D1.1

Initial Architecture Requirements Report

WP1 – Distributed System Architecture

SIFIS-Home

Secure Interoperable Full-Stack Internet of Things for Smart Home

Due date of deliverable: 31/03/2021
Actual submission date: 30/03/2021

Responsible partner: FSEC
Editor: Marko Komssi;
E-mail address: marko.komssi@f-secure.com

28/03/2021
Version 1.2

Project co-funded by the European Commission within the Horizon 2020 Framework Programme		
Dissemination Level		
PU	Public	**X**
PP	Restricted to other programme participants (including the Commission Services)	
RE	Restricted to a group specified by the consortium (including the Commission Services)	
CO	Confidential, only for members of the consortium (including the Commission Services)	

The SIFIS-Home Project is supported by funding under the Horizon 2020 Framework Program of the European Commission SU-ICT-02-2020 GA 952652

Authors: Riccardo Coppola (POL), Luca Ardito (POL), Andrea Saracino (CNR), Giacomo Giorgi (CNR), Domenico De Guglielmo (MIND), Marko Komssi (FSEC)

Approved by: Håkan Lundström (SEN), Max Dmitrichenko (INT)

Revision History

Version	Date	Name	Partner	Section Affected Comments
0.1	20/12/2020	Defined ToC	FSEC, CNR, POL	All
0.2	13/01/2021	Stakeholders and User definitions	FSEC, MIND, RIOTS, SEN	3.1
0.3	27/01/2021	Definition of context diagram	POL	3.4
0.4	27/01/2021	Added Background	POL	2
0.5	16/02/2021	Definition of User Stories	POL, CNR	5
0.6	22/02/2021	Use cases and first requirement version	POL	6,7
0.7	01/03/2021	Finalized requirement list	POL, CNR	5,6,7
1.0	07/03/2021	Ready for review	POL, FSEC, CNR	All
1.1	25/03/2021	Revised after review	POL, FSEC, CNR	All
1.2	27/03/2021	Ready for submission	CNR	All

Executive Summary

The main objective of the SIFIS-Home project is to provide a secure-by-design and consistent software framework for improving resilience of interconnected smart home systems at all stack levels. To address this goal, the software framework shall ensure correct functionality of the smart home system as well as to enforce security, privacy and safety of all SIFIS-Home users. This calls for eliciting both functional and non-functional requirements with a special focus on security and privacy aspects to ensure the functionality of the smart-home architecture.

This deliverable describes the initial architecture requirements for the SIFIS-Home system. The deliverable concentrates on three aspects. First, it presents a bottom-up research approach to address the requirements elicitation for the SIFIS-Home software framework to promote adaptability in various use cases. The approach takes a user-centric and holistic view to architecture requirements by embracing different contexts, use cases and stakeholders. Second, the deliverable presents the high-level preliminary conceptual architecture, SIFIS-Home context diagram and use cases. These three elements provide the overall foundation to the smart-home architecture requirements. Finally, the deliverable introduces the initial requirements for the SIFIS-Home system. The elicited requirements are divided into functional, non-functional and security requirements. While security requirements are typically seen as a part of non-functional requirements, they have a distinct and significant role in the SIFIS-Home architecture. Furthermore, the deliverable describes the priorities for the requirements and maps them to the SIFIS-Home use cases.

The proposed requirements have a significant role in the SIFIS-Home project. They will be utilised in WP3 and WP4 as well as validated and refined further within the activities of WP5 and WP6 at a later stage in the project. This deliverable also provides input to D3.1 and D4.1, to provide the finalized requirements list in D1.2.

Table of contents

1 Introduction

The SIFIS-Home project aims at providing a *software framework* that facilitates the management of security in smart home environments, ensuring certifiable levels of privacy and resilience in smart home applications and systems. As illustrated in Figure 1, this is achieved by leveraging two main components:

i. A software framework utilising secure IoT specific communication protocols that enables:
   a. securely managing and enforcing security functionalities,
   b. performing privacy-aware data handling, aggregation and analysis,
   c. ensuring secure communication in a resilient, easy and efficient way.

ii. A development toolkit that allows (third party) developers to provide applications that exploit the potential of the SIFIS-Home Security Architecture, to integrate security functionalities in their applications.

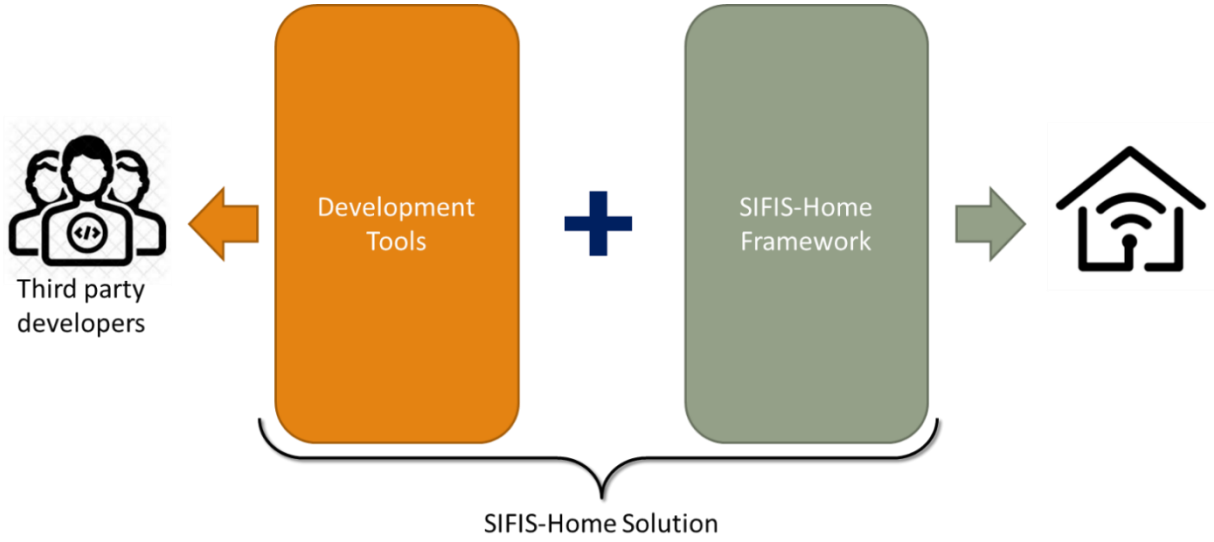


Figure 1: SIFIS-Home main building blocks

The SIFIS-Home software framework is devoted to ensure correct functionality of the smart home system. The framework must enforce the security of connected devices as well as the privacy and safety of home tenants. To this end the framework must be designed considering a great range of functional, technical and security requirements, reflecting the needs for a flexible, performing, resilient, configurable and privacy aware smart home framework.

This deliverable reports the process of requirements elicitation for the SIFIS-Home software framework. This process aims at identifying the basic functionalities for the software framework and the consequent functional and non-functional requirements, with a specific attention to security requirements. The methodology used to gather requirements has been based on a bottom-up procedure, to define architectural requirements which can adapt to different implementations. The proposed requirements will be validated and refined further within the activities of WP5 and WP6 at a later stage in the project.

This deliverable provides input to D3.1 and D4.1, to provide the finalized requirements list in D1.2.

2 Background

In this section we present and describe the typical components of a requirements elicitation document. The Requirements document is the result of the requirements engineering steps described in the following sections and formalizes all the requirements elicited [Sommerville, 1997].

2.1 *Requirement Gathering Approaches*

Requirements engineering is the procedure of finding out, analysing, and documenting all information about what the system should do and its purposes [Sommerville, 2011]. The requirements engineering process covers both *user requirements* (i.e., what the system is expected to provide to its final users) and *system requirements* (i.e., documentation of the exact implementation details of the system).

Software system requirements are typically classified in two sets: functional requirements and non-functional requirements:

- **Functional requirements (FR)** define the system behaviour: *what* the system should do or must not do. They can be thought of in terms of how the system responds to its external inputs.

- **Non-functional requirements (NFR)** specify *how* the system should perform its behaviour. They are not concerning the basic functionalities of the system, but instead pose constraints on the services or functions offered by the system (e.g., timing, performance, security, usability constraints). Requirements about software products quality are defined and categorized in the ISO/IEC 25010 standard. All non-functional requirements must be measurable, in order to be testable.

A taxonomy of *non-functional requirements*, proposed by Sommerville, is reported in Figure 2, and includes the following high-level categories:

- *Product requirements*: they specify or constrain the behaviour of the system (e.g., performance, reliability and memory requirements)
- *Organisational requirements*: they are derived from policies and procedures in the customer's and developer's organisation (e.g., operational, development process and standardisation requirements)
- *External requirements*: they are derived from external factors (e.g., regulatory, domain, ethical, legislative requirements).

The definition of requirements can follow one of two possible strategies: the *top-down* approach, or *bottom-up* approach.

- The **top-down approach** (also called *stepwise refinement*) starts from the analysis of the description of the whole business process that will be served by the system. This approach results in a progressive decomposition of the system's requirements into more fine-grained requirements that are then divided between the different components of the whole system picture. The most important benefit of a top-down design approach is the possibility of identifying from the beginning a set of reusable functionalities and patterns that shall not be repeated between the different components of the system. The downside of a top-down approach is that the analysis and design does not consider at all the possible existence of systems that will have to support the functionalities of the system under design.

- The **bottom-up approach** is based on the combination of the analysis of multiple existing systems together, to give rise to a more complex unitary system. The bottom-up approach starts with the specification of the individual sub-systems involved and their capabilities, then

proceeds through the abstraction of group-level behaviours [Crespi, 2008].

The top-down approach takes advantage from being specific for a set of pre-defined use cases and should thus be preferred when use cases or pilots are already defined. On the other hand, the bottom-up approach tends to be more general, abstracting from the actual final implementation and should be preferred when designing an architecture which is easily adaptable to any use case matching the architecture specification.

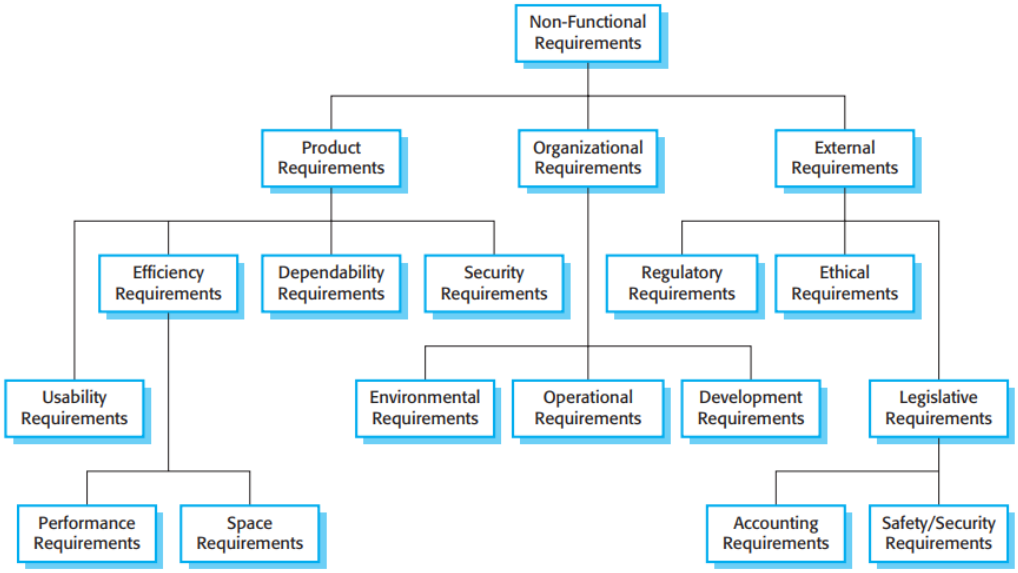


Figure 2: Non-Functional Requirements Decomposition [Sommerville, 1997]

2.2 *Context Diagrams*

The context modelling of the system is an essential part for the definition of the architectural requirements of the system, its environment, and its dependency and interfaces towards other components. The context model is typically defined at the earliest stage of the system specification, and clearly identifies the boundary between *the system* and *the external world*.

The most important elements addressed by the definition of the system's Context Diagram are:

- *Identity and responsibilities of external entities*: the set of external entities (actors) that interact with the system, their responsibilities and the way the actual interaction is carried out. The actors can either be final (human) users of the systems, or other external hardware systems that will interact with the system under design.
- *Interfaces with the external entities*: in the context modelling phase, it is crucial to identify how the external entities will interact with the system under design. This task is carried out by defining interfaces between the system and the external entities. For each external entity, a physical interface (i.e., physical means of connection and/or interaction with the system) and a logical interface (i.e., the protocols used for the interaction) are specified.
- *The system and the internal entities*: the set of components that belong to the system, and for which it is not needed an external communication interface.

While the UML standard does not provide specific tools for context modelling, a widespread means of visualizing the context view is through informal Context Diagrams (e.g., the "box and arrow" diagram

reported in Figure 3).

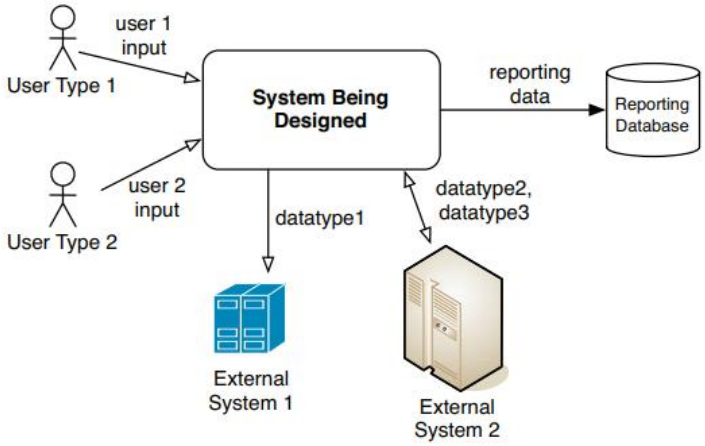


Figure 3: Sample Informal Context Diagram

The most important benefits of the usage of such type of diagrams are: (i) the clear definition of the scope and boundaries of the system at a glance, including other systems that will be interfaced with it; (ii) the readability of the diagram without the need of technical knowledge (henceforth readable by all stakeholders involved in the requirements elicitation and validation process).

2.3 *Use Cases and Use Case Diagrams*

Use cases are a common requirement discovery and elicitation technique, introduced by Jacobson et al. [Jacobson, 2004] and are a standard of the Unified Modelling Language (UML). Each use case identifies the actors that are involved in an interaction with a system and gives a name for the type of interaction. The specification of the use case provides information about the individual steps and tasks that take place in an interaction of a given type between an actor and the system. It describes the system's behaviour under various conditions, as it responds to a request from a *primary actor*. Each use case can also involve a *secondary actor*, meaning that another actor of the system is part of the interaction initiated by the primary one.

The set of use cases can be visually documented by a Use Case Diagram (UCD). It represents the complete set of all the possible interactions with the system, that will be formalized later by functional requirements of the system in the requirements document. Figure 4 reports a sample Use Case Diagram defined with the UML language.

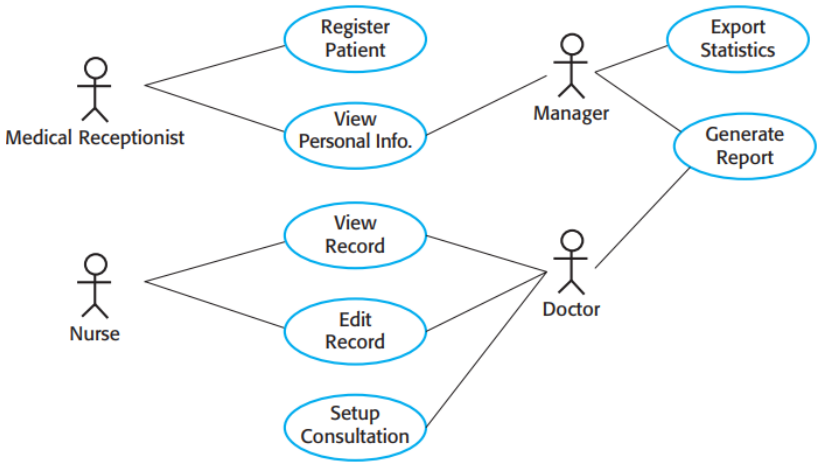


Figure 4: Sample UML Use Case Diagram

Figure 5 reports a basic use case template [Cockburn, 1998], containing the main information about each use case of a system. The template contains the following information:

- *Goal in context*: the functional goal that the actor achieves using the system, i.e. the reason for which the system is used. The goal can be formulated using the following template: as a <actor>, I want to <do something> so that <some outcome is reached>
- *Scope*: if a layered decomposition has been defined for the architecture of the system, the Scope defines the level inside such decomposition (e.g., summary use cases at the highest level, primary task, or subfunction);
- *Preconditions*: what can be assumed about the current state of the system and environment;
- *Success end condition*: The state of the system and environment if the goal of the primary actor succeeds;
- *Failed end condition* (or "minimum guarantees"): how the interests of the stakeholders are protected in all circumstances, even in failures;
- *Primary, secondary (or "support") actors*: the primary actor of the use case is the type of actor that is triggering the use case and that has the intention of interacting with the system. The secondary actors are optional supporting other actors that perform some steps of the use case narrative;
- *Trigger*: if the use case is not started by a voluntary operation performed by the primary actor, the trigger defines which event has started the use case;
- *Main success scenario*: a numbered sequence of steps that describes the scenario from trigger to goal delivery. The form used in the template is <step_number>.<action_description>
- *Extensions and sub-variations*: show conditions, branches and alternatives, in the following format: <step_altered> <condition> ":" <action_description> or <sub_use_case>

USE CASE #	< the name is the goal as a short active verb phrase>	
Goal in Context	<a longer statement of the goal in context if needed>	
Scope & Level	<what system is being considered black box under design> <one of : Summary, Primary Task, Subfunction>	
Preconditions	<what we expect is already the state of the world>	
Success End Condition	<the state of the world upon successful completion>	
Failed End Condition	<the state of the world if goal abandoned>	
Primary, Secondary Actors	<a role name or description for the primary actor>. <other systems relied upon to accomplish use case>	
Trigger	<the action upon the system that starts the use case>	
DESCRIPTION	**Step**	**Action**
	1	<put here the steps of the scenario from trigger to goal delivery,and any cleanup afte>
	2	<...>
	3	
EXTENSIONS	**Step**	**Branching Action**
	1a	<condition causing branching> : <action or name of sub.use case>
SUB-VARIATIONS		**Branching Action**
	1	<list of variation s>

Figure 5: Sample reporting scheme for a Use Case [Cockburn, 1998]

2.4 *Requirement Evaluation and Prioritisation*

After the requirements are defined, they must be evaluated in terms of their priority. Requirements prioritisation is considered one of the most important processes in the definition and construction of software processes [Ruhe, 2005].

Prioritisation helps identifying the most important requirements from the full set of requirements and can be performed taking many different aspects into account. Common aspects are importance, number of stakeholders with an interest in the requirement, cost and time for the development of the features satisfying the requirement, and penalty in case of failure in satisfying that requirement.

In the context of this requirements elicitation document, we opt for adopting the most common prioritisation technique, suggested by RFC 2119 and IEEE std. 830-1990, namely the *Numerical Assignment Prioritisation*. The practice consists in dividing the requirements in three different groups representing *critical*, *standard* and *optional* priority for the requirements [Berander, 2005].

In the context of the SIFIS-Home architecture, a prioritisation scheme can consider as *critical* all functional requirements related to the infrastructure management, and all non-functional requirements related to security, safety and access to the SIFIS-Home network. Standard requirements are related to individual features offered to the individual systems of the users. Optional requirements can be linked to the quality of the provided services.

3 Used Requirement Collection Methodology

Our Requirements elicitation procedure is summarized in the flow-chart in Figure 6.

The first step of the Requirements Elicitation Process has been an analysis of the stakeholders involved with the conceptual preliminary architecture of SIFIS-Home. We report a brief description for each stakeholder in Section 4. In Section 5, we report the informal preliminary architecture that can be derived from the analysis of the pilot systems. Based on the study of these systems, we derive a high-level definition of the main components of the architecture. In the same phase, we define a high-level conceptual security architecture necessary for the SIFIS-Home system.

Building on the stakeholders identified for the SIFIS-Home system, and on the preliminary architecture, we define the high-level context diagram in Section 6. The high-level context diagram identifies three different main typologies of actors for the SIFIS-Home system*: internal devices, external devices, and human users*. For each category of actor we define: (i) the interfaces with which it can interact with the system, both physically (e.g., network connections) and logical (e.g., APIs and user interfaces), and (ii) the typologies of data streams that it will exchange with the system.

In Section 7, we define the high-level Use Case Diagrams in the system. To build the Use Case diagram, we identify, for each of the users defined in the context diagram, the principal ways of interaction with the system, obtaining a set of User Stories (section 7.1). On top of each user story, we define a set of use cases for the specific actor (section 7.2) and we detail each of them by using the Narrative template described in the background section.

Both the use case and the context diagram provide information for the definition of the functional requirements of the system. The full requirements table, along with an informal clarification of each requirement, is reported in subsection 8.1. Non-functional requirements of the system reported in subsection 8.2. along with their categorisation according to the ISO/IEC 25010 standard, are elicited based on corresponding functional requirements.

Finally, in subsection 8.3, we report the security requirements, which encompass both functional and non-functional characteristics of the SIFIS-Home system. For all typologies of requirements, we report a prioritisation in three different priority levels: critical, standard and optional.

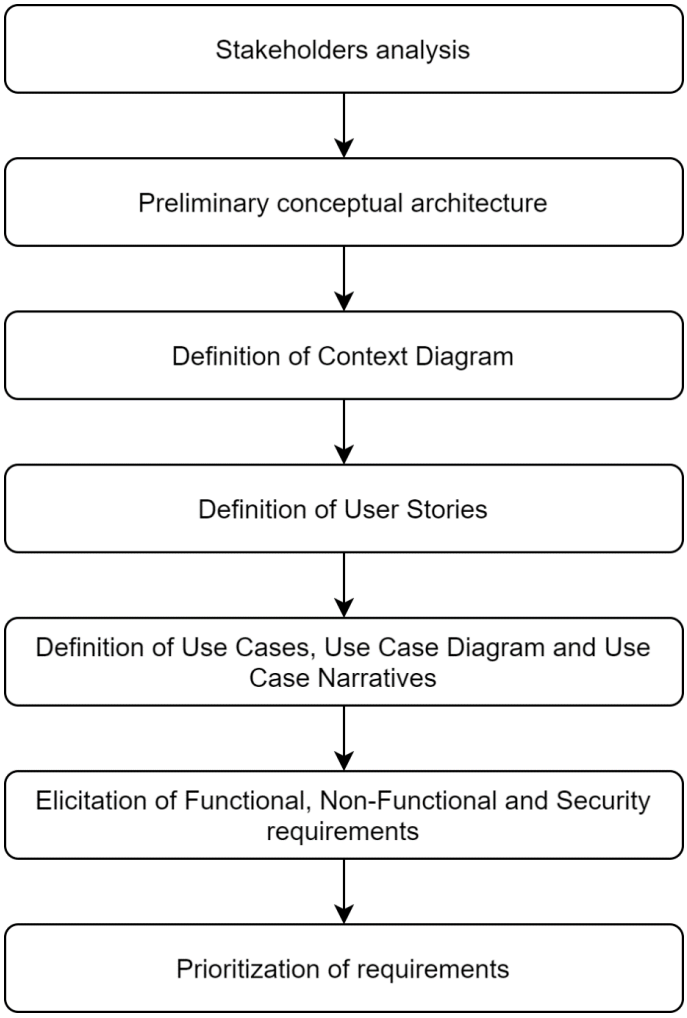


Figure 6: Workflow for the requirement gathering process

3.1 *SIFIS-Home Stakeholders*

In Table 1, we report the stakeholders of the SIFIS-Home system. By definition, a stakeholder is any group or individual who can affect or is affected by the achievement of the application, and whose actions can influence or be influenced by the development and use of the system whether directly or indirectly [Sharp, 1999].

Stakeholder	**Description**
Administrator (Admin)	He is the owner of the smart home system, responsible for managing the household's information security, setting up configurations and usage profiles.
Resident	A resident is the standard user of the smart home. He lives in the apartment, has a usage profile and uses the services provided by the smart home.
Restricted Users	Residential user with limited features (e.g. children).
Guest	Guest is temporarily visiting in the home and can use a subset of the services of the smart home.
Unknown User	Unknown User is an unhostile and unknown person or group associated with the apartment (e.g., a courier, dealer, plumber, postman …)
Housing Management Service Provider	Housing Management Service Provider provides management services for the (apartment) building. They can enhance and streamline their services to the property owners/residents with the data collected by the smart home system.
Property Maintenance Service Provider	Property Maintenance Service Provider provides maintenance services in the building. They can streamline and optimize their maintenance services towards the property owners/residents.
Internet Service Provider	Internet Service Provider (or telecom operator) provides services for the building and/or household for accessing and using the Internet.

Router Manufacturer	Router Manufacturer provides a router (networking device) for the household enabling to perform the traffic directing functions on the Internet
Building Constructor	Building Constructor can monitor the building functionalities connected to the system already during the construction stage and continue receiving data regarding the building functionalities throughout the constructor's liability period for streamlined corrective actions when and where needed.
Third-party Operator as Service Provider	Third-party Operator as Service Provider is a hotel/Airbnb owner/host, security service providers, or catering services that operates or conduct business on the apartment's premises.
Installer	Employees of the smart home solution company that take care to configure the house: they create the house structure (floors, rooms, objects) and can change the house settings (e.g. temperature control).
Developer	Developers of the SIFIS-Home system.
Third party developer	Developers of external applications that interact/communicate with a SIFIS-Home gateway.

Table 1. The stakeholders of SIFIS-Home system.

3.2 *High-Level Preliminary Conceptual Architecture*

Figure 7 illustrates an initial architecture of the SIFIS-Home system that is used to represent the base functionalities of the system that will be described by the general use cases. With the aim of defining the context diagram for the SIFIS-Home architecture we report in the following the logical architecture of a smart home which is SIFIS-Home capable.

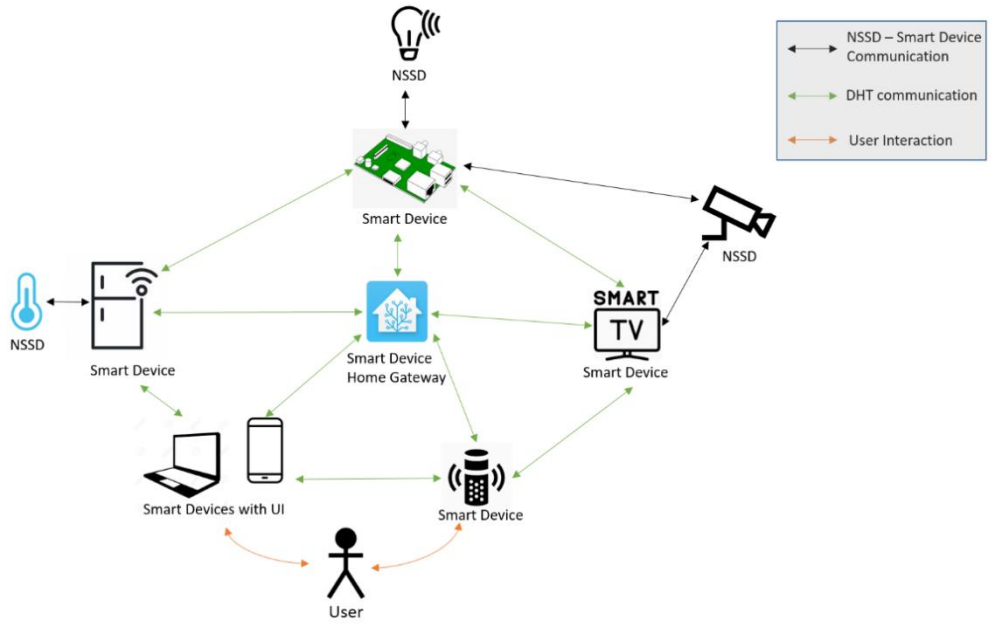


Figure 7: The preliminary architecture of the SIFIS-Home system

The architecture includes the following set of devices:

- *Smart Devices:* These devices are characterized by a relatively good computational capability; they are based on general purpose computational hardware and their functionalities are managed through an Operative System (OS). Smart devices can be customized by installing third party software and have the capability of directly communicating among them, autonomously exchanging information. This intercommunication enables a Peer-to-Peer (P2P) logical model, which is easily represented by means of a distributed hash table (DHT).
- *Not So Smart Devices (NSSD):* This set is made by those devices which present smart functionalities and present a network interface, yet they still have very limited computational power, and only present a firmware instead of a fully-fledged OS. For this reason, the NSSDs cannot be customized by installing third party software or applications.

Both smart devices and NSSD might or might not have interfaces, such as sensors, screens, touchscreens or other input peripherals to receive commands from the user. Smart devices might have network interfaces to connect to the Internet, being thus able to send and receive information and commands from outside the house logical perimeter.

This architecture description is only preliminary and acts as baseline for identifying the basic user interactions and the consequent functionalities on which we base the requirement gathering process.

4 SIFIS-Home Context Diagram

Figure 8 reports the Context Diagram of the SIFIS-Home system. In the Context Diagram we have divided the actors interacting with the system in three categories:

- **Internal entities:** Identifies a category of internal devices (i.e., located inside the home network) connected to the SIFIS-Home system.
- **External entities:** Identifies a category of external entities (e.g., devices or services) located outside the home network.
- **Users:** identifies a category of human users of the SIFIS-Home system, e.g. an utiliser of the smart devices in the SIFIS-Home network.

Detailed definitions of each category of actors are reported in the following subsections.

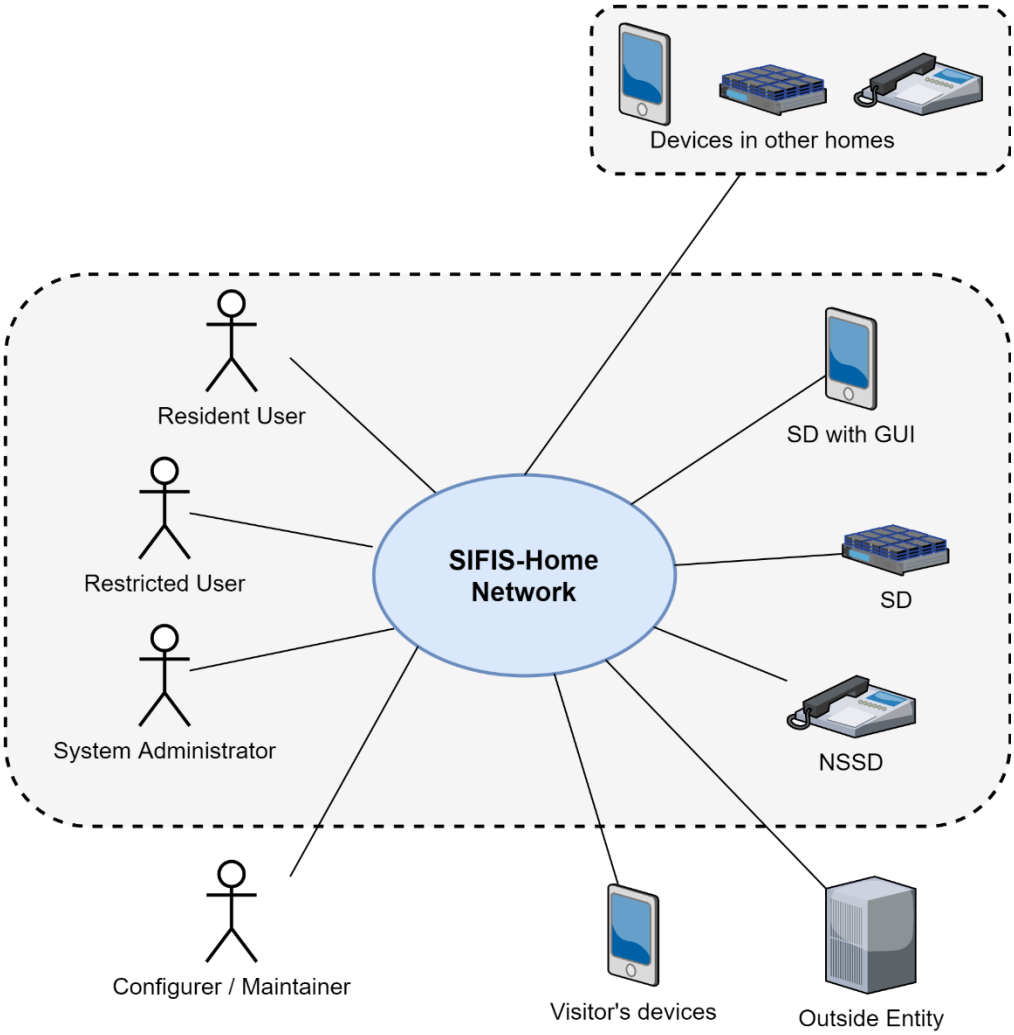


Figure 8. SIFIS-Home Context Diagram

4.1 *Internal Entities*

Table 2 reports the categories of internal entities defined for the interaction with the SIFIS-Home system. For each entity we report the physical and logical interfaces through which the entity will communicate with the system. Definitions, examples and possible variations inside a specific category

of entity are reported in the following:

- **Smart Device without local GUI (Graphical User Interface):** device with good computation capabilities, with a general purpose OS, where it is possible to install applications, and that should provide communication to the outside through an internet connection. Examples of devices in this category include Raspberry Pie devices, smart fridges, parked cars that can be controlled.
- **Smart Device with local GUI:** smart device with local GUIs exposed to the user. Examples of devices in this category include smartphones and smart TVs.
- **Not So Smart Device (NSSD):** device with connection capabilities but with low computational power and without the possibility of installing a general purpose OS. Examples of devices in this category include smart light bulbs, thermovalves, smart cameras, ESP 32-based devices, sensors, actuators.

Entity Type	Physical Interfaces	Logical interface
Smart Device without local GUI	WiFi connection, Ethernet, Bluetooth connection, IEEE 802.15.4, WiFi ad-hoc, IoT protocols, 5G (lowpan)	Software APIs
Smart Device with local GUI	WiFi connection, Ethernet, Bluetooth connection, IEEE 802.15.4, WiFi ad-hoc, IoT protocols, 5G (lowpan)	Software APIs
Not So Smart Device (NSSD)	WiFi connection, Bluetooth connection, IEEE 802.15.4, WiFi ad-hoc, IoT protocols, 5G lowpan	Software APIs

Table 2. Internal entities in the SIFIS-Home Context Diagram

4.2 *External Entities*

Table 3 reports the categories of external entities that we have defined for the interaction with the SIFIS-Home system. For each entity we report the physical and logical interface through which the entity will communicate with the system. Definitions, examples and possible variations inside a specific category of entity are reported in the following:

- **Visitor's device:** a smart device of a human user who is visiting a SIFIS-Home-served smart home. Its characteristics are equivalent to the smart device with local GUI category of internal devices.
- **Outside entity:** an external entity that can interact with the SIFIS-Home system. Examples of devices in this category include cloud applications, external storages or services, services that need a connection to the SIFIS-Home System (e.g., cloud based smart speaker).
- **Devices in other homes served by SIFIS-Home:** a device located in another smart home where another instance of the SIFIS-Home system is running. It can belong to any category of internal devices.

Entity Type	Physical Interface	Logical interface
Visitor's device	WiFi connection, Ethernet, Bluetooth connection, IEEE 802.15.4, WiFi ad-hoc, IoT protocols, 5G (lowpan)	GUI guest interface to the SIFIS-Home system
Outside entity	Internet connection	APIs with privacy policies
Devices in other homes served by SIFIS	Internet connection	APIs with privacy policies

Table 3. External entities in the SIFIS-Home context diagram

4.3 *Users*

Table 4 reports the categories of users that we have defined for the SIFIS-Home system. For each user we report the physical and logical interface through which the entity will communicate with the system. Definitions, examples and possible variations are reported in the following:

- **Resident user / Smart Home Resident:** A resident living in the smart home capable of using all the functionalities of the smart home system. Resident Users can configure their own profiles and install dedicated applications, but they do not have controls on the security policies in the house.
- **Restricted user:** User with restricted features according to the profiles defined in the current SIFIS-Home configuration. Examples of restricted users are visitor users, who are temporarily visiting in the apartment and can use the digital aspects of their home (apartment) connected to the network; children and elderly people, who can have a restriction on the features that they can access for security and safety reasons.
- **System administrator:** Admin is the owner of the smart home and is responsible for managing the household's information security. They configure security policies and have the rights to register and unregister devices in the smart home.
- **Configurer / Maintainer:** A service provider who offers maintenance services for the building itself or for a certain part of the system (e.g. the heating system, the ventilation system, access control & security services on the premises, etc.). Their responsibilities may include ensuring specific device or functionality operability; they may not need access all the user's personal information stored in the system, but need access to the system for example to perform specific device maintenance.

User	Physical Interface	Logical Interface
Resident User	Smart Device with GUI SIFIS-Home applications that connect to the FIWARE NSGI open API on the SIFIS-Home compliant middleware	Compliant and non-compliant apps + haptic, voice commands
Restricted user	NSSD, buttons, + physical interfaces in the SIFIS-compliant smart home	haptic, voice commands, etc.
System Administrator	Smart Device with GUI	Configuration panel of the GUI
Configurer / Maintainer	Smart Device with GUI	Configuration panel of the GUI

Table 4. Categorized list of users for the SIFIS-Home system

5 SIFIS-Home Use Cases

In this section we report the process of use case elicitation for the considered system. In subsection 5.1 we report the User Stories for the human users defined in the Context Diagram. In subsection 5.2 we report the Use Cases derived from the Stories, each detailed with a Use case Narrative.

5.1 *User Stories*

Seven main user stories have been defined for the human users of the SIFIS-Home system and are detailed in the following subsections. For each user story, we use Leffingwell's template [Leffingwell, 11], in the form *As a <actor>, I want <action>, so that <business value>*. We also report additional details about the actions in the user story, the main actors involved, and the acceptance test that should be applied to verify and validate the story.

5.1.1 SIFIS-US-01: Smart home handling through voice command

As a
Smart home resident
I want to
use voice commands
So that
the system recognises me, provides me with services and does not share private information about my requests with external entities.

Discussion
The smart home residents can exploit the speech recognition system offered by the SIFIS-Home framework to authenticate to the smart-home. After the authentication, the resident can access every home device with the modalities described in the access and usage policies and manage the smart devices through vocal commands expressed in natural languages (e.g., turn on the light, turn down the volume). Every vocal command registered by the system will be anonymized and maintained private to preserve the residents' privacy.

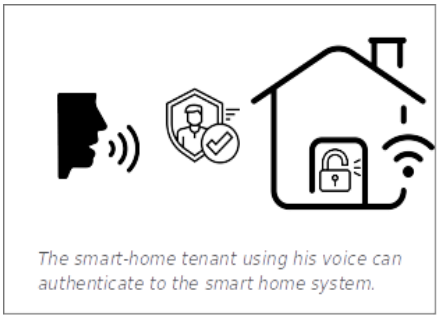
Main actors:

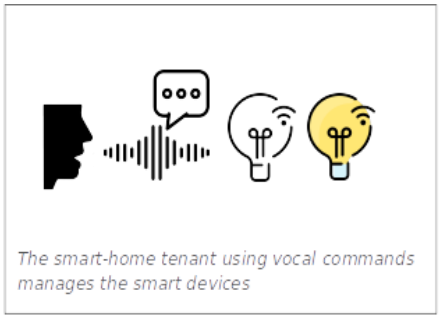
- Smart home resident, restricted user, smart home administrator

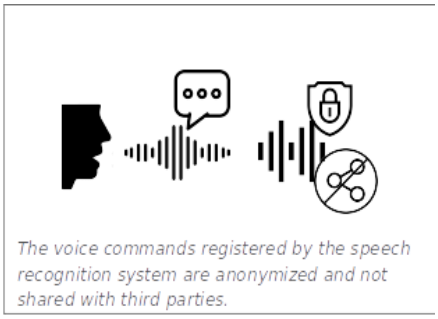
Acceptance test

- The actor can authenticate to the SIFIS-Home framework using their voice.
- The actor can manage the smart home components through voice commands.
- The speech recognition engine can maintain the privacy of any sensitive information.

Storyboard







5.1.2 SIFIS-US-02: Smart home configuration panel

As a

Smart home administrator

I want to

have a configuration panel

So that

I can set house usage, security and privacy policies to protect me, the residents and guests and visualize the smart home statistics.

Discussion

The smart home administrator can access to a smart-home configuration panel through which it is possible to define:

- High-level policies expressible in natural language (e.g., “Do not record sound in the living room tonight”). Those policies will trigger a reconfiguration of all the IoT devices to comply with this rule. They will be translated in a device configuration, which can limit the features of an IoT device, or inhibit the operation of a non-reconfigurable device.
- Device configurations which express how they will work in the smart environment.
- Visualize the statistics and the analytic results related to the smart home operation (e.g., energy consumptions, devices status).

The administrator will be able to define policy and device configuration in a remote way through the smart-home configuration panel offered by the SIFIS-Home framework.

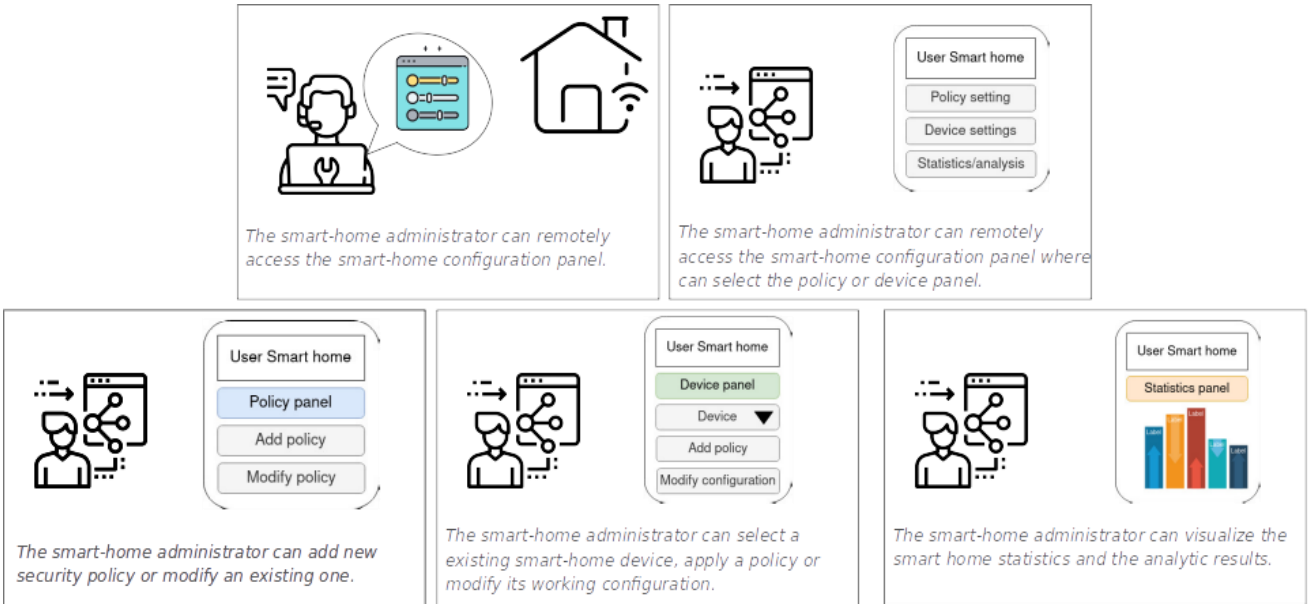
Main actors:

Smart home administrator

Acceptance test

- The smart-home administrator can remotely set security and privacy policies defining how each smart-home user can access and use the home devices.
- The smart-home administrator can remotely set the configuration of each home device.
- The smart-home administrator can visualize the smart-home statistics.

Storyboard



5.1.3 SIFIS-US-03: Physical anomaly detection in smart-home

As a

Smart home resident/administrator

I want to

be notified if someone enters my house without permission or if an anomalous action occurs

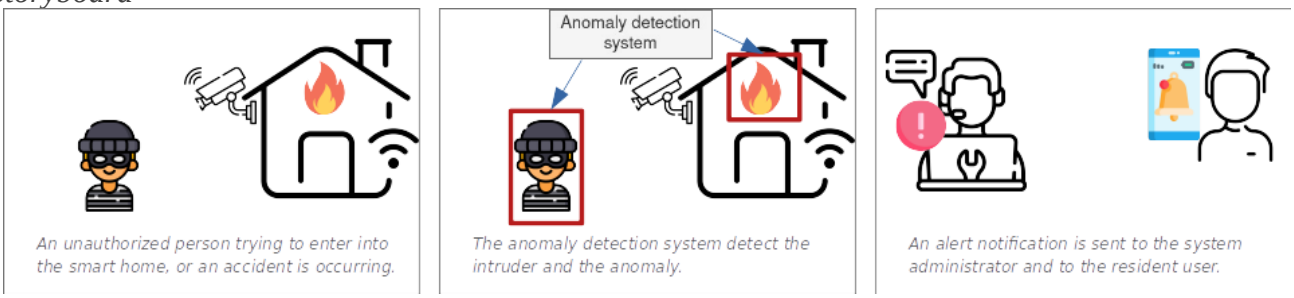
So that

I can protect myself and ask for help from the authorities.

Discussion

The smart-home resident and administrator can exploit the notification provided by the SIFIS-Home framework related to misbehaviours that occurred in the house. The SIFIS-Home framework will be able to detect physical anomalies by analysing video or audio streaming. The notification will be related to unauthorized persons trying to access the home, forbidden action or an accident that is occurring, or an unusual or forbidden object introduced into the home. The notification will arrive at the smart resident and administrator through a configurable communication channel (email, SMS, ...) to promptly react to the detected anomaly.

Storyboard



Main actors:

- Smart home resident
- Smart home administrator

Acceptance test

The smart home resident and the administrator must be notified as soon as an unauthorized person is being introduced in the smart home, or a dangerous situation is observed.

5.1.4 SIFIS-US-04: Software anomaly detection in smart-home

As a

Smart home administrator

I want to

be notified if some software intrusion is currently taking place

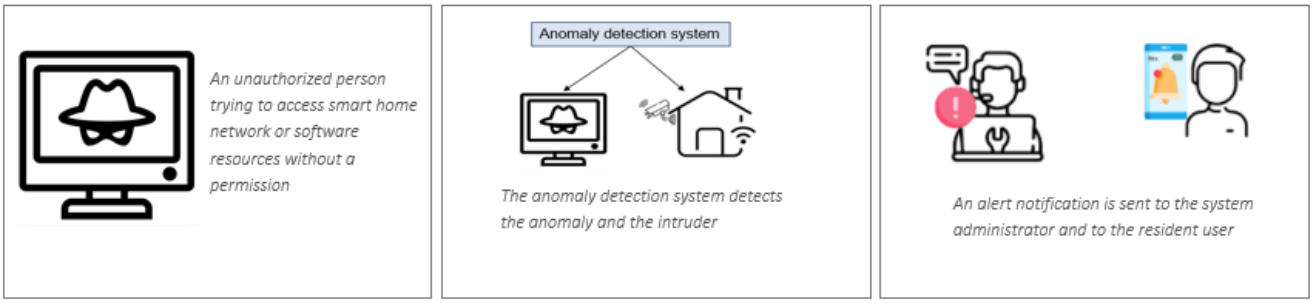
So that

I can take countermeasures to react to the attack.

Discussion

The smart-home administrator can exploit the SIFIS-Home framework's notification related to the software anomalies detected in the home devices or smart-home network. The software anomalies will be related to the unauthorized access to a smart device, malware or network attack to compromise nodes or steal sensitive information. The SIFIS-Home framework exploiting a software and network intrusion detection system will notify the smart-home administrator through a configurable communication channel (email, SMS) to make it possible a promptly react to the attack.

Storyboard



Main actors:

- Smart home administrator

Acceptance test:

- The smart-home administrator must be notified as soon as a software intrusion is detected in the smart-home system.

5.1.5 SIFIS-US-05: Register/Unregister device in the smart home

As a

Smart home resident/administrator

I want to

register and unregister new devices in the smart home

So that

I can use them in the smart home through the supervision of the SIFIS-Home framework.

Discussion

The smart-home resident or administrator user must have the possibility to register and unregister a new device (smart or not so smart) to the smart-home system managed by the SIFIS-Home framework. The system will allow the user to register a new device inserting the configuration settings related to how it must work. The device will be added to the user profile of the person who registered it to enable them to unregister the device and modify its settings. After the registration, the device will

be available to work with the SIFIS-Home framework and could be managed by the smart-home administrator or the resident user. The system will allow having limited registration, e.g. on daily or weekly basis, for devices that will have to be used only for a limited amount of time.

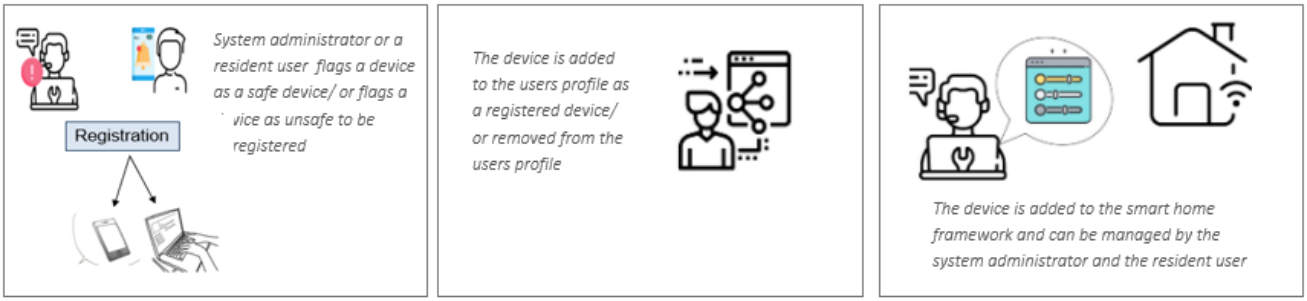
Main actors:

- Smart home resident
- Smart home administrator

Acceptance test

- The actor must have the possibility to register/unregister a new device in the smart home.
- The device must be added to the user profile of the person who registered it.
- The device must be available to be managed by the resident user and system administrator.

Storyboard



5.1.6 SIFIS-US-06: Installing third party application

As a

Smart home administrator/resident

I want to

Install new third party application in specific devices

So that

I can use it in the SIFIS-Home ecosystem

Discussion

The smart home administrator or resident user must have the possibility to install new functionalities in a specific device by installing a third-party application. The SIFIS-Home framework will check the smart home policies with the security and safety aspect of the application and its functionalities, then if the check was successful the SIFIS-Home framework will allow for the application to be installed. Otherwise, if some policies are violated the installation procedure will not start.

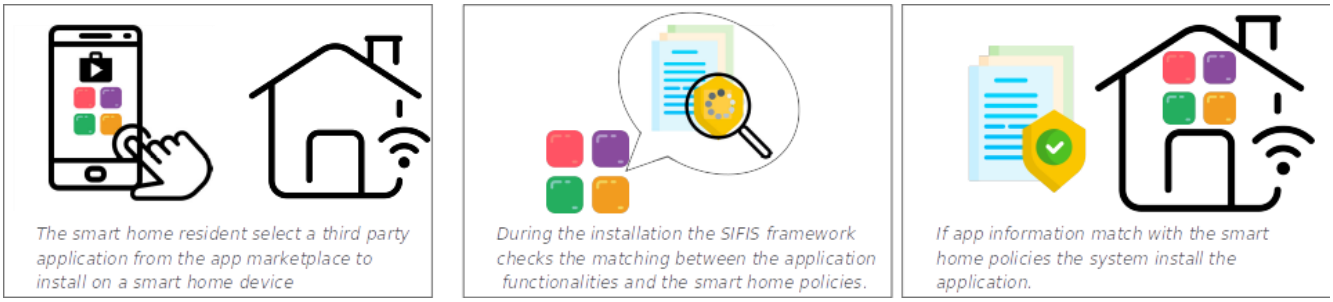
Main actors:

- Smart home resident.
- Smart home administrator.

Acceptance test

- The SIFIS-Home framework must allow the installation of a third-party application that matches the smart home policies.
- The SIFIS-Home framework must block the installation of a third-party application that does not match the smart home policies.

Storyboard



5.1.7 SIFIS-US-07: Creation and management of user profiles

As a

Smart home administrator/resident,

I want to

be able to create user profiles of different types and privileges and assign devices and applications to those profiles.

So that

profiles can be managed, and network can be monitored easily

Discussion

The system administrator or a resident user must have the possibility to create a new user profile that can operate in the smart home. It is possible to provide privileges and assign devices and applications to the new user profile. In addition, the actor can specify their preferences and enforcement conditions.

Main actors:

- Smart home resident
- Smart home administrator.

Acceptance test

- The actor must have the possibility to create a new user profile assigning them custom privileges and the device and application allowed.

Storyboard

5.2 *Use Case Diagram and Narratives*

Figure 9 reports the full Use Case Diagram of the SIFIS-Home system. The human actors interacting with the system reflect the human actors identified in the Context Diagram (described in previous section) and in the User Stories from which the Use Cases are elicited. The Use Case Diagram highlights the hierarchy of dependencies between different the different users. Users that are at lower levels in the hierarchy inherit all the use cases of the users higher in the hierarchy. Detailed description of the use cases, in the form of Use Case Narratives, are reported in Section 5.2.1 to Section 5.1.14. In Table 5, we map the use cases with the stories from which they are generated.

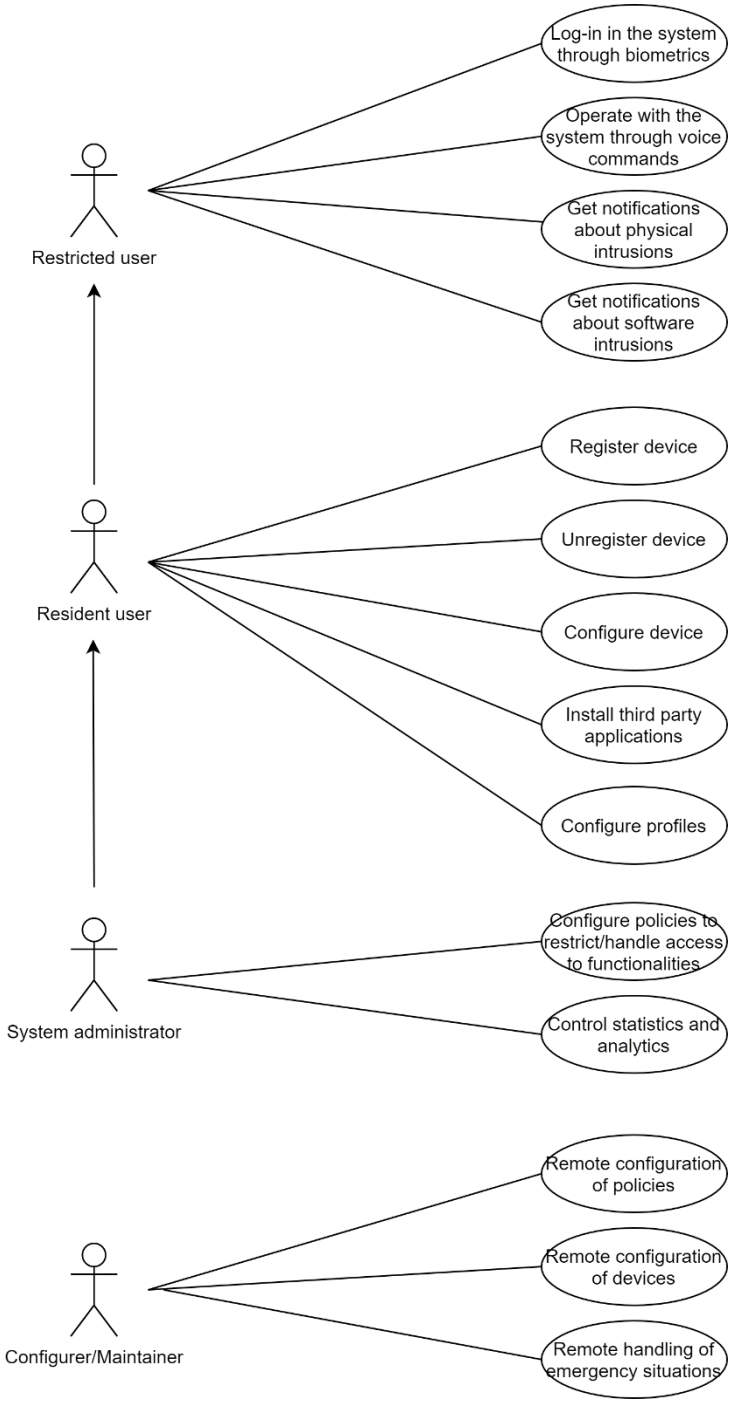


Figure 9. Use Case Diagram

In the following we provide the narratives for each use case, specifying goals, triggers, users and steps of each of the 14 use cases shown in the use case diagram.

5.2.1 SIFIS-UC-01: Log-in in the system through biometrics

Use Case #	SIFIS-UC-01
Goal in Context	The users want to be identified with biometrics for authentication or to receive dedicated services.
Scope & Level	User Goal
Preconditions	User is registered in the system.
Success End Conditions	The user is recognised and authorized to do specific actions according to the identity.
Failed End Condition	Voice commands are not recognised, and only low privileges operations are allowed.
Primary, Secondary Actors	Restricted User, Administrator
Trigger	-
Description	1. The user interacts (directly or indirectly) with a device reading biometrics (e.g. camera, smart speaker).
	2. The biometrics data are read.
	3. The identity is matched with the known ones.
	4. Specific actions are allowed or functionalities are activated according to policies and profiles.
Extensions	
	4a. The identity is not recognised a guest profile is applied.
	4b. The identity is not recognised and no residents are in the house. The person is considered as an intruder.

5.2.2 SIFIS-UC-02: Operate with the system through voice commands

Use Case #	SIFIS-UC-02
Goal in Context	Give voice commands to devices registered in the system.
Scope & Level	User Goal
Preconditions	User is registered in the system.
Success End Conditions	Voice commands are correctly handed by devices of the SIFIS-Home system.
Failed End Condition	Voice commands are not recognised by devices of the SIFIS-Home system and a new input is requested.
Primary, Secondary Actors	Restricted User, Administrator
Trigger	-
Description	1. The user gives a voice command to start the interaction with the system.
	2. The system asks which type of command the users wants to execute.
	3. The user gives the specific voice command.
	4. The system executes the command.
Extensions	1a. The command can also be issued via other input peripherals.
	3a. The user does not provide any command. The use case ends with failure.
	4a. The system does not recognise the command. The system asks the user to repeat the command. The use case goes to step 3.

	4b. The command cannot be executed from the device that captured it. It is forwarded to the device able to execute it.

5.2.3 SIFIS-UC-03: Get notifications about physical intrusions

Use Case #	SIFIS-UC-03
Goal in Context	Notify the user of an ongoing physical intrusion.
Scope & Level	Emergency management
Preconditions	Motion sensors and/or cameras are installed in the smart home.
Success End Conditions	The intrusion is detected and the users are notified.
Failed End Condition	-
Primary, Secondary Actors	Restricted User, Maintainer
Trigger	An intruder enters the smart home without authorisation; the intrusion is noticed by motion sensors and cameras and the alarm system is activated.
Description	1. The system shows a notification to the user about the intrusion.
Extensions	1a. Cameras start recording the intruder actions and store the identity if face is not covered. The recording can be analysed by the maintainer.
	1b. Authorities are contacted to give assistance.

5.2.4 SIFIS-UC-04: Get notifications about software intrusions

Use Case #	SIFIS-UC-04
Goal in Context	Notify the user of an ongoing software intrusion (malware).
Scope & Level	Emergency management
Preconditions	An anomalous behaviour can be identified.
Success End Conditions	The intrusion is detected, and the users are notified.
Failed End Condition	-
Primary, Secondary Actors	Restricted User, Maintainer
Trigger	Malware is installed on a smart device and the malicious code or behaviour is identified by the system.
Description	1. The system notifies the user about the installation malicious code
Extensions	1a. Self-healing algorithm is used to transfer functionalities of isolated device to others (if possible).
	1b. Maintainer or administrator verify that the malware has not spread to other devices..

5.2.5 SIFIS-UC-05: Register device

Use Case #	SIFIS-UC-05
Goal in Context	Registering a new device in the SIFIS-Home network.
Scope & Level	User Goal
Preconditions	Resident user is logged in the system.
Success End Conditions	A new device is registered in the SIFIS-Home network.
Failed End Condition	No device is registered in the SIFIS-Home network.
Primary, Secondary Actors	Resident User
Trigger	-

Description	1. The user opens the feature “Register new device”
	2. The system prompts the user version of the page with device characteristics to be inserted.
	3. The user inputs the characteristics and the name of the device.
	4. The system shows a recap of the information and asks the user for confirmation.
	5. The user confirms the registration.
	6. The system confirms that the registration is successful.
Extensions	3a. The user quits. The use case ends with failure.
	3b. The user selects a temporary registration for the device.
	5a. The user does not confirm the registration. The use case ends with failure.
	6a. The registration is not successful. The system prompts the user with an error.

5.2.6 SIFIS-UC-06: Unregister device

Use Case #	SIFIS-UC-06
Goal in Context	Removing a device from the list of registered ones in the SIFIS-Home network.
Scope & Level	User Goal
Preconditions	Resident user is logged in the system.
Success End Conditions	The selected device is no-longer present in the list of registered devices of the SIFIS-Home network.
Failed End Condition	No changes in the list of registered devices.
Primary, Secondary Actors	Resident User
Trigger	-
Description	1. The user opens the feature “Unregister device”
	2. The system prompts with the list of registered devices to the SIFIS-Home network.
	3. The user selects the device to unregister.
	4. The system shows the information of the device to unregister.
	5. The user confirms the decision of unregistering the device.
	6. The system prompts the user that the operation is successful.
Extensions	3a. The user quits. The use case ends with failure.
	5a. The user aborts the decision of unregistering the device. The use case ends with a failure.
	6a. It is not possible to unregister the device. The system prompts the user with an error.

5.2.7 SIFIS-UC-07: Configure device

Use Case #	SIFIS-UC-07
Goal in Context	Changing the settings for a device in the list of registered devices in the SIFIS-Home network.
Scope & Level	User Goal
Preconditions	Resident user is logged in the system.
Success End Conditions	The desired modifications in the settings are successfully applied for the selected device.

Failed End Condition	No changes in the settings.
Primary, Secondary Actors	Resident User
Trigger	-
Description	1. The user opens the feature “Device settings”
	2. The system prompts with the list of registered devices to the SIFIS-Home network.
	3. The user selects the device to configure.
	4. The system shows the configuration options for the user.
	5. The user selects the desired configuration options and clicks “save”.
	6. The system asks confirmations to save the changes.
	7. The user confirms the changes.
	8. The system prompts the user that the operation is successful.
Extensions	3a. The user quits. The use case ends with failure.
	5a. The user quits. The use case ends with failure.
	7a. The user does not confirm the changes. The use case goes back to step 4.
	8a. The system is not able to propagate the desired modifications to the configuration. The user is prompted with an error message.
	8b. An external event alters the context against the current configuration. The devices act autonomously to change the context according to configuration.

5.2.8 SIFIS-UC-08: Installing third party applications

Use Case #	SIFIS-UC-08
Goal in Context	Install new functionalities in specific devices by installing third party applications.
Scope & Level	User Goal
Preconditions	Resident user is logged in the system.
Success End Conditions	The application is integrated in the SIFIS-Home system.
Failed End Condition	The application violates the smart home policies and is not installed.
Primary, Secondary Actors	Resident User, Maintainer
Trigger	-
Description	1. The user opens the app marketplace and selects a new app.
	2. The user is notified about security and safety aspects of the app.
	3. The user confirms the intention to install the applications.
	4. App information are matched with smart home policies. The system installs the application and notifies it to the user.
Extension	2a. The user does not want to install the application: the use case ends with failure.
	4a. The app is not compatible with the smart home policies. The use case ends with failure.

5.2.9 SIFIS-UC-09: Configure policies to restrict/handle access to functionalities

Use Case #	SIFIS-UC-09
Goal in Context	Define policies and access rights to smart home functionalities to the various users and installed applications.

Scope & Level	User Goal
Preconditions	- Administrator is logged in the SIFIS-Home framework - Administrator has a smart device with GUI
Success End Conditions	The administrator successfully configures enforceable policies.
Failed End Condition	The administrator is not able to configure enforceable policies.
Primary, Secondary Actors	Administrator, Maintainer
Trigger	-
Description	1. The administrator opens the configuration panel on a smart device with GUI or remotely from a PC or smartphone and selects the feature to handle policies.
	2. The system asks the administrator to select the user/application to configure.
	3. The administrator selects a specific user or user group, or installed applications.
	4. The system shows the list of action/resources that can be allowed or forbidden.
	5. The administrator selects the list of allowed/forbidden action/resources for the user/application.
	6. The system saves the change. The policy enforcement starts upon saving. User is notified.
Extensions	3a. User leaves. Use case ends with failure.
	5a. User leaves. Use case ends with failure.
	6a. The change cannot be saved. Use case ends with failure.

5.2.10 SIFIS-UC-10: Configure profiles

Use Case #	SIFIS-UC-10
Goal in Context	define different usage profiles based on involved users, time of the day and security preferences.
Scope & Level	User Goal
Preconditions	- User is logged in the SIFIS-Home framework - User interacts with a smart device accepting voice commands or has a GUI.
Success End Conditions	The framework is able to define profiles to provide different working properties and conditions.
Failed End Condition	Profiles are not configurable or do not activate when needed.
Primary, Secondary Actors	Resident User
Trigger	-
Description	1. The resident opens the configuration panel on a smart device with GUI or remotely from a PC or smartphone, or gives a voice command, and selects the feature to configure profiles.
	2. The system shows the menu for configuration of profiles.
	3. The resident specifies their preferences and enforcement conditions.
	4. The system shows the summary of preferences and asks the user for confirmation.
	5. The user confirms.
	6. The preference is saved and enforced according to specified conditions. The user is notified.
Extensions	3a. User leaves. Use case ends with failure.

	5a. User leaves or does not confirm. Use case ends with failure.
	6a. It is impossible to save the configuration. Use case ends with failure.
	7. An external event alters the context against the current configuration. The devices act autonomously to change the context according to configuration.

5.2.11 SIFIS-UC-11: Control Statistics and Analytics

Use Case #	SIFIS-UC-11
Goal in Context	Visualize analysis about the system working.
Scope & Level	User Goal
Preconditions	- Administrator is logged in the SIFIS-Home network - Administrator has a smart device with GUI
Success End Conditions	The framework provides graphic statistics and analytics about the system.
Failed End Condition	The required information is not shown to the administrator.
Primary, Secondary Actors	Administrator
Trigger	-
Description	1. The administrator open the feature “system analytics”.
	2. The system opens the analytics menu with menu voices about device, user and profile usage.
	3. The administrator selects the device, user or profile for which he wants to see usage analytics.
	4. The system shows a GUI with the required analytics.
Extensions	3a. User leaves. The use case ends with failure.
	4a. The system is not able to provide the required analytics. The use case ends with failure.

5.2.12 SIFIS-UC-12: Remote Configuration of devices

Use Case #	SIFIS-UC-12
Goal in Context	Configure specific device functionalities from remote.
Scope & Level	User Goal
Preconditions	- Configurer is logged in the System - Configurer is enabled for the configuration of the smart home by the administrator.
Success End Conditions	The framework provides a GUI on which the configurer successfully sets up device functionalities.
Failed End Condition	Device configuration is not possible.
Primary, Secondary Actors	Configurer
Trigger	-
Description	1. The configurer opens the control panel for configuration.
	2. The system shows the houses they can manage.

	3. The configurer selects the smart home where the device is located.
	4. The system shows the list of devices in the smart home.
	5. The configurer selects the device to configure.
	6. The system shows the configuration options for the device.
	7. The configurer selects the desired configuration options and selects to save the changes.
	8. The system asks confirmations to save the changes.
	9. The user confirms the changes.
	10. The system prompts the user that the operation is successful.
Extensions	3a. The user leaves. Use case ends with failure.
	4a. It is not possible to retrieve the list of devices. The use case ends with failure.
	5a. The user leaves. Use case ends with failure.
	7a. The user leaves. Use case ends with failure.
	9a. The user leaves. Use case ends with failure.
	10a. It is not possible to apply the configuration changes. Use case ends with failure.

5.2.13 SIFIS-UC-13: Remote Configuration of policies

Use Case #	SIFIS-UC-13
Goal in Context	Define policies and access rights to smart home functionalities to the various users and installed applications, from remote.
Scope & Level	User goal
Preconditions	The smart home has Internet connectivity. The administrator allows access to the configurator. The configurer is logged in the system.
Success End Conditions	The framework provides a GUI on which the administrator successfully configures device functionalities.
Failed End Condition	Device configuration is not possible.
Primary, Secondary Actors	Configurer, Administrator
Trigger	User opening the policy configuration panel.
Description	1. The configurer opens his control panel for configuration of policies.
	2. The system shows the houses he can manage.
	3. The Configurer selects the smart home where the policy must be changed.
	4. The System shows the list of policies configured in the smart home.
	5. The Configurer selects the police to configure.
	6. The system shows the configuration options for the policy.
	7. The configurer selects the desired configuration options and clicks “save”.
	8. The system asks confirmations to save the changes.
	9. The user confirms the changes.
	10. The system prompts the configurator that the operation is successful and notifies the administrator about the policy.
	11. The smart home administrator accepts the policy.
Extensions	3a. The user leaves. Use case ends with failure.
	4a. It is not possible to retrieve the list of devices. The use case ends with

	failure.
	5a. The user leaves. Use case ends with failure.
	7a. The user leaves. Use case ends with failure.
	9a. The user leaves. Use case ends with failure.
	10a. It is not possible to apply the policy changes. Use case ends with failure.
	11a. The administrator does not accept the policy change. The use case ends with failure.

5.2.14 SIFIS-UC-14: Remote handling of emergency situations

Use Case #	SIFIS-UC-14
Goal in Context	Configurer wants to access to the smart home functionality remotely in order to quickly react to an emergency.
Scope & Level	User goal
Preconditions	- Configurer is registered in the SIFIS-Home framework - Configurer is allowed by the Administrator for the management of emergency situations in the smart home.
Success End Conditions	The framework allows the configurer to log in remotely to manage emergency situations from remote.
Failed End Condition	The configurer cannot access the smart home environment remotely and react to the emergency situation.
Primary, Secondary Actors	Configurer
Trigger	A physical or software intrusion to the SIFIS-Home network is detected.
Description	1. The system notifies the Configurer about the intrusion.
	2. The configurer selects the action to take to act against the intrusion.
Extensions	1a. Authorities are contacted to give assistance.
	1b. Cameras record the intrusion (in case of physical intrusion).

5.3 *Catalogue of use cases*

In Table 5, we report the mapping between the use cases and the user stories. Each user story has to be matched to at least a use case.

	SIFIS-US-01	**SIFIS-US-02**	**SIFIS-US-03**	**SIFIS-US-04**	**SIFIS-US-05**	**SIFIS-US-06**	**SIFIS-US-07**
SIFIS-UC-01	***X***						
SIFIS-UC-02	***X***						
SIFIS-UC-03			***X***				
SIFIS-UC-04				***X***			
SIFIS-UC-05					***X***		
SIFIS-UC-06					***X***		
SIFIS-UC-07		***X***					
SIFIS-UC-08						***X***	
SIFIS-UC-09		***X***					
SIFIS-UC-10							***X***
SIFIS-UC-11		***X***					
SIFIS-UC-12		***X***					
SIFIS-UC-13		***X***					
SIFIS-UC-14			***X***				

Table 5: Mapping between use cases and user stories

6 Requirements

In Table 6, we report the Functional Requirements that have been elicited for the SIFIS-Home system. For each requirement, we report a brief description, the use case(s) to which it is correlated (or none, if it is a general requirement of the system), the type and priority of the requirement, a unique ID for the requirements document, the pointers to corresponding non-functional requirements and their types.

We prioritize the requirements according to the following scheme:

- ***Critical***: a requirement that must be fulfilled for the execution of the essential system functionalities.
- ***Standard***: a requirement that should be fulfilled but whose absence will not impair the execution of the essential system functionalities.
- ***Optional***: a feature that may be fulfilled but whose absence will not impair the execution of the system functionalities.

6.1 *Functional Requirements*

In the following, we report the list of functional requirements identified for the SIFIS-Home architecture. All requirements identified are derived from the defined use cases. Table 6 also reports the mapping with the non-functional requirements (NFR) defined in the following section and the prioritisation level.

ID	Req. description	UC	Priority	NFR-Req. ID	NFR-Req. Type
F-01	The SIFIS-Home framework shall provide means of identifying the users inside the smart home through biometrics.	UC1	C	PE-03 US-09 DE-01	Performance Usability Dependability
F-02	The SIFIS-Home system shall provide means of authentication to the resident users and administrators inside the smart home.	UC1	S	PE-01	Performance
F-03	The SIFIS-Home system shall match read biometrics with a database of stored ones.	UC1	S	PE-01 PE-04	Performance Performance
F-04	The system shall activate features based on the user identity.	UC1	S	PE-05	Performance
F-05	The system shall activate a guest profile when the identity of the biometrics is not recognised.	UC1	S	PE-05	Performance
F-06	The SIFIS-Home system shall provide means of recognition of allowed users in the smart home.	UC1, UC3	C	PE-02	Performance
F-07	The SIFIS-Home system shall provide Automatic Speech Recognition (ASR) to provide the residents the facility to control their home appliances through their speech.	UC2	C	PE-02 PE-06 DE-02 DE-03	Performance Performance Dependability Dependability
F-08	The SIFIS-Home system shall receive and interpret the voice commands provided by the user.	UC2	C	PE-07	Performance
F-09	The SIFIS-Home system shall be able to execute all the recognisable voice commands.	UC2	C	PE-08	Performance
F-10	The SIFIS-Home system shall signal the presence of an intruder when the identity is not recognised and no residents are at home.	UC3	C	US-10	Usability
F-11	The SIFIS-Home system shall record intruder actions through cameras.	UC3, UC14	S	DE-04	Dependability
F-12	The SIFIS-Home system shall store the identity of the intruder if the face is recognised.	UC3, UC14	O	DE-05	Dependability
F-13	The SIFIS-Home system may grant the access to recording to the maintainer.	UC3, UC14	S	PE-09 US-10	Performance Usability

F-14	The SIFIS-Home system may allow to contact police to receive assistance in case of intrusions.	UC3, UC14	O	PE-10	Performance
F-15	The SIFIS-Home system shall provide means of identifying anomaly behaviours inside the smart home.	UC3, UC14	C	PE-10	Performance
F-16	The SIFIS-Home system shall provide means of recognition of allowed users in unusual locations or performing dangerous actions.	UC3, UC14	S	PE-10	Performance
F-17	The SIFIS-Home system shall provide means of recognition of forbidden objects inside the smart home.	UC3, UC14	S	PE-10	Performance
F-18	The SIFIS-Home system shall provide means of recognition of allowed objects in unusual positions.	UC3, UC14	O	PE-10	Performance
F-19	The SIFIS-Home system shall identify and isolate infected devices.	UC4	C	PE-11 US-11 DE-06	Performance Usability Dependability
F-20	The SIFIS-Home system shall notify the user when malware is detected.	UC4	C	PE-12 US-11	Performance Usability
F-21	The SIFIS-Home system may execute self-healing algorithms to transfer functionalities of isolated devices to the others.	UC4	C	PE-13 DE-06	Performance Dependability
F-22	The SIFIS-Home system may allow means of verifying that the malware has not spread to other devices.	UC4	S		
F-23	The SIFIS-Home system shall allow the resident user to register a new device.	UC5	C	PE-14 US-12 DE-07	Performance Usability Dependability
F-24	The SIFIS-Home system shall provide a list of the registered devices to the user along with their characteristics.	UC5, UC6, UC7, UC12	C	PE-15	Performance
F-25	The SIFIS-Home system shall allow the user to unregister a registered device.	UC6, UC12	C	PE-16 DE-08	Performance Dependability
F-26	The SIFIS-Home systems shall expose a section where the resident users and administrators can configure the devices.	UC7	C	PE-17, PE-18 US-13 US-14 DE-09 DE-10	Performance Performance Usability Usability Dependability Dependability
F-27	The SIFIS-Home system shall prompt the user when unsolicited configuration changes are propagated to the devices.	UC7, UC12	S	PE-18	Performance
F-28	The SIFIS-Home system must provide a marketplace function for the download of third-party applications on smart devices.	UC8	C	PE-19 US-15 DE-11	Performance Usability Dependability
F-29	The SIFIS-Home system shall provide information about the safety and security aspects of an application to the user.	UC8	C		
F-30	The SIFIS-Home system must provide a feature to show the administrator a list of currently active policies.	UC9, UC13	S	PE-22	Performance
F-31	The SIFIS-Home system must allow the administrator to configure the policies to restrict/enable access to functionalities.	UC9, UC13	C	DE-12	Dependability
F-32	The SIFIS-Home system must allow the administrator to configure policies for groups of users.	UC9, UC13	S	PE-20 US-16	Performance Usability
F-33	The SIFIS-Home system must allow the administrator to configure policies for group of devices.	UC9, UC13	S	PE-21 US-17 DE-12	Performance Usability Dependability
F-34	The SIFIS-Home system must allow the administrator to see the list of features/resources that are allowed or forbidden for all groups of users.	UC9, UC13	S	DE-12	Dependability
F-35	The SIFIS-Home system must allow the administrator to see the list of features/resources that are allowed or forbidden for all groups of devices.	UC9, UC13	S		
F-36	The SIFIS-Home system must provide the user with a feature to list all the currently available profiles.	UC10	S		
F-37	The SIFIS-Home system must allow the user to configure its profiles.	UC10	S	PE-23 US-18	Performance Usability

				DE-13	Dependability
F-38	The SIFIS-Home system must allow the user to switch his/her current profile.	UC10	O	PE-24 US-19 DE-14	Performance Usability Dependability
F-39	The SIFIS-Home system should show the user a summary of the preferences associated to its current profile.	UC10	O		
F-40	The SIFIS-Home system should show notifications to the user when the current profile is changed.	UC10	O		
F-41	The SIFIS-Home system should offer aggregate analytics and statistics about the usage of devices to the administrator.	UC11	S	PE-25 US-20 DE-15	Performance Usability Dependability
F-42	The SIFIS-Home system should offer aggregate analytics and statistics about the usage of profiles to the administrator.	UC11	S	PE-26	Performance
F-43	The SIFIS-Home system must offer remote log-in features to configurer/maintainers user profiles.	UC12, UC13	S	PE-27 DE-16	Performance Dependability
F-44	The SIFIS-Home system shall offer a panel with the remote houses that can be managed by a maintainer.	UC13	S		
F-45	The SIFIS-Home system must offer the maintainer a panel to react in case of intrusions.	UC14	S		
F-46	The SIFIS-Home system shall store personal resident information (video, audio, text).	UC10	C		

Table 6. The list of elicited functional requirements for the SIFIS-Home system

6.2 *Non-Functional Requirements*

In Table 7, we report the list of non-functional requirements which have been extracted from the above list of functional requirements. This set of requirements will pose the basis to define both the SIFIS-Home architecture and SIFIS-Home software framework. Satisfying these requirements is needed to ensure satisfaction of the functional requirements. The non-functional requirements are divided in Performance (PE), Reliability (RE), Availability (AV), Usability (US) and Dependability (DE). Non-functional Security requirements are collected in the next section.

Req. ID	Req. description	FR	Priority
PE-01	The user authentication shall happen in less than 2s.	F-02 F-03	Critical
PE-02	The user recognition (identification) shall happen in less than 2s.	F-06 F-07	Critical
PE-03	Identification through biometrics should be performed in less than 5 seconds.	F-01	Standard
PE-04	Biometric-based authentication should be performed in less than 5 seconds.	F-03	Standard
PE-05	Activation of features based on user identity (biometric recognition) should be performed in less than 5 seconds.	F-04 F-05	Standard
PE-06	Recognition of the start of an interaction through voice command should be performed in less than 2 seconds.	F-07	Standard
PE-07	The interpretation of the voice commands provided by the user should be performed in less than 2 seconds.	F-08	Standard
PE-08	The execution of the commands should be performed in less than 5 seconds.	F-09	Standard
PE-09	The mantainer must be able to access and watch the recording in less than one minute.	F-13	Standard
PE-10	The SIFIS-Home system shall contact police to receive assistance in less than 30 seconds.	F-14	Optional
PE-11	Identification of installed malware should be completed in less than 60 seconds from the execution of malware.	F-19	Optional

PE-12	The user should be informed of the presence of a malware in 5 seconds after the malware is recognised.	F-20	Standard
PE-13	Self-healing algorithms should be started in less than 60 seconds if available when malware is recognised.	F-21	Critical
PE-14	The registration of a new device should be completed in less than 30 seconds.	F-23	Standard
PE-15	The list of registered devices shall be shown by the SIFIS-Home system in less than 30 seconds.	F-24	Standard
PE-16	The de-registration of a device should be completed in less than 30 seconds.	F-25	Standard
PE-17	The configuration changes should be propagated successfully in less than 30 seconds.	F-26	Critical
PE-18	The current configuration of a device should be retrieved in less than 10 seconds.	F-26	Standard
PE-19	The marketplace should be accessed in less than 60 seconds.	F-28	Standard
PE-20	The configuration of policies for groups of users should be applied in less than 60 seconds.	F-32	Critical
PE-21	The configuration of policies for groups of devices should be applied in less than 60 seconds.	F-33	Critical
PE-22	The list of policies should be retrieved in less than 30 seconds.	F-30	Standard
PE-23	The configuration of profiles should be applied in less than 60 seconds.	F-37	Critical
PE-24	The change of current profile should be performed in less than 60 seconds.	F-38	Critical
PE-25	The statistics about usage of devices should be presented to the administrator in less than 30 seconds.	F-41	Standard
PE-26	The statistics about usage of profiles should be presented to the administrator in less than 30 seconds.	F-42	Standard
PE-27	Remote log-in should be performed in less than 60 second.	F-43	Critical
RE-01	The system shall not fail more than once a week (in average).	All	Critical
RE-02	The system shall not take more than one day to be repaired (in average).	All	Critical
AV-01	The system shall be available 99% of the time.	All	Critical
AV-02	The SIFIS-Home system shall ensure basic services availability in case of system failures.	All	Critical
US-01	The system shall be easy to use for average tech users.	All	Critical
US-02	The SIFIS-Home system shall anticipate strange, dangerous, or critical situations and raise an alert.	All	Critical
US-03	The SIFIS-Home system shall be autonomous and learn based on the users' habits.	All	Critical
US-04	The SIFIS-Home system shall consider special cases in its design, such as colour blindness.	All	Optional
US-05	The SIFIS-Home system shall preserve consistency among all devices, related database and constraints.	All	Critical
US-06	The SIFIS-Home hardware components should be easy to use for the elderly and users with no engineering background.	All	Optional
US-07	The SIFIS-Home system shall have an explorable interface.	All	Standard
US-08	Proper and easy hardware installation should be considered.	All	Standard
US-09	The identification through biometrics should be performed by the system in a radius of at least 10 meters from the device.	F-01	Standard
US-10	An untrained user should be able to recognise an intrusion in the SIFIS-Home system and contact the authorities in less than 1 minute.	F-10 F-13	Critical
US-11	An untrained user should be able to recognise a software intrusion in less than one minute.	F-19 F-20	Critical

US-12	An untrained user should be able to perform the device registration procedure in less than 5 minutes.	F-23	Standard
US-13	An untrained user should be able to perform the device de-registration procedure in less than 5 minutes.	F-26	Standard
US-14	An untrained user should be able to perform the configuration of devices in less than 5 minutes.	F-26	Standard
US-15	An untrained user should be able to perform the installation of an application in less than 5 minutes.	F-28	Standard
US-16	An untrained user should be able to complete the configuration of policies for groups of users in less than 5 minutes.	F-32	Standard
US-17	An untrained user should be able to complete the configuration of policies for groups of devices in less than 5 minutes.	F-33	Standard
US-18	An untrained user should be able to complete the configuration of profiles in less than 5 minutes.	F-37	Standard
US-19	An untrained user should be able to perform a profile change in less than 30 seconds.	F-38	Standard
US-20	An untrained user should be able to visualize and interpret the statistics in less than 5 minutes.	F-41	Standard
DE-01	The identification through biometrics should be performed correctly in more than 95% cases.	F-01	Critical
DE-02	The start of interaction command should be recognised properly in more than 99% of cases.	F-07	Critical
DE-03	The commands to execute should be recognised properly in more than 95% of cases.	F-07 F-08	Critical
DE-04	Record of intrusions must be available for six months after the recording.	F-11	Standard
DE-05	Identity of the intruders must be available for six months after the recording.	F-12	Standard
DE-06	Core functionalities should be replicated on multiple devices to avoid single points of failure.	F-21	Critical
DE-07	The registration of a new device should be successful in at least 99% of the cases.	F-23	Critical
DE-08	The de-registration of a new device should be successful in at least 99% of the cases	F-25	Critical
DE-09	The configuration changes should be propagated successfully to the devices in more than 99% of times.	F-26	Critical
DE-10	The SIFIS-Home system should be able to restore the previous configurations if there are error in the application of configuration changes.	F-26	Standard
DE-11	The installation of the selected app should be completed successfully in at least 95% of cases.	F-28	Critical
DE-12	The application of policies should be completed successfully in at least 99% of cases.	F-31 F-34 F-33	Critical
DE-13	The configuration of profiles should be completed successfully in at least 99% of cases.	F-37	Critical
DE-14	The change of current profile should be completed successfully in at least 99% of cases.	F-38	Critical
DE-15	The statistics must be shown correctly in at least 99% of cases.	F-41	Critical
DE-16	Remote log-in for the configurer should be successful in at least 99% cases.	F-43	Critical
DE-17	The SIFIS-Home system should be able to distribute the processing among multiple machines in different places if required.	All	Critical
DE-18	The SIFIS-Home system is required to be fault tolerant, it should continue to operate, even if one or more of the nodes fail.	All	Critical

DE-19	The SIFIS-Home system is required to be scalable dynamically by adding or removing nodes according to demand.	All	Critical

Table 7. A list of elicited non-functional requirements for the SIFIS-Home system

6.3 *Security Requirements*

Security requirements are a specific set of non-functional requirements which do not directly derive from specific functional requirements. Security requirements derive instead from laws and regulations, security standards, protocols and best practices. To discuss the security requirements, we will refer to the Open Security Architecture (OSA) standard to describe security requirements, dividing them into:

Testable Security Requirements (TSR) are security-related functions that the system must be able to perform. These requirements are testable, so one should be able to create test cases for them.

Non-Testable Security Requirements (NTSR) cannot be tested to be either working or not in a black-and-white fashion, but they can be measured by using metrics. Non-testable security requirements are of two sub-classes: one-time and continuous security requirements. One-time security requirements are implemented within a system and tested to be ensure that they are fulfilled. For example, user passwords are hashed to protect system against rainbow table attacks. Continuous security requirements are constraints on other requirements and may influence the system at any time. These include, for example, the validation of parameters in a HTTP POST request.

Table 8 distinguishes and summaries the characteristics between TSR and NTSR whereas Figure 10 connects the sources of security requirements with TSR and NTSR.

Security requirements				
Type	**Describes**	**Derived from**	**Example**	**Validation**
TSR	Security services, that the system has to provide.	Best practices, Policies, Regulations	Application X has to be possible to use only with user rights given by an administrator.	Testable
NTSR • One-time • Continuous	Architectural security requirements.	Architectural principals, Good practice, Standards	Availability, Integrity, Confidentiality	Measurable

Table 8. Characteristics of TSR and NTSR

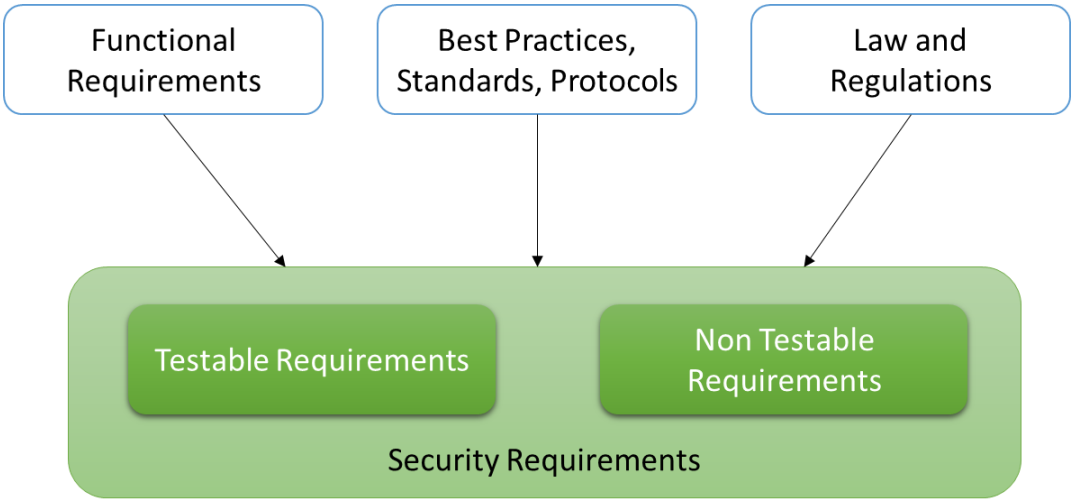


Figure 10. Security Requirements Gathering

Requirements coming from standards and protocols are generally intended to fulfil the standard objectives of security, namely:

Confidentiality is the property by which unauthorized users do not have access to information.

Integrity is the property by which information cannot be unnoticeably modified or destructed by unauthorized users, and information completeness and accuracy is maintained all over the information life cycle.

Availability in information systems is the property by which information and services have to be available when needed and requested. Consequently, security controls are used to protect information systems from service disruptions. Security controls also ensures high technical reliability of the communication channels.

From these basic security objectives should be derived also those requirements related to system *dependability*, management of *access control*, *authentication* and *authorisation*. Law and regulations requirements mainly derive from GDPR and the NIST regulation on data management, which provide requirements related to management and storage of data. In the SIFIS-Home project a deep analysis of these requirements is performed in Task 2.4. A particular attention for what concerns these requirements is dedicated to users' data privacy, in order to empower the user with a complete control on the data produced and collected by the framework and by supporting the development of analytics services privacy preserving and based on the minimum needed privilege paradigm.

Table 9 lists the elicited security requirements for the SIFIS-Home system. For the prioritisation of these security requirements, the same standard already used for functional and non-functional requirement will be used.

Req ID	Requirement Description.	FR	Testable	Priority
SE-01	APIs for the communication with internal devices must be secured.	C-02	NT	Critical
SE-02	APIs for the communication with external devices must be secured.	C-04	NT	Critical
SE-03	Personal data stored must be encrypted.	F-49 F-56	T	Critical

		S-04		
SE-04	The system shall protect and avoid disclosure of sensitive information.	F-56	NT	Critical
		F-62		
		S-04		
SE-05	The SIFIS-Home system shall prevent data alteration or deletion.	F-56	NT	Critical
		F-61		
		S-04		
SE-06	Wifi access should be protected against known WiFi security attacks.	C-01	T	Critical
		C-03		
SE-07	Biometrics must be stored safely in the SIFIS-Home database.	F-03	NT	Critical
SE-08	Log-in information should be stored in a protected database.	F-62	NT	Critical
SE-09	The information about the registered devices, their characteristics and their configurations should be stored in a protected database.	F-25	NT	Critical
		F-38		
		F-44		
SE-10	The information about policies should be stored in a protected database.	F-33	NT	Critical
SE-11	The information about user profiles and configuration aspects should be stored in a protected database.	F-39	NT	Critical
		F-42		
		F-44		
SE-12	Data paths should be identified to allow data tracking and detect data leaving the smart-home perimeter, according to policies.	General	T	Critical
SE-13	Data confidentiality shall be ensured all the time.	General	NT	Critical
SE-14	The system should not be affected by MITM attacks.	General	T	Critical
SE-15	Software and apps shall only be installed with authorisation of the smart home administrator or resident users.	General	T	Critical
SE-16	Users must be able to configure and allow the usage of data by the SIFIS-Home framework and third party software.	General	T	Critical
SE-17	Anomalous device behaviours should be identified and signalled in less than 60 seconds.	General	T	Critical
SE-18	Minimum needed privilege principle must always be enforced.	General	NT	Critical
SE-19	Access to devices functionalities should be protected and controlled	General	NT	Critical
SE-20	Access to critical functionalities and services of the SIFIS-Home framework shall be protected and controlled.	General	NT	Critical
SE-21	Privacy preferences shall be configurable for data, analytics and functionalities.	General	NT	Critical
SE-22	Analytics shall be able to work with anonymized data when possible.	General	NT	Critical
SE-23	The SIFIS-Home architecture shall be resilient to network-based attacks.	General	T	Critical
SE-24	The SIFIS-Home architecture shall be resilient DoS attacks.	General	T	Critical
SE-25	The SIFIS-Home architecture shall be resilient to sybil attacks.	General	T	Critical
SE-26	The SIFIS-Home architecture shall be resilient to device compromising attacks.	General	T	Critical
SE-27	The SIFIS-Home architecture shall be resilient to Internet connection failure.	General	T	Critical
SE-28	The SIFIS-Home architecture shall be resilient to physical device damage or failure.	General	T	Critical
SE-29	Devices must have unique identifiers.	General	NT	Critical

Table 9. A list of elicited security requirements for the SIFIS-Home system

6.4 *Mapping of Requirements on Use Cases*

Table 10 gives a visual representation and summary of the mapping of the functional requirements defined in Section 6.1 to the use cases that were used to generated them.

	SIFIS-UC-1	SIFIS-UC-2	SIFIS-UC-3	SIFIS-UC-4	SIFIS-UC-5	SIFIS-UC-6	SIFIS-UC-7	SIFIS-UC-8	SIFIS-UC-9	SIFIS-UC-10	SIFIS-UC-11	SIFIS-UC-12	SIFIS-UC-13	SIFIS-UC-14
F-01	x													
F-02	x													
F-03	x													
F-04	x													
F-05	x													
F-06	x		x											
F-07		x												
F-08		x												
F-09		x												
F-10			x											
F-11			x											x
F-12			x											x
F-13			x											x
F-14			x											x
F-15			x											x
F-16			x											x
F-17			x											x
F-18			x											x
F-19				x										
F-20				x										
F-21				x										
F-22				x										
F-23					x									
F-24					x	x	x					x		
F-25						x						x		
F-26							x							
F-27							x					x		
F-28								x						
F-29								x						
F-30									x				x	
F-31									x				x	
F-32									x				x	
F-33									x				x	
F-34									x				x	
F-35									x				x	
F-36										x				
F-37										x				
F-38										x				
F-39										x				
F-40										x				
F-41											x			
F-42											x			
F-43												x	x	
F-44													x	
F-45														x
F-46										x				

Table 10. Mapping of Requirements to Use Cases

7 Evaluation and Validation

The validation will be based on the acceptance tests defined for each use case. In particular, for each defined acceptance test, will be verified through the activities of WP5 and WP6, if the functionalities of the framework are able to satisfy or not each specific acceptance test. Specific acceptance tests will also be defined for the security testable requirements. Later, the validation methodology will be refined and better detailed in deliverable D1.2, following the integration of contributions from D3.1 and D4.1.

8 Conclusion

In this deliverable we have presented the work done to gather functional, non-functional and security requirements for the SIFIS-Home framework.

This deliverable has condensed the activities of Task 1.1 and Task 1.2, which are related to a background study of techniques for requirement gathering as well as the rationale behind the selection of a specific methodology. It has been driven by the lack of pilot-specific requirements at this stage, hence completely oriented towards the definition of an architecture by analysing its basic functionalities.

A set of seven user stories has been defined, from which we have extracted 14 use cases together with their acceptance tests. Finally, the sets of functional, non-functional and security requirements have been presented. The list of presented requirements is not final and will be analysed as part of the activities of WP3 and WP4. This deliverable will serve as an input to D1.2 and D1.3, which will also include the feedback provided from D3.1 and D4.1.

9 References

[Berander, 2005] Berander, P., & Andrews, A. (2005). Requirements prioritisation. In Engineering and managing software requirements (pp. 69-94). Springer, Berlin, Heidelberg.

[Cockburn, 1998] Cockburn, A. (1998). Use Case Template. CU-Boulder: Computer Science.

[Cohn, 2004] Cohn, M. (2004). Advantages of user stories for requirements. InformIT Network, available at: http://www. informit. com/articles.

[Crespi] Crespi, V., Galstyan, A., & Lerman, K. (2008). Top-down vs bottom-up methodologies in multi-agent system design. Autonomous Robots, 24(3), 303-313.

[Dennis, 2009] Dennis, A., Wixom, B. H., & Tegarden, D. (2009). Systems Analysis and Design UML Version 2.0. Wiley.

[Jacobson, 2004] Jacobson, I. (2004). Use cases–Yesterday, today, and tomorrow. Software & Systems Modelling, 3(3), 210-220.

[Leffingwell, 2010] Leffingwell, D. (2010). Agile software requirements: lean requirements practices for teams, programs, and the enterprise. Addison-Wesley Professional.

[Lincke, 2012] Lincke, S. J., Knautz, T. H., & Lowery, M. D. (2012, June). Designing system security with UML misuse deployment diagrams. In 2012 IEEE Sixth International Conference on Software Security and Reliability Companion (pp. 57-61). IEEE.

[Ruhe, 2005] Ruhe, G. (2005). Decision support in requirements engineering. In Engineering and managing software requirements (pp. 267-286). Springer, Berlin, Heidelberg.

[Sharp, 1999] Sharp, H., Finkelstein, A., & Galal, G. (1999, September). Stakeholder identification in the requirements engineering process. In Proceedings. Tenth International Workshop on Database and Expert Systems Applications. DEXA 99 (pp. 387-391). Ieee.

[Sommerville, 1997] Sommerville, I., & Sawyer, P. (1997). RE: a good practice guide. John Wiley and Sons.

[Sommerville, 2011] Sommerville, I. (2011). Software engineering 9th Edition. ISBN-10, 137035152, 18.

Glossary

Acronym	Definition
DHT	Distributed Hash Table
FR	Functional Requirements
NFR	Non-functional requirement
OS	Operative System
P2P	Peer to Peer
SIFIS-Home	Secure Interoperable Full Stack Internet of Things for Smart Home
UC	Use case
US	User story
SD	Smart Device
NSSD	Not So Smart Device