Integrated Framework for Predictive and Collaborative Security of Financial Infrastructures

Start Date of Project: 2018-05-01

Duration: 36 months

D2.4 FINSEC Reference Architecture I

Deliverable Details	
Deliverable Number	D2.4
Deliverable Title	FINSEC Reference Architecture I
Revision Number	3.0
Author(s)	GFT
Due Date	31/01/2019
Delivered Date	31/01/2019
Reviewed by	CINI & FUJI
Dissemination Level	CO
EC Project Officer	Christoph CASTEX

No.	**Contributing Partner**
1.	GFT (responsible)
2.	ATOS (contributor)
3.	IBM (contributor)
4.	FUJI (contributor)
5.	HPE (contributor)
6.	UTI (contributor)
7.	SILO (contributor)
8.	LIB (contributor)
9.	HDI (contributor)
10.	SIA (contributor)
11.	JRC (contributor)
12.	NEXI (contributor)
13.	WIRECARD (contributor)
14.	AS (contributor)
15.	CCA (contributor)
16.	INNOV (contributor)
17.	ZP (contributor)

18.	NRS (contributor)
19.	CNR (contributor)
20.	ORT (contributor)
21.	FBK (contributor)
22.	CINI (contributor)

Document Status

☒ draft ☒ WP leader accepted

☒ Consortium reviewed ☒ Project coordinator accepted

Revision History

Version	By	Date	Changes
1.0	GFT/INNOV	01/05/2018	Initial ToC, bullet points, Lorem Ipsum examples
1.1	GFT	27/12/2018	Contribution and Contents in co-authoring mode
2.0	GFT	18/01/2019	Complete Draft
2.2	GFT	23/01/2019	Revision for review
2.3	GFT	28/01/2019	Revision for QA
3.0	GFT	29/01/2019	Final version for submission

Abbreviations

AAA	Authentication, Authorization Accounting
ADC	Account Data Compromise
CA	Competent Authorities
CEBS	Committee of European Banking Supervisors
CERT	Computer Emergency Response Team
CMDB	Configuration Management Database
CIA	Confidentiality, Integrity and Availability
CSIRT	Computer Security Incident Response Team
CRUD	Create Read Update Delete
CSC	Common and Secure Communication
DPA	Data Protection Authority
DPO	Data Protection Officer
DSP	Digital Service Providers
EU	European Union
EUC	End-User Computing
GDPR	General Data Protection Regulation
ICT	Information Communication Technologies
IEC	International Electrotechnical Commission
IDS	Intrusion Detection System
eID	electronic Identification
eIDAS	electronic IDentification, Authentication and trust Services
eTS	electronic Trust Services
IaaS	Infrastructure as a Service
ISMS	Information Security Management System
ISO	International Organization for Standardization
ISP	Internet Service Provider
KYC	Know Your Customer
LOPD	Law for Protection of Personal Data
MiFID	Markets in Financial Instruments Directive
MSA	Micro Service Architecture
NDA	Non-Disclosure Agreement
NIS	Network and Information Systems
OES	Operators of Essential Services
PAN	Primary Account Number
PaaS	Platform as a Service
PCI DSS	Payment Card Industry Data Security Standard
PIA	Privacy Impact Assessment
PSD2	Payment Service Directive 2
PSP	Payment Service Provider
PSU	Payment Service User
P2PP	Peer-to-Peer Payment
RA	Reference Architecture
RTS	Regulatory Technical Standard
QTSP	Qualified Trust Service Provider
SCA	Strong Customer Authentication
SME	Small and Medium-Sized Enterprises
SA	Supervisory Authority

SECaaS	Security-as-a-Service
TI	Threat Intelligence
3DS	Three-Domain Secure

Note: other acronyms used in the rest of document will be fully explained before their use.

Executive Summary

The present document shows the first part of work done within FINSEC project to design and build a suitable Reference Architecture (RA) for the cyber & physical security platform to secure the Infrastructure of organizations in the Financial Sector.

The work on Reference Architecture has been conducted as team work that involved all Consortium beneficiaries. The work is also a continuous and incremental process and the present deliverable is a stable snapshot of the status of the RA at the date of submission. An updated version of this deliverable will be produced at month 18 of the project, taking into account all the improvements and due verifications over the period. Nonetheless, the actual RA fulfills the project milestone M2 fully specifying the RA of the FINSEC platform.

The FINSEC partners have selected a methodology to work on the RA, identifying it in the "4+1" architectural view model, which is presented in the document. The methodology is based on five different views, from which the structure of the system can be analyzed (logical view, process view, development view, physical view and scenarios). Moreover, it is demonstrated that all the functionalities of FINSEC environment are properly covered by this model.

The State of the Art analysis conducted underlined that some already existing Reference Architectures can give suitable input to FINSEC, such as the horizontal overlaying of RA over physical and cyber assets to be protected, and the use of Big Data infrastructures to support the functionalities of the platforms.

Relevant inputs to the task have been considered, in particular the input coming from use-cases considered in task T2.1 and a cross reference matrix

Finally, a multi-layer and high level reference view and a detailed logical view of the RA are presented. In particular, three main tiers have been structured (edge tier, data tier and service tier), containing the building blocks to perform the core services. The resulting RA ensures full communication and interaction between all the building blocks, from the lowest level (the field, communicating with the RA through probes) up to the application and visualization layers. Along with the edge and presentation tiers, the identified building blocks provide the functionalities of the FINSEC platform for a more general class of use cases. All the building blocks will be designed and implemented during the tasks belonging to work packages WP3, WP4 and WP5.

The feasibility of the proposed logical view has been proved by a complete mapping between the RA building blocks and the project use cases, whose functionalities will be assessed during WP6 activities.

TABLE OF CONTENTS

List of Tables

List of figures

1. Introduction

1.1. Scope and Purpose

This document represents the first mid-term outcome of task T2.5 *"Reference Architecture Specifications"* within the FINSEC project. The main goal of the report is to describe the Reference Architecture for the project, and the methodology adopted to build it. To this aim, the involved partners agreed upon a high level reference view and drew a logical schema for the FINSEC platform. The schema specifies all the building blocks that are needed to cover all the functionalities of the toolbox and services to be implemented according to the Description of Action.

The specifications of the Reference Architecture will be defined within the document: in particular, the requirements for each single building block will be formalized, together with the relationships and the data flows driving their integration.

Besides giving a high-level overview of the Reference Architecture, this deliverable aims at defining an overview for each element of the architecture, which will guide the partners during the next design and development phase.

The definition of the Reference Architecture takes into account other aspects relevant to the platform:

- the Probes that provides the collection of data from the physical and cyber worlds
- the Application Programming Interfaces APIs at the edge of the architecture
- Other applications (dashboard, collaboration, link with other information sources)

The collection and definition of the requirements for the Reference Architecture, contained within this document, also include the selection of the most suitable technologies to implement the different building blocks.

This document represents the basis for the implementation of the activities within task T2.5, which will end with the submission of the final outcome, i.e. the deliverable D2.5 *"FINSEC Reference Architecture II"* in Month 18.

1.2. Structure of the Document

Section 2 describes the methodology chosen to drive the activities towards the definition of the Reference Architecture: according to the Description of Action, a "4+1 view" methodology was adopted, in order to cover all the critical aspects completely.

In particular, this methodology defines a logical view, a process view, a development view, a physical view and the scenarios which characterize the Reference Architecture.

Afterwards, Section 3 deals with the State of the Art, the description of the most known and commonly used Reference Architectures for physical and cyber security systems, and a collection of the most adopted practices, together with the still unmet needs, that FINSEC Reference Architecture aims to meet.

In Section 4 there are a high level overview and description of the RA, detailed in Section 5, with a view of the logical design and the drill-down of the functionalities for each building block.

Section 6 maps the different use cases foreseen by the pilot activities on the RA building blocks involved in their deployment and testing actions. Finally, Section 7 concludes the deliverable and anticipates the work still to be done within task T2.5 and the implementation tasks.

1.3. Relevant Task input

Task T2.5 aims at defining the features and the structure of the Reference Architecture. For doing this, the task needs to get input information from some other tasks within work package WP2: in particular, task T2.1 *"Stakeholders Requirements and Use Cases"* contributes with the definition of the general requirements for the FINSEC platform from the end-user's point of view. Moreover, another important set of specifications comes from task T2.2 *"Review of Applicable Laws, Regulations and Standards"*, where the current international and national rules for physical and cyber security systems are analyzed, together with their impacts on the RA and its modules and functionalities.

Furthermore, task T2.3 *"Specifications for Integrated, Predictive and Collaborative Security"* defines the specifications for the toolbox, which are mapped on the different building blocks for their implementation.

Other project tasks will finally drive some specific modules of the RA and the detail features of their development on the future activities. This is the case of task T2.4 *"Integrated modelling of Physical and Cyber Assets"*, where the data model for FINSEC events is defined, thus driving the implementation choices for the data tier level building blocks in this architecture.

Finally, pilot activities are planned in task T6.1, which is thus involved in this information flow, giving the mapping of use cases on the FINSEC building blocks as an input for this report.

1.4. Relevant Task output

The main output of task T2.5 and of the present deliverable is the detailed specifications for the implementation of FINSEC Reference Architecture. These information will be preparatory to perform the core development activities within the project in work package WP3, WP4 and WP5.

The main outcome that will be given to the other tasks is the definition of each building block, and their relations. In particular, task T2.5 with the present document, will give input to tasks T3.1, T3.2, T4.1, T4.3, T4.4 and T4.5 (for the building block in probes level); to T3.2, T3.5, T4.1, T4.3 and T5.1 (for the data tier); to T3.3, T4.2, T4.3, T4.5, T4.6 and T5.4 (for the service tier); to T3.4, T4.7 and T5.3 (for the dashboard and the interface with outside world); to T5.2 and T5.3 (for infrastructural modules).

2. Methodology

2.1. Scope

In the following, a preliminary study to select the most suitable formal methodology defining the FINSEC Reference Architecture is presented. This study ensures that the partners follow a path with clear objectives and covering all the most important aspects of the final result of the project, i.e. the cyber-physical security platform with tools and technologies, integrated in a micro-services structure.

2.2. Definition of a Methodology for FINSEC Reference Architecture

One of the most important features of the FINSEC platform is its modularity and flexibility, or the capacity it must have to cope with a number of different situations taking place on different infrastructures and assets. In particular, one of the main consequences of this feature lies in the wide range of aspects of the platform, thus making the Reference Architecture something that can be structured, built and seen under different points of view.

Due to that complexity, the partners decided applying the "4+1" architectural view model as the most valuable methodology to cover all the needed aspects of FINSEC RA.

The "4+1" architectural view model[1] is a methodology to design an architecture for a software platform, having the main capacity of describing it from 5 concurrent "views".

These views represent different stakeholders who could deal with the platform and the architecture, from the management, development and user perspectives. The four main views which Facilitate the definition and design of the architecture are the logical, process, development and physical ones, while the "+1" view is represented by the use cases or scenarios, thus making this model an abstraction of the developed solution/platform and the basis for the development.

In the following, the meaning of the different views is explained.

- **Logical view:** it represents the range of functionalities or services that the system provides to the end users, and can be shown as block diagrams;
- **Process view:** it represents the system processes and data flows and how the different processes and building blocks communicate between each other, with details on the runtime behavior of the system. It can be represented by UML activity diagrams;
- **Development view (or implementation view):** it illustrates the software management aspects of the system, from the programmer's point of view. It is represented by UML components and package diagrams;
- **Physical view (or deployment view):** it describes the lower levels of the architecture, dealing with the physical infrastructures where the software components run and the physical connections between them. It can be represented by UML deployment diagrams;
- **Scenarios:** the architecture of the system can be explained from the end user's point of view too, with the description of a set of use cases. The use cases or "scenarios" aim at describing

[1] Kruchten 95 Kruchten, P.: "Architectural Blueprints - The "4+1" View Model of Software Architecture", Paper published in IEEE Software 12, November 1995, pp. 42-50

some possible functioning situations of the system and interactions between components. The validation and assessment of functionalities are usually performed by this view.

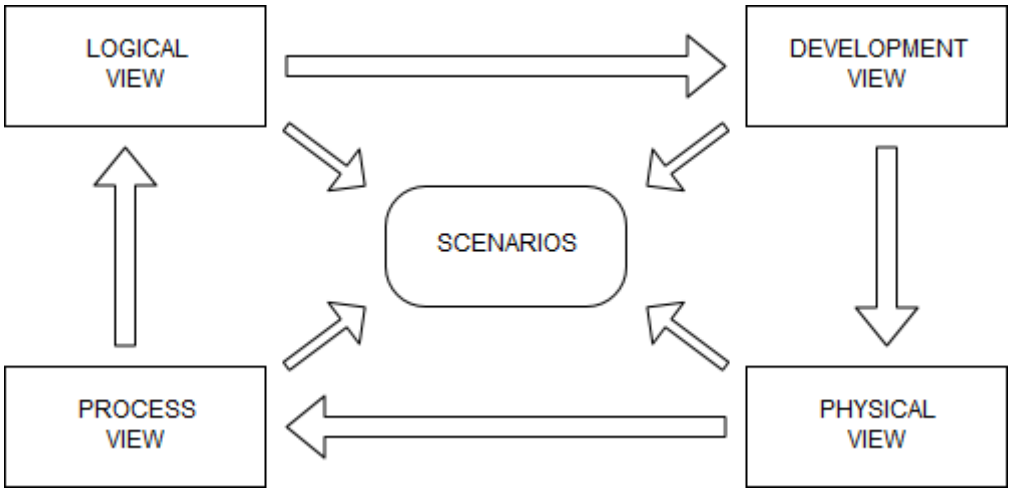

Figure 1: "4+1" views

In particular, for what the FINSEC project is concerned, the present document describes how the Logical view and the Process view were designed by the Consortium. The other views will be analyzed in the following tasks, in particular the Development view is the goal of work packages WP3 and WP4, while Physical view will be defined in WP5 and Scenarios in WP6.

Table 1 summarizes the main features and benefits of this methodology.

Table 1: "4+1" Pros and Cons

	Goal	FINSEC Tasks involved	Criticalities	Mitigation Strategy
Logical view	Describing the functionalities of the platform	T2.5	View defined in a preliminary way in an early phase of the project, quite far from the real implementation of the toolbox	Iterative definition and refining of the logical and process view until M18
Process view	Defining the interconnections between building blocks and runtime behavior of the system	T2.5	View defined in a preliminary way in an early phase of the project, quite far from the implementation and testing of runtime behaviour of the toolbox	Iterative definition and refining of the logical and process view until M18
Development view	Describing the system from the	T3.3, T3.4,	View of the RA built in a high number of	Strong internal coordination of WP3

	software programmer's point of view (technologies, technical details)	T3.5, T4.1, T4.2, T4.3, T4.4, T4.5, T4.6, T4.7	different tasks with many partners involved, with the risk of misalignments in the view itself	and WP4, with periodic communications and updates on the development of different modules and on the general strategy and goals
Physical view	Defining the physical layer of the architecture, the components of the hardware infrastructure and their integration	T5.1, T5.2 T5.3, T5.4	View of the RA built in a high number of different tasks with many partners involved, with the risk of misalignments in the view itself	Strong internal coordination of WP5, with periodic communications and updates on the deployment of different modules and on the general strategy and goals
Scenarios	Testing FINSEC platform functionalities	T6.2, T6.3, T6.4, T6.5, T6.6	High number of different use cases to be implemented to be started well before the end of tools development	Preliminary information on tools and technologies to be shared in advance during the project

According to the criticalities and the goals underlined in Table 1, the partners concluded that the "4+1 views" working methodology is fully suitable for the future development of FINSEC project

3. State of the Art

3.1. Reference Architectures for Physical & Cyber Security

In this section, various Reference Architectures (RA) for Security which pertain to the FINSEC RA specification activities are reviewed. Specifically, Internet of Things RA integrates information and services concerning the physical and cyber dimensions of systems. Likewise, security RAs and standards provide insights in the cyber and physical security aspects of the assets that comprise the systems of the financial sector. As such, these RAs are relevant to FINSEC and contribute to concepts and implementation guidelines for the development of the FINSEC Reference Architecture.

3.2. OpenFog RA

3.2.1. Overview

The OpenFog reference architecture (https://www.openfogconsortium.org/ra/) specifies a medium to high level view of system architectures for fog nodes and networks. It is specified by the OpenFog Consortium, which deals with specifications and standards associated with the fog computing paradigm. Fog computing defines system architectures that distribute computing, storage, control and networking functions closer to the users along a cloud-to-thing continuum. As such the fog computing paradigm has recently gained momentum for systems that combine information and services from cyber-physical systems i.e. systems and devices with both a cyber and a physical dimensions, which operate typically close to the field and their end-users.

The OpenFog Reference Architecture is illustrated in the Figure 2 and provides the structuring principles of fog systems towards achieving five properties that are collectively and conveniently called SCALE. In particular:

- Security: OpenFog RA specificies additional security for fog networks in order to ensure safe and trusted transactions.
- Cognition: Systems based on the RA ensure awareness of client-centric objectives to enable autonomy.
- Agility: The OpenFogRA makes provision for rapid innovation and affordable scaling under a common infrastructure.
- Latency: OpenFog RA emphasizes real-time processing and cyber-physical system control.
- Efficiency: The RA provides a means for dynamic pooling of local unused resources from participating end-user devices.

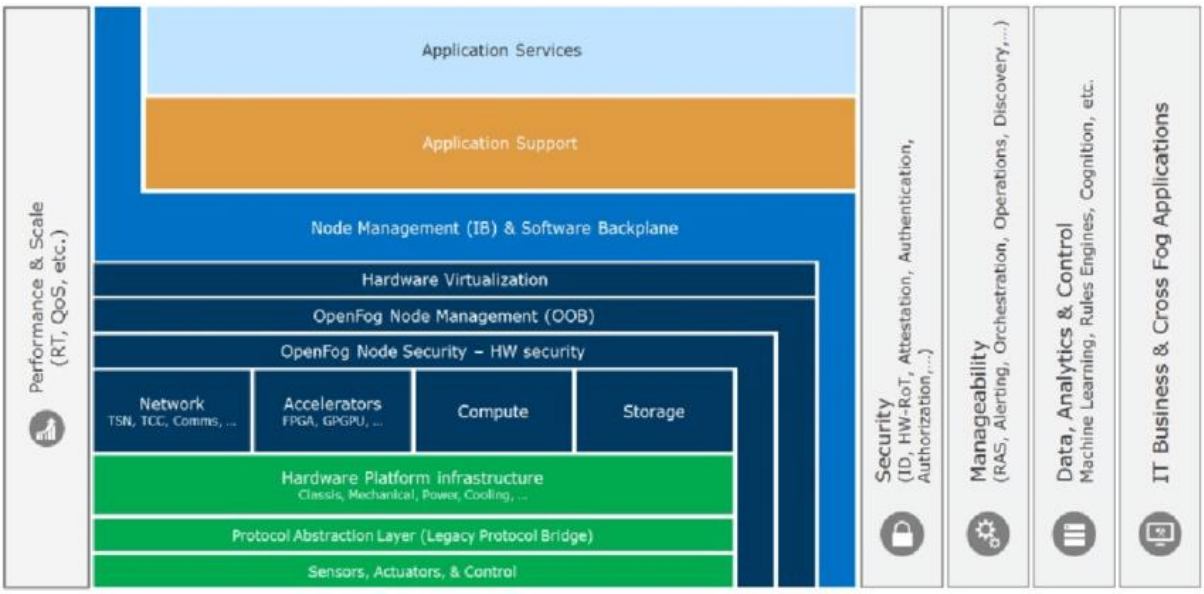

Figure 2: the OpenFog RA (source: https://www.openfogconsortium.org/ra/)

As illustrated in the Figure 2, the RA specifies the structure of fog computing systems and their functionalities in a layered format, which extends from hardware and network protocol functions up to application services. In this RA, Security is a cross-cutting function, which transcends both hardware nodes and their accompanying software backplanes.

3.2.2. Relevance to FINSEC RA

Security is one of the main pillars of the OpenFog RA, which defines the mechanisms needed to make a fog node secure from silicon to software application. Security functions must be applied across business cases, target markets, and vertical use cases, but also across the location of the node itself. The OpenFog RA also specifies that security implementations must cover various aspects such as privacy, anonymity, integrity, trust, attestation, verification, and measurement. Moreover, security must be end-to-end i.e. across OpenFog nodes, OpenFog networks, as well as OpenFog node orchestrations. Also, all the fog nodes must employ a hardware-based immutable root of trust, i.e. a trusted hardware component that receives control at power-on and accordingly extends the chain of trust to other hardware, firmware, and software components.

FINSEC is destined to protect both cyber and physical assets. As such the OpenFog RA is relevant to the FINSEC Architecture, as it addresses the security of cyber-physical systems structured in edge/fog computing configurations. The following elements of OpenFog RA are therefore considered in the scope of the FINSEC RA specification:

- Security as a Cross-Cutting, Overlay Function: Security in FINSEC will be implemented as an overlay function over financial cyber-physical systems and relevant assets. This shall be taken into account in specifying the logical view of the FINSEC architecture.
- End-to-End security: FINSEC shall support security end-to-end i.e. across all the systems and the devices that comprise the critical infrastructure of the financial institution. This shall be taken into account in specifying the logical and process views of the FINSEC architecture.
- Fog/Edge Computing Implementation: The FINSEC platform shall provide low-latency, security and real-time control functions and could therefore be implemented based on the edge/fog computing paradigm, much in the same way as OpenFog RA systems implement fog computing architectures. This shall be taken into account in the specification of the implementation and deployment view(s) of the FINSEC architecture.

3.3. IIRA and IISF

3.3.1. Overview

The Industrial Internet Consortium defined high-level structuring principles for Industrial Internet of Things (IIoT) systems (https://www.iiconsortium.org/IIRA.htm), which are detailed in the Industrial Internet Reference Architecture (IIRA). The IIRA emphasizes the implementation of systems with both cyber and physical parts, which are typical examples of OT (Operational Technology) and IT (Information Technology) convergence. IIRA suggests adherence to an edge computing architecture towards supporting scalable implementations of IIoT systems.

The IIRA is accompanied by a framework for IoT Security, which emphasizes IT/OT convergence and specifies various security functionalities, such as security configuration and management, security monitoring & analysis, protection of endpoints and more. The IIRA's security framework is called Industrial Internet Security Framework (IISF) and is illustrated in the Figure 3.

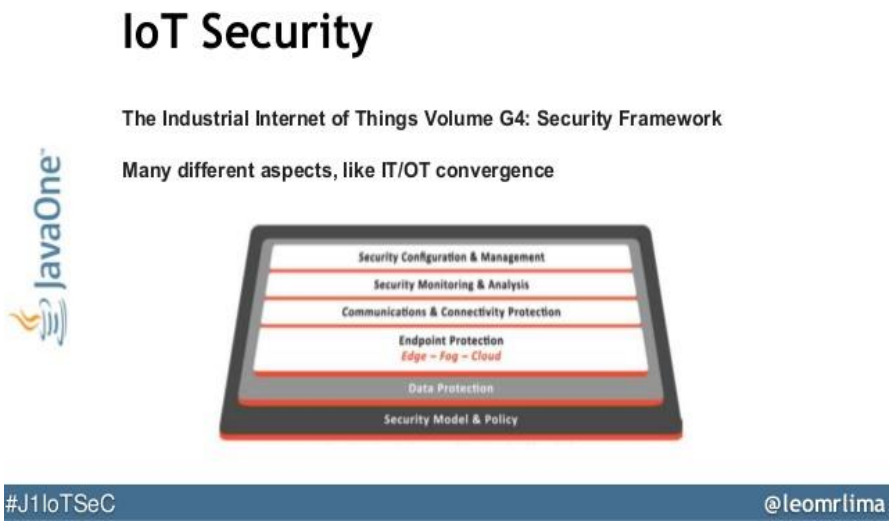

Figure 3: the IoT Security structure (source: https://www.iiconsortium.org/IIRA.htm)

3.3.2. Relevance to FINSEC RA

The IISF specifies security functions for IIoT systems that comprise cyber and physical assets. As such, it is relevant to FINSEC, which deals with integrated (cyber/physical) security for the assets of the critical infrastructures of financial organizations. Hence, IISF can influence the FINSEC RA in the following directions:

- **Horizontal Nature of the FINSEC Functionalities:** The specification of FINSEC security functions as an overlay and horizontal layer to all the functionalities of the FINSEC platform, which can be taken into account in the scope of the logical view of the FINSEC architecture.
- **Specification of FINSEC Functionalities:** IISF specifies the security functions that should be supported as part of an integrated OT/IT system, such as connectivity protection, endpoint protection, security monitoring and analytics, as well as security configuration. Security configuration and security monitoring functionalities should be supported by the FINSEC BigData platform for predictive and integrated security.

4. High Level Architectural Concepts

4.1. Driving Principles

Based on the review of the requirements and the reference architectures in the previous sections, this chapter summarizes some principles that drive the specification of the conceptual FINSEC architecture. Moreover, in the following subsections, some of the main building blocks of the architecture and the structuring principles between them are identified. The FINSEC architecture should adhere to the following principles:

- **Overlay and Cross-Layer Functionalities:** The FINSEC solutions will overlay functions running on top of existing systems for cyber and physical security. Therefore, the FINSEC architecture should specify cross-layer functionalities that will be provided on top of all the software, middleware and physical systems (e.g., devices, data centers, cameras) that comprise the critical infrastructures to be protected. Likewise, information from all these systems should be derived and processed by the FINSEC solutions.
- **Security Monitoring & Analysis:** The FINSEC DoA specifies the collection and the analysis of large amounts of security information as means of supporting its predictive security approach. This is in-line with some of the core functionalities of the presented reference architectures, such as the Monitor-Analyze-Act cycle specified in the IISF. Hence, the FINSEC architecture shall comprise data collection and analytics building blocks.
- **Multi-Tier Edge Computing Architecture:** The edge computing approach to the implementation of some of the above-listed reference architectures (e.g., OpenFog RA and IIRA) is relevant for the implementation of the FINSEC architecture and should be taken into account in the specification of the implementation and the deployment views. Note that the FINSEC architecture should comprise some cross-cutting functions (like the IIRA), such as visualization, configuration and management functionalities.
- **Blockchain infrastructure for Data Sharing:** Permissioned blockchain infrastructures provide compelling advantages when it comes to sharing information across different administrative entities. As such, they are a very good option for supporting the supply chain collaboration functionalities of the FINSEC architecture.

4.2. Structuring Principles and Building Blocks

In-line with the above listed principles, Figure 4 illustrates some of the building blocks that should be supported/implemented within the FINSEC architecture, along with their structuring based on a layered approach:

- **Monitoring Probes:** In order to collect security information, the FINSEC architecture should make provisions for monitoring probes on both cyber and physical security systems.
- **Legacy Security Systems:** The architecture shall include legacy security systems (notably systems of the project partners) that collect, analyze and persist security information & events (e.g., the SIEM and CCTV systems that will be advanced in work package WP4). Those systems could be classified as monitoring probes as well, but in this initial analysis they are listed separately as they typically provide a broader set of functionalities than simple probes.
- **Data Collection and homogenization:** in principle, FINSEC will be a data intensive system that will collect and consolidate security data from many different and heterogeneous sources (including multiple probes). To this end, it should offer as middleware the building blocks for data collection and unification from multiple different sources. Data unification refers to taking into account the heterogeneity of the different data models and formats that charactrerized the various data sources.

- **Data Storage (and Persistence):** A BigData infrastructure should be provided to ensure that very large datasets of security information can be effectively persisted.
- **Actuation and Automation:** FINSEC systems should facilitate interacting with the field and the systems of the critical infrastructure towards automating security actions (e.g., as part of the implementation of a security policy), as well as towards (re)configuring the operation of the probes. The reconfiguration functionality supports the implementation of intelligent and adaptive data collection through dynamically tuning parameters such as type, rate and amount of data collected.
- **Security Intelligence Kernel:** The security intelligence kernel is the building block that extracts security insights based on processing of the collected information performed by advanced analytics. The kernel may also interact with the FINSEC Security Knowledge Base.
- **Security Knowledge Base:** This knowledge base consists of readily available security knowledge, such as information about known threats, attacks, malware and more. It helps to resolve attack and threat patterns against known assets. The Security Knowledge Base will be primarily consulted by the security intelligence kernel and by the actuation & automation building blocks in order to resolve attack patterns at coarse and fine timescales respectively.
- **Dashboards and Visualization:** The architecture should facilitate the visualization of security information.
- **Risk Assessment and Compliance Auditing:** These two building blocks are already listed in the FINSEC DoA. They will be supported by the previously described building blocks of the FINSEC architecture and will be delivered as a service (i.e. based on a Security-as-a-Service (SECaaS) modality).
- **Supply Chain Collaboration:** This module will leverage information exchange/sharing capabilities of the FINSEC platform towards exchanging security information across financial institutions.

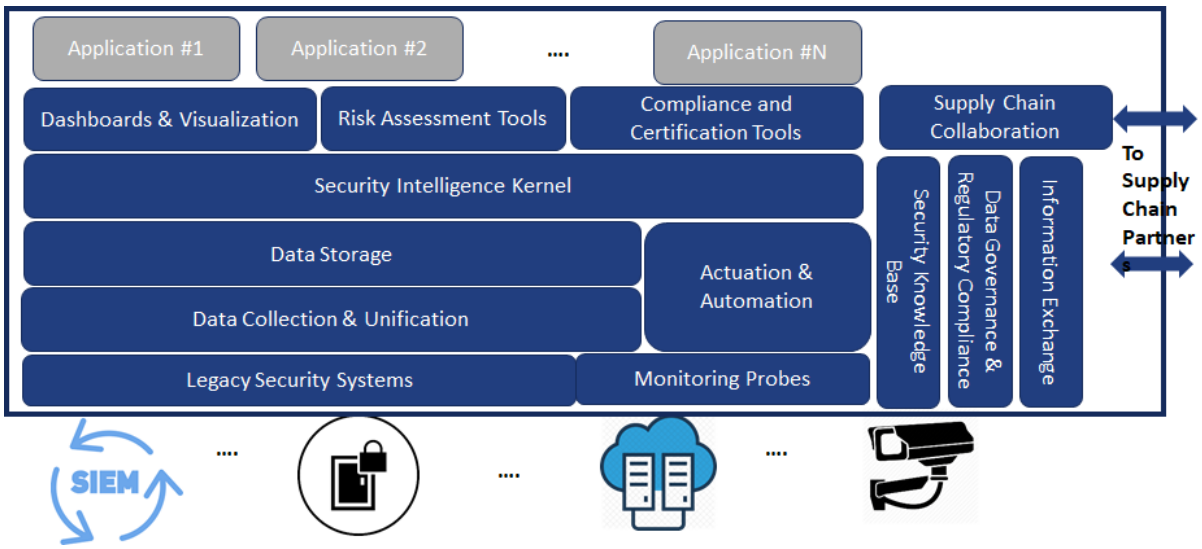

Figure 4: first version of the building blocks for FINSEC RA

Various applications (including the systems and the applications of the project use cases) will be implemented through leveraging the data intensive services of the FINSEC platform, including the risk assessment and compliance auditing services, the visualization services and more.

Note also that all the listed building blocks can be deployed and instantiated in a private cloud infrastructure for one or more financial organizations in order to enable the delivery of SECaaS services.

4.3. Data Collection Building Blocks

Figure 5 provides additional information about the data collection building blocks. Data collection should be based on streaming and batch processing middleware building blocks in order to support data collection from both streaming data sources and data at rest. Likewise, Figure 5 illustrates the use of a variety of data storage building blocks, such as data warehouses and data lakes, which shall hold, persist and manage security information. Note also that a regulatory compliance building block is envisaged in order to provide data governance functionalities, in-line with regulatory mandates such as GDPR and PSD2. For example, these mandates may include data encryption and anonymization.

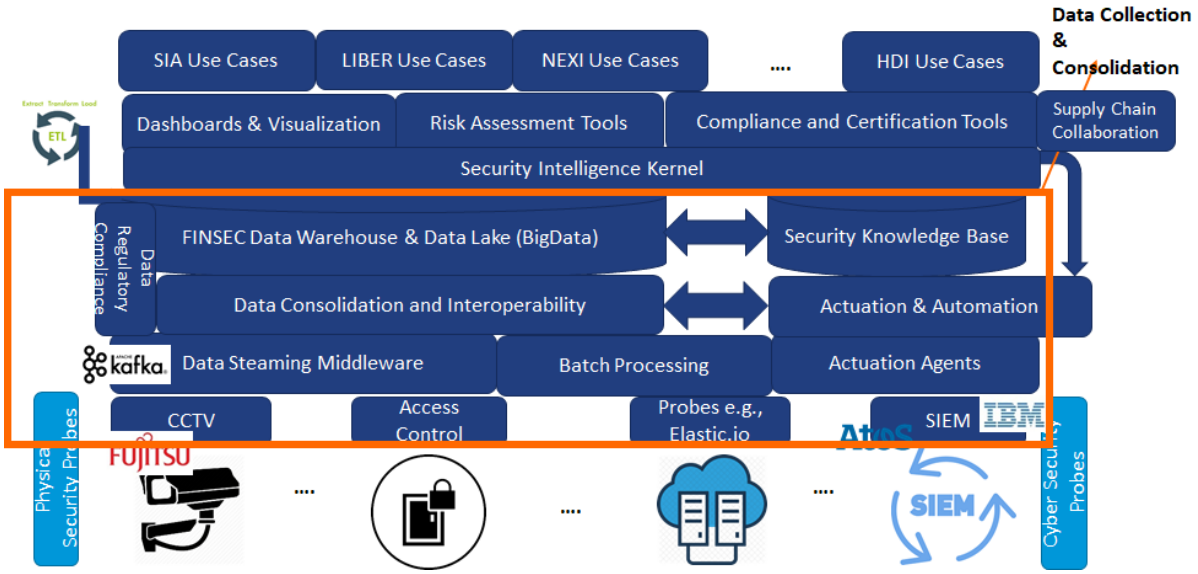

Figure 5: the Data Collection building blocks

4.4. Security Intelligence Kernel

Figure 6 illustrates the anatomy of the Security Intelligence Kernel. At its heart, lies a module based on advanced data analytics algorithms (such as machine learning and deep learning) used for extracting security patterns. Pattern extraction and its corresponding data processing algorithm should operate in a given context, such as location, time and thresholds stemming from specific business/security requirements. This context will configure the "Security Pattern Extraction" building block, which will be based on the "Patterns and Events Contextualization" building block.

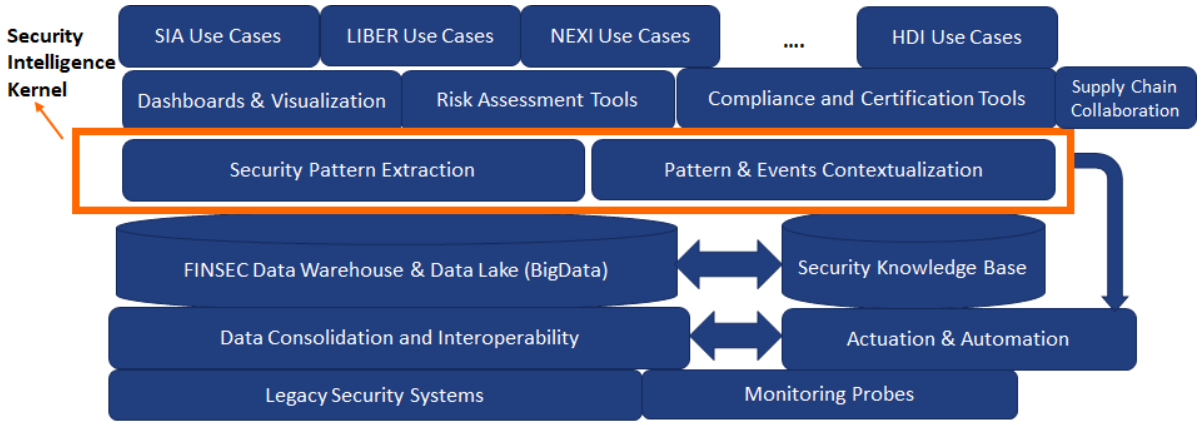

Figure 6: the Security Intelligence Kernel

4.5. Supply Chain Collaboration Concept

The FINSEC project will provide tools for stakeholders' collaboration in the financial supply chain to leverage the results of the previous tasks, while supporting assets modelling and interrelationships for financial services involving multiple participants, such as SWIFT network interactions.

Figure 7 depicts a collaboration building block based on a blockchain approach. The blockchain will be established and will operate in a fully decentralized manner, i.e. it will not be under the control of any administrative entity. Nevertheless, each participating entity (e.g., a bank or another financial institution) will interface to the blockchain infrastructure through a ledger client application, which shall make use of a blockchain API enabling the execution of the chain code. The entire data sharing process will be driven by a data model: the latter will specify the subset of security information that will be shared by each blockchain participant.

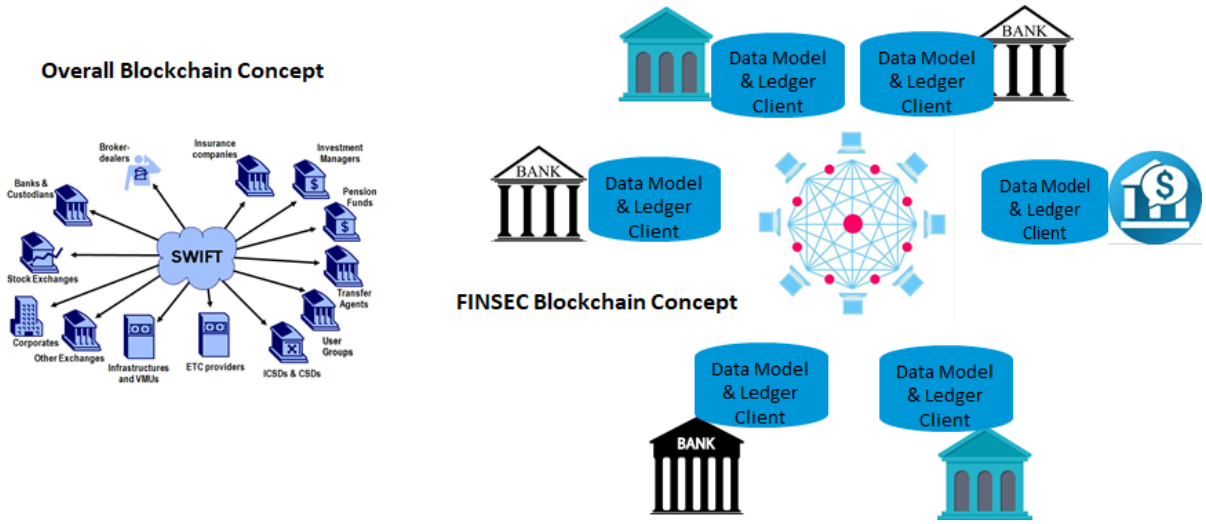

Figure 7: FINSEC blockchain collaborative proposed solution

The solution will provide a web-based tool for collaborative security processes, including assessment and mitigation, in the scope of the supply chain. The solution will also leverage the information sharing infrastructure developed in work package WP3.

5. FINSEC Platform Reference Architecture Logical View

5.1. Overview

The logical view of the high level Reference Architecture has been carried out to produce a more detailed view, in order to identify the specific building blocks that constitute the RA and the information flows among them.

In particular, a conceptual schema has been built by the involved partners, including different logical layers or tiers. This schema is referred to as the **Reference Architecture Logical Design (RA-LD)** and is coherent with the methodology followed by the FINSEC project and outlined in chapter [2] (**Logical View**).

In the following paragraphs, the basic principles and rationale that shape the logical design are described with more details.

5.1.1. Design Principles

In the process of refining the **Reference Architecture Logical Design** (RA-LD), some basic principles have been adopted to guide the design. These are listed in the following sub paragraphs.

5.1.1.1. One module should do one thing well

The Reference Architecture Logical Design will be defined in term of (services) modules. At the logical level, modules are black boxes with proper and well-defined interfaces that executes specific functions (business logic).

5.1.1.2. Every module should be defined in terms of its functionalities

In the Reference Architecture Logical Design, each module will be implemented as a manageable and independently deployable service component. In that respect, a module will follow a reference implementation.

5.1.1.3. Every module will expose a clearly defined Interface to other modules

Any module in the Reference Architecture Logical Design communicates with other modules via a well-defined set of Application Programming Interfaces. The definition of the API and the functionalities univocally characterizes a module (its behaviour, its communication means, the expected results, etc).

5.1.1.4. Every module will be assigned to only one partner for design and development

One module will have one partner as responsible. In other words, a partner will be responsible for the detailed design and implementation of its own software application services, although it will have other supporting partners for details specification, implementation, input of dataset, testing etc.

5.1.2. Synthesis

Albeit the FINSEC Reference Architecture can have multiple instances, being agnostic from implementation, the basic design principles suggest that the RA-LD could be easily designed to be implemented using a **Micro Service Architecture** (MSA). In particular, each module could be defined through its API (REST API) exposed to other services. These definitions will form the implementation view of the Reference Architecture and will be discussed in specific tasks/deliverables. A more advanced version of this document (deliverable D2.5 - Reference Architecture II) will be issued at month 18 to provide a coherent design of all the views of the methodology 4+1 described in Chapter [2].

5.2. Reference Architecture Logical Design

5.2.1. Assumptions

The idea behind the FINSEC project is to provide a solution to the problem of large enterprises and organizations, where usually the IT department doesn't share enough information with the physical security one (at least not before an incident escalates and they are required to do so). The result is that information from threats and incidents of one world (e.g. the physical infrastructure) may not be correlated with any information (incidents, threats or simply events) from the other world (e.g the IT infrastructure). Thus, the opportunity to obtain information that could be of use for a combined cyber and physical threat scenario, is missed.

In that sense, FINSEC is not a solution individually for the department responsible for physical security or for the IT department. The RA is supposed to be a solution offered to the top level management of large organizations (e.g. CSO or CEO), which is responsible of the security and the integrity of the company.

The FINSEC platform RA aims to provide a holistic approach to the overall security of theorganizations, especially in the Financial Sector.

The FINSEC platform is devised to be above the infrastructure (both IT and Physical) of a typical use case, e.g. of acompany organization which already has its own infrastructure and applications and procedures to monitor the IT assets from a cyber-security standpoint. In this scenario, the FINSEC platform does not substitute or duplicate the work of a probe, but rather uses any existing cybersecurity application (a SIEM, an enterprise antivirus application, a log scanning like fail2ban[2], etc.), to acquire data about incidents and alerts from the Logical Layer assets in the end users' Infrastructure. Where such tools do not exist, probes must be implemented specifically to retrieve data from logical assets such as servers and applications (logs, messages, etc.). In most cases, a SIEM can be an optimal solution and can be installed along the FINSEC deployment to provide information from the logical layer.

Accordingly, the physical end users' infrastructure will have its own applications and procedures: the FINSEC platform will exploit what is already in place (a Physical Security Information Management System (PSIM), a CCTV, an access control system, etc.) to obtain information of incidents coming from the physical world of the enterprise.

5.2.2. FINSEC Logical Design

This subsection presents the basic principles and explanations of the building blocks of the Reference Architecture logical view. The RA can be viewed as a tiered architecture, with the lower layer (the edge layer) that interfaces with the actual physical and logical infrastructures. Moreover, some cross-cutting elements have been added as infrastructural modules not belonging to one tier, but to the overall architecture.

However, according to the principles outlined in the previous paragraphs, the tier architecture is just a projection of a modular architecture, where every module can in principle communicate with any other module. In this respect, the architecture is closer to a modern micro-service architecture than a monolithic one. Figure 8 illustrates the actual stage of design of the RA.

[2] https://www.fail2ban.org

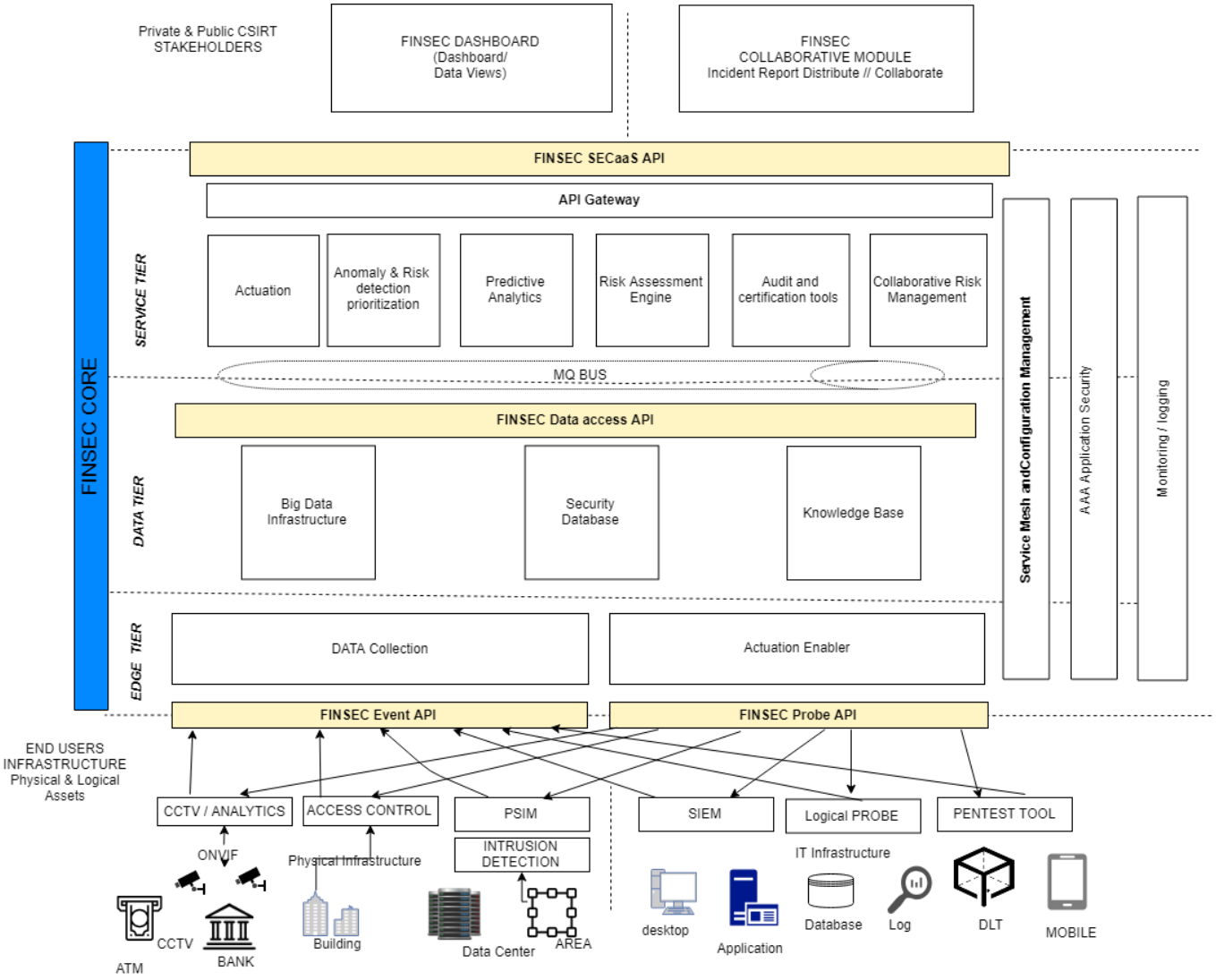

Figure 8: RA Logical view

With reference to Figure 8, the Logical Design can be viewed as an N-tier architecture composed by several layers:

- **Field Tier -** The **Field Tier** is the lower level and includes the probes and their APIs, whose role is extracting raw data from the physical and logical assets to be protected against threats. For example, CCTV analytics and SIEM are involved in this layer to give useful information about potential attacks to the upper tiers.
- **Edge Tier -** The **Edge Tier** contains the Actuation Enabler and a Data Collection module, which is necessary to filter the needed information during their flow towards the upper levels. The Actuation Enabler is responsible to allow some actions requested from the upper layers to be performed onto the probes, such as the shutdown of a server in case of threat or the closure of an automatic door of a protected room.
- **Data Tier -** The **Data Tier** is the logical layer where information are stored and is organized into three different storage infrastructures, providing consisting data access API to all other modules.
- **Service Tier -** The **Service Tier** is where the kernel applications of the FINSEC and the security toolbox will be running, also exposing the functionality to external entities.
- **Presentation and Communication Tier -** The **Presentation and Communication tier** offer interfaces to the rest of the world. This tier is where the dashboard will be provided to monitor data and where assets and the FINSEC Collaborative Module will be available to share information with other FINSEC instances.

The **Service Tier** defines the high level services that represent the "major intelligence" of the platform. The Service Tier services communicate with each other in 3 possible ways:

1. Synchronous communication: through their REST API (in this case, being the services internal to the platform, it is not necessary to use AAA);
2. MQ-based asynchronous access: via an MQ bus (in this case, therefore, queues and messages formats have to be defined);
3. DB-based asynchronous access: through the DB Infrastructure (presented below, in particular through the FINSEC REFERENCE DATA MODEL included in it).

The **Data Tier** provides an infrastructure to serve data that follow in the **FINSEC REFERENCE DATA MODEL** (defined by the project in tasks T2.3 and T2.4). It provides access in read/write via a Data Access API, exposed by an ad-hoc service of the platform (Data Manager). This module exposes convenient data access and manipulation functions to clients, is responsible for ensuring validation of input data against the data model and abstracts away the actual underlying DB engine(s), which can be changed without affecting upper-level services.

A possible alternative option, which allows to avoid an intermediate data access layer, is to use the CRUD REST API already exposed by the DB engine (if available, depending on the chosen DB engine) and to rely on DB validation rules for ensuring consistency and validation of data, with reference to the FINSEC REFERENCE DATA MODEL.

The FINSEC REFERENCE DATA MODEL can be used by the modules of the Service tier to communicate with each other (see above the 3rd mode of communication proposed).

In addition to data conforming to the common data model, the Data Base infrastructure may contain additional ad-hoc data stores for private data reserved to the individual Service tier modules, useful

for enabling their own internal logic. The concept in this case is that the individual service could still have a private DB schema for its own settings / local data, e.g. for processing with its own algorithms, and then proceeds to publish data on the common DB schema (via the Data Access API), following the FINSEC data model only once it has identified events useful for the common intelligence of the system, as previously mentioned.

The **Data Tier** provides the fundamental service for and will be based on:

- a **Data Base** suited to manage the non-structured Threat Information made by events, incidents, logs, etc. The task 5.1 on big data infrastructure will consider a noSQL DB (noSQL, memSQL) as well as SQL relational DB;
- a **BigData Infrastructure** to manage the large amount of data to be processed and distributed according to the requirements of the client modules, typically those ones of the Service tier needing BD / AI capabilities to perform their business logic. Deliverable D5.1 will identify the solutions to fulfil the requirement of this module;
- a **Security Knowledge Base** to be defined in task T3.5.

The **Edge Tier** communicates with the infrastructure (IT / Physical) through the southbound API interface. In fact such API consists on the union of two distinct APIs: **Event API** and **Probe API**. The Event API is implemented by FINSEC to receive events (possibly in push mode, to be evaluated if it is also needed in pull mode) and it is invoked by the probes; vice-versa, the Probe API is implemented by probes to receive commands from FINSEC (that in this case operates as an actuator vs probes, e.g. with actions, settings). In the latter case it is assumed that all the external products of the partners to be integrated are wrapped by PROBES (they're FINSEC plugins) that invoke the Event API.

Consequently, when probes want to publish data on the Data Tier (Data Base infrastructure and possibly ingestion in the Big Data Infrastructure), they have to send events based on the FINSEC Data Model. Instead Vs south (the integrated products), the probes communicate through an ad hoc interface depending on the backend product / resource that they have to integrate. FINSEC capabilities, as mentioned before, are exposed through proper **APIs** that guide both design and implementation. APIs should be designed according to principles of simplicity, elegance and coherence among the different FINSEC service modules. A consistent API design model will be leveraged and well known methodologies will be adopted.

The **FINSEC Core platform**, which is delimited by the blue bar in Figure 8, comprises three tiers, namely the edge, the data and the service tiers, which interact with the outside world with 2 main interfaces:

- The northbound API towards higher level applications (e.g. end-user applications), called **FINSEC SECaaS API** to align to the DoA and the core concept of FINSEC. It represents a consistent and unified view of the individual APIs exposed by the service tier high level services, which represent the "major intelligence" of the platform. The FINSEC SECaaS API is exposed by the API Gateway, which is the only entry point to the system for external clients. Among other capabilities, the API Gateway provides and supports Authentication, Authorization, and Accounting (AAA) services, which conceptually are part of the 2 vertical modules on the right side of Figure 8 (Application Security and Monitoring/logging).
- The southbound API interface, consisting of an **EVENT API** and **PROBE API**, allows communication between the Edge Tier and physical and cybersecurity probes.

The **FINSEC SECaaS API** is invoked by external (north end) Business Client Applications (upper side of Figure 88. The Business Client Applications are outside of the FINSEC core platform and interact with it only through the FINSEC SECaaS REST API.

Examples of Business Client Applications are:

- the **FINSEC Dashboard** application (developed and delivered by the project), which is a (WEB) GUI used by the profiled end-users of the platform.
- the **FINSEC Collaboration** application (developed and delivered by the project), which facilitates the collaboration of multiple platform instances (data sharing etc.).
- **the third parties applications** (in general not developed by the project itself but possibly provided by the project partners), which exploit the capabilities of FINSEC, for example the applications of WP6 end users' partners or 3rd parties (which can exploit the FINSEC market place, etc.).

5.3. Building Block as Service Application

The building blocks composing the logical architecture design respect the design principles: first of all, each of them performs an individual role and is conceived as a black box with proper interfaces executing specific functions; moreover, each module can be implemented as a manageable and independently deployable service respecting the micro-service architecture (MSA) paradigm and communicating with standard interfaces (REST API).

Finally, the design and the implementation of each single module is assigned to one responsible partner and will be detailed in the specific task and described in the corresponding deliverable.

Table 2 presents a list of the services supplied by the Reference Architecture, with high level descriptions and APIs used and provided. The definition of APIs and functionalities will be the objective for further developments in their specific tasks.

Table 2 – Building Blocks High Level description

Building Block	High Level Functional Description	API Provided	API Used	Task
FINSEC Dashboard	Web application that presents events, threats, incidents, logs, etc. in a User Graphical Interface. The will interact with the other micro-services to gather information to be presented to the dashboard graphically and intuitively.	API for asynchronous-Real Time messaging (web application provides push API).	Database CRUD API for security, knowledge base, incidents, threads, etc. Standard API for AAA, Log.	T5.3
FINSEC Collaborative Module	Service application for collaborative security information sharing and Threat Intelligence. The application will have a micro-service interface that will provide APIs for exchange information about threats and mitigations. Data will be based on the FINSEC Data Model.	API to create information about threats and mitigations.	Database CRUD API for security, knowledge base, incidents, threads, etc. Standard API for AAA, Log.	T3.4 T4.7
API Gateway	API Gateway is a fully managed service that provides to other services ways to create, publish, maintain, monitor, and secure APIs at any scale.	API to create, publish, maintain, monitor and secure APIs from other services. API to retrieve other service APIs.	Standard API for AAA, Log.	T5.3
Actuation	Application that offers API to other services to operate on the physical and logical infrastructure sending commands to the physical or logical components.	API for high level actuation commands (e.g. shutdown server, door closure…).	Database CRUD API for security, knowledge base, incidents, threads, etc. Standard API for AAA, Log.	T4.3

Building Block	High Level Functional Description	API Provided	API Used	Task
Anomaly & Risk detection prioritization	Application for anomaly and risk analysis. It consumes current data sources (logs, incidents, etc.) and produces incidents and alarms. Application consumes DATA Access API and pushes threat information using API of other services (eg. Dashboard, Collaborative Module, etc).	Std API only see below.	Database CRUD API for security, knowledge base, incidents, threads, asset modeling, etc. Dashboard/Collaborative Push API	T4.3 T4.2 T4.5
Predictive Analytics	Application that will analyze risk and threats from current data sources (logs, incidents, etc.) and predicts threats and patterns of threats. Application uses DATA Access API and push threat information using API of other services (eg. Dashboard, Collaborative Module, etc).	Std API only see below.	Database CRUD API for security, knowledge base, incidents, threads, asset modeling, etc. Dashboard/Collaborative Push API.	T3.3
Risk Assessment Engine	Application for real-time assessment of security risks, including business interpretation. It analyzes current model of assets associated with business risks levels stored as data model in DB and produces a risk assessment analysis. The Models are produced by the Audit and certification tool Application uses DATA Access API push threat information using API of other services (eg. Dashboard, Collaborative, etc).	Std API only see below.	Database CRUD API for security, knowledge base, incidents, threads, asset modeling, etc. Dashboard/Collaborative Push API.	T4.2
Audit and certification tool	Web Application with HMI to produce a data model representation of assets of the infrastructure. Application will be basically a Data Entry application plus input from other data sources. Application produces reports displayable and exportable (e.g. pdf).	Std API only see below.	Database CRUD API for security, knowledge base, incidents, threads, asset modeling, etc. Dashboard Push API.	T5.4

Building Block	High Level Functional Description	API Provided	API Used	Task
Collaborative Risk Management	Application for Risk Analysis & Management. Proposed implementation as Distributed Ledger with WEB interface.	API for other services to push threat information.	Std API only see below.	T4.6
MQ BUS	Asynchronous Message Passing Application. Provides Push/Pull API for basic message passing	API to pull message. API to push message.	Std API only see below.	T5.2
Security Database	A noSQL application for storing data according to the FINSEC Data Model.	CRUD API (create, read, update, delete) for storing STIX documents, Assets Documents, etc.	Std API only see below.	T3.2
Knowledge Base	A noSQL application for storing data according to the FINSEC Data Model. Provides CRUD (create, retrieve, update, delete) API for storing Knowledge Base documents.	CRUD API (create, read, update, delete) for storing knowledge base STIX documents, etc.	Std API only see below.	T3.5
Big Data Infrastructure	Distributed File System Application. Provides API for scaling data across multiple servers.	Std API only see below.	Std API only see below.	T5.1/T4.1/T4.3
DATA Collection	Application which provides API to EDGE services like CCTV or SIEM for pushing data (events, logs, etc.) to the Security Database. The application also performs normalization and prioritization to the information supplied by the EDGE applications.	API to insert STIX message.	Database CRUD API for security, knowledge base, incidents, threads, asset modeling,	T3.1/T3.2

Building Block	High Level Functional Description	API Provided	API Used	Task
Actuation Enabler	Application which provides API to the ACTUATOR service pushing action to the Logical and Physical infrastructure (e.g. shutdown of a server or close a door of a data center). The application performs abstraction and normalization to adapt to different EDGE components.	Specific API to operate on the actual infrastructure.	NO API based. Uses specific commands to interact with the actual infrastructure.	T4.3
CCTV/Analytics	Any Video Surveillance application that can be integrated will use FINSEC Event API to push information to the FINSEC core and provides FINSEC Probe API to interact with the EDGE components.	Std API only see below.	Database CRUD API for security, knowledge base, incidents, threads, asset modeling, Dashboard Push API.	T4.5
SIEM	Any Security Information and Event Management (SIEM) can be integrated as long as they will use FINSEC Event API to push information to the FINSEC core and provide FINSEC Probe API to interact with the EDGE components.	Std API only see below.	Database CRUD API for security, knowledge base, incidents, threads, asset modeling, Dashboard Push API.	T4.1
Logical Probe	Field-level logical sensor/actuator to give data on the status of the assets such as logs and actuation commands on logical assets (e.g., shutting down a server to protect it). Any probe can be integrated as long as it will use FINSEC Event API to push information to the FINSEC core and will provide FINSEC Probe API to interact with the EDGE components.	Std API only see below.	Database CRUD API for security, knowledge base, incidents, threads, asset modeling, Dashboard Push API.	T4.3
Pentest Tool	Service application acting on the field to extract information on the status of the assets. The Pentest is an assessment of the capacity of an asset to react to a penetration attempt by an actor, through the simulation of an attack. Service provides	API to start simulation of attacks. API to retrieve result of attacks.	Database CRUD API for security, knowledge base, incidents, threads, asset modeling, etc. Dashboard Push API.	T4.4

Building Block	High Level Functional Description	API Provided	API Used	Task
	APIs and will give information about the status of the asset and the attack typology under simulation.			
Service Mesh and Configuration Management	Service application that provides API to discover services within the infrastructure; moreover, its configuration will be done via this building block. Its APIs will need to give the user the current configuration for each service within the RA.	API for Discover, Configure, Set, Unset parameters of infrastructure.	Database CRUD API for security, knowledge base, incidents, threads, asset modeling, etc. Dashboard Push API.	T5.3
AAA Application Security	Service application that provides basic API for Authenticate, Authorize and Account users and other services within FINSEC platform. It is part of the "vertical" building blocks, not belonging to any specific tier.	Standard API for AAA.	Database CRUD API for security, knowledge base, incidents, threads, asset modeling, etc.	T5.3
Monitoring/ Logging	"Diagnostic" application, provides API for storing and retrieving logs from other services (e.g. connection/disconnection, search for services, use of certain services, getting warning and alarms, actuation on cyber or physical assets, etc.).	Standard API for Logging.	Database CRUD API for security, knowledge base, incidents, threads, asset modeling, etc.	T5.3
Std API ALL SERVICES	All services must provide the standard API to start/stop/ shutdown/monitor/status of the application.	API to start/stop/ shutdown/monitor/status of service. Push API to receive async message.	Standard API for AAA, Log. Database CRUD API for security, knowledge base, incidents, threads, asset modeling, etc.	ALL

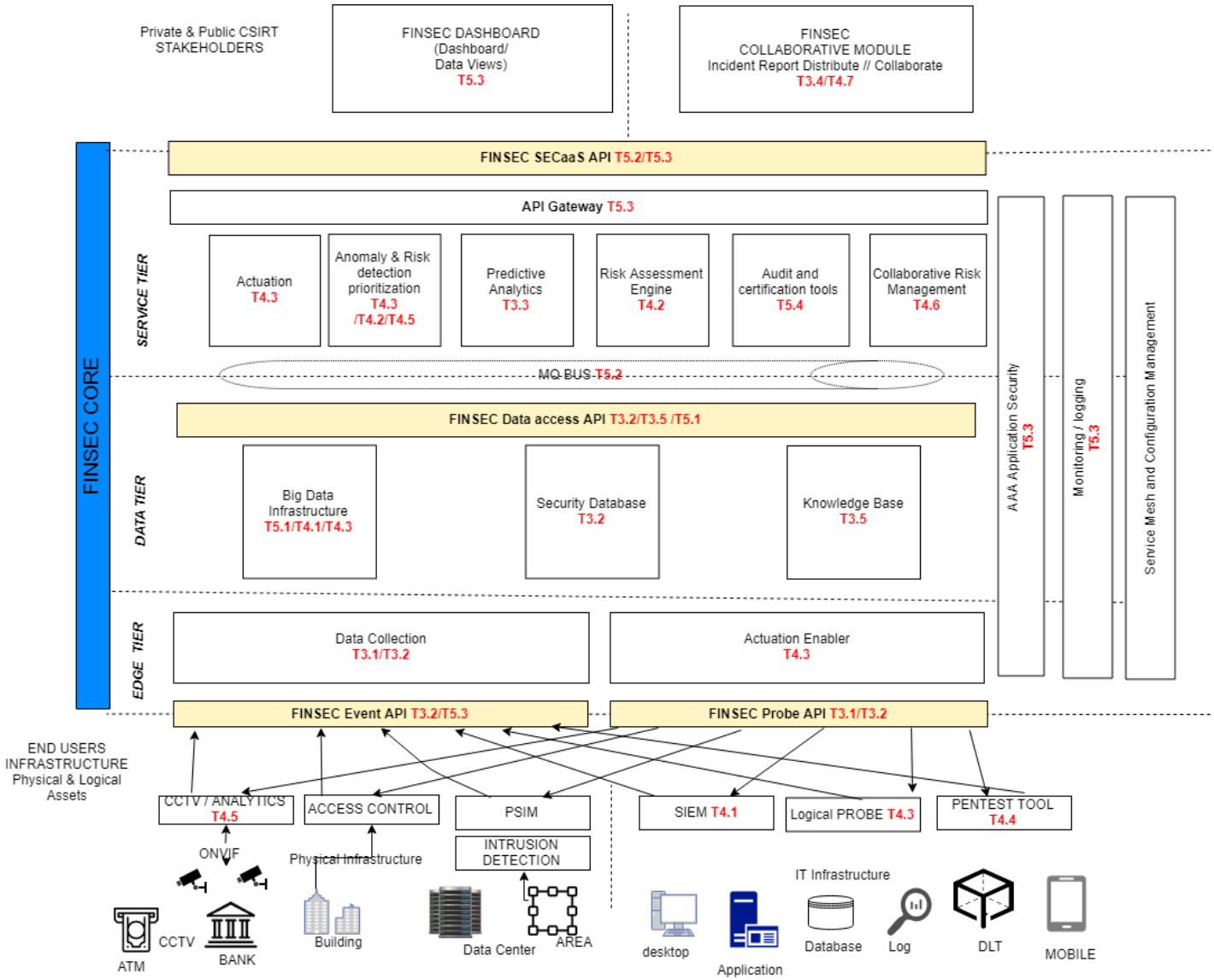

Figure 9: RA Logical view + Task assignment

6. Mapping of Use Cases to FINSEC Platform

The Reference Architecture presented in this document was designed with reference to the practical application of FINSEC toolbox, not only on specific pilots with their use cases, but also for a more generic use case. A number of foreseen scenarios will be validated. The functionalities will be assessed with reference to their capacity of detecting and defending the assets of an organization from cyber and/or physical attacks. However, the suitability of the Reference Architecture will be assessed, in terms of the building blocks on each single use case of the project pilots.

Table 3 shows the mapping of each use case on the Reference Architecture in terms of the specific building blocks involved to set up the tests.

Table 3: RA building blocks application on use cases

Pilots	Relevant RA Building Blocks
NEXI Pilot Use Case #1	CCTV Analytics, Data collection, Security Database, Anomaly & Risk detection prioritization, Predictive Analytics, FINSEC Dashboard.
NEXI Pilot Use Case #2	CCTV Analytics, Data collection, Security Database, Anomaly & Risk detection prioritization, Predictive Analytics, FINSEC Dashboard.
WIRECARD Pilot Use Case #1	CCTV Analytics, Data collection, Big data infrastructure, Anomaly & Risk detection prioritization, Predictive analytics, FINSEC Dashboard.
WIRECARD Pilot Use Case #2	CCTV Analytics, Data collection, Big data infrastructure, Anomaly & Risk detection prioritization, Predictive analytics, FINSEC Dashboard.
WIRECARD Pilot Use Case #3	CCTV Analytics, Data collection, Knowledge base, Big data infrastructure, Anomaly & Risk detection prioritization, Predictive analytics, FINSEC Dashboard.
SIA Pilot Use Case #1	Logical Probe, Data collection, Actuation Enabler, Big Data Infrastructure, Anomaly & Risk Detection prioritization, Predictive Analytics, Actuation, Audit and Certification tools, FINSEC Dashboard.
SIA Pilot Use Case #2	Logical Probe, Data collection, Actuation Enabler, Big Data Infrastructure, Knowledge base, Anomaly & Risk Detection prioritization, Predictive Analytics, Actuation, Audit and Certification tools, FINSEC Dashboard.
JRC Pilot Use Case #1	SIEM, Pentest Tool, Data Collection, Security Database/Big Data Infrastructure, Anomaly & Risk Detection prioritization, Predictive Analytics, Risk Assessment Engine, FINSEC Dashboard.
JRC Pilot Use Case #2	SIEM, Pentest Tool, Data Collection, Security Database/Big Data Infrastructure, Anomaly & Risk Detection prioritization, Predictive Analytics, Risk Assessment Engine, FINSEC Dashboard.
HDI Pilot Use Case #1	Logical Probe, Data Collection, Risk Assessment Engine, Anomaly & Risk Detection prioritization, Predictive Analytics.
SWIFT Pilot	The SWIFT use case will be finalized and presented in the next iteration of the Deliverable

7. Conclusions and Future Outlook

The deliverable presented the results of the task on Reference Architecture. The outcome of the task activity is a multi-tier design based on tiers providing core functionalities to presentation and edge tiers. The former lies on the lowest layer of the RA and deals with the collection of raw data from the field (physical sensors such as CCTV systems or logical probes) and with the communication of actuation security commands to the assets; the second tier has the role of storing the data into suitable infrastructures, according to the nature of the information and their potential use for the scopes of the project. The upper layer of the architecture contains the functionality exposed to the end users as micro-services, composing the toolbox that processes the data coming from the probes to detect and mitigate cyber or physical threats. These tiers are overlaid by an interface level containing the dashboard for the real time monitoring of critical parameters and warnings and a collaborative module to exchange information with other FINSEC instances. Moreover, some auxiliary vertical building blocks complement the architecture, such as the logging service.

The results reported, in this deliverable represent the starting point for the refinement of modules of the RA and for the implementation phase. Indeed, the building blocks defined within the document will represent the infrastructure or software modules to be developed during the work to be done in work packages WP3, WP4 and WP5.

The Reference Architecture presented is not only the basis for future work, but it also represents the first milestone achieved in the FINSEC project as a collaborative effort to outline the physical and cyber solution for critical infrastructure in the financial sector. In this respect, the milestone can be considered successfully achieved.

The validity of the design of the RA will be monitored and if needed, will be reviewed during the next months, until the end of task T2.5 in M18, with the preparation and submission of deliverable D2.5 “FINSEC Reference Architecture II”.