Deliverable 3.3.1

Conceptual Design of EMPOWER Architecture

Project title:	Support of Patient Empowerment by an intelligent self-management pathway for patients
Project acronym:	EMPOWER
Project identifier:	FP7-ICT-2011-288209
Project instrument:	STREP
Web link:	www.empower-fp7.eu
Dissemination level:	PU (public)
Contractual delivery:	2012-10-31
Actual delivery:	2012-10-31
Leading partner:	SRFG

The research leading to these results has received funding from the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement No 288209, EMPOWER Project.

Document History

Version	Date	Changes	From	Review
V0.1	06.07.2012	Initial Document		All Partners
V0.2	17.07.2012	Collected contributions to Section 4	SRFG, ICOM, SRDC	
V0.3	20.08.2012	Contributions to Sections 7, 9, 10, 12	SRFG, ICOM	All Partners
V0.4	04.10.2012	Details to Sections 7 Initial version of section 10.	HMGU, SRDC, ICOM, SRFG	All Partners
V0.5	11.10.2012	More contributions and details, More comments for contribution request	SRFG	All Partners
V0.6	16.10.2012	Sections 4 and 5 finalised.	HMGU, SRDC, SRFG	All Partners
V0.7	25.10.2012	Updates on Section 7, Completed List of Functional Requirements	ICOM, SRFG	All Partners
V0.8	25.10.2012	Preliminary Version for final proof-reading	SRFG	SRDC, HGMU
V0.9	29.10.2012	Including Priorities to Use Cases and Functional Requirements Further updates across the document.	SRFG, ICOM	All Partners
V1.0	30.10.2012	Final Proofreading	SRFG, HGMU, SRDC	

EMPOWER Consortium Contacts

Beneficiary	Name	Phone	E-Mail
SRFG	Manuela Plößnig	+43 662 2288 402	manuela.ploessnig@salzburgresearch.at
HMGU	Claudia Hildebrand	+49 89 3187 4182	hildebra@helmholtz-muenchen.de
GOIN	Siegfried Jedamzik	+49 8 41956161	siegfried.jedamzik@googlemail.com
USI	Peter J. Schulz	+41586664724	peter.schulz@usi.ch
SRDC	Asuman Dogac	+90 312 210 13 93	asuman@srdc.com.tr
ICOM	Ilias Lamprinos	+302106677953	labil@intracom.gr
MOH	Unal Hulur	+903125851907	unal.hulur@saglik.gov.tr

Table of Contents

Abbreviations

AP	Action Plan
APE	Action Plan Engine
ARR	Audit Record Repository
ATNA	Audit Trail and Node Authentication
BDT	Behandlungsdatentransfer (German)
BPM	Business Process Management
BPMN	Business Process Model and Notation
EHR	Electronic Health Record
EHRS	Electronic Health Record System
EMR	Electronic Medical Record Software
EIP	Enterprise Integration Patterns
epSOS	European Patients Smart Open Services
ESB	Enterprise Service Bus
GP	General Practitioner
HIS	Hospital Information System
HL7	Health Level Seven
IDE	Interactive Development Environment
IHE	Integrating Healthcare Enterprise
i18n	Internationalisation
JDK	Java Development Kit
JSON	JavaScript Object Notation
JSON-LD	JSON for Linking Data
JVM	Java Virtual Machine
LDT	Labordatenträger (German)
MP-DG	Machine Processable Diabetes Guideline
MVC	Model View Controller
MVVM	Model View Viewmodel
ODL	Observation of Daily Living
PHA	Personal Health Application
PHR	Personal Health Record
PHRS	Personal Health Record System
PIS	Practice Information System
PKI	Public/Private Key Infrastructure
REST	Representational State Transfer
RDF	Resource Description Framework
RDFa	RDF in Attributes
TLS	Transport Layer Security
SCM	Source Code Management
SL	Service Locator
SOA	Service Oriented Architecture
SOAP	SOAP Version 1.2
XML	Extensible Markup Language
XDS	Cross Enterprise Document Sharing
XPHR	Exchange of Personal Health Record

1 Summary

This document describes the conceptual architecture of the software components developed within the EMPOWER project. The conceptual architecture is the first breakdown into functional components, based on the technical requirements as collected in D3.2.1. Based on this conceptual architectural basis the implementation tasks will start developing the EMPOWER system architecture.

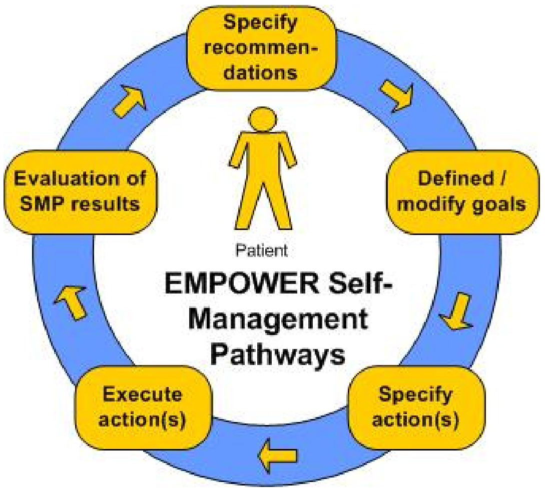
This document first collects architectural constraints brought either from the project requirements or from existing software and technologies that will integrate or connect to EMPOWER. Second, the architectural details are provided, like the breakdown into subsystems and components and their interoperability with each other and external systems. This includes also cross-cutting concerns like data persistence and security, as well as the description how the system will be finally deployed in the pilot applications.

2 EMPOWER in a Nutshell

Patient Empowerment involves patients to a greater extent in their own healthcare process and disease management becomes an integrated part of their daily lives. The capability of self-management opens to them the possibility for patients not only to contribute to their own healthcare but also to be more in control of their disease. EMPOWER develops a modular and standard-based Patient Empowerment Framework which facilitates the self-management of diabetes patients based on PHRs and on context-aware, personalised services. EMPOWER focuses the research and development efforts on a patient-centric perspective that also involves healthcare professionals. EMPOWER provides knowledge-based Self-Management Pathways for diabetes patients. This includes

(1) Services for the specification and execution of actions to change behaviour according to diabetes-specific health care needs. Patients can develop personalised action plans which include recommendations from the treating physicians and patients' preferences

(2) Services for monitoring of vital, physical, mental parameters as well as physical and lifestyle activities based on health standards.



EMPOWER semantically integrates multiple information sources (EHR/PHR, diabetes guidelines, patterns of daily living) for a shared knowledge model. The Self-Management Pathways facilitate the specification of recommendations that allow specifying individual goals for the patient. Based on these goals, relevant information and their preferences patients can specify their individual diabetes-specific actions. The Self-Management Pathways are an iterative process where executed actions and reported patterns of daily life can be evaluated. Recommendations, goals and actions can be updated iteratively according to current needs and preferences. Finally, the services in EMPOWER will embrace semantic interoperability based on health standards such as HL7[1] and IHE[2] profiles.

EMPOWER addresses long-term goals and short-term activities in order to facilitate the self-management of patients with diabetes and thus the treatment of chronic diseases. The pilot applications in Germany and Turkey will demonstrate that the holistic and patient-centric approach of EMPOWER can improve disease management by personalised self-management services helping diabetes patients to cope better with their condition.

[1] http://www.hl7.org
[2] http://www.ihe.net

3 Introduction

This document contains the Conceptual Design of the EMPOWER Architecture[3]. The document is based on the use cases and requirements gathered in "Deliverable 3.2.1 – Technical Requirements of EMPOWER Architecture" and "Deliverable 8.1.1 – Requirement Specifications and Scenario for the Pilot Application". The common architecture design in the EMPOWER project is expressed in UML in a separate file, using "Visual Paradigm for UML"[4] as design tool. Many of the figures depicted and described in this document are taken from the architectural model designed using this tool. Therefore, this deliverable documents the final snapshot of the conceptual design phase, after the basic architectural components have been defined and their interdependencies were specified. The architectural model itself will further evolve during the EMPOWER Architecture implementation phase and the UML model will be updated on demand as a living document. Furthermore, the project-internal wiki is used to document other living information (e.g. configuration parameters, glossary) during the implementation.

In the introduction a summary of the requirements including a reference to their detailed description is provided. Furthermore three rough levels of priorities are defined for requirements in order to rank them:

- **High (essential)**: Describes a basic functionality for the system core. The system cannot operate without these critical functions. Implies that the software will not be acceptable unless these requirements are provided in an agreed manner.
- **Normal (conditional)**: Are still required to provide a useful system, but the system can operate also without these functionalities. Implies that these are requirements that would enhance the software product, but would not make it unacceptable if they are absent.
- **Low (optional)**: Nice-to-have, currently the lowest priority. Implies a class of functions that may or may not be worthwhile. This level may be further refined during the exploitation phase.

After the summarisation of the use cases (Section 3.1) and functional requirements (Section 3.2) which were already collected in previous deliverables, the conceptual architecture documentation is structured as follows:

Section 4 contains the basic architectural constraints that are applied on the system implementation. Some of these constraints come already from the existing system context (like PHAs, EHR systems, etc), while other constraints have been derived from use cases, technical requirements and state-of-the-art implementation patterns. This section also provides conventions that have been agreed to support the distributed EMPOWER system development.

Figure 1 shows the EMPOWER environment, as already presented in previous documents. This figure presents the system scope and context. It shows that EMPOWER needs to communicate with human users (Patients, Health Care Actors), but also with existing healthcare applications. Those applications are described in Section 5 to define the borders between the developed system and other existing software components.

[3] We acknowledge that this document uses material from the arc 42 architecture template, http://www.arc42.de. Created by Dr. Peter Hruschka & Dr. Gernot Starke.
[4] http://www.visual-paradigm.com/product/?favor=vpuml

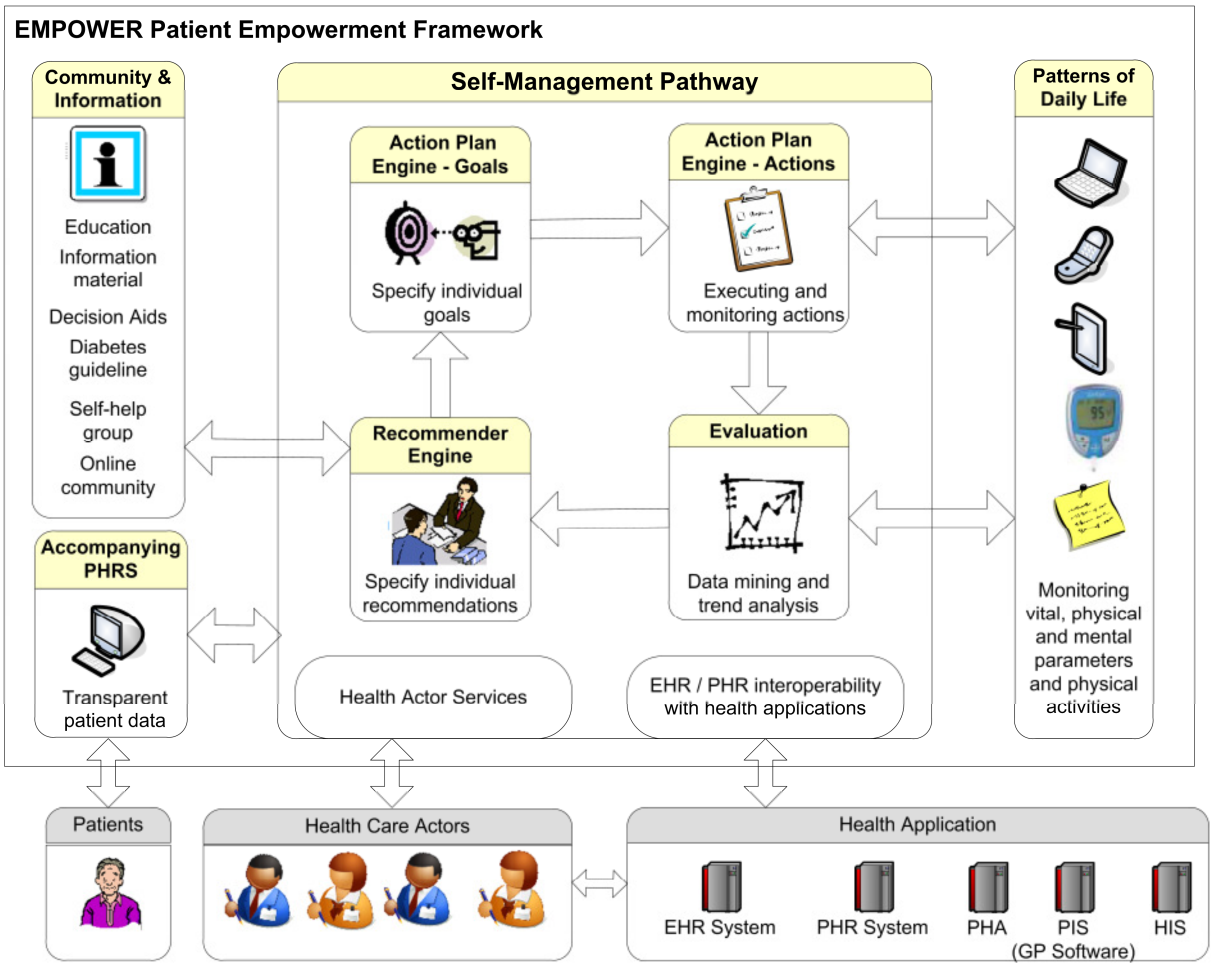


Figure 1: EMPOWER Patient Empowerment Framework

In Section 6 the basic solution ideas and strategies are collected. We define four strategies that will be followed within the project: (1) A service-oriented architecture with loose coupling between components, (2) the use of a rule-based business process engine for the modelling of the core EMPOWER business processes, (3) the support of multimodal user interfaces to have a wide range of possible application and (4) the extensive use of knowledge models to decouple the model development from the EMPOWER implementation and enable the inclusion of domain expertise via already existing knowledge models.

After the setting of this basic framework, the breakdown into building blocks of the EMPOWER system are shown in two levels in Section 0. First, the single subsystems are presented in a black-box view, i.e. by providing an outside view and to show the interdependencies with other components. Second, in the white-box view (Section 7.2), each subsystem is opened up and more detailed information about the functionality inside of the subsystems, like a further component breakdown is described and depicted graphically.

In order to show the most important interactions between components and subsystems in sequential order, five runtime views on the EMPOWER system are provided in Section 0.

In Section 0 the deployment of the EMPOWER architecture is shown. Separate subsystems must be able to run on different machines, which introduce the need of secure communication in-between. As EMPOWER serves two different pilot regions (Germany and Turkey), this section also covers the deployments as required due to the different pilot application conditions.

Section 0 defines the technical concepts and architectural aspects that are applied throughout the single building blocks of the EMPOWER architecture.

In Section 0 the basic approach on the mapping EMPOWER functionality to the user interfaces is provided. It shall be noted here, that a detailed description of user interfaces, including sketches and mock-ups will be available in deliverable D8.2.1.

In the final section, Section 0, a glossary of the most important terms used in the EMPOWER context is listed. The source of the glossary is a wiki-page, which is maintained by the project partners throughout the full lifetime of EMPOWER.

3.1 Use Cases Summary

In this section we provide the overview on the use cases collected in D3.2.1 and rank them to get knowledge in order to identify more and less important components in the architecture. The priorities will also be used be reflected during the implementation phase.

ID	Name	Priority
UC-4.1.1	Manage Pathways Knowledge Model	**High**
UC-4.1.2	Manage EMPOWER Inbox	**High**
UC-4.1.3	Open Inbox Message	**High**
UC-4.1.4	Get Chronicle View of Consultations	**High**
UC-4.1.5	Navigate EMPOWER UI Dashboard	**High**
UC-4.1.6	Patient Receives Messages	**High**
UC-4.2.1	Presentation of Trends and Visualization of Self-management Services	**High**
UC-4.2.1.1	Presentation of Trends and Visualization of Self-management Services for the patients	**High**
UC-4.2.1.1.1	Presentation of Trends and Visualization of measured and threshold glucose values for the patients	**High**
UC-4.2.1.1.2	Presentation of Trends and Visualization of pulse and stress level values for the patients	**High**
UC-4.2.1.2	Presentation of Trends and Visualization of Self-management Services for the medical professionals	Normal
UC-4.2.1.2.1	Presentation of Trends and Visualization of measured and threshold glucose values for the medical professionals	Normal
UC-4.2.1.2.2	Presentation of Trends and Visualization of glucose and mood values for the medical professionals	Normal
UC-4.2.1.2.3	Presentation of Trends and Visualization of pulse and stress level values for the medical professionals	Normal
UC-4.2.3	Present ODL Graphs	**High**
UC-4.3.1	Manage Patient Consent	**High**
UC-4.3.1.1	Patient Gives Consent to Medical Professional Using	**High**

	the Consent Editor Tool	
UC-4.3.1.2	Patient Removes Medical Professional from the “Given Consent” List	**High**
UC-4.3.1.3	Grant or Deny Access to Patient Data based on Consent	**High**
UC-4.3.2	Find Contact Information	Normal
UC-4.3.3	Manage EMPOWER User	High
UC-4.3.4	Patient Inserts/Modifies Contact Info	Normal
UC-4.3.5	Manage Patient Profile and Settings	Normal
UC-4.3.5.1	Select User Interface Language	Normal
UC-4.3.6	Manage Identifiers for User Identity	Normal
UC-4.3.7	Authenticate User	**High**
UC-4.3.8	Send Audit Record	High
UC-4.3.9	Authenticate EMPOWER Component (Node) for Secure Channel	**High**
UC-4.3.2	Find Contact Information from EMPOWER White pages	Normal
UC-5.2.1	Create Machine Processable Diabetes Guideline	**High**
UC-5.2.2	Assign Diabetes Guideline to Patient	**High**
UC-5.2.3	Execute and Monitor Diabetes Guideline	**High**
UC-5.2.4	Generate Recommendation about the Patient	**High**
UC-5.2.5	Create Recommendation Rules for Diabetes Patients	**High**
UC-5.2.6	Approve/Update/Extend Recommendations by the Recommender Engine	**High**
UC-5.3.1	Manage Goals for the Action Plan	**High**
UC-5.3.2	Manage Actions for the Action Plan	**High**
UC-5.3.3	Weekly View of Actions	**High**
UC-5.3.4	Manage Actions of the Running Action Plan	**High**
UC-5.3.5	System Reminds Patient to Take Medications or for Upcoming Visits and Tests	High
UC-5.3.6	Weekly Feedback and Evaluation	**High**
UC-5.3.7	Manage Action Plan Process Model	**High**
UC-5.3.8	View Calendar	**High**
UC-5.3.9	Extract Printable Calendar	Low
UC-5.4.1	Manage Information Material	Normal
UC-5.4.2	Assess Information Material	Normal
UC-5.4.3	Explore Patient Information Material	Normal
UC-5.4.4	Communication with Persons Sharing Similar	Normal

	Situations	
UC-5.4.5	Provide help information	Normal
UC-5.4.6	Patient views information material about diabetes-specific nutrition	Normal
UC-5.4.7	Import EHR and/or PHR Data to the used PHR System	Normal
UC-5.4.8	Retrieve PHR Data from the used PHR System for the EMPOWER Components	**High**
UC-6.1.1	Collection of ODLs	**High**
UC-6.1.1.1	Collect ODLs Manually	**High**
UC-6.1.1.2	Collect ODLs Automatically[5]	Normal
UC-6.1.2	Provide Mental Health Support	Normal
UC-6.1.2.1	Assess Mood	**High**
UC-6.1.2.2	Provide Stress Control Support	Low
UC-6.1.2.3	Provide Sleep Support	Normal
UC-6.1.4	Synchronize PHR Data with PHRS	Normal
UC-6.1.5	Record Medication Changes	Normal
UC-6.2.1	Manage Physical Activities	**High**
UC-6.2.1.1	Record Physical Activity	**High**
UC-6.2.2	Food Diary recording Eating Behaviours	**High**
UC-6.2.3	Evaluating Food Diary	High
UC-6.2.4	Calculate Nutrients Distribution	**High**
UC-6.2.5	Nutritional Information Material	Normal
UC-6.2.6	Insulin Reminder	Normal
UC-6.2.7	Manage Observations of Daily Living	Normal
UC-6.3.1	Select User Interface Modality	Normal
UC-6.3.1.1	Voice Input	Normal
UC-6.3.1.2	Speech Command	Normal
UC-6.4.1	Import EHR Data using IHE XPHR	**High**
UC-6.4.2	Import EHR Data through epSOS Network	Low
UC-6.4.3	Import PHR Data using IHE XPHR	Normal
UC-6.4.4	Export Data from PHR	Low

Table 1 Use Case Overview

3.2 Technical Requirements Summary

Also the technical requirements have been ranked according to the priorities high, normal and low. The complete list of functional requirements as identified in D3.2.1 is available in Table 2.

[5] Equipment to be used not yet decided

Reference[6]	Functional Requirements	Priority
Pathway Engine		
D3.2.1-9.10.3.1	1. The Pathway engine shall also provide RESTful services to consumers for monitoring processes and process tasks.	Low
D3.2.1-9.10.3.1	2. The Pathway engine and its process engine will communicate and interact with EMPOWER user via the EMPOWER inbox	High
D3.2.1-9.10.3.1	3. The Pathway engine shall provide the means to load and execute Knowledge models	High
D3.2.1-9.10.3.1	4. It is desired to load and run knowledge models or BPM process models from EMPOWER components e.g. from the Action Plan Engine or a PHRS.	High
D3.2.1-9.10.3.1	5. Pathway engine shall provide the necessary data to the process engine relevant to a particular process model	High
D3.2.1-9.10.3.1	6. Pathways engine shall access registered user data form metadata used in EMPOWER to support process models, interoperability, and persistence.	High
D3.2.1-9.10.3.1	7. Interactions with users (devices, sensors, information systems) shall provide the process engine with required data based on requirements of the process models. For example, an ODL is submitted and the required data is stored	High
D3.2.1-9.10.3.1	8. Process engine shall provide common services to exchange information between patient (or medical professional) and EMPOWER components e.g. Inbox related to patient or medical patient processes and tasks	High
D3.2.1-9.10.3.1	9. Security authentication and authorization shall be integrated and observed.	High
D3.2.1-9.10.3.1	10. If messages for the Action Plan Engine are received, it shall attempt to process them, otherwise, it shall forward to the Action Plan engine for processing.	High
EMPOWER Inbox Engine		
D3.2.1-9.10.3.2	1. The inbox engine provides services for PHRS client implementations to support the extension of existing PHRS inbox functionalities	Normal
D3.2.1-9.10.3.2	2. It shall allow reading messages from Inbox.	High
D3.2.1-9.10.3.2	3. It shall allow removing read and unread messages from Inbox.	Normal
D3.2.1-9.10.3.2	4. It shall allow classification of read messages as unread.	Normal
D3.2.1-9.10.3.2	5. It shall offer classification of messages using EMPOWER terminologies	Normal
D3.2.1-9.10.3.2	6. It shall include classification of messages using EMPOWER terminologies from other sources (system or other user)	Low
D3.2.1-9.10.3.2	7. It shall offer organization of messages according to the classifications	Normal
D3.2.1-9.10.3.2	8. It shall allow deletion of an incoming message	Normal
D3.2.1-9.10.3.2	9. It shall allow the system to perform housekeeping on messages sent by system to reduce information load on the user.	Normal

[6] Contains deliverable number (e.g. D3.2.1) and section number (e.g. 9.10.3.1)

D3.2.1-9.10.3.2	10. It shall provide services for status and metrics of inbox for total messages and by classification category, unread messages.	Normal
D3.2.1-9.10.3.2	11. It shall provide services to create or extend PHRS/PHA client inbox applications.	Normal
Reporting Module		
D3.2.1-9.10.4.1	1. It provides medical data to the user on the screen of personal computer, mobile device or information kiosk.	High
D3.2.1-9.10.4.1	2. It processes PHR data that is stored conforming to a standard format.	Normal
D3.2.1-9.10.4.1	3. It provides printing functionality for the desktop version.	Normal
D3.2.1-9.10.4.1	4. It allows users to set up the interface (scale, data to be displayed, font size, colour schema) and to save the settings.	Low
D3.2.1-9.10.4.1	5. It provides template modelling functionality for the knowledge worker in order to define new graphical representations to support user specific views.	Low
Security Module		
D3.2.1-9.10.5.1	1. It shall provide access to security related configuration	High
D3.2.1-9.10.5.1	2. It shall provide authorization services	High
D3.2.1-9.10.5.1	3. It shall provide configuration option to override secure communications during simulation or testing. The configuration component provides access to this configuration.	High
D3.2.1-9.10.5.1	4. There shall be support for IHE ATNA. OpenATNA software provides auditing with IHE profile support, application for visualization. This application shall be secured and available to particular authenticated users	Normal
D3.2.1-9.10.5.1	5. It shall provide a service to for components to send Audit log messages. EMPOWER Services must be assigned identifiers in the EMPOWER configuration.	High
Consent Editor		
D3.2.1-9.10.5.2	1. It shall allow multiple executions of different Consent Editors for the same patient.	Low
D3.2.1-9.10.5.2	2. It shall allow to display the available hospitals.	Low
D3.2.1-9.10.5.2	3. It shall allow to choose a hospital among the available.	Low
D3.2.1-9.10.5.2	4. It shall allow to display the available Medical Professionals of the selected hospital	Normal
D3.2.1-9.10.5.2	5. It shall allow to add Medical Professionals in the “Given Consent” list.	Normal
D3.2.1-9.10.5.2	6. It shall allow to display the list of the Medical Professional(s) the patient has given consent.	High
D3.2.1-9.10.5.2	7. It shall allow deleting multiple Medical Professionals from the “Given Consent” list.	High
D3.2.1-9.10.5.2	8. It shall allow to view contact information of the Medical Professionals in the “Given Consent” list.	High
Consent Engine		
D3.2.1-9.10.5.3	1. It shall either grant or deny a data access attempt based on the consent of the patient.	High
D3.2.1-9.10.5.3	2. This component will be accessed by the EMPOWER components. Therefore, this component can be reachable by Web service calls.	High
Configuration Module		

D3.2.1-9.10.5.4	1. Configuration tool shall provide services to EMPOWER components to access configuration properties either using simple property and/or XML based configuration.	High
D3.2.1-9.10.5.4	2. EMPOWER Services must be assigned identifiers in the EMPOWER configuration. This is used by the consent manager and audit trail log services	High
D3.2.1-9.10.5.4	3. Configuration shall provide configurations relating to particular profiles e.g. relevant for a particular deployment e.g. pilot application	High
D3.2.1-9.10.5.4	4. Configuration tool shall configure the Pathway Engine, Action plan engine configuration, terminologies. Where feasible, updates to the configuration can be performed when the system is running.	High
D3.2.1-9.10.5.4	5. Configuration tool shall provide services to access resource bundles for localization.	Normal
D3.2.1-9.10.5.4	6. Configuration tools shall provide services for localization of terminologies. This can use the resource bundles for localization of properties.	Normal
D3.2.1-9.10.5.4	7. Configuration tools shall provide access to common EMPOWER sources.	Normal
Identity Manager Module		
D3.2.1-9.10.5.5	1. Identity lookup (namespace & ID) and management	High
D3.2.1-9.10.5.5	2. Identity includes role (original, mapped to EMPOWER role), institution identifiers, Demographic data	High
D3.2.1-9.10.5.5	3. Should connects to Identity provider (adaptors)	Normal
D3.2.1-9.10.5.5	4. EMPOWER default adaptor should access static data sources (LDAP) rather than directly to institutions (supports pilot app and test scenarios)	Normal
D3.2.1-9.10.5.5	5. Identifiers should connect namespace and patient identifier in electronic health records, EHR or other sources	Normal
D3.2.1-9.10.5.5	6. Roles in other systems or PHRS should be mapped to common EMPOWER roles	High
D3.2.1-9.10.5.5	7. Identities for test users, non-medical related such as administrators, etc are provided in the LDAP configuration and assigned appropriate roles	High
D3.2.1-9.10.5.5	8. It is desired to offer identity mapping service to obscure patient identifiers on User interfaces or in applications	Low
D3.2.1-9.10.5.5	9. Shall provide default configurations for the pilot applications. The configuration component provides a selector for the particular profile	Normal
D3.2.1-9.10.5.5	10. Shall provide default configuration for demo and testing configurations. The configuration component provides a selector for the particular profile	High
Identity Provider Module		
D3.2.1-9.10.5.6	1. A test data source to simulate the connection to institution	High
D3.2.1-9.10.5.6	2. Secure Interface requirements	Normal
Audit Record Repository		
D3.2.1-9.10.5.7	1. It shall allow the EMPOWER components to send audit logs conforming to IHE ATNA Profile.	High
D3.2.1-9.10.5.7	2. It shall provide a GUI to System Administrators.	High
D3.2.1-9.10.5.7	3. It shall allow the System Administrators to monitor EMPOWER system operation.	High
Audit Trail/Log Sender		

D3.2.1-9.10.5.8	1. It shall allow the EMPOWER components to send audit logs to ARR.	High
D3.2.1-9.10.5.8	2. It shall allow the EMPOWER components to conform to IHE ATNA Profile.	High
Node Authentication Submodule		
D3.2.1-9.10.5.9	1. It shall provide confidentially and integrity of the communication.	High
D3.2.1-9.10.5.9	2. It shall provide authentication of the EMPOWER components.	High
D3.2.1-9.10.5.9	3. It shall provide these functionalities by conforming to IHE ATNA Profile.	High
Recommender Engine Module		
D3.2.1-9.11.3.1	1. It shall allow multiple executions of one MP-DG for different patients.	High
D3.2.1-9.11.3.1	2. It shall allow multiple executions of different MP-DGs for the same diabetes patient.	High
D3.2.1-9.11.3.1	3. It shall allow to suspend an execution.	High
D3.2.1-9.11.3.1	4. It shall allow to resume a suspended execution.	High
D3.2.1-9.11.3.1	5. It shall allow to abort an execution.	High
D3.2.1-9.11.3.1	6. It shall allow to view the results of previous MP-DG executions.	High
D3.2.1-9.11.3.1	7. It shall show all the executing MP-DGs and their statuses.	High
D3.2.1-9.11.3.1	8. It shall execute Recommendation Rules specified about the diabetes patient and shall provide direct recommendations to the medical professionals.	High
MP-DG System		
D3.2.1-9.11.3.2	1. The MP-DG System shall allow to create Machine Processable Diabetes Guidelines both graphically and manually.	Normal
D3.2.1-9.11.3.2	2. The MP-DG System shall allow to assign a MP-DG to a diabetes patient.	High
D3.2.1-9.11.3.2	3. The MP-DG System shall display the already existing MP-DGs to the medical professional.	Normal
D3.2.1-9.11.3.2	4. The MP-DG System shall display the recommendations to the medical professionals and allow them to edit the recommendations.	High
Recommender Rule Editor		
D3.2.1-9.11.3.3	1. The Recommender Rule Editor shall provide a GUI to help the medical professional to create recommendation rules.	Normal
D3.2.1-9.11.3.3	2. This component shall allow the user to delete an existing rule.	Normal
D3.2.1-9.11.3.3	3. This component shall allow the user to assign a rule to a diabetes patient.	Normal
D3.2.1-9.11.3.3	4. This component shall allow to personalize a rule/rule template to a diabetes patient.	Normal
Action Plan Engine Module		
D3.2.1-9.11.4.1	1. The Action Plan Engine (APE) shall provide Action plan services to support PHRS Action Plan client application	High
D3.2.1-9.11.4.1	2. Shall provide communications with the EMPOWER inbox via the Pathway engine	High
D3.2.1-9.11.4.1	3. Shall access the configuration from configuration component	Normal

D3.2.1-9.11.4.1	4. Shall share the process engine from the Pathway engine or utilize its own.	High
D3.2.1-9.11.4.1	5. Shall coordinate a patient's action plan with the Action plan knowledge model executed by the process engine	Normal
PHR system		
D3.2.1-9.11.5.1	1. PHRS shall utilize EMPOWER services to support client applications.	High
D3.2.1-9.11.5.1	2. PHRS shall provide Identity provider to support user authentication from EMPOWER PHAs.	Normal
D3.2.1-9.11.5.1	3. PHRS shall integrate inbox or extend an existing inbox with EMPOWER services	Normal
D3.2.1-9.11.5.1	4. PHRS shall integrate the EMPOWER Dashboard including navigation features and EMPOWER Inbox.	Normal
D3.2.1-9.11.5.1	6. PHRS applications (PHAs) for capturing and managing ODLs: physical/lifestyle activities, Vital Physical Mental parameters	High
D3.2.1-9.11.5.1	7. PHRS for importing and managing Patient information from heterogeneous data sources	Normal
D3.2.1-9.11.5.1	8. PHRS for Consent management where patient controls access to health info by services, applications and users	Normal
D3.2.1-9.11.5.1	9. PHRS for managing Patient Action Plan	Normal
D3.2.1-9.11.5.1	10. PHRS for exploring Patient Information materials and integration of materials with other PHRSs	Low
D3.2.1-9.11.5.1	11. ALL PHRSs and PHRs: Shared CSS style/themes Shared UI resource bundles and localization and Terminologies	Low
D3.2.1-9.11.5.1	12. Integration of patient materials, EMPOWER terminologies (standard and domain specific)	Low
D3.2.1-9.11.5.1	13. This PHRS supports the collection, management and visualization of a patient's Physical and Lifestyle parameters	Normal
Nutritional Information Submodule		
D3.2.1-9.11.5.3	1. It shall allow reading articles on nutrition.	Normal
D3.2.1-9.11.5.3	2. It shall allow reading pieces of advice on nutrition and diabetic diet.	Normal
D3.2.1-9.11.5.3	3. It shall allow reading articles on diabetes in general.	Normal
Diabetes Social Network		
D3.2.1-9.11.5.4	1. It shall allow the patient to add new friends in the Diabetes Social Network by sending requests to other participants.	Normal
D3.2.1-9.11.5.4	2. It shall allow posting pieces of advice on personal profile.	Normal
D3.2.1-9.11.5.4	3. It shall allow reading pieces of advice of friends' profiles.	Normal
D3.2.1-9.11.5.4	4. It shall allow adding personal information.	Normal
D3.2.1-9.11.5.4	5. It shall allow the patient to create a new group and invite friends to participate.	Normal
D3.2.1-9.11.5.4	6. It shall allow the patient to create a new event and invite friends to attend.	Normal
D3.2.1-9.11.5.4	7. It shall allow participating in a group a friend has created.	Normal
D3.2.1-9.11.5.4	8. It shall allow attending an event a friend has created.	Normal
Daily Living Forms		
D3.2.1-9.12.3.1	1. It shall allow multiple insertions in the Daily Living Forms for the same patient.	Normal

D3.2.1-9.12.3.1	2. It shall provide the patient with upcoming events in respect to their diabetes treatment based on their health condition. This includes actions and goals to be achieved by the patient. This info will originate from the Action Plan Engine component. Related to this, the following detailed functional requirements are applicable:	Normal
D3.2.1-9.12.3.1	a. It shall allow viewing a list of the goals the Medical Professionals have set for the patient.	Normal
D3.2.1-9.12.3.1	b. It shall allow reading details of the goals.	Normal
D3.2.1-9.12.3.1	c. It shall allow reading a list of the reminders.	Normal
D3.2.1-9.12.3.1	d. It shall allow popping up a message reminding the patient of an upcoming visit or test or to take a medication.	Normal
D3.2.1-9.12.3.1	e. It shall allow viewing the calendar	Normal
D3.2.1-9.12.3.1	f. It shall allow reading details of the events included in the calendar.	Normal
D3.2.1-9.12.3.1	g. It shall allow extracting the Calendar in a printable version.	Normal
D3.2.1-9.12.3.1	3. It shall allow storing the Calendar in a printable version on the patient's smartphone	Normal
D3.2.1-9.12.3.1	4. It shall allow the patient to view graphical representations of the Daily Living data.	Normal
D3.2.1-9.12.3.1	5. It shall allow limiting the graphical representation in a specific time interval.	Normal
D3.2.1-9.12.3.1	6. It shall allow viewing past meals consumed.	Normal
D3.2.1-9.12.3.1	7. It shall allow sending the graphical representations of the glucose level, weight and exercise results to the Medical Professional	Normal
D3.2.1-9.12.3.1	8. It shall allow multiple insertions in the Daily Living Forms for the same patient.	Normal
Mental Health Collector		
D3.2.1-9.12.3.2	1. It shall allow writing notes in a diary form about depression, stress and insomnia symptoms.	**High**
D3.2.1-9.12.3.2	2. It shall allow reading past diary entries.	**High**
D3.2.1-9.12.3.2	3. It shall allow reading articles on depression, stress and insomnia.	Normal
D3.2.1-9.12.3.2	4. It shall allow reading pieces of advice on depression, stress and insomnia.	Normal
D3.2.1-9.12.3.2	5. It shall allow listening to music for relaxation.	Normal
D3.2.1-9.12.3.2	6. It shall allow listening to instructions on how to keep a normal breathing tempo.	Low
Physical/Lifestyle Activity Data Collector Submodule		
D3.2.1-9.12.4.1	1. It shall allow the user to add, remove or update entries	High
D3.2.1-9.12.4.1	2. It shall allow the patient to view graphical representations of the Daily Living data.	Normal
D3.2.1-9.12.4.1	3. It shall allow the view of a single entry	Normal
D3.2.1-9.12.4.1	4. It shall allow viewing of history and sorting of information by column title.	High
D3.2.1-9.12.4.1	5. It shall help the user build a report for either download or to send to a medical professional. The report builder provides choices for the temporal range and types of information to include or exclude. Also, the builder provides options to include or exclude identifiers such patient name, EMPOWER user email identifiers, etc	Low
EHR/PHR Data Manager Submodule		

D3.2.1-9.12.5.1	1. The EHR/PHR Data Manager Submodule shall allow connecting to the epSOS Network.	Low
D3.2.1-9.12.5.1	2. This tool shall allow the patients to gather data from Hospital Information System through IHE XPHR.	High
D3.2.1-9.12.5.1	3. This tool shall allow the patient to send their medical data from PHR to EHR system through XPHR.	Normal
D3.2.1-9.12.5.1	4. This tool shall allow the patients to gather data from external PHR and PHAs through IHE XPHR.	Normal
XPHR Manager		
D3.2.1-9.12.5.2	1. It shall allow exchanging data through IHE XPHR.	High
D3.2.1-9.12.5.2	2. It shall provide mechanisms to insert the received data to the PHR's databases.	High

Table 2: Functional Requirements Overview

From the pilot application some high level requirements were stated, and initially mentioned in D8.1.1. Their importance has been estimated and priorities were specified as provided in Table 3.

Pilot Application Requirements	**Reference**[7]	**Priority**
Provide an Electronic Version of German Diabetes Passport	D8.1.1-3.1.1.1	Normal
Provide a PHRS for the pilot in Germany.	D8.1.1-3.1.2.2	High
Integration with patient terminals (Kiosks), which are not yet available.	D8.1.1-3.1.2.5	Low
Data download from medical health card (eGK), e.g. via 3rd party card reader.	D8.1.1-3.1.2.5	Low
Generate a customized guideline that is exclusive to EMPOWER	D8.1.1-3.2.1	High
Integration of data from Diabetes Minimum Health Data Set	D8.1.1-3.2.1.2	Normal
Integration with National Health Information System (via eSaglikKaydim PHR)	D8.1.1-3.2.2.1	High
Support of ASTM's CCR Standard (via eSaglikKaydim PHR)	D8.1.1-3.2.2.1	High
Integration with Health Coding Reference Server	D8.1.1-3.2.2.1.3	Normal
Integration with "Doctor Data Bank"	D8.1.1-3.2.2.1.4	Normal
Accordance to Ethical and Legal Requirements in Germany	D8.1.1-5.1	High
Accordance to Ethical and Legal Requirements in Turkey	D8.1.1-5.2	High
Interoperability with DocStar or other EHR.	D8.1.1-6.2.1	Normal
Development of "yet another blood-glucose monitoring/measurement tool" is explicitly excluded.	D8.1.1-3.1.2.1	-

Table 3: Pilot Application Requirements Overview

4 Architecture Constraints

The following general architectural constraints are derived after a careful analysis of the use cases section. They provide the basis for all further considerations.

4.1 Basic Technical Design Aspects

The following basic design decisions need to be considered for the further development of the architecture as well as the implementation:

[7] Contains deliverable number (e.g. D8.1.1) and section number (e.g. 3.1.1.1)

- The heterogeneous aspect of the EMPOWER system favours a Service-oriented Architecture (SOA).
- REST services are preferred vs. SOAP services. Most of the components require providing one or more service end points.
- Following concerns (aspects) are considered to be Cross-cutting concern and they must be applied transparent: security (authentication and authorization), logging (and audit), journaling, etc. The complete list with the concerns that are considered cross-cutting is provided in Section 0. Most of these concerns (aspects) are dependent on the underlying technologies.
- Transactional support is required. Transaction related details (e.g. rollback, concurrency, timeout, propagation, nested transaction, etc.) will be defined in Section 10.4; most of the enumerated transaction details will rely on the underlying technologies.

The actual architecture will have four logical layers, which are *presentation*, *application*, *business* and *persistence* layer[8]; the (end) users interact only with the presentation layer. All the involved components must be located in one of these layers. Each of the layers must be interchangeable. A list with all the layers (and the components that belongs to) will be provided in the next section. Cross-cutting concerns (upper mentioned) must be applicable across more layers.

4.1.1 Messaging

Message exchange between the single EMPOWER components will be required. The communicating components should take the following into account:

- Asynchronous message exchanging is required. According to the use cases, both standard asynchronous messaging exchange models are required: direct Point-to-Point component communication (e.g. ODL collection use case) and Publish-Subscribe via a central message instance (e.g. the inbox use case).
- Message routing abilities are also required. According with the use cases and the requirements, the message end point can/may be unknown in the moment when the message is send. In these cases the message end point(s) can be inferred (depending on the message content and meta-data) and routing abilities are required in order that the send message reach the needed end point(s).
- Messaging exchange patterns (e.g. request-reply, call-back, etc.) are also required.

It is presumed that extensive message related handling and operations are required (e.g. enrichment, trimming or message filtering, routing like described by Enterprise Integration Patterns (EIP)[9]. Therefore an integrated messaging solution within an Enterprise Service Bus (ESB) is preferred instead of a proprietary one. Messaging and ESB are relevant to the Use Cases involving the Pathway engine, ODL data access between EMPOWER and the PHRS or EHRS, or any interoperability related Use Case.

4.1.2 Configuration

The configuration is considered a central aspect to enable an easy configuration management among different deployments. There must be only one syntax and semantic to configure various aspects of the EMPOWER system. The configuration can be accessed (and eventually updated) via configuration service endpoints. It will be also possible to apply

[8] It is also reasonable to think that each layer is formed from sub-layers but at the moment only the upper listed layers are considered for simplicity reasons.
[9] EIP – Enterprise Integration Patterns

security constrains over the configuration (e.g. sensitive configuration can be access only by authorized actors).
Configuration is structured as tree of key – value pairs. If a configuration parameter is compound from different atomic configuration parts (e.g. the protocol:host:port/context/path is from 5 parts) it is preferable that each part has its own configuration entry (e.g. for the “protocol:host:port/context/path” is reasonable to provide 5 configuration entries). The exact syntax and the semantic for the configuration will be provided by the implementation of the single components.

4.1.3 Security

Identity of the users must be unique over the entire EMPOWER system. Different options are possible to accomplish this goal (e.g. single sign on, identity propagation, etc.). Details are described within the corresponding building blocks in the sections below.

4.1.4 Data materialisation (serialisation)

Standardised data serialisation formats like XML and JSON need to be used. Binary serialisation and proprietary formats (even if they are ASCII based) are discouraged. Standards format like (HL7 v3 for medical data) will be preferred. Custom created models (independent of language) are discouraged. The finally utilised standards will be defined in the knowledge models of Deliverable D3.4.1.

4.1.5 Test readiness

Each (main) component must be testable (against its contract, i.e. the API specification). The proof of the functionality (test) can be done together with other components or standalone. For the standalone case, any dependencies of other components must be emulated (mocked). The tests must be platform independent and they must run “out of the box”.

4.1.6 Deployment

The system must be deployable on a single machine but it must be also able to be deployed on several machines interconnected via network (running TCP/IP protocol).

4.2 Technical Constraints

In these subsections, specific technical constraints are listed in order to follow a common direction in the EMPOWER project. As EMPOWER system consists of several components, including communication to external, already existing systems, a further breakdown by components was made. As far as already known, the requirements and constraints are provided on a per-component basis.

4.2.1 Recommender Engine

- Hardware Requirements: This recommender engine will run on a desktop machine. A moderate machine with 4GB of RAM and 3 GHz of processor will be sufficient.
- Software Requirements: The graphical user interface of this component will be implemented through HTML5. Therefore any browser supporting this version of HTML will suffice. In addition to this communication with the server side will be accomplished through RESTful services. During the development the latest version of the Jersey[10] will be used on top of the latest version of Tomcat. In the persistency layer, the MySQL database is preferable. However, any database system can be used.
- Operating System Requirements: There is no constraint for this item.

[10] http://jersey.java.net/

- Programming Requirements: On the client side HTML5 (together with JavaScript) and on the server side latest version of Java will be used. Additionally, for the definition of recommendation rules Drools Rule Engine[11] will be used.

4.2.2 EHR/PHR Data Manager

- Hardware Requirements: This component will run on both desktop machine and mobile device. A moderate desktop machine with 4GB of RAM and 3 GHz of processor will be sufficient. On the device side, a device at least 1 GHz of processor and 1 GB of memory are required.
- Software Requirements: The graphical user interface of this component will be implemented through HTML5. Therefore any browser supporting this version of HTML will suffice. In addition to this communication with the server side will be accomplished through IHE XPHR[12] and IHE XDS[13] specification which uses MTOM[14] based HTTP SOAP messages. For this communication the latest versions of Axis2 and Tomcat will be used. In the persistency layer, the MySQL database is preferable. However, any database system can be used.
- Operating System Requirements: There is no constraint for the desktop version of this module. For the device, Android 4 is required.
- Programming Requirements: On the client side HTML5 (together with JavaScript) and on the server side latest version of Java will be used.

4.2.3 PHA - mobile

- Software development environment: Eclipse Helios v3.6 IDE with Android SDK v.1.6
- Programming language: JAVA
- Operating System Requirements: Android 3.2 Platform (Honeycomb) ,13 API Level for Tablets, Android 4.0 & 4.0.3 Platform (Ice Cream Sandwich) , 15 API Level for smartphones
- Network Protocols: Bluetooth Stack 2.3 or later (RFCOMM) , HTTP
- Database Management: SQLite

4.2.4 PHA - desktop

- Front-end Programming: JQuery plugins, AngularJS Javascript framework;
- Front-end IDE: Webstorm (Front-end)
- Backend services: RESTful services
- Backend programming language: JAVA
- Database: NoSQL preferably (MongoDB)

4.2.5 intLIFE PHR-S

- Hardware requirements: Intel Pentium, 4 CPU (4 X 3 GHz), RAM 4GB, 80G Hard Disk Partitioning in 40G, 40G, (Preferred with SATA Connectors)
- Operating System: LINUX Operating system, one of the following: Oracle Enterprise Linux 4 Update 7, Oracle Enterprise Linux 5 Update 2, Red Hat Enterprise Linux 4 Update 7, Red Hat Enterprise Linux 5 Update 2, SUSE Linux Enterprise Server 10 SP2, SUSE Linux Enterprise Server 11
- Software requirements: J2SE 5.0, Java SE 6, Glassfish Application Server v2, Apache + PHP server, MySQL db server, Oracle Express DB Server, VLC Media player (latest edition)

[11] http://www.jboss.org/drools/
[12] http://wiki.ihe.net/index.php?title=Exchange_of_Personal_Health_Record_Content
[13] http://wiki.ihe.net/index.php?title=Cross-Enterprise_Document_Sharing
[14] http://www.w3.org/TR/soap12-mtom/

4.2.6 New EMPOWER Components

In this sub-section, the general constraints are collected, that need to be considered when new components are built for the EMPOWER project.

4.2.6.1 Hardware Constraints

- 4 GB RAM
- 10 GB Hard Disk space
- For the end user (client) a display is required (XServer)
- TCP/IP network abilities (fire walls and security relevant operation may be necessary)

4.2.6.2 Operating System Constraints

- All the software must be able to be deployed and run on a UNIX-like system. As standard platform an Ubuntu Server 12.04 LTS will be provided. Support for other platforms may be provided, if necessary, e.g. an Enterprise Linux distribution to support Oracle XE.
- 64-bit platforms are preferred compared to the 32-bit ones.
- Snapshot abilities: we need to be able to take a system snapshot on a certain moment in time.
- Backup is provided in the development and testing deployments (for details refer to Section 9.2). For the pilot application deployments, the system should be integrated to the existing backup solutions.

4.2.6.3 Software Constraints

- Most of the software will be produced in Java or in other JVM-compatible framework (e.g. Groovy, Scala, Ceylon, etc).
- Java 1.7.0 in its open source release (openJDK) is encouraged.
- For the presentation layer in web browsers, JavaScript will be also used together with a suite of JavaScript libraries (like JQuery >= 1.7, AngularJS (MVC, MVVM framework) or other appropriate framework supporting Javascript application building), RDFa or microformats JSON, JSON-LD, etc.)
- HTML5-features should be compatible with the chosen browsers. Alternatively a special conversion library like modernizer[15] could be used to support HTML5 features.
- REST WS implemented in Java must follow the JSR 311 specification[16]
- Java-based Web Applications must follow the Servlet 3.0 API[17]
- For the process model declaration we will use BPMN 2.0[18] will be used. More details and argumentation is provided in Section 6.2 and in the description of the relevant architecture model elements.
- If 3rd party software is required, permissive open source licenses should be preferred. This option can be discussed in details depending of the open source license type and on the functionality to be implemented.
- All the possible IDE are allowed but to enable the use of a continuous integration server (CI-server) the code must compile, test and run IDE-independent.
- As CI-server a Jenkins[19] installation will be provided.
- For UML modelling Visual Paradigm 10 (e.g. Community Edition) is used.

[15] Modernizr (http://modernizr.com/) is a JavaScript library that detects HTML5 and CSS3 features in the user's browser. Taking advantage of cool new web technologies is great fun, until you have to support browsers that lag behind. Modernizr makes it easy for you to write conditional JavaScript and CSS to handle each situation, whether a browser supports a feature or not. It's perfect for doing progressive enhancement easily
[16] http://jcp.org/en/jsr/detail?id=311
[17] http://jcp.org/aboutJava/communityprocess/final/jsr315/index.html
[18] http://www.omg.org/spec/BPMN/2.0/
[19] http://jenkins-ci.org

- Generated code is allowed as far as it does not create dependency on any tool; if generating code is required the code generation process must be runnable from the command line.
- For persistence the use of PostGreSQL (for relational storage), MongoDB (for document based storage) and H2 (for demonstration and testing purposes) are preferred.
- Browser Requirements:
    - Support for Mozilla Firefox and Google Chrome available at beginning of 2013
    - Other Browsers may work but will not be tested extensively.

4.3 Conventions

4.3.1 Naming Conventions

- Directory and file names with whitespaces must be avoided, use "-" instead.
- Path and script names must be case-insensitive (7-bit ASCII).

4.3.2 Package Structure

- As high-level package structure the following breakdown is proposed (in a "main" folder):
    - eu.empower.<component>.api for the definition of the external interfaces of this component.
    - eu.empower.<component>.model
    - eu.empower.<component>.impl for the implementation of the classes
    - eu.empower.<component>.util for common purpose classes
- Unit Tests should be placed in the same package as the class under test, but located in the "test" folder, suffixed with "Test"
- Executable examples can also be placed in the test tree, but with suffix "Example"
    - Suitable documentation about parameters and eventually pre- and post-conditions is expected for examples.

4.3.3 Versioning

- All the EMPOWER sources and related resources (e.g. i18n, etc.) must be stored in the common repository
- As VCS, the cloud service from Bitbucket[20] is used. It allows non-public code sharing using "git"[21].
- Milestones and major releases will be labelled (tagged) by SRFG.
- Each component is versioned independently, but current version and release must be available in the root of the source tree (e.g. as version.properties file).

For the final prototype an EMPOWER version number 1.x is targeted. During the project runtime SNAPSHOTS will be provided, interim MILESTONEs will be defined and tagged as required. Before the FINAL RELEASE there will be RELEASE CANDIDATES (RC) available during the pilot phase. The software life cycle is specified in detail as follows (n=number):

- Snapshot-1.0 - Snapshot-n.n: development versions, there is very less integration between components, these releases purpose is to publish components (common API) for further tests.
- 0.1 - 0.n integration between components plays a crucial role here, the public components (common API) are engaged here in workflows.
- 1.00-RC1 - 1.00-RCn most of the components are ready (implemented) and they can communicate with each other. Those are the version designated to the

[20] https://bitbucket.org
[21] http://www.git-scm.com

complex test (close to the real life situations). This may be the beginning of the pilot phase.
- 1.00 Release all the components are ready there are no major bug/problems (reported in the previous phases).
- 1.0n Bug-fix release – enrichments and bug fixes detected in the last test phase.
- Eventually: Interim milestones (e.g. Review 1 milestone)

4.3.4 Source Code Structure

- The source code management (SCM) is handled using "git". The top-level structure of the repository is as follows:
    - "master-module": All EMPOWER components are located below this directory. This directory contains the master Maven pom.xml file.
    - "bin": scripts for general purpose (like empty module generation)
    - "doc": General documentation like UML-Diagrams, but no automatically generated documentation like Javadoc.
- Each top-level directory will contain a readme-file for proper use.
- The code provided by partners is in general placed within sub-modules of the master-module, and will be structured following the Maven standard directory layout[22].

4.3.5 Issue Tracking

- For issue tracking the issue tracker provided with Bitbucket is used. This should be the central place for bug reports, feature requests, etc. In the SCM commit-messages links to the solved issues should be provided.

5 System Scope and Context

Based on the System overview presented in Section 3 (Figure 1: EMPOWER Patient Empowerment Framework), the following types of external systems are identified as the relevant communication partners. To set the EMPOWER System into its context, a short description for each of the neighbouring components is provided in this chapter. The neighbouring components are mainly other health applications that need to exchange data to/from EMPOWER.

5.1 Electronic Health Record System

Electronic Health Records (EHRs) are used to digitally record health-related information on a per-patient basis, managed by the government. EHRs usually consist of a big range of data, from patient demographics to medical history, medication, laboratory results, but also binary data like radiology images or scanned doctor reports. Data exchange with EHRs is handled via XPHR as further described in Section 8.2. It must be noted that the EMPOWER user has an independent User-ID from the EHR Patient-ID or PhysicianID. Therefore mapping between the different identities must be managed within EMPOWER in order to query updates of the patient records and match it to existing EMPOWER PHRS users. A Master Patient Index such as OpenEMPI[23] is one alternative, however, another common approach would be to create a federated identity (EMPOWER PHRS user) to link users from an Identity provider. For example, patient or physician identities from one health record system are linked to the EMPOWER PHRS user. It is expected that each of the pilot applications will provide identity and contact information for both patients and medical professionals. This

[22] http://maven.apache.org/guides/introduction/introduction-to-the-standard-directory-layout.html
[23] http://openempi.kenai.com/

information will be available from Identity(s) provider specific for the pilot application rather than directly integrating to the EHR systems.

5.2 Personal Health Record System

In addition to the EMPOWER accompanying PHR system (PHRS), EMPOWER can interoperate with other existing PHR systems. One approach is to provide a plugin mechanism. In this case, there are existing PHR Users, which can directly access EMPOWER functionality without separate authentication mechanisms. The PHRS is responsible for authentication, user management, and either assigning users to particular EMPOWER authorisation roles.

5.3 Practice Information System

Practice Information System (PIS) in our context is considered as EMR software that combines additional functionality such as billing, scheduling or practice management. It is used in general practitioners offices that will take part in the German Pilot Application (e.g. DocStar). It typically offers very limited interoperability features, mostly only patient demographics import via BDT[24] or chipcard reader and import of medical data from devices like laboratory via LDT interface. IHE XPHR profile is not supported by the PIS. This requires EMPOWER-specific mechanisms to exchange data with the recommender engine and the PHR System.

5.4 Personal Health Applications

Personal Health Applications (PHAs) are usually operated by patients in a local environment, such as Desktop PCs, Smartphones or Tablets. Although patient data is directly managed by the patients, modern PHAs often provide data upload to web-based cloud services.

Supplementary EMPOWER tools will need to provide authentication and identity management and identity providers in order to facilitate the building of new personal health applications (PHAs).

5.5 Hospital Information System

A Hospital Information System (HIS) denotes the core/central information system of a hospital. Beside typical EMR functionality also administrative- and business-management features may be integrated. Interoperability is enabled by event-driven, message-based import and export interfaces that conform to the HL7 Version 2.X standard. Larger hospitals usually operate a central communication server that offers broad capabilities such as processing, transforming and routing of messages. The configuration of a new interface – like it would be required to interconnect to the EMPOWER system – is usually charged with a considerable amount of money for the required implementation and test of the interface.

6 Solution Ideas and Strategy

In this section, we describe candidate core technologies for use in the EMPOWER technology stack:

6.1 Service-oriented Architecture

Many SOA frameworks are available to support service oriented architectures. One key requirement is to utilize an integration framework that supports Enterprise Integration patterns (EIP). Apache Camel and Mule ESB are two prospective frameworks for EMPOWER.

These integration frameworks provide connectivity and adaptors to connect external systems using particular protocols and they provided Predefined EIP patterns to support the

[24] ADT, BDT, GDT and LDT (summarised as xDT) are acronyms for data exchange formats used in the German Health System to communicate health information to and from general practitioners' offices.

messaging infrastructure (smart routing, transformation, mediation, monitoring, and orchestration).

Both Camel and Mule can integrate with Business process engines and are configurable using XML, however Camel, provides additional capabilities to write Domain specific languages (DSL) in Java, Groovy, or Scala.

6.2 Rule-based Business Process Engine

To support the uptake of project results, EMPOWER will use an Open Source rule-based process engine; the license must also be permissive. The Business process engines must support BPMN2.0, human workflow integration and long executing stateful processes. The process engine and rule engine will support both rule-based business processes and in particular cases, processes regarding web page flows as required.

Two engines are considered: jBPM5 and the Activiti process engine support BPMN2.0 for process modelling. The process engine will utilize a rule engine based on available plug-ins. The JBoss Drools rule engine, for example, will be utilized by the process engine.

6.3 Multimodal Services and Interfaces

EMPOWER wants to reach a large number of diabetes patients, each of them with different abilities. The abilities are defined as maturity levels in different dimensions. Offering EMPOWER applications on different kinds of devices (e.g. Desktop-PCs with web browsers, Smartphones, Tablet-PCs) will enable access to a larger number of user groups. In addition optional input methods like voice input with speech commands are envisaged to support the acceptance of the provided solution.

6.4 Knowledge Models

A major feature of the EMPOWER approach is the emphasis on knowledge models. This will help to centralize the knowledge of medical information, business processes, and rules. The major advantage of using such knowledge models is to decouple the system architecture and implementation from the continuously changing and extending medical knowledge. The knowledge models itself can be based on various languages and formats (e.g. BPMN, ADL) and are described in a separate document (D3.4.1)

7 Building Block View

The EMPOWER subsystems description is divided into two levels: In the first level, the black-box view of the subsystems, including from their outside view and interactions with other subsystems are described. In a second level, the white-box descriptions of subsystems explain how the subsystems work internally.

7.1 Level 1 – Black Box Descriptions

The EMPOWER System is comprised of multiple subsystems as depicted in Figure 2. Two central subsystems are the *Configuration Subsystem* and the *EMPOWER Service Locator Subsystem*. All other subsystems are using those for retrieving their configurations ("get configuration") and locating other relevant services ("locate service"). The service locator subsystem also needs mechanisms to read information (descriptive meta-data) about the services (*Service Descriptor Subsystem*)[25]. These central components build up the *EMPOWER Core*. The other subsystems and their main functionality and packages are further described in the following subsections.

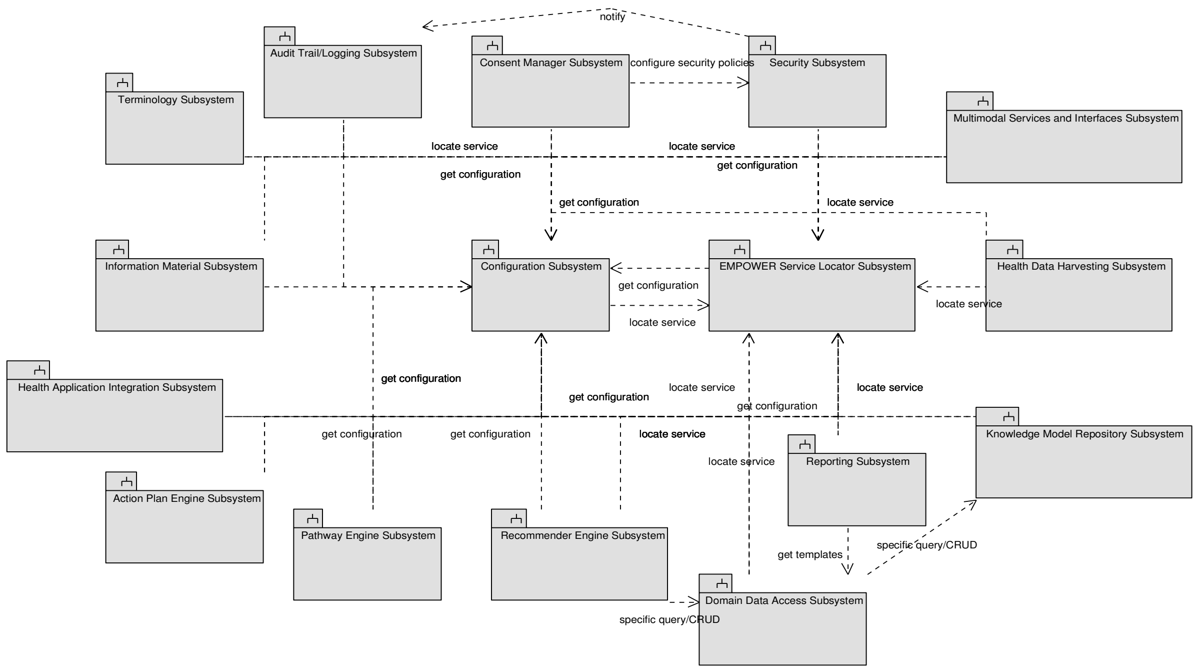

notify
Audit Trail/Logging Subsystem
Consent Manager Subsystem
configure security policies
Security Subsystem
Terminology Subsystem
Multimodal Services and Interfaces Subsystem
locate service
get configuration
Information Material Subsystem
Configuration Subsystem
EMPOWER Service Locator Subsystem
Health Data Harvesting Subsystem
Health Application Integration Subsystem
Knowledge Model Repository Subsystem
Reporting Subsystem
Action Plan Engine Subsystem
Pathway Engine Subsystem
Recommender Engine Subsystem
Domain Data Access Subsystem
get templates
specific query/CRUD

Figure 2: EMPOWER Subsystems view

7.1.1 Pathway Engine Subsystem

This subsystem (Figure 3) is responsible for many core features including messaging, execution of business process models including with tracking of user tasks, communication with users via an "Inbox" message queue, and monitoring of EMPOWER services and processes. To fulfil these tasks, a Notifier is provided in the pathway engine subsystem, to be used also by other components.

[25] The Service Descriptor Subsystem is not depicted as extra Model Element.

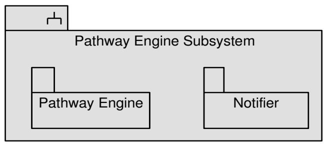


Figure 3: Pathway Engine – Black-box

The Self-Management Pathway (SMP) and Action Plan knowledge models will be comprised of sets of rule-based Business Process models (BPMs). A knowledge model includes both rules and other business logic utilizing EMPOWER terminologies, and EMPOWER services such as for the action plan and recommendations, patient information, user settings or patient state in the executing process model.

There can be more than one pathway knowledge model for diabetes patients; however, there is a default model and there can be models primarily for testing, demonstration or simulation (simulate user interaction). The choice of pathway model might also depend on the requirements of the PHRS, country or institution.

The process engine executes the process models and interacts with human actors via the EMPOWER inbox and other services. The engine can be notified and listen to activities and actions of the human user via the EMPOWER user's interactions with EMPOWER components. What the engine requires can be dependent on the BPM models. The Pathway Engine provides monitor components to monitor EMPOWER services, especially to notify the process engine and BPM related tasks or steps regarding the state of a particular action. For example, based on a process model, the monitor notifies the process engine when an ODL is accessed or stored. When stored, a particular step in a human process task is completed leading to the next step in the process model task. Communications between the EMPOWER user and Pathway engine or Action Plan engine are through the EMPOWER Inbox component. The inbox offers services to PHAs and the EMPOWER dashboard. Furthermore, using inbox services, it will be possible for PHAs to process inbox messages (filter, enrich, transform) according to the view requirements.

7.1.2 Knowledge Model Repository Subsystem

For each Knowledge Model as defined in D3.4.1, a separate Knowledge Model Handler (KMH) is configured and the appropriate adaptors and plugins provide interoperation services that support the Domain Data Access Subsystem (PHRS, EHRS) or database. For example, interoperation services would enable the PHRS or EHRS system to work together with the EMPOWER Domain Data Access Subsystem. The PHRS or EHRS would implement adaptors or plugins to handle XDS or XPHR messages from EMPOWER. The Knowledge Model Repository Subsystem with the currently considered Knowledge Models are depicted in Figure 4.

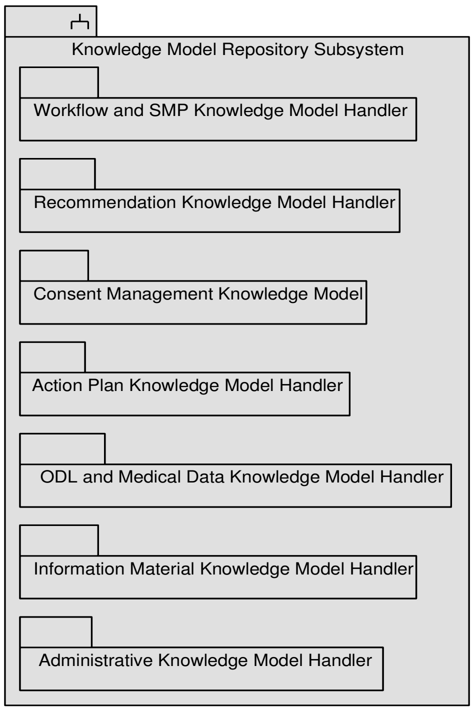

Knowledge Model Repository Subsystem
Workflow and SMP Knowledge Model Handler
Recommendation Knowledge Model Handler
Consent Management Knowledge Model
Action Plan Knowledge Model Handler
ODL and Medical Data Knowledge Model Handler
Information Material Knowledge Model Handler
Administrative Knowledge Model Handler

Figure 4: Knowledge Model Repository – Black-box

7.1.3 Domain Data Access Subsystem

The Domain Data Access Subsystem is responsible for accessing domain related information for patient and system information. This subsystem also enables semantic interoperability for exchanging of information between EMPOWER and the host PHRS or EHR (PIS) systems. Adaptors specific to target system (PHRS, EHRS, or PIS) will be in the Knowledge Model Repository Subsystem.

This subsystem also requires all handlers to utilise authorisation services to address data security and privacy of user or personal information. Other subsystems supporting data access to terminologies or diabetes information materials do not necessarily require user authorisation.

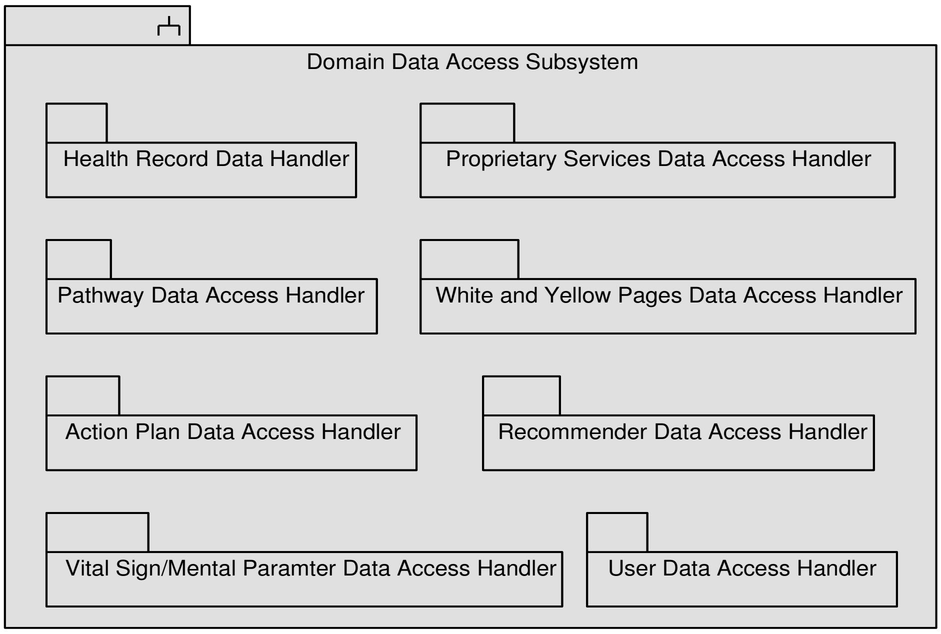

Domain Data Access Subsystem
Health Record Data Handler
Proprietary Services Data Access Handler
Pathway Data Access Handler
White and Yellow Pages Data Access Handler
Action Plan Data Access Handler
Recommender Data Access Handler
Vital Sign/Mental Paramter Data Access Handler
User Data Access Handler

Figure 5: Domain Data Access – Black-box

The Domain Data Access Subsystem provides at least the following data handlers to other subsystems:

- **Health Record Data Access Handler:** Data access for PHRS and EHRS assets
- **Action Plan Data Access Handler:** Data access for Action Plan Engine related assets
- **Pathway Data Access Handler:** Data access for Pathway engine related assets
- **Proprietary Services Data Access Handler:** Data access for proprietary data services provided by data centers
- **Recommender Data Access Handler:** Data access for recommender related assets. For example, the Action Plan Engine retrieves the recommender's recommendations for a patient and any associated patient encounter (consultation) data.
- **User Data Access Handler:** Data access for EMPOWER user related assets. Possible access of PHRS data. This includes also administrative data (like user settings, maturity levels, etc.)
- **Vital Sign/Mental Parameter Data Access Handler:** A common data access handler for all ODL related assets (e.g. Physical/Lifestyle Activity Data and Vital Sign/Mental Parameters). EMPOWER communicates with the host PHRS to exchange ODL related assets.
- **White and Yellow Pages Data Access Handler:** Data access for White and Yellow pages related assets.

7.1.4 Action Plan Engine Subsystem

The Action Plan Engine Subsystem provides Action Plan services primarily to PHAs and the Pathway Engine. The Action Plan Engine consumes many services from other subsystems including configuration, domain access handlers, security (authorisation), Audit trail, Reporting, Information materials, Terminology, Multimodal Services and Pathway subsystems.

The Pathway engine handles the process models of the Self-Management Pathway (SMP) and Action Plan Engine. These process models facilitate human interaction with both the Action Plan engine and Pathway Engine.

The Recommender Engine subsystem will provide information about patient encounters (consultation) with medical professionals including recommendations such as treatment goals to help the patient set their goals in their Action Plan. The patient encounter information will be stored by the Recommender and made accessible to the Action Plan Subsystem.

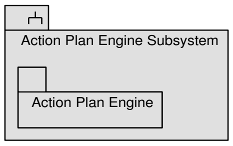


Figure 6: Action Plan Engine – Black-box

7.1.5 Recommender Engine Subsystem

The Recommender Engine is responsible for executing long-running Machine Processable Diabetes Guidelines (MP-DG) on behalf of the medical professionals. The guidelines in medical domain are developed by the best practices in treatment, monitoring or management of diseases. They are developed in textual formats by medical professionals. In order these guidelines to be executed with the help of computers, they should be described in a machine processable language and currently the mostly used language for this is Guideline Interchange Format (GLIF). To be more specific, the Recommender Engine Subsystem will execute guidelines for the management of diabetes. These guidelines are mostly long running (i.e. the actions in the guidelines may be performed in a specific duration like in usual business processes). In addition to the long-running MP-DGs, the Recommender Engine also provides direct recommendations based on the condition of the patient. Accordingly, the medical professionals can specify goals to support action planning by their diabetes patients.

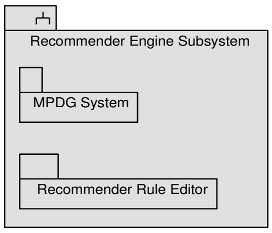


Figure 7: Recommender Engine – Black-box

As shown in Figure 7, this subsystem is composed of two modules. The MP-DG module is mainly a graphical user interface for the management (create, update, delete and execute) of the diabetes guidelines. On the other hand, the Recommender Rule Editor is for creating short running recommender rules graphically.

7.1.6 Multimodal Services and Interfaces Subsystem

The Multimodal Services and Interfaces subsystem (Figure 8) implements the differentiation of the views between clients, depending on client profile, configuration data and related META-TAGS. For this purpose, there is the Navigation Handler responsible for navigation issues between separated views and the Presentation Handler, which is responsible for presentation issues. The Multimodal Configuration subsystem is necessary in order to support a dynamic approach in the presentation layer, according to business needs.

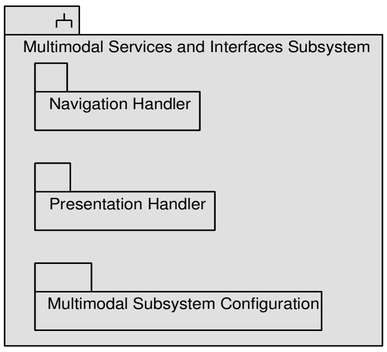

Multimodal Services and Interfaces Subsystem
Navigation Handler
Presentation Handler
Multimodal Subsystem Configuration

Figure 8: Multimodal Services and Interfaces Subsystem

7.1.7 Terminology Subsystem

The terminology subsystem provides terminology and internationalisation services and tools. Additional tools will be necessary for assembling and transforming internationalisation and terminology resources available from heterogeneous sources (i18n of Java or XML resources, PHRS resources for integration). The centralisation of this functionality is meant to harmonise and facilitate access to both terminologies, and internationalisation sources.

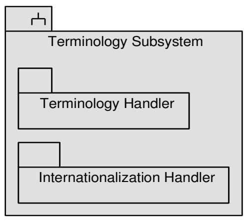


Figure 9: Terminology Subsystem – Black-box

7.1.8 Information Material Subsystem

The Information Materials Subsystem provides access to patient information material services. Patient information also includes help information about the EMPOWER to the User. Integration of 3rd party tools providing such functionality (like a Wiki or other CMS) is intended.

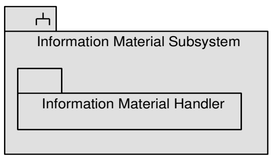


Figure 10: Information Material – Black-box

7.1.9 Reporting Subsystem

The reporting subsystem is responsible for creating visual representations (e.g. diagrams) of recorded medical data and for generating structured reports (e.g. printable reports, summaries). It offers interfaces to other components in order to provide its encapsulated functionality. The main consumer of these interfaces will be the multimodal services and interfaces subsystem and its presentation handler.

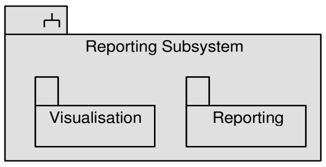


Figure 11: Reporting Subsystem – Black-box view

7.1.10 Consent Manager Subsystem

The Consent Manager Subsystem is for the management of the consents of the diabetes patients using the PHR system. It has two modules: (1) The Consent Editor is responsible for managing the consents the patient gives to Medical Professionals. With the Consent Editor the patient is able to give consent to Medical Professionals, remove Medical Professionals from the “Given Consent” list and view contact details of the Medical Professionals she has previously given consent, graphically. (2) On the other hand, the Consent Engine executes the consent rules given by the diabetes patient, when there is an access request to the patient data.

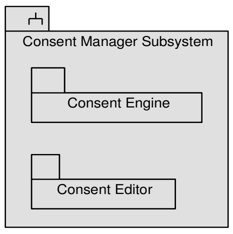


Figure 12: Consent Manager Subsystem – Black-box

7.1.11 Security Subsystem

The security subsystem offers services to the other subsystems related with identity management, user authentication, user authorisation and certificates for secure communication. The security subsystem must apply the rules that are specified by the users using the consent editor to each access to the other subsystems. Other subsystems will be able to request whether a user is allowed to access specific resources. Instead of developing a proprietary security solution from scratch, the use of existing AAA products like OpenAM[26] is preferred.

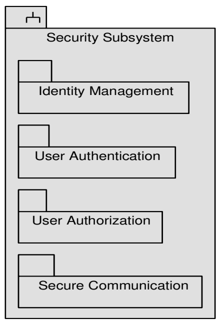


Figure 13: Security Subsystem – Black-box

[26] http://www.forgerock.com/openam.html

7.1.12 Audit Trail/Logging Subsystem

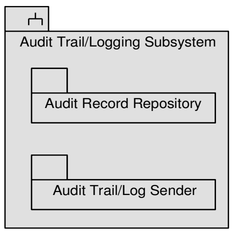


Figure 14: Audit Trail/Logging Subsystem – Black-box

The Audit Trail/Logging Subsystem is for the monitoring of audits in the EMPOWER system to enhance security. As shown in Figure 14, it contains two modules: (1) The Audit Record Repository (ARR) component collects all the logs sent from EMPOWER components and allows the System Administrator to monitor the system operation on “who sends which type of information to whom, who saw which type of information of whom, etc”. In this way, the security of the EMPOWER framework is enhanced. (2) On the other hand, through Audit Trail/Log Sender component, the EMPOWER components will be able to send audit logs to ARR through IHE ATNA profile to enhance traceability of the system execution.

7.1.13 Health Data Harvesting Subsystem

This subsystem provides services for PHRS and PHAs for harvesting and importing data for and by the patient. There are a variety of possible data sources, such as data from PIS/HIS, EHRS, PHRS, Data Centers (cloud services) for supporting patient devices, or file upload by a patient.

For the PHRS there might be various triggers for initiating the import of data, normally, this is patient-centric and handled by a particular SMP process model. The patient, however, might initiate an import from a particular system and then notified later by mail when the process has finished.

Data harvesting from data centers might also be prototyped (wearable sensors, home sensors e.g. bathroom scale, etc.); however, the data harvesting could instead be handled by the smartphone and forwarded to the EMPOWER PHRS using ODL data access services. Importantly, patient should know the data source provider from the EMPOWER PHRS and PHAs need to provide this information.

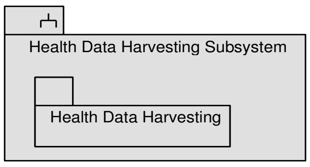


Figure 15: Health Data Harvesting Subsystem

7.1.14 PHR Integration Subsystem

This subsystem provides integration services and tools for the PHRS, and PHAs. For example, services could support identity resolution & mapping for the PHRS and EHRS, and provide service facades for other subsystems such as localisation, terminologies, internationalisation, and patient information materials. Error handling between the EMPOWER components and the PHRS will also be addressed in this subsystem.

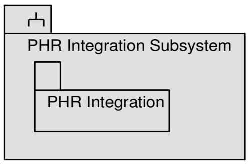


Figure 16: PHR Integration Subsystem

7.2 Level 2 – White Box Descriptions

This section contains the available “inside view” into the subsystems and packages that have been presented in the previous section. Note that these representations also contain dependencies to other packages and subsystems.

7.2.1 Service Locator

The service locator (SL) is the main entry point to the EMPOWER system. It provides the lookup of the EMPOWER services in the Service Registry and returns the requested service to the calling component. Together with the Configuration and the Service Descriptor it builds the *EMPOWER Core*.

The service locator uses two different design patterns that can be used to access the implemented services:

- Option A (*Proxy*[27]): In this case the SL returns an instance that exactly mimes the behaviour of the requested service.
- Option B: (*Façade*[28]): In this case the SL returns an instance with a more convenient behaviour with lower degree of complexity, e.g. by providing additional convenient methods within the Façade.

This layer of abstraction helps to enable separated development and working with mock-up components during the development phase.
Figure 17 shows an example for using the EMPOWER Service Locator. The Service Locator Proxy will do the authorisation check on the service level, while the service-specific authorisation checks are done by the services itself (either on the service proxy or the service façade). A runtime view on the use of the SL is provided in Section 8.1.

[27] Proxy design Pattern - http://en.wikipedia.org/wiki/Proxy_pattern
[28] Façade design Pattern- http://en.wikipedia.org/wiki/Facade_pattern

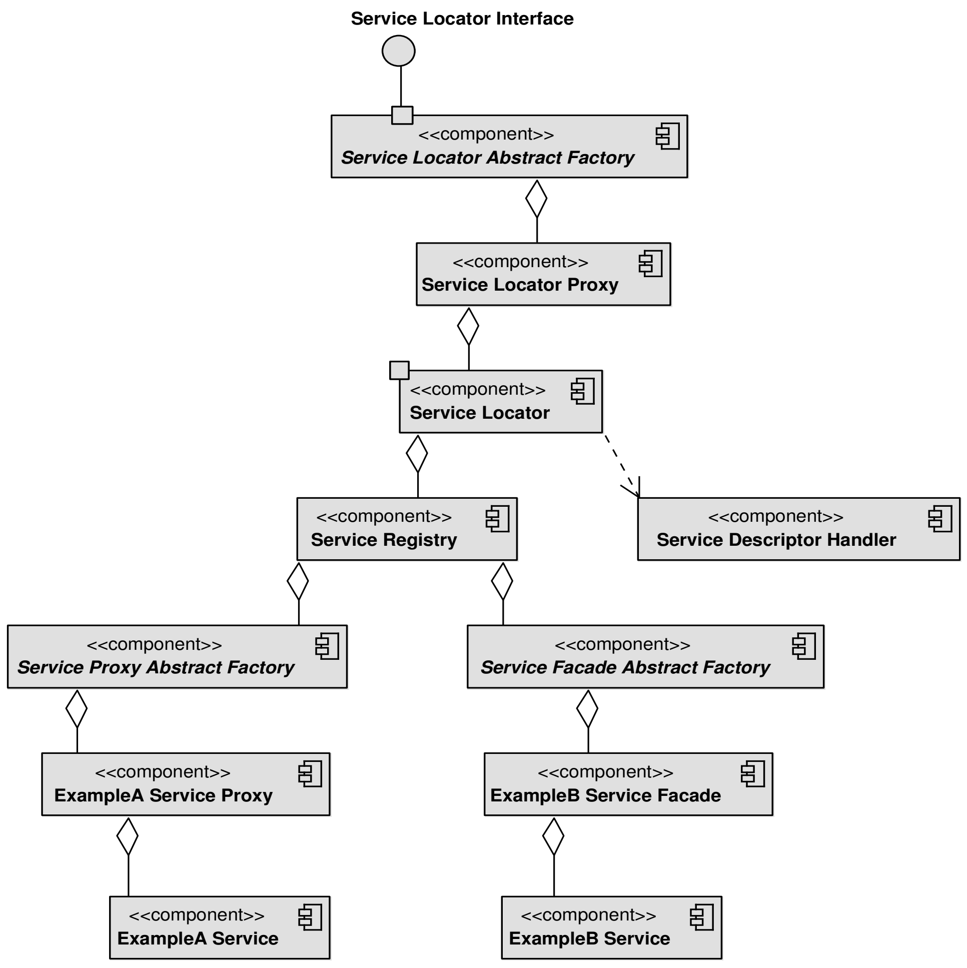


Figure 17: Service Locator with Example Services – White-box

7.2.2 Configuration

The configuration service is responsible to retrieve and distribute configuration information from other components and is therefore the central place to store EMPOWER configuration. Configuration should only contain non-sensitive information as the configuration service does not apply any access control constraints. The user is able to store in the configuration all kind of information. Unfortunately, not all kind of information is shareable and this is the reason why the EMPOWER system can apply security aspects on the configuration, if this is required.

As depicted in Figure 18, the configuration package also includes a configuration listener, in order to also enable configuration changes in runtime. Still, the major part of the configuration is meant to be static. Furthermore the configuration provides different access interfaces, including a RESTful web-service interface for storing and retrieving configuration information.

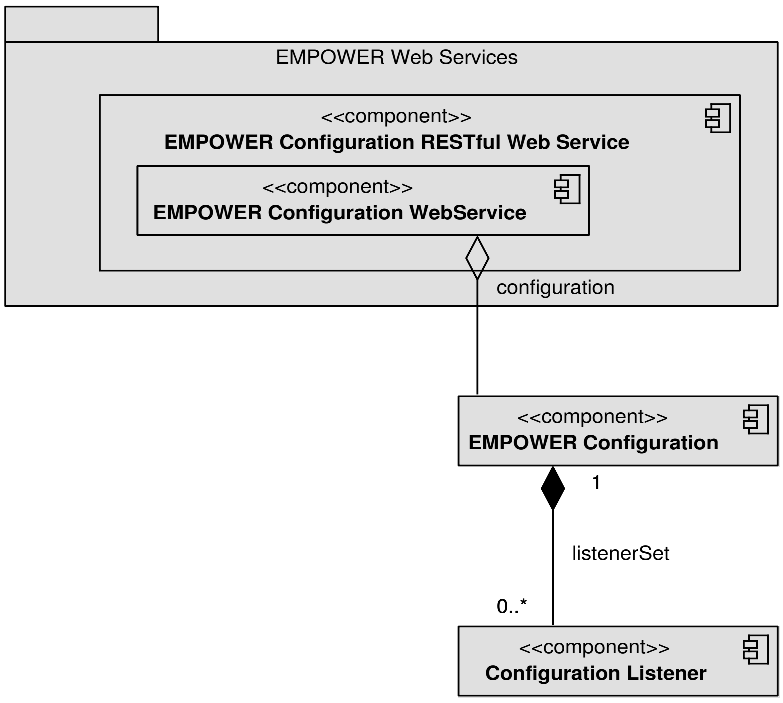


Figure 18: Configuration Service – White-box

7.2.3 Pathway Engine

The pathway engine needs to provide the following four services:

- PathwayMonitorCoreService
- PathwayEnginePHRSService
- PathwayEngineCoreService
- PathwayEnginePHAService

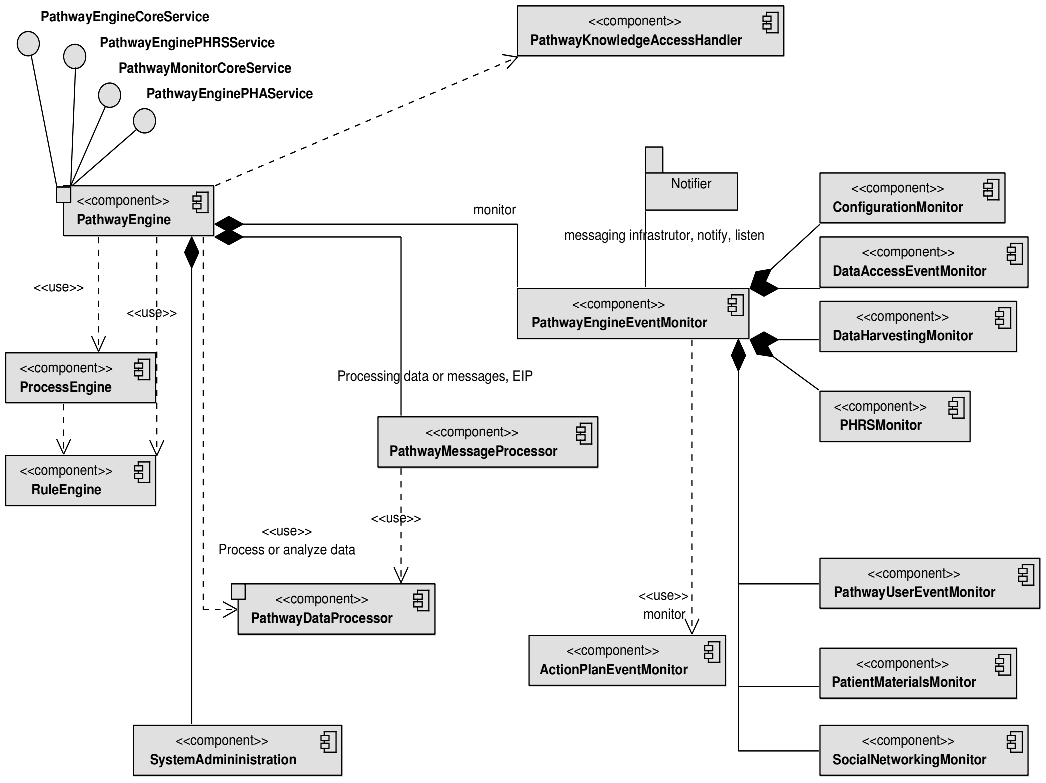

PathwayEngineCoreService
PathwayEnginePHRSService
PathwayMonitorCoreService
PathwayEnginePHAService
<<component>> PathwayKnowledgeAccessHandler
<<component>> PathwayEngine
monitor
Notifier
messaging infrastrutor, notify, listen
<<component>> ConfigurationMonitor
<<component>> DataAccessEventMonitor
<<component>> DataHarvestingMonitor
<<component>> PHRSMonitor
<<component>> PathwayEngineEventMonitor
<<use>>
<<use>>
<<component>> ProcessEngine
<<component>> RuleEngine
Processing data or messages, EIP
<<component>> PathwayMessageProcessor
<<use>>
<<use>>
Process or analyze data
<<component>> PathwayDataProcessor
<<use>>
monitor
<<component>> ActionPlanEventMonitor
<<component>> PathwayUserEventMonitor
<<component>> PatientMaterialsMonitor
<<component>> SocialNetworkingMonitor
<<component>> SystemAdmininistration

Figure 19: Pathway Engine – White-box

As depicted in Figure 19, the Pathway engine consists of several interdependent components, which are described as follows:

- **ProcessEngine:** An existing Open Source rule-based process engine
- **RuleEngine:** The rule engine required by the process engine, although the rule engine may be utilized by any component.
- **PathwayMessageProcessor:** A processor based on EIP patterns to enrich, filter, transform, route messages.
- **SystemAdmininistration:** Administrative services for the system.
- **PathwayDataProcessor:** A component (EIP patterns) that processes data for use by Pathway or other EMPOWER components. Heterogeneous sources or proprietary messages can be enriched, filtered, or transformed to a standard message (IHE profile XPHR), an EMPOWER domain specific message, or message required by 3rd party tools such as the process engine.
- **PHRSIntegrationHandler:** Provides integration services to the PHRS. The PHRSMonitor is accessible from this handler. Integration might include the handling of user sessions if necessary.
- **PathwayKnowledgeAccessHandler:** Handles the integration of dynamic and static knowledge relevant for the process engine, rule engine and terminology services.
- **ConfigurationMonitor:** Monitors Configuration changes.
- **PathwayUserEventMonitor:** Provides monitoring of user events to components, including all monitors and the process engine. This monitor can serve as a central monitor for user events.
- **DataAccessEventMonitor:** Monitors data access, especially by the process engine depending on the particular Business Process Model. For example, if the patient must provide one glucose measurement per day, then when an the glucose measure ODL is saved, the monitor notifies the process engine and the executing BPM task marked as completed on this day.
- **SocialNetworkingMonitor:** Monitors social networking related information. For example, a new forum is available about a particular topic and that topic is relevant to a goal or interest of patients. Daily RSS feeds likewise could be monitored.
- **PatientMaterialsMonitor:** Monitors patient information repository. Report updates by topic and quantitative information by topic.
- **DataHarvestingMonitor:** Monitor access.
- **PHRSMonitor:** Bidirectional monitoring of PHRS and Pathway engine subsystem.
- **PHAMonitor:** Bidirectional monitoring of PHA and Pathway engine subsystem.

7.2.4 Notifier

The Notifier is based on EIP software patterns. The MonitorManager will be the superclass for all the monitors of the EMPOWER system. This manager provides convenient methods to register and publish various (synchronous and asynchronous) messages. The nature of the message deployment may vary from peer-to-peer to publish subscribe messages. Security constrains can be applied over any message publish and/or subscribe operation, including the creation of audit records for the Audit Trail/Log System. The dependencies between the components are depicted in Figure 20.

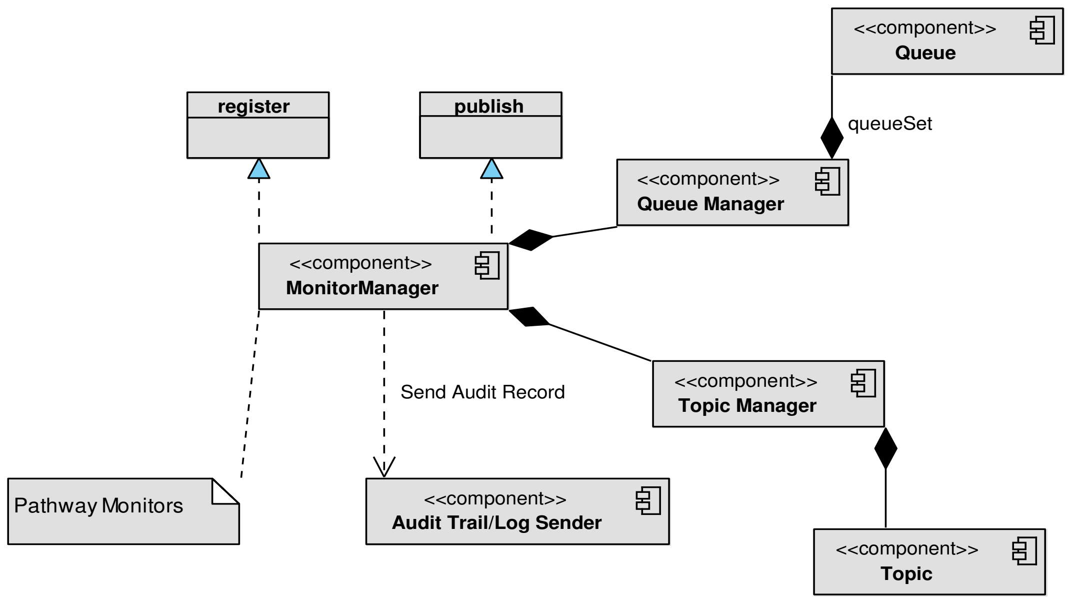

<<component>> Queue
queueSet
register
publish
<<component>> Queue Manager
<<component>> MonitorManager
Send Audit Record
<<component>> Topic Manager
Pathway Monitors
<<component>> Audit Trail/Log Sender
<<component>> Topic

Figure 20: Notifier – White-box

7.2.5 PHR Integration

The PHRS integration subsystem provides service facades to simplify the use of EMPOWER services by PHRS and PHAs. They support the communication and integration between the PHRS, PHAs and EMPOWER components. Particular services will address security and identity mapping issues.

Based on the facade software pattern, they simplify and reduce dependencies of PHRS and on the internal workings of the EMPOWER API. Paticular services will support identity resolution & mapping for the PHRS and EHRS, and provide service facades for other subsystems such as localisation, terminologies, internationalisation, and patient information materials. Error handling between the EMPOWER components and the PHRS will also be addressed in this subsystem.

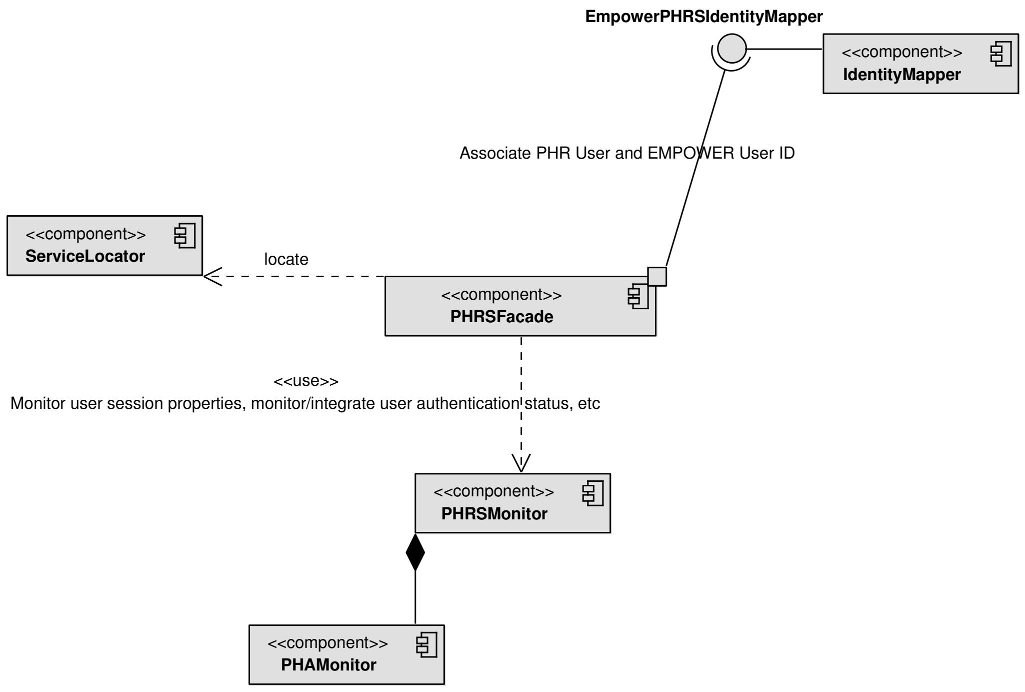

EmpowerPHRSIdentityMapper
<<component>> IdentityMapper
Associate PHR User and EMPOWER User ID
<<component>> ServiceLocator
locate
<<component>> PHRSFacade
<<use>>
Monitor user session properties, monitor/integrate user authentication status, etc
<<component>> PHRSMonitor
<<component>> PHAMonitor

Figure 21: Health Application Integration – White-box

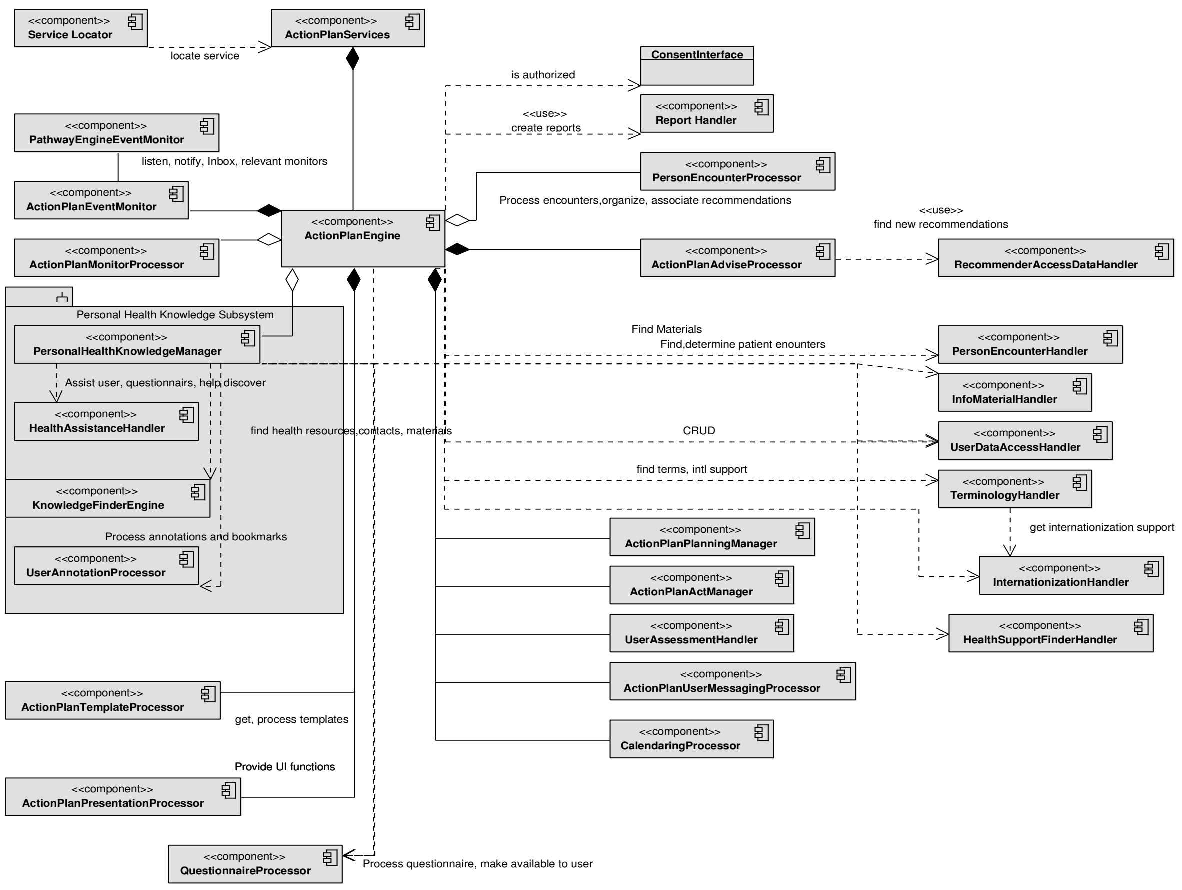
The PHRSFacade uses many EMPOWER services and in Figure 21, only the two integration related components are shown, the PHRSMonitor and Identity mapper. The PHRSMonitor enables the PHRS and Pathway Engine to communicate directly. The Identity mapper enables the PHRS to determine identity mappings of PHRS users to other systems.

7.2.6 Action Plan Engine (APE)

The action plan engine (APE) is responsible to execute the Action Plans (AP) of the Patients. It is comprised of the components listed below. Their interdependencies are shown in Figure 22.

- **AP Planning Manager**: This component supports the user to manage and plan their goals and associate potential activities and strategies to achieve either the activity or goal. To plan, the user needs to be informed about how to create goals and resources from which to derive goals. These supportive resources include recommendations and health data from each medical consultation (encounters) that are stored by the recommender engine after each consultation. Other supportive resources include patient information materials (decision aids, texts, summary charts supporting goals e.g. Mind maps) to help support their decision making for planning their goals, activities and strategies to achieve their activity and goals.
- **AP Advise Processor:** This component supports components to assemble, enrich deliver supportive materials to the patient. The type of information depends on the component using this processor. Supportive materials include information materials, charts, and questionnaires. As a processor component it is will be based on EIP patterns such as message routing, enrichment, filtering, transformation, etc.
- **AP Act Manager:** This component supports Activity event planning. Events are configured and associated or not with an activity. For each activity, the user might also activate, deactivate, prioritize, or apply other activity setting independent of the event planning. Naming the component with "Event" or "Activity" was deemed ambiguous.
- **AP Engine:** The Action Plan Engine is the main entry point offering Action Plan services to PHAs and other core components. It interacts with the Pathway Engine and other components via the PathwayEngineEventMonitor.
- **AP Event Monitor:** This monitor facilitates communication between an active action plan activity events and other components including AP Engine related components, PHAs and the PathwayEngineMonitor. A "monitor" design is based on appropriate EIP patterns / Message patterns.
- **AP Monitor Processor:** This component is the communication point between the AP Engine components and Pathway Engine & process engine, particularly the BPMs supporting the Action Plan Engine and SMP.
- **AP Presentation Processor:** This component processes presentation information. In general, it processes and prepares data, for example for use with a template. It processes and prepares data for the EMPOWER visualisation and reporting components to support the Action Plan PHA services.
- **AP Services:** This component offers services to PHAs and potentially to EMPOWER core components. Interfaces will be categorized according to the role within the User interface (Action Plan, Inbox, Dashboard Menu, etc.). Services will normally be RESTful. Usually the core components will communicate via a Monitor component.
- **AP Template Processor:** The template processor acquires the template.
- **AP User Messaging Processor:** The user messaging processor enables users to communicate via the inbox and provides inbox services to the PHA dashboard inbox menu. Additionally the Action Plan UI can filter and present messages outside of the inbox to the user.
- **Calendaring Processor:** This processor works with the activity event data and processes it for a calendar depending on the destination calendar. There is no calendar presentation handler, rather the calendar application provided by the device or PHA plugin (e.g. JQuery agenda plugin)

- **Pathway Engine Event Monitor:** The PathwayEngineEventMonitor supports communications between the PathwayEngine, the APEngine, and other components. It acts as a message bus and components can register their event listeners.
- **Person Encounter Handler:** This handler accesses heterogeneous data sources containing data about a patient's encounters with medical professionals, self-help groups, etc. The obvious sources of encounters are from the activity event plan and the stored recommendations from the EMPOWER recommender. In the processor, the patient might have scheduled an appointment that can be matched with encounter data from other sources such as the recommendations. The handler updates and maintains patient encounter summary information provided by the PersonEncounterProcessor.
- **Person Encounter Processor:** As a processor, this component processes (enrich, filter, transform) information about a patient encounter(s) and creates a common message for use by other Action Plan components. EIP patterns are employed to process messages. The encounter processor can then prepare updates to a patient encounter summary with cross referenced information, for example, linking doctors' appointments to consultation information, EHR patient data and recommendations.
- **Questionnaire Processor:** This component processes (routes/assembles, enrich, transform) questionnaires or responses from questionnaires. The questionnaire is an input form, a type of message that is directed to a target user and the results used by the sender and shared with the target user. It differs from a UI web form, although they could be accessed from the web UI or from an inbox message. The questionnaires are a means to either facilitate data collection or even disseminate information to the patient. The sender is usually the system acting on behalf of the user. Questionnaires will usually be associated with a BPM. For example, to support activity events such as preparing the patient for the consultation event (a doctor's appointment includes questionnaire for the patient), the settings of user settings and maturity levels; the dissemination of knowledge, information materials and decision aids and lastly one aspect of addressing motivation.
- **User Assessment Handler:** This component supports the PHA services and the continuous or periodic assessment of the action plan based on Action Plan and SMP Business process models via the APMonitorProcessor. This component processes the BPMs especially to support PHA user interactions, reporting and visualisation. Continuous assessment of the plan occurs while the plan is active and could provide the user with messages to the inbox and/or User interface feature. Depending on the BPM, each weekly assessment, the user is notified to view their results and other supportive. Subsequently, the patient provides feedback to modify their Action Plan accordingly.



<<component>> Service Locator
locate service
<<component>> ActionPlanServices
ConsentInterface
is authorized
<<use>> create reports
<<component>> Report Handler
<<component>> PathwayEngineEventMonitor
listen, notify, Inbox, relevant monitors
<<component>> PersonEncounterProcessor
<<component>> ActionPlanEventMonitor
<<component>> ActionPlanEngine
Process encounters,organize, associate recommendations
<<use>> find new recommendations
<<component>> ActionPlanMonitorProcessor
<<component>> ActionPlanAdviseProcessor
<<component>> RecommenderAccessDataHandler
Personal Health Knowledge Subsystem
<<component>> PersonalHealthKnowledgeManager
Find Materials
Find,determine patient enounters
<<component>> PersonEncounterHandler
Assist user, questionnairs, help discover
<<component>> InfoMaterialHandler
<<component>> HealthAssistanceHandler
find health resources,contacts, materials
CRUD
<<component>> UserDataAccessHandler
find terms, intl support
<<component>> KnowledgeFinderEngine
<<component>> TerminologyHandler
<<component>> ActionPlanPlanningManager
get internationization support
Process annotations and bookmarks
<<component>> UserAnnotationProcessor
<<component>> ActionPlanActManager
<<component>> InternationizationHandler
<<component>> UserAssessmentHandler
<<component>> HealthSupportFinderHandler
<<component>> ActionPlanUserMessagingProcessor
<<component>> ActionPlanTemplateProcessor
get, process templates
<<component>> CalendaringProcessor
Provide UI functions
<<component>> ActionPlanPresentationProcessor
<<component>> QuestionnaireProcessor
Process questionnaire, make available to user

Figure 22: Action Plan Engine (APE) – White-box

The following components have been grouped together into the “Personal Health Knowledge Subsystem”. This subsystem support the patient to aggregate information and discover information, including contact information and patient information materials, decision aids, questionnaires and questionnaires as decision aids, recommendations:

- **Personal Health Knowledge Manager:** Manages the knowledge that the patient aggregates: Contact info (medical, online & offline support groups), info material bookmarks and annotations.
- **User Messages Handler and User Messages Processor:** The processor routes the correct enricher or transformer. Enrichment includes parameterisation of the message template and Internationalisation of the message. It is noted, that EMPOWER should listen to changes to the PHRS for user language changes. This component is available to all EMPOWER components.
- **Health Assistance Handler:** This component handles messages directed to the user and applies a processor (UserMessagesProcessor). This component is available to all EMPOWER components.

7.2.7 Visualisation Module

The components in the visualisation module are divided as follows. The Visualisation Handler is the interface to outside components. This includes the following tasks:

- provide rendered chart or other graphical representation of relevant data to other components
- provide info about available chart templates and its configuration
- provide a GUI call to manage templates
- utilizes EMPOWER security and privacy components
- ATNA notification interface

The Visualisation Processor performs data analysis, enrichment, transformation, etc.

The Visualisation Renderer does the whole rendering, e.g. producing an ODL chart over a given time period. Finally, the Visualisation Template Manager is providing the view and controller for managing chart templates, including the following:

- create/import new templates
- modify existing templates (in respect to visual representation, permissions, assigned user groups etc.)
- delete or restore old version
- provide chart templates for rendering
- persist templates in Visualisation Template Repository

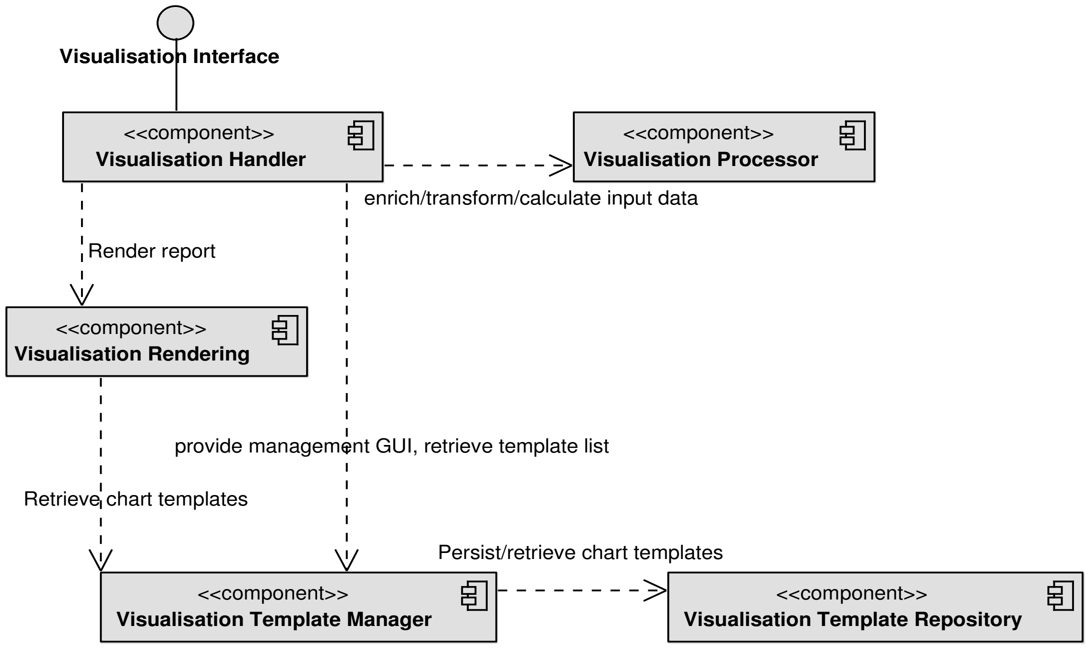

Visualisation Interface
<<component>> Visualisation Handler
<<component>> Visualisation Processor
enrich/transform/calculate input data
Render report
<<component>> Visualisation Rendering
provide management GUI, retrieve template list
Retrieve chart templates
Persist/retrieve chart templates
<<component>> Visualisation Template Manager
<<component>> Visualisation Template Repository

Figure 23: Visualisation – White-box

7.2.8 Reporting Module

The reporting module has a similar structure to the visualisation module, just without the separate processor. While the visualisations generated by the Visualisation Module include mainly the charts, reports can include such visualisations, but will also contain written text (for example medical summary) and additional (meta-)information required by the patient or medical professional (for example patient's or doctor's contact details).

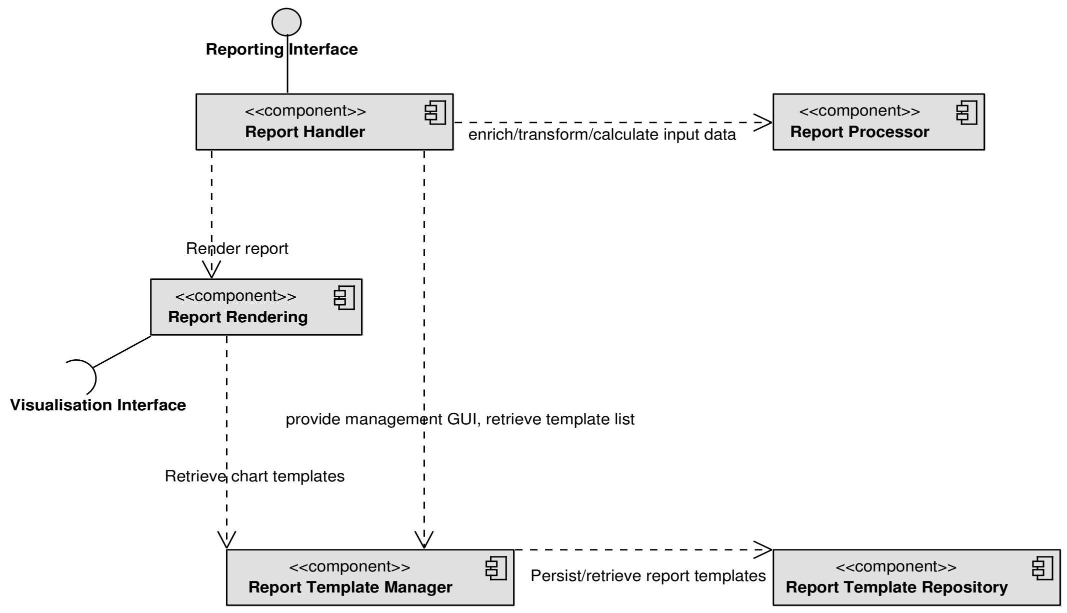

Reporting Interface
<<component>> Report Handler
<<component>> Report Processor
enrich/transform/calculate input data
Render report
<<component>> Report Rendering
Visualisation Interface
provide management GUI, retrieve template list
Retrieve chart templates
<<component>> Report Template Manager
Persist/retrieve report templates
<<component>> Report Template Repository

Figure 24: Reporting – White-box

7.2.9 Health Data Harvesting

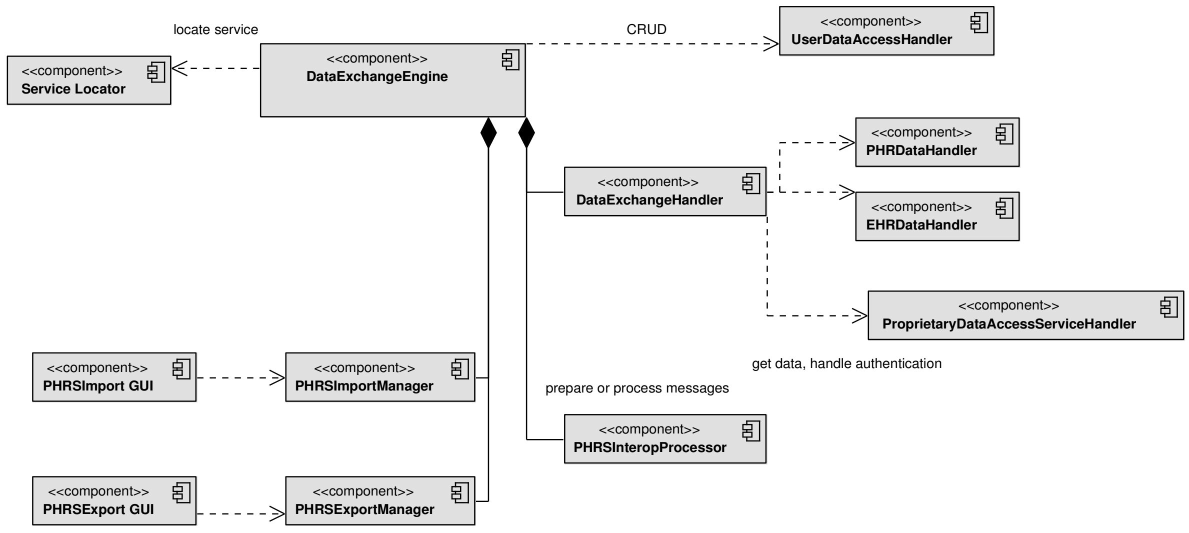

locate service
CRUD
<<component>> Service Locator
<<component>> DataExchangeEngine
<<component>> UserDataAccessHandler
<<component>> PHRDataHandler
<<component>> DataExchangeHandler
<<component>> EHRDataHandler
<<component>> ProprietaryDataAccessServiceHandler
get data, handle authentication
prepare or process messages
<<component>> PHRSImport GUI
<<component>> PHRSImportManager
<<component>> PHRSInteropProcessor
<<component>> PHRSExport GUI
<<component>> PHRSExportManager

Figure 25: Health Data Harvesting – White-box

The single components for health data harvesting provide the following functionality:

- **PHRSExport GUI Services:** Export Services are provided to the GUI.
- **PHRSExportManager:** Export manager provides services to download information in a particular format.
- **PHRSImport GUI Services:** Export Services are provided to the GUI
- **PHRSImportManager:** The import manager provides services to import data from a given data service
- **DataExchangeHandler:** Manages the data exchange between different data sources, using the data handlers from the Domain Data Access subsystems.
- **PHRSInteropProcessor:** Interoperability services are employed to transform data for the EMPOWER health record subsystem (PHRS)
- **DataExchangeEngine:** The data exchange engine coordinates the requests from the end-user, interaction with data services and interaction with EMPOWER health record subsystem (PHRS)

7.2.10 Navigation and Presentation Handler

The Navigation Handler includes services provided to clients to construct user interfaces. For example, responses to the following requests: which are the available services each time, related to an identification ticket and the client position; what are the next possible action(s), depending on the result of a specific service.

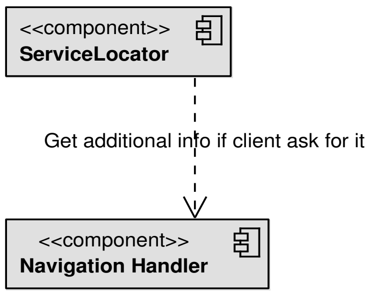


Figure 26: Navigation Handler

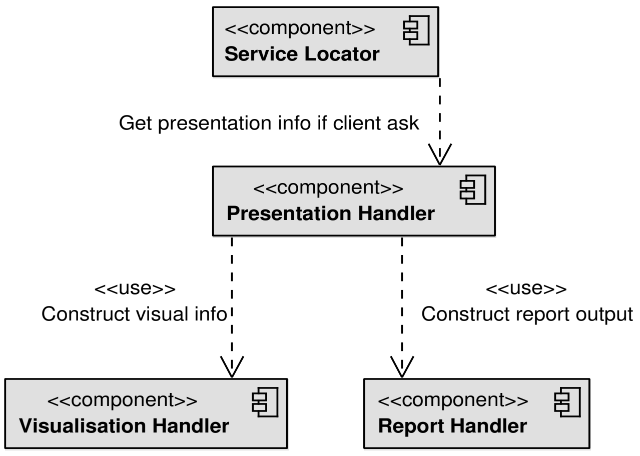


Figure 27: Presentation Handler

With reference to Figure 27, the Presentation Handler communicates with the Visualisation Handler in order to construct all the necessary visual components which will be delivered to the client. The collection of the appropriate components is based on META-TAGS and is the outcome of applying a set of predefined and configurable conditions, related to the client's profile and preferences. Moreover, the Presentation Handler communicates with the Reporting Handler in order to construct the necessary report which will be delivered to the client. The appropriate report type is also the outcome of applying a set of predefined and configurable conditions, related to the client's profile and preferences.
The interaction is also described in the runtime views in Section 8.5.

7.2.11 Recommender Engine

The internal structure of Recommender Engine is shown in Figure 28. The GuidelineEngine is the main component that executes the guidelines and orchestrates the other components. Through the PHRInterface, the GuidelineEngine retrieves patients' medical data used for the execution of the guideline from the PHR. At each important step, the engine provides necessary audit logs to the Audit Record Repository through SendAuditRecordInterface. During the execution, the GuidelineEngine handles persistency requirements (such as the status of an executing guideline, the guidelines currently executed, etc.) through the RecommenderAccessDataHandler component.

The front-end GUI of Recommender Engine subsystem is MP-DG System. This GUI has two editors (1) GuidelineEditor, through which the users create/manage diabetes guidelines graphically. (2) Recommender Rule Editor, through which the users create/manage short-running recommendation rules. Both of these editors use the RecommenderEngineModel through ModelInterface. The MP-DG system reaches the GuidelineEngine through the EngineInterface. Furthermore, the GuidelineExecutionManagement component is used by the user to manage and assign guidelines to the patients and control the execution of the diabetes guidelines.

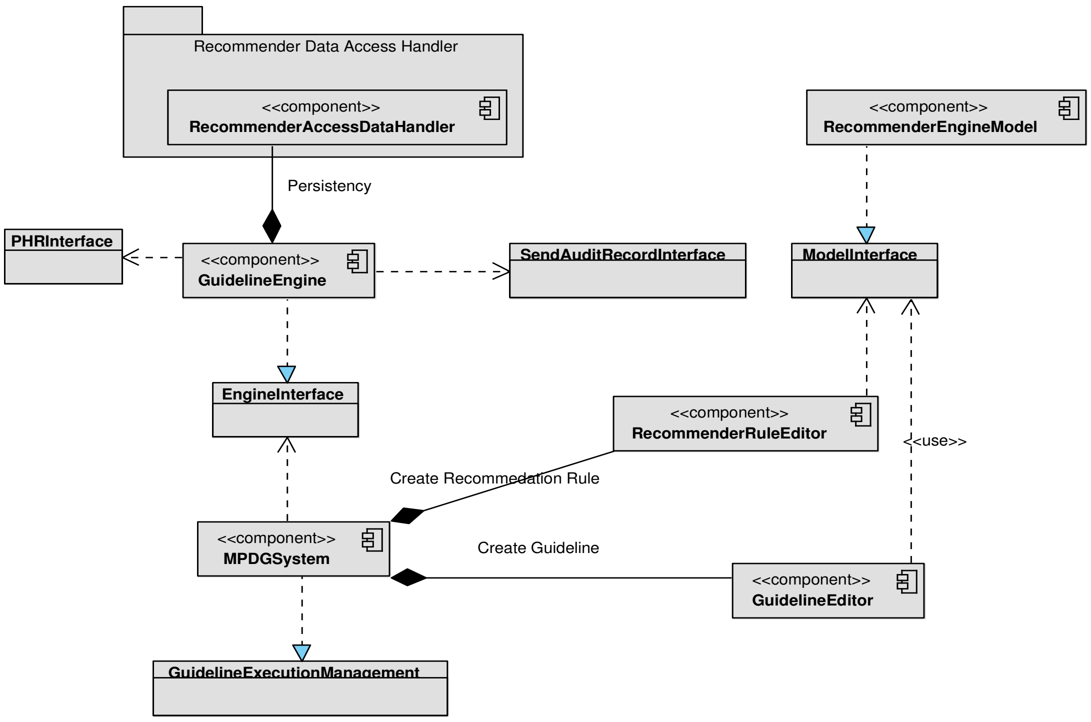


Figure 28: Recommender Engine – White-box

7.2.12 Terminology and Internationalisation Handler

The major components of the Terminology subsystem are the Terminology and Internationalisation Handlers, as well as the Terminology Processor. Interaction between them and the EMPOWER core services are depicted in Figure 29. The tasks of the single components are described in more detail below.

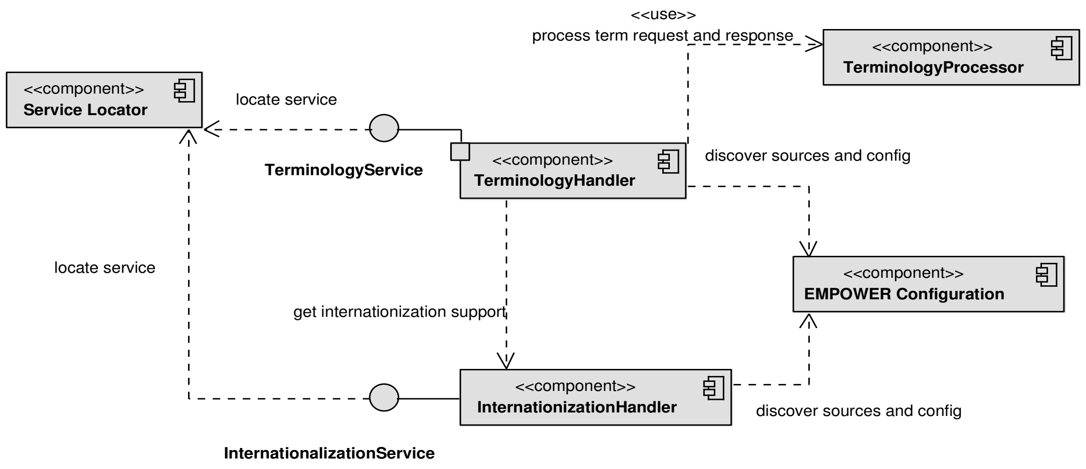

<<use>>
process term request and response
<<component>> TerminologyProcessor
<<component>> Service Locator
locate service
TerminologyService
<<component>> TerminologyHandler
discover sources and config
<<component>> EMPOWER Configuration
locate service
get internationization support
<<component>> InternationalizationHandler
discover sources and config
InternationalizationService

Figure 29: Terminology and Internationalisation Handler - White-box

- **InternationalizationHandler:** This component handles the configuration of services supporting internationalisation of terminologies, UI internationalisation properties, User messages (validation, tooltips, hints, process messages from SMP and Action Plan, etc.). The heterogeneous data sources supporting internationalisation are prepared and processed if necessary, and cached for use by the internationalisation services.
- **TerminologyHandler:** The TerminologyHandler is required to lookup terms by codes in different code systems (or namespaces) from heterogeneous sources. While simple terminologies can be defined in flat files (e.g. in configuration), more complex terminologies require more structure. The service offered is the TerminologyService.
- **TerminologyProcessor:** This component processes the terminology result(s) to enrich, transform, the information. It can also utilize particular templates and apply internationalisation using the internationalisation services
- **TerminologyAccessHandler:** This component handles the access to the terminologies.
- **TerminologyService:** These services are available to PHAs and other components.
- **InternationalizationService**: These services make the functionality of the Internationalisation Handler available to PHAs and other components.

7.2.13 Information Material Handler

The Information Material Handler provides services to discover and display relevant information resources to the patient. The required components and interfaces of the Information Material Handler are depicted in Figure 30. Strong use of existing tools like Wikis or CMSs is intended in this subsystem.

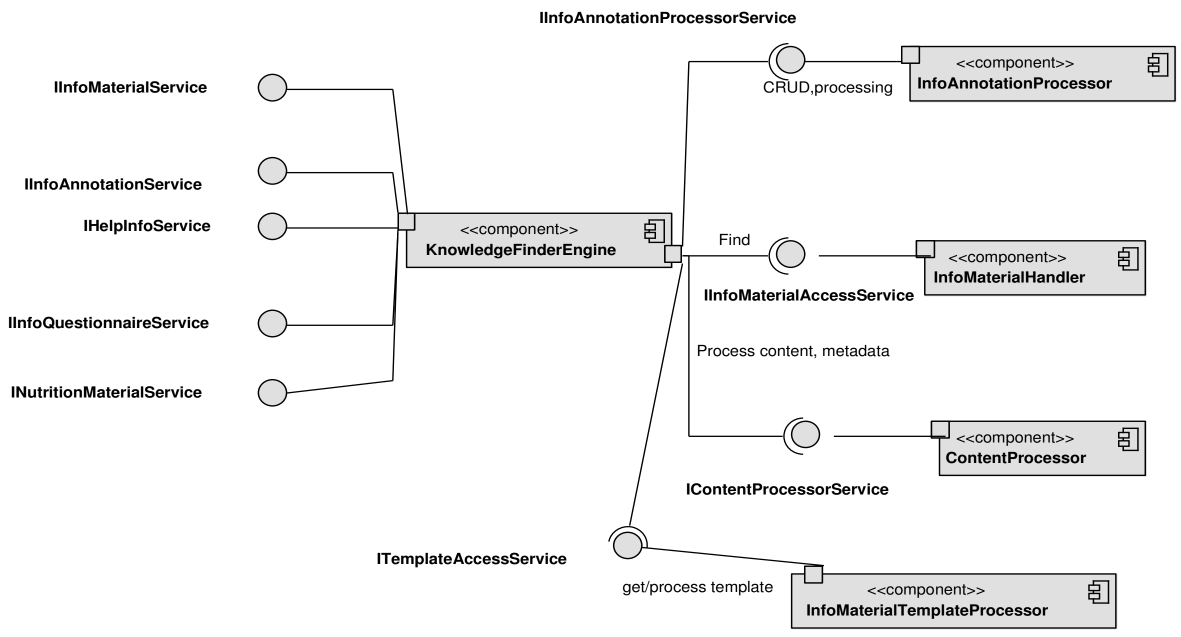


Figure 30: Information Material Handler – White-box

The depicted components are tentative depending on the APIs of the chosen 3rd party tools. The components' functions are as follows:

- **KnowledgeFinderEngine:** This engine is the main access point and supports the discovery of health resources and provides services supporting the ranking of information materials by users.
- **InfoMaterialHandler:** Provides services for patient information materials including and services for ranking materials.
- **InfoMaterialTemplateProcessor:** Handles templates for search results and information summaries.
- **InfoAnnotationProcessor:** The processor assembles and processes (e.g. enrich, filter, transform) annotations and metadata of the information materials. This is important for creating interactive summaries of information, potentially the semantically enrich or annotated content.
- **ContentProcessor:** The processor assembles and processes (e.g. enrich, filter, transform) the content of information materials. The content can be semantically enhanced or enriched using the InfoAnnotationProcessor.

7.2.14 Generic Services API

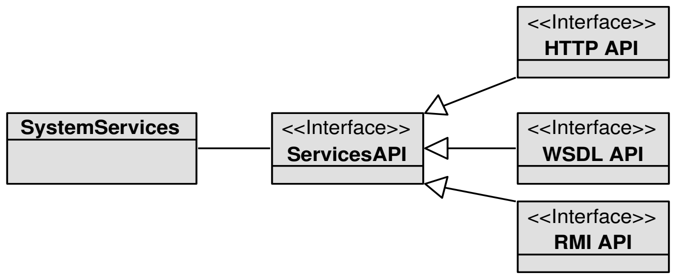


Figure 31: Services API – White-box view

The Services API Component is a component allowing clients to have access to system services. This component can be easy used, offering a satisfactory set of protocols and interfaces in order to fit with the most common client side implementations. There is for example the HTTP API, which consists of a set of service interfaces implemented behind of an HTTP listener waiting for HTTP requests. All service APIs can behave using the HL7 Protocol v3. The name “SystemServices” is meant as a generic abstraction of many different kinds of services to be offered through the EMPOWER architecture.

7.2.15 Consent Editor and Engine

The main component of this subsystem is the Consent Engine. Through the Consent Editor GUI, the user manages his/her consent rules. These rules are stored to PHR database through the PHR Interface to be used later. The Consent Editor is reached through the Consent Interface. In other words, the other EMPOWER components will use the Consent Interface, to check whether an access to the patient data should be granted or not.

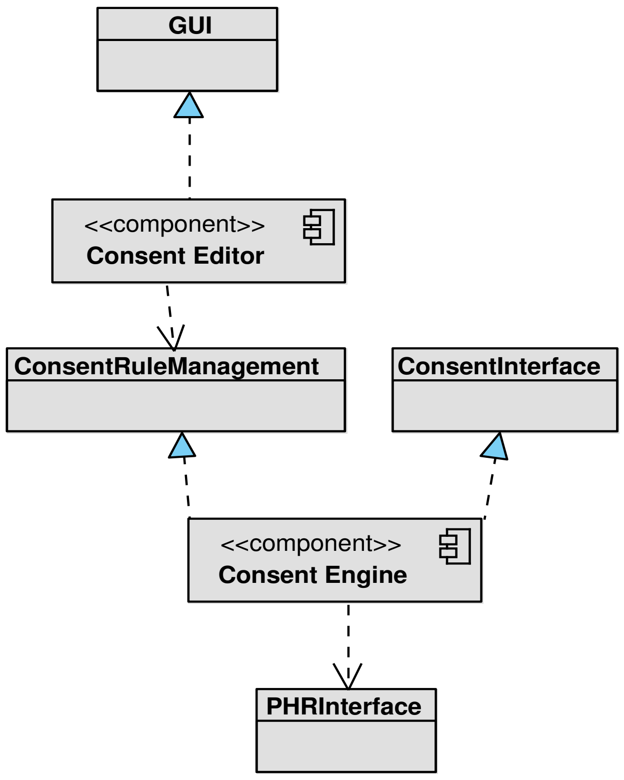


Figure 32: Consent Manager – White-box view

7.2.16 Audit Trail/Logging and Audit Record Repository

As shown in Figure 33, the Audit Record Repository component has a GUI through which all the actions in the EMPOWER components can be audited. The EMPOWER components logs every data exchange or other security related activities by sending necessary log with the Audit Trail/Log Sender component. This component is reached through the SendAuditRecordInterface component.

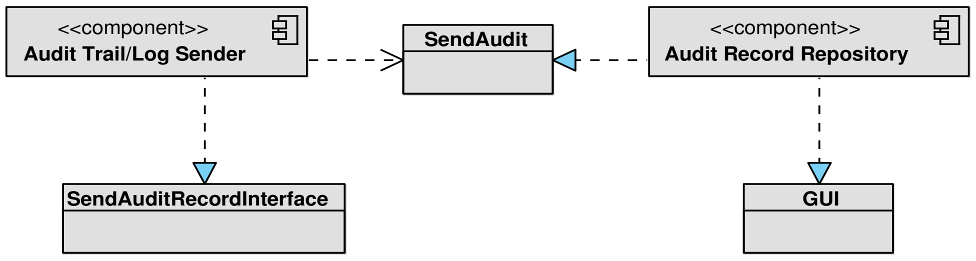


Figure 33: ATNA – White-box

7.2.17 User Authorisation

The user authorisation is required to enforce access control on the available resources. In EMPOWER a consent editor is provided to allow the users to manage the access control policies for their own resources. Therefore, the final access control decisions need to take into account user consent, which will be handled by the security policy manager. The dependencies between user authentication, authorisation, access control and security policies are shown in Figure 34. As depicted, the user needs to be authenticated before the user authorisation takes place. To grant user access to a specific resource, a corresponding security policy must exist. The security policies itself are created, deleted or configured by the consent engine.

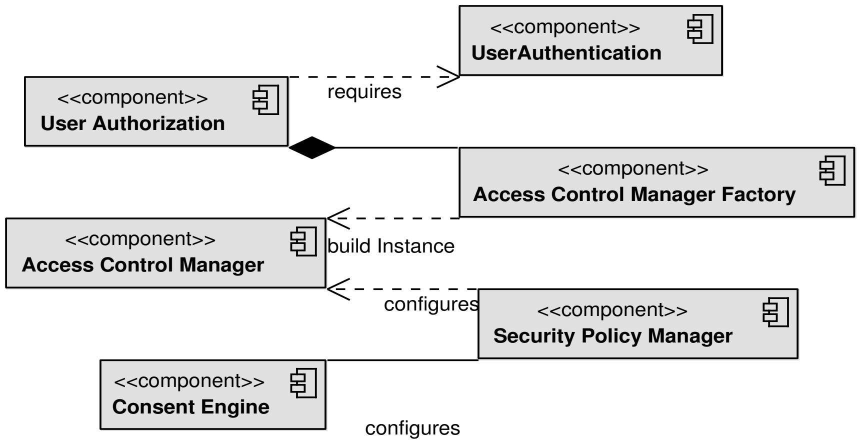


Figure 34: User Authorisation – White-box view

7.2.18 User Authentication

The user authentication is required before the user can be authorised to access a specific resource. An identity manager is responsible to request User-IDs from an (external) identity provider, and map this ID by using an identity mapper to the EMPOWER-managed User-ID. Similar, if the PHA uses its own user management with only locally valid User-IDs, they must be mapped to the EMPOWER-managed User-ID. Therefore, an interface towards the PHAs providing such functionality is offered.

Although some services might not need any user authentication, always a default user and role, which is mapped e.g. to “guest” must be used.

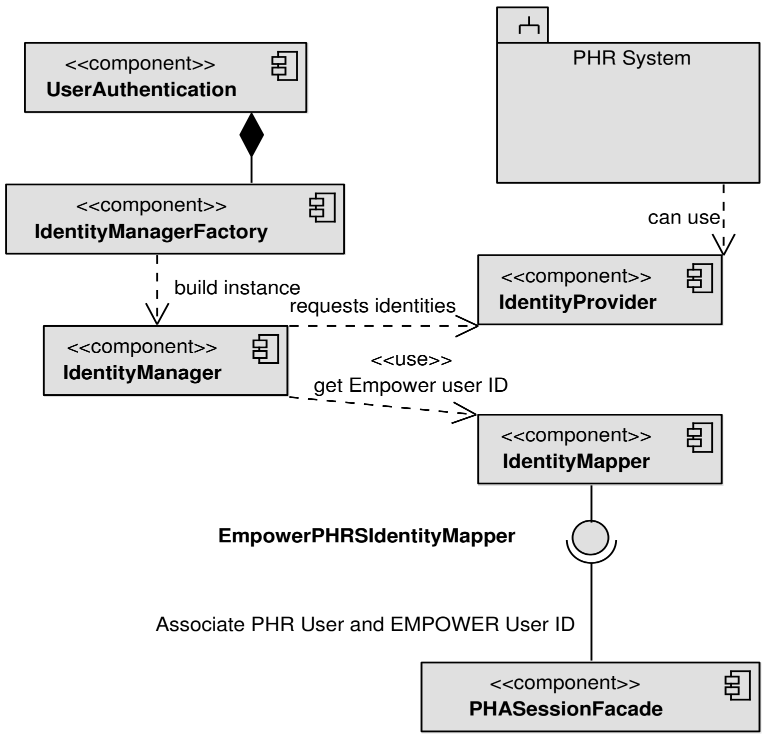


Figure 35: User Authentication – White-box view

7.2.19 Domain Data Access Handlers

7.2.19.1 User Data Access Handler

The UserDataAccessHandler provides secure access to administrative data including maturity levels, a user health profile (e.g. diabetes profile or "diabetes passport"), and ODLs.

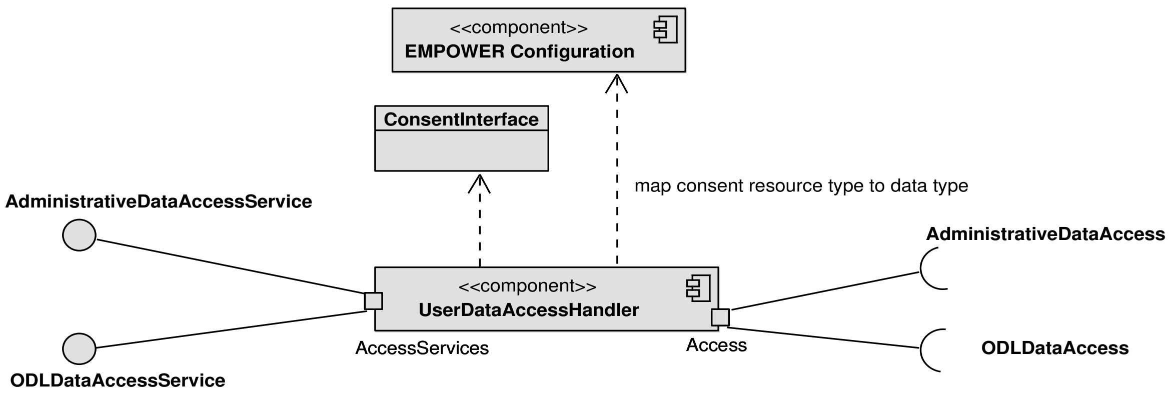


Figure 36: User Data Access Handler

7.2.19.2 Health Record Data Handler

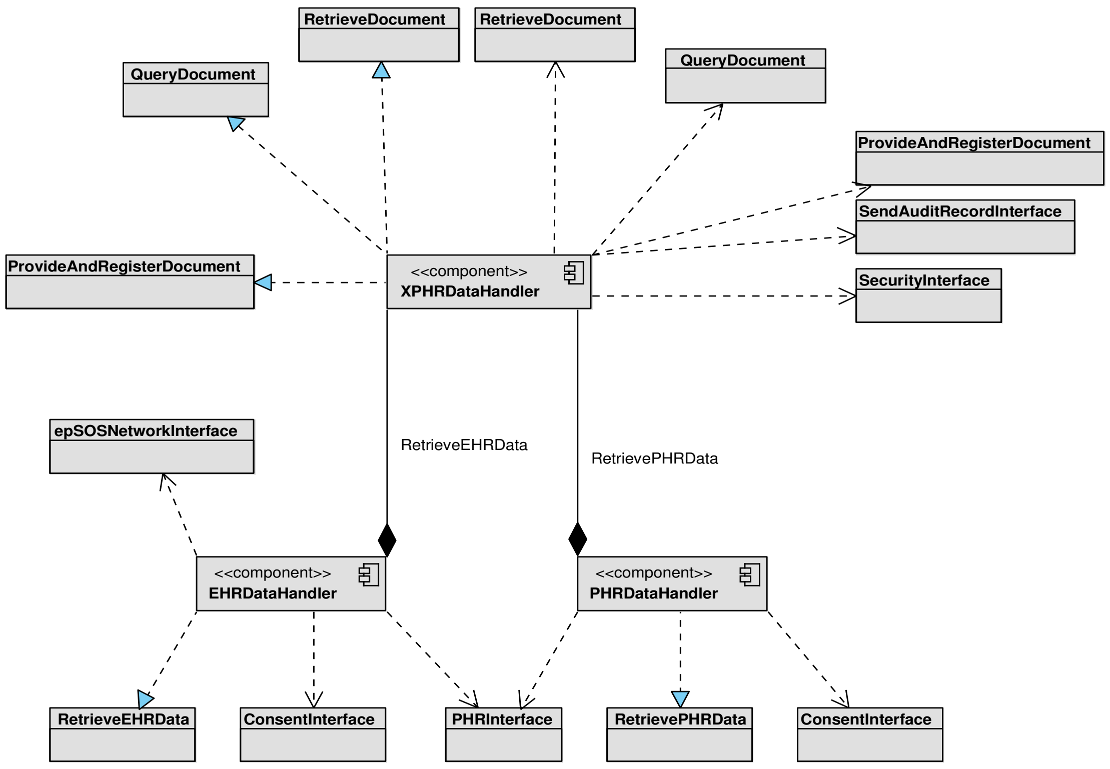


Figure 37: Health Record Data Handler – White-box

The integration of EHR systems that contains the data from the medical professional with the PHR system is handled through the XPHR Data Handler component. Through this component, integration with other PHA/PHR applications is also established. As shown in Figure 37, the XPHR Data Handler has two subcomponents (1) EHR Data Handler that is used to retrieve data from EHR systems and (2) PHR Data Handler, which is used to gather data from other PHR Systems. Both of these subcomponents use Consent Interface to check whether the patient gives the required consent before hand. After retrieving the medical data about the patient, it is stored to the used PHR through the PHRInterface component.

The access mechanism used to gather PHR and EHR data is IHE XPHR Profile, which is based on IHE XDS (Cross Enterprise Document Sharing) profile. In line with this profile, the data providers send their data to the PHR system through ProvideAndRegisterDocument component. On the other hand, the data receivers uses QueryDocument and RetrieveDocument component to gather medical data from the used PHR System. At each step in this process, necessary security precautions are taken through the SecurityInterface component and each step is logged through SendAuditRecordInterface.

7.2.19.3 Pathway Data Access Handler

This handler is configured by the configuration component and provides the Pathway Engine with data access services for the pathway engine and particularly the rule-based process engine and the process models. Additionally, this handler can provide service facades to the Pathway Engine for particular EMPOWER services from domain access, terminology and internationalisation and patient materials services.

7.2.19.4 Action Plan Data Access Handler

This handler is configured by the configuration component and provides data access services to the components of the Action Plan Engine Subsystem. The ActionPlanDataManager supports the Action Plan Engine with data management functionalities including support for the patient to manage their data associated with Action Plan.

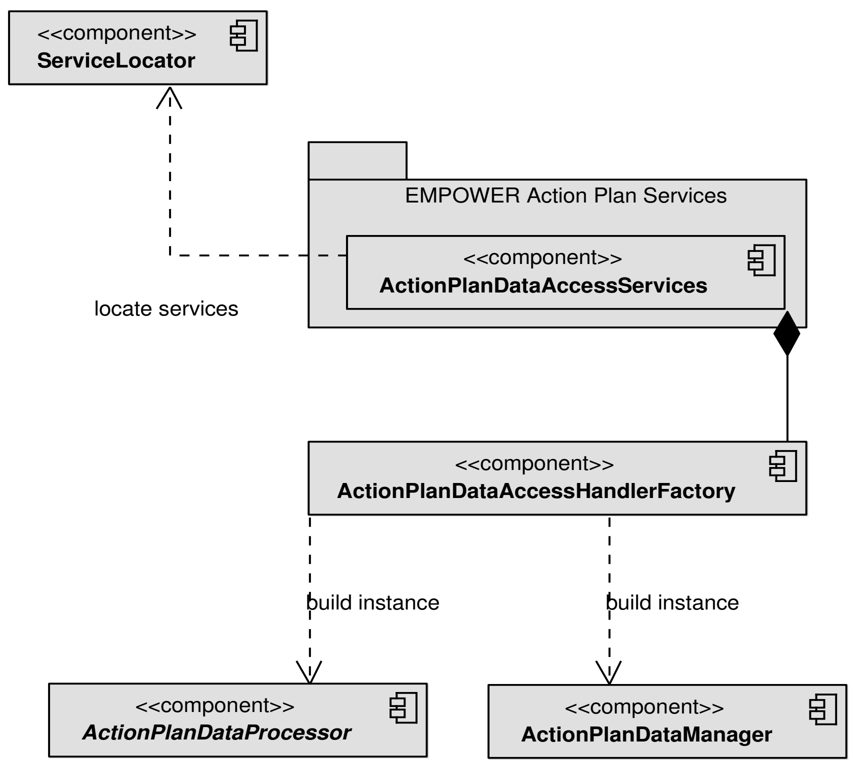


Figure 38: Action Plan Data Access – White-box

7.2.19.5 Vital Sign/Mental Parameter Handler

The generic Vital Sign/Mental Parameter Handler as depicted in Figure 39 can be used for all kinds of ODL data access, like Mental Health, Vital Sign and Nutrition Information.

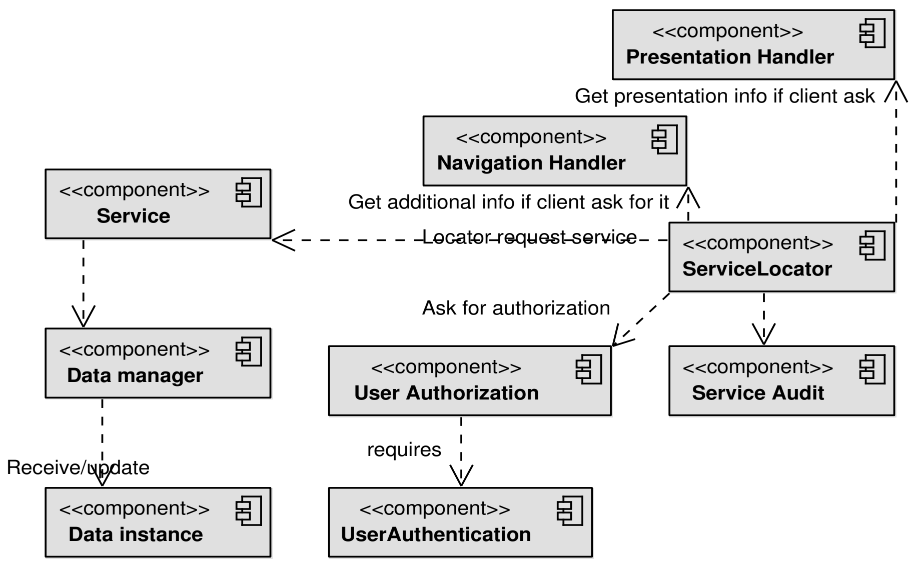


Figure 39: Vital Sign/Mental Parameter Handler

The Service Locator accepts a client's request and performs the following tasks:

- Calling the Audit Trail/Logging in order to track user's requests.

- Calling User authorisation in order to check if the client is authorized to use the service.
- Calling the specific Service in order to get the necessary info.
- Calling the Navigation handler in order to enrich the answer to the client with navigation info.
- Calling the Presentation handler in order to enrich the answer to the client with presentation info.

7.2.19.6 White and Yellow Pages Data Access Handler

White and yellow pages data access handler provides access to contact and location information of medical personnel, professional services include offline and online groups (e.g. forums) and patient information. Each pilot application must provide and approve personnel information, healthcare services information and patient information including the EMPOWER patient materials site and external health related web sites. Where relevant, the medical personnel should correspond to a user identity from an Identity provider either from EMPOWER or from an external system. The IHE Personnel White Pages (PWP) integration profile[29] will be consulted.

The term "Yellow pages" indicates that the information is categorised and accessible by particular categories using the EMPOWER terminology

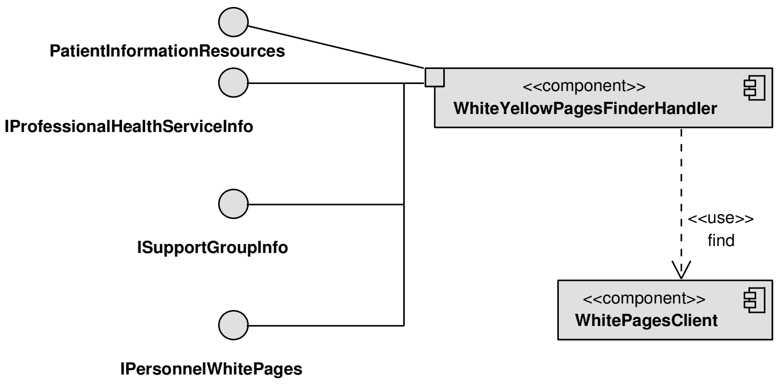


Figure 40: White and Yellow Pages Handler – White-box

The single components of the white and yellow pages handler have the following tasks:

- **WhitePagesClient:** The white page and yellow page client includes adaptors to the particular technology supporting a white page directory (like an LDAP directory of doctors).
- **WhiteYellowPagesFinderHandler:** This handler provides and configures the WhitePages client and software interfaces. It provides valid and authorized Contact information and/or identity information for medical professionals, organisations and healthcare services (support groups online and offline, forums, patient information sources including EMPOWER patient information materials). The types of contact information are categorized by the EMPOWER terminology for query and organisational purposes.
- **IPersonnelWhitePages, IProfessionalHealthServiceInfo, IPatientInformationResources** and **ISupportGroupInfo**: These are four tentative Service interfaces to access contact information and/or endpoints of personnel (medical professionals), patient information materials or support groups related information.

[29] http://wiki.ihe.net/index.php?title=Personnel_White_Pages

7.2.19.7 Recommender Engine Data Access Handlers

This data access handler is responsible to handle the persistency requirements of the execution of recommendation guidelines. For example, the medical professional may pause one of the instance of a guideline and restart it after a while. During this time duration, the engine may be shut down. In this case, the previously paused instance should not disappear. To overcome this, the state of the instance, before pausing, should be stored to a database. There may be other examples, such as the number of executing guideline instances, the guidelines themselves, etc. should be stored persistently. This Recommender Engine Data Access Handlers will realize these persistency mechanisms.

7.2.20 Knowledge Model Handlers (KMHs)

For each of the knowledge models a separate Handler will be provided and supported by the configuration component. The main task of the knowledge model handlers is to provide adaptors/connectors to either the existing PHRS data access services or to new repositories from EMPOWER for the PHRS.

The following seven handlers for knowledge models are currently considered to be required:

- Workflow and SMP Knowledge Model Handler
- Recommendation Knowledge Model Handler
- Action Plan Knowledge Model Handler
- Consent Management Knowledge Model Handler
- ODL and Medical Data Knowledge Model Handler
- Information Material Knowledge Model Handler
- Administrative Knowledge Model Handler (User Data)

Figure 41 presents an example handler for the pathway knowledge model. The selected process engine and rule engine will provide the necessary functionalities and services. The pathway knowledge model is composed of one or more rule-based BPM process models for the self-management pathways. More details on the knowledge models are available in D3.4.1.

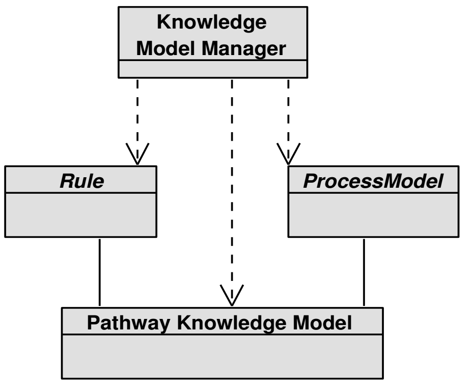


Figure 41: Pathway Knowledge Model – White-box

8 Runtime Views

The runtime views provide some of the most important processes to be performed within EMPOWER. While not every use case is covered in this document, the communications with architectural relevance are presented. Every scenario references to one or more Use Cases listed in Section 3.1.

8.1 Service Access Sequence

This sequence diagram presents the use of the service locator. This is usually the first step that needs to be made to access any of the provided EMPOWER services. It contains three different options: The first path ('1') depicts the "golden path", i.e. the optimal case, where both the authentication as well the authorisation was successful. In the second option ('2') already the user authentication fails in the beginning of the login-process. The third option ('3') allows the user accessing the service locator proxy due to positive authentication, but the access control (security policy enforcement point) denies the access for this user to the service itself.

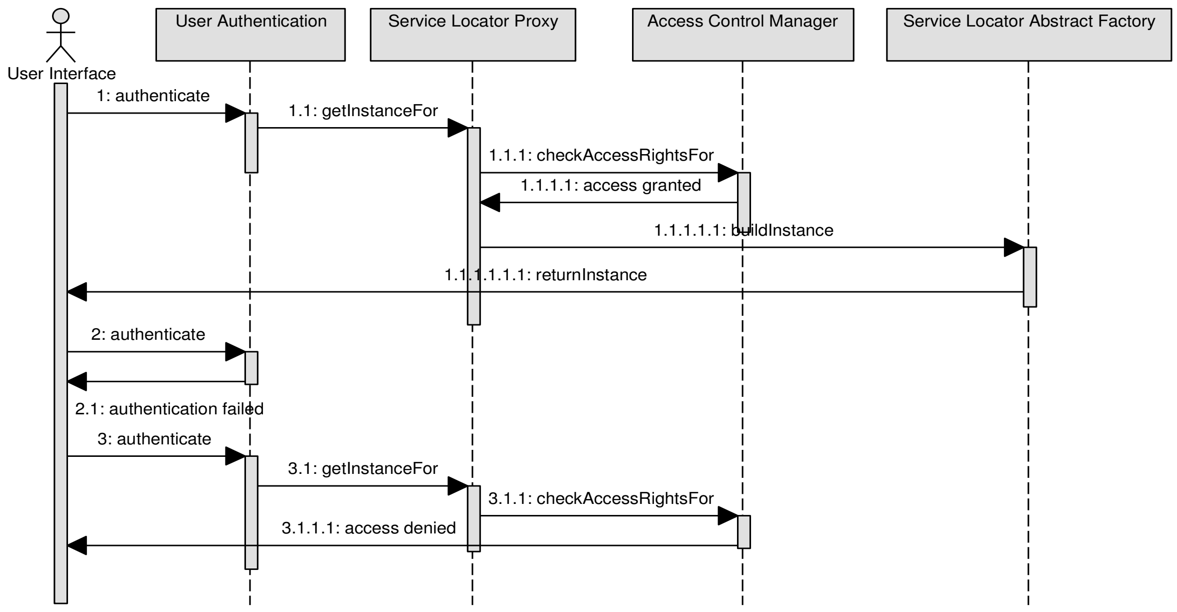


Figure 42 Service Access Sequence

It is noted here that the returned service instance itself is able to use the access control manager again for own purposes based on the defined security policies (e.g. to ensure that patients can only access their own patient data), while it does not need to perform another user authentication.

8.2 EHR-PHR communication

The PHR-EHR communication will be based on IHE XPHR Profile (UC-6.4.1 – UC-6.4.4). The EHR System (of a Hospital) will provide an ID Provider and XPHR Endpoint. The security of the communication line will be realized through IHE ATNA Profile (confidentiality, integrity and system level authentication). In other words, the certificates of the PHR and EHR would be exchanged first. In this way, it can be deduced that the EHR and PHR system trust each other.

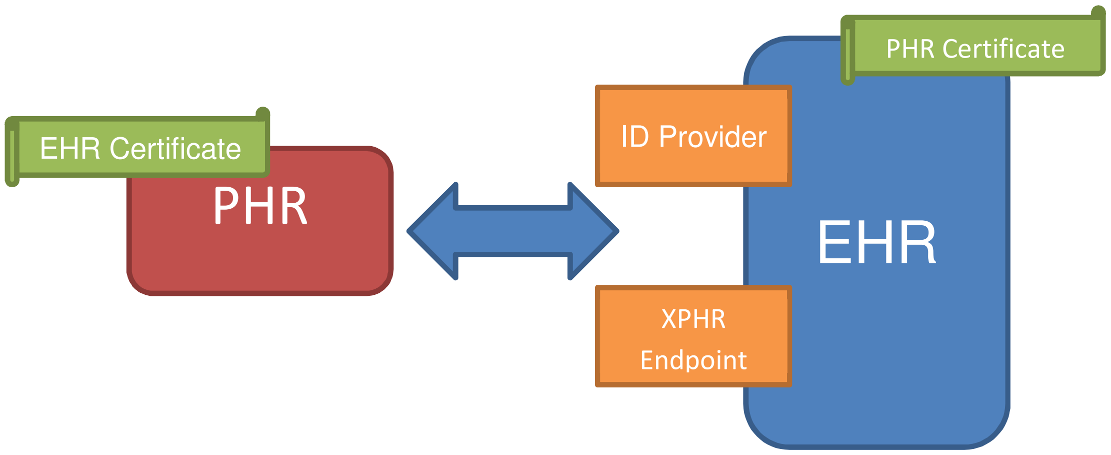


Figure 43: EHR-PHR Integration Architecture

Considering the user level authentication, when retrieving data from EHR to the PHR, the patient will first login to the ID Provider of the EHR and after successful authentication s/he will obtain a protocol id (unique id, secret id, etc.) to be used in the subsequent requests (This protocol id can be regarded as an authenticated session id like in Web browsing). After that the user will import his/her EHR data to the PHR system by using this protocol id. In the EMPOWER system, only the patients are allowed to import their EHR data to the PHR and the users can only retrieve their EHR data.

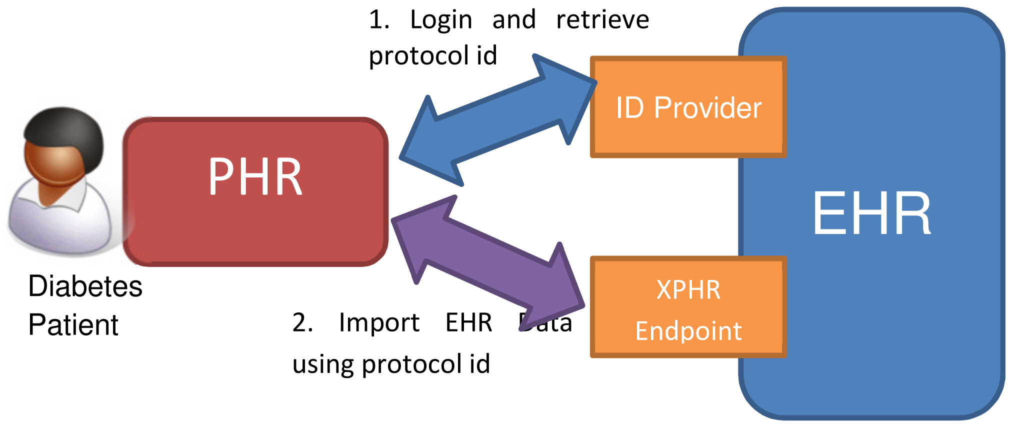

1. Login and retrieve protocol id
PHR
Diabetes Patient
ID Provider
EHR
XPHR Endpoint
2. Import EHR Data using protocol id

Figure 44: The EHR-PHR Integration – Executing Steps

After the EHR retrieval, the components of the EMPOWER system will be able to work on this data based on the consent of the patient. For example, Recommender Engine, during the execution of a diabetes guideline, requires some medical data about the patient. If the Recommender Engine is used by the doctor in a session and if the patient in question has not given consent to this doctor previously, the attempt to access to the medical data of the patient will be denied.

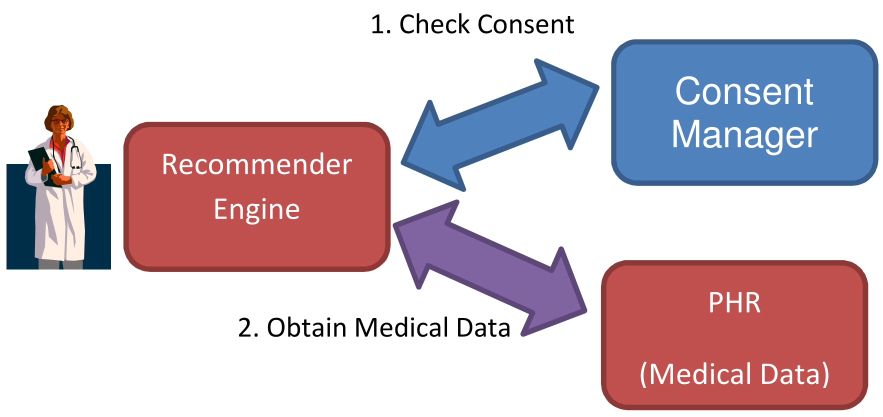


Figure 45: Recommender Engine – PHR System communication

It should be noted that at each point of interaction necessary audit logs will be sent to the Audit Record Repository.

8.3 PIS – PHR communication

In contrast to the scenario described Section 8.1, there will be also the need to export patient-related information (e.g. medication or diagnoses) from PIS systems like DocStar[30]. This requirement was introduced in D8.1.1 for the Pilot Application in Germany. In this case the following steps are performed:

1. GP opens patient record in PIS.
2. GP uses an “Export to EMPOWER” button to push the information required for the recommender and pathway engines to EMPOWER using EMPOWER web services.
    a. Because the patient related information stored in the PIS must be associated with the right patient a Master Patient Index is required for correct mapping. The Master Patient Index must be able to map the different IDs for the patient and guarantee the patient unique identity.
    b. Before data is exported, the patient consent needs to be requested from the consent manager.
3. Recommender Engine provides the recommendations.
4. A separate (browser) window to the EMPOWER system will be opened presenting the recommendations.
5. GP may modify (adapt/extend) the recommendations in EMPOWER.
6. GP pushes the (adapted/extended) recommendations to patients PHR system.
    a. Optional: Based on the PIS the recommendations may be also stored in the GPs patient record.
7. Depending on the patient consent, the accompanying EMPOWER PHR stores the patient information including recommendations or not.

As in the previous case, at each point of interaction necessary audit logs will be sent to the Audit Record Repository.

8.4 PHA-XPHR Data Transfer

During the execution of EMPOWER Framework, it may be necessary to exchange data between PHAs and the used PHR. The mechanism for this exchange will be based on IHE XPHR. The following steps will be applied in this communication:

[30] http://www.docstar.de

1. First, necessary certificates of the PHA and PHR should be exchanged so that they can trust each other and confidentiality can be achieved.
2. One important point to consider is that both of the applications have different user data base. Therefore, when sending a measurement data from PHA to PHR, the patient should log in to both systems. For this purpose, the PHR system will provide an ID Provider.
    a. The user will first log in to PHA.
    b. When the user wants to upload a medical data to PHR, the PHA will direct the patient to necessary log in page of the PHR.
    c. The patient will log in to PHR.
3. At this point, the data is formatted into IHE XPHR and sent to the PHR System through XPHR Client.

8.5 Reporting and Presentation

Figure 46 presents the workflow for creating reports or visualisations for presentation of EMPOWER information to the user. The information must be rendered depending on the abilities of the client device. After the service location, audit and user authorisation, the data is forwarded to the Navigation Handler, which takes a decision about the visualisation based on a set of meta-data provided by Multimodal services and interface subsystem and configuration. The information about rendering is processed (materialized) via the presentation handler in to the human visible user interface.

The workflow presents two optional paths: one for retrieving a plain visualisation (like a graph) and one to create a full report to the user.

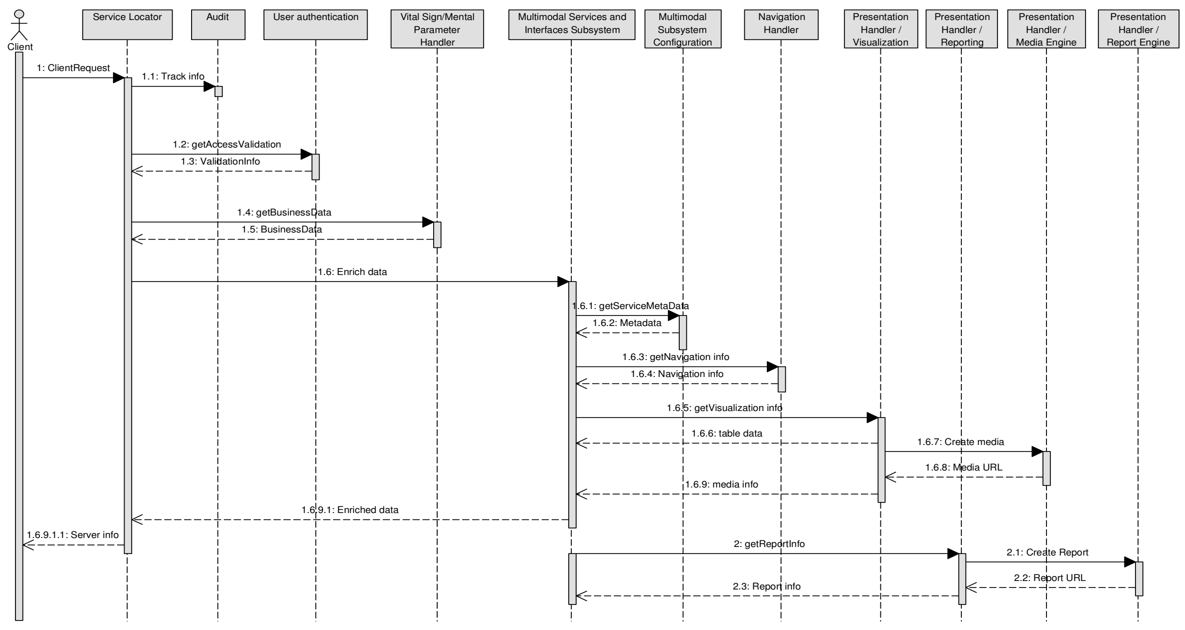


Figure 46: Workflows of Reporting or Presentation

9 Deployment Infrastructure

Running everything on one machine should be possible, to ease the implementation and integration process. However, the final deployment must allow the distribution of services to multiple machines.

9.1 Secure communication

To run on distributed machines, TLS must be enabled, as depicted in Figure 47. This requires the management and distribution of certificates. In order to enable certificates for multiple names on a single server, the use of TLS v1.2 with SNI support is recommended. Secure communication must be able to be turned off for the development purposes via the configuration.

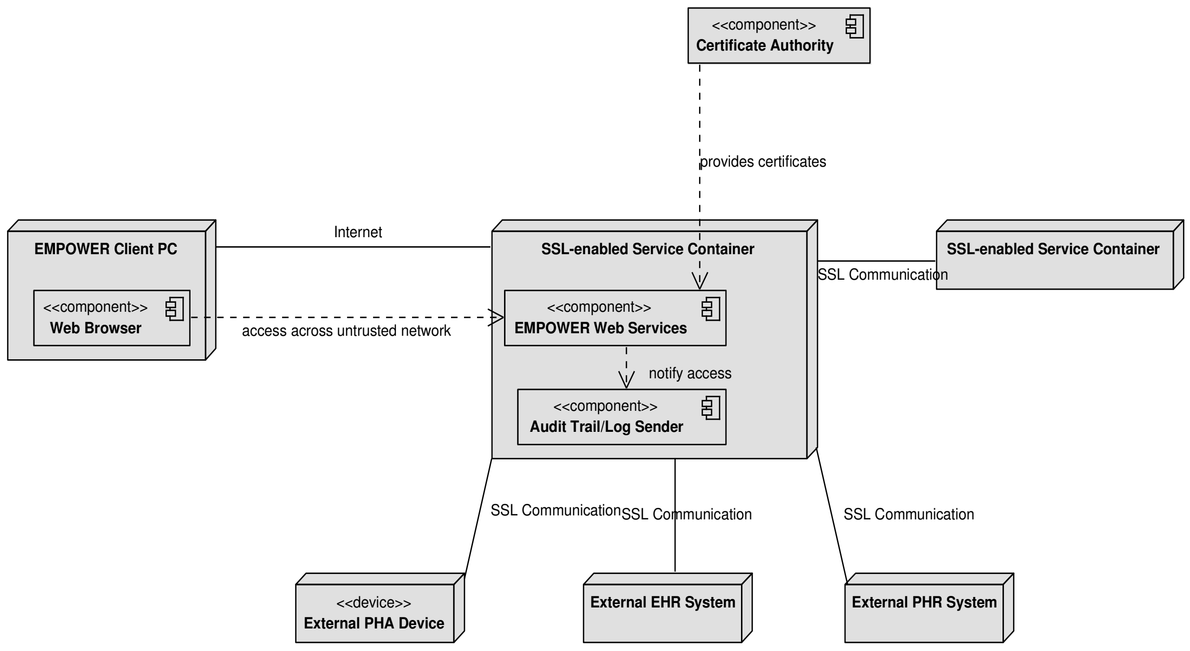


Figure 47: Secure Communication

9.2 Multiple Deployment Environments

The EMPOWER system should be deployable in different environments, especially without the need of multiple partners to separately install their own components. The following deployments are to be considered in the conceptual design:

- English demonstrator (e.g. for reviews/conferences, no real patient data)
- German demonstrator (no real patient data)
- Turkish demonstrator (no real patient data)
- German deployed pilot application (with test data for real patient)
- Turkish deployed pilot application (with test data for real patient)

Especially the deployments in the pilot applications may include multiple machines, due to the testing with real patient data.

The above listed demonstrator may use the latest Release Candidate, while the deployments of the pilot applications need to run only releases of bug-fix releases of the EMPOWER architecture implementation. In addition to these “final deployments”, the following deployments containing the working version will be available:

- Development Deployment: This deploy is triggered by changes in the repository by using a CI-Server. Always the latest version of the source code repository will be compiled, tested and deployed. In case of compilation or test failures a message to the latest committer responsible for failing should be generated.
- Testing Deployment: This is a local predecessor of the demonstrators. It may be used to run tests of the prototype before the demonstrators are finalised. This deployment can also be run by the CI-Service, but is always triggered manually on demand.

10 Technical Concepts and Architectural Aspects

This chapter covers cross-cutting concerns related to the EMPOWER system.

10.1 Configurability

- The Configuration Service provides a unified way to configure various aspects of the EMPOWER configuration.
- The Configuration Service will be able to support multiple configuration file formats based on the requirements: **xml**, java.properties (key value pairs), JSON.
- Maven can be used to generate configuration files, which however is not encouraged. Instead Maven profiles (e.g. one for each pilot deployment, test/production) must be used.
- Due to the different deployment environments, also multiple configuration profiles need to be supported:
    - English default demonstrator (e.g. for reviews/conferences)
    - German demonstrator
    - Turkish demonstrator
    - German deployed pilot application
    - Turkish deployed pilot application
    - Developer Profile Template (each developer can use his own configuration for development purposes)
- The configuration component will send notifications about changes to the configuration monitor (other components can subscribe to this monitor).

10.2 Persistency

EMPOWER system must be able to persist (and retrieve) information. According with the requirements the information can materialize in a variety of forms (e.g. open-EHR[31] ADL [32]artefacts, IHE – XPHR[33] profiles, property formats, etc.), one of the crucial aspect in EMPOWER is to define an atomic and unified unit of information able to satisfy all the needs/constrains defined with the upper mentioned profiles. Those atomic and unified units of information can be used at least in two ways:

- to duplicate the information
- to relate existing information (in different format) like meta-information, this meta information can be used by specified services in order to manage and manipulate the information. This approach allows harmonious usage of different kind of storage.

In most of the cases the access to information will be done via specialized services – all these services will be collected in to a single (pluggable) layer. This layer must accomplish at least:

- Independent of the underlying technologies (e.g. Relational database, NOSQL, Document Management System, etc.)
- Independent of any domain specific models, no specialisation will be permitted (by specialisation I mean a service able to serve a certain kind of domain model like blood pressure).
- Provide all the storage related abilities (e.g. transactional support, roll back, back-up, etc.)

The persistence layer will also provide information pipes for a lot of information related operations such as: transformation, content enrichment, filtering, routing, etc. For this functionality an ESB is foreseen to be used.

[31] open-EHR : http://www.openehr.org/home.html

[32] ADL: http://www.openehr.org/svn/ref_impl_eiffel/TRUNK/apps/adl_workbench/doc/web/index.html

[33] IHE – XPHR: http://wiki.ihe.net/index.php?title=PCC_TF-1/XPHR

In EMPOWER, basically two options for data handling are used. Depending on the underlying data model (relational or document-based) one of the approaches will be applied separately by each of the subsystems.

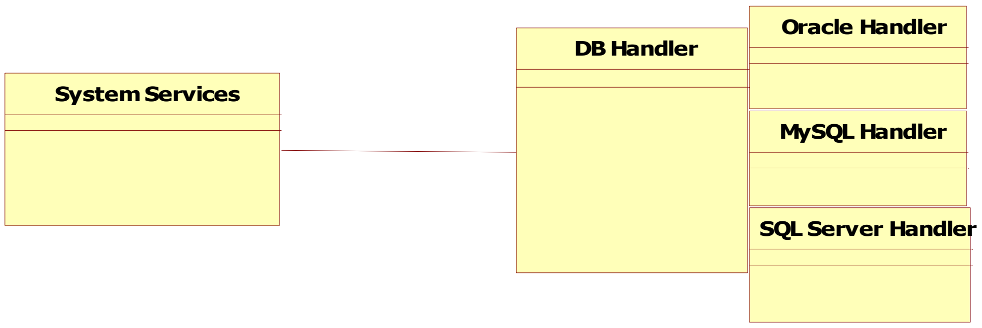


Figure 48: DB Handler for relational databases

In the relational case, as depicted in Figure 48, a DB Handler component offers the ability to choose between database vendors without affecting the design or the implementation of the system. This is similar with what the JPA[34] () or JDBD is doing. Beside this it is important to notice that the DB handler is part of the business layer and only the xxxHandler are part of the persistence layer. The component can cooperate with other independent vendor oriented components (Oracle, MySQL, SQL Server) offering to the system a common interface.

In addition to persisting data in relational databases, there will be other mechanisms to store information in RDF Triple Stores, noSQL databases (like MongoDB, CouchDB) or search indexes (like Apache Solr).

10.3 Flow Control

The flow control can be a stateless, and it must be applicable over different flows, in this way the flow control can be reused and even standardized (e.g. a standard way to react on a certain event or error). Here are some of the most popular flow controllers:

- Based on rules provided by rule engine (Process Engine)
- Based on events
- Based on exceptions

Hardcoding of flow control is therefore discouraged. As much logic as possible (like if's, switch, etc.) should be in the rules, rather than in the (Java) code.

10.4 Transaction Processing

Most of the actual frameworks provide support for the transactions. The transaction processing is done in most of the cases transparent to the user. One of the most popular transaction processing standards is JTA[35]. Most of the actual implementations are based in this standard. EMPOWER will also use a JTA based solution.

10.5 Session Handling

EMPOWER will use HTTP session according with the Servlet API 3.0[36]. The Servlet API 3.0 is implemented by most of the modern servlet (web) containers.
Sessions duration must be reasonably limited, i.e. no unlimited session timeout is allowed.

[34] http://www.oracle.com/technetwork/java/javaee/tech/persistence-jsp-140049.html
[35] JTA – http://www.oracle.com/technetwork/java/javaee/jta/index.html
[36] Servlet 3.0 API - http://tomcat.apache.org/tomcat-7.0-doc/servletapi/index.html

10.6 Security

EHR/PHR Authentication can be done by an external ID Provider. OAuth2.0 is preferably used. For the development deployment there will be a maven script to generate the certificates. The certificates should not be shared using the VCS or anywhere else. In the pilot phase there will be one partner responsible in the project to play the role of the certification authority.

10.7 Communications and Integration with other Software Systems

10.8 Distribution

EMPOWER components (sub-systems) will be able to run on different JVM instances located on different physical machines, connected via TCP/IP. Possible firewalls between the distributed systems are required to be configured appropriately. The service locator will be used to locate services (and interconnect components and subsystems).

10.9 Exception/Error Handling

Exception swallowing is discouraged. At least a log message must be produced. Whether exceptions are thrown further or locally handled depends on the used framework specification.

10.10 Logging, Tracing

Log4j using simple logging façade for java (SL4J) is recommended to be able getting a central log-file. The preferred logging configuration of the developers must stay in the test package. A per-module configuration in the main-tree is discouraged. Logging is not internationalized, only English must be used as far as possible (also avoid e.g. getLocalizedMessage() to allow easy log-file exchange for bug-fixing between partners.
The six log levels should be chosen appropriate, from detailed to minimum they are: TRACE > DEBUG > INFO > WARN > ERROR > FATAL.

All access to patient information must be at least logged using the Audit Trail/Logging Subsystem.

10.11 Parallelisation and Threading

Based on the high-level framework (like EJB, or Servlet-API specification) parallelisation should be preferred compared to low-level approaches like “synchronize” or other concurrency packages from the Java, Apache, etc.

10.12 Internationalisation

Internationalisation is required to support at least three different languages, which are German (de), English (en) and Turkish (tr). Most of the EMPOWER nomenclature will be derived from standard repositories. Mapping between languages will therefore be done by using domain-specific dictionaries like ICD, SNOMED-CT. Internationalisation in EMPOWER will therefore be handled by a separate subsystem, the Terminology Subsystem as described in Section 0, providing a single access point for all translation between languages. To support all required characters UTF-8 encoding will be used.

10.13 Testability

Unit tests should only be used for testing, while for manually triggering some process “Examples” should be created. For starting the examples accompanying sample scripts should be provided (e.g. containing a corresponding maven-call).

Unit tests should contain comments to indicate what aspects of the specification (contract) are covered. Unit tests will be triggered by the CI-Server. They should be independent, as the order of execution is not guaranteed.

10.14 Code Generation

In some cases code generation may be desirable (e.g. JAXB) but should be trigger-able from the command-line e.g. by Maven, rather than from modelling tools.

10.15 Build-Management

In the EMPOWER system development process the Maven 3 build lifecycle will be applied. A Maven target to create an empty EMPOWER module is available.

11 User Interfaces

The EMPOWER system will require multiple user interfaces, depending on the user role and the used device the user interfaces will change. As defined in the user cases, there will be a main entry point to EMPOWER, which is the "EMPOWER dashboard".

The details on user interfaces will be available in Deliverable 8.2.1 – Design of the EMPOWER Pilot Application, where also a storyboard of user interfaces will be sketched. For the conceptual architecture, the following is pointed out to allow for a flexible approach in the user interface implementation:

- The graphical style of the user interface must be separated from its functionality. Therefore theme(s) for all user interfaces will be handled separately in a central css depending on the used framework (e.g. by a jQuery Theme[37]). Separate designs may be needed for different end-user devices (e.g. Desktop-PC and Smartphone)
- Also the Frontend Layout shall be centralised and must support internationalisation (in best case by framework support)
- Validation rules are required for input data. They should be on a central place, in the best case they are already part of the used knowledge model.

Finally, it is important to note that the User Interface Design must support different maturity levels of the user. The maturity level shall therefore be reflected dynamically in the implementation, e.g. in the configuration, so that adaptation according to the users' maturity levels can be applied easily.

In the following subsections, three different options for the implementation of the user interfaces are shown. The Navigation Handler provides the abstraction between the system services and the front-ends running on the different devices.

11.1 Android Interface (mobile)

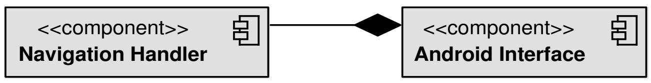


Figure 49: Android Interface

The component Android Interface offers the ability to Android clients to navigate through the system services using a service API following the configurable rules of the Navigation handler.

11.2 HTML Interface (web)

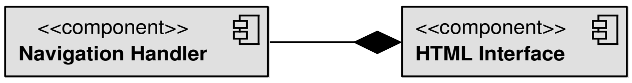


Figure 50: HTML Interface

The component HTML Interface offers the ability to HTML clients (using browsers) to navigate through the system services using the HTTP service API following the configurable rules of the Navigation handler.

[37] http://jqueryui.com/themeroller/

11.3 Swing Interface (desktop)

The component Swing Interface offers the ability to swing clients (stand-alone applications) to navigate through the system services using a service API following the configurable rules of the Navigation handler.

12 Glossary

Term	Description
Knowledge Model	Includes dynamic (rules, processes, workflows) and static (ontologies, schema, and vocabularies) knowledge. The knowledge models are based on well-established standards (e.g. CEN 13606, HL7 RIM-based application architecture), and on appropriate medical terminologies (e.g. LOINC, SNOMED-CT). Standards are required for data interoperability.
Rules	Rules in EMPOWER can categorised as following: 1. Rules for data validation (web form) 2. Rules for modelling a business process model 3. Rules for reminders or alerts 4. Rules for Decision making by a patient or doctor might be related to 2+3. Internationalization support must be provided for most rules that provide feedback to a user.
Knowledge Worker	A knowledge worker is an actor managing knowledge such as process models and EMPOWER terminologies
Personal Health Record System	A PHR System is a collection of PHA's.
Personal Health Application	PHA is a generic name for all different possible types of personal health applications. A PHA can be a desktop, mobile or another specific type of application.
Electronic Health Record System	An EHR System residing in a hospital can reach some of the functionality of the EMPOWER system. For example, the EHR system can reach the PHR information of a diabetes patient.
EMPOWER Inbox	The EMPOWER inbox provides communication services between the user and the system as expressed by the Business Process Models (Human task integration). A PHA inbox will use these services to integrate and process EMPOWER messages.
Action Plan	The patient uses an Action Plan to facilitate the self-management of their disease.
Information Material	Patient Information material is content created and/or assembled, and enriched by EMPOWER medical partners. This material is stored in an open source content management system and the content can include decision aids, texts, and associated multimedia resources. Metadata can include terms from EMPOWER terminologies to enhance searching
Patient	This actor is for the diabetes patients themselves. The main beneficiary of the EMPOWER project. The family member of a patient can also perform tasks on behalf of the patient if the patient gives consent. In EMPOWER, the family members are also assigned to Patient role as they may have an account in the PHR, too. In the PHR, a patient can set another patient as a family member. In other words, in addition to giving consent to a medical professional, the patients can give consent to other patients, too.
System Administrator	The System Administrator manages the system configuration and data including patient data within the system.
Medical Professional	This actor is assumed to be an expert in the diabetes. The Medical Professionals, general practitioners, dietician, nurses, etc. are considered to be in this category.
EMPOWER System	All EMPOWER components are defined inside the EMPOWER system
Content	The Content Administrator provides and updates the information

Administrator	material available in the EMPOWER PHRS
EMPOWER UI Dashboard	User is provided with a dashboard with menus to navigate EMPOWER menu and interact with EMPOWER functionality.
User Authentication	Is the access control element that authenticates the users.
Audit Record	A record of an auditable event. Usually saved in an Audit Record Repository.
Identity Manager	The identity manager provides the link between IDs of multiple systems, each of them maintains its own User-IDs.
User Authorisation	User Authorisation is the access control element that decides whether a user may access a specific resource.
Medication	Information about drug including dosage, administration, etc. taken by a Patient
EMPOWER configuration	EMPOWER Configuration is kept central.
ODL Collector	Any device/software for collecting observations of daily living.
EMPOWER Service Locator	EMPOWER Service Locator
Identity Provider	Internal or external ID provider
Mental Health Support	Mental illnesses could hamper self-care and diabetes treatment. Help and support should be accessible anytime. The system provides mental health support for depression and stress related issues.
Recommender Engine	The Recommender Engine component will provide recommendation to the Medical Professionals by examining the EHR/PHR data and well-known clinical guidelines
EMPOWER service	Any internal or external service provided by one of the EMPOWER subsystems.
Physical Activity	A specific ODL for a physical activity.
Machine Processable Diabetes Guideline	The medical best practice is currently documented through textual guidelines[38]. In order to support the execution of these textual guidelines through computers, they are described through GLIF (Guideline Interchange Format). In the EMPOWER context, Machine Processable Diabetes Guideline are the diabetes guidelines described through GLIF.
Pathway Engine	The Pathway Engine is the central messaging and notification service that may be used by all components.
Observations of Daily Living (ODLs)	ODLs are any patient-generated observation about their health; however, they are not clinically generated data. To be consistent, when the patient imports clinical data, EMPOWER categorizes the imported data as ODLs.
PHR Data	Information stored in the Personal Health Record
IDE for BPM Modelling	Interactive Development Environment with appropriate plugins to support rule based BPM modelling. To support iterative development: Storyboards can be used together with BPM visual design tool. The resulting diagrams might feedback to the story boards
EMPOWER Knowledge Model	A collection of rule based BPM process models, Knowledge base includes rule base and static EMPOWER terminologies.
Pathways Knowledge Model	EMPOWER Knowledge Model supporting the EMPOWER self-management pathway.
Action Plan Knowledge Model	EMPOWER Knowledge Model supporting Action plans. The Pathways Knowledge Model will provide particular high-level processes to coordinate the Action Plan among the system

[38] http://guidelines.gov/

	components and EMPOWER users.
System Knowledge Models	Knowledge models needed for the EMPOWER system. These are associated with the Pathways Engine
Component Knowledge Models	To support modularization and potentially reusability. These are not relevant to the EMPOWER Knowledge Model. Any component can use BPM and any process engine. For example, a PHRS or PHA has functionalities (workflows, page web flow) that can benefit from BPM. One idea is to support storyboards and their implementation with a BPM visual design tool.