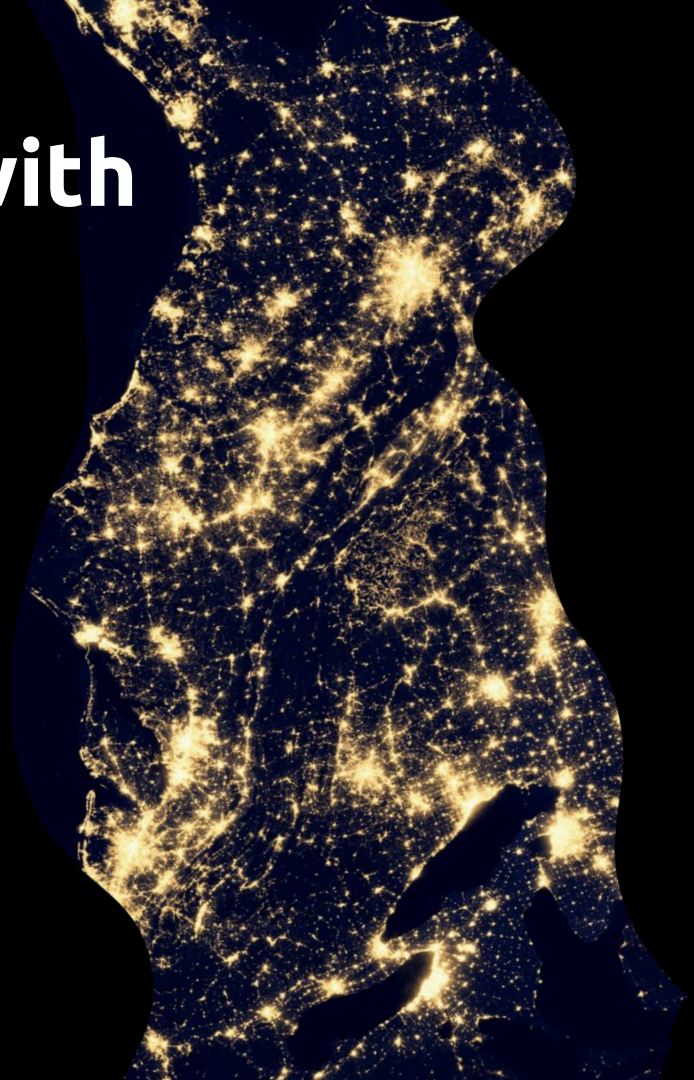
Scaling Architecture with AWS services

Sovathana Phat

Senior full stack software engineer

Slash



ជំរាបសួរ!

aws CADT /slash

Case study: Moosic social network platform

We want to create a social network platform by following the requirements:

- User can share short video to thousands of users.
- User activities like, comment and share.
- User can chat
- User can use and login only from Cambodia

/slash

On-Premises Approach:

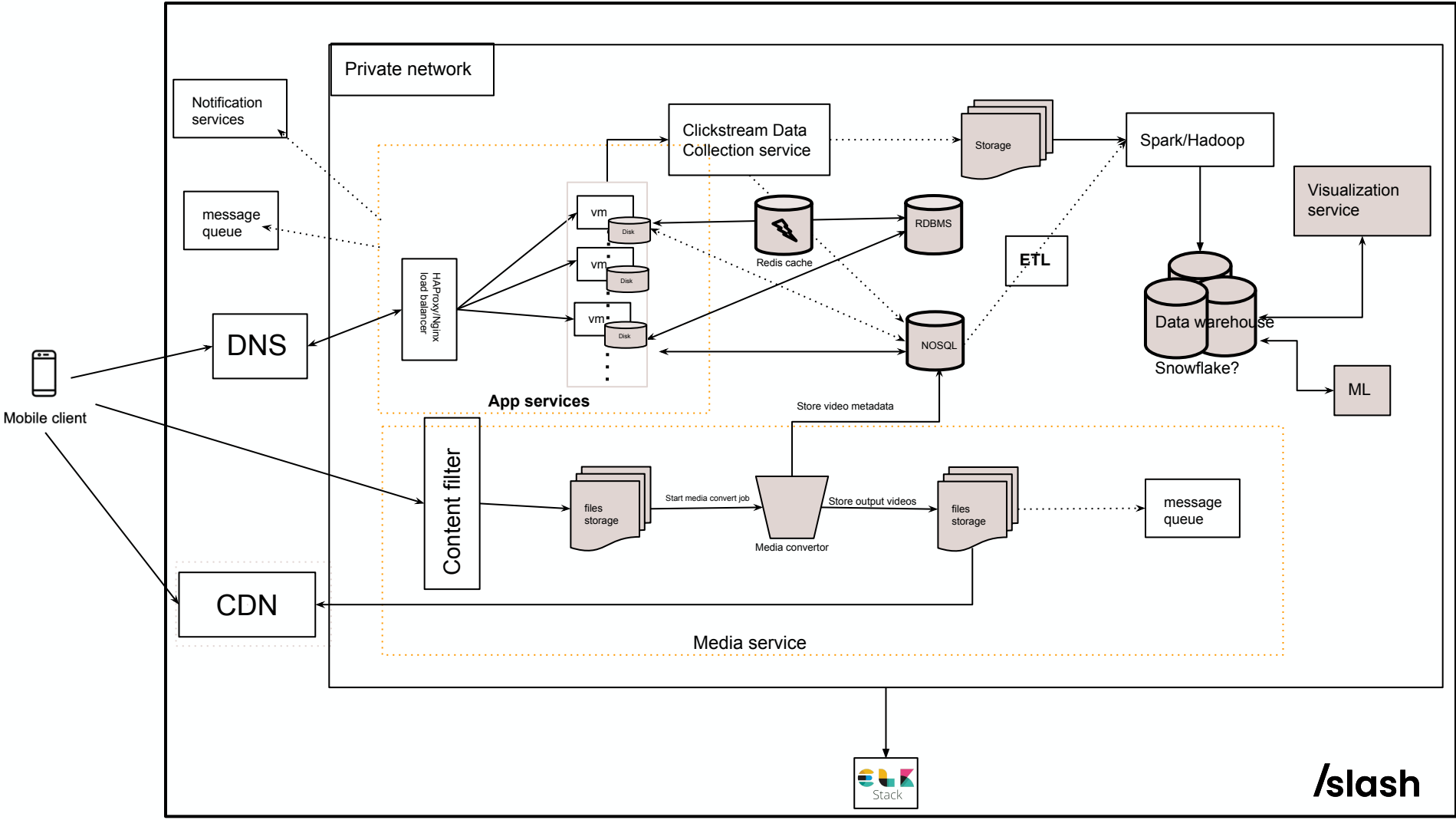
We need:

- On premises servers
- Datacenter Space
- Local ISP
- Electricity and Backup power Generator
- IT specialist

Time launch:

- At least 6 months to one year

Upfront cost is very high, while monthly cost is not small


Private network
Notification services
message queue
DNS
Mobile client
CDN
Clickstream Data Collection service
Storage
Spark/Hadoop
Visualization service
HAProxy/Nginx load balancer
vm
Disk
Redis cache
RDBMS
NOSQL
ETL
Data warehouse
Snowflake?
ML
App services
Store video metadata
Content filter
files storage
Start media convert job
Media convertor
Store output videos
files storage
message queue
Media service
Stack
/slash

Assume go with on-premise:

After 6 months, we launched successfully.

And after 3 more months,

- Hit with 500% users increase
- Our Cambodian people and other people from oversea wanted to use also

What are the challenges?

After a lot of demands,

- We need more servers on-premise (CPU, hard disk, RAM)
- Latency issue from India, USA, China, Australia

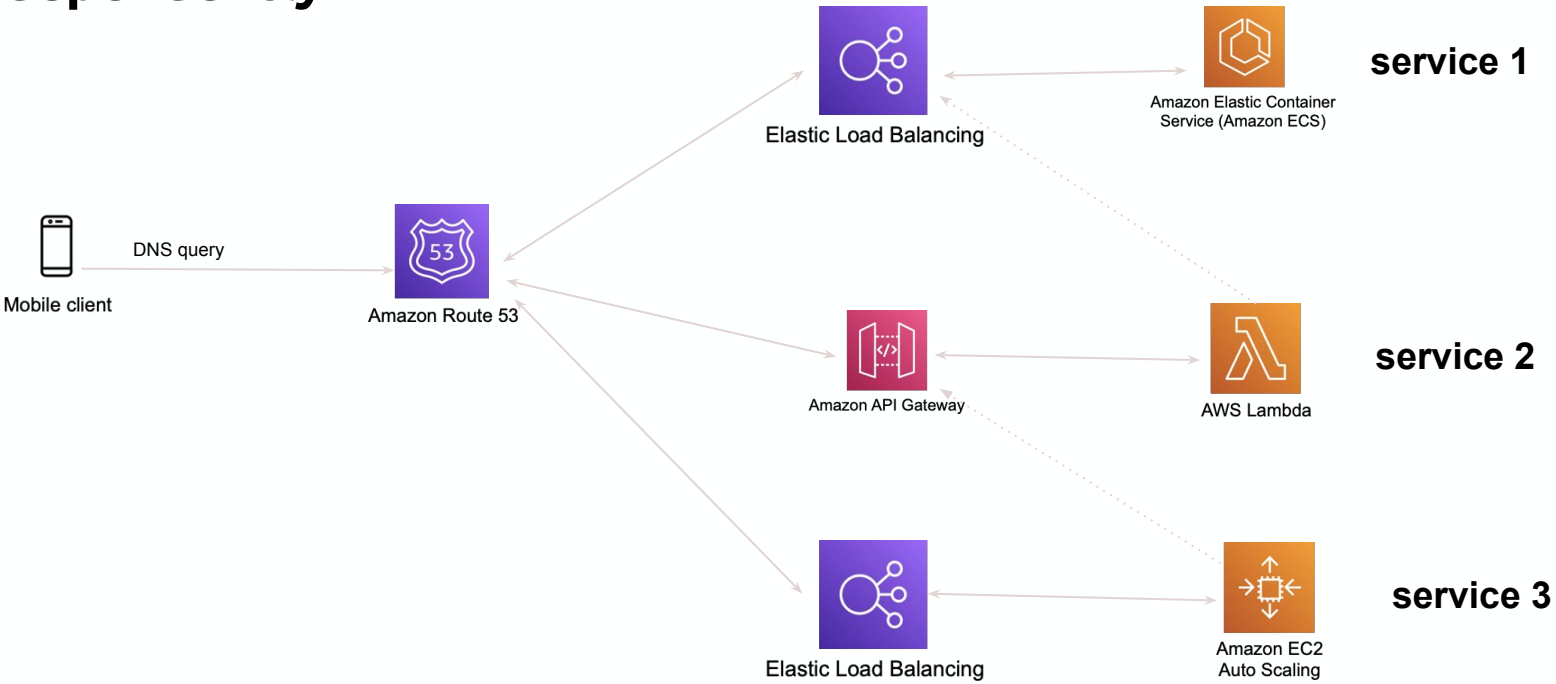
As a result, we lose users

Cloud Approach use AWS services:

- Scalability
  - Microservice Architecture with Serverless or EC2
  - Provision new instances fast (auto scaling)
- Security
  - Secure user’s videos by giving access to their own s3 folder
- Cost efficiency
  - Adaptive streaming with caching
- High Performance
  - The database should scale and high read throughput

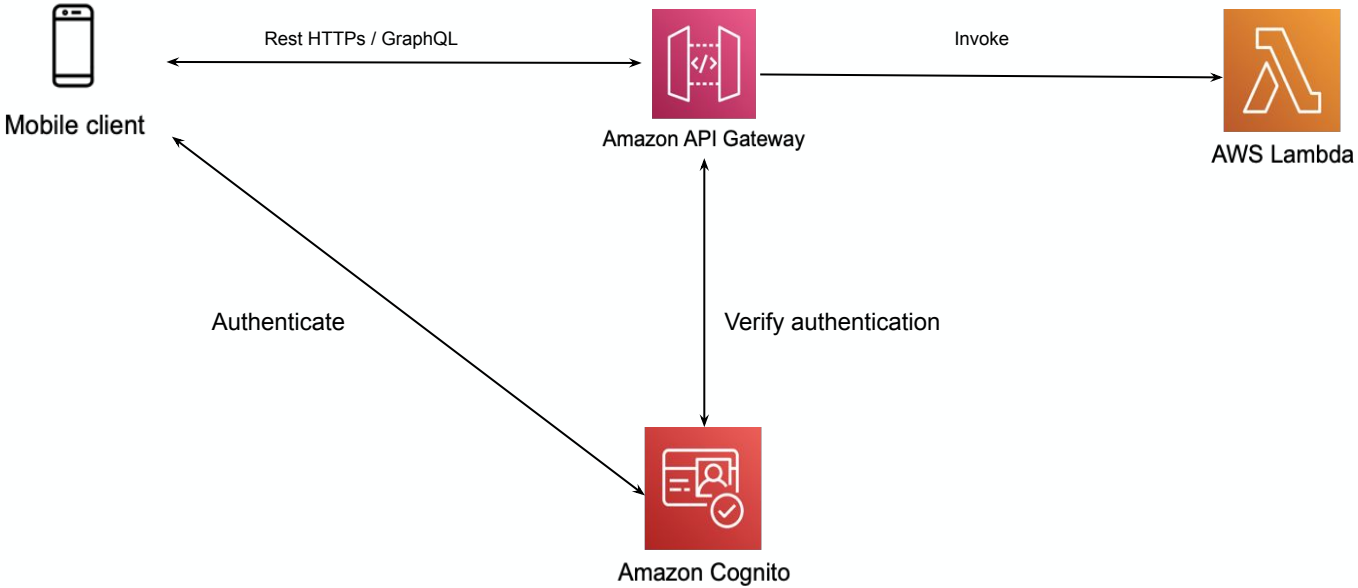
/slash

Microservice Architecture with Serverless or EC2 and scale independently

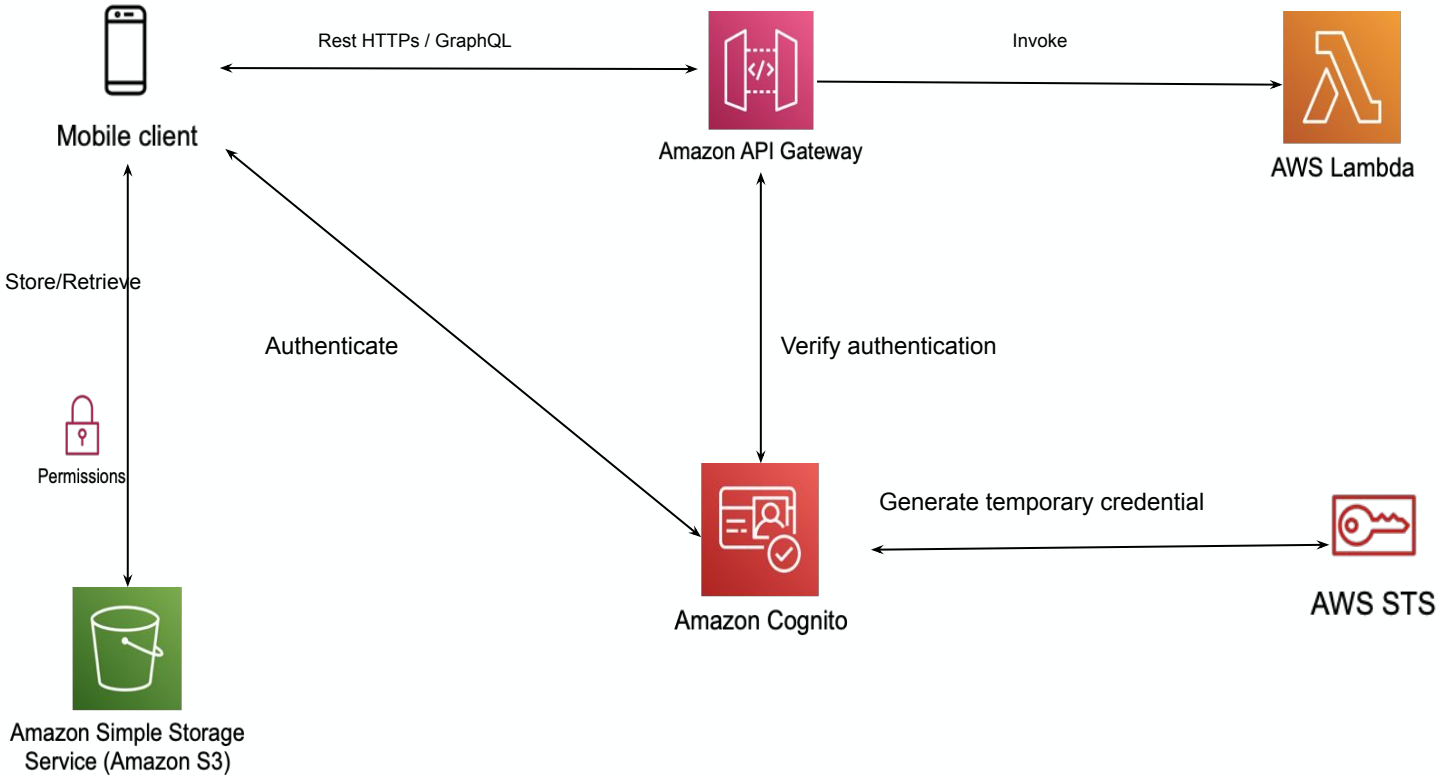


/slash

Mobile App Rest API with HTTPs

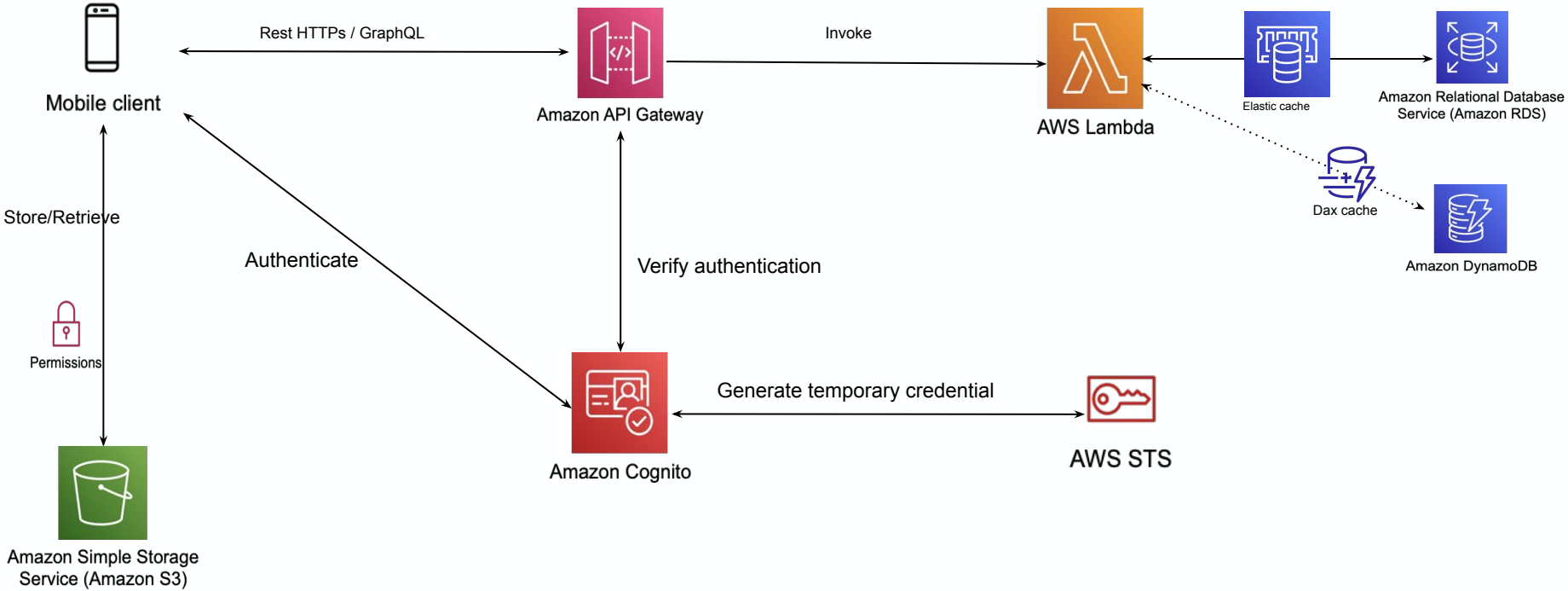


Giving user access to s3

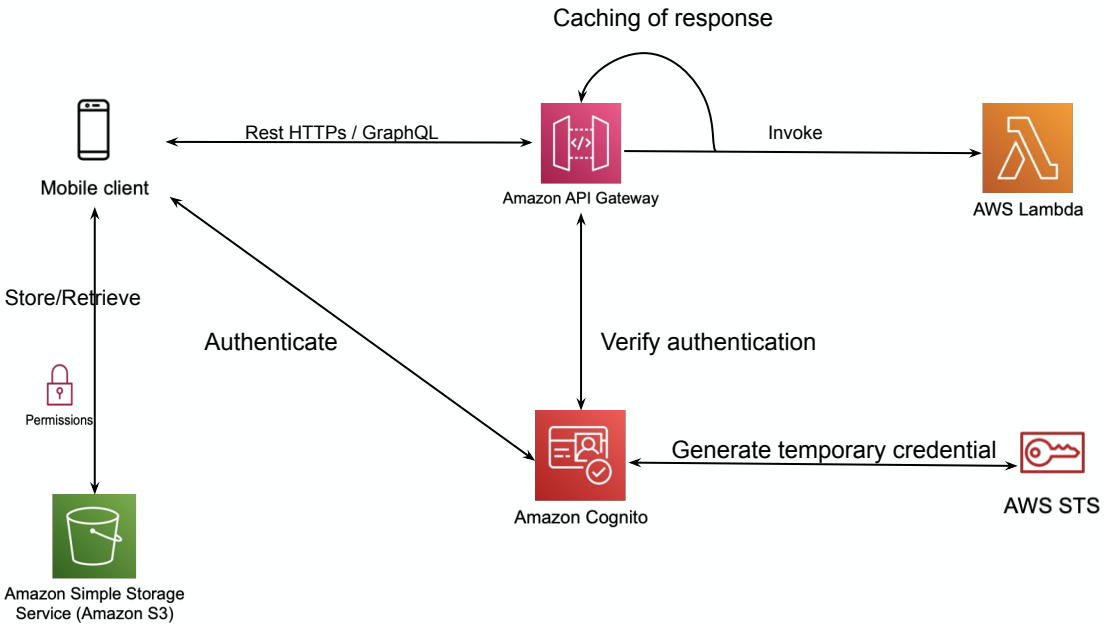


/slash

Reduce load off of databases for read intensive workloads

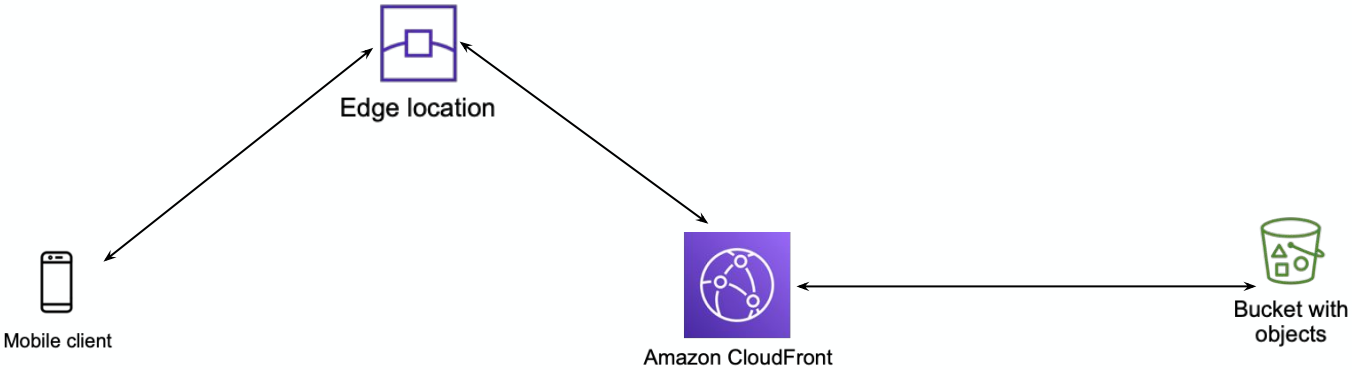


Caching at API gateway



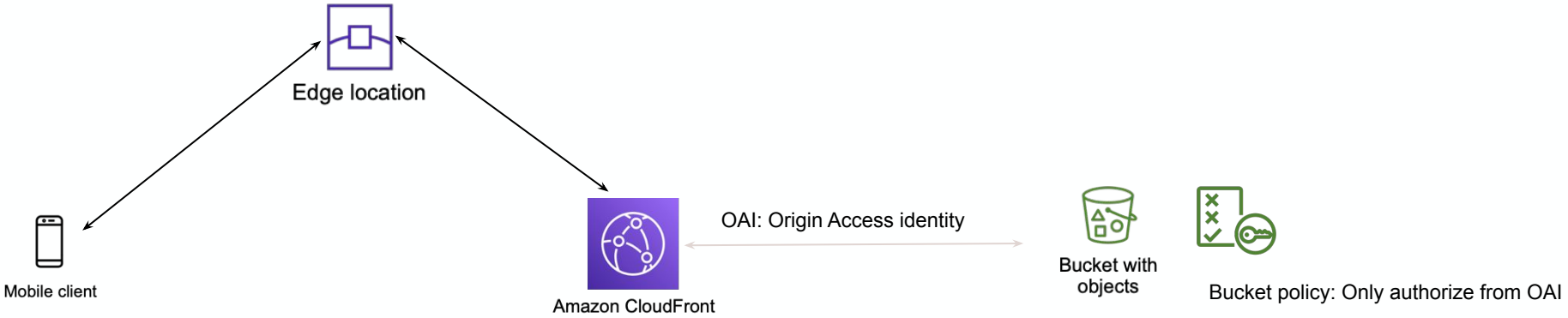
/slash

Serving videos content globally

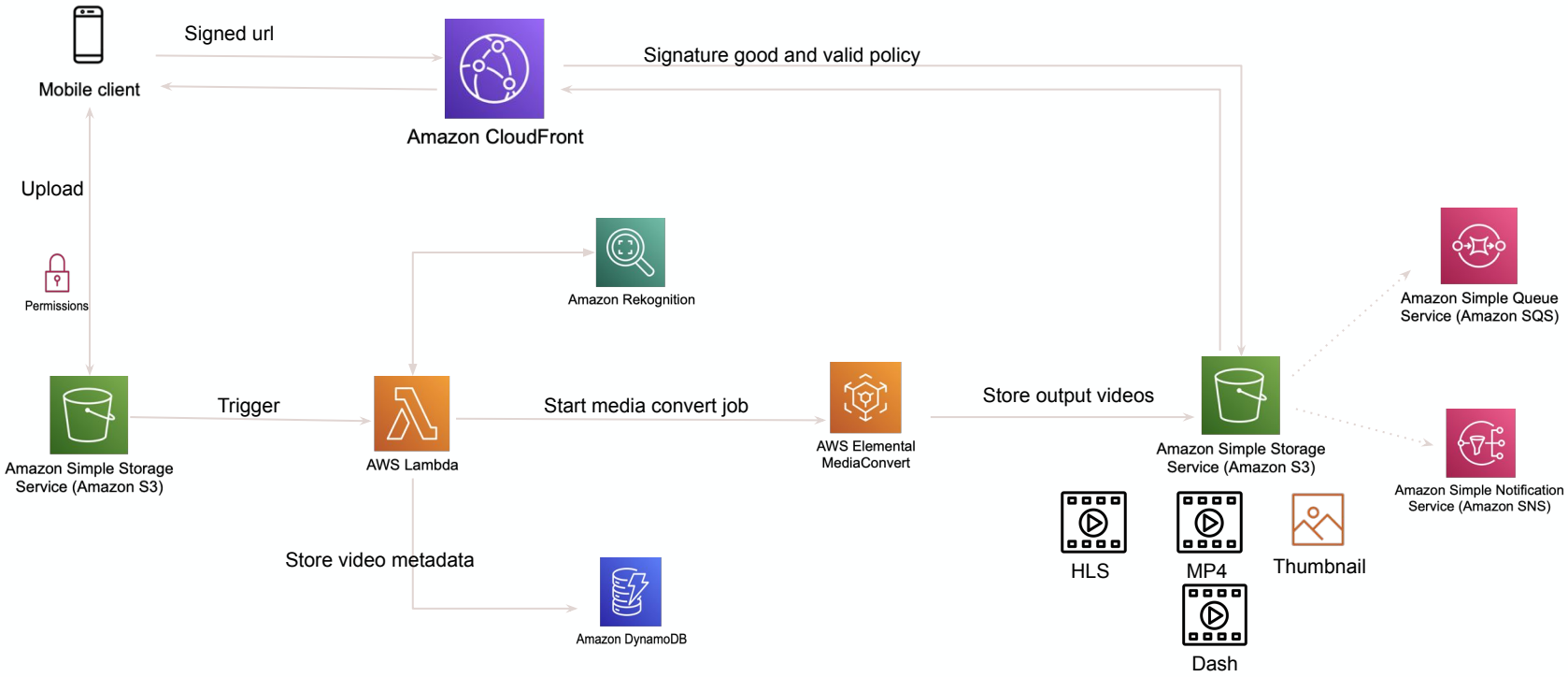


/slash

Serving static videos content globally: More secure

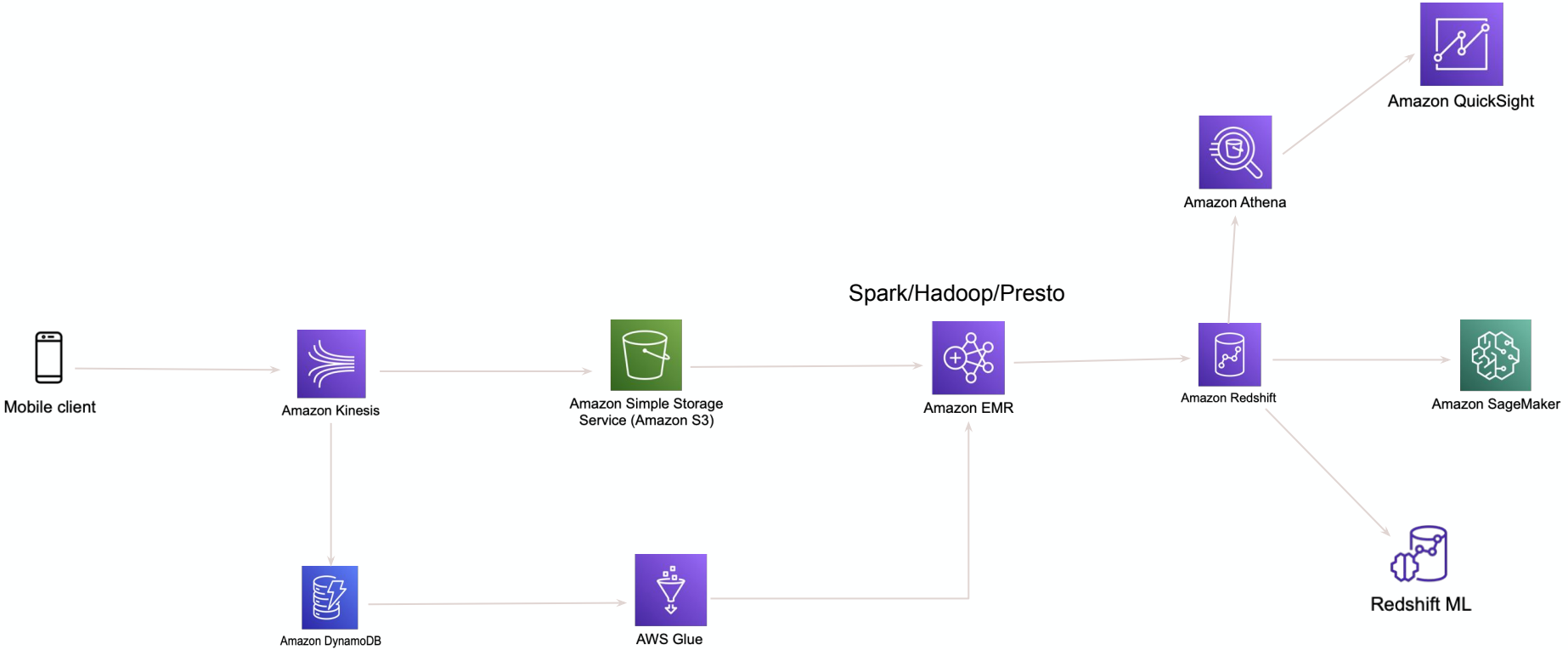


Compression, thumbnail and apply watermark on video content



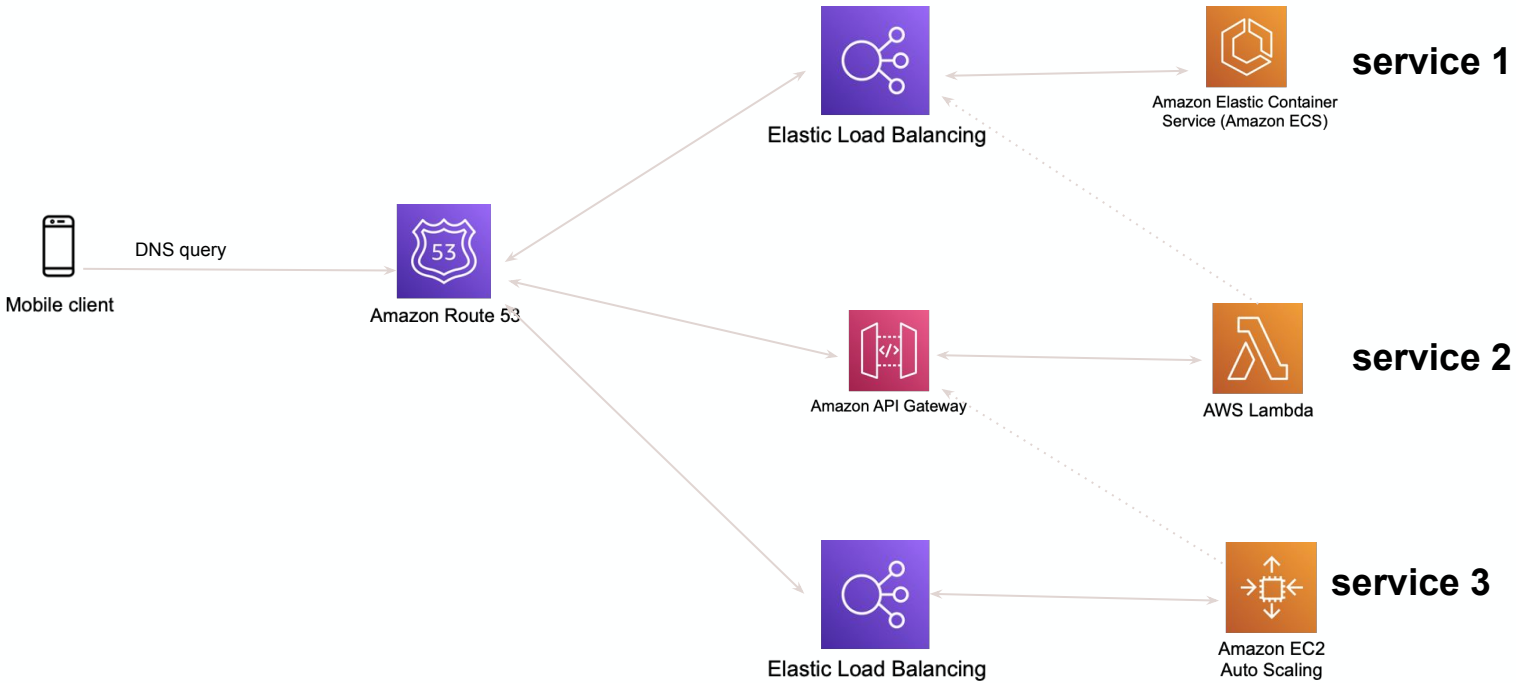
/slash

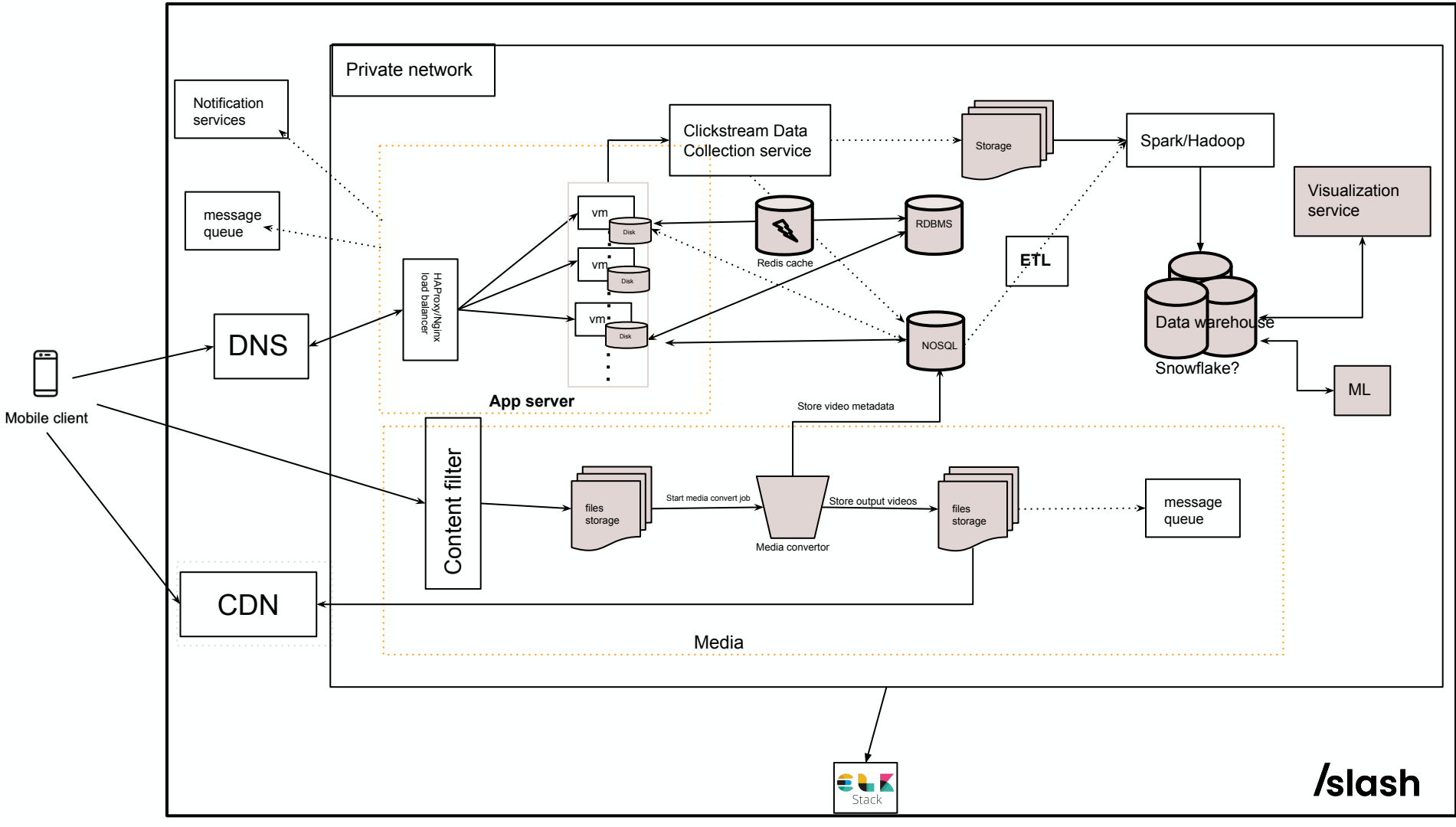
Clickstream data collection service

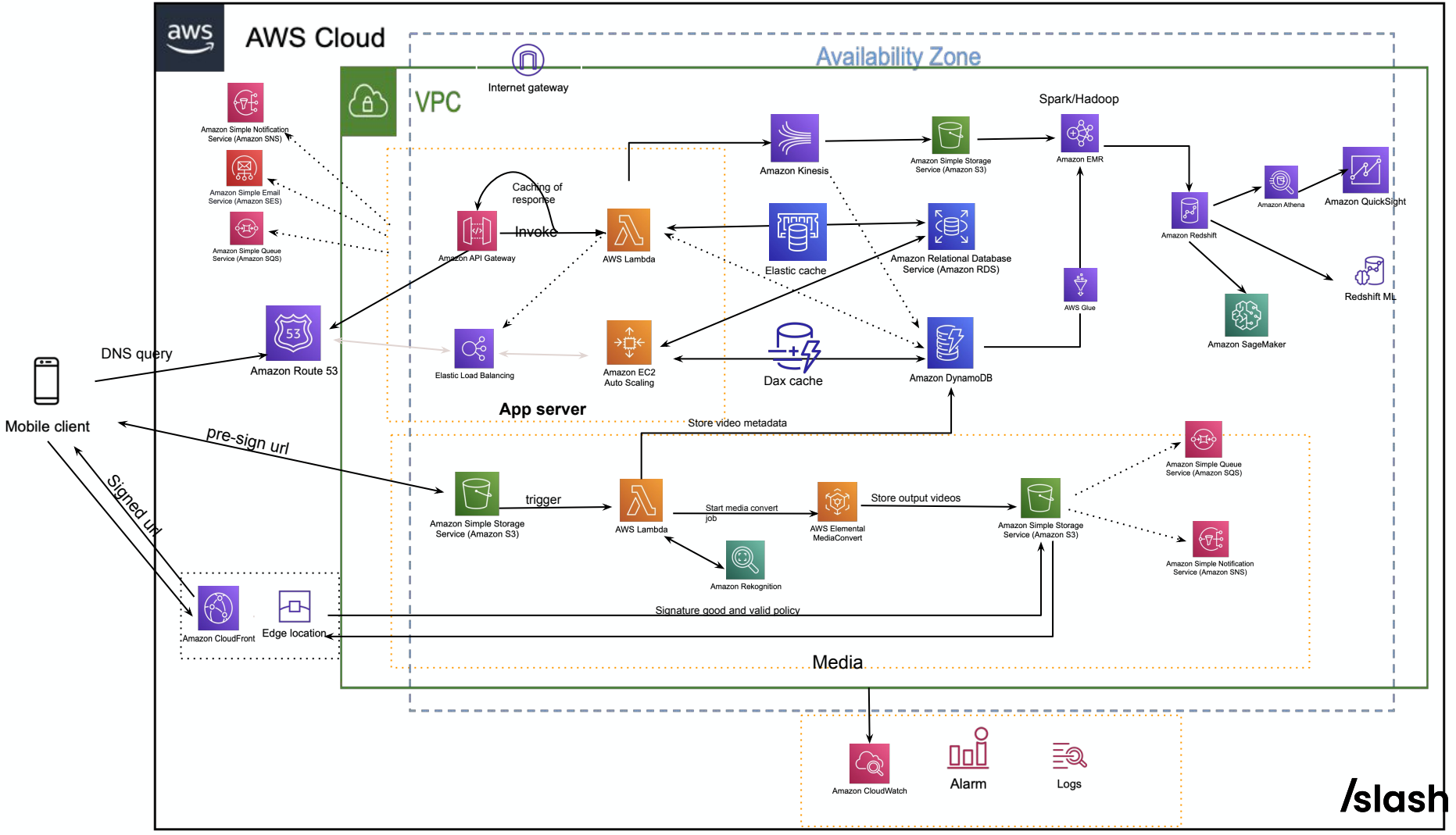


/slash

Microservices with scale independently




Private network
Notification services
message queue
DNS
Mobile client
CDN
HAProxy/Nginx load balancer
vm
Disk
App server
Clickstream Data Collection service
Redis cache
RDBMS
NOSQL
Storage
Spark/Hadoop
ETL
Visualization service
Data warehouse
Snowflake?
ML
Store video metadata
Content filter
files storage
Start media convert job
Media convertor
Store output videos
files storage
message queue
Media
Stack
/slash


AWS Cloud
Availability Zone
VPC
Internet gateway
Amazon Simple Notification Service (Amazon SNS)
Amazon Simple Email Service (Amazon SES)
Amazon Simple Queue Service (Amazon SQS)
Caching of response
Invoke
Amazon API Gateway
AWS Lambda
Amazon Kinesis
Amazon Simple Storage Service (Amazon S3)
Spark/Hadoop
Amazon EMR
Amazon Athena
Amazon QuickSight
Amazon Redshift
Redshift ML
Amazon SageMaker
AWS Glue
Elastic cache
Amazon Relational Database Service (Amazon RDS)
DNS query
Amazon Route 53
Elastic Load Balancing
Amazon EC2 Auto Scaling
Dax cache
Amazon DynamoDB
App server
Store video metadata
Mobile client
pre-sign url
Signed url
Amazon Simple Storage Service (Amazon S3)
trigger
AWS Lambda
Start media convert job
AWS Elemental MediaConvert
Store output videos
Amazon Simple Storage Service (Amazon S3)
Amazon Simple Queue Service (Amazon SQS)
Amazon Simple Notification Service (Amazon SNS)
Amazon Rekognition
Signature good and valid policy
Amazon CloudFront
Edge location
Media
Amazon CloudWatch
Alarm
Logs
/slash

With AWS cloud

Fast, flexible, secure, and budget-friendly solution

/slash

Thank you!

Plug and play together!

/slash