Contents

1. Problem statement

Challenge

- It's well known that ice cream shops like to push sales on customers. Our AI startup is developing a powerful way to identify the top 3 favorite ice cream flavors by analyzing a customer picture for a few milliseconds. A big ice cream chain could be interested in using our product to introduce cross-selling.
- First, we need you to design a secure public API to allow our clients to upload a frame from the stream of the camera included in the register and receive the top flavours in the response. The AI service was designed by you last year, and we assume that you remember how it works. According to the documentation, this service outputs the hypothesis containing the top flavours.
- The client-side software is not developed yet; your architecture will have to determine its specifications.

Objectives

- Design a high-level architecture of the public API that includes a map of the services, layers inside them, and main flows;
- Create a GIT repository with a docker-compose.yml that builds and runs the fake services needed in your architecture;

Presentation (15 minutes maximum).

Expectations

- A straightforward high-level design with relevant flows and components;
- Clear verbal explanation;
- A running docker-compose file;

Presentation quickly after instructions were sent (a week).

2. Problem understanding

Business Drivers

a) Increase ice-cream sales by customer personalization and recommendation.
b) Recommendation product can be used by different ice cream chains. Multi-tenancy and data separation must be considered in design.
c) Customer must be Authenticated and Authorized to upload picture to receive recommendation. However, company does not want overhead of maintaining and protecting customer data, as well as does not want customer to go through registration process.
d) Business will grow and customer base will expand, solution must scale, operational cost must remain low, runtime cost must be shared-model (multi-tenancy).

Key challenges

- Security
    a. Authentication and Authorization of customer
    b. One time customer Authorization to upload image

- Data
    a. Data type - Pictures are bytes/large-object.
    b. Storage must scale and availability must be high across multiple Availability zone.
    c. Data management and retention - storage size will increase very fast. Data retention policy has to be in place.
    d. Data versioning – customer can upload multiple pictures; data versioning will be required to avoid overwriting as well as keep versions of picture. Data versioning is required because recommendation depends on customer picture, and will be used to analyze correctness of recommendation.

Customer experience impact

a) Uploading customer picture requires large band width and data utilization. Customer might not like to upload picture, since data plans/utilizations are charged.
b) Recommendation must appear as real time, else customer may not pay attention to recommendation.

Product success and effectiveness

a) Correlation/relation between flavour recommendations and purchase.
    Example – Recommendation offered by flavour rating and flavour purchase.
    This will measure effectiveness of overall recommendation product and customer experience.
b) Correlation/relation between customer picture and recommendation.
    Example – Customer uploaded picture and recommended flavour rating.
    This will measure effectiveness of recommendation engine.

3. Proposed Technology

Security and Data transfer

a) Authenticate user using OpenID connect (OIDC). OpenID connect is on top of OAuth2.0 and allows exchange User data such as user-name, email and other public data of user.

   Use social media such as Google, Facebook, Instagram support such integration. Users trust such social media and will find it easy to Authenticate and share data through these trusted social media.

b) Use Signed URL to explicitly Authorize user to upload picture. This URL will have expiry and user can use only once.

   Signed URL will have other data such as User identification (example email Id), request-ID (random number uniquely identifying upload and request).

c) Use CDN (Content delivery network) such as AWS-Cloud-front, to get nearest touch point (LTM/VIP) to user, so that image can be uploaded quickly with minimum data/network-hops.

   CDN also provides geographic location of user, geographic location can be used for performance optimization such as data-locality, data-shard etc.

Data storage

a) Use Object storage such as AWS S3. Object storage are good fit for large/unstructured data storage. Such storage systems offer high throughput using multipart upload/download.

b) Object storage such as AWS S3 have built in mechanism for Signed URL validations, CORS (Cross origins scripts).

c) Cloud storage such as AWS S3 provides data-versioning, data-retention life cycle, cost optimization and on demand scaling.

   Example – For cost optimization you can move image files from hot-S3 to Cold/glacier S3.

d) Cloud storage such as AWS S3 are managed services, and provides high availability across availability zones (replication) at same time restricts data sharing across regions (data protection).

Since customer picture will be used by recommendation system, S3 is good option to integrate customer-picture with recommendation system. S3 takes care of replication and offer very high throughput using multipart upload/download.

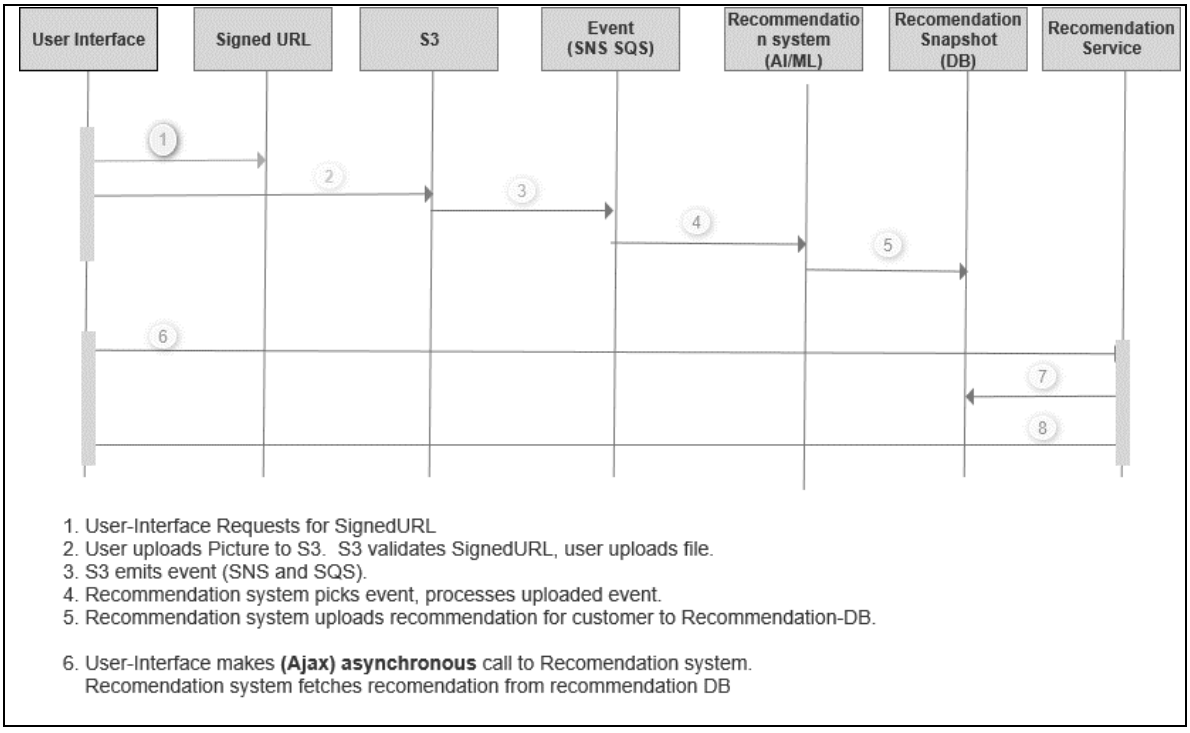
4. Integration points and Flow Sequence

Design principles

- Asynchronous communication (Event based), if possible.
- No hotspot or intense compute or intense network or intense storage activities in User Experience path.
- Services and components must be isolated, elastic, distributed and independent.

Integration points and sequence

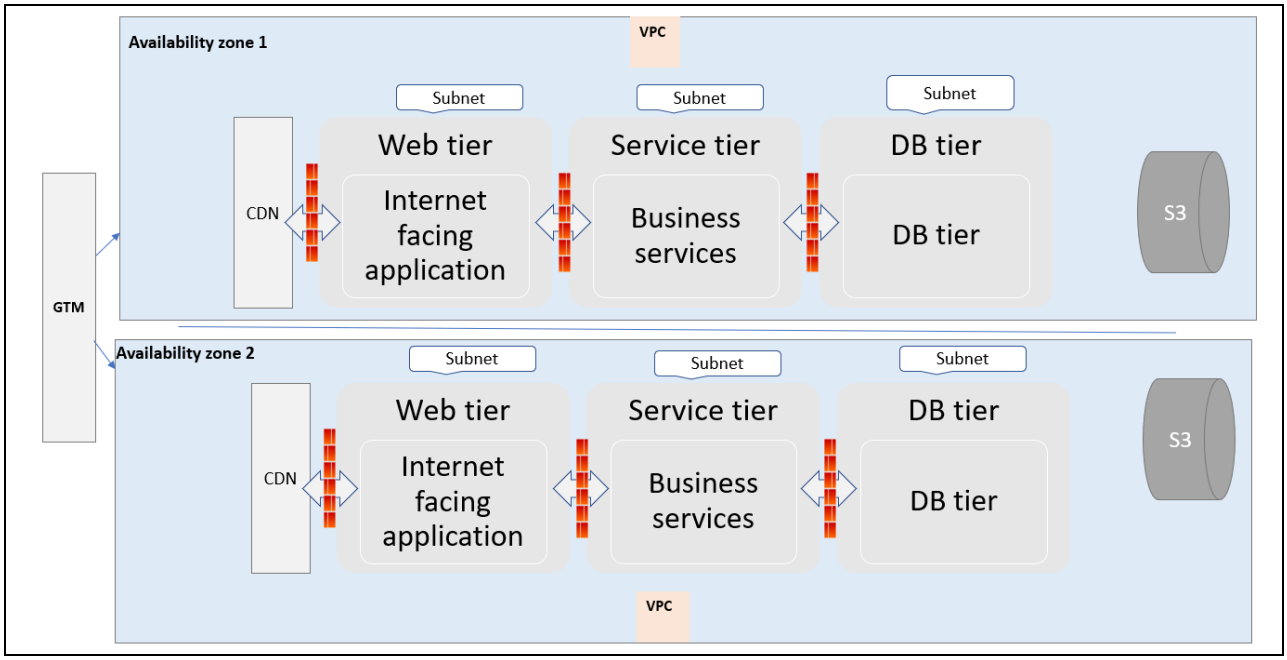
a) User Interface
b) Signed URL
c) Storage
d) Events
e) Recommendation system



1. User-Interface Requests for SignedURL
2. User uploads Picture to S3. S3 validates SignedURL, user uploads file.
3. S3 emits event (SNS and SQS).
4. Recommendation system picks event, processes uploaded event.
5. Recommendation system uploads recommendation for customer to Recommendation-DB.

6. User-Interface makes **(Ajax) asynchronous** call to Recomendation system.
   Recomendation system fetches recomendation from recommendation DB

5. Network diagram



Note:

- GTM – Global traffic manager
- CDN – Content delivery network
- VPC – Virtual private cloud (Virtual network)

6. Application Logical design

Services and purpose

Service	Purpose	Unit of task	Input	Output
authenticate()	This service integrates with social platforms such as Google, Facebook, Instagram for user authentication and user identification	Forward user for OAuth2.0 Authentication with partner IDP. Parse JWT token - user identification	User Confirmation to Authenticate with partner IDP	User Identification and Authentication
getSignedURL()	This Service provides signed URL. Signed URL will be used by customer to upload picture.	Service takes JWT token as input parameter. Validate JWT Extracts customer emailId, requestId Generates SignedURL (one time token)	Authenticated user (JWT token)	SignedURL
getRecommendation()	This service provides recommendation	Recommend ice-cream flavor, with ranking. Service fetches recommendation from recommendation DB based on userId/userEmailId.	User details (userId, emailId)	List of icecream recommendations, with ranking.
generateRecommendation()	This service computes recommendation from customer picture	Subscribe to S3 events. Generate recommendations based on customer uploaded picture. Update recommendation DB, for recommendation consumption by recommendation service.	Event from S3 Picture User emailId, requestId	Updates recommendation database.