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

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.

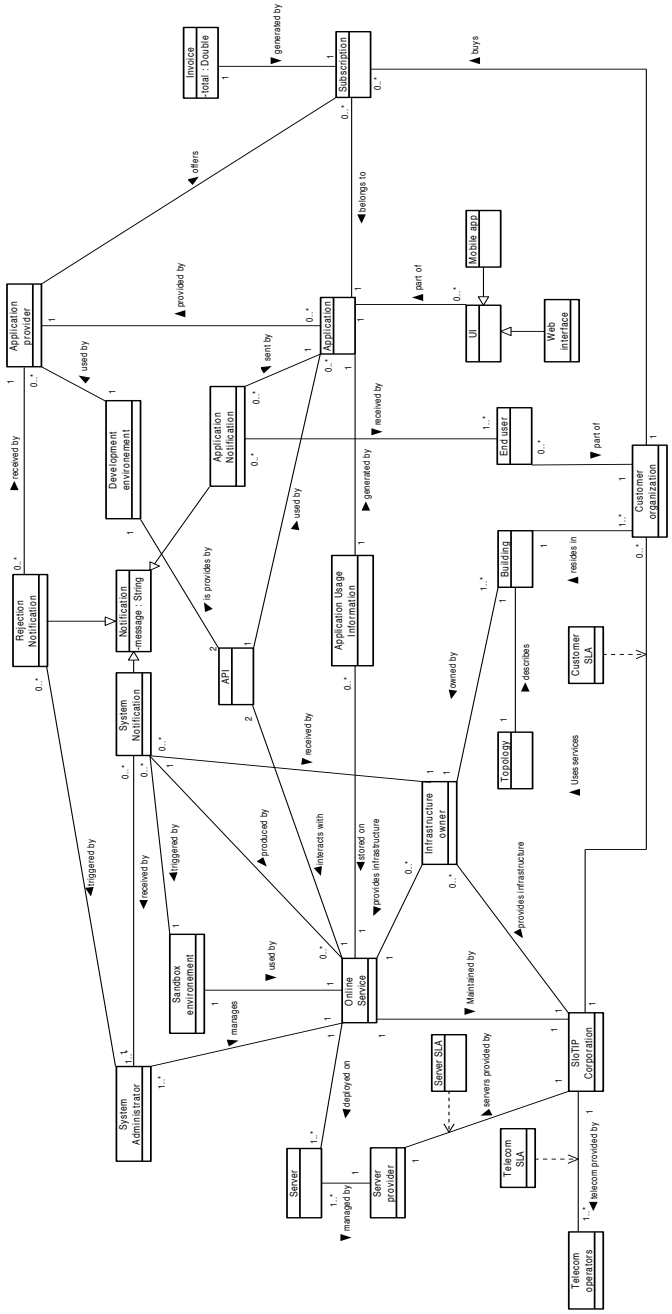

Figure 1.1: Domain model: General

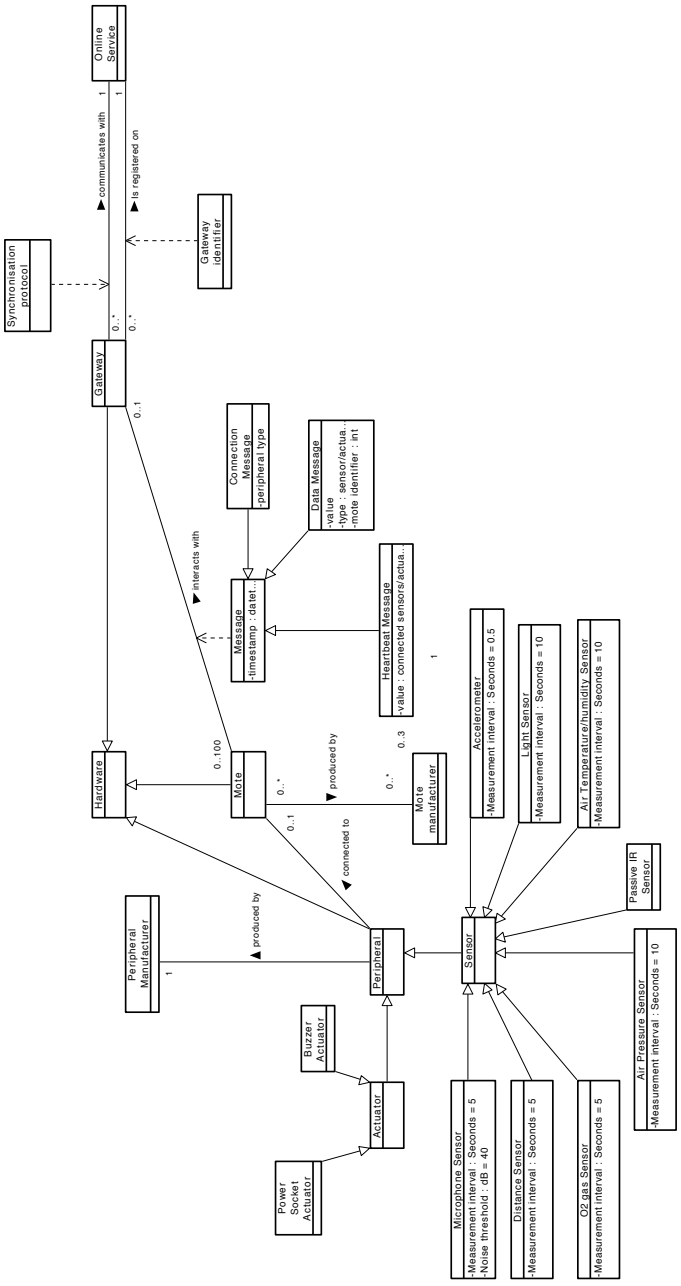

Figure 1.2: Domain model: Hardware and information flow

1.2 Domain constraints

In this section we provide additional domain constraints.

- A customer organisation can only subscribe to an application that is already registered with the system.
- A end-user only receives notifications from applications that the customer organisation 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 organisation.

- 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 the measures the air temperature in degrees Celsius and humidity in percentage.
- **Application Notification**: A type of notification send 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 organisations 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 keeps 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 SIoTIP corporation and the customer organisation.
- **Customer organisation**: The stakeholder in our system which subscribes to applications which optimise 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 environment**: The environment 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 buildings.
- **Invoice**: An invoice is associated with a subscription between a customer organisation and an application provider. It specifies the fees necessary to be paid by the customer organisation.
- **Light Sensor**: Type of sensor that measures the amount of light in LUX.
- **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_2 gas sensor**: Sensor that can measure the amount of O_2 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.
- **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 successful platform for deploying IoT devices. They have agreements with different third parties for providing an optimal platform with respect to availability, performance ..
- **Sandbox environment**: 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 administrator 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 specifies 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 communicate with the online service.

2. Functional requirements

Use case model

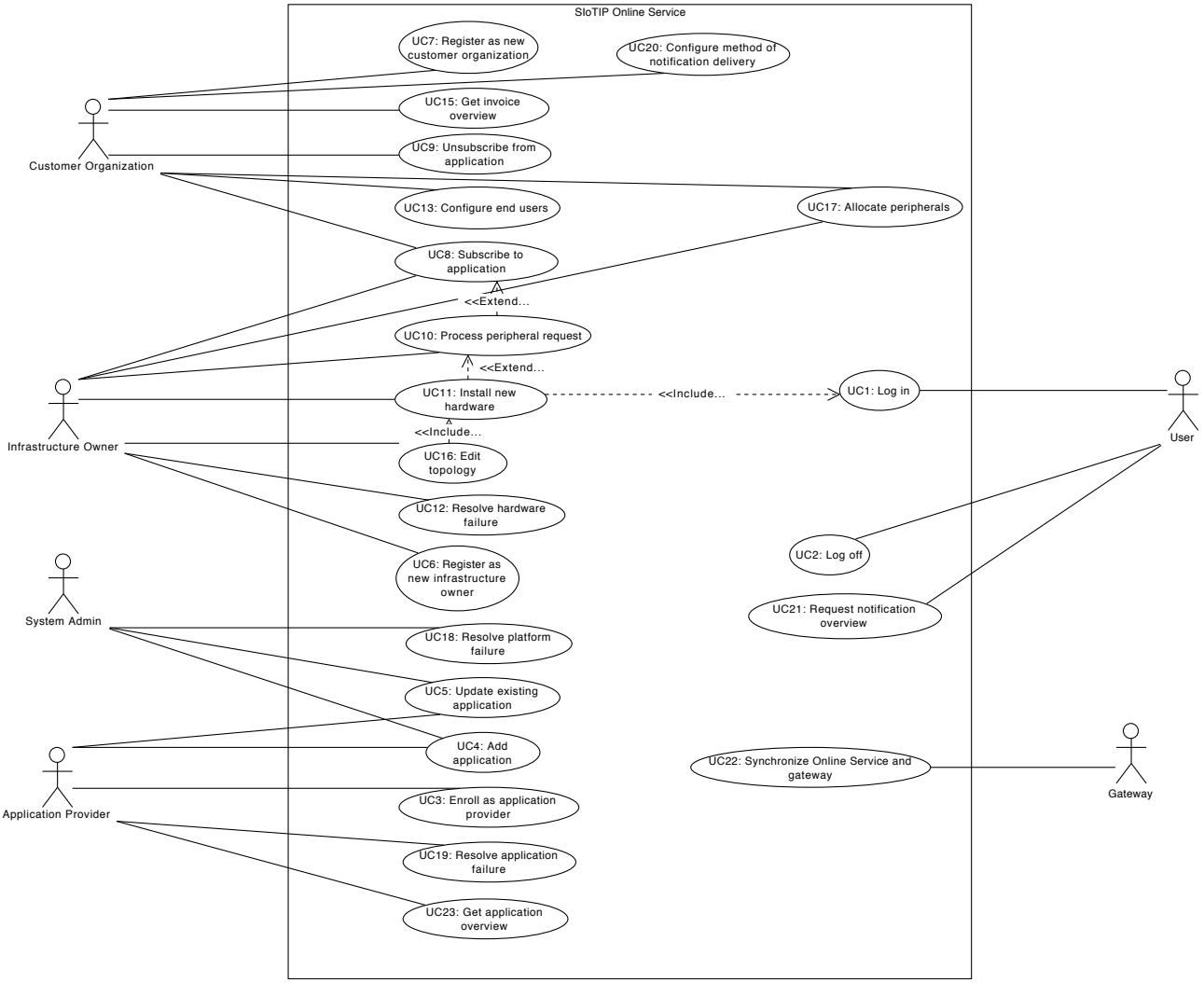

Figure 2.1: Use case diagram
(main functional requirements)

2.1 Use case overview

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

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

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

UC4: Add application The user logs into the his 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 organisations for subscription. If the application does not pass al 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 his 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 organisations 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 organisations 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 organisation).

UC7: Register as new customer organisation The organisation contacts or is contacted by SIoTIP and provides its billing and contact information. The organisation is then registered as a customer organisation in the online service.

UC8: Subscribe to application The user logs into his customer organisation 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 organisation has access to all the peripherals needed for the application. If not, the infrastructure owner is automatically notified of the subscription and the peripherals and gateways needed for that application (see UC10: process hardware 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 organisation dashboard and unsubscribes from the application. The user is informed by the online service that the application will be deactivated.

UC10: Process hardware request The infrastructure owner is notified of a new request for hardware. The infrastructure owner approves or rejects the purchase of the hardware and a notification of the decision is sent to the customer organisation which requested the peripherals. If the infrastructure owner approves 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 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 organisations in his building (See UC: Allocate 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 his customer organisation dashboard and assigns users to an application. These changes are saved in the online service.

UC14: Transmit data The sensor sends data to the corresponding gateway. The gateway stores the data and makes it available to all applications interacting with the sensor.

UC15: Get invoice overview The user logs into his customer organisation dashboard and selects the option to show his customer organisation's invoice overview. The online service then shows an overview of all the customer organisation's invoices.

UC16: Edit topology The user logs into his infrastructure owner dashboard and selects the option to edit the topology of his buildings. 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 organisations. He then assigns access rights to the customer organisations in his building to use the new peripherals. The online service automatically activates any application of that customer organisation which needed the newly allocated peripherals in order to run. Conversely, if a peripheral gets withheld from a customer organisation, the online service deactivates any application of that Customer organisation which needed the witheld peripheral to function properly.

UC18: 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).

UC19: 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) and a log is sent to the relevant application provider.

UC20: Configure method of notification delivery The user logs into his customer organisation dashboard and configures the method of notification delivery. The online service will save the changes made.

UC21: 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.

UC22: Synchronize Online Service and gateway The gateway sends all data acquired between this moment and the last synchronisation moment to the online service.

UC23: 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 *UC4*: Add application

- **Name:** Add application
- **Primary actor:** Application Provider

- **Secondary actor(s)**: System Administrator, Online Service
- **Interested parties:**
  - *Online Service:* wants to authenticate its users and keep track of applications.
  - *Application Provider:* wants his applications to be available on the SIoTIP platform.
- **Preconditions:**
  - The Application Provider is registered on the Online Service (cf. UC3).
  - The Application Provider is logged into the Online Service (cf. 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.

2.2.2 *UC8*: Subscribe to application

- **Name:** Subscribe to application
- **Primary actor:** Customer organisation
- **Secondary actor(s)**: Infrastructure owner, Online Service
- **Interested parties:**
  - *Online Service:* wants to authenticate its users and keep track of subscriptions.
  - *Customer organisation:* wants to subscribe to applications
  - *Infrastructure Owner:* Has to decide whether hardware will be bought for the applications or not.
- **Preconditions:**
  - The Customer organisation is registered on the Online Service (UC7).
  - The member of the Customer organisation is logged into hit Customer organisation dashboard (cf. UC1).

- **Postconditions:**
  - The Customer organisation is subscribed to the application.
- **Main scenario:**
  1. The member of the Customer organisation 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 organisation (**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

2.2.3 *UC10*: Process hardware request

- **Name:** Process peripheral request
- **Primary actor:** Infrastructure Owner
- **Secondary actor(s)**: Online Service
- **Interested parties:**
  - *Customer organisations:* Could use the requested hardware if it is allocated to them.
  - *Infrastructure Owner:* Has to decide whether or not the hardware will be bought.
- **Preconditions:**
  - The Infrastructure Owner received a request for new hardware.
  - The Infrastructure Owner is registered on the Online Service (cf. UC6).
  - The Infrastructure Owner is logged into his Infrastructure Owner dashboard. (cf. UC1)
- **Postconditions:**
  - The requester of the hardware is notified about whether or not the Infrastructure Owner approved the request.
- **Main scenario:**
  1. The Infrastructure Owner approves the request.
  2. The Online Service orders the hardware from SIoTIP.
  3. The Online Service notifies the requester of the hardware about the decision.
- **Alternative scenarios:**

  1b1. The Infrastructure Owner rejects the request, resume at step 3.

2.2.4 *UC11*: Install new hardware

- **Name:** Install new hardware
- **Primary actor:** Infrastructure Owner
- **Secondary actor(s)**:
- **Interested parties:**
  - *Online Service:* wants to authenticate its users and keep track of all hardware.
  - *Customer organisations:* can use the newly installed hardware if it is allocated to them.
  - *Infrastructure Owner:* has to maintain topology of the hardware and has to place and configure it.
- **Preconditions:**
  - The Infrastructure Owner is registered on the Online Service (cf. 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 organisations who are allowed to use them. (**UC17: Allocate peripherals**)

2.2.5 *UC17*: Allocate peripherals

- **Name:** Allocate peripherals
- **Primary actor:** Infrastructure Owner
- **Secondary actor(s)**: Online Service
- **Interested parties:**
  - *Online Service:* wants to authenticate its users and keep track of all allocations.
  - *Customer organisations:* can use the peripherals if they are allocated to them.
- **Preconditions:**
  - The Infrastructure Owner is registered on the Online Service (cf. UC6).
  - The Infrastructure Owner is logged into his Infrastructure Owner dashboard. (cf. UC1)
- **Postconditions:**
  -
- **Main scenario:**

1. The Infrastructure Owner indicates he wants to allocate or withhold peripherals to/from Customer organisations.
2. The Online Service presents the user with a means to do this.
3. The Infrastructure Owner allocates/withholds peripherals to/from Customer organisations.
4. The Online Service activates any application in a Customer organisation which needed the peripherals allocated to that Customer organisation.
5. The Online Service deactivates any application in a Customer organisation which needed the peripherals withheld from that Customer organisation.

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*: Communication channel between the gateway and the Online Service

Due to a failure of the intermediate telecom infrastructure or an error in/outage of the Online Service important functionalities of the Online Service are compromised: applications can not connect to the connected sensors/actuators to push modifications or get data.

- **Source:** external or internal (e.g. error in the Online Service which results in a loss of connection between gateway and Online Service).
- **Stimulus:**
  - the external communication channel between the gateway and the online service is failing
  - or, the internal communication service of the Online Service is failing and thus connection between applications and gateways is not possible
- **Artifact:** external communication channel, gateway, internal communication system
- **Environment:** normal execution
- **Response:**
  - Prevention:
    * the SIoTIP corporation has negotiated a SLA with the telecom provider which stipulates 97% availability over our communication channel
    * the SIoTIP corporation has negotiated a SLA with the server provider which provides 99% uptime of the servers used to run our Online Service
    * the gateways itself run minimal applications to ensure that when the connection to the Online Service is lost the actuators can still be used (e.g. temperature control)
  - Detection:
    * the gateways are able to detect that the Online Service is offline by receiving time-outs when sending data to the Online Service (e.g. when the external communication channel fails)
    * the Online Service detects the lack of updates from gateways and keeps track of how long no updates have been received
  - Resolution:
    In every scenario the system administrator is contacted.
    * if the external communication channel is at fault, the system administrator must contact the telecom provider to have this resolved
    * if the internal communication fails the system administrator resolves the technical issue and communication is automatically restored
- **Response measure:**
  - Detection of a failing external or internal communication link depends on the synchronisation protocol but happens within one minute.

- In case of an external communication failure, the SLA with the telecom provider stipulates that the problem is addressed within 15 minutes and resolved within one hour.
- In case of an internal communication failure, the system administrator has to resolve the issue within 20 minutes.

3.1.2 *Av2*: Availability of microPnP sensors

Due to the failure of a sensor, the correct behaviour of a critical application is compromised.

- **Source:** external
- **Stimulus:**
  - A sensor that is in use by a critical application (e.g. a fire alarm application) stopped working due to unforeseen circumstances (e.g. faulty battery in mote).
- **Artifact:** the application
- **Environment:** normal execution
- **Response:**
  - Prevention:
    - the system administrator has to check the battery level on all motes periodically.
    - the mote manufacturer guarantees a battery life of at least 3 years.
    - the sensor manufacturer guarantees a lifetime of at least 5 years.
  - Detection:
    - The online service detects the lack of incoming data.
  - Resolution:
    In every scenario the system administrator is contacted and the broken sensor is replaced.
    - If a alternative sensor can be found to replace the failing sensor, this sensor is used as a substitute.
    - if no alternative sensor can be found, the application and the infrastructure owner are notified.
- **Response measure:**
  - the detection of a sensor failure happens within 30 seconds.
  - in case of the presence of an alternative sensor the replacement in the application should be within 30 seconds.

3.2 Performance

3.2.1 *P1*: Transferring data from sensor to Online Service

Transferring periodic data from the sensors to the applications should happen in a timely fashion, even if a lot of sensors send at the same time.

- **Source:** Sensor data
- **Stimulus:**
  - synchronisation of data with the online service
- **Artifact:** applications using the data
- **Environment:** peak load

- **Response:**
  - When under heavy load, the system should increase the throughput as specified in the synchronisation protocol.

  The data packages should be as minimal as possible, without excluding necessary details.
- **Response measure:**
  - All packages should arrive within 5ms.

3.2.2 *P2*: Efficient testing of new applications

Transferring periodic data from the sensors to the applications should happen in a timely fashion, even if a lot of sensors send at the same time.

- **Source:** Sensor data
- **Stimulus:**
  - synchronisation of data with the online service
- **Artifact:** applications using the data
- **Environment:** peak load
- **Response:**
  - When under heavy load, the system should increase the throughput as specified in the synchronisation protocol.

  The data packages should be as minimal as possible, without excluding necessary details.
- **Response measure:**
  - All packages should arrive within 5ms.

3.3 Modifiability

3.3.1 *M1*: Adding a new type of sensor to the topology

DistriNet has developed a new type of sensors ('presence sensors') which are capable of detecting if a specific person is present in the neighbourhood of the sensor. This is based on radio signals emitted by the phones of the subjects. Of course, our Online Service and its applications must be able to process and use this data.

- **Source:** DistriNet research
- **Stimulus:** expanding the application 'horizon' and expanding the accuracy and reliability of current applications.
- **Artifact:** the stimulated artifact
- **Environment:** normal execution
- **Response:**
  - The infrastructure owner has to be notified so he is up to date with the latest technology
  - The application logic has to be expanded to be able to cope with this new information
  - A new sensor type has to be added to the system
  - Our library and API's need to be updated to be able to cope with the new sensor type
- **Response measure:**
  - This modification must be done within 30 minutes of introduction of the new sensor.

3.3.2 *M2*: Implementing a new and optimised sandboxing environment

Developers at SIoTIP have developed a new sandbox environment in which applications are tested before being accepted or rejected. This should not influence the currently deployed applications and should not prevent future applications from being tested.

- **Source:** SIoTIP
- **Stimulus:** optimising the application submission workflow and reliability, and by consequence the efficacy of our system
- **Artifact:** the sandbox environment
- **Environment:** normal execution
- **Response:**
  - The infrastructure owner has to be notified so he is up to date with the latest sandbox environment.
  - The current version of the sandbox environment has to be updated to the new version.
  - The update to this new sandbox environment does not affect the rest of the system.
- **Response measure:**
  - This update must be done within 1 day.

3.4 Usability

3.4.1 *U1*: Subscribing to an application

Subscribing to a new application should be possible with ease. If a subscription requires new hardware, this hardware is ordered automatically. Whenever the new hardware arrives and is connected, the application should automatically become active (without configuration).

- **Source:** customer organisation
- **Stimulus:**:
  - Customer organisation wants to use the system efficiently and with ease.
  - The customer organisation does not need to know underlying processes.
- **Artifact:**Online Service, application, the customer organisation dashboard (and indirectly with the hardware store), infrastructure owner, application provider
- **Environment:** normal execution
- **Response:**
  - The subscription is registered in the Online Service and key users are added in the application configuration.
  - It is checked with the infrastructure owner if the necessary sensors are available.
    - If the necessary hardware is available, the application is activated automatically and the end users of the customer organisation can start interacting with the application.
    - If the necessary hardware is not available, the system puts the necessary hardware in a 'shopping basket' and notifies the infrastructure owner associated with the customer organisation.
    - When the infrastructure owner accepts, the hardware will be delivered in a few days. When installed by a employee of the infrastructure owner, the application will become active automatically.

- **Response measure:**
  - Subscribing and configuring an application should be done in less than 10 minutes.
  - Ordering and installing the necessary hardware should be done in 5 working days.
  - Activating the application (when all hardware is available) should be done in less than 5 seconds.

3.4.2 *U2*: Adding an application

Developing and adding a new application should be easy and should follow a consequent and solid process. The application provider should be able to upload applications without breaking the system. Whenever an application does not meet the necessary requirements, the Online Service will notify the system administrator which can approve manually or can reject, which means the application provider has to modify its application.

- **Source:** application provider
- **Stimulus:**:
  - Application providers want to use the system as intended: the application developer does not need to have knowledge about the underlying process.
  - The application provider should be able to determine the flaws in their applications which should enable the provider to fix these errors.
- **Artifact:**Online Service, application, application developer, sandbox environment
- **Environment:** normal execution
- **Response:**
  - The application is automatically checked in the sandbox for, among others, memory leaks.
    - If the application does not produce any errors or trigger any security measures, the application is added to the Online Service and starts accepting subscriptions.
    - If the sandboxing environment produces an error, a system administrator should be notified for manual inspection.
    - If the system administrator accepts the application nonetheless, the application is published and starts accepting subscriptions. The application provider is notified about the successful application.
    - If the system administrator rejects the application, the application provider is notified with the errors and is able to fix the errors before submitting the application once more.
- **Response measure:**
  - Uploading and checking the application in the sandbox environment should be done within 10 minutes, if no errors occur.
  - A system administrator should respond to an possible rejection notification within 8 hours.
  - After rejection by the system administrator the application provider is notified within 5 seconds.
  - After acceptance by the system administrator, the application is published within 5 seconds.