robustness functions for model resizing and sharing
Security Plane	Network Flow Monitoring	First prototype of DDoS known attacks detection with 5GS capabilities	Final version implementation
		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	Data Extraction and log parsing from	Data extraction and log parsing

	Behaviour Monitoring	devices (Linux)	(smartphone / Android)
		Data preparation (transforming log info into ML model inputs)	Integration of data rebalancer (from Federated AI)
		Event classification	Generation of CTI events
		Implementation of central model aggregator	
		Federated training	
	CTI	Internal automated functions	IoC aggregation
		Event parsing and formatting	ML model manager
	Network Self-Healing	Topology discovery	Implementation of final version
		Prescriptive analytics	
		Integration with Network Flow Monitoring, Network Self-Protection components	
	Network Self-Protection	Enforce security mitigation rules	Implementation of final version
		Abstract different mitigation technologies	
		Integration with Network Flow Monitoring, Network-Self-Healing components	
	IoT device Self-Protection	Integration with Message bus and behaviour monitoring	Packaging for Android-based devices
		Self-protection policies (definition and storage)	Proposal of self-protection policies to device manager (Android devices)
		Enforcement of policies on device emulator (Linux)	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	Enhanced version of the security applet in terms of resilience and performance. Over the air management of the security applet.
		Middleware for integration of the HE with eSIM security applet in Python	Final version of the middleware in Python and an Android implementation of it
		HE library for encrypting data with ABE that can be used in Go, Python, C or Java	Final version of encryption library, public keys integrated with Blockchain
		Proof of concept of a decentralized key management system (deployed at XLABs servers)	Final version of key management 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)	Assess, specify and implement interfaces for integrating with ARCADIAN-IoT framework (e.g., Reputation System, Remote Attestation, Self-Recovery)
		Encryption / decryption middleware baseline solution, including encryption/decryption	
	Permissioned Blockchain	DID Method trust anchor	Implementation of final version
		ABE Key Management	
		Reputation scores	

Vertical ARCADIAN-IoT Planes

Table 4 - Prototype functionality roadmap for components belonging to ARCADIAN-IoT vertical planes

Plane	Component	Main P1 functionalities	Functionalities expected for P2
Identity Plane	SSI (Decentralized Identifiers & Verifiable Credentials)	Public Decentralized identifier for devices	IoT Device DID Exchange (DIDCOMM) - Implicit flow to connect to service DID on start up
		FE issuer of ARCADIAN-IoT Verifiable Credentials to Devices & Persons	Full SSI IoT Device Support (Ubuntu OS on Domain A)
		Integration with Authentication Component	
	eSIM (Network-based authentication)	Zero-touch authentication of devices in third-party services leveraging cellular networks credentials	Improvements and optimization.
	Biometrics	Pre-processing of video stream	CRUD operations and extension of functionalities
		Face detection model	
		Face recognition model	
		User registration	
		Identification results report	
	Authentication (Multi-Factor Authentication)	First prototype of Multi-Factor Authenticator (MFA) – person authentication - integrating with Biometrics, Network-based Authentication and Verifiable Credentials	Integration with Self-aware Data Privacy and Behaviour Monitoring
			Device MFA
			Person MFA using 2 devices as sources of data

Trust Plane	Network Authorization	Network-based Authorization running in a network testbed, integrated with Reputation System (with virtual test devices), with a base set of trust-based authorization policies	Implementation of the final set of trust-based authorization policies
		Distribution of trustworthiness information to the eSIM for self-protection and self-recovery actions	Integration of real devices running in real networks with the testbed running the Network-based Authorization
	Reputation System	RabbitMQ communication with components sending event information	Reputation score calculation
		Score reputation based on Alpha-Beta model (for events received)	Support entities' reputation storage in Blockchain
		Map reputation score with policies	Process Attestation Results for obtaining reputation scores (Remote Attestation integration)
		Simple component to allow user to specific policies (via a REST API)	
	Remote Attestation	Ability of Attester to request encryption & signing of dummy claims (via Hardened Encryption libraries), generate and format Evidence (CBOR) and transmit it to Verifier	Claims collection in real devices (Android, Linux OS's)
		Ability of Verifier to process encryption keys, receive dummy reference values and evidence	Appraisal of evidence for generation of Attestation Results and transmission to Reputation System
Recovery Plane	Self-Recovery	First prototype for eSIM-related operational recovery process	Final prototype of P1 functionalities
		Data backup and recovery client + server part	
		Backup encryption via integration with HE	
	Credentials Recovery	No functionalities available for P1.	Issue recovery key as QR Code for SSI Wallet (self-recovery integration)
			Support credential recovery on wallet
			IoT device public DID recovery supported by DID method
			IoT device DID connection used to recover other credentials (e.g., VCs)
			eSIM credential recovery

2.2.2 Integration status overview

In addition to the functionalities defined to be ready for P1, it was also necessary to define which interfaces would be available in P1. For that reason, a map showcasing the direct integration between each of the components was defined. Figure 3 depicts the integration status that each component currently has with other ARCADIAN-IoT components.

Planned for P1
Planned for P2
No integration planned

		Decentralized identifiers	Network-based IoT device auth	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	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				P2c	P1a	P1e					P2									P2	P2	P1b
Network-based IoT device auth	TRU				P1		P1e																
Biometrics	UWS				P1																		
Multi-factor Authentication	TRU	P2	P1	P1		P1	P1e				P2		P2		P2						P2		
Verifiable credentials	ATOS	P1a			P1		P1e					P2											P1b
Onboarding IdP	ATOS	P1e	P1e		P1e	P1e						P2											P1b
Network-based authorization	TRU								P1		P1f									P1			
Reputation Systems	IPN							P1		P2					P1	P1	P2	P2	P1	P1	P2	P2	P2
Remote Attestation	IPN								P2												P1	P2	
Self-recovery	XLAB				P2			P1	P2			P2								P2	P1	P2	
Credentials Recovery	ATOS	P2				P2	P2				P2									P2			
Self-aware data privacy	MAR				P2																P1	P2	
Federated AI	RISE														P1		P2						
Behaviour Monitoring	IPN								P1					P1			P2			P1			
Network Flow Monitoring	UWS								P1								P1	P1	P1				
Cyber Threat Intelligence	RISE								P2					P2	P2	P1	P2d			P2			
Network Self-Healing	UWS								P2							P1			P1				
Network Self-Protection	UWS								P1							P1		P1					
IoT device self-protection	IPN							P1	P1		P2	P2			P1		P2						
Hardened Encryption (w eSIM)	XLAB	P2			P2				P2	P1	P1	P1	P1										P2
Hardened Encryption (w/ cryptochip)	BOX2M	P2							P2	P2	P2	P2	P2										
Permissioned blockchain	ATOS	P1b				P1b	P1b		P2												P1		

Figure 3 - Component to component integration mapping

Notes supporting interpretation of Figure 3:

a Verifiable Credentials integrated with persons Mobile SSI Wallet in P1;
b Public DIDs created on private Ethereum sidetree node for VC issuer in P1;
c In the case of constrained devices support DIDs (with eSIM support) it will be explored to make use of DPOP token / SIOP for DID authentication;
d Federated operation;
e Onboarding IdP ready for integration in M20 and supports Persons for P1 and Services (IoT Devices planned for P2).

As it is possible to observe, a significant number of component integrations that were planned for P1 were achieved. The consortium has made it possible to fulfil most of the planned component integration, 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. Some of the key aspects that can be highlighted include:

- 20 of the 21 main components[1] are in the process of being integrated for P1, the only exception being the Credentials Recovery component, which will only start integration towards support in P2.
- 9 of the components are to be fully integrated with the remaining ARCADIAN-IoT components by P1, with 10 of the components having partial integration in P1 (and the remaining integration effort to be done by P2).

Some issues or deviations are worth noting. For instance, the need for a coordinated onboarding process was initially not contemplated (in WP2), and was added as a result of diverse discussions relating to integration and use case support. This has led to the inclusion of new sub-components and has resulted in additional unplanned e. There are also minor integration intersections that slightly ran over the planned deadline (described in section 3) and are estimated to be finalized by M21 (i.e., one month delay). Such aspects are mentioned in sections 3 and 4, where applicable.

2.3 Component to domain approach

2.3.1 Planned prototype functionality roadmap

The integration of technical components between themselves is a core part of integration efforts, there is also the need to integrate technical components with the specific use cases (initially specified in [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 also any necessary setup or configuration for smoothly running and executing the expected role in the use case.

Even though the thorough and complete integration of technical components with domains is only targeted for P2 (as the delivery of the domains implementation is formally due in M24), the consortium strived to achieve preliminary integration with some of the use cases that already demonstrate technical maturity for a functional integration.

The selection of use cases targeted for ARCADIAN-IoT validation in P1 was driven from the expected readiness from the domain owners (LOAD, BOX2M, RGB) perspective, and set between months 15 and 16. While this has influenced the integration planning of ARCADIAN-IoT components' owners – e.g., leading to prioritization of research and development in features or interfaces directly enabling the subset of "**P1 use cases**" - , other parallel implementation, feature provisioning or integration efforts that partially support use cases only planned for P2 ("**P2 use cases**") have also been pursued. Activities in the latter scope have purposely been left off the report (i.e., section 4), and can partially be drawn from D3.2 [4] and D4.2 [5] deliverables. For the sake of clarity, the list of use cases addressed in P1, as well as those left for P2, are listed in Table 5 and Table 6, respectively.

[1] Onboarding IdP can be considered as supporting component for enabling ARCADIAN-IoT's identity management, thus is not accounted for

Table 5 - Use cases addressed in P1

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 data
BOX2M	B1	New device registration
	B2	GMS IoT device data gathering and transmission process
RGB	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

Table 6 - Use cases planned for P2

Responsible	Use case ID	Use case name
LOAD	A4	Person requesting a DGA service
	A5	DGA service
	A6	Drone security or privacy incident
	A7	Personal device security or privacy incident
BOX2M	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
	C5	Patient MIoT devices security or privacy incident
	C6	MIoT Cloud services security or privacy incident
	C7	Medical 3rd party security or privacy incident

2.3.2 Integration status overview

Figure 4 represents the activities around the integration of technical components with the domains' use cases targeted for P1. The following are some of the key figures with respect to supporting P1 use cases:

- 8 of the specified 17 use cases are targeted for being supported and validated in P1.
- 12 of the 20 components are in the process or have complete integration efforts for supporting the targeted use cases in time for P1
- 1 component (Network-based authorization enforcement) is only expected to take part in integration activities for the P1 use cases in time for P2
- 6 components (e.g., those from Security or Privacy plane) are not involved and will not be demonstrated in the P1 use cases; thus, their integration will only be performed in the

context of (part of) the use cases targeted for P2 (i.e., black rows on the security and privacy plane).

Planned for P1 / Planned for P2 / No integration planned		USE CASES							
		A1	A2	A3	B1	B2	C2	C3	C4
		LOAD	LOAD	LOAD	BOX2M	BOX2M	RGB	RGB	RGB
Decentralized identifiers	ATOS	P1	P1		P2	P2	P1		P1
Network-based IoT device auth	TRU	P2	P1						
Biometrics	UWS	P1	P1	P2					
Multi-factor Authentication	TRU	P2	P2			P2			
Verifiable credentials	ATOS	P1	P1				P1		P1
Network-based authorization	TRU	P2	P2	P2	P2	P2	P2	P2	P2
Reputation Systems	IPN	P1	P1	P1	P2	P2	P2	P2	P2
Remote Attestation	IPN	P1		P1	P2	P2	P1		
Self-recovery	XLAB				P2				
Credentials Recovery	ATOS				P2				
Self-aware data privacy	MAR	P2		P2	P2	P2	P1	P1	P2
Federated AI	RISE								
Behaviour Monitoring	IPN	P2	P2	P2	P1	P1	P2	P2	P2
Network Flow Monitoring	UWS								
Cyber Threat Intelligence	RISE								
Network Self-Healing	UWS								
Network Self-Protection	UWS								
IoT device self-protection	IPN								
Hardened Encryption (via eSIM)	XLAB	P2	P2	P2			P1	P1	P1
Hardened Encryption (via cryptochip)	BOX2M				P1	P1			
Permissioned blockchain	ATOS	P1	P1	P1			P1		P1

Figure 4 - Use Case Integration Plan

The integration into ARCADIAN-IoT domains mostly depends 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 (only a portion is represented in this document with respect to P1), considerable efforts are made in parallel and iteratively. This leads to blocks (or sets) of functionalities and interfaces being ready in some of the use cases, at different speeds and at different levels.

The information presented so far, intends to grant the reader with the necessary information about the integration approach of ARCADIAN-IoT framework. Additional details will be provided in Section 3 (component integration) and section 4 (preliminary domain support).

The following section provides complementary information with respect to the tools that support the integration of components and respective use cases.

2.4 Supporting communication tools

The ARCADIAN-IoT framework is comprised of approximately 20 technical components, which, in cases of full-fledged deployment, may translate into the associated physical deployment locations spanning resources as diverse as IoT devices, mobile devices, 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 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 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>. Although the routing schema may change until the final release of ARCADIAN-IoT Framework (i.e., P2), 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, P1 currently requires authentication for all components connected to RabbitMQ's instance (deployed at IPN's premises and available to partners). Non-authenticated communication attempts are not allowed and requests are refused.

TLS-enabled communications, targeting ARCADIAN-IoT components authentication and encryption of the exchanged inter-components traffic (protecting against e.g., man-in-the-middle attacks), will only be available in P2. Other aspects such as time to time, dead letter exchange or dead letter routing key will be defined and set up in the next version (P2) of ARCADIAN-IoT framework. Other communication approaches (via REST APIs and DID Comm) will equally employ appropriate security considerations.

3 COMPONENT INTEGRATION STATUS

As mentioned in the previous section, Prototype 1 (P1) is comprised of a set of functionalities which can be characterized 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 4.

Thus, this section describes the outcomes of the integration between ARCADIAN-IoT components for P1: 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.

3.1 Integration status for horizontal planes

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

3.1.1 Privacy plane

3.1.1.1 Self-Aware Data Privacy (MAR)

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.

An integration or cooperation with the Multi-factor Authentication component is already agreed and will be explored in the second half of the integration cycle, in time for P2.

3.1.1.2 Federated AI (RISE)

Integration with Behaviour Monitoring (IPN)

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.

Since the 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 local 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 re-training phase of the models.

3.1.2 Security plane

3.1.2.1 Behaviour Monitoring (IPN)

Integration with IoT device self-protection (IPN)

The implementation of the Behaviour Monitoring 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 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 (UC)

As in the integration with IoT device self-protection, in case the component identifies a malicious event, the Behaviour Monitoring 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).

Integration with Federated AI (RISE)

The Behaviour Monitoring 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 in order to guarantee the privacy of the devices that integrate this component. The process implements the Weighted Federated Average algorithm, which takes all of the model parameters from the participating clients, and outputs an aggregated scaled mean of the Neural Network weights.

The Behaviour Monitoring 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.

3.1.2.2 Network Flow Monitoring (UWS)

The Network Flow Monitoring (NFM) component is the horizontal plane component of the ARCADIAN-IoT framework responsible of 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 alert is generated in a JSON format (see details in D3.2) and transmitted using AMQP over a RabbitMQ message bus.

3.1.2.3 Cyber Threat Intelligence (RISE)

Integration with Network flow monitoring (UWS)

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 D3.2) 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 analyzed by the ML-based subcomponents. Once the IoCs are generated, they are ready to be shared.

3.1.2.4 Network Self-healing (UWS)

The Network Self-Healing (NSH) is the horizontal plane component of the ARCARDIAN-IoT framework that receives the information about detected cyber threats from Network Flow Monitoring (NFM) component. It deploys multiple instances to discover the network topology and performs Predictive Analytics to create a decision that contains WHAT, WHERE, WHEN and for HOW LONG the attack should be stopped. Such 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 (UWS)

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. 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.

3.1.2.5 Network Self-protection (UWS)

The Network Self-Protection (NSP) is the horizontal plane component of the ARCARDIAN-IoT framework that stops the illegitimate traffic from attacks against the network. It is a distributed component that is deployed in every single point within the data plane where attacks could be stopped. NSP uses the decisions created by the Network Self-Healing component to stop the attack in the data plane.

Integration with Network Self-Healing (UWS)

When the Network Self-Healing create the decision, it sends a message through the Network Healing Instruction exchange where the Network Self-Protection is listening. 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.

3.1.2.6 IoT Device Self-Protection (IPN)

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. For Prototype 1, the threat information is sent by the Device Behaviour Monitoring.

Integration with Behaviour Monitoring (IPN)

The threat information comes from the devices Behaviour Monitoring component, which runs at device level, or elsewhere when the devices are not able to support local deployment. These messages are sent in JSON format via the RabbitMQ exchange (dbm_exchange) to the device self-protection queue, and contains data related to the IoT device 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.

A report of selected the self-protection actions is then relayed to other interfaces of the ARCADIAN-IoT project (e.g., to the reputation system (only available in P2). The intention is to provide information of protective measures taken when threats are detected so that the reputation of a device can be updated accordingly.

In scenarios where the Device Behaviour Monitoring and Device Self Protection are not deployed directly on the device, ARCADIAN-IoT message bus is the communication channel. On the other hand, when both run at device level, both are deployed as a bundle and use internal component communication.

3.1.3 Common plane

3.1.3.1 Hardened Encryption with RoT signatures (XLAB)

Integration with Self-recovery (XLAB)

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.

Currently, 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 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 are shared through an API provided by the key management subcomponent.

Integration with Remote Attestation (IPN)

Similarly, as with the Self-recovery, the Hardened encryption has been successfully used by the Remote Attestation component. More precisely, the component uses the encryption, while the RoT signatures with eSIM needs to be further supported (see Section 3.2 Remote Attestation).

Integration with eSIM (TRU)

The integration between the encryption technology and the eSIM security applet (Root of Trust) has been done and tested in a real device (Raspberry Pi 3, Model B+, with 2 different modems, the SIMCom SIM7600G-H-M2 R2 (LTE Cat4 M2), and the Monarch GM01Q (LTE Cat M1)). For this purpose, and according to (departing from) GSMA IoT SAFE specifications, 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. For the moment, and according to requirements related with the drone device implementation in Domain A, this *device middleware* is developed in Python. In the forthcoming period these components development and integration will target a different environment: an Android smartphone.

Integration with Authentication component (TRU)

A basic key management subcomponent of Hardened encryption has been implemented that needs to be further integrated with Authentication component, so that the appropriate cryptographic keys are delegated only to authenticated devices. The integration is planned for the next prototype.

Integration with Permissioned blockchain component (ATOS)

To enhance the thrust in the system, public cryptographic keys will be published on a blockchain. Currently the key management subcomponent of HE holds this information, hence needs to be trusted. Similar to the Authentication component, the integration is planned for the next prototype.

3.1.3.2 Hardened Encryption with cryptochip system (BOX2M)

A dedicated middleware API was designed and has been set up for secure (TLS)-enabled) communications with external systems. It has currently been tested with Self-aware Data Privacy (refer to section for domain B status description in Section 4 for further details).

Moreover, the integration with the message bus for consumption of events from the Behaviour Monitoring component is ongoing.

3.1.3.3 Permissioned Blockchain (ATOS)

Integration with Reputation / Hardened Encryption (XLAB)

The Permissioned Blockchain component is based on Hyperledger Fabric. The component implements smart contracts for publishing ARCADIAN-IoT data objects on top of a deployed blockchain network. It is noted that the actual data being published is stored off-chain while a cryptographic hash of the data is stored on-chain for verifying the integrity of the actual data. The components that plan to publish ARCADIAN-IoT data objects to the Hyperledger Fabric blockchain are the Reputation System (P2), and Hardened Encryption (P1). Consumption of the ARCADIAN-IoT data on Hyperledger Fabric is controlled by clients authenticated by X.509 certificates who will ultimately then control who it makes the data available too. End users of the data are any organisations that implement a peer node and are given access to the blockchain network where the data objects are published.

Integration with Decentralized Identifiers (ATOS)

The Decentralized Identifiers component has considered to make use of a dedicated smart contract to publish public Decentralized Identifiers to the Hyperledger Fabric blockchain but is presently pursuing an integration with Sidetree Node, that stores root anchor files hash stored on

the blockchain, which is deployed on a Private Ethereum (P1). For consumption of public Decentralized Identifiers, then it is more common that this access is given publicly as its name suggests, but currently it is restricted by only making it available to organisations that make up the private blockchain network.

3.2 Integration status for vertical planes

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

3.2.1 Identity plane

3.2.1.1 Decentralized Identifiers (ATOS)

Integration with Permissioned Blockchain (ATOS)

Public Decentralized Identifier is currently provided by a Sidetree implementation integrated with a Private Ethereum Blockchain in P1. It is still under consideration possible alternative integrations with the Hyperledger Fabric blockchain for P2.

Integration with Verifiable Credentials (ATOS)

(1) The ARCADIAN-IOT Framework makes use of the Ledger uSelf SSI Broker / Agent solution in the Verifiable Credential´s component in P1. As regards, Decentralize identifiers, the SSI Broker is currently integrated with the Sidetree implementation and requests to create a Public did:elem based on a public key 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 also 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 and has currently issued Person VCs to SSI Wallet signed by the private key associated to the frameworks did:elem.
(3) DID:WEBs are also supported for the piloting Service Providers to make use of to publish their private keys for their registering their application services to the framework. This is in the process of being made available to Service Providers for supporting end user, device and services registration.

3.2.1.2 Hardware-based authentication using network credentials (TRU)

This component integration happens with Solution Providers and ARCADIAN-IoT's Multi-factor Authentication (MFA). Currently a complete deployment is available and being tested (as standalone, not as part of the MFA) with Solution Providers (further description in section 4).

Integration with MFA (TRU)

The integration with the MFA has been performed and successfully tested (not integrated in Solution Providers technology yet – this is ongoing). The specifications and sequence diagram for such integration is made available in the project shared folder.

Further technical and research details about this component, its integration with the MFA, as well as next steps, can be found in the public deliverable D4.2.

3.2.1.3 Biometrics (UWS)

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 (TRU)

- 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 is already implemented including each of the previous interfaces, prototype 2 will further improve by adding the capability of updating/deletion of people's faces and reducing video streaming delay. Besides, most information will be shared with MFA in order to provide more details about the accuracy of the verified user.

Table 7 - Example of Biometrics authentication request and reply for facial recognition

Request	Reply

POST
IP:PORT/authenticate/
Headers: X-AIOT-AUTH-DID: DID
Body: {'BiometricsImage:' Image <Base64>}
	Error 400 if error in request.
	Success 200 with body:
	{
	"result": "Code with result of the authentication
	0: Authentication Complete
	1: No faces detected
	2: More than one face detected
	3: Other error"
	"verified": "Boolean (True or False)"
	}

3.2.1.4 Multi-factor Authentication (TRU)

ARCADIAN-IoT's multi-factor authentication (MFA) currently has integration performed and tested with the Hardware-based authentication (that uses network credentials), and with the Biometrics component. There is also integration ongoing with the Verifiable Credentials component. All the integrations target, in this first prototype, person authentication.

The technical specifications for integration with the Service Providers are also available and were discussed with the first two domains that will have integration with this component (Domain A and C). More information can be found in section 4.

The integration of this component with the Self-Aware Data Privacy and with the Self-recovery components is also agreed and will follow-up next.

3.2.2 Trust plane

3.2.2.1 Verifiable Credentials (ATOS)

As previously indicated, the ARCADIAN-IOT Framework makes use of the Ledger uSelf SSI Broker / Agent solution in the Verifiable Credential´s component.

At this time, it is supported the issuing of Person Verifiable Credentials to end user´s mobile SSI Wallets signed by the ARCADIAN-IoT´s public Decentralized Identifier (DID). This is provided by an SSI IdP (FE/BE) module implemented as part of the Verifiable Credentials component and integrated with the SSI Broker to request connections, issuing and verification of Person Verifiable Credentials.

Integration with Decentralized Identifiers (ATOS)

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

Integration with MFA (TRU)

The SSI IdP and Broker/Agent is currently being integrated for the authentication flow with the Multi-Factor Authentication component, which will then be able to fully test the interwork with the piloting Service Providers. Finally, the registration flow and its interwork will be integrated during the first prototype piloting to provide the Service Providers a complete solution to register and authenticate users based on their Verifiable Credentials.

3.2.2.2 Network-based authorization enforcement (TRU)

Integration with Reputation System (UC)

The network-based authorization enforcement integrates with the project' reputation system through ARCADIAN-IoT's RabbitMQ infrastructure. Currently the component already receives and enforces reputation policies according to the entities reputation scores (all provided by the reputation system). The current prototype is working just for simulated devices, in a network testbed. In the forthcoming period, the technology will be demonstrated with real devices.

Integration with Self-protection and self-recovery (eSIM-based actions) (TRU)

This component also integrates with ARCADIAN-IoT's eSIM security applet for informing it of devices' trustworthiness. This integration is performed using GSM over-the-air (OTA) services, and it has been done and successfully tested with Android devices. With the devices' trustworthiness, the security applet is expected to enforce actions of self-protection and self-recovery.

3.2.2.3 Reputation System & Policy Manager (UC)

The Reputation System integrates with the remaining components through the ARCADIAN-IoT RabbitMQ infrastructure in the respective queues. The following tables summarizes the

integration with the respective components and data exchange information.

Integration with Authorization (TRU)

The Reputation Systems exchanges information regarding the policies that are associated with a specific component. Such exchange is performed in JSON format in the *rs_policies* queue.

In addition, the Reputation Systems also exchanges information regarding the updates of reputation for a specific entity in the *rs_updates* queue.

Integration with Device Behaviour Monitoring (IPN)

The Reputation System also exchanges information regarding the updates of reputation for a specific entity in the *rs_updates* queue.

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 (IPN)

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 (UWS)

Reputation System uses the information of network flows to determine reputation. This exchange is performed via the *nfm_exchange*.

Integration with Network Self Protection (UWS)

The Reputation System uses the information provided by Network Self Protection in the *nsp_exhange* via JSON format to determine reputation.

3.2.2.4 Remote Attestation (IPN)

The implementation of ARCADIAN's Remote Attestation (RA2IoT) solution was organized in two phases. In a first stage, the mbedTLS library was used to implement the cryptographic operations. In the second phase, the solution was evolved by replacing the mbedTLS-based cryptographic functions with Hardened Encryption ones. The communication was later on addressed and validated.

Integration with Hardened Encryption (XLAB)

In order to enable an efficient and smooth integration process, with as few setbacks as possible, the demo python implementation provided by XLAB was reused, as it already implements all the necessary functions. In the current version, RA2IoT leverages (and has been tested with)

Hardened Encryption to encrypt/decrypt dummy data, between one Attester and one Verifier – more information on RA2IoT can be found in D4.2 [5].

To better depict the integration status – and the particular importance of integrating with Hardened Encryption -, the output of one execution of the current implementation of RA2IOT is shown in Figure 1. After the initial booting, the Attester (top sub-figure) waits for attestation requests form the Verifier (bottom sub-figure). When it receives the attestation request, it parses it from CBOR to an internal structure and generates the evidence according to the claim selections provided by the Verifier, also copying the nonce that came in the attestation request. Then, it encrypts and signs the evidence through Hardened Encryption's libraries and sends the attestation response to the Verifier. In turn, the Verifier parses the CBOR response into an internal structure, decrypts the evidence in it through the Hardened Encryption Libraries, appraises this evidence and prints the attestation results.

The support for eSIM as Root of Trust is subject to the work from XLAB on Hardened Encryption, and is expected to be validated with a real device in the scope of P1.

3.2.3 Recovery plane

3.2.3.1 Self-recovery (XLAB)

So far, the Self-recovery component is developed in a mostly stand-alone state, data backups and retrieval are enabled via a REST API, clustered storage backend (Ceph) and client-side recovery scripts written in Bash. Integration with other ARCADIAN-IoT components will be the focus of the next period of development.

Several integration activities will be further explored and implemented in P2, for which we provide a summary. Data about recovery operations, such as frequency and consistency of backups and the success rate of recoveries will be communicated to the Reputation System via RabbitMQ. The client-side portion of the Self-recovery component will be expanded with the IoT Device Self-protection component, which will provide event messages that inform Self-recovery when a recovery process should be triggered. Finally, in the context of supporting Credentials Recovery, the server-side component will provide an API endpoint allowing the creation of a recovery key and QR code, which can be used in an out-of-band recovery of a mobile SSI Wallet.

The following activities have taken place targeting P1 validation:

Integration with Hardened Encryption (XLAB)

The bash client-side scripts are integrated with Hardened Encryption, providing encryption and decryption capabilities in the recovery process, available in P1. During P2, the client-side scripts will be ported to Python (and any other languages required) and also integrated with HE.

Integration with Authorization (TRU)

The server-side component of Self-recovery will interface 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. This integration will be further explored and implemented in P2.

3.2.3.2 Credentials Recovery (ATOS)

Integration with Self-Recovery (XLAB), Verifiable Credentials (ATOS), Hardened Encryption (XLAB)

The solution for Credential Recovery is agreed for Persons credentials noting that eSIM network credentials are handled by radio network processes outside of the ARCADIAN-IoT Framework components and that biometric face, as a physical trait, is not a credential that can be backed up. Therefore, Credential Recovery for persons concentrates on the recovery of a user´s mobile SSI Wallet. The solution is agreed at high level and is currently finalizing low level design and so is not meeting the first prototype integration timeline.

It is expected to fulfil integration with SSI Wallet, Hardened Encryption, SSI IdP and Self-Recovery during the first prototype piloting phase. IoT-Device support is planned for the final prototype.

3.3 Overall integration status

The previous section presented what each component currently supports or provides in the scope of integration with other components. As the ARCADIAN-IoT framework is composed of several components, it may be hard to grasp the overall status of integration. To complement what was presented before (as well as what is available in the project's repository), a graphical representation is the best way to complement the current integration status reporting. With that goal in mind, Figure 5 shows an overview of the conceptual ARCADIAN-IoT framework architecture focusing the clear identification of a) the currently integrated interfaces, b) the currently ongoing interface integration activities and c) interfaces planned to be supported in P2.

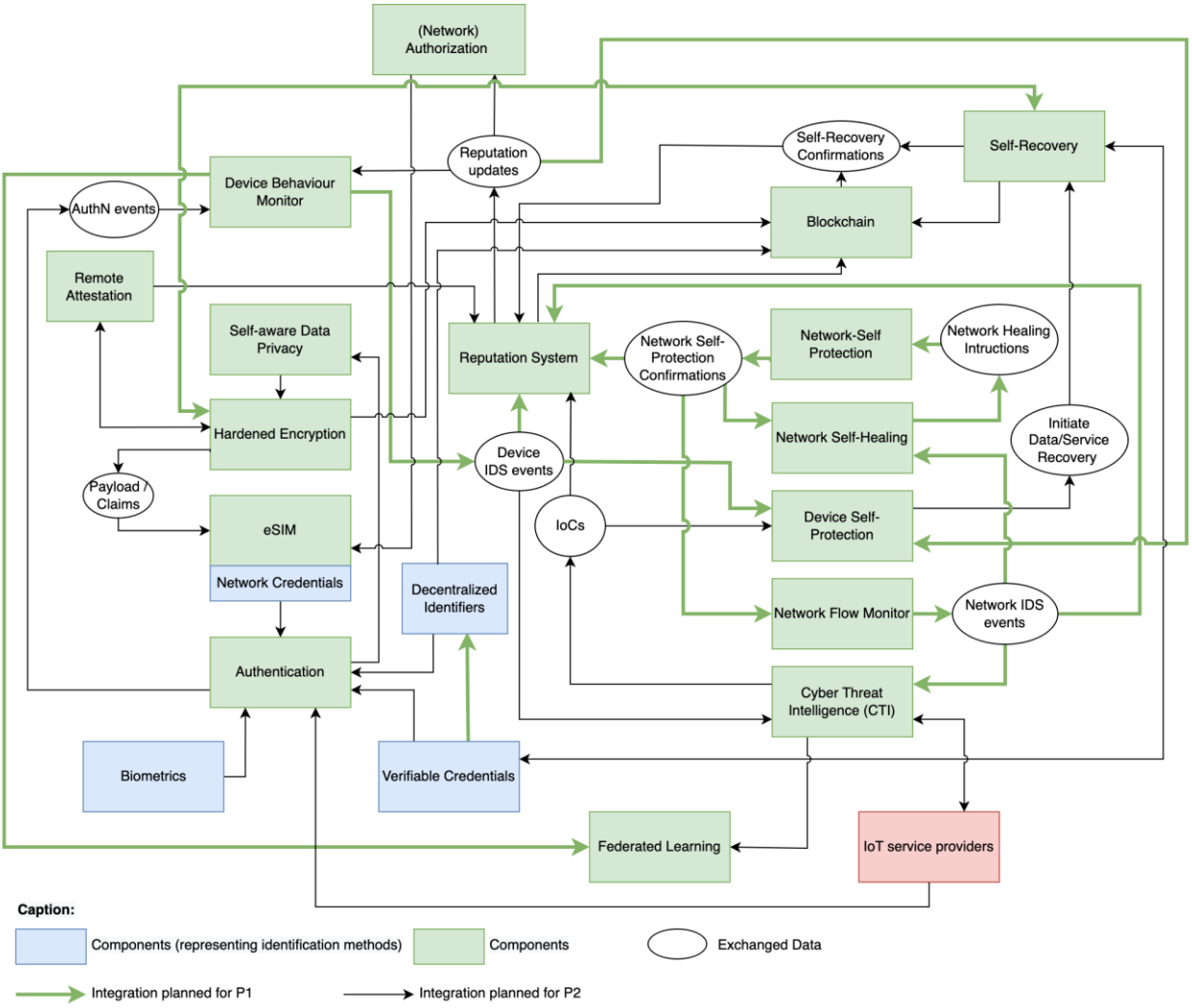

Figure 5 - Overview of integration status with respect to the functional architecture

As shown in the figure, a considerable number of interfaces is already functional or currently being integrated. The next action points are to not only to finalize the ongoing activities and address the P2 planned interfaces, but also to further validate and refine the currently integrated interfaces (as part of ARCADIAN-IoT's iterative implementation approach).

The following section considers different aspects of the integration - the integration with the domains' use cases.

4 DOMAIN INTEGRATION STATUS

This section provides an overview of the integration status of the ARCADIAN-IoT framework for each use case previously defined in D2.2 [1], spanning all three ARCADIAN-IoT domains, for the first ARCADIAN-IoT framework prototype - P1. This integration status covers the perspectives from both:

(1) **domain responsibles** (as IoT service provider), depicting the status over both the service-specific enablers and artefacts, as well as the preliminary support that each domain is providing for the integration with ARCADIAN-IoT

(2) **ARCADIAN-IoT technical component(s) responsible**, reporting the tailoring and adaptation performed to allow execution in the domain-specific environments.

Considering that a significant effort for P1 has naturally been in enabling the integration between the different ARCADIAN-IoT components (exposed in Section 3) – as most of its value directly depends on such coexistence -, the integration status of ARCADIAN-IoT for supporting the application domains is considerably more limited / lower at the time or reporting. Nevertheless, this integration work will be one of the main goals for the next period, considering its importance for enabling proper P1 validation (by M24).

4.1 Domain A - Emergency and vigilance using drones and IoT

Regarding P1 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, TRU, UWS, XLAB). Targeted ARCADIAN-IoT involvement in the domain is depicted via the Figure 6 below:

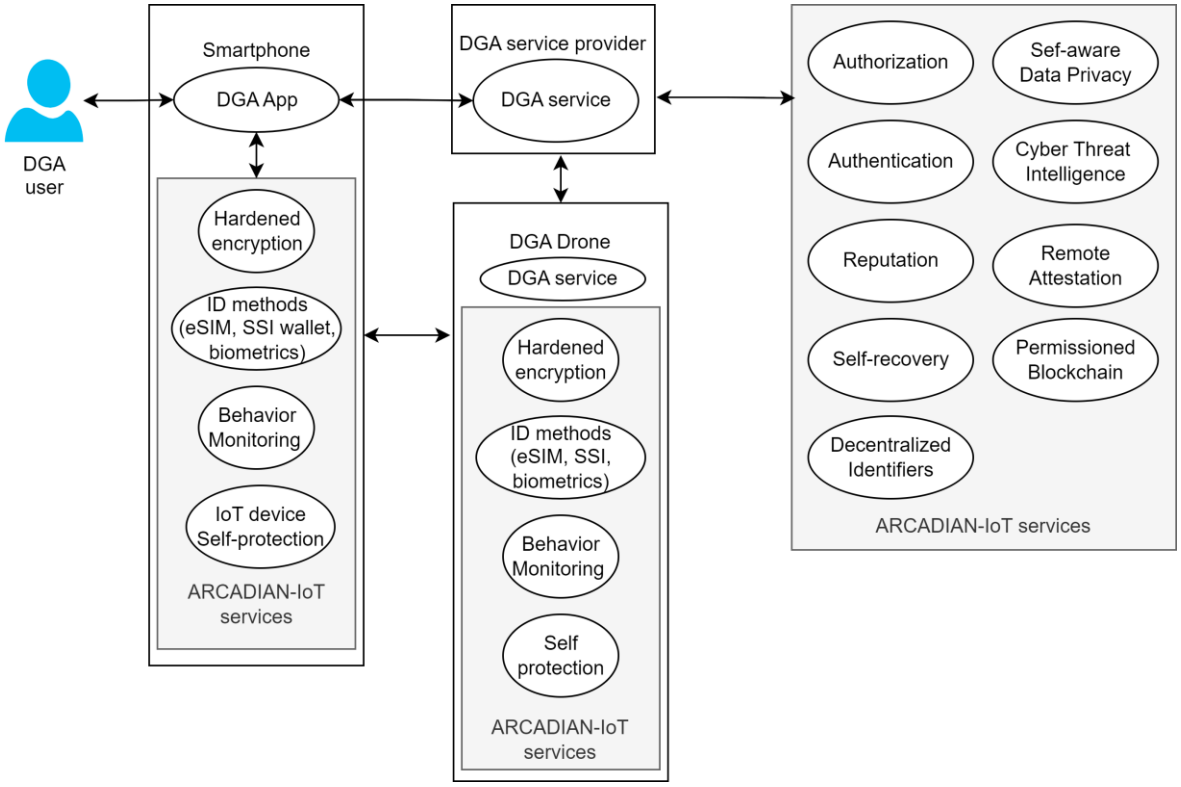

Figure 6 - High-level view of ARCADIAN-IoT involvement in Domain A

The use cases addressed in prototype P1 are A1, A2 A3, and the work done regarding each one of these use cases is described in the following subsections.

4.1.1 Use case A1 - Person registration at DGA service

4.1.1.1 Overall status

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:

App Development

- 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.
- Implementation of the module of face recognition (in 5 positions) to send to Biometrics for the user registration
- Exploring the interaction with Biometrics component
- Attended work sessions with partners regarding registration requirements

Backend Development

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

4.1.1.2 Component integration readiness

DIDs (ATOS)

- Peer DIDs are currently supported and working in the end user mobile SSI Wallet.
- Public did:elem is working for the ARCADIAN-IoT Decentralized Identity integrated with the Ledger uSelf SSI Broker / Agent.
- Service Provider and Service Registration using did:web is supported for P1, but as yet not integrated. This is needed to support:
    - Service Provider and service registration
    - End user Registration from the Service provider

Biometrics (UWS)

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. RTMP is used as to share the video stream in H264 video encoding from the drone to the Biometrics component; this is operational although it has not been tested yet as this will be for P2.

Once registered, the biometrics component will respond with the status of this registration. This integration has been successfully achieved. The results of the face verification algorithm when a video is received are provided by RabbitMQ, with an exchange named "bio_exchange". Also, this interface communicates the instructions of Registration, Update and Delete user information. AMQP interface for registration has been validated with LOAD.

Verifiable Credentials (ATOS)

- Issuing of Person VC is supported by the ARCADIAN-IoT SSI IdP sub-component requesting the frameworks SSI Broker/Agent to issue the Person VC to a user´s SSI Wallet. This is a pre-requisite for the registration use case.
- Requesting a Person VC by the SP to support registration flow is pending to start integration.

- Registration request to generate an ARCADIAN-IoT Identity (aiotID) and associate with user´s Person VC and network eSIM identity is pending to start integration.

Remote Attestation (IPN)

The transmission of reference values is accomplished by message bus (described in section 2.4) with the “ref_values” topic, through the rabbitmq exchange “ra_exchange”. We implemented a *Subscriber program* that subscribes to and manages reference values provided in JSON format by the service provider’s *publisher program* (implemented by LOAD). This interaction was established and information (dummy reference value) was exchanged successfully. As next steps for P1 completion, the transmission of one or more reference values for the agreed claims (e.g., DGA application fingerprint) will be performed.

Reputation System and Policy Manager (UC)

The sharing of person registration information was planned for P1 but is pending integration with the reputation system.

4.1.2 Use case A2 - Person authentication at the DGA service

4.1.2.1 Overall status

The work done to achieve the goals of use case A2 was related to the features allowing a user to register in the system by using a mobile app. This work involved the following activities:

App Development

- Elaboration of the UX/UI study and layouts of the authentication process of the user when logging in to the app with user credentials
- Implementing the module of face recognition to send to Biometrics for the user authentication
- Started the integration with the Network-based Authentication.
- Exploring the direct interaction with Biometrics component regarding authentication
- Attended work sessions with partners regarding interaction interface requirements

Backend Development

- Creating server infrastructure and initial services in LOAD's development server
- Component to support face images sending to Biometrics (in common with A1 use case)
- Component to support video streaming for the authentication via face recognition

4.1.2.2 Component integration readiness

DIDs (ATOS)

Same integration as mentioned in A1 (4.1.1.2).

Network-based Authentication (TRU)

Deployment of the needed technology has been done and integrated testing is ongoing.

Multi-factor Authentication (MFA) (TRU) – All the specifications have been discussed and agreed. Currently, as mentioned, LOAD is integrating the Network-based Authentication (already deployed) in an isolated manner, to allow to uncover issues with that particular authentication component. After finishing the integration of the Verifiable Credentials, the complete MFA will be ready for the integrated testing in Domain A, for person authentication.

Verifiable Credentials (ATOS)

Integration for authentication with SP is still pending, as it is waiting for MFA to Verifiable Credentials integration to be completed.

Reputation System and Policy Manager (UC)

The policies for authorization have been discussed with use case owner and in P1 they are configured through the frontend application. This application has been documented in D4.1. The policies management require information of ARCADIAN-IoT identifiers, as policies rely on such information. In P2 the policy's configuration will be performed though the Policy Manager API, relying on HTTPS.

The events to establish a reputation score have been discussed and partially documented in D4.1. Their integration into the reputation system requires the exchange of information through the Message Bus (e.g., RabbitMQ), thus still pending integration in P1. For instance, the information of failed login attempts.

Biometrics (UWS)

Please refer to the work performed for A1.

4.1.3 Use Case A3 - Person retrieving and editing personal data

4.1.3.1 Overall status

The work done to achieve the goals of use case A3 was related with updating user profile information that was first provided 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
- Started the integration with the Network-based Authentication
- Exploring the interaction with Biometrics component in what regards to update the Biometrics information, like the face images.
- Attended work sessions with partners regarding interaction interface requirements

Backend Development

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

- Component to support sending updated face images to Biometrics component

4.1.3.2 Component integration readiness

Remote Attestation (IPN)

Remote Attestation procedures for attesting device or service integrity have been implemented. The adopted approach has been executing one Attester instance and one Verifier instance in different containers. Watchdog-based remote attestation is supported, where remote attestation runs successfully (i.e., the claims / evidence are successfully appraised against the associated policies).

It is planned to do the necessary adaptations in the Attester for having it run properly in Android environment. Both service (appraising app fingerprint) and device attestation (appraising e.g., device model ID or build version) are scoped, but potentially only dummy values only be considered for P1.

Hardened Encryption (XLAB)

The encryption library is available to be used in Go, Java and Python, an interface with JS is planed soon. RoT signing has been tested with a Raspberry Pi, still needs to be developed for phones. A key management service is currently being prototyped.

Reputation System and Policy Manager (UC)

The operation of person, editing or retrieving information in the service, can be an indicator that the person cares about its data and the quality of the DGA service. Given this assumption, providing information regarding the operation of the person retrieving/editing information is relevant to update person's reputation. This integration was planned for P1 but is still pending.

4.2 Domain B - Secured early monitoring of grid infrastructures

Use cases B1 and B2 are the most mature use cases in domain B. A high priority was placed on these 2 use cases because they stand as a development base / core functionalities for the other use cases of domain B, led by BOX2M.

As such, they are the most suitable use cases for preliminary integration with ARCADIAN-IoT framework within the scope of P1. Targeted ARCADIAN-IoT involvement in the domain is depicted via the Figure 7 below:

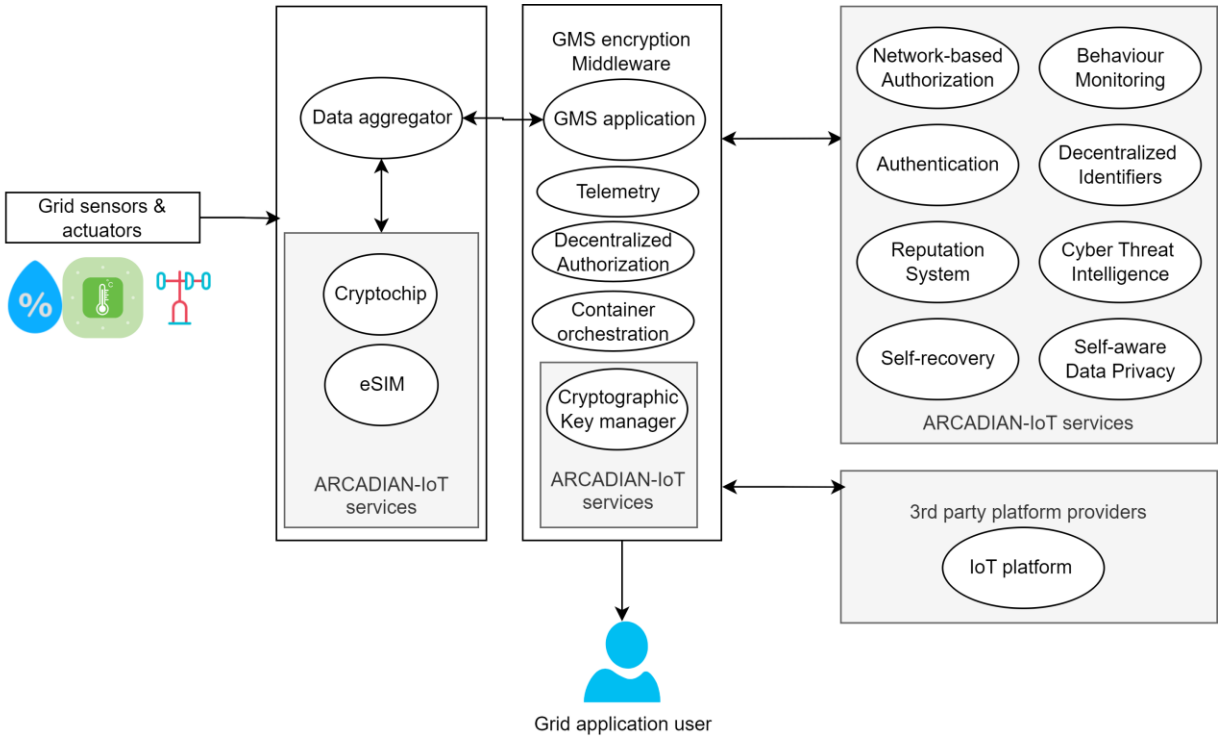

Figure 7 - High-level view of ARCADIAN-IoT involvement in Domain B

4.2.1 Use case B1 - New device registration

4.2.1.1 Overall Status

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 both when a new IoT device is provisioned, or when an existing device, due to operational lifecycle changes (e.g. broken crypto chip, stolen & recovered device, sabotage, hacking or cloning susceptions), has the crypto chip replaced or reprovisioned with new key pairs. 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 creates new associated ARCADIAN-IoT Identities (aiotID). Likewise, IoT devices must be issued with a Decentralized Identifier to perform DID authentication. Both activities have not yet been successful[2].

This work involved the following activities:

Hardware Development

2 The support of DIDs on the MCU with DIDCOMM specification was prototyped, but was found to consume too much resources. As such, no integration was achieved with the MCU for Decentralized Identifiers in P1. Other alternative approaches are being explored for the final prototype.

- Designing and deployment of device motherboard, powered by microcontroller, equipped with hardware crypto chip and with configuration port, according initial established HLD (high-level design)
- Design and development of communication and sensors interfacing plugin units, according initial established HLD (high-level design)
- Testing of device boards, 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 device exploitation tests

Firmware Development

- Designing and deployment of firmware structure and agents, including main firmware engine, communication firmware agent, edge computing firmware agent, encryption and decryption system firmware engine
- Testing of firmware, for each agent and wholistically (there were used GSM and ETH telecommunication modules, and 2 types of grid sensors interfacing boards)
- Regarding encryption and decryption system firmware engine, it was used the ST Electronics firmware, integrated / called into / by device microcontroller main firmware engine
- Credentials (user, pass) were provisioned for device registration and device unique authorization encryption key; the support of multiple devices was tested (to simulate a fleet).

Backend Development

- Setting up the cloud computing infrastructure on Microsoft Azure
- Setting 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 data base, API)
- Middleware backend testing (encryption and decryption were tested correlated with device provisioned keys, API was tested with BOX2M IoT platform as external interface)
- There were provisioned credentials (user, pass) for device registration into telemetry protocol broker and device unique authorization encryption keys into encryption and decryption system data base; there were tested multiple devices (to simulate a fleet)

Frontend Development

- Middleware front end code development (provisioning web pages for keys, API, monitoring process)
- Middleware front end testing (creation, removal, modifications, view)

4.2.1.2 Component Integration Readiness

Hardened Encryption using cryptochip system (BOX2M)

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 device ID and wrong key, corresponding respectively to the lack of ID and key provisioning, were performed sequentially, both in the device side and in the middleware side.

Self-aware data privacy (MAR)

Self-aware data privacy component is included in Martel's . Orchestra Cities is designed to be a multi-function platform allowing a broad range of IoT-enabled applications and services (e.g. waste and water management, transportation, air quality or energy consumption) to be effectively defined, offered and monitored. At this point in time, dedicated experiments were done testing Orchestra Cities integration with BOX2M live IoT platform, by connecting via secure TLS connection to the encryption / decryption middleware via BOX2M's dedicated API.

Behaviour Monitoring (IPN)

The current state of Behaviour Monitoring component is capable of receiving information from the encryption/decryption Middleware through a RabbitMQ queue. The listener was implemented and able to receive system call trace information and perform the classification of the incoming events remotely as it is not possible to run the component itself directly on the device. Concretely, the encryption/decryption middleware will provide the device behaviour monitoring with IoT devices' specific-events (e.g., authentication attempts).

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

4.2.2.1 Overall Status

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

4.2.2.2 Component Integration Readiness

At this stage, no additional activities regarding **Hardened Encryption (via crypto chip)**, **Self-aware Data Privacy**, or **Behavior Monitoring** were performed for specifically supporting use case B2. Please refer to the description included in use case B1.

4.3 Domain C - Medical IoT

Use cases C2, C3 and C4 are the most mature use cases in domain C, led by RGB. As such, they are the most suitable use cases for preliminary integration within the scope of P1.

Targeted ARCADIAN-IoT involvement in the domain is depicted via the Figure 8 below:

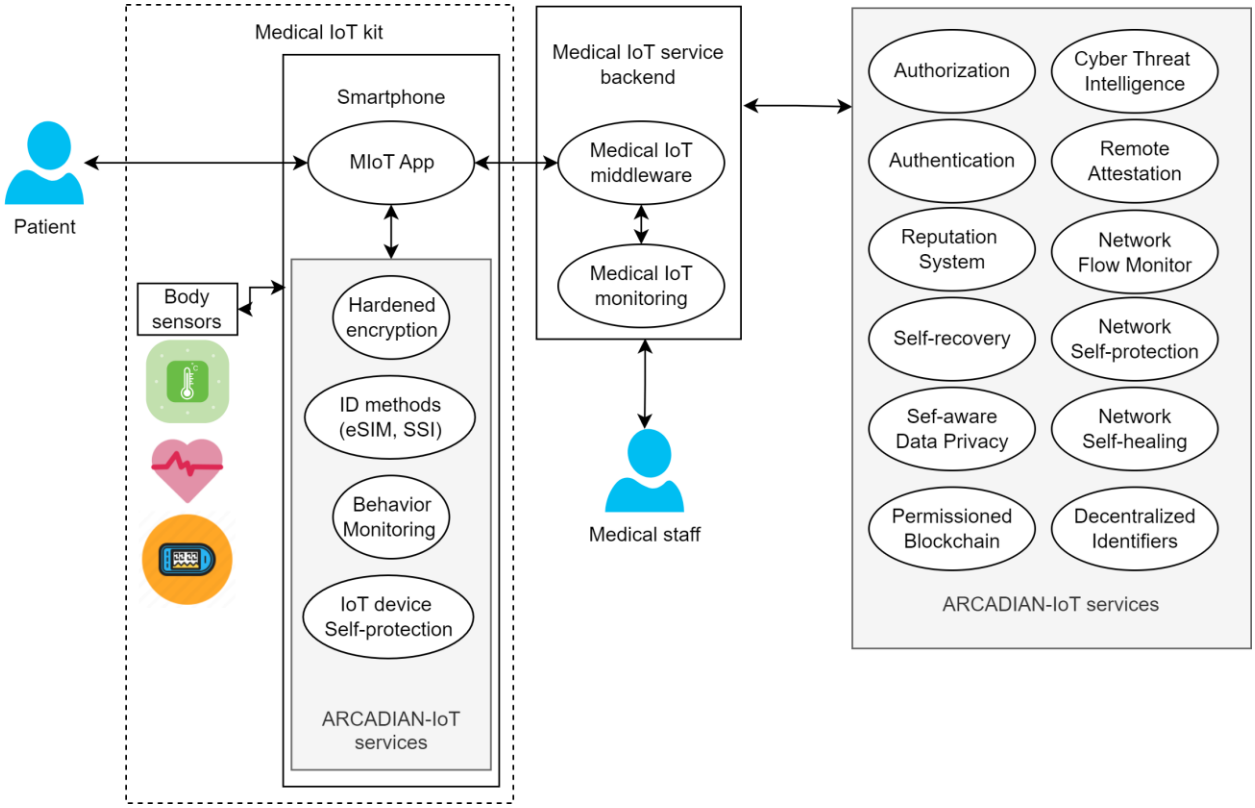

Figure 8 - High-level view of ARCADIAN-IoT involvement in Domain C

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

4.3.1.1 Overall Status

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. That part of the work is almost complete, being at a bug reviewing stage. Additionally, effort has been put in identifying and specifying necessary adaptations for integrating the Medical IoT service with ARCADIAN-IoT. In this scope, the next immediate step will be integrating ARCADIAN-IoT Multi-factor Authentication into the telemedicine system - the technical specification for this integration was already agreed between the involved partners.

4.3.1.2 Component Integration Readiness

Decentralized Identifiers (ATOS)

The following integration threads can be listed:

- Service Provider and Service Registration using did:web is planned for P1 but has not yet not taken place.
- Peer DIDs are supported and working in the end user mobile SSI Wallet.
- Public did:elem is working for the ARCADIAN-IoT Decentralized Identity integrated with the Ledger uSelf SSI Broker / Agent.

Verifiable Credentials (ATOS)

The following integration threads can be listed:

- Issuing of Person VC is supported.
- Integration for registration and authentication for persons with SP is as yet not started.
- Verifiable Credentials for constrained IoT Devices have been assessed and it has been concluded that they are not currently suitable.

Remote Attestation (IPN)

Remote Attestation procedures have been implemented and are available via one Attester instance and one Verifier instance running in different containers. In this use case, watchdog-based remote attestation will be supported, where remote attestation runs successfully (i.e., the claims / evidence are successfully appraised against the associated policies).

It is planned to do the necessary adaptations in the Attester for having it run properly in Android environment, allowing validation in RGB's targeted mobile device. Both service (e.g., appraising app fingerprint) and device attestation (appraising e.g., mobile device model ID or build version) are supported. Both service (appraising app fingerprint) and device attestation (appraising e.g., device model ID or build version) are scoped, but potentially only dummy values only be considered for P1.

Hardened Encryption (via eSIM) (XLAB)

The encryption library based on the attribute-based encryption paradigm is available to be used in Go, Java and Python. RoT signing has been tested with a Raspberry Pi, but still needs to be developed for phones which are the devices that are encrypting in this use case. A key management service is currently being prototyped.

Permissioned Blockchain (ATOS)

Not ready for integration at this time but will look to support Reputation System and Hardened Encryption during the first prototype piloting.

Reputation System and Policy Manager (UC)

The sharing of MIoT Kit delivery, the patient registration and authentication is planned for P1 but not integrated with the reputation system at this stage. In particular sharing this information through the message bus is required in order to determine initial reputation levels or devices and persons.

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

4.3.2.1 Overall Status

For supporting this use case, an Android application has been integrated with the telemonitoring web service. Such integration aims at enabling that the information collected by the telemedicine modules – as well as personal data - can be accessed as soon as possible by the doctor and nurses, for a prompt diagnosis. Moreover, interoperability tests have been done using similar APIs to those in ARCADIAN-IoT.

Towards this goal, secure data transmission, by sending encrypted events to hospital monitoring system, has been implemented.

The plan for the near future will be to use encryption systems (ARCADIAN-IoT's Hardened Encryption) to prevent this information from being sent unprotected.

4.3.2.2 Component Integration Readiness

Self-aware Data Privacy (MAR)

The component is consolidated, and the integration possibilities discussed but it is not yet implemented due to the maturity of the use case and the relationship with the existing technologies.

Hardened Encryption (via eSIM) (XLAB)

Similarly, as in Use case C2, HE component currently enables encryption and key management based on attribute-based encryption, while RoT signatures with eSIM need to be further developed to support Android phones.

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

4.3.3.1 Overall Status

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 pending to start.

4.3.3.2 Component Integration Readiness

Decentralized Identifiers (ATOS)

Peer DIDs are supported and working in the end user mobile SSI Wallet, while public did:elem is

working for the ARCADIAN-IoT Decentralized Identity integrated with the Ledger uSelf SSI Broker / Agent.

Service Provider and Service Registration using did:web is planned for P1 but has not yet taken place.

Verifiable Credentials (ATOS)

The issuing of Person VC is currently supported, while the integration for registration and authentication with SP is as yet not started.

Hardened Encryption (via eSIM) (XLAB)

Please refer to use case C2.

Permissioned Blockchain (ATOS)

Not ready for integration at this time but will look to support Reputation and Hardened Encryption during the first prototype piloting.

4.4 Validation considerations

The validation of the domain integration that is currently provided is outside the scope of this report. In M24 the consortium will release two deliverables that address the validation of ARCADIAN-IoT in the context of the domains and the associated use cases, namely D5.3 – "ARCADIAN-IoT use cases implementation" and D5.4 – "ARCADIAN-IoT use cases validation and legal compliance".

Deliverable 5.4 will present the implementation activities of the use cases. Domain owners (LOAD, BOX2M and RGB) will report, for instance, how the use cases have been implemented, the internal testing and validation that they were subject to, as well as the domain perspective on the interactions and activities related to the technical components.

On the other hand, Deliverable 5.4, will document the end-to-end validation of the use cases, considering not only the performance and functionality but also the legal aspects. The evaluation will consider the functionalities and integrations ready by M24 and its main support and validation ground is Prototype 1 (P1). From the time of writing of this report (i.e., M20) until M24, the consortium, will define the validation scenarios, bind the appropriate metrics and KPIs, to each scenario, prepared the necessary sequence diagrams to document the interactions between components, use cases and supporting tools. With respect to the legal validation, the consortium – guide by its legal partner (Elex) – will assess to what extent ARCADIAN-IoT framework and its use cases comply with existing cybersecurity regulations.

The outcomes and findings of the validation activities will serve as in input for Task 5.1 (responsible for the integration of ARCADIAN-IoT framework). Particularly, these outcomes will be required in order to optimize the performance and compliance of ARCADIAN-IoT framework in this final version (Prototype 2 (P2), which is due in M30.

5 CONCLUSIONS

This document reports the main integration outcomes and current status of the first prototype (P1) of ARCADIAN-IoT framework. These outcomes are part of Task 5.1 (Integration of ARCADIAN-IoT framework) and are associated to Milestone 3 (MS3), due on M20 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 P1.

In this document, 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 and the (2) component-to-domain support. It was possible to provide a description of the current ARCADIAN-IoT inter-component integration status, and the preliminary integration level with use cases targeted for P1 - as well as associated integration validation efforts. It was shown that the latter integration scope is still at a more preliminary stage, requiring additional effort in time for P1 validation.

The report on the use cases validation and legal compliance focusing on P1 functionalities is to be delivered in M24 (including the currently ongoing integrations planned for P1). The remaining ARCADIAN-IoT components' integration and domains' use cases artefacts will delivered in prototype 2 (P2), due in M30.

Overall, from a total of 21 technical components of the ARCADIAN-IoT framework, only credentials recovery (related to the SSI) has not yet been integrated in P1 due to delayed component specification (and consequent implementation). Nevertheless, the internal development and integration plans indicate an integration closer to the end of M21, thus it does not represent an issue. At the same time, 8 out of a total of 17 use cases were targeted for being supported in P1, with 12 of the 20 components in the process of having complete integration efforts for supporting the targeted use cases in time for P1.

The ongoing actions associated to the integration with ARCADIAN-IoT framework integration include the finalisation of the ongoing integration activities, the continuation of testing and validation activities across the integrated framework, and the integration of the interfaces envisioned for Prototype 2 (P2).

REFERENCES

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

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

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

[4] ARCADIAN-IoT, “D3.2: ARCADIAN-IoT Horizontal Planes - 2nd version,” 2022.

[5] ARCADIAN-IoT, “D4.2: ARCADIAN-IoT Vertical Planes - 2nd version,” 2022.

