# TFS Data Marketplace
# Vision

## Version <1.3>

# Revision History

| Date | Version | Description | Author |
|---|---|---|---|
| <10/03/2021> | <1.0> | First Version | TFS Data Marketplace team |
| <12/03/2021> | <1.1> | Second Version | TFS Data Marketplace team |
| <02/10/2022> | <1.2> | Third Version | TFS Data Marketplace team |
| <04/11/2022> | <1.3> | Forth Version | TFS Data Marketplace team |

# Table of Contents

# Vision (TFS Data Marketplace)

## 1. Introduction

Toyota Financial Services (TFS), being part of a highly regulated industry needs to ensure that all risk management, governance process and controls are in place to ensure compliance. This entails documenting all the business processes, definitions of our data elements, connecting the defined data elements to the physical attributes in our various applications and databases. Furthermore, we need to document the lineage of the data to ensure that it is flowing correctly through our ecosystem. In addition to these, we must ensure the data quality at the source and through the transformations it goes through while flowing in our ecosystem.

### 1.1 Background

Currently, we can record, store, check and correct all the data in our ecosystem. However, all of these are disjointed, and we don't have a holistic view.

- we can store the data definitions via crowdsourcing in our data governance platform (Informatica Axon).

- We can scan the applications and databases for the physical data attributes (Informatica EDC). The linking of the physical and business data elements, however, is a manual process.

- We can connect to the source applications and databases to do data profiling for quality checks and apply data quality rules (Informatica Data Quality) wherever applicable.

## 2. Business Requirements

We want to have a holistic view like the portal that we see at https://www.data.gov/ . We would like to have a similar portal where we can visualize data duly categorized that end users can use for visualization and solving their business problems.

### 2.1 Business Opportunity/Problem Statement

| The problem of | • *Disjointed system to record, store, check and correct all the data in the ecosystem/ no holistic view of data* |
|---|---|
| affects | • *Employers/ Business partners of TFS* |
| the impact of which is | • *Unorganized data, manual process of linking physical and business data elements which is time consuming* |
| a successful solution would be | • *Data to be organized by business areas such as Loan Originations, Insurance, Servicing etc.*<br><br>• *Ability to do google like search for any data element which would bring up business* |

| | *definitions, physical attributes, data quality rules & profile and any related data associated with it* |
|---|---|

## 2.2 Vision Statement

● We want to have a holistic view like the portal that we see at https://www.data.gov/ . We would like to have a similar portal where we can visualize data duly categorized that end users can use for visualization and solving their business problems.

  ● Leverage CKAN (https://ckan.org/) to store our business metadata definitions which is connected to the physical attributes and subsequently, the associated data quality and profiles of these data elements.

  ● Data to be organized by our business areas such as Loan Originations, Insurance, Servicing etc.

  ● ability to do google like search for any data element which would bring up business definitions, physical attributes, data quality rules & profile and any related data associated with it

## 2.3 Business Risks

1. CKAN has a lack of outside resources aside from the official CKAN documentation. The scarcity of information on CKAN would make production slow.
2. The Data extracted from different sources do not seem to have the same number of columns. Pre-processing data might be a challenge.
3.  Making last minute changes to the project before meetings with clients can sometimes be less efficient.
4. Lack of experience using Jira among team members has made the documentation process a little time consuming.
5. Not having the same schedule for meetings among team members was a major issue.
6. Change in technology from CKAN to postgresql increased the learning curve for all team members
7. Not having enough AWS resources was a little challenging as well.

## 2.4 Business Assumptions and Dependencies

● The user is assumed to have registered for the system to be able to use the website.
● The user will be able to register only via the link sent to them in the email by admins.
● The admins are assumed to send an invitation to the users or other admins, and also specify the role of the person while sending an invitation.
● The user will need a stable internet connection to be able to use the website.
● The project will have various framework(flask) dependencies.

# 3.      Stakeholder Profiles and User Descriptions

## 3.1     Stakeholder Summary

**TFS Members:**

### Project Leadership and guidance

- Gaurav Lall – gaurav.lall@toyota.com

### Project Co-ordination and technology support

- Tai Trieu – tai.trieu@toyota.com

### Axon and EDC (Data Catalog and Business Glossary)

- Aritra Das – Aritra.das@toyota.com

### Informatica Data Quality (IDQ)

- Arpita Santra – Arpita.Santra@toyota.com


**TCU Members:**

### Project Leadership and guidance
Bingyang Wei  – b.wei@tcu.edu
Krishna Kadiyala – k.kadiyala@tcu.edu

### Project Team
Keenan D'spain - KEENAN.DSPAIN@tcu.edu
Loc Pham - LOC.PHAM@tcu.edu
Sabina Khanal - SABINA.KHANAL@tcu.edu
Kundan Chaudhary - K.CHAUDHARY@tcu.edu
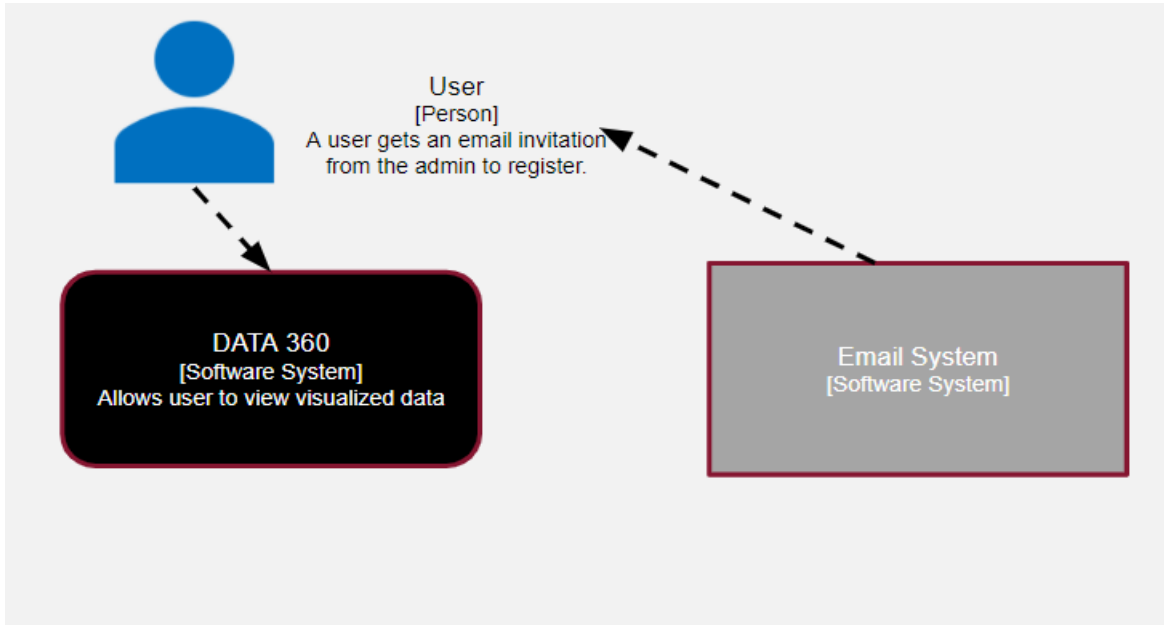Khiem Nguyen - KHIEM.NGUYEN@tcu.edu


## 3.2     User Environment

- The user gets an email with an invitation link to register for our software system.
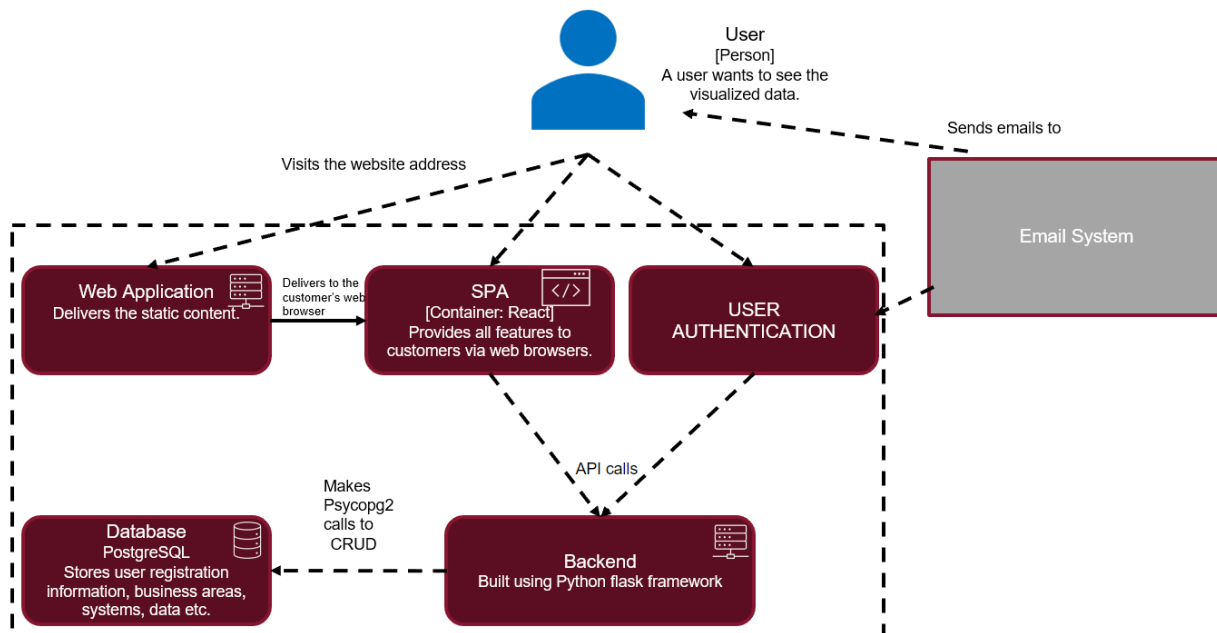- Our software needs to integrate with TFS email system.

## 4.  Scope and Limitations

### 4.1  Product Perspective



The user is allowed to register on the portal and use it via email invitation only.

Registered user makes request using RESTful API from frontend to backend. The backend processes the requests and makes PostgreSQL database calls using Psycopg2 library.
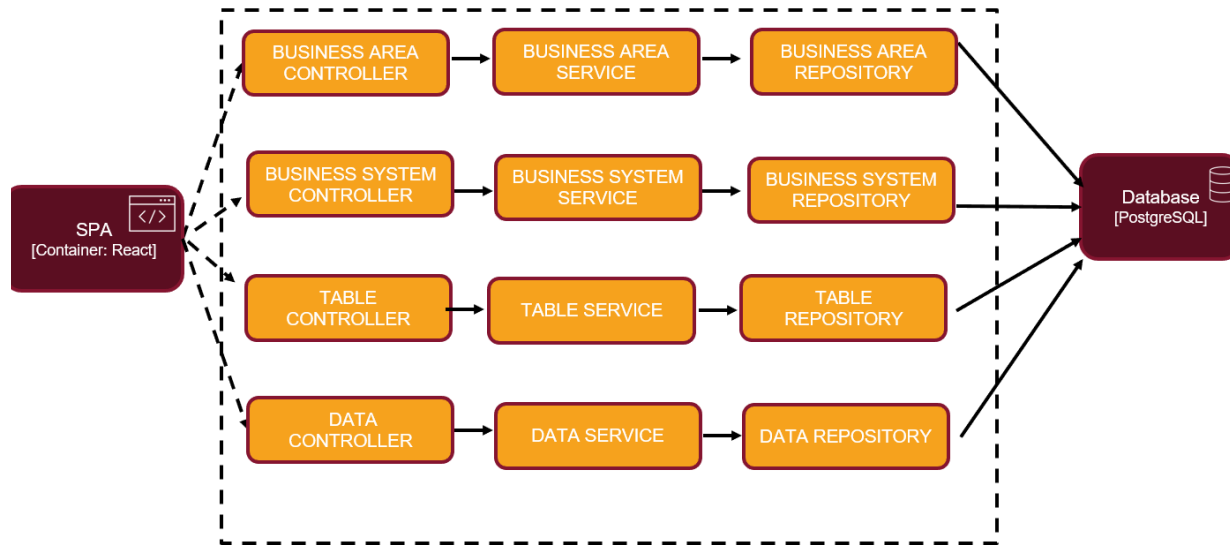


**Controller:** Takes requests from the front-end and passes it to the service or passes the results to the front-end.

**Service:** The logic layer of the application. Performs all the computations needed.

**Repository:** Fetches the data from the database or commits the data to database as needed.

### 4.2    Major Features / Scope

FE-1: User gets an email invitation to register for the software system.

FE-2: The home page lists all the business areas under TFS.

FE-3: Admins can create, edit, view or delete the business areas and systems. Admins will need to add Axon, EDC and IDQ files for the tables and data section.

FE-4: Users can only view the business areas, systems, tables or data.

FE-5: The user can look for a business area on home page or directly look for the data from the search bar present on the top of the page.

FE-6: When clicked on any business area, a page with all the systems under that business area opens.

FE-7: When clicked on any system, a page with some details about that system opens. The page also has the tables present under that system.

FE-8: When clicked on any table, a page with data under that table opens.

FE-9: Finally, when any data is selected, the 360 view of that data is displayed.

FE-10: There is also user management feature for admins on the left of homepage from where they can edit or delete information about an user/admin.

### 4.3    Deployment Considerations

The frontend was deployed in Vercel and the backend is deployed in AWS.
The user who has registered for the software system will be able to access the software system using the URL:
tfsdp.vercel.app/home