FP7-ICT-611140 CARRE

Project co-funded by the European Commission
under the Information and Communication Technologies
(ICT) 7th Framework Programme

D.2.5. CARRE Architecture

Z. Deng, G. Drosatos, G. Gkotsis, R. Kloda, E. Liu, V. Marozas,
J. Piwinski, A. Solosenko, X. Zhao

November 2014

CARRE Contacts

Project Coordinator:	Eleni Kaldoudi	kaldoudi@med.duth.gr
Project Manager:	Koula Zigeridou	kzygerid@alex.duth.gr

DUTH Democritus University of Thrace	Eleni Kaldoudi	kaldoudi@med.duth.gr
OU The Open University	John Domingue	john.domingue@open.ac.uk
BED: University of Bedfordshire	Enjie Liu	Enjie.Liu@beds.ac.uk
VULSK: Vilnius University Hospital Santariskių Klinikos	Domantas Stundys	Domantas.Stundys@santa.lt
KTU Kaunas University of Technology	Arunas Lukosevicius	arunas.lukosevicius@ktu.lt
PIAP Industrial Research Institute for Automation & Measurements	Roman Szewczyk	rszewczyk@piap.pl

Disclaimer

CARRE is a Specific Targeted Research Project partially funded by the European Union, under FP7-ICT-2013-10, Theme 5.1. “Personalized health, active ageing & independent living”.

Document Control Page

Project

Contract No.:	611140
Acronym:	CARRE
Title:	Personalized Patient Empowerment and Shared Decision Support for Cardiorenal Disease and Comorbidities
Type:	STREP
Start:	1 November 2013
End:	31 October 2016
Programme:	FP7-ICT-2013.5.1
Website:	http://www.carre-project.eu/

Deliverable

Deliverable No.:	D.2.5
Deliverable Title:	CARRE Architecture
Responsible Partner:	OU – G. Gkotsis and PIAP – J. Piwinski, R. Kloda
Authors:	(alphabetical order) Z. Deng (BED), G. Drosatos (DUTH), G. Gkotsis (OU), R. Kloda (PIAP), E. Liu (BED), V. Marozas (KTU), J. Piwinski (PIAP), A. Solosenko (KTU), X. Zhao (BED)
Input from:	All partners
Peer Reviewers:	Arunas Lukosevicius (KTU), Youbing Zhao (BED), Eleni Kaldoudi (DUTH) External reviewer: Debora Testi (CINECA)
Task:	T.2.5. CARRE environment design
Task duration:	6 months: 1 May 2013 to 31 October 2014
Work Package:	WP2: Domain analysis, requirements and design
Work Package Leader:	DUTH – Eleni Kaldoudi
Due Date:	31 October 2014
Actual Delivery Date:	21 November 2014
Dissemination Level:	PU
Nature:	R
Files and format:	Deliverable report: 1 pdf file
Version:	05
Status:	☐ Draft ☒ Consortium reviewed ☒ WP leader accepted ☒ Coordinator accepted ☐ EC accepted

Document Revision History

Version	Date	Modifications	Contributors
v01.0	20Oct 2014	New content – outline	J. Piwiński R. Kłoda
v1.1	21 October	BED input to visualisation	BED
v2.0	31 October	draft	
v3.1	10 Nov	updated structure	George Gkotsis
v3.2	11 Nov	Revised architecture high-level view, inserted early content in sections 3 & 4.	G. Gkotsis, E. Kaldoudi, J. Piwiński, E. Liu, X. Zhao
v.3.3	12 Nov	Added content from the earlier version of the deliverable (section 4 - components). Added content in sections 1 and 3.	G. Gkotsis,
v.3.4	13 Nov	Content added from BED (sections 3 and 4, subsection on security), PIAP (sections 3 and 4), OU (sections 3, 4 and 6).	E. Liu, X. Zhao, J. Piwiński, G. Gkotsis.
v.3.5	14 Nov	More content added in missing sections.	E. Liu, X. Zhao, J. Piwiński, G. Gkotsis.
v.3.6	14 Nov	Content added by Eleni Kaldoudi. Content based on Skype discussion of Friday, November 14.	E. Kaldoudi, E. Liu, X. Zhao, J. Piwiński, G. Gkotsis
v.3.7	15 Nov	Content in Sections 3, 4, and 5 Formatting to the required layout.	E. Kaldoudi, G. Drosatos
v.3.8	15 Nov	Content added by BED, PIAP. Content added in section 3 (CARRE objects & services) by George Gkotsis.	E. Liu, X. Zhao, J. Piwiński, G. Gkotsis.
v.3.9	16 Nov	Content added in section 3 and section 4	E. Kaldoudi
v.4.0	16 Nov	Content added in Section 5, Corrections, comments in several places, content added in subsection 4.1.2 and section 5.	G. Drosatos, G. Gkotsis, KTU
v.4.1	17 Nov	Formatting and style editing	E. Kaldoudi
v.4.2	18 Nov	Review from BED, KTU	G. Gkotsis
v.4.3	20 Nov	Review from DUTH, final edits from KTU, feedback from BED	G. Gkotsis
v.5	21 Nov	finalized	E. Kaldoudi

Table of Contents

List of Figures

List of Tables

Executive Summary

This document reports on the architectural design of the overall CARRE services environment. The CARRE architecture is described through the definition of the major software components and the description on how these components operate with one another. Furthermore, it also discusses security issues to be considered during the development stage of each component. Finally, an illustrative collection of sequence diagrams presents how the system operates for different user actions.

About CARRE

CARRE is an EU FP7-ICT funded project with the goal to provide innovative means for the management of comorbidities (multiple co-occurring medical conditions), especially in the case of chronic cardiac and renal disease patients or persons with increased risk of such conditions.

Sources of medical and other knowledge will be semantically linked with sensor outputs to provide clinical information personalised to the individual patient, so as to be able to track the progression and interactions of comorbid conditions. Visual analytics will be employed so that patients and clinicians will be able to visualise, understand and interact with this linked knowledge and also take advantage of personalised empowerment services supported by a dedicated decision support system.

The ultimate goal is to provide the means for patients with comorbidities to take an active role in care processes, including self-care and shared decision-making, and also to support medical professionals in understanding and treating comorbidities via an integrative approach.

Terms and Definitions

The following are definitions of terms, abbreviations and acronyms used in this document.

Term	Definition
.NET	.NET Framework is a software framework developed by Microsoft that runs primarily on Microsoft Windows. http://www.microsoft.com/net
API	Application programming interface (API) is a set of functions and procedures that allow the creation of applications which access the features or data of an operating system, application, or other service.
Architecture (software)	The high-level structure of a software system.
CBSE	Component-Based Software Engineering is a software engineering approach that asserts a separation of concerns using components.
Component (software engineering)	An individual software component is a software package, a web service, a web resource, or a module that encapsulates a set of related functions (or data).
CSV	A comma-separated values file stores tabular data (numbers and text) in plain-text form.
DB	Database
DSS	Decision Support Service: Acquired knowledge application to support non-expert in given area to take right decision and solve a problem
DoW	Description of Work.
GATE	General Architecture for Text Engineering (GATE) is a human language processing system. https://gate.ac.uk/
Google search	Google search is the most-used search engine on the World Wide Web, handling more than three billion searches each day.
HealthVault	A web-based platform from Microsoft to store and maintain health and fitness information. https://www.healthvault.com
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol over Secure Socket Layer
ICD	The International Classification of Diseases (ICD) is the standard diagnostic tool for epidemiology, health management and clinical purposes. http://www.who.int/classifications/icd/en/
ICT	Information and communication Technology
ID	Identification number.
IP	Internet Protocol
IPSec	Internet Protocol Security
IT	Information Technology
Java	A general-purpose computer programming language designed to produce programs that will run on any computer system. https://www.java.com/
JavaScript	An object-oriented computer programming language commonly used to create interactive effects within web browsers.
JSON	JSON or JavaScript Object Notation, is an open standard format that uses human-readable text to transmit data objects consisting of attribute–value pairs. http://json.org/

Lemur	An open-source (BSD license) software framework for building language modeling and information retrieval software. http://sourceforge.net/p/lemur/wiki/Home/
Linked Data	A method of publishing structured data so that it can be interlinked and become more useful.
Medical expert	End-user of CARRE project. Internist, Cardiologist, Nephrologist
MedLinePlus	The National Institutes of Health's Web site for patients and their families and friends, http://medlineplus.gov.
MyHealthAvatar	An EU-funded project for "A Demonstration of 4D Digital Avatar Infrastructure for Access of Complete Patient Information". http://www.myhealthavatar.eu/
OAuth	OAuth is an open standard to authorization. http://tools.ietf.org/html/rfc5849
Patient	End-user of CARRE project. Person, who is at risk of heart or renal disease
PHR	A Personal Health Record (or PHR) is a health record where health data and information related to the care of a patient is maintained by the patient.
PubMed	A service of the US National Library of Medicine that provides free access to MEDLINE, the NLM database of indexed citations and abstracts to medical, nursing, dental, veterinary, health care, and preclinical sciences journal articles. Accessible at http://www.ncbi.nlm.nih.gov/pubmed/.
Pub/sub	In software architecture, publish–subscribe is a messaging pattern where senders of messages, called publishers, do not program the messages to be sent directly to specific receivers, called subscribers. Instead, published messages are characterized into classes, without knowledge of what, if any, subscribers there may be. Similarly, subscribers express interest in one or more classes, and only receive messages that are of interest, without knowledge of what, if any, publishers there are.
Python	A high-level general-purpose programming language. https://www.python.org/
RESTful	A web service that complies with the REST software design style.
RDF	RDF (Resource Description Framework) is a standard model for data interchange on the Web.
REST	Representational State Transfer (REST) is a software design style reflecting the architecture of the HTTP protocol in which functionalities are provided through resources.
Sensor data	Measurements coming from sensor devices used by the patients/end-users of CARRE.
SPARQL	An RDF query language. http://www.w3.org/2009/sparql/
SSH	Secure SHell
SSL	Secure Socket Layer
SW	Software
TCP	Transmission Control Protocol
Twitter	An online social networking service that enables users to send and read short 140-character messages called "tweets". https://twitter.com/
UC	Use Case
UML	The Unified Modeling Language (UML) is a general-purpose modeling language in the field of software engineering, which is designed to provide a standard way to visualize the design of a system. http://www.uml.org/
UML	Unified Modelling Language
Virtuoso	Virtuoso Universal Server is a middleware and database engine hybrid that combines the functionality of a traditional RDBMS, ORDBMS, virtual database, RDF, XML, free-text, web application server and file server functionality in a single system.

	http://virtuoso.openlinksw.com/
Visualisation	A technique to create images and diagrams for CARRE data to be presented to end users in collaboration with the CARRE decision support services.
Vivaport	A patient health summary system, created by the cooperation of 19 partner organisations from 8 European countries. https://vivaport.eu/
Weka	Weka is a collection of machine learning algorithms for data mining tasks. http://www.cs.waikato.ac.nz/ml/weka/
WordNet	WordNet is a lexical database for the English language. http://wordnet.princeton.edu/
WP1 – WP8	Work package1-8
WS	Web Service
XML	Extensible Markup Language (XML) is a markup language that defines a set of rules for encoding documents in a format that is both human-readable and machine-readable.

1. Introduction

This document reports on the architectural design of the overall CARRE services environment. The CARRE architecture is described through the definition of the major software components and the description on how these components operate with one another.

CARRE architecture adopts the Component-Based Software Engineering approach (CBSE)[1]. CBSE asserts the separation of concerns into lower level software artifacts (called components) with the aim to simplify the development of a previously more complex system. CBSE requires an inherent *structural* approach towards the design of the system where the main parts of the system are identified from the beginning. At the same time, the system components are designed in such a way that they have a high degree of independence. Hence the effort for developing these components can be assigned to different parties while they can also be reused easily from third-party developers.

The present document is following up on project reports created until now and extends the understanding on how the CARRE services environment will be ultimately implemented. Hereby, we discuss how the architecture document is informed by already submitted CARRE deliverables:

– *D.2.1 Domain analysis & use case definition*[2]. The document gives an insight on the nature of cardiorenal comorbidities by providing an analysis of the relevant medical domain. It also discusses the current state concerning legal issues and provides a series of illustrative CARRE use cases. All of the above have been taken into account in the design of CARRE architecture.

– *D.2.2 Functional Requirements & CARRE Information Model*[3]. The document identifies and models the core concepts used for describing cardiorenal comorbidities. It proceeds with listing the functional requirements that need to be met from an end-user perspective. The model has already been used for the design of the CARRE components while the functional requirements allow us to design the components and their interactions having in mind the user requirements that need to be met.

– *D.2.3 Data source identification & description*[4]. The document provides a great deal of detail on the various data sources and how these sources can be used for acquiring the desired information (concerns both public and personal data).

– *D.2.4 CARRE metadata scheme & ontology*[5]. The deliverable is the first that includes a CARRE software artifact: the CARRE ontology. The ontology introduces the classes that are going to be used for describing risk associations as well as sensor data measurements. Furthermore, the report presents the "risk associations data entry" component that was developed to enable medical experts to insert risk factor descriptions into the public RDF[6] repository. The decisions made within D.2.4 have been taken into account for designing the low-level components of the CARRE services.

– *D.3.1 Aggregator module generic design*[7]. The report presents the design for the personal data sensor aggregator mechanisms. This has been taken into account when designing the overall CARRE architecture.

The current document aims to describe a technical roadmap for the remaining of the development activities. It identifies and defines the major software components that are crucial for meeting the requirements of CARRE. Depending on the progress of the project up until now (i.e. the progress of concurrent, on-going tasks dedicated to the development of individual components), where available we provide greater detail for each component. However, the aim here is to define the components, outline their purpose in relation to others and present how these will be integrated into the overall CARRE environment.

The remainder of this document is structured as follows: Section 2 presents the overall, high level CARRE architecture. Section 3 describes the basic objects used within CARRE together with the basic services. Section 4 presents in detail each of the architectural components (illustrated earlier in Section 2) while

[1] R. Weinreich and J. Sametinger. Component-based software engineering: putting the pieces together, 2001.
[2] http://www.carre-project.eu/?wpfb_dl=608
[3] http://www.carre-project.eu/?wpfb_dl=645
[4] http://www.carre-project.eu/?wpfb_dl=708
[5] http://www.carre-project.eu/?wpfb_dl=762
[6] http://www.w3.org/RDF/
[7] http://www.carre-project.eu/?wpfb_dl=709

discussing how the CARRE objects and services (presented in Section 3) are accounted for in each component. Moreover, Section 4 discusses some issues concerning the security policies to be considered during the development of the various components. Section 5 presents a representative list of sequence diagrams in order to show how different user actions will be propagated into the lower layers of the architecture. Finally, Section 6 concludes with a few final comments and remarks.

2. Architecture High Level View

As already discussed, the CARRE architecture adopts a component-based approach. Figure 1 presents a high-level view of the overall architecture illustrating the major components identified and how these are connected to one another. A discussion with details concerning the design decisions is provided in the following sections.

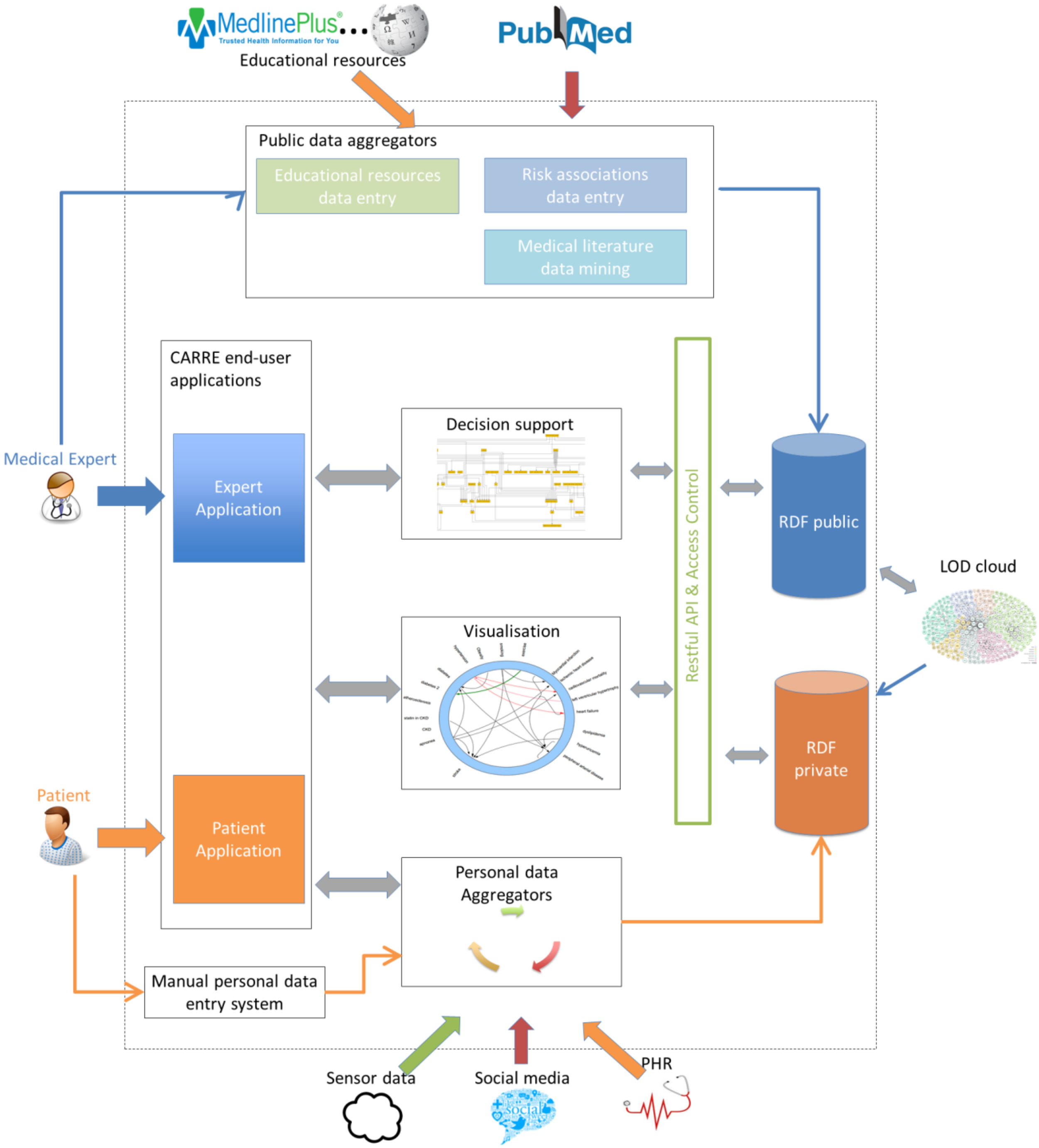

Figure 1: CARRE Architecture high-level view.

3. CARRE Objects and Services

This section outlines the data used during the different phases of CARRE information processing and the set of services that are going to be developed hereafter. The aim of this section is not to describe the above in full technical detail, but to draw a framework and a general overview of the anticipated CARRE system.

3.1. CARRE Objects

For ease of reading, we have categorised the data objects based on their visibility access rights into *public* and *personal* data.

Personal data include information acquired concerning a specific patient after the patient has given her consent. These data originate from heterogeneous data sources (external to CARRE) and introduce different types of information (stored within CARRE). The table below summarises the personal data anticipated in CARRE, along with a short discussion about their origin and purpose of usage. All of the harvested personal data are deposited onto the private RDF repository in semantically enriched form.

Apart from data referring to individuals, CARRE will capture, store and represent public knowledge regarding the medical domain. This information, which is by default public, together with the personal data will be processed by the middleware components and will be ultimately used by the end-user applications.

The tables below are organised in three different categories:

a) Data objects external to CARRE; this refers to all external data sources and the data they include. CARRE harvests these sources and aggregates certain parts of these external data.

b) CARRE data objects: these include all data as harvested from external sources and stored in the public and private CARRE RDF repositories.

c) Objects produced by CARRE: these refer to the output of the CARRE services.

3.1.1. Data Objects External to CARRE

Table 1 briefly describes types of data objects found in external 3rd party sources, which CARRE will harvest and utilise. A more detailed description of the external sources and their data has been given in CARRE D.2.3.

Table 1. Data objects external to CARRE.

personal/private	Sensor data	Data collected from personal sensors developed by 3rd party sensor devices manufacturers. These data include measurements on weight, blood pressure, physical activity etc. These data reside in the cloud, as provided by 3rd party providers. These data are collected, stored and managed under the responsibility of the patient. Access to such data is via secure channels and requires user (patient) authorisation.
	Medical data	Data on medical history as stored in 3rd party personal health record (PHR) systems. The patient is responsible to input and maintain these data using a 3rd party PHR system (examples include Vivaport, MyHealthAvatar and HealthVault) Access to these data is via secure channels and requires user (patient) authorization.
	Web social and lifestyle related data	These data include any interaction of the user with 3rd party web-based social systems, including the patient's web browser. Examples of such data are patient's browser history, patient's Google searches and any input in widely used social media, e.g. Twitter. Access to these data is via secure channels and requires user (patient) authorisation.

public	Medical literature	These data include scientific publications that are the commonly accepted sources of medical evidence. Normally these publications reside in the web repositories of individual scientific journals and can be reached via dedicated scientific indexing services available on the web, the main one considered here being PubMed. These data can be accessed via public APIs.
	Educational resources	This includes educational items on medical issues addressed to patients. Such data reside on 3rd party web repositories (e.g. MedlinePlus, Wikipedia) and can be accessed via public APIs.

3.1.2. CARRE Data Objects

Table 2 describes the CARRE data objects. These are derived directly from the CARRE external data objects via CARRE aggregators, and this process may involve a variable degree of processing (depending on the type of data, to be described in detail in D.3.2 and D.3.3).

Table 2. CARRE data objects.		
personal/private	CARRE Sensor data	These are data harvested from the cloud of 3rd party personal sensor providers via the CARRE sensor aggregators. These data are semantically enriched and stored into the private repository as CARRE RDF triples.
	Medical data	These are harvested from PHR systems via the corresponding CARRE aggregators. Only data relevant to CARRE are harvested, including demographic information about the individual, such as age, sex, medication taken etc. and information on medical history regarding disease status and incidents and drug administration. In the case that a patient does not use a PHR, some minimum personal medical data will be acquired via manual data entry. This will be achieved via a light patient data entry application, to be developed as part of T.3.3. These data are semantically enriched and stored into the private repository as RDF triples.
	Web social and lifestyle related data	This refers to information on intention and lifestyle as deduced by data harvested from social media accounts or other forms of web presence of the individuals (e.g. Google searches). These data are semantically enriched and stored into the private repository as RDF triples.
public	Risk factor descriptions	These are deduced directly from medical literature. The detailed description of the information comprising a risk factor description is given in CARRE D.2.2. These descriptions are manually entered by the medical expert, based on the medical expert's understanding of the medical literature. The process is also assisted by semantic data mining of medical literature (component developed in T.3.4). These data are semantically enriched and stored into the private repository as RDF triples.
	Educational resources	This includes metadata on educational resources as harvested from external educational resources repositories. The external harvested metadata is complemented with a rating of the educational resource by the medical expert. These data are semantically enriched and stored into the public repository as RDF triples.

3.1.3. Objects produced by CARRE

Table 3 describes the CARRE data objects. These are produced by the CARRE services and are presented to the end-users.

Table 3. Objects produced by CARRE.		
personal/private	Personalised visualisation of risk factor model	This refers to the output of the visualisation service. It includes an interactive graphical user interface and provides the set of pictograms (graphics), text and other visual components One of the features of all visualisation components are scalability and adaptability in order to allow easy configuration of presented elements (interface) on tablets or standard monitor display.
	Personalised alerts	This is a decision support service (DSS) object, which is the result of analysis and reasoning captured in CARRE ontologies. A basic feature of DSS is the timely identification of new health risks for the patient and creation of the respective alerts. Hence the DSS will provide the variables for the visualisation object on how to present a given piece of information using various techniques – for example presentation of comorbidity risk level. DSS will be used to inform patient about medical check-ups, monitoring, increased risk of disease progression and transition, the need to change diet etc.
public	Generic visualisation of risk factor model	This object illustrates graphically the CARRE comorbidity model. It is accessible by both authenticated and guest users and can be used to present an overview (e.g. including all risk elements) or present a detailed risk association with all relevant data. Functions of this object can be shared with those of the visualisation object in the personal/private data.
	Display of (links to) relevant educational resources	This data object refers to information indexed and provided through linking mechanisms by CARRE. The targets of the above links are web-based repositories with educational data. The links are provided complementarily to medical terminology in order to assist end-users to understand the CARRE-produced knowledge.

3.2. CARRE services

This sub-section aims to present the set of CARRE core services, which will consume and/or produce the CARRE object (Table 4). The aim here is to provide a complementary view of the overall CARRE architecture. Hence, CARRE objects together with CARRE services provide an overview of the operations that run across the whole system.

Table 4: CARRE core services.	
Service	Description
Risk description entry system	Service and respective user application for entering risk factor descriptions (risk elements, evidence etc.). Data entry is via a browser based application and the service produces automatically the appropriate RDF triples, to be stored in the public repository.
Risk association data mining	This service provides data mining methods to discover relevant risk associations in published medical evidence literature, as indexed in PubMed. The service presents results to the medical expert who is thus assisted into entering new (or updating existing) risk factor descriptions into the system.
Educational resources aggregation and annotation system	This service allows the annotation of educational resources by medical experts. This leads to the generation of appropriate RDF triples of educational resource metadata, to be stored into the public RDF repository.
Sensor data aggregation	This includes a collection of services that support the personal sensor data aggregation, such as: authentication with 3rd party data sources, pulling data from manufacturers' data cloud, subscription to pub/sub services, and transformation of

	data into RDF.
PHR data aggregation	This mechanism implements the collection of relevant health information from personal health record systems, namely demographic data, diagnoses, and medication summary). The service allows automatic harvesting from an existing PHR or, alternatively, manual data entry by the patient (if a PHR is not used by the patient). Additionally, the service transforms acquired data into the appropriate RDF triples to be stored into the private RDF repository.
Personal medical data manual entry system	This refers to the browser-based application that will allow end-users to manually enter personal health information. It is meant to be used complementary to the PHR data aggregation service (described above) in cases where a PHR system is not used by the patient.
Personal lifestyle data aggregation	This service provides patient lifestyle information as deduced from information generated by the patient's in web presence (e.g. Google searches, Twitter). The service also transforms information into the appropriate RDF triples to be stored in the private RDF repository.
Semantic enrichment	This service is part of data aggregation and concerns both public and personal data. It aims at automatically transforming data from different formats (JSON, CSV, RDF using different vocabulary) into CARRE RDF statements, conforming with the CARRE ontology already developed.
Content delivery for CARRE visualisation and decision support	This concerns the layer placed on top of the CARRE RDF repositories and serves the requests coming from the middleware components. It is comprised of a set of RESTful methods to serve CARRE-related requests. Typically, these requests are encoded in a web-friendly format such as JSON.
Access control & personalised content delivery	Together with the RESTful API, an access control mechanism is deployed on top of the RDF repositories to ensure the secure delivery of data to and from the middleware components. Furthermore, this access control service will be responsible for the personalisation of the delivered content, i.e. an end-user request sent to/from the application space will be processed accordingly here.
Personal data analytics	This service analyses all personal data collected through APIs, compares these with generic medical evidence data on risk factors and creates the personalised risk model for each patient.
Decision Support	This service assesses personal information for events and data that may create alerts for treatment guidance, patient education, diet and monitoring adjustments. The service also assesses public data for respective alerts to the medical expert (as described in WP6).
Comorbidity model visualisation	This service creates the visualisation of the generic and personalised risk model and also integrates the visual output of the decision support service.

4. Architectural Components

In this section we describe each of the components shown in Figure 1, and introduced in Sections 2 and 3. The aim here is to provide an overview of the purpose of each component, interacts communicate with the other components and give the high level approach for its implementation and testing. Detailed technical specifications for each component will be provided in the respective project deliverable describing the development of each component.

4.1. Detailed description of CARRE architectural components

4.1.1. Personal data aggregators

Sensor data aggregators

This is the set of components responsible for linking sensors to the CARRE system (Table 5). This linking includes acquiring authorisation from the end-users to connect to the sensor data storage (most cases the cloud) and import sensor data. After acquiring this authorisation, the aggregators fetch historical data, subscribe to pub/sub[8] notifications (monitoring mode) and fetch in real-time (where available) sensor data coming from the sensor data cloud. Finally, the aggregators are responsible for transforming this data into the appropriate RDF triples, conforming to the CARRE ontology (see D.2.4).

Table 5. Component description: sensor data aggregators.

Sensor data aggregators	
Main functionality	Retrieve sensor data and deposit it onto the private RDF repository
Partners	KTU, OU
Input data format	variable, specific to sensor
Output data format	RDF
Input communication protocol	variable, specific to vendor
Output communication protocol	SPARQL
Input from	sensor data cloud
Output to	RDF private repository
Development technologies	OAuth, Java, Python
Prerequisites & Dependencies	Sensors, portable device (tablet or smartphone or equivalent), RDF private repository
How and who will test	Test complete chain: subscription/unsubscription to services, sensor data

[8] In software architecture, publish–subscribe is a messaging pattern where senders of messages, called publishers, do not program the messages to be sent directly to specific receivers, called subscribers. Instead, published messages are characterized into classes, without knowledge of what, if any, subscribers there may be. Similarly, subscribers express interest in one or more classes, and only receive messages that are of interest, without knowledge of what, if any, publishers there are.

	acquisition, data fetch from vendors' data clouds, data integrity verification and preprocessing in CARRE system outliers detection/mitigation, missing data detection/filing. Responsible KTU and OU (real data is supplied from partners via test volunteers).

Personal Health Record aggregators

Patient Health Record aggregators are responsible to acquire permission from the users to access 3rd -party PHR systems (that provide an API for accessing their records). Having acquired this authorisation, the aggregators implement mechanisms that fetch relevant information from personal records. The data are then turned into RDF and deposited into the private RDF repository.

Table 6. Component description: personal health record aggregators.

Personal Health Record aggregators	
Main functionality	To insert patient health information and support the DSS and visualisation services
Partners	VULSK, OU, DUTH, BED
Input data format	variable, specific to PHR
Output data format	RDF
Input communication protocol	variable, specific to PHR
Output communication protocol	SPARQL
Input from	PHR aggregator/system (i.e. MyHealthAvatar/Vivaport)
Output to	private RDF repository
Development technologies	.NET
Prerequisites & Dependencies	PHR vocabulary, Private RDF repository
How and who will test	Test data from test volunteers, preferably by the same people using the sensors – each person should create a record in each system to be integrated.

Personal medical data manual entry system

The purpose of the manual entry system is to acquire personal health information when a PHR is not used by the patient (Table 7). This subsystem provides an end-user interface. Primarily, it is meant to be used complementary to the PHR data aggregation component (described above) in cases where PHR data is incomplete or not available. Acquired data are turned into RDF and stored in the private RDF repository

Table 7. Component description: personal medical data entry system.

Personal medical data manual entry system	
Main functionality	To provide an interface for patients to enter manually information concerning their health and medical care received.

Partners	VULSK
Input data format	User input
Output data format	RDF
Input communication protocol	Human Computer Interaction input
Output communication protocol	HTTP + SPARQL
Input from	Patient - end user
Output to	Private RDF repository
Development technologies	.NET
Prerequisites & Dependencies	PHR scheme, Private RDF repository
How and who will test	VULSK, BED

Web lifestyle data aggregators

This set of aggregators (Table 8) aim to collect information on user intention, mainly on the following topics: patients' inclination to search for health related information on the web and what types of health related information (which can better shaper the educational material brought to them by CARRE); patients' intention to travel (which may affect their diet and physical exercise and thus require changes in diet, monitoring etc.); and patients' searches on diet and food (which can better shape guidelines and educational material brought to them on diet and food preparation). The aggregators record information on web searches and patient tweets (if available) and explore proven data mining tools to extract patient's intention. This intention information is then stored in the CARRE private RDF store.

Table 8. Component description: web lifestyle data aggregators.

Web lifestyle data aggregators	
Main functionality	Capture lifestyle, extract patient/user intension information and store it into the private RDF repository.
Partners	DUTH, OU, BED
Input data format	Variable, depending on lifestyle provider
Output data format	RDF
Input communication protocol	Variable, depends on social media provider
Output communication protocol	SPARQL
Input from	Google searches, browsing history, Twitter
Output to	Private RDF repository
Development technologies	Java, Lemur[9], Weka[10], WordNet[11]

[9] http://www.lemurproject.org/

[10] http://www.cs.waikato.ac.nz/ml/weka/

Prerequisites & Dependencies	Portable device (tablet or smartphone or equivalent) RDF private repository
How and who will test	Test data from datasets or volunteers (DUTH, OU) Evaluation of extracted patient intention by using datasets or volunteers (DUTH, OU)

4.1.2. Public data aggregators

Risk associations data entry

This is a subsystem that provides an end-user interface for the medical experts (Table 9). The purpose of this system is to allow the insertion of risk associations into CARRE. The insertion includes the insertion of risk elements, evidence sources and observables as prescribed in D.2.4. A detailed presentation of this subsystem is included in the aforementioned deliverable.

Table 9. Component description: risk associations data entry.

Risk associations data entry	
Main functionality	Provide an interface for inputting risk associations. Semantically enrich the input and transform it to RDF Deposit the content onto the public RDF repository
Partners	OU, DUTH
Input data format	User input
Output data format	RDF
Input communication protocol	Human Computer Interaction input
Output communication protocol	SPARQL
Input from	User input
Output to	Public RDF repository
Development technologies	Content Management System enhanced with Semantic Web capabilities
Prerequisites & Dependencies	RDF public repository CARRE risk associations ontology
How and who will test	OU, DUTH and VULSK to troubleshoot and ingest content coming from D.2.2.

Medical literature data mining

This component acts as another data aggregator (Table 10). Its purpose is to gather medical knowledge with the aim to 1) enrich the evidences of the existing risk descriptions as entered manually by medical experts and 2) identify new risk associations for cardiorenal diseases and comorbidity as published in medical literature during and beyond the project's lifetime. This aggregator extracts and summarises key information from popular and trusted medical publications as they are indexed in PubMed. It harvests the necessary

[11] http://wordnet.princeton.edu/

metadata describing each literature item, and pushes such semantic descriptions to the CARRE public RDF repository. The function will ultimately extend the CARRE knowledge base. The actual data (i.e. the scientific paper) will not be retrieved, but only indexed and described along with information (link) for its retrieval. Various data mining tools (such as GATE) and ontology (such as MeSH ontology and ICD) are consulted for medical domain knowledge.

A user interface will be provided for the medical expert to test and validate retrieved literature and proposed risk associations.

Table 10. Component description: medical literature data mining.

Medical literature data mining	
Main functionality	Mining newly published risk associations and storing them into public RDF repository
Partners	BED, OU, VULSK
Input data format	User input
Output data format	RDF
Input communication protocol	HCI input
Output communication protocol	SPARQL over HTTP
Input from	User input
Output to	Public RDF repository (validated by medical experts)
Development technologies	Java, GATE[12]
Prerequisites & Dependencies	public RDF repository, CARRE ontology, Risk association entry system, public RDF repository API
How and who will test	Medical experts, VULSK and DUTH to test and validate the newly found risk associations

Educational resources aggregator and rating system

This module aggregates metadata on educational resources available in 3rd party repositories on the web (Table 11). Data aggregation is based on consuming existing APIs of educational resource repositories. The aggregator is triggered by new terms and concepts pertaining to risk factor descriptions (and their semantic enrichments) in the CARRE public RDF repository. Using such terms relevant educational resources are retrieved. Their metadata can be stored as RDF triples in the CARRE public repository. Alternatively, educational resources are shown to medical experts for expert rating. Rating information is then stored along with the educational resource metadata. This aggregator is also periodically triggered to query external educational resources for new content.

Table 11. Component description: educational resources aggregator and rating system.

Educational resources aggregator and rating system	
Main functionality	Aggregate educational resources metadata and allow rating by medical experts.

[12] https://gate.ac.uk/

Partners	DUTH, VULSK, OU, BED
Input data format	Variable, depends on social media provider
Output data format	RDF
Input communication protocol	Variable, depends on social media provider
Output communication protocol	SPARQL
Input from	Educational resources repositories (i.e. MedLinePlus and Wikipedia)
Output to	Public RDF repository
Development technologies	JavaScript client/server
Prerequisites & Dependencies	External educational resources, Public RDF repository
How and who will test	Test theme coverage - test if all currently available conditions and observables correspond to at least one educational content item

4.1.3. RDF Repository & Middleware

Public RDF repository

This component is part of the RDF repository, implemented using Virtuoso[13] and configured accordingly to provide public access (Table 12). The data hosted in this repository originate from the public data aggregators and conform to the CARRE ontology. The CARRE ontology is designed so as to link the CARRE vocabulary to external, well-known vocabularies. Hence external developers may access this repository directly and develop their own applications. Apart from its public use, the RDF public repository is connected with the RESTful API in order to pass CARRE public data to the middleware components of the CARRE architecture.

Table 12. Component description: public RDF repository.	
Public RDF repository	
Main functionality	Store data on risk associations and related concepts. Metadata on educational resources. Data on decision making (WP6).
Partners	OU
Input data format	RDF
Output data format	RDF
Input communication protocol	SPARQL
Output communication protocol	SPARQL
Input from	Public data aggregators

[13] http://virtuoso.openlinksw.com

Output to	RESTFul & access control, third-party developers Linked Open Data Private RDF repository
Development technologies	Virtuoso
Prerequisites & Dependencies	CARRE ontology Public data aggregators
How and who will test	Data entry by medical experts (DUTH, VULSK) Queries by BED and PIAP Ontology (clinical experts)

Private RDF repository

This is part of the RDF repository responsible for hosting personal data (Table 13). It implements a set of secure mechanisms for ensuring the privacy of data originating from the private data aggregators. The repository is connected to the RESTful API component in order to pass personal data information to the middleware components of the CARRE architecture. Both private and public RDF repositories are developed using Virtuoso[14].

Table 13. Component description: private RDF repository.	
RDF private repository	
Main functionality	Store private patient data (sensor data, input from PHR or manual patient data entry, lifestyle
Partners	OU
Input data format	RDF
Output data format	RDF
Input communication protocol	SPARQL (authenticated[15])
Output communication protocol	RDF
Input from	Personal data aggregators
Output to	RESTful API & access control
Development technologies	Virtuoso
Prerequisites & Dependencies	CARRE ontology Personal data aggregators
How and who will test	Sensor data entry by KTU, OU PHR data entry by VULSK Lifestyle data entry aggregator by DUTH Queries by BED, PIAP

[14] http://virtuoso.openlinksw.com/

[15] By, e.g., HTTP Basic Authentication over HTTPS, or OAuth.

RESTful API & Access Control

This is a middleware component sitting on top of the CARRE RDF repositories and responsible for lifting the data in a web-accessible format (Table 14). The component is accessing the RDF repositories through SPARQL over HTTP queries and provides JSON encoded CARRE objects. To achieve this, a set of web methods is provided that describe the basic CARRE services and objects. These methods and objects are consumed by the middleware components. Requests for personal data are sent through secure and authenticated channels. Finally, third-party developers may use this component either partially or as a whole.

Table 14. Component description: RESTful API and access control.

RESTful API & Access Control	
Main functionality	To provide a layer of CARRE-related RESTful services over the RDF repositories.
Partners	OU
Input data format	JSON requests, translates them to SPARQL queries that are passed to the RDF repositories
Output data format	Output to CARRE middleware services is a JSON response, output to the RDF repositories is SPARQL requests.
Input communication protocol	HTTP
Output communication protocol	HTTP
Input from	Requests originate from the middleware components, Responses returned originate from the CARRE RDF repositories.
Output to	Visualisation, DSS, personalisation, end-user applications
Development technologies	Python, RDF, SPARQL
Prerequisites & Dependencies	User database, data populated in CARRE RDF repositories, CARRE RDF repositories
How and who will test	Early stage of testing by OU. Testing will also occur during the development of middleware components by BED and PIAP.

4.1.4. CARRE end-user services and applications

Visualisation service

The visualisation service is a data-driven module which provides web-based interactive visual presentation of data stored in the public and private data repository, such as risk associations - generic and personalised (Table 15). The decision support service (DSS) can be another data source which provides real-time decision support data for patients. The data will be retrieved via RESTful web service APIs provided by the public and private RDF repository. Secure APIs may be used to provide encrypted communication, such as in the case of retrieving data from the private repository. The interactive visualisation client-side code will be downloaded online from the server side and run in web browsers in the Patient Application and the Medical Expert Application.

Various interactive visual analysis and information visualisation techniques will be investigated and exploited in the visualisation service component. To reduce both the data and visualisation complexity, interactive data analysis techniques will be employed in the visualisation service component to support client-side data filtering and clustering. Efficient user interactions will also be designed to promote the visualisation service for fast and effective insight gaining. To summarise, in the CARRE architecture, the visualisation component is used by the Patient Application and the Medical Expert Application and communicates with the public/private RDF repository and the DSS to provide integrated interactive tools for visual analysis.

Table 15. Component description: visualisation service.

Visualisation service	
Main functionality	Visualise generic model and personal model
Partners	BED, PIAP
Input data format	JSON (RESTful responses with data coming from RDF repositories or other data sources)
Output data format	Web-based SVG or image plot
Input communication protocol	HTTP/HTTPS (PUT, GET, POST, DELETE)
Output communication protocol	HTTP/HTTPS
Input from	RDF public and RDF private repository APIs or other data source APIs
Output to	Web browser based output to Android mobile devices or PC displays
Development technologies	Interactive network visualisation and visual analytics technologies. Web-based interactive plotting library in JavaScript.
Prerequisites & Dependencies	Data available in CARRE RDF Repository or other data sources, Decision support component, risk assessment, risk association and other variables
How and who will test	Testers: developer (BED), medical experts (DUTH, VULSK) and other related partners Test method: testing the service with the data set according to test requirements.

Decision support service

The purpose of the decision support services (DSS) component is to allow ontology-based reasoning for CARRE end-users (Table 16). This includes the analysis of the generic and personalised risk model so as to allow the CARRE stakeholders to identify and assess critical medical conditions. Its aim will be to produce meaningful information that will be passed to the end-user interface with the synergy of the visualisation component. The DSS component will consume the web service provided by the CARRE RESTful API and take into account the user input that is forwarded into it.

Table 16. Component description: decision support service.

Decision support service	
Main functionality	Decision support service will provide personalised links to educational resources, and a variety of health related alerts including alerts for monitoring, diet adherence, management and change.
Partners	PIAP, VULSK, DUTH

Input data format	RESTful response
Output data format	text and variables send in for example JSON to Visualisation service
Input communication protocol	HTTP
Output communication protocol	HTTP
Input from	RESTful API & access control
Output to	Visualisation service
Development technologies	to be decided in WP6
Prerequisites & Dependencies	RDF repositories populated with data, RDF repositories, personal and public data aggregators
How and who will test	PIAP, VULSK, DUTH, BED

Patient application

The patient application is the integration of all components that involve the patient as the end user (Table 17). This application thus includes the following: (a) authentication; (b) functionality that allows the patient to register personal sensors, personal health record systems and web lifestyle systems (browser history, Twitter); (c) manual medical data entry system; (d) output of visualization service; (e) output of decision support service.

Table 17. Component description: patient application.

Patient application	
Main functionality	Domain experts provide personalised decision supports on disease progression that will be implemented as DSS components and will be presented to the patient through user-friendly visualisation.
Partners	DUTH, PIAP, VULSK, BED, OU
Input data format	DSS, Visualisation, Personalisation
Output data format	Web based application
Input communication protocol	HTTP, API
Output communication protocol	HTTP
Input from	DSS, Visualisation, RDF Repositories
Output to	Patient
Development technologies	to be decided in WP3, WP5 and WP6
Prerequisites & Dependencies	RDF repositories populated with data, working Visualisation model, RDF repositories, personal and data aggregators, Visualisation, DSS,Manual personal data entry system
How and who will test	ALL

Medical expert application

The medical expert application is the integration of all components that involve the medical expert as the end user. This application thus includes the following: (a) authentication service; (b) risk factor data entry system, including the interaction with the medical literature mining service; (c) educational resource rating system; (d) output of the visualization service; and (e) output of the decision support service.

Table 18. Component description: medical expert application.

Medical expert application	
Main functionality	Domain experts provide medical decision support on disease progression that will be implemented as DSS components and will be presented to the medical experts through user-friendly visualisation.
Partners	VULSK, PIAP, DUTH, BED
Input data format	DSS, Visualisation
Output data format	Web based application
Input communication protocol	HTTP, API
Output communication protocol	HTTP
Input from	DSS, Visualisation, RDF Repositories
Output to	Medical experts
Development technologies	to be decided in WP3, WP5, and WP6
Prerequisites & Dependencies	RDF repositories populated with data, working Visualisation service and DSS, RDF repositories, public data aggregators
How and who will test	DUTH, VULSK

4.2. Security issues

Security is a cross concern of all CARRE architectural layers and services. Ideally the security solution will be centralised services with reusable modularized components that handle authentication, authorisation and audit. In order to interact with private information from CARRE platform, user must be authenticated and authorized. More specifically, the authentication will be ideally utilizing a centralised user store and SSO (single sign on) mechanism.

CARRE architecture is designed as separate modules/services, which interact with each other, and each components and services will take security into account. Services would verify requests from not only outside world, but also from each other. For example, RDF repository will only serve the information to query if proper security token is provided with the request; which means other services like DSS will need to obtain proper user authorisations to retrieve data from RDF repository. This defensive architecture will make sure that malicious user will not be able to penetrate CARRE platform through one single service component.

In current iteration, state-of-the-art industrial standards and general security guidelines have been studied and will be referenced in the following implementation iterations. Relevant standards should be followed by respective services, and the best practices in security for CARRE architecture are suggested as follows:

Basic general security recommendations

– Back up often and keep backups physically secure

- Keep the server physically secure, only give access to authorized users
- Secure the server (and other computer on same network) with strong passwords
- Always install latest security updates to operating system and server applications
- Close unused ports and turn off unused services
- Monitor the server CPU load/traffic, etc.
- Use firewall

Run application with least privileges

- Run the web application by user with minimum practical privilege and no shell access
- Do not run the application within the context of a system user (root)
- Keep files for the web application in limited locations and prevent user from access arbitrary path of server.

Guard against malicious user input

- Never assume the input from user is safe
- Filter user input to check HTML tags, which might contain script
- Never display unfiltered user input, always HTML encode before display
- If HTML input from user is required, explicitly define whitelist for them. Do not attempt to filter out all malicious input
- Do not assume information get from DNS query or headers is safe. Use safeguards for query strings, cookies, headers and so on
- Do not store sensitive information accessible to browsers or their debugging tools

Access databases securely

- Do not expose database to public internet if possible
- Give minimum permission to specific database user/role for a given web application
- Do not create SQL statements by concatenating strings, use query parameters instead
- Store user password using strong hash algorithm, e.g. Bcypt. Do not use plain hash (MD5/SHA1, etc) or even a salted version
- Run database in separate box/virtual machine

Keep Sensitive Information Safe

- Transfer sensitive information through TLS (SSL 3.0 is not safe due to POODLE[16])
- Do not create own encryption algorithm, use existing proofed one
- Do not write error message which contains information which might be useful to malicious user (e.g. user information)
- Do not display detailed error message to end users
- Do not store any sensitive information in cookies
- Have log/audit on any operation which access sensitive information

Guard against popular vulnerabilities

- Cross Site Request Forgery (CSRF) Protection
- Subscribe to relevant security notice board
- Click Jacking Protection

---

16 https://www.us-cert.gov/ncas/alerts/TA14-290A, accessed on 14/11/2014

- Use two step authentication[17]
- Use Security Headers[18]
    - Cache Control (Cache-Control: no-cache, no-store, max-age=0, must-revalidate Pragma: no-cache Expires: 0)
    - Content Type Options (X-Content-Type-Options: nosniff)
    - HTTP Strict Transport Security (Strict-Transport-Security: max-age=31536000; includeSubDomains)
    - X-Frame Options (X-Frame-Options: DENY)
    - X-XSS-Protection (X-XSS-Protection: 1; mode=block)

[17] http://lifehacker.com/5938565/heres-everywhere-you-should-enable-two-factor-authentication-right-now, accessed on 14/11/2014

[18] http://docs.spring.io/autorepo/docs/spring-security/3.2.5.RELEASE/reference/htmlsingle/, accessed on 14/11/2014

5. CARRE from the End User Perspective: Sequence Diagrams

This section discusses how the components will be put into use from an end-user point of view, via a representative collection of UML sequence[19] diagrams. The aim of these is to illustrate different cases of user actions and how the different components will be executed.

5.1. Patient inserts personal health information manually

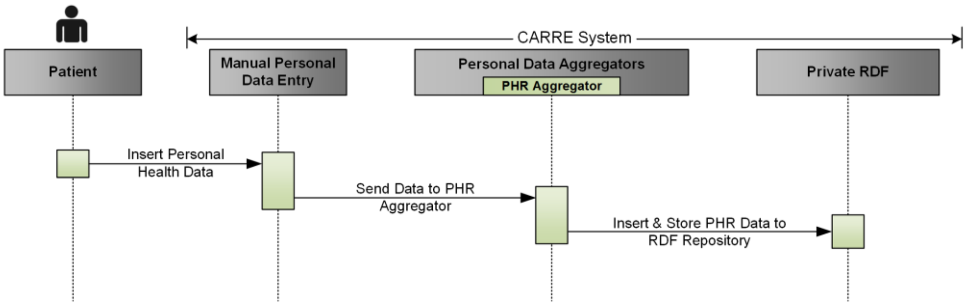

Figure 2: Patient inserts personal health information manually.

Figure 2 illustrates the case where a patient is using the Manual Personal Data Entry component to insert personal health information (all communication is taking place via a secure channel). The steps are:

- After authentication, the Patient accesses the interface provided by the Manual Personal Data Entry and fills one or more forms to submit information concerning her health.
- The Manual Personal Data Entry component sends the received information to the PHR Aggregator (part of the Personal Data Aggregators).
- The PHR Aggregator transforms the received data into RDF triples and stores them in the Private RDF Repository.

5.2. Personal data aggregation

Figure 3 describes two different phases of a patient managing her personal data. The first phase describes the steps that take place for different cases, such as registering a sensor device, linking her lifestyle-based web stream or uploading her PHR within CARRE. Initially, the patient accesses the web service (or interface) provided by the corresponding personal data aggregator. For all different cases, the user is prompted to grant access to the CARRE application for accessing her personal data. For the case of sensor data, the diagram shows how to make use of security mechanisms that implement secure delegated access, such as OAuth[20]. For all cases, the final step ends with the personal aggregator storing the acquired access token onto the private RDF repository.

The second phase describes the case where a periodical process takes place with the aim of keeping up-to-date personal data in the private repository. The diagram shows that the CARRE aggregators are responsible for handling the communication between the third-party data sources and CARRE. In all cases (sensors, PHR, lifestyle data), the aggregator ends up depositing the appropriate RDF onto the private RDF repository. If a pub/sub service is provided by the third-party data source (e.g. in cases of sensor data), a notification will initiate the process of fetching new personal data.

[19] http://www.uml.org/

[20] http://oauth.net/

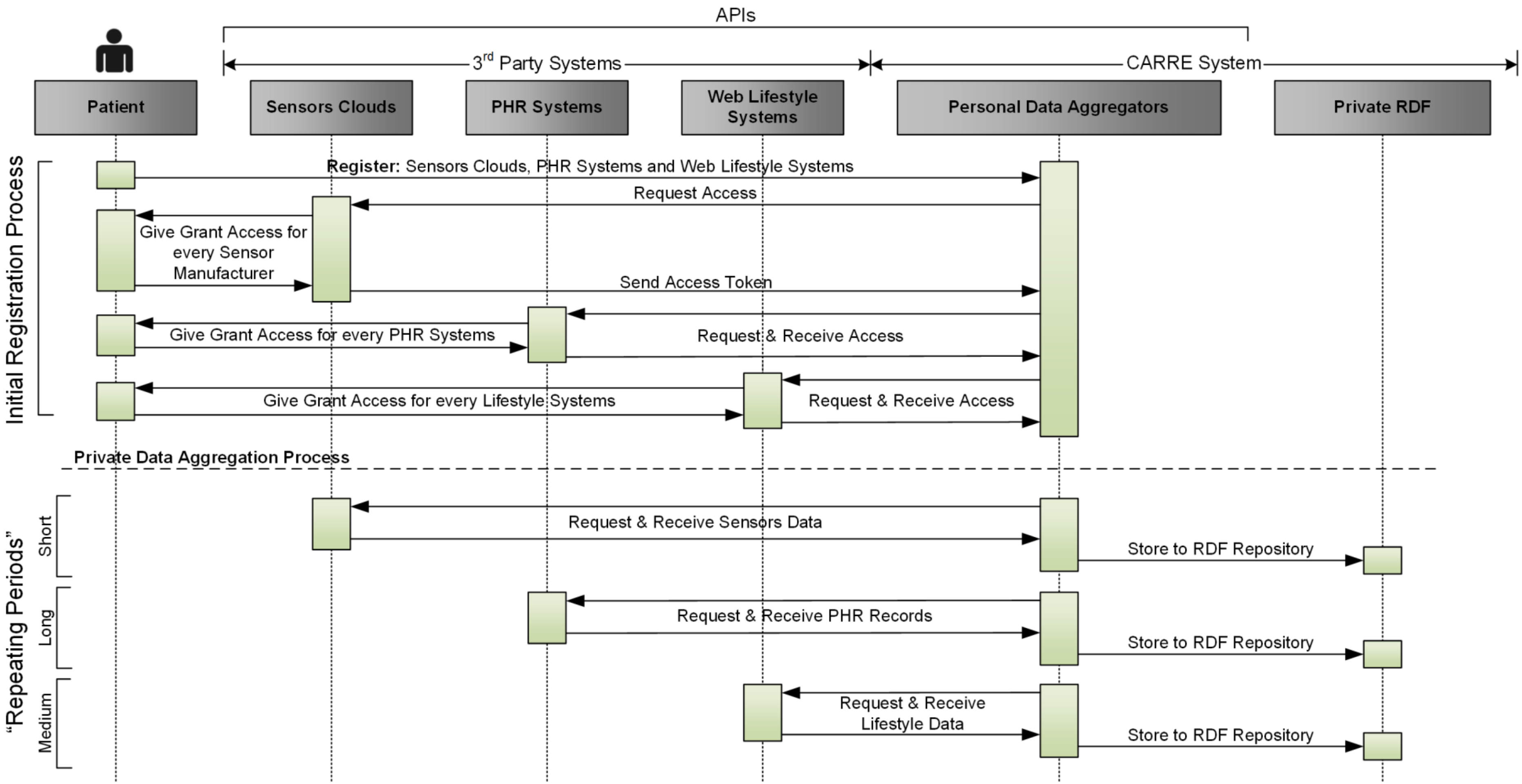

Figure 3: Patient linking personal data sources to CARRE and sensor data stored in the repository.

5.3. Public data aggregation

Figure 4 describes how public data (medical evidence and educational material) are aggregated from external sources with the input of medical experts. Similarly to the previous scenario, Figure 4 describes the generation and maintenance of all types of public data (i.e. risk associations, educational resources and medical literature data mining). The case comprises of two different phases.

The first phase illustrates the steps for inserting new data manually by a medical expert. Initially, the expert visits the web interface of the "risk associations data entry" and inserts information required to describe a new risk association. During this process, a service for searching in medical literature (third-party component) will support her in finding and selecting all relevant evidence source. Upon the insertion of the risk association, the data entry component automatically enriches the acquired information and turns it into appropriate RDF triples that are then stored in the public RDF repository. In parallel to this, the data entry component invokes the educational resources data entry component, informing it about concepts related to the risk association recently added. The educational resources component is then accessing third-party educational resources and searches for relevant information. The information retrieved is then passed back to the risk associations data entry component and prompts the medical expert to assess the presented resources. Finally, the feedback received from the medical expert is stored into the RDF repository.

The second phase describes the periodic and automatic case where the medical literature data mining tool searches into the medical literature (which is a component external to CARRE architecture). The result of this periodical search may return newly discovered evidence sources. These results are passed through the "risk associations data entry" to the user asking for her assessment. In cases where new risk association information is inserted or updated, the previous phase (described above) is repeated so as to supervise and ensure that educational resources remain valid and up-to-date. Finally, a similar procedure is triggered automatically by the educational resources data entry component with respect to educational resources. The only difference is that updating educational resources record does not trigger any investigation into the medical literature.

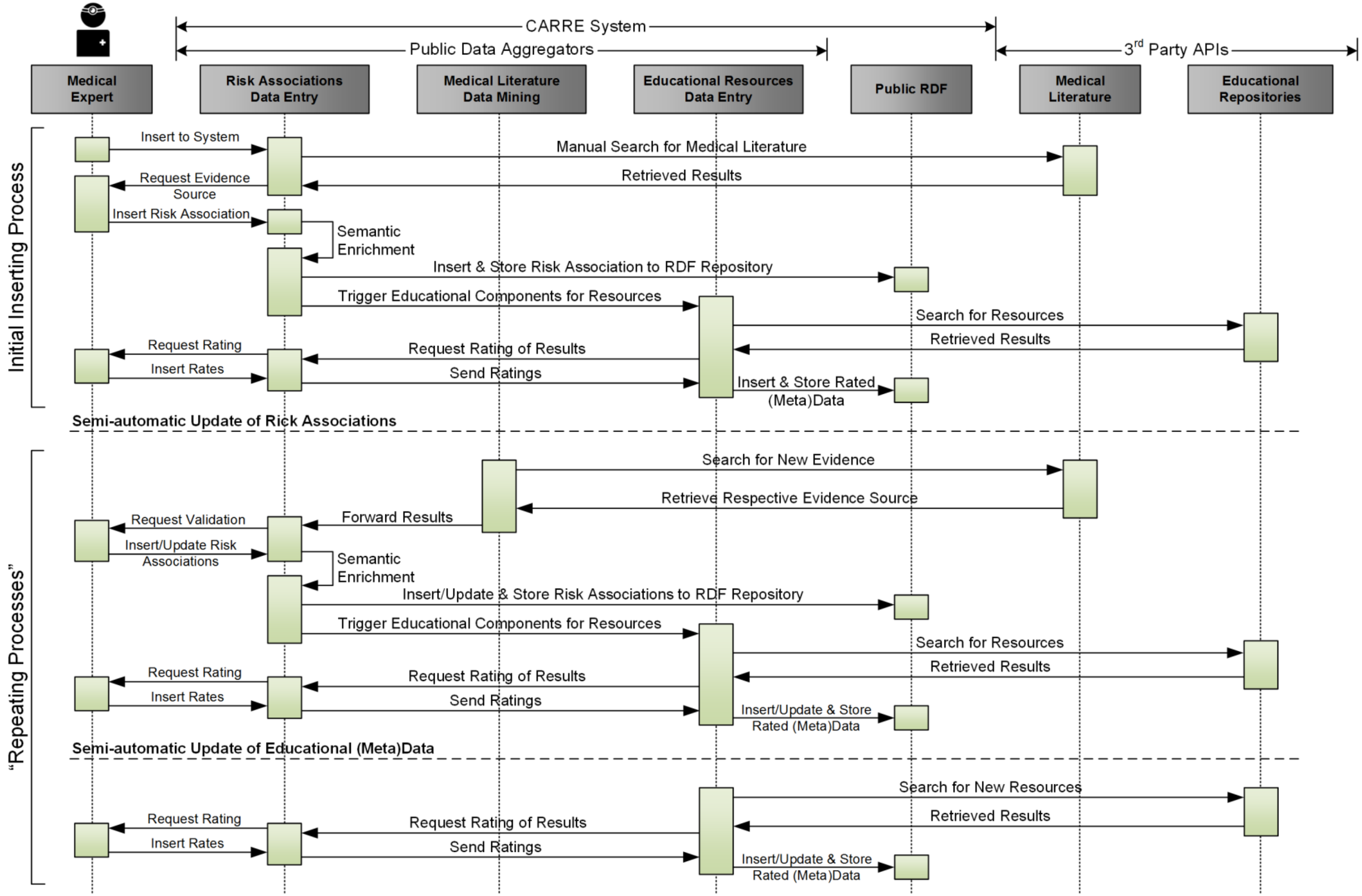

Figure 4: Medical expert inserting and updating CARRE public data.

5.4. Public access of risk associations

Figure 5 describes how users (both patients and medical experts) may access the generic risk associations model. The sequence starts with the user visiting the CARRE web interface and requesting the generic visualisation. This request is passed to the visualisation component. The visualisation component requests the data to build this model from the public RDF repository via the CARRE RESTful API. The construction of the model takes place within the visualisation component (analytics) and is then passed to its graphic engine for generating the desired graphic representation. Afterwards, this representation is forwarded to the end-user web interface. As an optional step, the user may pass parameters for investigating either a virtual medical scenario or for simply changing the visualisation layout. The user request is passed to the visualisation component which will generate a new graphic representation. Depending on the nature of the request, this last step is executed with the synergy of the CARRE decision support component.

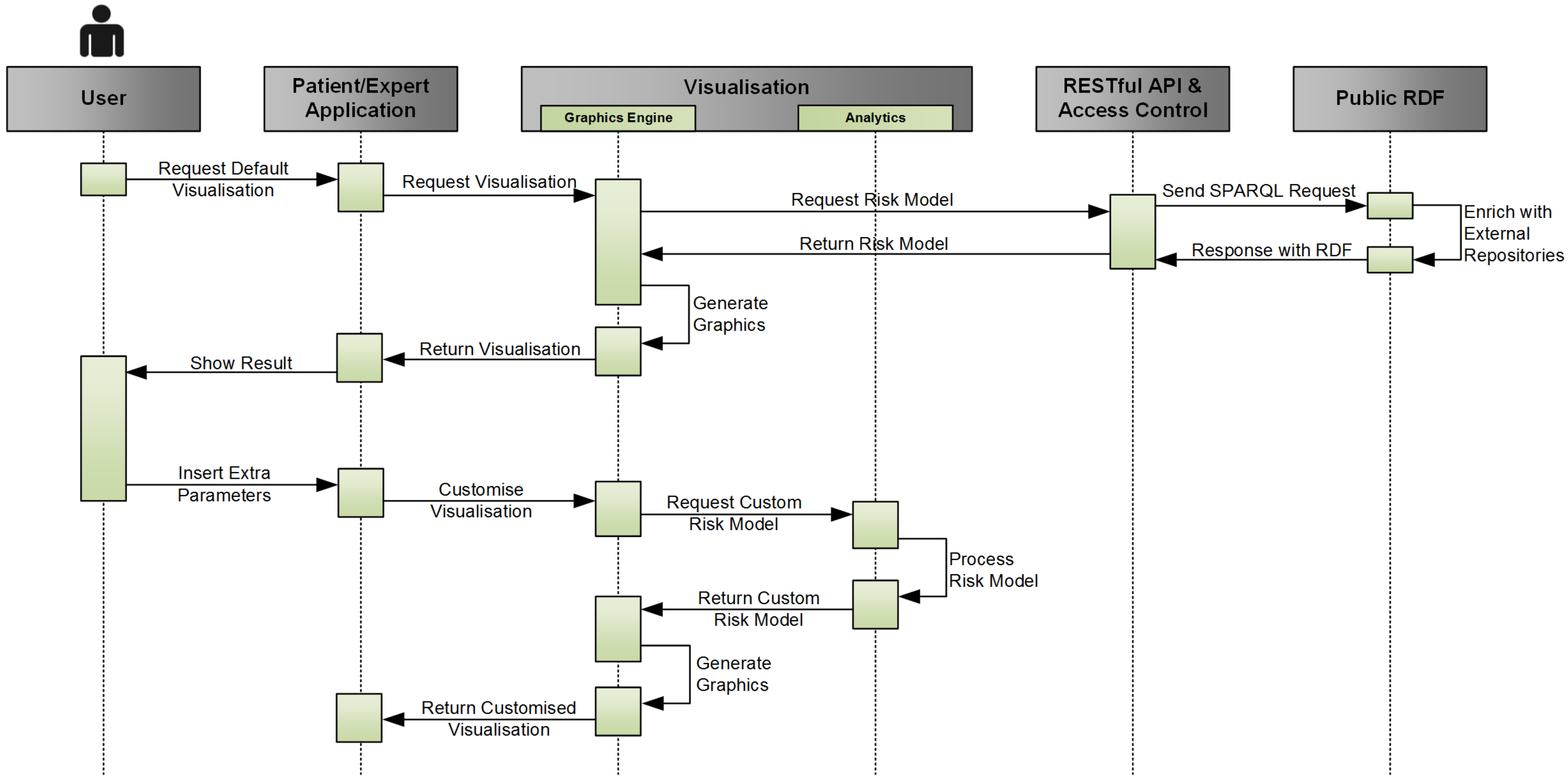

Figure 5: User accessing the CARRE risk associations model.

5.5. Personalised visualisation of patient's risk model

Figure 6 describes the case where a patient accesses her personalised risk model. The first step starts with the patient application forwarding a request for the appropriate visualisation from the visualisation component. The visualisation component then makes a request to the analytics subcomponent for the personalised risk associations model. The personalised model is using as input data from both the public RDF (i.e. risk associations and metadata) and the private RDF (i.e. personal data). This data is accessed via the RESTful API. The generated personalised models are passed back to the graphics engine which in turn displays it accordingly to the patient.

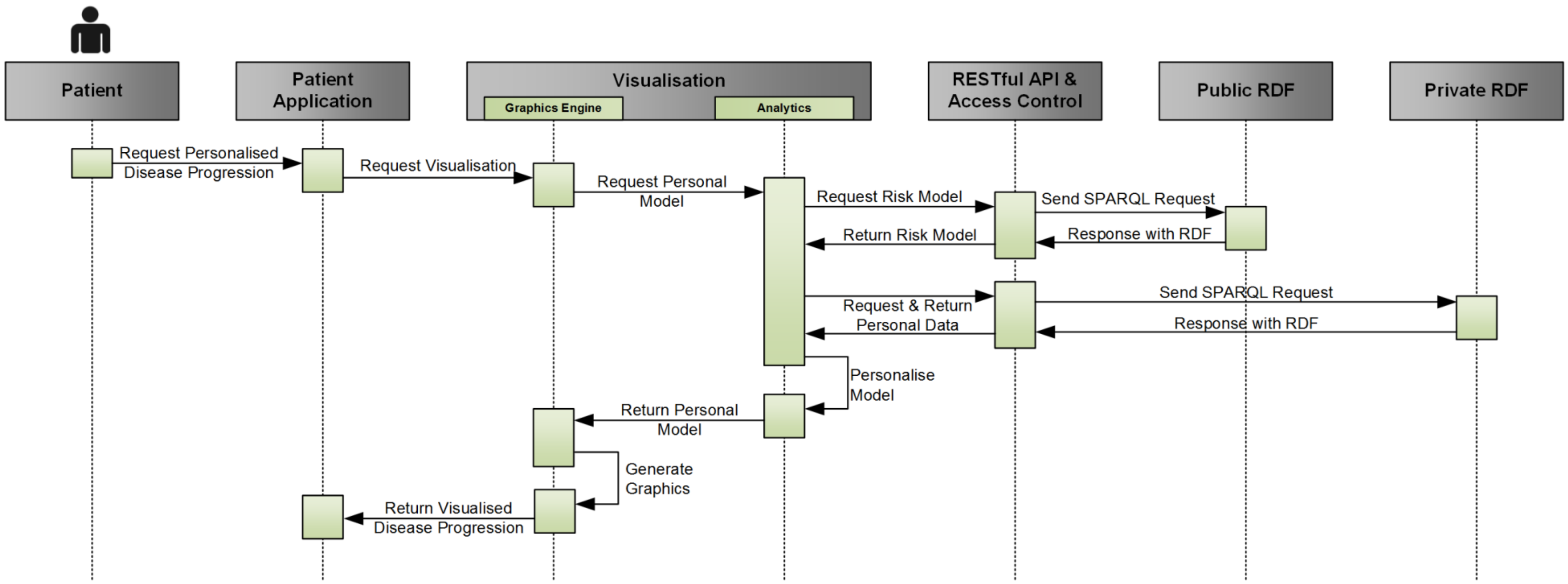

Figure 6: Patient accessing personalised CARRE services.

5.6. Alert generation for patient

This scenario describes how CARRE will push alerts to the patient. The specific scenario described in Figure 7 illustrates the generation of an alert for the patient after some personal data have been updated. More specifically, the sequence diagram shows that the DSS initially loads the risk associations model. It then periodically requests the personal data for patients. After an arbitrary number of requests, some change in the patient’s personal data result in an alert detection. The DSS component is generating this alert and is passing it to the visualisation component. Ultimately, this alert is received by the patient through the patient application.

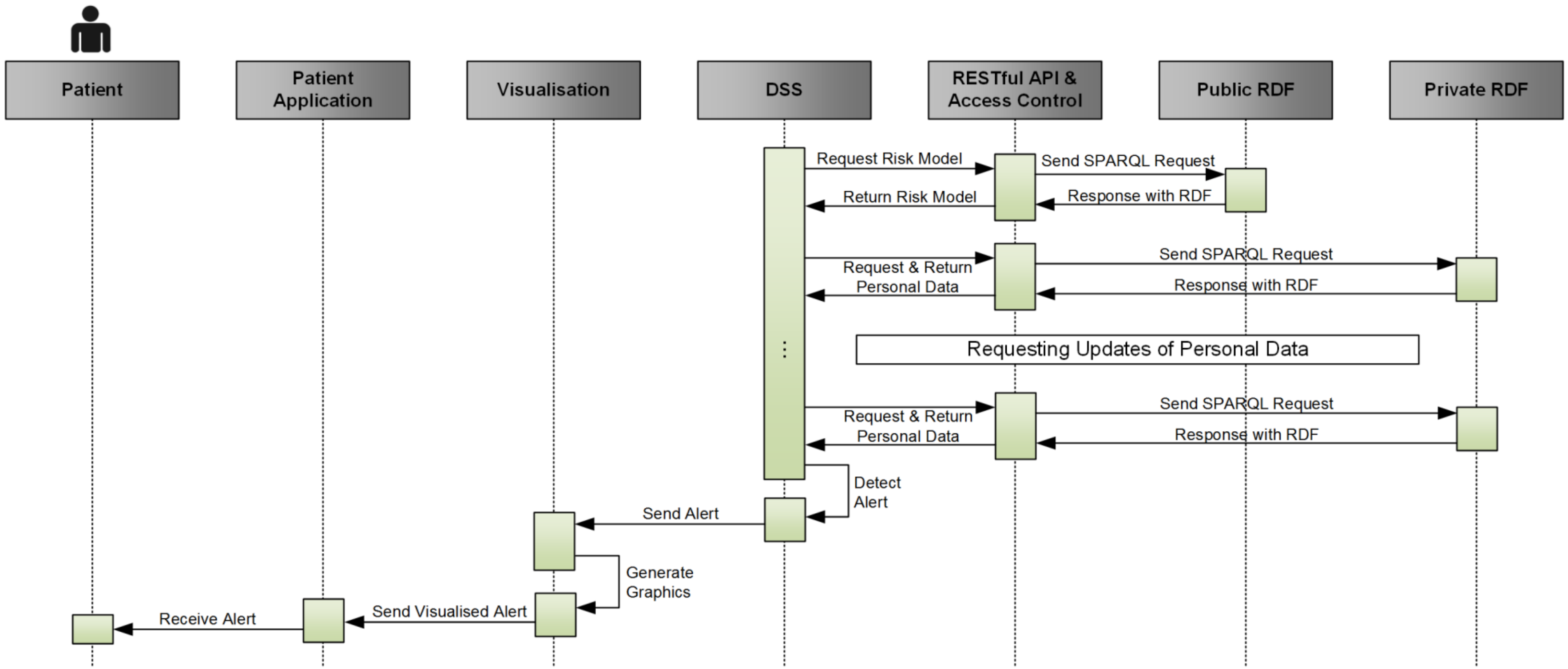

Figure 7: Patient receives an alert from CARRE.

5.7. External developer accessing the Public RDF Repository

Figure 8 illustrates a simple scenario where a third party developer wishes to develop an application/service using CARRE's already populated public data. To achieve this, the developer follows the standard way of connecting to an RDF repository. More specifically, she connects to the repository and sends one or more SPARQL requests over HTTP. The request results are typically returned in RDF and the session is complete by disconnecting from the repository.

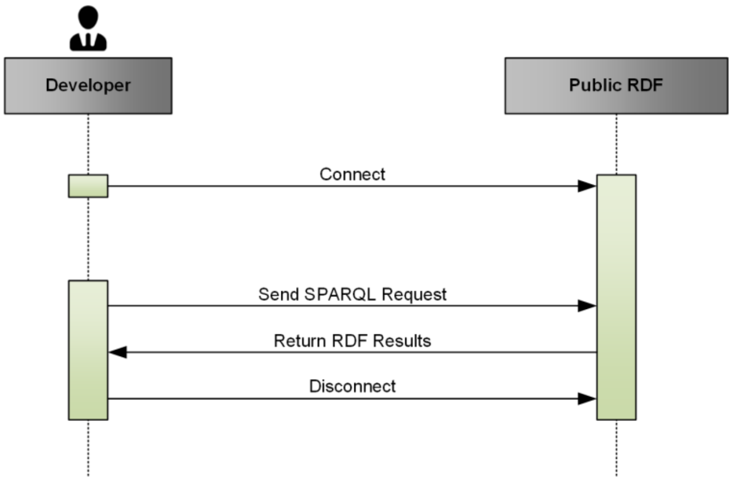

Figure 8: Developer using CARRE's Public Data.