An Integrated Solution for Sustainable Care for Multimorbid Elderly Patients with Dementia

WP2: User Requirement Definition and Design of CAREPATH System Architecture

D2.3: Holistic CAREPATH Architecture

Contractual Date of Delivery to the EC: 28 February 2022 (M8)

Actual Date of Delivery to the EC: 28 February 2022

Participant(s): [1]EXYS, [2]**Fraunhofer**, [4]WARWICK, [5]UHCW, [6]SESCAM, [7]UCLM, [8]CITST, [9]SRDC.

Author(s): [2]Yehya Mohamad, [2]Carlos Velasco , [2]Henrike Gappa, [1]Angelo Consoli, [1]Jaouhar Ayadi, [1]Luca Gilardi, [4]Omar Khan, [4]Sarah Lim Choi Keung, [4]Theodoros N. Arvanitis, [5]Timothy Robbins, [6]Pedro Abizanda Soler, [7]Luz Maria Peña Longobardo, [7]Juan Oliva Moreno, [8]Oana Cramariuc, [9]Mert Gencturk, [9]Gokçe Banu Laleci Erturkmen.

Type (P-prototype, R-report, O-other, ORDP-Open Research Data Pilot, DEM-Demonstrator, ET-Ethics): **R**

Dissemination level: PU

Version: 1v9

Total number of pages: 48

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 945169.

Executive Summary

This deliverable presents the holistic CAREPATH architecture developed in Task 2.2 of WP2, which defines the CAREPATH overall architecture and its components, providing solid boards of the platform and allowing in the design, development and deployment phases loosely coupling of its parts. The CAREPATH architecture is divided in several layers comprising the CAREPATH platform, these are top down the presentation layer, the security layer, the application layer, the service layer, the knowledge layer, and the persistence layer. Additionally, the architecture contains links through dedicated connectors to external systems.

Based on the CAREPATH architecture layers components have been identified. These components will form the basis for the different CAREPATH modules, which will be specified in Task 2.3.

Table of contents

List of Figures

List of Tables

Acronyms and Abbreviations

CAREPATH	An Integrated Solution for Sustainable Care for Multimorbid Elderly Patients with Dementia
FHIR	Fast Healthcare Interoperability Resources
H2020	Horizon 2020 Framework Programme
UI	User Interface
CSS	Cascading Style Sheets
HTML	Hypertext Markup Language
SSL/TLS	Secure Sockets Layer / Transport Layer Security
SQL	Structured Query Language
H/HMP	Home/Health Monitoring Platform
AICP	Adaptive Integrated Care Platform
HTTPS	Hyper Text Transfer Protocol Secure
CDSS	Clinical Decision Support Systems
TIS	Technical Interoperability Suite
SIS	Semantic Interoperability Suite
PEP	Patient Empowerment Platform
AAA	Authentication, Authorization and Accounting
API	Application Programming Interface
EWS	Early Warning Services

1. Introduction

1.1 Project information

CAREPATH is a Horizon 2020's funded project (Grant agreement ID: 945169), proposing an ICT based solution for the optimization of clinical practice in the treatment and management of multimorbid older patients with Mild Cognitive Impairment (MCI) or mild dementia. In order to achieve this, CAREPATH will elaborate on a methodology for computer interpretable clinical guidelines and computationally derived best clinical practice for best suitable treatment of this patient group. Thereby, a multidisciplinary care approach is considered, with a focus on the very individual needs of these patients to be translated into personalized care plans for increasing their independence and Quality of Life (QoL).

The CAREPATH project started on July the 1st, 2021 and will end on June the 30th, 2025.

1.2 Document scope

Based on the work done in Task2.1 and previous knowledge of consortium members [1], [2] a holistic system architecture will be presented so that the architecture will: (1) Drive prototype development according to user requirements; (2) Consist of a living document that will be updated during the project according to feedback from users and results of the development and in particular the evaluation WPs; (3) Include different layers describing all components and interrelations between them; (4) Describe the interrelations with external systems e.g. Hospital Information Systems and (5)Consider security, logging, authentication and single sign-on during the holistic design.

The work presented in this deliverable describes all components running in the technical platform.

1.3 Document structure

The deliverable is organized as follows:

- Chapter 1 - provides most importantly information about the scope deliverable
- Chapter 2 - provides an overview of the CAREPATH architecture with a high-level diagram (see Figure 1)
- Chapter 3 - explains in detail all components included in Figure 1
- Chapter 4 – provides conclusions.

2. Overview of the CAREPATH Architecture

Considering all the above, we designed CAREPATH architecture that is described in this section. It can be split into six logical layers: bottom up, the **persistence layer, the knowledge layer**, the **service layer**, the **application layer**, the **security layer,** and the **presentation layer.** Additionally**, connectors** to external world, which are an essential part of the described architecture.

- The **Persistence layer** includes the database hard- and software based on SQL and non-SQL repositories.
- The **knowledge layer** includes the subcomponents for organising the required data and knowledge. These subcomponents are domain knowledge, reports, history, and guidelines/policies.
- The **service layer** includes all the core CAREPATH components: the Home Monitoring, the CDS, the Polypharmacy, Interoperability components.
- The **Application layer** includes all core CAREPATH applications: Adaptive Integrated Care Plan Platform (AICP), Patient Empowerment Platform (PEP) and Home/Health Monitoring Platform (H/HMP).
- The **security layer** is responsible for user authentication, management of user rights according to his role and data encryption for data exchanged through external APIs. The authentication methods that can be used are included in it also.
- The **presentation layer** includes all the user interfaces needed for interaction with the services of CAREPATH. Its main component is the User Interface (UI) manager, which is responsible for the proper visualisation of the results in different devices (e.g. desktop PCs, laptops, mobile devices) and platforms in a unified manner. The user interfaces designed as wireframes, and they are presented in the deliverable D2.5" Specifications of all Components and Customisation Requirements".

These components along with their subcomponents are described in detail in the following sections.

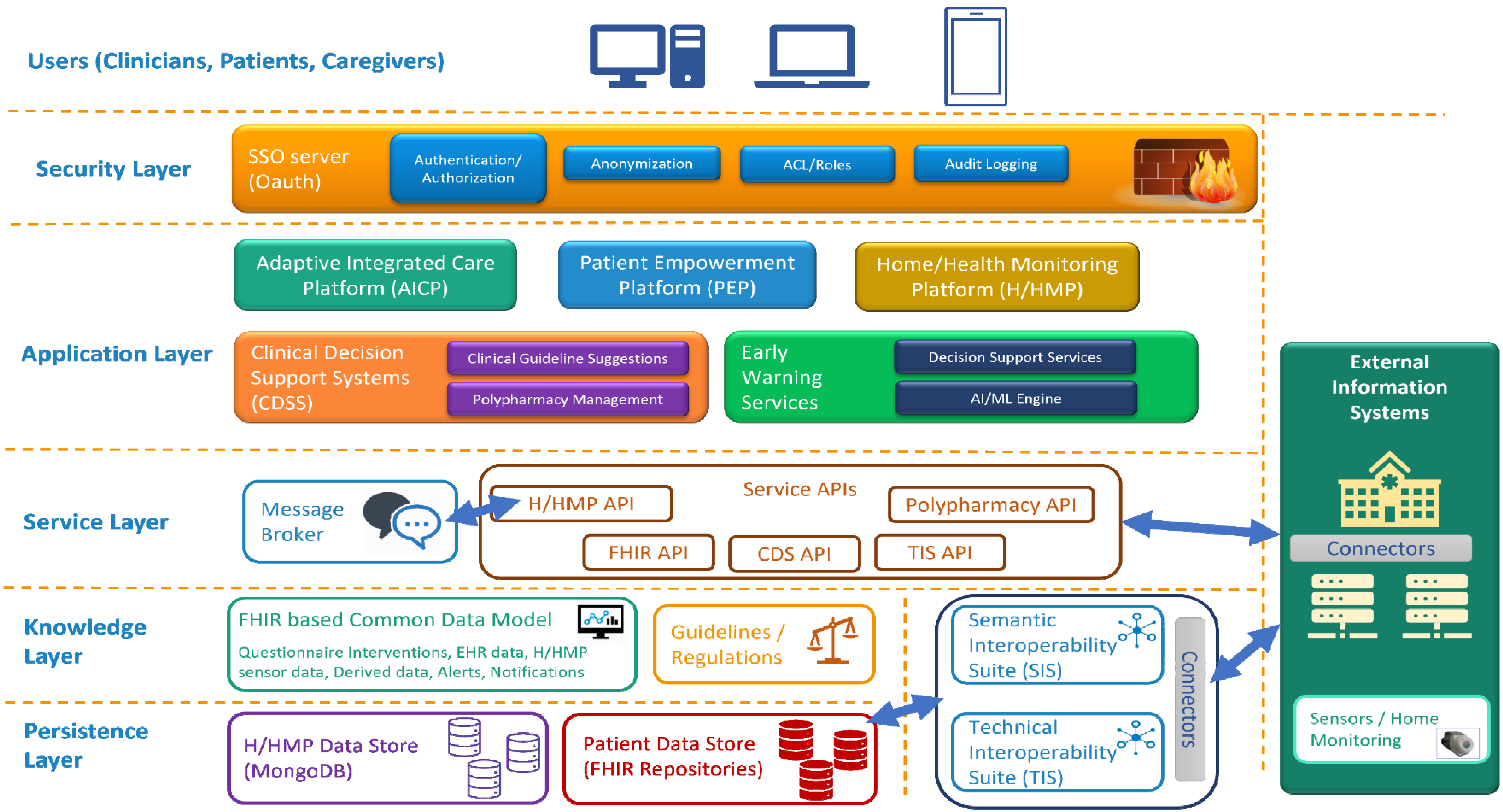


Figure 1: CAREPATH architecture overview

3. Description of CAREPATH Architecture Layers

In this section a detailed description of every layer and its components will be presented. This content serves multiple purposes e.g. creating a common understanding of the envisioned CAREPATH platform within all consortium partners, allowing them to present a joint view to the internal stakeholder of CAREPATH and to the external world. This Architecture will form a communication vehicle between users, designer, developer, and tester of CAREPATH providing them with a joint terminology of CAREPATH parts. As mentioned earlier this architecture will evolve during the project as new knowledge and circumstances will arise during the project.

3.1 Presentation Layer

The Presentation layer includes all the Graphical User Interfaces (GUIs) that enable user interaction with the CAREPATH services. Various GUIs for visualizing the various results by also allowing the users to work through the dashboards that will be developed. Further details as well as wireframes will be provided in deliverable D2.5.

The UI manager component is responsible for the proper visualisation of the data on different devices (e.g. desktop PCs, laptops, mobile devices) and platforms in a unified manner. This is achieved by implementing responsive web interfaces, leveraging web standards like HTML5[1] and CSS3[2], which can adjust the web content to different kinds of digital devices, by detecting their capabilities. The basic principles of responsive web design are:

- Fluid layout: The designer specifies content area width as percentages of the browser window; thus, it adjusts to the user's screen size., providing increased content accessibility.
- Flexible media: Media objects have a default size which hinders fluid-width content areas containing such media to scale below the media object's width. This problem is addressed by CSS's max-width property, which scales media content according to its parent container.
- Media queries: CSS3's media queries enable browsers to provide different styles for different display contexts.

3.2 Security Layer

The security layer is responsible for the authentication, encryption, and access rights management.

3.2.1 Authentication

When we refer to the term authentication, we mean the process by which the user reaffirms his identity. The process starts when a user tries to have access to information, login into a system. In order to gain access, users usually enter a username and a password. This login combination certifies access if the entered information has been assigned to this user.

The most common way to authenticate a user is with a username and password, entered in a form and sent to the web application for verification. Upon first registration with a web application, a username and password can be chosen, or are given. Future authentications depend on the knowledge of these credentials. The username and password are stored by the web application, typically in a database. Another way of authentication uses a client certificate, which is used to create a secure SSL/TLS connection [3], where both client and server are authenticated with their certificates. The server verifies whether the client certificate is valid (e.g., has a valid signature by an expected CA), and can extract user-specific information from the certificate. Client certificates can be explicitly installed in the browser or can be used in combination with electronic identity cards or other smartcards.

Alternatively, to providing authentication, a web application can also depend on a third-party authentication provider to take care of the user authentication process. Examples of large authentication providers are OpenID, Google, and Facebook. A web application simply directs the user to the authentication provider, who takes care of the whole authentication process. After a successful authentication, the application receives some evidence of the authentication, together with the user's details. The exact details depend on

[1] https://www.w3.org/TR/html5/

[2] https://www.w3.org/TR/2001/WD-css3-roadmap-20010523/

the implementation and user's configuration, but typically include a username, a name, an email address, etc.

There are many existing authentication techniques such as Basic Authentication [4], JSON WEB TOKEN (JWT) [5], OAuth 2.0[3], and Bearer Authentication[4].

A simple authentication system built into the HTTP protocol is Basic authentication. The client sends HTTP requests with the Authorization header containing the word Basic word followed by a username: password and base code64. Because base64 is easily decoded, basic authentication should only be used in conjunction with other security mechanisms such as HTTPS / SSL.

Another technique to authenticate a user is using JSON Web Token (JWT) that defines a compact and self-contained way for securely transmitting information between parties as a JSON object. The fact that this information is digitally signed allows us to confirm it and to consider it credible. A public/private key value can be used to sign JWTs. JWT used after the user is logged in, each subsequent request will include the JWT, allowing the user to access routes, services, and resources that are permitted with that token. JWT is widely used by the Single Sign On, as it has small generic exits and its ability to be easily used in different areas. In the CAREPATH platform, we will use JWT for the RESTful communication between CAREPATH components.

Authentication can also be used OAuth 2.0. More specifically, it is an authorization protocol that gives the API client limited access to user data on a web server. The APIs of GitHub, Google and Facebook are particularly useful. OAuth is based on authentication scripts called flows that allow the resource holder (user) to share the protected content from the resource server without sharing its credentials. To do this, an OAuth 2.0 server issues access identifiers that clients can use to access protected resources on behalf of the resource owner.

Another authentication scheme is the Bearer authentication, which also involves tokens. The name "Bearer authentication" can be understood as "give access to the bearer of this token." The cryptic string, which is generated by the server in response to a login request is the bearer token. This Authorization token must be sent by the client when making requests to protected resources. It was originally created as part of OAuth 2.0 but is sometimes also used on its own but always over HTTPS (SSL).

In CAREPATH platform, we will use OAuth 2.0 for user authentication. This process starts when the user accesses the login UI where s/he should give his credentials (username, password). If the user is successfully authenticated, s/he can have access to the CAREPATH platform.

3.2.2 Access rights management

A common security technique for controlling user access to specific parts of a platform is the use of user rights.

By applying security rights, to processes and to users, the site can divide super-user privileges among several administrators. User rights can also be restricted, such as for guest users.

In CAREPATH, we have following main roles: patient / informal caregiver, clinician, and administrator.

Patient / Informal caregiver: They are the main beneficiary of the CAREPATH platform and have access to the patient applications in form of the patient dashboard, the empowerment platform with serious games etc.

Clinicians: They are responsible for the treatment of the patients, creation of patient-specific care plans, and follow up the data of the patients on the clinician dashboard.

Administrator: His/her task is the management of the CAREPATH platform. He can manage the available services. S/He is responsible to approve or disapprove users request for signing up in CAREPATH and add them with their respective role. In addition, s/he handles the overall system storage, network, version updates, etc. and s/he can access to usage statistics of the CAREPATH components.

In CAREPATH, as we have clear roles as presented above, we can use role-based access control (RBAC) [5] to restrict system access to authorized users. Each user on the platform has an assigned role, and each role has a set of access permissions to resources/UIs. The UIs and the mapping with the corresponding roles are presented in the deliverable D2.5.

---

3 https://oauth.net/2/

4 https://jwt.io/introduction/

3.2.3 Communication Encryption

Data encryption translates data into another form, or code, so that only people with access to a secret key (formally called a decryption key) or password can read it and is widely used on the internet to ensure the sanctity of user information that is sent between a browser and a server. Through cryptography we can prove that the information is authentic, and its source is what it claims. It can be used to verify the origin of a message and confirm that it has not been altered during transmission.

In the CAREPATH platform, we will use encryption for the data that exchanged between CAREPATH and the external tools. The most widely used protocol is the Secure Sockets Layer (SSL) and Transport Layer Security (TLS) [3], through it, a secure channel is provided between two machines operating over the Internet or an internal network. Technically, SSL is a transparent protocol that requires little interaction from the end user when establishing a secure session.

3.2.4 Anonymization & De-identification Tool

The Anonymization & De-identification Tool (ADT) will be responsible for handling the privacy challenges on sensitive health data by applying several data de-identification and anonymization techniques. It is designed to work on an HL7 FHIR API so that it can be used on top of any standard FHIR Repository as a data de-identification, anonymization, and related actions toolset. The component is expected to access FHIR resources, present metadata, guide the user of the tool about the configuration to be applied and then output the processed FHIR resources. The resulting FHIR resources will be de-identified/anonymized based on the configurations provided.

ADT will be composed of four components, which are Metadata Analyzer, Algorithm Suggestion Service, Configuration Manager, and De-identification Engine:

- **Metadata Analyzer:** It will be the component for extracting the attributes from FHIR resources and enabling the annotation of them as Identifier, Quasi-identifier (QI), or Sensitive Attribute (SA).
- **Algorithm Suggestion Service**: Algorithm Suggestion Service suggests de-identification algorithms for each attribute based on its data type and annotation. Possible algorithms are:
    - *Pass Through*: No change.
    - *Substitution*: Replace value with another value (meaningful or meaningless)
    - *Recoverable Substitution*: Substitute with pseudorandom value with the choice of length
    - *Fuzzing*: Replace the value with the one produced as a result of a statistical algorithm or code set produced randomly (Adding noise)
    - *Generalization*: Making a value less specific
    - *Date Shifting*: Shifting dates by random intervals
    - *Redaction*: Deleting the value
- **Configuration Manager**: Some algorithms will require further configurations to be made. For instance, SAs may have some rare values and those rare situations should be indicated because they may cause identification of patients easily. In addition, ADT supports anonymization methods such as k-anonymity and l-diversity. For these methods to be applied, the k value and l value should be provided. Configuration Manager enables user to perform all these kinds of configurations.
- **De-identification Engine**: After all the configurations are done, De-identification Engine will de-identify data according to those configurations. When de-identification process will finish, new de-identified FHIR resources will be versioned and labelled accordingly and then saved back to the FHIR Repository.

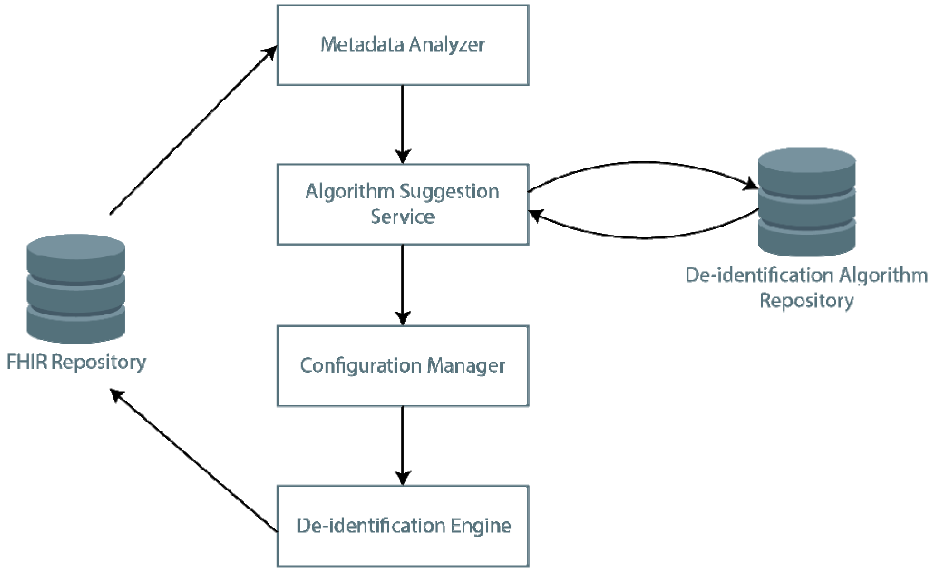


Figure 2: Architecture of Anonymization & De-identification Tool (ADT)

3.3 Application Layer

In this section all planned CAREPATH applications will be described in detail including their position within the architecture and interactions with other applications or subcomponents of the CAREPATH architecture.

3.3.1 Adaptive Integrated Care Platform (AICP)

The Adaptive Integrated Care Platform (AICP) is one of the core components of the CAREPATH system, it will facilitate collaborative management of the care of the patients with multimorbidity. It will be the direct interface to care team members, for defining, updating, reconciling, sharing care plans, utilization of clinical decision support modules supporting these operations, creating diet and exercise plans as part of care plans, showing the patient data retrieved from local care systems and the Patient Empowerment Platform. It will provide an easy to navigate dashboard for care team members to see patient's medical history along with his/her care plan history, review care plan progress and patient generated data as a dashboard. AICP will be directly connected with the Clinical Decision Support Systems (CDSS), Early Warning Services (EWS) and the Patient Data Store which is a FHIR Repository.

AICP will be composed of three components which are the AICP UI, AICP Engine, and Subscription Engine.

- **AICP UI:** It will provide a GUI for care team members to collaboratively create update, share, and export personalized care plan definitions.
- **AICP Engine:** AICP Engine will be the core sub-component of AICP component coordinating the internal business logic of the functionalities provided by AICP, facilitating the choreography of the communication between other subcomponents within AICP component. It will be the main interface to the main graphical interfaces of AICP, i.e., AICP UI, providing following APIs:
    - *CarePlanAPI:* This interface will serve to AICP UI, enabling the creation, update, view and deletion of care plan and its constituting entities managed within AICP. This interface will also serve to export the care plan snapshot in a standard based format and to share the care plan snapshot with all care team members, local care systems and Patient Empowerment Platform.
    - *CareTeamAPI:* This interface will serve the AICP UI, enabling editing, viewing care team member list, and also inviting, adding, removing care team members.
    - *CareTeamMemberDashboardAPI*: This interface will serve the AICP UI, enabling care team members to check their patients with upcoming or missing appointments, the tasks assigned to them, new messages and notifications received from patients and care team members.
    - *CDSSAPI:* This interface will serve the AICP UI to call specific clinical decision support modules for risk calculation, retrieving suggestions for interventions based on clinical guidelines, and polypharmacy management.
    - *EarlyWarningAPI:* This interface will serve AICP UI to call specific early warning modules for monitoring deviations in intrinsic capacity and dementia profile of the patient, deviation

of the patient from their target goals and his current treatment plan activities (adherence), potential adverse effects such as treatment conflicts and risk stratification and early warning of long-term outcomes.
  - *MessageAPI:* This interface will serve the AICP UI, to enable the care team members to view, read, write, tag, send messages.
  - *NotificationAPI:* This interface will serve Subscription Engine component to receive notifications about the events fired based on the pre-defined event subscriptions. It will also serve to AICP UI to enable care team members to review notifications and early warnings.
  - *PatientDataAPI:* This interface will serve the AICP UI, to enable the MDT members to view patient summary overview and provide updates when necessary.
- **Subscription Engine:** The aim of Subscription Engine will be to manage the subscriptions to specific resources or events, and generation of notifications based on subscription criteria to be forwarded to addressed actors. After receiving an event criterion from Patient Data Store or Early Warning Services, it will run subscription criteria and will push notification messages to the subscribed components via the specified notification interfaces.

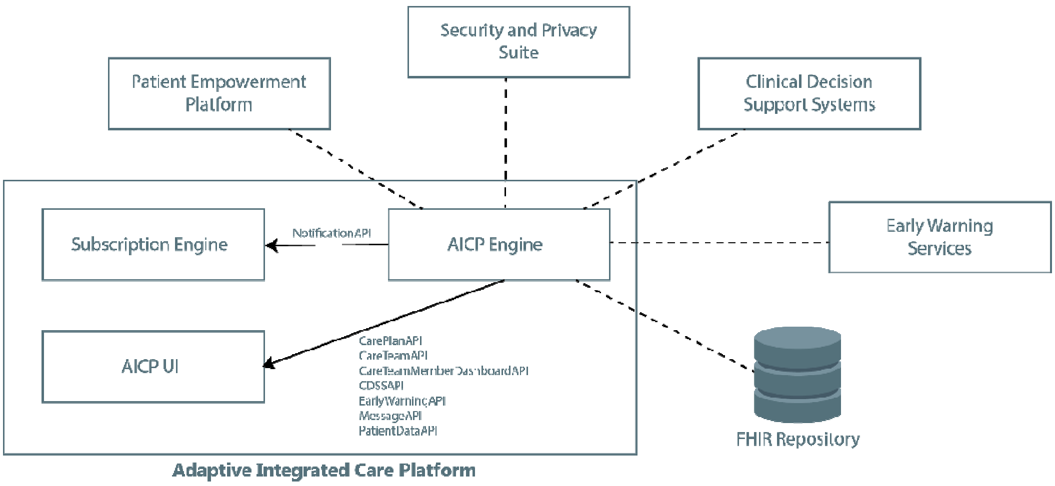


Figure 3: Architecture of Adaptive Integrated Care Platform (AICP)

3.3.2 Patient Empowerment Platform (PEP)

Patient Empowerment Platform (PEP) will be the main interface of patients and their informal care givers. Through PEP they will be able to review their care plan, display daily activities including exercises, diet plan, medication intake, home measurements, appointments, and referrals, provide feedback about care plan goals and activities, report symptoms and patient reported outcome measures (PROMs), access educational materials, exchange messages with care team members, and retrieve notifications and reminders.

PEP will be composed of three components, which will be PEP Mobile UI, PEP Web UI and PEP Engine.

- **PEP Mobile UI:** PEP Mobile UI will be provided to patients.
- **PEP Web UI:** PEP Web UI will be provided to informal caregivers. Both Mobile and Web UI will provide GUIs for patients and informal caregivers to follow patients' care plan activities, provide feedback on them, interact with care team members, and be notified with alerts and reminders.
- **PEP Engine:** PEP Engine will be the core sub-component of PEP component coordinating the internal business logic of the functionalities provided by PEP, facilitating the choreography of the communication between other subcomponents within PEP component. It will be the main interface to the main graphical interfaces of PEP, i.e., PEP Mobile UI and PEP Web UI, providing following APIs:
  - *CarePlanAPI:* This interface will serve to PEP Mobile UI and PEP Web UI, enabling view of care plan and its subcomponents. It will also facilities providing feedback about care plan components and marking care plan activities as performed or not.
  - *MessageAPI:* This interface will serve to PEP Mobile UI and PEP Web UI, to enable the patients and informal caregivers to view, read, write, tag, send messages.
  - *QuestionnaireAPI:* This interface will serve to PEP Mobile UI, to enable the patients to view, and complete questionnaires (PROMs) assigned to them as a part of their care plan.

- *MedicationAPI:* This interface will serve to PEP Mobile UI, to enable the patients to indicate the medications taken as part of their care plan or apart from it.
- *EducationMaterialAPI:* This interface will serve to PEP Mobile UI, to enable the patients to view the list of educational material as a part of their care plan and also related with their diseases and select and review one of them.
- *ReminderAPI:* This interface will serve to PEP Mobile UI and PEP Web UI, enabling creation and view of reminders to the patients.
- *ExerciseAPI:* This interface will serve to PEP Mobile UI, to enable the patients to view and complete exercise activities assigned to them as a part of their care plan.
- *DietPlanAPI:* This interface will serve to PEP Mobile UI, to enable the patients to view and complete dietary plan assigned to them as a part of their care plan.

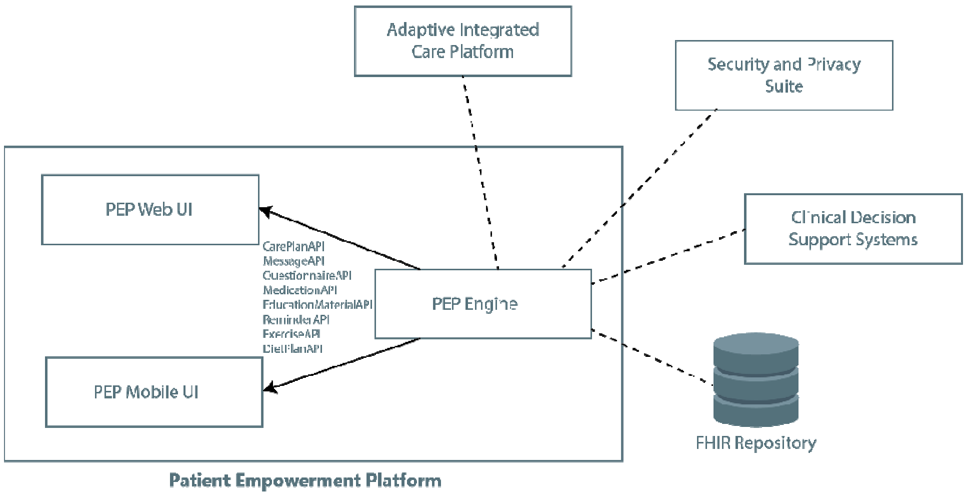


Figure 4 Architecture of Patient Empowerment Platform (PEP)

3.3.3 Home/Health Monitoring Platform (H/HMP)

The Home/Health Monitoring Platform (H/HMP) will be the CAREPATH component responsible for collecting, filtering and integrating data coming from health sensors and home automation devices, The H/HMP queries the Patient Data Store (FHIR Repositories) via the FHIR API, consolidates the patients' information (EHR data, information coming from PEP, data coming from external information systems, etc.) thus obtained with sensor data, and stores results in the H/HMP Data Store (Persistence Layer). Next, various other modules in the system will interact with the H/HMP by means of the H/HMP API (Service Layer) for their own processing purposes, specifically, but not only, the FHIR based Common Data Model (which stores in the FHIR Repositories) and the Early Warning Services (in particular the AI/ML Engine).

In CAREPATH, the definition and the design of H/HMP are performed in WP2, while its implementation will be carried out as an activity of Task 4.1.

3.3.3.1 Architecture of the H/HMP

The Home/Health monitoring platform operates at the Application Layer and, although it is a well decoupled subsystem, it interacts with modules residing in other layers, and consists of many components, as shown in Figure 5.

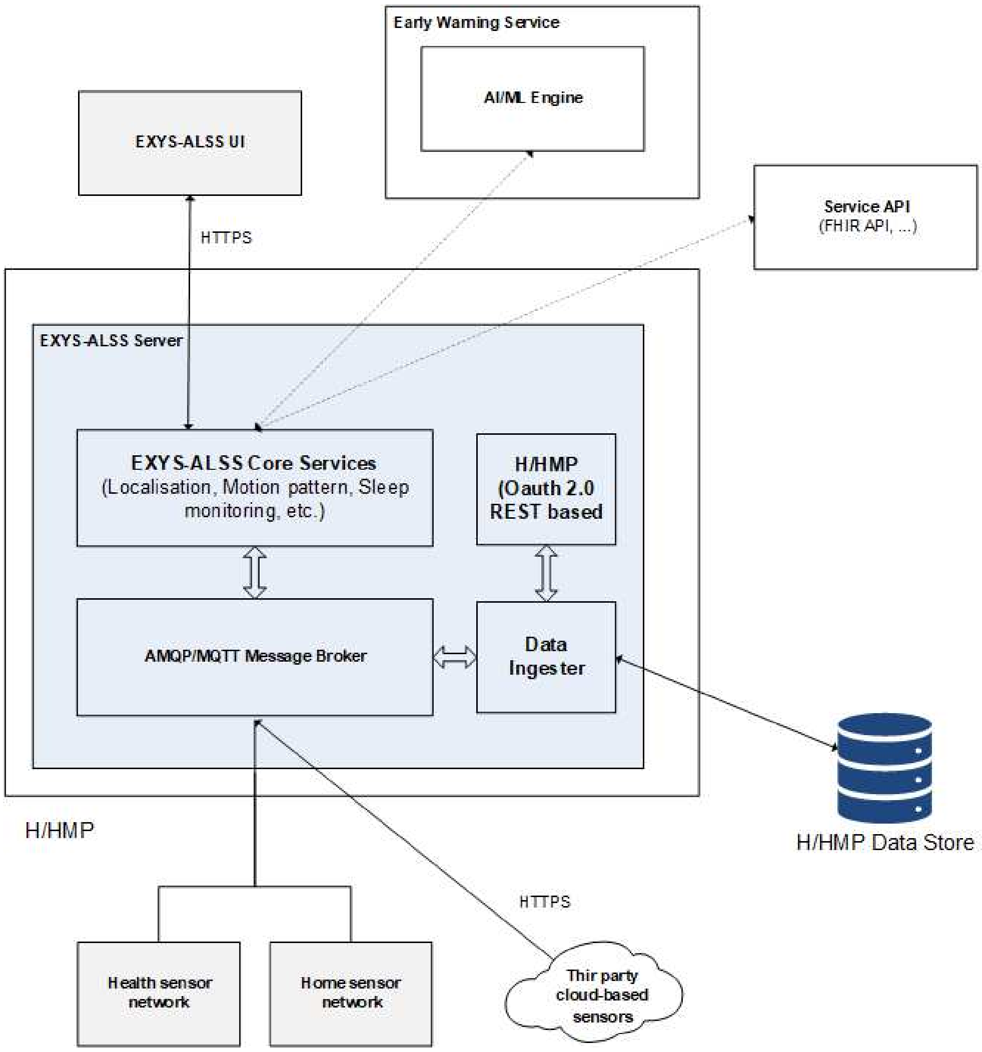


Figure 5: Architecture of the Home/Health Monitoring Platform (H/HMP)

The H/HMP will leverage on the EXYS-ALSS (Ambient Living Support System) platform. This includes the EXYS-ALSS Server which is in charge to orchestrate all the information flow for the H/HMP (i.e., collect data from the extern, process them, deliver the results to the other CAREPATH subsystems).

The H/HMP will include a set of devices installed in patients' home and wireless network link. Among the devices that will be installed there:

- The home automation sensor network: it will include a set of domotic and environmental sensors (temperature and light sensors, movement sensors, water and smoke detectors, smart plugs, etc.), connected in network.
- The health/medical sensor network: a network of wearable health and activity devices (blood pressure, pulse, oximeter, weight scale, glucose meter, fitness smartwatches, etc.).
- The Smart Network Gateway (EXYS9200-SNG): it is an electronic equipment at the heart of the H/HMP, implementing different communication protocols (ZigBee, Z-Wave, BT, BLE, Wi-Fi, etc.), acting as collector of data coming from the home automation and health sensor networks and forwarding the information to the EXYS-ALSS Server.

Some sensors will be directly connected to the EXYS9200 gateway and data is directly collected from the device and stored on EXYS servers (in the Eclexys datacentre or deployed on pilot sites, based on decisions mainly driven by privacy governance).

For some other devices, data will be stored at the manufacturer's cloud servers, therefore they don't pass through the EXYS9200-SNG gateway, instead will be available by querying manufacturer's cloud, previous registration.

3.3.3.2 The Smart Network Gateway (EXYS9200-SNG)

The H/HMP device will be in charge to collect data from the two different sensor networks (the health sensor network and the home/domotic sensor network). In the frame of the project, thus, the EXYS9200-SNG gateway will be enhanced and adapted to the CAREPATH needs, and will act as an interference-shielded, highly performing interface gateway between the raw data coming from the devices/sensor networks.

Interfacing a wide range of specific sensors/devices platform to the CAREPATH ecosystem will be delegated to "gateway services", either implemented as physical devices or micro services running on the gateway. The gateway services handle all low-level hardware, communication, and protocol specific details of interfacing a specific device. It is also the responsibility of the gateway service to enumerate devices and provide device specific setup and configuration interfaces when needed. Moreover, it acts as an intermediate layer aiming at clearly separating the sensor/devices context from the rest of the CAREPATH ecosystem, thus increasing its modularity and very loose-coupling character.

The EXYS9200-SNG has also local processing capabilities and is equipped with a performing database dedicated to local storage and processing needs.

Finally, the EXYS9200-SNG system comes along with is a complete framework (EXYS-ALSS) consisting of several components:

- A set of peripheral sensors and home automation devices allowing
    - monitoring health parameters;
    - detection of events (e.g. presence, movements, …), status of electrical infrastructures in the environment (status, power consumption, …);
    - measurements of environmental parameters (temperature, light, air quality, …);
    - activation of environmental devices (power switches, …);
- A technology automation and interfacing management layer:
    - based on an open-source software solution with native plugins
    - extended with ad-hoc plugins developed by EXYS to expand the gateway
- Forwarding service
    - this is the layer in charge of collecting all data and submitting raw measurements to the EXYS-ALSS Server;
- EXYS-ALSS Server, consisting of:
    - a server managing the back-end datastore (the H/HMP Data Store) for all collected data as well as derivative information that may be generated by the system in automated form or through the operation of care givers, also operable as single server or in cloud architectures;
    - the server infrastructure allows managing user profiles (end users or care takers, care givers, system administrators, …)
    - an API implementation serving as middleware to authenticate and interact with the data stored in the database;
- EXYS-ALSS UI
    - Adaptive Web application user interface that provides a complete interface to overview measures and operate one or more setups.

Figure 6 presents a description of the enhanced EXYS9200-SNG Software Stack Layers architecture.

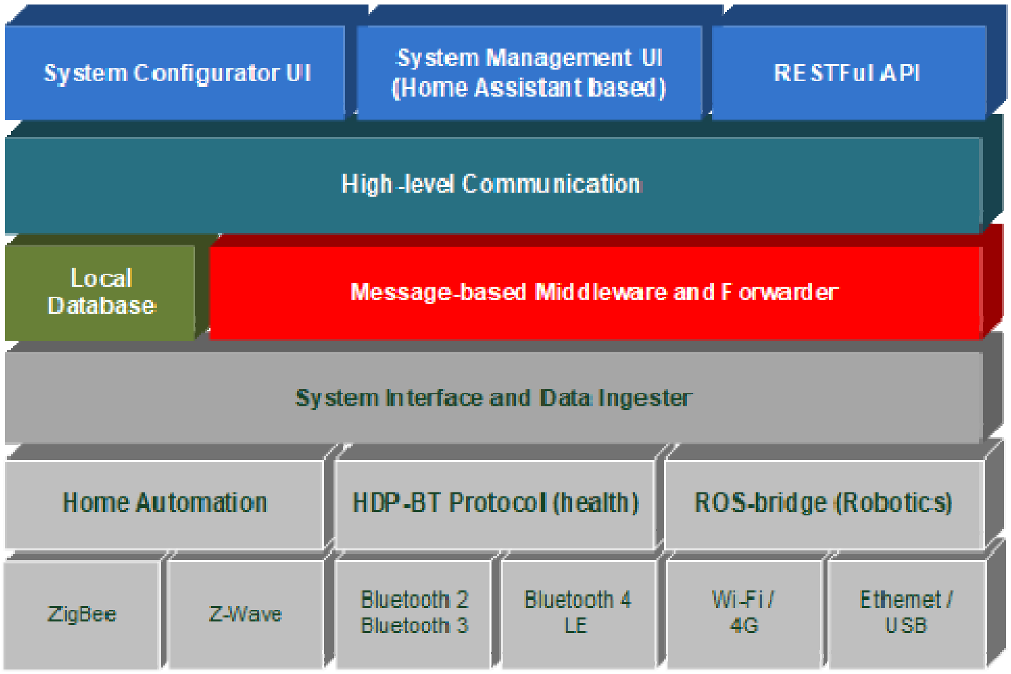


Figure 6: EXYS9200-SNG Software Stack Layers

The System Interface and Data Ingester layer is an intermediate layer has two tasks: firstly, it saves the collected data in a local database; seconds it passes the collected and aggregated data to the Message-based Middleware. This last, an AMQP/MQTT Message broker, enqueues the ingested information and forwards them to the High-level Communication layer. Eventually, the High-level Communication layer is in charge to dispatch the data to the EXYS-ALSS Server.

The two applications, also developed at EXYS premise using innovative Web-based technologies and laying on top of the software stack are web applications accessible from a common web browser (Google Chrome, Mozilla Firefox, Microsoft IE, …). The System Configurator UI allows the operator to setup and manage the gateway and the devices. The System Management UI (based on the Home Assistant GUI) allows the user to configure home device parameters, to check device information and to send device messages (data/commands) to(/from) a central server.

Communication with the EXYS9200-SNG is also possible through a securitized RESTful API.

3.3.3.3 Health monitoring subsystem

The EXYS9000–SNG is compliant with the Bluetooth Health Device profile (HDP-BT), which supports the vast majority of existing e-Health/medical equipment. The BT 4 LE (Low Energy), implemented by means of the module, is particularly adapted to applications where wearable devices are needed. Encrypted communication guarantees secured data communication between the gateway and the devices.

The EXYS9200-SNG enhanced data collector's event manager module gathers raw data coming from health sensors network mainly thorough the BLE module (moreover Z-Wave, ZigBee and other communication protocols are already available for interaction with medical devices). It then passes these data to the Processing Layer through specific software bindings. The communication is bidirectional, so the operator can also act on the EXYS9200-SNG interface to send configuration commands to health devices.

The CAREPATH system will integrate through the EXYS9200-SNG gateway a set of health and activity devices, some of which are:

- Blood pressure meters
- Weight scales
- Pulse oximeters
- Sleep monitoring mats
- Fitness/activity smartwatches

3.3.3.4 Home monitoring subsystem

The EXYS9200-SNG is capable to collect data from the home sensor network and forward them to the other subsystems of the CAREPATH framework.

The EXYS9200-SNG domotic and environmental devices, mainly ZigBee and Z-Wave based, comprises among others:

- Motion detection sensors
- Environmental temperature and humidity captors
- Environmental luminosity sensors
- Door / window Trigger sensors
- Air quality (CO2, smoke) sensors
- Flood detectors
- Smart Power plugs

The EXYS9200-SNG management interface (which is a Web UI). enables to configure and interact with the sensor infrastructure, to monitoring the parameters' values acquired from sensors (reported in graphical and tabular form), to perform basic computation and statistical analyses on the data.

3.3.3.5 Services enabled by the H/HMP

Among raw health, activity and environment data, the H/HMP makes available a set of services.

- Person indoor localization (high accuracy and low accuracy)
- Indoor objects localization
- Mobility pattern detection
  - indoor wandering patterns: looping, pacing, random, direct
  - speed (gait)
- Person outdoor localization
- Outdoor Geofencing
- Sleep monitoring and analysis
- Fall detection
- Smart Rule Engine and Alerting

3.3.3.5.1 Indoor localisation (high accuracy localization)

The high accuracy localization system is an infrastructure-based Ultra-Wideband (UWB, IEEE802.15.4a[5]) depending on the system functional architecture is shown in Figure 7. The system infrastructure is composed of anchor nodes, tags and a system controller. Anchor nodes perform measurements of signals transmitted by tags. Results are sent to the system controller over Wi-Fi links where tags' positions calculations take place. Localization data are sent to the EXYS9200-SNG and forwarded to the rest of the CAREPATH framework.

[5] https://standards.ieee.org/ieee/802.15.4a/3571/

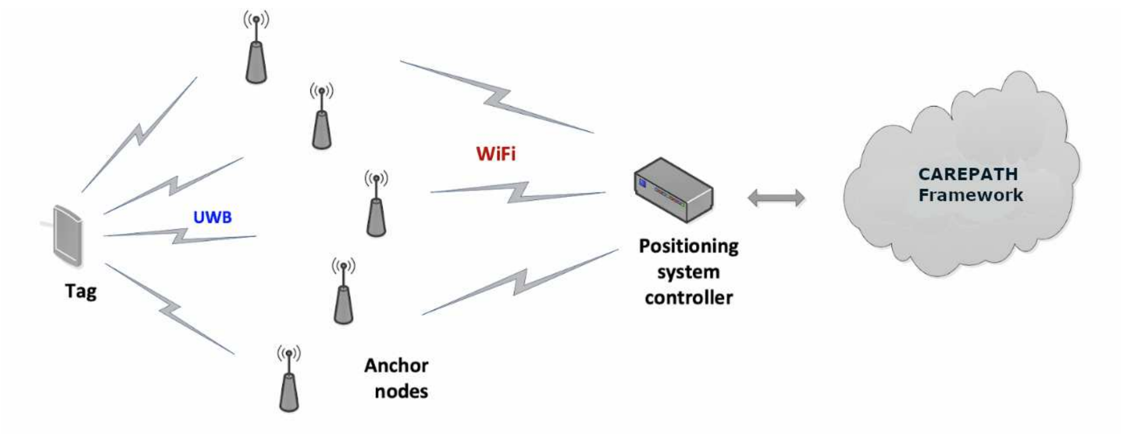


Figure 7: UWB Indoor Localisation architecture

Person Localisation

The anchor node (UWB interface used for positioning and reception of tag's data. It conforms to the IEEE802.15.4a standard) measures time of arrival of UWB packets transmitted by system tags and performs additional measurements enabling anchor nodes synchronization. The device is equipped with MEMS sensors: an accelerometer and an atmospheric pressure meter. The accelerometer is used for detection of node displacement, the barometer can be used for determination of tags altitude (these are used for fall detection purpose). The anchor node sends positioning results to the system controller and relays data received from the tags (e.g., messages related to falling down). Position accuracy under the meter can be achieved.

Object Localisation

Having high accuracy positioning capabilities, the UWB infrastructure is also used to localize objects (glasses, keys, purses, etc.)

- Tags attached to various objects periodically send advertisement packets.
- The anchor nodes measure signal levels and sent results to the system controller where tags' positions are calculated.
- The user selects an item on the tablet screen and the server sends localization request message to the system controller.
- The controller responds with object position.
- In parallel the system controller sends the start alerting message to a tag attached to the chosen item. The tag starts to blink and beep, which helps the user to localize it in the indicated room. Moving the found tag causes it to cease signalling.

3.3.3.5.2 Indoor positioning (low accuracy localization)

Indoor positioning detects in which room, roughly, the user is present.

Developed by EXYS internally, it is based on an infrastructure of anchors and user tags provided by smartwatches, bracelets, or portable devices.

The algorithm estimates the position (with several meters of approximation) using RF ranging based on signal power of anchors and tags. The accuracy is in the order of 5 meters.

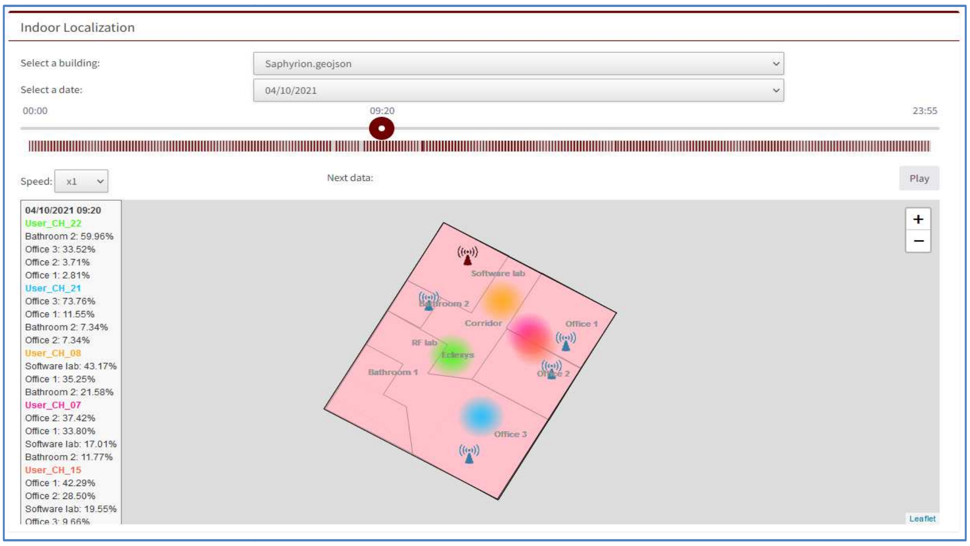


Figure 8: Map from the RF ranging indoor positioning shown in the EXYS-ALLS UI

3.3.3.5.3 Mobility pattern detection (wandering)

Wandering is one of the most frequent, problematic, and injurious behaviours for people with dementia, which is closely associated with adverse events such as falling or getting lost. Wandering is a typical behaviour of the people living with dementia. As the disease progresses, the patients become disoriented in time and space, and they feel lost. This means continuous concern for the caretakers of the patient, so there are a lot of ongoing research to monitor the behaviour of the patients and alert the relatives in case of wandering[6].

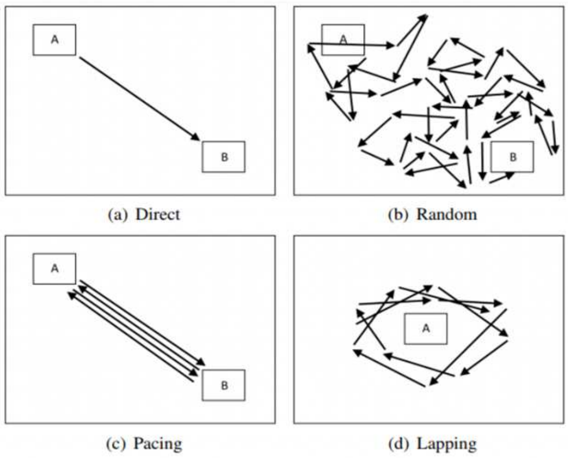
The wide-spread definition of wandering in this context is the following:

"a syndrome of dementia-related locomotion behaviour having a frequent, repetitive, temporally-disordered, and/or spatially disoriented nature that is manifested in lapping, random, and/or pacing patterns, some of which are associated with eloping, eloping attempts, or getting lost unless accompanied".

- Researchers tried to formulate the event of wandering with regards to locomotion. The most used papers classify movement patterns into 4 different categories (.

Figure 9). These are the following:

- **Direct:** locomotion from a point to a destination along a straightforward path without significant indecision. Direct movement is the only pattern which is considered non-wandering.
- **Pacing**: back and forth locomotion between two points, at which directional heading is reversed.
- **Lapping**: circuitous locomotion revisiting, at least, three points sequentially along the path with several directional changes.
- **Random**: locomotion along an aimless path with multiple changes in direction and several indecisions at any point along the path.



[6] A. Hammoud, M. Deriaz, D. Konstantas, Wandering Behaviors Detection for Dementia Patients: a Survey, https://www.researchgate.net/profile/Abbass_Hammoud/publication/325466176_Wandering_Behaviors_Detection_for_Dementia_Patients_a_Survey/links/5b0fb276a6fdcc80995bebe4/Wandering-Behaviors-Detection-for-Dementia-Patients-a-Survey.pdf?origin=publication_detail, last accessed October 2018.

Figure 9. Martino-Saltzman movements

Wandering pattern analysis is performed using an adaptation of the well-known Vuong algorithm[7]

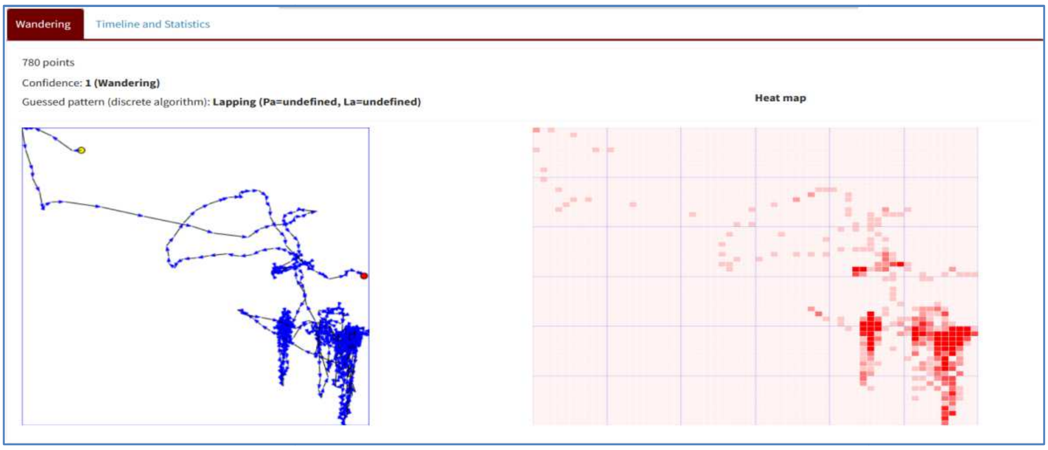


Figure 10: Wandering detection shown on the EXYS-ALSS UI

3.3.3.5.4 Outdoor localisation and geofencing

A typical symptom of dementia is spatial disorientation. The purpose of the outdoor geofencing service is to provide service for monitoring and limiting the motion area of the user.

The usual territory visited by the user can be defined on a map by drawing a polygon.

When the user crosses the boundary polygon an alarm message is issued by the system in order to warn the caregivers or relatives about the unexpected behaviour of the user.

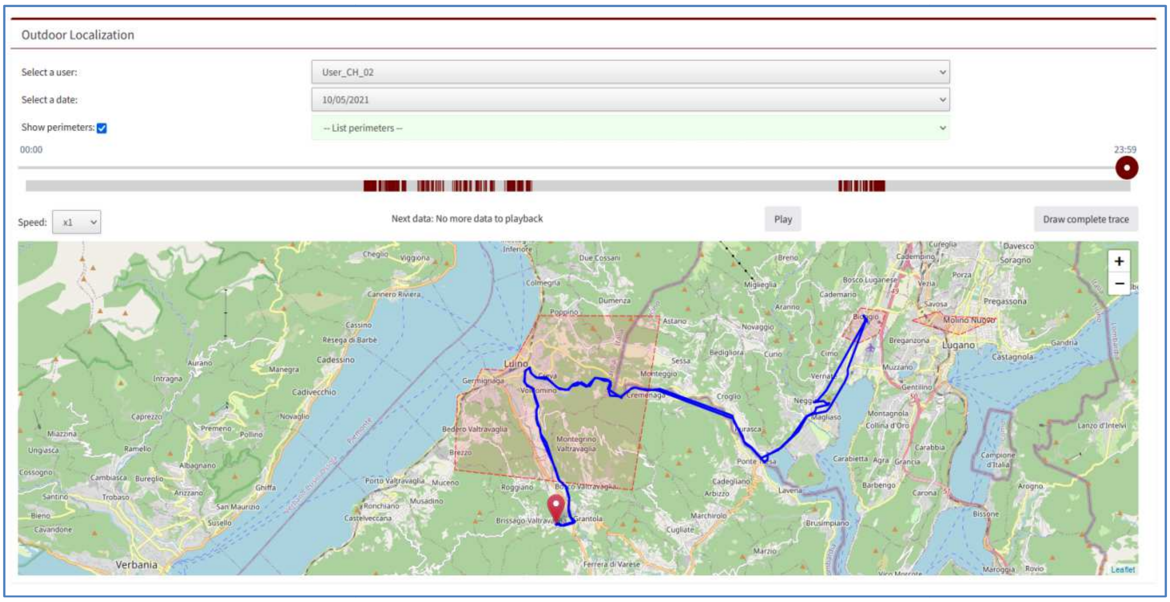


Figure 11: Outdoor localisation and geofencing on the EXYS-ALSS UI

3.3.3.5.5 Sleep monitoring and analysis

The tasks of the sleep monitoring and analysis service are the following.

[7] Automated detection of wandering patterns in people with dementia, Nhu Khue Vuong Beng, Syin Chan PhDa, Chiew Tong Lau PhDa, Gerontechnology 2014;12(3):127-147; doi:10.4017/gt.2014.12.3.001.00

- Sleep stages (light, deep, REM, awake) are continuously monitored using wearable devices.
- Sleep derived parameters and regression analysis are computed using Machine Learning algorithms (linear and polynomial regression, support vector machine, random forest).

Figure 12: Some sleep analysis outcomes

Using the data collected and processed by different machine learning algorithm[8], the correlations between sleep parameters are investigated in order to detect sleep disorders for people with dementia, e.g. progressively less REM (Rapid eye movement sleep) sleep throughout the night, as well as an increase of the night-time awakenings.
Figure 13 presents an example of a learning session of REM vs duration of sleep and awake vs duration of sleep using linear regression model.

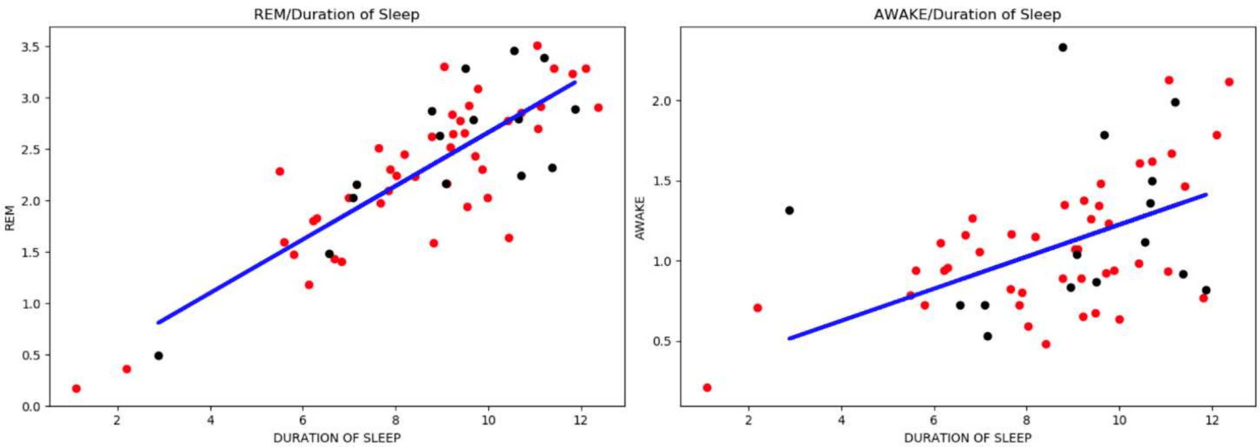


Figure 13. Training and testing REM vs duration of sleep, awake vs duration of sleep

In this case sleep data from 45 days are used for the training phase and the rest of 15 days represents the testing data. The blue line represents the learning model (based on red points). Black points are the testing values. In case of REM vs duration of sleep, the testing error is 0.19 and for the awake vs duration of sleep is 0.26.

For each user, a personal profile is created by extracting a set of rules from corelations learnt from different sets of sleep parameters

3.3.3.5.6 Fall detection

Another typical signal of dementia and decreasing intrinsic capacity is the increasing rate of falls. Although the fundamental role of the UWB indoor localization system (section 3.3.3.5.1) consists in delivery of

[8] In CAREPATH, a powerfull multi-core workstation will be dedicated to batch-running the AI/ML algorithms.

localization data, the tags are able to deliver additional information on monitored person behaviour. MEMS sensors embedded in the tags allow for detection of person's activity and falling down events.

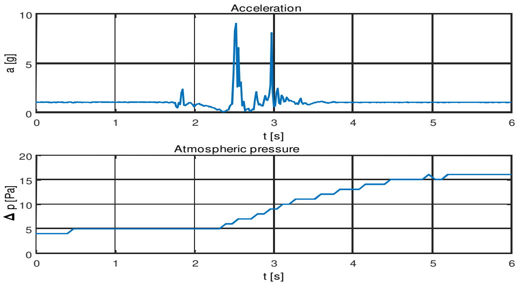


Figure 14. Acceleration and barometric (MEMS-based) fall detection

The fall detection procedure consists in analysis of acceleration and atmospheric pressure changes. The occurrence of three events is searched by the fall detection procedure:

- free fall event (acceleration drops to zero in such case),
- high-g event ((high acceleration values are recorded in case when the person is hitting the floor),
- atmospheric pressure change; increase in pressure value by a few Pa (decrease of altitude by 1 m corresponds to 11.3 Pa pressure increase) may result from the fall.

There are no devices and algorithms that can detect the fall with 100% probability (sensors' results recorded when the monitored person dropped the tag may resemble ones recorded during a real fall). Therefore, the tag not only sends information on the detected fall but also sends data informing on the fall probability. Relation between particular events occurrence and the likelihoods shown in Table 1.

Table 1: Likelihoods of fall event

	Event occurrence		
Fall likelihood	**Free fall**	**High g**	**Atmospheric pressure increase**
high	YES	YES	YES
medium	NO	YES	YES
low	YES	NO	YES
low	NO	NO	YES

3.3.3.5.7 Smart Rule Engine and Alerting

The H/HMP will also provide a Rule Engine (RE), allowing to create both basic and complex rules, based on parameters coming from multiple sources (health, environment, etc…). The RE will be an adaptation of the EXYS-ALSS Rule Engine for the needs of the CAREPATH project pilots.

Rules can combine data coming from different sources (sensors, personal information, patient EHR data, etc.), allowing the use of logical operator (AND, OR, NOT), relational operators (equal, greater than, greater equal than, less than, less equal than, between, etc.), arithmetic and mathematical functions, statistical functions on aggregated data, both for "real-time" and historical data.

Rules also include meta-information, such as:

- the rule sender and recipient,
- the rule priority (HIGH, LOW, MEDIUM),
- a datetime range of validity,

- a possible recurring timeline.

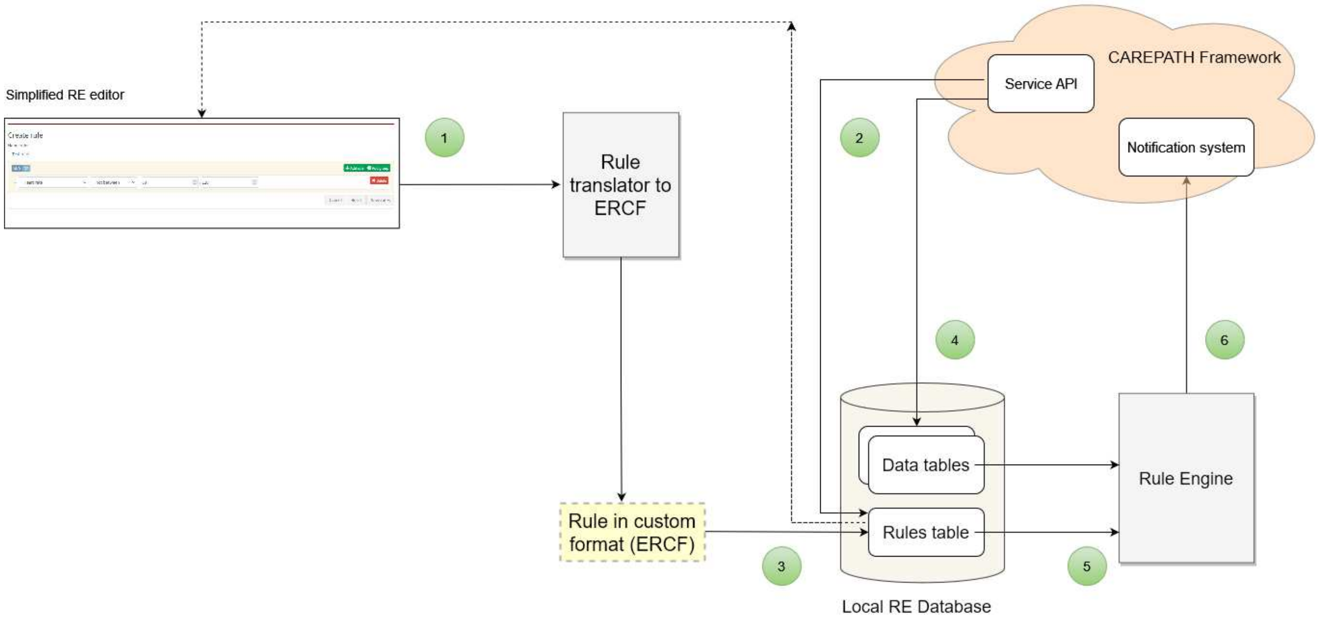


Figure 15. Smart Rule Engine

Figure 15 shows a flow chart of the RE.

1. A simplified RE GUI editor, available in the EXYS-ALSS User Interface, can be used by the operator (must have EXYS-ALSS admin credentials) to insert and associate simplified rules to the patients. The simplified rules are translated in a custom format (EXYS Rule Custom Format, ERCF).
2. More complex rules (already in ERCF format) are provided, along with measurement data, from the CAREPATH data stores through the Service API, and persisted to a local RE database (Rules Table).
3. The simplified rules translated in ERFC are also persisted in the local RE database.
4. Information coming from the CAREPATH Patient Data Store (FHIR Data Store) are also queried to the service API and persisted in the local RE database (Data Tables).
5. The Rule Engine periodically reads and parses all the active rules, and any activation by the patient data is transmitted to the CARPATH Notification system.

3.3.4 Clinical Decision Support Systems (CDSS)

Clinical Decision Support Systems (CDSS) provide decision support aids to provide clinical guideline-based treatment suggestions, to carry out risk assessments and to provide guidance about polypharmacy management and being utilized by AICP during the creation and update of care plans. This also includes an Early Warning System (EWS), utilising algorithms built using machine learning techniques to identify potentially preventable situations.

3.3.4.1 Clinical Guideline Suggestions

In CAREPATH, we will build clinical decision support services to deliver personalized guidance to healthcare professionals about the goals and interventions (treatment actions, patient monitoring activities and lifestyle management activities) that can be put into the active care plan of the patient. These suggestions will be built upon the recommendations of the clinical guidelines to achieve patient-centred and customised care.

The exact content of Clinical Decision Support Systems (CDSS) to be implemented in CAREPATH depends on the output of Task 6.2 in which a holistic patient centred CAREPATH best practice guideline will be established. Task 3.1 will deliver the rules that will be implemented as clinical decision support services based on the holistic clinical guideline flow.

The architecture design for CDSS for clinical guideline suggestions in this document focuses on the design of a generic CDS service that will be employed in Carepath project for delivering personalized care plan suggestions, independent of any content specific CDS modules.

The CDS services will consume patient's most recent health and wellbeing parameters and PROMs as input parameters (such as Electronic Health Records, parameters measured Home/Health Monitoring platform such as medical device readings, derived parameters such as dementia profile and intrinsic capacity detected by the home and health monitoring platform, risks detected by early warning services, patient provided data collected by patient empowerment platform such as symptoms and PROM scores), and in return will provide suggested care plan goals and interventions. These CDS services will be invoked by the Adaptive Integrated Care Platform (AICP), hence it is the role of AICP to gather all these input parameters and provide these as input parameters to CDSs. CDSs will in turn run the rules to deliver guideline-based care plan goal and intervention suggestions.

In order to allow AICP to access all CDS modules, a CDS-Hooks REST API will be designed, implemented and exposed to ensure all the messages are in conformance with its specification, granting interoperability with all the existing platforms. Details of this are provided in Section 2.3.2.4 CDS API.

In conformance to CDS-Hooks API, the input parameters are passed to CDS services as FHIR resources. In Carepath architecture, as described in Section 3.6.2, all information exchange among various Carepath components is achieved via the Patient Data Store which is a FHIR Repository. AICP retrieves the relevant CDS input parameters from FHIR repository (which has been previously stored to FHIR Repository by other components such as TIS/SIS, HHM/P, Early Warning Services), and passes them to CDS services. The response of CDS services can consist of textual recommendations communicated as information cards and computable recommendations communicated as suggestion cards in conformance to CDS Hooks API. In suggestion cards, the recommended goals and activities are represented as FHIR resources (such as MedicationRequest, Goal, Appointment resources).

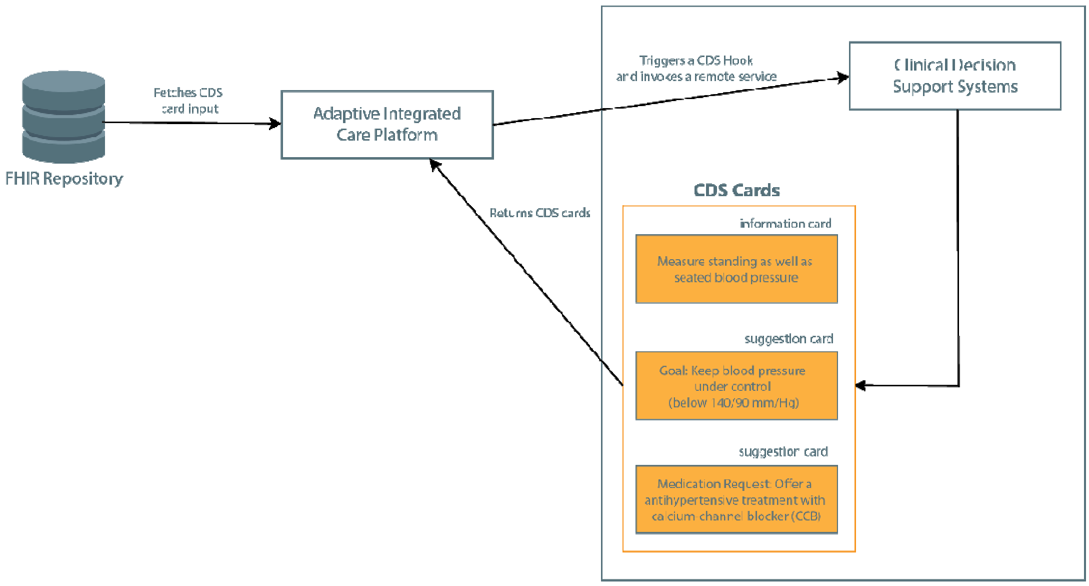


Figure 16: Architecture of Clinical Guideline Suggestions

3.3.4.2 Polypharmacy Management

More than half of all patients older than 65 years take more than 5 prescription drugs 90% of these patients additionally take more than one freely available drug, 50% take 2 to 4 of these preparations High risk for drug-drug interactions and other adverse events Potentially inappropriate prescribing (PIP) is a problem Further worsened due to incomplete case histories, low patient compliance or other inaccuracies and missing knowledge. There are criteria/guidelines for evaluating prescriptions so called screening tools for identifying polypharmacy. These guidelines are mainly paper based and not computerized, so they have to be applied manually by a medical professional. In the special case of elderly patients' development started with Mark Beers' list of inappropriate medications (1991) followed by other evidence-based rules:

- START (screening tool to alert to the right treatment)

NAME	patient	diagnosis	complication drugs	Recommendation
START A1	TRUE	icd == "I48.2"		START A1: Vitamin K antagonists or direct thrombin inhibitors or factor Xa inhibitors in the presence of chronic atrial fibrillation.
START A2	TRUE	icd == "I48.2"		START A2: Aspirin (75 mg – 160 mg once daily) in the presence of chronic atrial fibrillation, where Vitamin K antagonists or direct thrombin inhibitors or factor Xa inhibitors are contraindicated.
START A3	TRUE	hasICDPrefix("I25.1"),hasICDPrefix("I60"),hasICDPrefix("I61"),hasICDPrefix("I62"),hasICDPrefix("I63"),hasICDPrefix("I64"),hasICDPrefix("I65"),hasICDPrefix("I66"),hasICDPrefix("I67"),hasICDPrefix("I68"),hasICDPrefix("I69"),icd=="I73.9"		START A3: Antiplatelet therapy (aspirin or clopidogrel or prasugrel or ticagrelor) with a documented history of coronary, cerebral or peripheral vascular disease.

Figure 17: example START rules

- STOPP (screening tool of older people's prescript

NAME	subject drugs	diagnosis	complication drugs
STOPP B1	"C01AA05"	hasICDPrefix("I50")	
STOPP B2	"C08DA01","C08DB01"	hasICDPrefix("I50.04"),hasICDPrefix("I50.05"),hasICDPrefix("I50.13"),hasICDPrefix("I50.14")	
STOPP B3	"C07*"		"C08DA01","C08DB01"
STOPP B4	"C07*"	icd == "R00.1",icd=="I44.1",icd=="I44.2"	
STOPP B5	"C01BD01"	icd=="I47.1"	
STOPP B6	"C03C*"	hasICDPrefix("I1")	

Figure 18: example STOPP rules I

OutputNoDiag	OutputNoDrug
	STOPP B1: No clear evidence of benefit
	STOPP B2: May worsen heart failure
STOPP B3: risk of heart block.	
	STOPP B4: risk of heart block, asystole - (Drugs-ATC of group C07)
	STOPP B5: higher risk of side-effects than beta-blockers, digoxin, verapamil or diltiazem
	STOPP B6: Loop diuretic as first-line treatment for hypertension? (safer, more effective alternatives available).
	STOPP B7: Loop diuretic for dependent ankle oedema without clinical, biochemical evidence or radiological evidence of heart failure, liver failure, nephrotic syndrome or renal failure (leg elevation and /or compression hosiery usually more appropriate).

Figure 19: example STOPP rules II

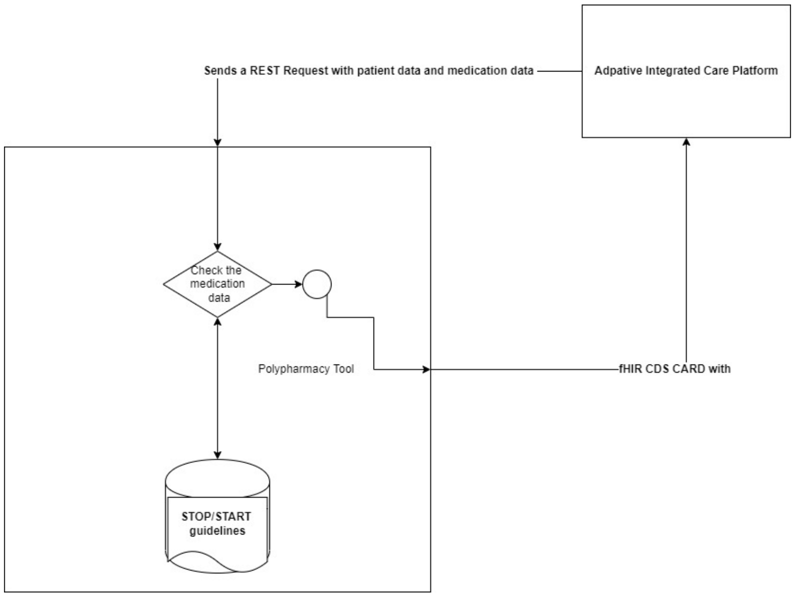


Figure 20: Architecture of the Polypharmacy system

The criteria try to minimize inappropriate prescribing. Published in 2008, updated in 2014 with a total of 114 rule statements. In CAREPATH will build on previous prototypes and customize the STOPP/START rules as computerized tool to identify in appropriate prescriptions of patients. The CAREPATH polypharmacy tool (see Figure 20: Architecture of the Polypharmacy system will provide a REST API interface that will be invoked by AICP providing patient data such as age and gender, conditions as ICD numbers and a list of medicaments including the ATC codes of each medicament. The CAREPATH polypharmacy tool will walk through all rules and check if any of them will fire. A firing rule will generate textual recommendations that will be compiled together and sent back to the AICP as FHIR CDS cards.

3.3.5 Early Warning Services (EWS)

3.3.5.1 EWS Technologies

The Early Warning Services (EWS) incorporate advanced technologies such as artificial intelligence and machine learning, to provide dynamic early warning about the development of the condition as well as the suitability of the treatment, namely:

1) effectiveness of medication and detection of side-effects.
2) development of a risk stratification model in collaboration with WP6 and definition of how the risk categories will be monitored by the various tools.
3) continuous monitoring of trajectories of intrinsic capacity and dementia profile by integrating patient data collected both passively and actively from home/health monitoring platform.

The Advanced Early Warning Smart Decision Tools gets inputs from the following components:

- Patient Empowerment Platform
- Home/Health Monitoring Platform
- Interoperability Framework
- Clinical Decision Support Modules

and outputs results to the Adaptive Care Planner to adapt the care pathway based on the current dementia profile of the patient. Figure 21 shows the EWS architecture in relation to the other CAREPATH modules.

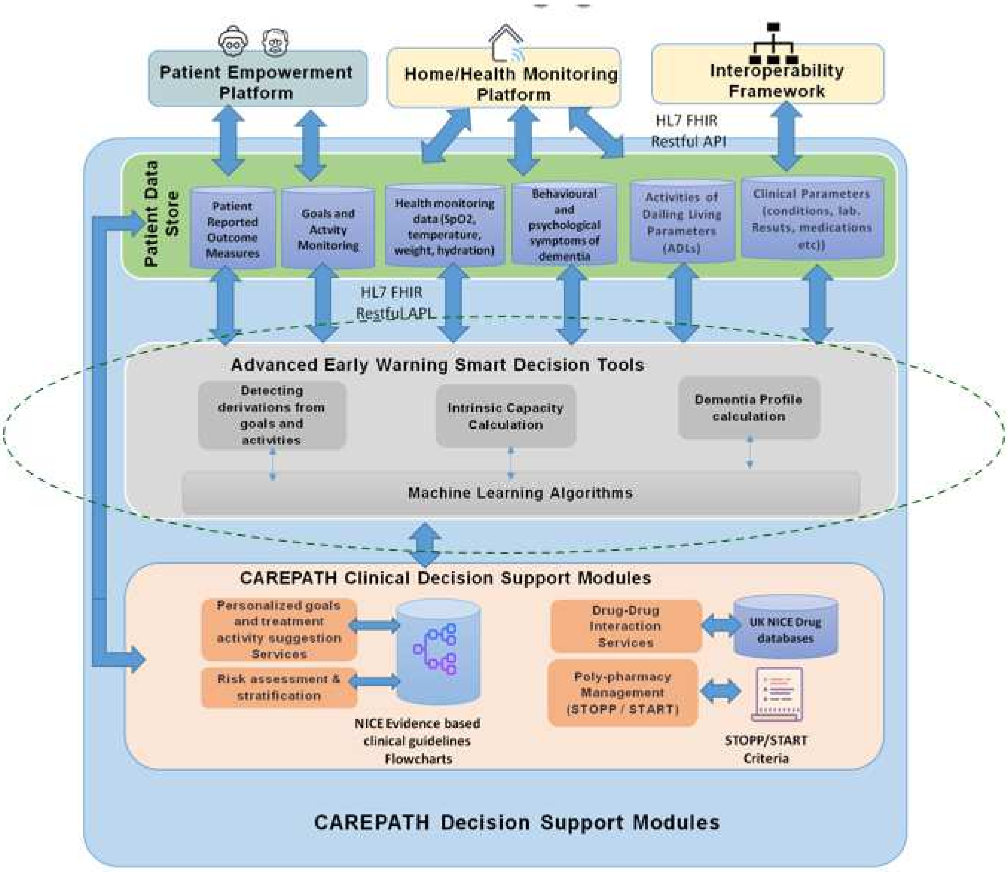
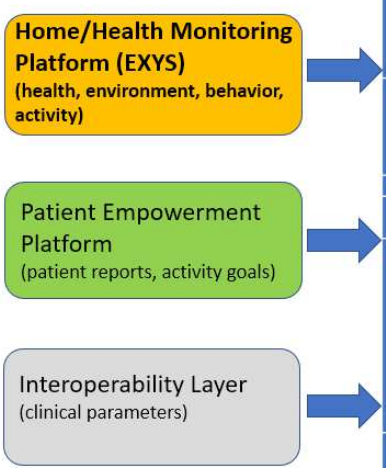


Figure 21: Early Warning Services

Monitoring the Dementia Profile

Figure 22 shows a scheme of the detection and identification of the dementia profile.



Category	Symptom	Data/service provision
Cognition	Gait speed (motor retardation or hyperactivity)	Mobility pattern detection (speed)
	Spatial disorientation	Geofencing
	Wandering (aberrant motor behaviour)	Mobility pattern detection (wandering)
	Memory loss	Reminders, Objects localization
	Language problems	TBD
Circadian rhythms	Sleep pattern changes (hypersomnia, insomnia, sleep-wake cycle reversal, fragmented sleep, and rapid eye movement sleep behaviour disorder)	Sleep monitoring and analysis
Mood	Depressive symptoms	TBD
Sensory	Poor vision	TBD
	Hearing capacity	TBD
Vitality/energy balance	Activity	• Health/home monitoring platform: activity raw data • Patient Empowerment platform: activity monitoring
	Weight	Health/home monitoring platform: health raw data
	BMI	Health/home monitoring platform: health raw data
Locomotion	Gait speed	Mobility pattern detection (speed)
	Muscle strength	?
	Sarcopenia	Health/home monitoring platform: health raw data (weight)
	Falls	Fall detection

Figure 22: Monitoring the Dementia Profile

Estimate the changes in Intrinsic Capacity

Intrinsic capacity is defined as the composite of all physical and mental capacities that an individual can draw upon during their life, in order to create a multidimensional indicator related to individual's functional status whose follow up over time may be useful to reach healthy aging (source: WHO).

Input (biomarkers *):	Output
Cognition: gait speed, memory loss **Psychosocial / Mood:** depressive symptoms **Sensory:** poor vision and hearing capacity **Vitality/energy balance**: activity, obesity or weight loss and low or high body mass index, diseases **Locomotion:** gait speed, muscle strength, sarcopenia [9]	Risk prediction model for the interpretation of patient's immediate and future health status by processing data from health and home monitoring environment, patient EHR's etc.

(*: biomarkers also included in Dementia Profile figure)

3.3.5.2 AI/ML Engine

Machine Learning algorithms will be employed for assessing both dementia profile and intrinsic capacity, as well as to aggregate information coming from different CAREPATH modules.

Types of ML algorithms that will be considered: **supervised** and **unsupervised**

Reinforcement learning could be also envisaged for particular functionalities.

Other data analysis methodologies that will be taken into consideration, envisioned to identify the most promising features and correlations among data, and to rationalize the computations, are **feature selection methods** (e.g., PCA principal component analysis) and **dimensionality reduction.**

The final choice of methodologies and a set of ML algorithms will strictly depend on the input and output variables, and on how the dementia and intrinsic capacity patterns that will be predicted/assessed will be precisely defined.

Finally, the AI/ML Engine will be deployed on a powerful multi-core workstation at the EXYS premise, which will batch-run all the selected AI/ML algorithms.

3.3.5.2.1 Overview of AI /ML algorithms

A structured list of possible AI/ML algorithms that will be taken into consideration is reported below.

Supervised Machine Learning Algorithms:

A set of training data is available against objective function

- **Regression algorithms,** to forecast single variables**:**
  - Linear, polynomial, logistic regression, etc.…
  - Bayesian networks (direct acyclic graphs), for inference
  - Evolutionary algorithms (genetic algorithms…)
  - Decision trees
  - K-nearest neighbours

[9] Operationalising the concept of intrinsic capacity in clinical settings, Islene Araujo de Carvalho, Finbarr C Martin, Matteo Cesari, Yuka Sumi, Jotheeswaran A Thiyagarajan, John Beard, WHO Clinical Consortium on Healthy Ageing, 21–22 November 2017 BACKGROUND PAPER

- Neural networks, particularly DNNs (in general effective only if large amount of training data is available, difficult to fine tune)

- **Classification algorithms:**
    - Support-vector machine
    - Naïve Bayes
    - Linear discriminant analysis
    - K-nearest neighbors
    - Evolutionary algorithms, such genetic algorithms
    - Neural networks, particularly DNNs (usually effective only if large amount of data, difficult to fine tuning

Unsupervised Machine Learning Algorithms:

No a-priori data patterns known, discover previously unknown patterns in data

- Clustering: hierarchical clustering, k-means, mixture models, DBSCAN, and OPTICS algorithm
- Anomaly detection: Local Outlier Factor, Isolation Forest

3.3.5.2.2 Some AI / ML tools

Some AI/ML tools that can be employed / supported by the system are listed here.

- TensorFlow
- PySpark
- Scikit Learn
- Pandas
- Acumos AI
- H2O
- PolyAxon
- Apache Kafka
- Jupyter

3.4 Service Layer

The service layer contains all the core software modules of the CAREPATH cloud-based architecture. These modules provide different but complementary functionalities which constitute the "heart" of CAREPATH. More details for each component are provided in the following subsections.

3.4.1 Message Broker

It is intended to enable *services* and applications to communicate with each other using messages.

3.4.2 Service APIs

3.4.2.1 H/HMP API

The H/HMP API will provide an interface between the H/HMP / H/HMP Data Store modules and the other CAREPATH's components. It is based on the OAuth 2.0 RESTful technology, assuring standard authentication/authorization access and security in data exchange. The authorization protocol Oauth 2.0 was already described in section 3.2.1.

The set of H/HMPI is given in JSON format using the Swagger Description Language for describing RESTful APIs. The RESTful architecture is based on the Internet's HTTP(S) protocol (v1.1).

The H/HMP API calls are categorized as shown in

Table 2

Table 2: H/HMP API categories of call

Category	Description
OAuth authentication/ authorization mechanism	This set of API calls enable the interaction of the operator with the OAuth 2.o authorization mechanism, i.e.: - Creation of the operators and caregivers. - Login (primary authentication) - Application creation and authorization - Token management (obtaining, refreshing, revoking the authorization token)
User configuration	These calls allow to configure the parameters associated to the users (caregivers, operators, admins) registered in the H/HMP Data Store
Device configuration	These calls allow to fully manage sensors and devices of the home and health sensor networks.
Health data query	These calls enable to query the H/HMP Data Store for data coming from sensors of the health sensor network.
Home / Environment sensor interaction	These calls enable to query the H/HMP Data Store for data coming from sensors of the home sensor network, as well as possibly activate some domotic actuators.

The detailed description for each of the available API's calls, as well as the authorized roles (who can call what), will be given later in the course of the project's implementation.

3.4.2.2 CDS API

The CDS API will be based on CDS-Hooks standard API[10]. CDS-Hooks API provides the specification of a "hook"-based pattern for invoking decision support from within a clinician's workflow. The API supports synchronous, workflow-triggered CDS calls returning information and suggestions.

In CDS Hooks, a CDS Service is a service that provides recommendations and guidance through RESTful APIs. The primary APIs are Discovery, which allows a CDS Developer to publish the types of CDS Services it provides, the Service endpoint that CDS Clients use to request decision support.

- **Discovery API**: A CDS Service is discoverable via a stable endpoint by CDS Clients. A RESTful operation needs to be implemented allowing to discover all available CDS modules, including information such as the purpose of the CDS Service, when it should be invoked, and any data that is requested to be pre-fetched. The discovery endpoint is always available at: {baseUrl}/cds-services. For example, if the baseUrl is https://carepath.care, the CDS Client MAY invoke: GET https://carepath.care/cds-services. As a response an object containing a list of CDS modules together with attributes within a CDSService description resource as depicted below.

ServiceDescription			
Field name	**Optionality**	**Type**	**Description**
hook	REQUIRED	string or URL	The hook this service should be invoked on.
title	RECOMMENDED	string	The human-friendly name of this service.
description	REQUIRED	string	The description of this service.
id	REQUIRED	string	The {id} portion of the URL to this service which is available at {baseUrl}/cds-services/{id}
prefetch	OPTIONAL	object	An object containing key/value pairs of FHIR queries that this service is requesting that the CDS Client prefetch and provide on each service call. The key is a string that describes the type

10 https://cds-hooks.org/

			of data being requested and the value is a string representing the FHIR query.

- **Service API:** A CDS Client SHALL call a CDS Service by POSTing a JSON document to the service. The CDS Service endpoint can be constructed from the CDS Service base URL and an individual service id as {baseUrl}/cds-services/{service.id}. The request SHALL include a JSON POST body including the following fields:

ServiceRequest			
Field name	**Optionality**	**Type**	**Description**
hook	REQUIRED	String	The hook that triggered this CDS Service call.
hookInstance	REQUIRED	string	A UUID for this particular hook call.
fhirServer	OPTIONAL	URL	The base URL of the CDS Client's FHIR server. The scheme should be https. In Carepath, we will not use this field as we do not expect the CDS service to fetch input parameters directşy from the FHIR server. AICP will pass the input parameters wşthin the prefetch field.
fhirAuthorization	OPTIONAL	object	The OAuth 2.0 authorization granting the CDS Service access to FHIR resources. As described above this will not be used in Carepath.
context	REQUIRED	object	Hook-specific contextual data that the CDS service will need (e.g. with the patient-view hook this will include the FHIR identifier of the Patient being viewed).
prefetch	REQUIRED	object	The FHIR data that was prefetched by the CDS Client. This will be prepared by AICP and passed to the service. This field is OPTIONAL in original specification but in our implementation it is required.
redirect	OPTIONAL	URL	The URL an app link card should redirect to upon completion of user interaction. Not used in Carepath.

For successful responses, CDS Services SHALL respond with a 200 HTTP response with an object containing a Cards array and optionally a systemAction array as described below

ServiceResponse			
Field name	**Optionality**	**Type**	**Description**
cards	REQUIRED	Array of Cards	An array of Cards. Cards can provide a combination of information (for reading), suggested actions (to be applied if a user selects them), and links (to launch an app if the user selects them). The CDS Client decides how to display cards, but displaying suggestions using buttons, and links using underlined text is recommended.
systemActions	OPTIONAL	Array of Actions	An array of Actions. An action should only be generated after interacting with the user through an app link. Actions are designed to convey any choices the user made in an app session. This field is not used in Carepath.

Card			
Field name	**Optionality**	**Type**	**Description**
summary	REQUIRED	string	One-sentence, <140-character summary message for display to the user inside of this card.
detail	OPTIONAL	string	Optional detailed information to display, represented in (GitHub Flavored) Markdown. (For non-urgent cards, the CDS Client may hide these details until the user clicks a link like "view more details…").
indicator	REQUIRED	string	Urgency/importance of what this card conveys.

			Allowed values, in order of increasing urgency, are: info, warning, critical. The CDS Client may use this field to help make UI display decisions such as sort order or coloring.
source	REQUIRED	object	Grouping structure for the Source of the information displayed on this card. The source should be the primary source of guidance for the decision support the card represents.
suggestions	OPTIONAL	Suggestion	Array of Suggestions that allows a service to suggest a set of changes in the context of the current activity (e.g., changing the dose of the medication currently being prescribed, for the medication-prescribe activity). If suggestions are present, selectionBehavior must also be provided.
selectionBehavior	CONDITIONAL	string	Describes the intended selection behavior of the suggestions in the card. Currently, the only allowed value is at-most-one, indicating that the user may choose none or at most one of the suggestions. Future versions of the specification may expand this behavior, so CDS Clients that do not understand the value must treat the card as an error. CDS Clients must support the value of at-most-one.
links	OPTIONAL	Link	Array of Links that allows a service to suggest a link to an app that the user might want to run for additional information or to help guide a decision.

Source			
Field name	**Optionality**	**Type**	**Description**
label	REQUIRED	string	A short, human-readable label to display for the source of the information displayed on this card. If a url is also specified, this may be the text for the hyperlink.
url	OPTIONAL	URL	An optional absolute URL to load (via GET, in a browser context) when a user clicks on this link to learn more about the organization or data set that provided the information on this card. Note that this URL should not be used to supply a context-specific “drill-down” view of the information on this card. For that, use link.url instead.
icon	OPTIONAL	URL	An absolute URL to an icon for the source of this card. The icon returned by this URL SHOULD be a 100x100 pixel PNG image without any transparent regions.
topic	OPTIONAL	Coding	A topic describes the content of the card by providing a high-level categorization that can be useful for filtering, searching or ordered display of related cards in the CDS client's UI.

Suggestion			
Field name	**Optionality**	**Type**	**Description**
label	REQUIRED	string	Human-readable label to display for this suggestion (e.g., the CDS Client might render this as the text on a button tied to this suggestion).
uuid	OPTIONAL	string	Unique identifier for this suggestion.
actions	OPTIONAL	Array of Actions	Array of Actions, each defining a suggested action. Within a suggestion, all actions are logically AND'd together, such that a user selecting a suggestion selects all of the actions within it.

Action

Field name	Optionality	Type	Description
type	REQUIRED	string	The type of action being performed. Allowed values are create, update, delete.
description	REQUIRED	string	Human-readable description of the suggested action may be presented to the end-user.
resource	CONDITONAL	object	Depending upon the type attribute, a new resource or the id of a resource. When the type attribute is create, the resource attribute shall contain a new FHIR resource to be created. For delete, this shall be the id of the resource to remove. In hooks where only one "content" resource is ever relevant, this attribute may be omitted for delete action types only. For update, this holds the updated resource in its entirety and not just the changed fields.
resourceId	CONDITIONAL	string	A relative reference to the relevant resource. SHOULD be provided when the type attribute is delete.

3.4.2.3 TIS/SIS APIs

3.4.2.3.1 Use Cases

Technical Interoperability Suite

TIS will provide interoperability interfaces to enable seamless data exchange between the CAREPATH solution and the systems external to the project, such as local care systems. The TIS component aims to allow data exchange while handling the heterogeneous protocols and clinical data representation formats which may be in use across the IT systems in local care sites. TIS will provide a **standard based data exchange protocol**. The following requirements have been identified as being relevant for defining the functionality/architecture of the interoperability tools:

- CARE-45: The Interoperability Framework, needs to retrieve the required patient data from local EHRs and migrate them into Carepath Patient Data Store as FHIR Resources, after handling the necessary structural and semantic mappings.

Additionally, based on the architecture of existing components within CAREPATH at the time of this deliverable, the following assumptions regarding the constraints/functionality of the interoperability services are made:

- FHIR will be adopted as the standard for data storage, representation and exchange between CAREPATH components.
- TIS will import patient data and/or clinical documents from local care system into CAREPATH via pull/query-based mechanisms but should also support importing of data through manual upload.
- TIS will utilize the structural mapping and terminology mapping services provided by Semantic Interoperability Suite (SIS) to transform data into FHIR format and terminology codes which can be interpreted and used by the CDS services.
- TIS will push data into CAREPATH FHIR repository (provided as part of Personalised Care Plan Management Platform) through its FHIR RESTful API;

It is important to note at this stage the specific structure of the FHIR resources applicable for the CAREPATH data model has been finalised and the actual data that will be made available by pilot sites still needs to be confirmed, through investigation of external APIs, identification of items in relevant clinical guidelines and development of the research protocol/needs of the CAREPATH tools.

Semantic Interoperability Suite

SIS will be developed to both handle structural mappings among different information models and resolve semantic mismatches due to use of different terminology systems and different compositional aggregations to represent the same clinical concept.

In the CAREPATH architecture, a patient's electronic health records will be received from local EHR systems via the Technical Interoperability Suite (TIS), which requests structural mapping from the Semantic Interoperability Suite (SIS) to transform this data into the FHIR format, used by the backend Data Store for the CAREPATH components. As part of these requests, SIS will also need to obtain semantic mappings for given FHIR resources. These mappings ensure correct linking between different data values coding systems, used by pilot sites and CAREPATH components.

Generally speaking, the Semantic Interoperability Platform has to handle both structural mappings and semantic mappings, ensuring that all data exchanged between pilot sites and all other CAREPATH components stay understandable, consistent and coherent. SIS provides a **standard based data exchange protocol** to support these structural and semantic mapping queries.

The Semantic Interoperability Suite will address content level interoperability challenges between the information systems in local care information systems and CAREPATH platform, by semantically mediating different clinical information representations. The design of SIS in based on FHIR. As part of the semantic mediation process, SIS will use a terminology service for terminology mappings and a semantic metadata registry to process interoperability.

In general, the SIS module runs in the background and so the effects of its use are not immediately visible to users. As such, specific requirements will not be extracted through user engagement, such as the clinical scenario storyboards being developed by pilot sites. Instead, the functionality of SIS will be inferred based on what is needed to support the requirements of user facing components

3.4.2.3.2 Component View

The following composition view identifies the major design constituents of the CAREPATH interoperability architecture, localizes and allocates functionality, responsibilities, or other design roles to these constituents. The view provides the design stakeholders with a high-level, architectural view of the system on the level of reusable subsystems and large-grained components. Components are considered autonomous, encapsulated units within a system or subsystem that provide one or more interfaces. This view is governed by the composition viewpoint declared by IEEE Std. 29148:2011. The design is represented by UML2 component diagrams.

Technical Interoperability Suite

The Technical Interoperability Suite (TIS) enables data exchange between the information systems in local care settings and CAREPATH. The design of TIS is based on the FHIR standard (release 4) for interoperability. TIS extends local information systems with FHIR endpoints to present a uniform interface for the exchange of patient data care plan. TIS uses the security and privacy suite to integrate into the local system's security environment.

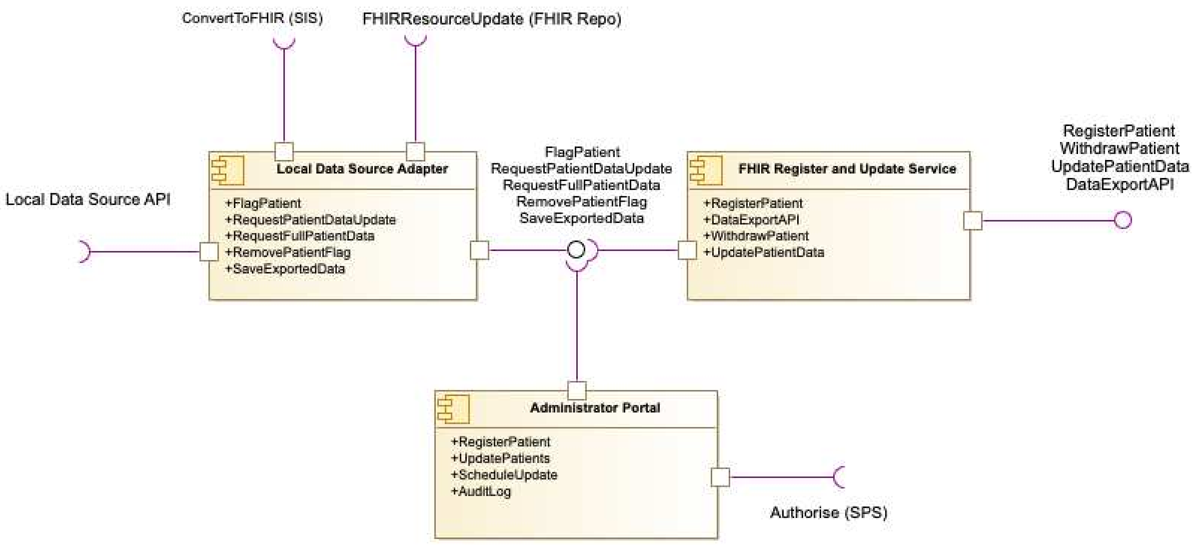


Figure 23: TIS component diagram

- ***<u>FHIR Register and Update Service</u>***

Type	Subsystem component
Purpose	Provides interfaces for the AICP and FHIR repository components enabling patient data from source systems to be

	requested and saved to the FHIR repo based on a pull mechanism.
Definition	A FHIR RESTful service for a client to query patient data related FHIR resources and save these resources to a FHIR repository
Provided Interfaces	
RegisterPatient	An endpoint for a client (e.g., AICP) to notify the local care record system that a patient has joined the CAREPATH study. This results in a flagging of the patient on the local care system and returns a full import of patient data for saving to the repository.
WithdrawPatient	An endpoint for a client (e.g., AICP) to notify the local care record system that a patient has been removed from the CAREPATH study.
UpdatePatientData	A FHIR RESTful endpoint for client (e.g., AICP) to retrieve patient data from a local care record system. This interface implements the FHIR RESTful API type level *search* interaction and includes a date parameter to limit response.
DataExportAPI	A collection of FHIR RESTful endpoints for clients (e.g., AICP) to send care plans, medication requests and referral requests to a local care information system. This interface implements the FHIR RESTful API instance level *update* interaction on FHIR *CarePlan*, *MedicationRequest* and *ReferralRequest* resources.
Required Interfaces	
None	

- ***Administrator Portal***

Type	Subsystem component
Purpose	Provides a user interface and associated functions for human administrators to view and edit update jobs and the internal audit log.
Definition	A UI and associated internal APIs for viewing, editing and executing update jobs.
Provided Interfaces	
RegisterPatient	Function to allow administrators to execute the register patient workflow directly from the TIS portal, notifying both the FHIR repo and the local care system.
UpdatePatients	Interface for executing batch jobs for updating multiple patients.
ScheduleUpdate	Interface for scheduling one-off or regular updates for registered patients.
AuditLog	Functions for viewing the internal audit log of update jobs.
Required Interfaces	
Authorize	The service requires SPS's AAA facilities to authenticate users, authorize access and manage audit trail.

- ***Local Data Source Adapter***

Type	Subsystem component

Purpose	Provides a plugin mechanism to connect each local information system through its native API.
Definition	A software module created for each local care information system to adapt the native API to a uniform internal API for the FHIR endpoint components to use.
Provided Interfaces	
FlagPatient	A uniform internal API to notify local care systems that a patient has been recruited to CAREPATH
RemovePatientFlag	A uniform internal API to notify local care systems that a patient has been removed from CAREPATH
RequestPatientDataUpdate	A uniform internal API to request patient data updates.
RequestFullPatientData	A uniform internal API to request all historical data for a patient.
SaveExportedData	A uniform internal API to export structured or unstructured care plan into local care systems.
Required Interfaces	
LocalDataSourceAPI	The local system API to request and send patient data
ConvertToFhir	SIS API to convert patient data in local format to FHIR format.
FHIRResourceUpdate	FHIR Repo API to save requested data from local care systems to the FHIR repository.

Semantic Interoperability Suite

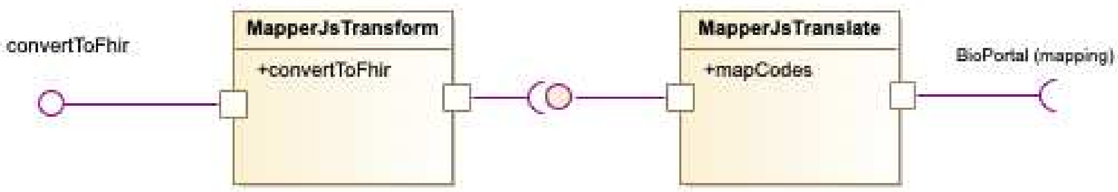


Figure 24: SIS component diagram

The Semantic Interoperability Suite addresses content level interoperability challenges between the information systems in local care information systems and CAREPATH platform, by semantically mediating different clinical information representations. The design of SIS in based on NodeJS, using templates for mapping between message formats. As part of the semantic mediation process, SIS uses a terminology service for terminology mappings and a semantic metadata registry for local codes.

- ***SIS:MapperJsTransform***

Type	Subsystem component
Purpose	Converts data provided by TIS to JSON/FHIR format through template-based substitution.
Definition	A FHIR RESTful service for client to convert input data to the CAREPATH FHIR format.
Provided Interfaces	
convertToFhir	A FHIR RESTful endpoint for TIS to convert patient data from local care information system to CAREPATH FHIR.

Required Interfaces	
mapCodes	The service requires transformation of CodableConcept data to CAREPATH terminology codes.

- ***SIS:MapperJsTranslate***

Type	Subsystem component
Purpose	Transcodes Concepts or Data Elements among distinct terminologies and formats.
Definition	A software client to request external terminology services for metadata mapping
Provided Interfaces	
mapCodes	Internal for the Transform subcomponent to request mappings of coded items
Required Interfaces	
External Terminology Mapping Services API	A RESTful endpoint to provide vocabulary/terminology mappings to CAREPATH selected terminologies. The BioPortal mapping service will be used for this feature.

3.4.2.3.3 Interface View

Technical Interoperability Suite

Interfaces provided by the TIS module for use by other components are limited to those included in the FHIR Register and Update Service subcomponent. All other interfaces are internal to TIS only and will not be documented here.

All interfaces provided by this module are HTTP POST only, as they are called to send information between systems and not to provide information back to the calling component. As such, return values will be HTTP status codes only, to provide information on whether the request was successful.

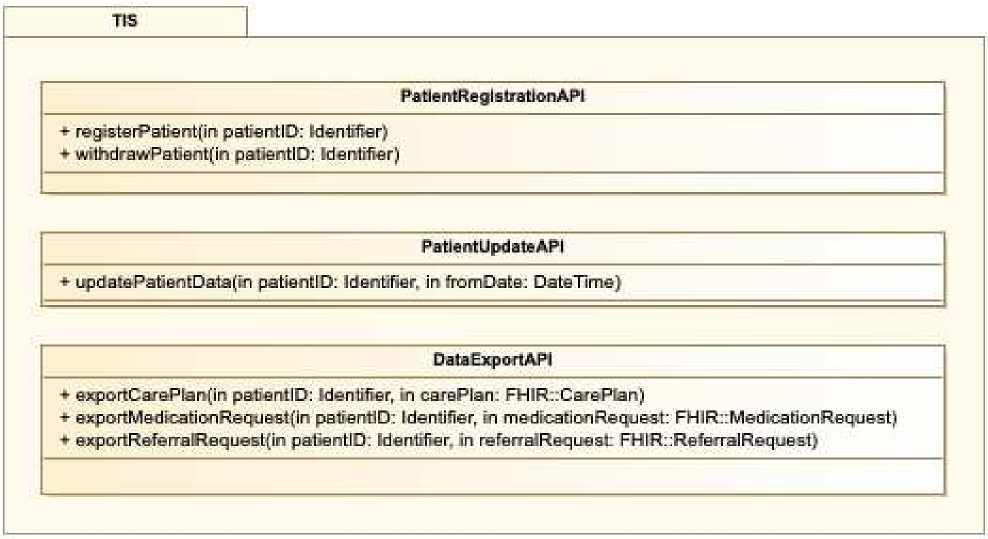


Figure 25: TIS interfaces

Interface: PatientRegistrationAPI		
Operation	registerPatient	A RESTful endpoint for systems to notify local EHR systems of a patient's registration to CAREPATH. Also triggers first import of

			patient data to the FHIR repository as a FHIR Bundle. Local EHRs are expected to flag the patient record as being included in CAREPATH.
Parameters	patientID	Identifier	Identifier of the patient to be registered (local EHR identifier).
Return	void		HTTP status code returned in header to indicate success or failure.
Operation	withdrawPatient		A RESTful endpoint for systems to notify local EHR systems of a patient's withdrawal from CAREPATH. Local EHRs are expected to remove CAREPATH flags from their local systems in response to a request from TIS.
Parameters	patientID	Identifier	Identifier of the patient to be withdrawn (local EHR identifier).
Return	void		HTTP status code returned in header to indicate success or failure.

Interface: PatientUpdateAPI			
Operation	updatePatientData		A RESTful endpoint for systems to request an update of patient data. Data will be uploaded directly to the FHIR repository using the FHIR::update or create requests
Parameters	patientID	Identifier	Identifier of the patient to be updated (local EHR identifier).
	fromDate	DateTime	Optional: DateTime in ISO format to indicate the earliest date of records to be imported. If omitted all patient data will be requested.
Return	void		HTTP status code returned in header to indicate success or failure

Interface: DataExportAPI			
Operation	exportCarePlan		A RESTful endpoint for CAREPATH components to export careplans for inclusion in local EHR systems, exported resources will be provided in FHIR format unless a suitable mapping to local formats is provided by the site.
Parameters	patientID	Identifier	Identifier of the patient whose data is being exported (local EHR identifier).
	carePlan	FHIR::CarePlan	The resource to be sent to the local EHR.
Return	void		HTTP status code returned in header to indicate success or failure
Operation	exportMedicationRequest		A RESTful endpoint for CAREPATH components to export medication requests for inclusion in local EHR systems, exported resources will be provided in FHIR format unless a suitable mapping to local formats is provided by the site.
Parameters	patientID	Identifier	Identifier of the patient whose data is being exported (local EHR identifier).
	medicationRequest	FHIR::MedicationRequest	The resource to be sent to the local EHR.
Return	void		HTTP status code returned in header to indicate success or failure
Operation	exportReferralRequest		A RESTful endpoint for CAREPATH components to export referral requests

			for inclusion in local EHR systems, exported resources will be provided in FHIR format unless a suitable mapping to local formats is provided by the site.
Parameters	patientID	Identifier	Identifier of the patient whose data is being exported (local EHR identifier).
	referralRequest	FHIR::ReferralRequest	The resource to be sent to the local EHR.
Return	void		HTTP status code returned in header to indicate success or failure

Semantic Interoperability Suite

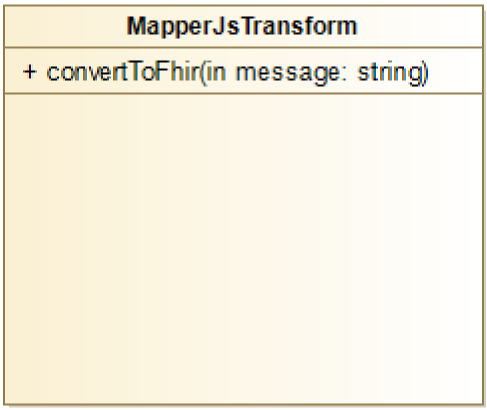


Figure 26: SIS interfaces

Interface: MapperJsTransform			
Operation	convertToFhir		A RESTful endpoint for systems to convert input resources to FHIR resources.
Parameters	message		Either a JSON or XML resource.
Return	string		FHIR resource or error message if conversion failed.

3.4.2.3.4 Interaction View

Technical Interoperability Suite(TIS)

- TIS-1: Import Patient Data, fromDate is an optional variable, omitting this variable should result in the full patient history being retrieved from the local data source. The initiating component is expected to be either the TIS UI (administrator portal) or AICP.
- TIS-2: Register Patient, this sequence diagram covers both registration and withdrawal. Registration involves both flagging the patient in the local system and triggering a first import of the patient history. The initiating component is expected to be either the TIS UI (administrator portal) or AICP.
- TIS-3: Export Patient Data, the same sequence is expected for CarePlans, MedicationRequests and ReferralRequests. The convertion to a local format via SIS is optional, depending on whether suitable mappings are provided by local sites. The initiating component is expected to be AICP.

3.5 Knowledge Layer

The Knowledge layer includes all of the knowledge/data needed for the proper functioning of the CAREPATH components as well as the knowledge/results that will be extracted by CAREPATH along with the corresponding storage mechanisms. The repositories included in the knowledge layer concern clinical guidelines, policies and other rules that should be followed in the CAREPATH platform, FHIR based Common Data Model, EHR data, H/HMP sensor data, Derived data, Alerts and Notifications.

Interoperability Services knowledge models

To deliver valid transformation between source data formats and the CAREAPTH data format, the structural mapping component of SIS needs to store the schema and mapping from the source data set to FHIR. Additionally, concepts in local terminologies need to be mapped to their equivalents in selected terminologies for use in CAREPATH, in order to develop clinical decision support services which can be used across regions. The following section details the logical models which need to be captured by the interoperability services to provide this functionality.

3.5.1 Logical View

The following logical views focus on the domain models of the CAREPATH system. A domain model is a conceptual model of the problem domain that a system is designed to solve. The objective of the model is to identify key actors and domain entities, and their logical relationships in the system, in order to formally model the problem and establish a common vocabulary to describe the problem and solutions. The following sections present detailed design of the logical view for each subsystem within the interoperability suite.

3.5.2 Technical Interoperability Suite

The logical view of technical interoperability suite extends and refines the interactions between the patient data, administrator and care record systems, with respect to the transformations needed to import and export data through the TIS subsystem.

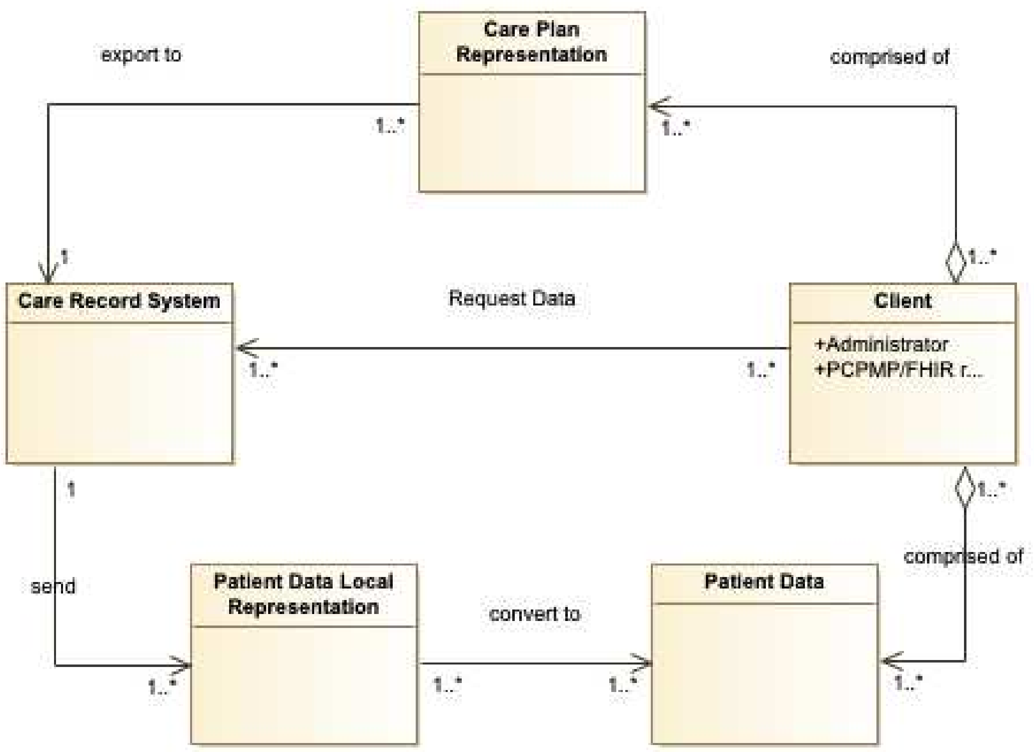


Figure 27: TIS logical diagram

Domain Entity	Description
Client	Abstract class of any client requesting data from a local care system, could include software or human actors
Care Plan Representation	Abstract class of care plan representation, which has physical implementations in each system.
Care Record System	A care record system processes patient data and care plan encoded in concrete representations.
Patient Data	Conceptual entity of patient data, which have physical encoding format in a concrete system implementation.
Patient Data Local Representation	The physical representation of patient data in a local care record system.

Semantic Interoperability Suite (SIS)

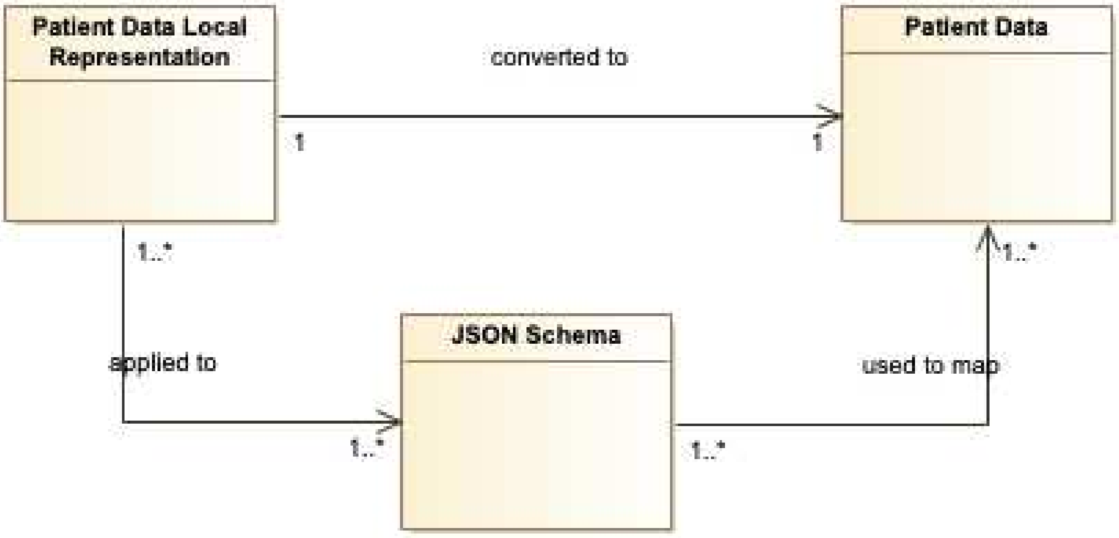


Figure 28: SIS logical diagram

Domain Entity	Description
Patient Data Local Representation	Resource given to transform containing either an XML or JSON representation of a care resource from TIS
Patient Data	Any type of clinical or administrative patient data that is relevant in the context of care planning. Represented as FHIR resources.
JSON Schema	Detailed representation of an exportable FHIR resource or Input Resource in a JSON format.

3.6 Persistence Layer

3.6.1 H/HMP Data Store

The H/HMP Data Store is in charge to persist data coming from the home and health sensor networks, as well as the derived information outcome from the processing activities performed by the H/HMP module.

The core of H/HMP Data Store is implemented using a no-sql database, namely MongoDB, ensuring easy storage of large amount of data, while guaranteeing fast access to information. If there is the need to persist more structured data in a more rational way and for a specific use, the core database can be made to work in parallel with a relational database (e.g., PostgreSQL).

3.6.2 Patient Data Store (FHIR Repositories)

Patient Data Store purposes to store data coming from various CAREPATH components (TIS, CDSS, PEP, AICP, H/HMP, EWS) required for personalised care planning and execution in a unified manner, and exchange of these data with authorized CAREPATH components.

In CAREPATH, Patient Data Store will be FHIR repository implementing HL7 FHIR R4 specifications. There will be support for tens of FHIR resource types including clinical (Condition, Observation, Procedure), care plan related (CarePlan, CareTeam, Goal), administrative (Patient, AuditEvent) and further (e.g., Questionnaire) resources. FHIR RESTful API will be provided for exchange of FHIR resources with other components. Resources will also be stored as JSON files inside a MongoDB database for fast access and manipulation.

The API methods that will be utilized in the CAREPATH are as follows:

- *FHIRResourceRead*: A read-only FHIR RESTful endpoint to access available FHIR resources in the repository. This interface will implement the FHIR RESTful API instance level interactions: read, vread and history and type level interaction: search on all supported FHIR resource types (Condition, Observation, CarePlan, AuditEvent, QuestionnaireResponse …).
- *FHIRResourceUpdate*: A FHIR RESTful for unlimited access (i.e. create, update, delete) to FHIR resources in the repository. This interface will implement the FHIR RESTful API instance level interactions: update and delete and type level interaction: create on all supported FHIR resource types (Condition, Observation, CarePlan, AuditEvent, QuestionnaireResponse …).
- *FHIRResourceSubscription*: A FHIR RESTful endpoint for subscribing to a resource according to a criterion (e.g., any update, or upon update of a specific attribute) by utilizing the FHIR Subscription resource type. The preferred “channels” for sending updates to subscribed systems is rest-hook and websocket.

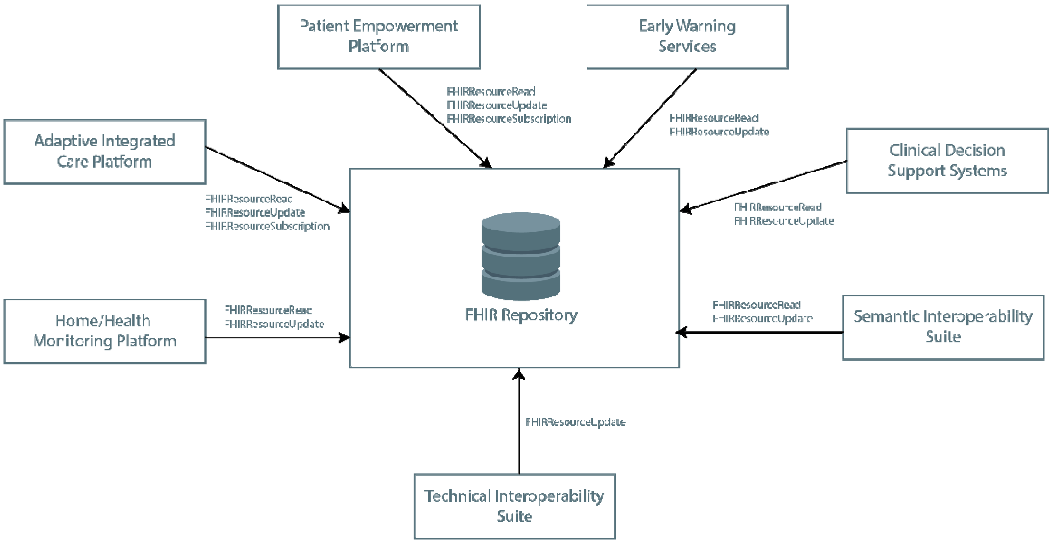


Figure 29: Architecture of Patient Data Store (FHIR Repository

4. Conclusions

This deliverable describes the initial version of the CAREPATH architecture that was designed within Task 2.2 of WP2 as a starting point and basis for future work. This high-level architecture builds on the work done in Task 2.1 where the user requirements were elicited in WP2.

As a result, a core set of CAREPATH platform is divided into several layers of the system based on their functionality were identified and presented in this document. First, the components of the presentation layer which constitute the user interfaces of CAREPATH system, then other layers are presented e.g. repositories of knowledge layer where all of the knowledge and data for the proper functionality of service layer components are stored as well as the outcomes of the system, subsequently the presentation layer and the UI manager which is responsible for the display of the various system views and lastly the security layer and the authentication, access rights management and communication encryption procedures it includes.

The results of this deliverable will form the input for the work on Specifications of all components and Customisation requirements, these results will be reported in D2.5.

It should be noted that although this initial architecture has been defined, the definition and implementation of the CAREPATH architecture is an ongoing task. The final version of the architecture will be presented in deliverable *D2.4 Final* CAREPATH *architecture definition* that will be delivered on M46.

References

[1] C. A. Velasco, Y. Mohamad and P. Ackermann, “Architecture of a web of things eHealth framework for the support of users with chronic diseases,” in *Association for Computing Machinery -ACM-DSAI 2016, 7th International Conference on Software Development and Technologies for Enhancing Accessibility and Fighting Info-exclusion.*, Vila Real, Portugal,, 2016.

[2] Y. Mohamad, C. A. Velasco, N. Kaklanis, D. Tzovaras and F. Paternò, “A Holistic Decision Support Environment for Web Accessibility,” in *International Conference on Computers Helping People with Special Needs*, 2018.

[3] A. Freier, P. Karlton and P. Kocher, “The Secure Sockets Layer (SSL) Protocol Version 3.0,” 2011.

[4] P. Dunlu, L. Chen and H. Huan, “An extended UsernameToken-based approach for REST-style Web Service Security Authentication,” in *2nd IEEE International Conference on Computer Science and Information Technology*, Beijing, China, 2009.

[5] M. Jones, B. Campbell and C. Mortimore, “ JSON Web Token (JWT) profile for OAuth 2.0 client authentication and authorization Grants,” Internet Engineering Task Force (IETF), 2015.

[6] P. Tsangaratos and I. Ilia, “Comparison of a logistic regression and Naïve Bayes classifier in landslide susceptibility assessments: The influence of models complexity and training dataset size,” *ENA,* vol. 145, pp. 164-179, 2016.

[7] B. Letham, C. Rudin, T. H. McCormick and D. Madigan, “Interpretable classifiers using rules and Bayesian analysis: Building a better stroke prediction model,” *The Annals of Applied Statistics,* pp. 1350-1371, 2015.

[8] X. Blasco, J. Herrero, J. Sanchis and M. Martinez, “Anew graphical visualization of n-dimensional Pareto front for decision-making in multiobjective optimization,” *Information Sciences,* pp. 3908-3924, 15 October 2008.

[9] A. Corbellini, C. Mateos, A. Zunino, D. Godoy and S. Schiaffino, “Persisting big-data: The NoSQL landscape,” *Information Systems,* pp. 1-23, 2017.

[10] J. R. Lourenço, B. Cabral, P. Carreiro, M. Vieira and J. Bernardino, “Choosing the right NoSQL database for the job: a quality attribute evaluation,” *Journal of Big Data,* 2015.

[11] E. Tang and Y. Fan, “Performance Comparison between Five NoSQL Databases,” in *7th International Conference on Cloud Computing and Big Data*, 2016.

[13] A. Spyrou, N. Kaklanis, D. Tzovaras, Y. Mohamad, H. Gappa, A. C. Velasco, M. Breidenbach, S. Caria, F. Paterno and F. Pulina, “WADcher: A unified web accessibility assessment framework,” in *SMART ACCESSIBILITY CONFERENCE*, Athens, 2019.

Review status

Deliverable leader:	**Yehya Mohamad, Henrike Gappa and Carlos A. Velasco (Fraunhofer)**
Contributors:	Yehya Mohamad, Omar Khan (Warwick), Mert Gencturk, Gokce Banu Laleci Erturkmen (SRDC), Luca Gilardi (EXYS)
Reviewers:	G. Erturkmen (SRDC)
Approved by:	Angelo Consoli (EXYS), Jaouhar Ayadi

Document History

Date	Changes	Version	Authors
7 October 2021	Initiation of Deliverable Table of Content (ToC) and first draft	1v0	Yehya Mohamad (Fraunhofer)
4 November 2021		1v1	
26 January 2022	Addition of content for interoperability services	1v2	Omar Khan (Warwick)
3 February 2022	Addition of content for Adaptive Integrated Care Platform (AICP), Patient Empowerment Platform (PEP), Clinical Guideline Suggestions, FHIR API, CDS API, and Patient Data Store (FHIR Repositories)	1v3	Mert Gencturk, Gokce Banu Laleci Erturkmen (SRDC)
4 February 2022	Addition of content about H/HMP, H/HMP Data Store, H/HMP API and Early Warning Services.	1v4	Luca Gilardi (EXYS)
10. February 2022	Consolidation of the contributions of the consortium and clarifying pending issues. Producing a new version	1V5	Yehya Mohamad (Fraunhofer)
17.February 2022	Corrections and proof reading before review. Document sent for review	1V6	Yehya Mohamad (Fraunhofer)
20. February 2022	Internal Review	1v7	Gokce Banu Laleci Erturkmen (SRDC) & Luca Gilardi (EXYS)
22. February 2022	Final corrections	1v8	Yehya Mohamad (Fraunhofer)
28. February 2022	Deliverable D2.3 last editing and submission to the EC.	1v9	Angelo Consoli (EXYS), Jaouhar Ayadi

- End of document -