Tagging of M&A Documents (Infineon)

Software Architecture Description

AMOS – 5

Contents

1. Description

1.1 Purpose of this document

1.2 Architectural requirements

1.2.1 Non-functional requirements

1.2.2 Use Case View (functional requirements)

1.2.3 Scope of the project

1.2.4 Technology stack

# Description

# 1.1 Purpose of Document

This document tries to aggregate the System architecture description form the different levels and provides a comprehensive overview of the software to illustrate different aspects of the system

# 1.2 Architectural requirements

The goal of this Software Architecture Document is to achieve the intended result of the proposed system. This section will briefly determine the functional and non-functional requirements of the system.

## 1.2.1 Non-functional requirements

This part of the document precise requirements that address characteristics of the system besides the functionalities but are relevant for the system architecture. The non-functional requirements of the software have constraints and qualities.

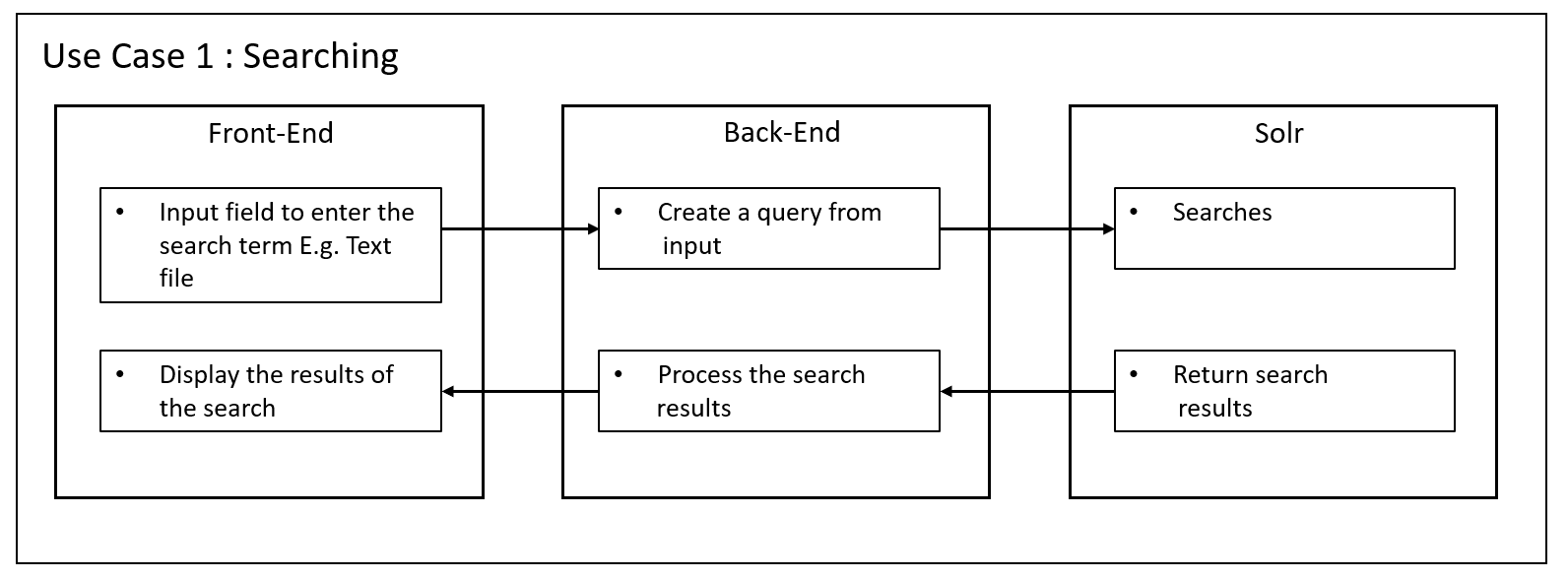
**Constraints -** it is the limitations and conditions on the system.

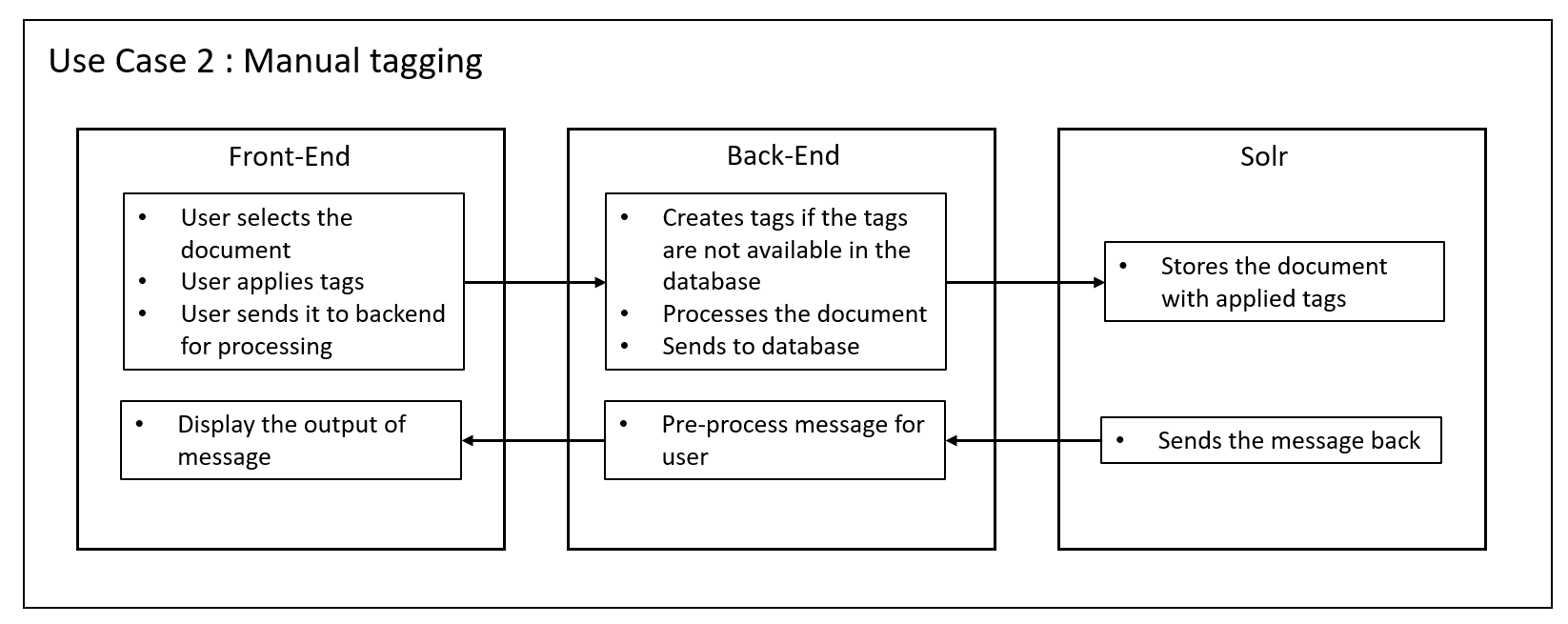
* The application needs to run on any web browser.

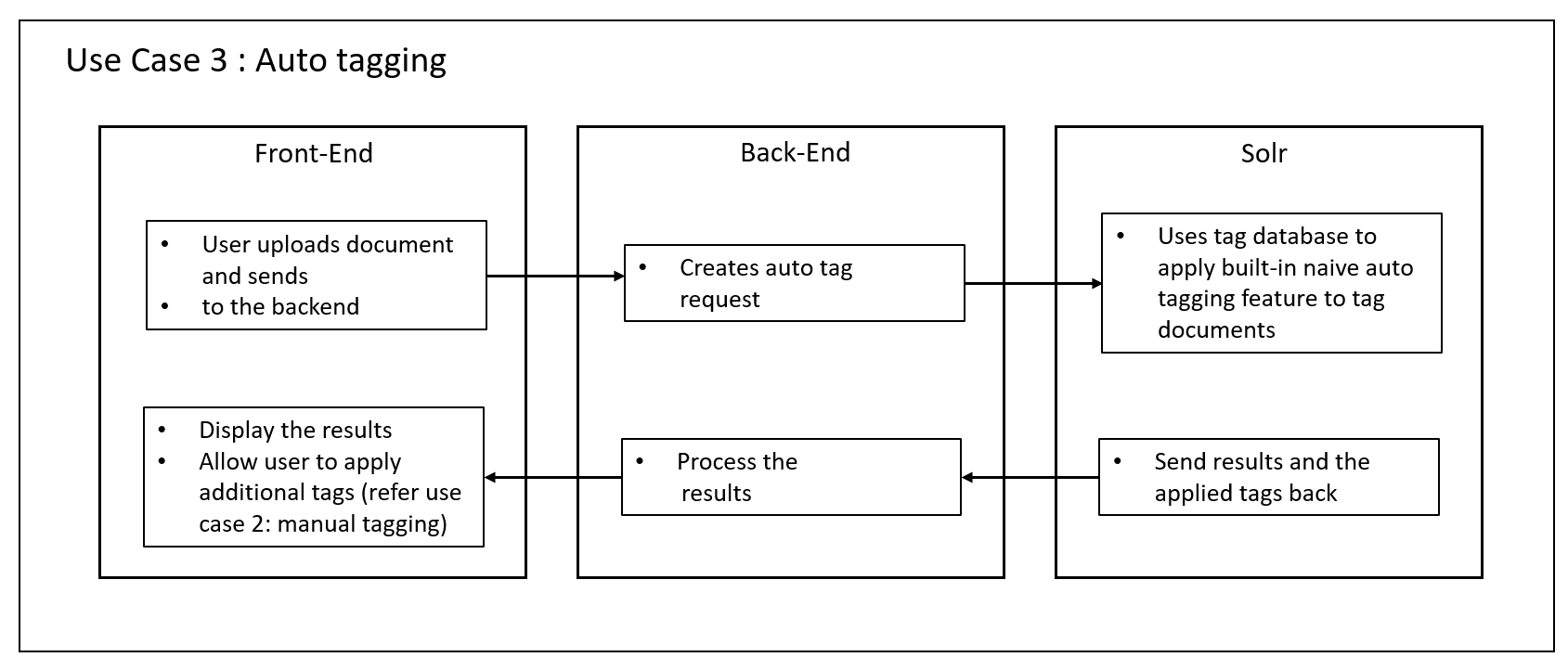
**Qualities -** it is the qualities of the website in terms of service and management.

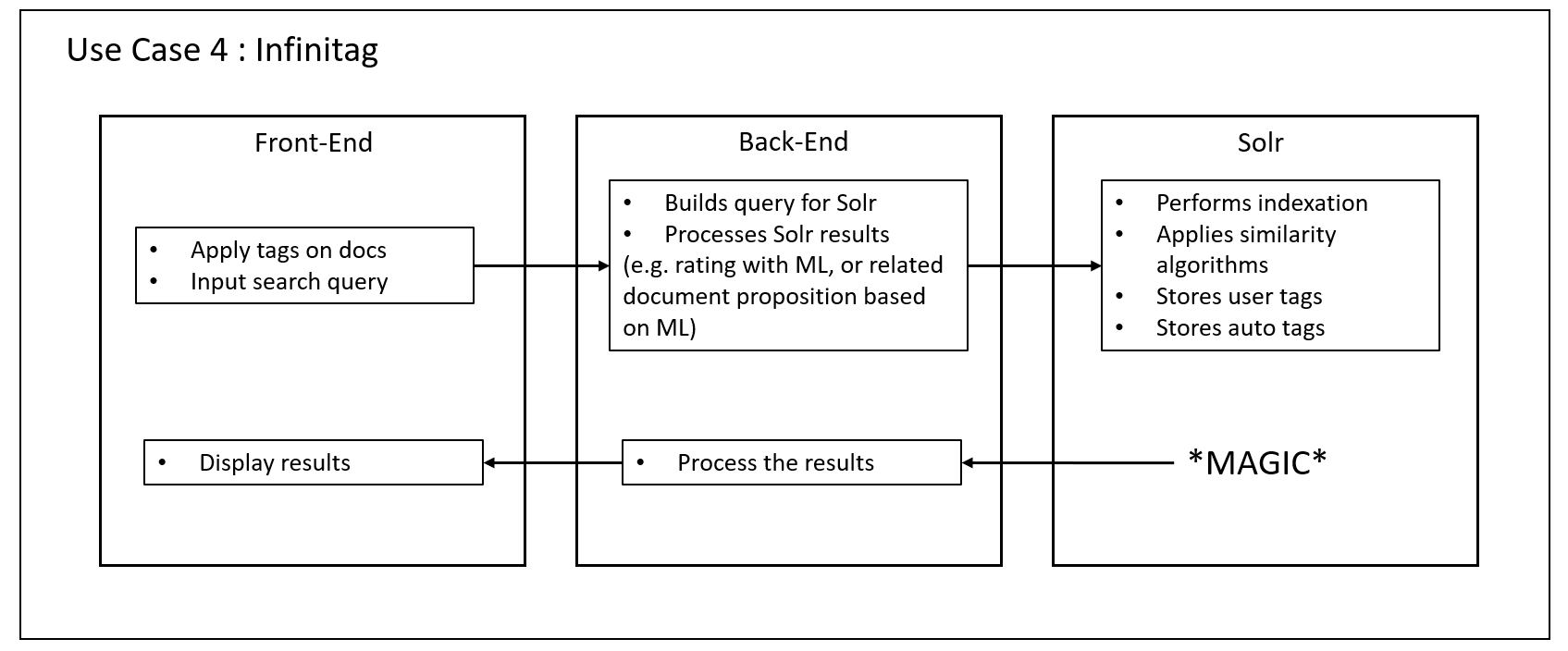
* The application is done through a lot of data gathering and research about the Machine learning algorithms and search engines
* The application is opensource, is licensed under MIT

## Use Case View (functional requirements)









## 1.2.3 Scope of the project

The following table has the constraints or scope of the project:

|  |  |
| --- | --- |
| 1 | Supports English and German language |
| 2 | Our software can only work with plain text, i.e. scanned or faxed documents that do contain “images of text” only will not work. |
| 3 | ML techniques used in tagging words to the text files |
| 4 | Structured data will be uploaded |

## Technology stack

|  |  |  |
| --- | --- | --- |
|  | **Language** | **Framework** |
| **Front-end** | Typescript | Angular Frontend |
| **Back-end** | Python [C++ optimization if required] | Flask, Solrlib |
| **Solr** | * Indexation * Feature extraction (tokenization, document preprocessing) * Language independent * Multi filetype support (eg. txt,html,json,pdf,doc,docx,pptx,etc.) * Built-in similarity features * Later: built-in machine learning feature which can be explored | |