

Ooredoo Oman

**IT EAD**

**High Level Design**

High Level Design for:

**PCID-SalesPoint-Phase 3 (eReload)**

|  |  |  |  |
| --- | --- | --- | --- |
| Version: | 2.0 | Version Release Date: | 07/08/2024 |
| Author(s): | Ali Shehadeh | Document Status: | **In Progress** |
| Storage Location: |  | | |
| Restriction: | C2 - COMPANY CONFIDENTIAL / INTERNAL USE ONLY | | |

**Reviewers & Approvals:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Name** | **Title** | **Division** | **Role** | **Signature** | **Date** |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

**Change History:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Date** | **Description** | **Author** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Contents

[1. Introduction 5](#_Toc174381565)

[1.1 Purpose 5](#_Toc174381566)

[1.2 Scope 5](#_Toc174381567)

[**1.2.1** In Scope 5](#_Toc174381568)

[**1.2.2** Out Scope 5](#_Toc174381569)

[1.3 Impacted Channels & Systems 5](#_Toc174381570)

[1.4 Intended Audience 5](#_Toc174381571)

[1.5 Reference documents 6](#_Toc174381572)

[1.6 Project & Architectural Dependencies 6](#_Toc174381573)

[1.7 Assumptions & Alignment 6](#_Toc174381574)

[1.8 Risks 6](#_Toc174381575)

[1.9 Outstanding Design Issues 6](#_Toc174381576)

[1.10 Architectural Decisions 7](#_Toc174381577)

[2. Business Process Impacts 8](#_Toc174381578)

[3. Solution Description 10](#_Toc174381579)

[3.1 Solution Overview: 10](#_Toc174381580)

[4. Requirements high level design 12](#_Toc174381581)

[4.1 Single eWallet Approach 12](#_Toc174381582)

[**4.1.1** **Channel Member eWallet** 12](#_Toc174381583)

[**4.1.2** **Wallet Balance Transfer** 14](#_Toc174381584)

[**4.1.3** **Wallet Transaction Receipt** 16](#_Toc174381585)

[4.2 Dealer Topup 18](#_Toc174381586)

[**4.2.1** **Customer Prepaid Recharge** 18](#_Toc174381587)

[**4.2.2** **Customer Postpaid Bill Payment** 21](#_Toc174381588)

[**4.2.3** **Customer Prepaid Bundle Subscription** 24](#_Toc174381589)

[**4.2.4** **eTopup Dashboard & Reports** 28](#_Toc174381590)

[4.3 Transaction Reversal 30](#_Toc174381591)

[**4.3.1** **Transaction Rollback** 30](#_Toc174381592)

[**4.3.2** **Wallet Balance Adjustment** 32](#_Toc174381593)

[**4.3.3** **Rollback & Adjustment Notification** 34](#_Toc174381594)

[**4.3.4** **Rollback & Adjustment Transaction Receipt** 35](#_Toc174381595)

[4.4 eVoucher 36](#_Toc174381596)

[**4.4.1** **Get Single Voucher by Sales App** 36](#_Toc174381597)

[**4.4.2** **Get Single Voucher by SalesPoint Web Portal** 39](#_Toc174381598)

[**4.4.3** **Get Multiple Voucher by Sales App** 42](#_Toc174381599)

[**4.4.4** **Get Single Voucher by USSD** 44](#_Toc174381600)

[**4.4.5** **Download Voucher** 47](#_Toc174381601)

[**4.4.6** **Void Voucher** 48](#_Toc174381602)

[**4.4.7** **Reserve Voucher (Offline Distributor)** 50](#_Toc174381603)

[4.5 Generate ERP Sale 52](#_Toc174381604)

[**4.5.1** **ERP Wallet Balance** 52](#_Toc174381605)

[**4.5.2** **Voucher Dashboard & Availability** 54](#_Toc174381606)

[**4.5.3** **Voucher PIN Generation via VMS System** 55](#_Toc174381607)

[**4.5.4** **Notification with SLA for VMS PINs Order** 58](#_Toc174381608)

[**4.5.5** **Safety Stock Notification for Each Denomination** 60](#_Toc174381609)

[**4.5.6** **OO Logistics Team Role Access in SalesPoint Admin Portal** 61](#_Toc174381610)

[**4.5.7** **Distributors Role Access in SalesPoint Web Portal** 63](#_Toc174381611)

[4.6 Integration with Enhance 65](#_Toc174381612)

[4.7 Postpaid Bill Payment eWallet 66](#_Toc174381613)

[4.8 Bulk Customer Recharge 68](#_Toc174381614)

[4.9 Bulk eWallet Topup 71](#_Toc174381615)

[4.10 PIN Code 74](#_Toc174381616)

[4.11 Customer Nickname 76](#_Toc174381617)

[4.12 eReload Reports 78](#_Toc174381618)

[4.13 Push eWallet Transaction to DWH 80](#_Toc174381619)

[4.14 Other Operators eVoucher Inventory 82](#_Toc174381620)

[4.15 Other Operators SalesPoint Web Portal Access 85](#_Toc174381621)

[4.16 POS Device Registration and Management 88](#_Toc174381622)

[4.17 Voucher Card Download and Print 90](#_Toc174381623)

[4.18 CVM Offers 91](#_Toc174381624)

[5. Service & Integration Design 95](#_Toc174381625)

[5.1 Interfaces 95](#_Toc174381626)

[6. System Impacts 97](#_Toc174381627)

[6.1 System Impacts Summary 97](#_Toc174381628)

[**6.1.1** USSD Gateway: 97](#_Toc174381629)

# Introduction

## Purpose

This document provides a high-level solution for delivering eReload system inside SalesPoint solution and also highlights the main integration services that are required for SalesPoint eReload system.

## Scope

### In Scope (Refer to approved SRS document listed in Reference documents “SRS - Ooredoo SalesPoint (DMS) Solution - Phase 3 v1.4.docx”)

* Dealer & Distributor eWallet Management
* Customer eTopup
* Selling eVoucher (SMS & Print)
* eReload Reports
* eVoucher Management & Distribution (By Channel and Shared Pool)
* CVM Offers
* New eReload solution

For more details about in scope kindly refer to the SOW document in reference documents section.

### Out Scope

* Any requirement does not exist in the approved SRS document will be considered as out of scope.

## Impacted Channels & Systems

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | | | | | | | | | |
| SalesPoint |  | USSD |  | ESB |  | DWH |  | SMSC |  |
| ERP |  |  |  |  |  |  |  |  |  |

## Intended Audience

The following document is intended for:

* IT EAD
* ADM\Third Party
* IT PM
* Business SPOCs

## Reference documents

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID | Name | Version | Author | Location and Attachment |
| 1 | SRS | V1.4 | Ali Shehadeh |  |
| 2 | SOW | V1.3 | Khalid Zukari |  |

## Project & Architectural Dependencies

|  |  |  |
| --- | --- | --- |
| ID | Details | Approved by |
|  |  |  |
|  |  |  |

## Assumptions & Alignment

|  |  |  |
| --- | --- | --- |
| ID | Details | Attachment |
|  |  |  |
|  |  |  |

## Risks

|  |  |  |
| --- | --- | --- |
| ID | Details | Approved by |
| 1 | Hard dependency on Digitalization project phase 2, as all the touch points will be through ESB, no direct integration with CRM. | Hussain Khabouri |

## Outstanding Design Issues

|  |  |  |
| --- | --- | --- |
| ID | Details | Status |
|  |  |  |

## Architectural Decisions

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Issue | Reasoning & Decision | Approved by |
| 1 | SalesPoint eReload integration will be direct from/to SalesPoint and USSD, ~~SMSC~~, SNG, ERP, VMS systems |  | ITEAD |

# Business Process Impacts

The solution is impacting the Customer-Centric Processes for Business customer segment of Ooredoo.

Customer-Centric Processes: A total of seven Customer Centric Processes represent the customer view and interaction with the telecommunication company. These processes start with the customer initiating the contact. They end with the fulfilment of his/her request. Customer Centric processes include activities such as handling information requests, new sale, billing and invoice generation or problem and complaint handling.

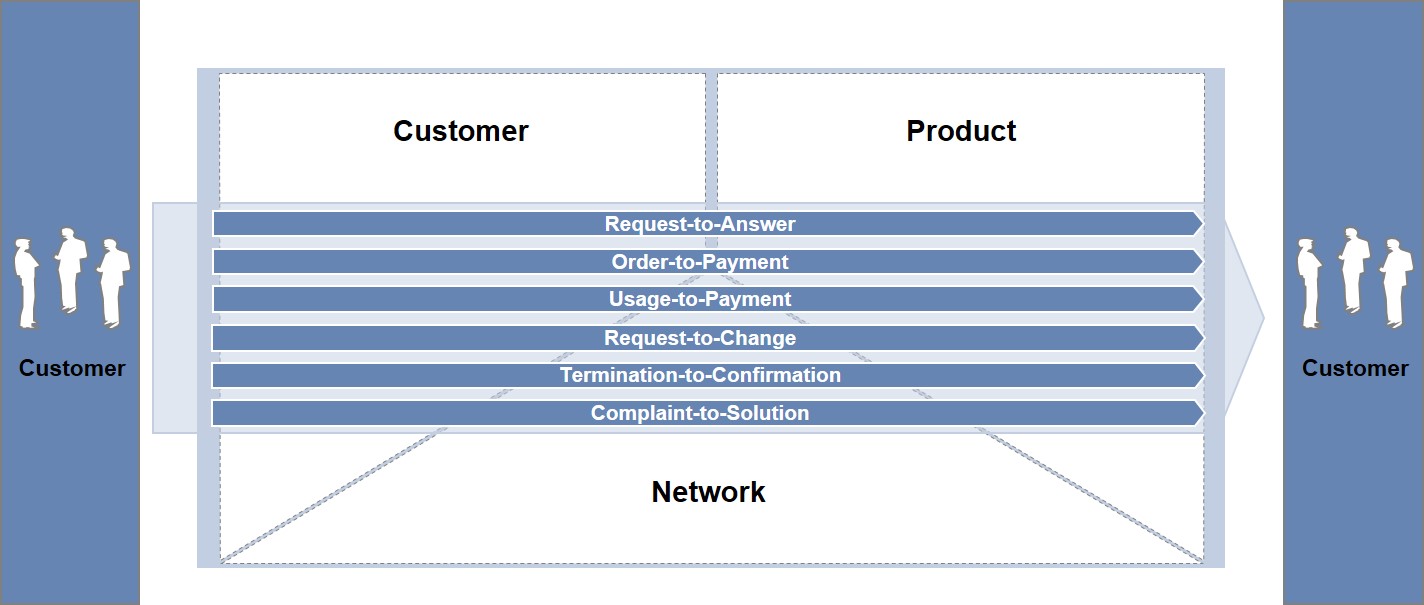


Figure 1: Customer-Centric Processes

The impact on the customer-centric process is listed below:

|  |  |  |  |
| --- | --- | --- | --- |
| # | Name | Description | Impact |
| 001 | Request-to-Answer | * This process comprises of activities relevant to managing customer requests across all communication channels (customer interfaces). * Specific information requests or product requests from the customer are qualified and addressed. * This could lead to the preparation of a pre-sales offer if the customer shows interest in a particular product. | No Impact |
| 002 | Order-to-Payment | * This process deals with all activities which convert the customer request or an accepted offer into a ‘Ready for use’ product. * This process involves capturing customer order information, triggering the relevant provisioning process and handing over the order to the Service layer. * Once the product is successfully provisioned, the customer order is closed and the customer satisfaction is validated. | No Impact |
| 003 | Usage-to-Payment | * This process deals with all activities related to the handling of the product/service usage. * The accuracy of pricing is ensured and all usage data is captured and duly processed for billing information requests and bill generation | No Impact |
| 004 | Request-to-Change | * This process deals with all activities which convert the customers change request into a ‘Ready for use’ product. * This process involves capturing customer order information, triggering the relevant provisioning process and handing over the order to the Service layer. * Once the product is successfully provisioned, the customer order is closed and the customer satisfaction is validated. | No Impact |
| 004 | Termination-to-Confirmation | * This process deals with all activities related to the execution of customer‘s termination request. * This process involves retention activities, capturing customer order information, triggering the relevant provisioning process and handing over the order to the Service layer. * Once the product is successfully terminated, the customer order is closed and the customer satisfaction is validated. | No Impact |
| 005 | Complaint-to-Solution | * This process deals with a complaint (problem) initiated by the customer, analyses it to identify the source of the issue, initiates resolution, monitors progress and closes the trouble ticket. * A complaint is a customer inquiry in which the customer is not pleased with a product or the handling speed of an inquiry etc. | No Impact |
| 006 | Problem-to-Solution | * This process deals with a technical complaint (problem) initiated by the customer, analyses it to identify the source of the issue, initiates resolution, monitors progress and closes the trouble ticket. * The basis for a problem is an unplanned interruption to a product / service or reduction in the quality of a product/service. (In comparison, the process “complaint-to-solution” deals with customer inquiries in which the customer is not pleased with a product or handling speed of an inquiry etc.). | No Impact |

# Solution Description

The new eReload Component inside SalesPoint will replace the existing OO eReload system with all the existing features that business users have it in the current solution. After the successful implementation, current eReload shall be sunset.

## Solution Overview:

SalesPoint product is designed as a sophisticated N-tier architecture built on modern technologies and open standards. Essentially, this proposed product functions as an all-in-one solution for telecom companies to oversee their complete sales operations. This includes customer activities, sales activities, managing growing sales trees of distributors and dealers, while smoothly handling the complexity of commission processing. Furthermore, it seamlessly integrates with the digital assets of telecom companies.

The diagram below illustrates the SalesPoint architecture and represents the central systems that will be seamlessly integrated into the solution.



Figure : SalesPoint Architecture

# Requirements high level design

## Single eWallet Approach

In this requirement, and as per business needs, eTopup eWallet and eVoucher eWallet will be combined in a single eWallet for dealer and distributor that allow the users to perform the below activities from same balance amount:

* Balance eTopup (Pinless recharge)
* Bundle Subscription
* Receive commission
* Selling eVoucher by SMS or Print from POS device.

### **Channel Member eWallet**

**1. Wallet Management**

* **Wallet Balance Addition:**
  + **Earned Commission:** Automatically adds earned commission to the One Wallet after a transaction or periodically as defined by the system rules. All the calculation for commission should be done on the backend not on the front end.
  + **Top-Up Using Voucher HRN:** Dealers can redeem specific HRN codes to add balance directly to their One Wallet. Invisible captcha shall be implemented for voucher redemption.
  + **Distributor Transfer:** Allows a distributor to transfer funds from their master wallet to a dealer's One Wallet, facilitating intra-network fund distribution from Sales App and Web Portal.
  + **ERP Integration:** The system will expose a secured API endpoint for ERP systems to trigger top-ups of the distributor's master wallet. This allows seamless fund management from corporate finance systems.
    - **For the APIs exposed to ERP, below shall be considered:**
      * Create a dedicated account on ERP with access limited to specific APIs
      * API should be exposed on secure encrypted channel as HTTPS
      * API should enforce authentication and authorization checks
      * API should main audit logs for all trans action performed on ERP & Salespoint end
* **Immediate Balance Update:**
  + Upon successful execution of any balance addition method, the One Wallet balance is updated in real-time. This ensures that users always have the most current information about their available funds.
* **Confirmation and Notifications:**
  + A confirmation message (SMS, push notification, and in-app alert) is sent to the user after any successful top-up action, providing transaction details and updated balance information. The notification will mention the transaction type like Commission, Topup, Bill payment …etc.
* **Real-Time Balance Display:**
  + The current balance of the One Wallet will be prominently displayed on the user’s dashboard in the SalesPoint system and mobile app, ensuring visibility at all times.
* **Transaction History:**
  + Users can access a detailed transaction history related to their One Wallet. This includes top-up events, transfers, payments, and commission credits, along with timestamps and descriptions for each transaction.

**2. Security Considerations**

* **Secure API Access:** The ERP integration for distributor master wallet top-ups will require secure authentication (using bearer token).
* **Transaction Logging:** All balance addition and transfer actions will be logged for audit purposes, ensuring traceability and accountability.
* **Error Handling:** The system will include robust error handling mechanisms to manage scenarios like failed transactions, insufficient funds, or incorrect HRN codes, with appropriate user notifications and system rollbacks.

**3. User Interface**

* **Dashboard Integration:** The One Wallet balance and quick actions (such as top-up or transfer) will be accessible directly from the user’s main dashboard in both the SalesPoint backend portal and the mobile app.
* **Transaction History View:** A dedicated section within the user profile or wallet menu will allow users to view their transaction history, filter by date, and search specific transactions.

**4. Integration Points**

* **Voucher HRN System:** Integration with the existing voucher redemption system for handling HRN-based top-ups.
* **Commission Engine:** Integration with the commission calculation and distribution engine to credit earnings directly to the One Wallet.
* **ERP System:** A dedicated API for ERP systems to trigger and manage top-ups for distributor master wallets.

### **Wallet Balance Transfer**

The Wallet Balance Transfer feature allows a distributor to transfer funds from their One Wallet (master wallet) to the One Wallets of other users (dealers) within their hierarchy in the eReload system. This functionality is part of the SalesPoint system and will be implemented for Ooredoo Oman.

**1. Transfer Interface**

* **User Interface in SalesPoint Portal:**
  + **Transfer Options:** Distributors will have access to a dedicated interface within the SalesPoint portal where they can initiate balance transfers. The interface will support:
    - **Batch Transfers:** Distributors can transfer balance to multiple users at once, streamlining bulk distribution of funds.
    - **Individual Transfers:** Distributors can also perform single transfers to individual dealers under their hierarchy.
* **Channel Member Profile Enhancement:**
  + A new field will be added to the channel member profile page to segregate between Ooredoo (OO) dealers and distributor dealers. This will help distributors easily identify eligible recipients within their hierarchy and ensure that transfers are made only to authorized users.

**2. Transfer Validation**

* **Recipient Account Validation:**
  + Before initiating the transfer, the system will validate the recipient's account to ensure it is active, under the correct hierarchy, and eligible to receive funds. This validation process will check:
    - **Account Status:** Ensures the recipient’s account is active and capable of receiving transfers.
    - **Hierarchy Verification:** Confirms that the recipient is under the distributor’s hierarchy, preventing unauthorized transfers.
    - **Eligibility Check:** Validates that the recipient is designated as a dealer in the system and not classified under a restricted category.
    - **Channel PIN:** Validates that the channel PIN is valid and matched the user PIN stored encrypted in the database.

**3. Balance Update and Notifications**

* **Real-Time Balance Update:**
  + Upon a successful transfer, the system will immediately update the One Wallet balances for both the sender (distributor) and the recipient (dealer). The master wallet balance of the distributor will be debited, and the recipient's wallet will be credited accordingly.
* **Transaction Logging:**
  + The system will log the details of the transfer, including the amount, timestamp, sender, source IP, hostname, channel type (web/mobile), Device ID, and recipient details, to ensure traceability and support audit requirements.
* **Notifications:**
  + **Sender Notification (SMS & Push Notification):** The distributor will receive a confirmation notification once the transfer is successfully completed. This notification will include details of the transaction such as the amount transferred, the recipient, and the updated wallet balance.
  + **Recipient Notification (SMS & Push Notification):** The dealer receiving the funds will also receive a notification informing them of the balance transfer, along with the updated balance in their One Wallet.

**4. Security and Error Handling**

* **Secure Transaction Processing:**
  + All transactions will be secured using HTTPS and authentication methods to protect against unauthorized access and fraud.
  + System will reject negative or zero values in transfer balance request.
* **Error Handling:**
  + In case of any issues during the transfer (e.g., invalid recipient account, insufficient balance), the system will provide appropriate error messages to the distributor and abort the transaction.
* **Security Validations:**
  + The Dealer wallet balance should not drop below 0, post transfer of money
  + System should have threshold on the amount which can be transferred from Distributor -> Sub Distributor -> Dealer per transaction, and cumulative limit per day.
  + System should have Audit logs for all transactions having details as Date, Time, Device ID, Source IP, Party A (sender), Party B (recipient), Amount, comments, mode (commission, voucher etc), Debit/Credit

**5. User Experience**

* **Intuitive Interface:** The transfer interface will be designed to be user-friendly, allowing distributors to easily select recipients and specify transfer amounts. Batch transfer options will include bulk upload features (Excel/CSV files) for ease of use. System will validate the file type and the enforce input validation of each cell content before processing the file
* **Transaction History:** Distributors will have access to a history of all balance transfers initiated from their master wallet, allowing them to track and manage their fund distributions effectively though Sales App and web portal.

### **Wallet Transaction Receipt**

**1. Receipt Generation**

* **Receipt Creation:**
  + For every transaction performed through the One Wallet (e.g., balance transfers, top-ups, commission credits, voucher purchases), the system will automatically generate a receipt, The receipt will be generated in real-time once requested by the authorized user from eWallet transaction history.
* **Receipt Details:**
  + The receipt will contain comprehensive details about the transaction, including:
    - **Amount:** The exact amount credited to or debited from the One Wallet.
    - **Date and Time:** The timestamp when the transaction was processed.
    - **Transaction Type:** Whether the transaction was a credit or a debit.
    - **Transaction Details:** A brief description of the transaction (e.g., "Top-up using Voucher HRN," "Balance Transfer to Dealer X").
    - **Created By:** The identifier (username or system process) of the entity that initiated or processed the transaction.

**2. User Access and Interaction**

* **Receipt Viewing:**
  + Users will be able to view receipts for all their wallet transactions directly within their SalesPoint account. This will be accessible through a dedicated section in the transaction history or account management area.
  + Receipts will be displayed in a list format, with each entry providing a summary of the transaction, and users can click on any entry to view the full receipt details.
* **Receipt Downloading:**
  + The system will provide an option for users to download their receipts in PDF format. This allows users to keep a personal record of their transactions outside of the SalesPoint system.
  + System should ensure to sanitize the data required for crafting the PDF receipt to prevent template Injection or XSS attacks.
  + Receipts will be formatted for easy reading and printing, including the company logo, and consistent styling for professional presentation.

**3. Security**

* **Secure Access:**
  + Access to receipt viewing and downloading will be restricted to authorized users based on their role and permissions within the system.
* **Security Validations:**
  + System shall enforce restriction such that the details of transaction, including receipts are accessible only with the parties involved.

**4. User Experience**

* **User-Friendly Interface:**
  + The receipt viewing and downloading interface will be designed to be intuitive, with clear navigation and options for sorting and filtering past transactions. Receipts will be easily accessible from both the SalesPoint web portal and the mobile app.
* **Receipt Format:**
  + Receipts will be formatted to be easily readable, including key transaction details in a structured layout. The format will also ensure that receipts are printer-friendly for users who may need physical copies.

## Dealer Topup

### **Customer Prepaid Recharge**

**1. User Interface in Sales App**

* **Recharge Interface:**
  + The Sales App will provide a dedicated interface for dealers to initiate prepaid recharges for customers.
  + **Recharge Options:** Dealers can recharge customer accounts using:
    - **FDN (Fixed Dial Number):** A predefined number associated with the customer's account.
    - **MSISDN:** The customer's mobile number.
    - **Nickname:** A user-defined name associated with the customer’s account for easy identification.
* **Recharge Amount Selection:**
  + The interface will allow the dealer to select the desired recharge amount, dealer can enter a custom amount as per the customer’s requirement. Amount will be in Rial Omani with only one fraction digit like 1.2 OMR, 3.4 OMR …etc.
  + System should not accept negative or Zero amount at any place, and it must be validated on the Backend.

**2. Validation and Processing**

* **Dealer PIN Validation:**
  + System should enforce Dealer’s PIN for recharge request.
* **Wallet Balance Validation:**
  + Before processing the recharge, the system will validate that the dealer’s One Wallet has sufficient balance to cover the requested recharge amount.
  + If the balance is insufficient, the system will display an error message and prevent the transaction from proceeding.
* **Recharge Confirmation:**
  + Upon confirmation of the recharge request, the system will deduct the specified amount from the dealer’s One Wallet balance.
* **Recharge Execution:**
  + The system will interface with the IN (Intelligent Network) system via the ESB (Enterprise Service Bus) to perform the actual top-up of the customer’s prepaid account.
  + **ESB Integration:**
    - **Recharge Request:** The SalesPoint backend will send a request to the IN system through the ESB to apply the recharge to the customer’s account.
    - **Response Handling:** The system will handle responses from the ESB, ensuring that the recharge is successful and the customer's account is updated.

**3. Post-Processing and Notifications**

* **Confirmation Message:**
  + After the successful recharge, the system will send a confirmation message to the dealer within the Sales App. This message will include details of the recharge, such as the amount, the customer's MSISDN, and the transaction ID.
  + The system may also send a notification via SMS and push notification to the dealer confirming the recharge.
  + ESB will handle customer notification for topup transaction.
* **Transaction History:**
  + The recharge transaction will be automatically recorded in the dealer's transaction history. Dealers can view this history in the Sales App, which will show details of the recharge, including the date, time, amount, and customer MSISDN.

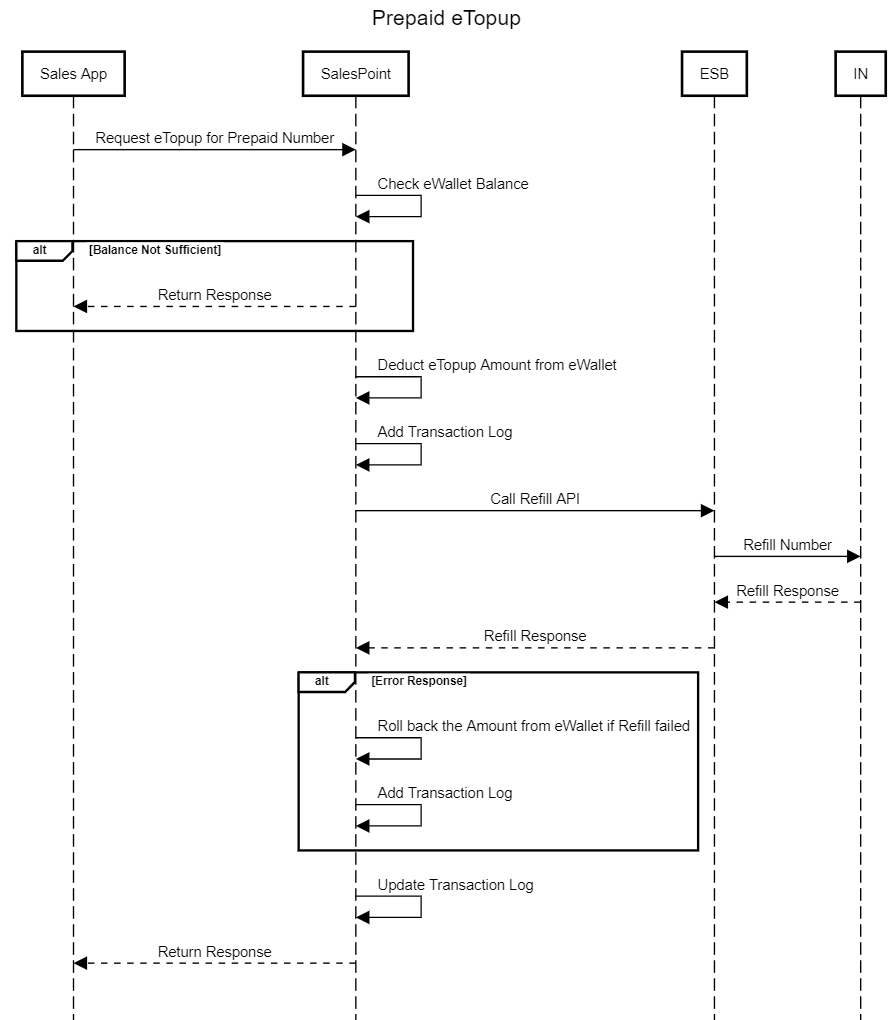
**4. Error Handling**

* **Insufficient Balance:**
  + If the dealer’s One Wallet does not have enough balance to cover the recharge, the system will immediately notify the dealer and prevent the transaction.
* **ESB/IN System Failure:**
  + If the recharge fails due to issues with the ESB or IN system (e.g., network errors, system downtime), the system will roll back the transaction. This includes reversing the deducted amount in the dealer's One Wallet and notifying the dealer of the failure.

**5. Security Considerations**

* **Security Validations:**
  + The system shall enforce strict input validation in all the parameters and input fields.
* **Data Protection:**
  + All transactions, including recharge requests and balance deductions, will be secured using HTTPS and authenticated access to prevent unauthorized use.
* **Audit Trail:**
  + The system will maintain an audit trail of all recharge transactions, including details of who initiated the recharge, when it occurred, and the outcome. This ensures accountability and traceability.

**6. Sequence Diagram**



### **Customer Postpaid Bill Payment**

**1. User Interface in Sales App**

* **Bill Payment Interface:**
  + The Sales App will provide a dedicated interface for dealers to initiate postpaid bill payments for customers.
  + **Payment Options:** Dealers can make payments using multiple identifiers, including:
    - **Account Number:** The unique identifier associated with the customer's postpaid account.
    - **FDN (Fixed Dial Number):** A predefined number associated with the customer’s account.
    - **MSISDN:** The customer's mobile number.
    - **Service ID:** A unique service identifier for B2B Fixed postpaid account.
    - **Nickname:** A user-defined name associated with the customer’s account.
* **Bill Amount Selection:**
  + The interface will allow the dealer to select the bill amount. This could be the full bill amount or a partial payment as per the customer's requirement. Amount will be in Rial Omani with only one fraction digit like 1.2 OMR, 3.4 OMR …etc.
  + System should not accept negative of Zero amount at any place, and it must be validated on the Backend.

**2. Validation and Processing**

* **eWallet Account Mapping:**
  + The system will determine under which eWallet account (e.g., Bill Payment Wallet) the bill payment activity is mapped. This ensures that the correct wallet is used for the transaction.
* **Wallet Balance Validation:**
  + Before processing the payment, the system will validate that the dealer’s Bill Payment Wallet has sufficient balance to cover the requested bill amount. If the balance is insufficient, the system will display an error message and prevent the transaction from proceeding.
* **Bill Payment Confirmation:**
  + Upon confirmation of the bill payment request, the system will deduct the specified amount from the dealer’s Bill Payment Wallet balance.
* **Bill Payment Execution:**
  + The system will interface with the Billing system through the ESB (Enterprise Service Bus) to perform the actual bill payment.
  + **ESB Integration:**
    - **Payment Request:** The SalesPoint backend will send a payment request to the Billing system via the ESB to apply the payment to the customer’s postpaid account.
    - **Response Handling:** The system will handle responses from the ESB, ensuring that the payment is successful, and the customer's account is updated.

**3. Post-Processing and Notifications**

* **Confirmation Message:**
  + After the successful bill payment, the system will send a confirmation message to the dealer within the Sales App. This message will include details of the payment, such as the amount, the customer's MSISDN, and the transaction ID.
  + The system will also send a notification via SMS and push notification to the dealer confirming the payment.
  + ESB will take care of sending the bill payment message to the customer.
* **Transaction History:**
  + The bill payment transaction will be automatically recorded in the dealer's transaction history. Dealers can view this history in the Sales App, which will show details of the payment, including the date, time, amount, and customer account number or MSISDN/FDN or Service ID.

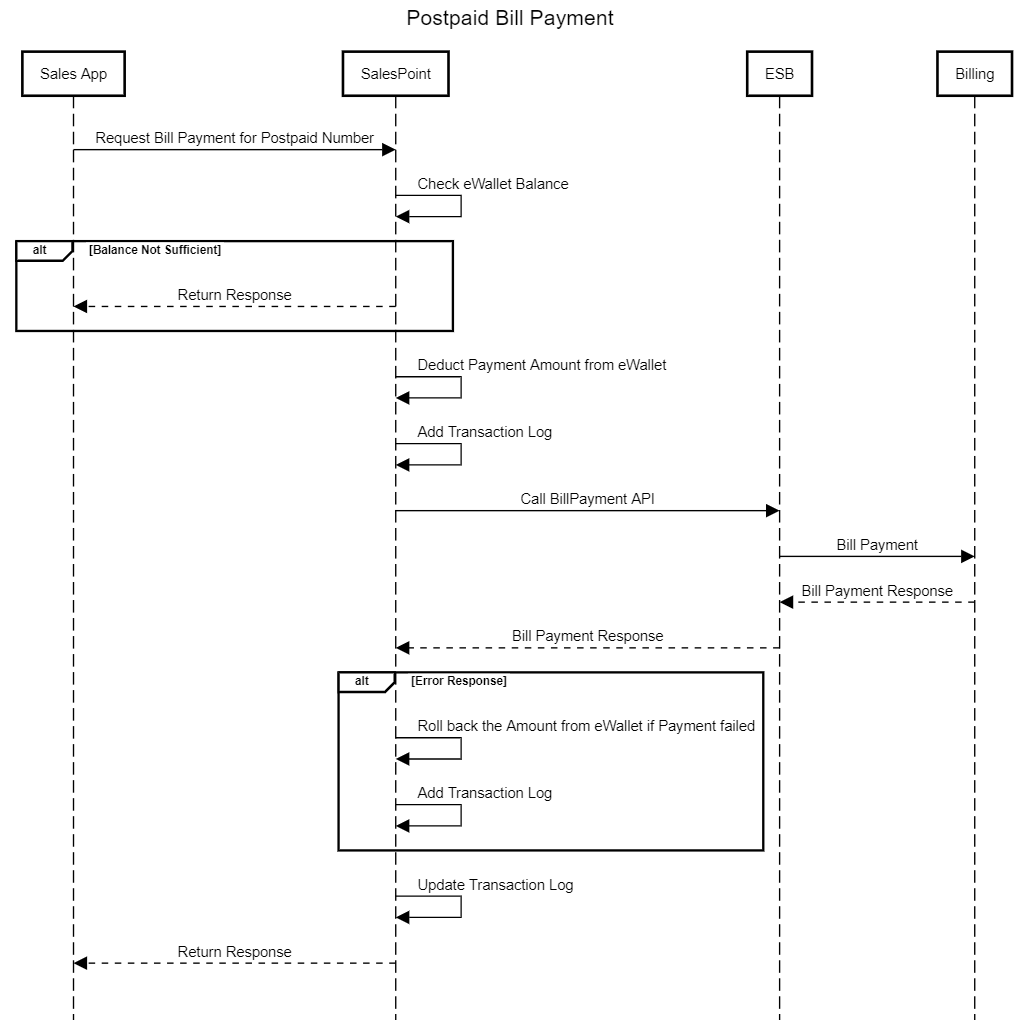
**4. Error Handling**

* **Insufficient Balance:**
  + If the dealer’s Bill Payment Wallet does not have enough balance to cover the payment, the system will immediately notify the dealer and prevent the transaction.
* **ESB Failure:**
  + If the payment fails due to issues with the ESB or Payment Gateway (e.g., network errors, system downtime), the system will roll back the transaction. This includes reversing the deducted amount in the dealer's Bill Payment Wallet and notifying the dealer of the failure.

**5. Security Considerations**

* **Data Protection:**
  + All transactions, including payment requests and balance deductions, will be secured using HTTPS and authenticated access to prevent unauthorized use.
* **Audit Trail:**
  + The system will maintain an audit trail of all bill payment transactions, including details of who initiated the payment, when it occurred, and the outcome. This ensures accountability and traceability.

**6. Sequence Diagram**



### **Customer Prepaid Bundle Subscription**

This feature enables the integration of the SalesPoint eReload system with Ooredoo Oman’s backend system to facilitate customer bundle subscription transactions. The process involves balance checks, amount deductions from the eWallet, and interaction with Ooredoo Oman’s ESB APIs for bundle offerings and subscription management.

**Key Components**

1. **SalesPoint eReload System:**
   * Manages customer eWallet balances, validates transactions, and handles the deduction process.
   * Interfaces with Ooredoo Oman’s ESB APIs to retrieve product offerings and submit subscription requests.
   * Manages the customer’s eWallet balance, ensuring funds are available before committing to the subscription.
   * Handles the deduction of funds based on the selected bundle or plan
2. **ESB APIs:**
   * **RetrieveProductOfferings API:**
     + Provides available bundles and plans based on the customer's MSISDN (Mobile or Fixed).
   * **CreateCart API:**
     + Manages the subscription of addons or changes to the base plan for the customer.
3. **SalesPoint Mobile App:**
   * User interface where customers can view available bundles, select a plan, and initiate the subscription process.

**Functional Flow**

1. **Retrieve Bundle Offerings:**
   * **Dealer nput:**
     + The dealer accesses the "Bundle Subscription" option in the SalesPoint Mobile App.
   * **API Call:**
     + The SalesPoint eReload system sends a request to the RetrieveProductOfferings API via the ESB, passing the customer’s MSISDN.
   * **Regex Check:**
     + The system determines whether the MSISDN is Mobile or Fixed based on a regular expression (Regex) check.
   * **Offerings Display:**
     + The API returns a list of available bundles and plans. The SalesPoint Mobile App displays these offerings, including details like price (inclusive of VAT) and validity.
2. **Customer Selection and Subscription:**
   * **Bundle Selection:**
     + The customer selects a bundle or plan from the list displayed in the SalesPoint Mobile App.
   * **Dealer PIN Validation:**
     + System should enforce Dealer’s PIN for recharge request.
   * **Wallet Validation:**
     + Before proceeding, the system checks the associated eWallet type to ensure sufficient balance is available to cover the cost of the selected bundle.
   * **Amount Deduction:**
     + The system deducts the bundle price (inclusive of VAT) from the customer’s eWallet.
3. **CreateCart API Call:**
   * **Subscription Request:**
     + The SalesPoint eReload system sends a request to the CreateCart API via the ESB to subscribe to the selected bundle or change the base plan.
     + The system specifies the appropriate activity (e.g., Subscribing to addons) based on the customer’s selection.
     + This API should validate if the bundle being subscribed was available in response of Bundle Subscription API. This is to prevent the attacker from directly calling CreateCard API with random bundle ID leading to unauthorized transaction.
   * **Response Handling:**
     + The ESB processes the request and returns a successful or failure response.
4. **Transaction Handling:**
   * **Successful Subscription:**
     + If the CreateCart API call is successful, the system confirms the subscription and finalizes the transaction.
   * **Failure and Rollback:**
     + If the CreateCart API call fails, the system rolls back the transaction by refunding the deducted amount to the customer’s eWallet.
   * **Error Messaging:**
     + The SalesPoint Mobile App displays a confirmation message upon successful subscription or an error message if the process fails.
5. **User Notifications:**
   * **Transaction Notifications:**
     + The user receives an SMS and a push notification confirming the wallet transaction and subscription status.

**Sequence Diagram**

A diagram of a company

Description automatically generated with medium confidence

### **eTopup Dashboard & Reports**

The eTopup Dashboard & Report feature in the SalesPoint Admin Portal provides administrators with a comprehensive view of key metrics and transaction summaries related to eTopup activities. This feature includes reporting capabilities with various filters and export options, ensuring that only authorized users can access and manage this data. This dashboard accessible within OO network only.

**1. Administrative Dashboard**

* **Dashboard Overview:**
  + The eTopup Dashboard will present an overview of key performance indicators (KPIs) and metrics related to eTopup activities. This includes:
    - **Total Topups:** The total number of eTopup transactions over a selected period.
    - **Total Amount:** The cumulative amount of money recharged through eTopup transactions.
    - **Active Users:** The number of dealers actively performing eTopup transactions.
* **Real-Time Updates:**
  + The dashboard will display real-time data for all eTopup transactions. The system will continuously update the dashboard as new transactions are processed, ensuring that administrators have the most current information available.
* **Visualization Tools:**
  + The dashboard will include various visual aids such as graphs, charts, and tables to help administrators quickly interpret the data. Examples include:
    - **Bar Charts:** Displaying the number of transactions by day.
    - **Pie Charts:** Showing the distribution of topup amounts by transaction type.
    - **Line Graphs:** Depicting trends in topup activity over time.

**2. Reporting Capabilities**

* **Report Generation:**
  + Administrators will have the ability to generate detailed reports on eTopup transactions. These reports can be customized using various filters, including:
    - **Date Range:** Select specific dates or periods (e.g., last week, last month, custom date range) to narrow down the report.
    - **Transaction Type:** Filter by the type of eTopup transactions (e.g., prepaid recharge, postpaid bill payment).
    - **User/Dealer:** Generate reports for specific users or groups of dealers within the system.
* **Export Options:**
  + The system will allow administrators to export the generated reports in multiple formats, including:
    - **CSV:** For easy manipulation and analysis in spreadsheet software.
    - **XLSX:** For more complex reporting and data presentation needs.
  + The export function will be accessible directly from the dashboard or from the report generation interface.

**3. Access Control**

* **User Authorization:**
  + The system will ensure that only authorized users have access to the eTopup dashboard and reporting features. This will be managed through user roles and permissions within the SalesPoint Admin Portal.

## Transaction Reversal

### **Transaction Rollback**

The Transaction Rollback feature in the SalesPoint Admin Portal provides administrators with the ability to reverse transactions that have been processed through the SalesPoint system. This feature ensures that erroneous or failed transactions can be corrected, and the system's financial integrity is maintained.

**1. Rollback Interface**

* **Administrative Access:**
  + The rollback feature will be accessible only to authorized administrators through the SalesPoint Admin Portal. This access will be controlled via role-based access control (RBAC), ensuring that only users with the necessary permissions can initiate a rollback.
* **User-Friendly Interface:**
  + The interface will be designed to be intuitive and straightforward, allowing administrators to easily search for and select transactions that need to be rolled back. Key functionalities include:
    - **Search and Filter Options:** Administrators can search for transactions by transaction ID, date, amount, user, or other relevant criteria.
    - **Transaction Details:** Upon selecting a transaction, the system will display all relevant details, including the transaction type, amount, date, time, involved users, and current status.

**2. Transaction Validation**

* **Eligibility Check:**
  + Before initiating a rollback, the system will perform an automatic validation to ensure the transaction is eligible for rollback. Validation criteria include:
    - **Transaction Status:** Only transactions with specific statuses (completed, pending) may be eligible for rollback.
    - **Dependency Check:** The system will check if the transaction has any dependent or subsequent transactions that could be affected by the rollback.
* **Admin Notification:**
  + If a transaction is found to be ineligible for rollback, the system will notify the administrator with the reasons why the rollback cannot proceed. This ensures that administrators are fully informed before taking further actions.

**3. Rollback Process**

* **Reason for Rollback:**
  + The system will prompt the administrator to select a reason for the rollback from a predefined list of options (e.g., incorrect amount, duplicate transaction, system error). Additionally, the admin can enter specific details or notes related to the rollback.
* **Transaction Reversal:**
  + Once the rollback is initiated, the system will reverse the original transaction. This includes:
    - **Updating Wallet Balances:** The system will adjust the respective wallet balances for both the sender and recipient (if applicable) to reflect the reversal.
    - **Inventory Adjustment:** If the transaction involved the allocation or reservation of items (e.g., eVouchers), the system will return these items to the inventory.
* **Notification:**
  + Both the original user(s) involved in the transaction and the administrator will receive notifications confirming the rollback (SMS & Push Notification). These notifications will include details of the rollback, including the transaction ID, amount, and reason for the rollback.

**4. Audit and Logging**

* **Audit Trail:**
  + The system will maintain a detailed audit log of all rollback transactions. This log will include:
    - **Rollback Details:** Transaction ID, date, time, original transaction details, and the reason for the rollback.
    - **Administrator Actions:** The identity of the administrator who initiated the rollback and any notes or comments provided.
* **Access to Audit Logs:**
  + The audit logs will be accessible to authorized users within the SalesPoint Admin Portal. These logs can be used for internal audits, compliance checks, and resolving disputes.

**5. Security and Compliance**

* **Role-Based Access Control:**
  + Access to the wallet transaction rollback feature will be restricted based on the administrator's role. Different levels of access can be configured, allowing some administrators to perform rollback only within certain limits, while others may have broader permissions.
* **Notifications:**
  + Upon completion of a rollback transaction, both the affected user and the administrator will receive notifications detailing the adjustment (SMS & Push Notification). This helps in maintaining transparency and provides immediate feedback on the action taken.
* **Enforce Google Captcha**

### **Wallet Balance Adjustment**

The Wallet Balance Adjustment feature allows administrators to manually adjust the balance of any Wallet within the SalesPoint system. This feature is essential for correcting discrepancies, applying special credits or debits, and managing wallet balances as needed. The adjustment process is tightly controlled and logged to ensure accountability and auditability.

**1. Balance Adjustment Interface**

* **Administrative Access:**
  + The wallet balance adjustment feature will be accessible only to authorized administrators via the SalesPoint Admin Portal. Access to this feature will be controlled through role-based access control (RBAC), ensuring that only users with the appropriate permissions can perform balance adjustments.
* **User-Friendly Interface:**
  + The interface for adjusting wallet balances will be designed to be straightforward and easy to use. Key functionalities include:
    - **Search and Selection:** Administrators can search for and select the wallet to be adjusted using criteria such as wallet ID, associated user ID, or dealer name.
    - **Adjustment Fields:** The interface will provide fields for the administrator to enter the adjustment amount only in positive and radio button to select Debit or Credit and the reason for the adjustment.**2. Balance Adjustment Process**
* **Reason for Adjustment:**
  + Before processing the adjustment, the system will prompt the administrator to provide a reason for the balance adjustment. This could be selected from a predefined list (e.g., correction of error, promotional credit, system reconciliation) or entered as a custom reason. This reason will be stored along with the transaction details for future reference.
* **Validation:**
  + The system will validate the entered adjustment amount to ensure it complies with predefined limits and rules set for the administrator's role. For instance:
    - **Positive Adjustments:** The system have a maximum credit limit that an administrator can apply. This will be done for Credit transaction type only
    - **Negative Adjustments:** The system will ensure positive amount to be accepted. This will be done for Debit transaction type only.
* **Immediate Update:**
  + Once the adjustment is confirmed, the system will immediately update the balance of the selected wallet. Both the adjusted amount and the updated balance will be displayed to the administrator for confirmation.

**3. Audit and Logging**

* **Audit Trail:**
  + Every wallet balance adjustment will be recorded in the system's wallet transaction history. This log entry will include:
    - **Transaction Details:** The amount of the adjustment, the type (credit or debit), the reason for the adjustment, and the date and time of the transaction.
    - **Administrator Actions:** The identity of the administrator who performed the adjustment, along with any comments or additional details provided during the adjustment process.
    - **Wallet Information:** The wallet ID, associated user ID, and the balance before and after the adjustment.
* **Access to Audit Logs:**
  + The audit logs related to balance adjustments will be accessible only to authorized users within the SalesPoint Admin Portal. These logs can be used for internal reviews, audits, and resolving disputes.
  + Any application user should not be able to modify or delete any of the audit logs.

**4. Security and Compliance**

* **Role-Based Access Control:**
  + Access to the wallet balance adjustment feature will be restricted based on the administrator's role. Different levels of access can be configured, allowing some administrators to perform adjustments only within certain limits, while others may have broader permissions.
* **Notifications:**
  + Upon completion of a balance adjustment, both the affected user and the administrator will receive notifications detailing the adjustment (SMS & Push Notification). This helps in maintaining transparency and provides immediate feedback on the action taken.
* **Enforce Google Captcha**

### **Rollback & Adjustment Notification**

**1. Notification Triggers**

* **Transaction Rollbacks:**
  + Whenever a transaction is rolled back by an administrator, the system will trigger notifications to inform the affected user(s).
* **Wallet Balance Adjustments:**
  + Any manual adjustment to a user’s wallet balance, whether credit or debit, will trigger notifications to ensure transparency and keep users informed.

**2. Notification Channels**

* **Push Notifications:**
  + **Trigger:** Upon a rollback or balance adjustment, the system will immediately send a push notification to the user's registered mobile device.
  + **Content:** The notification will include a brief message about the rollback or adjustment, specifying the amount, the date of the transaction, and the reason for the change.
* **In-App Notifications:**
  + **Trigger:** Users will receive an in-app notification when they next log in to the SalesPoint app.
  + **Content:** The in-app notification will provide similar details as the push notification.
* **SMS Notifications:**
  + **Trigger:** An SMS notification will be sent to the user’s registered mobile number for every rollback or adjustment.
  + **Content:** The SMS will briefly inform the user about the rollback or adjustment, including the key details (amount, date, reason).

**3. Notification Content**

* **Details Included:**
  + **Amount:** The exact amount of the rollback or adjustment (credit or debit).
  + **Date and Time:** When the rollback or adjustment occurred.
  + **Reason:** A brief description or code indicating the reason for the rollback or adjustment (e.g., "System Correction," "Customer Request," etc.).
  + **Transaction ID :** A reference to the original transaction that was rolled back or adjusted.

### **Rollback & Adjustment Transaction Receipt**

**1. Receipt Generation**

* **Trigger:**
  + A receipt will be generated upon requst by the system whenever a rollback or balance adjustment transaction is completed.
* **Content:**
  + **Amount:** The exact amount of the rollback or adjustment (credit or debit).
  + **Date and Time:** When the rollback or adjustment was processed.
  + **Reason:** A description or code indicating the reason for the rollback or adjustment (e.g., "System Correction," "Customer Request," etc.).
  + **Source IP**
  + **Type (Credit/Debit)**
  + **Receipt Account**
  + **Channel Type (Web, Mobile)**
  + **Created By:** The administrator or system user who initiated the transaction.
  + **Transaction ID:** A unique identifier for the rollback or adjustment transaction.

**2. Receipt Accessibility**

* **Availability:**
  + **Access:** Users can access their rollback and adjustment receipts through the SalesPoint web portal or Sales App.
  + **User Interface:** A dedicated section in the account history or transaction management area will be available where users can view and manage their receipts.
  + **Filtering:** Users will be able to filter receipts by date range, transaction type (rollback or adjustment), and other relevant criteria.

## eVoucher

### **Get Single Voucher by Sales App**

This feature enables dealers to retrieve a single eVoucher via the Sales App, ensuring secure handling of voucher PINs (HRN) using AES SHA-512 encryption. The system validates the dealer's POS, checks eVoucher inventory, and prioritizes vouchers based on reservation and expiry dates. System shall enforce RBAC based validation to restrict the dealer from accessing only their own voucher details.

**1. Dealer Interface in Sales App**

* **Voucher Denomination Selection:**
  + Dealers can select the voucher denomination they wish to get.
* **Delivery Method**
  + Dealer can select the delivery method (SMS or by Print)

**2. POS Validation**

* **Active Status and Assignment Check:**
  + The system validates whether the POS device is active and correctly assigned to the requesting dealer.
* **Balance Availability Check:**
  + The system checks if the dealer's account, associated with the POS, has sufficient balance to cover the requested voucher amount.

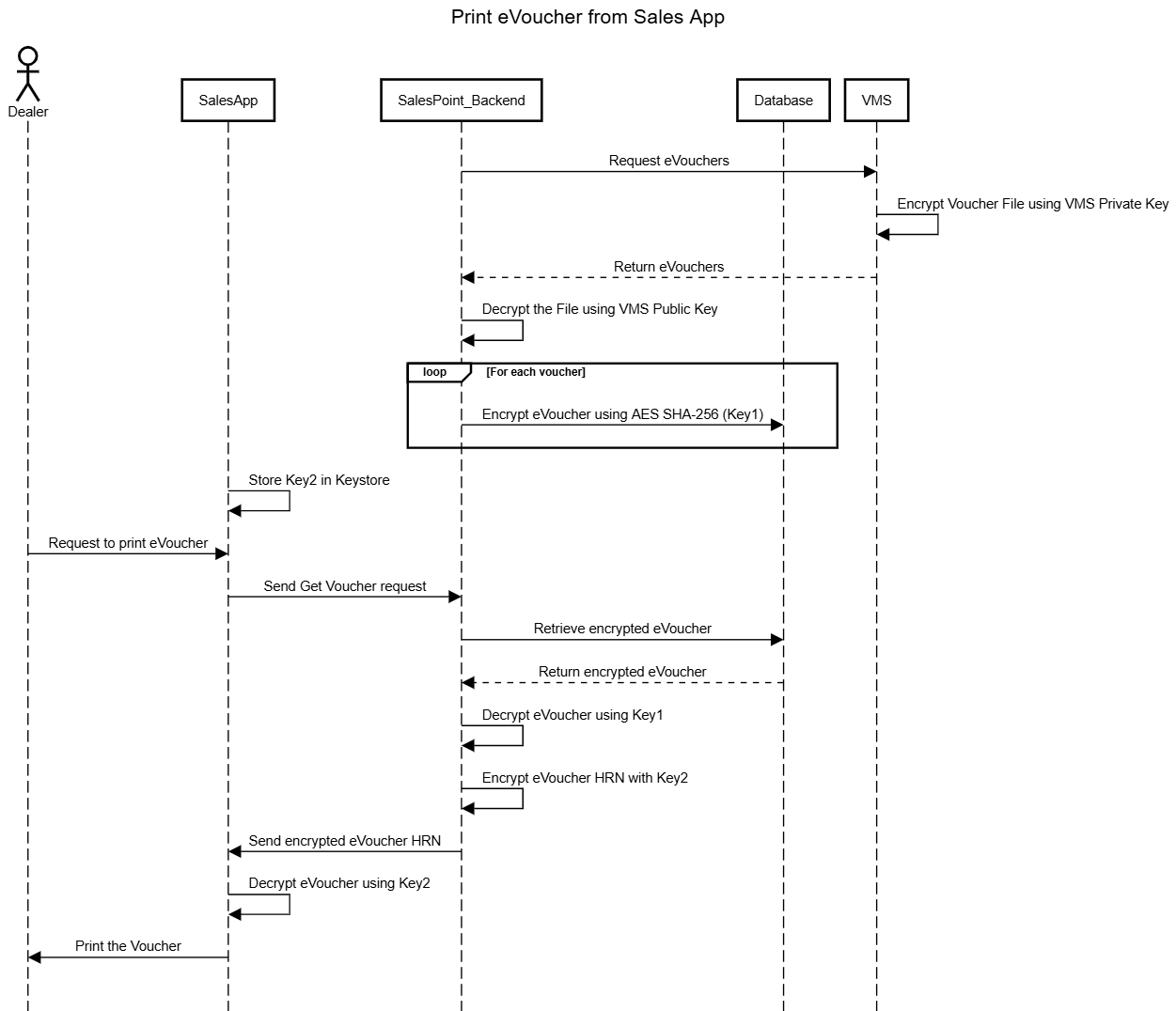
**3. Voucher Inventory Check**

* **Shared Pool Inventory:**
  + The system checks the shared pool for the requested voucher denomination.
* **Unavailable Vouchers Handling:**
  + If the requested voucher denomination is unavailable in the shared pool, the system notifies the dealer with an appropriate message.

**4. Voucher Prioritization**

* **Reservation and Expiry Date Prioritization:**
  + Vouchers are prioritized based on their reservation status and PIN expiry date.

**5. Voucher Commitment & Security**

* **Voucher Commitment Process:**
  + The system commits the voucher only after the dealer selects the "Print" option in case of By Print delivery method.
  + The system commits the voucher only after sending SMS to the customer in case of SMS delivery method.
* **HRN Encryption & Decryption:**
  + **AES SHA-512 Encryption:** The HRN is always encrypted using AES SHA-512 (encryption will be done in server side only) while stored in the system.
  + **Decryption During Print:** Decryption of the HRN occurs only during the print action on the POS device. We will store encryption keys securely on the mobile device using key storage mechanisms provided by the operating system, KeyStore on Android and Secure Enclave on iOS. Avoid hardcoding encryption keys in the application code or storing them in plaintext.
  + **Decryption During Send SMS:** Decryption of the HRN occurs only during send SMS in case of SMS delivery method.
  + **All Decryption actions will be audited in the system.**
  + Below intermediate design for encryption and decryption will be applied, until the permanent design placed in the Q2/2025:
* 

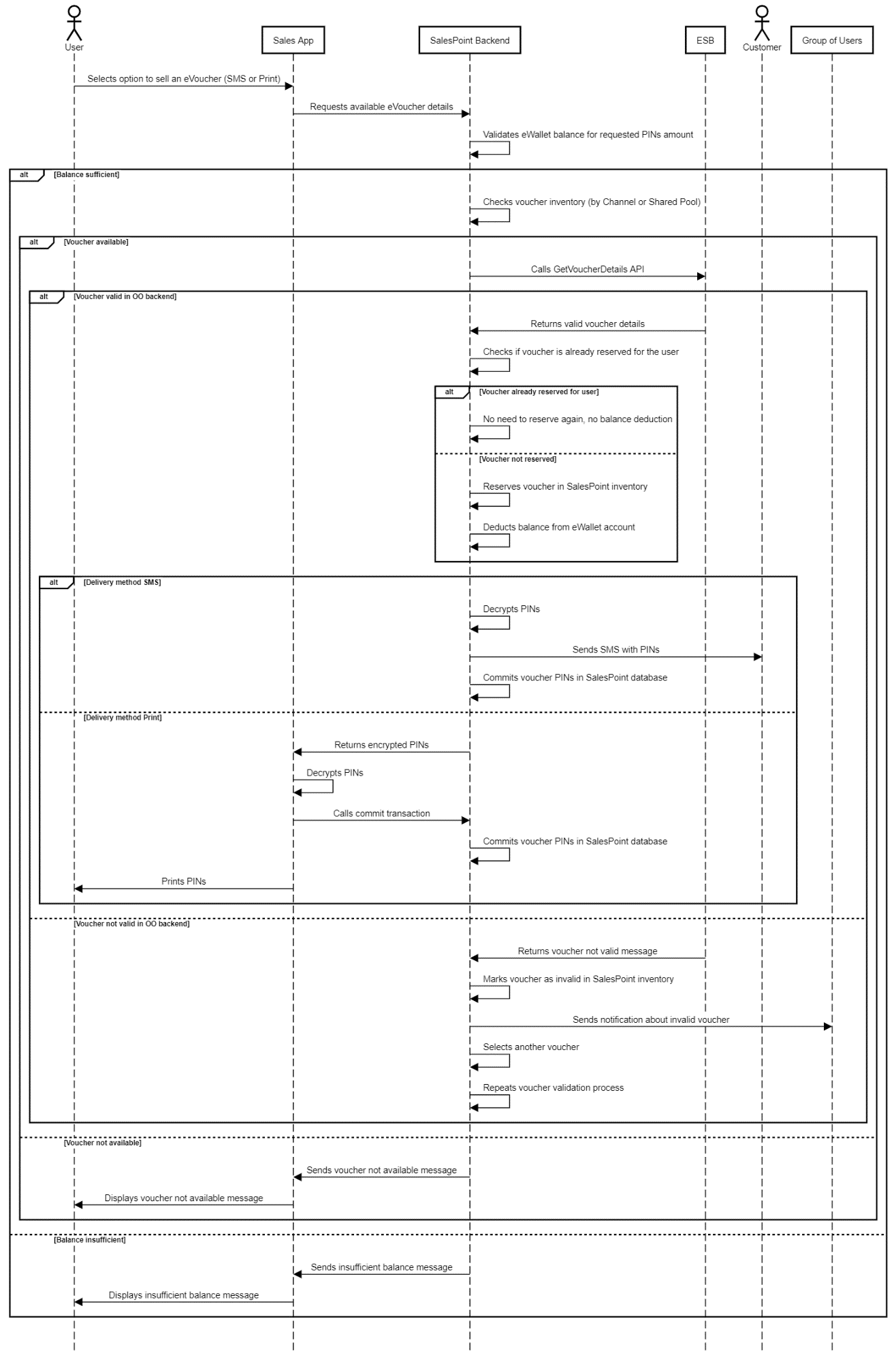
**6. Transaction History Update**

* **Logging eVoucher Details:**
  + The system logs all eVoucher transaction details into the dealer's transaction history, including voucher denomination, transaction ID, and timestamp.

**7. Security & Compliance**

* **Data Protection:** All data actions are securely logged, and HRN handling follows industry-standard security protocols to ensure data protection throughout the process.
* We will store encryption keys securely on the mobile device using key storage mechanisms provided by the operating system, KeyStore on Android and Secure Enclave on iOS. Avoid hardcoding encryption keys in the application code or storing them in plaintext.

**8. Sequence Diagram**



### **Get Single Voucher by SalesPoint Web Portal**

**1. User Interface for Dealers**

* **Voucher Selection Interface:**
  + Dealers can log into the SalesPoint Web Portal and access a dedicated interface to select and retrieve eVouchers.
  + Dealers will select the desired voucher denomination from available options.

**2. Dealer Balance Verification**

* **Dealer Balance Verification:**
  + Upon successful POS validation, the system checks the status and availability of the balance associated with the dealer to ensure sufficient funds are available for the transaction.

**3. Voucher Availability and Allocation**

* **Shared Pool Check:**
  + The system queries the shared pool of vouchers to find a voucher that matches the requested denomination.
  + If no available eVoucher PIN is found in the shared pool, the system displays an appropriate message to the dealer indicating the unavailability.

**4. Voucher Prioritization and Commitment**

* **Prioritization Logic:**
  + The system prioritizes vouchers based on their reservation status and the expiry date of the PIN, ensuring that the most viable voucher is selected for distribution.
* **Voucher Commitment:**
  + Once a suitable voucher is identified and all checks are passed, the system commits the voucher by marking it for delivery.
  + The voucher commitment process includes setting the voucher as 'Sold' to prevent other transactions from claiming the same voucher.

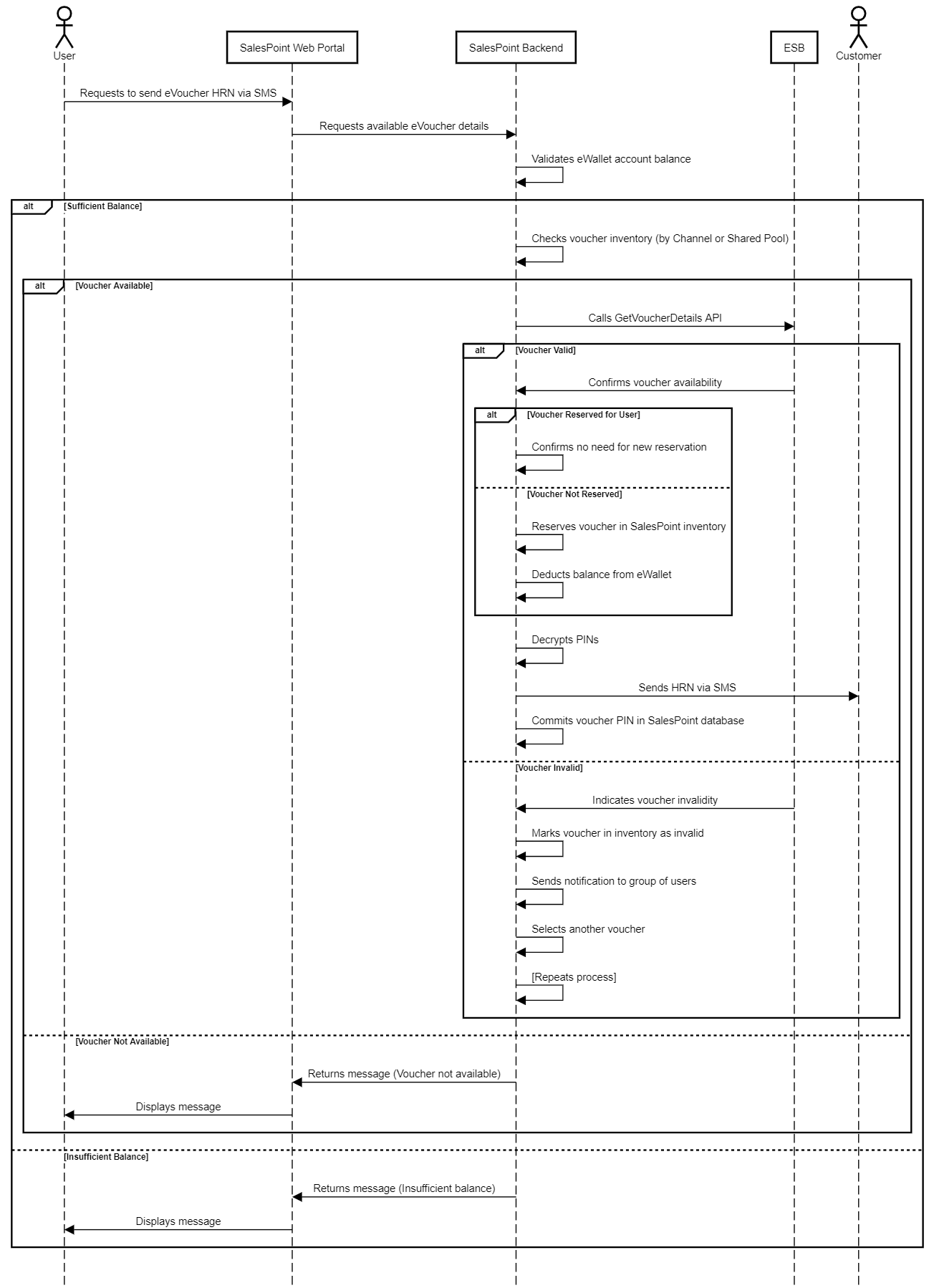
**5. Secure Voucher Distribution**

* **HRN Encryption and Delivery:**
  + The HRN (PIN) is encrypted using AES SHA-512, ensuring top-level security.
  + The system sends the decrypted HRN directly to the customer’s mobile number via SMS, ensuring that the voucher is delivered securely and promptly.

**6. Transaction History Updates**

* **Logging Transactions:**
  + Each transaction involving the retrieval and distribution of an eVoucher is logged in the dealer's transaction history.
  + The log includes details of the transaction such as voucher denomination, date, time, Source IP, Channel Type (Web/App), and the recipient's mobile number.

**7. Sequence Diagram**



### **Get Multiple Voucher by Sales App**

**1. User Interface for Dealers**

* **Voucher Selection Interface:**
  + Dealers can log into the Sales App and access a dedicated interface to select and retrieve multiple eVouchers.
  + Dealers will select the desired voucher denominations and specify the quantity for each.

**2. POS Validation and Dealer Balance Verification**

* **POS Status Check:**
  + The system validates whether the Point of Sale (POS) device is active and correctly assigned to the dealer initiating the request.
  + System must validate a combination of multiple parameters (POS ID, IMEI and Device ID) mapped to dealer account.
* **Dealer Balance Verification:**
  + Upon successful POS validation, the system checks the status and availability of the balance associated with the dealer to ensure sufficient funds are available for the requested vouchers.

**3. Voucher Inventory and Allocation**

* **Shared Pool Inventory Check:**
  + The system checks the shared pool for the requested voucher denominations and quantities.
  + If the requested eVoucher PINs are not available in the shared pool, the system displays an appropriate message to the dealer indicating the unavailability.

**4. Voucher Reservation and Deduction**

* **Voucher Reservation:**
  + If eVoucher PINs are available, the system reserves the required quantity of vouchers from the shared pool.
* **Balance Deduction:**
  + Upon successful reservation, the system deducts the total amount for the reserved vouchers from the dealer's eVoucher Wallet balance.

**5. Voucher Commitment & Security**

* **Voucher Commitment Process:**
  + The system commits the voucher only after the dealer selects the "Print" option in case of By Print delivery method.
  + The system commits the voucher only after sending SMS to the customer in case of SMS delivery method.
* **HRN Encryption & Decryption:**
  + **AES SHA-512 Encryption:** The HRN is always encrypted using AES SHA-512 while stored in the system.
  + **Decryption During Print:** Decryption of the HRN occurs only during the print action on the POS device.
  + **Decryption During Send SMS:** Decryption of the HRN occurs only during send SMS in case of SMS delivery method.
  + **All Decryption actions will be audited in the system.**

**6. Transaction History Update**

* **Logging eVoucher Details:**
  + Upon successful generation and printing of the eVouchers, the system logs all relevant transaction details in the dealer's transaction history, including each voucher's denomination, transaction ID, and timestamp.

**7. Sequence Diagram**

The same sequence diagram for section 4.4.1 will be followed except the system will get multiple vouchers.

### **Get Single Voucher by USSD**

This feature facilitates the acquisition of a single eVoucher via USSD, allowing dealers to interact directly with the SalesPoint system through mobile devices without needing internet access. The system will provide APIs to manage the voucher retrieval process, validate dealer balances, and ensure secure transmission of voucher details.

**1. USSD Integration and API Provision**

* **API for USSD Interaction:**
  + The system will provide a set of APIs to facilitate interaction between the USSD gateway and the SalesPoint system. These APIs will include:
    - **Wallet Account and Balance Validation API:** Validates the dealer’s wallet account and checks balance sufficiency based on input from the USSD system.
    - **Available Denomination API:** Provides information on available voucher denominations that dealers can select.
    - **Single Voucher Request API:** Processes requests for single vouchers based on the dealer’s denomination selection.

**2. USSD Dealer Interaction**

* **Voucher Denomination Selection:**
  + Dealers can select the desired voucher denomination directly via USSD commands, which interact with the SalesPoint system to fetch and display available options.

**3. Voucher Validation and Allocation**

* **Dealer PIN and Balance Validation:**
  + Upon a voucher request, the system first validates the dealer's identity and checks the associated eWallet balance to ensure it can cover the voucher cost.
  + The system will validate the dealer PIN passed in the API.
* **Shared Pool Inventory Check:**
  + The system checks the shared pool for the requested voucher denomination.
  + If no available eVoucher PIN is found in the shared pool, the USSD system displays an appropriate message to the dealer.

**4. Voucher Reservation and Commitment**

* **Voucher Reservation:**
  + If an eVoucher PIN is available, the system reserves the voucher, deducting the amount from the dealer’s eVoucher Wallet balance.
* **Voucher Commitment:**
  + Following successful reservation, the system commits the voucher and prepares it for distribution.

**5. Voucher Distribution**

* **HRN Encryption and SMS Delivery:**
  + The HRN is encrypted and securely stored until the commitment phase.
  + Once committed, the system sends the HRN to the customer’s mobile number via SMS, ensuring the voucher is only accessible to the intended recipient.
  + The system will be sent by SalesPoint backend system to customer mobile number via SNG/SendPoint system.

**6. Transaction Logging**

* **Update Transaction History:**
  + The system logs all details of the voucher transaction in the dealer’s transaction history, including the denomination, transaction ID, and timestamp of the distribution.

**7. Security and Compliance**

* **Secure API Communication:**
  + All API communications between the USSD gateway and the SalesPoint system are secured with HTTPS and bearer token to prevent unauthorized access and data breaches.
* **Audit Trails:**
  + The system maintains detailed audit trails for all operations related to eVoucher transactions via USSD, facilitating compliance with regulatory standards and internal audits.

**8. Sequence Diagram**

A screenshot of a computer screen

Description automatically generated

### **Download Voucher**

This feature allows dealers to download multiple eVouchers via the Sales App, targeting their Point of Sale (POS) devices. It involves validation of POS status, balance checks, voucher allocation from a shared pool, and secure PIN downloading.

**System Components**

1. **Sales App:**
   * Interface for dealers to select and initiate the download of eVouchers.
2. **SalesPoint Backend:**
   * Manages all backend operations including dealer validations, balance checks, voucher reservations, and transaction logging.
3. **POS Device:**
   * Endpoint for downloading and storing eVouchers for transactional use.
   * eVoucher should be stored in encrypted format in Keystore.

**Functional Flow**

1. **Dealer Interaction:**
   * Dealers log into the Sales App to select voucher denominations and quantities.
2. **POS and Balance Validation:**
   * System validates the POS device for active status and proper assignment using POS MAC address and if it’s assigned to the requested dealer.
   * Checks dealer’s balance for sufficiency against the total voucher cost.
3. **Voucher Allocation:**
   * System searches the shared pool for the requested denominations.
   * If unavailable, displays an appropriate message.
   * If available, reserves vouchers and deducts the corresponding amount from the dealer’s eWallet.
4. **Voucher Download and Commitment:**
   * Committed vouchers are downloaded directly to the dealer’s POS device.
   * POS devices use only Android OS, so vouchers PINs will be stored in Android Keystore system.
   * The system updates the transaction as complete upon successful download.
5. **Transaction History Update:**
   * Detailed transaction records are updated in real-time, reflecting the downloaded voucher details in the dealer’s account.

### **Void Voucher**

This feature enables DSMs and distributors to void and potentially re-print eVouchers through both the Sales App and SalesPoint Web Portal. The process involves validating the active status of the voucher, handling reprint requests, and ensuring transaction integrity and auditability.

**System Components**

1. **Sales App & SalesPoint Web Portal:**
   * Interfaces where DSMs and distributors can initiate voucher voiding actions.
   * Allow reprinting of eVouchers under specific conditions.
   * From Sales App shall use invisible captcha, from portal shall use google captcha.
   * System should validate that voiding or reprint of only those voucher shall be allowed which are already purchased or committed for a given dealer/distributor.
2. **SalesPoint Backend:**
   * Manages validation, voiding processes, and reprinting actions.
   * Handles transaction updates and system responses based on voucher status checks.
3. **Integrated Network (IN):**
   * Validates the active and unused status of eVouchers.
   * Ensures eVouchers are eligible for voiding and, if applicable, reprinting.

**Functional Flow**

1. **Voiding Request Initiation:**
   * DSM logs into the Sales App only.
   * Distributor logs into the Sales App or Web Portal.
   * Requests voiding of an eVoucher using the serial number provided by the dealer.
2. **Voucher Validation:**
   * SalesPoint Backend queries the IN to confirm the voucher is still active and has not been used via ESB API “**GetVoucherDetails**”.
   * If the voucher is valid and unused, the system processes the voiding request.
3. **Reprint Permissions and Validation:**
   * If a voided voucher needs to be reprinted:
     + Dealers if got permission from DSM or Distributor to re-print the voucher, re-print option will be available for the dealer in Sales App voucher history.
     + System re-validates the voucher's unused status before permitting reprint.
4. **Transaction History Update:**
   * All actions related to voiding and reprinting are recorded in the dealer's transaction history.
   * Detailed logs include the original voucher serial number, the action taken, and the outcome.
5. **Error Handling and Feedback:**
   * If the voucher is already used or if there are issues with dealer or POS validation, the system rejects the voiding request.

### **Reserve Voucher (Offline Distributor)**

This feature allows offline distributors to reserve and download multiple eVouchers through the SalesPoint portal. This will allow distributors to use their own system for selling eVouchers. The system manages voucher reservations, prevents duplication, handles offline storage, and tracks voucher usage.

**System Components**

1. **SalesPoint Portal:**
   * User interface for distributors to manage eVoucher reservations, downloads, and transactions.
2. **SalesPoint Backend:**
   * Handles all backend processing, including balance validation, voucher reservation, duplicate check, and transaction logging.
3. **Voucher Management System (VMS):**
   * Monitors voucher inventory, manages reservation priorities, and tracks usage status updates.

**Functional Flow**

1. **Voucher Selection and Request:**
   * Distributors log into the SalesPoint portal and select the desired voucher denominations and quantities.
   * The system allows them to input these selections through a user-friendly interface.
   * Distributor shall enter OTP code before proceeding in download vouchers.
2. **Balance and PIN Validation:**
   * The system shall validate the distributor Wallet PIN code before processing the transaction.
   * The system validates the distributor's eWallet balance to ensure it can cover the total cost of the requested vouchers.
   * It checks the availability of requested voucher denominations in the shared pool.
   * If the balance is insufficient or no vouchers are available, the system notifies the distributor with an appropriate message.
3. **Voucher Reservation:**
   * If the requested vouchers are available, the system reserves them and deducts the corresponding total amount from the distributor’s eWallet.
   * The vouchers are prioritized based on their reservation status and expiry date.
4. **Offline Storage and Download:**
   * The system generates an Excel sheet containing the reserved eVoucher PINs, which the distributor can download to their device.
   * System will push OTP using PRNG algorithm to the user email, and force the user to enter the received OTP. Resend OTP and number of allowed OTP failed attempts will be applied as per ISM recommendation which was implemented in other processes in SalesPoint system.
   * System will download the vouchers decrypted to allow distributor to feed their system as per current business process.
   * System will create the excel sheet with random and strong password to be able to open it.
   * System will send the password to distributor via SMS. Each time the password generated should be unique (using PRNG based algorithm), and random with minimum length of 12 characters.
5. **Voucher Usage and Status Update:**
   * As the vouchers are used, the distributor reports usage back to the SalesPoint system.
   * The system syncs daily with the VMS to update the voucher status to "Used" based on the VMS's voucher status updates.
   * The system ensures that no duplicate requests are processed, maintaining data integrity.
6. **Transaction Logging:**
   * The system updates the distributor transaction history with all relevant details of the reserved and downloaded vouchers, including the denominations, quantities, and status changes.

## Generate ERP Sale

### **ERP Wallet Balance**

This feature enables the SalesPoint system to interact with an ERP system to manage the balance of distributor wallets. It allows for the automated topping up of distributor master wallets based on requests received from the ERP system, ensuring proper validation and communication.

**System Components**

1. **ERP System:**
   * ERP system responsible for sending wallet balance requests to the SalesPoint system.
2. **SalesPoint Backend:**
   * Core component that processes requests from the ERP system, validates the sales order number, updates the distributor's wallet balance, and handles notifications.
   * The wallet that receives the total amount specified in the ERP request.
   * Sends transaction details to the corresponding distributors after processing the ERP request.

**Functional Flow**

1. **Wallet Balance Request from ERP:**
   * The ERP system sends a wallet balance request to the SalesPoint backend, specifying the total amount to be added to the distributor's master wallet.
2. **Sales Order Validation:**
   * The SalesPoint backend validates the sales order number to ensure it is unique and has not been processed previously.
   * If a duplicate request is detected, the system rejects it and logs the event for audit purposes.
3. **Distributor Wallet Top-Up:**
   * Upon successful validation, the system credits the specified total amount to the distributor’s master wallet.
   * Based on Item code system will credit the mapped item code to the assigned wallet account to that item code.
   * The transaction is recorded in the distributor's wallet history, ensuring full traceability.
4. **Distributor Notification:**
   * After the wallet top-up is completed, the system sends an email notification to the corresponding distributor.
   * The email includes details of the transaction, such as the sales order number, amount credited, and the updated wallet balance.

**Security and Compliance**

* **Audit Trail:**
  + The system maintains detailed logs of all sales order validations, wallet top-ups, and notifications to ensure compliance and support auditing processes.

### **Voucher Dashboard & Availability**

This feature provides administrators with a comprehensive dashboard within the SalesPoint system to monitor and manage the availability of vouchers in the shared pool. The dashboard allows for real-time visibility into voucher stock levels by denomination and includes tools for filtering and searching.

**System Components**

1. **SalesPoint Admin Portal:**
   * The interface where administrators access the voucher dashboard.
   * Allows for real-time monitoring, filtering, and searching of voucher availability by denomination.
   * Actual voucher code (HRN) will not be shown in UI.
2. **SalesPoint Backend:**
   * Manages the data processing, aggregation, and display of voucher availability.
   * Handles the filtering and search functionalities, providing administrators with accurate and up-to-date information.
   * Tracks and manages the stock levels of all voucher denominations in the shared pool.

**Functional Flow**

1. **Dashboard Access:**
   * Administrators log into the SalesPoint Admin Portal and navigate to the Voucher Dashboard section.
   * The dashboard interface is designed for ease of use, providing a clear overview of voucher availability.
2. **Real-Time Voucher Availability Display:**
   * The dashboard displays the current availability of vouchers in the shared pool, categorized by denomination.
   * The system automatically updates the dashboard as new vouchers are added or used, ensuring that administrators have the latest information.
3. **Filtering and Search Options:**
   * The system provides filtering options that allow administrators to narrow down the displayed data based on specific voucher denominations or other criteria.
   * Administrators can use the search function to quickly find information on a particular denomination, streamlining the process of managing voucher availability.

### **Voucher PIN Generation via VMS System**

This feature involves generating voucher PINs through the VMS (Voucher Management System) by utilizing a sequence of API calls from the SalesPoint backend. The process includes requesting PIN generation, monitoring the task progress, loading the generated file into an SFTP folder, and finally processing the file to import the PINs into the SalesPoint system.

This functionality shall be limited to admin users based on their role (logistics team)

**System Components**

1. **SalesPoint Backend:**
   * Manages the entire process of PIN generation by interacting with the VMS through a series of API calls.
   * Handles the monitoring of task progress, loading of voucher files, and final parsing of the PINs.
2. **Voucher Management System (VMS):**
   * External system responsible for generating and managing voucher PINs.
   * Provides APIs for PIN generation, task monitoring, file loading, and status checking.
3. **SFTP Server:**
   * Secure storage location where the VMS places the generated voucher PIN files.
   * Monitored by the SalesPoint backend for the presence of new files to be processed.

**API Integration Flow**

1. **GenerateVoucher API Call:**
   * **Purpose:** Initiates the voucher PIN generation process.
   * **Process:**
     + The SalesPoint backend sends a "GenerateVoucher" API request to the VMS, specifying the number of PINs required, denominations, and other relevant parameters.
     + VMS begins the PIN generation process and returns a task ID for tracking.
2. **GetGenerateVoucherTaskInfo API Call:**
   * **Purpose:** Monitors the status of the PIN generation task.
   * **Process:**
     + The SalesPoint backend repeatedly calls the "GetGenerateVoucherTaskInfo" API using the task ID until the VMS indicates that the PIN generation is complete.
     + Once the generation task is marked as ready, the process moves to the next step.
3. **LoadVoucherBatchFile API Call:**
   * **Purpose:** Requests the VMS to load the generated PIN file into the VMS Server ~~SFTP folder~~.
   * **Process:**
     + The SalesPoint backend calls the "LoadVoucherBatchFile" API to instruct the VMS to place the generated PIN file in the designated SFTP folder.
     + The VMS begins loading the file into the SFTP folder.
4. **LoadVoucherCheck API Call:**
   * **Purpose:** Confirms that the PIN file has been successfully loaded into the VMS Server ~~SFTP folder~~.
   * **Process:**
     + The SalesPoint backend periodically calls the "LoadVoucherCheck" API to verify the file’s loading status.
     + Once a successful response is received, confirming that the file is ready, the SalesPoint backend proceeds to check the VMS Server ~~SFTP folder~~.
5. **SFTP File Monitoring and Parsing:**
   * **Purpose:** Finalizes the process by Decrypt and importing the generated PINs into the SalesPoint system.
   * **Process:**
     + The SalesPoint backend checks the SFTP folder for the presence of the generated PIN file.
     + The PIN Files generated by VMS system must be encrypted using asymmetric crypto.
     + SalesPoint will decrypt the file first before importing.
     + The file is then parsed, and the PINs are Decrypted using separate symmetric key and imported into the eReload system.
     + System will encrypt each voucher individually using Asymmetic crypto key and store it in the database encrypted.
     + Once the voucher is imported, the file should be deleted securely.
     + The system updates the administrators on the status of the entire process, including any errors encountered during parsing.

**Security and Audit**

* **Audit Trail:**
  + Every API call and action related to the PIN generation and import process is logged, providing a complete audit trail for compliance and troubleshooting.

**Sequence Diagram**

**A diagram of a project

Description automatically generated with medium confidence**

### **Notification with SLA for VMS PINs Order**

This feature ensures that stakeholders are informed about the status of VMS PIN orders in a timely manner. The system monitors the order processing against predefined Service Level Agreements (SLAs) and sends notifications about the order status, expected completion times, and any delays. If an SLA breach occurs, the system escalates the notification to inform relevant stakeholders.

**System Components**

1. **SalesPoint Backend:**
   * Manages the processing of VMS PIN orders, including tracking order status and monitoring SLA adherence.
   * Handles the generation and distribution of notifications based on order status and SLA compliance.
   * Sends notifications via email and SMS to inform stakeholders of the status of VMS PIN orders.
   * Escalates notifications to stakeholders if SLA breaches occur.
   * Tracks the time taken for each step in the VMS PIN order process.

**Functional Flow**

1. **Order Status Monitoring:**
   * The SalesPoint Backend continuously monitors the status of VMS PIN orders, including the time taken at each processing stage.
   * The SLA Management Module tracks the elapsed time against the defined SLAs for each order.
2. **Notification Triggering:**
   * As the order progresses, the system triggers notifications at key stages, such as:
     + Order initiation
     + PIN generation in progress
     + File loading into SFTP
     + PIN import into the eReload system
   * Notifications include the current status of the order, the expected completion time, and any potential delays identified during processing.
3. **SLA Compliance Check:**
   * SalesPoint checks if the processing time for each order step remains within the defined SLA limits.
   * If an order is likely to breach the SLA, the system sends a warning notification to relevant stakeholders.
4. **Final Notification:**
   * Upon completion of the VMS PIN order, the system sends a final notification confirming the successful processing and delivery of the order.
   * The notification includes a summary of the processing time and adherence to the SLA.

### **Safety Stock Notification for Each Denomination**

This feature monitors voucher stock levels by denomination and sends alerts when stock falls below specific thresholds. Administrators can configure these thresholds, and the system will escalate notifications if stock levels continue to decline.

**Key Components**

1. **SalesPoint Backend:**
   * Monitors stock levels in real-time for each voucher denomination.
   * Triggers notifications when stock falls below set thresholds.
   * Sends alerts to stakeholders when stock is low.
2. **Admin Interface:**
   * Allows administrators to set and adjust stock thresholds for each denomination.

**Functional Flow**

1. **Stock Monitoring:**
   * The system continuously checks voucher stock levels by denomination.
2. **Threshold Configuration:**
   * Administrators set stock thresholds through the admin interface.
3. **Notifications:**
   * The system sends an alert when stock falls below the configured threshold.

### **OO Logistics Team Role Access in SalesPoint Admin Portal**

We will enable the OO (Ooredoo) Logistics team to access specific functionalities within the SalesPoint Admin Portal using their Ooredoo Active Directory (AD) accounts. The team will have role-based access to various tools and reports necessary for managing eVoucher operations.

**Key Components**

1. **SalesPoint Admin Portal:**
   * The interface through which the OO Logistics team will access their designated roles and functionalities.
2. **Active Directory (AD) Integration:**
   * Allows the OO Logistics team to log in using their Ooredoo AD credentials, ensuring secure and streamlined access.
3. **Role-Based Access Control (RBAC):**
   * Manages permissions, ensuring that the Logistics team only has access to specific functionalities within the SalesPoint Admin Portal.

**Functional Flow**

1. **Login via AD:**
   * The OO Logistics team members log in to the SalesPoint Admin Portal using their Ooredoo AD accounts.
   * The system authenticates users against the Ooredoo AD and grants access based on predefined roles.
2. **Access Roles for OO Logistics Team:**
   * **eVoucher Dashboard (Shared Pool):**
     + View real-time availability and status of eVouchers in the shared pool.
   * **Generate Voucher Requests (with status):**
     + Initiate and monitor the status of voucher generation requests.
   * **View Generate Voucher Request and Workflow:**
     + Access details and track the workflow of all voucher generation requests.
   * **Distributor & Dealer Wallet Balance & Transactions:**
     + View the balance and transaction history of distributors and dealers.
   * **eVoucher Transactions:**
     + Monitor all eVoucher-related transactions.
   * **eVoucher Inventory Details:**
     + Access detailed information about eVouchers, including serial numbers, expiry dates, value, denomination, and status.
   * **eReload Reports:**
     + Generate and view reports related to eReload activities.

### **Distributors Role Access in SalesPoint Web Portal**

This feature provides distributors with secure access to the SalesPoint Web Portal. It includes user creation, role-based access, and various tools specifically tailored for managing their distribution hierarchy and related transactions.

**Key Components**

1. **SalesPoint Web Portal:**
   * The interface through which distributors and their users access the system functionalities.
2. **User Management System:**
   * Enables the creation and management of user accounts for distributors by Ooredoo Admin.
3. **Multi-Factor Authentication (MFA):**
   * Adds an additional layer of security through OTP (One-Time Password) when distributor users log in.
4. **Role-Based Access Control (RBAC):**
   * Defines and manages the specific roles and permissions available to distributor users.

**Functional Flow**

1. **User Creation and Access:**
   * **User Creation:**
     + Administrators can create user accounts for distributors, assigning them specific roles based on their needs.
   * **Login Process:**
     + Distributor users log in using a username and password.
     + An OTP via SMS is sent to the user for multi-factor authentication (MFA) to ensure secure access.
2. **Access Roles for Distributor Users:**
   * **View Related DSM and Dealers Only:**
     + Users can view only the DSMs (District Sales Managers) and dealers associated with their distribution hierarchy.
   * **Related Wallet Access:**
     + Users can access the distributor's master wallet and related dealer wallets only.
   * **Wallet Transfer:**
     + Users can transfer funds between the distributor master wallet and related dealer wallets within their hierarchy.
   * **Distributor Wallet Balance & Transactions:**
     + Users can view the balance and transaction history of the distributor’s wallet.
   * **eVoucher Transactions:**
     + Users can monitor eVoucher transactions related to the dealers under their hierarchy.
   * **Create DSM/Dealer:**
     + Users can create DSMs and dealers within their own distribution hierarchy.
   * **Stock Transfer:**
     + Users can transfer stock (eVouchers) between entities within their hierarchy.
   * **Stock Balance Monitoring:**
     + Users can monitor stock balances under their hierarchy and check stock levels of their child entities.
   * **Movement Tracking:**
     + Users can track all stock and wallet movements within their hierarchy.
   * **View and Verify Vouchers:**
     + Users can view eVouchers and verify serial numbers against physical printers.
     + Voucher code (HRN) will not be visible in UI.

**Security**

* **MFA with OTP:**
  + Provides an additional security layer, ensuring that only authorized distributor users can access the portal.
* **Role-Based Permissions:**
  + Access to functionalities and data is strictly controlled by roles assigned to the distributor users, ensuring they only see and manage what is relevant to their hierarchy.

## Integration with Enhance

This feature integrates the SalesPoint system with Enhance Partner to retrieve EVD (Electronic Voucher Distribution) transactions and dealer balances. The data is synchronized daily, ensuring that the SalesPoint Admin Portal reflects the most up-to-date dealer balances and transactions.

**Key Components**

1. **SalesPoint Backend:**
   * Manages the integration with Enhance Partner, including retrieving and processing EVD transactions and dealer balances.
2. **Enhance Partner System:**
   * External system that provides EVD transaction data and dealer balance information.
3. **SalesPoint Admin Portal:**
   * Interface where administrators can view the synchronized dealer balances and transactions.
4. **Data Sync Module:**
   * Handles the daily synchronization of data between the Enhance Partner system and the SalesPoint system.

*Note: Integration Document and APIs not shared by Distributor Enhance yet.*

## Postpaid Bill Payment eWallet

This feature introduces a separate eWallet dedicated to postpaid bill payment transactions. Distributors can transfer funds to two different wallets for their child accounts, and administrators have the flexibility to configure bill payment deductions to occur from either one or both wallets. Additionally, ERP can topup the postpaid bill payment eWallet using new Item code in the same recharge wallet API.

**Key Components**

1. **Postpaid Bill Payment eWallet:**
   * A separate wallet specifically for handling postpaid bill payment transactions within the SalesPoint system.
2. **SalesPoint Backend:**
   * Manages wallet transfers, configuration settings, and integration with the ERP system for crediting the postpaid bill payment eWallet.
3. **SalesPoint Admin Portal:**
   * Interface for administrators to configure wallet usage settings and manage bill payment options.
4. **ERP Integration:**
   * Allows the ERP system to credit the postpaid bill payment eWallet with funds, ensuring the wallet is sufficiently funded for bill payments.

**Functional Flow**

1. **eWallet Creation and Management:**
   * The system creates a separate eWallet specifically for postpaid bill payments.
2. **Administrator Configuration:**
   * **Wallet Deduction Settings:**
     + Administrators can configure whether postpaid bill payments are deducted from the postpaid bill payment eWallet, the general eWallet, or a combination of both.
     + This setting can be adjusted based on market conditions, allowing flexibility in how payments are managed.
   * **Transfer Management:**
     + Admins can enable or disable the ability for distributors to transfer funds between the two wallets.
3. **ERP Integration for eWallet Credit:**
   * The ERP system sends credit instructions to the SalesPoint backend to add funds to the postpaid bill payment eWallet using the same Recharge wallet API with new Item code for postpaid bill payment balance.
   * The system processes these credits and updates the wallet balance accordingly.
4. **Transaction Handling:**
   * When a postpaid bill payment is initiated, the system checks the configured settings to determine from which wallet(s) the payment will be deducted.
   * The transaction is processed, and the wallet balances are updated in real-time.
5. **Reporting and Monitoring:**
   * All transactions involving the postpaid bill payment eWallet are logged and can be viewed in the SalesPoint Admin Portal.
   * Administrators can monitor wallet balances, transaction histories, and audit logs.

**Security**

* **Access Control:**
  + Only authorized users can configure wallet settings, transfer funds, or initiate postpaid bill payments.

## Bulk Customer Recharge

This feature allows system administrators to perform bulk top-ups of eWallet accounts by uploading an Excel or CSV file through the SalesPoint system. The system will validate the uploaded data, process the top-ups, and log the results, ensuring both efficiency and accuracy.

**Key Components**

1. **SalesPoint Admin Portal:**
   * User interface where administrators can upload the XLSX/CSV file for bulk eWallet top-ups.
2. **SalesPoint Backend:**
   * Validates the structure and content of the uploaded files, ensuring data integrity before processing.
   * Handles the efficient processing of valid top-up requests from the uploaded file and logs any errors.
   * Updates the eWallet balances based on the processed data and reflects these changes in the transaction history.
3. **Role-Based Access Control (RBAC):**
   * Ensures that only authorized administrators can access and use the bulk top-up feature.

**Functional Flow**

1. **Template Provision:**
   * **Download Template:**
     + The system provides an option for administrators to download a predefined template (Excel/CSV) that includes the necessary fields:
       - Channel Member Code (mandatory)
       - Amount (mandatory, must be positive)
       - eWallet Type (mandatory)
       - PO Number (mandatory)
       - Note (optional)
   * **Template Structure:**
     + The template ensures the correct structure for successful file validation and processing.
2. **File Upload and Initial Validation:**
   * **File Upload:**
     + Administrators upload the filled Excel/CSV file through the SalesPoint Admin Portal.
   * **Initial Validation:**
     + The system checks the file format and structure against the template.
     + It also validates the following:
       - Correct file format (Excel/CSV)
       - Presence of all mandatory fields
       - Valid Channel Member Codes
       - Valid eWallet Account Types
       - Positive top-up amounts
3. **Error Handling:**
   * **Invalid Data Detection:**
     + If the file contains invalid data, the system rejects the file and provides specific error messages indicating the issues (e.g., invalid Channel Member Code, negative amounts).
   * **Feedback to User:**
     + The system returns the error messages to the administrator, allowing them to correct the file and re-upload if necessary.
4. **Bulk Processing:**
   * **Processing Valid Entries:**
     + The system processes all valid entries in the file, updating the corresponding eWallet accounts.
   * **Logging Invalid Entries:**
     + Entries with errors are logged, and the system provides a summary of these errors to the administrator.
5. **Summary Report:**
   * **Operation Summary:**
     + After processing, the system generates a summary report, detailing:
       - The number of successful top-ups
       - Any errors encountered
       - The total amount credited to eWallets
   * **Downloadable Report:**
     + Administrators can download this report for audit and record-keeping purposes.
6. **Transaction Reflection:**
   * **Transaction History Update:**
     + The system reflects each top-up in the eWallet transaction history, ensuring transparency and traceability.
7. **Security:**
   * **Role-Based Access:**
     + The system uses RBAC to restrict access to the bulk top-up feature, ensuring that only authorized personnel can perform these operations.
   * System must validate the file type, extension, presence of any special character in file name, content type, MIME Type and Magic bytes.
   * System should also enforce strict input validation on the cell content (MSISDN & Amount) before processing the request.

## Bulk eWallet Topup

This feature enables distributors to perform bulk top-ups of prepaid customer accounts by uploading an Excel or CSV file through the SalesPoint web portal system. The system validates the uploaded data, processes the top-ups, and logs the results, ensuring both efficiency and accuracy.

**Key Components**

1. **SalesPoint Web Portal:**
   * User interface where distributor can upload the Excel/CSV file for bulk prepaid customer top-ups.
   * Provides a comprehensive view of all bulk prepaid recharge requests, including details like MSISDN, amount, and status.
2. **SalesPoint Backend:**
   * Validates the structure and content of the uploaded files, ensuring data integrity before processing.
   * Verifies the availability of funds in the corresponding eWallet.
   * Updates eWallet balances based on the processed data and reflects these changes in the transaction history.
3. **Bulk Processing Engine (Airflow):**
   * Orchestrates the efficient processing of valid top-up requests from the uploaded file, leveraging Airflow for workflow management.
   * Logs any errors encountered during the processing.
4. **Role-Based Access Control (RBAC):**
   * Ensures that only authorized administrators can access and use the bulk top-up feature.
5. **ESB:**
   * Handles the actual top-up of prepaid customer accounts based on the validated data from the uploaded file.

**Functional Flow**

1. **Template Provision:**
   * **Download Template:**
     + The system provides an option for administrators to download a predefined template (Excel/CSV) that includes the necessary fields:
       - MSISDN (mandatory)
       - Amount (mandatory, must be positive)
   * **Template Structure:**
     + The template ensures the correct structure for successful file validation and processing.
2. **File Upload and Initial Validation:**
   * **File Upload:**
     + Administrators upload the filled Excel/CSV file through the SalesPoint Admin Portal, specifically within the wallet page associated with the eTopup account type.
   * **Initial Validation:**
     + The system checks the file format and structure against the template.
     + It also validates the following:
       - Correct file format (Excel/CSV)
       - Sufficient eWallet balance for the total top-up amount in the sheet
       - Positive top-up amounts
       - No duplicate MSISDN entries in the same sheet
3. **Error Handling:**
   * **Invalid Data Detection:**
     + If the file contains invalid data, the system rejects the file and provides specific error messages indicating the issues (e.g., insufficient balance, duplicate numbers, negative amounts).
   * **Feedback to User:**
     + The system returns the error messages to the administrator, allowing them to correct the file and re-upload if necessary.
4. **Bulk Processing via Airflow:**
   * **Processing Valid Entries:**
     + The system uses Airflow to orchestrate the processing of all valid entries in the file, updating the corresponding prepaid customer accounts.
   * **Logging Invalid Entries:**
     + Entries with errors are logged, and the system provides a summary of these errors to the administrator.
5. **Summary and Reporting:**
   * **Operation Summary:**
     + After processing, the system generates a summary report, detailing:
       - The number of successful top-ups
       - Any errors encountered
       - The total amount debited from the eWallet
   * **Bulk Recharge Request Viewer:**
     + Administrators can view all bulk prepaid recharge requests, including details of each MSISDN, the amount recharged, and the status of each recharge.
6. **Transaction Reflection:**
   * **Transaction History Update:**
     + The system reflects each top-up in the eWallet transaction history, ensuring transparency and traceability of all eTopup activities.
7. **Access Control:**
   * **Role-Based Access:**
     + The system uses RBAC to restrict access to the bulk top-up feature, ensuring that only authorized personnel can perform these operations.

## PIN Code

This feature allows dealers to reset their PIN codes via the SalesPoint web portal. The system sends the new PIN code to the dealer via SMS, logs the reset transaction for auditing purposes, and provides administrators with the ability to view the dealer's PIN code. The length of the PIN code is configurable.

**Key Components**

1. **SalesPoint Admin Portal:**
   * User interface where admin can request a PIN code reset.
   * No one should be able to view the PIN code. Admin may have an option to trigger PIN reset which will send the newly generated PIN to dealer via SMS.
   * ~~Provides administrators with access to view dealer PIN codes.~~
2. **SalesPoint Backend:**
   * Manages the generation, configuration, and resetting of PIN codes.
   * Handles the delivery of the new PIN code to the dealer via SMS.
   * Records all PIN reset transactions for auditing purposes, including details such as the time of reset and the requesting user.
   * Allows administrators to configure the length of the PIN code (e.g., 4, 6, 8 digits).
3. **SMS Gateway:**
   * Responsible for sending the new PIN code to the dealer’s registered mobile number.

**Functional Flow**

1. **PIN Reset Request:**
   * **Admin Access:**
     + Administrators have access to view the current PIN code of a dealer through the portal and click reset PIN code.
2. **PIN Generation and Configuration:**
   * **PIN Length Configuration:**
     + The system generates a new PIN code based on the configured length (e.g., 4, 6, 8 digits).
   * **Secure Generation:**
     + The system securely generates the PIN code and ensures it meets the required security standards.
     + The system should use Secure PRNG based algorithm to generate PIN
     + PIN must be stored in encrypted format
3. **PIN Delivery via SMS:**
   * The new PIN code is sent to the dealer’s registered mobile number via the integrated SMS Gateway.
4. **Audit Logging:**
   * The system logs the PIN reset transaction in the audit logging module.
   * The log includes details such as the dealer’s ID, the time of reset, the admin or system user who initiated the reset, and the outcome (success or failure).
5. **Administrator View Access:**
   * Administrators can view the current PIN code assigned to each dealer within the SalesPoint web portal, subject to role-based access controls.

**Security**

* **PIN Code Security:**
  + PIN codes are generated using a cryptographically secure random number generator (CSPRNG) to generate a sequence of digits. **random.SystemRandom (Python):** Uses the underlying system’s secure random number generator.
* **Access Control:**
  + Only authorized users can initiate PIN resets or view dealer PIN codes, ensuring sensitive information is protected.
* **Audit Trail:**
  + All PIN-related activities are logged for audit and compliance purposes, providing a complete history of changes and access.

## Customer Nickname

This feature allows customers to manage a nickname for their mobile number via SMS. The nickname will be used for customer recharge and bill payment processes. The system supports the creation, retrieval, update, and deletion of nicknames through specific SMS commands.

**Key Components**

1. **SMS Gateway:**
   * Handles the receipt and processing of SMS commands sent by customers (e.g., "CREATE", "GET", "UPDATE", "DELETE").
   * Manages the communication between customers and the SalesPoint system.
2. **SalesPoint Backend:**
   * Responsible for creating, retrieving, updating, and deleting nicknames associated with customer mobile numbers.
   * Enforces nickname validation rules such as length, allowed characters, and uniqueness.
   * Processes the nickname management commands and interacts with the database to store and retrieve nickname information.
   * Sends responses back to the customer via the SMS Gateway.

**Functional Flow**

1. **SMS Command Processing:**
   * **Create Nickname:**
     + The customer sends an SMS with the text "CREATE <nickname>" to the designated number (e.g., "80001").
     + The SMS Gateway forwards the command to the SalesPoint Backend.
     + The system validates the nickname (length, characters) and, if valid, creates the nickname in the database.
     + A confirmation SMS is sent back to the customer.
   * **Retrieve Nickname:**
     + The customer sends an SMS with the text "GET" to "80001".
     + The system retrieves the nickname associated with the customer's mobile number from the database.
     + The nickname is sent back to the customer via SMS.
   * **Update Nickname:**
     + The customer sends an SMS with the text "UPDATE <new\_nickname>" to "80001".
     + The system validates the new nickname and, if valid, updates the existing nickname in the database.
     + A confirmation SMS is sent back to the customer.
   * **Delete Nickname:**
     + The customer sends an SMS with the text "DELETE" to "80001".
     + The system deletes the nickname associated with the customer's mobile number from the database.
     + A confirmation SMS is sent back to the customer.
2. **Validation Rules:**
   * The nickname must:
     + Contain only letters and digits (no special characters or spaces).
     + Be between 4 and 10 characters long.
   * The system validates these rules whenever a nickname is created or updated.
3. **Error Handling:**
   * If the customer sends an invalid command or if the nickname does not meet the validation criteria, the system sends an error message back to the customer with instructions on how to correct the input.
4. **Use in Recharge and Bill Payment:**
   * The nickname is stored and can be used by the customer in future transactions, such as mobile recharges and bill payments, to reference their mobile number.

## eReload Reports

This feature introduces a dedicated section in the SalesPoint Admin Portal for generating and downloading eReload transaction reports. Administrators can access detailed reports for every eReload transaction and download them as Excel files.

Using Reporting Engine module in SalesPoint, OO admin team can create the needed report by themselves with assigning the right security group for each report.

**Key Components**

1. **SalesPoint Admin Portal:**
   * User interface where administrators can access the eReload reports section and generate reports.
2. **eReload Reporting Module in SalesPoint:**
   * Generates detailed reports on eReload transactions, including all relevant information such as transaction ID, date, time, dealer, amount, and status.
   * Provides the functionality to download reports in Excel format.
3. **Excel Export Engine in SalesPoint:**
   * Converts the report data into an Excel file, ensuring that all transaction details are included as default.
   * Handles formatting and structuring of the Excel file for easy readability and analysis.

**Functional Flow**

1. **Accessing the eReload Reports Section:**
   * **Navigation:**
     + Administrators log into the SalesPoint Admin Portal and navigate to the dedicated "eReload Reports" section.
   * **Report Options:**
     + The section provides options to filter and generate reports based on various criteria such as date range, dealer, transaction type, and status.
2. **Report Generation:**
   * **Default Report View:**
     + The system displays a default report view with all eReload transaction details, including:
       - Transaction ID
       - Date and Time
       - Dealer/Distributor Name
       - Amount
       - Status (Success/Failure)
       - eWallet Type
       - Any additional metadata relevant to the transaction
   * **Filtering Options:**
     + Administrators can apply filters to narrow down the report to specific transactions or time periods.
3. **Downloading Reports as Excel Files:**
   * **Excel Export:**
     + For any generated report, the administrator can click an "Export to Excel" button to download the report as an Excel file.
4. **Security and Access Control:**
   * **Role-Based Access:**
     + Only authorized administrators have access to the eReload reports section, ensuring that sensitive transaction data is protected.
   * **Data Security:**
     + The system ensures that Excel files are generated without exposing sensitive information like HRN code.

## Push eWallet Transaction to DWH

This feature facilitates the daily transfer of eWallet transaction data from the SalesPoint system to a Data Warehouse (DWH). The data is compiled into a file and securely transferred via SFTP to the DWH for further processing and analysis.

eWallet transaction data shall not include voucher PIN code, it shall contain voucher serial number only.

**Key Components**

1. **SalesPoint Backend:**
   * Compiles and processes eWallet transaction data to be transferred to the DWH.
   * Manages the scheduling and preparation of the data file for daily transfers.
   * Generates the daily eWallet transaction data file in a predefined format (e.g., CSV, JSON).
   * Ensures that the file includes all relevant transaction details required by the DWH.
2. **Data Warehouse (DWH):**
   * Receives and stores the eWallet transaction data for analysis and reporting purposes.
   * Ensures the data integrity and availability for business intelligence operations.
3. **SFTP Server:**
   * Create a dedicated SFTP account with limited privileges and no terminal access.
   * Facilitates the secure transfer of the generated file from the SalesPoint system to the DWH.
   * Ensures that data is transmitted securely and reliably.

**Functional Flow**

1. **Daily Data Compilation:**
   * **Transaction Data Collection:**
     + The SalesPoint Backend collects all eWallet transactions that occurred within the last 24 hours.
     + The data includes transaction details such as:
       - Channel Member Code
       - Channel Member Name
       - Channel Member Type (Dealer, Distributor …etc.)
       - Transaction ID
       - Date and Time
       - Amount
       - eWallet Type
       - Activity (Topup, voucher, bundle …etc.)
       - Source and Destination Wallets
       - Transaction Status (Success/Failure)
   * **File Generation:**
     + The system compiles the collected transaction data into a structured file (CSV) that meets the DWH’s specifications.
2. **Secure File Transfer via SFTP:**
   * **Scheduled Transfer:**
     + The system schedules the transfer of the validated file to the DWH via SFTP on a daily basis, typically during off-peak hours to minimize impact on system performance.

## Other Operators eVoucher Inventory

This feature enables the management of eVoucher inventories for multiple operators (e.g., Friendi, Awasr, Apple, PlayStation) within the SalesPoint system. It includes functionalities for adding operators, uploading eVouchers, configuring denominations, linking operators to specific wallets, and providing role-based access for selecting operators during eVoucher transactions.

**Key Components**

1. **SalesPoint Admin Portal:**
   * Interface where administrators can add operators, upload eVouchers, and configure denominations.
   * Provides the ability to link each operator to a specific wallet.
2. **SalesPoint Core:**
   * Handles the upload, storage, and management of eVouchers for different operators.
   * Manages the configuration of different denominations for each operator.
   * Allows each operator to be linked to a separate wallet, enabling independent management of wallet balances.
3. **Sales App and SalesPoint Web Portal:**
   * Interfaces where users can select an operator during the download or print of eVouchers.
   * Access to operators is controlled by role-based permissions.

**Functional Flow**

1. **Adding Operators:**
   * **Operator Addition:**
     + Administrators use the SalesPoint Admin Portal to add new operators (e.g., Friendi, Awasr, Apple, PlayStation).
     + The system creates an entry for each operator in the database, allowing further configurations.
2. **eVoucher Upload:**
   * **File Upload:**
     + The system provides an option to upload eVoucher files for the newly added operators via the SalesPoint Admin Portal.
     + System shall Ensure below controls are met:
       1. Create a fresh asymmetric keypair for each Operator and share the public key with vendor.
       2. Respective private key should be used on eReload system should first decrypt the file before importing.
       3. Then it should encrypt the individual voucher by a different key (per vendor) before storing into the database
       4. Once the voucher is imported, the file should be deleted securely
       5. System should maintain audit logs for file processing.
     + Uploaded files contain eVoucher details such as serial numbers, HRN, denominations, and expiry dates. System shall validate the file content ensuring strict input validation.
   * **Validation:**
     + The system validates the uploaded file to ensure it meets the required format and structure.
     + Any errors in the file are reported back to the administrator for correction.
3. **Denomination Configuration:**
   * **Configure Denominations:**
     + Administrators can configure different denominations for each operator within the eReload Module in SalesPoint.
     + The system allows for flexible denomination settings tailored to the specific operator's needs.
4. **Wallet Linkage:**
   * **Linking to Wallets:**
     + Each operator can be linked to a specific wallet in SalesPoint System.
     + This linkage allows administrators to manage the wallet balances for each operator separately, ensuring that funds are tracked and controlled independently.
5. **Operator Selection in Sales App and Web Portal:**
   * **Role-Based Access:**
     + The system enforces role-based access control, ensuring that only authorized users can select and manage specific operators during eVoucher transactions.
   * **Operator Selection:**
     + Users can select the appropriate operator when downloading or printing eVouchers via the Sales App or SalesPoint Web Portal.
     + The system presents a list of available operators based on the user’s role and permissions.
6. **eVoucher Transactions:**
   * **Transaction Handling:**
     + When a user selects an operator and performs a transaction (e.g., downloading or printing an eVoucher), the system ensures that the correct wallet is debited and the appropriate operator’s eVoucher is used.
   * **Audit Logging:**
     + All transactions, including the selection of operators and the eVoucher activities, are logged for auditing purposes.

**Security**

* **Role-Based Access Control:**
  + Access to operators and related functionalities is restricted based on user roles, ensuring that only authorized personnel can manage and transact with specific operators.
* **Data Integrity:**
  + The system ensures that all eVoucher uploads, configurations, and transactions are accurately recorded and processed, maintaining the integrity of the operator inventories.
* **Encryption:**
  + eVouchers HRN are encrypted (using AES SHA-512) to protect sensitive information during uploads and transactions.
* **File Upload Validation (This shall be applicable for all file upload in all features):**
  + Only CSV & XLXS extensions are allowed for file upload.
  + System shall Validate the Magic Bytes and Content type Ensure that input validation is applied before validating the extensions.
  + System shall Validate the file type, don't trust the Content-Type header as it can be spoofed
  + System shall Change the filename to something generated by the application
  + System shall Set a filename length limit. Restrict the allowed characters if possible
  + System shall Set a file size limit
  + Only allow authorized users to upload files
  + System shall store them outside of the webroot
  + Protect the file upload from CSRF attacks
  + System shall Validate the file content, by checking each value that it does not contain any special character

## Other Operators SalesPoint Web Portal Access

SalesPoint enables other operators (e.g., Friendi, Awasr) to access the SalesPoint web portal with specific roles and permissions. Operators can manage their eVoucher inventories, monitor transactions, and view related distributor information. Access is secured with username and password authentication, supplemented by OTP-based multi-factor authentication (MFA) that will be sent via SMS.

**Key Components**

1. **User Management System:**
   * Allows administrators to create user accounts for operators to access the SalesPoint web portal.
   * Manages authentication credentials, including username, password, and OTP for MFA.
2. **SalesPoint Web Portal:**
   * The interface through which operator users access their eVoucher inventory, transactions, and distributor information.
3. **Role-Based Access Control (RBAC):**
   * Ensures that operator users can only access functionalities and data related to their specific operator pool.
4. **OTP Multi-Factor Authentication (MFA):**
   * Adds an extra layer of security, requiring users to enter a one-time password (OTP) sent to their registered mobile number or email during login.

**Functional Flow**

1. **User Creation and Access:**
   * **User Account Creation:**
     + Administrators create user accounts for each operator, assigning them specific roles and permissions within the SalesPoint web portal.
   * **Login Process:**
     + Operator users log in using their username and password.
     + The system prompts for an OTP, which is sent to the user’s registered mobile number via SMS.
     + Upon entering the correct OTP, the user gains access to the portal.
2. **Access Roles for Operator Users:**
   * **eVoucher Dashboard (Shared Pool):**
     + Users can view the eVoucher dashboard, but only for their specific operator pool.
     + The dashboard displays eVoucher availability, including details such as denominations, expiry dates, and current stock levels.
   * **Upload eVouchers via Excel/CSV:**
     + Users can upload eVoucher files in Excel or CSV format, adding new eVouchers to their shared pool.
     + The system validates the uploaded files for format correctness and data accuracy before processing.
   * **Distributor Wallet Balance & Transactions:**
     + Users can view the balance and transaction history of wallets associated with their distributors.
     + This includes only the distributors linked to the specific operator’s inventory.
   * **eVoucher Transactions:**
     + Users can monitor eVoucher transactions related to their distributors, providing insight into the usage and distribution of eVouchers within their pool.
   * **eVoucher Inventory Details:**
     + Users can view detailed information about their eVoucher inventory, including serial numbers, expiry dates, values, denominations, and status.
     + Access is restricted to the inventory associated with their operator’s shared pool.
3. **Security and Compliance:**
   * **MFA with OTP:**
     + The login process requires both a password and an OTP, ensuring that access is secure and compliant with best practices.
   * **Role-Based Permissions:**
     + Access to sensitive information and functionalities is restricted based on the user’s role, ensuring that operator users only manage their own data.
4. **Audit Logging:**
   * **Transaction Logging:**
     + All actions taken by operator users, including uploads, transactions, and inventory views, are logged for auditing purposes.
     + Including login activity.
   * **Access Logs:**
     + The system records login attempts and access times, providing a complete audit trail for security reviews.

## POS Device Registration and Management

The POS (Point of Sale) Management Module is a component of the eReload system within the SalesPoint dealer management solution. It provides a platform for registering, updating, and managing POS devices associated with dealers. This module enables administrators to manage POS device information efficiently, including registration, updates and view.

**Key Components**

1. **SalesPoint Admin Portal:**
   * User interface for administrators to manage POS devices, including registration, updates, and view.
2. **POS Device Management Module in SalesPoint:**
   * Handles the registration, updating, and deletion of POS devices.
   * Manages the mapping of POS devices to dealer accounts and other associated information.
   * Stores all POS device information, including POS serial numbers, dealer IDs.

**Functional Flow**

1. **POS Device Registration:**
   * **Interface for Registration:**
     + The system provides an interface within the SalesPoint Admin Portal for registering new POS devices.
   * **Information Entry:**
     + Administrators can input details such as the POS serial number (POS ID), dealer ID, account ID, shop name, main distributor name, and an associated icon.
   * **Data Validation:**
     + The system validates the entered data to ensure that it meets the required format and that all mandatory fields are completed.
   * **Mapping and Storage:**
     + The POS device is mapped to the corresponding dealer and account information and stored securely in the backend database.
2. **POS Device Information Update:**
   * **Editing POS Details:**
     + Administrators can access the POS Management Module to update POS device information, including the serial number, dealer details, and associated metadata.
   * **Change and Save:**
     + Changes are validated and saved in the backend, updating the corresponding records in the database.
3. **Mapping and Storage of POS Device Information:**
   * **Mapping:**
     + Each POS device is mapped to its respective dealer, account, and other related details within the system.
   * **Storage:**
     + All POS device details, including the serial number, dealer ID, and other metadata, are securely stored in the backend database for easy retrieval and management.
4. **Administration and Audit Logging:**
   * **Audit Trail:**
     + All actions related to POS device management (registration, updates, deletions) are logged for auditing purposes.
   * **Access Control:**
     + The system ensures that only authorized administrators can manage POS devices, with role-based access control implemented for security.

## Voucher Card Download and Print

This section already covered in section 4.4 eVoucher, as all of this will be part of Sales App voucher print feature.

## CVM Offers

This feature allows Ooredoo Oman customers to view and subscribe to CVM (Customer Value Management) offers based on their MSISDN through the Sales App. The system integrates with the ESB to retrieve available offers, validate wallet balances in SalesPoint eReload, and handle the subscription process.

**Key Components**

1. **Sales App (Mobile Application):**
   * User interface where customers can access the new "CVM Offer" menu, view available offers, and subscribe.
2. **SalesPoint Backend:**
   * Manages the communication between the Sales App and the ESB.
   * Handles wallet balance validation, transaction processing, and rollback in case of errors.
   * Manages customer wallet balances, handling deductions and rollbacks as required during the subscription process.
   * Sends SMS and push notifications to the user regarding wallet transactions and subscription status.
3. **Enterprise Service Bus (ESB):**
   * Facilitates the integration with external systems, providing APIs RetrieveSpecialOffers and CreateCart to manage offers and transactions.

**Functional Flow**

1. **Accessing CVM Offers:**
   * **New Menu Creation:**
     + A new menu called "CVM Offer" is added under the eReload widget in the Sales App.
   * **Customer Input:**
     + The user is prompted to enter their mobile number (MSISDN).
   * **Request to Backend:**
     + Upon entering the mobile number, the Sales App sends a request to the SalesPoint backend.
2. **Retrieve Special Offers:**
   * **API Call:**
     + The SalesPoint backend calls the RetrieveSpecialOffers API from the ESB using the provided MSISDN.
   * **Offer Retrieval:**
     + The ESB returns a list of offers available to the customer. If no offers are available, the ESB indicates this, and the backend relays the information to the Sales App.
   * **Offer Presentation:**
     + The Sales App displays the offers with details such as:
       - Plan Name
       - Description
       - Validity
       - Price (inclusive of VAT)
       - Offer Date
     + If no offers are available, a message is shown to the user indicating that no offers are available.
3. **Offer Subscription:**
   * **Offer Selection:**
     + The user selects an offer and clicks "Subscribe."
   * **Wallet Validation:**
     + The Sales App sends a subscription request to the backend.
     + The SalesPoint backend validates that the customer's eWallet has sufficient balance to cover the price, including VAT.
   * **Wallet Deduction:**
     + If the balance is sufficient, the backend deducts the offer price from the wallet.
   * **Create Cart API Call:**
     + The backend calls the CreateCart API from the ESB to process the subscription.
     + SalesPoint will validate that submitted offer ID is eligible for the given MSISDN before submitting the request to SubmitCart API to avoid any fraud.
4. **Transaction Handling:**
   * **Successful Subscription:**
     + If the CreateCart API call is successful, the backend confirms the subscription and finalizes the transaction.
   * **Failure and Rollback:**
     + If the CreateCart API call fails, the system initiates a rollback, returning the deducted amount to the customer's eWallet.
   * **Error Messaging:**
     + The Sales App displays a confirmation message upon successful subscription or an error message if the process fails.
5. **User Notifications:**
   * **Transaction Notifications:**
     + The user receives an SMS and a push notification confirming the wallet transaction and subscription status.

**Security**

* **Access Control:**
  + Role-based access controls are implemented to ensure that only authorized users can access and manage CVM offers.
* **Audit Logging:**
  + All transactions, including API calls, wallet deductions, and rollbacks, are logged for auditing and compliance purposes.

**Sequence Diagram**

A white sheet of paper with black text

Description automatically generated

# Service & Integration Design

## Interfaces

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| ID | Provider | Consumer | Nature | Type | Reuse | Description |
| 1 | ESB | SalesPoint | API | REST | Yes | SalesPoint will integrate with ESB to perform all the required customer activities, the below APIs will be used:   1. Refill 2. GetSubscriberBalance 3. RetrieveCustomerSubscriptions 4. RetrieveProductOfferings 5. GetVoucherDetails 6. UpdateVoucherStatus 7. GetBillingInquiry 8. CreatePayment 9. RetrieveSpecialOffers 10. CreateCart |
| 2 | SalesPoint | USSD | API | REST | No | USSD need to integrate with SalesPoint below APIs instead of existing eReload system:   1. GetBalanceEnquiry 2. SelleVoucherCard 3. CustomerRecharge |
| 3 | SalesPoint | ~~SMSC~~ (SNG/SendPoint) | API | REST | No | ~~SMSC~~ SNG/SendPoint need to integrate with SalesPoint below APIs instead of existing eReload system:   1. GetCustomerNickname 2. CreateCustomerNickname 3. UpdateCustomerNickname 4. DeleteCustomerNickname |
| 4 | SalesPoint | ERP | API | REST | No | ~~SMSC~~ ERP need to integrate with SalesPoint below APIs instead of existing eReload system:   1. CreditBalance 2. eVoucherSalesOrder 3. CommissionCredit 4. GetChannelMembers |
| 5 | VMS | SalesPoint | API | XML-RPC | No | SalesPoint shall integrate with VMS system to generate and load eVouchers PINs:   1. GenerateVoucher 2. GetGenerateVoucherTaskInfo 3. LoadVoucherBatchFile 4. LoadVoucherCheck |
| 6 | VMS | SalesPoint | File | SFTP | No | VMS shall load the eVouchers in SFTP folder, and SalesPoint shall read it and install it in SalesPoint database |
| 7 | SalesPoint | DWH | File | SFTP | No | SalesPoint will drop daily transaction file for eWallet transactions in SFTP folder, DWH need to parse it. |
| 8 | SalesPoint | Enhance/Whole-Sellers/3rd Party | API | REST | Yes | SalesPoint will provide below APIs to allow distributors and wholesellers to integrate with the new eReload system:   1. GetBalanceEnquiry 2. CustomerRecharge 3. ChangePIN 4. SelleVoucherCard 5. GetProductInfo 6. ReserveVoucher 7. CommitVoucher 8. DebitWalletBalance 9. CreditWalletBalance 10. GetSubscriptionTypes 11. GetBundleCategory 12. GetBundleCategoryPlanPrepaid 13. SubscribeBundleManagerPlan |

# System Impacts

## System Impacts Summary

### USSD Gateway:

* Need to integrate with new REST APIs from SalesPoint eReload instead of existing eReload system, below the APIs need to be changed:
  + Dealer Balance Enquiry
  + Customer Recharge (topup)
  + Sell eVoucher Card
    1. ESB:
* Refer to Interface Section for a list of the required APIs.
  + 1. DWH:
* Need to parse the CSV file pushed in daily basis mentioned in section 4.13
  + 1. SMSC:

* SMSC needs to integrate with new REST APIs from SalesPoint eReload instead of existing eReload system for customer nickname feature.
  + 1. ERP:
* ERP needs to integrate with new REST APIs from SalesPoint eReload instead of existing eReload system for ERP Sale Order to recharge distributor master wallet.
  + 1. VMS:
* SalesPoint needs to integrate with VMS APIs to generate PINs and load it to SalesPoint system.
  + 1. Distributors / WholeSellers / 3rd Parties:
* Distributors / WholeSellers / 3rd Parties needs to integrate with SalesPoint APIs to perform eReload activities.