COS20019 Assignment 3

Phan Vu 104222099
Tutor class: 1.00 pm Saturday
Swinburne University of Technology Ho Chi Minh, Vietnam
104222099@student.swin.edu.au

Tran Kim Thu 104061810
Tutor class: 1.00 am Saturday
Swinburne University of Technology Ho Chi Minh, Vietnam
104061810@student.swin.edu.au

***Abstract* - This report outlines the architectural plan for a flexible and easily navigable web application designed for a Multimedia Sharing platform. The presentation highlights the essential elements of the design, emphasizing incorporating various cloud services to guarantee strength and scalability. The goal is to delineate the data flow for fundamental operations such as uploading material, tracking user interactions, providing content suggestions, and enabling real-time search functionalities. The application's objective is to expand substantially, which requires ongoing development to meet increasing needs and provide customized user experiences.**

***Index terms* - Cloud infrastructure, web application, scalability, multimedia processing**

I. Introduction

Cloud computing has become essential for modern digital applications, providing flexible and practical solutions to address growing needs. This study proposes an architectural plan to improve and expand the Photo Album application, which has gained considerable success and is ready to meet the growing needs of users. Our design utilizes the functionalities of AWS and implements a serverless/event-driven architecture to align with modern cloud computing concepts.

Several problems and opportunities require the improvement of this application's features. This architectural solution utilizes AWS services and modern cloud computing principles to address the issues of escalating user traffic, global accessibility, optimization of media storage, and efficient processing of diverse media formats.

Our goal is to address the business scenario's objectives thoroughly, considering scalability, dependability, security, and cost-effectiveness. Our main goal as the architects of this proposed solution is to clarify the architectural elements and their reasoning to help you understand how each component smoothly fits into the system.

This study examines the use of cloud services to provide a substantial, scalable, and practical architecture for the Photo Album application while embracing technological progress. We evaluate complex architectural details and technology frameworks to determine a robust and future-oriented design and compare it to well-established industry benchmarks.

The following sections provide a complete analysis of the architectural elements, including cloud services, roles, interactions, and design justifications. UML collaboration diagrams and critical design analysis will offer a valuable understanding of the solution's efficacy.

This study aims to propose a comprehensive solution and offer direction for the long-lasting, expandable, and effective digital infrastructure of the Photo Album application, serving as a protectors of an innovative cloud architecture.

II. Architecture Overview

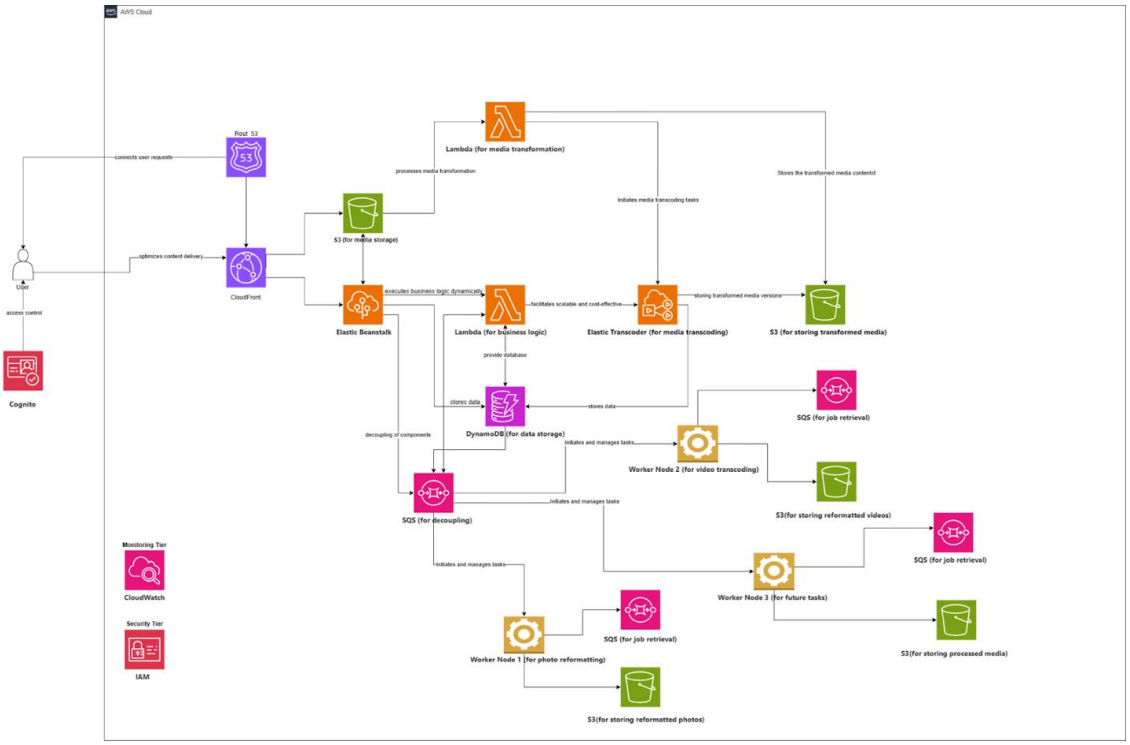

Figure 1: Architecture design overall

A. Five Tier Architecture

1. User Interface Tier

CloudFront: Provides low-latency, global access to the application, enhancing user experience.

Elastic Beanstalk: Simplifies deployment and scaling of web applications, ensuring the application layer can handle varying workloads.

2. Application Tier

Lambda (for business logic): Enables serverless execution of business logic, allowing for efficient scaling and cost optimization.

3. Data Storage Tier

S3 (for media storage): Provides scalable and durable object storage for storing large media files.

DynamoDB (for data storage): Offers single-digit millisecond performance at scale, suitable for quick data retrieval and storage.

4. Media Transformation Tier

Lambda (for media transformation): Enables on-demand media transformation, promoting efficiency and scalability.

Elastic Transcoder: Provides scalable media transcoding, ensuring compatibility with various devices.

5. Decoupling Tier

SQS (for decoupling): Enhances decoupling between components, allowing asynchronous communication and improving fault tolerance.

B. Detail each tier

User Interface Tier: The UI Layer is the main user interface to the media processing engine. Its purpose is to provide a seamless and engaging user experience. Amazon CloudFront leads the UI Layer as a worldwide CDN [1]. CloudFront secures, speeds up, and optimizes media distribution, increasing user experience. It reliably links customers to the media processing infrastructure by speeding up static and dynamic material delivery. AWS Elastic Beanstalk, which deploys and scales web apps, is seamlessly

integrated into the UI Layer. It coordinates the implementation and adjustment of user-interacted programs, enhancing system control.

Amazon S3 allows media asset storage and retrieval, providing a reliable and expandable media processing platform. Amazon Cognito optimizes user authentication, registration, and permissions. This ensures smooth and secure media processing system user interaction. Cognito enhances UI Layer security by managing identities. For effective domain routing, the UI Layer uses Amazon Route 53 [2]. This service connects user requests to CloudFront and other infrastructure. It optimizes traffic to AWS resources. Integration is necessary for a stable and customizable user interface. The UI Layer secures access with IAM policies. This protects sensitive user data and media material by restricting media processing system activities to authorized users.

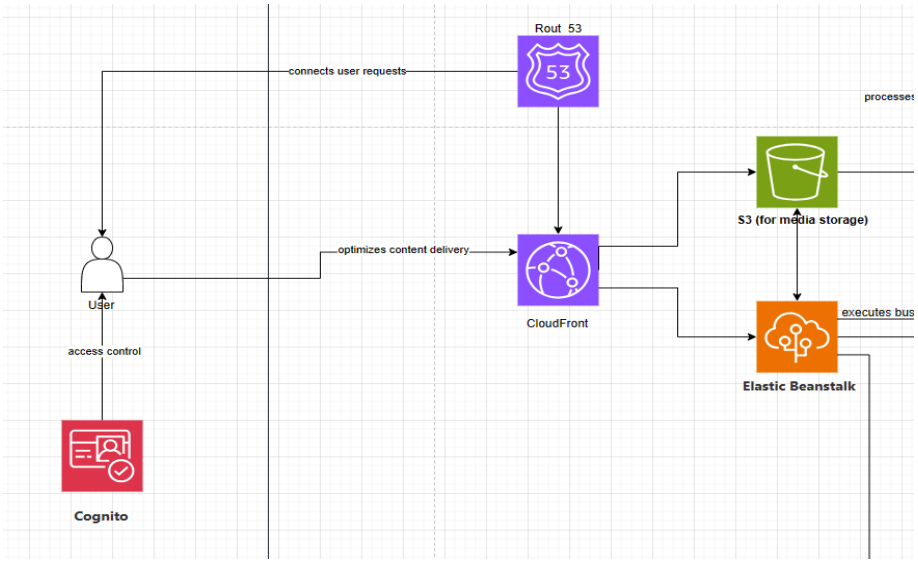

Figure 2: Architecture design for UI

Application Tier: The Application Tier controls media processing system logic and functionality. This layer is essential for user requests, business logic, and data storage and processing. AWS Elastic Beanstalk is an important Application Layer component for deploying, scaling, and managing web apps [3]. Its user-friendliness facilitates implementation, letting developers focus on application development rather than infrastructure management. Elastic Beanstalk easily supports multiple application architectures and frameworks, making it ideal for media processing. Flexible Beanstalk connects other major Application Layer components. It executes business logic with AWS Lambda for serverless computing. Data storage and retrieval are efficient using DynamoDB, a scalable and performant NoSQL database [4]. Integration with Amazon Simple Queue Service (SQS) allows component separation, ensuring the ability to tolerate increased workload and system faults by handling asynchronous communication between system pieces.

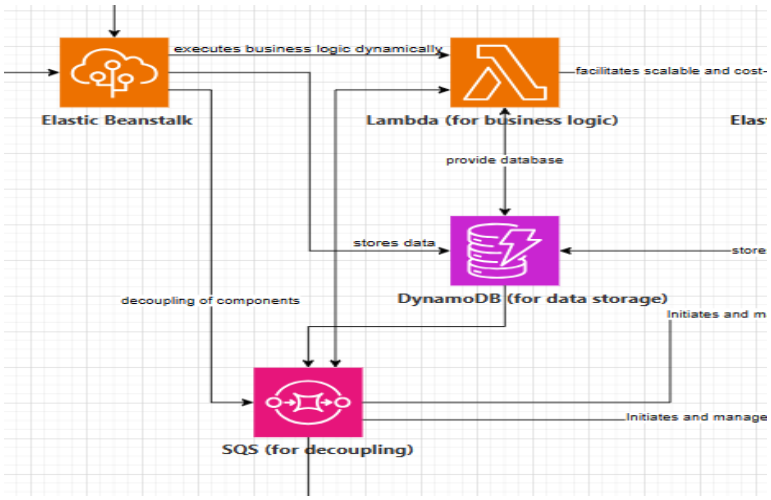

Figure 3: Architecture design for Application Layer

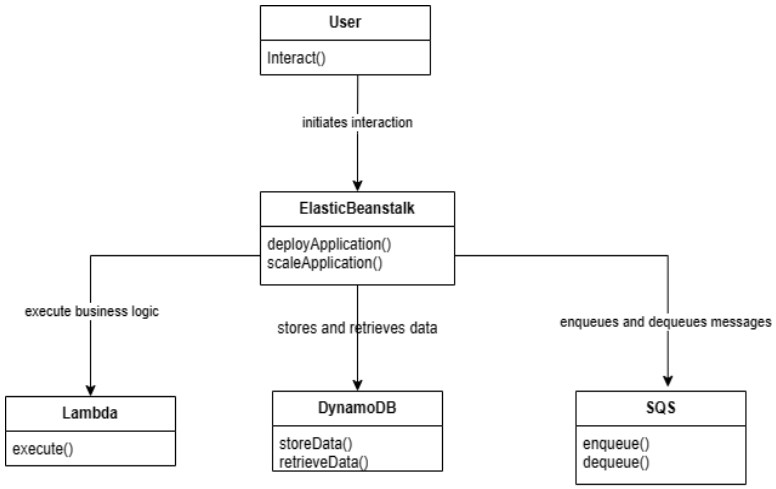

Figure 4: UML class diagram for Appication Layer

Data Storage Tier: The Data Storage Tier stores media assets and their data for reliable data management. This layer stores Amazon DynamoDB structured data and Amazon S3 media. Amazon S3 is highly scalable and stores objects, mostly video assets, in the Data Storage Layer. S3 provides unmatched data availability, security, scalability, and performance. It can store almost infinite data, making it ideal for managing media assets' diverse features in multiple formats. The easy integration of S3 and AWS Lambda allows real-time media conversion. New media objects uploaded to S3 trigger Lambda functions for dynamic media processing and modification. Media assets are efficiently prepared for multiple devices and platforms with this integration. Structured data is stored in Amazon DynamoDB, a managed NoSQL database. Regardless of volume, DynamoDB processes data in single-digit milliseconds. This makes it ideal for high-read/write applications. DynamoDB executes business logic easily with AWS Lambda. Integration ensures data retrieval and modification in

DynamoDB, it will automatically activate serverless functions. This allows for immediate and adaptable reactions to changes to the underlying data structure.

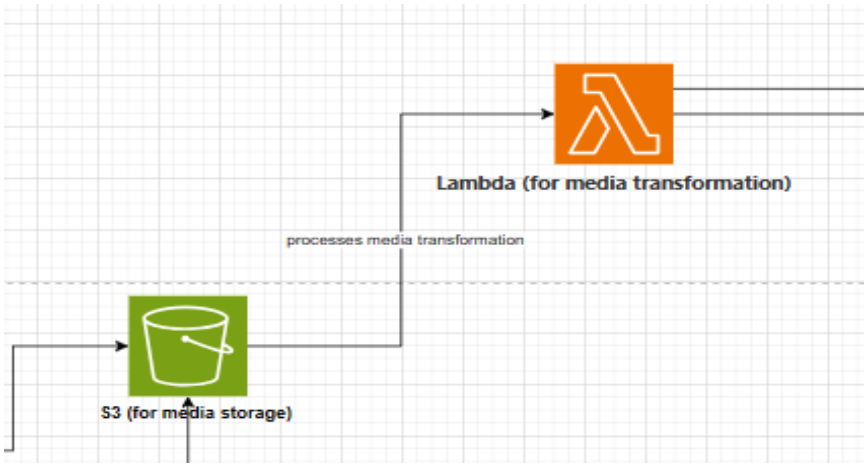

Figure 5: Architecture for data storage

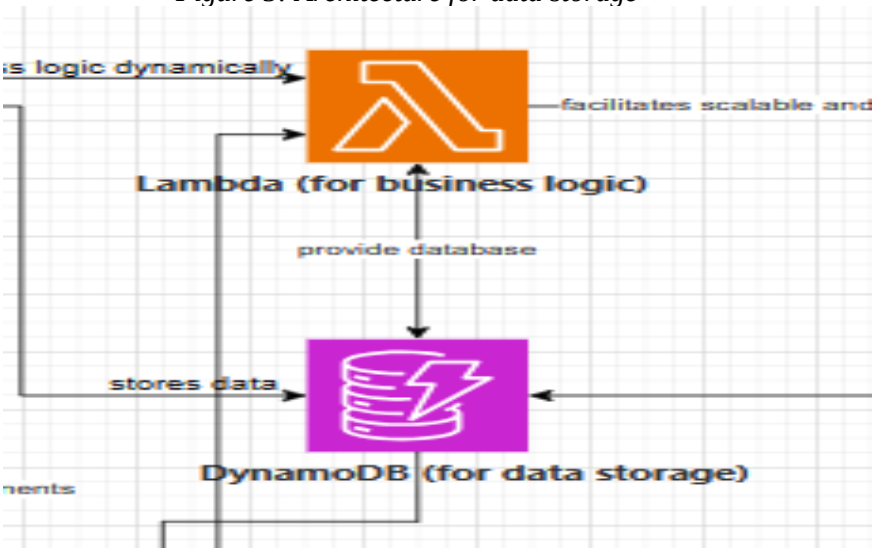

Figure 6: Architecture for data storage

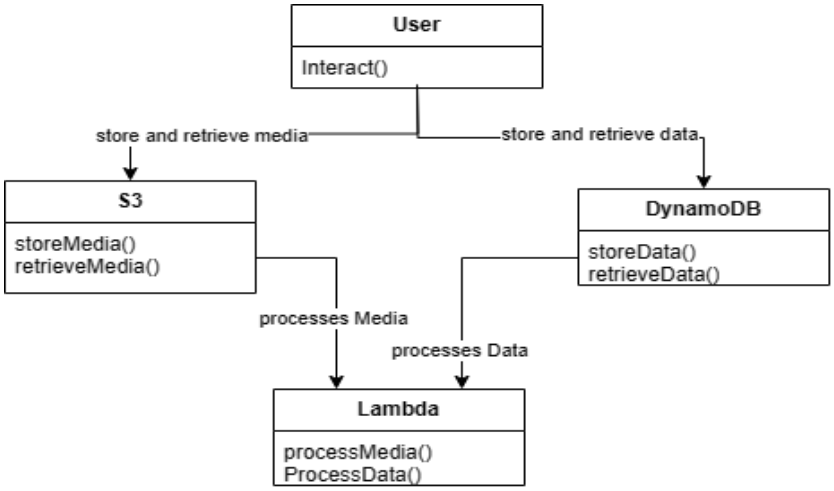

Figure 7: UML class diagram for data storage

Media Transformation Tier: For smooth media consumption across devices and platforms, the Media Transformation Tier dynamically processes and optimizes media assets. AWS Lambda and AWS Elastic Transcoder transform and transcode media in this layer. AWS Lambda, the Media Transformation Layer's core, provides serverless real-time media processing. Lambda routines dynamically and personally modify media assets as new objects are uploaded to Amazon S3. Media manipulation is possible with this layer's lambda functions. Photos and videos can be cropped, watermarked, and filtered. The Media Transformation Layer integrates well with AWS Elastic Transcoder. Scalability, usability, and cost are expertly designed into Elastic Transcoder. This program converts multimedia. Lambda's scalability and Elastic Transcoder's parallel processing make high-volume media transformation efficient.

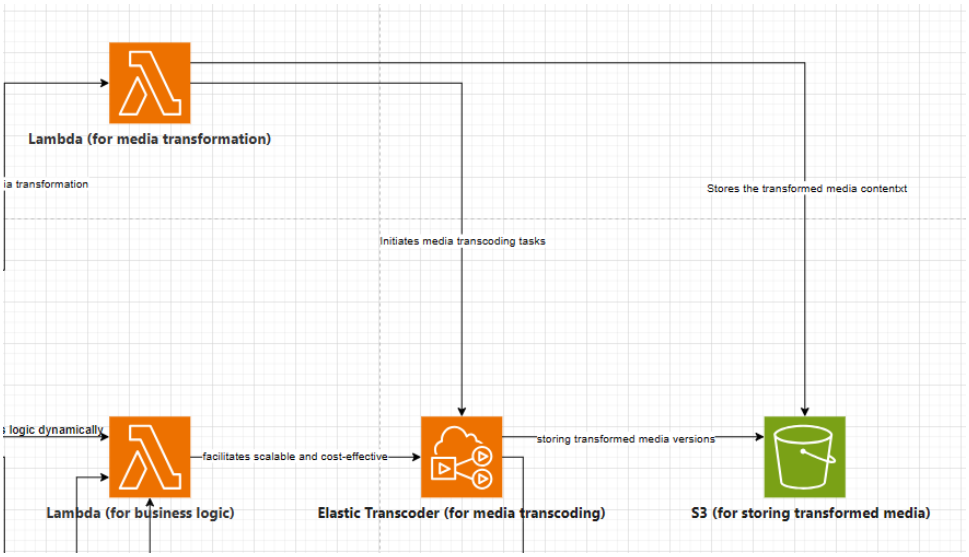

Figure 8: Architecture design for media transformation

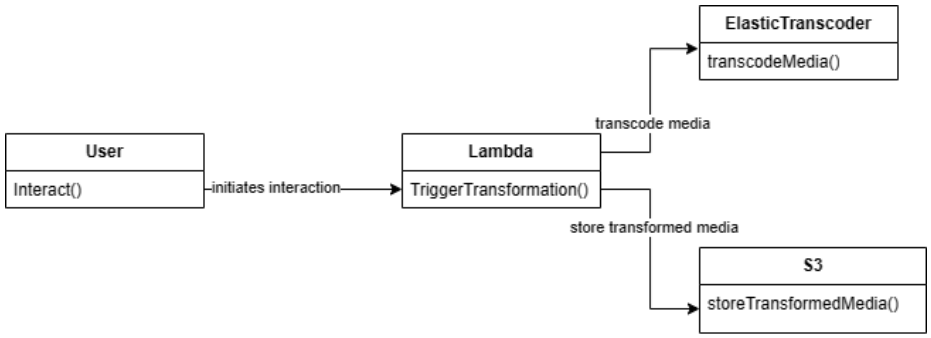

Figure 9: UML class diagram for media transformation

Decoupling Tier: The Decoupling Layer is essential to media processing architecture. Asynchronous communication and loose coupling are possible. Amazon SQS, a completely controlled message queuing service, forms the core of this layer. By controlling message queues and encouraging component communication, the Decoupling Layer scales and protects processing. Amazon SQS decouples media processing system members so they can work independently. Message queues provide reliable, extensible communication between system components. SQS seamlessly triggers AWS Lambda functions that implement exact business logic. By using SQS as an event source for Lambda, the Decoupling Layer eliminates direct dependencies and allows asynchronous communication and processing across components. SQS coordinates worker services through central communication. Each worker node can independently retrieve and process queue messages, enabling concurrent and distributed

task execution. SQS delays message delivery by keeping notices in the line for a set time before processing.

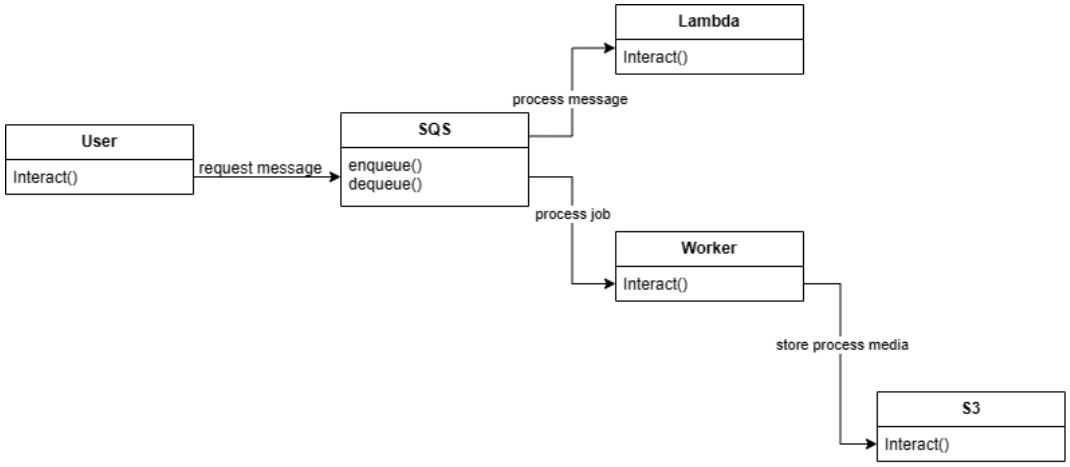

Figure 10:UML class diagram for Decoupling

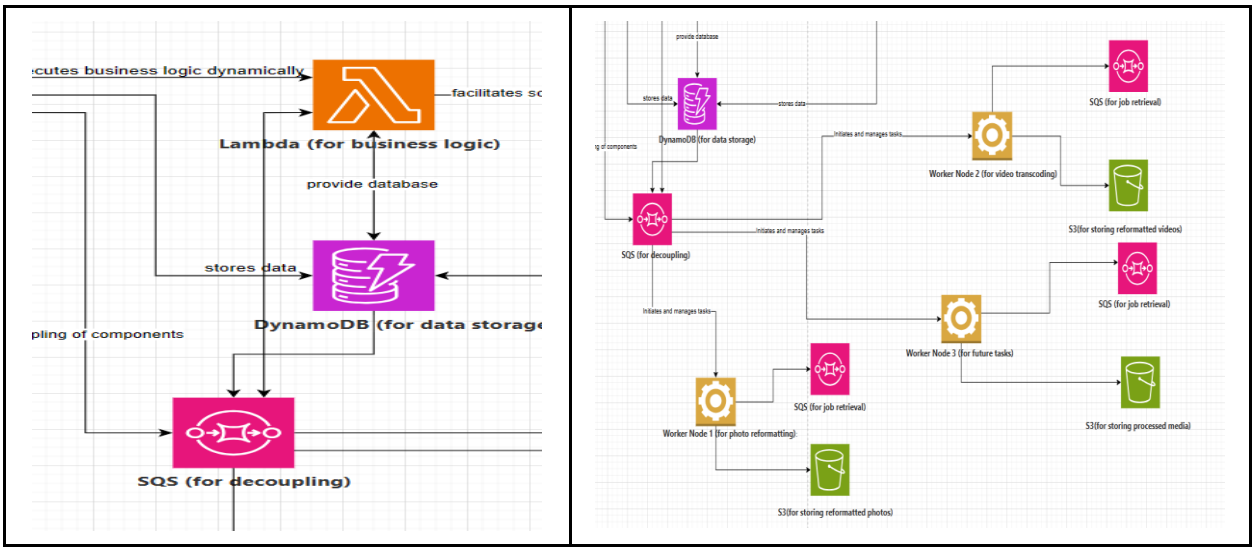

Figure 11:Architecture Design for Decoupling

III. Design Rationale

A. Business Scenario Fulfillment

1. Scalability for Global Demand

The business scenario entails managing fluctuating workloads of media processing jobs, which may encounter sudden increases in demand from users worldwide. By utilizing AWS CloudFront for worldwide content distribution and AWS Elastic Beanstalk for expandable web applications, the architecture guarantees that the system automatically adjusts resource allocation according to demand. By implementing responsive scaling, a uniform user experience is ensured across different regions, thereby tackling the issue of handling varying workloads.

2. Efficient and Dynamic Media Processing

The company needs flexible media processing for photo reformatting, video converting, and future assignments. It seamlessly integrates AWS Lambda for serverless computation and AWS Elastic Transcoder for media transcoding. Worker nodes dedicated to reformatting photos and transcoding videos

enable parallelized and efficient material processing. Due to its modular design, worker Node 3 is reserved for future media processing needs.

3. Cost-Effective Resource Utilization

Media processing cost efficiency and resource utilization are corporate priorities. The architecture uses auto-scaling, AWS Lambda pay-as-you-go models, and Elastic Transcoder tiered pricing to maximize resource use. This strategic cost management method lets the company adjust resources to demand, optimizing operational costs in high- and low-demand periods.

4. Secure and Resilient Data Management

Media and user data security is a top business priority. Amazon S3 stores media, and DynamoDB organizes structured data in the Data Storage Layer, providing security. Media assets and user data are protected by encryption, IAM, and version control protocols. This meets the business's robust and reliable data management plan.

5. Adaptability to Future Technologies

The company plans to use AI-powered tasks to improve media processing. Worker Node 3's reservation for future duties shows the architecture's progressive approach, ensuring technology adaptability. The modular architecture lets the company easily add new features and stay ahead of technology.

6. Asynchronous and Decoupled Processing

Media jobs must be handled efficiently and independently to maintain system responsiveness and fault tolerance. Asynchronous communication and loose coupling are enabled by the Decoupling Layer using Amazon SQS. This architecture improves the system's ability to handle many users or tasks, recover from failures, and multitask. The architecture guarantees autonomous charge processing, preventing obstacles and enabling modular scalability.

7. Real-time Monitoring and Operational Insights

The business requires continuous monitoring and practical insights into media processing system operational well-being and efficiency. Integration with Amazon CloudWatch allows instant metric tracking and measurement. This ensures that the business can actively manage design, respond to performance changes, and optimize resource use. The architecture enables real-time decision-making and system optimization for the company.

B. Alternative Solutions Considered:

1. Virtual machines vs. Containers vs. Serverless computing.

Responsibility and vendor lock-in depend on organization readiness. Containers and virtual machines provide more precise server resource control than serverless configurations [5]. Simpler serverless provisioning speeds up service launch. Serverless architecture, particularly Lambda, reduces operational burden, automates scaling, and meets user expectations for efficient scaling.

Aspect	*Virtual Machines (VMs)*	*Containers*	*Serverless*
Resource Allocation	Divided into multiple	Shares host OS resources	Utilizes cloud-based
	VMs	lightweight, no Guest OS	servers, abstracts
			underlying infrastructure
Isolation	Hard isolation between	Shares the host kernel,	Multi-tenant environment,
	different VMs	but separate user spaces	limited isolation between
		for applications	functions
Portability	Less portable	Highly portable, can move	Limited portability between

		between cloud environments	different cloud providers
Resource Efficiency	More resource intensive	Efficient resource usage,	Efficient, cost-effective
		shares OS resources	usage
Scalability	Scalable	Highly scalable	Highly scalable
Ease of Deployment	Dependent on VM setup	Quick deployment,	Rapid deployment,
		facilitated by container	abstraction from underlying
		images	infrastructure
Cost Efficiency	May require more	Cost-efficient due to	Cost-effective, billed on
	resources	efficient resource usage	usage, not idle time
Vendor Lock-In	Potential dependence	Potential dependence	Potential vendor lock-in,
	on specific VM tech	on specific container tech	dependent on provider APIs
Security	High isolation between	Security concerns due to	Security concerns due to
	VMs, dedicated OS	shared host kernel,	multi-tenancy, shared
		additional measures needed	server environment

2. SQL vs. NoSQL database

The preference for DynamoDB stemmed from its ability to function at scale with sub-millisecond latency, making it well-suited for rapid data retrieval and storage. This is particularly important for media-related applications that require high throughput[5].

Aspect	***SQL***	***NoSQL***
Definition	Structured Query Language	Not Only SQL
Database Architecture	Relational (RDBMS)	Non-relational (Various models)
Suitable for	Structured data with predefined schema	Unstructured and semi-structured data
Data Storage	Tables with columns and rows	Collections or documents
Query Language	SQL	Dynamic schemas, diverse query syntax
Scaling	Vertically scalable	Horizontally scalable
Transaction Management	ACID properties for transactions	May or may not follow ACID properties
Complex Queries	Supports JOIN and complex queries	Limited or no support for JOINs
Data Structure	Normalized data structure	Denormalized data structure
Examples	MySQL, PostgreSQL, Oracle	MongoDB, Cassandra, Redis

3. Caching options

Caching at the Edge (CloudFront):

- Rationale: Exploited due to the inherent characteristics of multimedia content. CloudFront's edge caching strategically delivers frequently visited material from edge locations, lowering latency and optimizing performance by serving information closer to users.

Alternative Considerations:

- AWS ElastiCache: While ElastiCache offers in-memory caching solutions, using CloudFront for edge caching aligns better with the distributed nature of multimedia content delivery, reducing the load on backend systems and enhancing user experience.

4. Push vs. Pull message handling

Pull Mechanism (SQS):

- Rationale: Selected for its advantages in decoupling, fault tolerance, and scalability. The pull mechanism of SQS enables components to obtain messages autonomously, fostering loose connectivity among various system components.

Alternative Considerations:

- Push Mechanism: While push mechanisms might offer real-time delivery, they can create tight dependencies between components, potentially leading to increased coupling and reduced fault tolerance.

5. The number of tiers in architecture

Five-Tier Architecture:

- Rationale: Chosen for its capacity to separate different aspects, support expansion, and encourage independent functioning of other components. This hierarchical strategy maximizes system efficiency and facilitates the wide range of capabilities necessary for multimedia processing and content distribution.

Alternative Considerations:

- Reduced Tiers (e.g., 3-tier): A simpler architecture might streamline operations but could sacrifice granularity and segregation of functionalities, potentially limiting scalability and manageability as the platform grows.

C. Design Criteria

	Criteria	**Solution**
Performance	1. Performance Boost	1. Amazon CloudFront enhances the performance of the media processing system by reducing latency and accelerating the delivery of media assets.
	2. Quick Response Times	2. AWS Lambda contributes to performance by providing serverless computing capabilities.
	3. Low-Latency Data Access	3.DynamoDB plays a key role in performance by offering low-latency data access.
	4. Cost-Effective Transcoding	4. Elastic Transcoder primarily addresses scalability, it indirectly contributes to performance. By offering scalable media transcoding capabilities, the service ensures the system can efficiently process and deliver media files of varying sizes and formats.

Scalability	1. Global Reach	1.CloudFront's extensive network of edge locations contributes to scalability by providing a distributed infrastructure that can efficiently scale to handle increasing demands.
	2. On-Demand Scalability	2.AWS Lambda is a cornerstone for scalability due to its serverless nature.
	3. Scalable and Reliable	3.DynamoDB directly addresses scalability by offering a highly scalable NoSQL database.
	4. Highly Scalable	4. Elastic Transcoder is explicitly designed for scalable media transcoding.
Reliability	1. Fault Tolerance	1. SQS is employed for asynchronous communication, enhancing fault tolerance by decoupling components.
	2. Cascading Failure Prevention	2. Critical data, including media assets, is stored redundantly using services like Amazon S3 and DynamoDB.
	3. Data Integrity and Retention	3. Decoupling and Workers are equipped with robust retry mechanisms and error-handling procedures.
	4. Scalability and Load Balancing	4. Lambda and Elastic Transcoder enable the architecture to scale dynamically based on demand.
Security	1. Secure Access Control	1. AWS IAM is employed to manage access to AWS services and resources securely.
	2. Authentication and Authorization	2. Amazon Cognito is integrated to add secure user sign-up, sign-in, and access control functionalities to web and mobile applications.
	3. Access Management	3. IAM policies are meticulously crafted to manage access control for each component in the architecture.

D. Cost Estimation.

Assumption Cost for 1TB[6]**:**

Service	Quantity	Total Monthly cost
S3 storage	1 TB per month x 1024 GB in a TB = 1024 GB per month **1024 GB x 0.09 USD per GB = 92.16 USD** Tiered price for: 1,024 GB **1,024 GB x 0.023 USD = 23.55 USD** Total tier cost = 23.552 USD (S3 Standard storage cost) S3 Standard cost (monthly): 23.55 USD	**115.71 USD**
CloudFront	Tiered price for: 5,120 GB 5,120 GB x 0.085 USD = 435.20 USD Total tier cost = 435.20 USD (Data transfer out) Data transfer out to internet cost: 435.20 USD	**547.60 USD**

	5,120 GB x 0.02 USD = 102.40 USD (Data transfer out to origin) Data transfer out to origin cost: 102.40 USD 10,000,000 requests x 0.000001 USD = 10.00 USD (HTTPS requests) Requests cost: 10.00 USD 435.20 USD + 102.40 USD + 10.00 USD = 547.60 USD (Total cost United States) CloudFront price (monthly): **547.60 USD**	
Lambda	10,000,000 requests x 10 ms x 0.001 ms to sec conversion factor = 100,000.00 total compute (seconds) 10 GB x 100,000.00 seconds = 1,000,000.00 total compute (GB-s) 1,000,000.00 GB-s x 0.0000166667 USD = 16.67 USD (monthly compute charges) 10,000,000 requests x 0.0000002 USD = 2.00 USD (monthly request charges) 0.50 GB - 0.5 GB (no additional charge) = 0.00 GB billable ephemeral storage per function 16.67 USD + 2.00 USD = 18.67 USD **Lambda costs - Without Free Tier (monthly): 18.67 USD**	**18.67 USD**
Rout 53	Tiered price for: 3 3 x 0.50 USD = 1.50 USD Total tier cost = 1.50 USD (Hosted Zone cost) 10,000 records x 0.0015 USD per record = 15.00 USD (RRset records cost) 1.50 USD + 15.00 USD = 16.50 Total Hosted Zones and RRset records cost Total Hosted Zones & RRset records cost: 16.50 USD 2 policy record per month x 50.00 USD = 100.00 USD (Traffic Flow cost) 10 million queries x 1000000 multiplier for million = 10,000,000.00 Standard queries in million Tiered price for: 10,000,000.00 Standard queries 10,000,000 Standard queries x 0.0000004 USD = 4.00 USD Total tier cost = 4.00 USD (Standard queries cost) 5 million queries x 1000000 multiplier for million = 5,000,000.00 billable Latency based routing queries Tiered price for: 5,000,000.00 Latency based routing queries 5,000,000 Latency based routing queries x 0.0000006 USD = 3.00 USD Total tier cost = 3.00 USD (Latency based routing queries cost) 1 million queries x 1000000 multiplier for million = 1,000,000.00 billable Geo DNS queries	**125.00 USD**

	Tiered price for: 1,000,000.00 Geo DNS queries 1,000,000 Geo DNS queries x 0.0000007 USD = 0.70 USD Total tier cost = 0.70 USD (Geo DNS queries cost) 1 million queries x 1000000 multiplier for million = 1,000,000.00 billable IP-based routing queries Tiered price for: 1,000,000.00 IP-based routing queries 1,000,000 IP-based routing queries x 0.0000008 USD = 0.80 USD Total tier cost = 0.80 USD (IP-based routing queries cost) Tiered price for: 1,000 IP (CIDR) blocks 1,000 IP (CIDR) blocks x 0.00 USD = 0.00 USD Total tier cost = 0.00 USD (IP (CIDR) blocks cost) 16.50 USD + 100.00 USD + 4 USD + 3 USD + 0.70 USD + 0.80 USD = 125.00 USD **Route53 Hosted Zone cost (monthly): 125.00 USD**	
Simple Queue Service	10 requests per month x 1000000 multiplier for million = 10,000,000.00 total standard queue requests Tiered price for: 10,000,000.00 requests 1,000,000 requests x 0.00 USD = 0.00 USD 9,000,000 requests x 0.0000004 USD = 3.60 USD Total tier cost: 0.00 USD + 3.60 USD = 3.60 USD (Standard queue requests cost) 10 requests per month x 1000000 multiplier for million = 10,000,000.00 total FIFO queue requests Tiered price for: 10,000,000.00 requests 1,000,000 requests x 0.00 USD = 0.00 USD 9,000,000 requests x 0.0000005 USD = 4.50 USD Total tier cost: 0.00 USD + 4.50 USD = 4.50 USD (FIFO queue requests cost) 3.60 USD + 4.50 USD = 8.10 USD (Total SQS cost) **Total SQS cost (monthly): 8.10 USD**	**8.10 USD**
Elastic Transcoder	DynamoDB data storage cost (Monthly): 256.00 USD Video assets transcoding cost (Monthly): 180.00 USD Audio (only) assets transcoding cost (Monthly): 9.00 USD Total Upfront cost: 0.00 USD **Total Monthly cost: 189.00 USD**	**189.00 USD**
CloudWatch	Tiered price for: 5 Dashboards 3 Dashboards x 0.00 USD = 0.00 USD 2 Dashboards x 3.00 USD = 6.00 USD Total tier cost: 0.00 USD + 6.00 USD = 6.00 USD (Dashboards cost) CloudWatch Dashboards and Alarms cost (monthly): 6.00 USD Tiered price for: 10 metrics 10 metrics x 0.30 USD = 3.00 USD	**9.00 USD**

	Total tier cost = 3.00 USD (Metrics cost (includes custom metrics)) CloudWatch Metrics cost (monthly): 3.00 USD **Total Monthly cost: 9.00 USD**	
DynamoDB	Monthly read cost (Monthly): 2.76 USD Monthly write cost (Monthly): 23.38 USD DynamoDB Data export to Amazon S3 cost (Monthly): 100.00 USD DynamoDB Data import from Amazon S3 cost (Monthly): 75.00 USD DynamoDB data storage cost (Monthly): 256.00 USD Upfront read cost (Upfront): 30.00 USD Upfront write cost (Upfront): 150.00 USD Total Upfront cost: 180.00 USD **Total Monthly cost: 457.14 USD**	**457.14 USD**
Cognito	5,000 MAUs x 0.10 SAML or OIDC federation requests = 500.00 SAML or OIDC federation MAU requests 500.00 SAML or OIDC federation MAUs - 50 free SAML or OIDC federation MAU requests per month = 450.00 billable SAML or OIDC federation MAU requests Max (450.00 billable SAML or OIDC federation MAU requests, 0 minimum billable SAML or OIDC federation MAU requests) = 450 total billable SAML or OIDC federation MAU requests 450 MAUs x 0.015 USD = 6.75 USD (SAML or OIDC federation MAU requests) SAML or OIDC federation cost (monthly): 6.75 USD 5,000 MAUs - 50000 free MAU requests per month = -45,000.00 billable MAU requests Max (-45000.000000 billable MAU requests, 0 Constant Unit) = 0.00 total billable MAU requests Tiered price for: 0.00 MAUs Total tier cost = 0.00 USD (User Pool MAUs) User Pool MAU cost (monthly): 0.00 USD Advanced security feature cost (monthly): 0 USD **Cognito MAU cost (monthly): 6.75 USD**	**6.75 USD**
AWS SWF	3,000 executions per month x 0.0001 USD = 0.30 USD (Workflow Executions cost) 20 Tasks, Markers, Timers and Signals x 3,000 executions per month x 0.000025 USD = 1.50 USD (Tasks, Markers, Timers and Signals cost) 30 days + 60 days = 90.00 Workflow days 90.00 days x 3,000 executions x 0.000005 USD = 1.35 USD (Open and retained workflow cost) 0.30 USD + 1.50 USD + 1.35 USD = 3.15 USD (Simple Workflow Cost)	**3.15 USD**

	Simple Workflow Cost (monthly): 3.15 USD	
Total:		**1.480,12 USD**

IV. References

[1]"What is Amazon CloudFront? - Amazon CloudFront," *What is Amazon CloudFront? - Amazon CloudFront*. https://docs.aws.amazon.com/AmazonCloudFront/latest/DeveloperGuide/Introduction.html

[2]L. Cushman, "Manage your clients' domains easily with Amazon Route 53," *Pax8 Blog*, Aug. 30, 2023. https://www.pax8.com/blog/manage-domains-easily-with-amazon-route-53/ (accessed Nov. 25, 2023).

[3]S. Cholewinski, "Scaling your business with Elastic Beanstalk," *Scaling your business with Elastic Beanstalk*, Aug. 14, 2023. https://us.nttdata.com/en/blog/2023/august/elastic-beanstalk (accessed Nov. 25, 2023).

[4]B. Damue, "Exploring the NoSQL Powerhouse Amazon DynamoDB," *DEV Community*, Jul. 11, 2023. https://dev.to/brandondamue/exploring-the-nosql-powerhouse-amazon-dynamodb-394b (accessed Nov. 25, 2023).

[5]P. Manager, "Virtual Machine vs Container vs Serverless," *Whitebox Solutions*, Jul. 11, 2022. https://www.whiteboxsolution.com/blog/virtual-machine-vs-container-vs-serverless/

[6]Coursera, "SQL vs. NoSQL: The Differences Explained + When to Use Each," *Coursera*. https://www.coursera.org/articles/nosql-vs-sql (accessed Nov. 25, 2023).

[7]"AWS Pricing Calculator," *AWS Pricing Calculator*. https://calculator.aws/