



Document Version / Details: Ver. 1.0/ 01-Mar-2018

**Technical Design Document**

**Record of Release**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Version No. | Modified By | Reviewed By | Authorized By | Release Date | Modifications Done |
| 1 | Siva/ Mayuresh |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

**Table of Contents**

[1 Introduction 4](#_Toc2335914)

[1.1 Purpose 4](#_Toc2335915)

[1.2 Scope 4](#_Toc2335916)

[1.3 Document Reference 6](#_Toc2335917)

[1.4 Glossary 6](#_Toc2335918)

[2 Technical Overview 7](#_Toc2335919)

[2.1 Participating Applications 7](#_Toc2335920)

[2.2 Architecture Overview Diagram (AOD) 7](#_Toc2335921)

[2.3 Design Assumptions 7](#_Toc2335922)

[2.4 Design Constraints 7](#_Toc2335923)

[2.5 Performance Considerations 7](#_Toc2335924)

[2.6 Integration Infrastructure Versions 7](#_Toc2335925)

[3 Integration Solution Flow 8](#_Toc2335926)

[3.1 Integration Process Flow Diagram 8](#_Toc2335927)

[3.2 {flow\_name} Integration Flows 8](#_Toc2335928)

[3.2.1 Composite Description 8](#_Toc2335929)

[3.2.2 Service Details 9](#_Toc2335930)

[3.2.3 Component Details 9](#_Toc2335931)

[3.2.4 Reference Details 10](#_Toc2335932)

[3.2.5 Adapters 10](#_Toc2335933)

[3.2.6 Common Services 10](#_Toc2335934)

[3.2.7 DVMs 11](#_Toc2335935)

[3.2.8 Cross References 11](#_Toc2335936)

[3.2.9 Error Handling 11](#_Toc2335937)

[4 Open and Closed Issues 12](#_Toc2335938)

[4.1 Open Issues 12](#_Toc2335940)

[4.2 Closed Issues 12](#_Toc2335941)

# Introduction

This document contains Integration requirements

1. The introduction of the **Requirements Specification** provides an overview of the entire document. It includes the purpose, scope, definitions, acronyms, abbreviations, references, and overview of the document.
2. The **Requirments Specification** captures the complete software requirements for the system, or a portion of the system. Following is a typical outline for a project **using use case modeling**. This artifact consists of a package containing use cases of the use case model and applicable Supplemental Specifications and other supporting information.
3. This subsection describes what the rest of the **Requirements Specification** contains and explains how the document is organized.

This document will provide technical design of Data Management – SmartForms.

## Purpose

The purpose of this document is to explain the technical design and overview of SmartForms implementation for the Data management.

Its main purpose is to –

* Provide the link between the functional specification and the detailed technical design documents
* Detailed functionality provided by each component or group of components and show how the various components interact in the design
* Provide a basis for SmartForms detailed design and development
* This document is not intended to address installation and configuration details of the actual implementation. Installation and configuration details are provided in technology guides produced during project

As is true with any high-level design, this document will be updated and refined based on changing requirements.

## Scope

Build a web-application (Smart Form) to:

* Serve as a single-entry point for data management requests globally.
* Redirect the appropriate transactions to PeopleSoft self-service.
* Intake all transaction for Data management including includes Personal and Job changes
* Self-Service: Below is the list of transactions:

Address

Contact Details

Disability

Emergency contact

Ethnicity

Gender

Preferred Name

Veteran Info

Business title

Company

Department

FCLM Area and FCLM job

Full time - Part Time

Pay group

Product Line

Shift Pattern and ADP shift

Supervisor change

* Non-Self-service: Below is the list of non-self-service transaction

Bank Account details

Bank Account Number

CRI National ID

Date of Birth

Educational qualification

Legal Name

Marital Status

National ID Upload

SSN/SIN/National ID

Visual Survey Conducted

Work Permit Details

Agency

Amazon Discount Code

Business Unit

Compensation

EA (Executive Assistance)

Eligibility

Employment Date

Job Code

Location

M&A Hire/Re-Hire

Office Location Change

Part time Work Arrangement- FWA

Pay rate

Promotions

Public Transportation Reimbursement

Seasonal-Individual

Shift- Individual (CR)

Standard Hours

Status Change

Suspension

Telecommuting-Individual

* Validate bulk requests for accuracy of each row for transaction that are in scope for bulk uploads
* Escape valve is created to allow the requester to generate a ticket out from Smart form. Enable escape valve to capture the reason if the user is unable to use self-service or generate a ticket.
* Reduce the volume of manual ticket processing such as Fake self Service (department for initial release)

For Non self-service transaction provide a web form which will validate and capture request from user before creating the ticket (via API or Quicklink) or making output ready for RPA

## Document Reference

| Author | Document Name | URL | Comments | |
| --- | --- | --- | --- | --- |
|  |  |  |  | |
| Varun Verma | FunctionaRequirenentsDocument\_SF |  | |  |
| Siva/ Koteswararao/ Mayuresh | Solution Design Document |  | |  |

## Glossary

|  |  |
| --- | --- |
| **Term** | **Definition** |
| ESS | Employee Self Service |
| GSS | Generalist Self Service |
| MSS | Manager Self Service |
| RPA | Robot Processing Automation |
| SF | SmartForm |

# Technical Overview

## Participating Applications

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Sl. # | Application(Name of the application) | Type(If Source/Target System) | Interfacing Methodology(Protocol/Message Type to be used to integrate) | Remarks |
|  |  |  |  |  |

## Architecture Overview Diagram (AOD)

<Diagram>

Goals of this design:

<Define goals in bullet list format>

## Design Assumptions

<List all assumptions in bulleted items>

## Design Constraints

<List all constraints in bulleted items>

## Performance Considerations

<List performance considerations in bulleted items>

## Integration Infrastructure Versions

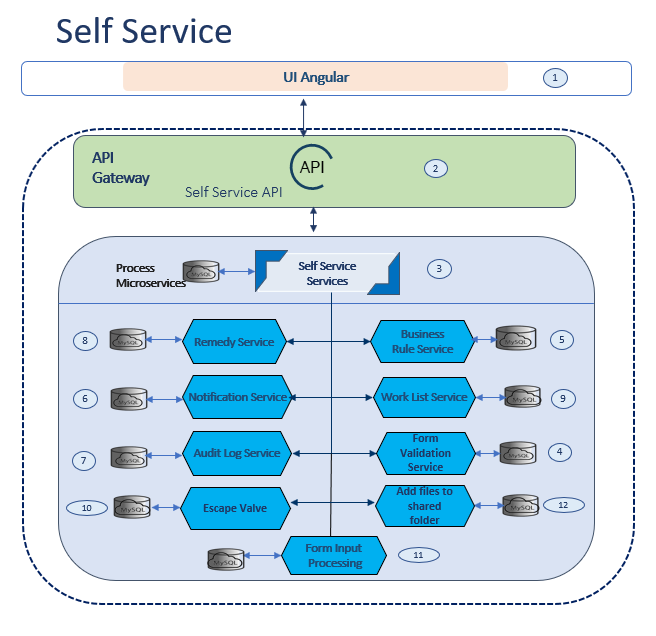
<Details about the integration infrastructure>

| Product | Version | Operating System | Database |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# Integration Solution Flow

## Integration Process Flow Diagram

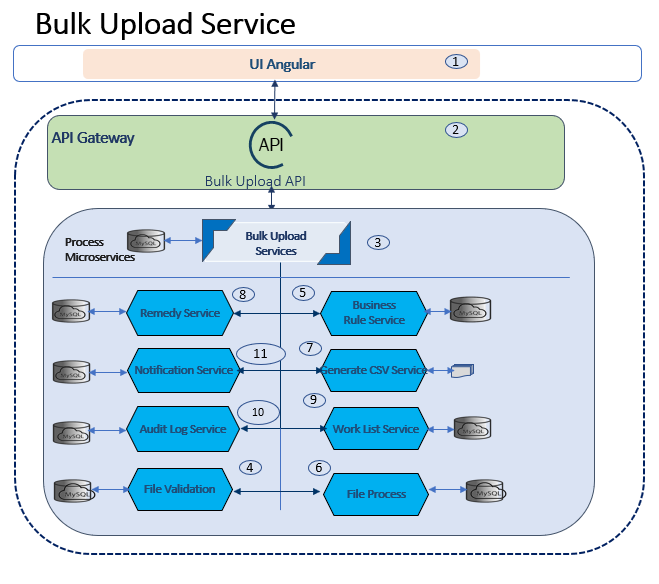
### Self Service



Flow execution steps:

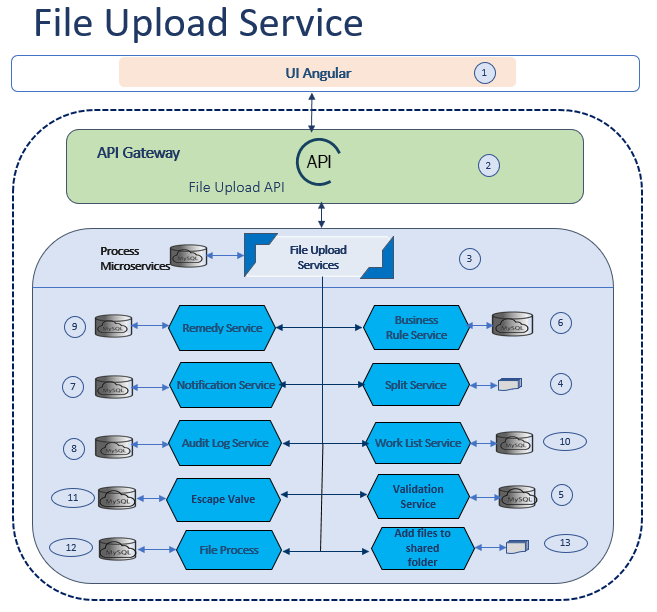
1. User Selects option from Self Service transaction from UI which calls Self Service API
2. Self Service API calls Self Service services
3. Self Service services retrieve configuration details and orchestrate system services calls in sequence configured for this use case
4. Form validation service performs basic validations on form input data.
5. Business rules are performed on the input and if any failure happens it will be logged and notified
6. Notification service will be invoked to send notification
7. Audit log service performs logging
8. Raise remedy ticket for each form submission & get remedy ID.
9. Insert worklist entry for every form submission
10. Escape Valve is present so user can use at any point he feels he cannot proceed further.
11. Form input processing saves data in automation database

### Bulk Upload Service

  
Flow execution steps

1. User Selects Bulk Upload from UI which calls Bulk Upload API
2. Bulk Upload API calls Bulk Upload Service
3. Bulk Upload Service retrieve configure details and orchestrate system services calls in sequence configured for this use case
4. Business rules are performed on the message and if any failure will be logged and notified as configured
5. Validate the Uploaded file based on business rules
6. File process
7. Generate CSV file
8. Raise ticket for each CSV file and get the Remedy ID
9. Insert work list entry for each CSV file
10. Insert Audit Logs
11. Send Notification

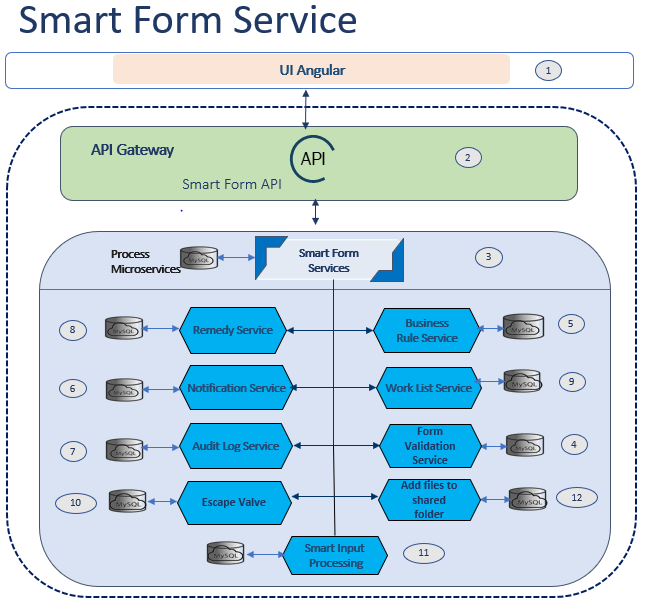
### File Upload Service



Flow execution steps:

1. User Selects option file upload from UI which calls file upload API
2. File Upload API calls File Upload services
3. File Upload services retrieve configuration details and orchestrate system services calls in sequence configured for this use case
4. Split service breaks down input file into equal sized parts and as per transaction.
5. Validation service checks for duplicate and null values in data
6. Business rules are performed on the data in the files and if any failure happens it will be logged and notified
7. Notification service will be invoked to send notification
8. Audit log service performs logging
9. Raise remedy ticket for each file upload & get remedy ID.
10. Insert worklist entry for every file upload.
11. Escape Valve is present so user can use at any point he feels he cannot proceed further.
12. File process writes data to automation database.
13. Files are then uploaded to shared folder

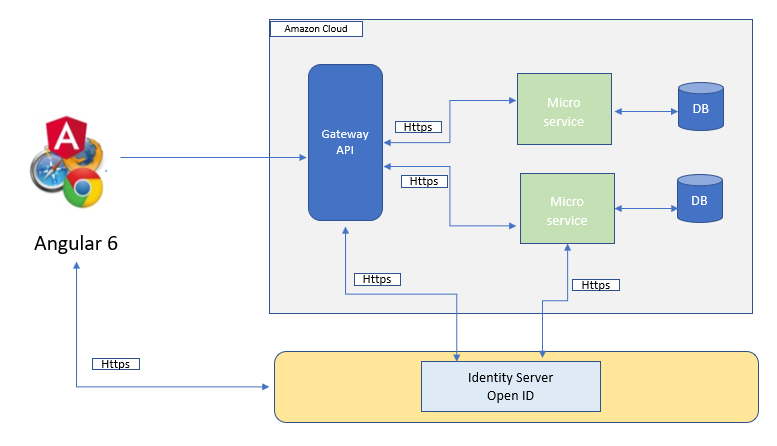
### Smart Form Service



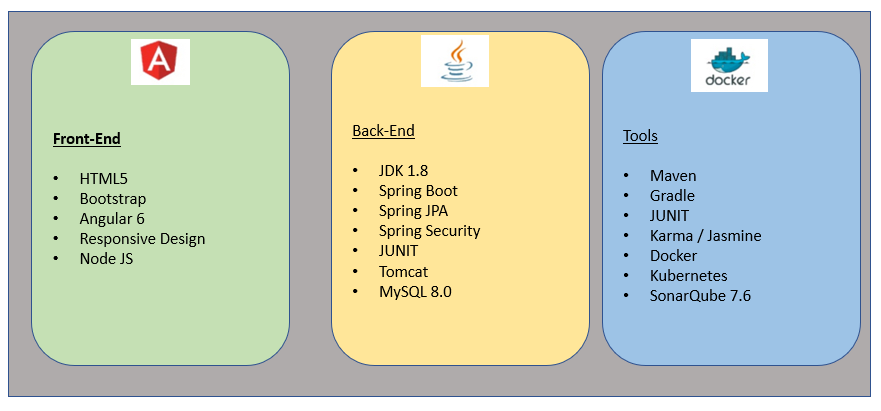
Flow execution steps:

1. User Selects option smart form service from UI which calls smart from service API
2. Smart Form API calls Smart Form services
3. Smart Form services retrieve configuration details and orchestrate system services calls in sequence configured for this use case
4. Form validation service performs basic validations on form input data.
5. Business rules are performed on the input and if any failure happens it will be logged and notified
6. Notification service will be invoked to send notification
7. Audit log service performs logging
8. Raise remedy ticket for each form submission & get remedy ID.
9. Insert worklist entry for every form submission
10. Escape Valve is present so user can use at any point he feels he cannot proceed further.
11. Smart input processing saves data in automation database

### SSO



## Technical Stack



### DVMs

### Cross References

### Error Handling

# Open and Closed Issues

1. Add open issues that you identify while writing or reviewing this document to the open issues section. As you resolve issues, move them to the closed issues section and keep the issue ID the same. Include an explanation of the resolution.  
     
   When this work product is complete, any open issues should be transferred to the project- or process-level Issue Log (PJM) and managed using a project level Issue Form (PJM). In addition, the open items should remain in the open issues section of this work product, but flagged in the resolution column as being transferred.

## Open Issues

| **Number** | **Description** | **Date Raised** | **Priority** | **Owner** | **Action** | **Target Date** |
| --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |
| **1.** |  |  |  |  |  |  |
|  |  | | | | | |
| **2.** |  |  |  |  |  |  |
|  |  | | | | | |
| **3.** |  |  |  |  |  |  |

**Priority:**  H= High; M=Medium, L=Low

## Closed Issues

| **Number** | **Description** | **Date Raised** | **Priority** | **Owner** | **Resolution** | **Date Closed** |
| --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  | | | | | |
|  |  |  |  |  |  |  |

**Priority**: H= High; M=Medium, L=Low