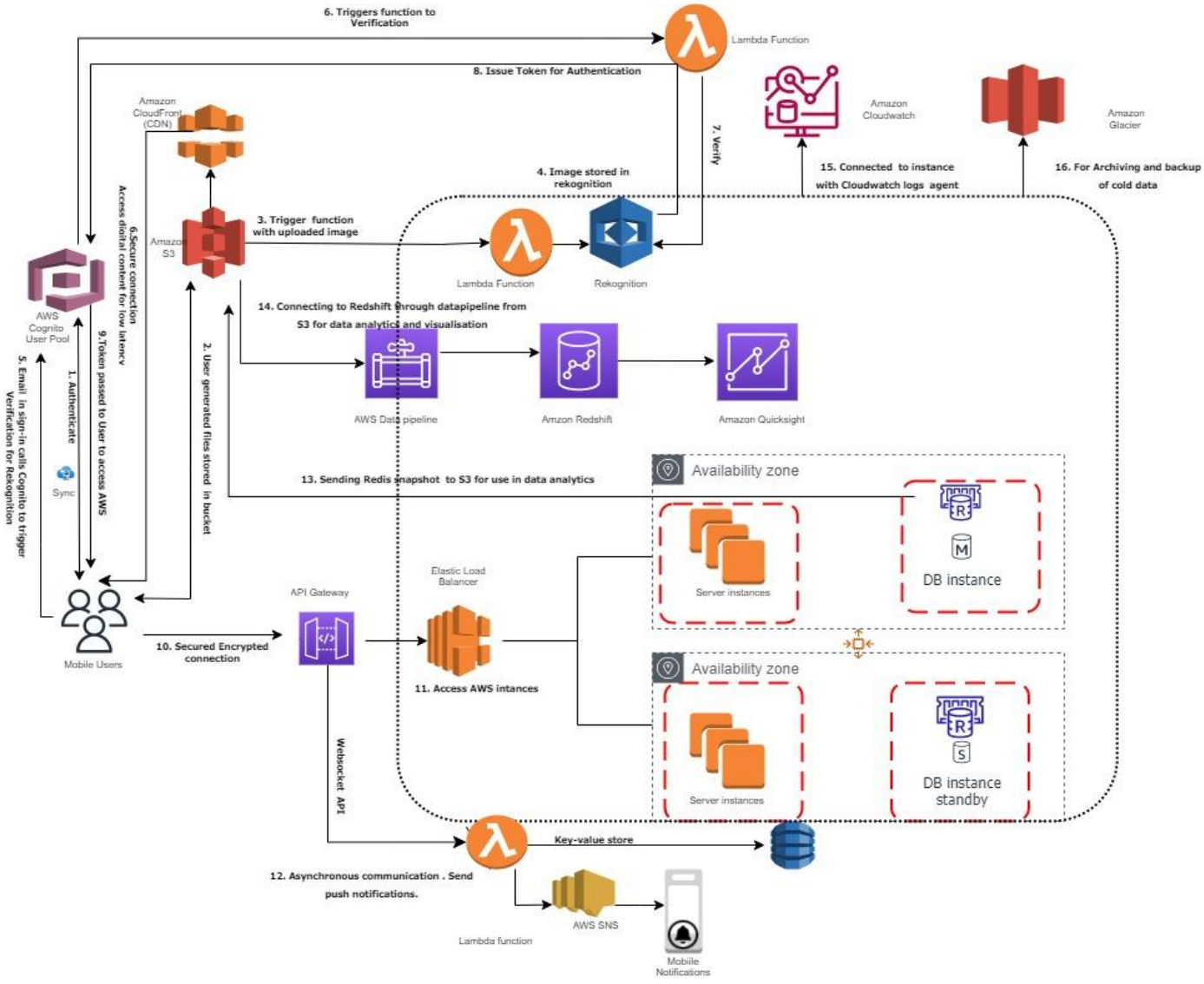
Introduction:

Architectural design and solution for a social media app specifically made for women. The app can be used as an audio-only chat and content creator application.

Solution Architecture:

The proposed mobile-based architecture is based on AWS framework and its services. It shows the RESTful backend that uses AWS-managed services for backend resource requirements. The diagram below outlines the services used and the flow. The few requirements which are included in the app along with different features for scalability, high performance, load distribution, high throughput and replication are -

- User Registration, Verification and Authentication.
- Individual(per-to-peer) and group messaging of audio content.
- Push notifications.
- Storage for later usage for analytics and monitoring.



Solution Flow:

1. We make use of **Amazon Cognito** which offers Identity Management, Authentication,Verification and Data synchronization across devices.
   The user generated files are stored in **Amazon S3,** an highly available and durable storage and then passed onto the **Amazon Rekognition** service via a **Lambda function**. Amazon Rekognition is used for verification of the clients to be Women only ,through facial recognition features (probably using a selfie pic during sign up) ,as it's a woman only App. Once the user has received the token after verification it can access the AWS services in an encrypted manner**.**We also make use of **Amazon CloudFront** to access digital assets for low latency content delivery**. (flow operation 1-9 in the architecture**)

2. In applications where mobile clients need to communicate over a channel, implementation is mostly using **Websockets**. Websockets are also used to asynchronous updates from server to client as well. Once the connection is secured and we get access to the AWS. We used **AWS API gateway** to leverage **AWS websockets API** for bidirectional communication between client and servers. Mobile users send requests to API Gateway to access the application and data. **.(flow operation 10 in the architecture**)

3. There is an Amazon **Elastic load balancer** for effective load distribution across availability zones and server instances. Each availability zone is equipped with replicas for the instances holding the application logic for users and their communication. It also consists of **Amazon RDS** for data storage and **Amazon Elastic cache for Redis** for in-memory cache and message broker service to transfer data between services. Also, **auto-scaling groups** are present for instances. **.(flow operation 11 in the architecture**)

4. When a user gets connected, a **lambda function** will save the connection id and other user data to a **Amazon Dynamo DB** table. It then formats each communication request and sends a push notification to specific user via **Amazon SNS .(flow operation 12 in the architecture**)

5. The data snapshots from **Redis** as stored in **S3 bucket** for later analysis **.(flow operation 13 in the architecture**)

6. **AWS Redshift** is used as a data warehouse to analyse structured and unstructured data across databases etc**.(flow operation 14 in the architecture**).

7. **Amazon Cloudwatch** is connected to instance logs for monitoring and then auto-scaling.**(flow operation 15 in the architecture**).

8. Archiving can be done through **Amazon Glacier.(flow operation 16 in the architecture**)

Addressing the requirements/concerns for the start-up:

Scaling to meet the demand, but with uncertainty around when and how much this demand will be – they are very concerned about buying too much infrastructure too soon or not enough too late!

- AWS API Gateway Websocket API is used for communication which can be scaled as per the need .

- Instances of the EC2 , Amazon RDS, Elastic cache redis, Lambda all can be auto scaled as per the need.

Manage user identities & sync user specific data across multiple devices

- AWS Cognito provides scalable, Identity management , MFA along with synchronisation capabilities across multiple devices.

Ability for Service Providers to send notifications to consumer

- AWS SNS service provides us with push notification features required.

Ability to run analytics on top of collected data, with analytics they should be able to visualize & understand app data usage.

- Using AWS Redshift along with AWS QuickSight gives us good analysis and visualisation dashboard for understanting

Their ability to configure their database and data access layer for high performance and throughput

- Amazon RDS is used as a managed database server which can be increased in storage and scaled for high performance and throughput.

Effective distribution of load

- Load distribution is effectively done by the Elastic Load balancers in conjunction API Gateway for balancing out network traffic which is useful in microservice architecture.

A self-healing infrastructure that recovers from failed service instances

- AWS CloudFormation stacks or AWS Opworks stack can be used to recover instances.

Security of data at rest and in transit

- PII (Personal Identifiable Information eg phone number, address, email, National ID), Data in Amazon RDS , EBS are encrypted at rest. Passwords will be stored as Hash.
  Encryption can be provided by common methods like AES etc.
- HTTP and Web socket communication is secured with TLS for data in transit.
- Separate VPCs to segregate envs (Dev, Test, Prod) and CI CD pipelines for each env.

Securing access to the environment as the delivery team expands

- User IAM roles , groups , MFA are used .
- Security groups are created.
- Only API gateway is given as entry point to the system.

Ability to easily manage and replicate multiple environments based on their blueprint architecture

- Application to be containerised and orchestrated in order for easy management and replication . Elastic Beanstalk can be used .

Elimination of system downtime and Disaster recovery.

- Blue-Green deployment could be used to eliminate or reduce downtime.
- Disaster discovery can be achieved by cross-region backups.

Assumptions:

- We will be using AWS as the cloud provider as its easy to use, scalable ,flexible ,reliable and cost effective .Also it covers 60% of the market for cloud providers.
- Only Audio chats are being uploaded. Audio can be of any language.
- Voice to text conversion , translation to common language and analysis on these will be taken care off in future releases .