CALL H2020-SC1-FA-DTS-2018-2020
Trusted digital solutions and Cybersecurity in Health and Care
TOPIC DT-TDS-01-2019
Smart and healthy living at home

SMART BEAR

"Smart Big Data Platform to Offer Evidence-based Personalised Support for Healthy and Independent Living at Home"

D6.8 (D74) – Report on integrated SMART BEAR platform v0.1

Due date of deliverable: 31-03-2021
Actual submission date: 21-05-2021

Grant agreement number: 857172
Start date of project: 01/09/2019
Lead contractor: CNR
Duration: 60 months

Project funded by the European Commission within the EU Framework Programme for Research and Innovation HORIZON 2020	
Dissemination Level	
PU = Public, fully open, e.g. web	✓
CO = Confidential, restricted under conditions set out in Model Grant Agreement	
CI = Classified, information as referred to in Commission Decision 2001/844/EC.	
Int = Internal Working Document	

D6.8 (D74) – Report on integrated SMART BEAR platform v0.1

Editors
Manuel M Perez Perez - ATOS
Alberto Acebes - ATOS
Marco Decandia - ARIA

Contributors
Adrian Arroyo Perez - ATOS
Jose Gato - ATOS
Elsa Prieto - ATOS
Alberto Zanini - ARIA
Othonas Soultatos – STS
Konstantina Koloutsou - STS
Ioannis Basdekis - STS
Jonatan Maggesi - UMIL
George Zissis - ATC
Ioannis Kouris - ICCS
Stefano Siccardi - UMIL
George Giotis - ATC
Luigi Gallo - CNR
Konstantin Pozdnyakov - CITY
Christos Kloukinas - CITY
Manolis Chatzimpyrros - STS

Reviewers
Stéphane Railhet - SV
Ioannis Basdekis - STS

The work described in this document has been conducted within the project SMART BEAR, started in September 2019. This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 857172

©Copyright by the SMART BEAR Consortium.

Executive Summary

This document shows the methodology used for the implementation of the basic functionalities of the SB@Cloud components when integrated for the first prototype to be used for the Pilot of Pilots (PoP).

Both the initial design of the architecture specified in deliverable (D2.2), as well as the subsequent adaptations, define a series of functional blocks and interfaces between them. Security, scalability, and modularity are prioritised providing a procedural mechanism to facilitate integration.

Also, as described in the architecture design, among the possible integration and deployment options, the containerisation of the components has been chosen as the most adequate and advanced mechanism, in line with the current trend, to facilitate the independence between components and the hypothetical promotion of the chosen architecture to any real production environment, with stability and scalability requirements.

The selection of this integration mode entails an extra degree of complexity that more traditional deployments would not have (bare metal, Virtual Machine) with a demanding set of tools and procedures. But the realisation of the project with orientation and implementation of an efficient Continuous Integration procedure (CI), with proper testing software engineering approach, and also Continuous Delivery/Deployment (CD), has been considered from the beginning an important requirements for the technical development of the project.

The research done to select the appropriate tools is not described in the present document as they will be properly documented in the integration deliverables, D6.1 and D6.2, to be released at month 30, where the DevOps methodology will be completely set up. The full integration process will use the most widespread and well-known DevOps tools, including CI/CD from the source code repository (GitLab) to the deployment of the application in a Kubernetes[1] cluster. Applying CI/CD principles from the very beginning, (i.e. Agile-based software development, branching workflows) is key for avoiding the misuse or even withdrawal of the CI/CD processes. Therefore, the selected tools are thoroughly discussed and tested for usefulness. The current CI/CD configuration is the basis for future integrations and deployments of SMART BEAR, although, at the present state, the focus is on the basic integration of the use cases proposed by the reviewers.

These use cases are the basis for the demonstration of the current prototype. Therefore, it includes only basic functionalities to test data workflow and components interaction. The first prototype can be considered as the basis for the Pilot of Pilots (PoP) to be carried out from June 2021.

The PoP main advantage is to receive early feedback from all stakeholders involved in the project, including medical professionals.

This report documents the integrated platform's functionalities, the final amendment made to the SMARBEAR architecture and the outcomes of the testing activities.

[1] https://kubernetes.io/ Open-source system for auto deployment, scaling, and management of containerized applications.

Contents

Executive Summary **3**
List of acronyms **6**
List of tables **7**
List of figures **8**
1 Introduction **9**
1.1 Overview 9
1.2 Goals of the deliverable 10
2 SMART BEAR platform for the Pilot of Pilots **11**
2.1 SB@HomeHub 12
2.2 SB@App 12
2.3 SB@Cloud 12
2.4 Cross pilot interaction and synergies 12
2.4.1 HOLOBALANCE 12
2.4.2 Smart4Health 13
3 Pilots of Pilots Functionalities **14**
3.1 Use Case 1: Participant registration & devices pairing to participants 14
3.2 Use Case 2: Security authentication authorization 15
3.3 Use Case 3: Flows of data from devices to repository 16
3.4 Use Case 4: FHIR support and semantic annotations 17
3.5 Use Case 5: Data analytics (synthetic data) & interventions. 17
3.6 Use Case 6: Mobile app (view of own data, delivery of interventions) 18
3.7 Use Case 7: Synergies support (data ingestion –ID correlations) 19
3.8 Use Case 8: Security (anonymization, access control, component and user authentication) 19
4 Prototype integration **21**
4.1 SB@Cloud Components 22
4.1.1 Secure Manager One 22
4.1.2 Data Repository 23
4.1.3 Dashboard 23
4.1.4 BDA 24
4.1.5 Non-FHIR_database 24
4.1.6 DSS 25
4.1.7 SB@App 25
4.1.8 Garmin intermediate server 25
4.1.9 Notification manager 26
5 SMART BEAR components requirements for CI/CD **27**
6 Testing the PoP (Pilot of Pilots) **28**

6.1 Methodology 28
6.2 Goals and scope 28
6.3 Testing Pre-conditions 29
7 Test procedure and plan 31
7.1 PoP_F1_TC1 31
7.2 PoP_F1_TC2 31
7.3 PoP_F2_TC1 32
7.4 PoP_F2_TC2 33
7.5 PoP_F3_TC1 34
7.6 PoP_F4_TC1 34
7.7 PoP_F4_TC1 35
8 Concluding Remarks 37

List of acronyms

Acronym	Description
ACME	Automatic Certificate Management Environment
API	Application programming interface
BDA	Big Data Analytics
CA	Certificate Authority
CDR	Clinical Data Repository
CI/CD	Continuous Integration / Continuous Deployment
DSS	Decision Support System
EHR	Electronic health record
GDPR	General Data Protection Regulation
HB	HOLOBALANCE
HIS	Health Information System
MVP	Minimum Viable Product
OSGi	Open Service Gateway Initiative
PoP	Pilot of Pilots
REST API	(or RESTful API) is API that conforms to the constraints of REST architectural style and allows for interaction with RESTful web services
SB	Smart Bear
SB@App	Smart Bear smartphone app
SB@Cloud	Smart Bear Cloud infrastructure
SB@Dashboard	Smart Bear Dashboard component
SB@HomeHub	Smart Bear Home Hub component
SB@Repository	Smart Bear Repository component
SB@SecurityComponent	Smart Bear Security component
S4H	Smart for Health
TLS	Transport Layer Security
VM	Virtual Machine

List of tables

Table 1: Summary of the functional components implemented for the PoP. 21
Table 2: List of SB@Cloud components as containers. 22
Table 3: SB@Cloud component Secure Manager One details. 23
Table 4: SB@Cloud component Data Repository. 23
Table 5: SB@Cloud component Dashboard. 24
Table 6: SB@Cloud component BDA. 24
Table 7: SB@Cloud component Non-FHIR Database. 24
Table 8: SB@Cloud component DSS. 25
Table 9: SB@Cloud component SB@App. 25
Table 10: Component Garmin intermediate server. 25
Table 11: Component Notification manager. 26
Table 12: List of SB@Cloud components as containers. 27
Table 13: Testing scenarios. 29
Table 14: Testing preconditions, component responsability 30
Table 15: Testing functionality components relation 30
Table 16: Test PoP_F1_TC1 31
Table 17: Test PoP_F1_TC2 32
Table 18: Test PoP_F2_TC1 33
Table 19: Test PoP_F2_TC2 33
Table 20: Test PoP_F3_TC1 34
Table 21: Test PoP_F4_TC1 35
Table 22: Test PoP_F4_TC1 36

List of figures

Figure 1 Integration with CI/CD containers orchestration vs manual deployment to VM. 9
Figure 2: SMART BEAR Overall updated architecture showing the interdependencies among the main building blocks. 10
Figure 3: Scheme of the SB PoP showing the different layers 11
Figure 4: The HOLOBALANCE application communicates with the SMARTBEAR platform via an API Rest. The patient is registered in both platforms. 13
Figure 5: The Smart4Health platform interacts with SMARTBEAR. 13
Figure 6: Use case 1, workflow of user/device registration process 14
Figure 7: Use case 2, workflow of user authentication and authorisation 15
Figure 8: Use case 3, flows of data from devices to repository 16
Figure 9: Use case 4, security FHIR support and semantic annotations 17
Figure 10: Use case 5, data analytics (synthetic data) & interventions 17
Figure 11: Use case 6, mobile app (view of own data, delivery of interventions) 18
Figure 12: Use case 7, synergies support (data digestion –ID correlations) 19
Figure 13: Use case 8, security (anonymisation, access control, component, and user authentication) 19
Figure 14: Hi Level whole set of functional components 21
Figure 15: Security component diagram showing the interaction between the different security components. 22

1 Introduction

1.1 *Overview*

In order to integrate and operate with the complexity of the SMART BEAR platform, CI/CD methodology is proposed. The initial cumbersome of setting up the operating framework for CI/CD is compensated later on, as it paves the way for integration and deployment of new releases.

The steps defined by the architecture for integration with the selected methodology would be:

1. Set up a centralised *git* repository (GitLab provided by ATOS).
2. Development of the different components with the ability to be containerised.
3. Installation of a Kubernetes cluster/s in the ICCS cloud infrastructure, with proper security and networking (considered a prerequisite to deploy the SMART BEAR platform on it).
4. Deploy and run the SMART BEAR platform administrator tools.
5. Deploy and run the SMART BEAR application services.
6. Run tests and pilots.

However, this process requires the setup of the Kubernetes framework to work robustly and, although the initial processes and configurations for Kubernetes have been set and tested, the framework needs further refinements to work smoothly as a number of configuration issues arose. As the time given for setting up the PoP is tight, we decided, for this first prototype, to integrate the components in a hybrid environment, using containerised components installed in a Virtual Machine (VM) without enabling a container orchestrator, i.e. using the docker images in manual mode without Kubernetes. Therefore, to integrate and operate the PoP on SMART BEAR, a traditional approach based on docker compose containerisation is proposed.

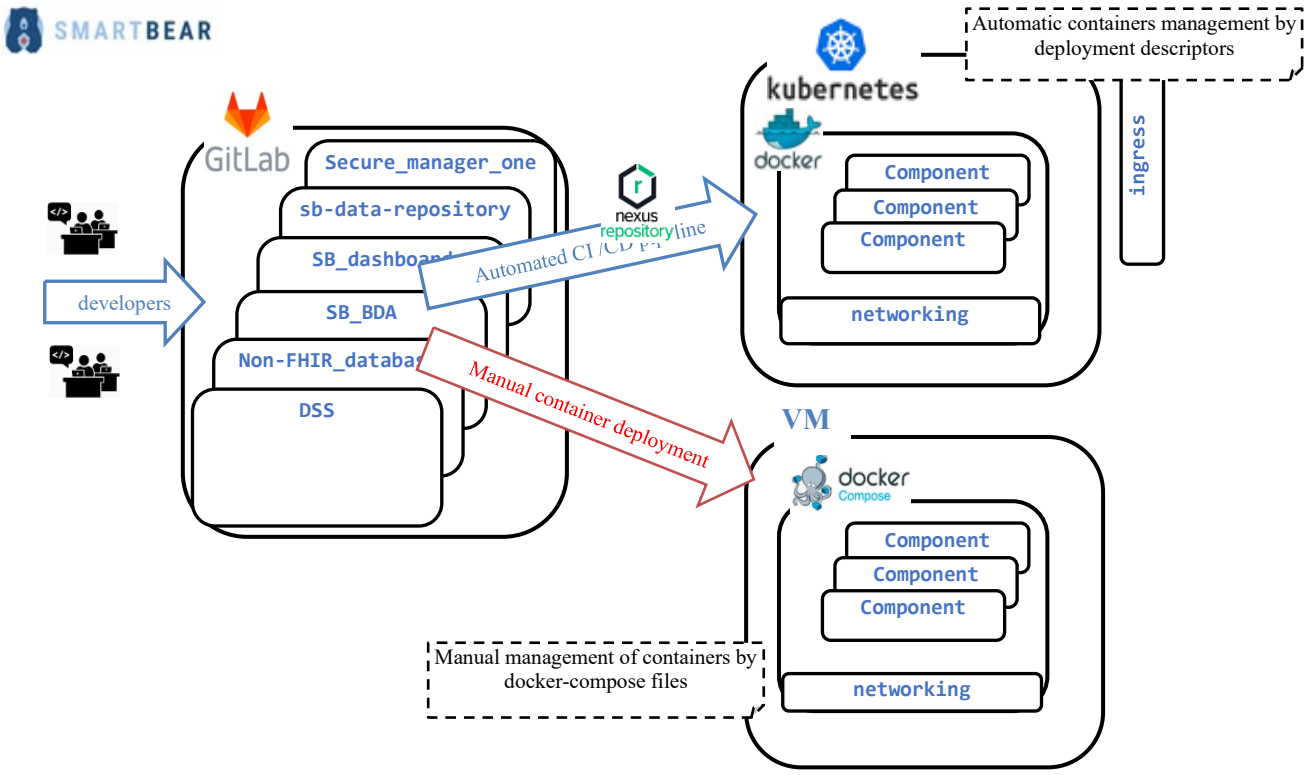

Figure 1 Integration with CI/CD containers orchestration vs manual deployment to VM.

In this deliverable, we describe the activities carried out to integrate those different components into a cohesive SMART BEAR Platform for the PoP (see Table 1).

1.2 *Goals of the deliverable*

We consider the SMART BEAR Platform as the integration of the cloud components (which will run in a containerised environment over a Kubernetes cluster) plus a set of common application services, methods and tools provided by SMART BEAR as well as the functionalities of HOLOBALANCE[2] AND SMART4HEALTH[3] projects to facilitate the development of decentralised applications.

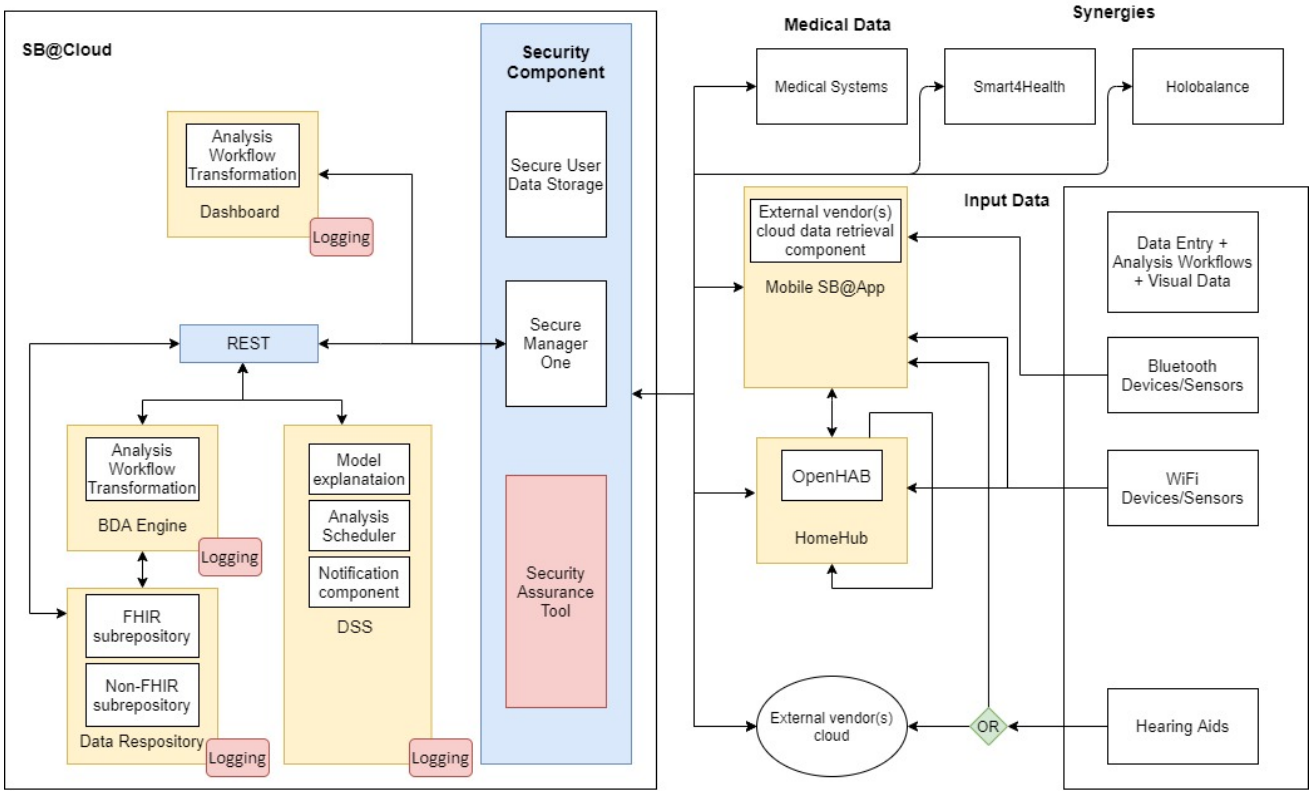

Figure 2: SMART BEAR Overall updated architecture showing the interdependencies among the main building blocks.

Specifically, we describe the dashboard, the security components, the BDA, the DSS and the databases and which are the procedures for the integration and installation. Furthermore, since those components have been documented in the architecture specification deliverable (D2.2) and development WPs (D4.2. D5.2), in this deliverable we are focused on those aspects which are needed for the integration of SMART BEAR.

[2] https://www.holobalance-project.eu

[3] https://smart4health.eu

2 SMART BEAR platform for the Pilot of Pilots

This Pilot of Pilot (PoP) aims to demonstrate the basic functionalities of the SMART BEAR platform in a real clinical site. The lesson learnt during the pilot will provide valuable input for refinement of the platform and for easing the integration and deployment during the pilot phase.

Therefore, the PoP includes only basic functionalities according to the suggestions provided during the first review of the project. The PoP will demonstrate the basic functionalities for SMART BEAR in order to tests the data workflows and components interaction. The prototype for the PoP will allow to get feedback from all stakeholders involved in the project, including medical professionals.

The figure below describes how the information is flowing in the SMART BEAR ecosystem and how the components are interconnected. This schema would help to break down acceptance criteria tests to check that what it has been promised to the users it is actually what it is delivered.

The objective is to show the different modules interconnected in order to provide the final SMART BEAR system ready for the piloting stage. Ten different uses cases have been defined for being evaluated during the PoP.

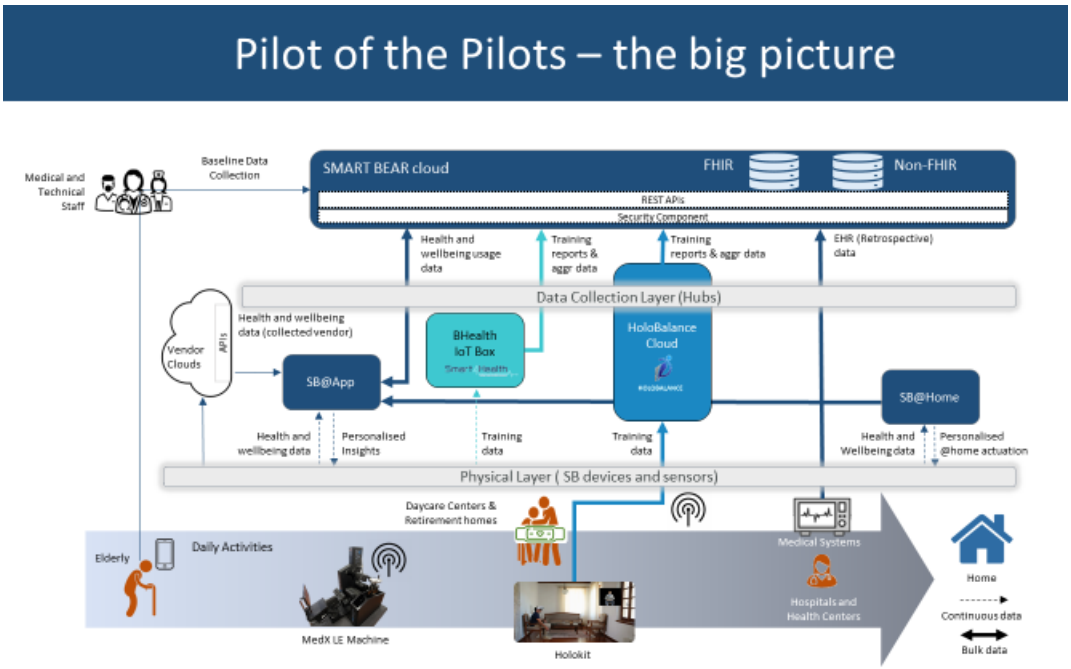

Figure 3: Scheme of the SB PoP showing the different layers

As described in the deliverable D2.2 “SMART BEAR Architecture”, three main systems are included in the SMART BEAR solution, namely: the Mobile Phone App (SB@App), the SMART BEAR SB@HomeHub and the SMART BEAR Cloud (SB@Cloud). In addition, the SMART BEAR platform integrates some services from the HOLOBALANCE and the Smart4Health projects.

2.1 *SB@HomeHub*

The Smart Home-HomeHub component of the SMART BEAR project is based on the openHAB software The openHAB is an open-source implementation of the Eclipse Smart Home Framework, which aims to provide a common approach for solving problems related to protocol connectivity, security and software development in the world of Internet of Things, including common smart home and smart city solutions. The HomeHub component is the subsystem accumulating inputs from house-bound sensors such as house temperature sensors, house light sensors, etc.

2.2 *SB@App*

The SB@app gathers physiological information from the patients´ devices (thermometer, oximeter, scale, etc...). The SB@App mobile application acts as gateway between all the sensors and the SB@Cloud backend. The data transmitted will be firstly formatted in HL7FHIR before being transmitted.

2.3 *SB@Cloud*

The SB@Cloud refers to the data storage and analytics services. The SB@Cloud will store the data transmitted by the SB@HomeHub and the SB@App.

Raw data to be analysed and aggregated, as well proposed interventions are stored in the non-FHIR database. Clinical data, physiological data sent by the SB@App and aggregated data are transformed to HL7 standard to be stored in the HAPI FHIR[4] server. The data is available to the Big Data Analytics (BDA) to perform decision models feed the rules of the Decision Support System (DSS), to suggest recommendations and personalised interventions.

2.4 *Cross pilot interaction and synergies*

The SMART BEAR project will interact and provide combined services with the HOLOBALANCE[5] and Smart4Health[6] projects

2.4.1 HOLOBALANCE

The SMARTBEAR platform interacts with the HOLOBALANCE platform. That platform provides rehabilitation services aimed at assisting patients to perform physiotherapist treatments. Once the physiotherapist set up the personal patient rehabilitation training program. The patient can follow the treatment using the *Holobox*, a projected holographic physio coach. The patient's wearable and IoT sensors will allow patient follow up. All the patient's rehabilitation training program is stored in the HOLOBALANCE cloud. The patient registered in SB platform can request and receive these FHIR-compliant data.

The Figure 4 below shows the interaction between the HOLOBALANCE and SMARTBEAR services

[4] https://hapifhir.io/

[5] www.holobalance.eu

[6] www.smart4health.eu

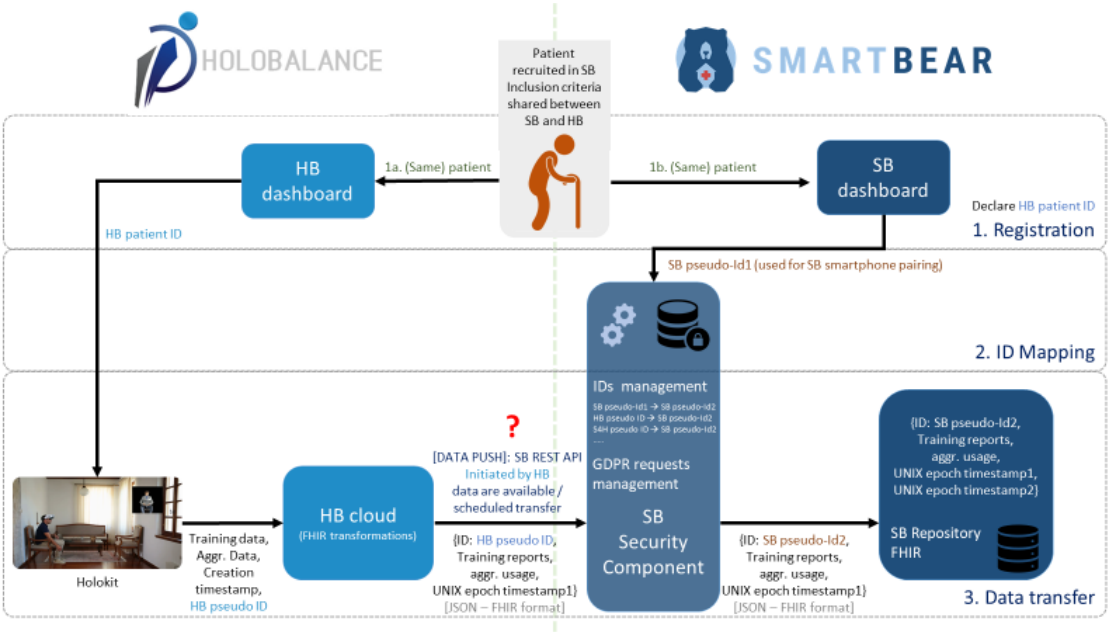

Figure 4: The HOLOBALANCE application communicates with the SMARTBEAR platform via an API Rest. The patient is registered in both platforms.

2.4.2 Smart4Health

The SMART BEAR platform also interacts with the Smart4Health platform. The Smart4Health platform takes advantage of the MedX machine.[1] for spine therapy. The platform allows the physiotherapist to perform the necessary machine calibration procedures and the initial force test to set the baseline back health parameters. As soon as the exercise is concluded, data is sent to both the Smart4Health Health Monitor and the SMART BEAR platform

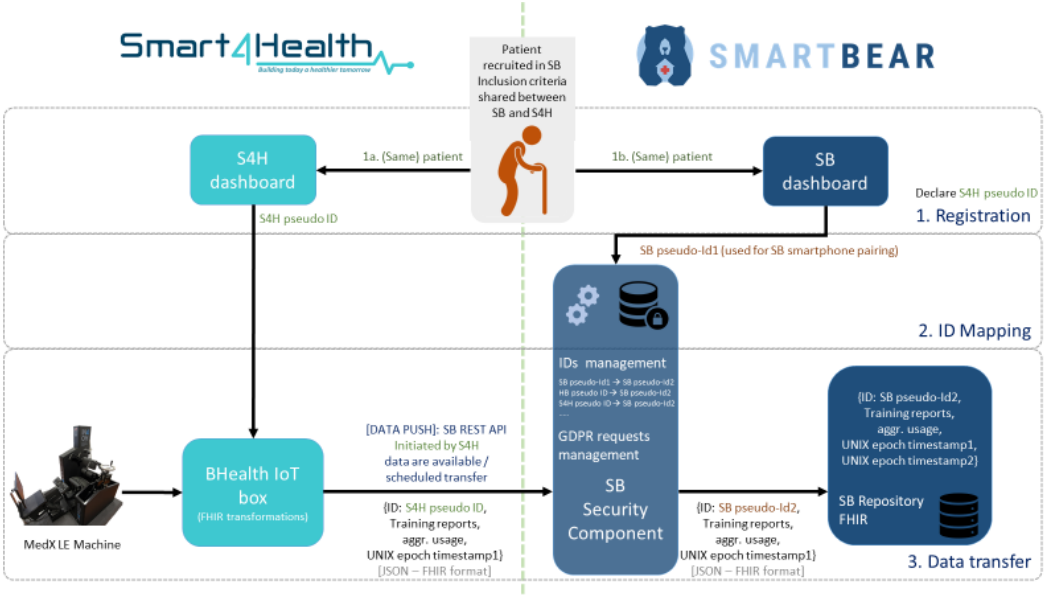

Figure 5: The Smart4Health platform interacts with SMARTBEAR.

3 Pilots of Pilots Functionalities

This chapter describes the proposed uses cases for the PoP and how the information is flowing in the SMART BEAR ecosystem and how the modules are interconnected. Furthermore, it gives an idea of what it is being shown and retrieved from a high-level perspective, and where data comes from.

This schema would help to break down acceptance criteria tests to check that what it is been promised to the users it is what it is delivered.

The objective is to show the different components Interconnected to provide the final SMART BEAR system ready for the piloting stage. Eight different uses cases are defined. For each use case the workflow has been defined for better understanding and easing the integration:

- **Use case 1**: Participant registration & devices pairing.
- **Use case 2**: Security: authentication and authorisation
- **Use case 3**: Flows of data from devices to repository
- **Use case 4**: FHIR support and semantic annotations
- **Use case 5**: Data analytics (synthetic data) & interventions
- **Use case 6**: Mobile app (view of participant´s own data, delivery of interventions)
- **Use case 7**: Synergies support (data digestion –ID correlations)
- **Use case 8**: Security (anonymisation, access control, component and user authentication)

3.1 *Use Case 1: Participant registration & devices pairing to participants*

The Dashboard allows the doctors setting the baseline configuration parameters. The configuration parameters and the interventions are formatted and stored in the HAPI FHIR implementation of the HL7 FHIR[7].

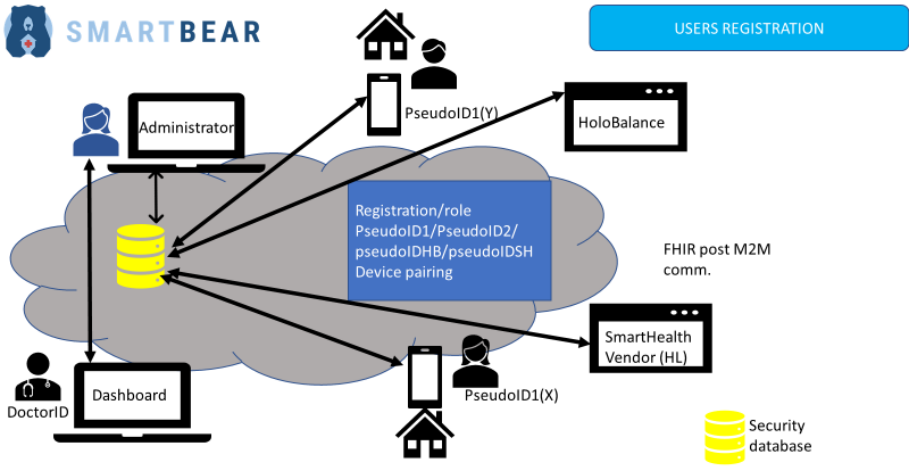

Figure 6: Use case 1, workflow of user/device registration process

[7] http://hapifhir.io/

Below, the workflow is broken down into steps:

1. The Dashboard allows the doctors setting the baseline configuration parameters. The configuration parameters and the interventions are modelled using HL7 FHIR stored in the HAPI implementation of the HL7 FHIR. Below the workflow is broken down in steps:
2. The patient must register the SMART BEAR application, the anonymous id is propagated to HAPI (to unify the patient id). For the prototype this is done manually (one patient)
3. The patient list is displayed showing the psychological profile and the pending notifications coming from the DSS.
4. The doctor sets the schedule (see parameters above) using the Dashboard web interface:
5. The DSS reads the information about the schedule and uploads the intervention into HAPI HL7 FHIR.
6. The Dashboard displays to the doctors the pending notification (the exercise plan) generated by the DSS. They can accept or deny them. Finally, if the exercise plan is accepted it is notified to the patient via the SB@App
7. SMART BEAR account creation, assuming that the S4H and/or HB integration has been completed: First, the HB and S4H user's accounts are created. The HB user ID and the S4H user ID are associated with the SB participant ID.

3.2 *Use Case 2: Security authentication authorisation*

This use case aims to demonstrate the security functionalities of the platform related to access control and user authentication.

Once the clinicians are registered, they can log in to the SB platform, and after successful authentication, the security component controls the access allowing only authorised actions.

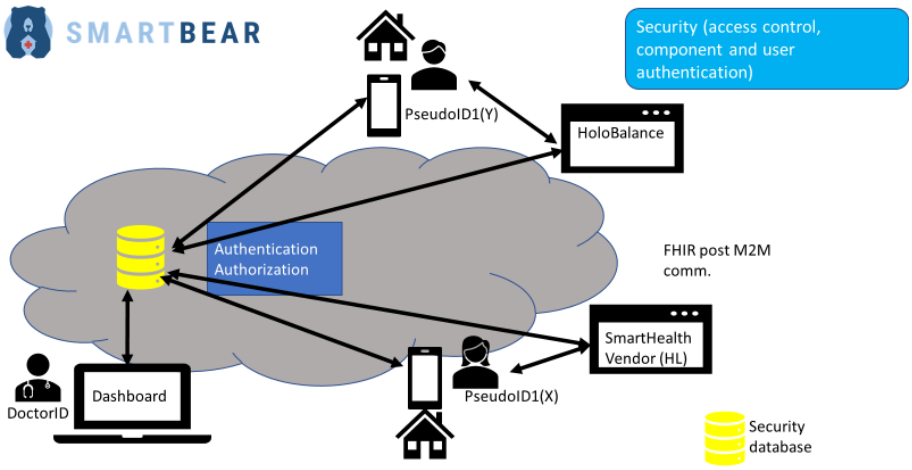

Figure 7: Use case 2, workflow of user authentication and authorisation

1. The doctor logs-in in to the SB platforms and the security component checks the user is registered in the security database.
2. The user is associated with his/her profile, and based on that, they can perform different tasks.

3. The first-time patients are registered in the systems they are assigned with an internal pseudo-id.

3.3 *Use Case 3: Flows of data from devices to repository*

This use case aims to demonstrate how the data is transmitted from the sensors to the FHIR HAPI server.

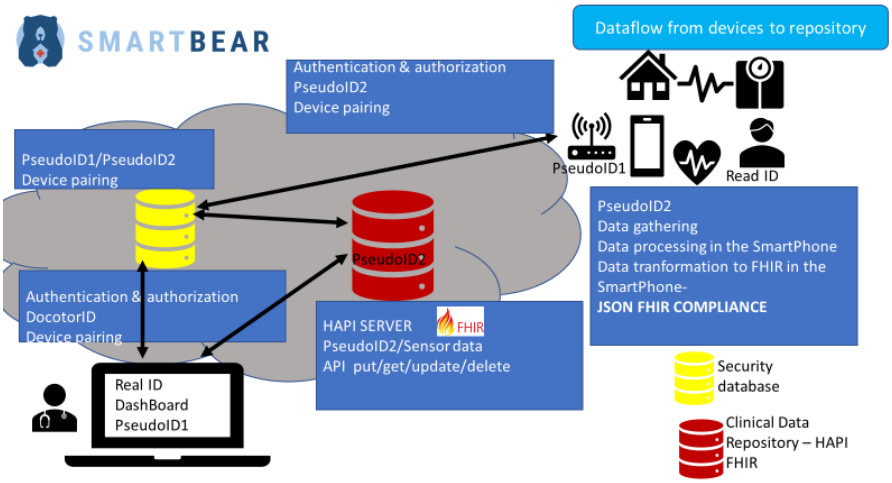

Figure 8: Use case 3, flows of data from devices to repository

1. The sensors and devices at home are paired with SB@app.
2. The sensors collected data are finally transmitted from the SB@app to the cloud once the users are authenticated and authorised.
3. The data is transmitted in HL7 FHIR format.
4. The security component authenticates the process and correlates patient IDs.
5. If there are not security issues the data is stored in the HAPI FHIR database.

3.4 *Use Case 4: FHIR support and semantic annotations*

This use case aims to demonstrate data transformation to the HL7 FHIR standard. In this use case for testing purposes, synthetic data is used.

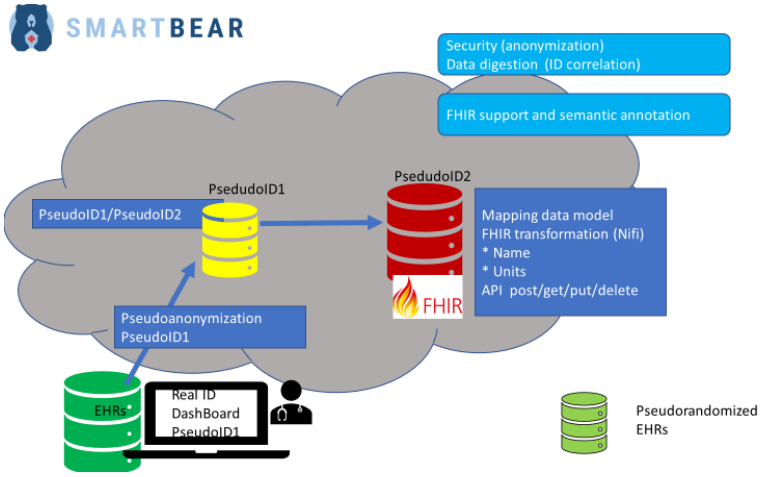

Figure 9: Use case 4, security FHIR support and semantic annotations

1. The doctor logs on to the platform
2. The doctor uses the dashboard to request the ingestion of EHRs into the HAPI FHIR repository.
3. The data is firstly pseudo-anonymised and then transformed to the HL7 FHIR standard.
4. The transformed data is stored in the database using the internal SMART BEAR pseudoID2 for the patients.

3.5 *Use Case 5: Data analytics (synthetic data) & interventions.*

This use case aims to demonstrate how the interventions are assigned based on patient's data and BDA models. In this case, synthetic data is also used.

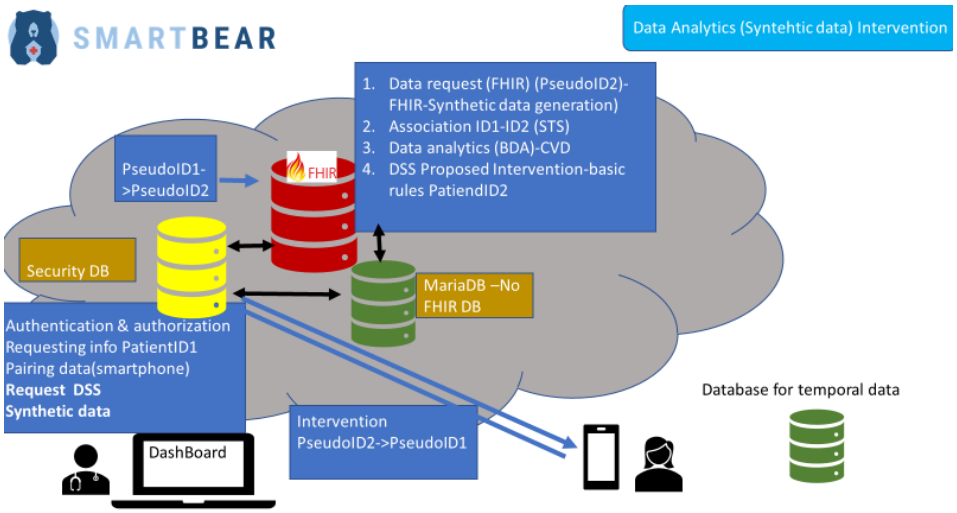

Figure 10: Use case 5, data analytics (synthetic data) & interventions

1. The doctor log on the platform
2. The doctor uses the dashboard to request statistics and information for a specific patient.
3. The security component allows or denies the operation based on the doctor´s profile and request.
4. The SMART BEAR analytics access patient´s information and perform the requested operation.
5. The DSS returns the information to the doctor.

3.6 *Use Case 6: Mobile app (view of own data, delivery of interventions)*

This use case aims to demonstrate how the patient requests to see his/her data and interventions.

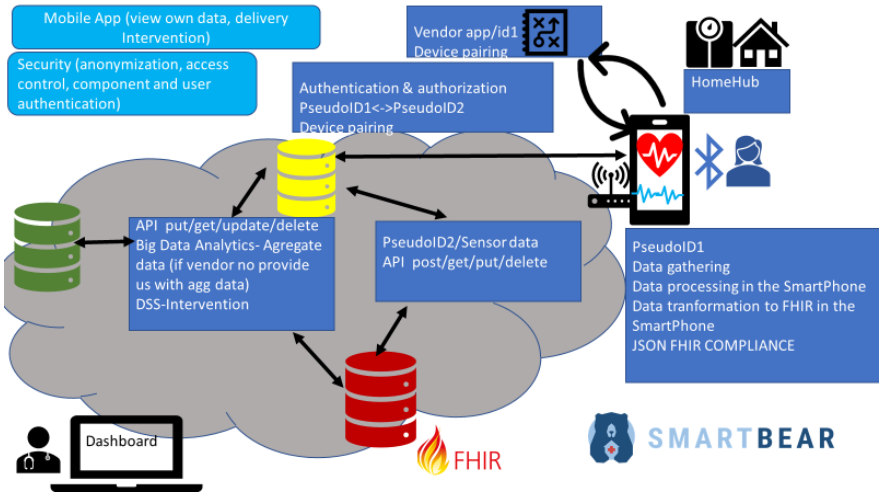

Figure 11: Use case 6, mobile app (view of own data, delivery of interventions)

1. The sensors and devices transmit the data from the SB@app to the SB@cloud.
2. The data is transmitted in HL7 FHIR format.
3. The security component authenticates the process and correlates patient pseudoIDs.
4. If there is not security issues the data is stored in the HAPI-FHIR repository.
5. The patient requests to see his/her data and provided interventions.
6. The request after being checked according by the security component is sent to the patient.

3.7 *Use Case 7: Synergies support (data ingestion –ID correlations)*

This use case aims to demonstrate the digestion of the data and the pseudonymisation process.

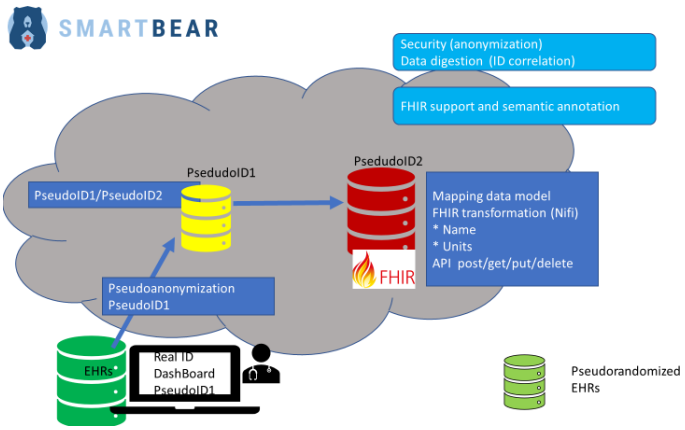

Figure 12: Use case 7, synergies support (data digestion –ID correlations)

1. The doctor logs in.
2. The doctors requests via the dashboard to ingest patient's data.
3. The security component accepts or denies the request according doctors profile.
4. The data is transmitted, pseudorandomised.
5. The security component checks the process and correlates patient IDs.
6. The data is transformed in HL7 format.
7. The e data is *stored in the HAPI-FHIR repository*

3.8 *Use Case 8: Security (anonymisation, access control, component and user authentication)*

This use case aims to demonstrate the authentication and authorisation of the users and assigning them with the pseudoIDs for operating in the SMART BEAR, HOLOBALANCE and Smart4Health platforms.

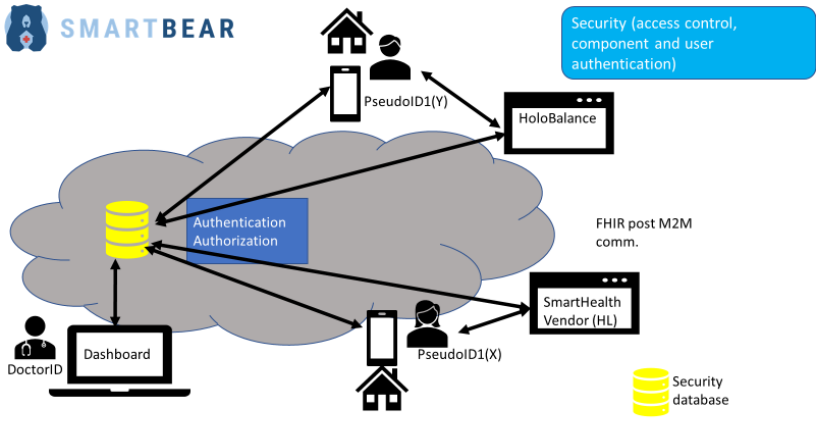

Figure 13: Use case 8, security (anonymisation, access control, component, and user authentication)

1. The doctor logs in.
2. The doctors requests via the dashboard to see patient's data.
3. The security component accepts or denies the request according doctors profile.
4. Patient data is transmitted from the SB@App to the SB@Cloud
5. The security component checks the process and correlates patient IDs.
6. The security component authorised users (patients, doctor) to see their own data(patients) and see patients´ data (doctors according their authorisation profile)

4 Prototype integration

The starting point of the first prototype integration was the identification of the components. The components are deployed in the SB@Cloud or at the home/patient environment. The SMART BEAR platform interacts with both the HOLOBALANCE and Smart4Healht platforms.

The table below summarises the list of components, the infrastructure/location where they are deployed and their functionalities (This table is reproduced in D6.7 for better clarification).

ID	Infrastructure	Component	PoP (functionality)
1	SB@Cloud	SB@Cloud (infrastructure/integration)	MVP:[8] Intermediate integration framework to support PoP
2	SB@Cloud	SB@REST	MVP: Registration, Device paring, GDPR requests management, Medical history…
3	SB@Cloud	SB@SecurityComponent	MVP: Registration, Device paring, GDPR requests management, Medical history.
4	SB@Cloud	SB@Dashboard	MVP: Registration, Device paring, GDPR requests management, Medical history.
5	SB@Cloud	SB@BDA	MVP: Analytics
6	SB@Cloud	SB@Repository	MVP: Medical history, Questionaries answers
7	SB@Cloud	SB@DSS	MVP: Interventions, Alerts
8	Home environment User environment (wearables)	SB@App	MVP: Pseudonymised registration, Notifications, Heart Rate, Sleep Statistics…
9	Home environment	HomeHub/openHAB	MVP: Home Devices & Sensors
10	Other	HA Vendor Cloud	HA usage and other aggregated data per study patient
11	Other	Smart4Health data HoloBalance data	TBC
12	Other	Security/Privacy assessments	

Table 1: Summary of the functional components implemented for the PoP.

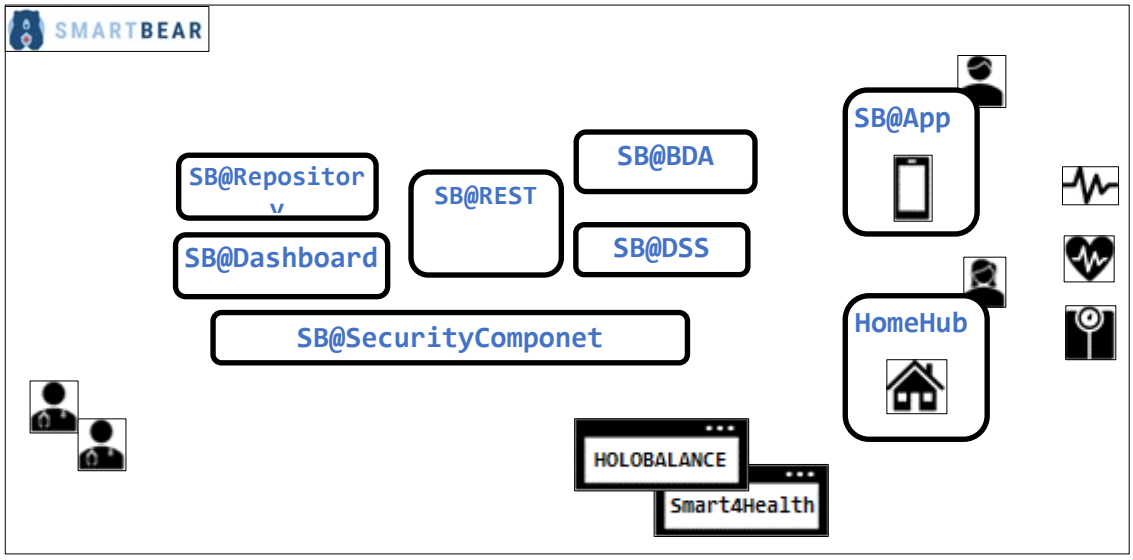

Figure 14: Hi Level whole set of functional components

[8] MVP stands for Minimum viable product (MVP). MVP indicates mandatory features to support to support PoP

4.1 *SB@Cloud Components*

Once detailed the whole set of the platform ecosystem components, we focus now in the ones specifically part of the SB@Cloud.

Since the architectural decision was to perform the integration by means of a containerised components, which is the current standard for integration of components and services in a cloud environment, each of the following represents a single container and, also due to the implementation and technologies used, them will not always correspond with the conceptual functional unit.

	Internal component ID	Subcomponent	Description
1	Secure_manager_one	usermanager	User manager API under /users
		am	WSO2 API Management UI
		is	WSO2 API Identity Server UI
		postgres	Relational DB supporting Identity server
2	sb-data-repository	sb-data-repository	HL7 FHIR repository based on HAPI FHIR server to manage the clinical data
		mariadb	Relational DB supporting sb-data-repository
3	SB_dashboard	sb_dashboard	SB Dashboard
4	SB_BDA	Sb_bda	Big Data Analytics
5	Non-FHIR_database	database	DSS auxiliary data base
6	DSS	api_dss	Decision Support System
		dss_core	DSS core services

Table 2: List of SB@Cloud components as containers.

4.1.1 Secure Manager One

Security manager one is the only component that exposes it interface to be consumed. The Security manager one acts as an api-gateway [1]. Also, it is the component that provides/exposes the REST APIs for Authentication and user management.

The Authentication is done with Oauth2 and user management with SCIM[2]

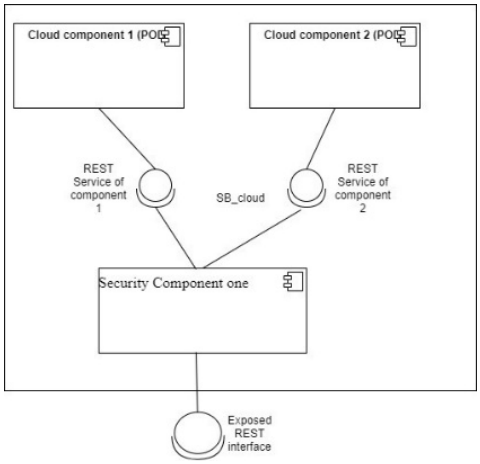

Figure 15: Security component diagram showing the interaction between the different security components.

The component is splitted in four subcomponents/containers

- *usermanager*: User API
- *am*: API Manager
- *is*: Identity Server
- *postgres*: BD for storing user data.

Secure Manager One	
Functional Description	Identity manager, api-gateway for cloud applications
Type of component	*Web Service, API-mediation service*
Interaction interface	*RESTFul API*
Component maintainer	STS
License	apache2
Delivery	Docker images
Deployment	Cloud

Table 3: SB@Cloud component Secure Manager One details.

4.1.2 Data Repository

The main data repository provides the Clinical Data Repository (CDR) supporting the H7 FHIR standard and an also standardised API REST interface, as well as the settings and infrastructure needed to provide a stable, secure, and homogenised storage of clinical and aggregated data.

Data Repository	
Functional Description	Clinical data repository based on the HAPI FHIR server
Type of component	Data repository
Interaction interface	REST API
Component maintainer	Atos
License	Apache 2.0
Delivery	Docker images
Deployment	Cloud

Table 4: SB@Cloud component Data Repository.

4.1.3 Dashboard

The Dashboard component provides a User Interface (UI) for the users of the platform.

Dashboard	
Functional Description	The application is responsible for managing the interactions between the cloud platform and the users who must use it to register patients into the system, enter clinical data, launch analytics on aggregated data and check that patients are actually using the devices in their possession to save the data within the platform.
Type of component	Web Service
Interaction interface	RESTfull API

Component maintainer	UMIL
License	Apache 2.0
Delivery	Docker image
Deployment	Cloud

Table 5: SB@Cloud component Dashboard.

4.1.4 BDA

The Big Data Analytics component (BDA) is the one in charge of providing advanced analytics over the data.

BDA	
Functional Description	Implementation of analytics, both basic and complex
Type of component	Web Service
Interaction interface	RESTful API https://app.swaggerhub.com/apis/ssiccardi/sbpopbda1/v1
Component maintainer	UMIL
License	Apache 2.0
Delivery	Docker image
Deployment	Cloud

Table 6: SB@Cloud component BDA.

4.1.5 Non-FHIR_database

The Non-FHIR database component is used by DSS as temporally storage in the analysis of data than can not be handled directly from the sb-data repository which is stored in FHIR format.

Non-FHIR database	
Functional Description	Management of raw data from sensors (analytics results, notifications, interventions)
Type of component	Web service - database
Interaction interface	RESTful API
Component maintainer	ICSS
License	MIT
Delivery	Docker image
Deployment	Cloud

Table 7: SB@Cloud component Non-FHIR Database.

4.1.6 DSS

The DSS provides recommendation related to the clinical interventions.

DSS	
Functional Description	Implementation of the DSS to generate notifications, interventions and explainability of the analytics
Type of component	Web Service, Job
Interaction interface	RESTful API
Component maintainer	ICCS
License	Apache 2.0
Delivery	Docker image
Deployment	Cloud

Table 8: SB@Cloud component DSS.

4.1.7 SB@App

The Smart Bear app retrieves data from the devices of the participant and sends them to the SB@Cloud backend. These data are transmitted in HL7 FHIR format, analysed and sent back to the SB@App to display useful information and analytics to the participant.

SB@App	
Functional Description	The SB@app manages the processes related to sensor data collection, HL7 FHIR transformation and transmission, and visualisation of the participant information.
Type of component	Smartphone -App
Interaction interface	Web service/API
Component maintainer	ATC
License	Open source
Delivery	Android application APK file
Deployment	Participant smartphone

Table 9: SB@Cloud component SB@App.

4.1.8 Garmin intermediate server

This service acts as an intermediate between the Garmin smartwatch and the SB platform.

Garmin intermediate server	
Functional Description	It integrates the Garmin Health API to retrieve data from vendor cloud.
Type of component	Web service
Interaction interface	N/A
Component maintainer	ATC
License	Apache 2.0
Delivery	Docker image
Deployment	Cloud

Table 10: Component Garmin intermediate server.

4.1.9 Notification manager

The notification manager accomplishes all the tasks related to the notification processes between the DSS to the Android application.

Notification manager	
Functional Description	It handles message delivery to SB@App
Type of component	Web service
Interaction interface	API
Component maintainer	Giorgos Giotis g.giotis@atc.gr
License	Apache 2.0
Delivery	Docker image
Deployment	Cloud

Table 11: Component Notification manager.

5 SMART BEAR components requirements for CI/CD

The CI/CD of the SMART BEAR components is based on Kubernetes. "*Kubernetes is a portable and extensible open-source tool for managing workloads and services in complex environments*[9]". One main advantage of Kubernetes is that it allows declarative configuration making in order to facilitate the automation processes. Kubernetes is a container-centric management environment initially developed by Google.

The installation and configuration of the integrated components are described in D6.7.

Each component, after their compilation, tests and packaging according to the selected technology, will have the specific configuration to generate a containerised version of the service provided.

Below is the chosen nomenclature for each container generated. Note that a single functional component can be composed of one or several containers (subcomponents). The same table is reproduced in D6.7 for clarification purposes.

Component	Subcomponent	Description
sb-data-repository	sb-data-repository	HL7 FHIR repository based on HAPI FHIR server to magane the clinical data
	mariadb	Relational DB supporting sb-data-repository
Secure_manager_one	usermanager	User manager API under /users
	am	WSO2 API Management UI
	is	WSO2 API Identity Server UI
	postgres	Relationa DB supporting Identity server
SB_dashboard	sb_dashboard	SB Dashboard
SB_BDA	Sb_bda	Big Data Analytics
Non-FHIR_database	database	
DSS	api_dss	Decision Support System
	dss_core	

Table 12: List of SB@Cloud components as containers.

[9] https://kubernetes.io/docs/concepts/overview/what-is-kubernete

6 Testing the PoP (Pilot of Pilots)

The following section describes the steps followed to test the "Pilot of Pilots".

6.1 *Methodology*

The chapter aims to define a basic methodology applied for the execution of the test for the integrated platform.

1. The first step is to identify the goals to be achieved by the platform in the current version and consequently to specify which use cases and functionalities must be demonstrated.
2. The second step is to define the list of pre-requirements, i.e. which components of the architecture must be set-up and ensure they are correctly working in order to proceed with the test of the identified uses cases.
3. The third step is to draw up the test procedures and plans and to describe in detail all the test results. It is worthy to point out that it is necessary to specify the environment where the test will be performed, the responsibilities (which partner will test which component), the list of all the test items and their related description, how the results will be tracked and a complete test plan.

6.2 *Goals and scope*

SMART BEAR Pilot of the pilots has been conceived as a way to demonstrate key functionalities of the platform as well as an interim step to assess and to refine the overall approach for the integration and the testing of the current prototype. Hence, we can summarise the PoP's goals as follows:

- Design and test integration infrastructure and methodology
- Design and testing methodology
- Test platform's key functionalities

The pilot of pilots will demonstrate the following set of functionalities

- Participant registration and devices pairing to participants
- Flows of data from devices to repository
- FHIR support semantic annotations
- Data analytics
- Mobile App
- Synergies support
- Security

In the following table we describe the overall scenario followed by the seven Use Cases included by that scenario.

Scenario	Pilot of Pilots
Actors	• Doctors • Patients • Administrators
Pre-conditions	Components enabling the main functionalities demonstrated by the Pilot of the Pilots have to be integrated in the SMART-BEAR platform and testing of basic functioning is a prerequisite. The PoP testing is focused on the integrated platform.
Main scenario	• Devices are paired to participants • Users register to the system • Data are analysed by the DSS • Recommendations are given to patients
Actors	• Doctors • Patients • Administrators
Main scenario	• Clinical Case Manager (CM) is created • CM creates patient • CM does patient's assessment • CM accepts patient • Devices are paired to participants • Data are analysed by the DSS • Interventions are generated and notified to both CM and patients Analytics are showed in the BDA

Table 13: Testing scenarios

In this scenario we are not testing the interaction with S4H as, at the moment of writing this deliverable, the work associated to that is still in progress to allow data transfer from SM4H and further elaboration by SMB. Moreover, due to the more focused scope of the Pilot of Pilots, we do not map testing cases to KPI which will be done for the testing of the overall Pilot.

In general, we point out that debugging and fine tuning of system will continue after closing this deliverable

6.3 *Testing Pre-conditions*

The following table describes how the test environment is made and the related responsibilities. Working of individual components has been tested by the responsible partner before and during the integration process, before testing the integrated system.

Component	Responsability
Security Component	STS
Dashboard	UMIL
DSS	ICCS
BDA	UMIL
Non-FHIR repository	ICCS
HAPI-FHIR Repository	ATOS
Sensors and devices	UOI
Wearables @app	ATC
Pseudo ID	STS

Table 14: Testing preconditions, component responsability

The components listed above, once integrated, will enable the different functionalities demonstrated by the Pilot of Pilots. The following table shows the functionalities to be demonstrated associated to the components enabling them.

Functionality	Components
Participant registration and devices pairing to participants	• Dashboard-GUI • Security authentication and authorisation components • Administrative database • Pseudo-IDs
Flows of data from devices to repository	• Security authentication and authorisation • Sensors and devices @home • Wearables @app • Pseudo IDs • HAPI-FHIR repository
FHIR support an semantic annotations	• Transformation • Clinical Data Repository • API
Data analytics on synthetic data and interventions	• DSS • BDA
Mobile App, view of own data and delivery of interventions	• Security authentication and authorisation components • Mobile App • DashBoard GUI • Pseudo IDs
Synergies support, data digestion and ID correlation	• Dashboard-GUI • Pseudo ID • Clinical Data Reposiroty
Security, anonymisation, access control, component and user authentication	• Pseudo IDs • Authentication and authorisation

Table 15: Testing functionality components relation

7 Test procedure and plan

7.1 *PoP_F1_TC1*

Test case Id	**PoP_F1_TC1**
Description	Creation of user, creation of patient
Use case/functionality covered	This scenario covers the creation of a Clinical Case Manager (CM). • Registration of a clinical case manager • Login as CM
Expected result	• CM user created and login enabled
Actual result	• Clinical Case Manager created.
Comments	• During first round of testing, the registration procedure showed an error when the user clicked on the confirmation email, even though the registration worked anyway. On the second round of test that had been fixed.
Feedback (if any)	Registration procedure looks easy to perform for the user
Test date	10th May 2021 12th May 2021 17th May 2021

Table 16: Test PoP_F1_TC1

7.2 *PoP_F1_TC2*

Test case Id	**PoP_F1_TC2**
Description	Creation of patient
Use case/functionality covered	In this scenario, the CM creates a patient and accepts him if pre-set criteria have been met. • Login in the Dashboard as CM • Create Patient • Add an assessment for that patient • Answer questions on exclusion criteria • Accept patient if criteria have been matched
Expected result	• Patient created • Assessment added • Patient accepted if criteria have been met

Actual result	• Patient created • Assessment added • Patient created in FHIR Repository • Patient accepted
Comments	• During first round of test, upon creation of the user in the Dashboard, it had to be created in the FHIR Repository manually. By the second round of test, that has been fixed • During first round of test, upon accepting the patient the patient's status label had to be changed to Accepted manually. By the second round that had been fixed.
Feedback (if any)	Patient's creation assumes some knowledge of the platform and of the process but, considering it is performed by internal users, that should not be an issue; it should easy to teach to operators.
Test date	10th May 2021 12th May 2021 17th-May-2021

Table 17: Test PoP_F1_TC2

7.3 *PoP_F2_TC1*

Test case Id	**PoP_F2_TC1**
Description	Creation of data
Use case/functionality covered	In this scenario, patient's medical history is completed. • Login in the Dashboard as CM • Select Patient • Fill in medical history of created patient
Expected result	• Medical history data added to the repository
Actual result	• Medical history data are added correctly
Comments	Some refining activities are ongoing. • First lines of the General Information have to be reorganised (e.g., Diabetes should be an alternative not compulsory) • Only a few data are actually saved (e.g., ECG data are not) because the mapping between data in the medical history and the FHIR repository has to be clarified

	• Currently working in the assessment of accepted patients: medical history and cardiovascular tables. In future, it will be all and will be predetermined based on patient.
Feedback (if any)	For future testing, we suggest to get a set o realistic data from clinicians, so to perform a test that is as close as possible to real life scenario
Test date	10th May 2021 12th May 2021 17th-May-2021

Table 18: Test PoP_F2_TC1

7.4 *PoP_F2_TC2*

Test case Id	**PoP_F2_TC2**
Description	Pairing of devices (mobile phone)
Use case/functionality covered	In this scenario, a mobile device (a smartphone for the sake of this test) is associated to the patient. • Login in the Dashboard as CM • Go in Device Management • Add device • In the mobile add IMEI of Phone • In the Mobile add the Pseudo ID of user
Expected result	• Mobile device added
Actual result	• Mobile device added and paired to the user
Comments	• SMARTPHONE is the Device required for collection of data. • Data on mobile for paring now has to be done manually.
Feedback (if any)	Future improvements should be in the direction of making more user friendly the mapping. Maybe using QR codes to make the process of entering pairing codes easier
Test date	10th May 2021 12th May 2021 17th May 2021

Table 19: Test PoP_F2_TC2

7.5 *PoP_F3_TC1*

Test case Id	**PoP_F3_TC1**
Description	Send data from mobile to FHIR Repository
Use case/functionality covered	In this scenario, data from the smartphone are sent to the FHIR Repository. • Data are sent from the mobile to the FHIR repository
Expected result	• Data are transmitted from Smartphone to FHIR Repository
Actual result	• Searching on the FHIR repository, we see that data have arrived and available
Comments	The FHIR
Feedback (if any)	FHIR Repository must be checked with more data for evaluating performance.
Test date	10th May 2021 12th May 2021 17th May 2021

Table 20: Test PoP_F3_TC1

7.6 *PoP_F4_TC1*

Test case Id	**PoP_F4_TC1**
Description	Generation of an intervention for a specific user and his CM, based on generated data and pre-set criteria.
Use case/functionality covered	In this scenario, the CM set criteria for generation of intervention and the system triggers the intervention based on data available. • The clinical CM manually enters the threshold that will generate the intervention • The device reads parameters and send them to the system • If the system identifies a threshold violation, the system generates the intervention • Generation are sent to both CM and patient
Expected result	• Threshold generated • Data sent from device to system

	• Intervention generated when threshold is exceeded
Actual result	• Interventions are generated.
Comments	• Criteria for generation of the intervention are set in the Dashboard for each specific user (e.g. blood pressure thresholds). • When the systems check new data and reads parameters outside the pre-set thresholds, it creates an alert that is shown in the "interventions" tab of the CM associated to that patient. • Intervention messages must be generated in two languages, English and Portuguese. For this testing, they are generated in English.
Feedback (if any)	NA
Test date	10th May 2021 12th May 2021 17th May 2021

Table 21: Test PoP_F4_TC1

7.7 *PoP_F4_TC1*

Test case Id	**PoP_F4_TC1**
Description	Showing analytics in the BDA
0Use case/functionality covered	• Operator connects to the BDA interface • User is selected • Analytics are showed
Expected result	• When you check the data, you expect to see the data just transmitted from the Mobile
Actual result	• The BDA correctly shows the data just received
Comments	• In the testing environment there are a lot of data generated during various tests conducted on both single components and the platform. To make future test easier, data from previous test should be removed.
Feedback (if any)	N/A
Test date	10th May 2021

	12th May 2021 17th May 2021

Table 22: Test PoP_F4_TC1

8 Concluding Remarks

The integration process started setting up the GitLab repository and investigating the CI/CD tools and framework that better suited the project necessities. The main objective was to have a first prototype ready for demonstration and testing as soon as possible, i.e. early May 2021. As the setup of CI/CD infrastructure and framework needs a huge amount of effort and time for setting up the Kubernetes, an alternative plan based on virtualisation was proposed.

This alternative plan consists of integrating all the components manually using Docker compose in a Virtual Machine. This alternative plan saves some time allowing a faster integration of the SMART BEAR components. Still, the framework for CI/CD based on Kubernetes will be provided for the PoP.

Kubernetes set up initially is cumbersome, and we faced unexpected problems in the infrastructure implementation, but they are mostly solved, and Kubernetes will provide easier integration and deployment. It is important to point out that the alternative integration plan does not preclude the continuation of the CI/CD approach, as CI/CD offers a bunch of advantages once the Kubernetes infrastructure is working properly.