Job based architecture

Job management system for gift card operations team

1. Overview

This document defines the design specification for an automated system intended to handle the operations work done by the Ops team in the business of creating prepaid vouchers. The system focuses on automating the process of managing voucher creation, gift card program creation, distribution, onboarding, reconciliation, and operational oversight for different stakeholders, including distributors, B2B customers, and mall principals. The system will leverage a Job Queue architecture, where each job represents a specific action that must be taken as part of the voucher lifecycle. The vouchers can either be purchased for resale or manufactured with banks, specifically for use in malls using the UPI-based eRupi product.

2. Business Context

The voucher business has two main components:

1. **Distribution Model**: Works with large distributors and B2B customers who purchase vouchers for resale or gifting purposes.
2. **Manufacturing Model**: Focused on malls, where vouchers are specifically manufactured using the eRupi mechanism to restrict their usage to certain properties or categories within the mall (e.g., food outlets). The control is implemented by collecting the UPI VPA of all tenants inside the mall, with the banking partners enforcing usage restrictions.

3. System Goals

- Automate the tasks performed by Ops, including voucher creation, gift card program creation, distribution, onboarding, reconciliation, monitoring, and operational compliance.
- Build a scalable, event-driven architecture that supports the dynamic requirements of voucher management.
- Ensure that all system actions are logged, tracked, and automated as jobs to reduce manual intervention and enable efficient workflow.

4. System Components

- **Job Queue System**: The backbone of the solution, providing a mechanism to create, store, and manage different types of jobs related to voucher operations.
- **Job Types**: Each job type represents a unique kind of activity required for the voucher lifecycle. Jobs can be initiated by human agents, other application services, or AI

agents.

- **APIs**: The system will expose multiple APIs, including those for creating jobs, interacting with the Job Queue, and managing integration with other services.
- **Persistence Layer**: A database to store information about vouchers, tenants, UPI VPA details, distributor wallet balances, and job status.
- **Dashboard and Monitoring**: A UI interface for Ops personnel to review current job status, manually create jobs if needed, and monitor operational performance.

5. Functional Requirements

Task	Attributes	Actions
Principal Onboarding	Entity information, UPI VPAs, Pricing, Discounts, Commissions, Marketing Materials	Collect and validate entity details, Onboard principals, Define pricing structure, Set commissions and discounts
Gift Card Program Creation	Program details, Pricing, Discounts, Commissions, Artworks, Email Messages, Terms and Conditions, Marketing Communications	Define program attributes, Set pricing structure, Assign commissions and discounts, Configure marketing details
Distributor Onboarding	Distributor details, Wallet setup, Gift card program associations	Collect distributor information, Configure wallet, Assign relevant gift card programs after the gift card programs are created
Distributor Wallet Management	Distributor ID, Deposit verification, Wallet balance update	Verify bank deposits, Update distributor wallet balance
Reconciliation	Voucher records from system and principal portals	Daily reconciliation of voucher data, Identify discrepancies, Generate reconciliation reports
Fund Transfer to Principals	Principal ID, Bank details, Transfer amount	Verify payment requirements, Initiate fund transfers, Confirm receipt of funds in bank
Commission and Discount Calculation	Gift card program rules, Distributor details	Calculate commissions and discounts, Generate invoices, Track payments
Monitoring and Alerts	Job type, Job state, Exception conditions	Monitor job progress, Trigger alerts for manual intervention, Log failures for auditing

1. **Job Queue Definition**

- The system should maintain a **Job Queue** capable of holding different types of jobs.
- Jobs can be assigned one or more of the following attributes:
  - **Type** (e.g., Gift Card Program Creation, Voucher Distribution, Payment Validation, Principal Onboarding, Distributor Onboarding, Reconciliation).
  - **Priority** (e.g., Low, Medium, High).
  - **Expiry Date**: A job may have an expiry period, beyond which no action is required.
  - **State**: Created, In Progress, Completed, Failed, or Expired.

2. **Job Creation and Lifecycle**
   - Jobs can be created by:
     - **Human Agents**: Via a dashboard to create new jobs manually for special scenarios.
     - **Application Services**: Automatically created by the voucher creation/distribution system.
     - **AI Agents**: AI-based decision-making can create jobs based on operational rules (e.g., replenishment requirements).
   - A **Job Creation API** should be available to allow programmatic creation of jobs.
   - Jobs should be **tracked** and must follow a **state machine** for lifecycle management.
3. **Job Processing**
   - Each job type requires specific **attributes** that will serve as inputs.
   - Jobs will be processed by executing specific actions:
     - Example: **Gift Card Program Creation Job** will require program details, pricing, discounts, commissions, and marketing communication details.
     - Example: ** - Example: **Principal Onboarding Job** will require entity information, UPI VPAs for tenants, pricing, discounts, commissions, and marketing materials.
     - Example: **Distributor Onboarding Job** will require distributor details, wallet setup, and gift card program associations.
     - Integration with **System APIs** to perform actions such as notifying banks, onboarding principals, or updating voucher inventory.
   - **Job Queue API** should be available to monitor and update the job state programmatically.
4. **Job Expiry and Retention**
   - Jobs may have an **expiry time**, after which no action is needed.
   - Expired jobs should be automatically marked as **Expired** and archived for auditing purposes.
5. **Integration with Banking Partners**
   - Ensure voucher restrictions using tenant **UPI VPA** lists are enforced by integrating with **banking partners**.

- The system should send relevant voucher data to banks to restrict payment approvals outside mall properties.
- Ops must **verify distributor deposits** into bank accounts before setting wallet balances.
- **Transfer funds to principals** and ensure that funds are deposited into banks to initiate voucher manufacturing.

6.

- Include **approval workflows** where necessary, such as for newly onboarded mall tenants or external principals.
- **Reconciliation Mechanism**: Regularly reconcile vouchers created via API against records from principal portals to identify anomalies such as missing vouchers or mismatched details.

7. **Distributor Management**
   - **Distributor Onboarding**: Allow Ops to onboard new distributors, defining which gift card programs are available to them.
   - **Wallet Management**: Track distributor wallet balances and ensure that distributors deposit funds before setting balances.
   - **Program Assignment**: Assign gift card programs to distributors based on agreements and availability.
8. **Commission and Discount Management**
   - Calculate **commissions** and **discounts** for each gift card program based on pre-defined rules.
   - Generate **invoices** for commissions owed by distributors and track **payments**.
9. **Monitoring and Notifications**
   - **Dashboard for Monitoring**: A user-friendly dashboard to display the current status of all jobs in the system.
   - **Alerts and Notifications**: Real-time notifications for jobs that require manual intervention or have failed.
   - **Reconciliation Alerts**: Alerts for any discrepancies found during reconciliation between principal portals and voucher records.

6. Non-Functional Requirements

- **Scalability**: The system should handle an increasing number of jobs, especially during peak business periods.
- **Security**: Protect sensitive information such as UPI VPAs using encryption, and ensure secure API interactions.
- **Reliability**: Ensure high availability for job queue processing to prevent disruption in voucher creation.

- **Auditing**: Maintain logs for all actions performed, particularly around voucher creation, modifications, payments, and reconciliation.

7. APIs Specification

- **Job Creation API**
  - Method: `POST`
  - Endpoint: `/api/job/create`
  - Payload:

```
{
        "attributes": {
    "voucher_details": { /* voucher-specific attributes */ },
    "customer_id": "string",
    "tenant_vpa_list": ["vpa1", "vpa2"]
  },
  "priority": "High"
}
```

- **Job Queue API**
  - Method: `GET`
  - Endpoint: `/api/job/queue`
  - Response:

```
[
      ]
```

8. Technical Architecture

- **Job : ** Job will be central construct. Job is an interface. Different types of jobs will be implementation. Job will involve series of actions. Action is another interface which almost always will tie to Rest API.
- **Job Queue Implementation**: Job queues will be built using database table.
- **Database**: A relational database (**PostgreSQL**) to job job metadata.
- **Cloud Integration**: Will be deployed on AWS. But can also work as a standalone system on laptop.

9. Future Considerations

- **AI Agent for Smart Decisions**: Integrate AI models to Identify and create jobs in Job queue
- **Analytics and Reporting**: Add a reporting layer for understanding trends in voucher usage, distributor performance, and customer insights.

- **10. Summary**

The system will help automate the current manual processes in managing voucher operations, focusing on a Job Queue architecture that will allow efficient, scalable, and flexible operations management. The automation will improve turnaround time, reduce the operational load on human agents, and ensure accurate enforcement of restrictions on the usage of vouchers.