CloudCV

Ankita Singh, Nimisha Srinivasa, Tanuj Mittal

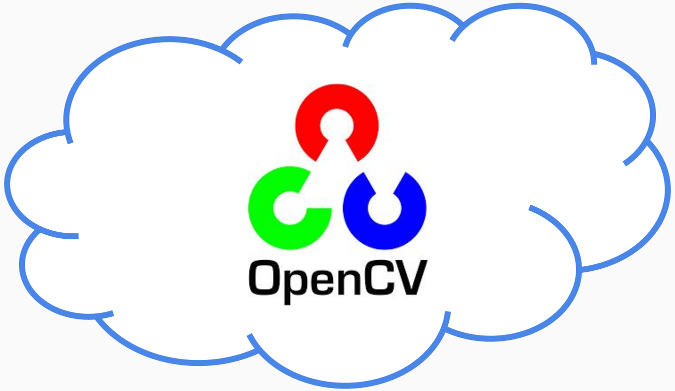
CS 293B

UCSB Spring 2016

CloudCV

- OpenCV-as-a-Service
- On-demand Image Manipulation Service

Goal: A Highly Available and Scalable service



API

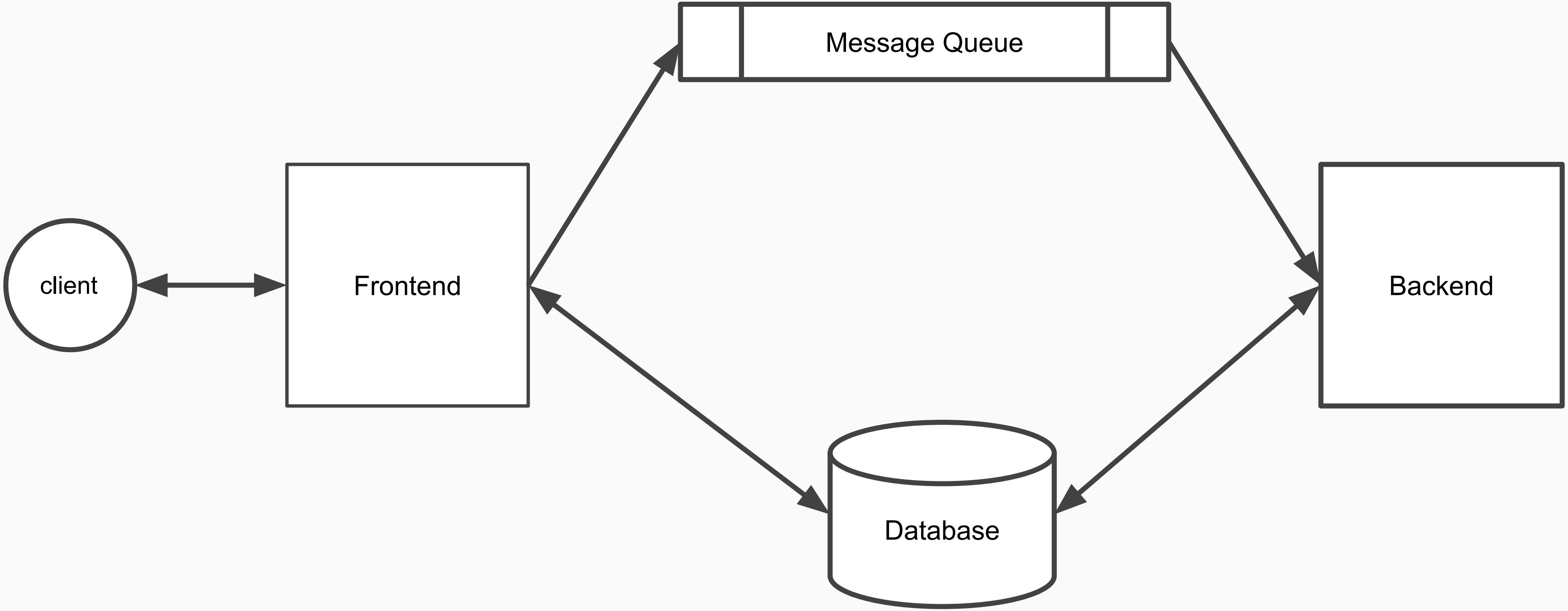
POST

```
 /api

{
        "src": "source_image_url",
        "filters": [
                {
                        "type": "grayscale"
                },
                {
                        "type": "pyramid_down"
                }]
}
```

GET

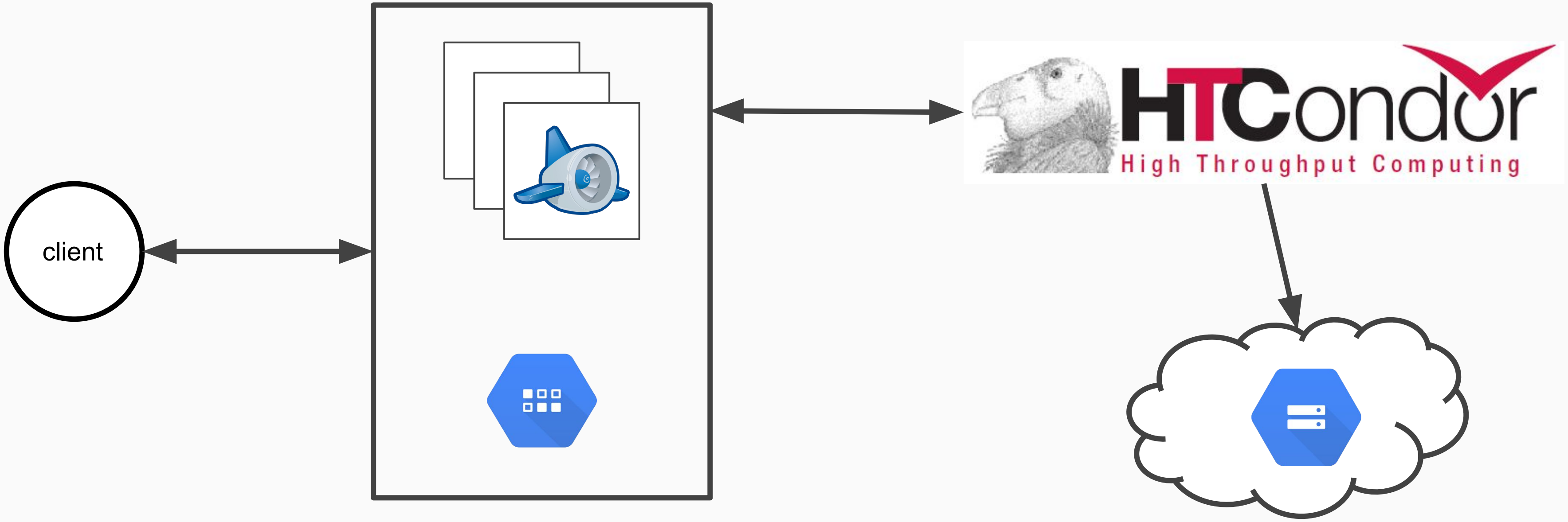
```
/api/<id>

{
        "status": 0,
        "res": "result_image_url"
}
```

Design



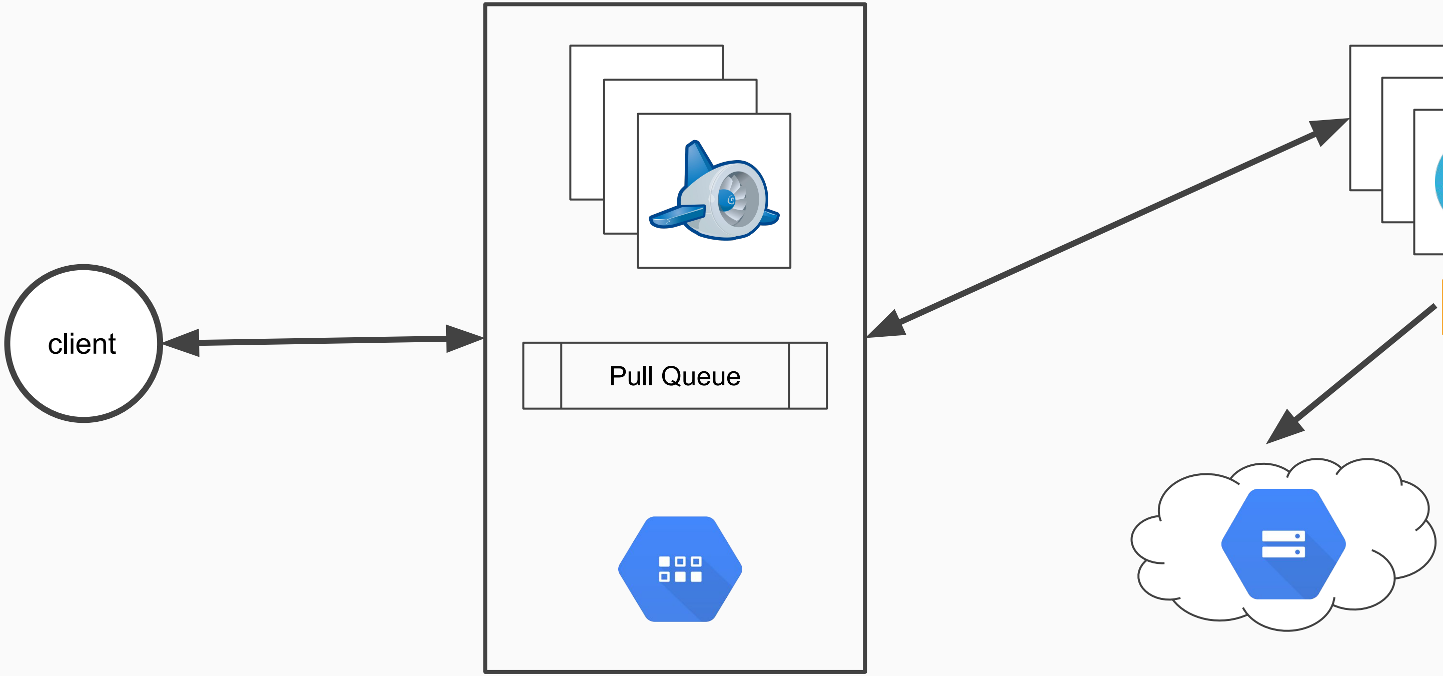
Implementation

Version 1



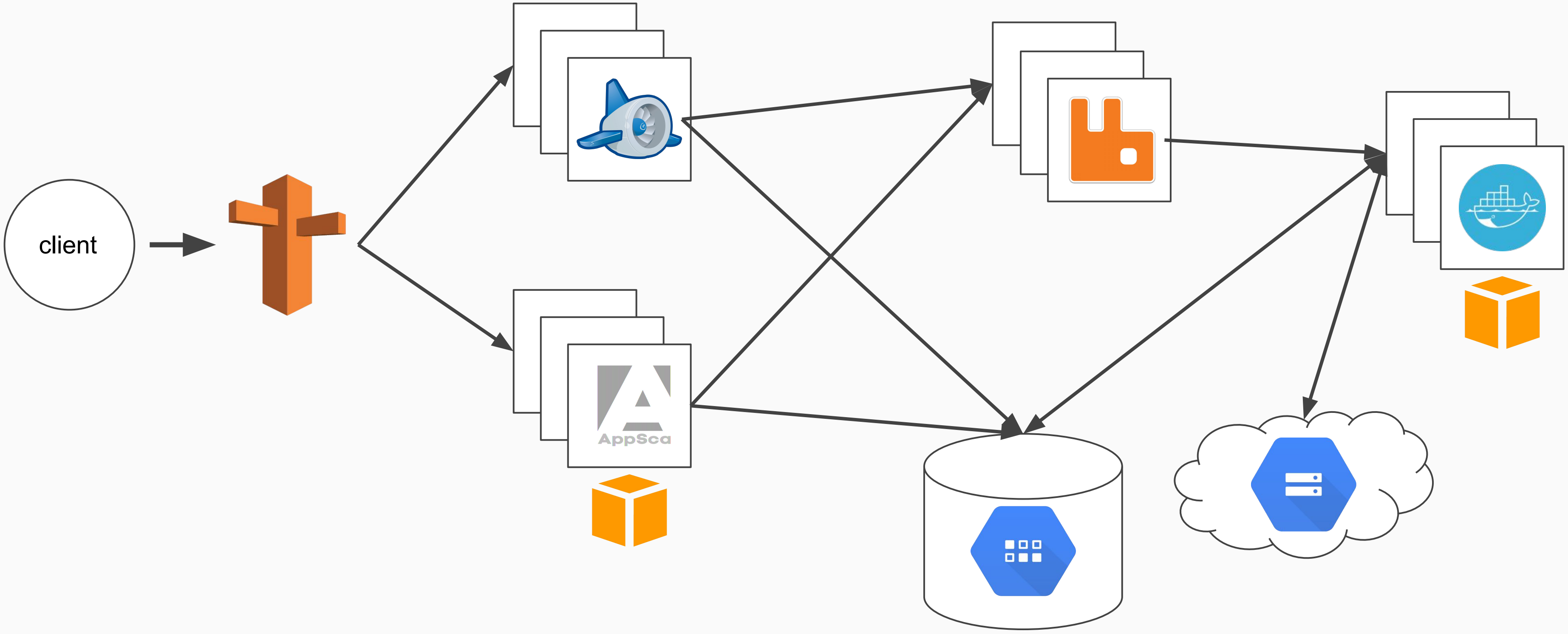
- Difficult to setup Java Universe for OpenCV
- Required setting up all dependencies every time a job is started

Version 2



- No failovers
- If GAE goes down, frontend & pull queue goes down too

Version 3: Final



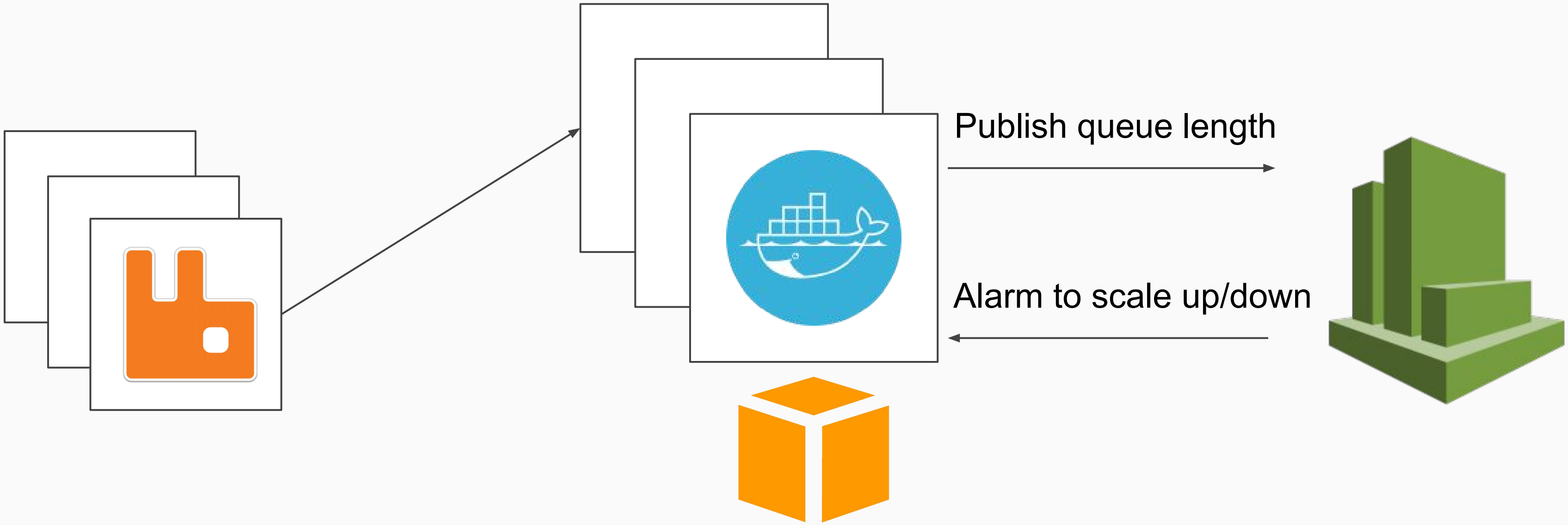
Scaling

Scaling

- Frontend
  - Depends on number of requests and resource utilisation
  - Auto scaling by GAE and Appscale (on AWS)

- Backend
  - Depends on queue length
  - Custom cloudwatch metrics, alarms and autoscaling groups

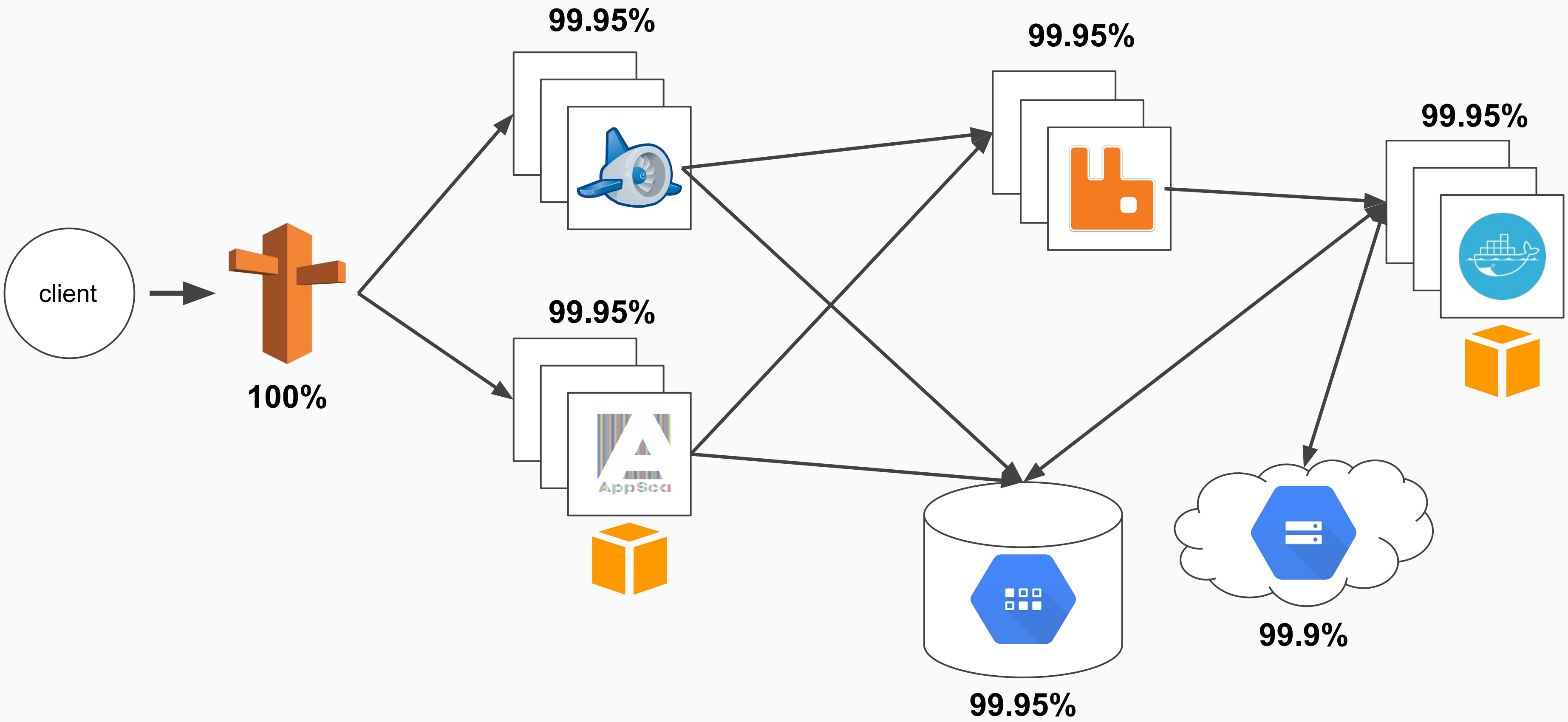
Scaling



Publish queue length

Alarm to scale up/down

Let’s talk about SLA


client
100%
99.95%
99.95%
AppSca
99.95%
99.95%
99.95%
99.9%

Read Availability

SLA for just frontends = 1 - (0.0005 * 0.0005) = 99.999975%

SLA for GCD = 99.95%

SLA* = 99.949975%

Bottleneck: GCD

*Not really, if we consider the result url

Write Availability

SLA for just frontends = 1 - (0.0005 * 0.0005) = 99.999975%

SLA for GCD = 99.95%

SLA for rabbitmq = 99.95%

SLA = 99.900000024994%

Bottlenecks: GCD and Rabbitmq

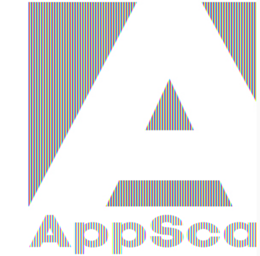
Demo

Future Work

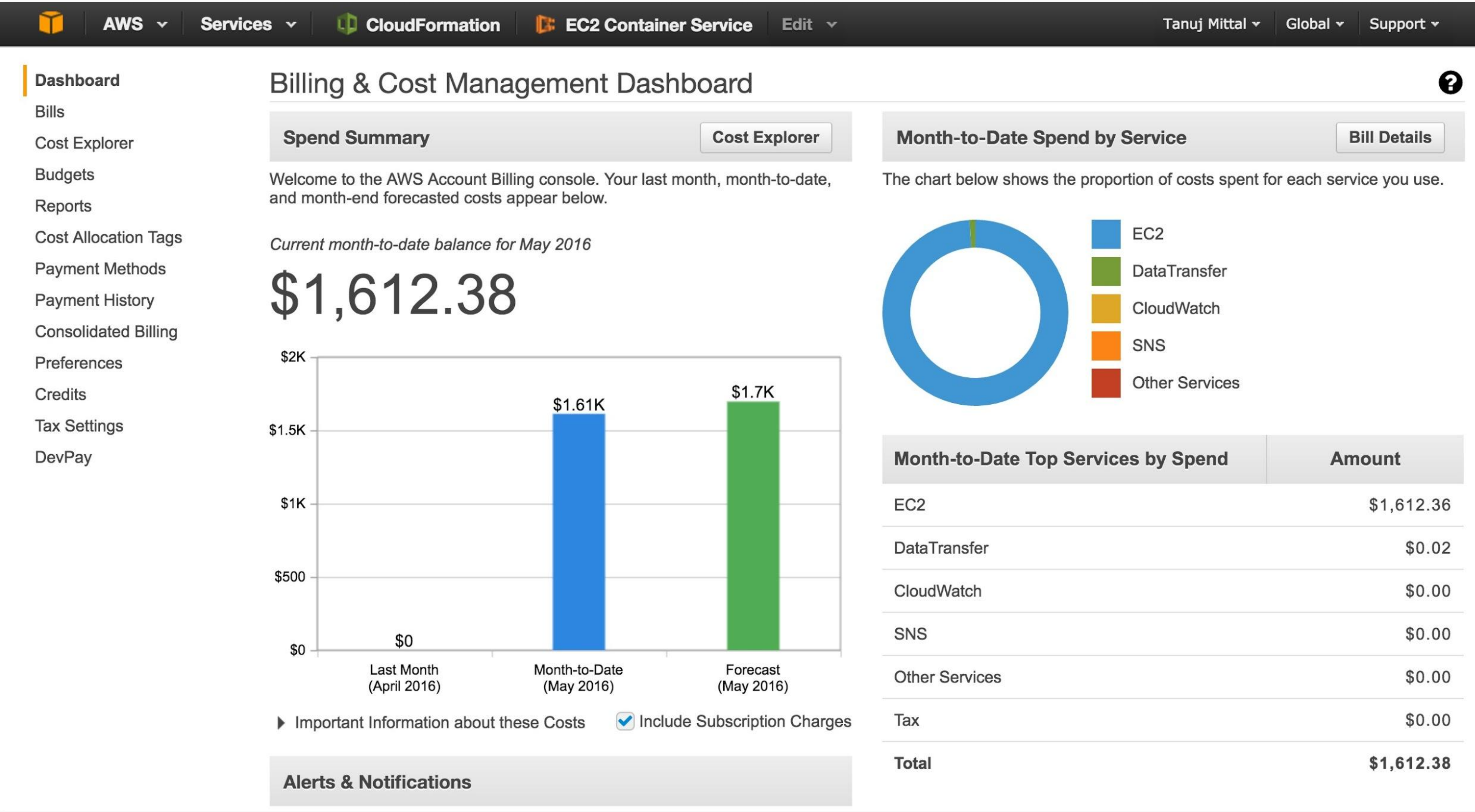
- Multiple Availability zones
- Adding failover support for Google Cloud Storage
- Rabbitmq cluster on different cloud infrastructures?
- Integrate more features from OpenCV
- Allow uploading images

Learnings

- Service Oriented Architecture is great for guarantees
  - We used Database aaS, rabbitmq aaS, Storage aaS, Platform aaS…
- Making everything work together is hard
  - Outdated documentation, unmaintained APIs
- Monitoring is essential
- Learnt new technologies...


AppSca

HTCondor
High Throughput Computing



Billing & Cost Management Dashboard

Spend Summary

Welcome to the AWS Account Billing console. Your last month, month-to-date, and month-end forecasted costs appear below.

Current month-to-date balance for May 2016

$1,612.38

▸ Important Information about these Costs

☑ Include Subscription Charges

Alerts & Notifications

Month-to-Date Spend by Service

The chart below shows the proportion of costs spent for each service you use.

Month-to-Date Top Services by Spend	Amount
EC2	$1,612.36
DataTransfer	$0.02
CloudWatch	$0.00
SNS	$0.00
Other Services	$0.00
Tax	$0.00
Total	**$1,612.38**

Thank You!