Wireless bi**O**monitoring stickers and smart bed architecture: to**W**ards untethered patients

D1.2: Functional Specification and System Integration Architecture

Cofinanciado por:

This page is intentionally left blank.

Document Identification

Title	**WoW: Wireless biOmonitoring stickers and smart bed architecture: towards untethered patients**
Reference	CENTRO-01-0247-FEDER-045931
Duration	2020/04/01 – 2023/03/31
Head of Consortium (***Líder do Consórcio***)	Glintt Healthcare Solutions, S.A. (Glintt)
Copromoters (***Copromotores***)	Carnegie Mellon University (CMU), Institute of Systems and Robotics of the University of Coimbra (ISR-UC), University of Coimbra (UC), Coimbra Hospital and Universitary Centre (CHUC)
Coordinator (***Coordenador***)	Filipa Fixe Santos (Glintt)

Work Package / Deliverable

Work Package (***Atividade***)	**WPx. Work Package Title** (***Título da Atividade em PT***)
Deliverable (***Entregável***)	D1.2. Functional Specification and System Integration Architecture Especificação Funcional e Arquitetura Integradora do Sistema
Lead Copromoter	Glintt
Due Date (***mês***)	2020/10/31 (M7)

Change Record

Version	Date	Edited by	Brief Description / Changes Summary
V0.1	2020-11-13	David Santiago (Glintt)	First draft sections 2-6.
V0.2	2020-11-19	David Portugal (ISR-UC)	Added section 1, 7, and placeholders of other sections to fill.
V0.3	2020-11-26	Fernanda Famá (ISR-UC)	Filled all sections with ISR-UC contributions + comments (with David’s assistance).
V0.4	2020-11-27	José Faria (ISR-UC)	Added text/table concerning the FHIR specification on the IMU sensor (section 6.8).
V0.5	2020-11-30	David Santiago (Glintt)	Updated Conceptual Model diagram and descriptions Updated Logical View components description Added Security and Privacy section Added Hardware Requirements Section

			Updated Process View and Use Case Views
V0.6	2020-12-03	Fernanda Famá and José Faria (ISR-UC)	Updated section 4.4 and 5.3 and added the attachment with the FHIR observation for the IMU data.
V0.7	2020-12-06	Mahmoud Tavaloki (ISR-UC)	Updated section 3.4.
V1.0	2020-12-11	José Faria (ISR-UC)	Added section 8.8 (DocumentReference), updated section 8.9 ánd Attachments.

Contents

List of Figures

List of Tables

1. Introduction

The aim of this deliverable is to provide a description of functional specification of the WoW project as well as the architecture integration system.

The architecture integration system in this deliverable aims to understand how the key components in this architecture connect to each other. For this, the document will present an overview about each component and their specific functionalities, and how they integrate in the decoupled and modular system architecture proposed for the WoW project.

In order to bring user context to life and to give detail how patient monitoring may take place, and how the system can add daily value to the end users, a complete and detailed set of use case mapping with the scenarios presented in deliverable 1.1 is presented in this document to achieve some of the key objectives of WP1, namely providing the foundations of the project to drive the technical work, and allowing for an adequate planning and integration directives for components within the project.

The document is structured as follows. In Section 2 a conceptual model is presented. Then, Section 3 a logical view focuses on the functionality that the system provides to end-users. In Section 3 deals with the dynamic aspects of the system, explaining their main interactions. A use case mapping is shown in Section 5. In the next Section is presented the FHIR Resources that will be implemented to ensure an interoperable application. Finally, conclusions are presented in Section 7.

2. **Conceptual Model**

To guide the remainder of this deliverable, and provide a functional specification for the WoW project, we start by proposing a conceptual model that presents an overview of the key components planned for the overall system.

The conceptual model describes basic concepts, and defines the basic attributes of these concepts and the relationships between them.

Figure 1 illustrates the proposed conceptual model for the WoW project.

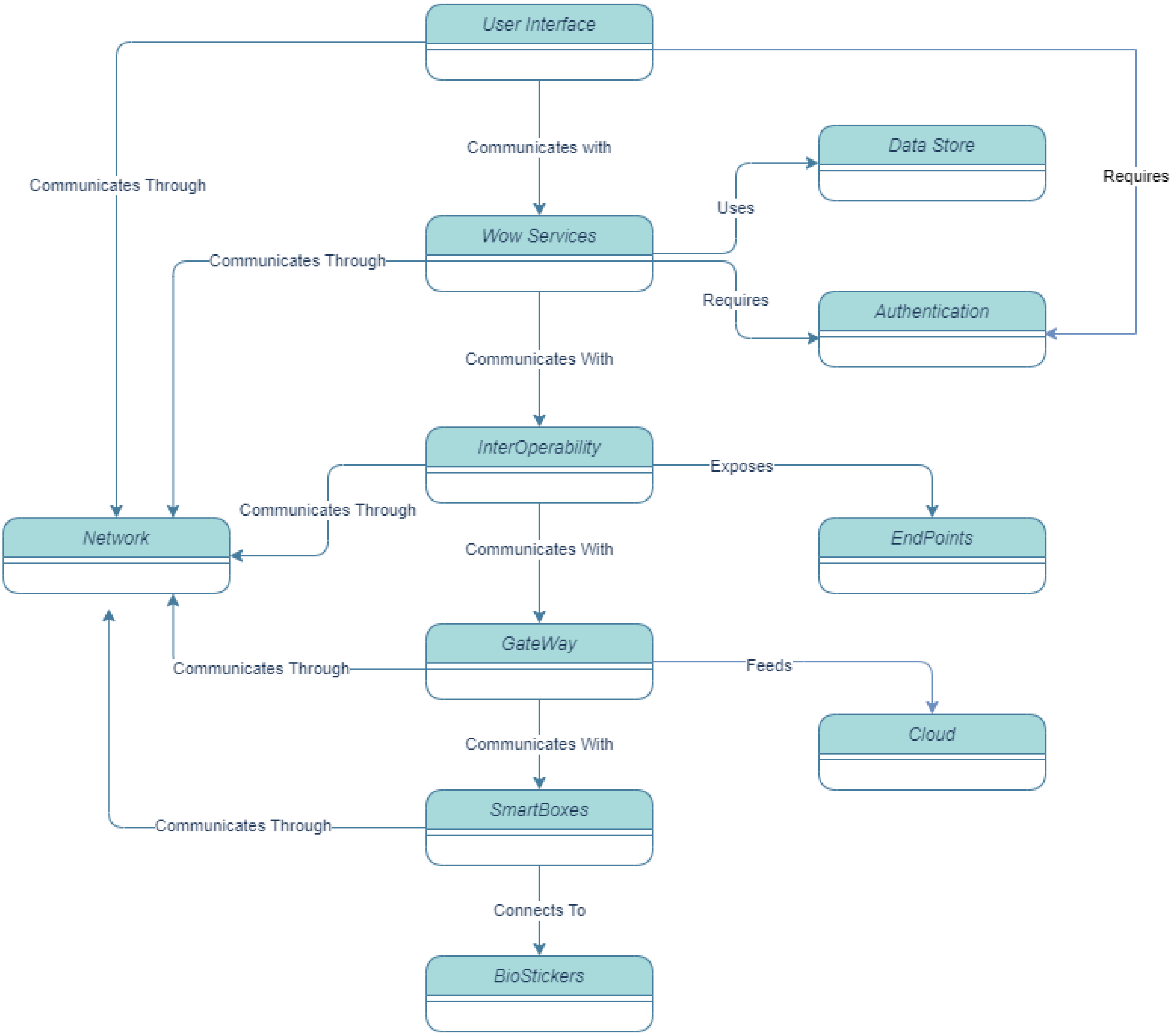

Figure 1. Conceptual Model.

2.1 User Interface

The user interface represents a graphical interface where the user can access the system functionalities.

2.2 Wow Services

Wow Services represent the layer that contains all functional components and business logic. It performs all the complex processing and data extraction from the databases.

2.3 Interoperability

The interoperability is responsible for receiving a request, detect its destination, and transform the requests into a version compatible with the destination.

2.4 Data Store

The Data Store is responsible for the persistence of the system data.

2.5 Authentication

The Authentication is responsible for validating the entity that made a request and check if it has permission to do it.

2.6 EndPoints

An Endpoint implements an interface that allows systems to communicate with each other.

2.7 Biostickers

The biostickers are electronic skin patches that adhere to the human epidermis to collect physiological and behavioural data, encompassing multiple sensors. Therefore, they are responsible for remote sensing and monitoring patients, signal processing and power regulation.

2.8 Smart Boxes

This module represents a smart IoT unit responsible mainly for data acquisition from the biostickers embedded sensors and transmission of measured data through the internet to the remaining system. Besides, we intend to explore methods for transferring energy to the biostickers through near field and far field wireless energy harvesting.

2.9 Gateway

This module is responsible to connect the smart boxes and the Hospital Information System (HIS), managing devices, data and users, for monitoring the registration of smart beds and patients

2.10 Network

The network is the foundation of the proposed architecture, whereas it guarantees the connection between all modules of the system. The network allows the patient and hospital administration access to the user interface, the data to be monitored, managed, stored, transformed and analyzed. Besides, it enforces authentication and security within the communication process.

2.11 Cloud

This module provides a scalable and elastic environment where the heterogeneous data captured can be modelled, analyzed and classified to detect anomalies and discovery of new digital biomarkers.

3. **Logical View**

In this section, we provide a logical view of the system architecture, having in mind the functional specifications described in the previous section. The logical view is concerned with the functionality that the system provides to end-users. It is a technology-agnostic view that describes the dependencies and interactions between the software components.

Figure 2 shows the logical view of the architecture of proposed, followed by a brief technical description of its components.

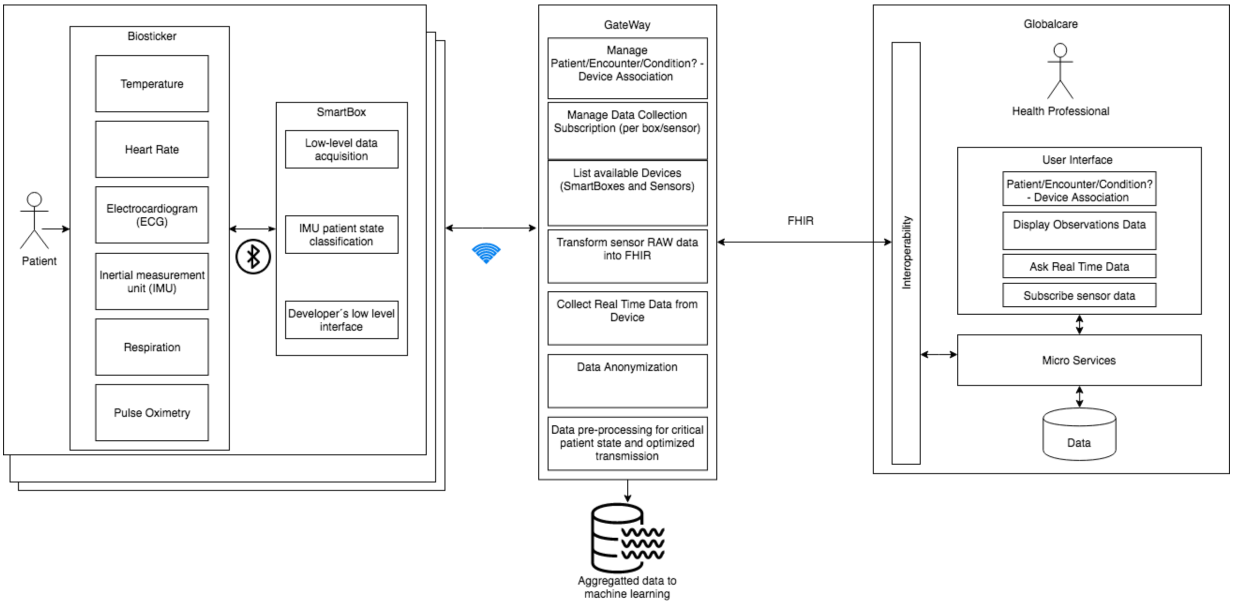

Figure 2. Logical View.

3.1 **Gateway**

This component links the SmartBox to GlobalCare, so it is generally responsible to manage users and data. It is in the gateway, where we maintain a list of Smart Boxes and sensors available and their related devices. In addition, this component collects data from devices via Wi-Fi, pre-processes raw data, transforms data into the FHIR interoperability standard to be sent to the Interoperability layer, and more.

3.2 Cloud

The cloud will collect the data and apply intelligence artificial techniques to analyse the data. Using data analytics, patterns and anomalies should be found to identify unhealthy and dangerous conditions in patients, also physiological and emotional responses. All this process will be controlled exclusively by the WoW project team, therefore the cloud service that will be used is being evaluated, either via a paid subscription cloud service or a local cloud deployed on site at the hospital.

3.3 Smart Box

The smart box consists of a single board computer which interfaces with the low-level data transmitted by the sensors embedded in the biostickers via bluetooth (BLE5) and eventually RFID. In this component, we will have a first data classification approach to detect the patient’s state (e.g. laid down, sitted, etc.) using an Inertial measurement unit (IMU), as well as a developer interface to be used by the technical team while implementing the data acquisition solution.

3.4 Biosticker

The biostickers which are connected to the patients encompass several sensors, that may include body temperature, heart rate, electrocardiogram, respiration rate, body motion, and pulse oximetry. These stickers transmit data through wireless communication to the SmartBoxes data acquisition module. Therefore, these Biostickers are responsible for acquisition and transmission of electrophysiological data to the smart box.

3.5 GlobalCare

The GlobalCare component represents the GlobalCare software. The GlobalCare software handles the administrative functions related to the identification and management of a patient in hospitals and clinics. It allows the registration and monitorization of all the patient's processes and flows, from the first contact with the health entity to its discharge.

The next subsection will present and describe the logical view of the GlobalCare components that will be integrated into this project.

3.5.1 GlobalCare's Logical View

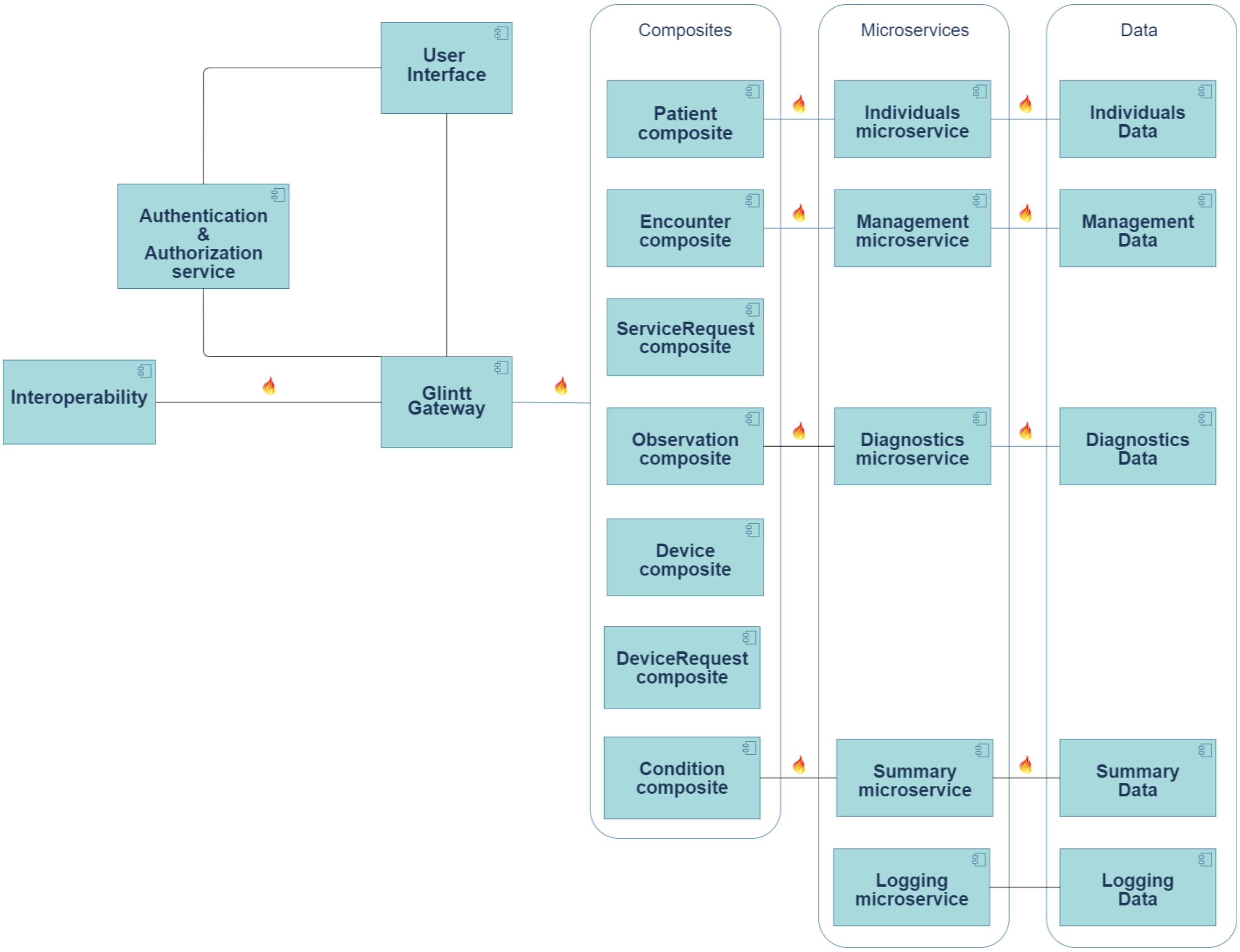

Figure 3 GlobalCare's logical view

3.5.2 **Interoperability**

The Interoperability component has the ability to make health information systems work together within and across organizational boundaries in order to advance the effective delivery of healthcare systems. It allows systems of different natures to communicate with each other.

This component, in this project, will allow the GlobalCare and the Gateway to communicate.

3.5.3 **GlobalCare Gateway**

The Glintt Gateway is the sole entry point into the GlobalCare system. Any party that needs to interact with the GlobalCare will have to do so using the gateway as the entry point. It is responsible for a great part of the system security, for the service discovery, and for the audit and logging.

At the level of security, it requires that the calls contain an authentication token that will be validated by the GlobalCare Gateway itself.

In what concerns to the service discovery, the gateway is responsible for mapping external requests addresses into internal addresses, forwarding the request to the respective internal address.

Regarding Auditing and Logging, the GlobalCare Gateway must be capable of logging the requests flow allowing to easily track the full tree of all the requests.

3.5.4 **Composites Layer**

The main necessity that the composite layer resolves is the composition of data from different microservices. Since the microservices do not communicate with each other, the capability of data aggregation relies solely on this layer. The composite modules are also able to communicate with each other. In this project, the composite layer is responsible for receiving the requests from the Glintt Gateway, call the microservices required to fulfill them, consolidate the data, and return it.

The following subsections will describe the composites of this project.

3.5.4.1 **Patient Composite**

This composite addresses business needs regarding a patient and uses the Individuals microservice and the Patient resource to implement them.

It has the goal to manage information related to the patient by listing or updating its information or creating one if it doesn't exist.

In this project, it will be responsible for listing the patient information so it can be associated with the observations made by the sensors, for machine learning purposes.

3.5.4.2 **Encounter Composite**

This composite addresses business needs regarding an Encounter and uses the Microservice Management and the Encounter resource to implement them.

It has the goal to manage information regarding the Encounters by listing them, updating them, or creating a new one. It is also responsible for discharging the patient from the encounter.

In this project, it will be responsible for listing the patient information so it can be associated with the observations made by the sensors, for machine learning purposes.

3.5.4.3 **ServiceRequest Composite**

This composite addresses business needs regarding the requests for procedures and diagnostics and uses the Gateway and the ServiceRequest resource to implement them.

It has the goal to manage information regarding the ServiceRequests by listing, updating, or creating them.

In this project, it will be responsible for listing the sensors subscriptions, update them, and creating new ones.

3.5.4.4 **Observation Composite**

This composite addresses business needs regarding the output of the sensors and uses the Gateway, the Diagnostics microservice, and the Observation resource to implement them.

It has the goal to manage information regarding the Observations by listing, updating, and creating them. It is also able to receive external observations from external devices, associate them with a GlobalCare patient, and persist them in the system.

In this project, it will be responsible for receiving an observation from the Gateway and persist it and for the listing the existent observations.

3.5.4.5 **Device Composite**

This composite addresses business needs regarding the medical devices and uses the Gateway and the Device resource to implement them.

It has the goal to manage information regarding the Devices by listing, updating, or creating them.

In this project, it will be responsible for listing the existent sensors, biostickers and smartboxes.

3.5.4.6 **DeviceRequest Composite**

This composite addresses business needs regarding the request for the use of a device by a patient and uses the Gateway and the DeviceRequest resource to implement them.

It has the goal to manage information regarding the DeviceRequests by listing, updating, or creating them.

In this project, it will be responsible for requesting the association of smartboxes, biostickers, and sensors to the patients.

3.5.4.7 **Condition Composite**

This composite addresses business needs regarding a patient condition, problem, or diagnosis and uses the Microservice Summary and the Condition resource to implement them.

It has the goal to manage information regarding the Conditions by listing them, updating them, or creating new ones.

In this project, it will be responsible for listing the patient conditions so they can be associated with the observations made by the sensors, for machine learning purposes.

3.5.5 **Microservices Layer**

Microservices represent the layer that contains the business logic. Each of the microservices is capable of handling the full specter of a small part of the domain. This allows the development of highly decoupled microservice modules, which will make the system capable of evolving each of the domains without needing to change the other modules. To promote the goal of decoupling, the microservices do not communicate directly with each other. As referred to in the previous section, the aggregation of the microservices data is made in the composition layer.

The following subsections will describe the microservices of this project.

3.5.5.1 **Individuals Microservice**

The individuals microservice allows to perform CRUD (Create, Read, Update and Delete), search, and get all the items operations regarding Practitioner Role, Practitioners and Patients, according to the correspondent FHIR Resources.

In this project, it is responsible for allowing the search and filtering of patients.

3.5.5.2 **Management Microservice**

The management microservice allows to perform CRUD (Create, Read, Update and Delete) and search operations regarding patient's encounter and episodes of care, according to the correspondent FHIR Resources.

In this project, it is responsible for allowing the search and filtering of encounters.

3.5.5.3 **Diagnostics Microservice**

The diagnostics microservice allows to perform CRUD (Create, Read, Update and Delete) and search operations regarding devices' observations, according to the FHIR Resource.

In this project, it is responsible for allowing the creation and search and filtering of the observations made by the devices.

3.5.5.4 **Summary Microservice**

The summary microservice allows to perform CRUD (Create, Read, Update and Delete) and search operations regarding the patient's conditions, according to the FHIR Resource.

In this project, it is responsible for allowing the search and filtering of the patient's conditions.

3.5.5.5 **Logging Microservice**

The Logging microservice is responsible for aggregating all the logs generated by the various services, persist and fetch them from the logging data layer, enabling a unified view of all the events in the system.

3.5.6 **Data Layer**

The Data Layer is responsible for data storage. It allows seamless access to the databases for data persistence.

This layer is composed of the following components:

3.5.6.1 **Individuals Data**

This component is responsible for the storage of the patients data.

3.5.6.2 **Management Data**

This component is responsible for the storage of the encounters data.

3.5.6.3 **Diagnostics Data**

This component is responsible for the storage of the sensors output data.

3.5.6.4 **Summary Data**

This component is responsible for the storage of the conditions data.

3.5.6.5 **Logging Data**

This component is responsible for the storage of the system logs.

3.5.7 **User Interface**

The User Interface layer is meant to be the point of human-computer interaction and communication. It should deliver a graphical interface so the end-user can explore the system's full potential simply and intuitively.

3.5.8 **Authentication and Authorization**

This layer represents an external module responsible for providing authentication, allowing the system to detect the entity that made the request. It should implement the OAuth protocol.

4. Security and Privacy

Every online system, in order to be trustworthy, needs a security and privacy infrastructure that guarantees the maximum protection of the system and its users. With that in mind, several solutions should be implemented. The following subsections describe the designed solutions at a system and user level.

4.1 Authentication and Authorization

To prevent users from accessing data or functionalities that they are not allowed to, every time a request is made, the system will validate if the entity that triggered it is authenticated and if it has the authorization to do it. This authentication process happens in the interoperability and GlobalCare.

4.2 Data Anonymization

In order to guarantee that the data provided by this system to other systems (ex: Machine Learning mechanisms) respects the principles of user data privacy, a data anonymization component will be implemented, responsible for irreversibly altering classified data before making it externally available.

4.3 Contingency and Mitigation Stategies

Towards the goal of developing a secured system, different security risks and threats have been identified and mitigation strategies are provided below.

4.3.1 Social Engineering of Username and Password

Every system based on username and password authentication is vulnerable to a third party gaining access to the system via social or technical engineering exploits such as phishing.

To prevent this kind of situation, the users are advised to define strong passwords that don't contain any personal references, and not share their password with any entity.

4.3.2 Password Cracking or brute force attacks

By requiring that user passwords have a minimum length and a combination of lower and upper case, letters, numbers, and symbols, the system can hardly be affected by this kind of threat.

Cracking this kind of password would require years of processing.

4.3.3 Data Breaches

A data breach would is an attempt to gain access to the system data storages to collect the sensitive information stored there. By defining a strict list of origin addresses that can directly access the databases, the system makes this kind of attack virtually impossible.

4.3.4 Data Tempering Attacks

A data tampering attack is an attempt to generate an unexpected response on a web service by

manually entering data. The most common is the SQL injection attack, in which SQL database commands are inserted in text fields with the intention of cheating the system and getting information of the contents of the database that would be inaccessible in another way.

By limiting and validating the type, format, and length of the data inputted by the users inside our servers, these attacks can be prevented.

4.3.5 Denial of Service Attack

A Denial of Service (DoS) Attack is an attempt to make an online service unavailable by overloading it throws a massive number of automated requests.

This kind of attack can be easily stopped by blocking the origin of the requests.

However, there is a variation of this attack, where the origins of the requests are distributed - Distributed Denial of Service (DDoS) – making it almost unstoppable. To manage to keep working properly when under this kind of attack, the system should be properly scaled so it can work under heavy loads.

4.4 Security in Data Transmission

4.4.1 Biostickers and SmartBoxes

In our approach the communication between the Biostickers and the Smartboxes will be made using the Bluetooth Low Energy (BLE) 5.0. This technology has a peer-to-peer communications over short distances which supplies a security system at the application layer and the link layer. The BLE security model includes five district features:

- **Pairing**, the process for creating shared secret keys;
- **Bonding**, the act of storing the keys created;
- **Device authentication**, check that the two devices have the same keys;
- **Encryption**, ensuring message confidentiality;

- **Message integrity**, preventing malicious tampering and data corruption in transit;

The BLE connection of the biostickers will use the Security Mode 1 with Security level 3 of the standard BLE Generic Access Protocol (GAP). This method will enforce security by means of authenticated pairing and AES-CCM encryption of the patient data that is beeing transmitted to the smartbox.

4.4.2 Smart Boxes and Gateway

The Wi-Fi connections will be implemented between the Smart Boxes and Gateway. This communication encompasses a WPA2 protocol version that used AES (Advanced Encryptation Standard) to ensure confidentiality, authenticity and integrity in a wireless network. Besides, a private network with SSID hiding, MAC ID filtering and static IP addressing will be implemented to avoid exposing the devices to external interference.

4.4.3 Gateway and GlobalCare

Wi-Fi or an Ethernet connection will be implemented between the Gateway and the GlobalCare, using similar security mechanism between the Smart Boxes and the Gateway.

5. Hardware

This section describes the component’s hardware modules and their requirements for the quality and production environments. The quality environment is set with the goal of testing the software while the production environment is meant to be accessed by the end-users.

5.1 Hardware Modules

This section describes the modules contained in each component

5.1.1 Interoperability

The interoperability is composed of 6 modules: Proxy, Bus, Processor, Webapp, Repository, and Queue.

The **Proxy** is a load balancer that allows the system to have high availability and low downtime impact. This module is only implemented in production.

The **Bus** is the communication integrator that allows the component to forward messages inside itself and allows the traceability of communications.

The **Processor** is responsible for processing events, transform data, communicate to an MDM to convert between systems and it allows a business creation based on events.

The **Webapp** is a user interface that allows the visualization of the metrics related to interoperability. It allows the integration of plugins and implements a REST API.

The **Repository** is a centralized repository that integrates data from several origins and allows to analyze and index several messages contained on events, allowing the system to trace the messages accordingly to the GDPR.

The **Queue** is responsible for the queuing of events.

The next subsections will describe the hardware requirements for each module in quality and production.

5.1.2 GlobalCare

The GlobalCare is composed of 4 modules: Haproxy, Docker Swarm Manager, Docker Swarm Worker, Database.

The **Haproxy** is responsible for the routing, load balancing, and centralization of certificates.

The **Docker Swarm Manager** is responsible for the docker cluster managing and implements system redundancy.

The **Docker Swarm Worker** is the applicational server that has all the GlobalCare containers, including the logs server.

The **Database** is composed of the system database.

5.2 Hardware Requirements

5.2.1 Interoperability

5.2.1.1 Quality

Table 1 Interoperability quality hardware requirements

Module	Scope	Machine	RAM	CPU	Disk1	Disk2
BUS	Base	hsh-q-bus0	2GB	2	15GB	5GB
PROCESSOR	Base	hsh-q-proces-sor0	2GB	2	15GB	5GB
WEBAPP	Base	hsh-q-webapp	2GB	2	15GB	
REPOSITORY	Base	hsh-q-repo	4GB	4	15GB	10GB
QUEUE	Base	hsh-q-queue	2GB	2	15GB	5GB

5.2.1.2 Production

Table 2 Interoperability production hardware requirements

Module	Scope	Machine	RAM	CPU	Disk1	Disk2
PROXY	Redundancy	hsh-p-proxy0	512MB	1	15GB	
	Redundancy	hsh-p-proxy1	512MB	1	15GB	
BUS	Base	hsh-p-bus0	2GB	2	15GB	20GB
	Redundancy	hsh-p-bus1	2GB	2	15GB	20GB
PROCESSOR	Base	hsh-p-proces-sor0	2GB	2	15GB	20GB

	Redundancy	hsh-p-processor1	2GB	2	15GB	20GB
WEBAPP	Base	hsh-p-webapp	2GB	2	15GB	
REPOSITORY	Base	hsh-p-repo	4GB	4	15GB	200GB
QUEUE	Base	hsh-p-queue	2GB	2	15GB	20GB

5.2.2 GlobalCare

5.2.2.1 Quality

Table 3 GlobalCare quality hardware requirements

Module	Scope	Machine	RAM	CPU	Disk1	Disk2
DOCKER SWARM Manager	Base	docmanager-q	8GB	2	10GB	128GB
DOCKER SWARM WORKER	Base	Dockworker-q	8GB	2	10GB	128GB
DATABASE	Base	dockerdb-q	16GB	4	10GB	256GB

5.2.2.2 Production

Table 4 GlobalCare production hardware requirements

Module	Scope	Machine	RAM	CPU	Disk1	Disk2
HAPROXY	Base	dochaproxy	4GB	2	50GB	
DOCKER SWARM MANAGER	Base	docmanager0-p	8GB	2	10GB	256GB
	Redundancy	docmanager1-p	8GB	2	10GB	256GB
	Redundancy	docmanager2-p	8GB	2	10GB	256GB
	Base	docworker0-p	8GB	2	10GB	256GB
	Base	docworker1-p	8GB	2	10GB	256GB

DOCKER SWARM WORKER	Op-tional(Recomended)	docworker-logs-p	8GB	2	10GB	256GB
DATABASE	Base	dockerdb-p	16GB	4	10GB	512GB

5.3 SmartBoxes and Gateway specifications

Table 5 Device specification.

Module	Device	OS	RAM	Disk
Smartboxes	Raspberry Pi 4 Model B	Ubuntu	8 GB	32 GB
Gateway	Intel NUC Core i7 - 8559U	Ubuntu	16 GB	1 TB

6. **Process View**

The process view deals with the dynamic aspects of the system. It explains the system processes and how they communicate. This view focus on the runtime behavior of the system

The following figure describes a general process view of the system and it is followed, in the next subsections, by a description of the main interactions in the system as a set of collaborating architectural components.

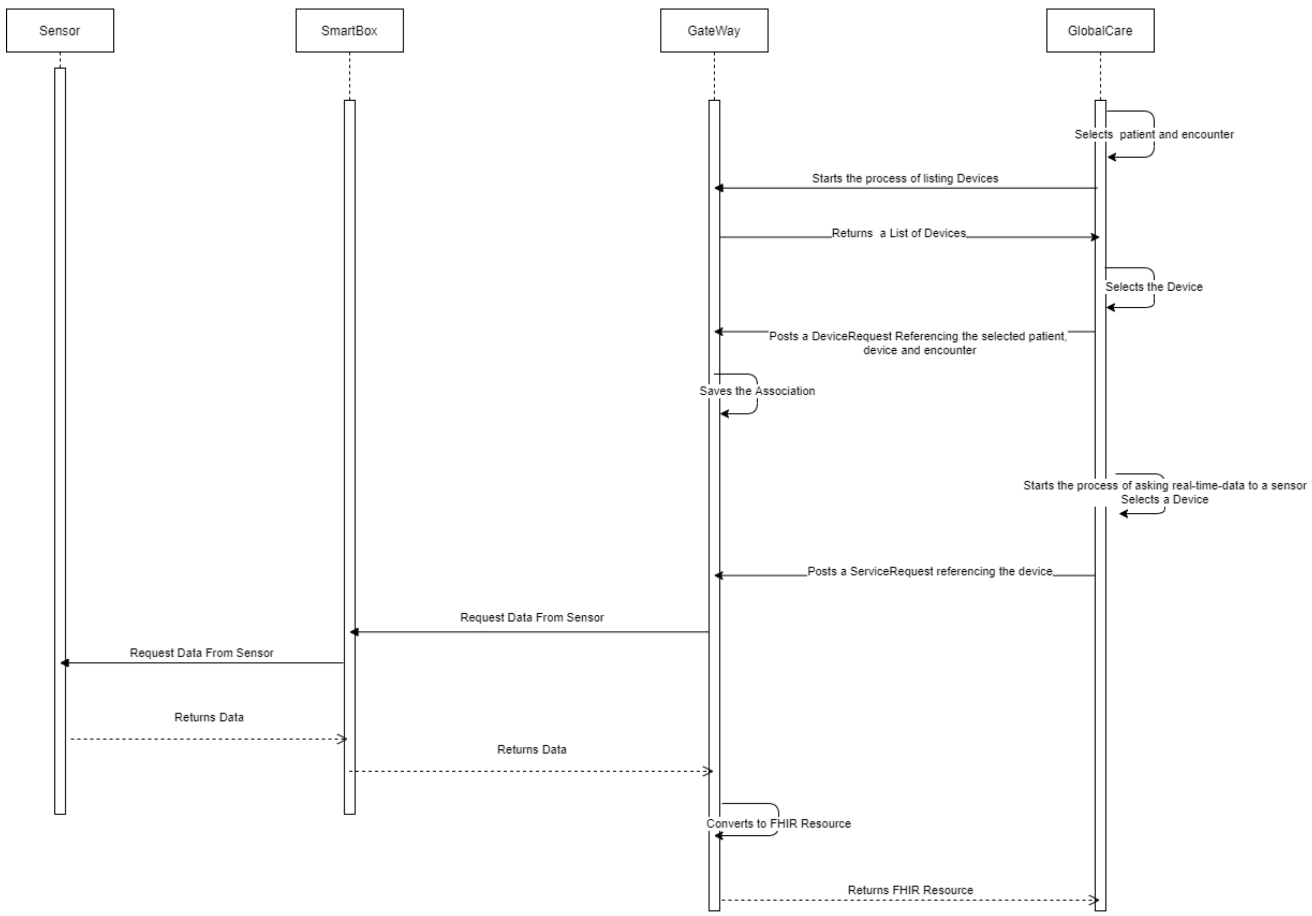

Figure 4 General Diagram

6.1 **List Devices**

In order to list all the devices, the GlobalCare should be able to make a GET request of the resource Devices to the Gateway. GlobalCare should give the user the possibility to implement the following filters to the request:

- Device Hierarchy
- Device Type
- Device assign

Device Hierarchy represents if the device is a Parent Device (SmartBox) or a Child Device (Sensor).
Device Type represents the type of device (thermometer).
Device Assign represents the patient to whom the device is assigned or the value null if not assigned.
The Gateway should return the respective list of FHIR Devices.

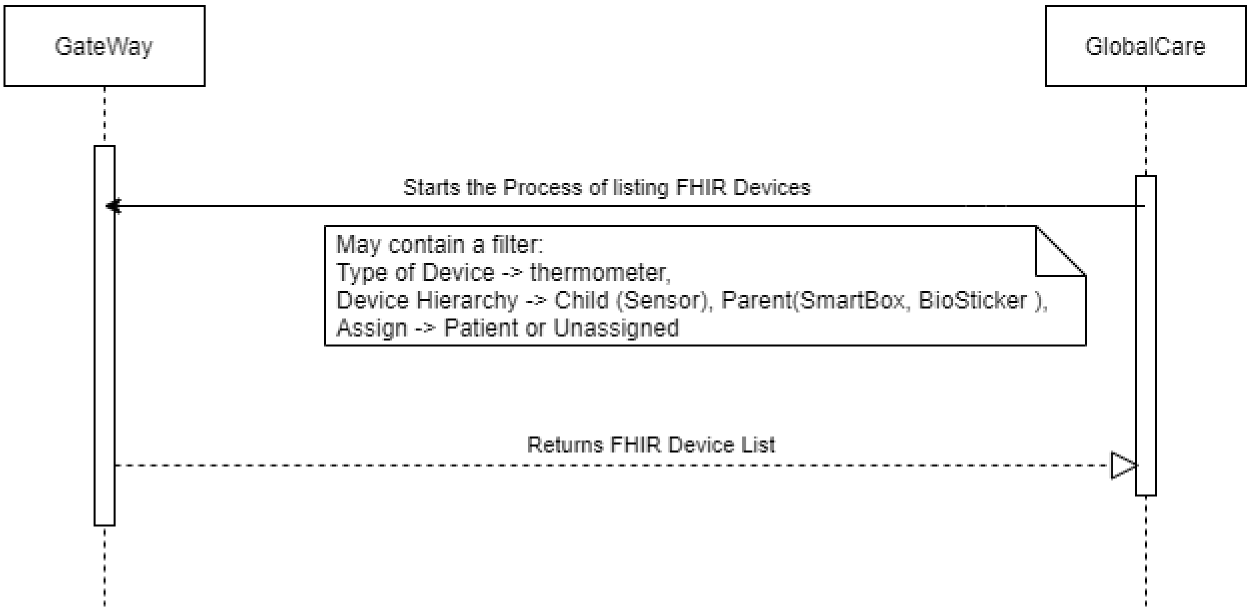

Figure 5. List Devices - Sequence Diagram.

For this process, the Gateway should implement a REST API GET route at /Device.

6.2 Attach Device to Patient

In order to associate (attach) a Device with a Patient and Encounter, the GlobalCare interface should list the unassigned devices so the user can select one of them, the GlobalCare should then list the Patients and their encounters to be associated with the device. Then the GlobalCare should post a DeviceRequest to the Gateway. This DeviceRequest should reference the Device, the Patient, and the Encounter selected by the user. The Gateway should receive the DeviceRequest and set its status to active.

The Gateway is responsible for managing the associations and the GlobalCare acts only as an interface.

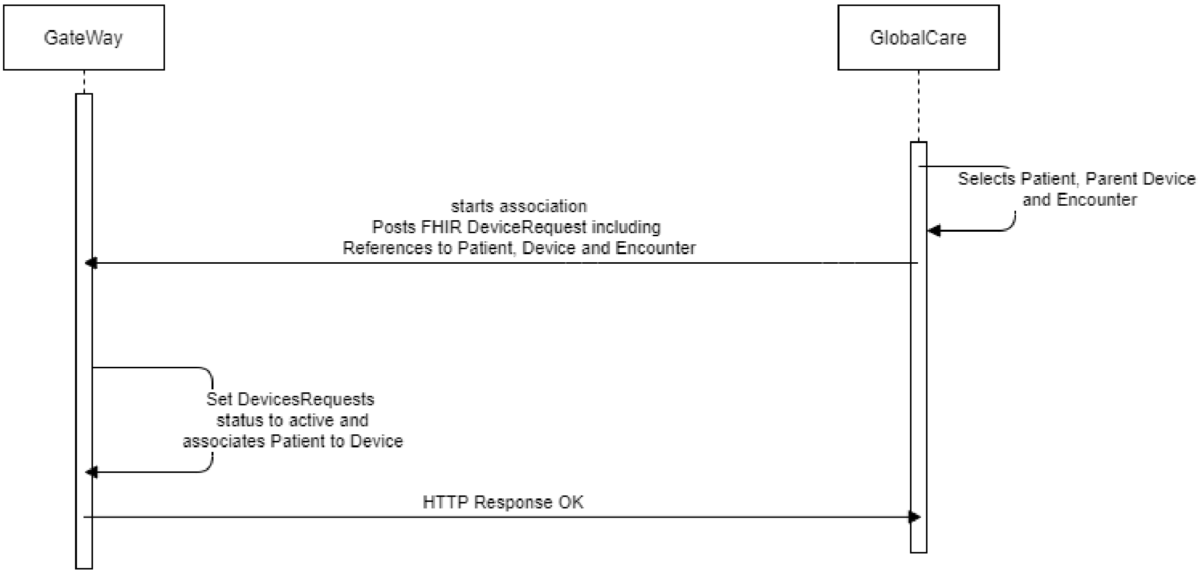

Figure 6. Associate Device to Patient and Encounter - Sequence Diagram.

For this process, the Gateway should implement an REST API POST route at: /DeviceRequest.

6.3 Detach Device

In order to update the association's values, GlobalCare should list the Parent Devices (Biostickers and Smart Boxes). This list should be able to be filtered by device status and associated patient. Then it should make a PatchRequest. This PatchRequest should contain a DeviceRequest with the fields to be changed.

In case this update occurs because of a mistaken association, the DeviceRequest should contain a status=Entered in Error and the GlobalCare should delete any observations received from the device while the association was active. If the goal is to finish the association the GlobalCare should set the DeviceRequest status to Completed.

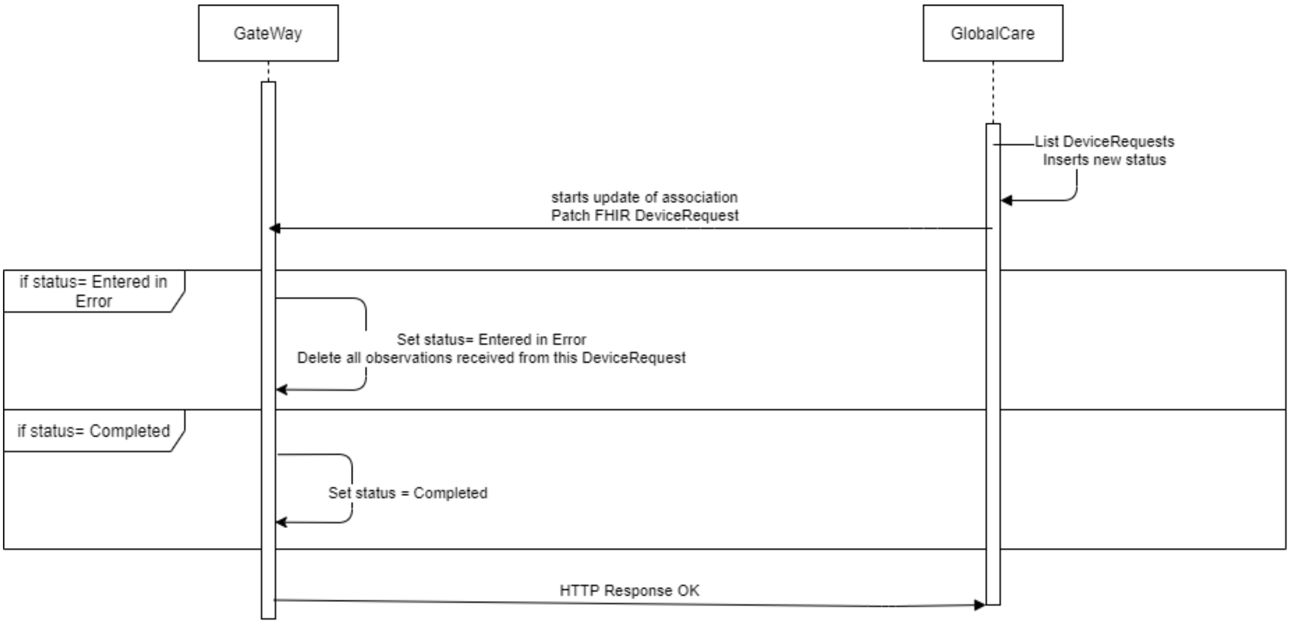

Figure 7. Update Device-Patient-Encounter Association - Sequence Diagram.

For this process, the Gateway should implement an REST API PATCH route at: /DeviceRequest.

6.4 Update Encounter

In order to update the encounter associated with a DeviceRequest, the GlobalCare should list the Parent Devices (Biostickers and Smart Boxes). This list should be able to be filtered by device status and associated patient. Then it should make a PatchRequest. This PatchRequest should contain a DeviceRequest with the new encounter.

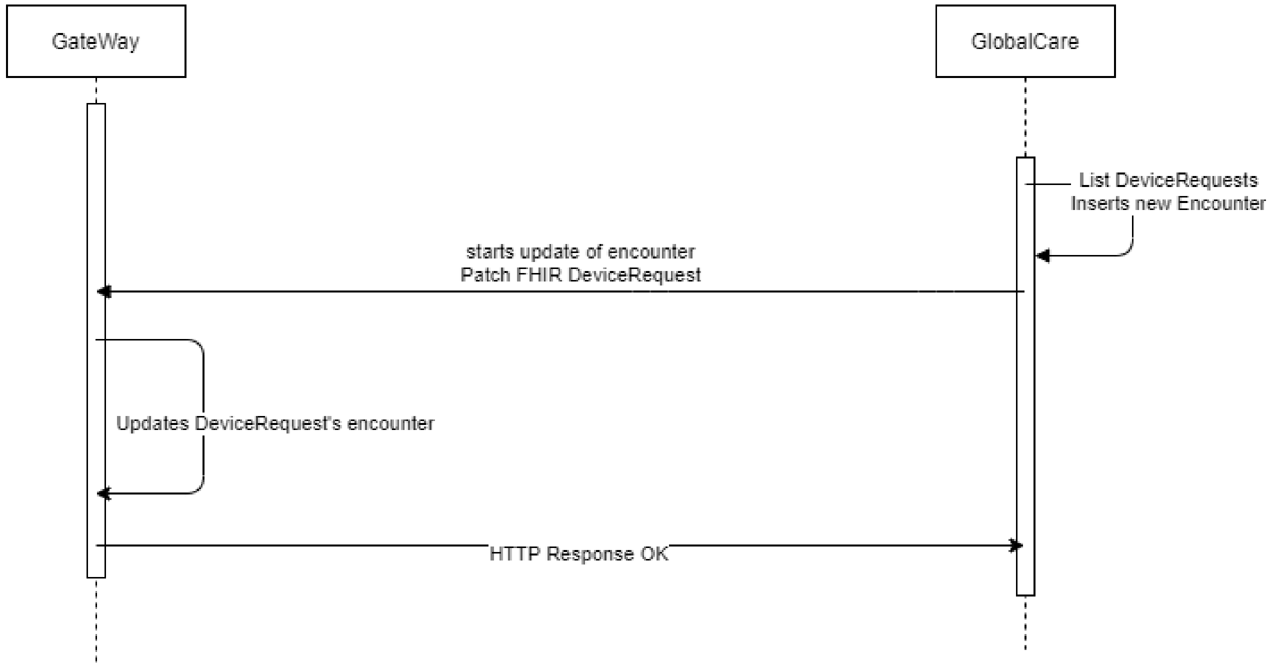

Figure 8- Update Encounter – Sequence Diagram.

For this process, the Gateway should implement an REST API PATCH route at: /DeviceRequest.

6.5 Subscribe Sensor

The GlobalCare should List the Patients. After the practitioner selects the patient the GlobalCare should list the devices associated with the patient. The practitioner should select a device and the timing in which that device should be activated (ex: every 30min). The GlobalCare should translate this in a ServiceRequest and Post it to the Gateway.

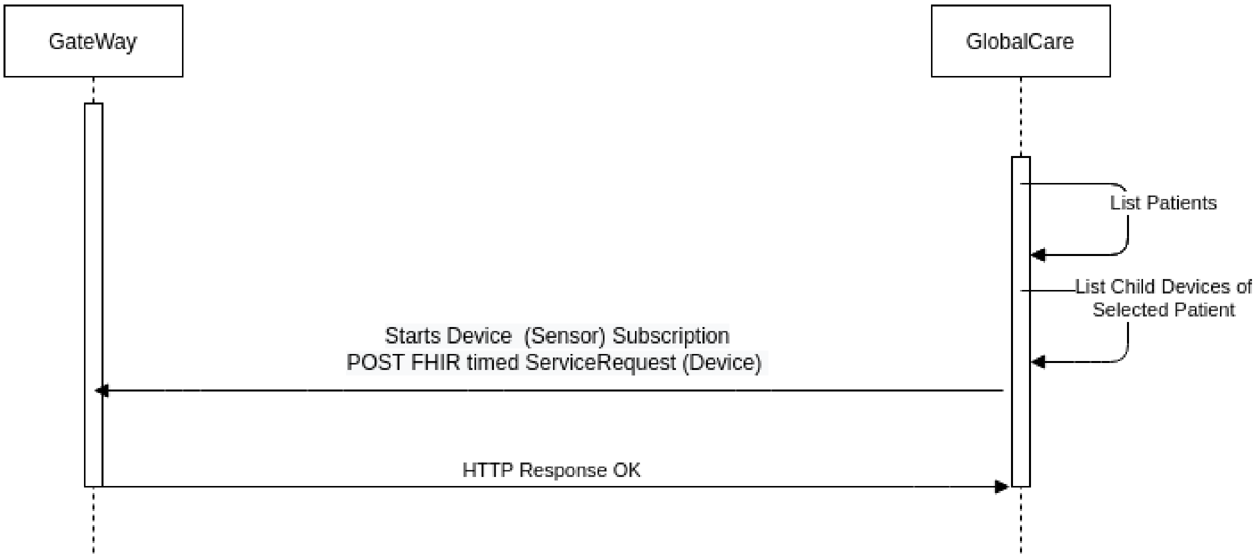

Figure 9. Subscribe Sensor - Sequence Diagram.

For this process, the Gateway should implement an REST API POST route at: /ServiceRequest.

6.6 Update Subscription

In order to update a subscription, GlobalCare should make a PATCH request to the Gateway containing a ServiceRequest with the fields to be changed.

If this update is made with the goal of changing the subscription timing, the ServiceRequest should contain a new timing, if the goal is to finish the subscription, the ServiceRequest status should be set to completed.

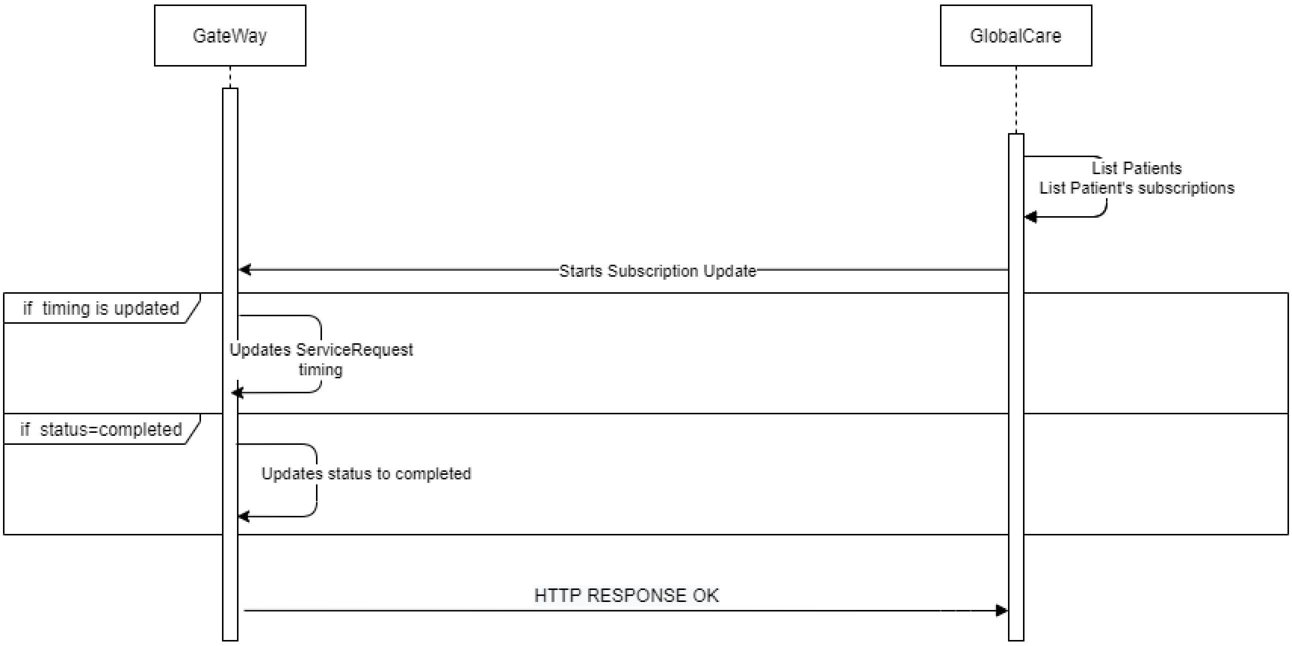

Figure 10. Update Subscription – Sequence Diagram.

For this process, the Gateway should implement an REST API PATCH route at: /ServiceRequest.

6.7 Ask Real-Time Data

In order to obtain real-time data from a device, the GlobalCare should make a POST request to the Gateway containing a ServiceRequest where the occurrenceDateTime is the DateTime at the moment the request is made. The Gateway will ask for the information to the SmartBox. The SmartBox will ask the sensor for data and will return it to the Gateway. The Gateway will convert it to an FHIR Resource and return it to GlobalCare.

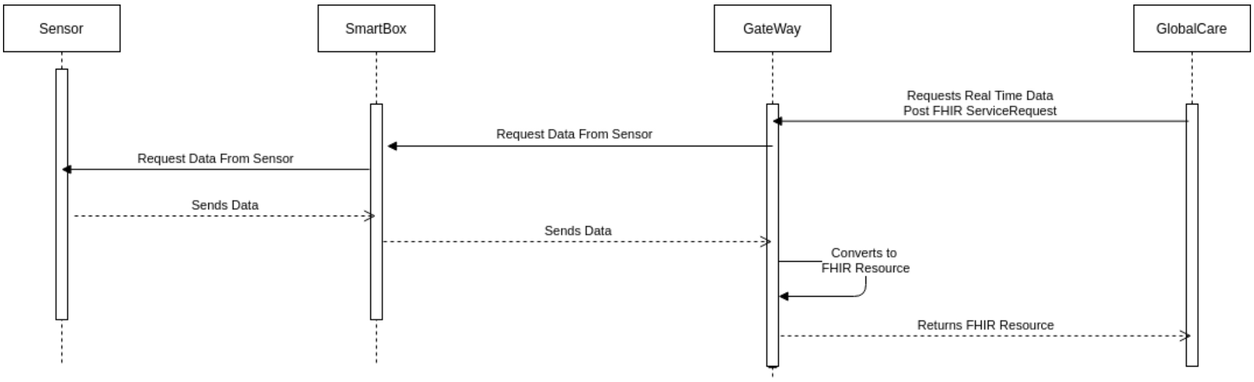

Figure 11. Ask Real-Time Data - Sequence Diagram.

For this process, the Gateway should implement an REST API POST route at: /ServiceRequest.

6.8 Receive Observation

In order to receive an observation triggered by a subscription, the Gateway should be able to initiate an event every time the subscription timing passes, The Gateway will ask the SmartBox for the data subscribed and the SmartBox will get it from the sensor. The SmartBox will then return the data to the Gateway that will transform it in a FHIR resource and post it to the GlobalCare.

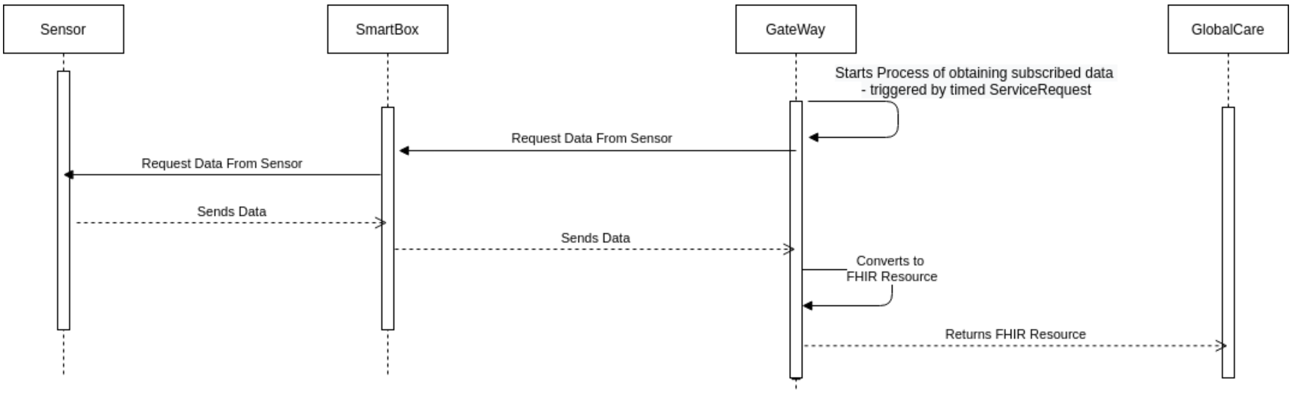

Figure 12. Receive Observation - Sequence Diagram.

For this process, the GlobalCare should implement an REST API POST route at: /Observation.

7. Use Case Mapping

7.1 List Devices

In order to list all the devices, the GlobalCare should be able to make a Devices GET request to the Gateway.

This list should be able to be filtered by the Device Hierarchy (Parent if Device.parent is null else is a child), Device Type (thermometer, etc) and Device Assign (Patient to whom is assigned, null if unassigned). This filter should follow the FHIR filter protocols.

7.2 Attach Device

The GlobalCare should be able to post a DeviceRequest to the Gateway. This Device-Request should reference the Device, the Patient, and the Encounter to be associated.

7.3 Detach Device

In order to update a Device-Patient association, the GlobalCare should be able to make a Patch Request containing a DeviceRequest with the fields to be changed.

These fields must be defined according to the following situations:

- The association is being updated because it was made by mistake. In this case, the Device request should contain a status=Entered in Error. After the patch request is successfully made the Global Care should delete any observations received from the device while the association was active.
- The association is being updated because is finished (The Device is no more connected to the patient). In this case, the GlobalCare should set the DeviceRequest status to Completed.

7.4 Update Encounter

In order to update a Device-Encounter association, the GlobalCare should be able to make a Patch Request containing a DeviceRequest referencing the new Encounter.

7.5 Subscribe Device

The GlobalCare should be able to make a Post request containing a ServiceRequest to the Gateway.

This ServiceRequest should reference the device and should have a timing defined. This timing will define the interval in wich the ServiceRequest should be executed (ex: if the timing is 30min, the ServiceRequest will be executed every 30min).

7.6 Update Subscription

In order to update a subscription, GlobalCare should make a PATCH request to the Gateway containing a ServiceRequest with the fields to be changed.
These fields must be defined according to the following situations:

- The goal of the update is to change the subscription timing. The ServiceRequest should contain the new timing.
- The goal is to finish the subscription. The ServiceRequest should set the status to Completed.

7.7 Ask Real-Time Data

In order to obtain real-time data from a device, the GlobalCare should make a POST request to the Gateway containing a ServiceRequest.

This ServiceRequest should define the occurenceDateTime as the DateTime at the moment the request is made.

7.8 Receive Observation

In order to receive an observation triggered by a subscription, the Gateway should have an available endpoint where the Gateway can post an observation. This observation must be persisted.

8. FHIR Resources

This section will describe all the implemented FHIR Resources and their usage in this project.

8.1 Device

A Device is a type of a manufactured item that is used in the provision of healthcare without being substantially changed through that activity. The device may be a medical or non-medical device (FHIR Device, 2020).

In this project, the sensors are referred to as a child device and the Smartboxes as a parent device.

8.2 DeviceRequest

This resource describes the request for the use of a device by a patient (FHIR DeviceRequest, 2020). In this project, this resource is created when a device(sensor or Smartbox) is associated with a patient.

8.3 Encounter

A patient encounter is further characterized by the setting in which it takes place. Amongst them are ambulatory, emergency, home health, inpatient, and virtual encounters (FHIR Encounter, 2020). It contains zero or more diagnoses and it is used in this project for machine learning purposes.

8.4 Condition

This resource is used to record detailed information about a condition, problem, diagnosis, or other events, situation, issue, or clinical concept that has risen to a level of concern (FHIR Condition, 2020). In this project, it is referred to in the encounter and it is also used for machine learning purposes.

8.5 Observation

Observations are measurements and simple assertions made about a patient, device, or another subject (FHIR Observation, 2020). In this project, an observation represents the output of a sensor (ex: patient's temperature).

8.6 Patient

The patient resource represents the Demographics and other administrative information about an individual or animal receiving care or other health-related services (FHIR Patient, 2020). In this project is used for machine learning purposes, allowing the system to relate the device's outputs and patient's conditions to some of the patient's data.

8.7 ServiceRequest

ServiceRequest is a record of a request for a procedure or diagnostic or other service to be planned, proposed, or performed (FHIR ServiceRequest, 2020). In this project, it is used to make a request of subscription or real-time data to a sensor.

8.8 DocumentReference

DocumentReference is a resource used to reference a general purpose document (FHIR DocumentReference, 2020). Documents can be any serialized object with a mime-type. In this project, it is used to reference IMU's raw data within the FHIR Observations.

8.9 Sensor data observation

According to the FHIR specifications, the applications developed using this protocol must follow specific standards and formats when exchanging messages to ensure interoperability. Thus, the sensors involved in this project should make use of the assignments shown in Table 1.
The technical team has defined that FHIR standards will be used in all communications between the Gateway and the Globalcare interface.

Table 6: Definition of FHIR message formats.

Sensor	Profile	Content Resource	Observation example (JSON format)
Body Temperature	BodyTemp	ID, meta, status, category, code, subject, effective, device and value	observation-example-body-temperature
Electrocardiogram	-	ID, status, category, code, subject, effective, components, and device	observation-example-electrocardiogram
Heart Rate	HeartRate	ID, meta, status, category, code, subject, effective, device and value	observation-example-heart-rate
IMU	-	ID, status, category, code, subject, effective, device and value	See Attachment
Pulse Oximetry	OxygenSat	ID, meta, identifier, partOf, status, category, code, subject, effective, device, value and interpretation	observation-example-pulse-oximetry
Respiration	RespRate	ID, meta, status, category, code, subject, effective, device and value.	observation-example-respiration

The observation examples shown above are used as general guidelines for supporting the decision on converting sensor data to the FHIR standard, and thus are not normative.

In this project, it is fundamental to estimate the patient's pose in order to monitor the patient at hospital or at home. To fulfill the FHIR standards and formats, the team has been working in developing a message for the Inertial Measurement Unit according to the FHIR specification. Therefore, the FHIR specification has been deeply analyzed to ensure the feasibility of an interoperable system.

Regarding IMU sensors, a predefined message format does not exist in the FHIR specification. To implement a compliant message, the FHIR observation must implement the property valueCodeableConcept that describes the patient's body position according to a value set from LOINC, "8361-8" (Body position with respect to gravity). Optionally, for long term data analytics purposes, the raw data from the IMU can also be provided. To accomplish this, the 3D orientation (raw, pitch, yaw) obtained from the sensor should be structured in a JSON format and a DocumentReference containing the JSON data should be created. The DocumentReference resource can then be referenced in a FHIR observation through the derivedFrom property. This feature will be defined later during project development.

9. Conclusions

The WoW project functional specifications and the final system architecture have been defined, and the main components, software layers as well as communication interfaces have been detailed.

This deliverable provides the foundations of the project to drive the technical work, allowing the understanding of how the WoW's integration architecture should be conducted in general and provides a detailed perspective regarding the functional specification, the interaction between the components and the behavior of the runtime system.

In addition, through the use case mapping and the sequence diagrams presented, it becomes clear how the data exchange between all the components will occur. Finally, the FHIR resources description provides an understanding of how the interoperabity between the Gateway and the GlobalCare will be implemented.

In the mid-term future, the functional specification will be taken into consideration for the fulfilment of the technical work ahead, also allowing for adequate planning and integration directives for components to be implemented within the project.

10. References

FHIR Condition. (2020). Obtido de FHIR Condition: https://www.hl7.org/fhir/condition.html

FHIR Device. (2020). Obtido de FHIR Device: https://www.hl7.org/fhir/device.html

FHIR DeviceRequest. (2020). Obtido de FHIR: https://www.hl7.org/fhir/devicerequest.html

FHIR DocumentReference. (2020). Obtido de FHIR DocumentReference: https://www.hl7.org/fhir/documentreference.html

FHIR Encounter. (2020). Obtido de FHIR Encounter: https://www.hl7.org/fhir/encounter.html

FHIR Observation. (2020). Obtido de FHIR Observation: https://www.hl7.org/fhir/observation.html

FHIR Patient. (2020). Obtido de FHIR Patient: https://www.hl7.org/fhir/patient.html

FHIR ServiceRequest. (2020). Obtido de FHIR ServiceRequest: https://www.hl7.org/fhir/servicerequest.html

Attachment

```
// FHIR Observation for IMU sensor
{
    "resourceType": "Observation",
    "id": "IMU",
    "status": "final",
    "category": [
        {
            "coding": [
                {
                    "system": "http://terminology.hl7.org/CodeSystem/observation-category",
                    "code": "activity",
                    "display": "Activity"
                }
            ],
            "text": "Activity"
        }
    ],
    "code": {
        "coding": [
            {
                "system": "http://loinc.org",
                "code": "8361-8",
                "display": "Body position with respect to gravity"
            }
        ],
        "text": "Body position with respect to gravity"
    },
    "subject": {
        "reference": "Patient/example"
    },
    "effectiveDateTime": "1999-07-02",
    "device": {
        "reference": "Device/example"
    },
    “derivedFrom”: [
        {
           "reference": "DocumentReference/example"
        }
    ],
    "valueCodeableConcept": {
        "coding": [
            {
                "system": "http://loinc.org",
                "code": "LA11870-5",
                "display": "Standing"
            }
        ],
        "text": "Standing position"
    }
}
```