



SMART BEAR

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"

D5.2 – Report on Continuous Security Assurance & Privacy by design - enabling mechanisms v1

Due date of deliverable: 31/01/2021
Actual submission date: 2/01/2021

Grant agreement number: 857172
Start date of project: 01/09/2019
Revision

Lead contractor: CNR
Duration: 48 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	



D5.2 – Report on Continuous Security Assurance & Privacy by design - enabling mechanisms v1

Editors

Ioannis Basdekis (STS)
Konstantina Koloutsou (STS)

Contributors

Othonas Soutatos (STS)
Konstantin Pozdniakov (CITY)
Fady Copty (IBM)
George Zissis (ATC)
Giorgos Giotis (ATC)

Reviewers

Manolis Michalodimitrakis (FORTH)
Maria Pateraki (FORTH)
Christos Kloukinas (CITY)



Executive Summary

The aim of the SMART BEAR platform is to integrate heterogeneous sensors, assistive medical and mobile devices to enable the continuous data collection from the everyday life of the elderly, which will be analysed to obtain the evidence needed in order to offer personalised interventions promoting their healthy and independent living. The platform will also be connected to hospitals and other health care service systems to obtain data of the end-users (e.g., medical history) that will need to be considered in making decisions for interventions. SMART BEAR will leverage big data analytics and learning capabilities, allowing for large scale analysis of the above-mentioned collected data, to generate the evidence required for making decisions about personalised interventions. Privacy-preserving and secure by design data handling mechanisms, covering data at rest, in processing, and in transit, will cover comprehensively all the components and connections utilized by the SMART BEAR platform. This deliverable covers the development, modelling, configuration, and execution of mechanisms to guarantee the security and privacy of data held in the SB platform (SB@Cloud). These mechanisms are built on top of Open Source software and have been selected in order to maximize the prevention of possible security and privacy incidents. Initially, we provide the platform security and privacy requirements, afterward, we describe the architecture of the deliverable, and finally for each of the mechanism, we provide a description of its functionality. Lastly, we oversee how it can be used in the SMART BEAR platform and finally we provide a roadmap of the deployment.



Contents

Executive Summary	3
List of acronyms	5
List of tables.....	7
List of figures.....	8
1 Introduction	10
1.1 Purpose of the document.....	10
2 Smart Bear Security Design.....	11
2.1 Security Building Blocks	11
2.2 Security Areas.....	11
2.2.1 Device security	11
2.2.2 Connectivity security.....	11
2.2.3 Cloud security.....	11
2.3 Authentication, Authorisation.....	12
2.3.1 OAuth 2.0	12
2.3.2 OpenID Connect.....	13
2.4 Access Control	14
2.4.1 Attribute Based Access Control	14
2.4.2 Role Based Access Control	14
2.4.3 RBAC vs. ABAC	14
3 Adhering to Privacy by Design Principles.....	16
3.1.1 Introduction	16
3.1.2 Privacy by Design in the context of Smart Bear	16
3.1.3 Functionality description.....	17
3.2 SB@SecurityComponent: Core Component Specification	26
4 Security Infrastructure	40
4.1 Architecture	40
4.2 Security Component	41
4.2.1 Cloud Infrastructure Security	41
4.2.2 Mobile Security	42
4.2.3 Secure Manager One	42
4.3 Roadmap	45
5 Conclusion	46
6 References.....	47



List of acronyms

ABAC	Attribute Based Access Control
AES-256	Advanced Encryption Standard algorithm
AMQP	Advanced Message Queuing Protocol
API	Application programming interface
APIM	API-Manager
BDA	Big Data Analytics
CIAM	Customer Identity and Access Management
CVE	Common vulnerabilities and exposures
DoS	Denial of Service
DPO	Data protection officer
EHRs	Electronic Health Record systems
GDPR	General Data Protection Regulation
HA	Hearing Aids
HB	HOLOBALANCE project
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
IAM	Identity Access Management
IETF	Internet Engineering Task Force
IS	Identity server
JSON	JavaScript Object Notation
JWT	JSON Web Token
MQTT	Message Queuing Telemetry Transport
OAuth	open standard for access delegation
OTS	off-the-shelf
PET	Privacy Enhancing Technologies
PII	Personal Identifiable Information
RBAC	Role Based Access Control
REST API	(also known as RESTful API) is an application programming interface (API or web API) that conforms to the constraints of REST architectural style and allows for interaction with RESTful web services
S4H	Smart4Health project
SAML 2.0	Security Assertion Mark-up Language 2.0
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
SCIM	System for Cross-domain Identity Management specification
SDK	Software Development Kit
TLS	Transport Layer Security
UML	Unified Modelling Language
X.509	X.509 is a standard defining the format of public key certificates
XACML	eXtensible Access Control Mark-up Language
XML	eXtensible Mark-up Language



List of tables

Table 1: Requirements and corresponding actions	22
Table 2: Login: REST service definition	28
Table 3: Logout: REST service definition	29
Table 4: End-user management: REST service definitions	30
Table 2: My profile: REST service definitions	33
Table 2: GDPR request management: REST service definitions	35



List of figures

Figure 1: OAuth 2 flow	13
Figure 2: Smart Bear Architecture (presented in D2.2): Color-coded components indicate presence of personal (red) or pseudonymised (blue) data	17
Figure 3: Security Component replaces different IDs to a SB-generated one (Pseudo-Id2) by which data record is stored in SB@Repository	19
Figure 4: Colour-coded graphical representation of the 'flow' of data.....	19
Figure 5: Sequence diagram: end-user registration.....	27
Figure 6: Design mock-up: end-user registration	28
Figure 7: Sequence diagram: end-user’s login.....	28
Figure 8: Design mock-up: end-user’s login.....	29
Figure 9: Sequence diagram: System administrator activates end-user’s registration request (unlocks a previously validated account)	29
Figure 10: Sequence diagram: System administrator revokes end-user’s access (locks/suspends access)	30
Figure 11: Design mock-up: Activate an account: Step 1/4: System administrator selects the ID of a newly registered end-user	31
Figure 12: Design mock-up: Activate an account: Step 2/4: System administrator triggers a new action for the specific account	31
Figure 13: Design mock-up: Activate an account: Step 3a/4: System administrator alters account status	31
Figure 14: Design mock-up: Activate an account: Step 3b/4: System administrator changed account status to “active”	32
Figure 15: Design mock-up: Activate an account: Step 4/4: System administrator reviews updated account status	32
Figure 16: Sequence diagram: End-user updates his/her personal details	33
Figure 17: End-user’s (e.g., Clinical Case manager) profile information.....	33
Figure 18: Editing end-user’s profile information	34
Figure 19: End-user’s (e.g., Clinical Case manager) updated profile information.....	34
Figure 20: Sequence diagram: End-user creates a GDPR request via his/her SB@App.....	35
Figure 21: System Administrator selects to view GDPR request details.....	36
Figure 22: System Administrator views request details and triggers the assignment of actions to specific end-users (SB System Administrators).....	36
Figure 23: Assign actions (in respect to a GDPR request): System administrator may assign one or more actions to designated end-users (SB System Administrators)	37
Figure 24: GDPR actions been assigned to a System administrator.....	37
Figure 25: GDPR assigned actions: System administrator may edit the detail of an assigned to him/her action.....	38
Figure 26: Edit GDPR assigned action: System administrator insert associated to the action log file and description.....	38
Figure 27: GDPR assigned actions: Updated action details	38
Figure 28: Updated GDPR request details	39
Figure 29: Auditor views GDPR request details and associated log files	39



Figure 30: Clinical Case Manager can view all GDPR requests created by him/her 39

Figure 31: Smart Bear Architectural Overview 40

Figure 32: Security component interaction 41

Figure 33: Message Flow 43

Figure 34: Authentication and Authorisation sequence 44



1 Introduction

1.1 Purpose of the document

WP5 is designing and developing mechanisms to ensure the security and privacy of the data held in the Smart Bear (SB) platform and the integrity of the platform itself. The security properties focus on the integrity, confidentiality and availability of data at rest, in transit and in processing for data flows (KPI-15) acquired by the interactions with external devices and systems (such as hospital medical systems, smart house sensors and IoT devices usage data via smartphone application, and other). The implemented mechanisms comply with the privacy and security-by design approach, the notion by which security and privacy measures and enhancing technologies (PETs) are being embedded directly into the design of a system. On this axis, and given the legal obligations imposed by the General Data Protection Regulation (GDPR), data minimization, pseudonymisation, transparency in processing of personal data, and other appropriate technical (and organisational) measures have been taken into account at early stage of the platform design to ensure that GDPR requirements are met. This way it is possible to preserve the anonymity of big data (i.e., usage, anonymized medical history, analytics, interventions stored in the Repository), the confidentiality of any Personal Identifiable Information (PII) (i.e., IDs association records, and Dashboard end-users profile information stored in the Security Component), the privacy and the integrity of data (KPI-14, KPI-16) according to state-of-the-art guidelines (e.g., encryption guidelines of NIST (NIST, 2020), role-based and application-level security access control, Secure Sockets Layer, Denial-of-Service Prevention (KPI-17, KPI-18, SmartBear Key Performance indicators are presented in Table 1 of Grand Agreement).

Data protection is a critical issue for SB platform and for this reason data minimisation, authentication and other security and privacy aspects are materialized with an in-between (REST API, SB@Cloud components, SB smartphone application, and external systems) component. Smart Bear Security Component (SB@SecurityComponent) provides mechanisms that perform pseudonymisation and IDs associations, used for the authentication and authorisation of SB@Cloud RESTful API to protect the transmission of any (sensitive or not) data, and it is also utilised for the management of privacy-related requests to demonstrate compliance with the GDPR. In this respect, the SB@App (technical aspects of which are presented in D3.2) that periodically ingests data from the everyday life of the elderly (i.e., heterogeneous sensors, assistive medical and mobile devices) to the SB@Repository, has data protection mechanisms in place to provide security both while data are flowing from/to the SB@App (data in transit) as well as while data stays temporarily in the smartphone's internal database (data at rest).

This deliverable focuses on the requirements of the SB@SecurityComponent, and encompasses the implemented security and privacy mechanisms, through the prism of this component's use. The initial privacy requirements (defined in D2.1) were revisited in the light of partners feedback received during the past six months, and comments and suggestions made in the context of the review meeting (in ref. Ares (2020)2400656 of 06.05.2020), by the EC and project reviewers (Recommendation 8). As a result, new requirements were added.

The structure of the deliverable is the following – after the executive summary of the deliverable (“Executive summary”), section 1 gives the general overview, and introduces the goal and structure of the document (“Introduction”). Then section 2 provides more details on the privacy and security provisions of the SB architecture that are in accordance to the Privacy by Design principle, including core mechanisms for preserving the anonymity of the data. Section 3 describes the SB Security Design targets. Lastly, section 4 describes the software components implementing the proposed design approach and concludes with our development roadmap.



2 Smart Bear Security Design

2.1 Security Building Blocks

Confidentiality, Integrity, Availability are the three main cybersecurity principles of any security control (Neumann et al. 1977). Confidentiality is the property where unless users, processes or devices are authorized to access, information is not disclosed. Integrity is the property whereby information has not been modified or destroyed in an unauthorized manner. Availability is the property of being accessible. Each of these three principles involve relevant protection mechanisms, which are described below, as they are derived from the various standards such as Common Criteria Evaluation Methodology (CEM) (ISO/IEC 15408 1996-2018) and the Open Source Security Testing Methodology Manual (ISECOM 1988-2918) and related research efforts (Hatzivasilis et al. 2016).

- Confidentiality: To achieve confidentiality authentication can be utilised so that we can verify the identity of the user;
- Integrity: Secure authorisation mechanisms and access controls should be used;
- Availability: In the context of Smart Bear, service availability can be achieved by utilising a cloud infrastructure;

The guidelines for secure IoT development suggested by large computer and software vendors (Betts et al. 2018) and surveys (Lin et al. 2017; Andrea et al. 2015; Bekara 2014), include the following three security areas: i) Device security ii) Connectivity security iii) Cloud security.

2.2 Security Areas

2.2.1 Device security

Device security implements various ways of authenticating a device and also securing the data storage of this device. To satisfy these dependencies two main components are required:

- A unique identity key or security token for each device;
- An on-device X.509 certificate and private key;

In a typical operation the ID token is used for authentication for each message transmitted. The certificate file and private key is used to secure the communication between devices by validating the transmitted messages and to encrypt the data at rest.

2.2.2 Connectivity security

Data confidentiality and integrity could be compromised as soon as a device is connected over the internet. All data that are transmitted between devices and also between the devices and the cloud must be encrypted. Seamless communication is supported by relevant protocols, such as the Advanced Message Queuing Protocol (AMQP), Message Queuing Telemetry Transport (MQTT), and Hyper Text Transfer Protocol (HTTP) (Hatzivasilis et al, 2018), and is safeguarded by the security mechanisms that are implemented by each one of them such as TLS for HTTP (HTTPS).

2.2.3 Cloud security

Cloud computing suffers from several security issues that overlooking them may lead to catastrophic consequences. As seen on (Jansen and Grance, 2011; Fernandes et al. 2014) the main security problems can be categorized as below.

- Shared technologies: As seen in (Kocher et al. 2018; Lipp et al. 2018) attackers can exploit shared memory to gain access to unauthorized content such as encryption keys.
- Data breach: Personal data containing sensitive information such as credit card information can be lost or worse can be leaked.



- Account/service hijacking: If login credentials are lost or leaked, this can lead to attackers gaining access to critical areas of services and could potentially compromise confidentiality, integrity and availability.
- Denial of Service (DoS): As seen in (Deshmukh & Devadkar, 2015) cloud infrastructure mechanisms cope with DoS attacks by providing scaling up of its resources, but this firstly provides the attacker with more resources to achieve his malicious goals and secondly this type of attack can have monetary impacts.
- Malicious insiders: A cloud company's employee can leverage his position to access sensitive information of the hosted services.

2.3 Authentication, Authorisation

2.3.1 OAuth 2.0

OAuth 2.0 is a widely adopted¹ protocol for authorisation. This specification and its extensions are being developed within the IETF OAuth Working Group. The OAuth 2.0 authorisation framework enables a third-party application to obtain limited access to an HTTP service, either on behalf of a resource owner by orchestrating an approval interaction between the resource owner and the HTTP service, or by allowing the third-party application to obtain access on its own behalf.

In OAuth2 jargon there are four (4) different roles specified:

1. Resource owner: An entity capable of granting access to a protected resource. When the resource owner is a person, it is referred to as an end-user.
2. Resource server: The server hosting the protected resources, capable of accepting and responding to protected resource requests using access tokens.
3. Client: An application making protected resource requests on behalf of the resource owner and with its authorisation.
4. Authorisation server: The server issuing access tokens to the client after successfully authenticating the resource owner and obtaining authorisation.

Moreover, it should also be mentioned that OAuth 2 defines 4 different Authorisation Grants as well as an extensibility mechanism for defining additional types which is out of scope of this deliverable. An authorisation grant is a credential representing the resource owner's authorisation (to access its protected resources) used by the client to obtain an access token:

- authorisation code: The authorisation code is obtained by using an authorisation server as an intermediary between the client and resource owner. Instead of requesting authorisation directly from the resource owner, the client directs the resource owner to an authorisation server which in turn directs the resource owner back to the client with the authorisation code;
- implicit: In the implicit flow, instead of issuing the client an authorisation code, the client is issued an access token directly. The grant type is implicit, as no intermediate credentials (such as an authorisation code) are issued (and later used to obtain an access token).
- resource owner password credentials: The resource owner password credentials (i.e., username and password) can be used directly as an authorisation grant to obtain an access token;
- client credentials: The client credentials (or other forms of client authentication) can be used as an authorisation grant when the authorisation scope is limited to the protected resources under the control of the client, or to protected resources previously arranged with the authorisation server;

¹ https://en.wikipedia.org/wiki/OAuth#OAuth_2.0

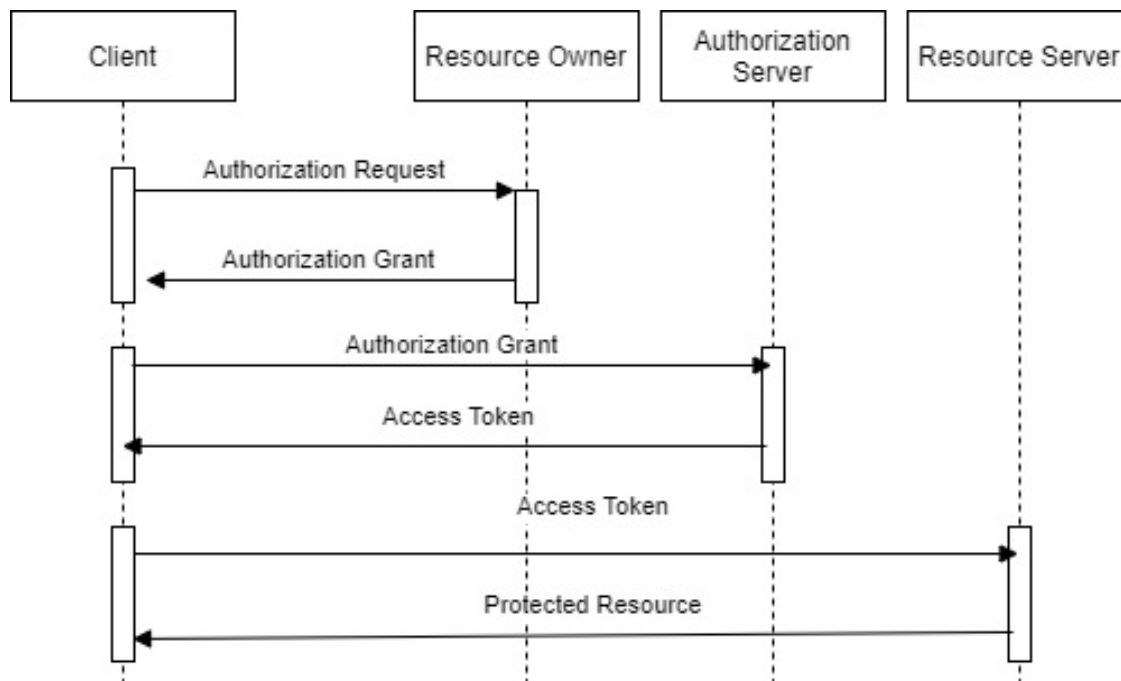


Figure 1: OAuth 2 flow

An abstract OAuth2 flow is depicted in Figure 1. This flow describes the interaction between the four (4) different roles and includes the following six (6) steps:

1. The client requests authorisation from the resource owner.
2. The client receives an authorisation grant, which is a credential representing the resource owner's authorisation, expressed using one of the four authorisation grant types.
3. The client requests an access token by authenticating with the authorisation server and presenting the authorisation grant.
4. The authorisation server authenticates the client and validates the authorisation grant, and if valid, issues an access token.
5. The client requests the protected resource from the resource server and authenticates by presenting the access token.
6. The resource server validates the access token, and if valid, serves the request.

2.3.2 OpenID Connect

The OAuth 2.0 Framework describes overarching patterns for granting authorisation but does not define how to perform authentication. OpenID Connect is an interoperable authentication protocol based on the OAuth 2.0 family of specifications. It uses straightforward REST/JSON message flows. OpenID Connect lets developers authenticate their users across websites and apps without having to own and manage password files. OpenID Connect allows for clients of all types, including browser-based JavaScript and native mobile apps, to launch sign-in flows and receive verifiable assertions about the identity of signed-in users. OpenID Connect uses the ID token data structure that enable end-users to be authenticated. The ID token has a JSON Web Token (JWT) format, which is a standard way to generate authentication tokens. The JWT contains various user information which are called claims. Also, it contains information about the validity of the token, such as issue datetime, expiry period etc. The token is normally signed by the token issuer with the issuer's public key to be easily verified using PKI. So, in the context of Smart Bear, OpenID connect will be used.



2.4 Access Control

Authentication and authorisation described above present the basic logic of how an end-user can login into the system, as well as on how we can get the user info. Nevertheless, not only we should have that info but also, we should define an access control mechanism of how we apply this information.

2.4.1 Attribute Based Access Control

Attribute Based Access Control (ABAC): A logical access control methodology where authorisation to perform a set of operations is determined by evaluating attributes associated with the subject, object, requested operations, and, in some cases, environment conditions against policy, rules, or relationships that describe the allowable operations for a given set of attribute:

- Attributes are characteristics that define specific aspects of the subject, object, environment conditions, and/or requested actions that are predefined and preassigned by an authority.
- A subject is an active entity (generally an individual, process, or device) that causes information to flow among objects or changes the system state.
- An object is a passive information system-related entity containing or receiving information.
- An operation is the execution of a function at the request of a subject upon an object.
- Policy is the representation of rules or relationships that define the set of allowable operations a subject may perform upon an object in permitted environment conditions.

2.4.2 Role Based Access Control

Role-based access control (RBAC) refers to the idea of assigning permissions to users based on their role within an organization. It is less prone to error because of its simplicity and manageability instead of assigning each user permissions. Based on user requirements, users are grouped and then each group is assigned a role. Afterward, each user is assigned on or more roles and each role is assigned with one or more permissions.

Benefits of RBAC

- create systematic, repeatable assignment of permissions
- easily audit user privileges and correct identified issues
- quickly add and change roles, as well as implement them across APIs
- cut down on the potential for error when assigning user permissions
- integrate third-party users by giving them pre-defined roles
- more effectively comply with regulatory and statutory requirements for confidentiality and privacy

In the possibility that there are overlapping roles, the permissions that the user has is the union of the permissions of each role the user has, since RBAC is an additive model.

2.4.3 RBAC vs. ABAC

	RBAC	ABAC
Simplicity/Complexity	Easy to establish roles and permissions.	Requires more processing power and time.



In practice	Most popular: is most commonly implemented in small and medium-sized enterprises (simple workflows, limited number of roles, simple hierarchy).	Assignment of a business rule of any complexity can be done but with high cost.
Limitations/Cons	Can't set up a rule using parameters that are unknown to the system. Permissions can be assigned only to roles. Restrict access to certain actions in your system but not to certain data.	Hard to configure due to the way policies must be specified and maintained. Difficult to perform a before the fact audit and determine the permissions available to a specific user. Difficult to determine risk exposure for any given employee position.

Based on the above criteria, since RBAC is simpler to implement, less prone to administrative errors and suits Smart Bear needs, it is decided to follow this access control model. Below is the initial set of roles that are defined for SMART-BEAR and will be used in RBAC.

1. System Administrator (SA)
2. Patient (P)
3. Clinical Case Manager (CCM)
4. Caregiver (C)
5. Data Scientist (DS)
6. Policy maker (PM)
7. Auditor (A)



3 Adhering to Privacy by Design Principles

3.1.1 Introduction

The Smart Bear (SB) project is motivated by the need for personal data management and services that comply with the General Data Protection Regulation (GDPR, 2016), which has already been imprinted during the requirements stage (in D2.1, and in particular section 6). GDPR grants the industry the ability to create knowledge derived from (personal or not) data analysis, provided the data subjects are given control of knowing and administering who and how their data are being processed in a verifiable way. Adhering to this, in addition to administrative procedures in place (D2.1, section 6.5 “Core obligations under the GDPR”) to ensure that personal data are processed fairly and transparently, the “Privacy by design” principle has been embedded into the architecture of SB@Cloud (in D2.2), while specific features (e.g., GDPR rights, privacy audits) are supported by the SB@SecurityComponent. This SB@Cloud component consists of the following two sub-components:

1. the Secure Manager One and
2. the Secure User Data Storage

Interactions between components via REST services, as well as authentication (i.e., SB@Dashboard, SB@App, interoperable systems) are secured by the Secure Manager One, while user account information is held in the Secure User Data Storage, a separated data repository in which personal data (i.e., names, surnames, individual profile data, usernames and passwords of all registered SB end-users) and IDs associations that may lead to participants identification (i.e., Pseudo-Id1 and Pseudo-Id2 associations of all study participants) reside in an encrypted fashion. Access to encrypted data is only provided once the end-user has been successfully authenticated, and has a role granted with this access. These sub-components guarantee the appropriate level of security, compliant with the GDPR rules.

As a fundamental requirement, SB@Cloud was designed in such a way that the full length of the collected or produced Big data (i.e., usage data, medical data, resulted analytics and interventions) should be kept in a fully anonymised fashion, while a small subset consisting of IDs and Personal Identifiable Information (PII) as far as it concerns participants (SB study patients), and login-profile information of the SB end-users, stored with high security safeguards (i.e., encrypted, logged and limited access to specific role(s)). In particular, participants IDs associations records to be used by semi-automated processes in accordance with GDPR rights, and with the involvement of responsible-for-treatment clinicians under specific conditions (e.g., medical protocol, GDRP rights, privacy audits), might lead to a patient identification. In this respect, SB@SecurityComponent allows via configurable rules (RBAC) REST access to records of Ids associations, to ensure that SB organisational measures are in place to comply with the GDPR.

This section lays out the meaning of the underpinning Privacy by Design principle by which components forming the SB@Cloud were accounted for, having as main objective to explain and document the under-implementation privacy aspects of the Security Component.

3.1.2 Privacy by Design in the context of Smart Bear

(ENISA, 2018) provides a technical definition of pseudonymisation, which entails “*the process of de-associating a data subject's identity from the personal data being processed for that data subject*”. Pseudonymisation is defined in detail in (GDPR art. 4(5), 2016) as: “*‘pseudonymisation’ means the processing of personal data in such a manner that the personal data can no longer be attributed to a specific data subject without the use of additional information, provided that such additional information is kept separately and is subject to technical and organisational measures to ensure that the personal data are not attributed to an identified or identifiable natural person;*”.

In practice, the pseudonymisation process ensures that a record after-the-fact can no longer be attributed to a specific data subject, without having previously been associated with additional

information. Consequently, as long as this additional key information that may lead to subject’s identification is kept separately, and its usage is subject to technical and organisational measures, the original data set cannot lead to subject’s identification by any means (provided for instance that one has previously removed all direct identifiers and other not essential indirect identifiers with high disclosure risk, or unusual unique characteristics have been merged into data groups). Still this small critical dataset, is the one that allows Data Controllers to meet specific GDPR obligations. Subsequently, its existence (during SB project’s lifecycle) allows the exercise of all GDPR Individual rights (such as subject’s request as to who viewed his/her pseudonymised data aka ‘Right to be informed’) or the Data controller’s obligation to keep records, while its absence will convert all SB big data collected and produced to a fully anonymised dataset and as such it can be used under specific conditions (‘appropriate safeguards’) even beyond the duration of the SB project (‘storage limitation’). Notably, privacy audits to be conducted (Task 5.2) will assert these claims.

3.1.3 Functionality description

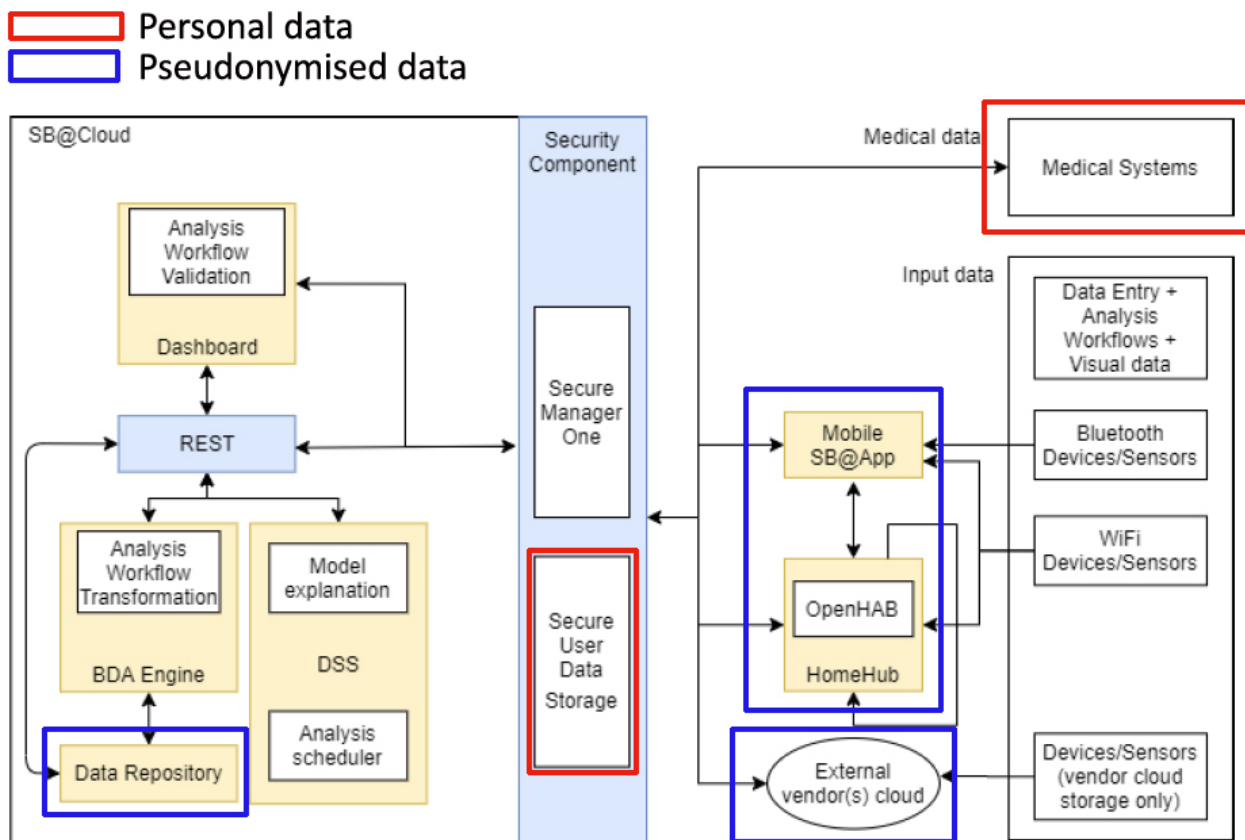


Figure 2: Smart Bear Architecture (presented in D2.2): Color-coded components indicate presence of personal (red) or pseudonymised (blue) data

In the particular context of SB (Figure 2), anonymised data stored in SB@Repository are subjected to different types of analysis, including statistical analysis techniques (e.g., descriptive statistics, statistical testing and inferencing) and data mining techniques (e.g., clustering, prediction), to obtain the evidence needed in order to offer personalised interventions promoting their healthy and independent living. The SB@Cloud, by virtue of its design, supports privacy. For stored data, PII for the subject of the data is masked or removed from it altogether. Regarding the type of personal information contained in transmitted data, there are four (4) different origins of data collected and associated types: a) data arriving from EHRs or systems of the synergetic projects S4H and HB (e.g., a



portion of personal health record stored in Medical systems of Pilots), b) IoTs usage data transmitted via the SB@HomeHub to the SB@App along with questionnaires' answers stored in smartphone's local storage later to be transmitted to SB@Cloud, c) data collected via the SB@Dashboard (e.g., medical information typed manually during the first visit), and d) aggregated data from a Vendor's cloud (case of Hearing Aids). Ids management and Personal Identifiable Information removal techniques used were introduced in (Basdekis et al, 2019) and properly enhanced to meet SB needs.

A. Ids management

- For data arriving from EHRs (i.e., portion of personal health record stored in Medical systems of Pilots), records will contain an Id associated with a unique participant, as well as potential PIIIs. To those, Ids will be replaced to a SB-generated Id (Pseudo-Id2), while PIIIs will be removed (Figure 3).
- Each pseudonymised usage data record transmitted by the SB@App is marked with a non-identifiable External pseudo Id (Pseudo-Id1) utilized for associating and pairing all devices handed over to a participant. The way this external pseudo Id is stored in smartphone's local storage is such that it cannot be associated with the participant's smartphone in case of a loss, since no one may identify, directly or indirectly a subject. As previously, Ids contained in REST JSON data will be replaced by a SB-generated Id (Pseudo-Id2) prior to those stored in SB@Repository.
- Aggregated data from External Vendor's cloud (case of Hearing aids) will be associated with the same Pseudo-Id1, later to be replaced via the SB@SecurityComponent mechanism.
- Exchangeable European Electronic Health Record (EHR) data to be collected in the context of Smart4Health (S4H) and HOLOBALANCE (HB). Associations of those Ids to the SmartBear-generated Pseudo-ID (Pseudo-Id1) and the one to be used later on (Pseudo-Id2) will be stored in SB@SecurityComponent.
- The Pseudo-Id1 that was provided by SB@Cloud will be used to configure all the devices (e.g., smart watch, sensors, and other wearables) that will be given to the particular patient, before handing them over to him/her. Pilot EHR will maintain the association between the real id of the patient and Pseudo-Id1, as whenever it will be necessary in the future to send data to SB@Cloud about the particular patient to do so under Pseudo-Id1 (Figure 4, Data Flow Type 1). Data flows of type 1 will thus be pseudonymised, as the real ID of the patient will not be conveyed to SB@Cloud. Similarly, when any of the devices that has been given to the particular patient sends data to the SB Cloud, it will transmit the relevant patient record under Pseudo-Id1 (Figure 4, Data Flow Type 2). Data flows of type 2 will thus be also pseudo anonymised, as the real ID of the patient will not be conveyed to SB Cloud. When the SB@Cloud receives data through the data flows of type 1 or type 2, it will not store the data under Pseudo-Id1. Instead, it will first replace Pseudo-Id1 with Pseudo-Id2 and then it will store the relevant data record. Thus, the data in the SB Cloud data store will be pseudo anonymised at rest.

The Id replacement process is performed by the Secure Manager One, during every REST trigger that transmits data into the SB@Repository (i.e., SB@App, "Synergies integrated" services, Push/Pull Vendor's API).

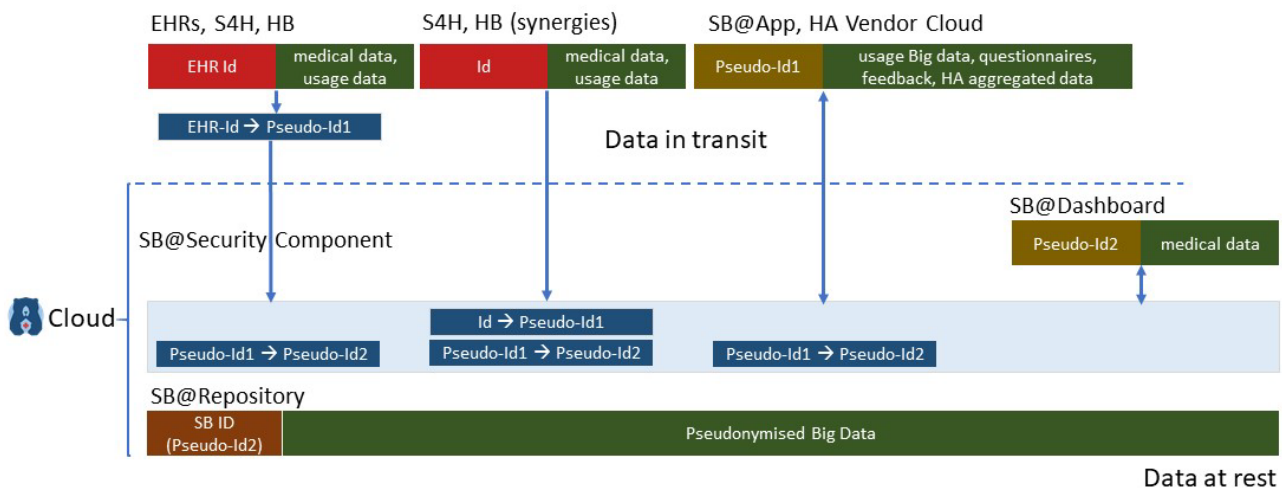


Figure 3: Security Component replaces different IDs to a SB-generated one (Pseudo-Id2) by which data record is stored in SB@Repository

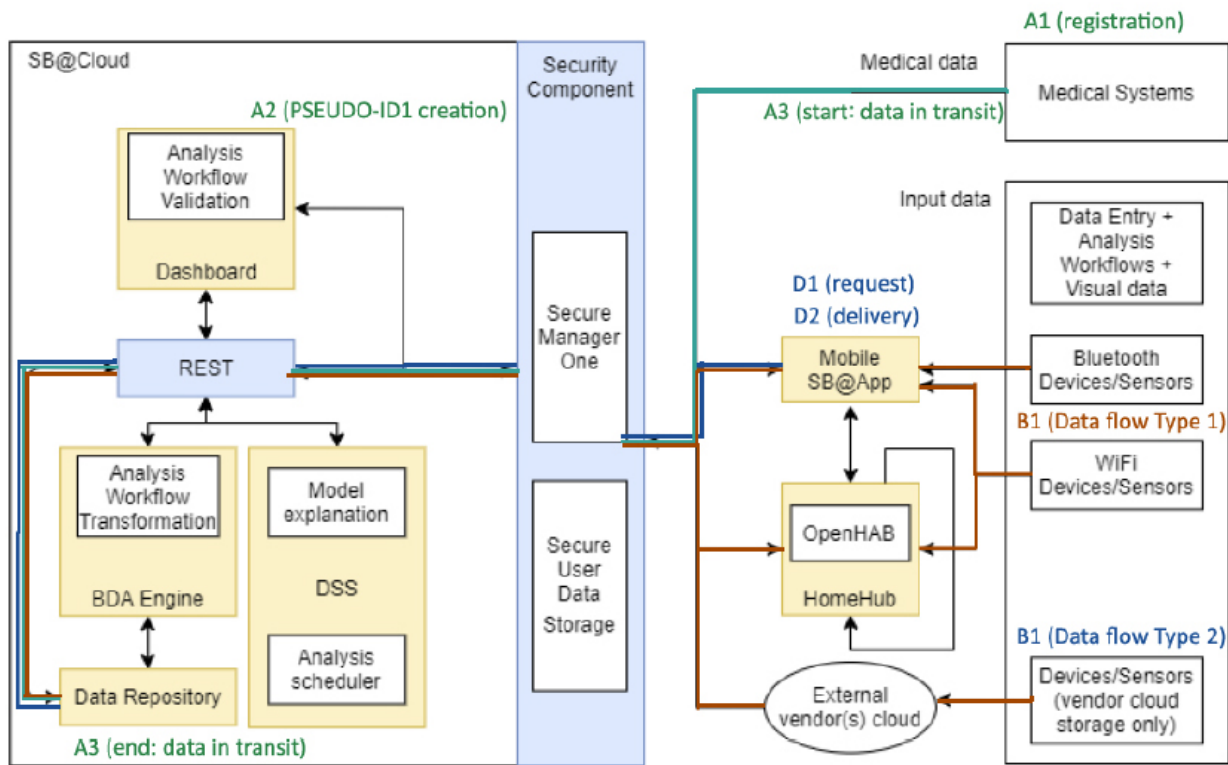


Figure 4: Colour-coded graphical representation of the 'flow' of data

B. Personal Identifiable Information

- Dates and timestamps, for instance, either referred to patients (e.g., birthdate) or to operations in hospitals (e.g., hospitalization dates) are PIIs. Location information are also notoriously PIIs, too, such as place of residence and work, travelling locations, and so forth.
 - pseudonymisation of Dates: this process removes actual dates as these could potentially lead to the identification of a specific patient (e.g., in cases where the data of only one SB patient have been recorded in EHS on a specific date). Actual dates are replaced by the date of the Sunday that follows it. For example, if the value of a date column in an EHS table is "19/9/2017", this date will be replaced by "24/9/2017". The later date will



- be used for all other records in EHS, which have an original date falling in the period from 18/9/2017 to 24/9/2017;
- pseudonymisation of Timestamps: this process removes actual timestamps as these could potentially lead to the identification of a specific patient (e.g., in cases where the data of only one SB patient have been recorded in EHS on a specific date). Actual timestamps are replaced by the date of the Sunday that follows it and the custom time 12:00:00. For example, if the value of a date column in an EHS table is "19/9/2017", this date will be replaced by "24/9/2017 13:23:21". The later date will be used for all other records in EHS, which have an original date falling in the period from 18/9/2017 to 24/9/2017 12:00:00;
- Deletion of values (direct identifiers): default values used in cases where personal data should be removed by the data anonymization process (e.g., email, first name, last name, postal code if applicable). Notably, direct identifiers are not anticipated to be transmitted via any of the interoperable systems (hospital EHRs, S4H, HB).
- Reduction of the level of detail as for indirect identifiers (e.g., gender, uncommon characteristics of weight and/or height, individuals' mental/physical well-being, profession, geolocation): in the event of such occasion resulted data will not leave the EU. During the initial stages of the data digestion (pre-processing stage), bracketing techniques (i.e., combining/grouping the categories of a variable), top-coding and disturbing (as for GPS data) techniques will be applied.

Records of IDs associations (i.e., EHR IDs, S4H IDs, Pseudo-Id1, Pseudo-Id2) will be stored in an encrypted form within the Secure User Data Storage of the SB@SecurityComponent, while pseudonymised data will be stored within the main SB@Repository.

C: Data transfer

- Any transfer of personal data which are undergoing processing or are intended for processing after transfer to a third country or to an international organisation shall take place only if, subject to the other provisions of the GDPR, the conditions laid down are complied with by the controller and processor, including for onward transfers of personal data from the third country or an international organisation to another third country or to another international organisation. All provisions shall be applied in order to ensure that the level of protection of natural persons guaranteed by the GDPR is not undermined.
- Whenever SB processes personal data, this action will take place in the ICCS server room (Greece). In no case resulted data leave the EU (ICCS act as Data controller).

D. Data integrity

- Data integrity will be achieved by a variety of data protection methods including backup and replication, auditing (of access logs, GDPR requests status, etc.), data and input validation, duplication removal and access controls.

E. Data minimisation

- Access to the anonymized data in the SB@Repository and those pseudonymised ones kept in SB@SecurityComponent, will be granted through the realization of Role-Based Access Control (RBAC) to specific end-user roles, limiting/preventing the access/process of Dashboard services (i.e., indirectly to data and information stored in SB repository and to all other SB@Cloud components) to what is directly relevant and necessary to accomplish a specified purpose.



- As described in previous Section 2, all RESTfull APIs will need to first obtain a secure token from the “Secure Manager One”. SB authentication manager will utilize the WSO2 Identity Server and API manager. The former builds agile, extensible CIAM solutions (by using the XACML rules) and the latter is a complete enterprise-class API management solution that combines easy, managed API access with full API governance and analysis. Each REST will check whether a specific end-user has access to a specific REST within a context (e.g., pair a device, create an analysis). The APIManager will be used by each REST service and will invoke an additional Header named "X-JWT-Assertion". This header contains an Authentication and authorisation token that contains information about the identity of the end-user that has invoked the REST API. The developer of the API will use the token to verify that the user that invoked the REST call holds the relevant permissions. In addition, via the Secure Manager One limiting the access of each-user on specific Organisation-based data will be accomplished (during the initial data gathering). In that sense, end-users of each pilot site will be able to have access to pseudonymised data (i.e., personal health records, smartphone pairing records) only to patients of their site/organisational unit. The SB system administrators will be able to manage end-users from all sites and perform actions such as system monitoring, services health check etc. Clinicians (Clinical Case Managers, Caregivers) will have access only to relevant views of the dashboard that relate to the pseudonymised medical history of patients they monitor, while they will all be able to view resulted analytics and interventions (performed to all data). Technicians that will support the pilots will also have access to views that provide tools for the installation and maintenance of the system and will also be able to have an overview of alarms, for example for devices that do not send data and possibly would require support from the technical team. Following this:
 - SB end-users (SB internal roles such as System Administrator, Clinical Case Manager, Caregiver, Data Scientist, Auditor and Patient) will be registered via the SB authentication manager.
 - SB@App and SB@HomeHub that trigger REST API will be associated with a Patient.

Notably, even if in the unlikely event of malicious access to both SB@Repository and SB@Secure User Data Storage is achieved, or data are leaked by administrator negligence, still will not be possible to reidentify study participants. External EHRs administrators or other EHR end-users who are aware of Ids are been designated to their patients, even if they gain access to data been stored in SB@Repository, still they will not be able to generate the association between participant's Pseudo-Id1 and Pseudo-Id2 (stored in encrypted fashion in SB@Secure User data Storage), and consequently cannot correlate Pseudo-Id2 with the Id stored in their systems.

F. Storage limitation

- SB pseudonymous data (as a whole) will become anonymous when the separately stored IDs association records will be deleted. Thus, after the completion of the SB project, data kept in SB@SecurityComponent will no longer be needed to conduct the research (e.g., analytics, interventions), and consequently will be erased and not further used for any data process.

G. Fair data process

- Data collected is provided by the study participants on a voluntary basis, in accordance with well-defined informed consent procedures, goals of data processes and individual’s rights. In principle, SB data and resulting analytics and interventions do not consider any personal, discriminative references. As such, there is no risk of discrimination due to data processing conduct in the context of SB.

The aforementioned procedures are accompanied by other security aspects covering the whole range of components, such as:

- Token-based access to RESTful APIs;
- Role-based access control: Access to the anonymized data in the SB platform will be granted through the realization of role-based access control to specific end-user roles, limiting/preventing also the access/process of personal information to what is directly relevant and necessary to accomplish a specified purpose;
- Protected logging system for monitoring data access;

In D2.1 (Sections 5 and 6), a list of functional end-user requirements has been established for the purpose of end-user management and most importantly for GDPR compliance. The following table (Table 1) maps those functional requirements to specific functionalities to be supported by the SB@SecurityComponent.

Table 1: Requirements and corresponding actions

Requirement	Provision
<p>R202 Secure Authentication</p>	<p>Security Authentication – Authorisation - Monitoring personal data access</p> <ul style="list-style-type: none"> • SB@App transmits via a secure HTTPS channel. SB@App has been granted access if is associated to a valid external Pseudo-Id1 (valid study participant) • SB@SecurityComponent requests authentication in the form of a token to either reject or allow this authentication and data transmission • SB@SecurityComponent holds end-user’s personal data and provides authentication mechanism ensuring access by authorized person only • SB@Dashboard provides authentication subject to Authorisation policy. This authentication ensures that an entity (i.e., SB registered active end-user, SB@App with valid external Pseudo-Id1) can access SB@Cloud REST services if a valid username-password authentication or an OAuth 2.0 certificate-based authentication is provided respectively • Access to structured pseudo-anonymised data as allowed only to authorized persons (i.e., SB registered active end-users) • Access to participants IDs association records allowed only to authorised persons (i.e., SB Sys administrators and Auditors) and is been monitored for auditing purposes
<p>R230</p>	<p>Security - End-user registration</p> <ul style="list-style-type: none"> • An end-user might gain access to the SB@Cloud if he/she provides a valid email, full name, organisation and role



<p>Register/update/lock/suspend a end-user (Dashboard end-user)</p>	<ul style="list-style-type: none"> Registration request will be subject to SB Sys administrator approval. Administrator will contact a responsible for the specific organisation (each organisation must nominate at least one “responsible person”) to confirm the end-user's identity and email. Upon receiving the relevant information, logins to admin area to approve or deny the registration request
<p>R230 Register/update/cancel a new user (participant)</p>	<p>Security - Recruitment (When a patient is recruited for SB, he/she goes to the local pilot partner)</p> <ul style="list-style-type: none"> Step 1. the patient is registered at the local clinical system (EHR) of the pilot partner Step 2. During the registration process, EHR will request the SB@Cloud to create a pseudo identifier for the patient and send it back to it in order to complete the registration process. This pseudo identifier will be called Pseudo-Id1 Step 3. When it creates Pseudo-Id1, the SB@Cloud will also create a second (internal) pseudo identifier (i.e., Pseudo-Id2) and associate it with Pseudo-Id1 simultaneously IDs association (i.e., between Pseudo-Id1 and Pseudo-Id2) is stored in Secure User Data Storage. This Pseudo-Id2 will be used by SB Cloud onwards as the id for any data that will ever be received for the particular patient. The association will be used in data ingestion and extraction. Pseudo-Id2 will NOT be communicated back to EHR
<p>R231 Assign a role to the user</p>	<p>Security - Privacy</p> <ul style="list-style-type: none"> RBAC
<p>R232 Assign access right to roles</p>	<p>Privacy Separation of data - Authentication – Authorisation</p> <ul style="list-style-type: none"> Different level of data obfuscation is been applied depending on the level of authorisation of the person accessing the data
<p>R301 Privacy of cloud data</p>	<p>Security – Privacy - Local de-identification and anonymization</p> <ul style="list-style-type: none"> Devices and sensors collect data using non-permanent buffer or addressing privacy and security if permanent SB@App stores external Pseudo-Id1 in encrypted fashion. Data stored in smartphone’s local storage does not contain any PII’s SB@App and sensors transmission are protected by state-of-the-art security and privacy technologies and methods SB@SecurityComponent pseudo-anonymises data prior of their storage in SB@Repository SB@Repository digests anonymised data



<p>R303 Implement privacy by design</p> <p>R304 Implement privacy by default</p>	<p>Privacy by design - Data minimisation - Accountability</p> <ul style="list-style-type: none"> • Recipients of devices and sensors transmission keep the usage data received and data further elaborated as confidential • Behavioural and environmental data collected do not contain any PIIs. Behavioural and environmental data are been treated in a confidential manner • Access to pseudo-anonymised data is limited to those needed by analytics • SB@SecurityComponent when replaces IDs and PIIS performs the action securely avoiding data scavenging • Privacy and security of structured data is been kept in all circumstances • RBAC (via configuration of OAuth 2.0 based on REST OpenAPI Specification) link access permissions to particular roles • Logging features are protected (separated encrypted repository, logged access) • SB@Repository backup mechanism and SB@SecurityComponent logging mechanism guarantee the integrity of the data
<p>R305 Implement measures to demonstrate compliance</p>	<p>Privacy – Monitoring mechanisms</p> <ul style="list-style-type: none"> • GDPR requests management follows the transparency regulation while informing end-users • Data kept in SB@SecurityComponent will no longer be needed and scheduled to be deleted upon SB conclusion • SB@Cloud will adopt a continuous compliance monitoring approach supporting auditability (T5.2) to ensure the security and privacy of the data held in the SB platform and the protection of the platform itself
<p>R313 Notify and communicate personal data breaches</p>	<p>Privacy - Data breach</p> <ul style="list-style-type: none"> • If a personal data breach occurs, Data controllers shall without undue delay and, where feasible, not later than 72 hours after having become aware of it, notify the supervisory authority of the personal data breach, unless the personal data breach is unlikely to result in a risk to the rights and freedoms of stroke survivors. Where the notification to the supervisory authority is not made within 72 hours, it shall be accompanied by reasons for the delay. • The notification of a personal data breach to the supervisory authority shall at least: <ul style="list-style-type: none"> ○ describe the nature of the personal data breach including where possible, the categories and approximate number of data subjects concerned,



	<p>and the categories and approximate number of personal data records concerned.</p> <ul style="list-style-type: none"> ○ communicate the name and contact details of the data protection officer or other contact point where more information can be obtained. ○ describe the likely consequences of the personal data breach; describe the measures taken or proposed to be taken by the controller to address the personal data breach, including, where appropriate, measures to mitigate its possible adverse effects. <ul style="list-style-type: none"> ● When the personal data breach is likely to result in a high risk to the rights of a participant, the controller will <ul style="list-style-type: none"> ○ communicate the personal data breach to the data subject without undue delay, describing in clear and plain language the nature of the personal data breach ● In addition to aforementioned actions, appropriate organisational actions on how to handle different cases of data breaches will take into account recent guidelines (EDPB, 2021)
<p>R306 Respect data subject rights R308 Provide information</p>	<p>Privacy – GDPR requests management</p> <ul style="list-style-type: none"> ● When a participant wishes to receive his/her data which are held in SB@Cloud: <ul style="list-style-type: none"> ○ he/she may initiate a GDPR request through his/her smartphone or through the pilot partner (CCM) who recruited him/her through the SB@Dashboard ○ in both these cases the request will arrive at the SB@Cloud under the Pseudo-Id1, which is known to the device of the patient and EHR ○ upon the receipt of such a request, SB@Cloud will deliver the record of the patient to the smartphone of the patient under Pseudo-Id1 (Figure 4, Data Flow Type 3) ● Response (and relevant data in machine-readable format) will be delivered to participant’s smartphone or the relevant clinical partner respectively (patients association, however, will not be communicated to EHR)
<p>R309 Prevent transfers of personal data outside the EEA</p>	<p>Privacy - Data transfers</p> <ul style="list-style-type: none"> ● SB@Cloud will not transmit anonymised data back to EHRs or elsewhere, however a participant may choose to share a SB report to his/her clinician or other ● Pilot partners may request access to the data of their patient (the purpose for which they may wish to do so needs to be clarified in the consortium). Such requests may come from EHR and cannot refer to individual patients. Upon the receipt of such requests

	<ul style="list-style-type: none"> ○ SB Cloud will send back the data of the patients of the relevant clinical partner (Figure 4, Data flow of type 4) but these data will be under another pseudo id (Pseudo-Id3) that will be generated by SB@Cloud and associated with Pseudo-Id2 (and as a consequence) with Pseudo-Id1. This association however will not be communicated to EHR. Also, the transfer of data under this flow will be filtered in order to ensure that it will be unlikely for users of EHR to identify a patient from other (non-ID) values in the patient’s record. ● In no case resulted data leave the EU
<p>R311 Maintain records of processing activities</p> <p>R316 Allow for the DPO to fulfil his tasks</p> <p>R318 Respect third-party rights</p>	<p>Privacy – GDPR requests management</p> <ul style="list-style-type: none"> ● Log (record-keeping system for processing activities) will record access to “Secure User Data Storage” and any GDPR-requests and their status ● Logging mechanism of GDPR requests tracks the progress of individual cases (i.e., participant’s Id, requested-on behalf, status, request timestamp, time-limits, assigned-to, justification, completion timestamp (if any)) and demonstrates compliance with the GDPR (compliance with timescales, maintained logs for audit, evidence to be requested by the supervisory authority)
<p>R315 Implement pseudonymisation, anonymization or deletion</p>	<p>Privacy by design - Data minimisation – Accountability</p> <p>In addition to R303, R304:</p> <ul style="list-style-type: none"> ● PII’s will be deleted or obfuscated or pseudonymised ● Records of IDs associations to be stored in an encrypted form

3.2 SB@SecurityComponent: Core Component Specification

The SB@SecurityComponent will provide an interface (REST API) to be used by i) the SB@Dashboard end-users, ii) the SB@App and iii) HA Vendor’s cloud (i.e., Push/pull REST services). In this section a more detailed view of the basic interactions between the SB@Dashboard, the SB@SecurityComponent and the REST component is presented, using the UML sequence diagrams representation, REST service definitions and associated design mock-ups.

3.2.1.1 End-user registration and login

In order to register in the SB@Cloud, an end-user must go through the following (Figure 5, and Figure 6):

1. Provide a valid email, along with a full name, organisation and a role in the registration form. Form will also ask for the end-user’s consent to register with the SB@Cloud.
2. Upon successful creation of the registration request, the end-user will receive an email in order to verify his/her email
 - The SB@Cloud provides e-mail notification services to the end-users

3. Upon confirmation, a registration request (email notification) will be sent to the SB Sys administrator for approval. Administrator will contact a responsible for the specific organisation (each organisation must nominate at least one “responsible person”) to confirm the end-user's identity and email. Upon receiving the relevant information, logs into admin area to approve or deny the registration request:
 - in case of acceptance: user receives an email with a link that prompts him/her to insert a strong password
 - in case of rejection: user will be notified that his/her registration has been rejected
4. Upon successful registration (i.e., active status) end-user will go through the authentication process (i.e., enters his/her credentials)

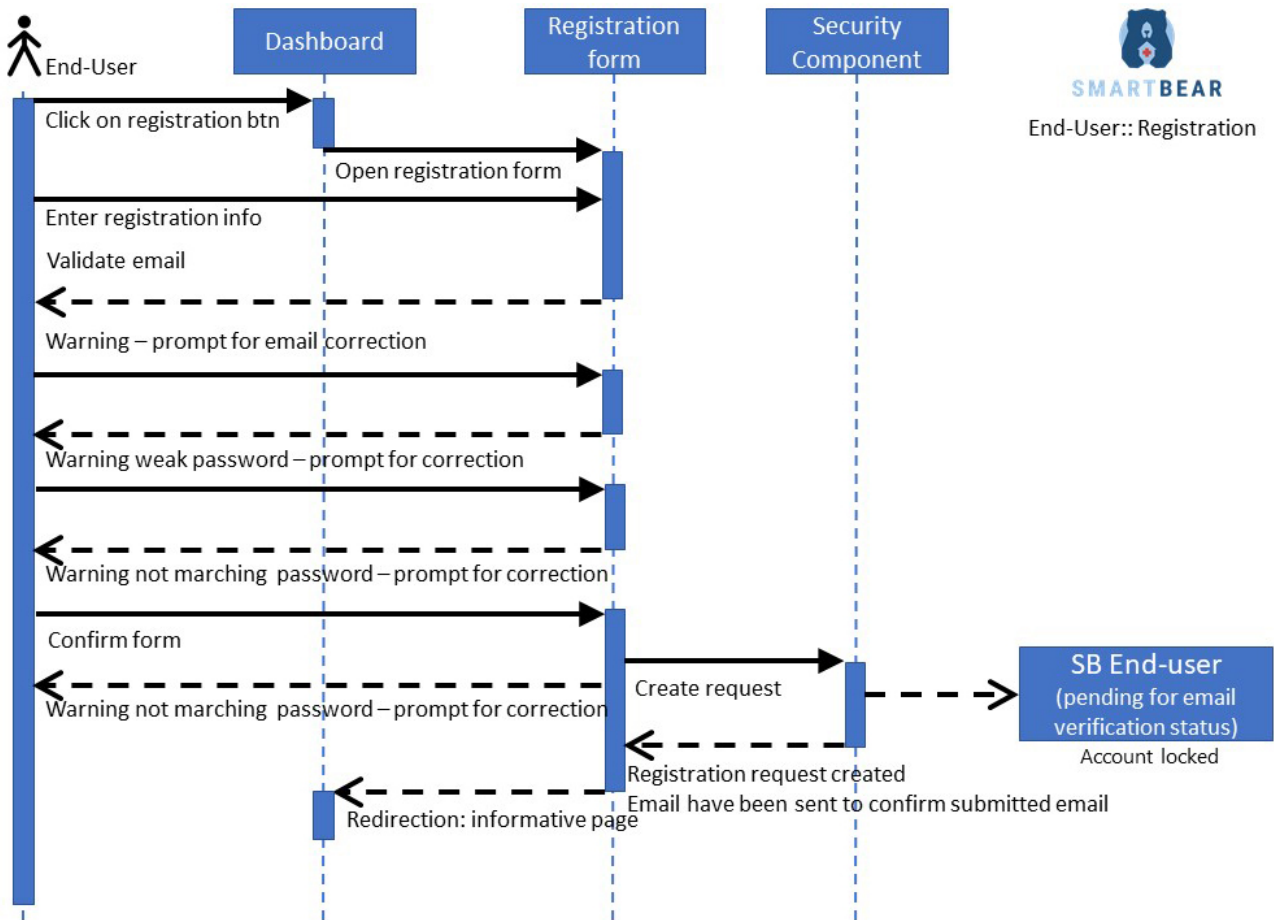


Figure 5: Sequence diagram: end-user registration

Method: registration	Params	Response
POST	Email, personal details, password	HTTP 200 Response Code JSON with the attributes token and status (success/error)

Registration

i Informative text

* First name:

* Organisation: ▼
{2B, ANA, ATC, ATOS, CATEL, CITY, CNR, CSC, FCSR, FORTH, IBM, ICCS, INV, ITTS, LISPA, MPF, NKUA, NOVA, PHILIPS, QUIRON, ROP, SRS, STS, SV, UMIL, UOI, UPV, ...}

* Email:

* Last name:

* Requested role: ▼
{System Administrator, Patient, Clinical Case Manager, Caregiver, Data Scientist, Policy maker, Auditor}

Register
Cancel

SC-2
REST-2
D-2

Figure 6: Design mock-up: end-user registration

Via a login form, end-users can enter their credentials in order to access the SB@Cloud services (Figure 7, and Figure 8). The REST authentication service communicates with the SB@SecurityComponent:

1. to verify whether the end-user has an active account status, and if it does, whether the password he/she typed is correct or not
2. if credentials used are the correct ones, a valid TOKEN is generated and used throughout the end-user’s session until the web browser is closed or he/she chooses to log out

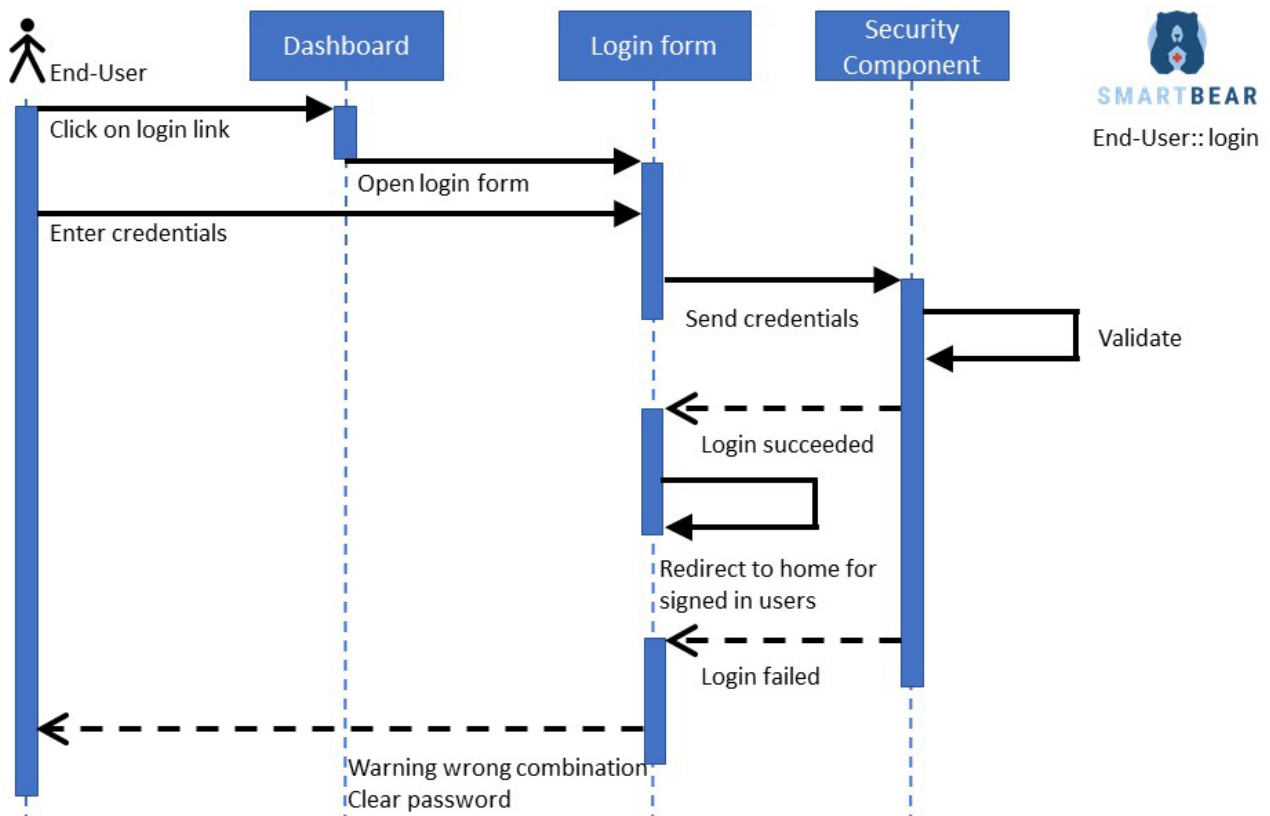


Figure 7: Sequence diagram: end-user’s login

Table 2: Login: REST service definition

Method: login	Params	Response
POST	Email - password	HTTP 200 Response Code

		JSON with the attributes token and status (success/error)
--	--	---

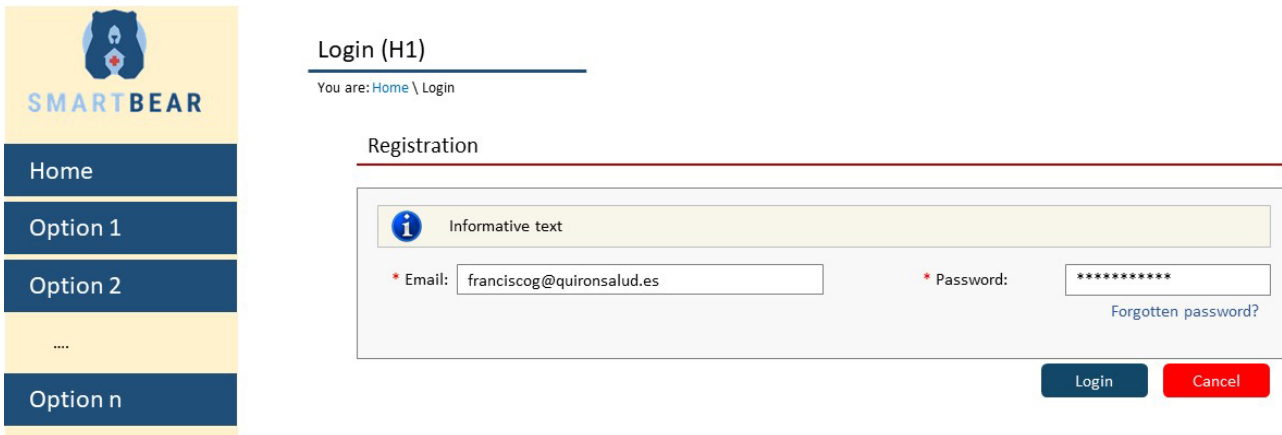


Figure 8: Design mock-up: end-user's login

Whenever a signed-in end-user wishes to exit, trigger of logout service will logout and make the access token deprecated.

Table 3: Logout: REST service definition

Method: logout	Params	Response
GET		HTTP 200 Response Code

3.2.1.2 End-user management

The user management service allows the management of SB@Dashboard end-user accounts. Access is not granted automatically, but via a moderation mechanism the SB System Administrator assesses whether end-user registration information is a valid one and he/she is allowed to gain access according to a role. SB System Administrator may assign a different role that the one originally requested (during registration process), and as such different privileges and access to the services supported (Figure 9, Figure 10, Figure 11, Figure 12, Figure 13, Figure 14, and Figure 15).

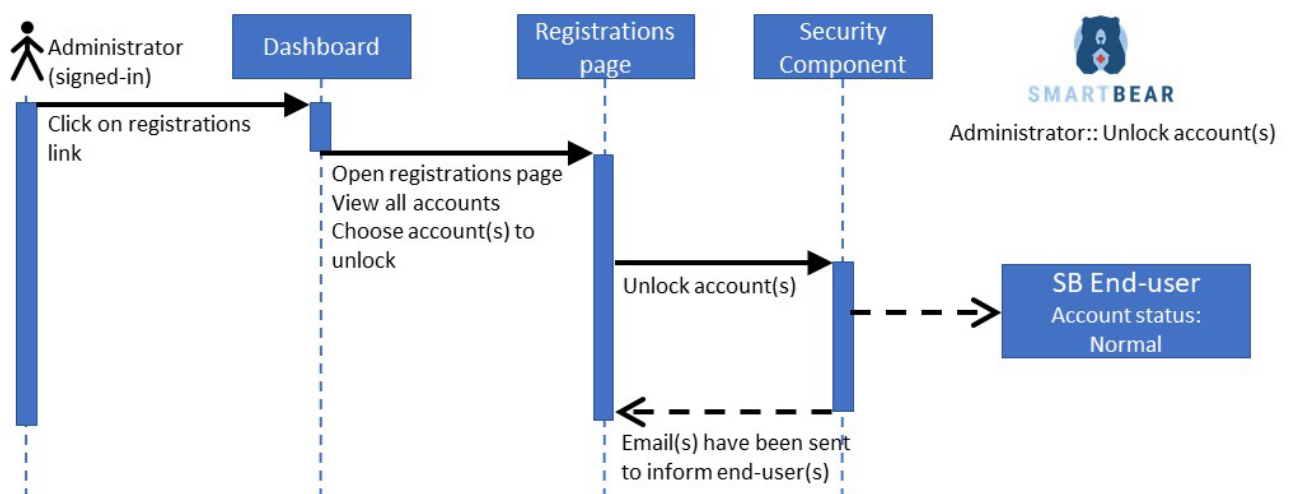


Figure 9: Sequence diagram: System administrator activates end-user's registration request (unlocks a previously validated account)

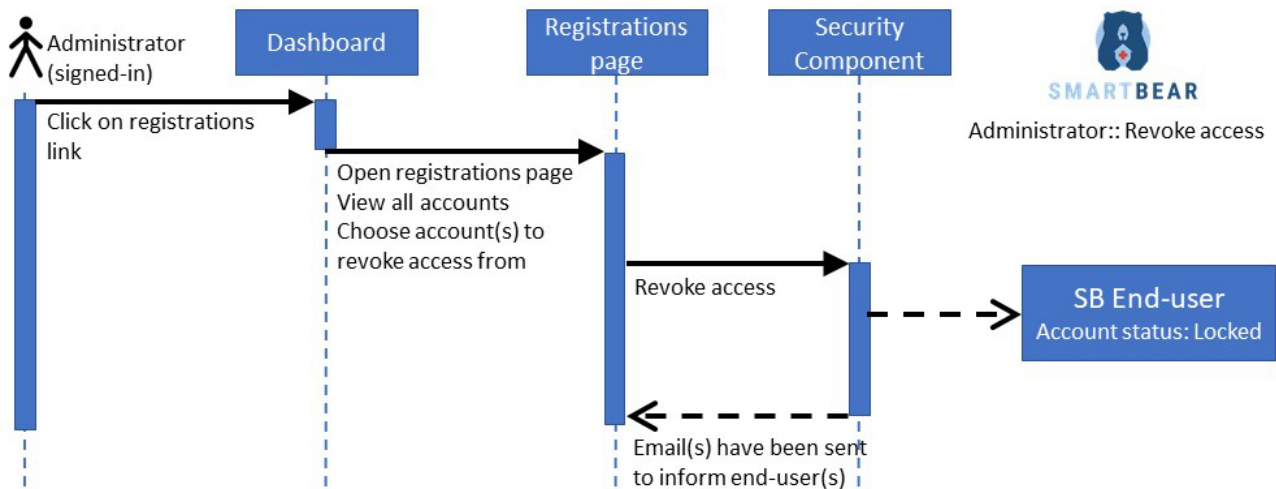


Figure 10: Sequence diagram: System administrator revokes end-user’s access (locks/suspends access)

Table 4: End-user management: REST service definitions

Method: Get all	Params	Response
GET		JSON records of all end-users containing UserId, fullname, organisation, role, and last update of account status
Method: Get		
GET	UserId	JSON record of specific end-user ID containing UserId, fullname, organisation, role, status, last update of account status and associated action(s) (JSON: type, assigned to, status and timestamp)
Method: Edit		
POST	UserId, first name, last name, organisation, role, email and status	JSON record containing UserId, fullname, organisation, role, status, last update of account status and associated action (JSON: type, assigned to, status and timestamp)
Method: Create action		
POST	UserId, type, and assigned to	JSON record containing UserId, fullname, organisation, role, status, last update of account status and associated action (JSON: type, assigned to, status and timestamp)
Method: Edit action		
POST	UserId, type, assigned to, and status	JSON record containing UserId, fullname, organisation, role, status, last update of account status and associated action (JSON: type, assigned to, status and timestamp)

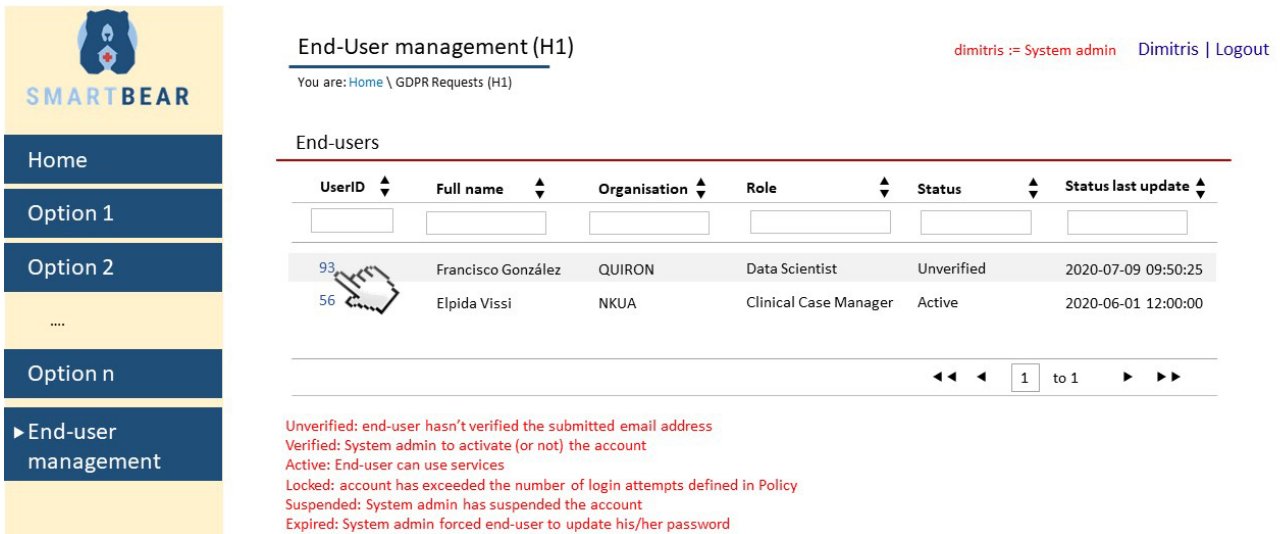


Figure 11: Design mock-up: Activate an account: Step 1/4: System administrator selects the ID of a newly registered end-user

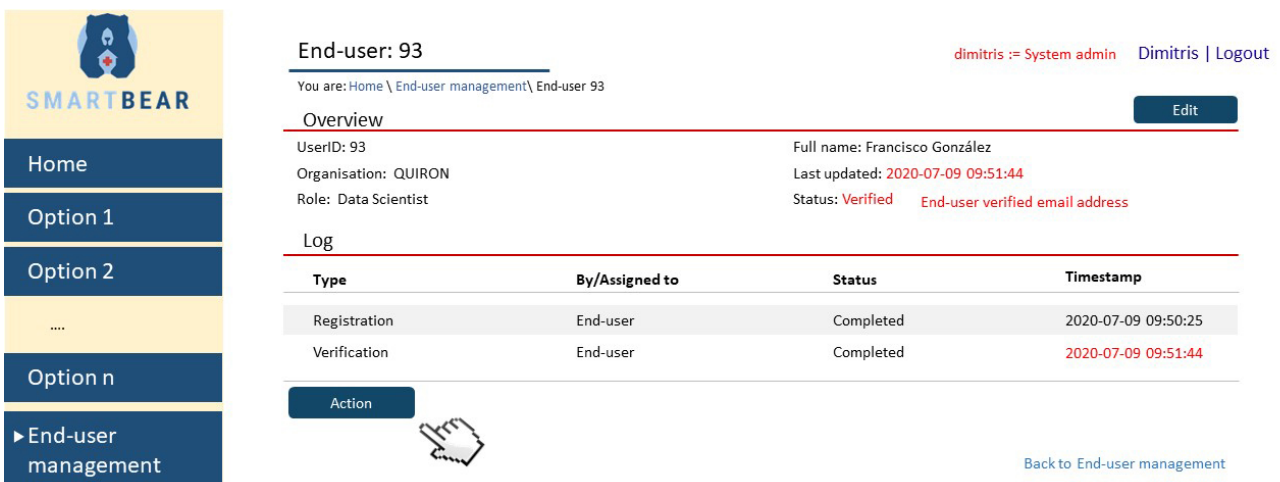


Figure 12: Design mock-up: Activate an account: Step 2/4: System administrator triggers a new action for the specific account

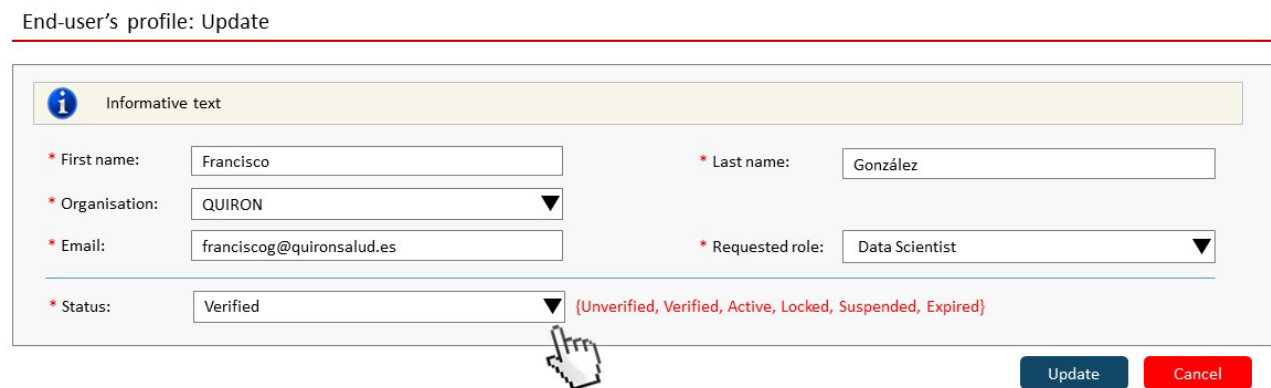


Figure 13: Design mock-up: Activate an account: Step 3a/4: System administrator alters account status

End-user's profile: Update

i Informativo text

* First name:

* Organisation:

* Email:

* Status: {Unverified, Verified, Active, Locked, Suspended, Expired}

* Last name:

* Requested role:

Figure 14: Design mock-up: Activate an account: Step 3b/4: System administrator changed account status to “active”

- Home
- Option 1
- Option 2
-
- Option n
- ▶ End-user management

End-user: 93

dimitris := System admin Dimitris | Logout

You are: Home \ End-user management \ End-user 93

Overview

UserID: 93

Organisation: QUIRON

Role: Data Scientist

Full name: Francisco González

Last updated: 2020-07-10 10:11:00

Status: Active Sys admin activated the account

Log

Type	By/Assigned to	Status	Timestamp
Registration	End-user	Completed	2020-07-09 09:50:25
Verification	End-user	Completed	2020-07-09 09:51:44
Activation	Dimitris	Completed	2020-07-10 10:11:00

[Back to End-user management](#)

Figure 15: Design mock-up: Activate an account: Step 4/4: System administrator reviews updated account status

Notably, SB@SecurityCompoment records all activities related to end-users’ accounts.

3.2.1.3 End-user’s profile

My Profile service enables registered end-users to manage their personal details and login credentials. The end-user can click on a link named by his/her first name to view and edit his/her profile information, with the option to edit those, or to alter the password (Figure 16, Figure 17, Figure 18, and Figure 19).

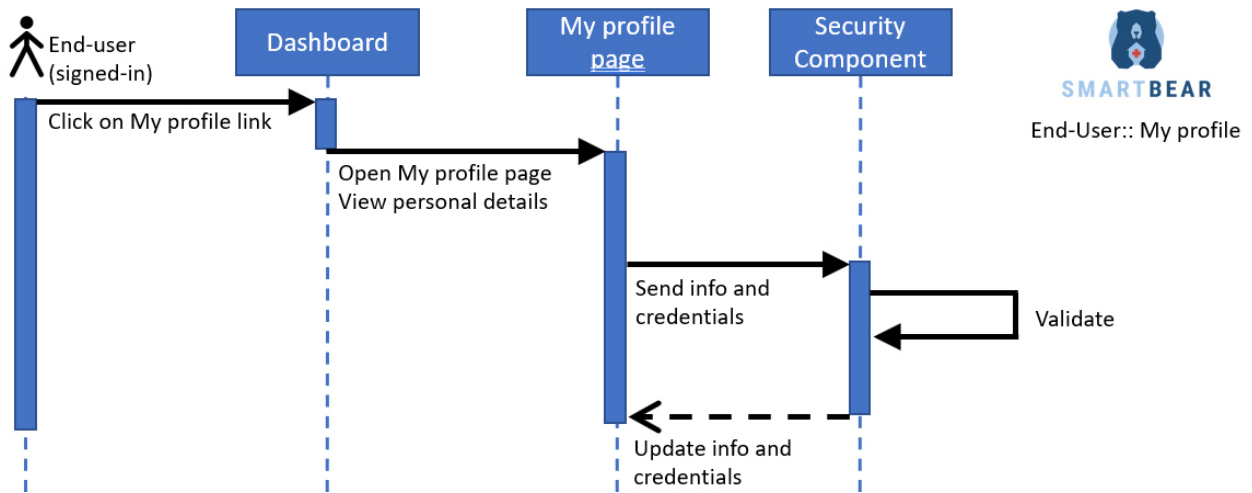


Figure 16: Sequence diagram: End-user updates his/her personal details

Table 5: My profile: REST service definitions

Method:	My profile	Params	Response
GET		UserId	JSON record of specific end-user ID containing UserId, fullname, organisation, role, status, last update of account status and associated action(s) (JSON: type, assigned to, status and timestamp)
Method: Edit			
POST		UserId, first name, last name, organisation, role, email and status	JSON record containing UserId, fullname, organisation, role, status, last update of account status and associated action (JSON: type, assigned to, status and timestamp)

End-user: 56 Elpida Vissi := Clinical Case Manager | Elpida | Logout

You are: Home \ End-user management \ End-user 56

Overview

UserID: 56	Full name: Elpida Vissi
Organisation: NKUA	Last updated: 2020-06-01 12:00:00
Role: Clinical Case Manager	Status: Active

[Edit](#)

Log

Type	By/Assigned to	Status	Timestamp
Registration	End-user	Completed	2020-06-01 10:45:00
Verification	End-user	Completed	2020-06-01 11:00:00
Activation	Dimitris	Completed	2020-06-01 12:00:00

Figure 17: End-user's (e.g., Clinical Case manager) profile information

My profile: Update

i Informative text

* First name:

* Organisation:

Email:

Alter email address

* Last name:


Requested role:

Alter role

Unmoderated changes (first and last name, Organisation)

Moderated changes: email (re-verification by end-user, re-activation by sys admin), role (re-approval by sys-admin)

Figure 18: Editing end-user's profile information



- Home
- Option 1
- Option 2
-
- Option n

End-user: 56

You are: Home \ End-user management \ End-user 56

Elpida Vissi := Clinical Case Manager

Elpida | Logout

Overview

UserID: 56	Full name: Elpida Vissi
Organisation: NKUA	Last updated: 2020-06-01 12:00:00 No change
Role: Clinical Case Manager	Status: Active

Log

Type	By/Assigned to	Status	Timestamp
Registration	End-user	Completed	2020-06-01 10:45:00
Verification	End-user	Completed	2020-06-01 11:00:00
Activation	Dimitris	Completed	2020-06-01 12:00:00

Figure 19: End-user's (e.g., Clinical Case manager) updated profile information

3.2.1.4 GDPR Requests management

Collecting and managing privacy-related requests been handled by the GDPR requests service, which accept requests and forwards them to SB System Administrators for processing. Participants may issue a request via the SB@App (alternatively via the SB@Dashboard by the Clinical Care Manager on their behalf) by which can also track status, follow-up, and provide feedback to those handling their requests. In addition, an Auditor may track the progress of the request and see who is currently working on it. The following roles are assigned to this service to support request creation, handling and tracking (Figure 20, Figure 21, Figure 22, Figure 23, Figure 24, Figure 25, Figure 26, Figure 27, Figure 28, Figure 29, and Figure 30):

- System Administrator
 - View GDPR request(s) created by participant(s) (or by Clinical Case Managers in their behalf)
 - View notification(s) for pending action(s)
 - View GDPR request details and associated actions
 - View action details
 - Assign action(s) in respect to request solving
- Auditor
 - View GDPR request(s) created by participant(s) (or by Clinical Case Managers in their behalf)
 - View notification(s) for pending action(s)

- View GDPR request details and associated actions
- View action details

- Clinical Case Manager
 - View GDPR request(s) created by him/her on behalf of a participant
 - Create GDPR request (on behalf of a participant)
- Patient (or participant)
 - View GDPR request(s) created by him/her
 - Create GDPR request

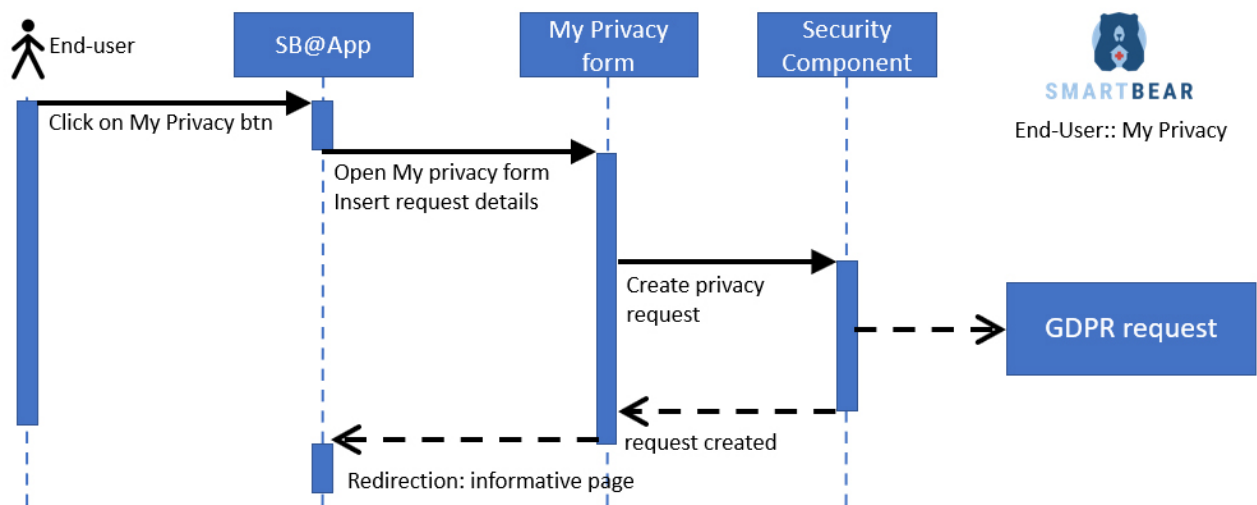
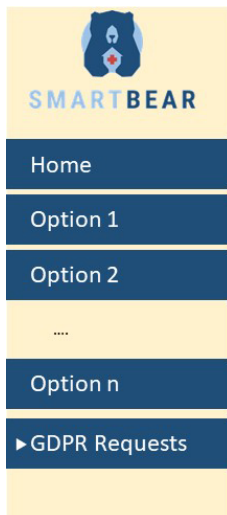


Figure 20: Sequence diagram: End-user creates a GDPR request via his/her SB@App

Table 6: GDPR request management: REST service definitions

Method: Get all Requests	Params	Response
GET		JSON records of all GDPR requests containing requestID, participantID/UserId, category, status, and last update of request status
Method: Get		
GET	requestID	JSON record of specific GDPR request containing requestID, participantID/UserId, category, status, and last update of request status and associated action(s) (JSON: type, assigned to, status, priority and timestamp)
Method: Edit		
POST	requestID, participantID/UserId, category, status	JSON record of specific GDPR request containing requestID, participantID/UserId, category, status, and last update of request status and associated action(s) (JSON: type, assigned to, status, priority and timestamp)

Method: Create request		
POST	participantID/UserId, category	JSON record containing requestID, participantID/UserId, category, status, and last update of request status and associated action(s) (JSON: type, assigned to, status and timestamp)
Method: Assign action		
POST	requestID, action type, assigned to, and priority	JSON record containing requestID, participantID/UserId, category, status, and last update of request status and associated action(s) (JSON: type, assigned to, status, priority and timestamp)



GDPR Requests (H1) dimitris := System admin Dimitris | Logout

You are: Home \ GDPR Requests (H1)

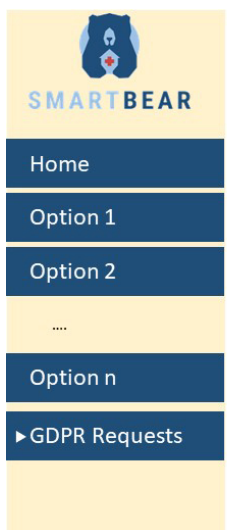
Participants Requests

RequestID	ParticipantID	Category	Status	Last updated
1	44844	Right to be informed	Initiation	2020-07-09 09:50:25

Notifications for assigned actions

None.

Figure 21: System Administrator selects to view GDPR request details



GDPR Request: 1 dimitris := System admin Dimitris | Logout

You are: Home \ GDPR Requests \ Request 1

Overview Print Edit Delete

ParticipantID: 44844
 Creator: Participant
 Category: Right to be informed
 Last updated: 2020-07-09 09:50:25
 Status: Initiation

Actions

Type	By/Assigned to	Status	Timestamp
Initiation	Participant	Completed	2020-07-09 09:50:25
Access log creation	Not assigned yet	Pending	
Access log review	Not assigned yet	Pending	
Access log transmission	Not assigned yet	Pending	

Assign action(s)

[Back to GDPR Requests](#)

Figure 22: System Administrator views request details and triggers the assignment of actions to specific end-users (SB System Administrators)

Assign action(s)

i Informative text

RequestID: 1 Category: Right to be informed

Pending action: Access log creation To: Dimitris Priority: High

* Assignment 1: + X

Advise:

Pending action: Access log review To: Paul Priority: High

* Assignment 1: + X

Advise:

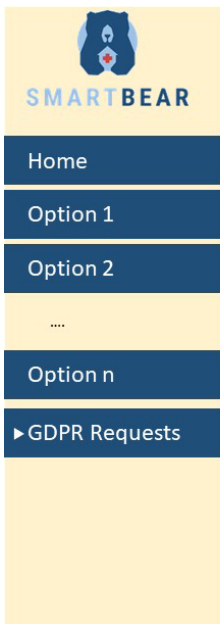
Pending action: Access log transmission To: Paul Priority: High

* Assignment 1: + X

Advise:

☞ Assign Cancel

Figure 23: Assign actions (in respect to a GDPR request): System administrator may assign one or more actions to designated end-users (SB System Administrators)



GDPR Requests (H1)

You are: Home \ GDPR Requests (H1)

dimitris := System admin Dimitris | Logout

Participants Requests

RequestID	ParticipantID	Category	Status	Last updated
1	44844	Right to be informed	Created	2020-07-09 09:50:25

◀◀ ◀ ▶▶ ▶▶ 1 to 1

Notifications for assigned actions

ActionID	Action	Category	Status	Last updated
1-2	Access log creation	Right to be informed	Pending	2020-07-09 09:50:25

Figure 24: GDPR actions been assigned to a System administrator

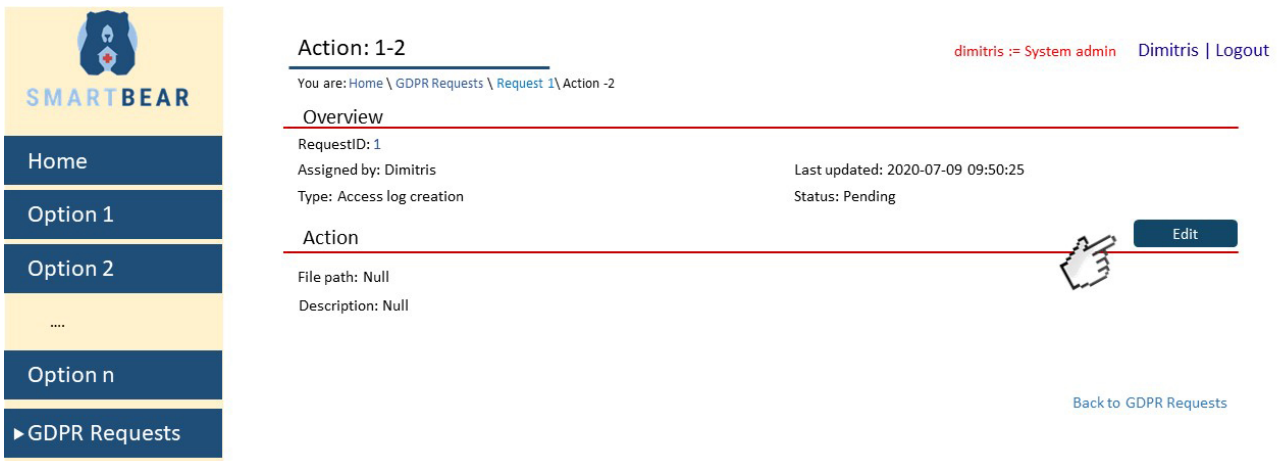


Figure 25: GDPR assigned actions: System administrator may edit the detail of an assigned to him/her action

Edit action



Figure 26: Edit GDPR assigned action: System administrator insert associated to the action log file and description

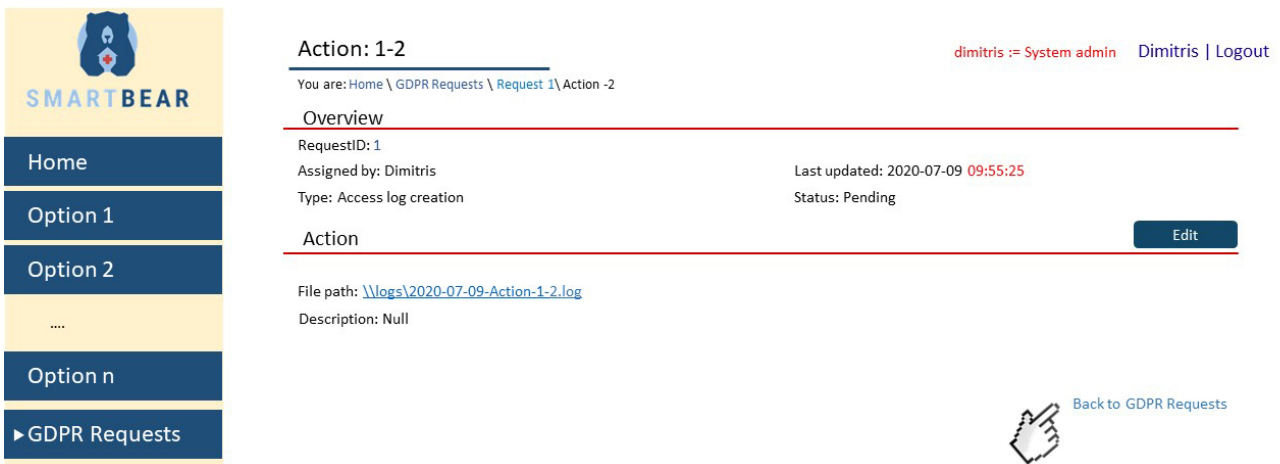
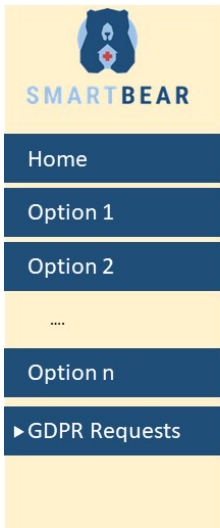


Figure 27: GDPR assigned actions: Updated action details



GDPR Request: 1

dimitris := System admin Dimitris | Logout

You are: Home \ GDPR Requests \ Request 1

[Print](#) [Edit](#) [Delete](#)

Overview

ParticipantID: 44844 Gender: Female
 Creator: Participant Last updated: 2020-07-09 09:50:25
 Category: Right to be informed Status: Access log creation
 Status updated

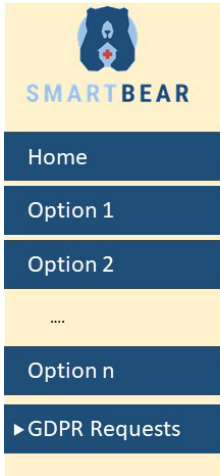
Actions

Type	By/Assigned to	Status	Timestamp
Initiation	Participant	Completed	2020-07-09 09:50:25
Access log creation	Dimitris	Completed	2020-07-09 09:55:25
Access log review	Paul	Pending	Status and timestamp updated
Access log transmission	Paul	Pending	

[Assign action\(s\)](#)

[Back to GDPR Requests](#)

Figure 28: Updated GDPR request details



GDPR Request: 1

Maria := Auditor Maria | Logout

You are: Home \ GDPR Requests \ Request 1

[Print](#)

Overview

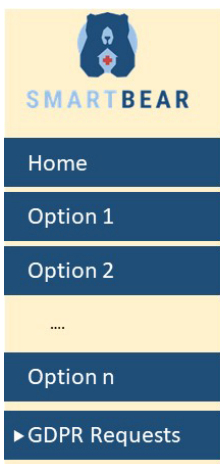
ParticipantID: 44844 Last updated: 2020-07-09 09:50:25
 Creator: Participant Status: Access log creation
 Category: Right to be informed

Actions

Type	By/Assigned to	Status	Log	Timestamp
Initiation	Participant	Completed		2020-07-09 09:50:25
Access log creation	Dimitris	Completed	Available	2020-07-09 09:55:25
Access log review	Paul	Pending		
Access log transmission	Paul	Pending		

[Back to GDPR Requests](#)

Figure 29: Auditor views GDPR request details and associated log files



GDPR Requests (H1)

Luisa := Clinical Case Manager Luisa | Logout

You are: Home \ GDPR Requests (H1)

Participants Requests

RequestID	ParticipantID	Category	Status	Last updated
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	
1	44844	Right to be informed	Access log creation	2020-07-09 09:50:25
44	3355	Right to rectification	Initiation	2020-08-10 12:00:00
177	1144	Right to data portability	Initiation	2020-09-09 11:50:25

[Create request](#)

Luisa has initiated requests for several participants

Figure 30: Clinical Case Manager can view all GDPR requests created by him/her

4 Security Infrastructure

4.1 Architecture

The Smart Bear architecture consists of the SB@App, the SB@HomeHub and the main SB@Cloud components (Figure 31).

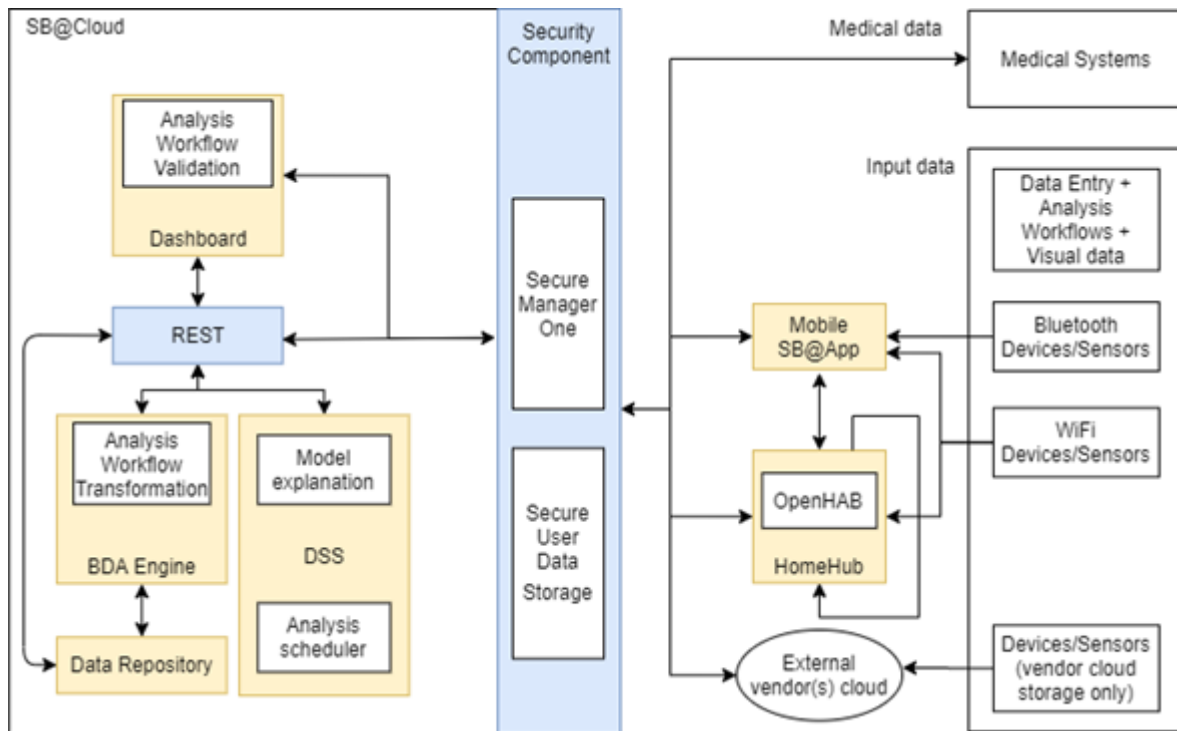


Figure 31: Smart Bear Architectural Overview

The SB@Cloud supports the main SB functionality such as: analysis, transformation, execution of the analytics workflows, generation of Machine Learning (ML) models, decision support for different types of interventions and data storage maintenance. It provides a dashboard to clinicians/data analysts to access services mentioned above. The dashboard supports the visualisation of the analytics outcomes as well. The SB@Repository component is intended to store SB data such as classifications, nomenclatures, terminologies, value sets, ontologies, data retrieved from the devices and Electronic Hospital Record (HER) systems, analysis workflows, results of the analytics workflow executions, intervention models, etc. Decision Support System (DSS) interacts with Big Data Analytics (BDA) engine to trigger its analytics workflows, run the intervention models, explain ML results and send notifications. BDA engine implements the analytics functionality using different ML techniques. It has a direct connection to the Data Repository to retrieve input data and store the executions results. SB@HomeHub and SB@App components are installed on the patient side and aim to handle mobile devices/home sensors data collection.

REST services are used by all components for communication. Such solution guarantees fast performance, reliability, and integration of new elements without affecting the system as a whole.

Security component is one of the main elements of the SB Architecture. It provides the protection for the Smart Bear ecosystem by implementing security mechanisms, in order to achieve the main goals of the project such as: collection, storage and access to the data in a secured way. Additionally, the security component provides a secure user data storage. It is intended to implement the authentication and the role-based access control of the SB infrastructure. The authentication process is implemented by the Secure Manager One (Figure 32). The SB architecture in more details is described in D2.2.

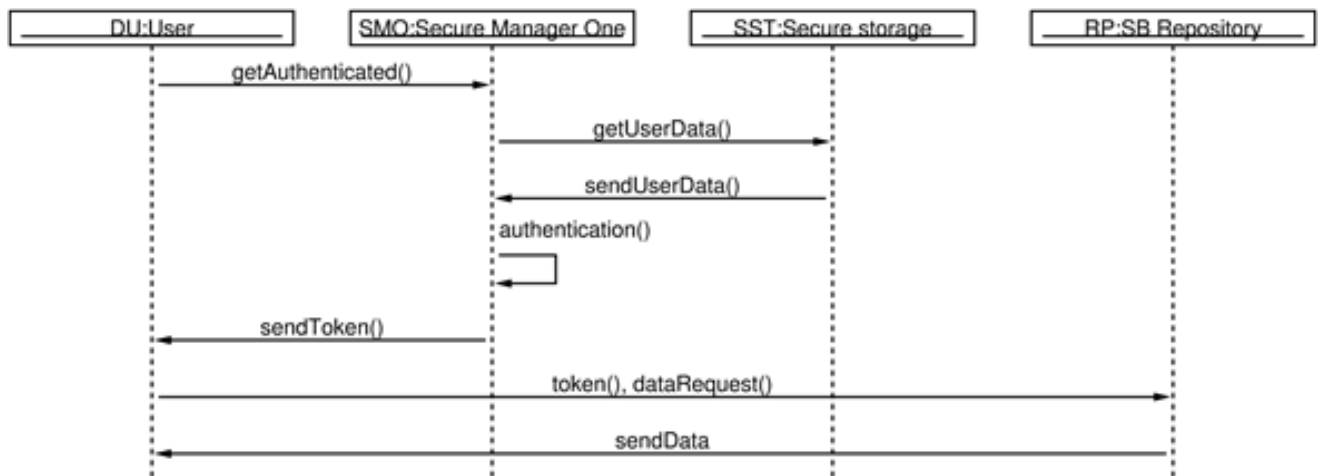


Figure 32: Security component interaction

4.2 Security Component

4.2.1 Cloud Infrastructure Security

The SB architecture follows a modern hybrid-cloud model, where part of the services is deployed on private cloud, and another part is deployed on a public cloud. To secure this hybrid-cloud model there is a need for a platform that provides security services for the hybrid cloud model.

As an initial solution, the architecture chooses Kubernetes for container orchestration to cope with the deployment, management, scaling, and networking of containers across different environments without needing to redesign it. Kubernetes is the most popular container orchestration system, it can be deployed on private and public cloud, and it is supported by various public cloud providers. Kubernetes offers several application security capabilities, but it lacks platform capabilities like: Authorisation and Authentication, Logging, Monitoring of user activity, DevSecOps pipelines, Vulnerability Scanning and so on. These platform security capabilities are required by regulations, and SB will be compliant with them.

OpenShift is an open source on-premises platform as a service built around Docker containers that are orchestrated and managed by Kubernetes. OpenShift can be deployed on private cloud, or on a public cloud setup. OpenShift provides all the security capabilities required for platform security and augments the application security capabilities provided by Kubernetes. Security and compliance capabilities provided by OpenShift:

- Authentication and Authorisation – this component will be used for authentication and authorisation of admins and developers of the SmartBear project. Authentication of end-users is answered by a different component;
- Logging – this component will be used for logging of all security related events. Logs will be archived for the number of days required by regulations;
- Monitoring – this component will be used to monitor user activity and alert on possible malicious insider activity (usually this indicates a breached account);
- Networking – this component will be used to configure required network security components;
- Pipelines – this component will be used to setup a CI/CD pipelines that ensure security measures are built-in by design in SB code;



- Service Mesh – this component will be used to ensure that communication between microservices is secure;
- Container security – scan container images for common vulnerabilities and exposures (CVE);

An updated and detailed version of the integrated framework will be reported in D6.2 (due in M19).

4.2.2 Mobile Security

The SB smartphone app (SB@App) acts as a gateway concerning the data flows from the various health, wearable and domotic devices integrated either through their SDK (Bluetooth connection) or via the web API (REST API connection) exposed by the corresponding vendor cloud. The SB@App stores temporarily all collected data in an internal database and periodically sends them to the SB data repository. In this respect, data protection mechanisms are implemented to provide security both while data are flowing between the SB@App and the SB@Cloud (data in transit) as well as while data stays temporarily in the SB@App internal database (data at rest).

Concerning data at rest, symmetric key encryption, based on the Advanced Encryption Standard (AES-256) algorithm, is applied to all collected data. The SB@App generates a cryptographic key while it is installed and registered for the first time. The cryptographic key is then stored in a keystore based on the Android Keystore system, which is an embedded security feature that protects key material from unauthorized use by preventing extraction outside the application processes. Moreover, it enforces the restrictions imposed by the authorized uses outside of the application's processes. The authorized uses are specified when the key is generated, and authorisations cannot be changed afterwards. This way, the SB@App process is the only one that can access the cryptographic key, ensuring also that the SB key cannot be extracted from the smartphone at all. In any case, all data stored in the application's internal database are pseudo anonymised and are also transmitted as such, making the actual linking between any health/wellbeing data to an identifiable user as hard as much as possible.

The proper management of the Android permission model highly affects the application's privacy. A privacy policy, that allows the SB@App users to be adequately informed and consent for any type of personal data processed, is attached to the SB@App. The main privacy design strategy followed is to minimise the personal data as much as possible.

Data transfers from the SB@App to the SB@Repository are encrypted under the HTTPS protocol. In the case of a vendor application integration, the SB@App follows the OAuth 2 user registration workflows and stores the generated user tokens encrypted in the internal database. Communication over HTTPS can be also applied whenever the SB@App requests home sensor data from the local SB@HomeHub component.

A more detailed report on the security provisions of the SB@App is provided in D3.2.

4.2.3 Secure Manager One

Secure Manager One consists of different off-the-shelf (OTS) components. For Identity Access Management (IAM), the WSO2™ identity server² (IS) which is a leading open-source software identity server will be deployed. WSO2 IS was decided to be used due to the in-house experience of one of the technical partner. For authorisation and data pseudonymisation it was decided for the same reason to use an OTS component also. The component of our choice is WSO2 API-Manager (APIM). Following is a more detailed description of the Secure Manager One components.

² <https://wso2.com/>



4.2.3.1 WSO2 Identity server

The WSO2 IS provides secure identity and access management by managing the identity of the users efficiently and by centralizing their administration and monitoring. WSO2 IS, enables extensible customization to fit the needs of the SB project, despite the fact that it is an OTS product. WSO2 IS will be used by the SB administrators to manage the users through both its management console and its RESTful API. Apart from the registered users, WSO2 IS can be used as an identity provider for third party systems that have their own set of users. SAML 2.0, OAuth 2.0 and OpenId Connect are some of the supported authentication protocols.

WSO2 Identity Server provides REST APIs for user management based on SCIM³. OAuth2 and OpenID connect provide the notion of claims, which are extra information that we can add to the user's profile to suit the goals of the project. In this context, during the registration a unique Pseudo-ID is generated for each user. This Pseudo-ID is then used for data-pseudonymisation by WSO2 API manager as described in Adhering to Privacy by Design Principles section.

4.2.3.2 WSO2 API Manager

The WSO2 API Manager's main features are the definition of RESTful APIs, the integration of governance policies and access control with OAuth2. It supports OpenAPI (formerly Swagger) specifications and manages throttling, access level and integration with various identity providers. It secures, protects, manages, and scales API calls by intercepting API requests and applying policies, such as throttling and security, using handlers and managing API statistics. Upon validation of a policy, the Gateway passes Web service calls to the destination backend. If the API call is a token request, the Gateway passes it directly to the WSO2 Identity Server (Figure 33).

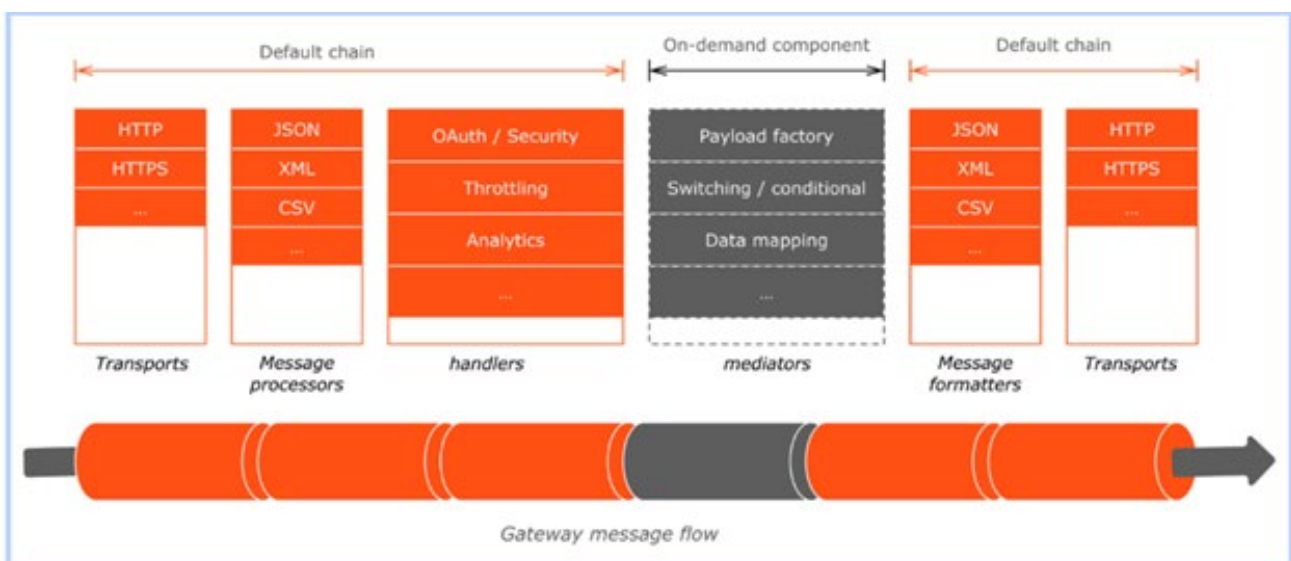


Figure 33: Message Flow

Messages that reach the Gateway are processed as follows:

1. When a request hits the API Gateway, it is received by the Transports module, that is responsible for carrying messages in a specific format. The transport provides a receiver and a sender (for receiving and sending messages accordingly).

³ <http://www.simplecloud.info/>

2. The receiving transport selects a message builder, based on the message's content type, and uses the selected one to process the message's raw payload data and convert it into a common XML, which the Gateway mediation engine can then read and understand.
3. The request is passed through a set of handlers that applies the quality of services on the request message, enforces security, limits request rate, and applies transformations on API requests if applicable
4. After all the requests are routed to the backend endpoint, a message formatter (selected based on the initial message's content type) is used to build the outgoing stream back into its original format based on the message.
5. The transport sends the message out of the Gateway.

In the SB context, the WSO2 API-Manager Gateway will be the only exposed REST interface of SB@Cloud, all other SB@Cloud components will expose their rest interfaces only on a private network that is accessible only to components running in the SB@Cloud. By utilizing this approach, we guarantee that we provide authentication and authorisation to the SB@Cloud services. During the deployment of the SB@Cloud components, each RESTful service of the components that will be exposed will be assigned with roles that the user needs to have in order to invoke the rest service. After the initialisation of the REST services a sequence of how the REST call can be invoked is depicted in Figure 34.

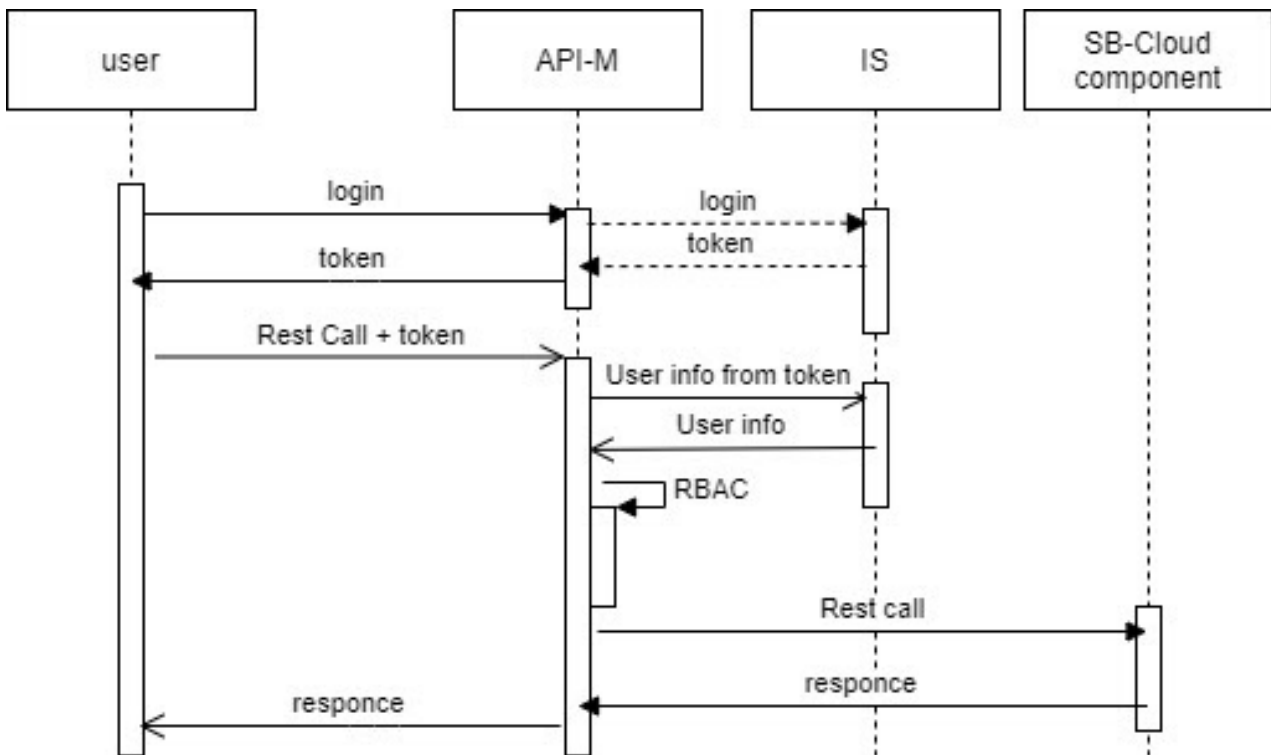


Figure 34: Authentication and Authorisation sequence

1. During the login phase of the user, the user gets an authentication token. As already mentioned as soon as APIM receives a token related call, it redirects it to the IS.
2. Afterward when the user makes a REST call to the SB@Cloud (through the SB@App or the dashboard) it uses also the authentication token.
3. The APIM then will get the users info (claims, roles, etc) from IS.
4. Then based on the defined roles specified APIM will decide if the user can invoke this service. If so, then APIM will proxy the request to the SB-Cloud component.



4.3 Roadmap

1. OpenShift described in Cloud security, will be installed on the SB@Cloud infrastructure. The specification of the infrastructure was described in D2.2.
2. After the successful deployment of OpenShift, the system administrator of the cloud will install the Secure Manager One elements and make the appropriate settings so that Secure Manager One will be the only exposed element to the internet.
3. Following the successful set-up of Secure Manager One, the appropriate roles will be added.
4. After that, each cloud service will be added to the Secure Manager One by adding the specification of this service.
5. Roles will be assigned to each cloud service specification in Secure Manager One. Moreover, for each service that is added to Secure Manager One, a data anonymisation mediator that is developed will be added to each service as described in Message Flow. For more information about data-pseudo anonymisation please see Adhering to Privacy by Design Principles chapter.
6. Apart from the services definition, for other SB service to interact the SB@Cloud services proxied by Secure Manager One, these services need to be registered in Secure Manager One. The SB@Dashboard which is based on web technologies will utilise the user credentials authorisation grant flow, since a user redirection between the identity server and the dashboard was undesired and it may confuse the end users. Moreover, for the SB@App, the best approach is to use the client credentials authorisation grant, since the application can store client credentials securely.



5 Conclusion

This document provides the initial version of the privacy and security mechanisms of the SB@Cloud platform, to encompass detailed descriptions of assets developed, as well as their rationale and interoperability. The implemented solution is based on security-by design approach and comply with GDPR, providing anonymity, confidentiality, privacy and integrity of the data.

After an introductory section overviewing the main security and privacy mechanisms touched by the SB@Cloud platform components, the presentation is structured on a per-component basis. Furthermore, the deliverable takes advantage of submitted deliverables on the set of user requirements D2.1 and the SB architecture to further highlight the relevance of the planned security assets.

This deliverable paves the road to the following versions, in which more features on the aforementioned will be added in order to further strengthen the security and privacy mechanisms.



6 References

- Andrea, I., Chrysostomou, C. and Hadjichristofi, G., 2015. Internet of Things: security vulnerabilities and challenges, 3rd IEEE International Workshop on Smart City and Ubiquitous Computing Applications (ISCC), IEEE, 6-9 July, Larnaca, Cyprus, pp. 180-187.
- Basdekis I., Pozdniakov K., Prasinos M., and Koloutsou K, 2019. Evidence Based Public Health Policy Making: Tool Support, 2019 IEEE World Congress on Services (SERVICES), Milan, Italy, 2019, pp. 272-277, doi: 10.1109/SERVICES.2019.00080.
- Bekara, C., 2014. Security issues and challenges for the IoT-based smart grid, International Workshop on Communicating Objects and Machine to Machine for Mission-Critical Applications (COMMCA), Procedia Computer Science, Elsevier, vol. 34, issue 2014, pp. 532-537.
- Betts, D., Street, C. and Diogenes, Y., 2018. Internet of Things security architecture. Microsoft Azure documentation. Available on-line: <https://docs.microsoft.com/en-us/azure/iot-suite/iot-security-architecture>
- Deshmukh, R. V. and Devadkar, K. K., 2015. Understanding DDoS attack & its effect in cloud environment. Procedia Computer Science. Elsevier Masson SAS, 49(1), pp. 202–210.
- European Parliament, 2016. Regulation (EU) 2016/679, European Union. Available on-line: <http://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX%3A32016R0679>
- EDPB, 2021. Guidelines 01/2021 on Examples regarding Data Breach Notification, online: https://edpb.europa.eu/our-work-tools/public-consultations-art-704/2021/guidelines-012021-examples-regarding-data-breach_el
- Fernandes, D. A. B. et al., 2014. Security issues in cloud environments: a survey. International Journal of Information Security, 13(2), pp. 113–170.
- Hatzivasilis, G., Fysarakis, K., Soutatos, O., Askoxylakis, I., Papaefstathiou, I. and Demetriou, G., 2018. The Industrial Internet of Things as an enabler for a Circular Economy Hy-LP: a novel IIoT protocol, evaluated on a Wind Park's SDN/NFV-enabled 5G Industrial Network. Computer Communications, Elsevier, vol. 119, pp. 127-137.
- Hatzivasilis, G., Papaefstathiou, I. and Manifavas, C., 2016. Software security, privacy and dependability: metrics and measurement. IEEE Software, IEEE, vol. 33, issue 4, pp. 46-54.
- Jansen, W. and Grance, T., 2011. Guidelines on Security and Privacy in Public Cloud Computing. Director, 144(7), pp. 800–144.
- Kocher, P. et al., 2018. Spectre Attacks: Exploiting Speculative Execution. Available at: <http://arxiv.org/abs/1801.01203>.
- Lin, J., Yu, W., Zhang, N., Yang, X., Zhang, H. and Zhao, W., 2017. A survey of Internet of Things: architecture, enabling technologies, security and privacy, and applications, IEEE Internet of Things Journal, IEEE, vol. 4, no. 5, pp. 1125-1142.
- Lipp, M. et al., 2018. Meltdown. Available at: <http://arxiv.org/abs/1801.01207>.
- Neumann, A. J. et al, 1977. "Post-processing audit tools and techniques" (PDF). US Department of Commerce, National Bureau of Standards. pp. 11-1-4.
- NIST, 2020. NIST Special Publication 800-175B Guideline for Using Cryptographic Standards in the Federal Government: Cryptographic Mechanisms, online: <https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-175b.pdf>
- GDPR, 2016. REGULATION (EU) 2016/679 OF THE EUROPEAN PARLIAMENT AND OF THE COUNCIL of 27 April 2016, on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation), online: <https://eur-lex.europa.eu/legal-content/EN/TXT/HTML/?uri=CELEX:32016R0679&from=EN>



ENISA, 2018. Recommendations on shaping technology according to GDPR provisions - An overview on data pseudonymisation, online: <https://www.enisa.europa.eu/publications/recommendations-on-shaping-technology-according-to-gdpr-provision>

Guidelines 01/2021 on Examples regarding Data Breach Notification, Ver 1.0, online: https://edpb.europa.eu/sites/edpb/files/consultation/edpb_guidelines_202101_databreachnotification_examples_v1_en.pdf