Project Description

Name : Payment Process
Author : Ravi Sah
Date : 05/04/2021

Steps to be followed

- Understand Requirement
- Create RAML
- Develop API using RAML
- Testing
- Munit Testing
- Code Check-in

Objective of this exercise

- To simulate a real world scenario, where a solution architecture document will be handed down to the development team to build & implement.
- Make use of MuleSoft out of the box capabilities (as much as possible) to come out with a solution to address the design problems addressed in the use case.
- Include previous best practices experience when coming out with the solution.
- Make use of Design Center to create RAML projects and design RAML specification base RAML best practices based on the business scenario given.
- Organize and create the mule projects based on the business scenarios, and solutions described in this document.
- Creating mule message flows, making use of mule components and connectors. implement a solution based on the scenario and specifications given in this document.
- M-unit test is created to test each of the possible scenarios.
- Test scripts creation to test the end point using testing tools such as Postman / Soap UI (if needed).
- Demo the working example at the end. During the presentation, outline in detail what you are trying to solve in your Mule project and the approach to resolve it.
- Extend and enhance your application in real-time.

Business Scenario

Alpha Insurance is a startup company providing a platform to buy electronic device insurance across the world.

This platform will enable customers to see available insurance for different electronics devices and customers can buy insurance.

This platform will connect to the respective insurance provider and will search their insurance plan and will show it to customers. Customers will appropriate the plan from the list of available plans and will make payment. Once payment is completed, customers can see plan details in the order page with other order information.
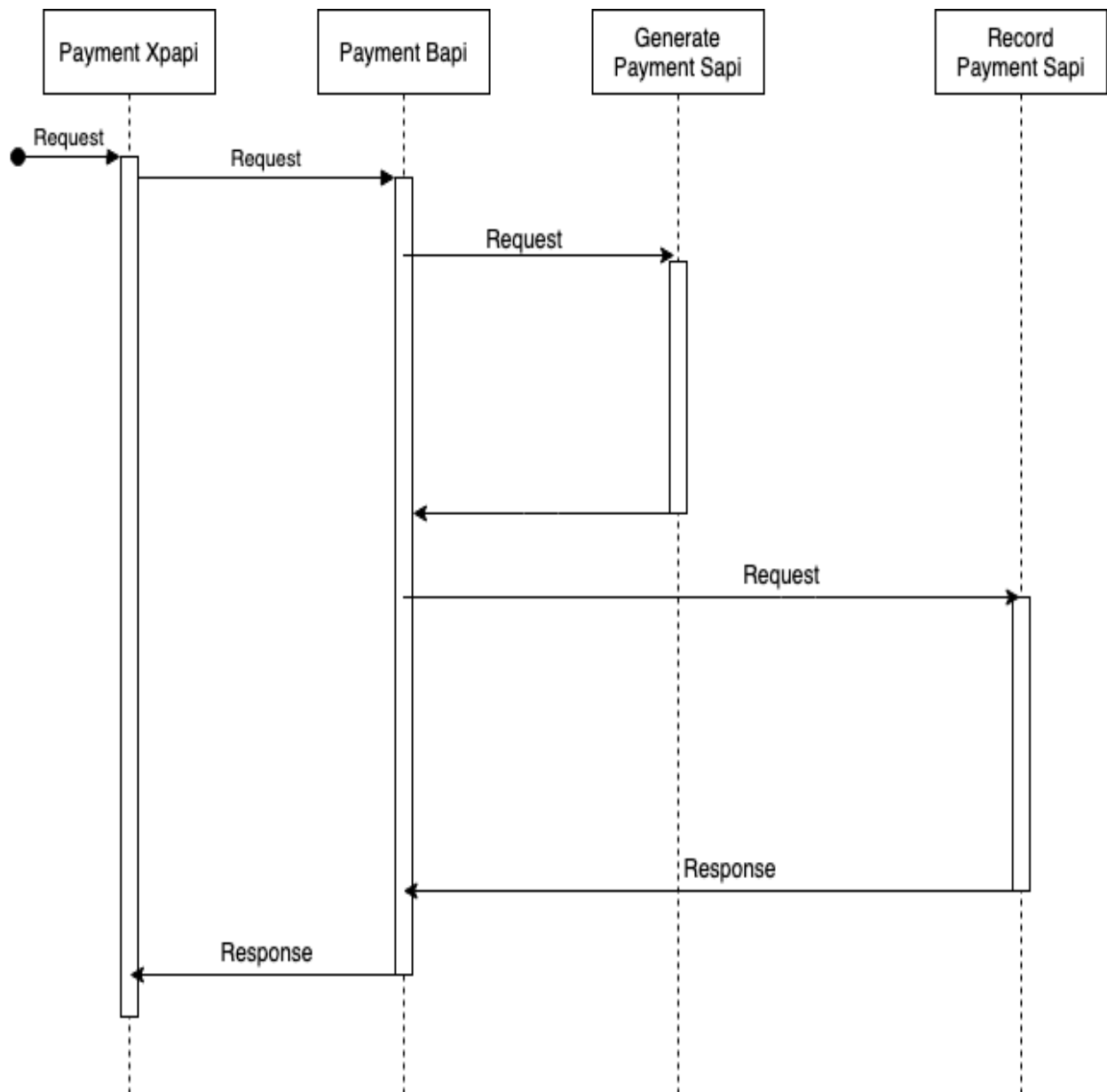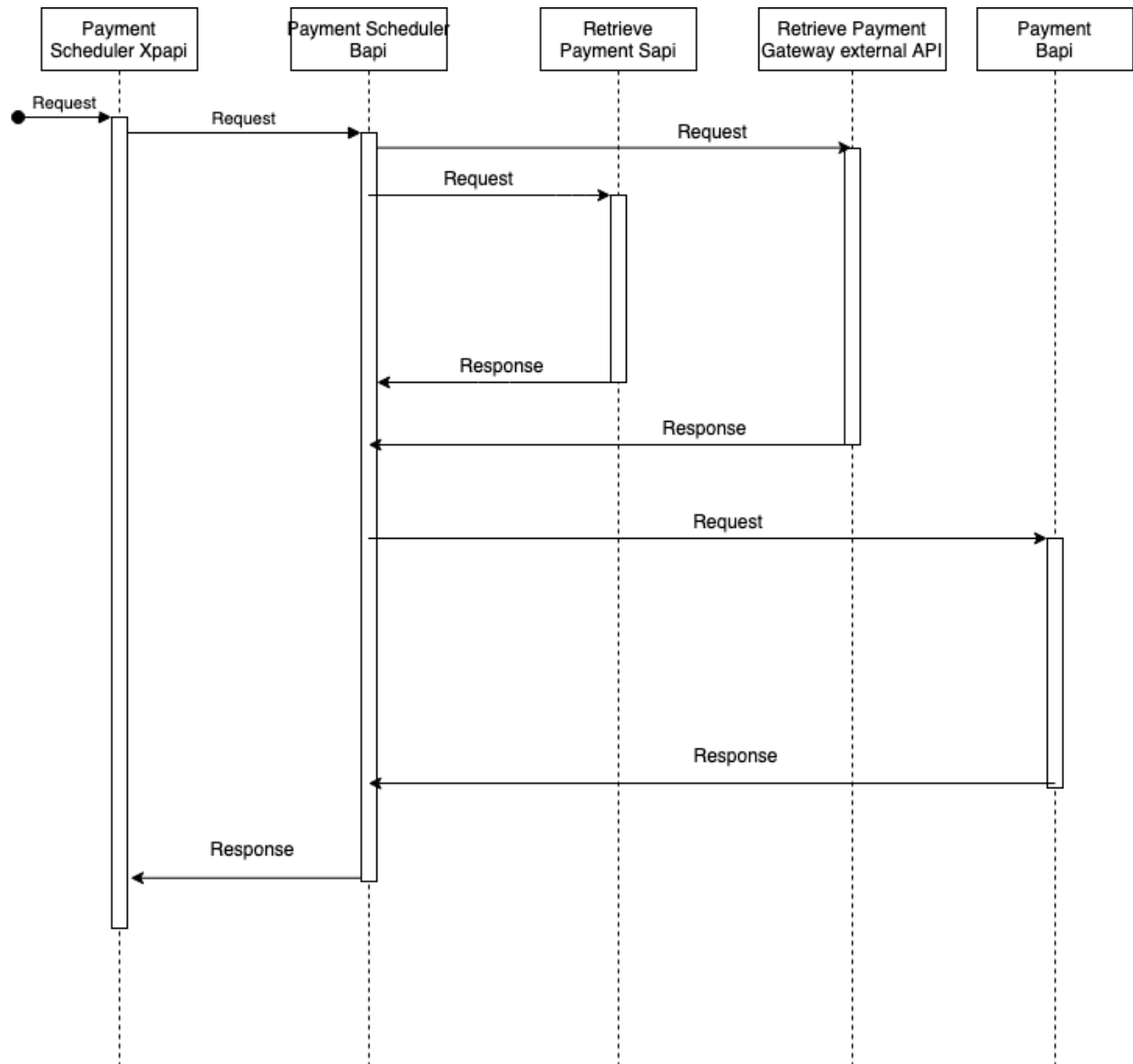
Payment Flow Diagram
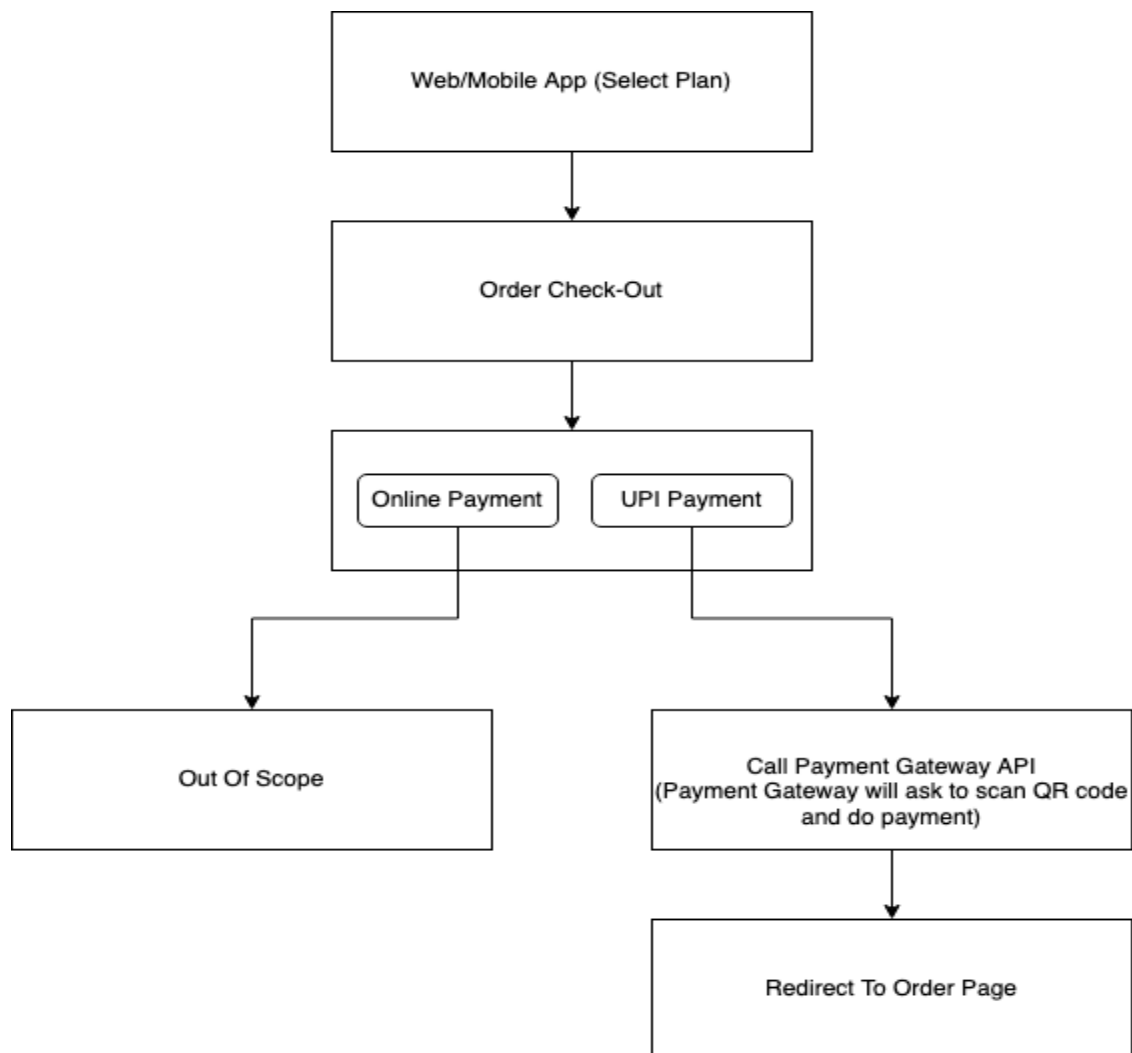
API Landscape
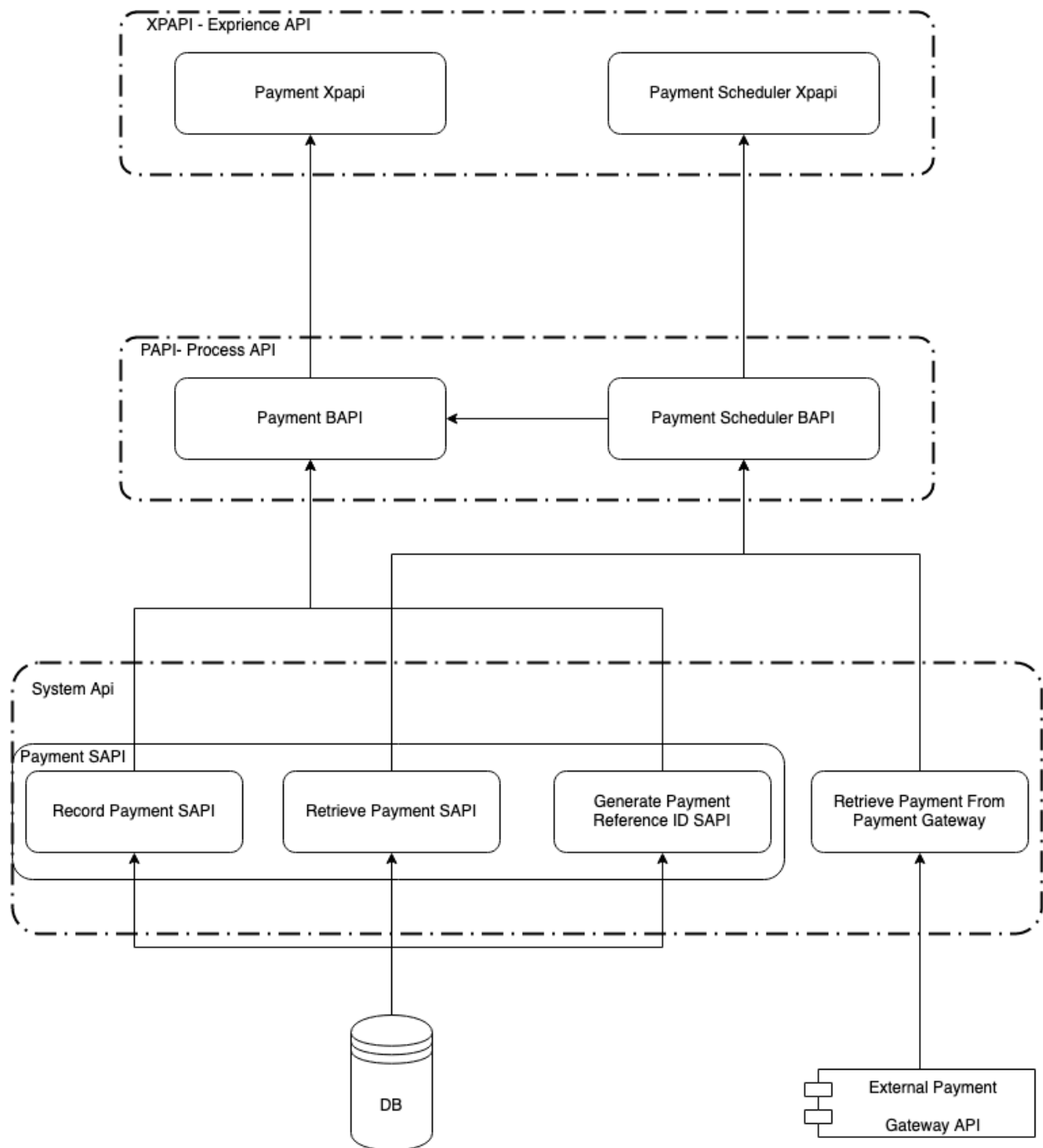
Sequence Diagram Payment

Sequence Diagram Payment Scheduler

Payment Scheduler Xpapi	Payment Scheduler Bapi	Retrieve Payment Sapi	Retrieve Payment Gateway external API	Payment Bapi

Request

Request

Request

Request

Response

Response

Request

Response

Response

Payment Xpapi :

Payment Gateway will call this api to record payment Transaction. This API will do JWT token validation.

FD Name : RecordPayment-XPAPI-FunctionalDesignDocument-V1.0

Scheduler Payment Xpapi:

EveryDay scheduler will call this api to record missing transactions.

FD Name : Record Payment Scheduler-XPAPI-FunctionalDesignDocument-V1.0

Payment Bapi :

This Api will call by Payment xpapi and Schedular payment bapi, where all business logic will be applied.

FD Name : RecordPayment-BAPI-FunctionalDesignDocument-V1.0

Scheduler Payment Bapi :

This Api will be called by Schedular payment Xapi.

FD Name : Record Payment Scheduler-BAPI-FunctionalDesignDocument-V1.0

Generate Payment ReferenceId Sapi:

Record payment bapi will call this api to generate unique payment referenceId.

FD Name : GeneratePaymentReferenceId-SAPI-FunctionalDesignDocument-V1.0

Record Payment Sapi :

This api will be called by payment bapi to record transactions.

FD Name : RecordPayment-SAPI-FunctionalDesignDocument-V1.0

Retrieve Payment Sapi:

This api will be called by payment bapi to retrieve payment sapi.

FD Name : RetrievePayment-SAPI-FunctionalDesignDocument-V1.0

Retrieve Payment External Api :

This api will be called by payment bapi to get missing transactions from payment gateway.

FD Name : Retrieve Transaction Payment Gateway-External API-FunctionalDesignDocument-V1.0

Note : For Details go to API detail documents.