OTTO VON GUERICKE
UNIVERSITÄT
MAGDEBURG

GLEIF PARSER and Visualizer

Technical Design Document

20 April 2019

Version 1.0

AUTHORS

Name	Role	Department
Bhuvanesh Leelakrishnan	Developer & Team Lead	Data and Knowledge Engineering
Aarthi Ilangovan	Developer & Documentation	Data and Knowledge Engineering
Sowmya Prakash	Developer & Documentation	Data and Knowledge Engineering
Yamuna Nagasandra Rajaiah	Developer & Documentation	Data and Knowledge Engineering

APPROVALS

Approver Date	Approver Role	Approver
21.04.2019	Project Supervisor	Stefan Willi Hart

Contents

1. Introduction

This document describes the high-level architecture of GLEIF Parser and Visualizer. GLEIF stands for Global Legal Entity Identifier Foundation, which manages LEI's of all the legal entities participating in financial transactions. GLEIF stores these LEI data in various formats and one such format is XML which can be downloaded for further user processing or use. The requirement of this project is to parse the XML files which holds the Level 1 and Level 2 data of GLEIF and store it in SAP HANA database. Furthermore, reports are displayed in a web page enabling user-search facility to apply different filter criteria for viewing the GLEIF data.

1.1. Purpose/Audience

The audience for this document is the development team and manager of this project. This document explains the architecture diagram that would be used for developing a software product. The architecture diagram provides an overview of an entire system, identifying the main components of the system and the relation with their interfaces. The project manager verifies the design to see if all the functionality is covered as part of the project.

1.2. Functional Requirements

- Application should be able to load GLEIF XML files from depending on user needs.
- GLIEF data source: Understanding final HANA table structure.
- Application should be able to parse XML files:
- Automatically parse XML files into SAP HANA using JAVA.
- XML files which is of unstructured data type. Parsing XML structure of nested fields are challenging.
- Application user interface should provide a upload button to upload GLEIF zip file from which XML will be extracted.
- Search criteria to be Country name, File Content Date, Registration Status and Active LEI.

1.3. Non - Functional Requirements

- Performance (optimum response time)
- Fault tolerance
- Portability
- Usability
- Availability
- Maintainability

2. Tech Stack

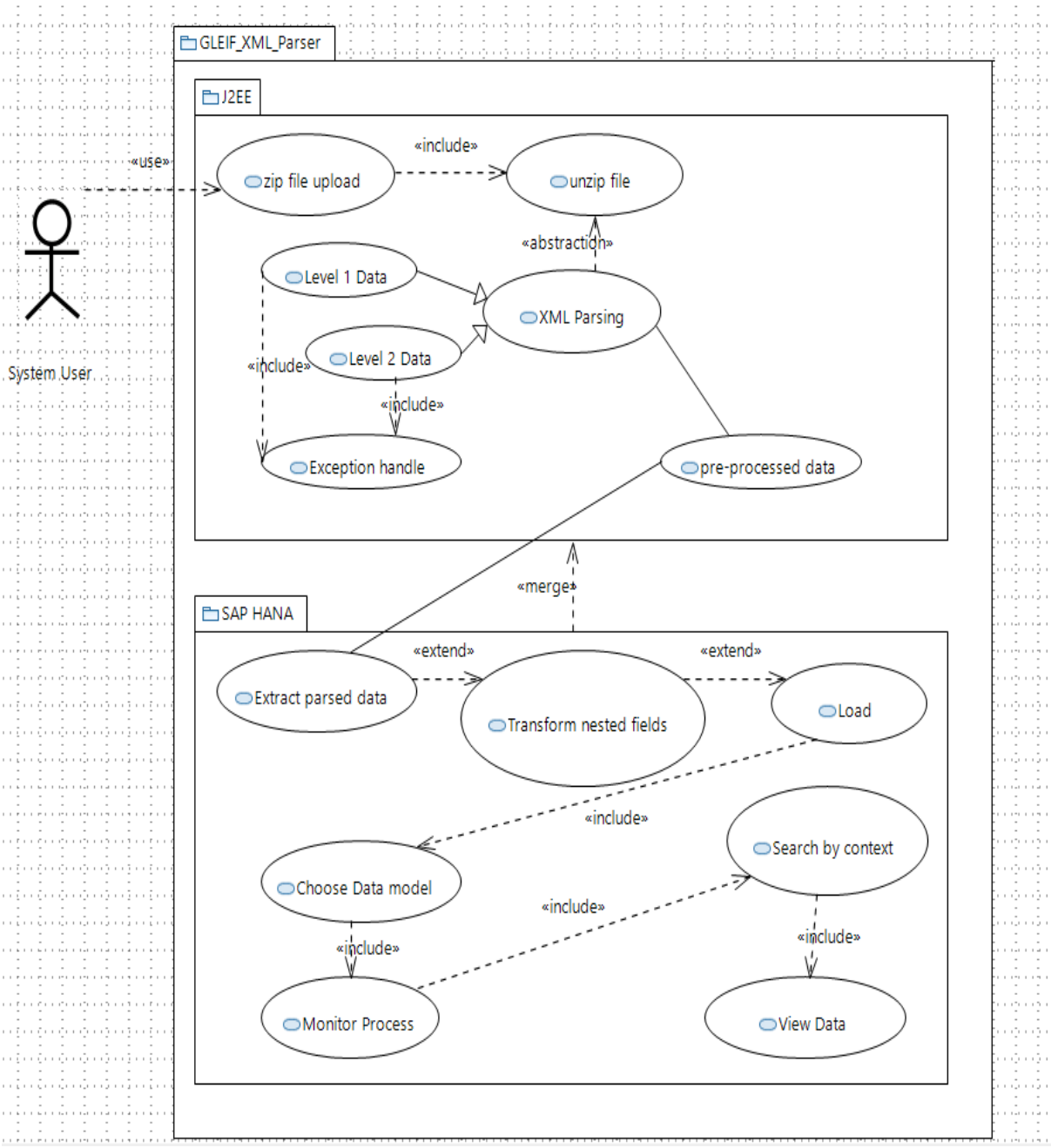
Area	Framework Name	Description
J2EE/Web framework	Spring boot 2.1 (jdk 8 and above) / Spring 5.0	Spring Boot makes it easy to create stand-alone, production-grade Spring based Applications that one can "just run". The Spring Framework provides a comprehensive programming and configuration model for modern Java-based enterprise applications - on any kind of deployment platform.
Parsing XML files	JAXB/ Stax	
Database interaction	JPA/Hibernate	
Database	SAP HANA	Java connects to SAP HANA via ngdbc.jar
Frontend template engine	Thymeleaf	Thymeleaf is a Java template engine for processing and creating HTML, XML, JavaScript, CSS, and text.
Front End	HTML/CSS/Jquery	User needs upload button from visualization of data.

2.1. Field level specifications

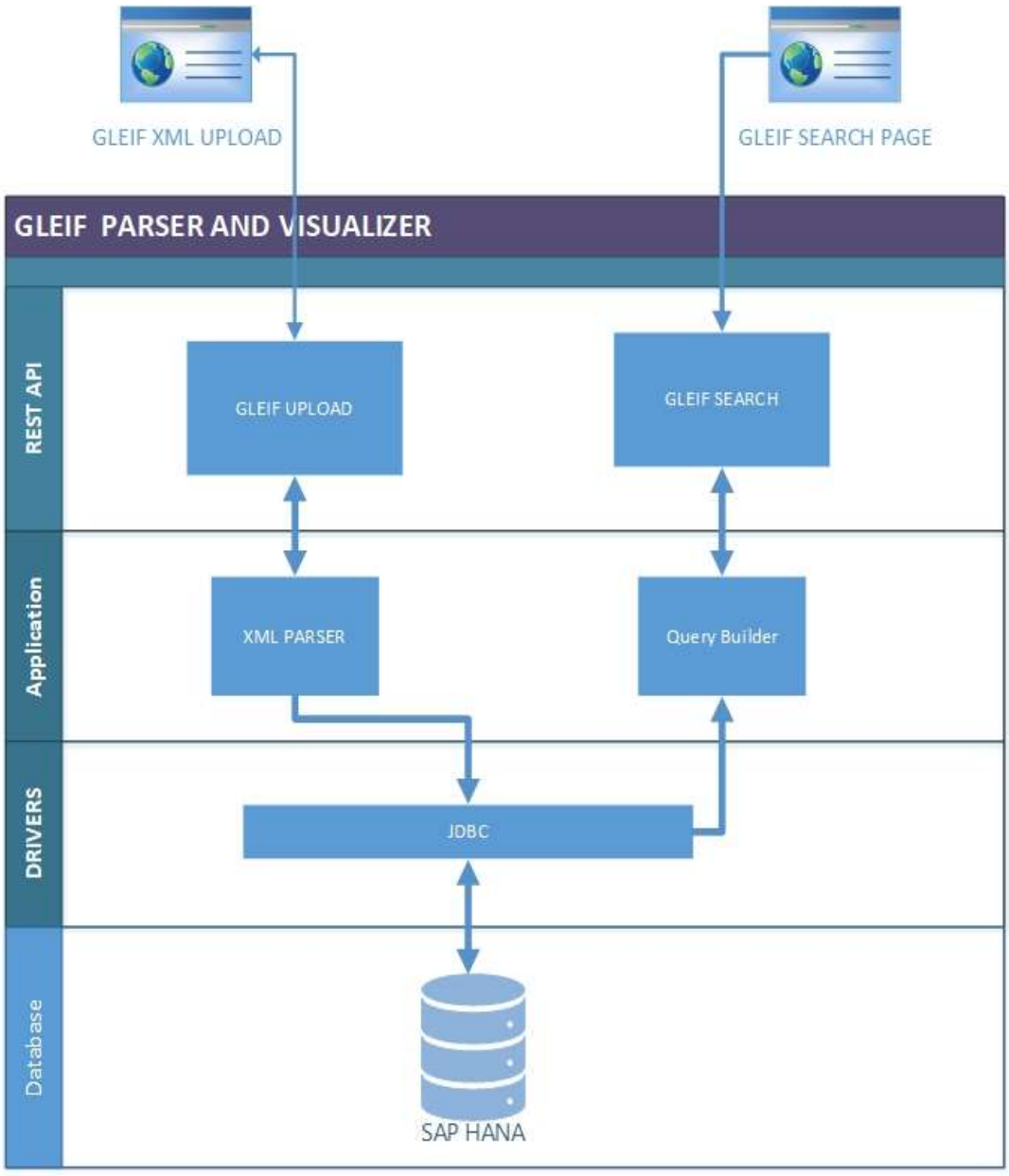
Form Elements:

Call-out	Field Label	UI Control	Mand ?	Editable	Data Type	Value Set	Default Value	Data Example	Data Source
<upload>	File upload	button	yes	no	NA	NA	NA	XML File	GLEIF Website
<legal name>	Legal Name	textbox	No	yes	string	NA	NA	Company Legal Name	User entry
<country name>	Country Name	textbox	No	yes	string	NA	NA	Country	User entry
<search>	Search	search	yes	no	NA	NA	NA	NA	NA

3. System use-case diagram



3.1. Solution Overview



4. Database Design

TABLE NAME	TABLE DATA
GLEIF_HEADER	GLIEF file headers.
LEVEL1_LEI_RECORD	GLIEF level 1 records.
LEVEL2_EXCEPTION_REASON	GLEIF data exception reason.
LEVEL2_EXCEPTION_REFERENCE	GLEIF data exception reference.
LEVEL2_REPORTING_EXCEPTION	GLEIF reporting exceptions.
LEVEL2_RR_RELATIONSHIP_QUALIFIER	GLEIF relationship qualifier.
LEVEL2_RR_RELATIONSHIP_QUANTIFIER	GLEIF relationship quantifier.
LEVEL2_RELATIONSHIP_RECORD	GLEIF relationship records.

4.1. System Files

The GLEIF application takes input in the form of 3 files:

SYSTEM FILES	CONTENTS
Level 1 LEI	This contains legal entity identifier, registered address of legal entity, country, codes for the representation of names of countries and their subdivisions, date of last update information etc.,.
Level 2 RR	This holds relationship records of LEI registered companies, whose parents have obtained an LEI.
Level 2 Exception reporting	This comprises of parent entity exceptions like no parent LEI or other exceptional reasons which may occur in child entity.

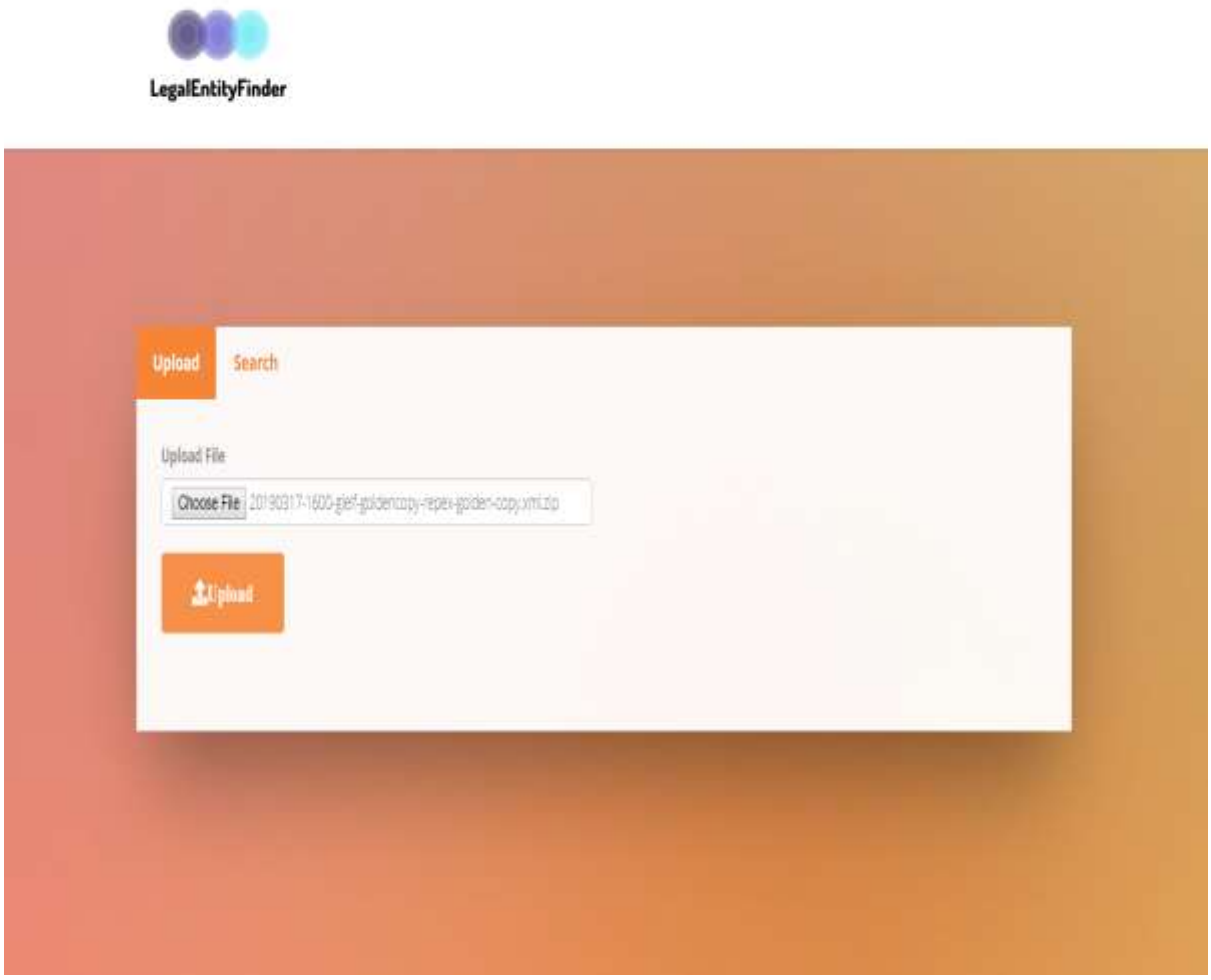
5. Implementation of Optimization Techniques

CONCURRENT PROCESSING: The GLEIF system files are processed through multithreading process and thus, the below results are achieved:

1. File upload time reduced by 80% as compared to previous implementation. This avoids system latency.
2. Parallel insertion of records into HANA database.
3. Processing time reduced while retrieving records for search queries.

6. Output results

1. Upload the available downloaded XML data copy into the GLEIF Parser and Visualizer



2. User ability to retrieve the required search fields out of available Content filters such as Legal Name, Country Name