



**Advanced Cyber-Threat Intelligence, Detection, and Mitigation
Platform for a Trusted Internet of Things
Grant Agreement: 786698**

D4.1 Architecture and design specifications: initial

Work Package 4: Cyber-threat framework, platform design and architecture

Document Dissemination Level

PU	Public	
CO	Confidential, only for members of the Consortium (including the Commission Services)	X

Document Due Date: 31/01/2019

Document Submission Date: 31/01/2019



Co-funded by the Horizon 2020 Framework Programme of the European Union



Document Information

Deliverable number:	D4.1
Deliverable title:	Architecture and design specifications
Deliverable version:	1.0
Work Package number:	WP4
Work Package title:	Cyber-threat framework, platform design and architecture
Due Date of delivery:	31/01/2019
Actual date of delivery:	31/01/2019
Dissemination level:	CO
Editor(s):	Raymond Binnendijk (CGI) Gohar Sargsyan (CGI)
Contributor(s):	Dimitrios Kavallieros, Vasiliki Georgia Bilali, Georgios Kokkinis (KEMEA) Nicholas Kolokotronis, Costas Vassilakis, Spiros Skiadopoulos, Christos Tryfonopoulos, Athanasios Chantzios, Christos-Minas Mathas (UOP) Stavros Shiaeles (CSCAN) Liza Charalambous, George Boulougaris, Michael Skitsas (ADITESS) Michele Simioli, Emanuele Bellini, Simone Naldini (MATHEMA) Pavué Clément (SCORECHAIN) Paul Quinn, Olga Gkotsopoulou (VUB)
Reviewer(s):	CSCAN KEMEA
Project name:	Advanced Cyber-Threat Intelligence, Detection, and Mitigation Platform for a Trusted Internet of Things
Project Acronym	Cyber-Trust
Project starting date:	01/05/2018
Project duration:	36 months
Rights:	Cyber-Trust Consortium

Version History

Version	Date	Beneficiary	Description
0.10	07-12-2018	CGI	Proposed outline
0.20	02-01-2018	CGI	Internal CGI review
0.30	11-01-2018	CGI	Technical Leaders preview
0.40	14-01-2018	CGI	Technical Partners review (Part-I) + input (Part-II)
0.70	18-01-2019	CGI	Technical Partners review (Part-I) + finalize input (Part-II)
0.90	28-01-2019	CGI	Final review
0.91	30-01-2019	CGI	Final review (REVISED)
0.92	31-01-2019	CGI	Final review (REVISED)
0.93	31-01-2019	CGI	Final review (REVISED)
1.0	31-01-2019	CGI, KEMEA	Final version for submission

Acronyms

ACRONYM	EXPLANATION
ACT	Attack Countermeasure Tree
ADT	Attack Defense Tree
AFT	Attack Fault Tree
AG	Attack graph
AIV	Annual Infrastructure Value
ALE	Annual Loss Expectancy
API	Application Programming Interface
ARC	Annual Response Cost
ART	Attack response Tree
AT	Attack tree
BAG	Bayesian Attack Graph
CAG	Core Attack Graph
CMS	Content Management System
CoAG	Conservative Attack Graph
CPE	Common Platform Enumeration
CSV	Comma-Separated Values
CUI	Character User Interface
CVE	Common Vulnerabilities and Exposures
CVRF	Common Vulnerability Reporting Format
CVSS	Common Vulnerability Scoring System
CWE	Common Weakness Enumeration
DAG	Directed Acyclic Graph
DDoS	Distributed Denial of Service
DNS	Domain Name Server
DLT	Distributed Ledger Technology
DT	Defense Tree
EDG	Exploit Dependency Graph
eVDB	enriched Vulnerability DataBase
GCF	Greenbone Community Feed
GPL	General Public License
GPO	Group Policy Object
GPRS	General Packet Radio Service
GPS	Global Positioning System
GrSM	Graphical Security Model
GSF	Greenbone Security Feed
HARM	Hierarchical Attack Representation Model
HTTP	Hypertext Transfer Protocol
HVAC	Heating, Ventilation, and Air Conditioning
IDPS	Intrusion Detection and Prevention System
IDS	Intrusion Detection System

IEC	International Electrotechnical Commission
iIRS	intelligent Intrusion Response System
IoT	Internet of Things
IPS	Intrusion Prevention System
ISO	International Standards Organization
LGA	Logical Attack Graph
NASL	Nessus Attack Scripting Language
NCCIC	National Cybersecurity and Communications Integration Center
NFC	Near Field Communication
NGFW	Next Generation FireWall
NIST	National Institute of Standards and Technology
NSE	Nmap Scripting Engine
NVD	National Vulnerability Database
OS	Operating System
OSINT	Open-Source INTelligence
OVAL	Open Vulnerability and Assessment Management
OWAT	Ordered Weighted Averaging Tree
PAG	Personalized Attack Graph
PT	Protection Tree
RDF	Resource Description Framework
RM	Risk Mitigation
SCAP	Security Content Automation Protocol
SCT	Security Compliance Toolkit
SDN	Software Defined Network
SNMP	Simple Network Management Protocol
SQL	Structured Query Language
TCP	Transmission Control Protocol
TMS	Trust Management Service
TVA	Topological Vulnerability Analysis
UDP	User Datagram Protocol
UI	User Interface
URI	Uniform Resource Identifier
URL	Uniform Resource Locator
VDB	Vulnerability DataBase
VM	Virtual Machine
VPN	Virtual Private Network
XML	eXtensible Markup Language

Table of Contents

Introduction.....	14
Project Overview.....	14
Purpose of the document.....	15
Relations to other activities in the project.....	15
Iterative approach.....	16
A mixture of research and development.....	16
Structure of the document.....	16
PART I – General Architecture	17
1. Methodology	18
1.1 Risk and Cost Drive Architecture (RCDA).....	18
1.2 Architectural views.....	21
1.3 Compliance & Security.....	22
2. Requirements	23
2.1 Requirements engineering.....	23
2.2 Requirements tracking and alignment between deliverables.....	23
2.3 Overview of key processes and components.....	24
2.4 Architectural requirements.....	27
2.5 Maturity requirement.....	28
3. Key design decisions and concerns	29
3.1 Key design decisions.....	29
3.2 Key open concerns.....	31
4. Operational view	33
4.1 Context.....	33
4.2 Key processes and components Conceptual Operational View (COV).....	33
4.3 Distributed Ledger Technology (A02, A15).....	38
4.4 Visualization Portal (A01).....	38
5. Delivery breakdown view.....	39
5.1 Solution Breakdown Structure (SBS).....	39
5.2 Development approach.....	41
5.2.1 Version Control System.....	41
5.2.2 Build Automation & Continuous Integration.....	41
5.2.3 Issue tracking.....	42
5.3 Integration strategy.....	42
5.3.1 Interface guidelines.....	42
5.3.2 Integration plan.....	43
5.4 Validation and testing.....	43

6. Infrastructure view	45
6.1 Component deployment	45
6.2 Integrated development environments.....	45
7. Security view	46
7.1 Product aim	46
7.2 Security area's	46
7.3 Privacy and Security by Design approach	47
7.4 Security mechanisms.....	48
7.5 Ongoing process.....	48
Part II - Tools high-level design specifications.....	49
8. Network repository (A16).....	50
8.1 Responsibilities.....	50
8.2 Key Functionality.....	50
8.3 Key Quality attributes	50
8.4 Open concerns	51
9. Visualization Portal (A01)	52
9.1 Responsibilities.....	52
9.2 Key Functionalities	52
9.3 Key Quality attributes	53
9.4 Open concerns	53
10. Profiling Service (A17)	54
10.1 Responsibilities.....	54
10.2 Key Functionality.....	54
10.3 Key Quality attributes	55
10.4 Open concerns	56
10.5 High-Level Design.....	56
11. Cyber-Defence Service (A04)	58
11.1 Responsibilities.....	58
11.2 Key Functionality.....	58
11.3 Key Quality attributes	59
11.4 Open concerns	59
12. Trust Management (A05 + A08)	60
12.1 Responsibilities.....	60
12.2 Key Functionality.....	60
12.3 Key Quality attributes	62
12.4 Open concerns	63
12.5 High Level Design	63

13. CT Registration Module (A06)	65
13.1 Responsibilities.....	65
13.2 Key Functionalities	65
13.3 Key Quality attributes	66
13.4 Open concerns	66
14. Enriched Vulnerability Database (eVDB) (A07 + A09)	67
14.1 Responsibilities.....	67
14.2 Key Functionality.....	67
14.3 Key Quality Attributes	68
14.4 Open concerns	69
15. Distributed Ledger Technology (A02, A15)	70
15.1 Responsibilities.....	70
15.2 Key Functionality.....	70
15.3 Key Quality attributes	70
15.4 Open concerns	71
15.5 High-Level Design.....	71
16. Crawling Service (A10)	72
16.1 Responsibilities.....	72
16.2 Key Functionality.....	72
16.3 Key Quality attributes	73
16.4 Open concerns	73
17. Smart Device Agent (A03m, A05m, A08m, A12, A14)	75
17.1 Responsibilities.....	75
17.2 Key Functionality.....	75
17.3 Key Quality attributes	76
17.4 Open concerns	76
18. Smart Gateway Agent (A03g, A04g, A05g, A08g, A11, A13)	77
18.1 Responsibilities.....	77
18.2 Key Functionality.....	77
18.3 Key Quality attributes	79
18.4 Open concerns	80
Conclusions	81
Annex: Architecturally significant requirements.....	82
Use-cases and scenarios	82
End-user requirements (functional)	88
End-user requirements (non-functional).....	90

List of Figures

Figure 0.1: Work Package relations	15
Figure 1.1: RCDA Architectural Micro cycle.....	19
Figure 1.2: RCDA core practices	20
Figure 1.3: Indicative high-level overview of RCDA practices applied within Cyber-Trust project process....	20
Figure 1.4: Architectural work is done throughout the project lifecycle	21
Figure 2.1: Work package interrelations (Athens meeting may-2018)	23
Figure 2.2: Threat intelligence processes and dataflow (Florence meeting Sept-2018).....	24
Figure 2.3: Threat intelligence actor interaction (Florence meeting okt-2018).....	25
Figure 2.4: Trust establishment and risk assessment actor interaction (Plymouth meeting Nov.-2018).....	25
Figure 2.5: Threat actors' attack strategies and application in the mitigation process. Source [D2.5].	26
Figure 2.6: TRL levels at EU H2020 RIA/IA projects	28
Figure 4.1: High-level solution overview. Source: [DoA].....	33
Figure 4.2: Vulnerability discovery and threat detection.....	35
Figure 4.3: Defense policy enforcement	37
Figure 4.4: Visualization Portal COV	38
Figure 5.1: Solution Breakdown Structure	40
Figure 10.1: A17 High-Level Design	56
Figure 12.1: A05+A08 High-Level Design.....	64
Figure 15.1: A02 + A15 High-Level Design.....	71

List of Tables

Table 0.1: Three pillars of Cyber-Trust	14
Table 1.1: Architectural views	21
Table 3.1: Key design decision.....	29
Table 3.2: Key design decision.....	29
Table 3.3: Key design decision.....	30
Table 3.4: Key design decision.....	30
Table 3.5: Key design decision.....	31
Table 3.6: Open concern	31
Table 3.7: Open concern	31
Table 3.8: Open concern	32
Table 3.9: Open concern	32
Table 3.10: Open concern	32
Table 4.1: Logical databases.....	34
Table 5.1: Validation and testing.....	44
Table 8.1: Key functionalities	50
Table 8.2: Quality Attributes	50
Table 9.1: Key functionalities	52
Table 9.2: Quality Attributes	53
Table 10.1: Key functionalities	54
Table 10.2: Quality Attributes	55
Table 11.1: Key functionalities	58
Table 11.2: Quality Attributes	59
Table 12.1: Key functionalities	60
Table 12.2: Quality Attributes	62
Table 13.1: Key functionalities	65
Table 13.2: Quality Attributes	66
Table 14.1: Key functionalities	67
Table 14.2: Quality Attributes	68
Table 15.1: Key functionalities	70
Table 15.2: Quality Attributes	70
Table 16.1: Key functionalities	72
Table 16.2: Quality Attributes	73
Table 17.1: Key functionalities	75
Table 17.2: Quality Attributes	76

Table 18.1: Key functionalities	77
Table 18.2: Quality Attributes	79
Table 0.1: Use cases - Architectural requirements.....	82
Table 0.2: End-user - Architectural requirements (functional)	88
Table 0.3: End-user - Architectural requirements (non-functional).....	90

Executive summary

Cyber-Trust Work Package 4 (WP4) is responsible for architecture. Through its recommendations and requirements from WP2 and WP3 are translated into technological, beyond the state of the art solutions, to be implemented in WP5, 6 and 7. This document is deliverable D4.1 within WP4: Cyber-Trust framework, platform design and architecture. The document contains the preliminary version of the architecture and the design specifications that have been selected to form the basis of the Cyber-Trust platform. Later in the project D4.4 will follow, containing the "definite" version of the architecture.

The architecture approach followed in WP4 is Risk- and Cost-Driven Architecture (RCDA) based on advantages versus other approaches that the consortium partners agreed upon at the proposal writing stage suitable for the Cyber-Trust platform. According to RCDA principles, the architecture work starts with identifying architectural concerns with the highest impact in terms of risk and cost, and addressing those concerns by making architectural decisions. Hence, this document D4.1 contains the results of the most impactful architectural decisions made, and concerns open still, in the Cyber-Trust project. These results have been documented in multiple views, where each view shows how the architecture addresses specific key stakeholder concerns.

The Cyber-Trust architecture process so far has taken place largely in parallel with the work on end user requirements and legal, ethical, security considerations in Work Packages WP2 and WP3. This has allowed the architecture and requirements processes to mutually benefit from each other's progress, and resulted in good cohesion between requirements and architecture. The price for this cohesion is some rework in maintaining traceability: D4.1 requirements traceability is based on an early analysis of the requirements in the Description of Action (DoA) section of the Cyber-Trust Grant Agreement, which in D4.4 will be extended to references to the output of WP2 and WP3.

The concerns with the highest impact in terms of risk and cost identified at the start of the project were especially integration, but also compliance and security.

Integration is a concern because the Cyber-Trust solution is composed of many separate components which are being developed by various development and research teams. This concern is addressed by shaping a modular architecture composed of various loosely coupled components where the interfaces between these components are shaped via integration guidelines. In addition, the architecture includes the approach chosen to develop or otherwise obtain the deliverable elements that make up the technical solution.

Compliance is an important concern, especially with respect to legal, ethical, social and privacy rules. This concern is mainly addressed in D2.3 - Cyber-Trust use case scenarios, D2.4 - Cyber-Trust end user requirements and especially D3.3 - Legal and ethical recommendations. WP2 and WP3 outputs are applied in indicates that, and explains how, compliance concerns vary based on the use cases, the tools to shaping the architecture and preparing this document.

Security is always a key concern in such complex platforms especially on designing and developing cyber-threat intelligence, detection, and mitigation platforms. Chapter 7 will address this concern, aligned with and complementary to D3.3 recommendations.

This deliverable consists of two main parts: Part I – General Architecture and Part II - Tools High Level Design Specification.

Part I – General Architecture

Chapter 1 provides the overview on the architecture methodology applied in Cyber-Trust, the Risk and Cost Driven Architecture methodology. The architecture views are introduced in Section 1.2 and the Compliance and Security in the Section 1.3.

The Requirements as depicted in Chapter 2. Followed by the requirements engineering and Requirements tracking and cohesion, in Section 2.3 Architectural requirements are identified which originate from use-case and scenarios, functional end-user requirements and non-functional end-user requirements. Solution maturity requirements is another important aspect Cyber-Trust platform and is explained in Section 2.4.

Chapter 3 introduces the key design decisions and concerns, which includes decision template, key design decisions and key open concerns. The operational view is presented in Chapter 4 which includes the context diagram, conceptual operational view , Distributed Ledger Technology and Visualization portal.

A delivery breakdown view is presented in Chapter 4, where the solution break down structure is introduced alongside with development approach, integration strategy and validation and testing. Infrastructure view is presented in Chapter 6 and the Security View in Chapter 7.

Part II – Tools High Level Design Specification

Each module of Cyber-Trust system is described in Part II following the template provided. The following sections are covered for each deployable module:

- Responsibilities
- Key Functionality
- Key Quality Attributes
- Open Concerns
- High Level Design (optional)

Introduction

Project Overview

Cyber-Trust | Advanced Cyber-Threat Intelligence, Detection, and Mitigation Platform for a Trusted Internet of Things is a 36-month long research project in the Digital Security Focus Area, co-funded by the Horizon 2020 Framework Programme of the European Union, under the Grant Agreement no. 786698. Its principal goal is to revolutionise the way cyber-security systems are built and operate.

By establishing an innovative cyber-threat intelligence gathering, detection, and mitigation platform, as well as, by performing high-quality interdisciplinary research in key areas, the Cyber-Trust project aims to develop novel technologies and concepts to tackle the grand challenges towards securing the ecosystem of IoT devices. It is structured around three pillars: a. key proactive technologies, b. cyber-attack detection and mitigation, and c. distributed ledger technologies, as seen in the table below.

Key proactive technologies	Attack detection and mitigation	Distributed Ledger Technologies
<ul style="list-style-type: none"> ▪ cyber-threat intelligence ▪ cyber-threat sharing ▪ reputation/trust management ▪ security games 	<ul style="list-style-type: none"> ▪ advanced targeted attacks ▪ network infrastructure attacks ▪ network visualisation ▪ mitigation and remediation ▪ forensics evidence collection 	<ul style="list-style-type: none"> ▪ registration ▪ update ▪ verification ▪ modelling ▪ consensus ▪ privacy

Table 0.1: Three pillars of Cyber-Trust

During Phase 1 The user requirements and regulatory framework have been set up to pave the way for the system design and architecture. During this phase, emerging trends in cyber-attacks have been identified to guide the definition of use case scenarios and the collection of the end-user requirements and the regulatory framework is being analysed and the impact of the proposed methods to fundamental rights, data protection and privacy is being assessed. The use cases have been identified. Phase 1 includes the work packages

- WP2. Cyber-threat landscape and end-user requirements;
- WP3. Legal issues: data protection and privacy.

Currently, the project is in Phase 2 - Platform design. In this phase, the Cyber-Trust platform reference architecture is created, incorporating inputs from the first phase, translated into technological tools to be built in Phase 3. The three tools above comprising the integrated platform are being designed and prototyped and the consortium is in the initial stage of the platform design. The design and architecture of the system is implemented under the work package

- WP4. Cyber-Trust framework, platform design and architecture.

The main outputs of this phase are the platform’s prototype, verifying the milestone Mi2 (Cyber-Trust rapid prototype) on month M12, and its final specifications at the end of the phase (month M16) which are associated with milestone Mi3 (Cyber-Trust architecture and design specifications). This document refers to the initial version of the system design and architecture. To ensure compliance and security privacy consideration, WP3 continues to be active in this phase to review and advice on the requirements.

Purpose of the document

This deliverable illustrates the initial architecture and the design specifications that are selected to constitute the basis of the Cyber-Trust platform. This deliverable is part of work package four (WP4, Cyber-Trust framework, platform design and architecture), Task 4.1 (T4.1): Setting up the framework: intake of recommendations and requirements. Led by CGI and contributed by partners, in this task the recommendations delivered from WP2 and WP3 have been translated into technical requirements which are covered by the Cyber-Trust platform. As part of the system design and architecture work, the recommendations and requirements are being made ready for consumption for setting up solution architecture. In this task, technical opportunities and constraints are being discussed and agreed. The outcome of the is the deliverable D4.1 – Architecture and Design Specifications – Initial; in terms of the initial draft of fine-tuned content to be compiled in following WP4 tasks.

Relations to other activities in the project

The inter-dependencies between Cyber-Trust project’s work packages are depicted in the following PERT diagram. It shows how work package WP4 is related and positioned within the project.

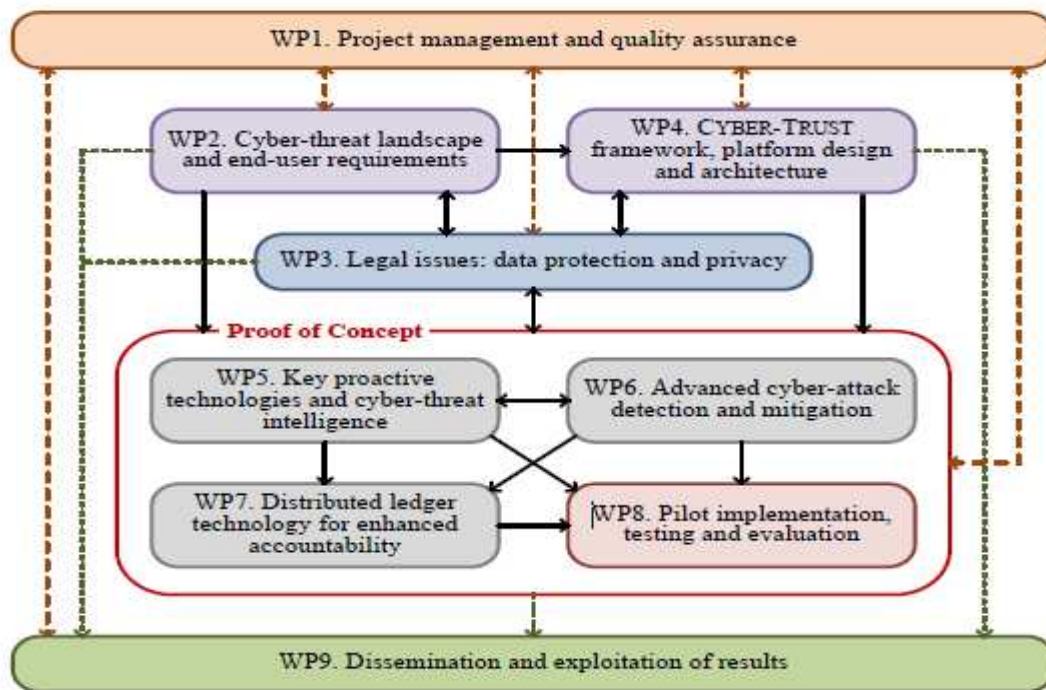


Figure 0.1: Work Package relations

This document, D4.1, is the first work package four deliverable and the primary outcome of T4.1 - Setting up the framework: intake of recommendations and requirements. D4.1 is intended as an initial architecture and design draft, aimed to be a base for and validated in tasks T4.3 - Implementation of architectural rapid prototype and T4.4 - Clickable UI mock-ups.

T4.3 will validate the architecture making a first draft version of the system using the architectural structures. This prototype aims to communicate, evaluate and validate the current architecture, as documented in D4.4. The deliverable D4.2 will be prepared as a result of this task. Lessons learned will be documented (also updating D4.4) and processed to the following technical WPs.

The aim of T4.4 task is to evaluate and validate the current state of front-end components and capture gaps for further development; moreover, it will validate UI of all front-end components via clickable UI-mock-ups. The prototype shall consist of clickable mockups of the user-interfaces and demonstrate future system

functionality. This task will deliver D4.3 (also updating D4.4 if needed). Lessons learned will be documented and processed to the following WPs for implementation and then WP8.

Iterative approach

Feedback gathered during T4.3 and T4.4 will be processed during T4.2 -Platform reference architecture and design specification. This task (T4.2) is responsible for the reference architecture and high-level system design specifications. The task will create a document describing the architecture and high-level design taking input from T4.1 so as to comply with WP2 and WP3 recommendations and requirements. Each technical partner will contribute to the design and will provide, explain and share technology that will serve as the base building blocks for implementing the CYBERTRUST platform during WP5, 6 or 7. Analysis of integration concerns will be performed. The architecture of the Cyber-Trust platform will be designed and documented. It will describe and explain all high impact design concerns and decisions to guide concrete tool implementation in the subsequent work packages. During T4.2, T4.3 and T4.4 D4.1 will be adjusted and elaborated on, resulting in D4.4, the final version of the Architecture and design specifications.

A mixture of research and development

Work packages WP 5 Key Proactive Technologies and cyber-threat intelligence, WP6 – Advanced cyber-attack detection and mitigation and WP7 – Distributed ledger technology for enhanced accountability follow (and partly go parallel) work package WP4 – Cyber-Trust framework, platform design and architecture activities and aim to implement the solution architecture (Proof of Concept). These implementation work packages are comprised of a mixture of research and development activities, where relevant state of art technology is identified, used and extended and new tools are custom developed. The research and technology partners will closely work together with clear identified roles and responsibilities to ensure efficient, high quality and smooth delivery of Cyber-Trust platform.

Structure of the document

The document is divided into two parts, Part I – General architecture and Part II – Tools design specifications.

Part I describes the solution architecture as a set of multiple views, based on CGI's RCDA approach. This includes a solution decomposition into multiple components

Part II describes each of this component in more detail. Covering the initial, high-level design aspects, aimed to jumpstart and smoothen WP5, 6 and 7 implementation activities.

For D4.1, being early in the project, Part-II contribution will be very high-level. Aspects like technology-stack, interfaces and high-level design views will be added later-on, in D4.4.

PART I – General Architecture

1. Methodology

The purpose of this chapter is to provide a brief description of the architecture methodology.

1.1 Risk and Cost Drive Architecture (RCDA)

As stated in DoA, the consortium has chosen Risk and Cost-Driven Architecture (RCDA) to be used as a method for architecture design. The advantage of applying this method is that it supports architectural decision making throughout the whole design process. Concerns and decisions are weighed throughout the design process and stakeholders' requirements are constantly validated against the design. The design process is iterative to ensure high-quality results. The fact that RCDA is a recognized method in the Open Group Certified Architect program¹, it is an extra advantage for the project and consortium partners to promote openness and collaboration on the most efficient way of shaping the design and architecture.

RCDA practices were applied while **initially shaping the Cyber-Trust project**.

The following concrete measures were applied:

- The architect is involved during the requirements WP2 to help and guide aiming at improving WP2->WP4 connection.
- Architecture (WP4) is delivered in two increments, with the ability to verify and learn:
 - The initial architecture (D4.1) is delivered early, to be able to align requirements engineering with software development and to have the opportunity to validate design decisions.
 - This architecture is validated by building and testing working software (D4.2 - Rapid Prototype)
 - The architecture is determined (D4.4) after processing the feedback gathered through D4.1, D4.2 and D4.3.
- Architecture focuses on critical design decisions and should not over-specify, and start early in the project, but the architectural work does not stop at D4.4. Technical Design and tools selection is performed later in the project (WP5, 6 and 7) and pilots in WP8. The architect is involved during these work packages where the architecture is validated and elaborated. The architect will help and guide but not lead.

¹ <http://www.opengroup.org/openca/cert/>

During the project, at the highest level of abstraction, the architectural specification process follows a simple **workflow** loop with three steps:

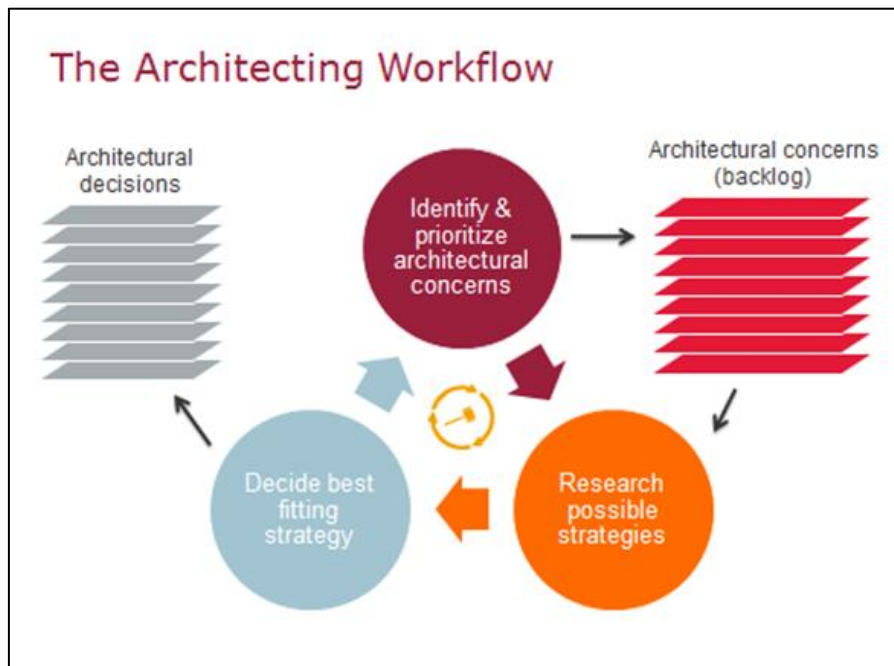


Figure 1.1: RCDA Architectural Micro cycle

We call this the "Architecture Micro cycle". This workflow loop is driven by a backlog of unresolved architectural concerns, resulting from the ARCHITECTURAL REQUIREMENTS PRIORITIZATION practice. The architectural decisions taken, resulting from the ARCHITECTURAL DECISION-MAKING practice, to address these concerns are added to an ever-growing stack of Architectural Decisions.

This micro cycle representation is a severe oversimplification. In real life, the architectural decisions usually affect more than one concern, and can hardly ever be made sequentially. The architect has to make sure that the total set of decisions maximally supports the total set of concerns.

In addition to the first two practices (prioritization and decision making) mentioned above, RCDA offers a set of core **practices** that are applied throughout the lifecycle of the project.

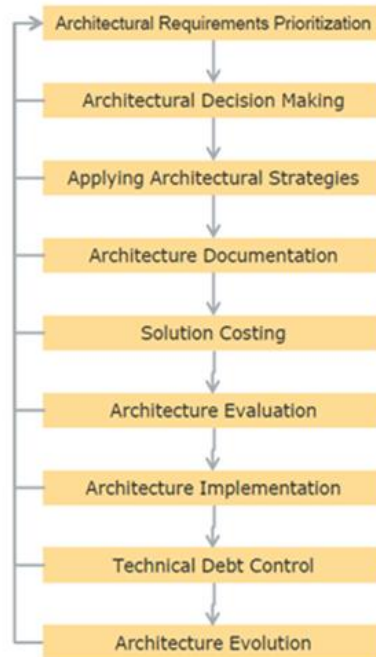


Figure 1.2: RCDA core practices

For more information about RCDA, see: <https://www.cgi.com/sites/default/files/white-papers/risk-and-cost-driven-architecture.pdf>

Figure 1-3 below shows how RCDA practices are applied within the Cyber-Trust project process. Practices are applied incrementally, continuously identifying and prioritizing concerns and making (and applying and documenting and validating etc.) decisions to mitigate these concerns.

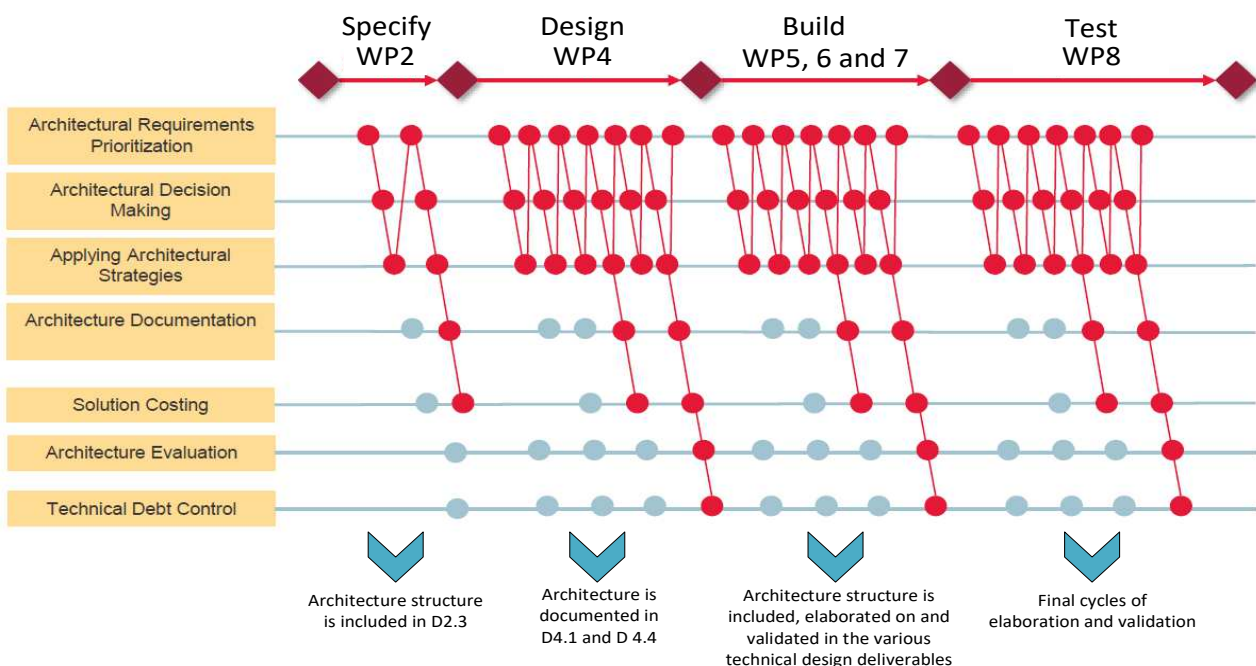


Figure 1.3: Indicative high-level overview of RCDA practices applied within Cyber-Trust project process

The Cyber-Trust project is mainly based on a traditional, phased, approach (waterfall). Although phases overlap and the architect is involved through the entire lifecycle, most of the work is performed in the design phase (WP4).

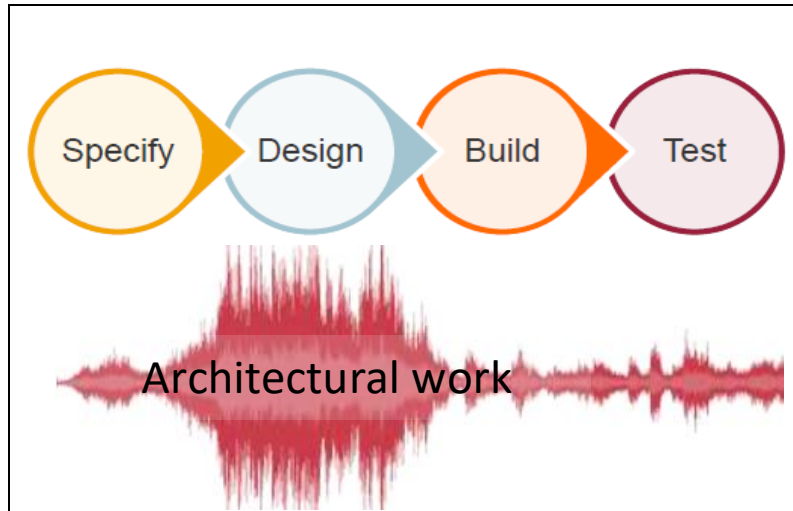


Figure 1.4: Architectural work is done throughout the project lifecycle

1.2 Architectural views

The Cyber-Trust solution architecture is shaped by the various architectural requirements and decisions made and documented in a set of views (see Table 1.1). These views focus on effectively communicating the architecture to the relevant stakeholders. Around and beyond these views, additional documentation is provided to complete technical systems design.

The views are detailed in subsequent chapters.

Table 1.1: Architectural views

Architecture views		
Chapter	View	Goal
2	Requirements	To identify, understand and prioritize architecturally significant requirements.
3	Decisions and concerns	To describe key decisions and open concerns.
4	Operational view	To describe how the system behaves in an operational environment.
5	Delivery breakdown view	To serve as a basis for planning solution delivery.
6	Infrastructure view	To identify and explain hardware, infrastructure software and deployment aspects of the solution.
7	Security view	To describes the set of processes, mechanisms and components used to make the system secure.

1.3 Compliance & Security

Compliance is an important concern, especially with respect to legal, ethical, social, privacy rules. This concern is mainly addressed in D2.3 - Cyber-Trust uses case scenarios, D2.4 - Cyber-Trust end-user requirements and especially D3.3 - Legal and ethical recommendations. WP2 and WP3 output indicate that, and explains how, compliance concerns vary based on the use cases, the tools to be developed within the Cyber-Trust project and the architecture will have to be flexible enough to address these variances.

Security is always a key concern in such complex platforms especially in designing and developing cyber-threat intelligence, detection, and mitigation platforms. Chapter 7 will address this concern, aligned with and complementary to D3.3 recommendations.

The approach that we will apply on designing the Cyber-Trust platform is the following: the requirements (end-user requirements and architectural requirements) should be legally and ethically compliant. A legal and ethical review will be provided by VUB iteratively to guarantee this compliance while designing and developing the system. Within WP4 we will focus on the architecture and design aspects, these are described in chapter 7, Security view. In WP5, 6 and 7 the security view guidelines and requirements compliance principles will be continued on a development level.

2. Requirements

This chapter will describe all requirements that the solution architecture will take into account.

Starting with explaining how these requirements are engineered and how alignment is achieved between the various WP deliverables.

After this an overview of the Cyber-Trust solution is given, followed by a more detailed summary of specific architecturally significant requirements.

Chapter 2 ends with a description of the required level of product maturity that has to be taken into account while realizing the architecture, into actual working software.

2.1 Requirements engineering

WP4 bridges the gap between requirements and technology were the recommendations delivered from WP2 and WP3 are translated into a solution architecture for the Cyber-Trust platform.

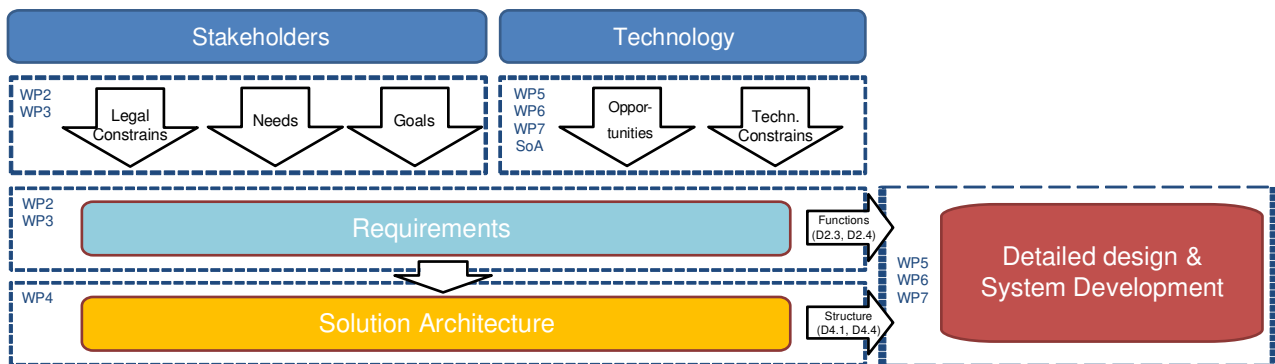


Figure 2.1: Work package interrelations (Athens meeting may-2018)

In practice, T4.1 activities and WP2 and 3 tasks overlap and therefore arrows between WP's will be bidirectional. Meaning architectural work started early by working together with WP2 and 3 partners in shaping, scoping, documenting and prioritizing requirements, making them ready for consumption for WPs 4-7. In doing so, technical opportunities and constraints are discussed, agreed and documented in deliverable D2.3. In general: Lessons learned in a specific WP can influences other WP's.

2.2 Requirements tracking and alignment between deliverables

Before going into requirements and architectural view requirements tracking must be explained. This is achieved by alignment of the structure and terms used in D2.3 and, early preliminary, D4.1 views:

- Solution Breakdown Structure (SBS, chapter 5.1), as a base for **component names** and cross-reference to D2.3 **actors** as well as to **partners** and DoA **tasks**.
- Conceptual Operational View (COV, chapter 4), for presenting SBS components in **runtime**, as well as D2.3 actors, **functions** and logical **databases**. The COV also adds **dataflow** requirements, deducted from D2.3.
- Key architectural requirements per component, are linked to **D2.3 use-case references** in the next section below.

At the end of this document an Annex has been included providing details regarding the link between the initial architectural requirements with the uses-cases (D2.3) and the end-user requirements (D2.4) respectively.

2.3 Overview of key processes and components

This section describes the key Cyber-Trust components, processes, data and interrelations. These based on slides that are presented and discussed during various technical project meetings and email conversations. These slides provide an overview of the key Cyber-Trust functionality, starting with figure 2-1 that describes how crawled data flows into the system.

For more functional details see: D2.3, use cases and scenario's.

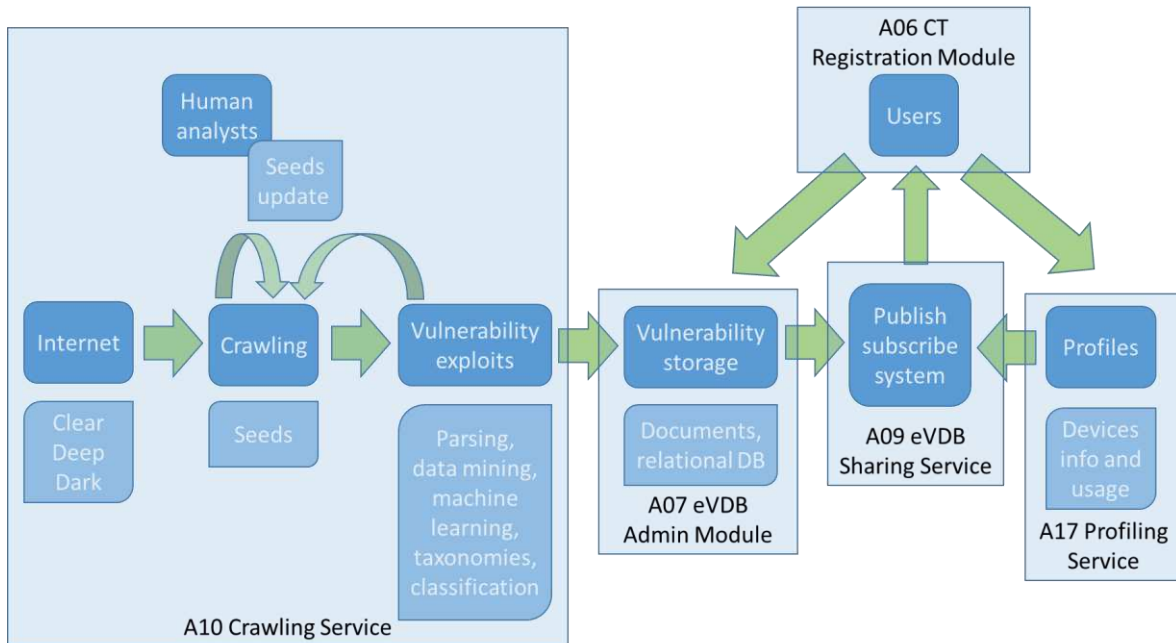


Figure 2.2: Threat intelligence processes and dataflow (Florence meeting Sept-2018)

The crawling service contains methods for harvesting the data available on the surface/deep/dark web and store it for further analysis, resulting in enriched data about vulnerabilities and exploits which is stored in the eVDB.

Users can register themselves, their organizations and devices via the CT registration module. The eVDB sharing service can automatically send relevant eVDB data to a registered user via a publish/subscribe mechanism.

In addition, figure 2-3 describes in more detail the various sub-processes, data categories and stakeholder roles involved. Again, behaviour and data are mapped to specific actors (A05, A07, etc.).

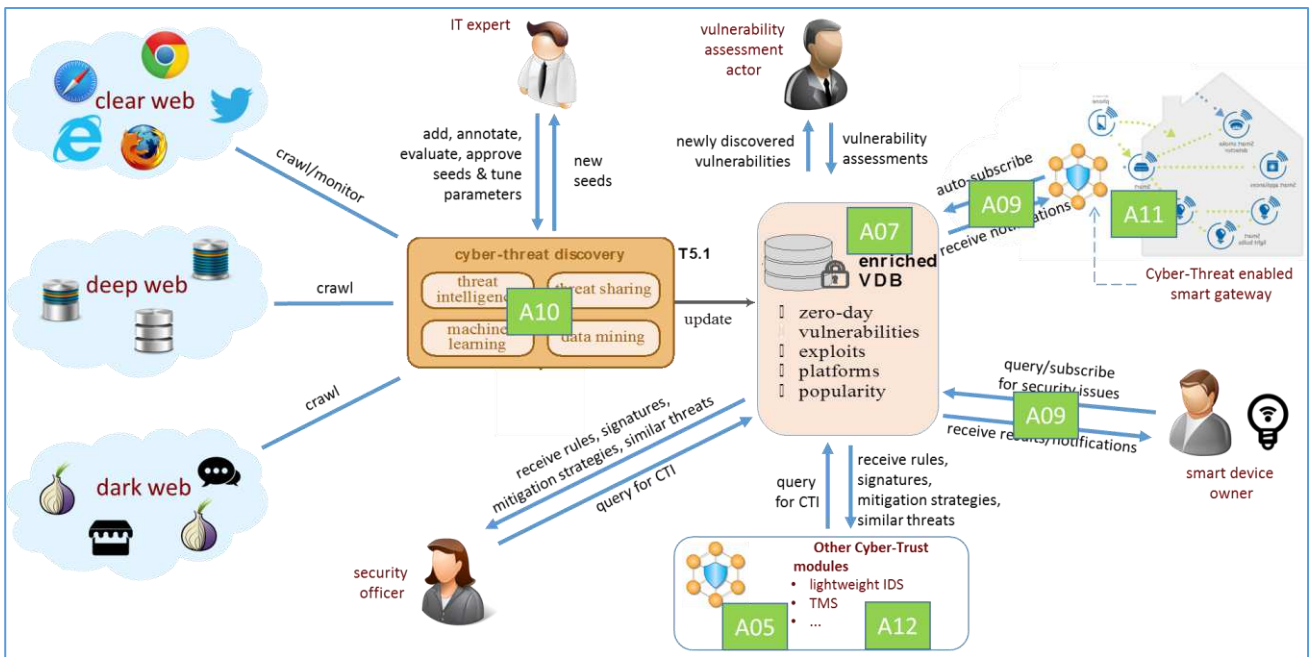


Figure 2.3: Threat intelligence actor interaction (Florence meeting okt-2018)

Figure 2-3 (again) describes how crawled data is stored, enriched using data mining and fed to the eVDB. From the eVDB this generic information about vulnerabilities and exploits can be used by various Cyber-Trust components. Specifically, by the Trust Management System (TMS, A05) and IDS (A12), who will be able to use this information to detect threats, as described further in figure 2-4.

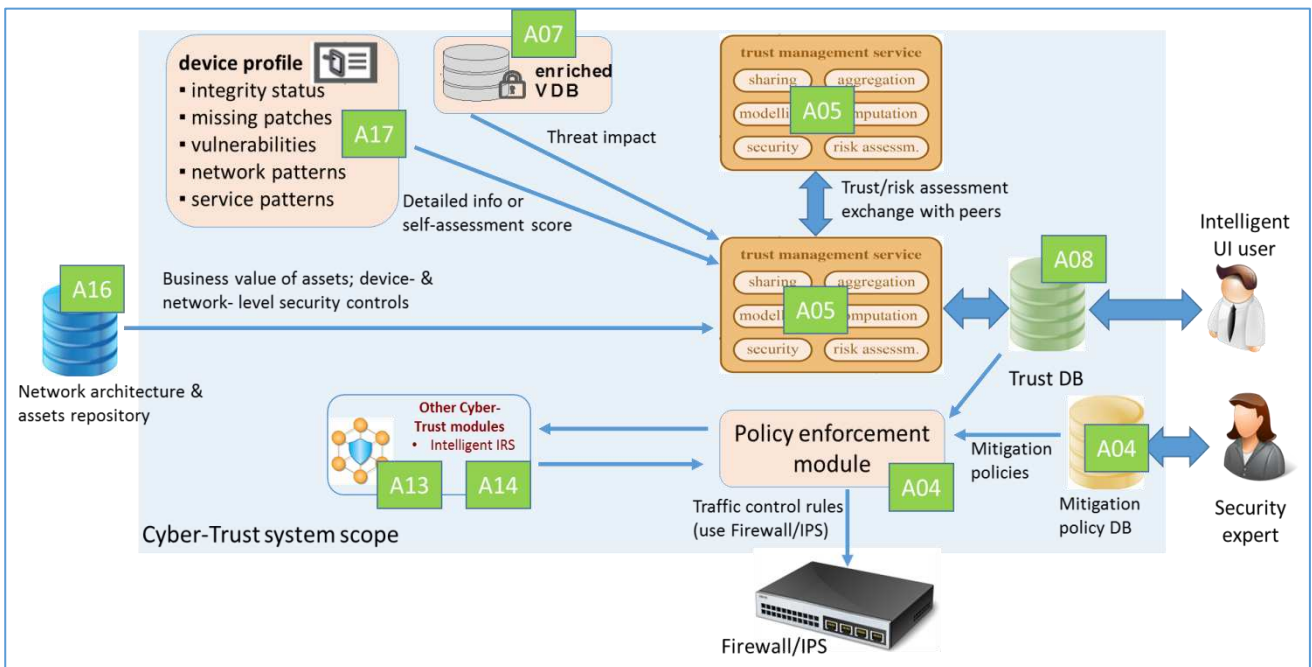


Figure 2.4: Trust establishment and risk assessment actor interaction (Plymouth meeting Nov.-2018)

The Trust Management System uses generic vulnerabilities and exploits to calculate specific device trust levels (stored in the TrustDB). Trust calculation also takes the device and network specific properties into account. Low trust scores trigger the Policy Enforcement Module (or Defense Service) to take action. Mitigation rules are applied directly or via components running on devices and/or gateway.

Figure 2-4 below- Threat actors' attack strategies and application in the mitigation process- originates from D2.5, and describes how the mitigation enforcement rules are established. Network Asset, Profiling and vulnerability and exploits information is combined as input for generation GrSM attack graphs. These attack graphs are fed to the Defense Service, from where mitigation enforcement rules will emerge to mitigate threats.

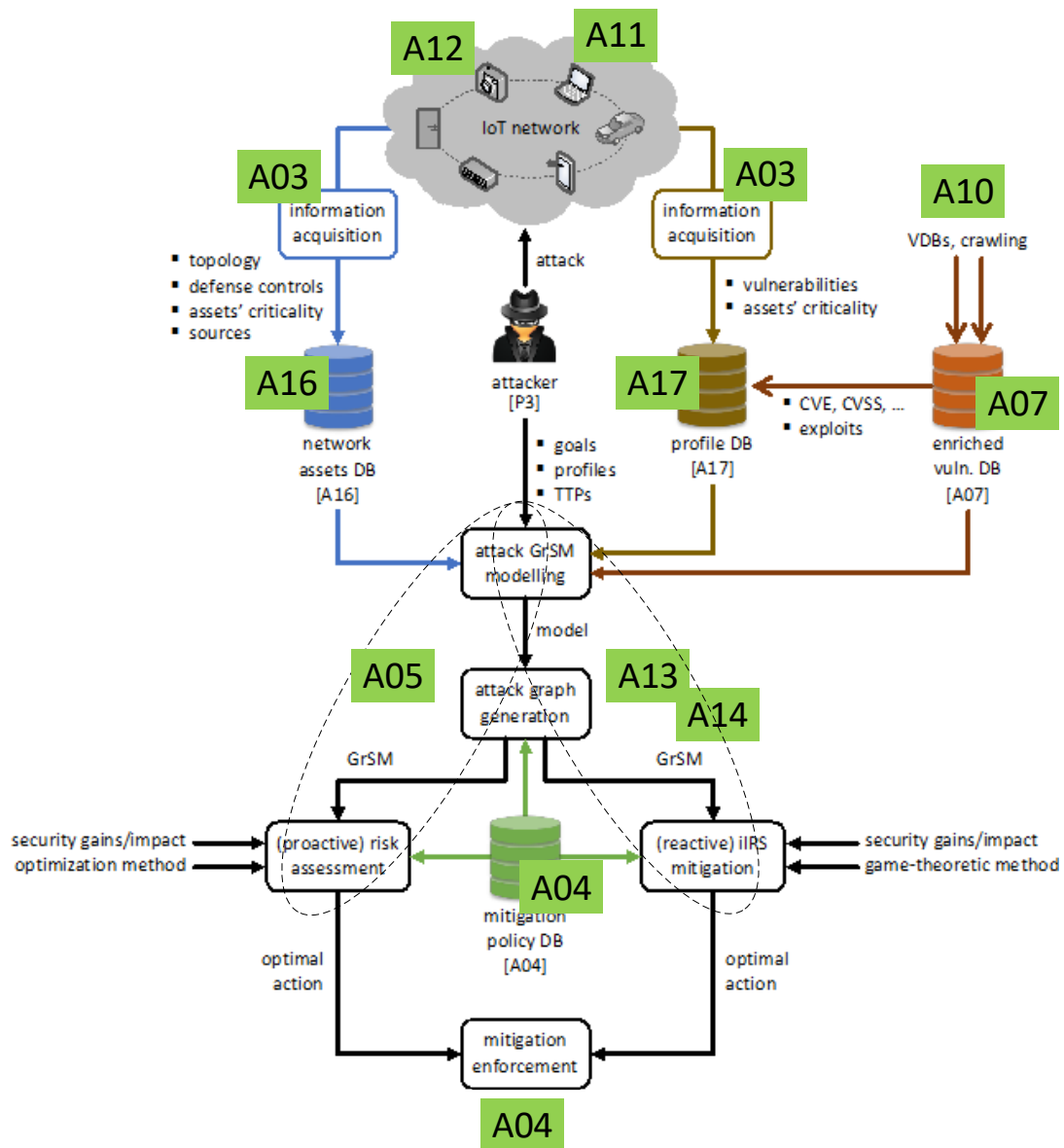


Figure 2.5: Threat actors' attack strategies and application in the mitigation process. Source [D2.5].

D2.5 explains how modelling of attackers and attack strategies with respect to potential cyber- attacks targeting at any part of the Cyber-Trust platform (devices, networks and CIIs). Such modelling will in turn allow for developing appropriate mitigation measures, being either proactive or reactive.

More precisely, a proper identification of attackers' profiles is essential in effectively addressing the security threats, as well as in appropriately responding to cyber-attacks. Constructing attackers' profiles rests with considering the attacker as an entity with varying (depending on the profile) constrained resources, like budget, tools, etc., aiming at exploiting vulnerabilities of any kind to maximize his profit

(access level, degrade QoS, etc.). Depending on the profile, some attack strategies will be more probable than others. Therefore, the attack strategies should be modelled in a systematic way to confront them. To this end, there are known tools to model the attack strategies – the most prominent being attack trees and attack graphs. These tools allow for presenting the possible paths that an attacker of any kind might follow (possibly in an adaptive manner) towards achieving his goals, whilst they also provide information on what needs to be done to alleviate security issues.

Utilizing appropriate tools to model attack strategies necessitates collection of appropriate information, including information on the network topology, on reachability amongst several nodes/devices (e.g. information on firewall rules), as well as on devices/software vulnerabilities (which in turn is contingent on system's elements configuration). All these pieces of information should feed the attack model, which will be developed in terms of appropriately estimating and combining the so-called preconditions that must be met with respect to exploiting specific vulnerabilities, as well as the so-called post conditions corresponding to the consequences occurred in case that some attackers' actions succeed. Moreover, such modeling tools for attack strategies allow for performing a risk analysis on the overall system, taking into account the relevant vulnerabilities and their corresponding probabilities of occurrence in conjunction with their impact. These systematic procedures allow for properly identifying and evaluating possible weaknesses, which in turn result in making proper decisions with regard to the security measures (mitigation steps) that need to be implemented.

The mitigation actions are computed in a proactive manner (by conducting risk assessment) and reactive manner (via the iIRS) will consider not only the attack graph's properties, but also the security gains and impacts that an action will have (in the long-term) on a network's security, operation, etc., and the predefined mitigation policies of organizations.

2.4 Architectural requirements

D2.3 describes Cyber-Trust **use-cases within scenarios in the context of various business domains (Smart Home and Mobile Devices)**, and thus explains a lot about data flows.

D2.4 identifies and describes **individual Cyber-Trust end-user requirements (functional and non-functional)**, derived from use-cases (D2.3), questionnaires and similar products and projects, **in more detail**.

Architecturally significant requirements, derived from D2.3 and D2.4, are consolidated and grouped and mapped to specific Cyber-Trust solution components (system actors). See: "Annex - Architecturally significant requirements" for all the details.

These architectural requirements, together with the design decisions in the next chapter 3, will shape the solution architecture as visualised in the architectural views that follow later in this document.

2.5 Maturity requirement

The maturity level of the Cyber-Trust platform will be measured via TRL (Technology Readiness Level). Technology readiness levels (TRL) are a method of estimating technology maturity of Critical Technology Elements (CTE) of a program during the acquisition process. They are determined during a Technology Readiness Assessment (TRA) that examines program concepts, technology requirements, and demonstrated technology capabilities. TRL is based on a scale from 1 to 9 with 9 being the most mature technology. The use of TRLs enables consistent, uniform discussions of technical maturity across different types of technology.

This general TRL maturity level approach is translated the following way for the EU H2020 research and innovation projects, which will be followed by the Cyber-Trust consortium partners.



Figure 2.6: TRL levels at EU H2020 RIA/IA projects

As stated in the Cyber-Trust project DoA under the sub-section "From lab to the market" (p. 145) the Cyber-Trust platform and the developed technologies will be validated in the relevant environments through two PoC (proof of concept) pilots (as mentioned above), therefore leading to technology readiness level (TRL) 5.

This however, does not limit the consortium partners to aim at higher ambitions in trying, if possible, to improve and advance the platform maturity level if the time and efforts allow. Also, architectural requirements aim to not limit, and support, future product maturity regardless the TRL levels. The Cyber-Trust solution will be validated in an intended environment while the architectural approach and methodology that apply in Cyber-Trust is validated in commercial environment. Therefore, the system is being set to be stable and resilient.

3. Key design decisions and concerns

As described in section 1.1, one of the principles of RCDA is to view architecture as a stream of design decisions. These decisions mitigate (prioritized) architectural concerns which emerge from architectural requirements (described in the previous chapter).

Most of chapter 2 architectural requirements can be dealt with directly in WP’s 5, 6 and 7. However, some additional design decisions are needed to (further) mitigate remaining concerns. These decisions are described in section 3.1 and need to be validated during the development of the Rapid Architectural Prototype (D4.2). Architectural views, described later in this document, are shaped by applying these architectural decisions (and also the architectural requirements).

Some key concerns are not yet mitigated at this time. Section 3.2 lists these open concerns. The Prototype (D4.2) development activities will be used to explore design strategies to mitigate these open concerns.

Key design decisions are documented using the RCDA light-weight architectural decision record template.

3.1 Key design decisions

During this initial architecture phase five high-impact concerns were mitigated by five key design decisions. These decisions are described below, including context, alternatives not chosen, the criteria that were applied and accepted drawbacks.

Table 3.1: Key design decision

ID: 01- Component development cadence	
Context	Concern
The Cyber-Trust solution is composed of many separate components which are being developed by various development and research teams	A lot of development dependencies, potentially being tied to big-bang deployments.
Decision	Alternatives not chosen
Design for autonomous components. Autonomy is more important than normalization. <ol style="list-style-type: none"> 1. Minimize centralized logic (see: 4.2 and 5.1) 2. Interfaces should be loosely coupled (see: 5.3) 3. Support flexible deployment cadences, provide decoupling through downwards interfaces compatibility (see: 5.3) 	Implementing decoupling façade/adaptor components.
Criteria	Drawbacks
<ol style="list-style-type: none"> 1. Flexibility 2. Modifiability 	Duplicate and decentralized logic. Peer2peer interfacing complexity.

Table 3.2: Key design decision

ID: 02- Modularity	
Context	Concern
The Cyber-Trust solution is composed of many separate components.	Future implementation problems due to the lack of integration flexibly
Decision	Alternatives not chosen
Design for modular components. <ol style="list-style-type: none"> 1. Strict separation of concerns (see: 4.2 and 	

<p>5.1)</p> <ol style="list-style-type: none"> Interfaces should be loosely coupled (see: 5.3) Component updates should be decoupled (see: 5.3) 	
Criteria	Drawbacks
<ol style="list-style-type: none"> The solution can run without all components to be operational The solution is able to run with client specific plugins. 	Suboptimal performance.

Table 3.3: Key design decision

ID: 03-Integration	
Context	Concern
The Cyber-Trust solution is composed of many separate components, based on various technology stacks.	Integration complexity.
Decision	Alternatives not chosen
<ol style="list-style-type: none"> Integrate early and often (see: 5.2) Use a central development environment (see: 5.2) Interfaces should be loosely coupled, and asynchronous (see: 5.3) Component updates should be decoupled (see: 5.3) 	Detailed interfacing specification up-front.
Criteria	Drawbacks
<ol style="list-style-type: none"> Flexibility Risk mitigation (validation by working-software). 	Technical partners need to be active and cooperating early in the project.

Table 3.4: Key design decision

ID: 04- Performance	
Context	Concern
The Cyber-Trust solution is composed of many separate components. Data exchange is crucial. Data-sets can be of various sizes.	Potential data exchange performance bottlenecks.
Decision	Alternatives not chosen
<p>Regulated peer2peer communication. Apply asynchronous communication patterns as defined in chapter 5.3.</p> <p>Combine multiple logical components into one deployable component (minimizes the number of inter-component peer2peer interfaces).</p>	<p>ESB. Shared databases. Fully custom peer2peer interfaces.</p>
Criteria	Drawbacks
<ol style="list-style-type: none"> Flexibility Loosely-coupled Optimize for both small and large messages Cloud-native. 	Peer2peer communication complexity, no central component to provide an overview.

Table 3.5: Key design decision

ID: 05- Messaging	
Context	Concern
The Cyber-Trust solution is composed of many separate components. Data exchange is crucial. Very important to "get the interfaces right".	To resist the urge to decide on interfacing design too early in the project. Get-the-interfaces-right just in time.
Decision	Alternatives not chosen
Draft high-level integration strategy and include interface guidelines in D4.1 (see 5.3). Based on loosely coupled REST-based peer2peer communication pattern. Use Rapid Prototype to validate and elaborate. Finalize in D4.4.	Central ESB. No design up-front. Big design up-front.
Criteria	Drawbacks
Just-enough-architecture (just-in-time). Do not over specify the architecture so technology experts have room to provide the optimum decision.	Solution-shape has not been decided on.

3.2 Key open concerns

During this initial architecture phase there are five high-impact concerns still open, to be mitigated later in the project. These open concerns are described below, including their context.

Table 3.6: Open concern

ID: 06- Pilot environment	
Context	Concern
The Cyber-Trust pilot will run in a controlled environment (WP8).	The challenge to run the pilot in an environment that represents (near) real life.
Decision	Alternatives not chosen
Test environment basics in are covered chapter 6. These basics need to be further elaborated on during Rapid Prototype development. This includes finding a partner to provide the pilot hosting environment.	
Criteria	Drawbacks

Table 3.7: Open concern

ID: 07- Security	
Context	Concern
Being a cybersecurity project, we should take fundamental security measures into account. Chapter 2 defines various security and privacy related architectural requirements: <ul style="list-style-type: none"> • Authentication (SSO, tokens) • Authorization (role-based) • Message security (encryption, hashing) • Transport channel security (TLS) • Data storage security (encryption, hashing, DLT) 	How to implement the security guidelines described in chapter 7 and security-related requirements in chapter 2 in more detail.
Decision	Alternatives not chosen
This needs to be further elaborated on during Rapid Prototype development.	
Criteria	Drawbacks

Table 3.8: Open concern

ID: 08- Requirements coverage & feasibility	
Context	Concern
A lot is written about the Cyber-Trust solution, especially in DoA and WP2 deliverables. Priority differentiation in D2.4 is limited (mostly M).	How to cover all requirements. Is everything technically feasible? Do we have enough development time and capacity?
Decision	Alternatives not chosen
Will be decide how to embed prioritization into way-of-working during WP's 5, 6 and 7.	
Criteria	Drawbacks

Table 3.9: Open concern

ID: 09- Research vs development	
Context	Concern
The implementation work packages are comprised of a mixture of research and development activities, were relevant state of art technology is identified, used and extended and new tools are custom developed.	Balance and corporation between research and development. When to stop research and push for delivery of TRL 5 mature software.
Decision	Alternatives not chosen
Will be decided how to embed this balance into way-of-working during WP's 5, 6 and 7. Use Rapid Prototype to explore.	
Criteria	Drawbacks

Table 3.10: Open concern

ID: 10- UI channels	
Context	Concern
Cyber-Trust functionality is provided via a smart-device-app, smart-gateway-app and multiple web portals.	Sort out the separation of functions. How to divide the required functions over the various UI channels.
Decision	Alternatives not chosen
Tis needs to be determined during Mock-ups (D4.3).	
Criteria	Drawbacks

4. Operational view

4.1 Context

The Cyber-Trust project is built upon three main cyber-security research thrusts, that is key proactive technologies, cyber-attack detection and mitigation, and distributed ledger technologies. The proposed approach aims to capture different phases of a large-scale cyber-attack before and after existing (and possibly unknown) vulnerabilities of devices have been widely exploited by cyber-criminals to launch the attack. Some novel methods and tools will be developed to deal with the fundamental problems of prevention, detection, and mitigation of advanced cyber attacks involving IoT devices and networks.

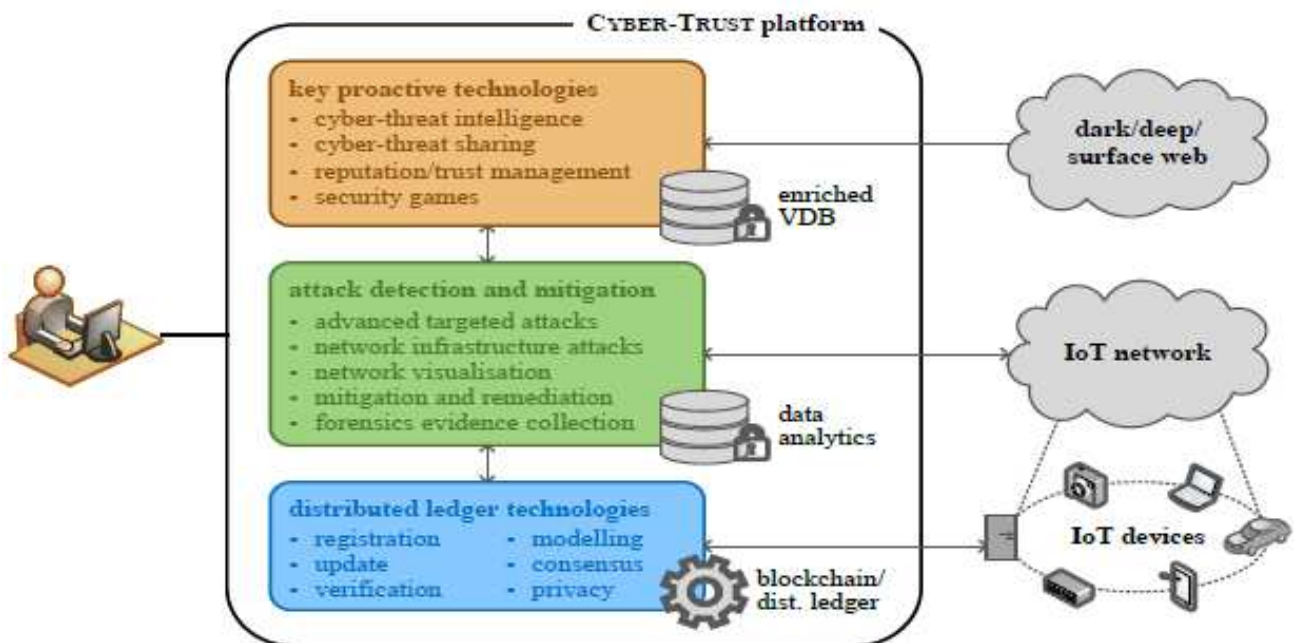


Figure 4.1: High-level solution overview. Source: [DoA]

A high-level overview of dark/deep/surface-web crawling, as well as threat detection and mitigation are described in chapter 2.3 – Overview of key processes and components.

The sections below will describe these processes, and related components, in more detail.

4.2 Key processes and components Conceptual Operational View (COV)

This operational decomposition shows the solution's structure during operation and illustrates the relationship of elements during run-time. The major focus here is on functional decomposition, separation of concerns, dependencies and especially data flow.

This view is totally aligned with the SBS (Solution Breakdown Structure, see chapter 5.1), which will be described later in this document.

Detailed responsibilities and high-level design aspects of each component are described in Part-II of this document. Some of the functional modules will be physically merged. See SBS.

The COV will contain all Cyber-Trust (logical) databases, as described in D2.3, as summarized in the table below. A data view will be elaborated on later in this project and added to the final architecture (D4.2).

Table 4.1: Logical databases

Cyber-Trust databases	
Actor	Database
A02	Distributed Ledger Data (on-chain, see section 4.3)
A04	Mitigation Policy Database
A04	Forensic Evidence Database
A07	Enriched Vulnerabilities Database (eVDB)
A08	Trust Database (TrustDB)
A10	Vulnerabilities Database (raw data)
A16	Network Architecture and Asset Repository
A17	ProfileRepository (Device and usage)

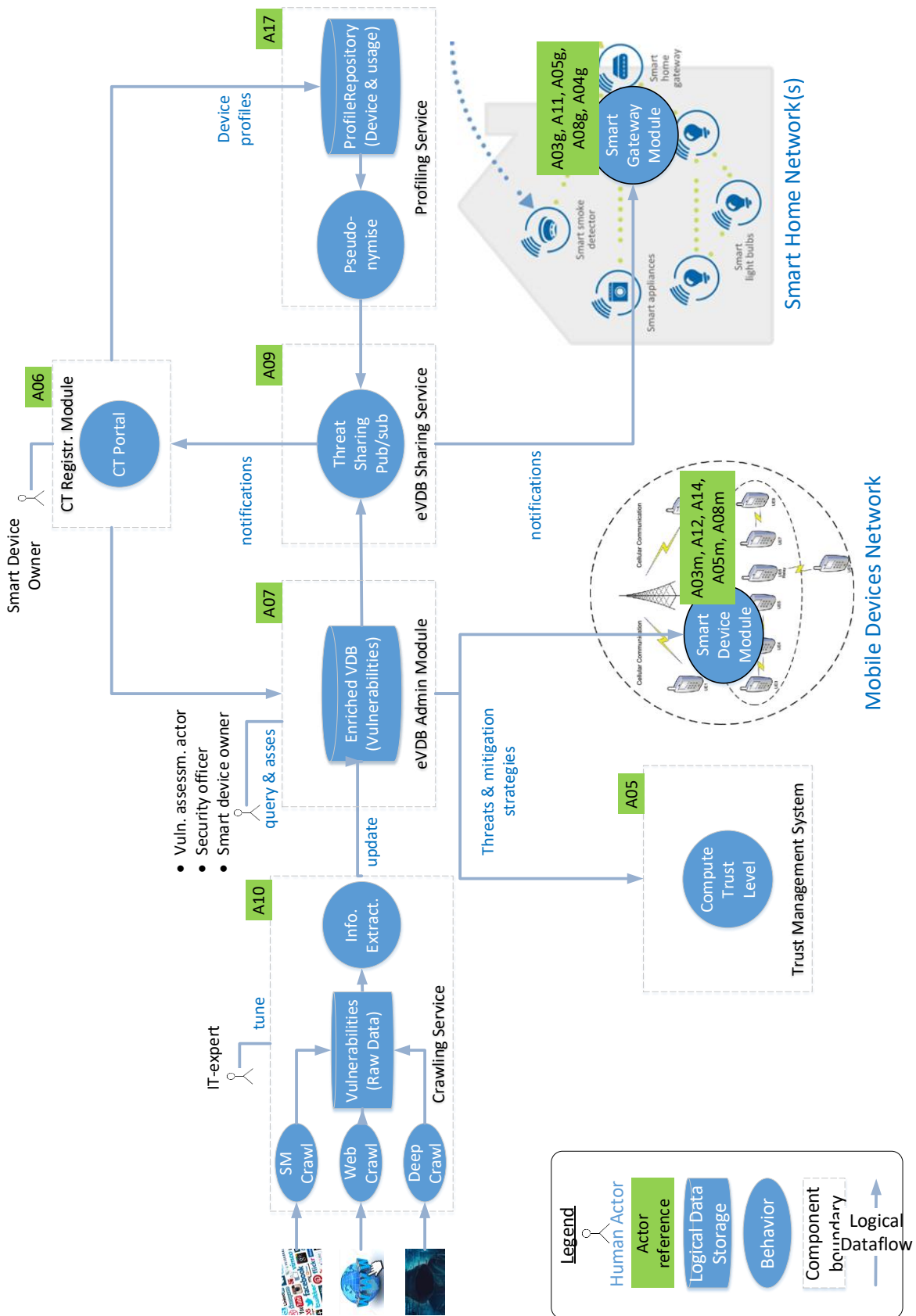


Figure 4.2: Vulnerability discovery and threat detection

Above: Further detailing figure 2-1 and 2-2, and identifying, separating and grouping the various processes, databases and data flow; Raw data is crawled and vulnerabilities are extracted. Generic vulnerability information is combined with specific Mobile and Smart-Home Network information to compute risk and trust levels.

Cyber-Trust users register and are centrally authenticated via the registration module.

Below: Further detailing figure 2-3 and 2-4; the TMS can trigger Defense services. Defenses services enforce policies at the device and/or network level to mitigate and/or remediate attacks.

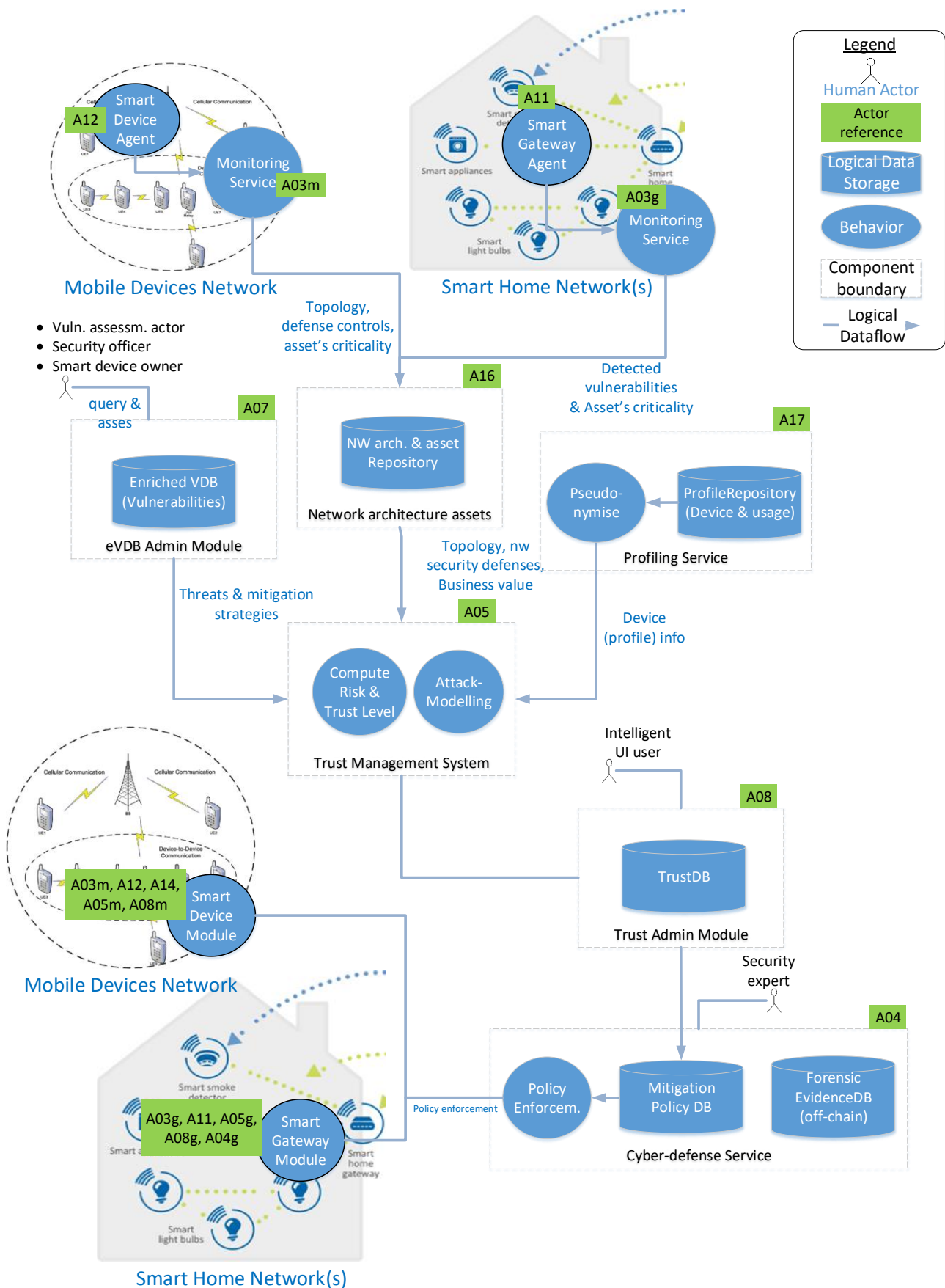


Figure 4.3: Defense policy enforcement

4.3 Distributed Ledger Technology (A02, A15)

Distributed Ledger Technology is part of the Cyber-Trust platform in order to provide secure and scalable storage and sharing capabilities via an enhanced blockchain ledger.

Distributed Ledger Technology is provided by A02 (DLT Service) and administrated via A15 (DLT Admin Module). A02 is used for on-chain storage to enable decentralized **secure storage and sharing** of:

- User and device registration data (Related to A06 and A07 off-chain)
- Forensic Evidence (Related to A04 off-chain)
- Critical Files (Related to A16 and A17 off-chain)

4.4 Visualization Portal (A01)

The Visualization Portal (A01) provides various network monitoring and health status views (2D and 3D) and functions. The data and functions are provided by A02, A03, A04, A05, A06, and A17.

A01 is also provides decision support and alerting, aimed to acknowledge the existence of vulnerabilities, using the eVDB (A07), as the primary source of data.

Data can be securely exported to file.

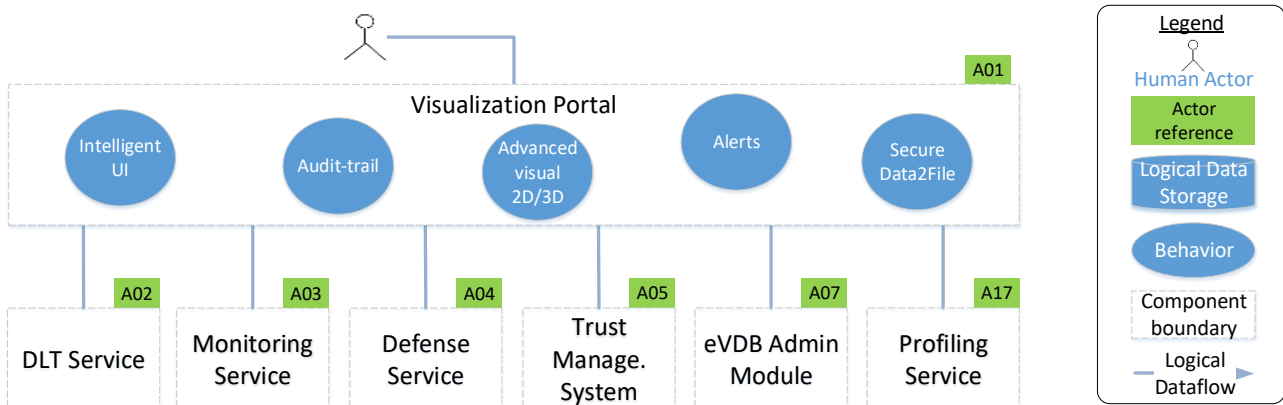


Figure 4.4: Visualization Portal COV

During the development of the Clickable Mockups (D4.3), A01 data, functions and look-and-feel will be elaborated on in more detail.

Also, separation-of-concerns, in relation to other Cyber-Trust UI-components will be discussed and decided on.

Note: more components will need an UI (e.g.: A06, A08). To-be-determined if these will be included into A01 (during development of the mockups, T4.4).

5. Delivery breakdown view

This chapter contains all the architectural information that will serve as a basis for planning the **integration and delivery** of the Cyber-Trust solution.

5.1 Solution Breakdown Structure (SBS)

In the Solution Breakdown Structure, the solution is broken down into deliverable elements. The Solution Breakdown Structure (SBS) is the basis for further component level design.

See PART-II for more detailed information per component. The PART-II chapter names and chapter order are aligned with the SBS.

The SBS is a tree showing how the solution decomposes into deliverable elements and which technical partner is responsible for delivery (WP5, 6 and 7) of every element.

Some logically decoupled modules will be physically combined as described in the legend.

A03, A04, A05 and A08 are split into multiple deployable units.

The SBS is linked to DoA tasks and D2.3 actors to provide clarity, traceability and alignment.

- 1) Out-of-project implementation scope. Well know available tool will be selected and used.
- 2) This components (and subs) will mainly implement UI functionality
- 3) These subcomponents will be combined into one platform service deployment.
- 4) These subcomponents will be combined into one device app deployment
- 5) The components will be combined into one gateway app deployment

Legend

- Logical solution component
- Actor reference (D2.3)
- Task reference (DoA)
- Technology Partner reference

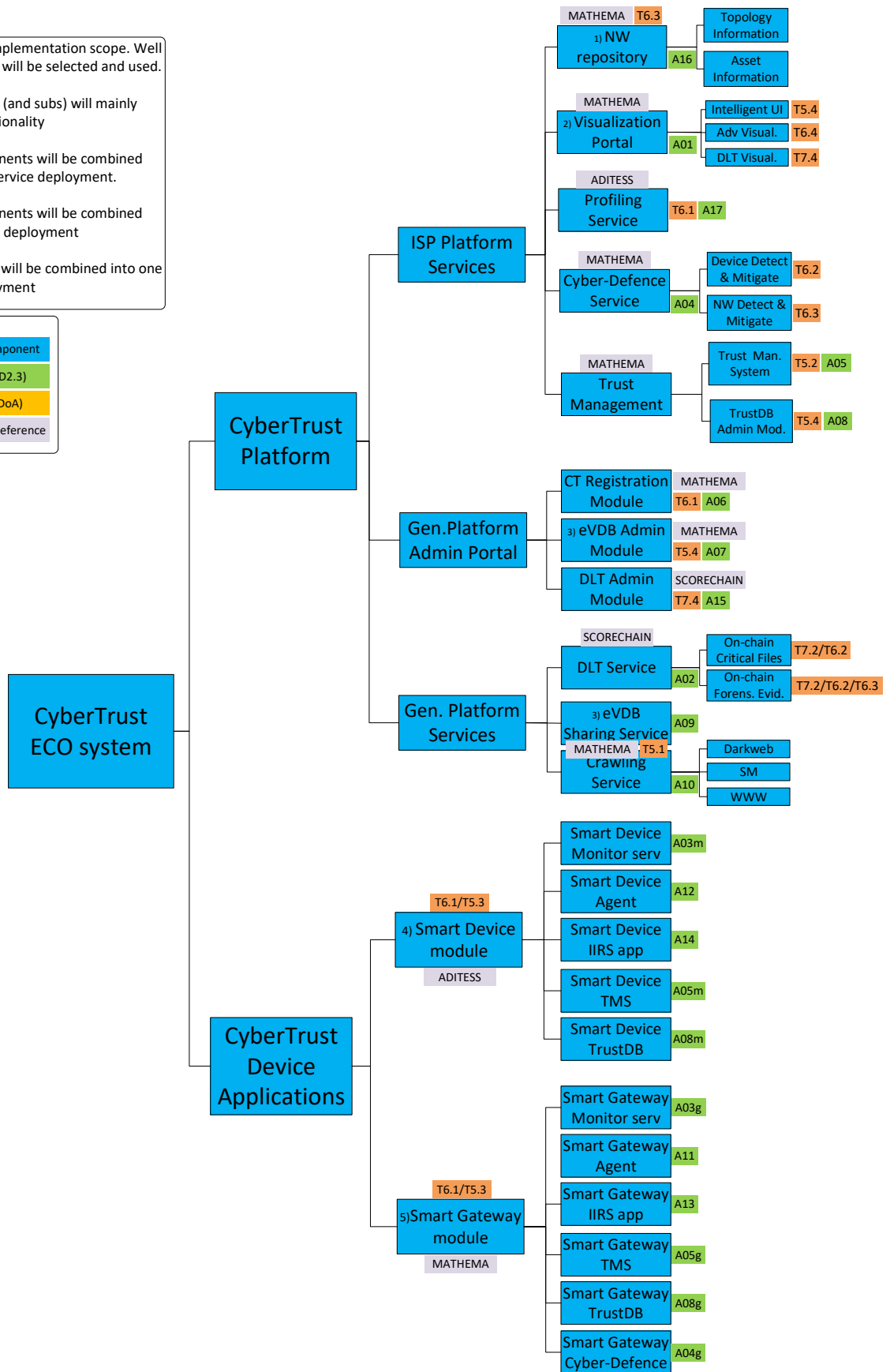


Figure 5.1: Solution Breakdown Structure

5.2 Development approach

This chapter describes the approach chosen to develop or otherwise obtain the deliverable elements that make up the technical solution. This approach will be used by **all technical partners** while carrying out work within the framework of WPs 5, 6 & 7. The development of the integrated Cyber-Trust Platform components will be **a continuous process** which will contain all the required **steps** to assure quality during the entire lifetime of the project.

This process can be represented as a virtual circle that contains the following functional components:

1. Source-code-versioning and management,
2. Continuous integration,
3. Issue tracking

The primary tool that will be used to support the development cycle is **Gitlab** and **GitHub** for the release of stable versions to the wider community.

Gitlab will be used by partners as not just as a revision control system but also as an environment where technical management and progress evaluation takes place. The source code of each component will be maintained as a separate project including clear documentation for tool setup and user documentation. Additionally, Gitlab issues will be used to keep track of improvements and bugs for each project.

Finally, a separate project will be used to include all architectural and development views, along with information regarding architectural concerns development milestones and a detailed breakdown of the expected outcome and functionalities from each component/service. The remaining of this section includes further information with regards to each of the three components in our development cycle.

5.2.1 Version Control System

A Version Control System (also known as a Revision Control System) is a repository of files, often the files for the source code of computer programs, with monitored access. Every change made is tracked, along with who made the change, why they made it, and references to problems fixed, or enhancements introduced. Version control systems are essential for any form of distributed, collaborative development. Whether it is the history of a wiki page or large software development project, the ability to track each change as it was made, and to reverse changes when necessary can make all the difference between a well-managed and controlled process and an uncontrolled 'first come, first served' system.

In the context of Cyber-Trust, integration is a major concern and great priority is assigned to speed and flexibility. Due to the nature of the project, branches should be created, managed and merged constantly. This happens whenever a new feature is introduced, a bug is fixed or enhancement is applied. As such, branch management should be inexpensive in terms of time and space consumption. Developers are not centrally located, but dispersed throughout many countries. In addition, code decoupling to a central repository is vital. Git fits all these requirements and is used as the primary version control system.

5.2.2 Build Automation & Continuous Integration

The version control system is a crucial part of the Cyber-Trust development lifecycle. However, one additional crucial development aspect that contributes significantly to the quality of the development is built automation. Build automation is the act of scripting or automating a wide variety of tasks that software developers do in their day-to-day activities including:

1. compiling computer source code into binary code;
2. packaging binary code;
3. running tests;
4. deployment to production systems;
5. creating documentation and/or release notes.

Continuous Integration (CI) is a software development practice where the members of a team frequently integrate their work – usually each contributor integrates his software code at least daily, leading to frequent integrations. Each integration cycle is verified by an automated build (including automated tests) to detect integration errors as quickly as possible. Many teams find that this approach leads to significantly reduced integration problems and allows a team to develop cohesive software more rapidly. CI is achieved through the configuration and usage of a CI server.

The benefits of Continuous Integration are the risk reduction and facility in bug fixing. The decision on whether or not build automation will be adopted in Cyber-Trust is not yet finalized as decisions on behalf of the partners with regards to developing technologies and individual approaches are still pending.

5.2.3 Issue tracking

The last step of the development lifecycle is the issue/bug tracking. The internal Gitlab tracking system will be used as the **only source** to register and manage issues (bugs, features, questions, work assignments, etc.). The issue tracker is accessible for every developing and tester partner.

5.3 Integration strategy

Integration of all individual components is a major concern we have to address. That is why it is **crucial to get the interfaces right** (or: focus on what's going on between the boxes) and to follow a plan focused on smooth integrating of all software components.

This section describes the interfaces guidelines and an integration plan to be followed in order to address this concern.

5.3.1 Interface guidelines

Each technical partner should design and build its interfaces following a number of guidelines, or explicitly explain why a guideline is violated.

These guidelines are:

1. Asynchronous chunky (not chatty) dialogues between components to promote loosely coupling and reduce overall bandwidth use.
2. Use REST². Aim to reach at least maturity level 2³.
3. Interfaces should abstract the underlying service complexities. Naming is semantic (not just CRUD) and thereby easy to understand by humans.

² https://en.wikipedia.org/wiki/Representational_state_transfer

³ <http://martinfowler.com/articles/richardsonMaturityModel.html>

4. Transport big chunks of data using the claim-check-pattern⁴. However, no integration over shared databases.
5. Design for interfacing resilience:
 - Use timeouts and retries in order to address (too) long wait times.
 - Validate incoming or received data. Fail fast.
 - Be able to deal with disconnected/unavailable peers to make your component more autonomous.
 - Use the circuit-breaker-pattern⁵ to protect against cascading failures and to limit the number of incoming requests.
6. Use decentralised over centralised orchestration for managing business process communication logic.
7. Avoid big bang breaking of interfaces by potentially supporting multiple (two at the minimal) coexisting interface versions.
8. Adapt Continuous Integration, described in earlier in this document, as an important part of the “Development Approach”, also a crucial practice in securing successful integration.
9. Consider security and privacy practices (see chapter 7) while implementing: authentication, authorisation, message security, transport channel security to all interfaces.

5.3.2 Integration plan

Integration is a continuous development process. Technical partners should deliver software to the ITE fast and often using the tools and approach described in the previous sections.

The ITE will host multiple integrated solution components, working together as one solution. Applying continuous integration practices and loosely coupled interfaces (as described in earlier in this document) is crucial here.

The ITE solution will build incrementally following the various project deliverables and milestones, starting with building the rapid prototype (D4.2) and finishing by providing the solution for running the pilots (WP8).

5.4 Validation and testing

To deal with the levels of complexity of the Cyber-Trust solution, components will experience the following sets of tests:

⁴ <https://www.enterpriseintegrationpatterns.com/patterns/messaging/StoreInLibrary.html>

⁵ <https://dzone.com/articles/circuit-breaker-pattern>

Table 5.1: Validation and testing

Types of tests		
Test category	Description	Responsible
Unit Test	Isolated technology faced tests, executed (preferably automatically) in the tech partner development environment.	Tech partner, developers
Component Test	Single component/service test. Stubs are used to simulate external components. Executed in the technical partner test environment.	Tech partner, system testers
Prototype Test	Implementation of a rapid architectural prototype (T4.3). Based on D4.1 architectural requirements, a rapid prototype of the system will be implemented and tested by the developers in the developers ITE. This prototype aims to validate and detail the major architectural decisions made in D4.1 and to identify new architectural concerns or opportunities to be included in the final architectural description (D4.4).	CGI (WP4 leader)
UI mock-ups Test	Clickable UI mock-ups of the Cyber-Trust platform (T4.4). Based on the analysis of the functional and non-functional requirements sourcing from WP2 and D4.1, UI mockups will be made. These mockups aim to evaluate and to capture gaps between expectation and current status. The mockups will be built for all components that include an user-interface and demonstrate how the information is handled by the back office platform. The mock-ups will be demonstrated at a workshop.	CGI (WP4 leader)
System Integration and overall Technical Testing	During WP5, 6 and 7 software development tasks, components are build following the guidelines and requirements originated from WP4. These components are integrated and tested, early and continuously. Deviations from the D4.1 and D4.4 are explicitly documented as design decisions and discussed with the WP4 leader. Details and deviations from D2.3 and D2.4 are documented and discussed with WP2 leader.	CSCAN (WP6 leader)
Pilots	WP8 aims to test and validate the Cyber-Trust platform. WP's 5,6 and 7 working software component deliverables are integrated and tested in a (near) real-life environment.	KEMEA (WP8 leader)

6. Infrastructure view

This chapter identifies and explains the major hardware, infrastructure software and some deployment aspects of the Cyber-Trust solution, although only high-level aspects are known at this time in the project.

6.1 Component deployment

The SBS describes the “Cyber-Trust ECO system” into various levels of abstraction. Deployment grouping is done on the third level, as described below:

- “ISP Platform Services” are envisioned to be hosted within a **ISP datacenter**.
- The “Generic Platform Administration Portal” will be implemented as a **web portal**, and hosted within the **central Generic Cyber-Trust cloud environment**.
- The “Generic Platform Services” will also be hosted within the **central Generic Cyber-Trust cloud environment**.
- “Smart Gateway Modules” will run as an app on the smart home **gateways**.
- “Smart Device Modules” will run as an app on the end-user’s **smart phone**.

6.2 Integrated development environments

WP’s 5, 6 and 7 partners will deploy and integrate their software early and often within an **integrated test environment**, using the tools described in chapter 5.2.

Each partner will **host their own components**, preferably based on Docker containers. The solution will be **integrated by connecting** the various environments. Individual environment setup and inter environment integration will be initially done during the development of the prototype (D4.2) and elaborated on in the final architectural document (D4.4).

7. Security view

7.1 Product aim

Being a cyber security project, we must take fundamental security measures into account. From an architectural perspective it is key to shape an architecture that addresses fundamental security aspects now and can address more and more detailed security aspects in the (near) future.

7.2 Security area's

This section provides a brief overview of the security areas to be considered, in addition to the D3.3 considerations, within the various Cyber-Trust components. These areas are based on the top vulnerabilities, as identified by the Open Web Application Security Project (OWASP⁶).

1. **Injection and Cross-Site scripting;** attack on a parser or interpreter that relies on user-supplied input (e.g.: SQL injection or XXE injection).
See OWASP website for prevention cheat sheet defensive measures.
2. **Broken Authentication and Session Management;** attackers can hijack or illegally obtain a session by intercepting or guessing the session ID.
See OWASP guidelines for safe handling of sessions ID's.
3. **Broken Access Control;** access data an attacker should not have access to.
Make sure objects check authorization in the first place. Also, don't let the interface (e.g., URL) expose information like database or session ids. This is applicable not only to components that are connected to the outside world (deep security).
4. **Security Misconfiguration;** e.g., no or default security configuration settings, like usernames and passwords.
Use personal, non-group, non-standard usernames and passwords. Add audit-trail logging.
5. **Sensitive Data Exposure;** attackers obtain sensitive data.
Identify which data is to be considered "sensitive". Only use/store the data you really need. Use transport security, encrypted storage and authorization mechanisms to protect these data.
6. **Cross-Site Request Forgery;** obtain access to your services via a (poorly protected) external service that already has access.
Use modern web framework, because they probably provide CSRF protection out-of-the-box. Also, see the OWASP CSRF prevention cheat sheet.
7. **Using components with known vulnerabilities;** must attacks are not based on zero-day vulnerabilities. Therefor make sure to keep your component versions up2date.
8. **Under protected APIs;** APIs are meant to be used by other programs. This makes them extra vulnerable for automatic attacks.
Use secure transport channels (TLS) for communication with the outside world. Make sure information parsers (be it JSON, XML, etc.) to be hardened and able to detect malicious input

⁶ www.owasp.org

7.3 Privacy and Security by Design approach

In addition to the considerations described above we will embed a Privacy and Security by Design approach. This is done by incorporating a consolidated set of preconditions when designing and implementing the system:

1. Privacy and Security as the Default Setting:

- **Least Privilege** - The principle that security architecture should be designed so that each entity is granted the minimum system resources and authorizations that the entity needs to perform its function.
- **Need-To-Know** - A method of isolating information resources based on a user's need to have access to that resource in order to perform their job but no more.
- **Least Trust** - The principle that security architecture should be designed in a way that minimizes 1) the number of components that require trust, and 2) the extent to which each component is trusted.
- **Mandatory Access Control** - A means of restricting access to objects based on the sensitivity of the information contained in the objects and the formal authorization of subjects to access sensitive information.
- **Segregation of Duties** - Separating certain areas of responsibility and duties in an effort to reduce fraud and unintentional mistakes.

2. Privacy and Security Embedded by Design:

- **Full Lifecycle Approach** - Security must be addressed throughout the full development of a software product from requirements and design to implementation, testing and deployment. Security must be considered at and engineered into, every step of a product's lifecycle.
- **Comprehensive Threat Analysis** - The sensitivity of the data used by a product, the system processes that handle them, and the potential repercussions from the loss, misuse or unauthorized access of any data must be assessed and prioritized.
- **Security Built Into the System Architecture** - Security measures to address any potential threats must be designed into the architecture of the system.
- **Regular Code Review** - Exploitable flaws in the source code must be discovered through repeated code reviews and audits and fixed through recoding and/or redesigning of the system.
- **Rigorous Security Testing** - The secure functionality of the system must be assured through structured testing and methods-based evaluation of the software features being delivered.

3. **Full Functionality — Positive-Sum, not Zero-Sum:** privacy and data security should not fully compromise the functionality of the proposed system. In contrast, they should enhance the functionality by giving the user more control and transparency over the data managed. Cavoukian and Dixon (2013) argue that privacy and security need not to diminish the functionality of proposed IT systems. When properly understood and implemented, privacy may work in conjunction with the IT and enhances its functionality insofar as it increases end-user satisfaction, consumer confidence, trust and use.

4. **End-to-End Security — Full Lifecycle Protection:** Two key areas of information security, Database Security and Identity and Access Management (IAM), are vital to end-to-end security of privacy. The following controls may be considered:

- **Preventative Security Controls** - prevent illegitimate actions from data in the database.
- **Detective Security Controls** - monitor and analyses cases of illegitimate actions that do happen in the database.

- **Identity Governance function** - ensure that the right people get access rights and the wrong people do not, provide knowledge of who has access to what, disable access rights when people leave, and enforce audit policies (ensure compliance).
- **Directory Services repository** - definitive, unified source for who has access and WHAT access they have.
- **Access Management mechanisms** – know who the user is (authenticate), grant the right access (authorize) and enforce security policies (Web, mobile, cloud).

7.4 Security mechanisms

Chapter 2 and 7 define various security and privacy related architectural requirements. These will shape mechanisms for:

- Authentication (SSO, tokens)
- Authorization (role-based)
- Message security (encryption, hashing)
- Transport channel security (TLS)
- Data storage security (encryption, hashing, DLT)

Note: At this time these mechanisms are not further designed in detailed. This will be done in the coming months. Using the Rapid Prototype to validate and elaborate.

7.5 Ongoing process

Securing software is an ongoing and custom process. In the various work packages that follow we should take security into account, working together and leverage the extensive cyber security expertise we have to our disposal in this project.

For this deliverable D4.1, being the initial draft of the Cyber-Trust architecture, the security aspect addressed above will be validated and elaborated on during the creating of the Cyber-Trust prototype D4.2. More in detail security architecture will be included in the final architecture, D4.4.

Part II - Tools high-level design specifications

Part II describes each of the various Cyber-Trust components in more detail. Covering the initial, high-level design aspects, aimed to jumpstart and smoothen WP5, 6 and 7 implementation activities.

For D4.1, being early in the project, Part-II contribution will be very high-level. Aspects like technology-stack, interfaces and detailed design views will be added later-on, in D4.4.

Part-II contains multiple chapters, each chapter describes one deployable component, which can be composed of one or more logical components, as described in the SBS. Component composition, naming and order is fully aligned to the SBS. Below the explanation on each section of the template.

Responsibilities

Each component is assigned to responsibilities that need to be documented to have a clear overview and understanding about the system. This part of the template highlights the main component responsibilities.

Key Functionality

A selection on the key functionality of each component has been made by partners collaborating for its development based on the functionality's priority and significance to Cyber-Trust architecture. These functionalities are strictly aligned with D2.3 – Use cases, therefore, the co-relation between each key functionality is presented towards specific use case ID from D2.3. This ongoing work will eventually cover all the functionalities presented in D2.3 chapter 6.2, which will be thoroughly reviewed, analysed, prioritised, adjusted if needed and eventually implemented. D4.4 will contain the complete and exclusive list of all the functionalities with associated use cases, which will be implemented.

Key Quality Attributes

It is important to identify the key quality attributes for each component early in the project. Here each partner, responsible for a specific component, introduces these key attributes for each component to ensure quality control and focus from the start to the delivery stage.

Open Concerns

To ensure transparency and quality and fast problem solving the partners collaborating for the development of each component, highlighted the open concerns they have at this point. It is natural to have such concerns which are being worked out and solved every single day. Thus, the project consortium partners intensively work to shape the design and the architecture of the Cyber-Trust system. While this document reflects the work performed so far D4.4 will contain section reflecting the follow-up of this section and solutions provided.

High Level Design (optional)

At this stage, the components have different levels of maturity in terms of the design. This part of the template is created to help partners collaborating in the development of each component to visualize it and present high-level design if possible.

8. Network repository (A16)

8.1 Responsibilities

This is a set of tools allowing to get information on a network’s architecture (including the topology and the security defences deployed therein), assets and their values, etc.

8.2 Key Functionality

Table 8.1: Key functionalities

Key Functionality		
Description	Category	Ref
<p>The network architecture and assets repository have the following functionalities within the ISP platform:</p> <ul style="list-style-type: none"> - Host-based vulnerability scanning: monitoring each end user device involves the correlation of information gathered in the eVDB [A07] with vulnerabilities and device characteristics gathered at device level. Information such a communication protocol, open ports, running services, installed firmware etc constitute correlation parameters for the detection of possible vulnerabilities specific to each device. - Monitor device critical OS files / vulnerabilities: The network architecture monitors critical OS files/directories [D5]. It also performs a device scanning using for example (NMAP tool) for open ports and running processes. Information is synced with the central backend database. In case any attempt is made in modifying the state of the device, backend services are triggered to check the status of vulnerability. - Monitor activity on the device: monitoring of communication and data transactions on the monitored device. It involves the logging of key device communication. 	Monitoring and scanning	DoA T6.3 UCG-07-02 UCG-09-01 UCG-09-02
<ul style="list-style-type: none"> - Gather device forensic evidence: The procedure of gathering evidence especially in the IoT environment differs based on the device, it is storage capabilities and software. A collection and storage of forensic evidence is part of the functionality of the network architecture and assets repository (e.g. device log files, timestamps etc.) from the Cyber-Trust registered devices [A06]. - Gather network forensic evidence: The process (automatic) and conditions (e.g. with the identification of an attack) under which the Cyber-Trust will start collecting relevant network data [A03] in order to be used as digital forensic evidence in the court of law as well as the collection mechanisms/techniques (e.g. DPI) - Curate forensic evidence database: The forensic evidence are stored in the Forensic Evidences eVDB (off-chain) [A07] while the hash values of these data, time stamps regarding the data, information regarding the owner of the data etc. will be stored in the DLT (on-chain) [A02]. Thus, this UC will show how the forensic data stored in the evidence DB will be annotated, organised and presented in the blockchain (e.g. hashes of the actual evidence, chain of custody etc.). 	Forensic Evidence Collection and Data Repositories	DoA T6.3 UCG-11-01 UCG-11-02 UCG-14-04
<ul style="list-style-type: none"> - Compute device trust level: The trust module collects all needed information [D5] and recomputes the trust level of the device [A05, D5]. - Compute device risk level: The TMS computes a new value for the risk level of a device [D5]. Information about the current device trust level, the current status of network attacks and network traffic related to the device. The device vulnerabilities and their exploitability, the device health level and views of peer-level TMSs [A05] are taken into account [D5, D6]. 	Cyber-Trust Trust and risk Characterisation	DoA T6.3 UCG-13-02 UCG-15-02

8.3 Key Quality attributes

Table 8.2: Quality Attributes

Key Quality Attributes
<p>Performance</p> <p>The performance of this component will be dependent on a number of attributes, these are the allocated computational power on the server or VM hosting the network architecture and assets repository, the available network resources as well as deployed</p>

mechanisms during implementation. Also, the type of scanning and monitoring tools will have a different requirement as some require high computational power and other need high bandwidth in order to perform simultaneously network scans. Metrics include the throughput which is the number of serviced requests per time unit, response time for benchmarking requests and the uptime as no errors should cause the system to crash. Also, the concurrent processes which include the maximum number of concurrent processes allowed running at any time on the gateway device will be measured. In addition to the number of active devices can be handled which include the maximum number of devices that the service can handle at the same time

Configurability

The network architecture and assets repository are employing a variety of tools allowing to get information on a network's architecture (including the topology and the security defenses deployed therein), assets and their values. Different tools have different requirement and configurations. A template-based configurations will be developed to aid initializing the scanning and monitoring tasks.

Security

Similar to the profiling service, the component network architecture and assets repository will be imposing security restrictions as it serves as the central data warehouse of the project. Towards this, incoming and outgoing information will go through a security module which will be responsible for admitting only agreed types and formats of information. The repositories of the system will be encrypted while data integrity will be preserved with the retention of the admitted info in a separate repository.

Communication with other Cyber-Trust modules will be done through the use of secure connections. No external component or user will be allowed to have direct access to any functionality of the system. Access is only granted to authorized Cyber-Trust architecture components.

Some types of security and variabilities scans will be extensive, and it is needed to be ensured that it does not cause and denial of service attack during the launch time.

8.4 Open concerns

A simple port scan such as SYN scan uses TCP functionality to find out if a port is open or closed will not cause a denial of service for the running daemons because they almost surely leave it to the operation system to handle the resulting events. However, when it comes to more extended scans that for example are used to guess which operation system is running on a host it can theoretically cause problems to very old or poorly configured hosts. This is because OS guessing is (besides other techniques) done by causing unusual situations to the host which then has to react. These reactions in many cases are specific to a TCP/IP Stack implementation which can be attributed to an OS or a Kernel Version. In addition, vulnerability scanner links Nessus without the needed knowledge and/or care. There are thousands of modules and some of them can cause a DoS (and they are labelled as those) to the scanned system.

9. Visualization Portal (A01)

9.1 Responsibilities

The Visualization Portal is a component concerned with inspecting, understanding and partially controlling a series of data concerning the status and health of the network, the IoT devices, and the collected logs. It is an interface between the users of the Cyber-Trust ecosystem and its backend services. It runs on the ISP infrastructure, where the server side is located, and at the client end, in the form of a standard web browser on PCs, smartphones or other portable devices, but also un 3D Virtual Reality devices for some use case scenarios. The aim of the Visualization Portal is to help the IoT user and the security officers to ‘make sense’ of the current state of things, through the presentation accurate, timely and synthetic information, in order to allow appropriate decisions to be made.

9.2 Key Functionalities

Table 9.1: Key functionalities

Key Functionality		
Description	Category	Ref
<p>This functionality is utilized by the end user, a Smart Home Owner[P1] or Smart Device Owner[P2], via a 2-dimensional User Monitoring Panel (UMP), a web browser component with a lightweight and intuitive interface, which allows the following operations:</p> <ul style="list-style-type: none"> • Synthetic health and risk level of the network • Current traffic levels on the network • Current risk level scores for the IoT devices owned • Receive alerts of current possible threats • Check of the devices patching status with the possibility of proceeding to the installation of the latest patches • Apply mitigation policies for current threats 	User Monitoring Panel	UCG-05-01, UCG-05-04, UCG-05-06, UCG-07-01, UCG-10-03, UCG-18-01
<p>This functionality is for the IOT Service Provider[O4], ISP Network Operator[O2] and LEA Operator[O3], and is implemented in a 2-dimensional panel, the Operator Monitoring Control Panel (OMCP), a browser component with a highly informative interface, consisting of several different pages. It allows the following operations:</p> <ul style="list-style-type: none"> • Display of a detailed health and risk level of the network • View network traffic with several graphs and at different detail levels • View historical data of different types and selection of time interval displayed • Establish baseline traffic statistics • Display topological and geographical information for the administered IoT devices • Check devices patching status and recommend installation of patches • Retrieve profile information for specific IoT devices and end users • Compute and display devices risk level • Explore trusted logs • Gather and display network forensic evidence • Curate the forensic evidence database • Visualize the status of devices during remediation • View applied mitigation policies <p>This is a highly complex tool, with many features and data sources. It will be implemented in different modules, so as to maximise performance and resiliency.</p>	Operator Monitoring Control Panel	UCG-05-01, UCG-05-04, UCG-05-06, UCG-05-09, UCG-06-03, UCG-10-03, UCG-11-02, UCG-12-03, UCG-12-04, UCG-14-04, UCG-15-02, UCG-16-02, UCG-17-01, UCG-18-01
<p>This functionality is for the IOT Service Provider[O4] and ISP Network Operator[O2] and LEA Operator[O3], and is intended for a novel and highly effective way of inspecting and interacting with the current security state of the network. It consists of an immersive 3-dimensional display implemented in Virtual Reality, the 3D-VR Operator Monitoring Environment (OME). The operator will don a Head Mounted Display providing virtual senses of sight and hearing, and haptic interfaces or manipulators for control actions. The OME will allow the operator to:</p>	Operator Monitoring Environment	UCG-05-02

<ul style="list-style-type: none"> • ‘be’ in the network, represented in the current time • Move, expand, change visual, drill down with several aspects of the data • View network traffic • View network threats • View a topological or geographical representation of the administered IoTs <p>It is anticipated that novel ways of using this interface might surface with practice.</p>		
--	--	--

9.3 Key Quality attributes

Table 9.2: Quality Attributes

Key Quality Attributes
<p>Usability</p> <p>Any User Interface needs to provide a tool which is fit to display the required information in a manner that results understandable, accurate and current. Considering the uncertainty in the data displayed, a measure of the probable error should also be indicated in the interface.</p> <p>The 2D interfaces, under the form of Panels and Dashboards, should be calibrated in user experience according to the end user:</p> <ul style="list-style-type: none"> • Intuitive and simple for the IoT User; colours and shapes of the data and controls should be what any user just expects • With more detail and a professional cut for the Administrators, with the possibility of drill down into the data; several different graph and data display styles should be available <p>The 3D interface will be quite novel but should be targeted to a relative novice, not an experienced gamer. The replication of 2D views in a 3D environment should be avoided. The VR interface will be a ‘world’ and networks, devices and nodes will be objects; health, risks and threats will be qualities. There will be positive and negative feelings represented.</p>
<p>Performance</p> <p>The interface needs to have a low latency time for responses to commands. Many display functionalities will require the transfer of data: sufficient bandwidth should be available, with feedback on the transfer state, and the possibility of aborting any current transfer operation.</p> <p>Servers should accommodate the simultaneous sessions of a high and scalable number of connected 2D displays. Most processing will be done on the servers, with rendering only or limited processing on the monitoring panels.</p> <p>The 3D-VR OME will support only a limited number (one or two) hardware devices. It requires a dedicated computer, a PC with at least 2 Core fast CPU and an Nvidia graphics GPU.</p>
<p>Security</p> <p>The display of data on users’ and IoTs’ profile is sensitive: access should be in an encrypted session. Particular care should be taken to avoid session hijacking.</p> <p>Different utilizers of the OMCP will need a different degree of access to the data they may inspect.</p> <p>The UCP and OMCP panels should be in the form of compiled apps, tamper-proof and as self-contained as possible, with security maintenance and upgrades. Penetration testing ought to take place to elevate security assurance in these tools.</p> <p>The specific problems of 3D-VR for what concerns security need further to be investigated in detail and addressed.</p>

9.4 Open concerns

- The volume of Traffic flow from the data sources to the data analysis components and the visualization components – there may be bottlenecks in bandwidth or latency
- Many different data sources might mean many different interfaces and data transfer structures
- User experience design (UX) of the UCP and OMCP panels
- Design of the 3D-VR worlds
- Development tools, testing, maintenance for 3D-VR apps

10. Profiling Service (A17)

10.1 Responsibilities

The Profiling Service is a Cyber-Trust component running on the Cyber-Trust backend. This component has a twofold responsibility: the central storage of device profiles and the correlation of existing information with newly acquired data from other repositories and sources (for patch and firmware updates as well as manufacturer use documentation). The profiling service is the primary interface with the Cyber-Trust backend components, and the responsible component for gathering and collecting information from the deployed SDAs (Smart Device Agents) and SGAs (Smart Gateway Agents). Due to its role the Profiling service is a component that runs on the ISPs infrastructure.

10.2 Key Functionality

Table 10.1: Key functionalities

Key Functionality		
Description	Category	Ref
<p>This functionality is responsible for the gathering of as much information as possible with regards to the state and characteristics of a device and as a result of the following sources</p> <ul style="list-style-type: none"> - Information gathered at the device level (CPU, memory, running processes, network usage etc.) - End-user input while user and device registration. - Information extracted from retrieved Manufacturer Use Directions (MUD) whenever these are available; - Historical data on track of firmware update, encountered attacks, trust and risk level scores 	Device Profiling	DoAT6.2, T6.3 UCG-10-01, UCG-10-05
<p>Trust and Risk level scores: store the trust and risk level scores score of the device as calculated with the use of device information and the state of network architecture assets from the TMS service.</p> <p>Match device profile with eVDB contents: The device profile is matched against the contents of the VDB. Detailed information on a device's vulnerabilities (and other related information is then retrieved.</p> <p>Check device patching status: Intelligence regarding the latest versions of firmware is stored in the Cyber-Trust backend system. Periodically, the installed firmware and software on monitored devices is checked and when outdated the end user is notified.</p> <p>Ensure Device firmware integrity: A backend service runs between the host and the devices information database to ensure that activated devices operate with the vanilla firmware. In case a fraudulent or altered firmware is detected then backend services for DPI are triggered for remediation and mitigation actions to take place.</p> <p>Update device critical OS files/vulnerabilities: In case a legitimate update is performed on the OS, firmware or any critical device files, key device parameters are recalculated and updated to the central database. Then the process of detecting vulnerabilities is also performed.</p>	Monitor Current Status of Active Devices	DoA T5.3, T5.4 T6.2, T6.3 UCG-13-02, UCG-14-08, UCG-15-02, UCG-07-01, UCG-07-03, UCG-14-01
Data related to the operation of Cyber-Trust is useful throughout the ecosystem and also to external platforms. Such information is anonymised and shared within the system for the improvement of system operations. This use case is also involved in cases sharing of information with other external platforms is desired	Data Anonymisation	DoA T5.3, T5.4 T6.1, T6.2, T6.3 UCG-10-02

<p>For each type of registered device, the Patch database contains information related to the latest security fixes of the firmware as well as the relevant binaries. Hash information is also securely stored to ensure the integrity of the vanilla patch versions.</p>	<p>Manage available patch databases</p>	<p>DoA T6.1, T6.2, T6.3 UCG-14-08, UCG-14-02</p>
<p>The Profiling supports a flexible access roles scheme. One's access role does not only determine its eligibility in accessing system functionality (vertical access rights) but also eligibility in accessing parts of information.</p>	<p>User Access Rights</p>	<p>DoA T5.3, T5.4 T6.1, T6.2, T6.3 UCG-10-01, UCG-10-05</p>
<p>Management of information will be accomplished with the extraction of metadata information from each piece of data. The basic abstract unit of storage is the media object which consists of a binary together with textual metadata in the form of key-value pairs. Each media object is identified by a universally unique identifier (UUID). A media object can be retrieved by specifying its UUID. The media object also has a type defined by a type field. The type defines the constraints/requirements on the key names. The extraction of the necessary set of meta data will depend upon the type of information as well as the set of actions to which the media object is eligible.</p> <p>The different functionalities will be accommodated through a number of centres of which each one is specialised for specific tasks according to their nature. Through this practice the Profiling Service can be scaled to handle more functionalities if required and enables combining information and embedding expert knowledge.</p>	<p>Data Management & Storage Principles</p>	<p>DoA T5.3, T5.4 T6.1, T6.2, T6.3 UCG-10-01, UCG-10-05</p>

10.3 Key Quality attributes

Table 10.2: Quality Attributes

Key Quality Attributes
<p><u>Performance</u></p> <p>The performance of this component will be dependent on a number of attributes, these are the allocated computational power on the server or VM hosting the Profiling Service, the available network resources as well as deployed mechanisms during implementation. The first two influencing factors shall be determined at a later stage in the project and are expected to be finalized when a solid deployment plan of the Cyber-Trust platform has been developed.</p> <p>However, since the performance of this module is critical, the Profiling Service could also deploy advanced mechanisms to ensure stable performance even at periods of high demand. The system maintains objects in its repositories for its TTL (Time to Live) period. In the case where performance degradation is noted, the system will activate mechanisms that will resume the component to a stable state.</p> <p>The Profiling Service will only communicate with Cyber-Trust internal components and therefore there is no such demand for a maximum number of accommodated users. However more solid and specific metrics will be presented when all communication means and exchange unit information (formatting conventions) have been determined. Such metrics include the throughput which is the number of serviced requests per time unit, response time for benchmarking requests and the uptime as no errors should cause the system to crash.</p>
<p><u>Configurability</u></p> <p>The system deploys a design where more functionality and components could be added at any stage. However the core mechanisms of the Profiling Service will not be subject to changes to avoid compromises in both security and performance. With regards to scalability and the need to accommodate more workload, the Profiling Service could allow the implementation of new processing centers for increased performance; these need to be determined and configured during setup. The Profiling Service also provides configurability and flexibility in accommodating more types of content.</p>

Security

Security in the Profiling Service will be provided only through the deployment of stable versions of software modules. This component will be imposing security restrictions as it serves as the central data warehouse of the project. Towards this, incoming and outgoing information will go through a security module which will be responsible for admitting only agreed types and formats of information. The repositories of the system will be encrypted while data integrity will be preserved with the retention of the admitted info in a separate repository.

The Profiling Service also supports auditing features for monitoring activity and service requests. This includes information such as who has created/accessed/modified/renewed etc. a media object. Communication with other Cyber-Trust modules will be done through the use of secure connections. No external component or user will be allowed to have direct access to any functionality of the system. Access is only granted to authorized Cyber-Trust architecture components.

10.4 Open concerns

- Avoid duplication of stored information, considering a hybrid communication system will be used combining centralized and decentralized methods,
- All communication between network and end devices (both centralized and peer-to-peer) needs to be monitored to allow sufficient performance in terms of detection and monitoring

10.5 High-Level Design

The following diagram illustrates the conceptual view of the Profiling Service. The architecture follows a loosely coupled approach allowing for scalability and extensibility where necessary. Direct interfacing with internal components and repositories is not supported due to security reasons. The Profiling Service ensures data integrity by maintaining a data revisioning system allowing tracking and monitoring of data evolution.

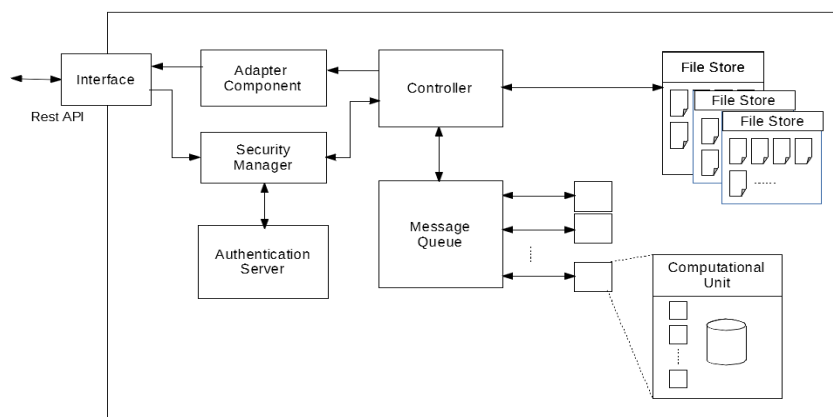


Figure 10.1: A17 High-Level Design

With regards to authentication and security, the Profiling Service supports the following features:

1. User management (creation/deletion/modification etc.)
2. Data access rights (customisation and configuration of access rules for each piece of stored data)

3. Flexible user access rights (customisation is supported not only per user type but also per data object)
4. Certificate negotiation is performed through existing standards. Currently, the system supports OAuth 2.0. However, support to additional standards is possible if project requirements suggest so.
5. Communication between both internal and external components is routed through secure connections

11. Cyber-Defence Service (A04)

11.1 Responsibilities

This covers the detection and mitigation of Cyber-Attacks on networks and device level. This service involves the decision making at defending the Cyber-Trust system and limiting the impact of the imposed threat; the detection and mitigation centre depending on the findings of the analysis determines the most appropriate mitigation policy to be taken at the device level.

11.2 Key Functionality

Table 11.1: Key functionalities

Key Functionality		
Description	Category	Ref
<p>Device-level monitoring and detection</p> <ul style="list-style-type: none"> - Monitor device at the gateway (network traffic filtering): The gateway is running network intrusion detection systems (NIDS) to check for signatures and anomalies based on the signature. <p>Network-level monitoring and detection</p> <ul style="list-style-type: none"> - Detect network attacks: Intrusion detection system (IDS) is utilized to monitor packets on the network in order to detect the attacks and malicious threats in the network. IDS compare packets signature against a database of signatures or attributes from known malicious threats. 	Continuous Monitoring	T6.2 T6.3 UCG-08-01 UCG-09-04 UCG-10-03
<p>Device -level mitigation</p> <ul style="list-style-type: none"> - Apply device security defense rule: Network security combines multiple rules and layers of defenses [A04] at the edge and in the network and as such represents the decision implementation element of the intersection of vulnerability (built from trust, threat and DLT-assured device profiles [A02]). These include access control, application security, Intrusion prevention system, firewall and many more. 	Mitigation	T6.2 UCG-18-01
<p>Device-level mitigation</p> <p>Ensure Device firmware integrity: A backend service runs between the host and the devices information database to ensure that activated devices operate with the vanilla firmware. In case a fraudulent or altered firmware is detected then backend services for DPI are triggered for remediation and mitigation actions to take place.</p>	Mitigation and health check	T6.3 UCG-07-03

11.3 Key Quality attributes

Table 11.2: Quality Attributes

Key Quality Attributes
<p><u>Performance</u></p> <p>The cyber-defense service requires high resource capabilities in order to handle large volume of data. The performance of the cyber-defense service may be affected by the following parameters:</p> <ul style="list-style-type: none"> • Available CPU and Memory for the service • Supported communication technologies and protocols • Average response time: The amount of time that the SGA takes to respond to a request • Network topology size and number of active devices • Type of WAN Network Connectivity and consumed bandwidth • Number of active devices to be monitored within the cyber trust platform
<p><u>Reliability</u></p> <p>The reliability for the execution of operations might be affected by the underlying hardware and exposed interfaces. Also, other external factors such as issues that directly affecting wireless performance such as radio interference and coverage may exist.</p>
<p><u>Security</u></p> <p>The Cyber-defense service requires a secure storage while performing inline traffic analysis. This requires ensuring that data is processed within a high confidentiality and integrity. Also, a secured communication will be supported between the cyber-deface service and Cyber-Trust backend services.</p>

11.4 Open concerns

Some types of smart devices require high-level of OS permission in order to be able to apply some security role. For instance, the device-level detection scenario, will Android-based device will be rooted in order to run the required agents. Another challenge is when there is an ongoing attack covered by encrypted traffic in which the deep packet inspection will be limited.

12. Trust Management (A05 + A08)

12.1 Responsibilities

The Trust Management Service (TMS) Cyber-Trust component that may run on different levels of the Cyber-Trust platform architecture:

- (a) at ISP/Cyber-Trust security provider level,
- (b) at smart gateway level and
- (c) at the smart mobile device level.

This component plays a central role in the operation of the Cyber-Trust platform, undertaking the computation of risk level and trust level, triggering actions to redefine communication rules based on the current trust and risk level assessments and liaising with peer-level TMSs to exchange trust assessments. The trust and risk levels are recomputed whenever the underlying data are modified (or their change exceeds a specified threshold). The Trust Management Service also offers a user interface, through which the internally maintained trust database can be viewed managed. To accomplish its tasks, the TMS collects information from various other Cyber-Trust components, namely the profile service (A17), the enhanced vulnerability database (A07) and the network architecture and assets repository (A16), while it also communicates requests to the IIRS module (A13). The TMS fully encapsulates the trust database (A08), and is responsible for updating its contents and also accepting and serving requests for retrieving content from it (in particular trust and risk levels).

12.2 Key Functionality

Table 12.1: Key functionalities

Key Functionality		
Description	Category	Ref
<p><u>Defining mitigation policies and parameters</u></p> <ul style="list-style-type: none"> - Characterize asset’s importance: The security officer prioritizes the attributes of the devices and their services owned based on his knowledge. This information (i.e. which exploits to block, etc.) is used for optimizing the defense actions to be applied and to maximize security. - Define applicable mitigation actions: The security officer consults the system’s vulnerabilities and exploits and views/defines/updates the mitigation actions which are applicable. - Define mitigation actions’ impact: The security officer quantifies the impact that the various mitigation actions have on the availability of the network resources to trusted devices (e.g. by refusing communication requests, shutting down running services, etc.). 	Mitigation	DoA T5.3, T5.4, T6.3 UCG-04-02 UCG-18-06 UCG-04-03
<p><u>Computation and delivery of trust and risk assessments</u></p> <p>The TMS undertakes the task of computing the trust level of devices (i.e. the degree to which a device is considered trustworthy) as well as the respective risk level (i.e. a measure of the potential demotion that known assets may sustain due to the current state of the respective device). In this respect, the TMS needs to:</p> <ol style="list-style-type: none"> 1. Collect a comprehensive view of the current state the device’s security aspects (including its integrity status and security controls), its behavioral patterns (e.g. attacks it has participated in, alignment with nominal network flow patterns), as 	Mitigation	UCG-13-02 UCG-15-02 UCG-13-01

<p>well as the associated risk (which vulnerabilities is the device exposed to and what is their expected impact).</p> <ol style="list-style-type: none"> 2. Regarding the expected impact, in particular, the TMS will examine not only the direct demotion of asset values hosted on the particular device, but also the potential to use the device under examination as a springboard for launching attacks to other devices (which possibly host assets with greater value). 3. Gather from peer-level trusted TMSs assessments of the devices' trust and risk levels. 4. Compute the overall trust and risk scores and store them in a database. 5. Accept and honour requests from other Cyber-Trust system modules for retrieval of the trust and risk level of designated devices. 		
<p><u>Computing mitigation policies and attack surface</u></p> <p>When a device's trust level is demoted or the corresponding risk level is raised, the TMS consults the mitigation policy database to identify actions that may need to be applied in order to protect the assets.</p> <ul style="list-style-type: none"> - Compute cyber-attack graphical security model: Create an attack graph that presents how exploits relate to security conditions. These conditions denote system attributes and represent the attacker's capabilities. Information about the exploits is retrieved by the eVDB and attacker's privileges on hosts, the profile of devices from the profile DB, and information about network hosts, connectivity, hosts' trust relationships, etc., from the network assets DB. - Compute attack's likelihood and success probability: The system consults the Profile DB to get information about the CVSS score (and other info) associated with a device's vulnerabilities (or it contacts the eVDB if this is not available) and use it for computing attacks' likelihood and success probabilities. - Identify and prioritize cyber-threats: Based on the above information, the TMS will use the graphical security model to identify the cyber threats and calculate the level of exposure to these threats and moderate their impact. Based on the final impact assessment, cyber-threats will be prioritized. - Compute device risk level: The TMS computes a new value for the risk level of a device. Information about the current device trust level, the current status of network attacks and network traffic related to the device (as compared with the baseline), the device vulnerabilities and their exploitability, the device health level and views of peer-level TMSs are considered. - Compute optimal risk mitigation actions: Optimal risk mitigation actions are computed based on the graphical security model and the prioritization of cyber-threats. 	Mitigation	<p>DoA T5.3, T5.4, T6.3</p> <p>UCG-15-01 UCG-15-03 UCG-16-04 UCG-15-02 UCG-15-04 UCG-18-05</p>
<p><u>Supporting mitigation and communication</u></p> <ul style="list-style-type: none"> - Communicate risk mitigation actions to the security officer: After having computed the optimal action, the security officer is informed about every defense action having been applied by the system or about specific defense actions, e.g. the most critical ones. - Retrieve mitigation policy information: Retrieve mitigation policy from the database on the threat detected. If no mitigation applies then default measure is to block connectivity. 	Mitigation	<p>DoA T5.3, T5.4, T6.3</p> <p>UCG-06-07 UCG-18-02</p>
<p><u>Storage and retrieval of trust information</u></p> <p>The trust-related information (trust and risk assessments for devices) shall be stored within the trust database, to be later retrieved and processed or returned as a response to queries.</p>	Data organization and storage	<p>UCG-13-02 UCG-15-02</p>

12.3 Key Quality attributes

Table 12.2: Quality Attributes

Key Quality Attributes
<p><u>Configurability</u></p> <p>The software should be configurable to accommodate variations in the way that trust and risk levels are computed, and in particular the weight that is assigned to different aspects of the trust and risk calculation (e.g. the level to which the presence of some security defense moderates the risk level associated to a device). This configuration should be performed without needing to modify the module’s code. Configurability also applies to the definition of trusted peer-level TMSs, which are consulted in the process of computing trust and risk assessments.</p>
<p><u>Performance/Scalability</u></p> <p>The TMS is a focal point in the Cyber-Trust architecture, being consulted by different modules and with high frequencies. In this respect, the performance of the TMS is of critical importance for the effective operation of the Cyber-Trust platform. To maintain a high level of performance, especially under the scenario of an extension of the Cyber-Trust user base, horizontal and vertical scaling techniques can be considered. Horizontal techniques apply not only to the spreading of TMS-related tasks to multiple machines at the ISP level, but also distribution of TMS-related tasks to different levels in the Cyber-Trust architecture: realizing a number of (or the full set) of TMS-related tasks to the TMS located at the Smart Gateway –which is expected to have sufficient resources– is expected to substantially contribute to the support of scalability, not only distributing processing power but also limiting the network latencies stemming from data transmission activities to/from Cyber-Trust security provider-hosted and cloud-hosted infrastructures.</p>
<p><u>Timeliness / Information up-to-dateness</u></p> <p>The cyber threat landscape is highly dynamic in multiple respects, including (a) identification of new vulnerabilities and exploits (b) the introduction and retraction of protected assets and threat agents (c) the state and behaviour of all monitored resources. Any change in the aforementioned elements necessitates the recomputation of risk and trust levels for a number of affected devices, and the potential application of appropriate defense measures. The effective level of cyber threat mitigation provided by Cyber-Trust clearly depends on the timeliness of TMS activities (in particular the recomputation of trust and risk assessments and application of mitigation actions), which in turn depends on (a) the up-to-dateness of information available to the TMS and (b) the timeliness of notifications that the TMS receives regarding changes in the different information repositories. To that effect, efficient event communication mechanisms should be deployed in the Cyber-Trust platform, allowing relevant modules to notify the TMS about changes in the underlying data and thus trigger the appropriate activities.</p>
<p><u>Availability / Robustness</u></p> <p>Trust assessments moderate the operation of communications throughout the scope of the Cyber-Trust platform. Therefore it is critical to ensure a high degree of availability for them. In a real-world environment, cases such as host and network outages, service malfunctioning, certain types of cyber-attacks (e.g. DOS/DDOS) etc. demote the level of services’ availability; such issues can be tackled by providing resource replication, caching mechanisms or local information computation processes (possibly suboptimal), which can be triggered when access to remote resources is hindered. Being able to operate in a disconnected/isolated/limited access environment entails the challenges of data synchronization, appropriate resource access exception handling and working with outdated data. Relevant mechanisms and solution to these challenges will be considered in the context of the Cyber-Trust platform.</p>
<p><u>Privacy/Security</u></p> <p>The computation of the risk and trust levels for devices entails the use of information concerning the health level of the devices, their network traffic behaviour, security controls that apply to them etc.; these data should be safeguarded, both for privacy reasons (they can be considered personal data) and for security purposes (e.g. the disclosure of security controls would provide potential attackers with valuable information). Moreover, the risk and trust levels themselves constitute ratings, and in this respect fall under the personal data domain. Taking these into account, the computation and storage of the risk and trust levels for devices should be best performed at machines under the user’s full control, without disclosing information to broader levels (e.g. ISP level). Naturally, when ratings or underlying data are necessary for the operation of the Cyber-Trust platform, these would be made available to the platform, under the principles of informed consent and reciprocity.</p> <p>Finally, since the TMS includes functionality to report the trust and risk levels of devices and a prioritized list of threats, clearly</p>

access to these functionalities should be safeguarded with appropriate authentication and authorization mechanisms.

Resource availability

The computation and storage processes that are performed in the context of the TMS necessitate the consumption of processing, memory, storage and power resources. Correspondingly, the three envisioned architectural elements on which the TMS module will be hosted (ISP/Cyber-Trust security provider level, Smart Gateway level, smartphone level) have significantly different amounts of resources available. In particular, at the smartphone level the processing, memory and power resources are limited, hence processing should be limited; to this end, algorithms employed at that level may trade off resource efficiency with result optimality; alternatively, computations entailing heavy resource usage may be delegated to trusted architectural levels with ample resources (i.e. the ISP/ Cyber-Trust security provider level or the Smart Gateway level), i.e. instead of performing the computations locally, a corresponding service hosted at the ISP/ Cyber-Trust security provider level or the Smart Gateway level could be invoked. Regarding this approach, the network context of the smartphone should be taken into account since (a) networking may not be available (b) networking could be available but bandwidth usage would be limited or charged (a typical case when mobile data are used) or (c) network resources could be in abundance (a typical case when the smartphone is connected to a Wi-Fi network).

12.4 Open concerns

- Selection of base tools that the TMS will be built upon; the open source domain hosts a number of tools, but documentation is scarce and available functionalities appear limited.
- Tuning the synthesis of individual aspects considered in the trust and risk computation (exposure to risks, value of assets, behavior of device, security controls available, placement of device into the network architecture, current threat level) into a single, comprehensive and accurate risk/trust level score.
- Efficient implementation of the risk/trust computation algorithms and provision of a tunable or adaptive version for use in mobile platforms.
- Optimally balancing between non-functional aspects related to (a) timeliness / information up-to-dateness (b) availability and robustness and (c) resource consumption.

12.5 High Level Design

The following diagram illustrates the conceptual view of the Trust Management Service. Its architecture is designed to allow for exposing a coherent API, allowing the adaptation aspects to be implemented internally, considering all the appropriate contexts (network & resource availability, situation criticality etc.). A separate communication channel for triggering event reception is also accommodated. This can be realized either as the subscribing part of a pub/sub channel (fostering loose coupling between event producers and consumers); triggering actions could also be submitted through the endpoints realized in the API. Adaptation, where needed, will be supported by an adaptation component to be developed and maintained separately from the computational aspects, promoting separation of concerns.

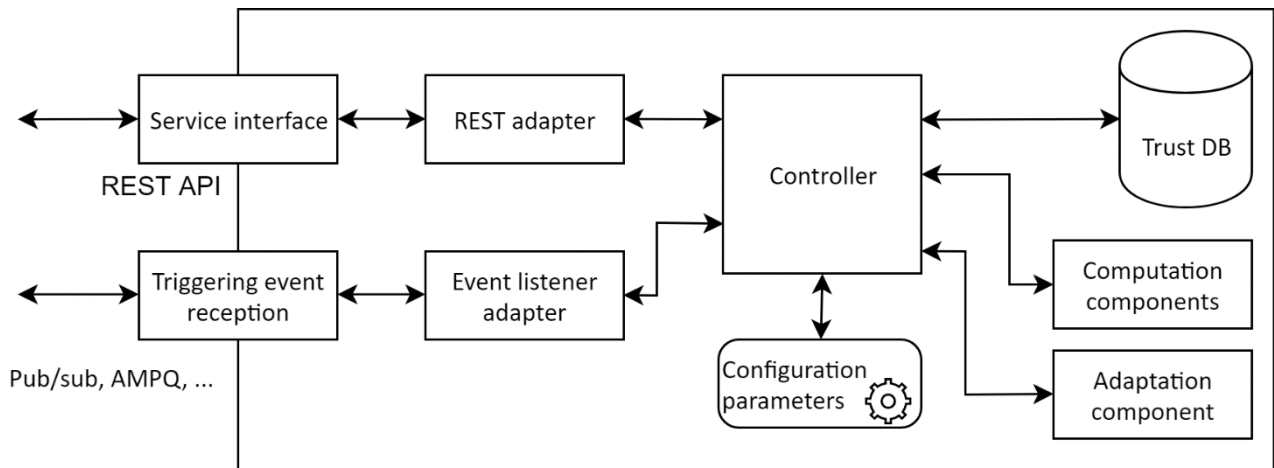


Figure 12.1: A05+A08 High-Level Design

13. CT Registration Module (A06)

13.1 Responsibilities

The Cyber-Trust Registration Module is part of the Admin Portal and is responsible for the administration of the identities and attributes of all the participating actors. This entails:

- Registering new Person and Organization Class Actors, provide them with a unique, secure and anonymous identifier
- Register identities of IoT devices to be administered and their association with Persons and Organizations
- Register into profiles the attributes associated with identities
- Associate profiles to permissions for operations, according to policies
- Allow the modification of registration data to authorized parties, and its deletion
- Provide a secure Login, Logout and Session Control for access to the Cyber-Trust platform
- Provide data on identities, attributes and profiles to all use-case activities which require them

The Registration Module intervenes in a high number of operations and is connected to many use-case activities. It provides a positive assurance that the entity involved is authentic, that the associated data are authoritative. Its concerns are the maintenance of data Confidentiality, Integrity and Availability. It ensures conformance to policies. It provides extensive logging for auditing and security purposes.

13.2 Key Functionalities

Table 13.1: Key functionalities

Key Functionality		
Description	Category	Ref
<p>The main functionality concerns the registration of actors and comprises:</p> <ul style="list-style-type: none"> • Register a user into the Cyber-Trust platform • Register an organization into the Cyber-Trust platform • Register a device into the Cyber-Trust platform • Change a device configuration • Get the information on a device <p>Closely associated is the handling of login and logout activities, and the corresponding handling of an active session:</p> <ul style="list-style-type: none"> • Log on to the Cyber-Trust platform • Log off from the Cyber-Trust platform <p>All these activities are based on a single registration repository, in the form of a database or similar, which reflects the current state of participating identities.</p> <p>This function will also maintain logs of registrations, attribute variations, deletions, as well as successful and unsuccessful login and logout activities, and session durations.</p> <p>The module will perform standard insert, update and delete operations on the identities repository, all such requests being in turn performed by an authenticated party, on an encrypted channel, through a standardized REST interface.</p>	Registration and Session	UCG-02-01, UCG-02-02, UCG-02-03, UCG-02-04, UCG-03-01, UCG-10-06, UCG-19-03
<p>Other processes, services and use-cases will query the registration repository and obtain information on the identities, which will be used directly, or correlated with data in other repositories or databases.</p> <p>Such activities include:</p> <ul style="list-style-type: none"> • Deployment of device agents • Activation/deactivation of device agents • Visualization of device vulnerability and trust levels • Device profiling • Host and Net-based vulnerability scanning • Collection of device forensic evidence • Matching a device profile with the eVDB database 	Queries and Confirmations	UCG-01-01, UCG-01-02, UCG-05-05, UCG-05-07, UCG-07-02, UCG-09-03, UCG-10-01, UCG-11-01, UCG-14-01, UCG-14-02, UCG-14-08

<ul style="list-style-type: none"> • Updating critical OS files • Managing and applying security patches • Applying remediation measures <p>These activities do not perform insert, update or delete operations on the identities registry, but only queries. Such queries can return one typed result, a set of homogeneous results or a boolean value for confirmation queries.</p> <p>Metadata will provide information on who can perform queries, on which types of identities, and which attributes will be returned.</p> <p>Data returned will mostly consist of a limited sequence of bytes in an encrypted format, and the module will communicate with a standardized REST interface.</p>		
--	--	--

13.3 Key Quality attributes

Table 13.2: Quality Attributes

Key Quality Attributes
<p>Security</p> <p>Any Identities DB request, of any type – insertion, deletion, variation, query – will be performed on an encrypted channel by an authenticated party. Authentication tokens and session keys will be provided by a centralized Key Distribution Centre. The organization of the service will closely correspond to the usage of standard LDAP/Kerberos services. Logons and logoffs will be of the Single Sign-On/Off type.</p> <p>For sensitive operations, a two-factor authentication will be requested, in the form of a variant to the OAuth protocol.</p> <p>All queries and responses will be on encrypted channels using TLS and X.509 certificates, the issuing CA being within the Cyber-Trust project. Such certificates cannot be used outside the project, and outside issued certificates cannot be used in the project.</p> <p>Measures will be taken to contrast Man-In-The-Middle attacks.</p>
<p>Replication and Data Protection</p> <p>Both the Identity Server and Identity Repository components of this module will be replicated and a transparent failover capability will be provided.</p> <p>All DB operations will be transactional and ensure a sufficient isolation level of the data.</p>
<p>Performance</p> <p>It should be possible to perform update operations in the range of tens per second, and query operations in the range of hundreds per second.</p> <p>A provision for batch update should be provided.</p>

13.4 Open concerns

- Where to store the private keys of operators with CRUD permissions
- How to proceed securely to the first registration of end users
- Whether to separate the KDC/CA functionalities from the Registration Module

14. Enriched Vulnerability Database (eVDB) (A07 + A09)

14.1 Responsibilities

This component encompasses the *enriched vulnerability database* (eVDB) admin [A07] and the *cyber-threat intelligence* (CTI) sharing [A09] modules. The eVDB admin module is responsible for the usage and the maintenance of the database storing enriched data about vulnerabilities, exploits, etc., that are collected through CTI techniques. The eVDB sharing services link the eVDB to Cyber-Trust registration portal [A06]. It enables the dissemination of results and information regarding vulnerabilities, exploits, cyber-attacks, etc. with affiliate members and individuals.

14.2 Key Functionality

Table 14.1: Key functionalities

Key Functionality		
Description	Category	Ref
<p>Update the eVDB with information crawled be the clear/deep/dark web</p> <p>The Cyber-Trust system continuously crawls popular social media streams, popular security-related websites and deep/dark web forums and marketplaces [A10, D6]. Cyber-Trust searches for cyber-threat information including zero-day vulnerabilities, exploits, signatures, executables, and other related information. The collected data will update the eVDB [A07, D6]</p>	Continuous monitoring	DoA T2.3 T5.1, T5.2, T5.4 UCG-16-05
<p>Review and validate eVDB entries</p> <p>The vulnerability assessment expert examines and assesses newly discovered cyber-threats, reviews the new vulnerabilities that were surfaced by Cyber-Trust and decides if there exists enough evidence to update the report confidence (RC) field of the discovered vulnerabilities [D6].</p>	Maintenance - Updating	DoA T2.3 T5.1, T5.2, T5.4 UCG-06-05
<p>Provide feedback/rating on sources of vulnerabilities.</p> <p>A vulnerability assessment expert provides feedback on the quality of the information gathered from the crawling of new seeds [A10, D6]. He also approves and annotates approved seeds for usage by the crawl module.</p>	Maintenance - Updating	DoA T2.3 T5.1, T5.2, T5.4 UCG-06-06
<p>Match device profile with eVDB content</p> <p>The device profile is matched against the contents of the eVDB [A07] to retrieve vulnerabilities (and other related information) that pertain the device [D5, D6].</p>	Querying - Continuous monitoring	DoA T2.3 T5.1, T5.2, T5.4 UCG-14-08
<p>Use of the eVDB sharing service</p> <p>This functionality includes</p> <ul style="list-style-type: none"> Registration to the eVDB sharing service, in order to create a unique user profile based on the role he/she has. Dissemination of information on new vulnerabilities and the available mitigations actions as well as any subsequent updates to users of different expertise and roles (security officers, smart device owners, etc.). Users that are already registered in the platform search and retrieve for any security issues and intelligence that pertain to devices. 	Registration and dissemination	DoA T2.4 T5.1, T5.2, T5.4 UCG-02-05 UCG-18-06 UCG-06-04 UCG-14-08
<p>Raise alert for the device owner</p> <p>This use case study is to raise an alert to the [P1, P2] user when for example, there is a threat</p>	Querying	DoA T2.3 T5.1, T5.2, T5.4

is detected in one of the owners' s IoT devices.		UCG-06-02
<p>Notify about updates and security-related issues</p> <p>The device profile is matched against the contents of the eVDB [A07] to retrieve vulnerabilities (and other related information) that pertain the device [D1, D5, D6]</p>	Querying - Continuous monitoring	DoA T2.3 T5.1, T5.2, T5.4 UCG-14-07
<p>Visualization</p> <p>Compute cyber-attack graphical security model</p> <p>Create an attack graph that presents how exploits relate to security conditions. In doing so, information about the exploits [D4, D5] is retrieved by the eVDB [A07].</p> <p>Visualize known and zero-day vulnerabilities</p> <p>The user can obtain from the vulnerability search interface of the eVDB [A07] (MISP) the list of known, zero-day, etc. vulnerabilities retrieved using different classification criteria assigned during the deep/dark web processing.</p>	Querying - Visualization	DoA T2.3 T5.1, T5.2, T5.4 UCG-15-01 UCG-05-08
<p>Compute attack's likelihood and success probability</p> <p>The Security officer consults the eVDB [A07] to compute the attack's likelihood and success probabilities. This information [D2, D4, D5] is vital for the iIRS [A13] to compute the best mitigation actions, because of the stochastic nature of the decision-making process of the iIRS [A13].</p>	Querying	DoA T2.3 T5.1, T5.2, T5.4 UCG-15-03

14.3 Key Quality Attributes

Table 14.2: Quality Attributes

Key Quality Attributes
<p><u>Scalability</u></p> <p>eVDB is expected to store a huge volume of (complex) information about security and vulnerability. Additionally, the number of users that are registered and are expected to use eVDB is expected to be large. Thus, eVDB should be able to accommodate the extensive number of both data and users.</p>
<p><u>Performance</u></p> <p>The performance of a database depends on the complexity and the volume of stored data and the number of users that simultaneously access it. eVDB will store complex information of an extensive volume and should be able to service a huge number of concurrent accesses. Thus, performance is very crucial and appropriate measure will have to be taken to assure prompt response times.</p>
<p><u>Security</u></p> <p>Secured communication and data exchange between the eVDB and its users (either physical or Cyber-Trust components) should also be safeguarded. Only registered users will have access and receive information regarding relevant (to their profile) vulnerabilities, exploits etc.</p>
<p><u>Reliability</u></p> <p>Because the eVDB Sharing Services interact with such a massive volume of information,</p> <ul style="list-style-type: none"> - the information that is produced by the algorithms utilised should be reliable. <p>the channels that connect these sources of information should be reliable too.</p>

Privacy-awareness

Since a significant part of the stored information is automated crawled, it is possible to accidentally gather personal data. To avoid accidentally exposing such data appropriate data anonymization/aggregation techniques should be applied.

14.4 Open concerns

- Selecting the proper mix of tools and technologies to deliver efficient and effective services is under consideration as part of T5.1.
- System specifications (e.g., computational power, network resources) for supporting high-throughput query evaluation and answering.
- Avoiding the collection (if possible) and exposure of personal information is a research challenge.
- The domain of cyber-defense in IoT is growing rapidly, and the eVDB sharing methods should adhere to broad architectural prototypes, to adapt in any kind of future IoT software changes.

15. Distributed Ledger Technology (A02, A15)

15.1 Responsibilities

The distributed ledger technology (DLT) is a Cyber-Trust component running on the professional actors' (ISP, LEA, ...) back-end. It will also run on the Cyber-Trust back-end. It aims to store data provided by other Cyber-Trust component regarding privacy and scalability issues.

15.2 Key Functionality

Table 15.1: Key functionalities

Key Functionality		
Description	Category	Ref
The data stored in the DLT are critical so we have to ensure that one has the rights to interact with the data stored in the DLT. Moreover, the data can be exported and saved out of the DLT with the right authorization.	Authority management	Doa 7.2, UCG-12-01, UCG-12-02, UCG-12-03, UCG-12-04
Metadata's related to each attacked encountered by Cyber-Trust user will be stored to keep track to who owns the actual information. These data will be used mostly by the ISP to help them during their investigation.	Forensic evidence storage	Doa 7.2, UCG-11-01, UCG-11-02, UCG-12-02, UCG-12-04, UCG-12-05, UCG-14-06
For each patch contained in the patch database, metadata will be store in the DLT. This way a smart-device will a trusted oracle to query about what firmware is safe and the URL to download it.	Trusted storage files	Doa 7.2, UCG-10-06, UCG-12-01, UCG-12-03, UCG-12-05, UCG-14-02, UCG-14-05, UCG-14-06, UCG-19-02, UCG-19-03
The DLT will be used to represent which company designed each device class and information relative to the class such as the risk level associated with the device.	Ownership management	Doa 7.2, UCG-02-02, UCG-02-03, UCG-03-04, UCG-03-03

15.3 Key Quality attributes

Table 15.2: Quality Attributes

Key Quality Attributes
<p>Security</p> <p>Using a blockchain instead of a traditional DB will create a secure decentralized network, where devices can connect and interact in a trusted way.</p>

Performance

Cyber-Trust DLT needs a good performance because of the increasing number of IoT device in the world. Using private permissioned blockchain will help us with that. Indeed, all the nodes on the network will not be asked to process the transactions. This will also result in high transaction throughput.

15.4 Open concerns

- Legal concern is the biggest concern as there is no strict legislation. Regarding the forensic evidence we will only be able to store metadata. Same for the trusted files.

15.5 High-Level Design

We will use HyperLedger as a framework to create Cyber-Trust DLT, as it is an open source framework to create private permissioned blockchain with great community support. Nodes will be running on the different partners of Cyber-Trust, companies such as ISP or IoT device manufacturers. LEA agencies will be running to get aware of the addition of new metadata related to forensic evidence. One central will also be running inside Cyber-Trust back end.

Here is an example of how the system will add a new forensic evidence block into the DLT.

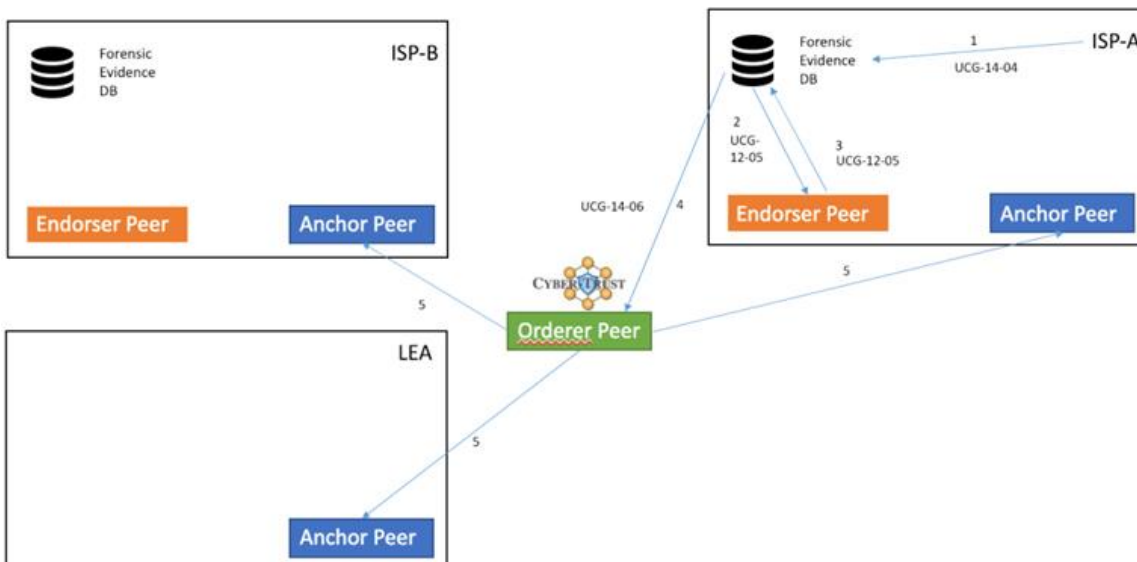


Figure 15.1: A02 + A15 High-Level Design

16. Crawling Service (A10)

16.1 Responsibilities

The Crawling Service component lies at the core of the cyber-threat intelligence gathering envisioned by Cyber-Trust. It is responsible for (a) collecting public cyber-threat intelligence information from the social/clear/deep/dark web, including related forums, marketplaces and security-related websites, (b) leveraging the collected information to identify emerging threats, zero-day vulnerabilities and new exploits to IoT devices, and (c) making the leveraged information available to the rest of the Cyber-Trust platform by storing it in the eVDB. To do so it utilizes an ensemble of state-of-the-art data processing and machine learning techniques to identify the web pages that should be crawled and to extract/contextualize all relevant threat information. The Crawling Service also offers a user interface, through which the crawling process can be supervised, managed and tuned. It interacts only with the eVDB Sharing Service, which is used for storing and sharing of the actionable intelligence that has been discovered.

16.2 Key Functionality

Table 16.1: Key functionalities

Key Functionality		
Description	Category	Ref
<p>This functionality is responsible for automatically and continuously crawling a variety of web sources to discover cyber-threat intelligence pertaining to the IoT domain and to surface new relevant seeds that will be added to the crawl frontier. The crawling functionality is expected to span over a variety of different (information-rich) sources such as:</p> <ul style="list-style-type: none"> - the clear web (e.g., popular security-related websites, publicly accessible vulnerability databases, security-related portals) - the social web (e.g., popular social media streams, technical forums, security-related blogs and wikis) - the deep/dark web (e.g., hack forums and related marketplaces) 	Crawling & seed update	DoA T5.1, T5.4 UCG-16-04, UCG-19-04
<p>The crawling process mainly focuses on surfacing information at two different levels of abstraction. At the lower level, it aims at discovering cyber-threat intelligence (that will be at a later stage leveraged to actionable intelligence) for the IoT domain, while at a higher level it aims at identifying new promising seeds for widening the crawl frontier. Thus, classification methods will be applied at both levels: classification of crawled pages will help identify the ones that will contain useful cyber-threat intelligence, while site classification will be used to enrich the crawl seeds and widen the crawl frontier.</p>	Information classification	DoA T5.1, T5.4 UCG-16-04
<p>Data collected during the crawling process are contextualized and subsequently leveraged to create actionable cyber-threat intelligence using an ensemble of machine learning techniques. Initially parsing, and clustering/classification techniques will be applied to contextualize the collected data, while appropriate machine learning techniques will be used to leverage the data to cyber-threat intelligence. The extracted intelligence will also be appropriately annotated with meta-information such as the credibility of the source and the level of confidence.</p>	Information extraction	DoA T5.1, T5.4 UCG-16-04
<p>Identified cyber-threat intelligence, alongside meta-information such as the credibility of the source and the level of confidence, is stored in the eVDB and is thus made available to the rest of the Cyber-Trust platform.</p>	Storage	DoA T5.1, T5.4 UCG-16-04, UCG-06-04
<p>The function of this service is supervised by an IT expert to ensure that it monitors credible and information-rich sources. The main functionality encompasses:</p> <ul style="list-style-type: none"> - the supervision of the crawling process regarding efficiency and effectiveness, - the management (e.g., addition, approval, annotation) of new seeds and the evaluation of 	Component management	DoA T5.1, T5.4 UCG-16-04, UCG-19-04,

<p>existing seeds, - the tuning of the crawl parameter to improve efficiency and effectiveness, the anonymisation/aggregation of collected content before presentation (calculation of top-ranked terms/tags, anonymisation of text snippets, removal of user-related information).</p>		<p>UCG-06-06</p>
---	--	------------------

16.3 Key Quality attributes

Table 16.2: Quality Attributes

Key Quality Attributes
<p><u>Configurability</u></p> <p>The service should be configurable enough so as to allow expert users to modify key software parameters, threshold values, and logic structures that define the behaviour of the service without resorting in the modification of the code. Moving such attributes out of the code will increase the reusability, flexibility and maintainability of the service without compromising any functional requirements. Configurability will allow for easy supervision and tuning, while along with customizability are key quality attributes for the Crawling service.</p>
<p><u>Performance/Scalability</u></p> <p>The performance of this component will be dependent on a number of attributes such as the allocated computational power on the server or VM hosting the Crawling Service, the available network resources, the internal architecture of the component and the deployment choices. As the number of crawled/monitored is expected to steadily increase appropriate measure will have to be taken for accommodating either vertical or horizontal scaling techniques. System requirements that incorporate attributes like computational power or network resources are expected to be finalized at a later stage in the project, typically alongside the deployment plan of the Cyber-Trust platform.</p>
<p><u>Effectiveness/Efficacy</u></p> <p>The effectiveness of this component, i.e., the ability to produce desired output, will be dependent on the technologies and tools that will be used for (a) acquiring, parsing and classifying data, (b) extracting, mining and contextualising meaningful information from the acquired data, and (c) leveraging the extracted information to actionable intelligence. It is also important to note that combining the best technologies from different fields is not a guarantee for the effective operation of the component as the interdependencies between the tools and techniques are non-trivial and may distort the final outcome. This is a key direction in T5.1 that involves the design of the component at hand. Finally, the efficacy of the component, i.e., the ability to produce the maximum achievable output, is also important in this context as the component aims at producing output that is greater than the sum of its parts.</p>
<p><u>Extensibility/Expandability</u></p> <p>The extensibility of the Crawling Service mainly refers to the systems design principles that should be followed to accommodate future developments (e.g., addition of different types of information sources), rather than to the systemic measure of the ability to extend the component (e.g., level of effort). To this end, both architectural and technology decisions have been focusing (e.g., on D2.2) and will also focus (e.g., within T5.1) on thoroughly reviewing possible directions and performing an informed decision regarding the key component modules. In this way the Crawling Service is designed as a modular component, with an internal structure and dataflow that is minimally (if not at all) affected by new/modified functionality.</p>
<p><u>Privacy-awareness</u></p> <p>Since the amount of information being crawled is vast and crawling is automated, it is possible to (accidentally) gather personal data. To avoid accidentally exposing such data appropriate data anonymisation/aggregation techniques will be applied. Moreover, the component will specifically focus on publicly available information, taking specific precautions to avoid infringing intellectual property rights or violating the terms of use of the specific websites (see also D3.1).</p>

16.4 Open concerns

- Selecting the proper mix of tools and technologies to deliver efficient and effective services is under consideration as part of T5.1.

- System specifications (e.g., computational power, network resources) for supporting high-throughput, high-quality cyber-threat discovery.
- Avoiding the collection (if possible) and exposure of personal information is a research challenge.

17. Smart Device Agent (A03m, A05m, A08m, A12, A14)

17.1 Responsibilities

The Smart Device Agent (SDA) is component running on end-user devices. This component will service a range of smart devices and will be responsible mainly for two modes of operation: one being for continuous/ real-time operation and the latter for ad-hoc operation when the circumstances call for it. The SDA is responsible primarily for the monitoring of device's usage, critical files and network transactions, and secondly for the application of mitigation policies and remediation actions.

17.2 Key Functionality

Table 17.1: Key functionalities

Key Functionality		
Description	Category	Ref
Installation/deployment and activation of the SDA on the registered device. Depending on the type of the end device setup and activation will follow.	Setup & Initiation	DoA T6.2, UCG-01-01, UCG-01-02
<p>The SDA is responsible for a number of operations aiding towards the early detection and prevention of compromise:</p> <ul style="list-style-type: none"> - Monitor device critical OS files/ vulnerabilities: event-based monitoring when critical OS files become subject to change. In this instance, a relevant message will be sent to the profiling service for investigation on whether this action is legitimate or a result of the malicious intervention. Additionally, the device is scanned for open ports and malicious active processes. Information is synced with the Profiling Service. In case any attempt is made in modifying the state of the device, backend services are triggered to check the status of vulnerability. - Check Device Patching status: when a new patch or firmware update becomes available for an actively registered device, the profiling service notifies the SDA and requests checking whether the end device is up-to-date. - Ensure Firmware integrity: A backend service runs between the SDA and Profiling Service to ensure that activated devices operate with the vanilla firmware. In case a fraudulent or altered firmware is detected then backend services for DPI are triggered for remediation and mitigation actions to take place. - Monitor activity on the device: this functionality involves the monitoring of communication and data transactions on the monitored device. It also involves the logging of key device information such as CPU and memory usage as well as running processes. 	Continuous Monitoring	DoA T6.2, UCG-07-01, UCG-07-02, UCG-07-03, UCG-09-01, UCG-09-02
<p>When the Cyber-Trust backend services have detected and verified an attack which could endanger monitored devices then the SDA is instructed to apply mitigation and remediation actions as necessary.</p> <ul style="list-style-type: none"> - Apply Mitigation Policy: At device level the decisions taken at the network level by the Cyber Defense Service are applied. - Remediate Device: This functionality involves restoring a device to a healthy state. Remediation takes place once the detection of an attack is confirmed meaning that either abnormal device behaviour is detected or the existence of fraudulent content is identified. The remediation process involves the isolation of affected files and their recovery to a previous healthy state or the notification of the end user about advised actions 	Mitigation & Remediation	DoA T5.3, T5.4 T6.2, T6.3 UCG-17-01, UCG-18-01

17.3 Key Quality attributes

Table 17.2: Quality Attributes

Key Quality Attributes
<p><u>Performance</u></p> <p>Different flavours of the SDA will be developed to capture enough smart device types. Due to the heterogeneity of capabilities of the targeted end devices the performance of the SDA may be affected (i.e. not all functionalities might be offered to all smart devices). Parameters that will impact this are:</p> <ul style="list-style-type: none"> • Operating OS and capabilities profile • Supported communication technologies and protocols • Average response time: The amount of time that the SDA takes to respond to a request. • Concurrent processes: The maximum number of concurrent processes allows to run at any time on the host device. • Requests per second: The maximum number of requests that are host device may service <p>Throughput: Indicates the amount of information SDA can process in a set period of time.</p>
<p><u>Reliability</u></p> <p>Because the SDA targets applicability to a wide range of smart devices, the operating environment can be vastly different and diverse. The reliability for the execution of operations might be affected by the underlying hardware and exposed interfaces.</p>
<p><u>Security</u></p> <p>Secured communication will be supported between the SDA and Cyber-Trust backend services. However restrictions may apply with low-end devices that do not support trusted communication channels or encrypted communication.</p>

17.4 Open concerns

- IoT devices adhere to vast and diverse communication protocols, the implementation methodology of the SDA needs to adapt to these specifications
- The satisfactory provision of envisioned functionalities depends on the capabilities of the end devices.
- Development and maintenance of different versions of the SDA to capture applicability to enough smart devices will be a challenge and therefore prioritization needs to be followed.

18. Smart Gateway Agent (A03g, A04g, A05g, A08g, A11, A13)

18.1 Responsibilities

The Smart Gateway Agent (SGA) is a Cyber-Trust component running on a network gateway at the user premises. This component will service a range of smart gateways and will be responsible mainly for two modes of operation: one being for continuous/ real-time operation and the latter for ad-hoc operation when the circumstances call for it. The SGA is responsible primarily for the monitoring of network traffic, and secondly for the computation and application of mitigation policies and remediation actions.

18.2 Key Functionality

Table 18.1: Key functionalities

Key Functionality		
Description	Category	Ref
<p>Network-based health check</p> <p>The SGA is responsible for a number of operations aiding towards the early detection and prevention of compromise on network level:</p> <ul style="list-style-type: none"> - Monitor device at the gateway (network traffic filtering): The gateway is running network intrusion detection system (NIDS) to check for signatures and anomalies based on the signature. In this instance, the system checks each packet passing through the SGA to detect any potential attacks. It uses a network probe to capture raw packet data as the network probe retrieves packet information such as source and destination IP address, source and destination ports, flags, header length and checksum. Then, the SGA compares the packets information with known attack signatures to identify potential threats. Finally, a report of the attack is issued to the responsible security officer. - Capture and classify network packets (DPI): the SGA automatically captures and classify the packets based on their contents; this can achieve using Deep Packet Inspection (DPI) approach. The DPI will characterise the packets into various categories such as benign, anomaly, suspected. Type: Network use case - Detect network attacks: the SGA is utilised to monitor packets on the network in order to detect the attacks and malicious threats in the network. SGA compares packets signature with against a database of signatures or attributes from known malicious threats. - Gateway Network Device Profiling: The SGA module profiles traffic for each device the user accepted the terms and conditions to allow the necessary monitoring [A03g] to detect abnormal traffic. 	Continuous Monitoring	DoA T6.3, UCG-08-01, UCG-08-02, UCG-09-05, UCG-10-05
<p>Discovery & Intelligence:</p> <p>The SGA participates in the Discovery & Intelligence of ongoing attacks on network level:</p> <ul style="list-style-type: none"> - Discover network: The exploitation of the Cyber-Trust device profiles conjoined with location information to allow for support to visualisation capabilities [A01], wither via dynamic (flow) or static (GID) graphs. - Receive intrusion detection system(s) alerts: In case of an attack discovery, or a false alarm event, the intrusion detection system is activated and generates alerts [D2, D4] and informs the iIRS [A13]. 	Discovery & Intelligence	DoA T6.3, UCG-16-02 UCG-16-03
<p>Defining mitigation policies and parameters</p> <ul style="list-style-type: none"> - Characterise devices' importance: Users prioritise the attributes of the devices and the services they own according to their preferences. This information (i.e. which exploits to block, etc.) is used for optimizing the defense actions to be applied and to 	Mitigation	DoA T5.3, T5.4, T6.3 UCG-04-02

<p>maximise users' satisfaction.</p>		
<p>Computing mitigation policies and attack surface</p> <ul style="list-style-type: none"> - Compute cyber-attack graphical security model: Create an attack graph that presents how exploits relate to security conditions. These conditions denote system attributes and represent the attacker's capabilities. Information about the exploits is retrieved by the eVDB and attacker's privileges on hosts, the profile of devices from the profile DB, and information about network hosts, connectivity, hosts' trust relationships, etc., from the network assets DB. - Compute attack's likelihood and success probability: The system consults the Profile DB to get information about the CVSS score (and other info) associated with a device's vulnerabilities (or it contacts the eVDB if this is not available) and use it for computing attacks' likelihood and success probabilities. - Compute device risk level: The TMS computes a new value for the risk level of a device. Information about the current device trust level, the current status of network attacks and network traffic related to the device (as compared with the baseline), the device vulnerabilities and their exploitability, the device health level and views of peer-level TMSs are considered. - Compute a belief on current security status: The alerts provided by the IDS are used to update the belief about the system's security state. The belief is a probability distribution over all the possible security states indicating the system's comprised security conditions. - Compute optimal intrusion response actions: Optimal defense action is computed based on information about the system's security state and the attacker's profile. Optionally, an action is suggested to the user, before being applied automatically. 	<p>Mitigation</p>	<p>DoA T5.3, T5.4, T6.3 UCG-15-01 UCG-15-03 UCG-15-02 UCG-15-04 UCG-18-05</p>
<p>Computation and delivery of trust and risk assessments</p> <p>The TMS undertakes the task of computing the trust level of devices (i.e. the degree to which a device is considered trustworthy) as well as the respective risk level (i.e. a measure of the potential demotion that known assets may sustain due to the current state of the respective device). In this respect, the TMS needs to:</p> <ol style="list-style-type: none"> 6. Collect a comprehensive view of the current state the device's security aspects (including its integrity status and security controls), its behavioral patterns (e.g. attacks it has participated in, alignment with nominal network flow patterns), as well as the associated risk (which vulnerabilities is the device exposed to and what is their expected impact). 7. Regarding the expected impact, in particular, the TMS will examine not only the direct demotion of asset values hosted on the particular device, but also the potential to use the device under examination as a springboard for launching attacks to other devices (which possibly host assets with greater value). 8. Gather from peer-level trusted TMSs assessments of the devices' trust and risk levels. 9. Compute the overall trust and risk scores and store them in a database. 10. Accept and honour requests from other Cyber-Trust system modules for retrieval of the trust and risk level of designated devices. 	<p>Mitigation</p>	<p>UCG-13-02 UCG-15-02 UCG-13-01</p>
<p>Supporting mitigation and communication</p> <ul style="list-style-type: none"> - Receive intrusion detection system(s) alerts: In case of an attack discovery, or a false alarm event, the IDS generates alerts and informs the iIRS to be evaluated in order to infer the system's security state by considering possible miss-detections and false alarms. - Communicate iIRS actions to the user: After having computed the optimal defense action, the user is informed about every defense action having been applied by the 	<p>Mitigation</p>	<p>DoA T5.3, T5.4, T6.3 UCG-16-03 UCG-06-07 UCG-18-02</p>

<p>iIRS or about specific defense actions, e.g. the most critical ones.</p> <ul style="list-style-type: none"> - Retrieve mitigation policy information: Retrieve mitigation policy from the database on the threat detected. If no mitigation applies, then the default measure is to block connectivity. 		
<p>Enforcing Network Mitigation:</p> <ul style="list-style-type: none"> - When the IDS has detected and verified an attack which could endanger monitored devices traffic, then the SGA is instructed by the iIRS to apply a mitigation and remediation actions as necessary. 	Mitigation	DoA T5.3, T5.4 T6.2, T6.3 UCG-18-03
<p>Storage and retrieval of trust information</p> <p>The trust-related information (trust and risk assessments for devices) shall be stored within the trust database, to be later retrieved and processed or returned as a response to queries.</p>	Data organization and storage	UCG-13-02 UCG-15-02

18.3 Key Quality attributes

Table 18.2: Quality Attributes

Key Quality Attributes
<p><u>Performance</u></p> <p>Different flavours of the SGA will be developed to capture enough smart gateway devices types. Due to the heterogeneity of capabilities of the targeted gateway devices, the performance of the SGA may be affected (i.e. not all functionalities might be offered to all devices). Parameters that will impact this are:</p> <ul style="list-style-type: none"> • Operating OS and capabilities profile, most of which runs a modified UNIX-based kernel and has limited RAM • Supported communication technologies and protocols • Average response time: The amount of time that the SGA takes to respond to a request • WiFi capabilities and standards • Type of WAN Network Connectivity Interface, this include broadband, GSM or 4G connectivity • Concurrent processes: The maximum number of concurrent processes allows to run at any time on the gateway device • Number of active devices can be handled: The maximum number of devices that the gateway may service at the same time
<p><u>Reliability</u></p> <p>Because of the SGA targets applicability to a wide range of gateway devices, the operating environment can be different and diverse. The reliability for the execution of operations might be affected by the underlying hardware and exposed interfaces. Also, other external factors such as issues that are directly affecting wireless performance such as radio interference and coverage may exist.</p>
<p><u>Security</u></p> <p>SGA could have different authentication functions to maintain their integrity. There are several authentication standards for gateways when operates in Wi-Fi mode. Some use shared keys, others rely on unique MAC addresses or verifiable digital certificates. Secured communication will be supported between the SGA and Cyber-Trust backend services. However, restrictions</p>

may apply with those low resource devices that do not support trusted communication channels or encrypted communication.

18.4 Open concerns

- Android limitation for monitoring traffic
- Which method will be used along with Machine Learning for optimal detection
- Which IPS solution will be utilized to enhanced Anomaly Detection

Conclusions

In this deliverables D4.1 we provide the Architecture and Design Specifications V1, which contains the preliminary version of the architecture and the design specifications that have been selected to form the basis of the Cyber-Trust platform. Later in the project D4.4 (Architecture and Design Specifications - final) will follow D4.1 containing the "definite" version of the architecture.

The Cyber-Trust solution architecture is documented in a set of views, focused on effectively communicating the architecture to all stakeholders. Beyond these views, additional documentation is provided to complete technical systems design.

By pro-actively engaging all technical partners in early architectural decision making, we have been able to deliver D4.1 according to the original Cyber-Trust schedule. It forms a good basis for further elaboration and validation of the architecture during T4.3 (Rapid Prototype) and T4.4 (Mockups). To be able to keep up this pace, collaboration among the relevant consortium partners is crucial.

The next formal version of this document will be D4.4, which will give more complete, detailed and validated information about how the Cyber-Trust key concerns will be addressed. Crucial input will be delivered from each component owner-partner and relevant responsible contacts (research, technological and management). As an immediate next step the finalisation of roles and responsibilities of each component's different aspects will be done to ensure smooth and efficient delivery of the Cyber-Trust platform.

Annex: Architecturally significant requirements

Use-cases and scenarios

The table below summarizes the architecturally significant aspects that were identified in the various D2.3 use-cases. These aspects are grouped per D2.3 Use-Case-Group (UCG), and mapped to specific actors. Being architecturally significant, these requirements will be explicitly considered while designing the specific individual components and/or creating the overall solution architecture.

Table 0.1: Use cases - Architectural requirements

UC-Architectural Requirements (AR)			
D2.3 UCG	UC AR ID	Consolidated description	System Actor
01	01	Setup End Device (install device apps)	
	02	A12 is installed on the smart device by P2	A12
	03	On installation, A03 links to A12 to get monitoring information. A12 implements “active monitoring”, in addition to the passive monitoring done by A03. Is A03 installed on the Smart-Gateway.	A03, A12
02	01	Register(and login) Policies and Sessions (register people and devices (A06))	
	02	A06 provides web portal, used by all platform users. A06 acts as the central Cyber-Trust platform portal, users use to login	A06
	03	A06 used for registration, stores user and devices profiles (implements users sign-up and login). A06 has a dedicated database for storage. Device profiles are stored in A17.	A06, A17
	04	A06 acts as the central Cyber-Trust platform portal, users use to login and redirect to A07 and A15. The (generic platform admin portal, see SBS) Cyber-Trust website is composed of A06, A07 and A15.	A06, A07 A15
	05	User registration profile is stored into the eVDB which is part of A07	A07
	06	A06 links to A03, A17 profile settings determine A03 monitoring behaviour	A03, A06
	07	A06 connects to A02 for safe storage of specific user or device profile data	A02, A06
	08	A06 connects directly to A07 to register the user's device profile in the eVDB (A07). Indirectly this enables eVDB sharing (A09)	A06, A07 A09
03	01	Deactivation (unregister and logout) Policies and Sessions	
	02	A06, being the central registration and login component is also the central deregistration and logoff component. Architectural impacts are already covered in UCG-02.	A06
04	01	System Definitions (source to provide (anonymized device data) for investigation and defense	

UC-Architectural Requirements (AR)			
D2.3 UCG	UC AR ID	Consolidated description	System Actor
		purposes	
	02	A17, via A03, is linked to A11 and A12 to get data	A03, A11 A12, A17
	03	A17 is linked to A08, via A05. A08 uses the pseudonymized data	A05, A08 A17
	04	A16 includes asset characteristics importance, inserted via A05. A13 is connected to A16 to get this information to be able to determine iIR measures.	A05, A13 A16
	05	Mitigation actions are enforced by A04.	A04
	06	O2 consults the mitigation actions (A04), via a cyber-attack graphical security model (using info from A16).	A04, A16
05	01	PORTAL: Visualization Tools (Advanced UI to monitor the health status of devices and network and detect misbehaving devices).	
	02	A01 provides various network monitoring and health status views (2D and 3D). The data is provided by A03, A05, A07 and A16.	A01, A03 A05, A07 A16
	03	After A06 logon users can be directed to eVDB search module -> link to A07	A06, A07
	04	Crawled data (A10) will be enriched (information extraction) and stored in the eVBD, which is part of A07.	A07, A10
06	01	PORTAL: Decision Support and Alerting (Viewing and notification aimed to acknowledge the existence of vulnerabilities, using the eVDB, A07, as a primary data source. A01 as the provides the UI).	
	02	Alert is risen by the user via the intelligent UI (A01) based on monitoring info provided by A03. User is O2.	A01, A03
	03	A07 raises the alert (IoT device threat). This alert is provided to the user via the intelligent UI (A01). User is P1, P2.	A01, A07
	04	After threat detection by A07, A05 recalculated trust levels. Thereby, A07 is connected to A05 for sending triggers.	A05, A07
	05	A01 is connected to A09 to query for eVDB entries	A01, A09
	06	A13 alerts the security officer (O2) about optimal defense actions.	A13, A14
07	01	Host-based health check (scan devices for detection of possible vulnerabilities. Device level, eg: firmware updates registered in A07 can trigger. Critical device level information is stored in A05	

UC-Architectural Requirements (AR)			
D2.3 UCG	UC AR ID	Consolidated description	System Actor
		and partly on the blockchain (A02) for integrity preservation)	
	02	A03 knows firmware version by A11, A12 monitoring. A03 detects outdated firmware by checking with central patch database (A04, which also included the MUD service).	A03, A04 A11, A12
	03	A03 monitoring and A04 detection involve correlation of A16 (registration), A07(vulnerabilities) and A11 (network), A12(device) data.	A03, A04 A07, A11 A12, A16
	04	A12 is connected, via A03, to A01 to report health status. A12 alerts when the firmware is outdated.	A01, A03 A12
	05	A04 runs between hosts and devices information databases. Carries out defense operations. Provides MUD services and patchDB. MUD provides the normal/typical way of using a given device.	A04
	06	A05 calculates trust scores if A12 raises alerts.	A05, A12
	07	A05 trust calculation can trigger A04 defense operations	A04, A05
08	01	Network-based health check (Monitor real-time network traffic for all registered devices at the smart gateway level. Combine with UCG-01 and UCG-10 output to generate insights on behaviour trends and patterns)	
	02	A11 is running on the smart-gateway. A11 uses DPI to identify anomalies.	A11
09	01	Monitoring Activities (Active monitoring (in addition to passive monitoring done by A03). Monitoring IoT devices in the network at the smart gateway level, providing input for UCG-06 and UCG-10)	
	02	A12 and A04 manage device monitoring. A16 knows device types and MUD, so A04 and its realisations in A11 (NIDS) and A12 (HIDS) can detect threats.	A04, A11 A12, A16
	03	A03 uses A12 data to detect suspicious content and trigger backend service A04 for defense operations.	A03, A04
	04	A16 collects and provides information on the network's architecture including in-place defenses. This helps A03 and A12 to detect threats, also using A07.	A03, A07 A12, A16
	05	The end device is monitored (by [A03]) in terms of communication and data transactions.	A03
	06	On detecting anomalies, forensic data is stored (on and off chain) on A02 and A04.	A02, A04

UC-Architectural Requirements (AR)			
D2.3 UCG	UC AR ID	Consolidated description	System Actor
	07	A11 detects network attacks, utilising a network probe.	A11
10	01	IoT profiling and Data Analytics (IoT device data, gathered via monitoring (UCG-08 and 09), as base data for threat detection, stored in profiling repository A17. Device profile metric eg. CPU and memory consumption and running processes)	
	02	A03 gathers device and network level data, and is able to guide A11 and A12	A03, A11 A12
	03	Device metrics are synced and stored in A17 (aka CMS)	A17, A12
	04	A17 input is provided by A03 (and its realizations in A11 and A12 [monitored data]) and A06.	A03, A06 A17
	05	A04 receives data for improvement of system operation. This data is provided by A17.	A04, A17
	06	A01 receives data for visualization. This data is provided by A17.	A01, A17
	07	A05 receives data for trust level calculation. This data is provided and anonymized by A17.	A05, A17
	08	A08 holds the data (TrustDB). A05 does the trust level (re)calculation	A05, A08
11	01	Forensic Evidence Collection	
	02	A05 (low trust level) or A04 (abnormal behavior detected) triggers forensic evidence collection (A04).	A04, A05
	03	A03 provides forensic data (device and network level). Off-chain forensics are stored in A03 (Forensic Evidence DB) and on-chain in A02 (DLT). Both types of data are visualised via A01.	A01, A02 A03
12	01	Forensic Evidence Exploration and Export (provide validated DLT data for incident examination)	
	02	A01, as a UI, is connected to A02.	A01, A02
	03	A02 provides validation, making sure DLT data is not compromised.	A02
13	01	Trust Characterization (TMS (A05) uses IoT profiling data (A17), eVDB (A07) and network asset data (A16) to detect device vulnerability and triggers to calculate newest device level trust score)	
	02	A05 calculates device level trust scores	A05
	03	Trust levels are stored in A08	A08
	04	Intelligent UI (A01) can trigger trust level calculation and retrieve result (A05).	A01, A05

UC-Architectural Requirements (AR)			
D2.3 UCG	UC AR ID	Consolidated description	System Actor
		A01 can view trust levels (A08).	A18
	05	Data from A17, A16, A12 and A04 are queried by A05 to calculated trust level.	A04, A05 A12, A16 A17
	06	A05 uses information from A17, A13 and A04 to calculate trust levels. Defenses present in the network and on the device (known in A16) influence trust level.	A04 A05, A13 A17
14	01	Data Repositories and Correlation (keep mitigation policy db (A04) up to date, validate against A07. Compare A05 (monitoring target data source) and A07 to detect threats. Protect logs by storing parts of the A05 in the DLT, A02)	
	02	A11 and A12 monitor devices and detect abnormal behavior. A04 detects suspicious content.	A04, A11 A12
	03	A04 uses A02 for secure patch versions and forensic evidence integrity. Off-chain forensics are stored in the Forensic Evidence DB A04.	A02, A04
	04	Search capabilities and pub/sub mechanism is part of A09, connecting to A07 for eVDB access. Notification channels are email and portal UI.	A07, A09
	05	A03 monitoring status is visible in the visibility portal A01	A01, A03
	06	Mitigation policies are stored in A04. Trust levels are stored in the TrustDB, A08. Changes to the mitigation policies can impact device trust levels.	A04, A08
	07	eVDB (A07) device profile data (A17) is matched by A09. If needed an alert will be generated, indicating a security-related issue.	A07, A09 A17
15	01	Computation of attack surface and metrics (trust level is UCG-13, the risk level is UCG-15. Computation based on A07, A16 and A17. Risk level is linked to threats, info crawled and registered into A07. TMS (A05) updates info on strategies based on attack graphs)	
	02	A13 decision making is based on the cyber-attack graphical security model.	A13
	03	The cyber-attack graphical security model is created based on A07 and A16 data.	A07, A16
	04	A09 provides UI for making graphical security model computation. This model is build using A07 and A16 data.	A07, A09 A13, A16

UC-Architectural Requirements (AR)			
D2.3 UCG	UC AR ID	Consolidated description	System Actor
		This model results into A13 output	
	05	Risk levels are stored in A08 (TrustDB), computed by A05. The risk is computed per device.	A08
	06	A13 is alerted by A11 on abnormal behaviour detection.	A11, A13
16	01	Discovery and Intelligence (remote detection techniques enable profiling data (A17) to be updated. Crawling of dark/deep-web info to be informed about vulnerabilities)	
	02	A11 detects, A13 responds. A11 sends an alert (alarm) to A13.	A11, A13
	03	A17 holds device profile repository of all monitored devices, through remote detection techniques.	A17
	04	O1 user configures A10 crawling parameters	A10
17	01	Remediation Policies (E.g.: block malicious requests)	
	02	A03 monitoring, provided by data via A12, allows A04 to detect abnormal behaviour. A03 collects input data (forensics) to further handle this potential threat. A04 alerts A01.	A01, A03 A04, A12
	03	Remediation measures are taken by A04	A04
18	01	Mitigation Policies (take appropriate defense actions for prevention. Data source for mitigation policies is A04. Ability to or real-time response action selection, aim to mitigate attacks. Eg.: block mac-address network traffic)	
	02	Network level decisions are taken by A04 (detection and mitigation center). Device level response decisions are taken by A13 and A14. Handling the threat is coordinated via A01 and conducted by A04, via A13 or A14.	A01, A04 A13, A14
	03	Mitigation policies are stored in A04 and used by A05.	A04, A05
	04	A05, risk calculation, takes mitigation options (A04) into account.	A05, A08
	05	A04 retrieves appropriate mitigation policy, sends it to A05 (resp. A13, A14) to compute the mitigation actions, which are then applied by A04.	A04, A05 A13, A14

End-user requirements (functional)

The table below summarizes the architecturally significant aspects that were identified in the various D2.4 end-user requirements. These requirements referenced by an ID and mapped to specific actors. Being architecturally significant, these requirements will be explicitly considered while designing the specific individual components and/or creating the overall solution architecture.

Table 0.2: End-user - Architectural requirements (functional)

FR-Architectural Requirements (AR)			
D2.4 FR_ID	FR AR ID	Consolidated description	System Actor
01-11 40-42 70 83-84	01	Data about attributes and health status of networks and connected devices and users' actions must be stored and visualised in 2D and 3D. This includes present and historical data (e.g.: before, current and after mitigation).	A01
12-18 71	02	Forensic data will be stored on- (A02) and off-chain and visualised (A01). Off-chain data includes information regarding various data sources (A03-A07).	A01, A02 A03, A04 A05, A06 A07
19-28 88-90 92	03	Alerting is triggered by detecting vulnerabilities (A03m, A12). Alerting is filtered depending on importance relative to the device status (trust/risk score, A05). Alerting is routed depending on the type of equipment and user role (A05, A17) Alerting channels are: mobile-app-messages (A12), email, WhatsApp, SMS (A05) and visual-portal icons (A01).	A01, A05 A03m A05, A12 A17
43-48 91	04	A11 will collect structured DPI data for various types of data to be analyzed within the system (A08).	A08, A11
49	05	Forensic data file export function is part of the portal (A01). Data shown on the screen can be exported into CSV format to be analyzed in standard tools (EXCEL).	A01
50-53 57-63 69-77 85-87	06	UI WEB PORTALS: Multiple web portals provide various Cyber-Trust functionalities UI. Portals provide the UI . Underlying services provide API for the portals to use. These functions are granted via role-based authorization . REGISTRATION: Web portal (A06) provides UI and underlying services API (includes un-registration and login/logoff). VULNERABILITIES:	A01, A03 A04, A05 A06, A07 A09

FR-Architectural Requirements (AR)			
D2.4 FR_ID	FR AR ID	Consolidated description	System Actor
		<p>Web portal (A07) provides UI and the underlying services API for eVBD and management, A09 for eVDB (security issue's) info.</p> <p>NETWORK STATISTICS: Device and gateway monitoring services (A03) provides underlying services API. Statistics will be visualised in A03 (app) and A01 (web portal).</p> <p>TRUST MANAGEMENT Web portal (A05) provides TrustDB management UI. TMS (A05) provides underlying services API. Device trust information will be visualised on A01.</p> <p>MITIGATION Portal (A01) provides UI to create the cyber-attack graphical security model. Defense service (A04) provides underlying services API to enforce mitigation.</p>	
54-55	07	Network topology will be created by A16.	A16
56 64-69 79-82	08	<p>Mitigation rules determined by A04. Rules can be proactive (E.g., pushed auto updates). Based on and triggered by information from various data sources channelled via A08. Trigger can also be manual via UI (A01). Mitigation actions decided by A13, executed by A04. Trust scores will be subsequently updated automatically (A05).</p>	A01, A04 A05, A08 A13
64	09	<p>Alerting is triggered on new patch (A17). Alerting is routed depending on the type of equipment and user role (A17) Alerting channels are: mobile-app-messages (A12) and visual-portal icons (A01).</p>	A01, A12 A17
78	10	Tuning crawling parameters via dedicated UI (not A01).	A10

End-user requirements (non-functional)

The table below summarizes the architecturally significant aspects that were identified in the various D2.4 end-user requirements. These requirements referenced by an ID and mapped to specific actors.

Being architecturally significant, these requirements will be explicitly considered while designing the specific individual components and/or creating the overall solution architecture.

Table 0.3: End-user - Architectural requirements (non-functional)

NF-Architectural Requirements (AR)			
D2.4 NF_ID	NF AR ID	Consolidated description	System Actor
01-02	01	GDPR and EU-legal compliance by applying the recommendations from D3.3. Note: A16 will be selected from available tools. Take D3.3 criteria into consideration.	A02, A03 A16, A17
03	02	Apply authentication and role-based authorisation strictly.	A07, A10
04-06	03	Apply hashing and digital signatures to be able to validate data sender authenticity. Apply encryption to secure sent data. Use https for secure transport channel	A02, A05 A07, A16
07-10	04	Only collect/store data that is strictly needed. Take D3.3 consideration into account. Involve WP3 partners during component development (early).	A03, A04 A05, A06 A08, A17
11	05	Apply security view (chapter 7) guidelines and hardening practices on deployment units (Docker containers)	A02, A03 A11
12 14	06	Encrypt forensic off-chain data. Apply strict authorisation on the database storage system.	A06
13	07	Audit -logging on critical actions via portal and DLT activities	A01, A02
15	08	Terms and conditions disclaimer during app install	A12
19	09	Implement the right-to-be-forgotten on unregistering the user. Keep in mind the on-chain data cannot be deleted.	A06
33-35	10	Device agent will have rules-based configuration options for detecting anomalies. On detection, A04 will store metrics as/and evidence data.	A04, A12
36-37	11	“User-friendly UI” will be evaluated during Clickable Mockups (D4.3). Mockups should therefore represent envisioned UI.	A01, A02
38-39	12	A04 will contain the “ PatchDB ” off-chain part.	A04

NF-Architectural Requirements (AR)			
D2.4 NF_ID	NF AR ID	Consolidated description	System Actor
16-17	13	<p>Authentication and role-based authorisation for all Cyber-Trust modules.</p> <p>Involve WP3 partners during component development (early).</p> <p>Make sure not to put data on the DLT that needs to be deleted.</p> <p>Take D3.3 considerations into account.</p>	ALL
Other	14	<p>Some NFR's can be applied directly from D2.5.</p> <p>No addition comments needed or added value by grouping applicable from an architectural perspective.</p>	ALL