

RDF to Salesforce extension

Software Requirements Specification

Prepared for  
Enterprise Information Systems Lab  
Summer semester 2015

Project mentor:

Niklas Petersen

Group members:

Alexander Melnyk

Mintcho Tzatzarov

Omar Rana

## Revision history

Date	Author	Comment
08.05.2015	Alexander Melnyk	Added revision history since the requirements will change during the Lab and we need keep the history in mind.

# 1. Introduction

This document gives an overview about the software requirements specification of the **RDF to Salesforce extension (RDF2SF)** that should provide an extended functionality for the cloud CRM system salesforce and enhance the data exchange between companies.

This document is divided in four chapters. The first part introduces the document and gives an brief overview of the system. The overall description of the project and the future system is in the second chapter, where the product perspective, product function, user characteristics and constraints are presented. The detailed description of the requirements, such as functional and nonfunctional requirements, is in the third chapter. The last chapter is an appendix with figures, that are referenced in other chapters.

## 1.1 Purpose

The purpose of this document is to give an detailed overview of the requirements of the system, that should extend the salesforce REST API by RDF functionality. This document can be used by project managers and software developers in order to identify, validate, design and implement the overall technical features and limitations of the system.

## 1.2 Scope

This document details the functionality required for the design and the implementation of the **RDF to Salesforce extension (RDF2SF)**. It has an overall description with product perspective, product functions, user characteristics and constraints. It also contains functional and nonfunctional requirements such as performance, reliability, usability, interoperability and security.

## 1.3 System overview

One of the common tasks of enterprises is to gather, exchange and manage the information about other companies. Due to rapid changes on the market, this information needs to be updated constantly. But the different data structures of each enterprise make the exchange significantly harder. At the same time, many companies use salesforce.com as a cloud CRM system, where they manage some of this data. The proposed system should use salesforce REST API in order to make the exchange of the data easier and to provide RDF support.

## 2. Overall description

### 2.1 Product perspective

Many companies gather, store and exchange basic information about other companies such as names, addresses and contact data. This information may change often and companies need to update it constantly, which induce many problems and challenges. One of the problems in such situation is the different data structure of this information. While there are some standardized data formats like vCard, no standard exists now for the entire complexity of the data of each company. Furthermore there is no convenient process for exchanging such information.

Salesforce is a cloud based CRM system that is widely used in US and in some countries in Europe. Many companies uses this platform and integrate the functionality in their own systems. The proposed system should use the salesforce API capabilities in order to exchange information between companies. The system should be able to consume the salesforce REST service for import and export of the data as JSON documents. Furthermore the system should transform JSON document into a RDF graph and vice versa.

### 2.2 Product functions

The salesforce REST API provides a rich functionality and access to many data models of the system. The RDF2SF extension should use this REST API and support the import and the export of the salesforce data structure. The extension uses a RDF graph from a vocabulary as an input, that is developed in the context of the LUCID project and will be provided for this system. This vocabulary contains much more information that could be synchronized with salesforce data mode and handled during the given time of the Lab. Therefore the system should only contain the functionality for **Sales Objects** (see appendix 4.1).

The system should provide a roundtrip of data processing between sales objects and RDF graphs. Given RDF graph the extension should import this information into salesforce. The extension should also export the data from salesforce in JSON document format and transform this output into RDF graph.

### 2.3 User characteristics

The main user characteristic is the knowledge about the salesforce: terms, data structure and workflows. The RDF2SF extension should be modelled after this characteristic and help the user and developer to work this salesforce.

## 2.4 Constraints, assumptions and dependencies

The system would require an salesforce account. We are using a free developer salesforce account which could have limited functionality. There can be constraints and restrictions of REST API provided by salesforce such as number of API calls, file storage size and bandwidth. The system will therefore depend on salesforce REST API data structure. The RDF data processing will be reused from some 3rd party libraries and needs to be evaluated during the development phase.

## 3. Specific requirements

### 3.1 External interface requirements

The core dependency of the system is salesforce REST API. In order to extend the functionality the system should have access to an valid salesforce account. The restrictions and data structure of the salesforce system can change in the future, which makes this dependencies more critical but unavoidable. The extension should be implemented as a service as well and does not have a client application. The programming language of the system is Java. For development, testing and running the application we will need a webserver such as Jetty, Tomcat, Glassfish or others. The salesforce REST API responses are represented as JSON documents.

### 3.2 Functional requirements

**Name:** Read Sales Objects

**Summary:** Read the data structure of the Sales Objects Model (SOM)

**Rationale:** In order to exchange the data between different companies, the system must be able to read the data from salesforce. The response of the REST API are JSON documents.

**Requirements:** The system should read any object from the SOM.

**Name:** Write Sales Objects

**Summary:** Write the data structure of the SOM

**Rationale:** In order to exchange the data between different companies, the system must be able to write the data to salesforce. The requests of the REST API

**Requirements:** The system should write any objects from SOM.

**Name:** Transform Sales Objects to RDF

**Summary:** Transform a Sales Object into an RDF graph

**Rationale:** In order to provide an uniform data structure for company data, the system must be able to transform the SOM JSON document into an RDF graph.

**Requirements:** The system should transform any objects from SOM into an RDF graph.

**Name:** Transform RDF graph to Sales Objects

**Summary:** Transform an RDF graph into one or many Sales Objects

**Rationale:** In order to support the exchange of the uniform data structure, the system must be able to transform an RDF graph into one or many Sales Objects as JSON document.

**Requirements:** The system should transform an RDF graph into one or many Sales Objects.

### 3.3 Performance requirements

The system must meet the performance requirements of the salesforce API restrictions such as number of API requests per day, file storage size and bandwidth usage per 24 hours. The response time of a request should not be longer than 5 seconds. This requirement depends on the performance of the salesforce API, since the designer might not have influence on the performance of the consumed service.

### 3.4 Nonfunctional requirements

#### 3.4.1 Security

The system will use the salesforce account of the provided user. Therefore the user credential must be protected. Furthermore the system must have some logging mechanism in order to verify and trace any changes done by the system.

#### 3.4.2 Reliability

The system must provide reliable functionality, thus 99% of all requests need to be executed successfully.

#### 3.4.4 Usability

The user of the system should be familiar with salesforce system, including terms and workflows. In order to make the system more usable, it must contain the same terms, workflows and object structure as salesforce API.

#### 3.4.4 Interoperability

In order to extend the Salesforce REST API the system must be designed and implemented as a REST service as well. This includes the a running REST webservice on a web server and processing JSON data format as input and output.

## 4. Appendix

### 4.1 Sales Objects Model of Salesforce.com API

