

Chitrakosh

14.06.2022

Raj Keshri

8210247321

Akatr
BOI CSP

Simri Bakhtiyarpur, BR 852127

Overview

High Quality photos and video sharing application accessible across web and android.

Functional Requirements

1. User Sign Up / Login features
2. Search User features
3. Follow and followers features
4. Photo and video upload feature.
5. User Feed / Photo Feed

Non Functional Requirements

1. Highly Available
2. Low latency
3. Highly scalable
4. Eventually consistent

Specifications

Application Stack

1. UI :- HTML, CSS, JavaScript
2. Backend :- Python
3. Framework:- Django
4. Database:- RDBMS

Cloud Component

Usages of GCP services /component to design system components.

Milestones

I. High Level Design Decision

Deciding upon overall architecture, microservices and associated system component related to various microservices.

II. Low Level Design Decision

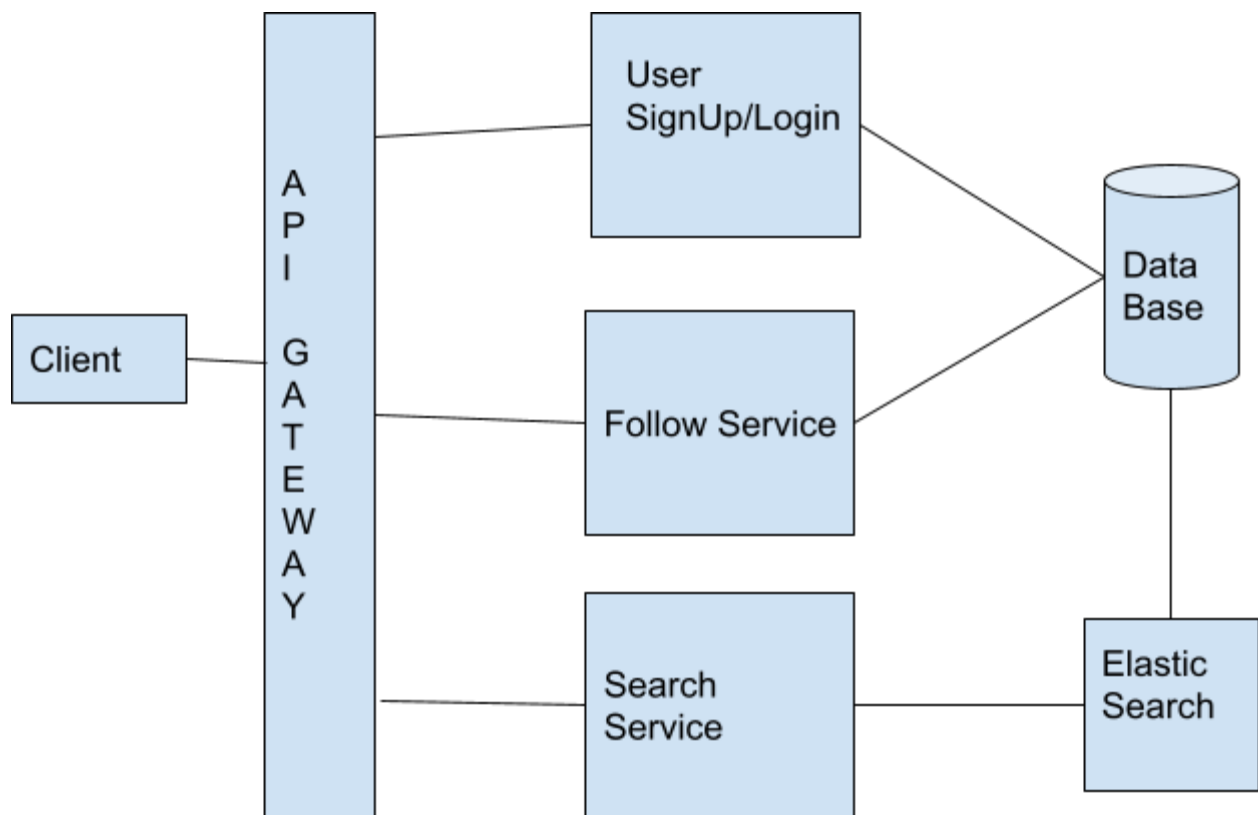
Efficient use of suitable design pattern and following SOLID principle ,OOPS principle to create a functional, extensible code. Use of multithreading , Socket Programming when needed.

III. Implementation

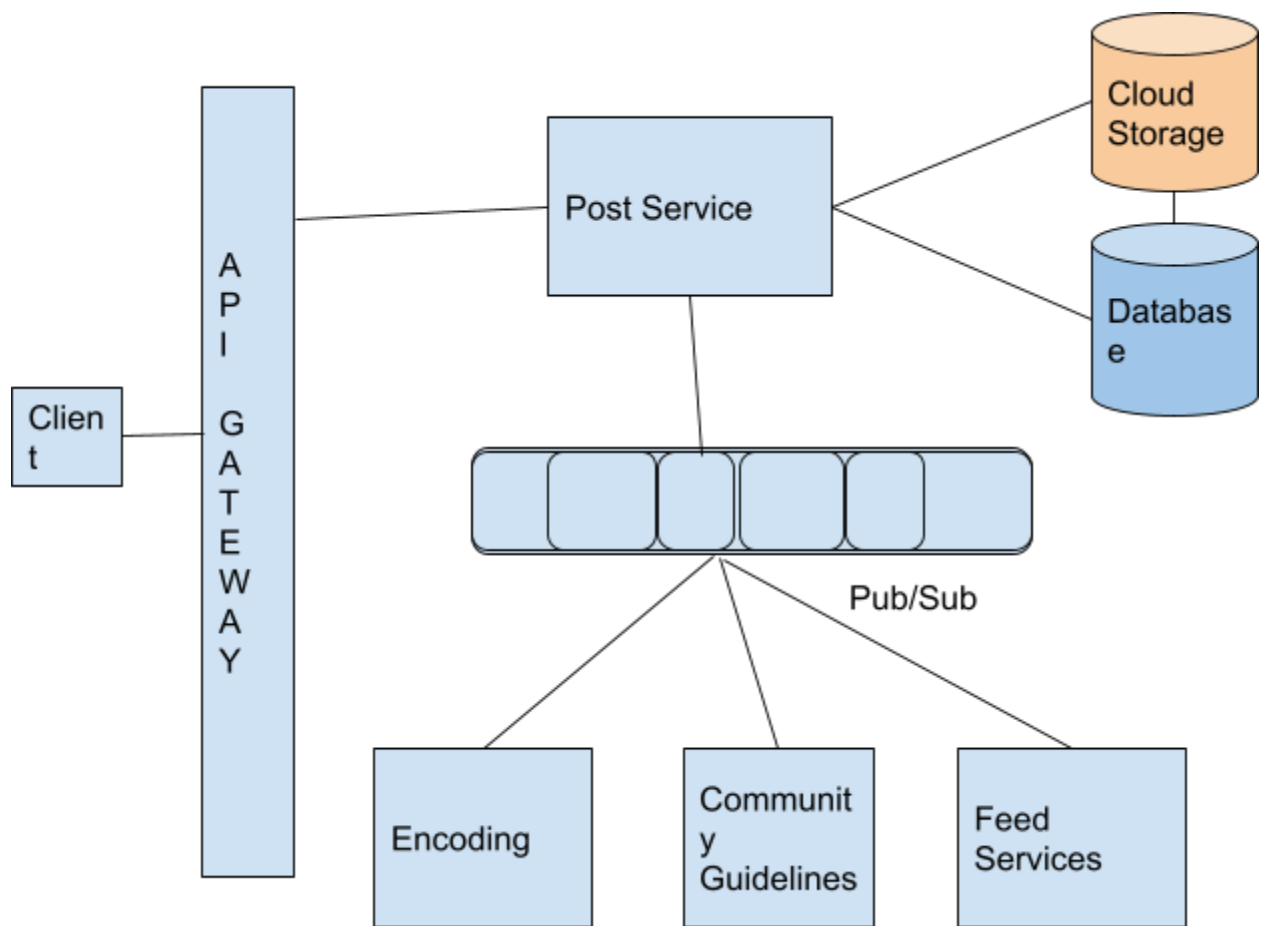
Implementation of abstract classes using python /Java/Kotlin.

High Level Design (microservices wise)

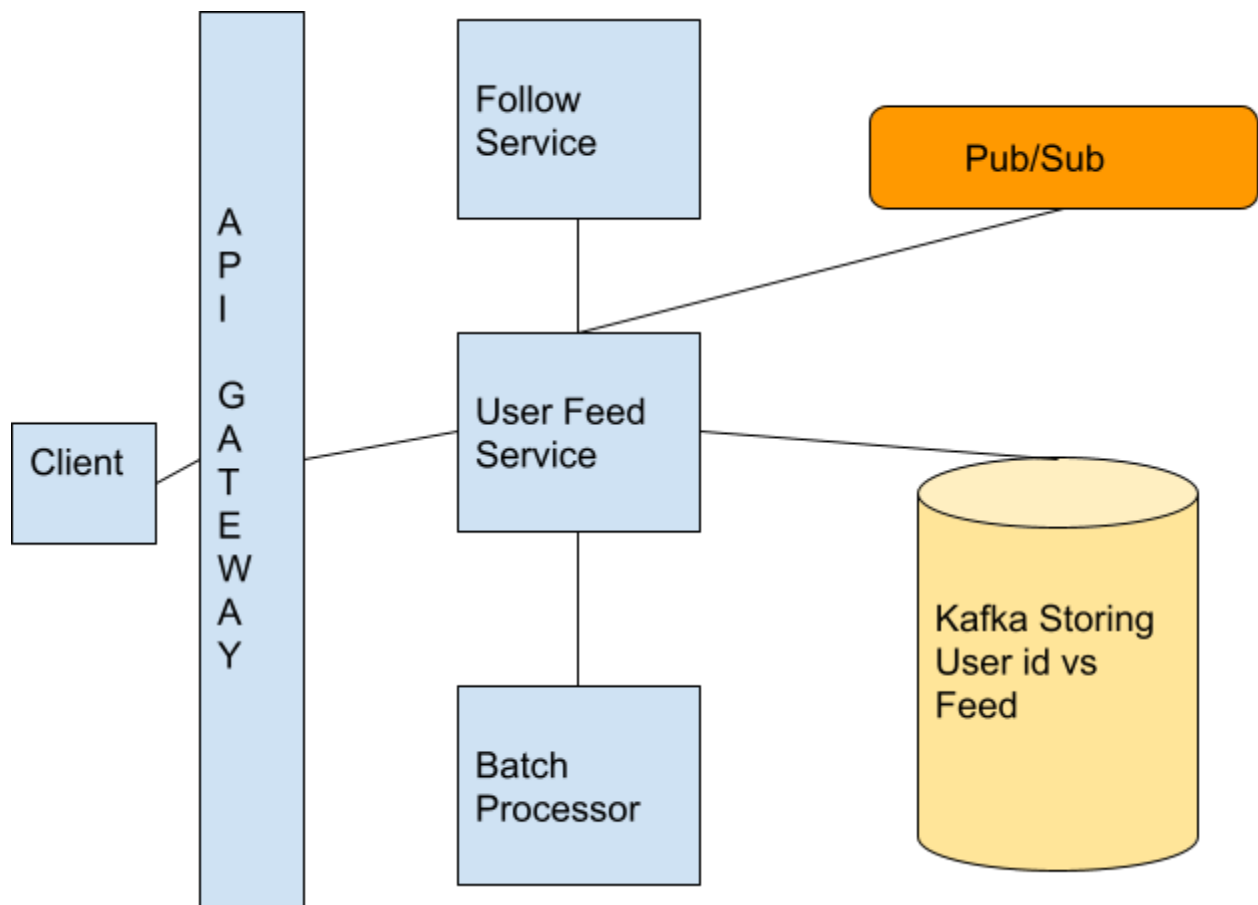
User SignUp/Login , Search User & Follow User Service



Post Photo/Video Service



User Feed Service



Service / Component	Cloud Solution
User SignUp/Login Service	Container Kubernetes
Follow Service	Container Kubernetes
Search Service	Container Kubernetes
API Gateway	APIgee
User DataBase	Cloud SQL
Elastic Search	** Kibana
Post Service	App Engine
Pub/Sub	GCP Pub/Sub
Feed Service	Container Kubernetes

Batch Service	Container Kubernetes
Community Service	Big Query
Kafka	** to be asked

Flow Diagram:-

Low Level Design (microservices wise)

User SignUp

class User:

```
    current_id = 0
```

```
    @staticmethod
```

```
    def gen_id():
```

```
        User.current_id +=1
```

```
        return User.current_id
```

```
    def __init__(self,email,phone,name,id,pwd):
```

```
        self._email= email
```

```
        self._phone= phone
```

```
        self._name = name
```

```
        self._id = User.get_id()
```

```
        self._pwd = pwd
```

Login

```
def authenticate(pwd):
```

```
    Pass
```

```
def addPhoto(url):
```

```
    pass
```

```
def change_name(newname):
```

```
    pass
```

```
def change_email(newemail):
```

```
    pass
```

```
def change_phoneno(newphone):
```

```
    pass
```

```
def changepwd(newpass):
```

```
    pass
```

```
def delete(user):
```

```
    pass
```

```
def deactivate(user):
```

```
    pass
```


Search User

```
def search_user_by_name(name):  
    pass
```

```
def search_user_by_id(id):  
    Pass
```

```
def blockuser(user_id):  
    pass
```

Follow User Service

```
def follow_user(id):  
    pass
```

```
def apporveuser(user_id):  
    pass
```

```
def removeuser(user_id):  
    Pass
```

```
def myfollowerlist():  
    pass
```

```
def followinglist():  
    pass
```

