# Design Documentation
# for
# Product Information Services

## Table of Contents

## Introduction

This document describes the design and implementation of product information system which publishes product information updates.

## Title

Scalable and Resilient Event Based Message Exchange for Real-Time Product Information Updates.

## Objective

Implement a robust and scalable system for exchanging event messages that updates product information promptly. Ensure these updates reach downstream consuming systems promptly, considering the global distribution of various apps across a multi-hybrid cloud environment. A simple visualisation/dashboard about product information updates would be nice to have. Follow best practices in code maintenance for releasing features through testing in production and deploy the application in cloud-native infrastructure. System Design Architecture is important.

## Assumptions

Source Data Provider

- Product Information System: Assume it is running in On-premises datacenters.

Downstream Systems

- Market-Based Product Promotion System for US: Assume it is running in Microsoft Azure.  (list of products, category, features, target group, promotional channel, promotional message, imageUrls)
- Customer Portal for Mobile Application System for EU: Assume it is running in Google Cloud Platform (GCP).

Sample Events to Handle

- Product Price Update
- Country
- Specific Product Discount Update

## Design Considerations

- Ensure eventual data consistency.
- Address geographical distribution and latency challenges.
- Strategy for safe feature release.
- Auto-scale based on compute capacity or processing backlog.
- Cost efficiency.

# Defining requirements

## Functional Requirements

1. Provide product details to product information system
2. Receive product details in customer portal from product information system
3. Publish changes from product information system to channels like customer portal and promotion system like
    - Product Price Update
    - Country – product available/unavailable for country
    - Specific Product Discount Update

## Non-Functional Requirements

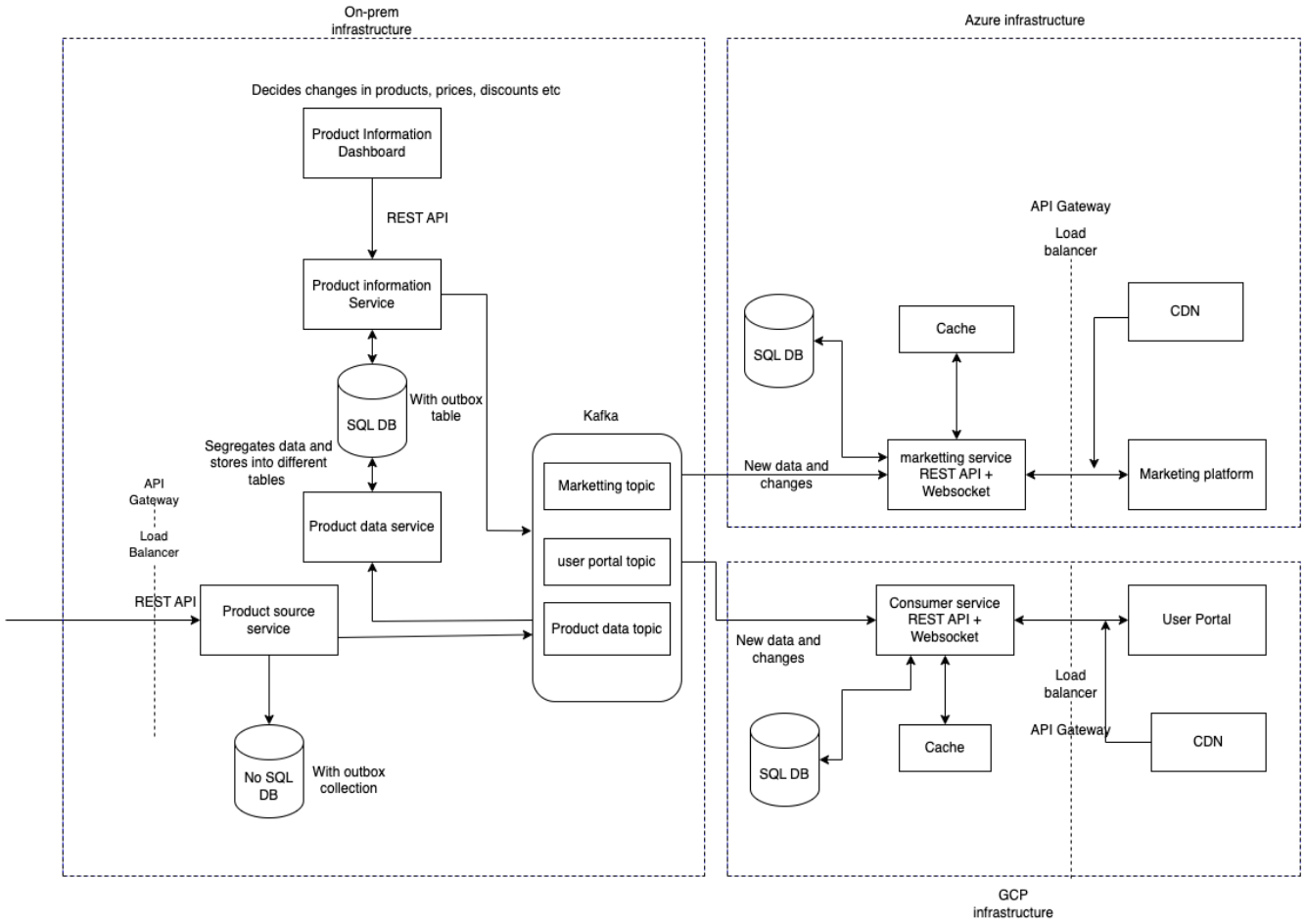
Low latency

High availability

Code quality

Portability – containerization and deployment

# System Design



## Services

1. **Product source service**

Provides the APIs to get the product details from various sources.

Receives the unstructured data and store it into MongoDB. This service also pushes the data into Kafka which will be then handled by product data service.

Adapting outbox pattern for eventual consistency.

2. **Data segregation Service**

Receives product data from Kafka and converts it into structured data and saves it into PostgresDB.

3. **Product Information Service – with dashboard**

Provides the APIs for modifying products, promotions and publishing changes to respective Kafka topics.

Adapting outbox pattern for eventual consistency.

4. **User portal service**

Provides APIs to get products details to user.

It receives product information from product information service via Kafka and stores it in PostgresDB. User portal receives product information through API call and messages from service through websocket.

5. **Marketing service**

Provides APIs to get promotion data. It receives promotion information from product information service via Kafka and stores it in PostgresDB.

## Dashboard and User interfaces

1. **Product information dashboard**

   User interface to change and publish product and promotion information

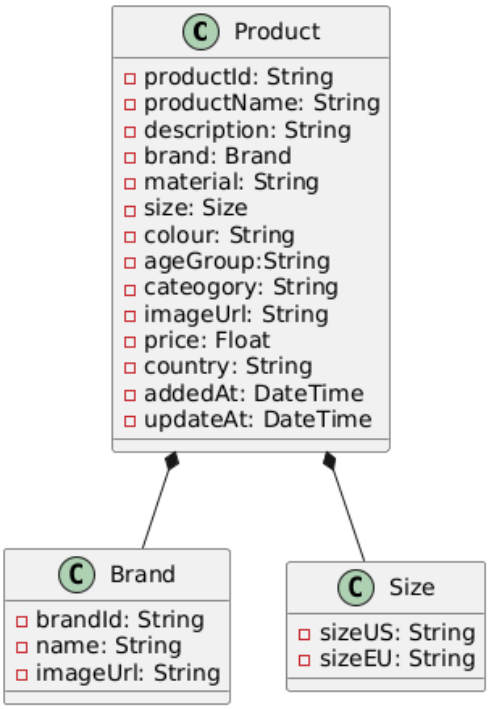
2. **User portal App**

   For customer to access product information

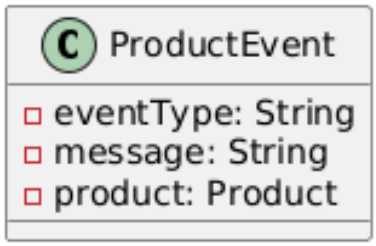
3. **Promotion portal**

   User interface to access promotion information

## Data Models


Product
productId: String
productName: String
description: String
brand: Brand
material: String
size: Size
colour: String
ageGroup:String
cateogory: String
imageUrl: String
price: Float
country: String
addedAt: DateTime
updateAt: DateTime
Brand
brandId: String
name: String
imageUrl: String
Size
sizeUS: String
sizeEU: String


PromotionData
productList: List<Product>
category: String
targetGroup: String
promotionalChannel: String
promotionalMessage: String
imageUrls: List<String>


ProductEvent
eventType: String
message: String
product: Product


ProductEvent
eventType: String
message: String
promotion: Promotion

## API Documentation

Documentation of APIs with OpenAPI and Yaml

## Testing

A test project with test cases for functional testing with the help of Junit testing framework.

## Code Quality

Analyse projects for code quality using SonarQube