

Project Design Design

Design document for services discovered at Family Dept (eGov, Karnataka)



Version : V0.4

**Document Control**

|  |  |
| --- | --- |
| **Document Information** | |
| *Document Name* | *Project Design Document* |
| *Project Name* | *Service Implementation Prilims and Design Proposal at Family DB* |
| *Client* | *Department of e-Governance, Government of Karnataka* |
| *Document Author(s)* | *Prashanth Nuggehalli Chandrashekar* |
| *Document Version* | *V1.0* |
| *Document Status* | *Work In Progress (W.I.P)* |
| *Date Last Update* | *12th July 2021* |
| *Date Released* | *Not yet released with official review* |

**Document Edit History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Date** | **Additions/Modifications** | **Prepared / Revised by** |
| V0.1 | 23rd Mar 2021 | Initial Draft | Prashanth Chandrashekar |
| V0.2 | 6th May 2021 | Added diagrams specific to FDB Integration | Prashanth Chandrashekar |
| V0.3 | 25th June 2021 | Updated after demo to ACS / PD | Prashanth Chandrashekar |
| V0.4 | 12th June 2021 | Updated eJanma / RTO service details | Prashanth Chandrashekar |
|  |  |  |  |

**Document Review/Approval History**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **#** | **Name**  ***(Prepared By)*** | **Organization/Title**  ***(Approved By)*** | **Date**  ***(Approval Date)*** | **Comments**  ***(Description)*** |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

*This page is intentionally left blank*

**Table of Contents**

[**Introduction**](#_gjdgxs) **11**

[Overview](#_4d34og8) 11

[Quick Summary](#_2s8eyo1) 12

[Organization Background](#_17dp8vu) 12

[**Key Challenges**](#_49x2ik5) **14**

[Observations](#_2p2csry) 14

[Recommendations](#_1v1yuxt) 14

[**Discovery**](#_2lwamvv) **14**

[Discovery outcome](#_3l18frh) 15

[**Architecture**](#_2dlolyb) **18**

[Narrow View](#_sqyw64) 19

[Detailed View](#_4bvk7pj) 19

[Architecture Components](#_1664s55) 19

[Proposed High Level Objective](#_kgcv8k) 21

[**Design**](#_2nusc19) **24**

[Design Considerations](#_1302m92) 24

[Common Components](#_3ep43zb) 25

[Message Logging](#_4du1wux) 25

[Message Auditing](#_3s49zyc) 26

[Exception Handling](#_45jfvxd) 29

[Notifications](#_3jtnz0s) 31

[SMTP Details : << TBD >>](#_2y3w247) 33

[Use Case Implementation](#_1d96cc0) 33

[Use case 1 : Straight passthrough](#_3x8tuzt) 33

[Service Detail](#_2ce457m) 33

[Operation Sequence](#_3bj1y38) 34

[Source](#_4anzqyu) 35

[Destination](#_2pta16n) 35

[Inbound Payload](#_14ykbeg) 36

[Outbound Payload](#_3oy7u29) 36

[Participating APIs and their endpoints](#_243i4a2) 36

[Outbound Connector Detail](#_338fx5o) 1

[Request Field Mapping](#_42ddq1a) 37

[Response Field Mapping](#_wnyagw) 38

[Error Messages](#_1vsw3ci) 38

[Business Exception](#_4fsjm0b) 39

[Use case 2 : Filter and Render](#_2uxtw84) 39

[Service Detail](#_1a346fx) 39

[Operation Sequence](#_2981zbj) 40

[Source](#_38czs75) 41

[Destination](#_1nia2ey) 41

[Inbound Payload](#_47hxl2r) 42

[Outbound Payload](#_2mn7vak) 42

[Participating APIs and their endpoints](#_11si5id) 42

[Outbound Connector Detail](#_20xfydz) 43

[Request Field Mapping](#_302dr9l) 43

[Response Field Mapping](#_3z7bk57) 43

[Error Messages](#_thw4kt) 44

[Business Exception](#_3dhjn8m) 44

[Use case 3 : Fetch, Merge and Enrich](#_1smtxgf) 44

[Service Detail](#_4cmhg48) 44

[Operation Sequence](#_16x20ju) 46

[Source](#_261ztfg) 47

[Destination](#_l7a3n9) 47

[Inbound Payload](#_356xmb2) 47

[Outbound Payload](#_1kc7wiv) 48

[Inbound Connector Detail](#_44bvf6o) 1

[Outbound Connector Detail](#_ymfzma) 50

[Request Field Mapping](#_1xrdshw) 50

[Response Field Mapping](#_2wwbldi) 1

[Error Messages](#_3w19e94) 51

[Business Exception](#_2b6jogx) 52

[Use case 4 : Notification Over Pub Sub](#_qbtyoq) 52

[Service Detail](#_3abhhcj) 52

[Operation Sequence](#_49gfa85) 54

[Source](#_13qzunr) 55

[Destination](#_3nqndbk) 55

[Inbound Payload](#_22vxnjd) 55

[Outbound Payload](#_i17xr6) 55

[Inbound Connector Detail](#_320vgez) 55

[Outbound Connector Detail](#_415t9al) 55

[Request Field Mapping](#_vgdtq7) 1

[Response Field Mapping](#_1ulbmlt) 56

[Error Messages](#_2tq9fhf) 56

[Business Exception](#_18vjpp8) 56

[Use case 5 : Notification Over Email](#_3sv78d1) 56

[Service Detail](#_280hiku) 56

[Operation Sequence](#_375fbgg) 58

[Source](#_46ad4c2) 59

[Destination](#_2lfnejv) 59

[Inbound Payload](#_10kxoro) 59

[Outbound Payload](#_3kkl7fh) 60

[Inbound Connector Detail](#_1zpvhna) 60

[Outbound Connector Detail](#_2yutaiw) 60

[Request Field Mapping](#_3xzr3ei) 1

[Response Field Mapping](#_sabnu4) 61

[Error Messages](#_1rf9gpq) 61

[Business Exception](#_4bewzdj) 61

[**API / Service NFR**](#_2qk79lc) **61**

[Security](#_15phjt5) 61

[Auditing](#_24ufcor) 62

[Additional NFRs](#_jzpmwk) 62

[Performance](#_434ayfz) 64

[Application monitoring](#_3hej1je) 66

[Cloudhub](#_1wjtbr7) 66

[Batch processing](#_4gjguf0) 66

[Build and deployment solution](#_2vor4mt) 66

[Code Versioning](#_1au1eum) 66

[Git](#_3utoxif) 66

[Continuous Integration](#_29yz7q8) 67

[Jenkins](#_p49hy1) 67

[Tools and technologies](#_393x0lu) 67

[Development Tools](#_1o97atn) 67

[Anypoint Studio](#_488uthg) 67

[Postman](#_12jfdx2) 68

[Putty / FileZilla](#_i9cwu0ttqkdm) 69

[Text Editor](#_gtnh0h) 70

[Technologies](#_3zy8sjw) 70

[Java / JDK 1.8](#_2f3j2rp) 70

[Mulesoft Runtime](#_u8tczi) 71

[CloudHub](#_3e8gvnb) 71

[Dataweave](#_1tdr5v4) 71

[Maven](#_4ddeoix) 71

[SonarQube](#_2sioyqq) 71

[Shell Scripting](#_17nz8yj) 71

[Environment](#_3rnmrmc) 71

[Non Production - Development](#_26sx1u5) 71

[Development](#_ly7c1y) 71

[Testing](#_35xuupr) 71

[Production](#_1l354xk) 71

[Production Active](#_452snld) 71

[Production Passive (DR)](#_2k82xt6) 72

[**Testing**](#_3jd0qos) **72**

[Testing Approach](#_1yib0wl) 72

[Functional testing](#_4ihyjke) 72

[Fetch and Render](#_dzakj25azktp) 72

[Fetch, Filter and Render](#_nex5ad7m3pu) 72

[Fetch, Merge and Render](#_hm2atn8hk7cj) 73

[Notify over PUB-SUB](#_7cm5ck5g37jm) 73

[Notify over Email](#_j6r285k7tsng) 73

[Non Functional testing](#_2xn8ts7) 73

[Testing Tools](#_1csj400) 73

[Mocking Frameworks](#_3ws6mnt) 74

[**Appendix**](#_2bxgwvm) **74**

# Introduction

Information Technology (IT) is no longer only the steward of middle- and back-office functions. Citizen facing initiatives and digital transformation will drive on how to manage the Government initiatives, programs, governance that bring value to citizens. The explosion in connected devices and platforms, combined with the abundance of data from field devices and the rapidly changing technology landscape, has made it imperative for everyone to quickly adapt to new products and technologies and move from the physical world to a digital one. At the core of each of these challenges is the need for a new level of connectivity.

To thrive in this era, organizations must seamlessly integrate applications, data and devices and it is those organizations that are able to embrace these challenges and will be able to provide state of facilities to the citizens.

The State has witnessed tremendous growth and e-Governance has played a vital role in the socio-economic development of the State, particularly in the rural areas and in agriculture and allied sectors. Many IT projects across departments have been implemented to cater to Citizen Centric services and many more are planned to be rolled out. To make this a success the following objectives must be achieved.

* Understand the Department's businesses and its processes.
* Enable departmental applications to exchange data and provide seamless services to citizens
* Transform it through Business Process Reengineering
* Digitise the reengineered citizen centric business process

## Overview

The purpose of this document is to provide the design of best possible service implementation pattern for identified service as an outcome of the workshop. The 3 use cases identified are based on the operational type and on specific patterns involved in the integration. This also serves as a guideline for some of the best practices followed by architects and developers based while implementing Mulesoft services.

## Quick Summary

The workshop #2 conducted with the Family DB dev team has yielded in discovering different services that can be categorized based on the operational type. In this phase of discovery and development, the integration team (ESB team) shall be designing the services they are discussed in detail under various subheadings in this document.

1. Straight Passthrough
2. Filter and Render
3. Fetch, Merge and Enrichment
4. Notification over PubSub
5. Notification over Email

## Organization Background

The Government of Karnataka is a pioneer and exemplary model in the governance of Information and Communication Technology (ICT) and is active in implementing e-governance initiatives in the country. The Centre for e-Governance (CeG)established in the year 2006 by the Department of Personnel and Administrative Reforms (DPAR), Government of Karnataka. It is an autonomous and independent body set up for conceptualizing, implementing and monitoring various e-Governance initiatives in Karnataka. The Centre has been established under the Karnataka Societies Registration Act 1960 and its area of operation is spread across the entire State of Karnataka.

The Centre for e-Governance is unique as it is placed in the Department of Personnel & Administrative Reforms (DPAR) which directly reports to the Chief Minister of Karnataka.

Over the past few years, CeG has contributed immensely by helping the citizens of Karnataka to reap the benefits of Information and Technology (IT). CeG has also spread its wings to various innovative projects with the help of e-Governance to ensure that the common man gets access to the benefits and schemes rolled out by the State Government.

CeG has also helped the State Government to set up two state-of-the-art data centers, MPLS technology-based Karnataka State Wide Area Network (KSWAN) and core applications such as e-procurement and HRMS.

**CeG at its Core**

The Centre for e-Governance is a Society setup by the Department of Administrative Reforms (DPAR), Government of Karnataka, under the Karnataka Societies Registration Act 1960. The area of operation of the society extends to the whole of Karnataka State. The Society was established in the year 2006. The Centre for e-Governance is an autonomous and independent body set up for conceptualizing, implementing and monitoring of various e-Governance initiatives in Karnataka.

**Projects under Centre for e-Governance**

Projects managed by Centre for e-Governance (CeG) are broadly classified as follows:

|  |  |
| --- | --- |
| **Core Infrastructure Projects** | These projects namely SDC, KSWAN, SecLAN, M1, Mailing Service and SSDG, etc., act as underlying foundation or pillars on which all the e-Governance projects in the state can run and operate. |
| **Core Enterprise Applications** | The enterprise applications such as HRMS, e-Procurement and Sachivalayavahini that cuts across various Departments, Users, Citizens, etc. |
| **Citizen Delivery Services** | Offers UID enrolment to residents of Karnataka State |
| **Capacity Building** | To build specific competencies and expertise among Government personnel in implementing and managing e-Governance Projects.  To assist Government Departments in Planning and implementing e-Governance initiatives. |

Table 1

# Key Challenges

## Observations

To provide integrated services to citizens through a free flow of information, and to usher in an era of good governance, characterized by efficiency, effectiveness, transparency, innovation and foresight. The different dimensions of the vision are described below:

* Departments and government agencies interoperate with ease and provide integrated services to citizens and businesses.
* Citizens and businesses will have a seamless and smooth interface with the government.
* This demands to have Enterprise service bus (ESB) as technology stacks with the government.
* Providing common infrastructure and application services will enable one government vision
* Faster time to deploy GoK services by enabling quick integration
* Increased productivity and efficiency by providing access to single source of data
* Readiness to create a richer portfolio of services
* Ability to create cross-Governmental services through interoperability, service discovery, and invocation
* The medium of paper to be minimized in all G2C, C2G, G2B, B2G, and G2G interactions by listing all the services through ESB Portal

## Recommendations

* **Technology** - CEG recommends MuleSoft Anypoint Platform as the Integration layer to address all the requirements mentioned in the RFP
* **Product Support** - The product support will be directly provided by MuleSoft on 9\*5 business hours for the current scope. This support is recommended for the initial period and can always be upgraded to a ‘Platinum level’ to achieve high availability and 24\*7 support at an additional cost amounting to uplift of 35% of the overall platform cost
* **CEG** ‘Rapid Launch’
  + Discovery and Reference Architecture
  + Platform Setup and Configuration
  + Mentor CeG team and assist them to deliver 5 sample integration use cases
* **Professional Services** - Professional services will be carried out by CEG in an onsite / offsite model as required
* **Technical Governance** - CEG to provide technical governance and help implement MuleSoft Best practices on design, development and deployment
* **Enablement** - CEG will facilitate training and knowledge transfer to CeG teams as required

# Discovery

With a series of discussions with department stakeholders (Mentioned in the Stakeholder list aforementioned) and with a senior program management group, we have discovered a set of activities that help CeG to implement ESB across the organization. We start with defining the AS IS picture and will eventually arrive at the proposed ESB system using MuleSoft technology. These discoveries lead us to conduct workshops, training and set up best practices to implement ESB in the most efficient way that brings noticeable changes to the end ushers.

The AS IS diagram shown in 1 illustrates the clutter that is caused with the point-to-point communication across. Easely this leads to points mentioned below.

* Messy Integration
* Rigid Solution
* Rise of security issues with patch updates
* Incompatibilities with connections as the system integration grows
* Changes are often out of control
* Locked to suppliers and their technical debt
* Primary application upgrade becomes difficult
* Complex point-to-point integration leads to instability
* Discontinued support by vendor is born by organization

*✍* ***Note*** *: Due to lack of complete IT landscape diagram / document and due to security reasons each block is mentioned / illustrated as a department. Departments can be Family, Healthcare, Finance etc..*

Discovery activity included talking to the program director who understands the service priority and service landscape. Whilst the objective is to integrate the existing services with various departments, the immediate attention is on Family DB (Highlighted in Red), and based on this 3 use cases are identified as proof of concept. This pattern is going to be standard across the integration but not limited to the same. While this document helps in understanding the best practices and guidelines to be followed by the development team, this document hosts important sections for the users / developers to refer to during post development.

## 

## Discovery outcome

Each service identified as a part of discovery is explained in detail in the below mentioned sections. Some of the sections are common to each service while others are very specific

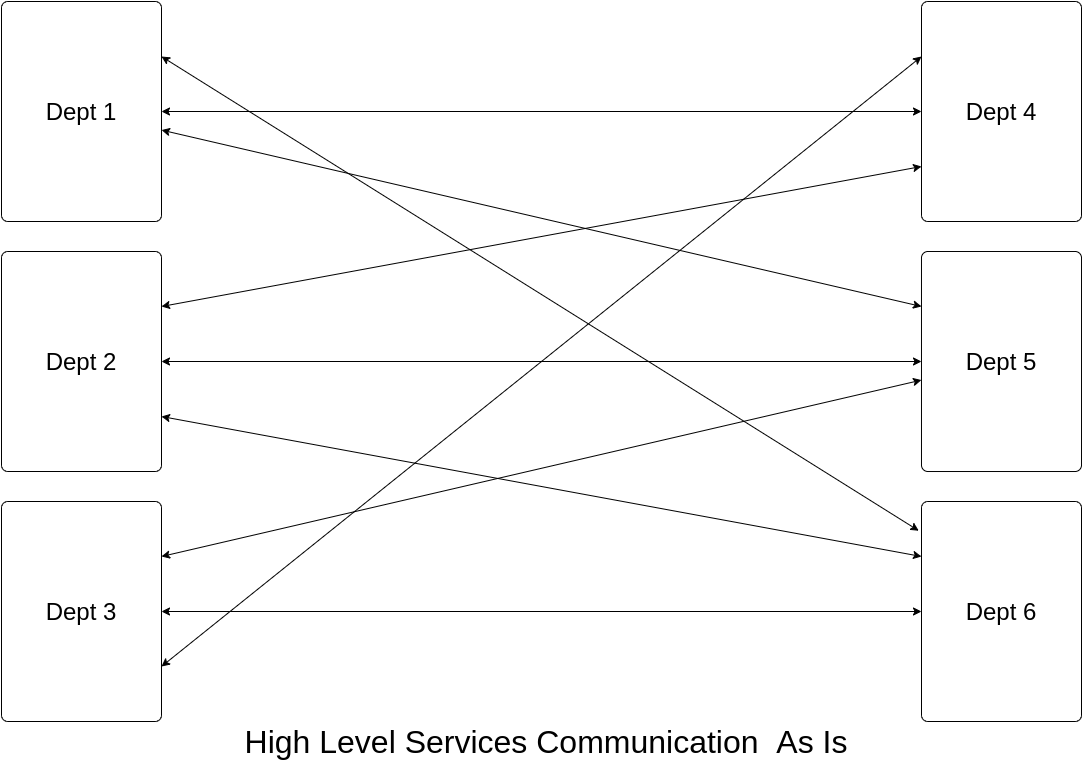


Figure 1

1. Use case 1 : Straight Passthrough : In this type
   1. Mulesoft service listens at a http connector (Details are provided in table )
   2. Caller (Dept) invokes the Mulesoft service
   3. Mulesoft service invokes another existing external server (Family DB)
   4. Mulesoft service receives the response,
   5. Mulesoft service sends the response back to the caller in the format that it receives from external service (Family DB)
   6. On error send appropriate message as per section

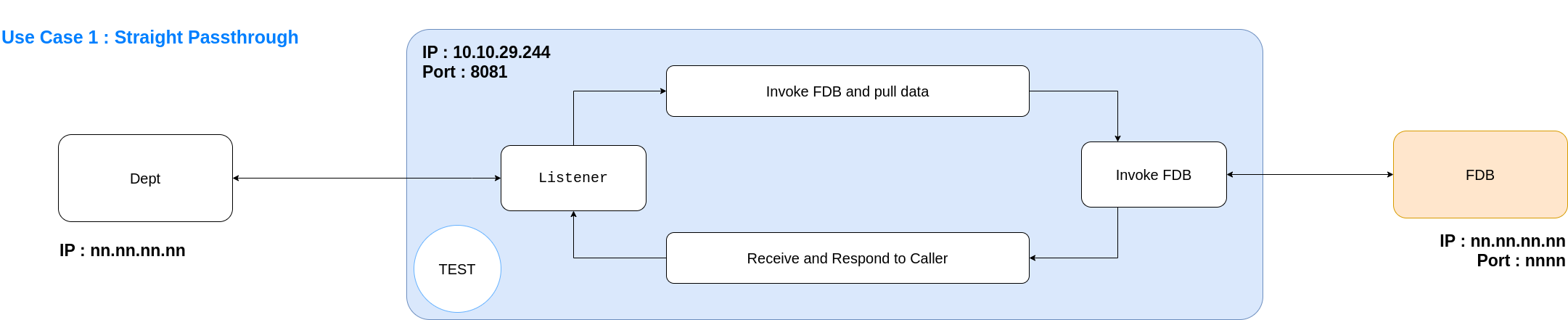


Figure 2

1. Use case 2 : Filter and Render
   1. Mulesoft service listens at a http connector (Details are provided in table )
   2. Caller (Dept) invokes the Mulesoft service
   3. Mulesoft service invokes another existing external server (Family DB)
   4. Mulesoft service receives the response
   5. Mulesoft service receives the response filters and maps the request that is required by caller as per mapping document
   6. Mulesoft service sends the response back to the caller in the format that it receives from external service (Family DB)
   7. On error send appropriate message as per section

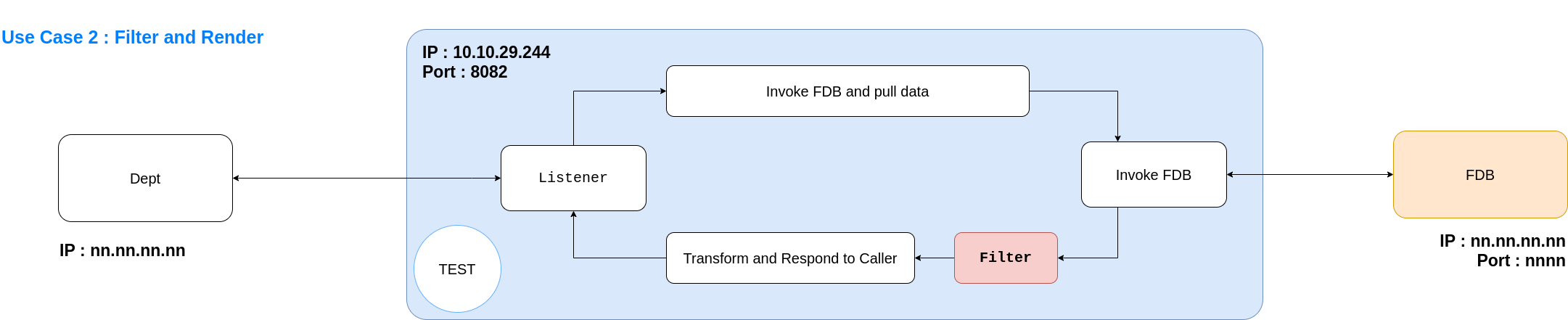


Figure 3

1. Use case 3 : Fetch,Merge and Enrich
   1. Mulesoft service listens at a http connector (Details are provided in table )
   2. Caller (Dept) invokes the Mulesoft service
   3. Mulesoft service invokes another existing external server (Family DB)
   4. Mulesoft service receives the response
   5. Mulesoft service invokes another external service (FCS)
   6. Mulesoft receives the response from FCS
   7. Mulesoft intercepts both the responses received from step d and step f, Merges the responses and
   8. Mulesoft maps the output of Merge (as in step g) to a format that the caller needs
   9. Mulesoft service sends the response back to the caller
   10. On error send appropriate message as per section

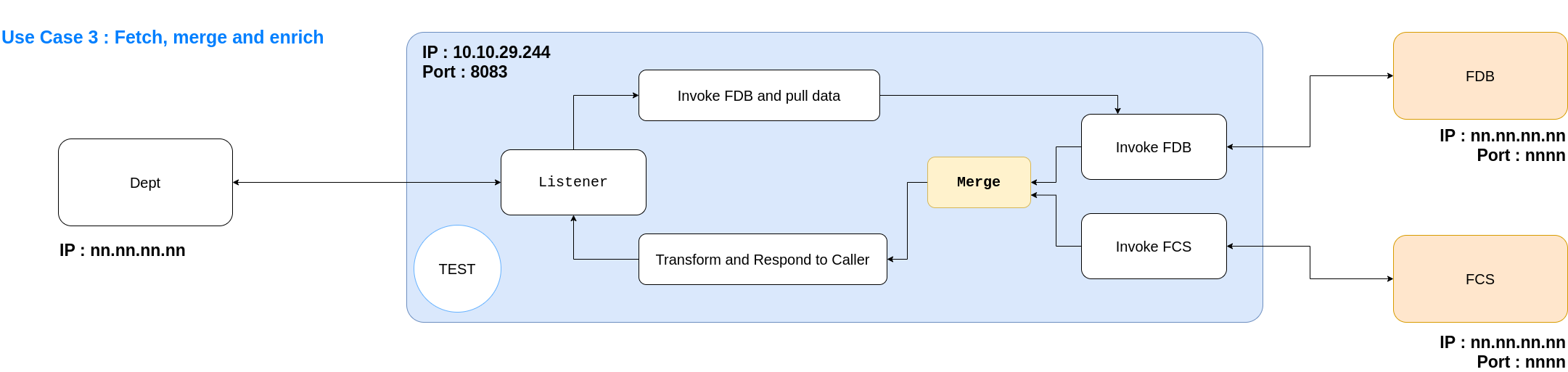


Figure 4

# Architecture

## Narrow View

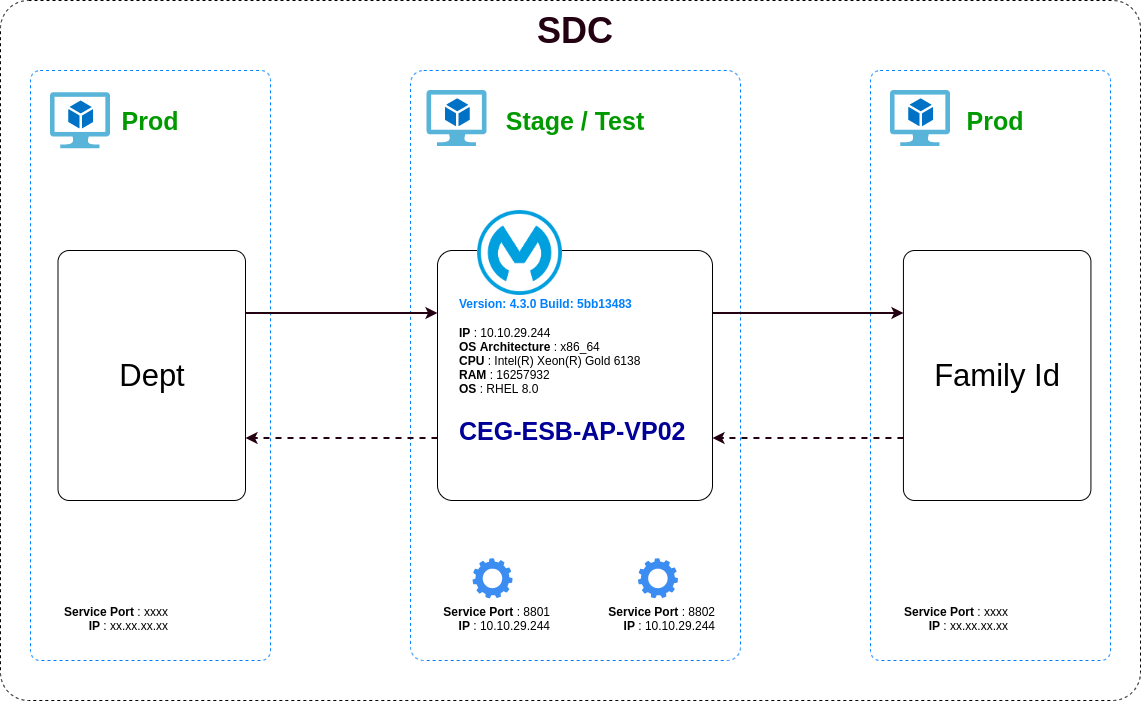


Figure 5

## Detailed View

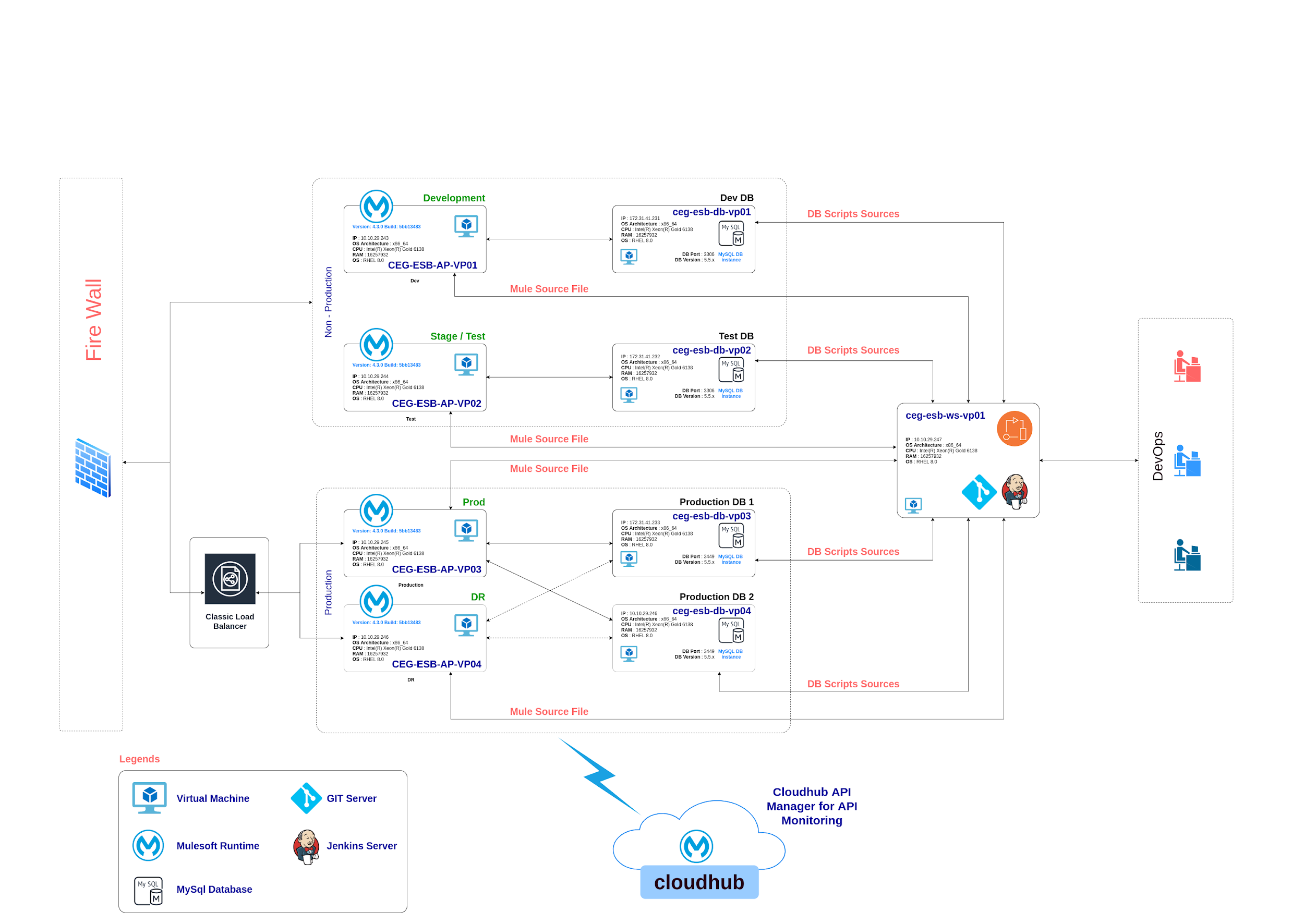


Figure 6

## Architecture Components

|  |  |  |
| --- | --- | --- |
| **Name** | **Technical Details** | **Description** |
| **Load Balancer** | Physical LB provided by SDC | Details of the general Load Balancer (mostly Classic LB) shall be updated once the details are provided by SDC |
| **Mule Runtime on Dev** | Hostname : CEG-ESB-AP-VP01  IP : 10.10.29.243  RAM : 16 GB  Mule Core : 2 Core NP  VM Core : 2 Cores  OS : RHEL 8.2  JVM : openjdk-1.8.0.265.b01-0.el8\_2.x86\_64 | Mule server on which the development of API and services are carried out before rolling out to Stage / Test server |
| **Mule Runtime on Stage / Test** | Hostname : CEG-ESB-AP-VP02  IP : 10.10.29.244  RAM : 16 GB  Mule Core : 2 Core NP  VM Core : 2 Cores  OS : RHEL 8.2  JVM : openjdk-1.8.0.265.b01-0.el8\_2.x86\_64 | Mule server on which the Testing of API and services are carried out before rolling out to Production server. |
| **Mule Runtime on Prod 1 (Active)** | Hostname : CEG-ESB-AP-VP03  IP : 10.10.29.245  RAM : 16 GB  Mule Core : 1 Core Prod  VM Core : 2 Cores  OS : RHEL 8.2  JVM : openjdk-1.8.0.265.b01-0.el8\_2.x86\_64 | Production Server that hosts the live API and Services |
| **Mule Runtime on Prod 2 (Passive)** | Hostname : CEG-ESB-AP-VP04  IP : 10.10.29.246  RAM : 16 GB  Mule Core : 1 Core Prod  VM Core : 2 Cores  OS : RHEL 8.2  JVM : openjdk-1.8.0.265.b01-0.el8\_2.x86\_64 | Production Server that hosts the live API and Services |
| **Cloudhub** | Mulesoft Cloudhub @ <https://anypoint.mulesoft.com/login/> | Mulesoft Provided Cloudhub to monitor the API state. This is also known as API Manager |
| **MySQL Database for Dev** | Hostname : ceg-esb-db-vp01  IP : 172.31.41.231  RAM : 16 GB  MySQL : 5.5.x  VM Core : 2 Cores  OS : RHEL 8.2 | Standalone Database Server used for auditing and logging and for development activities. |
| **MySQL Database for Test** | Hostname : ceg-esb-db-vp02  IP : 172.31.41.232  RAM : 16 GB  MySQL : 5.5.x  VM Core : 2 Cores  OS : RHEL 8.2 | Standalone Database server used for auditing and Logging and for Testing activities |
| **MySQL Database 1 for Prod** | Hostname : ceg-esb-db-vp01  IP : 172.31.41.233  RAM : 16 GB  MySQL : 5.5.x  VM Core : 2 Cores  OS : RHEL 8.2 | Standalone Database Server for auditing and logging and used as a production server. |
| **MySQL Database 2 for Prod** | Hostname : ceg-esb-db-vp01  IP : 172.31.41.234  RAM : 16 GB  MySQL : 5.5.x  VM Core : 2 Cores  OS : RHEL 8.2 | Standalone Database Server for auditing and logging and used as production server replica of Database 1 |
| **DevOps Server** | Hostname : ceg-esb-ws-vp01  IP : 10.10.29.247  RAM : 16 GB  VM Core : 2 Cores  OS : RHEL 8.2  JVM : openjdk-1.8.0.265.b01-0.el8\_2.x86\_64 | Dedicated VM for hosting Devops Server |

Table 2

## Proposed High Level Objective

One of the objectives is to establish an ESB that can seamlessly integrate various existing and new services enabling each department to be ready for digital convergence.

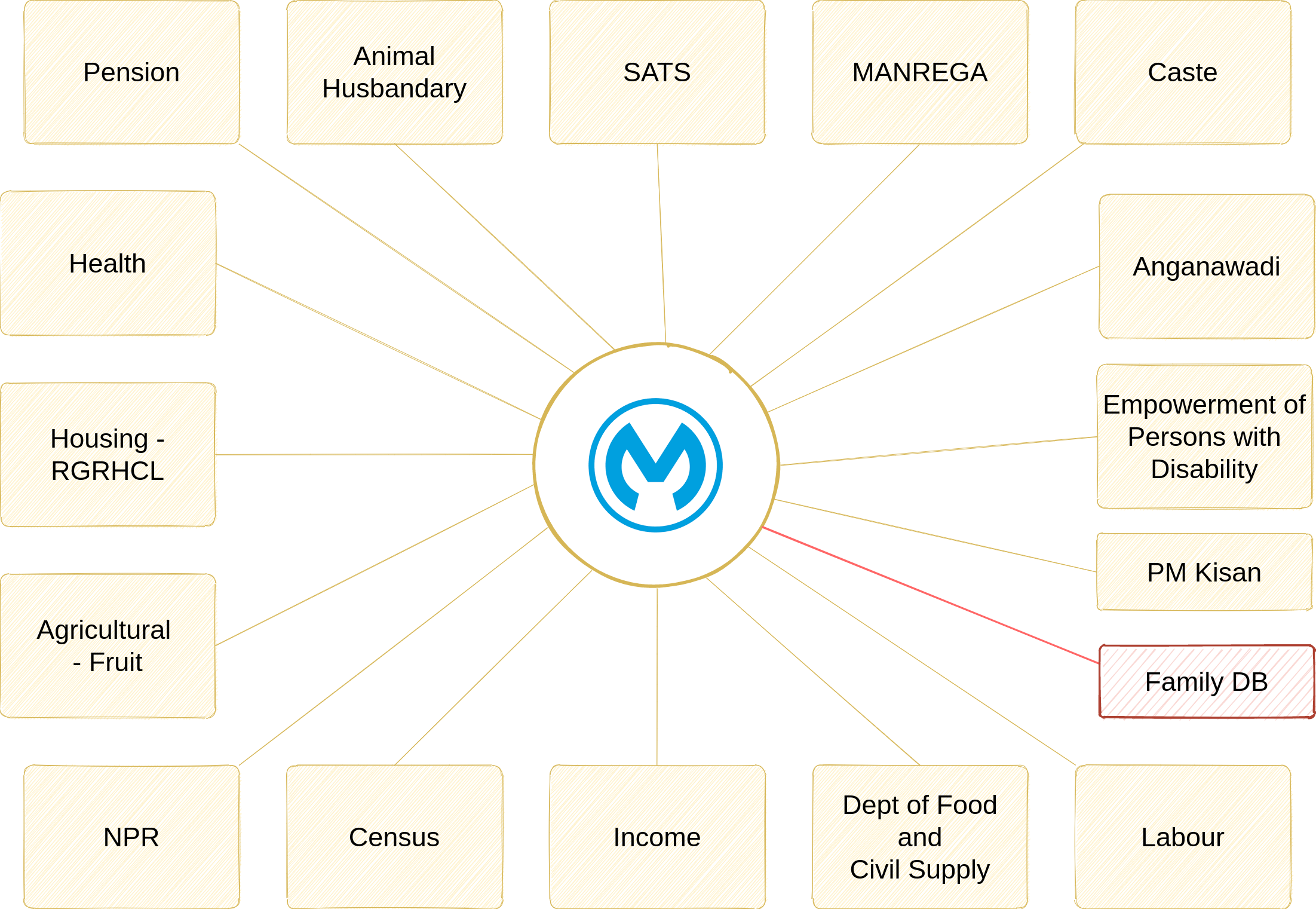


Figure 7

The proposed system is designed and referenced by keeping three different types of application / system integration in mind in the organizations IT landscape. Protected, Private and Public network system. Please refer to figure 2.

|  |  |  |
| --- | --- | --- |
| **Protected** | **Private** | **Public** |
| There are systems that interact within a defined intranet. In most of the cases they are among the systems within the State Data Center. They are governed by IP provisioning at the VM level or at the Application Level. They often talk over HTTP and exchange data, process data within the SDC. Communication is established among the applications, server or DB based on the internal IP of the individual system. | These are the systems that interact over the internet. The communication happens between SDC and a certified external network (WAN) outside over the internet. IP whitelisting, inbound and outbound traffic is controlled at Firewall around SDC. External IPs are bound based on the service need to meet business as usual activities. Typically this can include communication between SDC and University networks or with NIC data centers etc. | These are systems that communicate with each other over the public internet. They are governed by SSL certificates and are open to the general public. An application API is exposed to the general public to consume or use and traffic is allied to flow into SDC for data query, input or as a message. Each API exposed over the public net is governed by security policies. |

Table 3

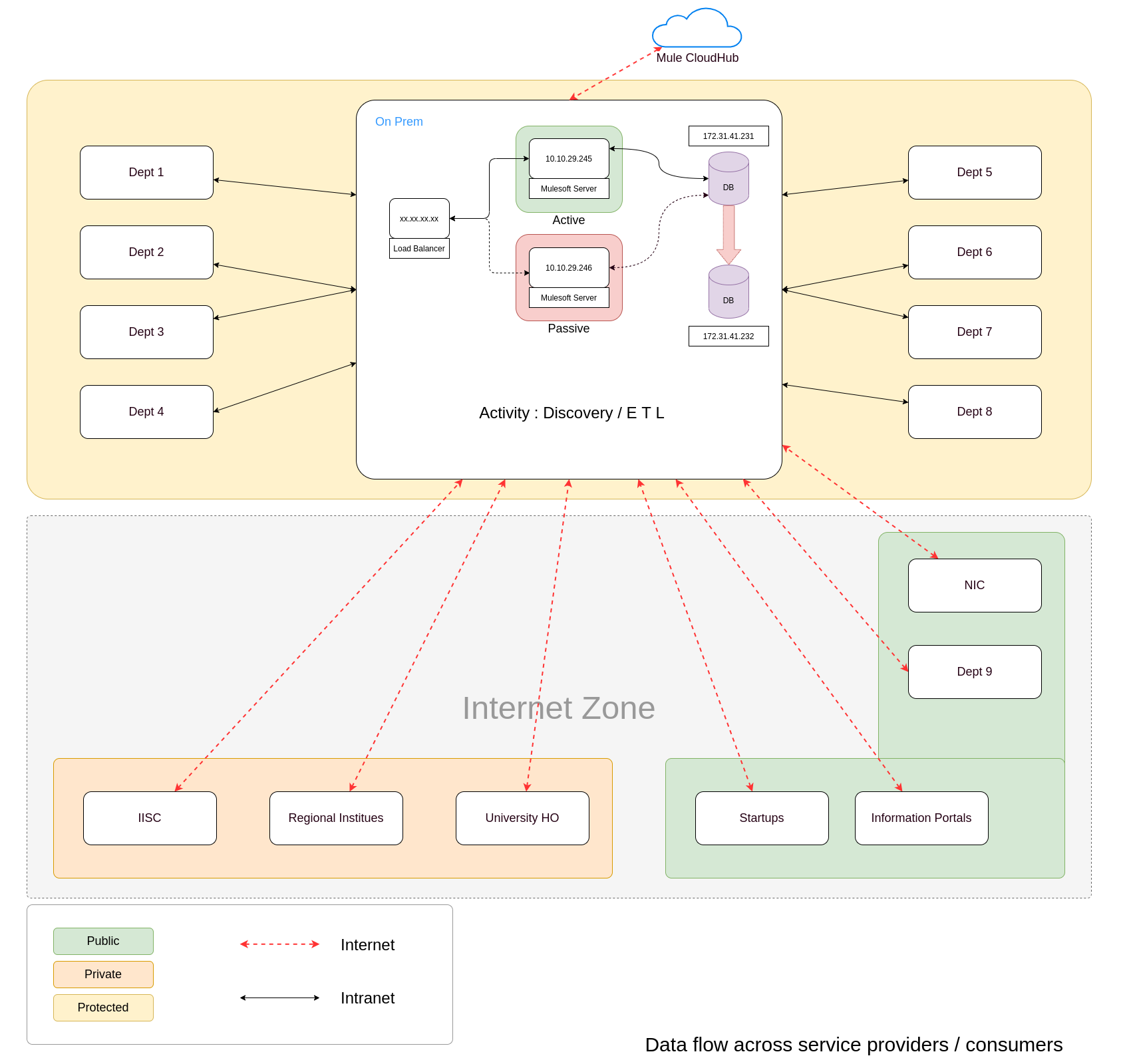


Figure 8

In order to achieve the PoC’s we are proposing above architecture with respective teams involved in this conducting following activities.

* Source EndPoint (Details) provided Program Management Group
* Payload Samples (Some are shared, more details are needed as per the questionnaire mentioned in the reference section)
* Translation Rules
* Transformation Rules
* ESB Service Design
* ESB Service Implementation (Minimal as per RFP)
* ESB Service Deployment
* Monitoring ESB Logs / Error Handling
* Infrastructure Team Support for IP configuration / Enablements
* CI/DC Pipeline for Deployments
* Server Side Scripts for Starting and Stopping Mule runtime
* Stress / Load Testing Team for conducting Request Analysis or TPS analysis
* Capacity Planning Report after the Analysis

# Design

Each service identified as a part of discovery is explained in detail in the below mentioned sections. Some of these sections are common to each service while others are very specific.

## Design Considerations

Following are general guidelines and assumptions considered to begin the ESB implementation design.

* Mule Servers shall reside within SDC
* Mule Servers shall be installed over Virtuele Machines
* Selection of OS is RHEL
* Audit Logs are Stored in MySQL database
* The Source code is maintained in GIT Server within SDC
* The build pipeline for mule source is over Genkins
* 3 Environments are chosen for development, testing and production operations
* Cloudhub is used for API monitoring

## Common Components

Common components help implement services to log, audit, notify and handle exceptions in a most organized way. Below mentioned diagram is common for both audit and logging activities.

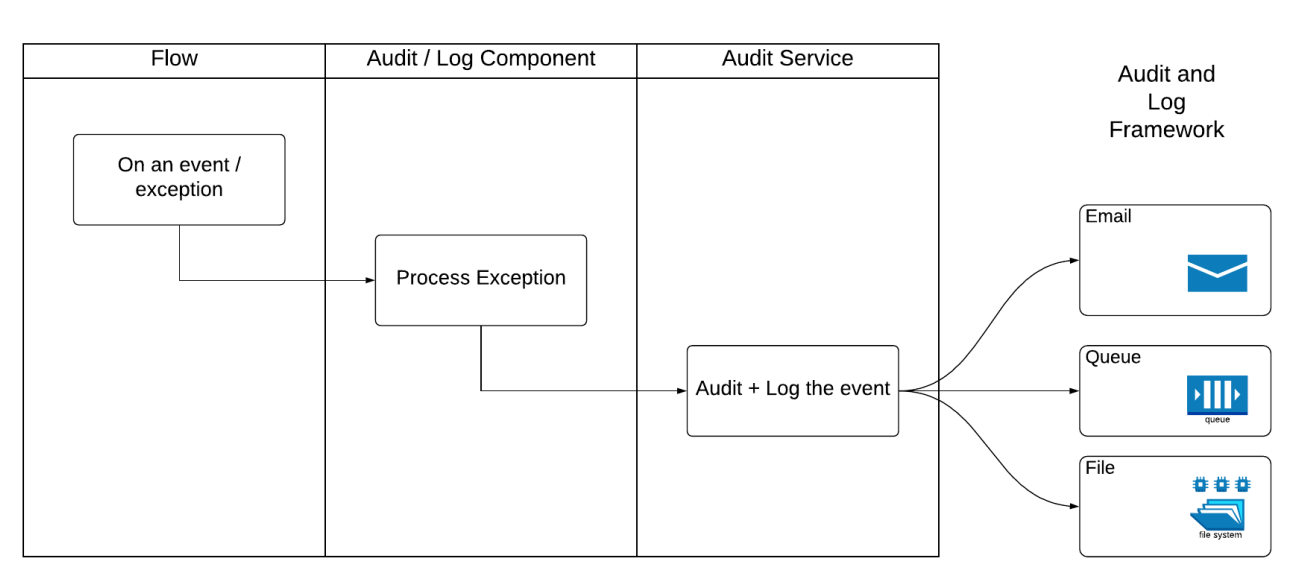


Figure 9

**Flow**: This is a normal flow or sub flow where the exception / event or a log is intercepted.

**Audit / Log Component**: A common code that can enrich captured events or allow a direct pass through.

**Audit Service**: An isolated service that is usually deployed at the system layer and monitored separately. This has an HTTP connector that can scale multiple threads and allows it to function independently.

### Message Logging

CEG has developed a framework for logging asynchronously that effectively and efficiently meets the logging requirements needed by CeG. This follows a simple design as mentioned in Fig

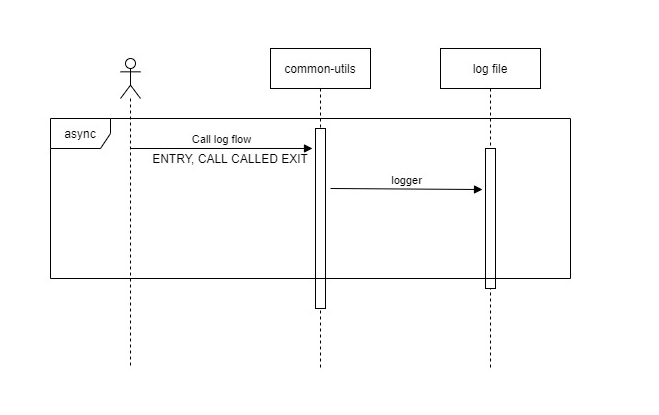


Figure 10

Default logging (Mulesoft Logger).

|  |
| --- |
| CEG recommends to use this feature very carefully as this can hinder the overall performance. It is recommended to use the audit / log framework as shown in figure 9 |

You can configure what gets logged, where it gets logged, and how by editing a configuration file at project level. By creating a configuration file, you can define what kind of messages to log, in what way (asynchronously or synchronously), and where they get logged (such as to the console, to disk, to an endpoint or to a database).

### Message Auditing

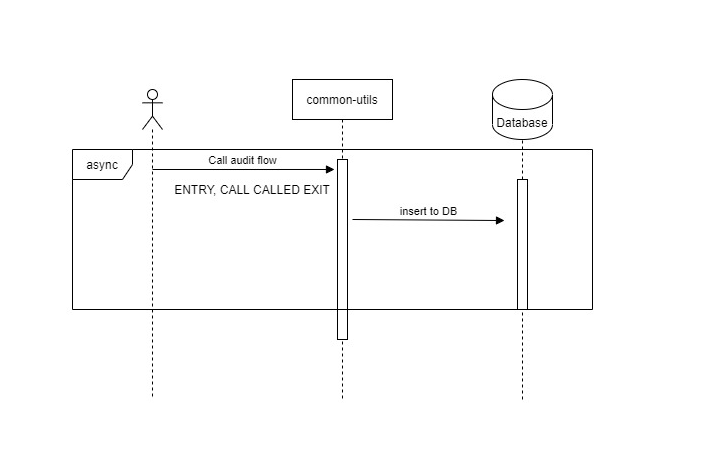
CEG recommends the aforementioned diagram 9 framework for auditing and logging. This gives a better control on the logs making the audit processing an isolated service that does not share resources with other integration flow hence contributing to the overall application performance.

What qualifies for audit ?

|  |  |  |
| --- | --- | --- |
| **#** | **Attributes** | **Yes / No** |
| **1** | Inbound Message | As per custom style |
| **2** | Outbound Message | As per custom style |
| **3** | Payload \*\* | As per custom style |
| **4** | Sensitive Data \* | As per custom style |
| **5** | Flow Name | As per custom style |
| **6** | Date and Time of Transaction | As per custom style |
| **7** | Error Codes | As per custom style |
| **8** | Error Messages | As per custom style |

Table 4

|  |
| --- |
| \* *Categorised based on org’s security policy / data policy etc...*  \*\* *Sometimes, during critical error tracking Payloads might be required to store temporally. There are many other factors that influence auditing, this section will be updated as and when needed.* |

*Figure 11*

Sequence Diagram shows the audit call is an asynchronous call and the common utility class takes the call and pushes the audit messages to the Database.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **#** | **Field Name** | **Attribute Name (Variable)** | **Datatype** | **MySQL** |
| **1** | Record Create Time | recordCreateTime | Time Stamp | TIMESTAMP |
| **2** | API Reference | apiReference | String | VARCHAR(100) |
| **3** | Flow Name | flowName | String | VARCHAR(100) |
| **4** | Message Id | messageId | String | VARCHAR(100) |
| **5** | Correlation Id | correlationId | String | VARCHAR(100) |
| **6** | Log Type | logType | String | VARCHAR(100) |
| **7** | Log Message | logMessage | String | LONGTEXT |
| **8** | Error Message | errorMessage | String | LONGTEXT |

Table 5

### Exception Handling

Exceptions are handled in combination with the audit framework as mentioned in 12 Whenever an exception or error occurs, this will be passed to a common component to process and normal execution continues.

CEG recommends keeping separate service to handle exceptions. The overview is provided in the below mentioned diagram. In all the services, Exception handling is based on 3 exception categories.

1. System Exceptions
2. Message Exceptions and
3. Business Exceptions

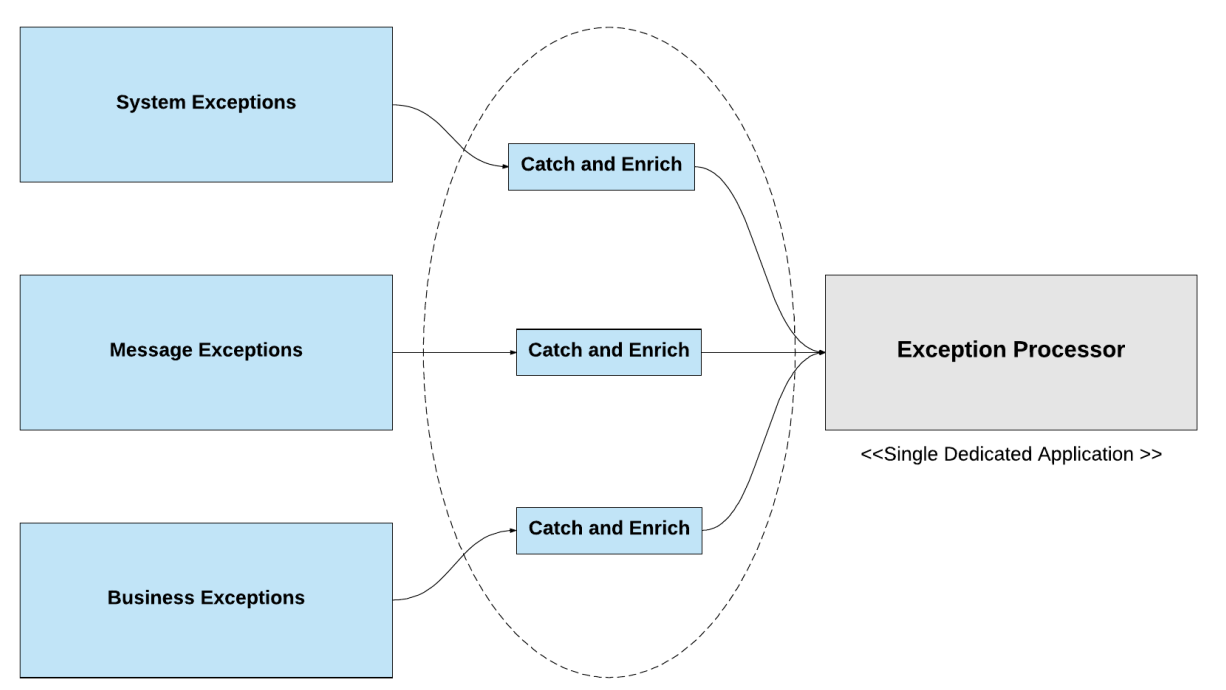


Figure 12

**System Exception**

Mule invokes a System Exception Strategy when an exception is thrown at the system-level (that is, when no message is involved, exceptions are handled by system exception strategies). For example, system exception strategies handle exceptions that occur:

* During application start-up
* When a connection to an external system fails

When a system exception occurs, Mule sends an exception notification to registered listeners, logs the exception, and — if the exception was caused by a connection failure — executes the reconnection strategy. System Exception Strategies are not configurable in Mule.

**Message Exception**

Mule invokes a Messaging Exception Strategy whenever an exception is thrown within a flow. When a message is processed through a Mule flow, an exception is thrown, normal flow execution stops. Mule transfers the message to the message processor sequence within the exception strategy. The diagram below illustrates what happens when a message throws an exception.



Figure 13

**Business Exceptions**

These are the exceptions that occur during

1. Message Transformation
2. Message Translation
3. Message Validation

### Notifications

Notification component uses SMTP connector to notify stakeholders or support team. In the event of any business incidents, the notifications are sent that hold basic information as mentioned in the table below.

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **Field Name** | **Attribute Name (Variable)** | **Datatype** |
| **1** | Record Create Time | recordCreateTime | Time Stamp |
| **2** | API Reference | apiReference | String |
| **3** | Flow Name | flowName | String |
| **4** | Message Id | messageId | String |
| **5** | Correlation Id | correlationId | String |
| **6** | Log Type | logType | String |
| **7** | Log Message | logMessage | String |
| **8** | Error Message | errorMessage | String |

Table 6

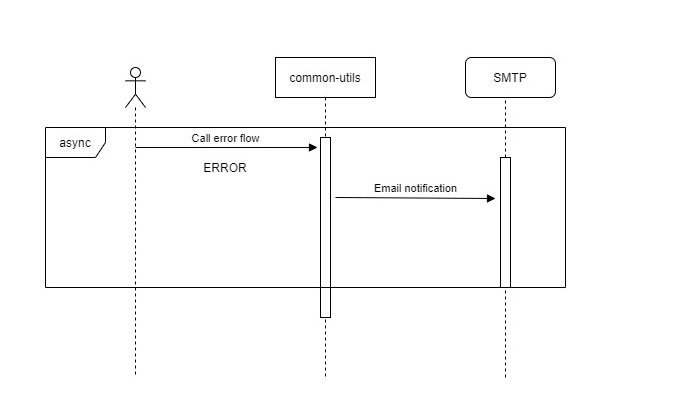


Figure 14

#### **SMTP Details :** << TBD >>

## Use Case Implementation

### Use case 1 : Straight passthrough

#### Service Detail

Detailed Service description of the service, this helps as a quick snapshot of the service.

|  |  |  |
| --- | --- | --- |
| **API / Service Title** | Straight Pass Through |  |
| **API / Service Name** | s-straight-passthrough |  |
| **API / Service Version** | v1 |  |
| **Brief Description** | This is a straight passthrough service that is invoked by a caller client over HTTP connector, in turn invokes the external web service over HTTP, and the response received from external web services is provided as is to the caller. | |
| **Summary of Methods** | GET api/department/data/single | |
| **Criticality** | Low | (Low, Medium High) |
| **Layer / Category** | System |  |
| **Usage** | Asynchronous on demand / Real time |  |
| **BASE URI** | **Internal** : http://10.10.29.245:6001/api/department/data/single  **External**: https://kutumba.karnataka.gov.in/Testfidapi/GetBeneficiaryData | |
| **Transport Protocol** | HTTP(s) | |
| **Service Level Access** | Policy 1 (TBD), Policy 2 (TBD) | |
| **Quality of Service (Proposed)** | Avg. Response Time (Secs) | 6 |
|  | Avg. Payload Size (KB/MB) | 10 KB |
|  | Max Payload Size (KB/MB) | 100 KB |
|  | Avg. Volumes (per day) | 10,000 |
|  | Peak Volumes | 1,00,000 |
| **Batch** | | |
| **Input** | N / A | N / A |
| **Processing** | N / A | N / A |
| **Output** | N / A | N / A |
| **Exception** | TBD | TBD |

Table 7

#### Operation Sequence

This shows the call sequence

**Department** (caller) >> **Mulesoft** >> **Family DB** >> **Mulesoft** >> **Department** (caller)



Figure 15

#### Source

Family DB

Get Beneficiary Data Integration

#### *Destination*

Department

<https://kutumba.karnataka.gov.in/Testfidapi/GetBeneficiaryData>

#### *Inbound Payload*

|  |
| --- |
| {  "deptId":1,  "input":{  "landStatus":"W",  "farmerId":"FID0101000000294"  },  "method":"GetFarmerDetailFarmerIDSet"  } |

#### *Outbound Payload*

|  |
| --- |
| {  "DeptID": "",  "BenID": "",  "RC\_Number": "MLVR00122746",  "Aadhar\_No": "",  "ClientCode": "1963936142",  "HashedMac": "keMGTnm8IhYoneDfgcQqC0bsJS38LoOYM39iy/CP9wM=",  "APIVersion": "1.0",  "IsPhotoRequired": "0",  "Member\_ID": "",  "Request\_ID": "9999999999"  } |

#### Participating APIs and their endpoints

|  |  |  |
| --- | --- | --- |
| **XAPI** | http://10.10.29.245:6001/api/department/data/single | |
| **PAPI** | http://10.10.29.245:8002/esb/fdb/department/single | |
| **SAPI** | http://10.10.29.245:7002/esb/kutumba/GetBeneficiaryData | |
| **Path** | /api/\* |  |
| **SSL** | No |  |

Table 8

#### *Outbound Connector Detail*

|  |  |  |
| --- | --- | --- |
| **HTTP Listener Name** | N/A |  |
| **IP** | N/A |  |
| **Port** | N/A |  |
| **Path** | N/A |  |
| **SSL** | N/A |  |

Table 9

#### Request *Field Mapping*

|  |  |  |
| --- | --- | --- |
| **#** | **Source Fields** | **Destination Fields** |
| **1** | N/A |  |
| **2** | N/A |  |
| **3** | N/A |  |
| **4** | N/A |  |

Table 10

#### Response Field Mapping

|  |  |  |
| --- | --- | --- |
| **#** | **Source Fields** | **Destination Fields** |
| **1** | N/A |  |
| **3** | N/A |  |
| **4** | N/A |  |

Table 11

#### *Error Messages*

|  |
| --- |
| A Possible Error Message from Get Beneficiary Data  Sample 1  {  "StatusCode": -3,  "StatusText": "Invalid HMAC code.",  "Response\_ID": "13072021092717715572211",  "Request\_ID": "9999999999",  "ResultDataList": ""  }  Sample 2  {  "StatusCode": -13,  "StatusText": "Permission denied.",  "Response\_ID": "13072021092804370847670",  "Request\_ID": "9999999999",  "ResultDataList": ""  } |

#### *Business Exception*

N/A

### Use case 2 : Filter and Render

#### Service Detail

This is a service that is invoked by a caller over HTTP connector, in turn invokes the external web service over HTTP, and the response received from external web service is filtered and output that is needed by caller is provided to the caller.

|  |  |  |
| --- | --- | --- |
| **API / Service Title** | Filter and Render |  |
| **API / Service Name** | s-filter-render |  |
| **API / Service Version** | v1 |  |
| **Brief Description** | This is a service that is invoked by a caller over HTTP connector, in turn invokes the external web service over HTTP, and the response received from external web service is filtered and output that is needed by caller is provided to the caller. | |
| **Summary of Methods** | GET api/department/data/single | |
| **Criticality** | Low | (Low, Medium High) |
| **Layer / Category** | System |  |
| **Usage** | Asynchronous on demand / Real time |  |
| **BASE URI** | **Internal** : http://10.10.29.245:6001/api/department/data/single  **External**: https://kutumba.karnataka.gov.in/Testfidapi/GetBeneficiaryData | |
| **Transport Protocol** | HTTP(s) | |
| **Service Level Access** | Policy 1 (TBD), Policy 2 (TBD) | |
| **Quality of Service (Proposed)** | Avg. Response Time (Secs) | 6 |
|  | Avg. Payload Size (KB/MB) | 10 KB |
|  | Max Payload Size (KB/MB) | 100 KB |
|  | Avg. Volumes (per day) | 10,000 |
|  | Peak Volumes | 1,00,000 |
| **Batch** | | |
| **Input** | N / A | N / A |
| **Processing** | N / A | N / A |
| **Output** | N / A | N / A |
| **Exception** | TBD | TBD |

Table 12

#### Operation Sequence

This shows the call sequence

**Department** (caller) >> **Mulesoft** >> **Family DB** >> **Mulesoft** >> **Department** (caller)

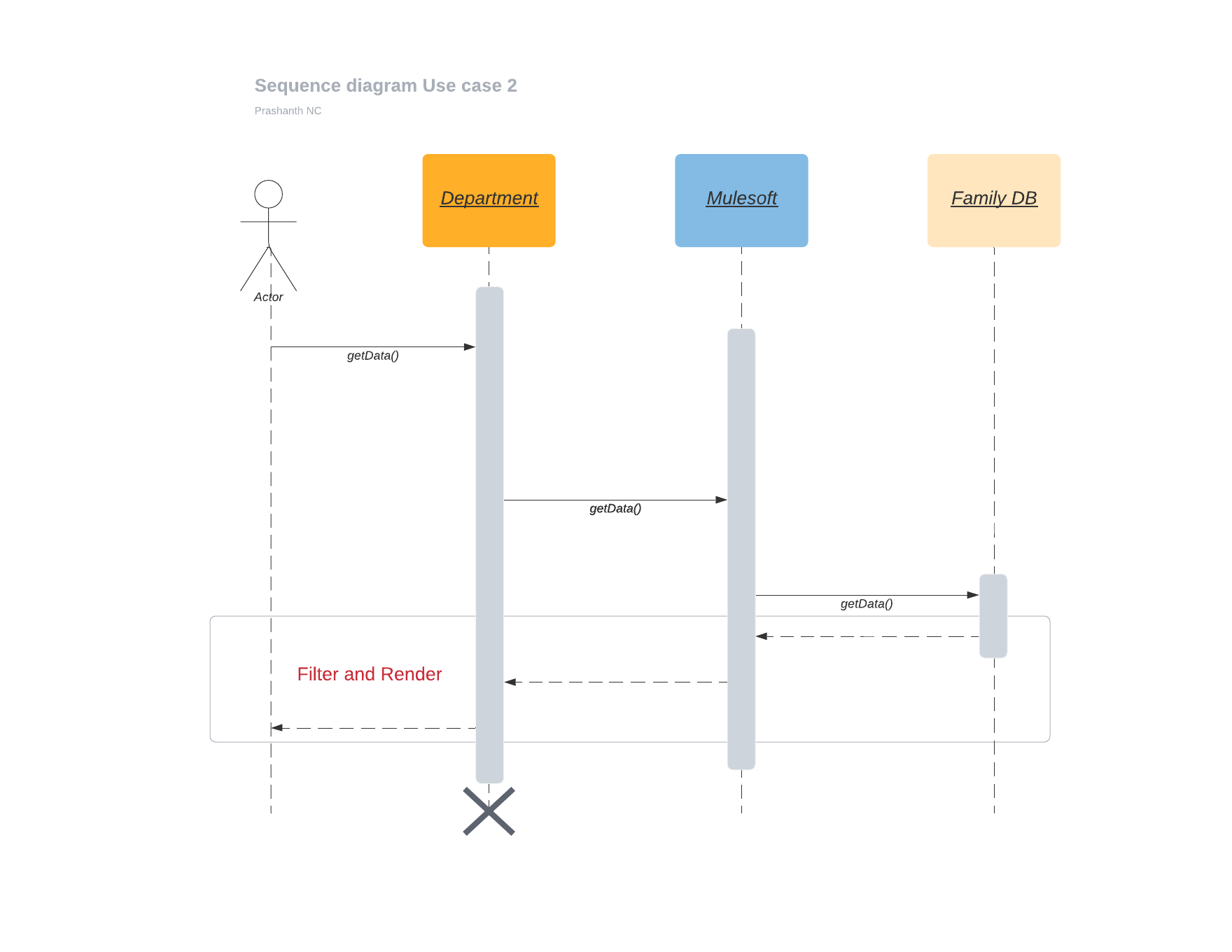


Figure 16

#### Source

Family DB

Get Beneficiary Data Integration

#### Destination

<https://kutumba.karnataka.gov.in/Testfidapi/GetBeneficiaryData>

#### Inbound Payload

|  |
| --- |
| {  "deptId":1,  "input":{  "landStatus":"W",  "farmerId":"FID0101000000294"  },  "method":"GetFarmerDetailFarmerIDSet"  } |

#### Outbound Payload

|  |
| --- |
| {  "DeptID": "",  "BenID": "",  "RC\_Number": "MLVR00122746",  "Aadhar\_No": "",  "ClientCode": "1963936142",  "HashedMac": "keMGTnm8IhYoneDfgcQqC0bsJS38LoOYM39iy/CP9wM=",  "APIVersion": "1.0",  "IsPhotoRequired": "0",  "Member\_ID": "",  "Request\_ID": "9999999999"  } |

#### Participating APIs and their endpoints

|  |  |  |
| --- | --- | --- |
| **XAPI** | http://10.10.29.245:6001/api/department/data/single | |
| **PAPI** | http://10.10.29.245:8002/esb/fdb/department/single | |
| **SAPI** | http://10.10.29.245:7002/esb/kutumba/GetBeneficiaryData | |
| **Path** | /api/\* |  |
| **SSL** | No |  |

Table 13

#### Outbound Connector Detail

|  |  |  |
| --- | --- | --- |
| **HTTP Listener Name** | N/A |  |
| **IP** | N/A |  |
| **Port** | N/A |  |
| **Path** | N/A |  |
| **SSL** | N/A |  |

Table 14

#### Request Field Mapping

|  |  |  |
| --- | --- | --- |
| **#** | **Source Fields** | **Destination Fields** |
| **1** |  |  |
| **2** |  |  |
| **3** |  |  |
| **4** |  |  |

Table 15

#### Response Field Mapping

|  |  |  |
| --- | --- | --- |
| **#** | **Source Fields** | **Destination Fields** |
| **1** |  |  |
| **3** |  |  |
| **4** |  |  |

Table 16

#### Error Messages

|  |
| --- |
| A Possible Error Message from Get Beneficiary Data  Sample 1  {  "StatusCode": -3,  "StatusText": "Invalid HMAC code.",  "Response\_ID": "13072021092717715572211",  "Request\_ID": "9999999999",  "ResultDataList": ""  }  Sample 2  {  "StatusCode": -13,  "StatusText": "Permission denied.",  "Response\_ID": "13072021092804370847670",  "Request\_ID": "9999999999",  "ResultDataList": ""  } |

#### Business Exception

N/A

### Use case 3 : Fetch, Merge and Enrich

#### Service Detail

This is a service that is invoked by a caller over HTTP connector, in turn invokes the two external web services over HTTP, when the responses are received from both this service merges the responses and creates a data that is required by caller, maps and sends the response back to caller.

|  |  |  |
| --- | --- | --- |
| **API / Service Title** | Fetch Merge and Enrich |  |
| **API / Service Name** | s-fetch-merge-enrich |  |
| **API / Service Version** | v1 |  |
| **Brief Description** | This is a service that is invoked by a caller over HTTP connector, in turn invokes the two external web services over HTTP, when the responses are received from both this service merges the responses and creates a data that is required by caller, maps and sends the response back to caller. | |
| **Summary of Methods** | GET http://10.10.29.245:6001/api/department/data | |
| **Criticality** | Low | (Low, Medium High) |
| **Layer / Category** | System |  |
| **Usage** | Asynchronous on demand / Real time |  |
| **BASE URI** | **Internal** : http://10.10.29.245:6001/api/department/data  **External**:  <https://fruits.karnataka.gov.in/fruitsTestservice/fruitsdata.asmx>  <https://nadakacheri.karnataka.gov.in/sms_ws/WebService.asmx>  <https://kutumba.karnataka.gov.in/Testfidapi/GetBeneficiaryData> | |
| **Transport Protocol** | HTTP(s) | |
| **Service Level Access** | Policy 1 (TBD), Policy 2 (TBD) | |
| **Quality of Service (Proposed)** | Avg. Response Time (Secs) | 6 |
|  | Avg. Payload Size (KB/MB) | 10 KB |
|  | Max Payload Size (KB/MB) | 100 KB |
|  | Avg. Volumes (per day) | 10,000 |
|  | Peak Volumes | 1,00,000 |
| **Batch** | | |
| **Input** | N / A | N / A |
| **Processing** | N / A | N / A |
| **Output** | N / A | N / A |
| **Exception** | TBD | TBD |

Table 17

#### Operation Sequence

This shows the call sequence

**Department** (caller) >> **Mulesoft** >> **Family DB** >> **Mulesoft** >> **FCS** >> **Mulesoft** >> **Department** (caller)

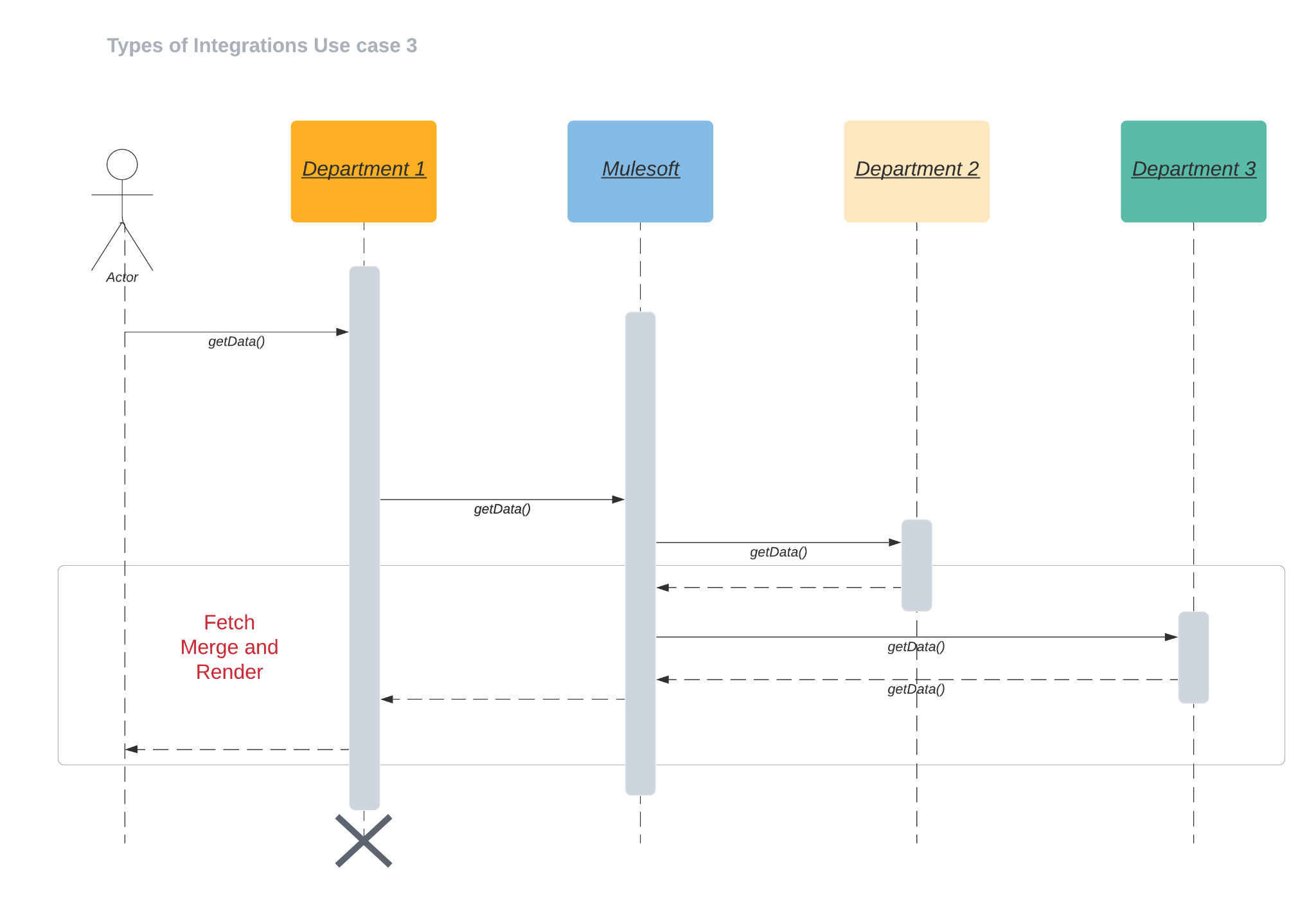


Figure 17

#### Source

Family DB

<http://10.10.29.245:6001/api/department/data>

#### Destination

Departments

<https://fruits.karnataka.gov.in/fruitsTestservice/fruitsdata.asmx>

<https://nadakacheri.karnataka.gov.in/sms_ws/WebService.asmx>

<https://kutumba.karnataka.gov.in/Testfidapi/GetBeneficiaryData>

#### Inbound Payload

|  |
| --- |
| {  "departments": [  {  "deptId": 1,  "input": {  "landStatus": "W",  "farmerId": "FID0101000000294"  },  "method": "GetFarmerDetailFarmerIDSet",  "creds":{  "client-id":"12d34567",  "client-secret":"7654321"  }  },  {  "deptId": 2,  "input": {  "rdNumber": "RD0034130083004"  },  "method": "GetXmlDataWoDSC",  "creds":{  "client-id":"12d34567",  "client-secret":"7654321"  }  },  {  "deptId": 3,  "input": {  "BenID": "",  "RC\_Number": "MLVR00122746",  "Aadhar\_No": "",  "IsPhotoRequired": "0",  "Member\_ID": "",  "Request\_ID": "9999999999"  },  "method": "GetBeneficiaryData"  },  {  "deptId": 6,  "input": {  "BM\_Beneficiary\_Id\_Govt": "990018060420171159"  },  "method": "GetMyPensionDetails",  "creds":{  "client-id":"1234567",  "client-secret":"765d4321"  }  }  ]  } |

#### Outbound Payload

|  |
| --- |
| <?xml version="1.0" encoding="utf-8"?>  Input 1  <soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">  <soap12:Body>  <GetFarmerDetailFarmerIDSet xmlns="http://tempuri.org/">  <UserName>EgovTest</UserName>  <Password>E$Test0521</Password>  <FarmerID>FID0101000000293</FarmerID>  <LandStatus>W</LandStatus>  </GetFarmerDetailFarmerIDSet>  </soap12:Body>  </soap12:Envelope>  Input 2  <?xml version="1.0" encoding="utf-8"?>  <soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">  <soap12:Body>  <GetXmlDataWoDSCV xmlns="http://tempuri.org/">  <GSC\_No>RD0034130083004</GSC\_No>  </GetXmlDataWoDSCV>  </soap12:Body>  </soap12:Envelope>  Input 3  {  "DeptID": "",  "BenID": "",  "RC\_Number": "MLVR00122746",  "Aadhar\_No": "",  "ClientCode": "1963936142",  "HashedMac": "keMGTnm8IhYoneDfgcQqC0bsJS38LoOYM39iy/CP9wM=",  "APIVersion": "1.0",  "IsPhotoRequired": "0",  "Member\_ID": "",  "Request\_ID": "9999999999"  } |

#### Inbound Connector Detail

|  |  |  |
| --- | --- | --- |
| **XAPI** | http://10.10.29.245:6001/api/department/data | |
| **PAPI** | http://10.10.29.245:8002/esb/fdb/department/multiple | |
| **SAPI** | <http://10.10.29.245:7002/esb/kutumba/GetBeneficiaryData>  [http://10.10.29.245:7002/esb/](http://10.10.29.245:7002/esb/kutumba/GetBeneficiaryData)ajsk/GetXmlDataWithDSC  http://10.10.29.245:7002/esb/fruits/GetFarmerDetailFarmerIDSet | |
| **Path** | /api/\* |  |
| **SSL** | N/A |  |

Table 18

#### Outbound Connector Detail

|  |  |  |
| --- | --- | --- |
| **HTTP Listener Name** | N/A |  |
| **IP** | N/A |  |
| **Port** | N/A |  |
| **Path** | N/A |  |
| **SSL** | N/A |  |

Table 19

#### Request Field Mapping

|  |  |  |
| --- | --- | --- |
| **#** | **Source Fields** | **Destination Fields** |
| **1** | N/A |  |
| **2** | N/A |  |
| **3** | N/A |  |
| **4** | N/A |  |

Table 20

#### Response Field Mapping

|  |  |  |
| --- | --- | --- |
| **#** | **Source Fields** | **Destination Fields** |
| **1** | N/A |  |
| **3** | N/A |  |
| **4** | N/A |  |

Table 21

#### Error Messages

|  |
| --- |
| A Possible Error Message from Get Beneficiary Data  Sample 1  {  "StatusCode": -3,  "StatusText": "Invalid HMAC code.",  "Response\_ID": "13072021092717715572211",  "Request\_ID": "9999999999",  "ResultDataList": ""  }  Sample 2  {  "StatusCode": -13,  "StatusText": "Permission denied.",  "Response\_ID": "13072021092804370847670",  "Request\_ID": "9999999999",  "ResultDataList": ""  } |

#### Business Exception

N/A

### Use case 4 : Notification Over Pub Sub

#### Service Detail

This service uses PUB-SUB based where when there is a change in database, a poller polls the data, picks up the delta change and pushes the delta to all the subscribers.

|  |  |  |
| --- | --- | --- |
| **API / Service Title** | Message notification orver pubsub |  |
| **API / Service Name** | s-notify-over-pubsub |  |
| **API / Service Version** | v1 |  |
| **Brief Description** | This service uses PUB-SUB based notification where when there is a change in database, a poller polls the data and pushes the delta to all the subscribers.. | |
| **Summary of Methods** | GET | |
| **Criticality** | Medium | (Low, Medium High) |
| **Layer / Category** | System |  |
| **Usage** | Asynchronous on demand / Real time |  |
| **BASE URI** | **Internal** :  **External**: | |
| **Transport Protocol** | HTTP(s) | |
| **Service Level Access** | Policy 1 (TBD), Policy 2 (TBD) | |
| **Quality of Service (Proposed)** | Avg. Response Time (Secs) | 6 |
|  | Avg. Payload Size (KB/MB) | 10 KB |
|  | Max Payload Size (KB/MB) | 100 KB |
|  | Avg. Volumes (per day) | 10,000 |
|  | Peak Volumes | 1,00,000 |
| **Batch** | | |
| **Input** | N / A | N / A |
| **Processing** | Fetch Delta Data | 6 Sec |
| **Output** | Provide JSON payload that contains the delta changes in DB | 6 Sec |
| **Exception** | TBD | TBD |

Table 22

#### Operation Sequence

This shows the call sequence

**Mulesoft** >> **Family DB** >> **Mulesoft** >> **Messaging Service** >> **Department** (subscriber)

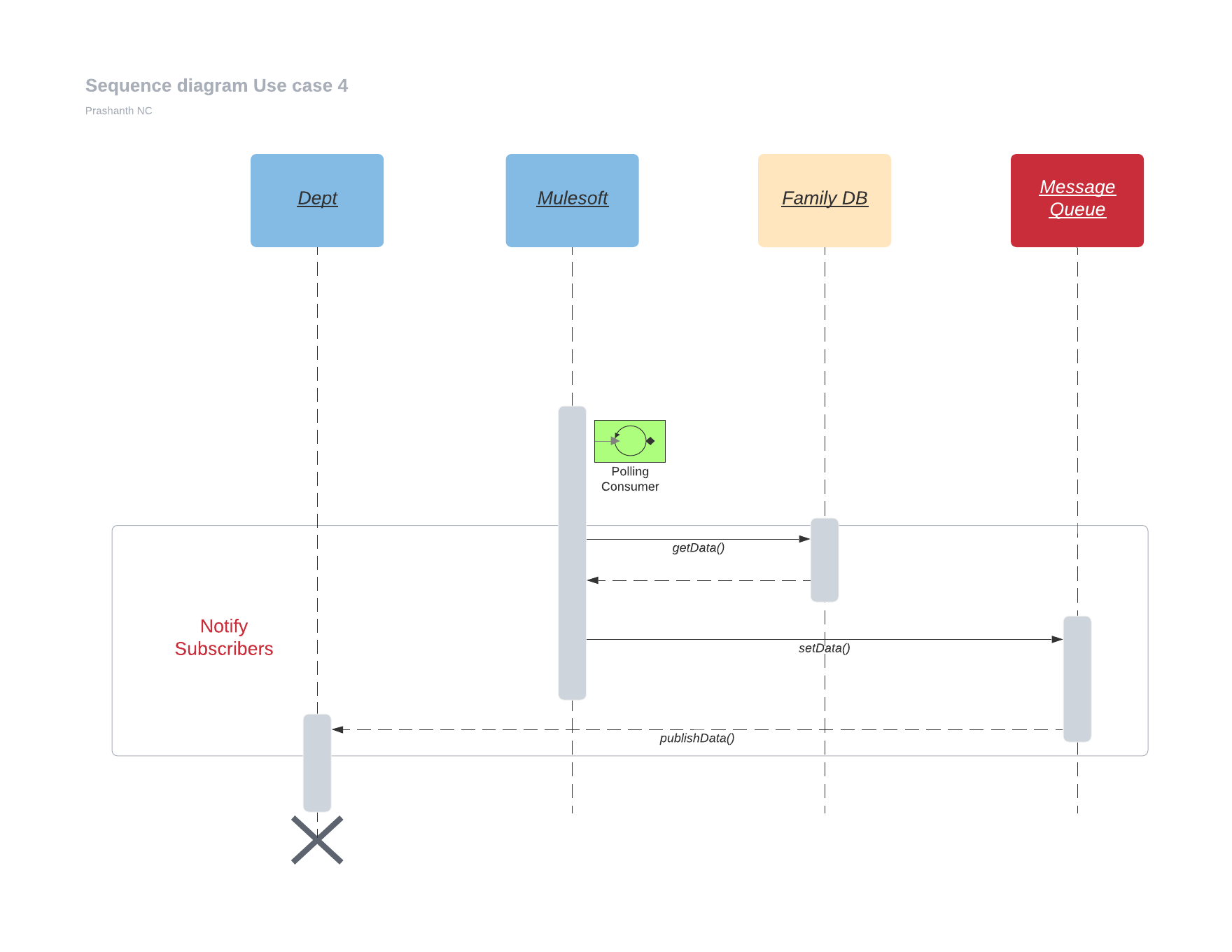


Figure 18

#### Source

Family DB

#### Destination

Department << Specific Department name shall be provided later >>

#### Inbound Payload

<< TBD >>

#### Outbound Payload

<< TBD >>

#### Inbound Connector Detail

|  |  |  |
| --- | --- | --- |
| **HTTP Listener Name** | N/A |  |
| **IP** | N/A |  |
| **Port** | N/A |  |
| **Path** | N/A |  |
| **SSL** | N/A |  |

Table 23

#### Outbound Connector Detail

|  |  |  |
| --- | --- | --- |
| **HTTP Listener Name** | N/A |  |
| **IP** | N/A |  |
| **Port** | N/A |  |
| **Path** | N/A |  |
| **SSL** | N/A |  |

Table 24

#### Request Field Mapping

|  |  |  |
| --- | --- | --- |
| **#** | **Source Fields** | **Destination Fields** |
| **1** | N/A |  |
| **2** | N/A |  |
| **3** | N/A |  |
| **4** | N/A |  |

Table 25

#### Response Field Mapping

|  |  |  |
| --- | --- | --- |
| **#** | **Source Fields** | **Destination Fields** |
| **1** | N/A |  |
| **3** | N/A |  |
| **4** | N/A |  |

Table 26

#### Error Messages

N/A

#### Business Exception

N/A

### Use case 5 : Notification Over Email

#### Service Detail

This service uses PUB-SUB based where when there is a change in database, a poller polls the data, picks up the delta change and pushes the delta to all the subscribers.

|  |  |  |
| --- | --- | --- |
| **API / Service Title** | Message notification orver Email |  |
| **API / Service Name** | s-notify-over-email |  |
| **API / Service Version** | v1 |  |
| **Brief Description** | This service uses Email based notification where when there is a change in database, a poller polls the data and pushes the delta to all the subscribers.. | |
| **Summary of Methods** | GET | |
| **Criticality** | Medium | (Low, Medium High) |
| **Layer / Category** | System |  |
| **Usage** | Asynchronous on demand / Real time |  |
| **BASE URI** | **Internal** :  **External**: | |
| **Transport Protocol** | HTTP(s) | |
| **Service Level Access** | Policy 1 (TBD), Policy 2 (TBD) | |
| **Quality of Service (Proposed)** | Avg. Response Time (Secs) | 6 |
|  | Avg. Payload Size (KB/MB) | 10 KB |
|  | Max Payload Size (KB/MB) | 100 KB |
|  | Avg. Volumes (per day) | 10,000 |
|  | Peak Volumes | 1,00,000 |
| **Batch** | | |
| **Input** | N / A | N / A |
| **Processing** | Fetch Delta Data | 6 Sec |
| **Output** | Provide JSON payload that contains the delta changes in DB | 6 Sec |
| **Exception** | TBD | TBD |
| **Email Server** | | |
| **Connection Type** |  |  |
| **From Email** |  |  |
| **To Email(s)** |  |  |
| **Email Port** |  |  |

Table 27

#### Operation Sequence

This shows the call sequence

**Mulesoft** >> **Family DB** >> **Mulesoft** >> **Email >>**  **Department** (subscriber)

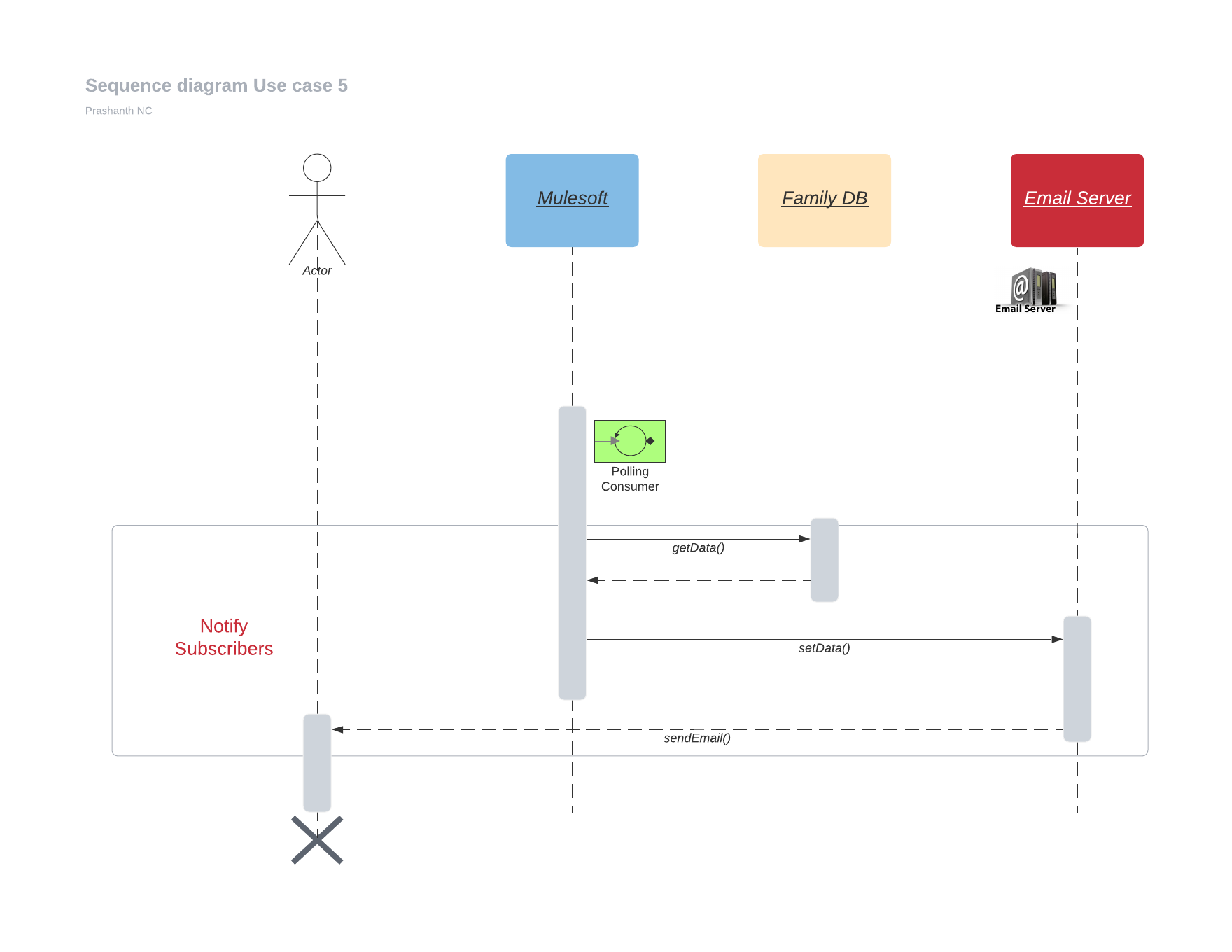


Figure 19

#### Source

Family DB

#### Destination

Department << Specific Department name shall be provided later >>

#### Inbound Payload

<< TBD >>

#### Outbound Payload

<< TBD >>

#### Inbound Connector Detail

|  |  |  |
| --- | --- | --- |
| **HTTP Listener Name** | http-uc05-config |  |
| **IP** | 10.10.29.244 |  |
| **Port** | 8085 |  |
| **Path** | /api/\* |  |
| **SSL** | No |  |

Table 28

#### Outbound Connector Detail

|  |  |  |
| --- | --- | --- |
| **HTTP Listener Name** | N/A |  |
| **IP** | N/A |  |
| **Port** | N/A |  |
| **Path** | N/A |  |
| **SSL** | N/A |  |

Table 29

#### Request Field Mapping

|  |  |  |
| --- | --- | --- |
| **#** | **Source Fields** | **Destination Fields** |
| **1** | N/A |  |
| **2** | N/A |  |
| **3** | N/A |  |
| **4** | N/A |  |

Table 30

#### Response Field Mapping

|  |  |  |
| --- | --- | --- |
| **#** | **Source Fields** | **Destination Fields** |
| **1** | N/A |  |
| **3** | N/A |  |
| **4** | N/A |  |

Table 31

#### Error Messages

<<TDB>>

#### Business Exception

<<TDB>>

# API / Service NFR

## Security

Businesses must ensure that the valuable information they store and make available through software applications and Web services is secure. Locked away and protected from unauthorized users and malicious attackers, protected resources — such as credit card information or Social Security numbers — must still be accessible to authorized legitimate users and systems in order to conduct business transactions. CeG is interested to but not limited to understanding / document the implementation of security w.r.t following points.

* VPN
* Authentication
* Encryption
* Certificates
* Whitelisted IPs
* API Policies
* User groups, roles and privileges
* Monitoring agents

Mulesoft and underlying technology does not store any data that is valuable, protected, private and confidential either on secondary data storage or short lived platforms. \*\*\*

|  |
| --- |
| \*\*\* *Should there be a need to store, appropriate data encryptions are used to ensure that the information or data is obfuscated / hashed / encrypted and stored with prior approvals at various levels.* |

## Auditing

CeG needs a robust solution to audit messages flowing through Mulesoft / ESB. This will aid finding the root, perform analysis and to pinpoint messages last seen, specifically to track messages. Technically the audit need to consider following points

* Request / Response tracking
* Message Origination
* Message Delivery Status
* Message Transaction Status
* Error Scenarios
* Notification Status
* Message life time in Middleware
* Worker that processed the message

## Additional NFRs

|  |  |  |  |
| --- | --- | --- | --- |
| **Identifier** | **NFR** | **Category** | **Description** |
| NFR001 | Security | Login | Create, Read, Update, and Delete (CRUD) levels. Access permissions between mulesoft and other systems must be governed by system administrators |
| NFR002 | Security | Password | Password policies must be defined by CeG team |
| NFR003 | Security | Encryption | (data in flight and at rest) – All communications between Mulesoft others must be encrypted  In transit - TLS 1.2  In Rest - << TBD >> |
| NFR004 | Audit | Traceability | The system must maintain full traceability of messages |
| NFR005 | Capacity | Throughput | The system must be able to handle 20 concurrent transactions per second, per  interface |
| NFR006 | Capacity | Storage | Internally Mulesoft must be able to persist 2 days’ worth of messages.  Since payloads on an average are of 1MB, we need to persist 10GB of data. Year on year growth for storage requirements will be about 20%. Current benchmark: << ?? >> |
| NFR007 | Performance | Response Time | The average API response time will be at Max 6 second |
| NFR008 | Availability | Hours of operation | << ?? >> |
| NFR009 | Availability | Peak Times | << ?? >> |
| NFR010 | Availability | Maintenance Time | << ?? >> |
| NFR011 | Reliability | Mean Time between failure | Expected failure rate is 0.01% with 99.99 uptime |
| NFR012 | Reliability | Mean to recover | If a failure occurs recovery time must be a maximum of 3 hours |
| NFR013 | Recoverability | Recovery Process | << ?? >> |
| NFR014 | Recoverability | Recovery Point Objectives | 4 hours RPO, is defined by business continuity planning. It is the maximum targeted  period in which data might be lost from an IT service due to a major incident. |
| NFR015 | Recoverability | Recovery Time Objectives | RTO, is the targeted duration  of time and a service level within which a business process must be restored after a disaster (or disruption) in order to avoid unacceptable consequences  associated with a break in business continuity. |
| NFR016 | Recoverability | Backup Frequencies | How often do we back up code, data  etc.? |
| NFR017 | Integrity | Information Integrity | << ?? >> |
| NFR018 | Maintainability | Conformance | Conformance to Enterprise  Architecture standards   * Conformance to Technical design standards * Conformance to coding standards * Conformance to best practices |
| NFR019 | Security | Data Boundary | Regional Boundary for  User Data: << ?? >> |

Table 32

## Performance

Specific to this project, each flow shall be governed by following 12 factors that influence high performance of application

1. Keeping optimal log data (not a huge string, or a string object)
2. Encouraging Asynchronous logging
3. Increasing the shut downtime in the Global Settings
4. Incrementing the thread while using polling based components
5. High Level Design
6. Reduce total number of threads while using VM in flows
7. While processing the heady loads, split the connector per endpoint
8. Configure an optimal thread profiles for each flows
9. Tune the receiver thread profile
10. Tune the dispatching thread profile
11. Adjusting and tuning the flow level
12. Adjusting and tuning the pooling profile
13. Use queued asynchronous processing strategy where necessary

CEG proposes to use an asynchronous way of processing the messages. This is achieved by using a variety of patterns. Of such design approaches, one is API led connectivity. This design method helps in reusing many services and distributes the load. In general, a Mule application is a collaboration of a set of flows. Conceptually, messages are processed by flows in three stages:

1. Message receipt by the inbound connector
2. Message processing
3. Message transmission an outbound connector

Stage 1 always happens first. Stages 2 and Stage 3 can be interleaved, since a flow can intermix message processors and outbound endpoints.

Messages are processed asynchronously, unless one of the following is true:

* The flow uses the synchronous processing strategy
* The flow takes part in a transaction
* The inbound endpoint which received the message uses the request-response message exchange pattern

The following diagram illustrates these threads

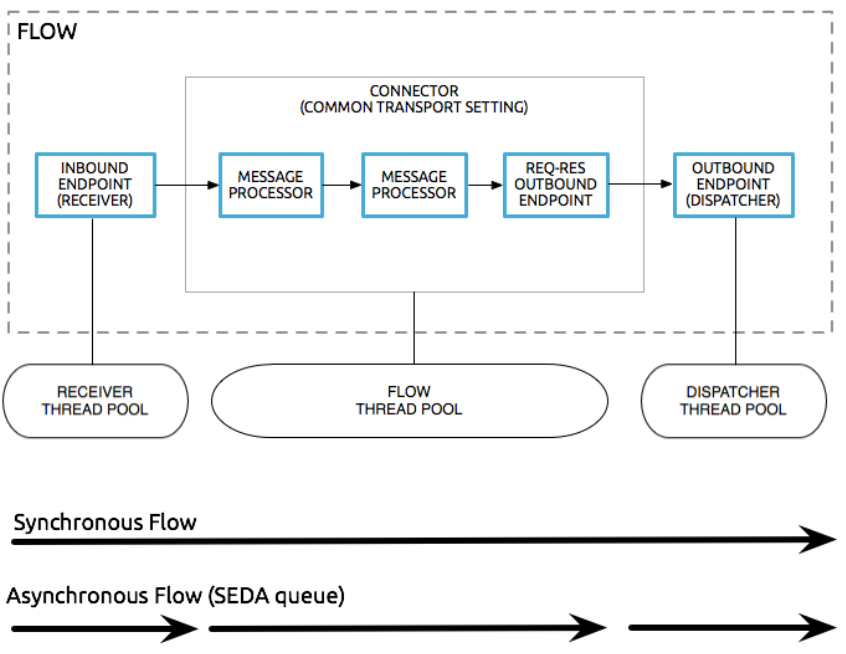


Figure 20

## 

## Application monitoring

### Cloudhub

Cloudhub provides a basic monitoring system for cloudhub / onprem setup. CeG shall use this basic monitoring of services and provide useful insights to the stakeholders. Detailed setup shall be provided in the below mentioned sections after configuring them. This is provided only for the production setup.

## Batch processing

## Build and deployment solution

### Code Versioning

#### Git

Git is a distributed version-control system for tracking changes in source code during software development. It is designed for coordinating work among programmers, but it can be used to track changes in any set of files. Its goals include speed, data integrity, and support for distributed, non-linear workflows

Details of GIT configuration are mentioned below.

### Continuous Integration

#### Jenkins

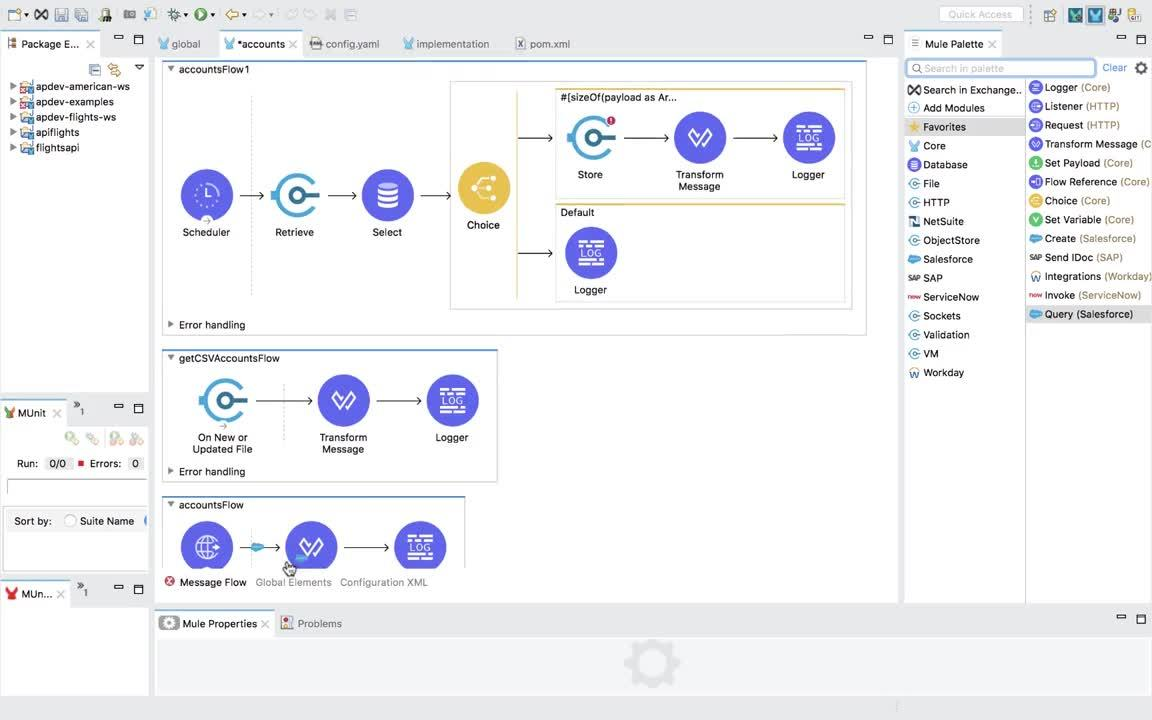
Jenkins is a free and open source automation server. It helps automate the parts of software development related to building, testing, and deploying, facilitating continuous integration and continuous delivery. It is a server-based system that runs in servlet containers such as Apache Tomcat. Details of Jenkins setup are provided below.

## Tools and technologies

### Development Tools

#### *Anypoint Studio*

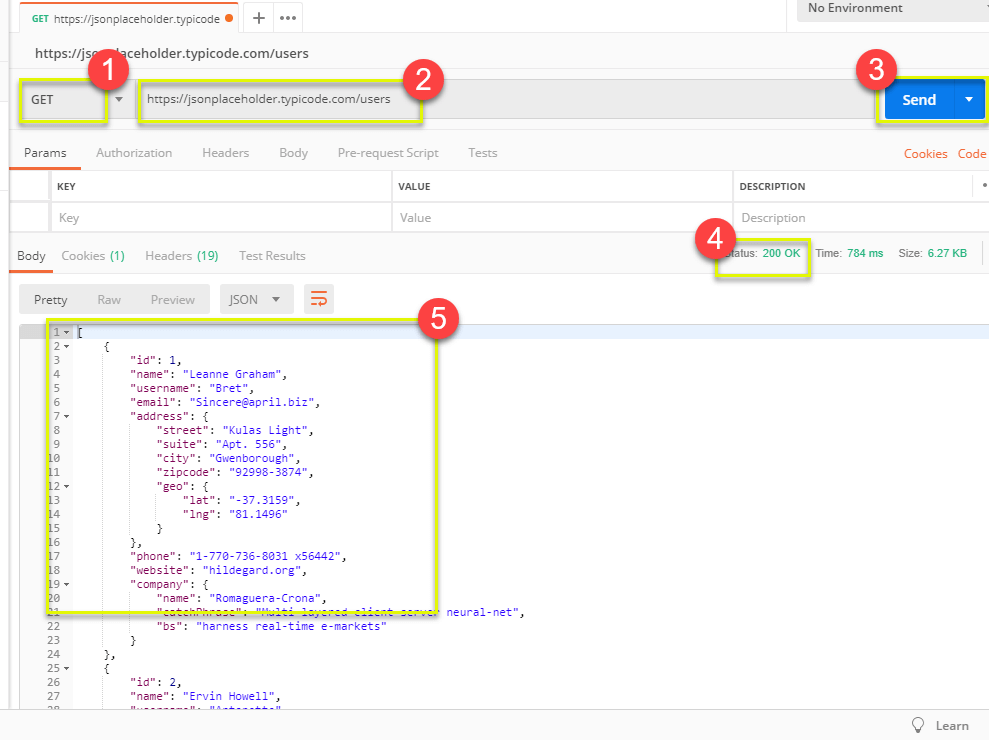
MuleSoft's Anypoint Studio is a user-friendly IDE (integration development environment) used for designing and testing Mule applications. It is an Eclipse-based IDE. We can easily drag Connectors from the Mule Palette. In other words, Anypoint Studio is an Eclipse based IDE for development of flow, etc



Screen Capture 1

#### Postman

Postman is a popular API client that makes it easy for developers to create, share, test and document APIs. This is done by allowing users to create and save simple and complex HTTP/s requests, as well as read their responses



Screen Capture 2

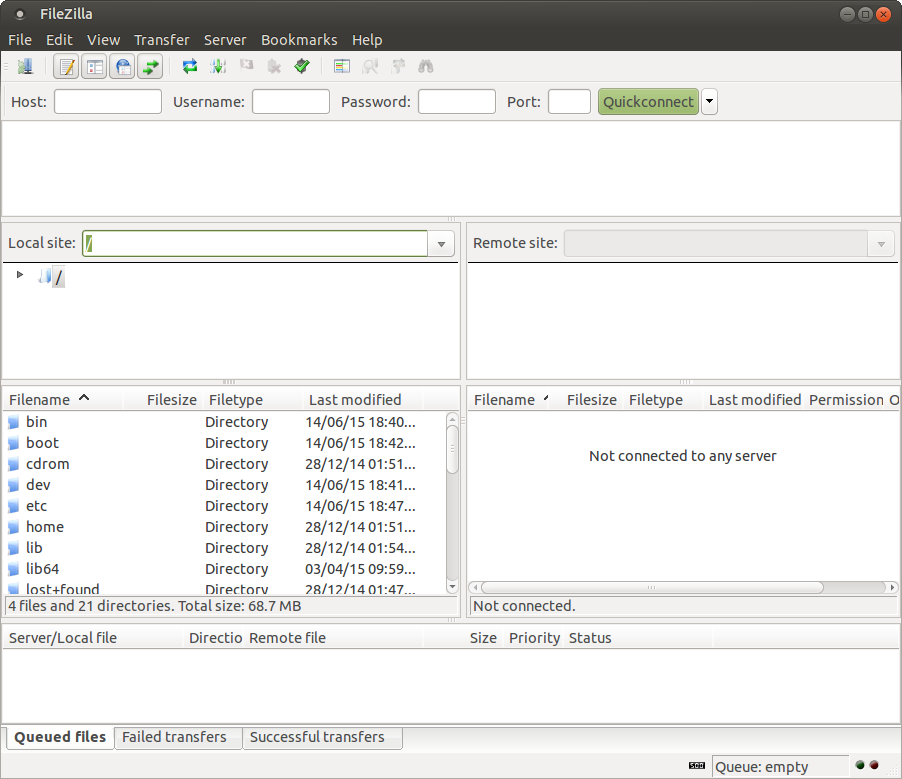
1. HTTP Method
2. URL Space
3. Action Button
4. Response Status Code
5. Payload Window

#### *Putty / FileZilla*

PuTTY is a free and open-source terminal emulator, serial console and network file transfer application. It supports several network protocols, including SCP, SSH, Telnet, rlogin, and raw socket connection.

FileZilla is a free, cross-platform FTP application, consisting of FileZilla Client and FileZilla Server. Features of FileZilla

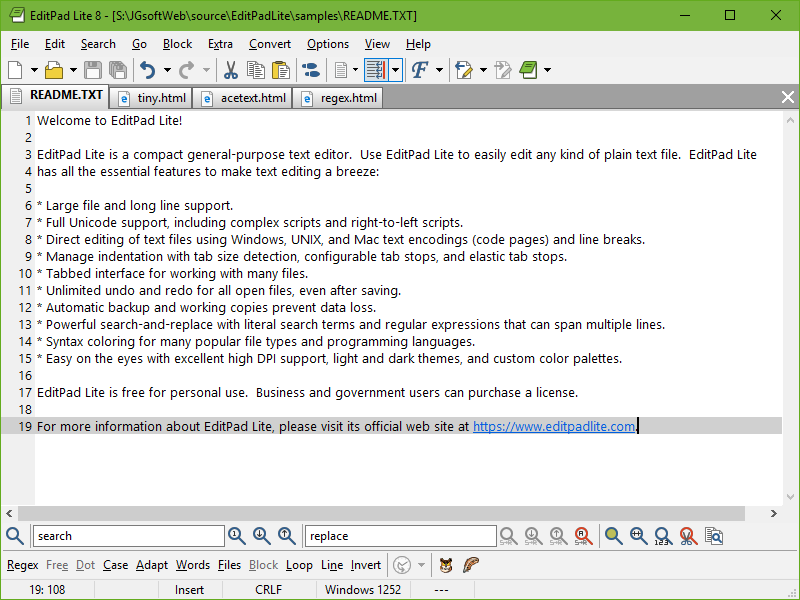
* Transfer files using FTP and encrypted FTP such as FTPS (server and client) and SFTP.
* Support IPv6 which is the latest version of internet protocol
* Supports resume which means the file transfer process can be paused and continued



Screen Capture 3

#### *Text* Editor

A simple word processing application used for editing text content.



Screen Capture 4

### Technologies

#### Java / JDK 1.8

The Java Development Kit is an implementation of either one of the Java Platform, Standard Edition, Java Platform, Enterprise Edition, or Java Platform, Micro Edition platforms

#### *Mulesoft Runtime*

Mule is a lightweight enterprise service bus and integration framework provided by MuleSoft. The platform is Java-based, but can broker interactions between other platforms. Mule runtime engine (Mule) is a lightweight integration engine that runs Mule applications and supports domains and policies. Mule applications, domains, and policies share an XML DSL (domain-specific language)

#### *CloudHub*

CloudHub is the platform as a service (PaaS) component of Anypoint Platform™ — a global, fully-managed, multi-tenanted, secure, and highly available platform for APIs and integrations.

#### Dataweave

DataWeave is a programming language designed for transforming data. It is MuleSoft's primary language for data transformation, as well as the expression language used to configure components and connectors. However, DataWeave is also available in other contexts, like as a command line tool.

#### Maven

Maven is a build automation tool used primarily for Java projects. Maven can also be used to build and manage projects written in C#, Ruby, Scala, and other languages. The Maven project is hosted by the Apache Software Foundation, where it was formerly part of the Jakarta Project.

#### SonarQube

SonarQube is an open-source platform developed by SonarSource for continuous inspection of code quality to perform automatic reviews with static analysis of code to detect bugs, code smells, and security vulnerabilities.

#### *Shell Scripting*

A shell script is a computer program designed to be run by the Unix shell, a command-line interpreter. The various dialects of shell scripts are considered to be scripting languages. Typical operations performed by shell scripts include file manipulation, program execution, and printing text.

### Environment

#### Non Production - *Development*

##### Development

Please refer to Setup Document (MuleSoft Setup.pdf) available at Source Control link

##### Testing

Please refer to Setup Document (MuleSoft Setup.pdf) available at Source Control link

#### Production

##### Production Active

Please refer to Setup Document (MuleSoft Setup.pdf) available at Source Control link

##### Production Passive (DR)

Please refer to Setup Document (MuleSoft Setup.pdf) available at Source Control link

#### 

# Testing

## Testing Approach

#### *Functional testing*

Functional testing of below mentioned use cases marks the standard types of use cases that are foreseen in integration @ CeG,Govt of Karnataka. If anything new is discovered, this section shall be updated.

Test Case template suggested for functional testing is as follows

|  |
| --- |
| **TestCase ID** |
| **Scenario** |
| **TestCase** |
| **Pre Conditions** |
| **Test Steps** |
| **Test Data** |
| **Expected Result** |
| **Post Condition** |
| **Actual Result** |
| **Test Status** |

Details are attached in the Test Case Document <https://docs.google.com/spreadsheets/d/1jYULsHUSrqPwWWZwkAyxvRtT4StQf_cw65vc9j61yr4/edit#gid=0>

Test restults are recorded and maintained at the following link.

<https://drive.google.com/file/d/1sE1Wk6EW4MJ4J-s1uYdvdm0nl8K2Hgjz/view?usp=sharing>

#### Fetch and Render

Please refer to Test Cases for Discovery Usecases under source control link

#### *Fetch, Filter and Render*

Please refer to Test Cases for Discovery Usecases under source control link

#### *Fetch, Merge and Render*

Please refer to Test Cases for Discovery Usecases under source control link

#### *Notify over PUB-SUB*

Please refer to Test Cases for Discovery Usecases under source control link

#### *Notify over Email*

Please refer to Test Cases for Discovery Usecases under source control link

#### *Non Functional testing*

This is going to be determined based on the department's SLA, that is not available at the time of preparing this document.

#### *Testing Tools*

Please refer to section 6.8.1.2

#### *Mocking Frameworks*

N / A

#### 

# Appendix

|  |  |  |
| --- | --- | --- |
| S. No. | Acronym | Expanded Form |
| 1 | CI / CD | Continuous Integration and Continuous Delivery |
| 2 | DR | Disaster Recovery |
| 3 | HA | High Availability |
| 4 | JDK | Java Development Kit |
| 5 | MVN | Maven |
| 6 | MW | Middleware |
| 7 | NP | Non Production |
| 8 | P | Production (when used next to cores or vCores) |
| 9 | RAML | RESTFul API Modeling Language |
| 10 | REST | Representational Stage Transfer |
| 11 | RHEL | Red Hat Enterprise License |
| 12 | TBC | To Be Considered |
| 13 | UAT | User Acceptance Test |
| 14 | VPC | Virtual Private Cloud |
| 15 | VPN | Virtual Private Network |
| 16 | WSDL | Web Service Description Language |
| 17 | eGov | e-Governance |
| 18 | GoK | Government of Karnataka, India |
| 19 | CeG | Center for e-Governance |

Table 33