**Emkan Finance**

A picture containing text, tableware, plate, dishware

Description automatically generated

Migration

Mobile Flow

**September 15, 2024**

Contents

[**Document Version History** 4](#_Toc175431736)

[**Document Review & Approval History** 4](#_Toc175431737)

[**Document References** 4](#_Toc175431738)

[1. Introduction 5](#_Toc175431739)

[Document Purpose 5](#_Toc175431740)

[Solution Objectives 6](#_Toc175431741)

[Key Requirements: 6](#_Toc175431742)

[**Scope** 7](#_Toc175431743)

[**Assumption** 7](#_Toc175431744)

[**High Level Business Requirements** 8](#_Toc175431745)

[**BRD References :-** 8](#_Toc175431746)

[**Architecture Overview** 9](#_Toc175431747)

[Architecture Layers 10](#_Toc175431748)

[ Diagram 10](#_Toc175431749)

[Integration Tools and Systems 13](#_Toc175431750)

[ Apache Kafka 13](#_Toc175431751)

[ IBM MQ 13](#_Toc175431752)

[ Informatica ETL 13](#_Toc175431753)

[ Oracle MFT 13](#_Toc175431754)

[ Incorta 13](#_Toc175431755)

[Core Systems 14](#_Toc175431756)

[Logical Application Architecture 14](#_Toc175431757)

[1-Transact Loan Management System (LMS) 14](#_Toc175431758)

[2-Newgen iBPS LOS 15](#_Toc175431759)

[3- MS-Dynamics 15](#_Toc175431760)

[4- ENME 16](#_Toc175431761)

[1- Emkan Payment System Hub (PSH) 17](#_Toc175431762)

[2- Emkan Notification Engine 17](#_Toc175431763)

[3- Emkan Rules Engine 17](#_Toc175431764)

[6- Registration Microservice 18](#_Toc175431765)

[Quality and rule engine 18](#_Toc175431766)

[Parallel Run and Coexistence 19](#_Toc175431767)

[Inquiries: 19](#_Toc175431768)

[1-Aggregator Logic 19](#_Toc175431769)

[2-Dispatcher Logic 20](#_Toc175431770)

[ Description 20](#_Toc175431771)

[**Execution Flow View** 22](#_Toc175431772)

[.1. Configuration 22](#_Toc175431773)

[1. Onboarding Flow 25](#_Toc175431774)

[2. Dashboard Flow S2 25](#_Toc175431775)

[3. Loan Eligibility Flow (Lead Generation Flow ). S1 31](#_Toc175431776)

[4. Customer declaration s2 32](#_Toc175431777)

[5. product Eligibility check 34](#_Toc175431778)

[6. IBAN Verification 35](#_Toc175431779)

[7. Upload Document Flow S3 36](#_Toc175431780)

[8. Quality Flow 37](#_Toc175431781)

[9. Offering Flow 38](#_Toc175431782)

[10. IVR Flow 40](#_Toc175431783)

[11. BuyCommodity Flow 42](#_Toc175431784)

[12. ContractSigning Flow 43](#_Toc175431785)

[13. Sanad Flow 44](#_Toc175431786)

[14. view sale authorization Flow 45](#_Toc175431787)

[15. Sell commodity Flow 45](#_Toc175431788)

[16. Disbursment Flow 46](#_Toc175431789)

[17. MissDocument Flow 46](#_Toc175431790)

[**Deployment and Security View** 47](#_Toc175431791)

[Appendix A - Architectural Decisions 47](#_Toc175431792)

[1. 48](#_Toc175431793)

[1. 48](#_Toc175431794)

[Appendix D – Failing Retry Policy 49](#_Toc175431795)

[Appendix E – Cooling Period 50](#_Toc175431796)

[Appendix R – Risks 50](#_Toc175431797)

[Appendix I – General Impacts 50](#_Toc175431798)

[Appendix T – Third Parties 50](#_Toc175431799)

# **Document Version History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Ver** | **Date** | **Author** | **Change Description** |
| 0.1 | 15/09/2024 | Mohammed Ishaq  Mohamed Hosni |  |
| 04 | 14/10/2024 | Mohammed Ishaq  Mohammed Hosni | Updated Iban Verification flow and Contract Signing |

Table 2 Document Version History

# **Document Review & Approval History**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Version** | **Reviewer** | **Approver** | **Review Date** | **Approval Date** |
| 0.1 |  |  |  |  |
|  |  |  |  |  |

Table 3 Document Review & Approval History

# **Document References**

|  |  |
| --- | --- |
| **Document Name** | **Document Link** |

**Acronym Defintion**

|  |  |
| --- | --- |
| **Acronym** | **Description** |
| ARB | Al Rajhi Bank |
| API | Application Programming Interface |
| OTP | One Time Password |
| BAU | Business as Usual (already implemented / As is) |
| MS | Microservice |
| AML | Anti-Money Laundering |
| IVR | Interactive Voice Response |
| SAMA | Saudi Central Bank |
| SIMAH | Saudi Credit Bureau |

**System Roles**

List of all related systems and their roles as per the solution

|  |  |
| --- | --- |
| **System** | **Role** |
| **Mobile App** | Customer Interface |
| **Emkan Services** | Integration , Business and Mobile BE Services |
| **Kastle** | Loan Orignation Banking System(LOS) has following modules:  Product Catalog  LMS Loan management System |
| **LOS** | Loan Origination system (newgen) |
| **Qarar** | Decision Engine Service[third party], it implements a decision engine based on credit bureau data. it implements decision rules to accept, reject or refer card applications. |
| **DMS** | Document Management System |
|  |  |
| **Unifonic** | Third party for sending notification to customer like SMS or IVR |
| **Emkan NE** | Emkan Notification Engine for sending notification like IVR ,SMS, push notification |
| **Emkan Rule Engine** | System to apply rules provide set of rules result |
| **Newgen** | Loan Origination System |
| **T24** | Loan Management System |
| **Middleware** | Integration Layer |
| **CRM** | Master Customer Data |

# Introduction

Emkan vision is to implement cutting-edge digital lending solutions that can power Emkan's next growth phase. This will be achieved through a phased migration of the existing loan portfolio and implementation of an integrated lending platform.

Key objectives are:

* Completely migrate customer records and loan accounts from Kastle to TRANSACT Core Banking.
* Implement a Newgen LOS to automate end-to-end digital lending workflows for all retail products.

## Document Purpose

The purpose of this document is to produce a High-Level Design (HLD) and architecture for Retail Lending platform implementation and migration, the key building blocks of the solution and how those are going to address the key business and technology requirements. It will also provide the key architectural recommendations and architecture decisions discussed and agreed during the discovery workshops.

This document covers Direct PF and Advance Salary.

## Solution Objectives

* Implement retail products using Newgen LOS and Transact LMS
* Migrate LMS loans accounts and contracts from Kastle to TRANSACT.
* Accelerate the decommissioning of the handicapped legacy core system, impacting time-to-market and stability.
* Mitigate the typical high-risk associated with this strategic initiative.
* Minimal impact on Emkan business and operation.
* Seamless transition through phased retail products data and functional migration.
* Readiness to host new products on the target platform as early as possible.
* Capitalize on Customer-Centricity through enterprise-wide Customer Master.
* Minimal or no impact on customer channels, B2B Gateways and external systems.
* Minimal integration impact on Emkan systems.
* Converge to Emkan new architecture and technology with superior qualities.

## Key Requirements:

LMS Migration Requirements:

* Migrate customer, loan and product data from Kastle to TRANSACT
* Replicate functionalities for loan origination, management and collections
* Interfaces with accounting, reporting, downstream systems
* Role-based access controls and authorization protocols
* Performance, availability, reliability and scalability
* Audit trails and regulatory compliance for the solution

LOS Implementation Requirements:

* End-to-end digital workflows for loan applications
* Configurable approval hierarchies and conditions
* Customizable repayment and settlement options
* Real-time decisioning via underwriting and modeling
* Straight-through processing for majority of applications

# **Scope**

-in scope

The scope of the document covers key architectural viewpoints including business architecture, solution architecture, logical application architecture, data architecture, and technology architecture. The document also captures all the business processes. The HLD provides the definition of the business processes, for Emkan’s Retail Products, solution architecture, logical application architecture, key technology services including integration services, data objects, deployment model. The Integration Team would provide the description of the interfaces proposed in septate IRD, the integration capabilities (messaging, connectivity, transformation, distribution, etc) and the supporting integration formats (XML, JSON, EDI, Bus, File/FTP, DB, etc.).

I

* Build new retail technology platform (LMS & LOS).
* Configure the new LMS on TRANSACT and new LOS on Newgen to cater for the following existing business products:

1. Retail Finance.
2. Micro finance.
3. Top-up.
4. Refinance.
5. Buyout.
6. Auto-lease.
7. Advanced Salary.
8. Direct PF.
9. Credit Card (LOS only).

-out of scope

* Kastle journey out of scope and required changes will covered in separate Document

# **Assumption**

* Emkan Registration flow /Login Flow [BAU]
* Emkan Notification Engine [BAU]

# **High Level Business Requirements**

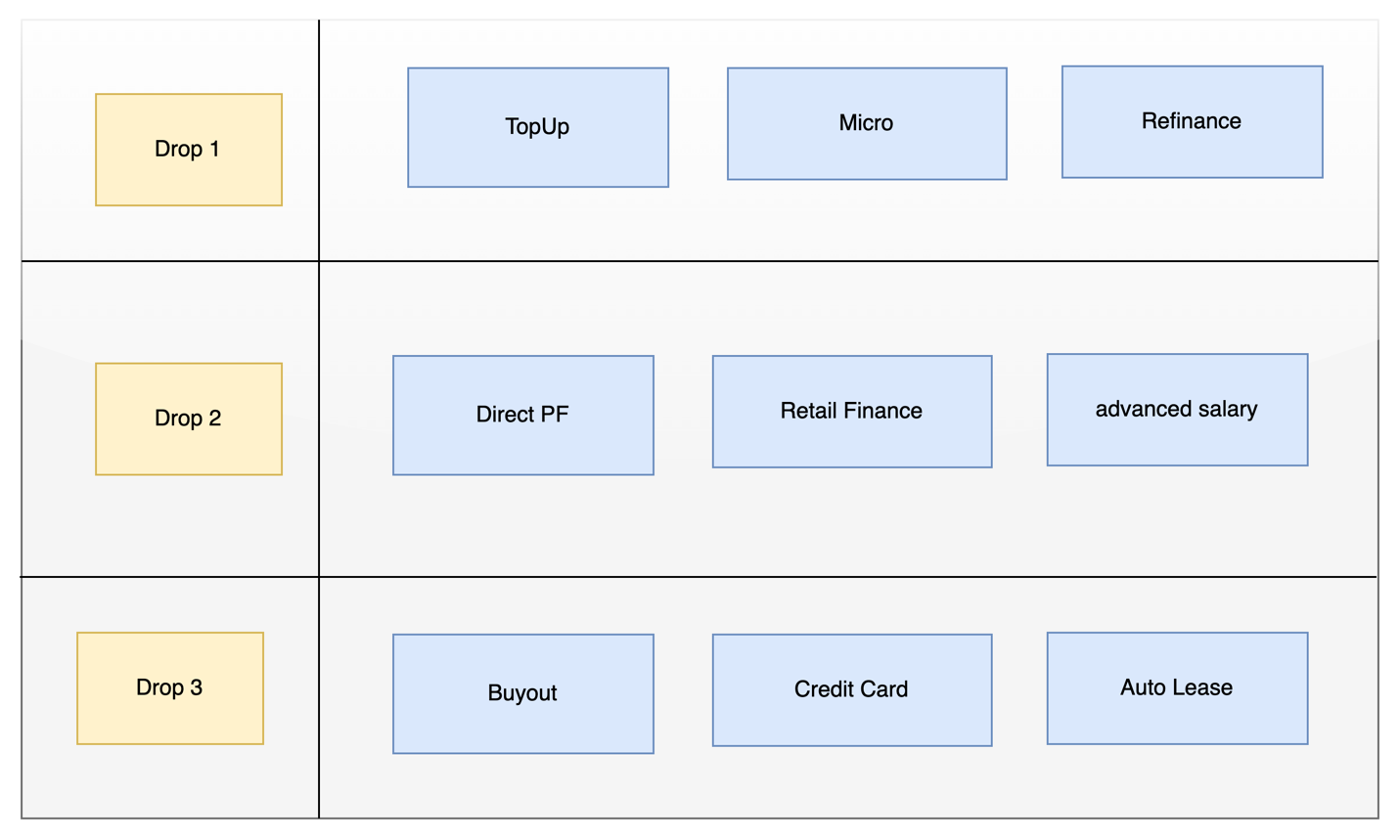
|  |  |  |  |
| --- | --- | --- | --- |
| ID | JIRA Ref. | Statement of Requirement | |
| Migration Drop 2 (LOS) | | | [https://jira.emkanfinance.com.sa/browse/](https://jira.emkanfinance.com.sa/browse/EDF-16) |

# **BRD References :-**

The following are the key strategic objectives of the Emkan Fintech Platform, as below:

* A fully- digital Loan Origination and Loan Management service platform that provides services throughout the value chain (loan origination to the disbursement and management)
* A digital platform that offers seamless connectivity with both internal and external systems and is created in accordance with financial industry best practices and standards as per Saudi laws and Emkan’s business model
* An internal loan underwriting capacity and approval/rejection capacity, that is fully automated and highly optimized, with improved support for Analytics based on rules engine for decision making as per B2B inputs
* Highly configurable workflows to serve the needs of different Loan products, offered by Emkan Finance
* Enabling end to end security to ensure the privacy and confidentiality of the customer’s data and Realtime customer data with CRM
* Customer onboarding- Customer registration or login
* Loan application- Customer selects a particular Loan product start application from Emkan channel.
* Loan eligibility and Contract sign- Customer due diligence, and eligibility based on Credit reports, income and monthly liabilities, with contract sign customer.
* Disbursement and Loan Management- Payment to customer and account and manage the further updates on monthly payments by customer, till last payment and contract closure in addition to loan early repayment, pas due management, loan amendment.
* Collection- Update collection management system, for regular or defaulted payments daily.

**Drops :**



**Risks on Defined Drops:**

1. **Customer will not be able to refinance Direct PF Applications**
2. **If customer is eligible for direct PF and he apply on topup or micro Qarar should disable offering Direct PF and vice versa**

**Recommendation**

1. **Moving Direct PF to Drop 1**
2. **Adding functionality to allow customer to apply for specific product if this product is QARAR decision is other product**

# **Architecture Overview**

Our architecture will be Mix of the following:

* Cloud based Architecture using OCI
* Microservice Architecture
* Service Oriented Architecture
* Event Driven Architecture

### Architecture Layers

1. Channel
2. External Gateway
3. User Repository and Identity Management
4. Back-end For Front-end (BFF)
5. Integration Layer and internal API Gateway
6. Core Systems
7. Data Layer

## Diagram

The below diagram will show how the architecture will be

|  |  |  |
| --- | --- | --- |
|  | Name | Description |
| Channels | | |
| 1 | Mobile Applications | Emkan Mobile Application |
| 2 | Emkan’s web site | Emkan Mobile Application (Retail Finance RF) |
| 3 | Kastle Back office | Branch journey and underwriting tasks |
| 4 | Newgen Back office | and underwriting tasks |
| 5 | Transact Back Office | LMS back-office tasks |
| 6 | One Portal | Emkan one portal Staff portal |
| 7 | Emkan’s Merchants Portal | Emkan merchant portal |
| 8 | Merchant’s E-commerce Platforms | Like Apple or Jarir |
| 9 | Emkan Branch | Newgen Branch journey |
| Internal Systems and Core Backends | | |
| 1 | Temenos Transact | LMS Core Banking |
| 2 | iBPS | Newgen LOS |
| 3 | Kastle ULS | Kastle unified loan system |
| 4 | MS Dynamics | CRM |
| 5 | Oracle GL | GL |
| 6 | Oracle MFT | Secured file transfer |
| 7 | ENME | Sanction |
| 8 | SAS | Provisioning |
| 9 | Exus | Collection |
| LOS External 3rd Parties | | |
| 1 | GOSI | None government employee income details |
| 2 | Dakhli | government employee income details |
| 3 | Saudi Post SPL | All the National address record of the customer |
| 4 | Qarar | Qarar is credit decision and limit engine fetch and evaluates the customer cedit report from SIMH and provides the scoring, rating, rule-based pricing and decision (accept, reject or refer) |
| 5 | Unifonic | sending notification (SMS, IVR) |
| 6 | Emdha | Digital Signature service |
| 7 | DDCAP | This service is to be triggered for purchase/sale of commodity as per Sharia Law |
| 8 | Sanad-Nafith | Promissory Note- Nafith Platform, approves and certifies the promissory note and as the final step the promissory note is saved for future reference |
| 9 | Nafath | National single sign on and Absher details |
| 10 | Elm | sending OTP /getting Absher details, verify mobile ownership |
| 11 | ARB-Channel | Al-Rajhi Bank provides the information for if the customer is an ARB customer and its Account information, IBAN verification |
| 12 | Saudi Payment | IBAN verification for none ARB customers |
|  |  |  |
|  |  |  |
| LMS External 3rd Parties | | |
| 1 | Neolep PGW | Customer payment |
| 2 | ARB-Mokafaa | ARB Loyalty program |
| 3 | SIMAH | SIMAH file as part of compliance |
| 4 | Natheer | Natheer watch list |
| 5 | ARB-DD | ARB direct deduction file |
| 6 | Nafith | Cancel promissory note |
| 7 | MOJ Najez | Capture Court Case ID |

### GWs Integration

## Integration Tools and Systems

The integration layer offers other tools and systems for integration

### Apache Kafka

### IBM MQ

### Informatica ETL

### Oracle MFT

### Incorta

## C:\Users\user\AppData\Local\Microsoft\Windows\INetCache\Content.MSO\A33F8287.tmp

## Core Systems

Figure 1 Core Backends

|  |  |  |
| --- | --- | --- |
| **Core Backends** | | |
| **1** | Temenos Transact | LMS |
| **2** | Newgen iBPs | LOS |
| **3** | Kastle | ULS |
| **4** | MS Dynamics | CRM |
| **5** | Oracle GL | GL |
| **6** | ENME | Sanction |
| **7** | Exus | Collection |
| **8** | Emkan Payment System Hub PSH | Payment MS |
| **9** | Emkan Notification Engine | Notification engine |
| **10** | Emkan Rules Engine | Rule engine |
| **11** | SAS | Provisioning |

## Logical Application Architecture

Figure 2 LOS,LMS,CRM AND ENME LOGICAL ARCHTECTURE

### 1-Transact Loan Management System (LMS)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **System** | **Transact** | **Vendor Name** | | **Temenos** |
| **Description** | Core Banking system by Temenos used to manage LMS | | | |
| **Infrastructure** | OCI | **Deployment** | Containerization | |
| **Container Orchestrator** | OpenShift | **Release** | R22 | |
| **System Components** | | | | |
| **Transact Core** | LMS Core Banking | **Temenos Data Hub** | Analytics and reporting platform | |
| **Transact Backoffice** | Transact user interface for operation team | **Data Event Streaming DES** | Event streaming component capturing real-time transactions | |
| **IRIS Framework** | API Framework |  |  | |

### 2-Newgen iBPS LOS

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **System** | **Transact** | **Vendor Name** | | **Temenos** |
| **Description** | Core Banking system by newgen used to manage LOS | | | |
| **Infrastructure** | OCI | **Deployment** | Virtual machines | |
| **Container Orchestrator** | N/A | **Release** | xxx | |
| **System Components** | | | | |
| **iBPS** | Process engine and application server | **BRMS** | rule engine (will be used for retail products ) | |
| **OmniDocs** | DMS |  |  | |

### 3- MS-Dynamics

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **System** | **MS Dynamics** | **Vendor Name** | | **Microsoft** |
| **Description** | CRM System | | | |
| **Infrastructure** | OCI | **Deployment** | Virtual machines | |
| **Container Orchestrator** | N/A | **Release** | xxx | |
| **System Components** | | | | |
| **Front-End** | Web based front-end for operation and administration teams | **Back-end App** | Core CRM Logic | |
| **Integration services** | API layer |  |  | |

### 4- ENME

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **System** | **ENME** | **Vendor Name** | | **EJADA** |
| **Description** | Sanction, screening and Customer Rating | | | |
| **Infrastructure** | OCI | **Deployment** | Virtual machines | |
| **Container Orchestrator** | N/A | **Release** | xxx | |
| **System Components** | | | | |
| **ENME Front-end** | Web based front-end for operation and administration teams | **Screening engine** | Sanction and screening engine | |

### 1- Emkan Payment System Hub (PSH)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **System** | **Emkan Payment System Hub** | **Vendor Name** | | **Emkan** |
| **Description** | Payment system receive online payment from different sources and reflect it to core systems | | | |
| **Infrastructure** | OCI | **Deployment** | Containerization | |
| **Container Orchestrator** | Kubernetes | **Release** | N/A | |
| **System Components** | | | | |
| **Transact Dispatcher** | Reflects payment to Transact | **Kastle Dispatcher** | Reflects payment to Kastle | |
| **Cortex Dispatcher** | Reflects payment to cortex |  |  | |

### 2- Emkan Notification Engine

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **System** | **Emkan Notification Engine** | **Vendor Name** | | **Emkan** |
| **Description** | Handle multiple type of notifications sent across the system | | | |
| **Infrastructure** | OCI | **Deployment** | Containerization | |
| **Container Orchestrator** | Kubernetes | **Release** | N/A | |

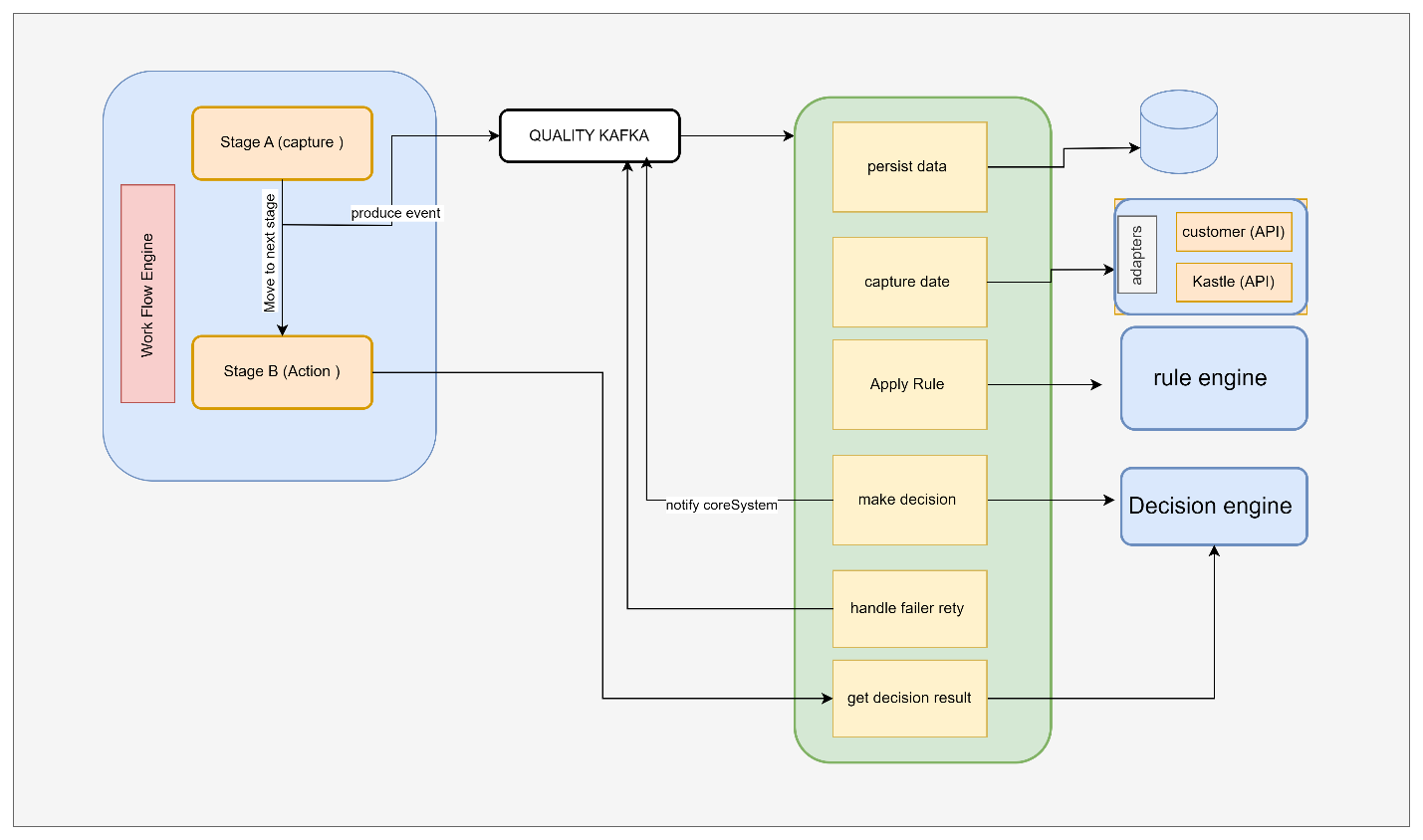
### 3- Emkan Rules Engine

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **System** | **Emkan Rules Engine** | **Vendor Name** | | **Emkan** |
| **Description** | Rules engine offer friendly user interface used by Emkan Business Team to define different product eligibility rules | | | |
| **Infrastructure** | OCI | **Deployment** | Containerization | |
| **Container Orchestrator** | Kubernetes | **Release** | N/A | |

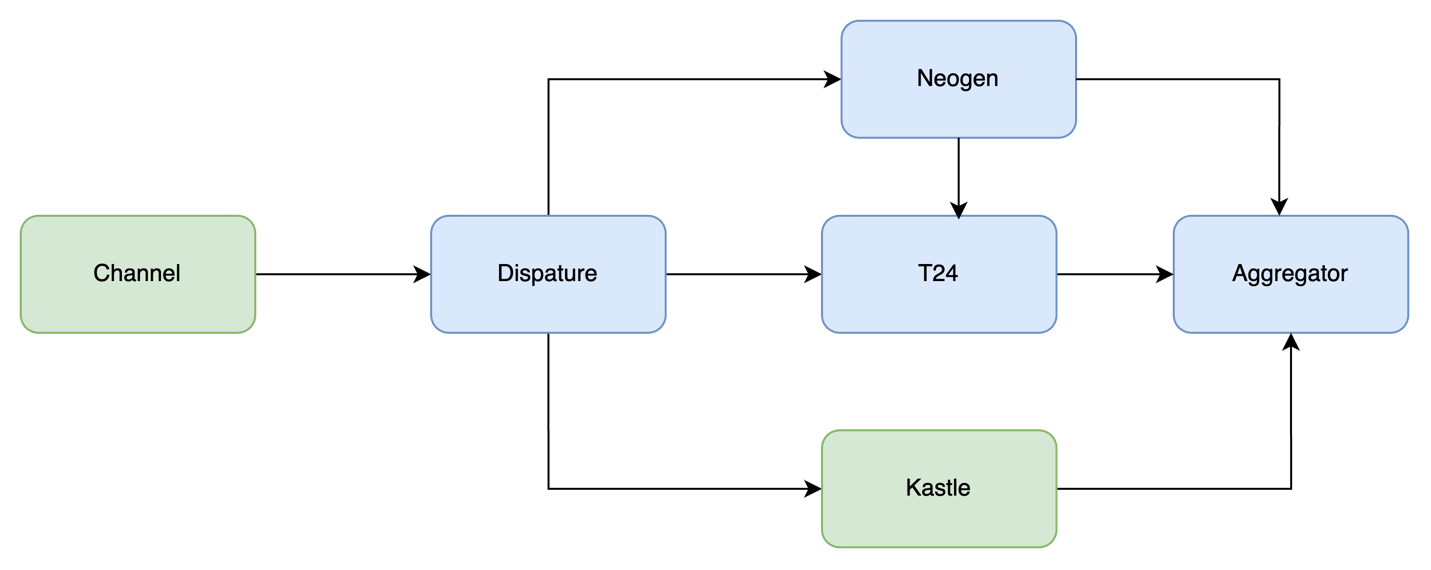
### 6- Registration Microservice

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **System** | **Emkan Rules Engine** | **Vendor Name** | | **Emkan** |
| **Description** | Handle customer onboarding and integrate with SPL | | | |
| **Infrastructure** | OCI | **Deployment** | Containerization | |
| **Container Orchestrator** | Kubernetes | **Release** | N/A | |

## Quality and rule engine



## Parallel Run and Coexistence



The migration project will be carried out as a series of drops each of them will have a group of Emkan’s Retail products. Business availability and continuity is a must.

During the project Emkan’s Retail Products Portfolio will be as the following:

* 1. Products not migrated and it will be availed through the Kastle stack
  2. Products migrated availed through the Newgen, Transact stack.
  3. Applications started and not completed before their products moved to the new technology stack at cutoff time will continue old stack until Loan contracts are created and will be migrated to new stack as delta migration

### Inquiries:

1. none migrated Products Inquires will be executed and fetched from old stack.
2. migrated Products Inquires will be executed and fetched from new stack.

To achieve these requirements two components would be implemented inside IBM integration layer

### 1-Aggregator Logic

Integration layer will handle the coexistence aggregation logic by consuming Kastle (LOS& LMS) endpoints and Transact and Newgen endpoints related to

* Dashboard (to retrieve active contracts and ongoing applications from different systems )
* Active and ongoing application on both system

For more details you can refer to execution flow

### 2-Dispatcher Logic

Mobile will call new APIs exposed on B2C GW and will reach to new stack

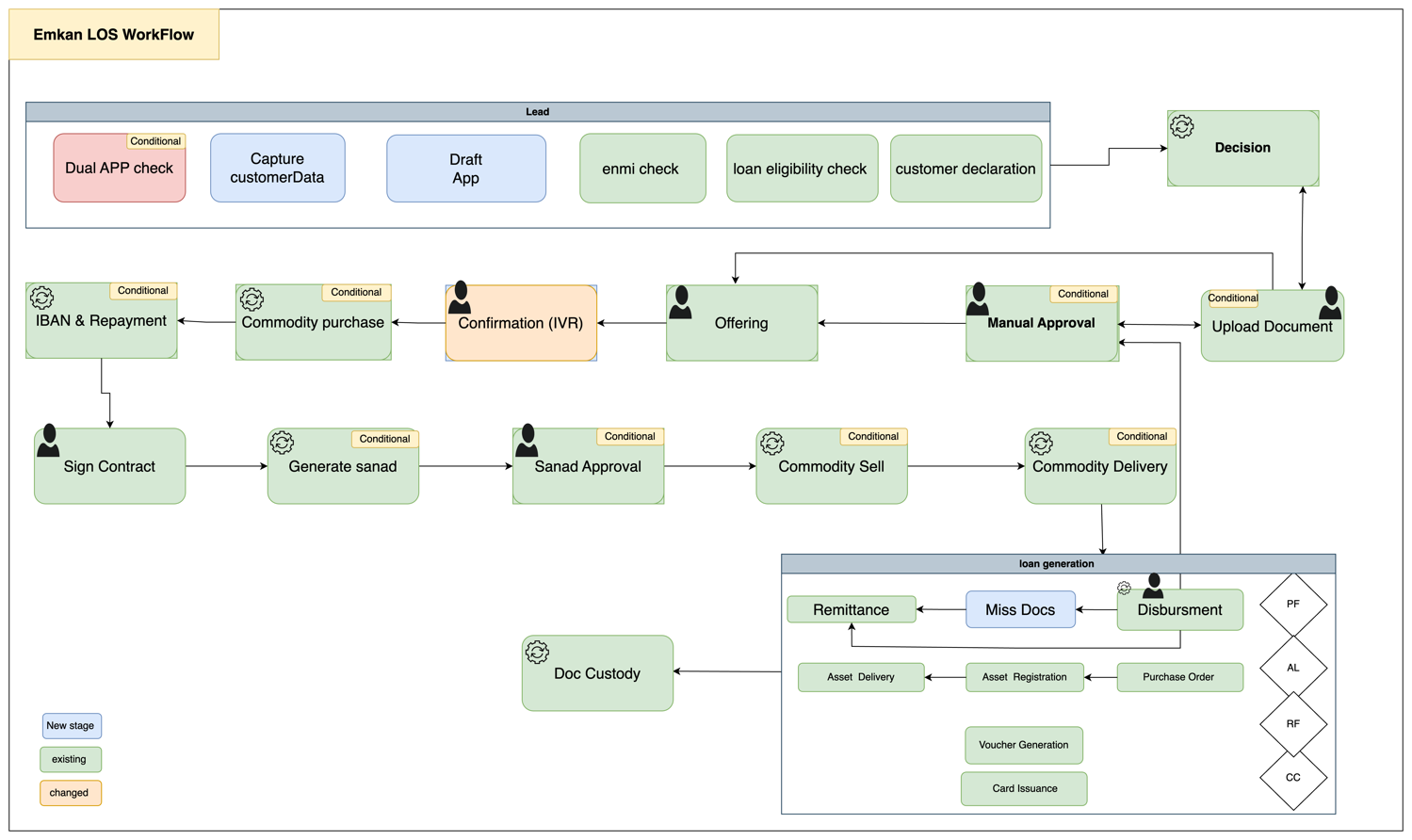
## Description

|  |  |  |
| --- | --- | --- |
| **Layer** | **Service/component** | **Role** |
| **Channels** | Staff | One Portal, Core Systems |
|  | Customer | Emkan Mobile App, Website |
|  | Partners | Merchant Portal, Merchant E-Commerce Platform |
| **Gateway** | B2C  IBM DataPower | Receiving traffic from Channel Customer |
| B2B | Receiving traffic from Channel Partners |
| IG | Receiving traffic from BFF and Middleware Integration Microservice layer |
| Dashboard | MS for doing transformation , aggregation,segregation ,... related to application dashboard communications like fetching all applications within stattistics from different core banking system,... |
| **Security** | IDAM(WSO2) | Identity Server for  identity and access management, |
| Identity Service | MS for identifying user [authentication/authorization] |
| Device Management | MS for managing registered device of customer |
| **Support Services** | Emkan NE | Notification Engine for sending notification like SMS,IVR, push Notification Message and feeback caller |
| Emkan Scheduler Engine | Scheduler Engine to Handling Job Tasks for Emkan |
| Emkan Rule Engine | Rule Engine to apply Ruel Sets |

# **Execution Flow View**

# Configuration

* + 1. **New Gen Configuration**:
* Configure below stage for all products



Description:

|  |  |  |
| --- | --- | --- |
|  | **Task Name** | **Description** |
| APP GENERATION | | * Validate if application already exists. * Capture customer personal information for creating an application. * Create Draft application. * Perform customer sanction and screening through ENMIE. * Apply loan eligibility rules * Capture customer information like Income,Expenses,total obligations as part of Customer Declaration. |
| Decision | | Final decision and final offer if accepted.  If any document missing return required documents as part of Qarar response. |
| Upload Documents | | Upload required documents. |
| Manual Approval | | Move application to underwrite (L1-L9 ) for manual check |
| Miss Document | | Ask customer to upload missed Document |
| Customer Offering Approval | | Customer decides accept or reject final offer |
| Confirmation stage [IVR] | | Triggering IVR call for loan confirmation |
| Commodity purchase maker | | Purchasing commodity |
| Sign contract | | Customer accept /reject Sign contract of all applications |
| Generate Sanad | | System generates PN for customer |
| Sanad Approval | | Customer decides accept or reject Sanad |
| Commodity sell maker | | Selling commodity |
| Disbursement | | Disbursement Stage |
| Remittance | | Remittance Stage |
| Document custody | | Complete of process |

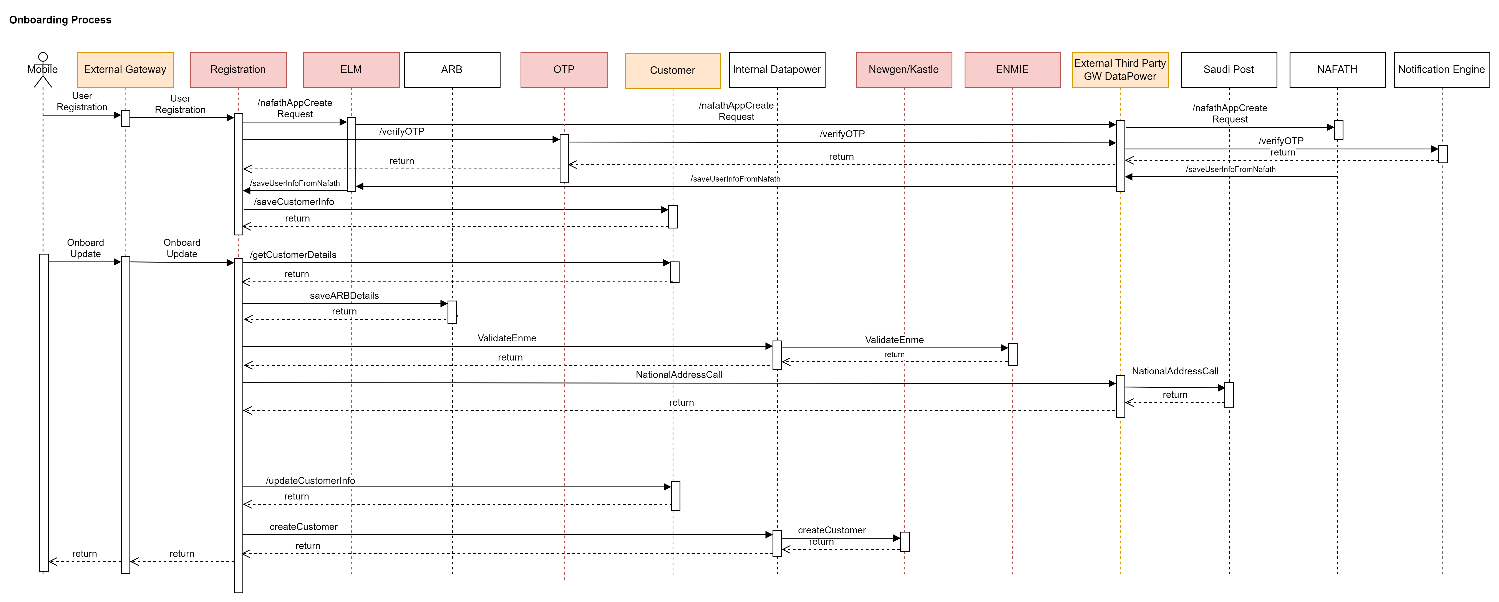
Configure Emkan Employee Users across all Emkan Functions which use Newgen Interface

* + 1. **DMS**
* Product Required Documents
  + Salary Certificate
  + ID Certificate
* Application Required Documents
  + Contract summary
  + Certificate of ownership
  + Digital selling contract
    1. **T24**
* Configure Accounting Finance cycle
  + 1. **Oracle GL**
* Configure Accounting Finance cycle
  + 1. **Qarar**
* AS IS except
  + - Disable Direct PF
    - If customer applying for TopUp or Micro Qarar should not offer DirectPF
    - Refinance will be applicable for topup and micro products only
    - Qarar to return required/missing documents.
    1. **Kastle**

1. Freeze DirectPF application
   * 1. **Payment Gateway**

* Configure new merchant at new-leap
  + 1. **Robotics**
* Configure RPT to use new screens on Newgen Portal

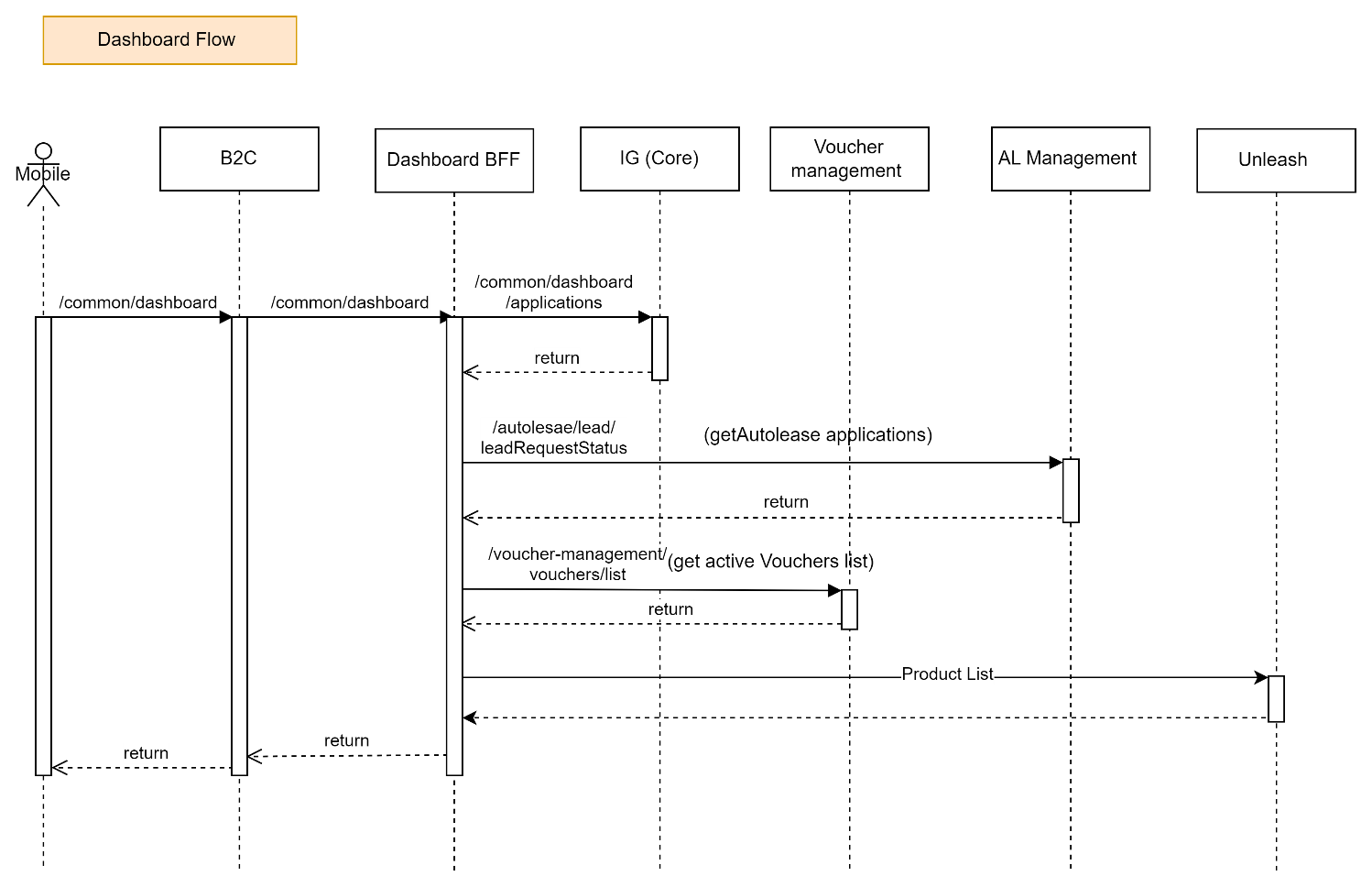
# Onboarding Flow

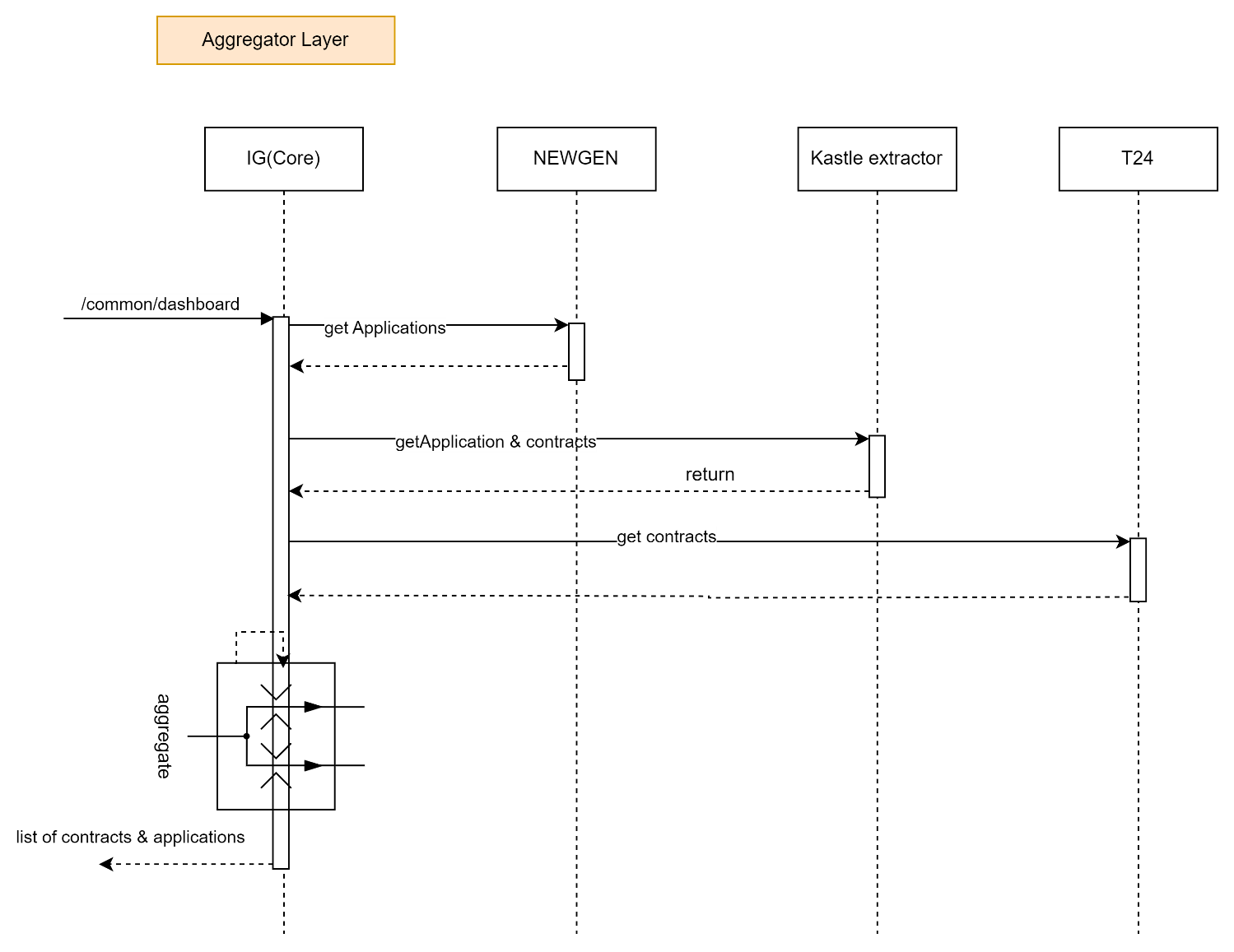


|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **System Provider** | **Action** | **Request** | **Response** | **Type** | **Description** |
| Registration | /nafathAppCreateRequest | AsIs | AsIs |  |  |
| OTP | /verifyOTP | AsIs | AsIs |  |  |
| Nafath | /saveUserInfoFromNafath | AsIs | AsIs |  |  |
| Customer | /saveCustomerInfo | AsIs | AsIs |  | Make sure request is in synch with Newgen. |
| Custome | /getCustomerDetails | AsIs | AsIs |  |  |
| ENME | /validateENME | AsIs | AsIs |  | As part of migration we will be calling ENME instead of Amlock.  Validate request and response from ENME. |
| Customer | /updateCustomerInfo | AsIs | AsIs |  |  |
| Customer | /createCustomerInfo | AsIs | AsIs |  |  |

# Dashboard Flow S2

1. Diagram

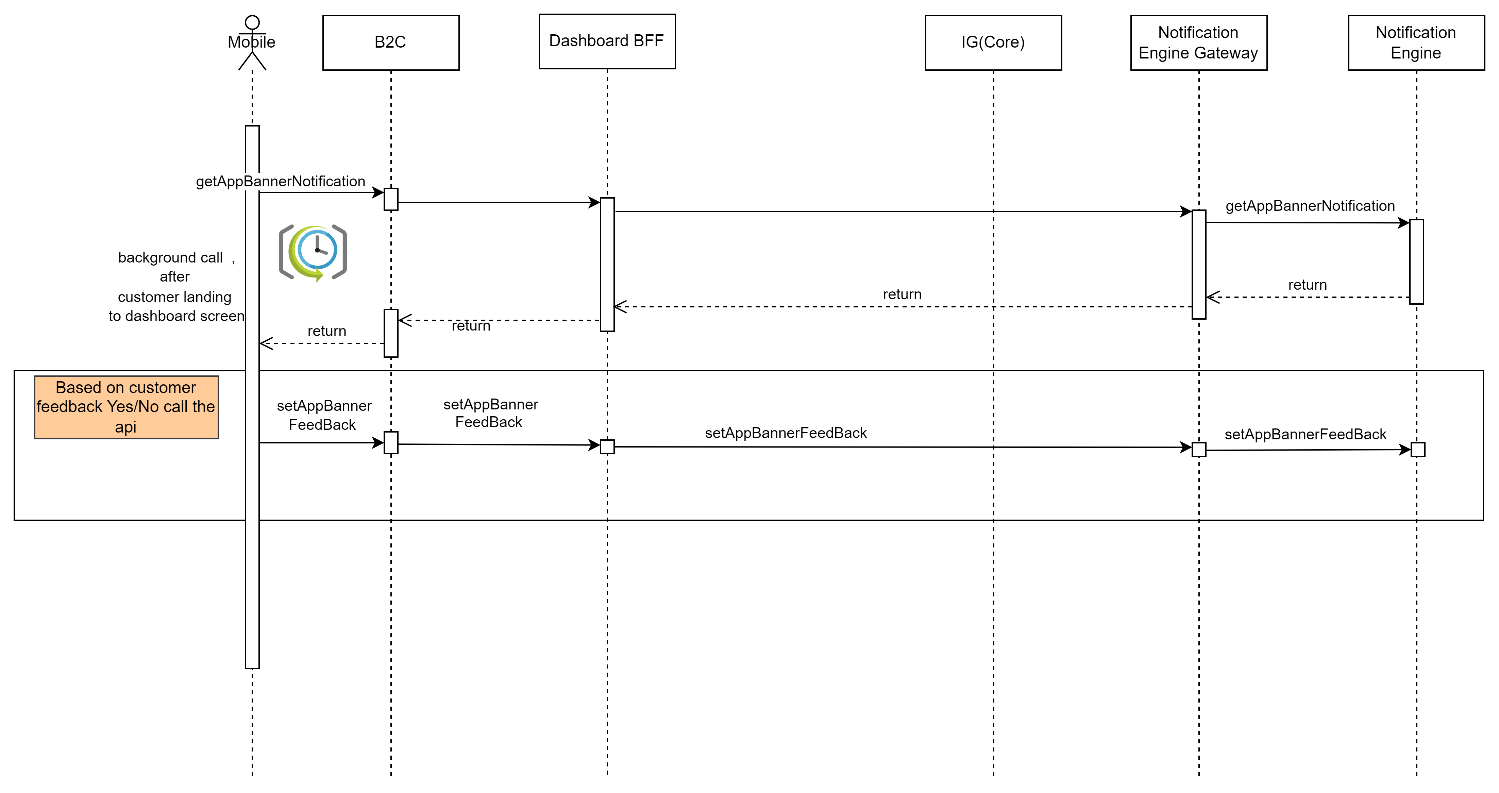




Description:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **System Provider** | **Action** | **Request** | **Response** | **Type** | **Description** |
| Dashboard | /common/dashboard | {  "status": "active",  "nationalId": "NID123456789"  } | {  "id": 123456,  "applicationId": "APP001",  "lastInstallmentDate": "2024-07-15",  "subProduct": "SubProduct A",  "product": "Product A",  "installmentAmount": 500.00,  "stage": "In Progress",  "totalFinancedAmount": 10000.00,  "amountRemaining": 3000.00,  "nextInstallmentDue": "2024-08-15",  "loannumber": "LN123456789",  "status": "active",  "substage": "Underwriting",  "subsourcingChannel": "Online",  "bp": "BP123",  "source": "Web",  "installmentRemainDuration": 5,  "remainingInstallmentNumber": "5",  "reasonCode": "RC001",  "reasonMessage": "Payment overdue",  "reasons": [  {  "reasonCode": "RC001",  "reasonmsg": "Payment overdue"  },  {  "reasonCode": "RC002",  "reasonmsg": "Insufficient funds"  }  ]  } |  | Instead of calling Kastle-Extractor call Integration Layer(Internal Data power) end point which will aggregate response from T24/Kastle/Newgen and send all applications. |
| Autolease | /leadRequestStatus | nationalId  phoneNumber | nationalId  dealerOffers  {  leadId,  leadStatus  }  autoleaseApplications{  } | REST/JSON |  |
| Voucher-Management | /vocucherList | nationalId | voucherId  amount  redeemedAmount  currency  merchantCode  applicationId  customerId  state  lastStateChange  piryDate  transactionId  voucherCode  createdAt  categoryCode  voucherIssuerName | REST/JSON |  |

AppBanner Dashboard Flow:

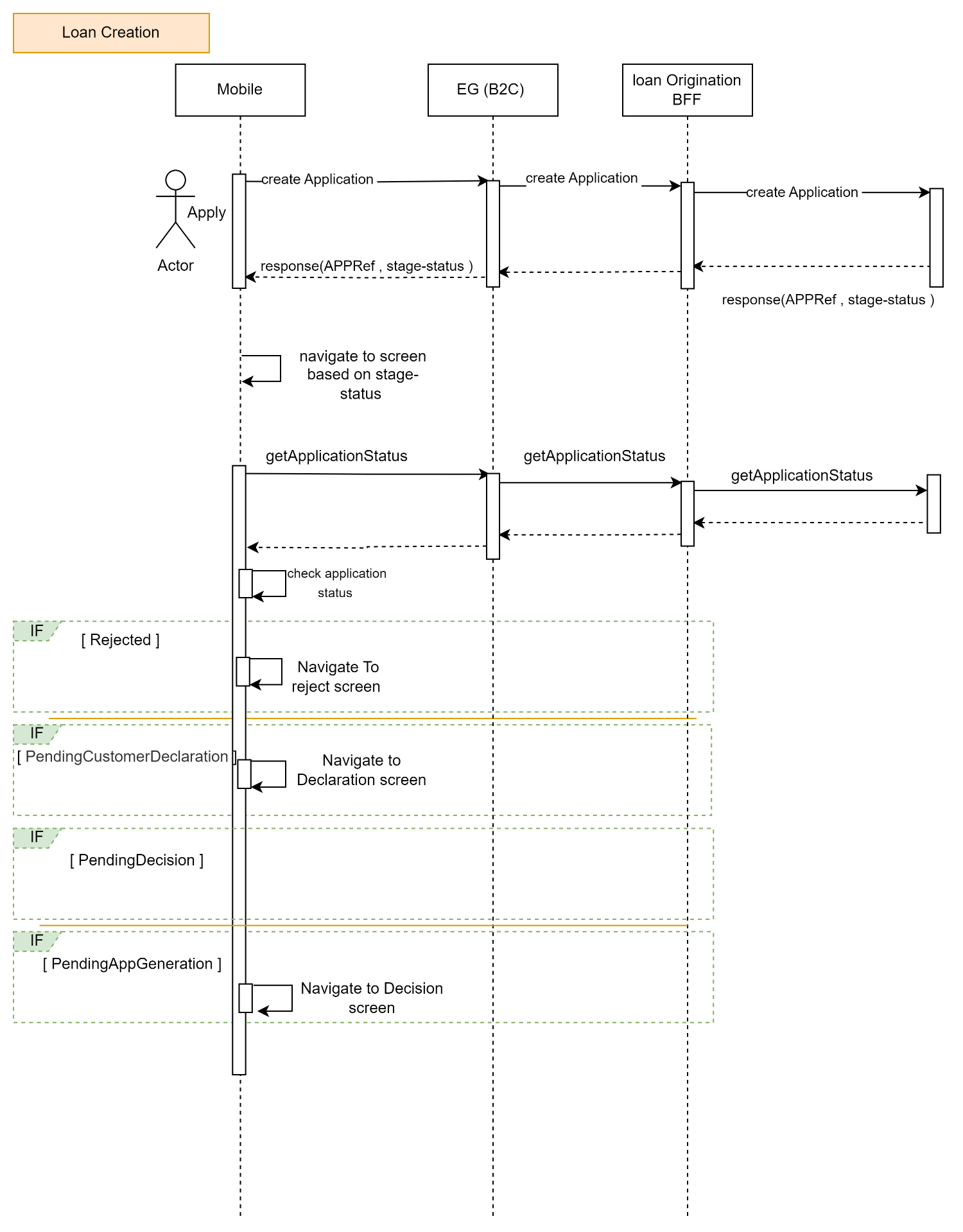


|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **System Provider** | **Action** | **Request** | **Response** | **Type** | **Description** |
| Dashboard | /common/dashboard | {  "status": "active",  "nationalId": "NID123456789"  } | {  "id": 123456,  "applicationId": "APP001",  "lastInstallmentDate": "2024-07-15",  "subProduct": "SubProduct A",  "product": "Product A",  "installmentAmount": 500.00,  "stage": "In Progress",  "totalFinancedAmount": 10000.00,  "amountRemaining": 3000.00,  "nextInstallmentDue": "2024-08-15",  "loannumber": "LN123456789",  "status": "active",  "substage": "Underwriting",  "subsourcingChannel": "Online",  "bp": "BP123",  "source": "Web",  "installmentRemainDuration": 5,  "remainingInstallmentNumber": "5",  "reasonCode": "RC001",  "reasonMessage": "Payment overdue",  "reasons": [  {  "reasonCode": "RC001",  "reasonmsg": "Payment overdue"  },  {  "reasonCode": "RC002",  "reasonmsg": "Insufficient funds"  }  ]  } |  | Instead of calling Kastle-Extractor call Integration Layer(Internal Data power) end point which will aggregate response from T24/Kastle/Newgen and send all applications. |
| Dashboard BFF | getAppBannerNotification | AsIs | AsIs |  | Response should contain  eventCode,feedback,notificationStage,message,action,title,nid. |
| Dashboard BFF | setAppBannerFeedBack | AsIs | AsIs |  |  |

# Loan Eligibility Flow (Lead Generation Flow ). S1

* Diagram

**TOPUP & MICRO**



**REFINANCE**



A screenshot of a computer

Description automatically generated

Description:

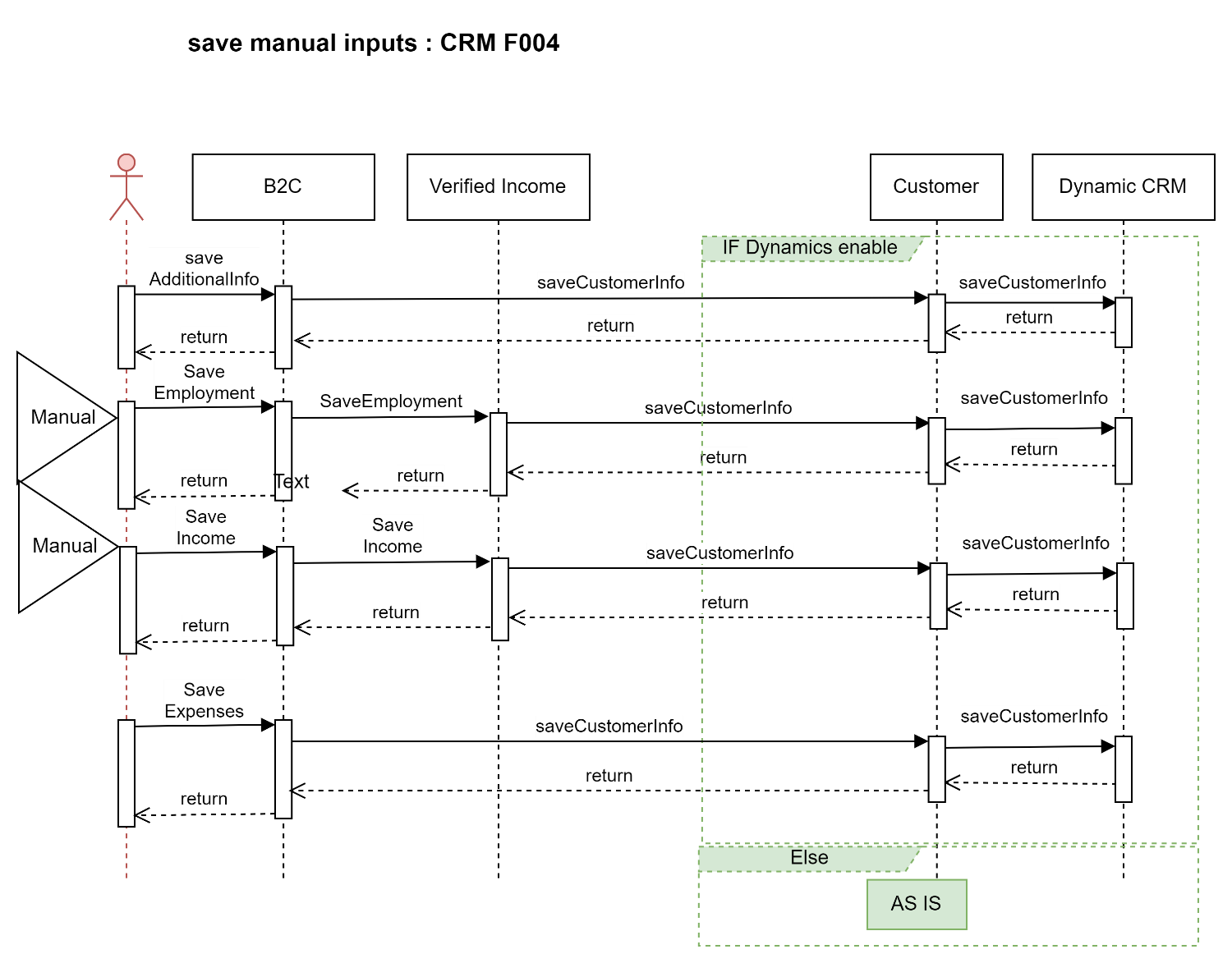
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **System Provider** | **Action** | **Request** | **Response** | **Type** | **Description** |
| Loan-Origination | /getActiveContractsEligibileForRefinance | nationalId  preferredLanguage | Proceedflag  Reason  transactionId  refinanceApplications{  appId  subProductId  paidAmount  totalAmount  paidPercentage  isRecommended  } | Rest/Json |  |
| Loan-Origination | /createApplication | nid  emkanProduct  subProduct  preferredLanguage | compAppId,  stageStatus,  emkanProduct,  subProduct | Rest/Json |  |
| CoreSystem | LoanEligibility | Nid  productCode | proceedFlag | Rest/Json | Perform Loan Eligibility if eligible for Refinance |
| CoreSystem | getActiveApplications | Nid | Application information  contractId | Rest/Json | Get all details related to applications along with contractId |
| CoreSystem | getLoanInfoByContractId | Nid  contractId | Status,  Tenure,  totalInstallments,  remainingInstallments,  totalFinancingAmount,  remainingBalanceAmount,  lastInstallmentDate,  nextInstallmentDate,  apr,  excessAmount,  principleoutstanding,  profitoutstanding,  totalprofitdue,  earlyterminationfees,  remainingunpaidinstallment,  paymentschedule  {  Applicationno,  installmentAmount,  principalAmount,  pricipalpaid,  profitAmount,  profitPaid,  dueDate,  installmentNo,  status  } |  | Fetch Loan information and perform Refinance Eligibility check AS IS. |
| Loan-Origination | getApplicationStatus | Nationalid  compAppId  preferredLanguage | compAppId,  stageStatus,  emkanProduct,  subProduct |  | Based on status navigate to respective screen. |

# Customer declaration s2

* Diagram

A screenshot of a computer screen

Description automatically generated



Description:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **System Provider** | **Action** | **Request** | **Response** | **Type** | **Description** |
| Customer | /getCustomerReviewDetails | "nid": "1184113444",  "compAppId": "1",  "dateOfBirthHijri": "1440-02-01"  "getCustomerInfo" : ["basicCustInfo","addInfo","salInfo","expInfo","empInfo","autoLeaseCustomerInfo","arbAccounts","ibanInfo","buyOutInfo","naInfo","companyInfo","loanDetails"]   |  | | --- | |  | |  | |  | |  | | "addInfo": {    },  "salInfo": [  {    }  ],  "expInfo": {    },  "empInfo": {    },  "autoLeaseCustomer": {    },  "arbAccounts": [],  "naInfo": {  "Addresses": [  {    }  ]  },  "compInfo"  "loanDetails":[  {  }]  ibanInfo:  {  }  "basicCustInfo": | Rest/Json | Internally calls api from CRM to fetch customer related information. |
| CRM | getCustomerProfile | Nid  "getCustomerInfo" : ["basicCustInfo","addInfo","salInfo","expInfo","empInfo","autoLeaseCustomerInfo","arbAccounts","ibanInfo","buyOutInfo","naInfo","companyInfo","loanDetails"] | nationalId  arbcic  isArB  emkanCustomerInfo{  }  addInfo{  }  salInfo[  {  }  ]  expInfo{  }  empInfo  {  }  "naInfo": {          "Addresses": [  ]  }  arbAccounts  {  }  autoLeaseCustomer  {  } | Rest/Json | Fetch customer information from CRM DB. |
| Verified-Income | /employment-incomeinfo | asIs | asIs | Rest/Json | Fetch employment income information from third party when cooling period is off. |
| Customer | /saveAdditionalInfo | addInfo  "dataCategory": "addInfo",  "source": "M",  "info": {  "loanPurpose": "HMREN",  "maritalStatus": "MARRIED",  "familySize": "3",  "educationalFees": "1",  "houseOwner": "RENTH",  "workingStatus": "SAL",  "associateCode": "EMK00222",  "source": null,  "occupationId": "5",  "occupationName": null,  "politicalExposed": "1",  "otherOccupation": null  } | status | Rest/Json |  |
| CRM | /saveCustomerInfo |  | status |  | This api will be exposed by CRM to emkan. Emkan will call this api to save customer related information in Dynamic CRM.  Below objects will be passed as part of this request.  "basicCustInfo","addInfo","salInfo","expInfo","empInfo","autoLeaseCustomerInfo",ibanInfo","buyOutInfo","naInfo","companyInfo","loanDetails" |
| Customer | /saveIncome | salInfo  {  "source": "M",  "dataCategory": "incomeInfo",  "info": {  "basicSalary": 10000,  "housingAllowances": 2100,  "monthlyAllowances": 3000,  "extraIncome": 122  },  "nid": "{{nid}}"  } | status |  | This api will internally call /saveCustomerInfo api exposed by CRM. |
| Customer | /saveEmployment | empInfo  {  "nid":"1928403847",  "customerEmployment":    {  "fullName":"omar Ahmed",  "basicWage":"12000",  "housingAllowance":"2000",  "otherAllowance":"2000",  "employerName":"شركة ميد التجارية",  "dateOfJoining":"22/08/2020",  "workingMonths":"12",  "employmentStatus":"نشيط",  "salaryStartingDate":"2015/06/12 00:00:00.000",  "establishmentActivity":"",  "channel":"MANUEL",  "source":"M",  "sectorId":"GOV",  "departmentID":"MIN00006",  "militaryRank":"WAKEEL",  "active":1,  "updatedAt":"",  "manualCoolingPeriod":""  }    } | status |  | This api will internally call /saveCustomerInfo api exposed by CRM. |
| Customer | /saveExpenses | expensesInfo  {  "nid": "1662468493",    "source": "manual",  "dataCategory": "expInfo",  "info": {  "totalIncome": "244",  "housing": "22",  "expatFees": "22",  "foodBeverage": "22",  "tuitionFees": "22",  "healthcareInsurance": "2",  "transportationCommunication": "22",  "totalAdditionalExpenses": "22",  "totalObligations": "22",  "otherLoans": "false",  "otherLoansWithBank": ""  }  } | status |  | This api will internally call /saveCustomerInfo api exposed by CRM. |



**IRD for reference**

# Desision

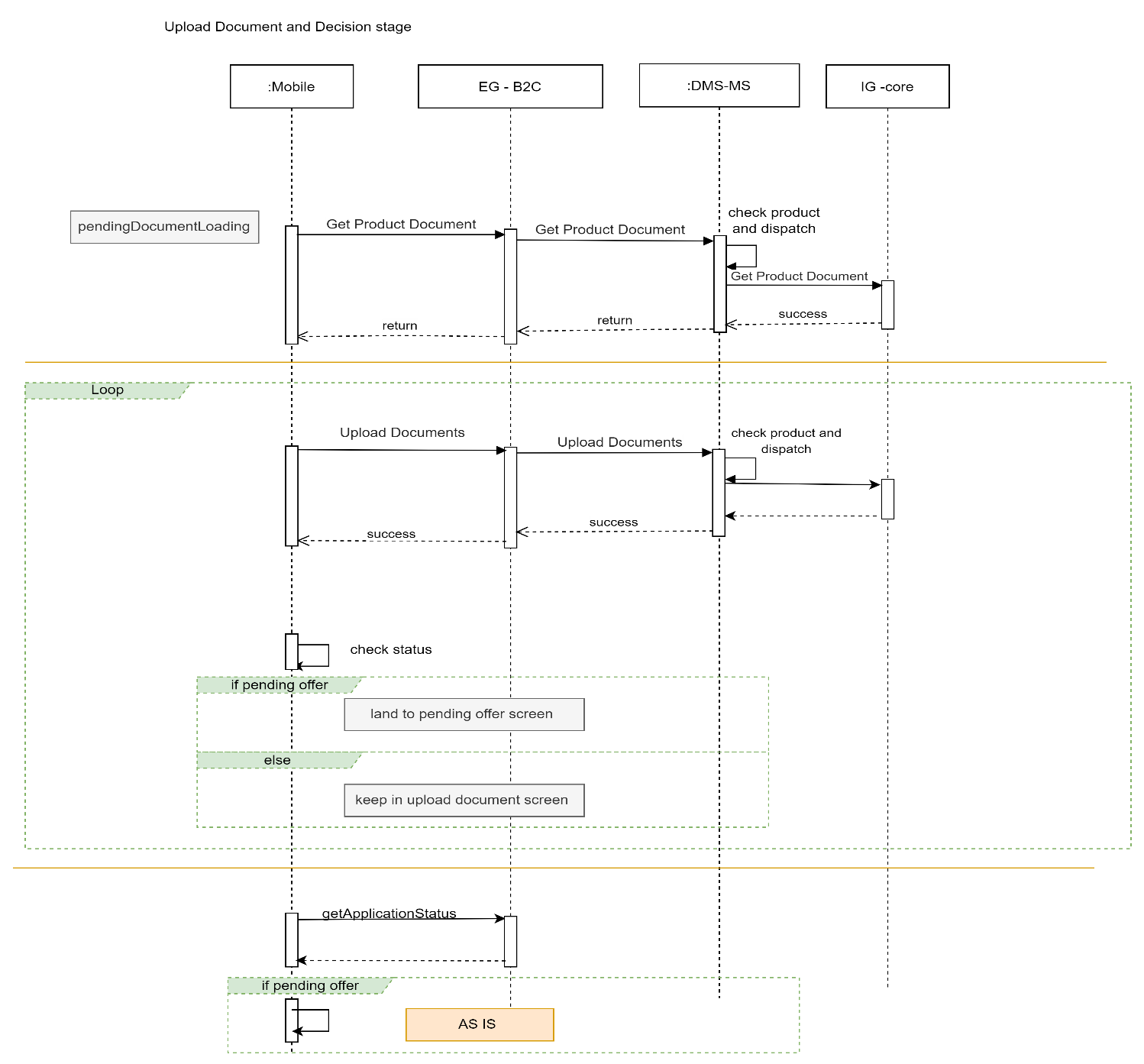
Design Overview

* Diagram

# Upload Document Flow S3

Design Overview

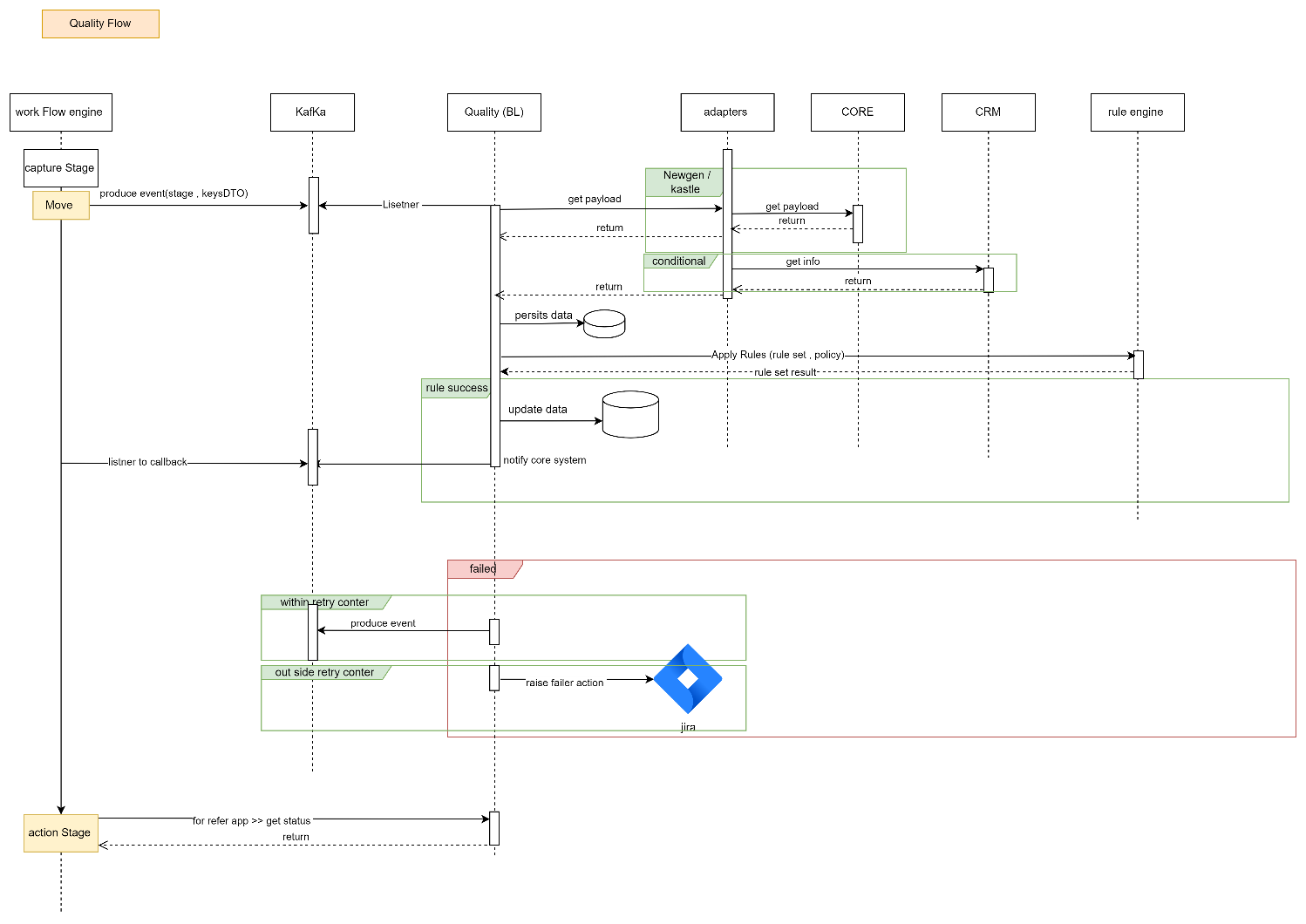
* Diagram



Description:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **System Provider** | **Action** | **Request** | **Response** | **Type** | **Description** |
| DMS | /getProductDocument | documentCode  cic  nationalId  source | source  documents  {  fileContent  filename  docId  } | Rest/Json |  |
| DMS | /uploadDocuments | compAppId  docId  replaceOrAddDoc  data  fileName | status  message  attachmentresponse  {  filename  uploadstatus  } |  |  |
| Loan-Origination | getApplicationStatus | nationalid  compAppId  preferredLanguage | compAppId,  stageStatus,  emkanProduct,  subProduct |  | Based on status navigate to respective screen. |

# Quality Flow



Description:

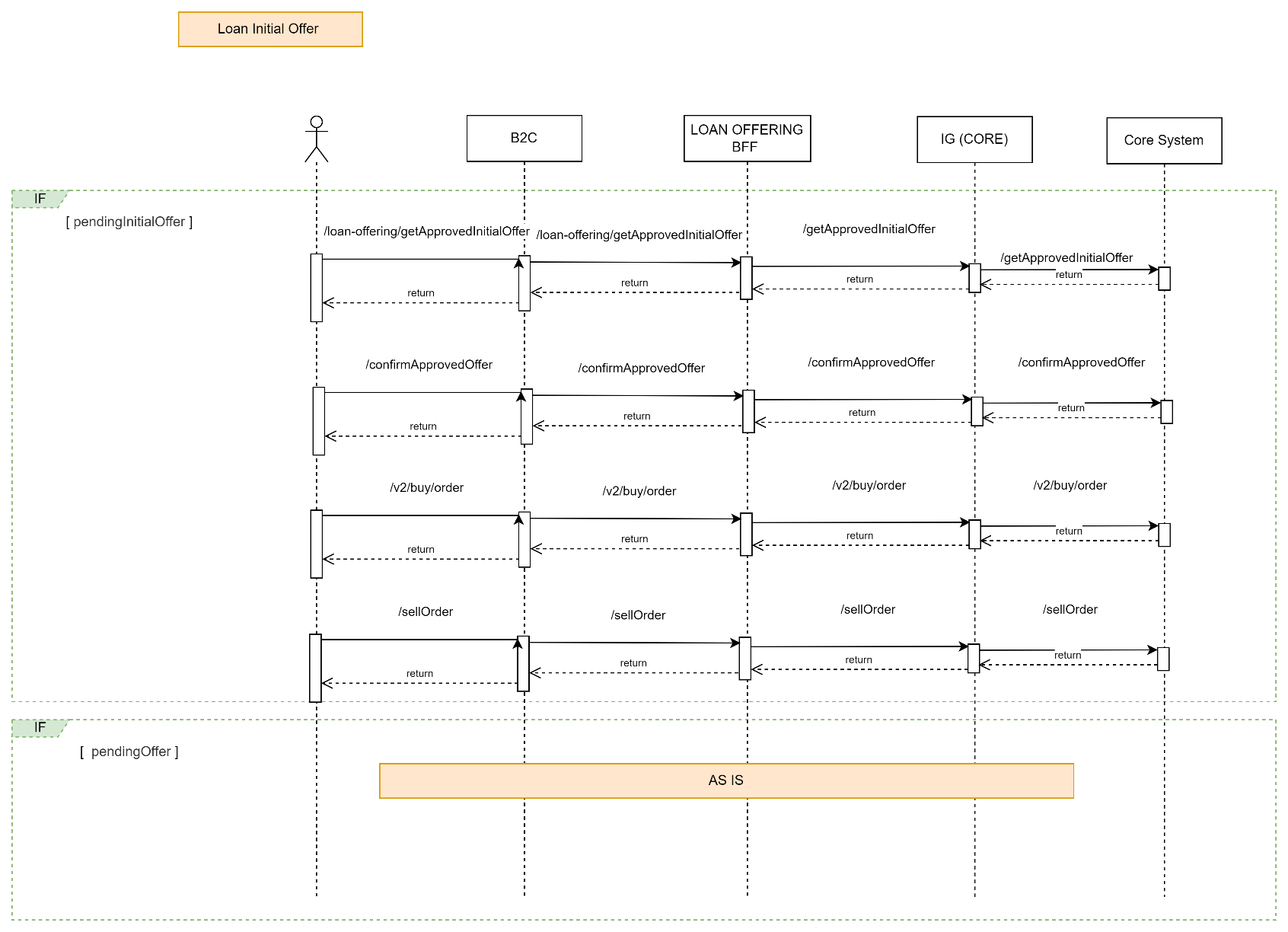
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **System Provider** | **Action** | **Request** | **Response** | **Type** | **Description** |
| Kafka | /send/{topic}  Topic Name: QTY.QUALITY\_EVENTS | eventCode  captureStageKey(Create Application)  compAppId  nationalId  emkanProduct |  | Kafka Queue | Push request to Kafka Topic  Topic Name: QTY.QUALITY\_EVENTS |
| Quality-Control | /status | eventCode  stageKey  compAppId  nationalId  containerId  emkanProduct | message  action  containerName  captureStage |  |  |
| Kafka | /callback/{topic} | compAppId  nationalId  emkanProduct  status |  |  |  |

# Offering Flow

1. Initial Offer

Design Overview

* Diagram



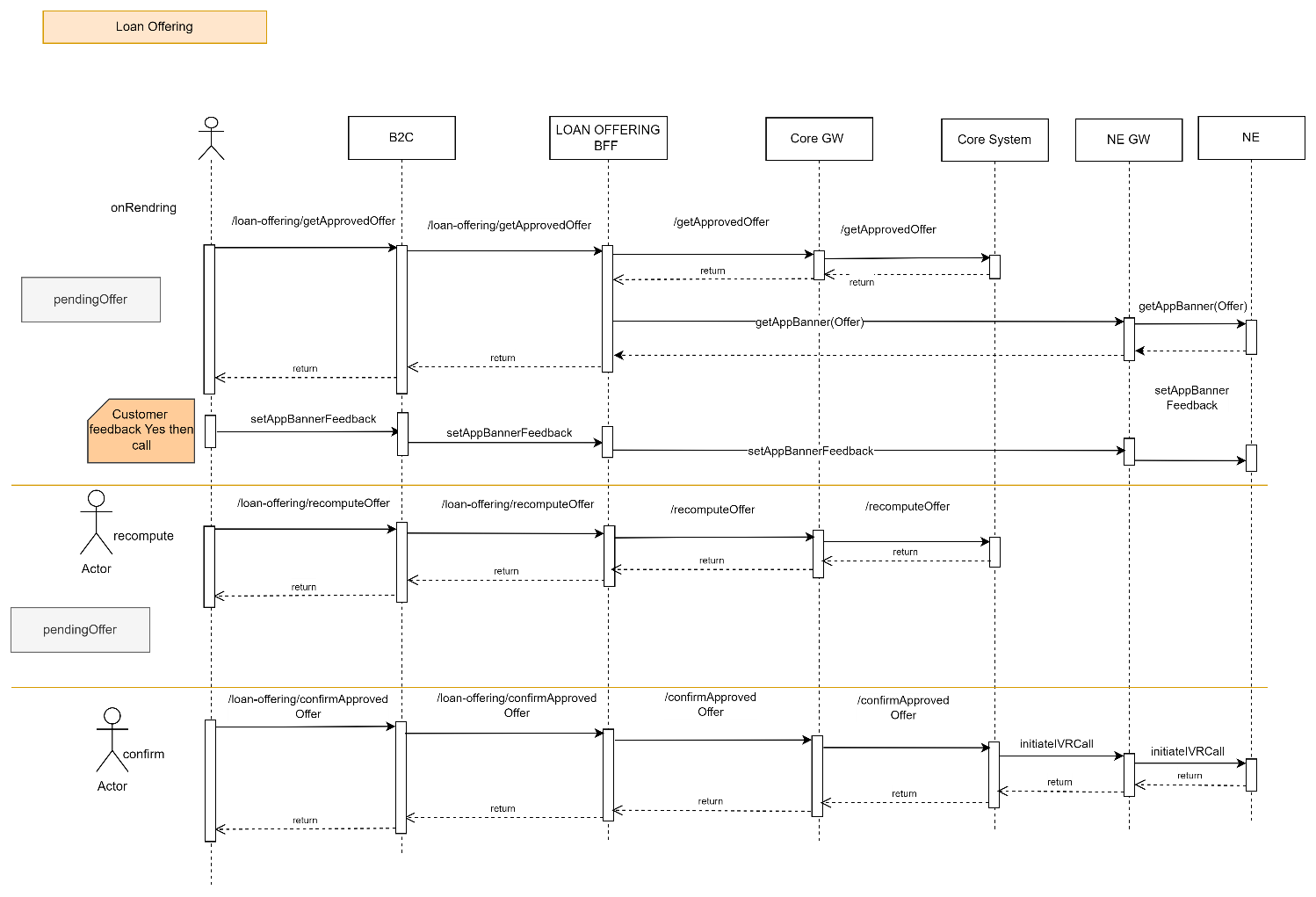
Description:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **System Provider** | **Action** | **Request** | **Response** | **Type** | **Description** |
| Loan-Offering | /getApprovedInitialOffer | nationalId  compAppId  preferredLanguage | compApplId  requestFinanceAmount  offerFinanceAmount  currency  purchasePrice  totalDownPayment  profitAmount  totalFinanceAmount  financeAmount  minimumFinanceAmount  maximumFinanceAmount  installmentAmount  extraAmount  apr  flatRate  debitAmount  tenure  minimumTenure  maximumTenure  minimumAmount  maximumAmount  managementFees  totalFees  vat  commodityFees  subProductCode  voucherAmount  totalAmountWithProfit  deferredLoan  deferredMessage | Rest/Json |  |
| Loan-Offering | /confirmApprovedOffer | AsIs | AsIs |  |  |
| Loan-Offering | /v2/buy/order | AsIs | AsIs |  |  |
| Loan-Offering | /sellOrderForRefinance | AsIs | AsIs |  |  |

1. Final Offer

Design Overview

* Diagram

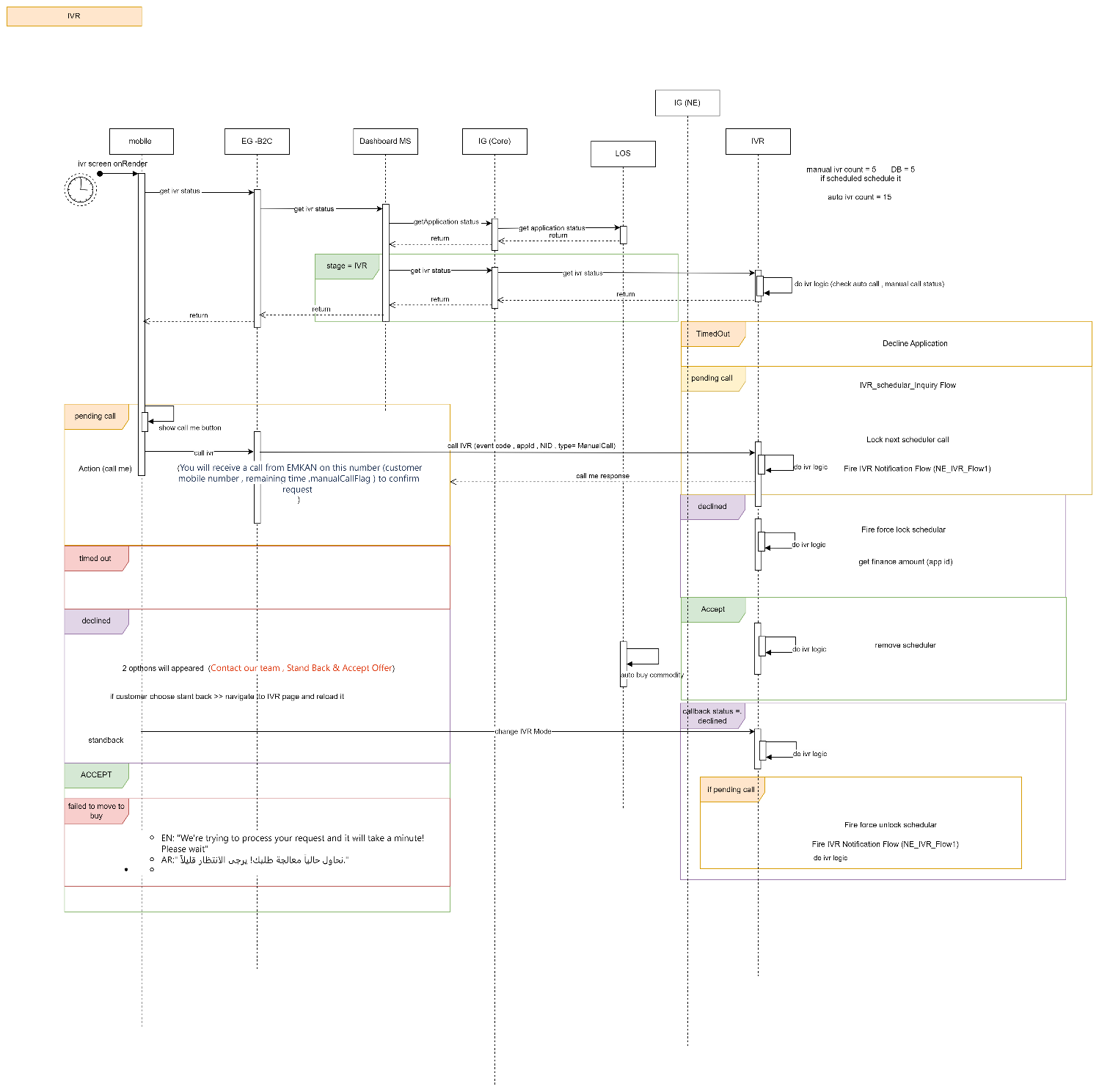


Description:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **System Provider** | **Action** | **Request** | **Response** | **Type** | **Description** |
| Loan-Offering | /getApprovedOffer | AsIs | AsIs | Rest/Json |  |
| Loan-Offering | /recomputeOffer | AsIs | AsIs | Rest/Json |  |
| Loan-Offering | /confirmApprovedOffer | AsIs | AsIs | Rest/Json |  |

# IVR Flow

Mobile Flow



Callback Flow:

A screen shot of a computer

Description automatically generated

Description:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **System Provider** | **Action** | **Request** | **Response** | **Type** | **Description** |
| IVR | /getIVRStatus | AsIs | AsIs | Rest/Json |  |
| Loan-Origination | getApplicationStatus | nationalid  compAppId  preferredLanguage | compAppId,  stageStatus,  emkanProduct,  subProduct |  | Based on status navigate to respective screen. |

# BuyCommodity Flow

Design Overview

* Diagram

A black screen with white squares

Description automatically generated

Description:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **System Provider** | **Action** | **Request** | **Response** | **Type** | **Description** |
| NewGen | buyCommodity | AsIs | AsIs | Rest/Json |  |
| NewGen | updateBrokerCertificate | AsIs | AsIs | Rest/Json |  |
| T24 | createIslamicContract | AsIs | AsIs |  |  |
| NewGen | updateStatus |  |  |  | Post succesful purchase of commodity, Newgen to update status to pendingIbanVerification. |

* NewGen Diagram

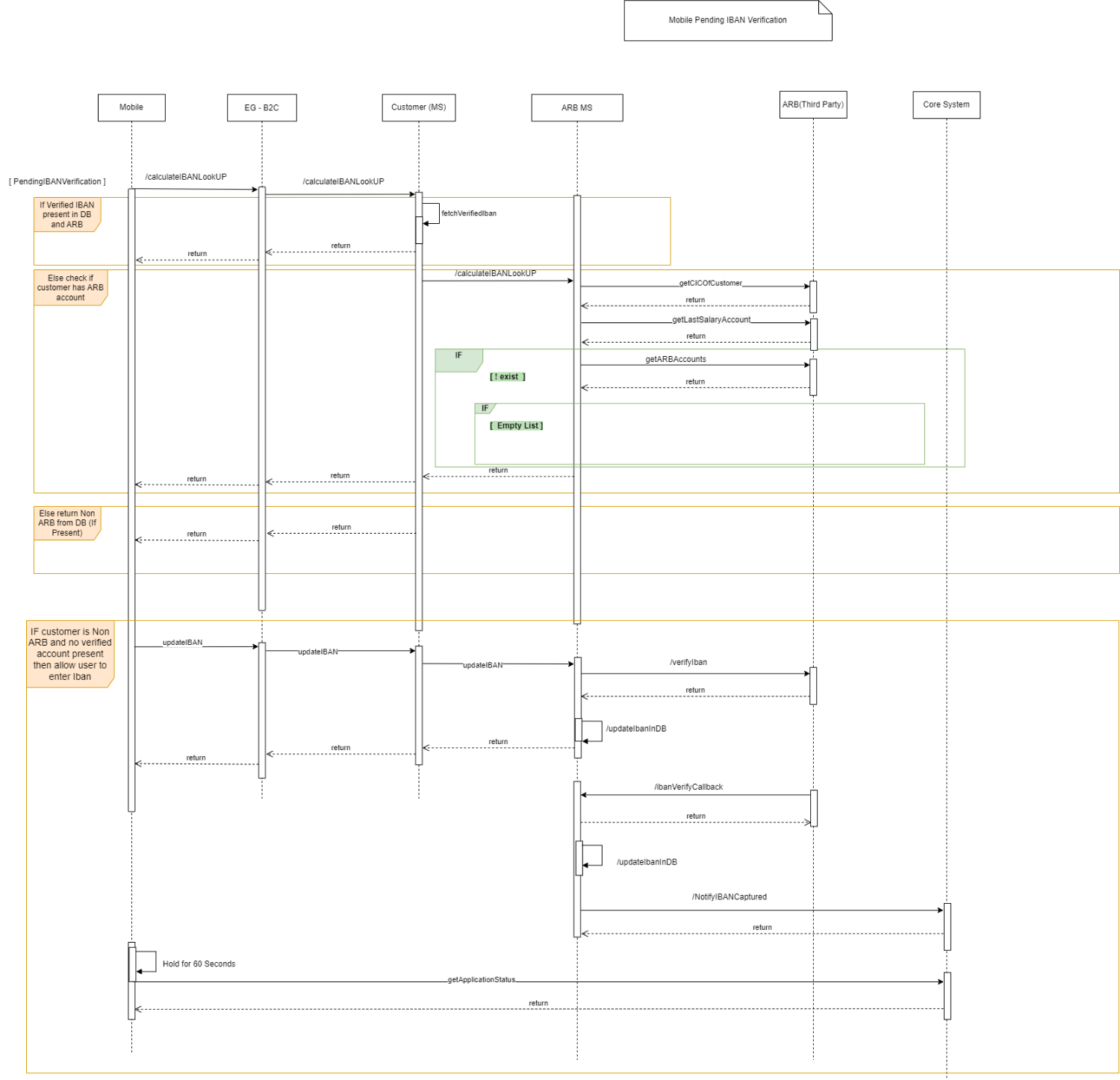
A screenshot of a computer

Description automatically generated

Description

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **System Provider** | **Action** | **Request** | **Response** | **Type** | **Description** |
| Newgen | /NotifyIbanCaptured | NID  Message: Iban Updated Successfully | Status  NID  ibanInfo | Rest/Json | 1) Newgen to call /calculateIbanLookup to fetch verified ibans from DB.  2) If verified iban is non arb then calculateibanlookup will call ARB to check if any ARB account exists.  3) Once iban information is gathered. Newgen to update repayment mode and disbursment account and status accordingly. |

# IBAN Verification Flow

* Mobile Diagram

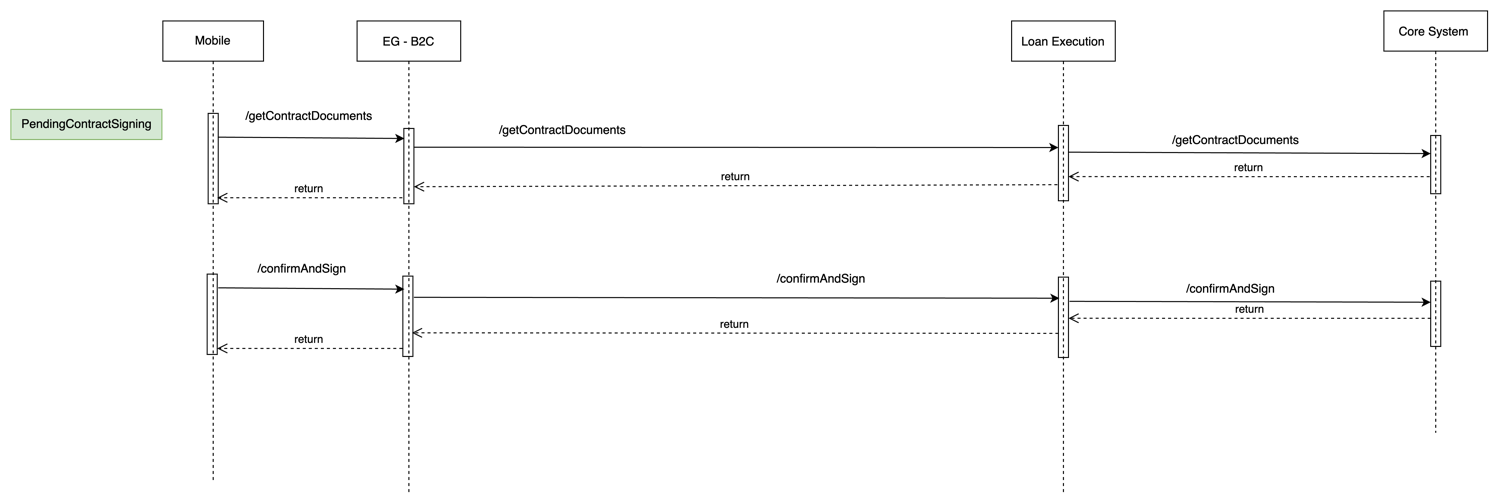
Description

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **System Provider** | **Action** | **Request** | **Response** | **Type** | **Description** |
| Customer | /calculateIBANLookUp | NationalId  ApplicationId | nationalId  ibanNumber  active  bankCode  status  primaryIban  verifiedFlag  isARB | Rest/Json | **FetchVerifiedIban:**  1)Fetch Iban information from DB  2) If iban is ARB and verified then return the response.  3) If iban from DB is non ARB then validate if ARB accounts are present by calling arbAccounts.  4) If no ARB account details present then return NON ARB account fetched from DB. |
| ARB | getCICOfCustomer | NID | cicNumber | Rest/Json | Get CIC Number of customer to fetch ARB accounts |
| ARB | getLastSalaryAccount | NID  arbCIC | isARB  accountNumber  iban  bban  accountName  accountStatus  accountOpeningDate  accountClosingDate | Rest/Json | If Last Salary account is present then return this response |
| ARB | getARBAccounts | NID  arbCIC | isARB  accountNumber  iban  bban  accountName  accountStatus  accountOpeningDate  accountClosingDate | Rest/Json | Return list of ARB accounts and by default pick the first record. |
| Customer | updateIban | NID  IBAN | NID  verifiedFlag  iban  bankCode  active  status  primaryIban  accountNumber | Rest/Json | Enter Non ARB iban from screen.  Save iban in iban\_verification table.  Mark primaryIban as Y |
| ARB | /verifyIban | Iban  poiType  poiNumber  source  IbanMetaDto  {  Name  Value  } | AsIs | Rest/Json | Please refer request and response from Sarie microservice.  /verifyIban |
| ARB | /ibanVerifyCallBack | AsIS | AsIS | Rest/Json | Callback recieved from ARB.  Update callback information in DB. |
| ARB | updateIbaninDB |  |  |  | 1) Update iban information into DB once verifyIban is completed.  2) Update iban information in DB once callback recieved from ARB |
| Newgen | /NotifyIbanCaptured | NID  Message: Iban Updated Successfully | Status  NID  ibanInfo |  | Once callback was received succesfully. Notify Newgen with iban status. |
| Newgen | /getApplicationStatus | NID  ApplicationId | Status |  | 1) Mobile will hold for 60 seconds post updateIban.  2) Get latest status from Newgen and proceed based on the status. |

# ContractSigning Flow

Design Overview

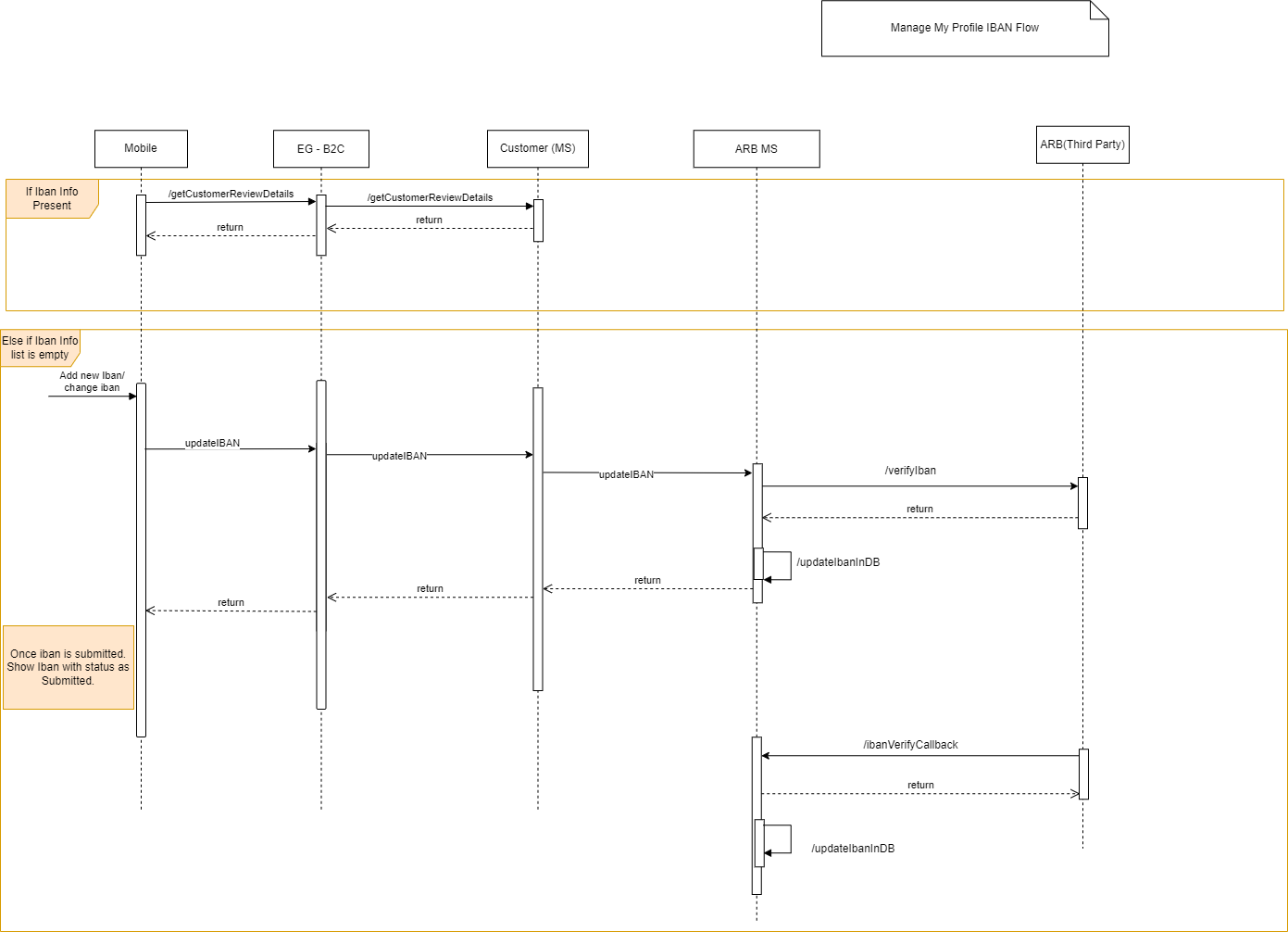
* Diagram



Description

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **System Provider** | **Action** | **Request** | **Response** | **Type** | **Description** |
| Loan-Execution | getContractDocuments | AsIs | AsIs | Rest/Json |  |
| Loan-Execution | /confirmAndSign | AsIs | AsIs | Rest/Json |  |

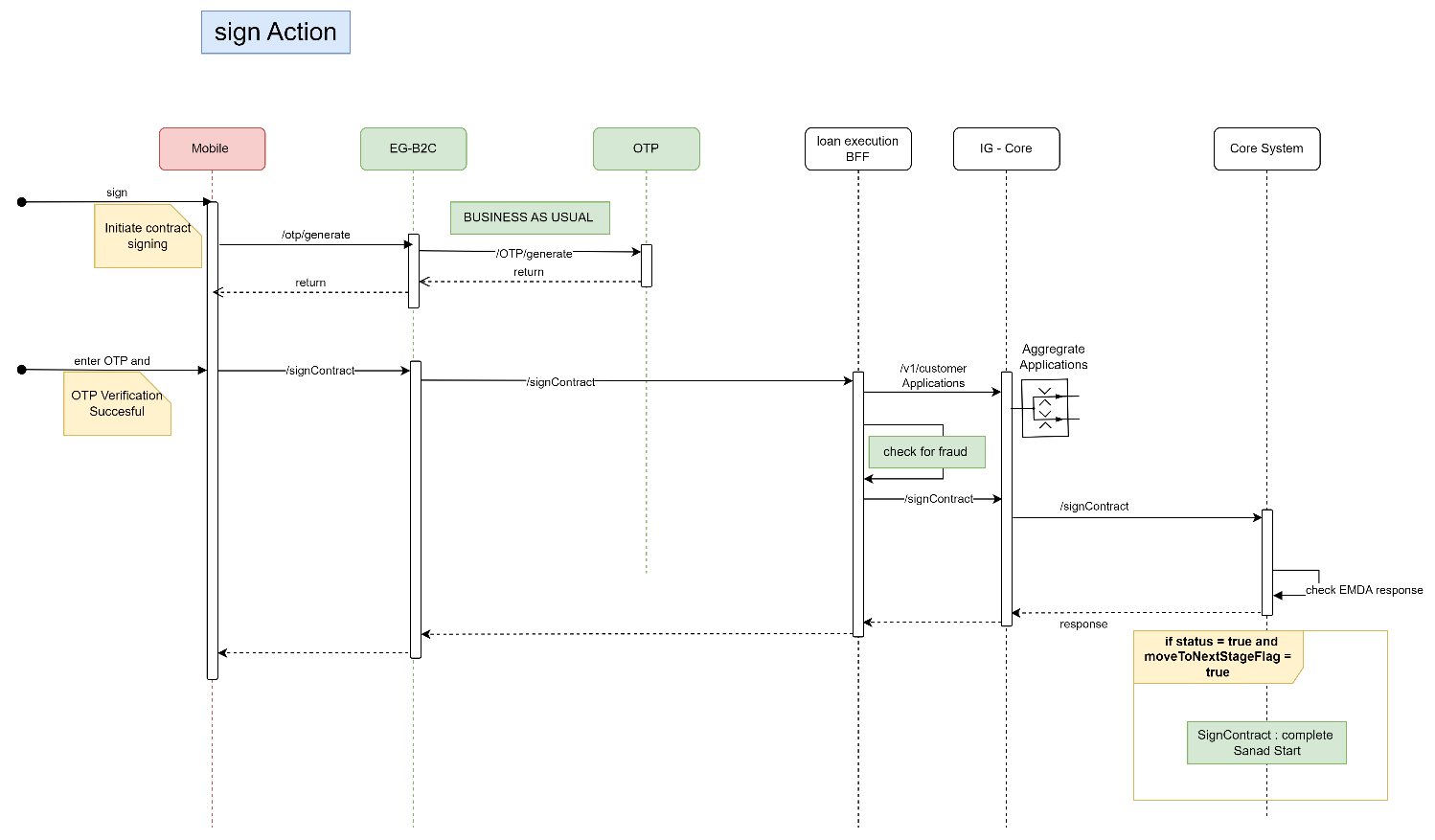
Mange My Profile



Description

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **System Provider** | **Action** | **Request** | **Response** | **Type** | **Description** |
| Loan-Execution | getCustomerReviewDetails | AsIs | AsIs | Rest/Json | 1) Make sure IbanInformation is fetched from iban\_verification table with attributes  i)isVerifiedFlag  ii) status  iii) isPrimary  iv) ibanInfo  2) If iban list is empty then allow user to enter iban and follow the update iban flow. Mark PrimaryIban with Y  3) If iban list is returned with isVerified flag as False then display iban with status as Submitted. |
| ARB | /updateIban | AsIs | AsIs | Rest/Json |  |
| ARB | /ibanVerifyCallback | AsIs | AsIs | Rest/Json |  |

Signe Flow



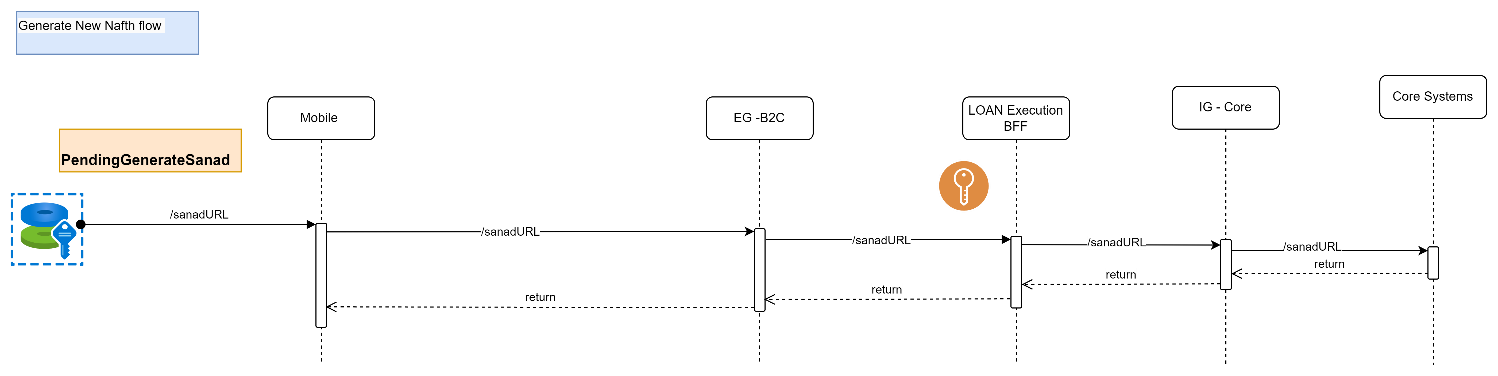
Description:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **System Provider** | **Action** | **Request** | **Response** | **Type** | **Description** |
| OTP | /otp/generate | AsIs | AsIs | Rest/Json |  |
| Loan-Execution | /signContract | nationalId  compAppId | compAppId  status  isSanad  sanadURL |  |  |

# Sanad Flow

Design Overview

* Diagram



Description:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **System Provider** | **Action** | **Request** | **Response** | **Type** | **Description** |
| Loan-Execution | /sanadURL | status  isSanad  sanadURL | status  isSanad  sanadURL |  |  |

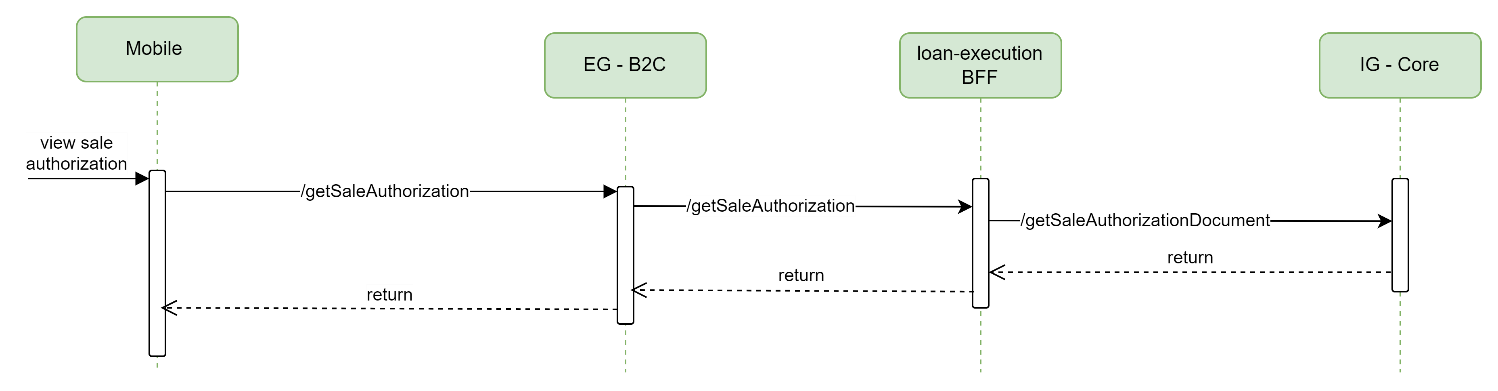
A computer screen shot of a black background

Description automatically generated

Description:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **System Provider** | **Action** | **Request** | **Response** | **Type** | **Description** |
| Loan-Execution | /digital/sanadCallBack | AsIs | AsIs |  |  |

# view sale authorization Flow



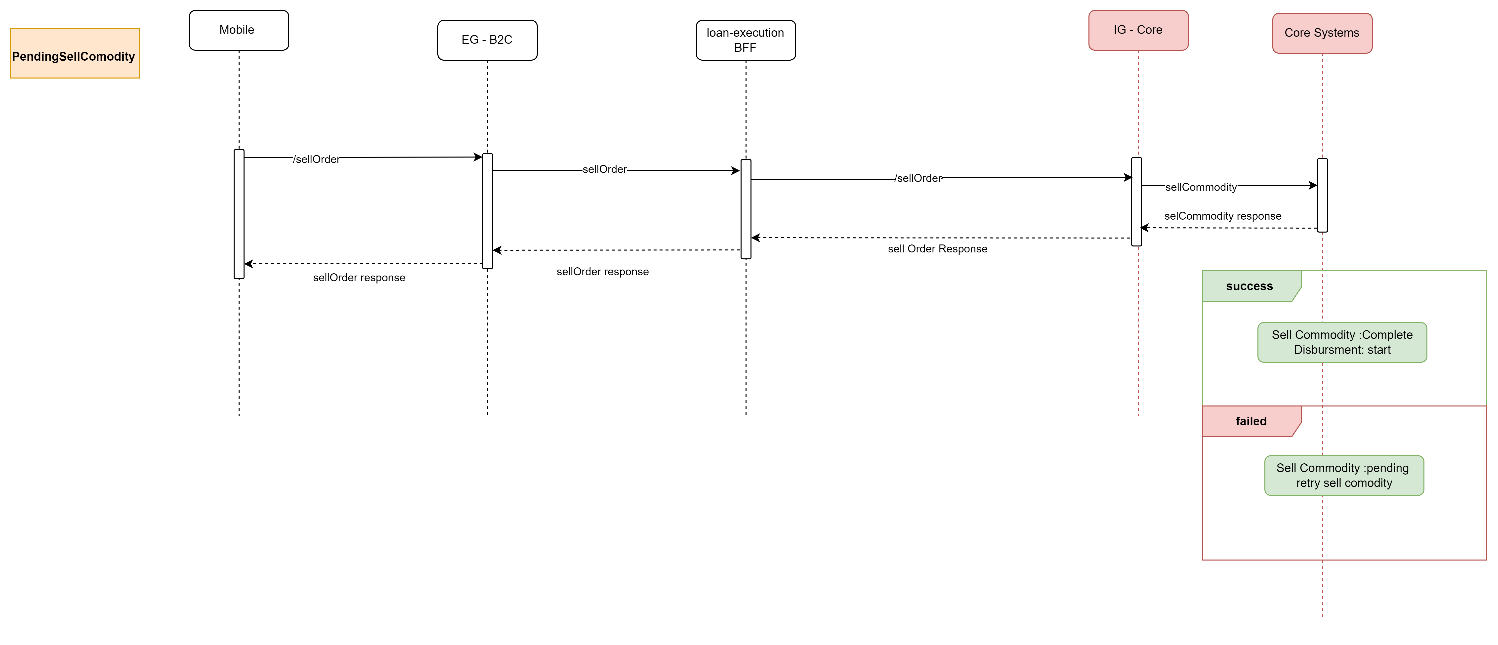
Description:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **System Provider** | **Action** | **Request** | **Response** | **Type** | **Description** |
| Loan-Execution | /getSaleAuthorization | AsIs | AsIs |  |  |

# Sell commodity Flow

Design Overview

* Diagram



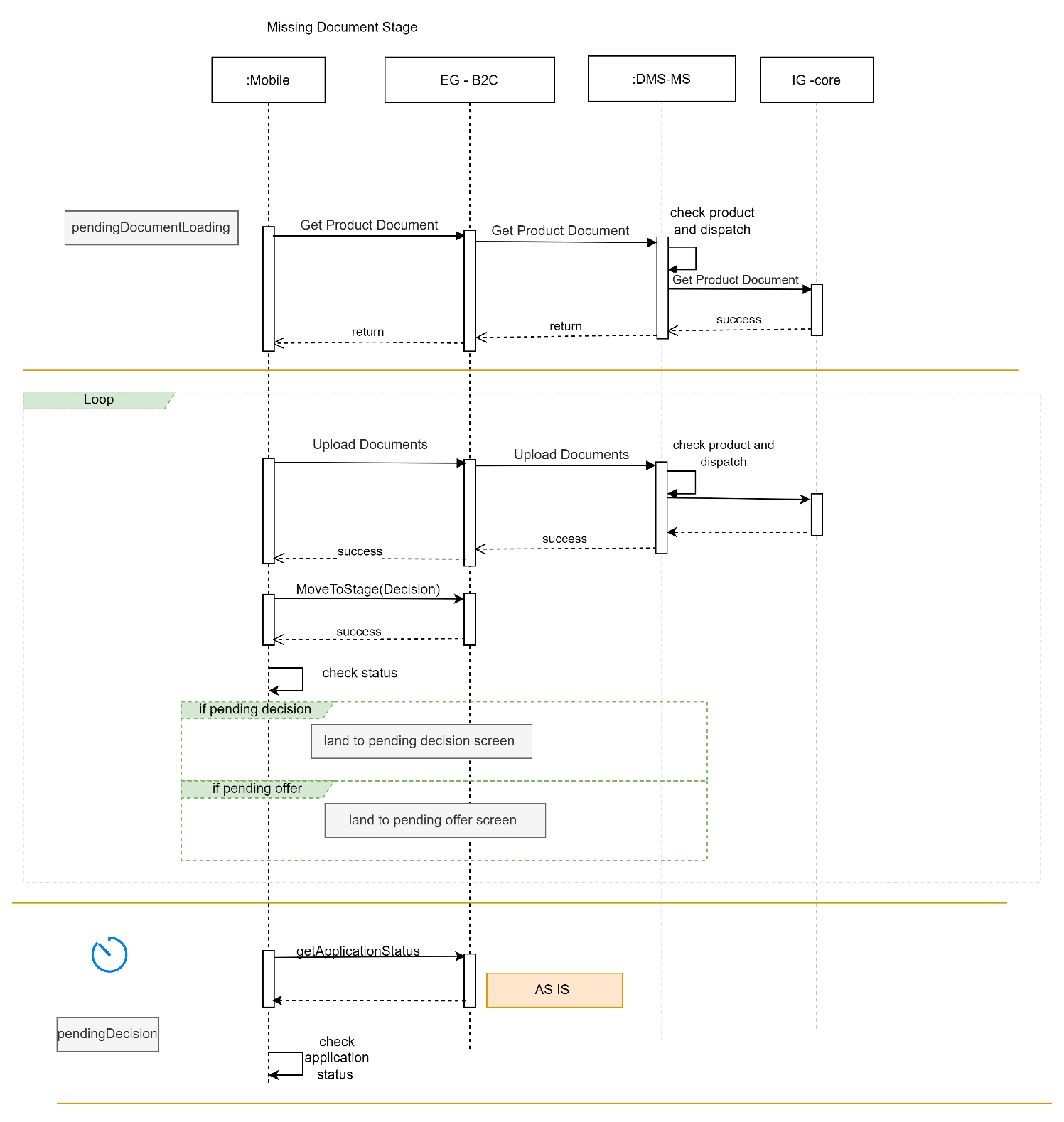
Description:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **System Provider** | **Action** | **Request** | **Response** | **Type** | **Description** |
| Loan-Execution | /sellOrder | nid  compAppId | compAppId  status |  |  |

# Disbursment Flow

LOS internal Flow

# MissDocument Flow



# **Deployment and Security View**

Description:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **System Provider** | **Action** | **Request** | **Response** | **Type** | **Description** |
| Loan-Origination | getApplicationStatus | Nationalid  compAppId  preferredLanguage | compAppId,  stageStatus,  emkanProduct,  subProduct |  | Based on status navigate to respective screen. |
| DMS | getProductDocument | AsIs | AsIs |  |  |
| DMS | uploadDocument | AsIs | AsIs |  |  |

# Appendix A - Architectural Decisions

## 

## 

# Appendix D – Failing Retry Policy

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **System** | **Error Code** | **Retry Mechanism** | **Max Number of Retry** | **Alternative after Pass Max No of Retry** |
| Emkan Services | Not http “200” | Exponential back off  Starting with interval 5 mins | 20 | Create Jira ticket |
| “Newgen”,MW | Not http “200” | Retry every 10 min | 20 | Create Jira ticket |

# Appendix E – Cooling Period

|  |  |  |
| --- | --- | --- |
| **Stage/Type** | **Cooling Period** | **System Owner** |
| IVR |  |  |
| signing contract | 24 h |  |
| generate Sanad |  |  |
| Sanad approval | 24 h |  |
| commodity sales maker | 24 h |  |
| commodity delivery |  |  |
| document custody |  |  |

# Appendix R – Risks

|  |  |  |
| --- | --- | --- |
| **Stage/Type** | **Risk** |  |
| Lead | Customer cann’t refinance DirectPF or Buyout |  |
| Offer | Qarar can’t offer Direct PF if customer apply on topup or Micro |  |
|  |  |  |

# Appendix I – General Impacts

|  |  |  |
| --- | --- | --- |
| **owner** | **stage** | **impact** |
| MW |  |  |
|  |  |  |
|  |  |  |

# Appendix T – Third Parties

|  |  |  |
| --- | --- | --- |
| Third Parties | **Require change** | **System Owner** |
| Qarar | Freeze Direct PF | qarar |
| Unleash | Freeze Direct PF | MW |
| Sanad |  |  |
|  |  |  |
|  |  |  |
|  |  |  |