



Katholieke  
Universiteit  
Leuven

Department of  
Computer Science

# Shared Internet Of Things Infrastructure Platform: Domain Analysis Software Architecture (H09B5a and H07Z9a) – Part 1

HAVENEERS-VERREYDT-LEMMENS

**Robin Haveneers (r0450702)**  
**Stef Verreydt (r0456110)**  
**Axel Lemmens (r0462440)**

Academic year 2016–2017

# Contents

<b>1</b>	<b>Domain analysis</b>	<b>2</b>
1.1	Domain models . . . . .	2
1.2	Domain constraints . . . . .	3
1.3	Domain constraints . . . . .	4
1.4	Glossary . . . . .	5
<b>2</b>	<b>Functional requirements</b>	<b>8</b>
2.1	Use case overview . . . . .	8
2.2	Detailed use cases . . . . .	10
2.2.1	<i>UC8</i> : Subscribe to application . . . . .	10
2.2.2	<i>UC4</i> : Add application . . . . .	11
2.2.3	<i>UC11</i> : Install new hardware . . . . .	12
<b>3</b>	<b>Non-functional requirements</b>	<b>13</b>
3.1	Availability . . . . .	13
3.1.1	<i>Av1</i> : Name of the quality attribute scenario . . . . .	13
3.2	Performance . . . . .	13
3.2.1	<i>P1</i> : Name of the quality attribute scenario . . . . .	13
3.3	Modifiability . . . . .	14
3.3.1	<i>M1</i> : Name of the quality attribute scenario . . . . .	14
3.4	Usability . . . . .	14
3.4.1	<i>U1</i> : Name of the quality attribute scenario . . . . .	14

# 1. Domain analysis

## 1.1 Domain models

This sections contains the domain models for our system. For readability reasons, the model is split into two parts. In the first figure the general domain model is shown, in the second figure the model containing the hardware and information flow is illustrated.

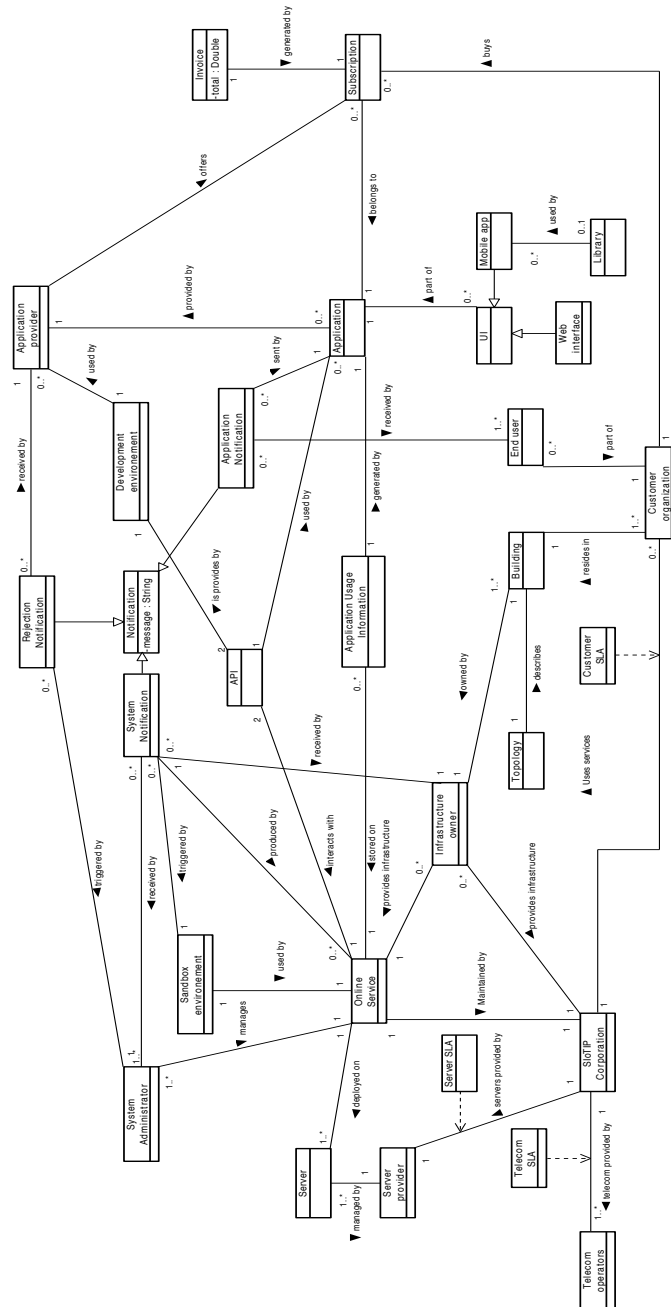


Figure 1.1: Domain model: General

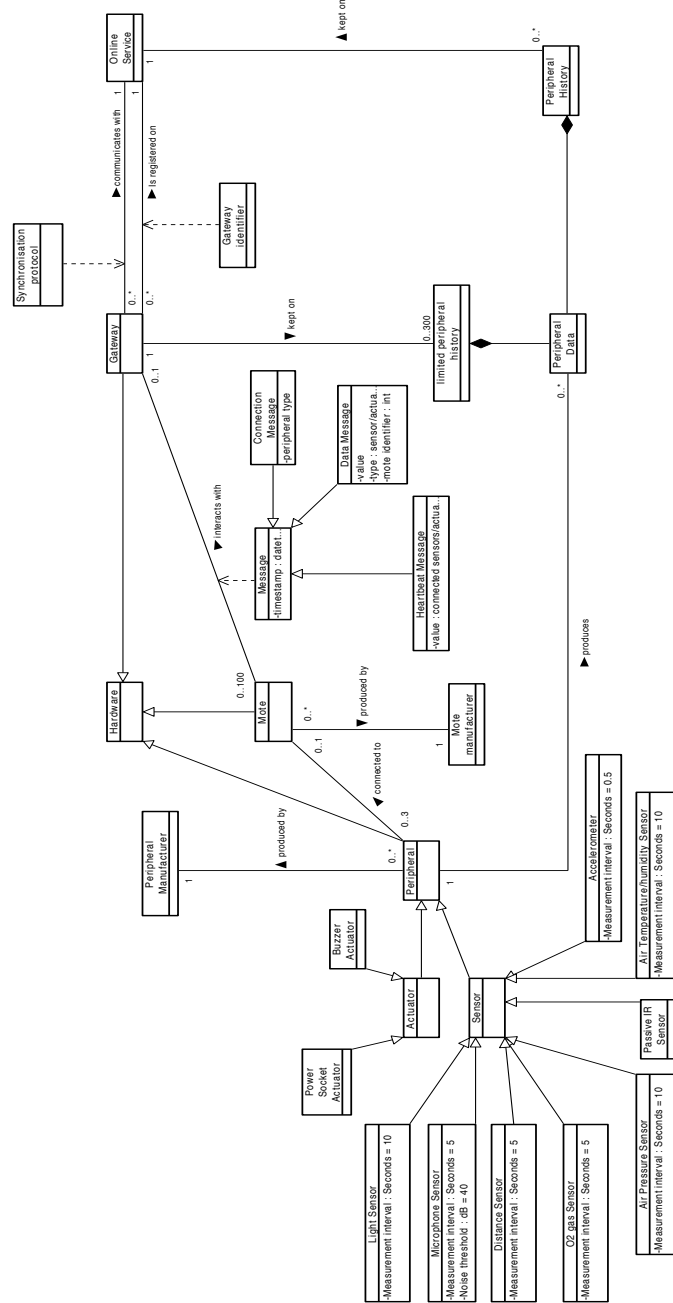


Figure 1.2: Domain model: Hardware and information flow

## 1.2 Domain constraints

In this section we provide additional domain constraints.

- A customer organization can only subscribe to an application that is already registered with the system.
- A end-user only receives notifications from applications that the customer organization he works for is subscribed to. He only receives these notifications if he/she is an individual who should receive notifications as configured in the application by the previously mentioned customer organization.

- The Online Service can only communicate with peripherals via their gateway.
- The connectivity between the network manager and the attached devices is done via 6LoWPAN.
- Application developers only receive rejection notifications about the applications they want to deploy.
- An application should be idle until all the necessary hardware is available and if the hardware is available it should be activated automatically. For example, a fire application requires at least 2 specific sensors per room.
- If a component of the hardware fails, the system should automatically switch to a nearby, similar device if possible, based on the topology provided by the infrastructure owner.
- Only persons who are allowed to do so, can issue commands to an application.
- Gateways contain limited storage space. Whenever they're running low on space, the oldest information should be removed.
- When communicating with the system, customer organisations, infrastructure owners, system administrators and application providers each have their own, personal, dashboard to do so.

## 1.3 Glossary

In this section, we provide a glossary of the most important terminology used in this analysis.

- **API:** Application programming interface. The interface used by applications to communicate with the online system and gateways.
- **Accelerometer:** Type of sensor that measures the acceleration in 3 dimensions.
- **Actuator:** Hardware device that has actions associated with it (e.g. switch, light, buzzer ...)
- **Air Pressure Sensor:** Type of sensor that measures the air pressure in bar.
- **Air Temperature/Humidity Sensor:** Type of sensor that measures the air temperature in degrees Celsius and humidity in percentage.
- **Application Notification:** A type of notification sent by the application to a relevant end-user.
- **Application Usage Information:** Data collected by applications and stored on the online service. Useful for providing insight in application usage and big data analytics.
- **Application provider:** Stakeholder that provides (develops and deploys) applications on the online system, to which one can subscribe.
- **Application:** Piece of 'software' running on the online service and gateway provided to customer organizations to subscribe to with the goal of simplifying their day to day business.
- **Building:** Place owned by infrastructure owner where customer organisations are housed. The infrastructure owners keep a topology about the building they own.
- **Buzzer:** Actuator that is able to play a sound.
- **Connection Message:** The message that is sent by the mote to the gateway whenever a new sensor or actuator is inserted.
- **Customer SLA:** The Service-Level Agreement between SLoTIP corporation and the customer organisation.
- **Customer organization:** The stakeholder in our system which subscribes to applications which optimize their workflow.

- **Data Message:** The message send by the mote from one of its connected devices to the gateway which contains the value it's reading, the mote identifier and a timestamp. This message also specifies whether it was sent by a sensor/actuator which is necessary to be able to correctly interpret the data.
- **Development environnement:** The environmennt available to application providers to freely test and debug their application before deploying it.
- **Distance Sensor:** Type of sensor which measure
- **End user:** A person involved in the customer organisation who interacts with applications.
- **Gateway identifier:** The identifier which is used to register the gateway with the Online Service.
- **Gateway:** A device that is connected to the online service and which provides access to sensors and actuators. It relays data from connected devices to the Online Service.
- **Hardware:** Umbrella concept used for all the connected devices (such as peripherals, gateways ...)
- **Heartbeat Message:** Periodical message sent by the mote to indicate it is alive. This message includes a list of connected sensors and actuators as well as a timestamp indicating when the message was generated.
- **Infrastructure owner:** Stakeholder in our system which manages the physical infrastructure as well as the hardware used as described in the topology. Customer organisations reside in these buidlings.
- **Invoice:** An invoice is associated with a subscription between a customer organization and an application provider. It specifies the fees necessary to be paid by the customer organization.
- **Library:** Set of methods that can be used by application providers in their mobile applications to simplify development.
- **Light Sensor:** Type of sensor that measures the amount of light in LUX.
- **Llimited peripheral history:** Small set of peripheral history maintained by the gateway. Useful for application to, for example, learn certain behaviourial patterns.
- **Message:** Umbrella term used for the different types of messages a mote can emit.
- **Microphone Sensor:** Type of sensor which allows detection of sound.
- **Mobile app:** One type of interface end-users can use to communicate with the online system.
- **Mote manufacturer:** Third party which produces the motes used by our system.
- **Mote:** Device that hosts sensors and actuators.
- **Notification:** Umbrella concept used for all the different types of notifications used by the system. Notifications are sent out on several events. The precise information depends on the type of event.
- **O<sub>2</sub> gas sensor:** Sensor that can measure the amount of O<sub>2</sub> gas in the air in percentage.
- **Online Service:** The collective concept that are the services provided via our online back-end. This service is used to test, develop and host applications, to store application configurations, keep track of sensor data, collect usage statistics etc.
- **Passive IR (Presence) Sensor:** Type of sensor which allows detection of motion achieved by taking pictures and comparing subsequent pictures.
- **Peripheral Data:** The data which is produced by peripherals/hardware. Hardwa
- **Peripheral History**
- **Peripheral Manufacturer:** The third party company that is responsible for the manufacturing of sensors and actuators.

- **Peripheral Type:** The type of the peripheral. Currently sensor and actuator are possible.
- **Peripheral:** Umbrella concept for sensors and actuators.
- **Power socket:** Type of actuator which can be turned on or off.
- **Rejection Notification:** The type of notification which is triggered by the sandbox environment if the application that is waiting to be deployed is rejected by our system. This notification is send to the application provider.
- **SIoTIP Corporation:** This stakeholder wants SIoTIP to be a succesful platform for deploying IoT devices. They have agreements with different third parties for providing an optimal platform with respect to availability, performance ..
- **Sandbox environement:** The environment which is used to test submitted applications before they become available on the Online Service. When there are problems with the application being tested, a system admininstrator performs a secondary, manual review. If he rejects the application, the application provider is notified and is required to fix the application.
- **Sensor:** A type of peripheral that produces measurements and sends this data to the gateway to be used in applications.
- **Server provider:** The third-party provider which provides (virtual or physical) servers to deploy the Online Service on.
- **Server:** The server (virtual or physical) on which the Online Service is deployed. They are provided by the third party server provider.
- **Server SLA:** The SLA between SIoTIP corporation and the server provider which, among others, specifies the up-time constraints and reaction time in case of a hardware failure.
- **Subscription:** An agreement between the application provider and a customer organisation in which is specified for which application a customer organisation is subscribed.
- **Synchronisation protocol:** The protocol used in the communication between the gateways and the online service.
- **System Administrator:** The person responsible to act when a submitted application is rejected. He or she can accept the application nonetheless or reject it as well which notifies the application provider.
- **System Notification:** A notification which notifies the system administrator when a submitted application needs attention. This notification is triggered by the sandbox environment.
- **Telecom SLA:** The SLA between the SIoTIP corporation and the telecom provider which specifes the the capabilities of these communication channels.
- **Telecom operators:** Third party who provides means for gateways to communicate with the Online Service and vice versa.
- **Topology:** Overview of the hardware of the infrastructure spread out over the building(s) he owns. This can be used when new, for example, new applications are added to find out which new hardware is necessary.
- **UI:** The interface of a application used by end users of a customer organisation.
- **Web interface:** A possible UI for an application used by end users to communict with the online service.

## 2. Functional requirements

### Use case model

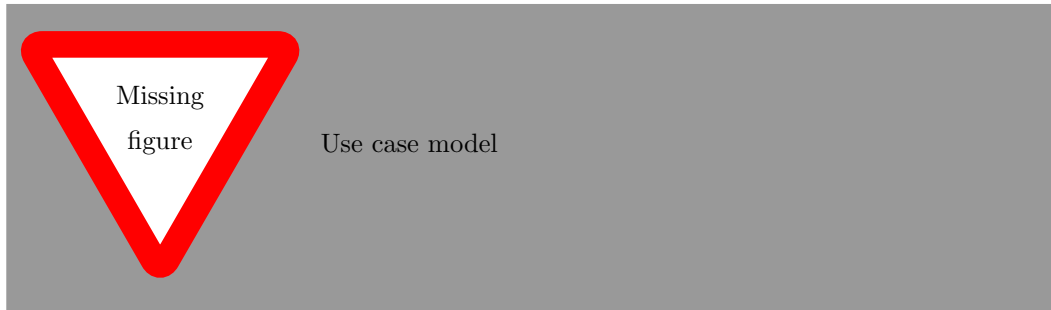


Figure 2.1: Use case diagram for the system.

### 2.1 Use case overview

**UC1: Log in** The user wishing to use the system provides his credentials. The system verifies the credentials and authenticates the user. If the provided credentials were not correct, the system does not authenticate the user.

**UC2: Log off** The user indicates he wants to log off from the system. The system logs him off.

**UC3: Enroll as application provider** The organization wishing to become an Application Provider contacts SIoTIP and they negotiate the contract. The organization receives API documentation and rules of conduct from SIoTIP. When the negotiations are conducted, the organization is provided dashboard accounts for the individual developers employed by the organization and the organization is registered as an Application Provider in the Online service.

**UC4: Add application** The user logs into the Application Provider dashboard and uploads the new application. The Online Service initiates a number of automated tests and shows the progress on the user's Application Provider dashboard. If the application passes all tests, the user receives a notification and the application is made available to Customer Organizations for subscription. If the application does not pass all tests, a SIoTIP System Administrator performs a secondary review and decides whether to accept or reject the application. In the latter case, the user is notified of the reason of rejection.

**UC5: Update existing application** The user logs into the Application Provider dashboard and uploads the updated application. The user also indicates whether to automatically update existing instances of the application or to require customer organizations to subscribe to the application. The Online Service initiates a number of automated tests and shows the progress on the user's application provider dashboard. If the application passes all tests, the user receives a notification and the application is made available to customer organizations for subscription. If the application does not pass all tests, a SIoTIP system administrator performs a secondary review and decides whether to accept or reject the application. In the latter case, the user is notified of the reason of rejection.



**UC6: Register as new infrastructure owner** The infrastructure owner contacts SIoTIP and negotiations are started. An Infrastructure Owner dashboard account is set up for the new user. The infrastructure owner provides the names of the currently renting companies, and SIoTIP contacts them for registration (See UC: Register new customer organization).

**UC7: Register as new customer organization** The organization contacts or is contacted by SIoTIP and provides its billing and contact information. The organization is then registered as a Customer Organization in the Online Service.

**UC8: Subscribe to application** The user logs into his Customer Organization dashboard, subscribes to the application and provides the needed information. The user is informed by the Online Service that the application will be activated once the required peripherals are installed. The Online Service checks whether or not the customer organization has access to all the peripherals needed for the application. If not, the infrastructure owner is automatically notified of the subscription and the needed peripherals (see UC10: process peripheral request). Once all required hardware is installed, the application is activated and the user is notified.

**UC9: Unsubscribe from application** The user logs into his Customer organization dashboard and unsubscribes from the application. TODO notification nodig? The user is informed by the Online Service that the application will be deactivated.

**UC10: Process peripheral request** The Infrastructure Owner is notified of a new request for peripherals. The Online Service automatically adds sufficient gateways needed to support these peripherals to the request, if any. The infrastructure owner approves or rejects the purchase of the hardware and a notification of the decision is sent to the user which requested the peripherals. If the infrastructure owner approved the request, the hardware is ordered from SIoTIP.

**UC11: Install new hardware** The Infrastructure Owner receives the hardware from SIoTIP and installs it. The infrastructure owner configures any new gateways to connect to the (WiFi?) local network. Once online, the gateways immediately connect to the Online Service to register themselves. The infrastructure owner then logs in to the infrastructure owner dashboard and provides the necessary topology information. Lastly, he allocates the new peripherals to the Customer Organizations in his building (See UC: Allocato peripherals).

**UC12: Resolve hardware failure** The Online Service detects that a hardware component is no longer sending data and sends a notification to the Infrastructure Owner. The Online Service also notifies all applications currently using the failing hardware component so that the applications can search for equivalent sensors in the topology.

**UC13: Configure end-users** The user logs in to the Customer Organization dashboard and assigns users to an application. These changes are saved in the Online Service. Nodig?

**UC14: Transmit data** TODO nodig als use case? The sensor sends data to the corresponding gateway. The gateway stores the data and makes it available to all applications running on that gateway.

**UC15: Get invoice overview** The user logs into his Customer Organization dashboard and selects the option to show his Customer Organization's invoice overview. The Online Service then shows an overview of all the Customer Organization's invoices.

**UC16: Edit topology** The user logs into his Infrastructure Owner dashboard and selects the option to edit the his building's (TODO meerdere buildings?) topology. When the user is done editing the topology, the updated topology will be saved in the Online Service.

**UC17: Allocate peripherals** The user logs into his Infrastructure Owner dashboard and selects the option to allocate peripherals to Customer Organizations. He then assigns access rights to the Customer Organizations in his building to use the new peripherals. The Online Service automatically activates any application of that Customer Organization which needed the newly allocated peripherals in order to run. Conversely, if a peripheral gets withheld from a Customer Organization, the Online Service deactivates any application of that Customer Organization which needed the withheld peripheral to function properly.

**UC18: Develop application** TODO nodig? Niet echt interactie tussen context en systeem?

**UC19: Resolve platform failure** An event causes the platform to misbehave. The Online Service detects this event and sends a notification to the System Administrator(s).

**UC20: Resolve application failure** An event causes an application to misbehave. The Online Service detects this event and sends a notification to the System Administrator(s). TODO ook notifiati  naar app provider?

**UC21: Configure method of notification delivery** The user logs into his Customer Organization dashboard and configures the method of notification delivery. The Online Service will save the changes made.

**UC22: Request notification overview** The user logs into his dashboard and requests an overview of previously received notifications and alarms. The Online Service provides this overview.

**UC23: Synchronize Online Service and gateway** The gateway sends all data acquired between this moment and the last synchronization moment to the Online Service.

**UC24: Get application overview** The user logs into his Application Provider dashboard and request an overview their applications. The Online Service provides this overview.

## 2.2 Detailed use cases

### 2.2.1 UC8: Subscribe to application

- **Name:** Subscribe to application
- **Primary actor:** Customer Organization
- **Secondary actor(s):**
  - Infrastructure Owner: Needs to approve/reject the subscription if extra hardware is needed
- **Interested parties:**
  - *System*: wants to authenticate its users.
- **Preconditions:**
  - The User is registered into the system and has credentials to prove his identity.
  - Second precondition.
- **Postconditions:**
  - First postcondition.
  - Second postcondition.
- **Main scenario:**

1. The member of the Customer Organization indicates he wants to subscribe to an application.
2. The Online Service presents the user with a means to search for and subscribe to applications.
3. The user subscribes to an application.
4. The Online Service saves the subscription.
5. The Online Service notifies the user that the application will be activated once the required peripherals are installed.
6. The Online Service notifies the Infrastructure Owner that new peripherals are needed (**Include:** UC10: Process peripheral request).
7. The Infrastructure Owner installs the hardware (**Include:** UC11: Install new hardware).
8. The Infrastructure Owner allocates the new peripherals to the Customer Organization (**Include:** UC17: Allocate peripherals )
9. The Online Service activates the application
10. The Online Service notifies the user that the application is activated.

- **Alternative scenarios:**

- 3b. Alternative at step 3

- **Remarks:**

- First remark

**UC1: Name** Short summary of this use case scenario

### 2.2.2 *UC4*: Add application

- **Name:** Add application

- **Primary actor:** Application Provider

- **Secondary actor(s):** System Administrator

- **Interested parties:**

- *Online Service:* wants to authenticate its users and keep track of applications.

- **Preconditions:**

- The Application Provider is registered on the Online Service (UC3).
- The Application Provider is logged into the Online Service (UC1).

- **Postconditions:**

- The Application Provider is notified about whether or not the application got accepted by the Online Service

- **Main scenario:**

1. The Application Provider indicates he wants to add an application
2. The Online Service presents the Application Provider with a means to add his application to the Online Service.
3. The Application Provider adds the application.
4. The Online Service initiates a number of automated tests on the application.
5. The application passes all tests.

6. The Online Service notifies the Application Provider that the application got accepted.
7. ...

- **Alternative scenarios:**

5b. The application did not pass all automated tests.

5b1. The Online Service instructs a System Administrator to perform a secondary review.

5b2a. The System administrator approves the application, resume at step 6.

5b2b. The System Administrator rejects the application. The Application provider is notified of this decision.

- **Remarks:**

- First remark

### 2.2.3 UC11: Install new hardware

- **Name:** Install new hardware

- **Primary actor:** Infrastructure Owner

- **Secondary actor(s):** secondary actor(s)

- **Interested parties:**

- *Online Service:* wants to authenticate its users and keep track of all hardware.
- *Customer Organization:* TODO?

- **Preconditions:**

- The Infrastructure Owner is registered on the Online Service (UC6).

- **Postconditions:**

- The new hardware is installed.
- The new hardware is added to the topology of the building it is installed in.

- **Main scenario:**

1. The Infrastructure Owner receives the hardware from SIoTIP.
2. The Infrastructure Owner places the new hardware where it needs to be.
3. The Infrastructure Owner configures any new gateways to connect to the local network.
4. The new gateways automatically connect to the Online Service and register themselves.
5. The Infrastructure Owner logs into the Online Service. (**Include:** UC1: log in)
6. The Infrastructure Owner edits the topology of his infrastructure. (**Include:** UC16: Edit topology)
7. The Infrastructure Owner allocates the new peripherals to the Customer Organizations who are allowed to use them. (**UC17: Allocate peripherals**)

- **Alternative scenarios:**

3b. Alternative at step 3

- **Remarks:**

- First remark

## 3. Non-functional requirements

In this section, we model the non-functional requirements for the system in the form of *quality attribute scenarios*. We provide for each type (availability, performance and modifiability) one requirement.

### 3.1 Availability

#### 3.1.1 *Av1*: Name of the quality attribute scenario

Shortly describe the context of the scenario.

- **Source:** source
- **Stimulus:**
  - Description of a first stimulus.
  - Description of a second stimulus.
- **Artifact:** the stimulated artifact
- **Environment:** the condition under which the stimulus occurs
- **Response:**
  - Describe how the system should respond to the stimulus.
- **Response measure:**
  - Describe how the satisfaction of a response is measured.

**UC2: Name** Short summary of this use case scenario

### 3.2 Performance

#### 3.2.1 *P1*: Name of the quality attribute scenario

Shortly describe the context of the scenario.

- **Source:** source
- **Stimulus:**
  - Description of a first stimulus.
  - Description of a second stimulus.
- **Artifact:** the stimulated artifact
- **Environment:** the condition under which the stimulus occurs
- **Response:**
  - Describe how the system should respond to the stimulus.
- **Response measure:**
  - Describe how the satisfaction of a response is measured.

## 3.3 Modifiability

### 3.3.1 *M1*: Name of the quality attribute scenario

Shortly describe the context of the scenario.

- **Source:** source
- **Stimulus:**
  - Description of a first stimulus.
  - Description of a second stimulus.
- **Artifact:** the stimulated artifact
- **Environment:** the condition under which the stimulus occurs
- **Response:**
  - Describe how the system should respond to the stimulus.
- **Response measure:**
  - Describe how the satisfaction of a response is measured.

## 3.4 Usability

### 3.4.1 *U1*: Name of the quality attribute scenario

Shortly describe the context of the scenario.

- **Source:** source
- **Stimulus:**
  - Description of a first stimulus.
  - Description of a second stimulus.
- **Artifact:** the stimulated artifact
- **Environment:** the condition under which the stimulus occurs
- **Response:**
  - Describe how the system should respond to the stimulus.
- **Response measure:**
  - Describe how the satisfaction of a response is measured.