Cloud Computing Architecture - Assignment 3

Serverless/Event-driven Architectural Design Report

Nguyen Gia Binh - 104219428

Student

Swinburne University of Technology

Hanoi, Vietnam

[104219428@student.swin.edu.au](mailto:104219428@student.swin.edu.au)

Bui Viet Hoang - 104060260

Student

Swinburne University of Technology

Hanoi, Vietnam

[104060260@student.swin.edu.au](mailto:104060260@student.swin.edu.au)

Pham Hung Manh - 104169507

Student

Swinburne University of Technology

Hanoi, Vietnam

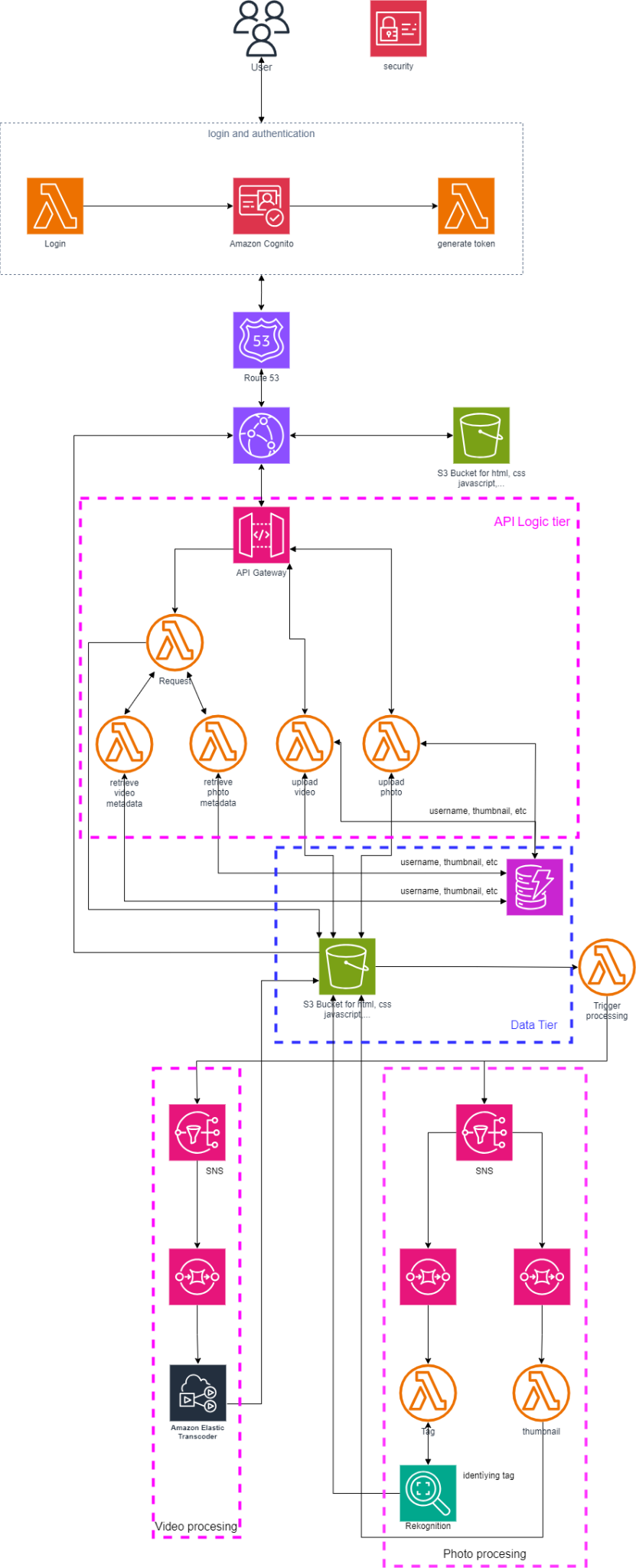
104169507@student.swin.edu.au

**Abstract - The Photo Album application, developed previously, met with amazing success. Since then, it has been observed that the demand will grow tremendously in the future, doubling every 6 months recently. Furthermore, problems regarding speed were also mentioned, along with additional changes the company wants to make. Within this report, a serverless cloud framework is introduced, and strategically developed to cater to the demands of the Photo Album application. The design not only ensures the sustained operation of the website but also aligns with essential design criteria, encompassing performance, reliability, scalability, security, and cost-effectiveness.**

I) INTRODUCTION

This document outlines a serverless cloud architecture tailored to the Photo Album application, effectively addressing various requirements. These include transitioning to managed cloud services, accommodating fluctuating demand, embracing a serverless approach, replacing the sluggish and costly relational database, optimizing global response times, and efficiently managing video and photo uploads.

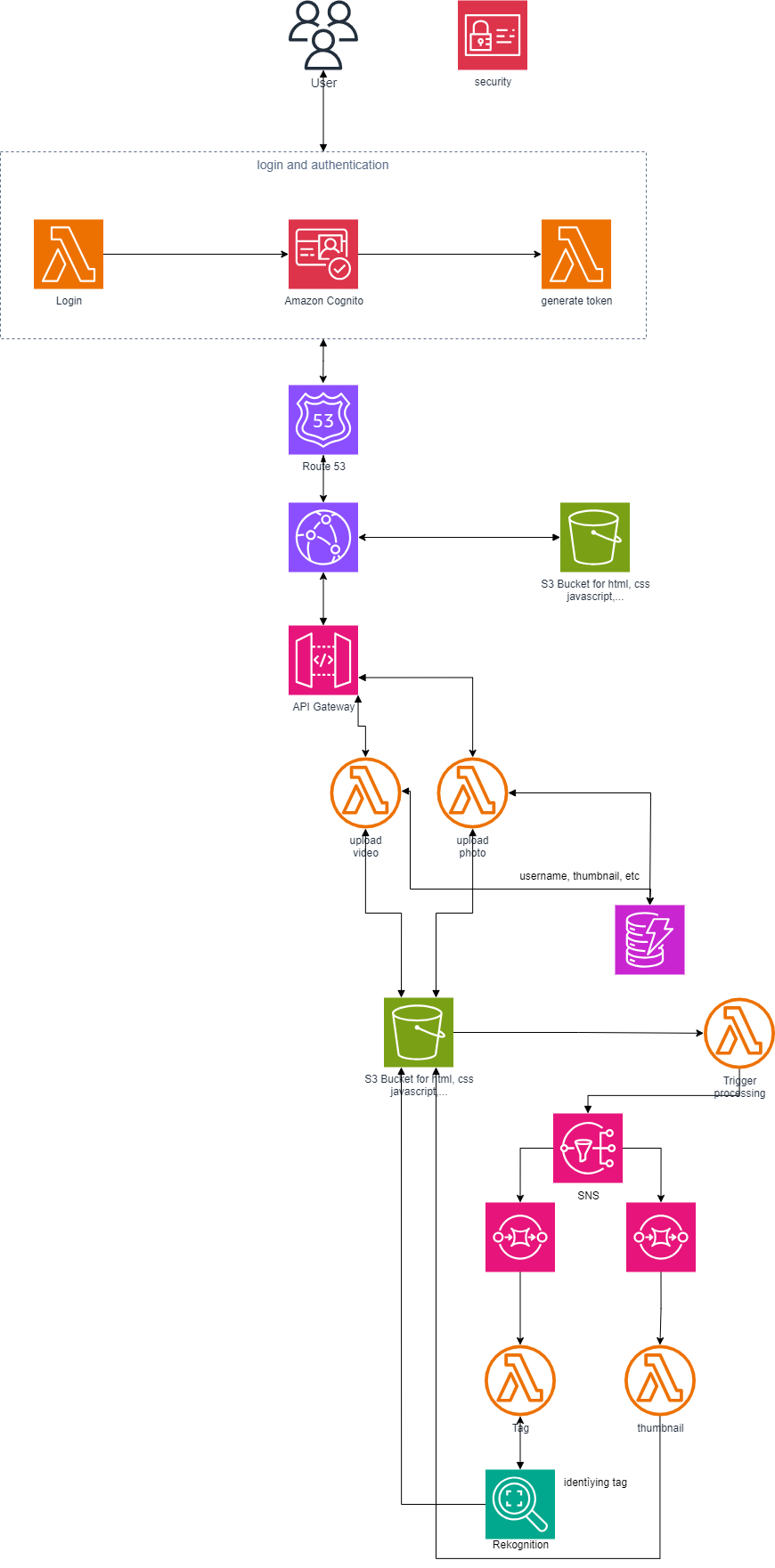
AWS's service autonomy stands as a key advantage, empowering us to adopt a serverless/event-driven approach. This methodology enhances the flexibility and efficiency of our web operations. By eliminating the burden of server management, serverless computing allows us to focus on app development and implementation, freeing us from the intricacies of infrastructure administration. Additionally, event-driven architecture ensures that actions are triggered in response to specific events, fostering a more adaptable and scalable system.

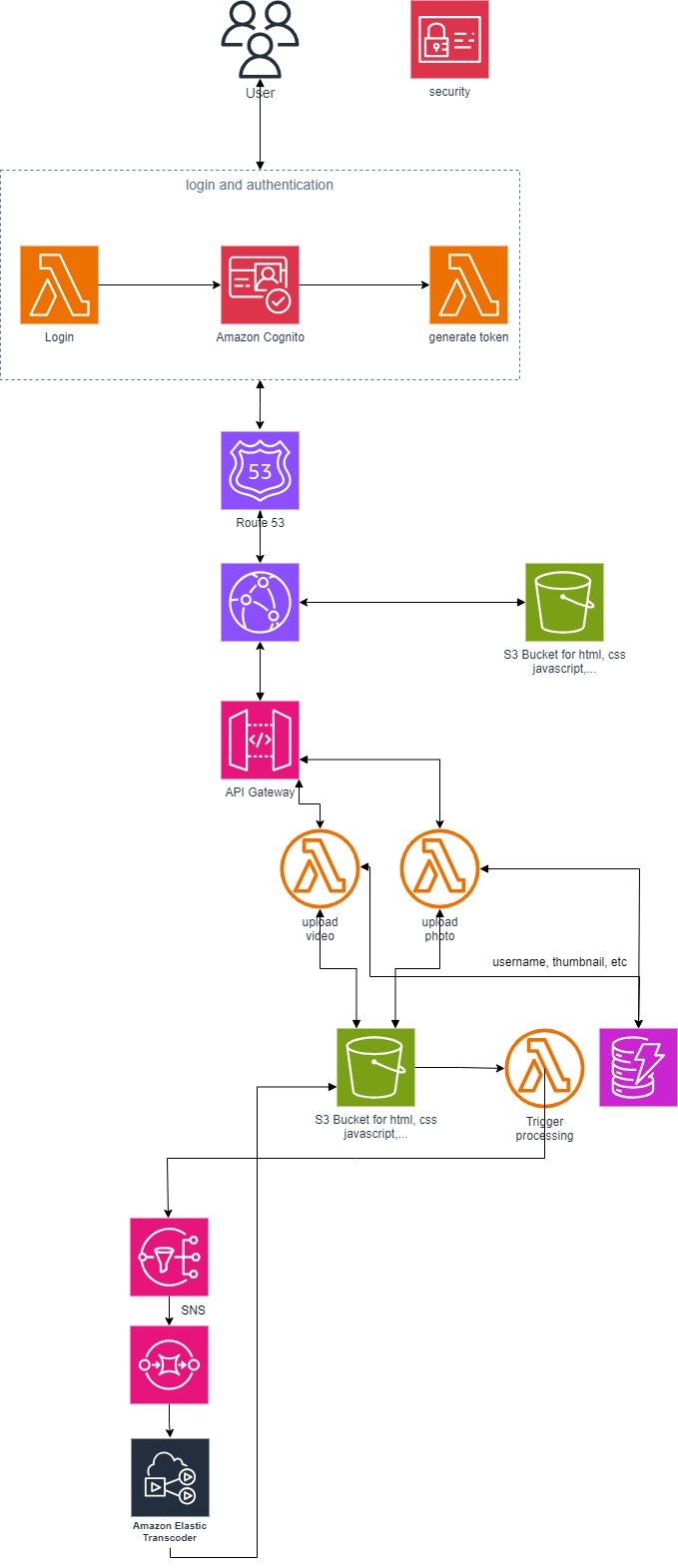
II) ARCHITECTURAL DIAGRAM

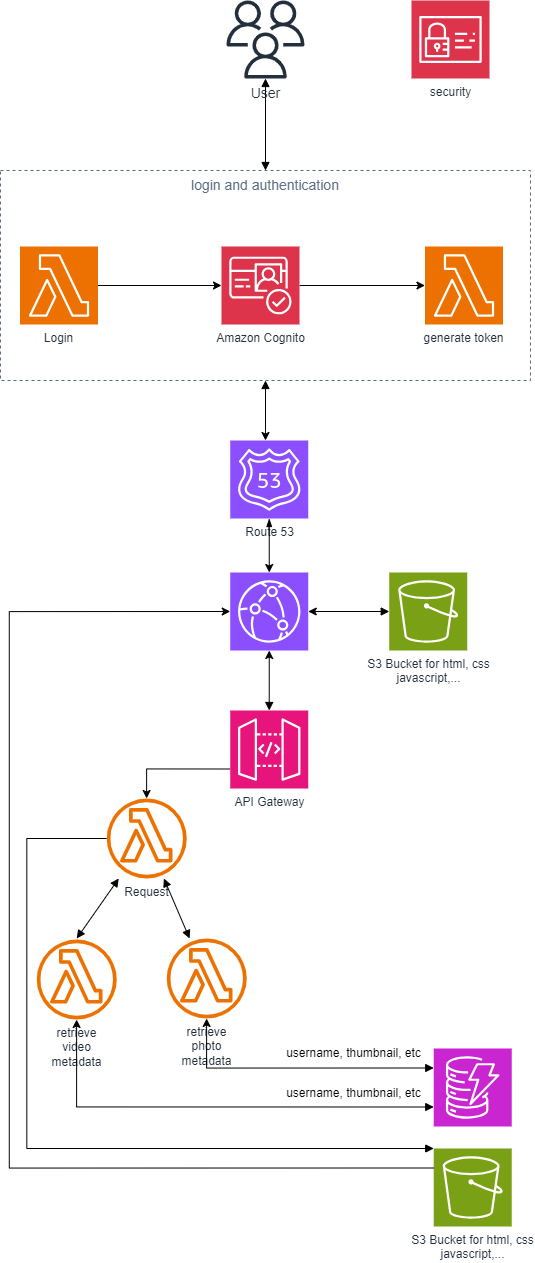


The architectural diagram shown above is the final architectural solution, as well as all AWS services used for this project.

III) UML COLLABORATION DIAGRAMS

Uploading Photo

Uploading Video

Retrieving Photo/video



IV) AWS SERVICES USED

* **Amazon Simple Notification Service (Amazon SNS)**:

*Service description*: a managed service that provides message delivery from publishers to subscribers (also known as producers and consumers). Publishers communicate asynchronously with subscribers by sending messages to a topic, which is a logical access point and communication channel.

*Purpose:* The application's photo and other media processing pipelines can be decoupled via the SNS service. SNS connects many application components, promoting efficient information sharing and communication. It provides feedback to different system components on detection, real-time alarms, and the state of media processing. It notifies users of newly tagged media, improving both the user's experience and the service's functionality.

* **Amazon Simple Queue Service (Amazon SQS):**

*Service description*: a fully managed message queuing service that makes it easy to decouple and scale microservices, distributed systems, and serverless applications. Amazon SQS moves data between distributed application components and helps you decouple these components.

*Purpose:* SNS is linked to SQS in order to decouple the application's pipelines for processing photos and videos. Jobs are queued up in SQS, and nodes can take them out to process at their own speed, avoiding overload and keeping queued jobs stored, increasing their reliability.

* **Amazon Rekognition:**

*Service description*: a deep learning-powered image recognition service that detects objects, scenes, and faces; extracts text; recognizes celebrities; and identifies inappropriate content in images. It also allows you to search and compare faces.

*Purpose:* Rekognition can identify tags from uploaded photos using machine learning. We don't need to create a custom model, eliminating the cost of making, training, and deploying said custom model. Plus, Rekognition is ready as is. Furthermore, it can identify faces so there are further improvements that can be made

* **Amazon Elemental Media Converter:**

*Service description:* a file-based video processing service that formats and compresses offline content for delivery to televisions or connected devices. With AWS Elemental MediaConvert's high-quality video transcoding, you can create on-demand video assets for playback on virtually any device.

*Purpose:* Making uploaded video to be usable across devices and platforms

* **Amazon Simple Storage Service:**

*Service description:* Amazon Simple Storage Service (S3) is a leading cloud-based storage solution by Amazon Web Services, offering secure, scalable, and highly durable object storage through virtual containers called "buckets." It allows users to effortlessly store, retrieve, and manage various data types, ensuring 99.999999999% durability through redundant storage across multiple locations within chosen regions. S3 prioritizes security with encryption, access control, and IAM integration, while its diverse storage classes cater to different needs, optimizing costs for varying usage patterns. Its versatility makes it a pivotal component for applications, enabling seamless data management, backups, content delivery, and analytics for individuals, startups, and enterprises.

* **Amazon DynamoDB:**

*Service description:*  Amazon DynamoDB is a fully managed NoSQL database service provided by Amazon Web Services (AWS), offering seamless scalability, high performance, and low latency for applications requiring flexible, fast, and reliable data storage. It operates on a key-value and document-oriented model, enabling users to store and retrieve structured data with lightning-fast performance. DynamoDB automatically handles hardware provisioning, setup, maintenance, and scaling, allowing developers to focus solely on building their applications without worrying about infrastructure management. Its scalability is unparalleled, adjusting capacity to accommodate any level of throughput, and it ensures consistent single-digit millisecond latency for data access. With features like global tables for multi-region deployments, encryption at rest and in transit, and flexible querying capabilities, DynamoDB is a go-to choice for applications demanding high availability, reliability, and seamless scalability for their data needs.

V) DESIGN RATIONALE

1. Business scenarios fulfilled:

**A:** Minimize the need for in-house systems administration

The foundation of our architectural approach is the utilization of fully managed cloud services to lessen the burden of internal systems management. The following services have been chosen for this purpose:

* DynamoDB:

Purpose: DynamoDB has been chosen as the database solution to effectively manage and store information associated with user-uploaded images and videos.

Features: Its key-value data style and schema-less architecture make it ideally suited for this purpose, offering exceptional performance and scalability. Furthermore, the security features, multi-AZ replication, and automated scalability perfectly match the specifications.

* API Gateway:

Purpose: Serving as a secure bridge between frontend and backend functionality, the API Gateway is an essential part of the system design.

Features: It simplifies the administration, publishing, and building of APIs. Optimal performance and improved reliability are ensured through the use of automatic load balancing and fault tolerance methods. Integration with AWS Cognito and customized authentication techniques offers a highly reliable system for controlling user access.

* Amazon S3:

Purpose: Amazon S3 is the best option for hosting a static website and storing user-uploaded material.

Features: Its outstanding qualities, such as scalability, durability, and strong security features, align perfectly with the specific technical requirements. It offers scalable and long-lasting media file storage, guaranteeing the security and safety of stored material.

* Amazon Cognito:

Purpose: Amazon Cognito delivers strong identity verification and user management features to provide safe and managed application access.

Features: It efficiently manages user registrations, logins, and user data storage on a large scale. It also has the ability to support multi-factor authentication, improving overall security.

**B:** Rising demand

The design prioritizes scalability and dependability in order to meet the increasing demand. Serverless computing, effective storage, and content delivery are used to maximize performance.

* Lambda:

Lambda ensures seamless scalability and reliability with its automatic scaling and efficient resource utilization. It efficiently handles surges in incoming requests and is event-driven for optimal concurrency management.

* DynamoDB Auto-scaling:

DynamoDB's auto-scaling capabilities are essential for maintaining scalability. It allows the database tables to adjust read and write capacity based on real-time traffic patterns, preventing throttling issues during increased traffic.

* API Gateway Dynamic Adjustment:

The API Gateway dynamically adjusts to incoming traffic, ensuring overall scalability and reliability. With automated load balancing and fault tolerance, it optimizes performance and adapts to varying traffic patterns efficiently.

**C:** Serverless / Event-driven solution

**D:** Replace the slow and costly relational database

**E:** Global response time needs an improvement

**F:** Video Media

Since business requirements expect the system to handle video media in the future, we make a video processing system similar to the photo processing system. The video file then goes through the same procedure as the photo does before getting processed, which means the metadata of the video as well as the user\_id from which the video came in will be stored in DynamoDB and the actual content of the video is stored in S3 bucket. Then a Lambda function will trigger and the video is put in the queue(SQS) waiting to be processed. We use Amazon Elemental Media Converter to transcode the video so devices can watch it across platforms, and devices with high-quality audio.

An alternative option to this is Amazon Elastic Transcoder, which provides a lower cost for a reduced monthly upload quantity and is simpler to use. On the other hand, Elemental Media Convert offers a more extensive range of features, providing control over the processing procedure, a lower price for a higher upload limit, and superior quality output.

**G:** Media processing

When a photo or video is uploaded to S3, the S3 bucket will send out a notification, and a Lambda function will handle the request, placing the media into its appropriate processing lane. At the beginning of each lane is an SNS node with SQS subscribed to it. For photos, there are two SQS subscriptions to one SNS, creating a parallel processing procedure. Consequently, the photo processing procedure is decoupled because each process runs in parallel and independently. Furthermore, since SNS allows for multiple subscribers, this design is scalable and extensible.

The video processing procedure follows the same architecture but only with one lane because there is only 1 requirement for it.

An alternative service to this is AWS Step Functions, It enables the coordination and visual representation of distributed application components as a sequence of state machine steps. Step Functions offers a graphical user interface for creating processes together with support for a number of AWS service interfaces, custom code, and error-handling features. But it is not included in this architecture due to its nature of complex workflows or coordinating multiple AWS services and custom logic

1. Design criteria

**A:** Performance, scalability (extensibility, decoupling, etc.)

* CloudFront: uses an extensive worldwide network of edge sites to cache and distribute content closer to end users, lowering latency and speeding up response times. Because it automatically adjusts to traffic variations, it can readily handle huge traffic volumes and abrupt surges in demand.
* Route 53: It is built for high availability and offers a globally dispersed network of DNS servers. It also lets you route traffic to various endpoints based on parameters like geography, latency, or set weights. By doing this, you can make sure that your DNS records are accessible and able to process large amounts of DNS queries
* API Gateway: automatically scales to accommodate large amounts of API calls. High availability is ensured by distributing traffic among several networks, and you may set up caching to lighten your backend services.
* S3 bucket: highly scalable, allowing you to store virtually unlimited amounts of data. It automatically scales to accommodate growing storage needs without any upfront configuration. S3 provides a reliable data storage solution and further guarantees great durability by storing data redundantly across many sites. S3 enables dependable and quick data transport, making it possible to upload and download big files and objects. Because of its large bandwidth capacities, it may be used to host websites, stream media, and distribute static content.
* DynamoDB: designed to scale horizontally and handle high volumes of reads and writes. It automatically partitions data across multiple servers, allowing it to scale seamlessly as application traffic increases. DynamoDB's low-latency data access makes it ideal for applications demanding rapid response times. It accomplishes this by replicating data across multiple availability zones within a region, enabling local, swift data retrieval.
* Lambda function: automatically scales your code in response to the incoming request rate. It dynamically allocates the required computing power to manage the workload and automatically scales down when demand subsides.
* Simple Notification Service: It adheres to a publish-subscribe messaging model. You can broadcast messages to a topic, and subsequently, subscribers who have indicated interest in that topic receive those messages. This arrangement disentangles the message originator (publisher) from the message recipient (subscriber).
* Amazon Simple Queue Service: decouple the components of your applications. Senders and receivers do not need to directly communicate, allowing for loose coupling and improved scalability. Messages are held in queues until they are processed, ensuring reliable and asynchronous communication between components. SQS is designed for high availability and durability. It distributes messages across multiple availability zones within a region to ensure redundancy and fault tolerance.

**B:** Reliability

* CloudFront: Because it automatically adjusts to traffic variations, it can readily handle huge traffic volumes and abrupt surges in demand.
* Route 53: lets you route traffic to various endpoints based on parameters like geography, latency, or set weights. By doing this, you can make sure that your DNS records are accessible and able to process large amounts of DNS queries
* API Gateway: High availability is ensured by distributing traffic among several networks, and you may set up caching to lighten your backend services.
* S3 bucket: S3 provides a reliable data storage solution and further guarantees great durability by storing data redundantly across many sites. S3 enables dependable and quick data transport, making it possible to upload and download big files and objects.
* DynamoDB: DynamoDB's low-latency data access makes it ideal for applications demanding rapid response times. It accomplishes this by replicating data across multiple availability zones within a region, enabling local, swift data retrieval.
* Simple Notification Service: It adheres to a publish-subscribe messaging model. You can broadcast messages to a topic, and subsequently, subscribers who have indicated interest in that topic receive those messages.
* Amazon Simple Queue Service: Messages are held in queues until they are processed, ensuring reliable and asynchronous communication between components. It distributes messages across multiple availability zones within a region to ensure redundancy and fault tolerance.

**C:** Security

* CloudFront: CloudFront integrates with other AWS services, such as AWS Shield and AWS Web Application Firewall (WAF), to provide protection against DDoS attacks and other security threats.
* Route 53: supports DNSSEC (DNS Security Extensions), which adds an extra layer of security by digitally signing DNS records, preventing unauthorized modifications or DNS spoofing.
* API Gateway: provides various security features, such as authentication and authorization options, including AWS Identity and Access Management (IAM), Cognito User Pools, and custom authorizers. It also supports SSL/TLS encryption for secure communication
* S3 bucket: provides various security features to protect your data. Access to buckets and objects can be controlled through bucket policies, Access Control Lists (ACLs), and AWS Identity and Access Management (IAM) roles and permissions. Additionally, you can enable server-side encryption to encrypt data at rest.
* DynamoDB: offers robust security features to protect your data. It integrates with AWS Identity and Access Management (IAM), allowing you to control access to tables and operations at a granular level. Encryption at rest and in transit is also supported for data protection.

**D:** Cost

* Pay-per-use pricing: Most of the services proposed use pay-per-use pricing, so the business only pays for the resources it actually uses. As a result, this service is very cost-effective.
* No upfront fee: This service doesn't ask for an upfront fee, and it doesn't require any long-term commitment.
* Scalability: Services such as Lambda and DynamoDB automatically scale up as demand increases and scale down when demand subsides, ensuring that costs accurately reflect the resources used.
* Data Storage: S3 and DynamoDB provide efficient data storage, allowing you to save money while ensuring high availability and durability.

VI) COST CALCULATION