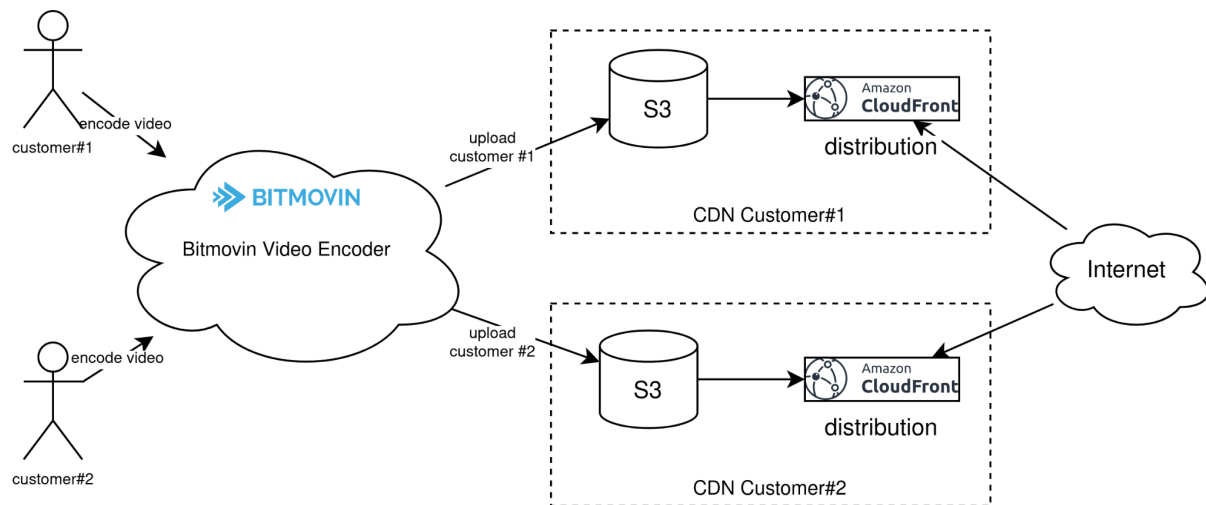


# Context

Bitmovin provides its customers with a preconfigured CDN that is integrated in our Encoding product so that customers can onboard quickly, i.e., encode their videos and immediately distribute them to their own end customers for viewing.

We don't provide our own CDN but integrated [AWS CloudFront](#) and give each customer a [distribution](#) that is backed by an [S3 bucket](#) (hidden from customers). When customers want to upload a video to their CDN our encoder will - behind the scenes - upload the content to S3 bucket and the distribution will automatically distribute the content globally.



For each customer we store in the database information about their setup which looks somewhat like this:

customer_id	bucket_id	distribution_id
customer#1	bucket#1	distribution#1
customer#2	bucket#2	distribution#2

With this information we can identify where we have to upload content to for each customer and which AWS resources (s3 bucket and distributions) they are using.

## Problem

### Overview

We want to bill our customers for their distribution usage. We will invoice customers for every GB of data that is transferred over the CDN to their own end-customers. This billing data should be updated at least once per day and customers should be able to see the amount that will be on their next bill at most 24 hours later.

Currently our customers are billed at the end of the month. To prevent unpleasant surprises (huge bills that customers might not be able to pay) or fraud (customers without intent to pay at all) we want to automatically cap the usage of the CDN if it exceeds the following limits:

- 100 GB within 15 minutes
- 500 GB within 3 hours

Additionally we want to make this usage available to customers via a REST API so that they can check what has been consumed so far within a certain period of time.

## Billing

This is an existing system that can already handle messages with the following syntax:

```
{
  "type:" "EVENT_CDN_USAGE"
  "payload": {
    "id": "<identifier>",
    "customerId": "...",
    "startPeriod": "<start timestamp>",
    "endPeriod": "<end timestamp>",
    "trafficUsageGb": "<traffic within period in GB>"
  }
}
```

The task is to send this message at least every 24 hours with the usage since the last time this usage has been exposed.

## AWS Usage Data

We can get usage data for CDNs in two different ways:

- AWS billing data via [AWS Cost Explorer](#)
- [AWS CloudWatch](#)

AWS billing data is about 24h delayed but this is the most correct one and should be used to bill our own customers and provide the final usage data.

AWS CloudWatch only gives you data for the last 3 hours and can be slightly incorrect, which should not matter too much for some workflows but makes it unreliable for billing. This data can be used for short-term workflows and to provide some usage estimation to customers.

# Task

- Create a [Spring Boot](#) application
- Provide a REST API that exposes usage data for every customer for a time period with a resolution of 24 hours (i.e., one data point per 24 hours). Every request is - by our system - guaranteed to contain a header `X-Customer-Id` that allows us to distinguish between customers.
- Regularly retrieve the usage data (up to you how to implement that) from AWS and provide
  - Usage data to the billing service at least once every 24h
  - Usage data to the customers with a delay of 15 minutes
  - Usage capping based on short-term usage data
    - Limits are 100 GB within 15 minutes and 500 GB within 3 hours
    - If exceeded
      - Disable the CDN (can be simulated by setting a status)
      - Publishing an event (up to you to design it) to the message broker
- Note that our services are running in a high availability environment so we will have it deployed more than once. If you do not have time to fully implement a HA mode be prepared to discuss how you can implement this in a future iteration.
- Also consider how you want to add Logging and - especially - Metrics to the system to make it easier to observe it.