TODO:

Choose proper architecture pattern for Instagram services
Find what will be the bottleneck in each case
Find what might be a SPOF in each case

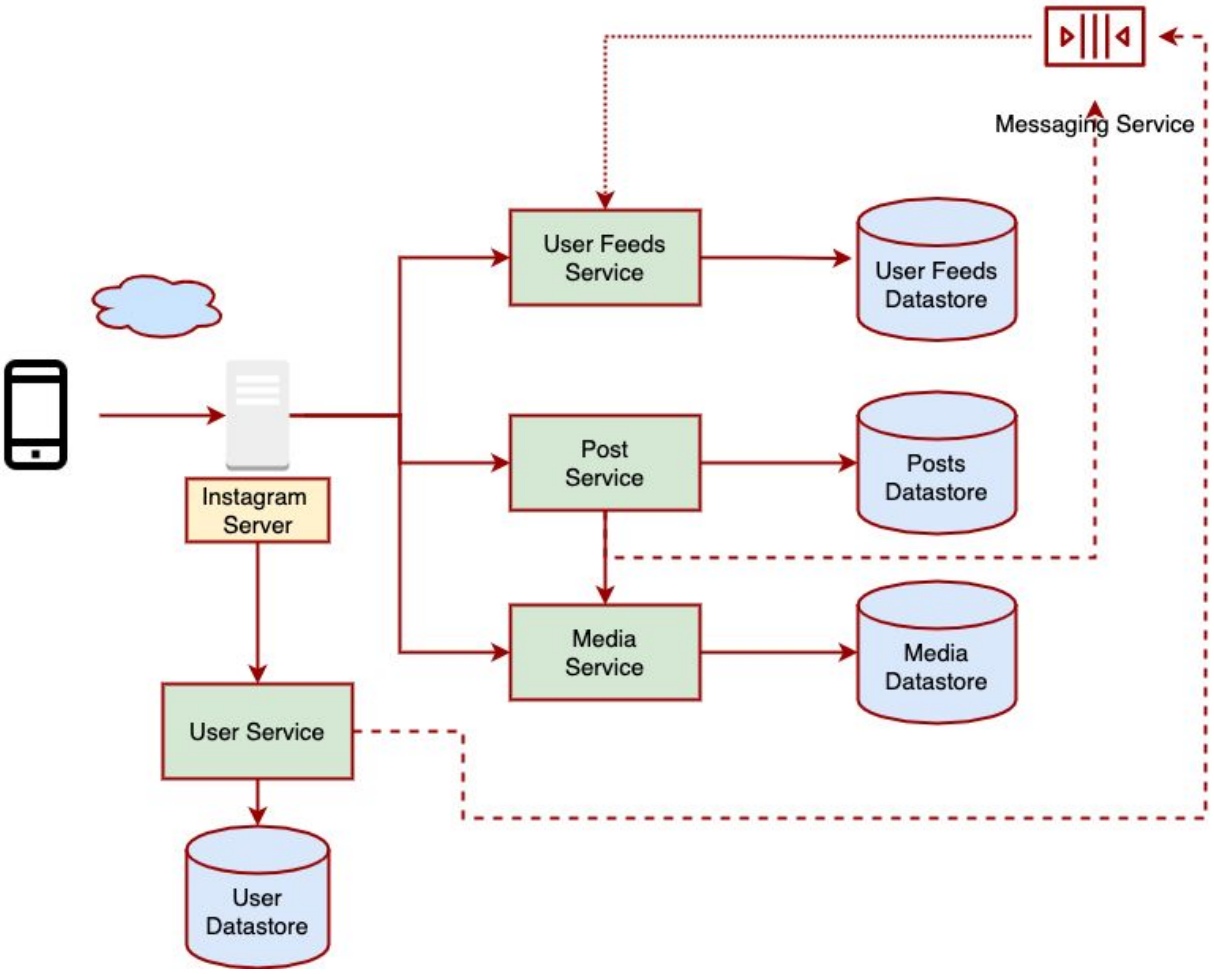
Instagram requirements:

1. User upload images from mobile client
2. Users follow other users
3. User see news feed with content
4. Scallability: 1 billion users a month

Resources estimations:
1 billion users a month
2 photos per month
2 mb per photo

1000000000 * 2 * 2 = 4000000000 mb = 4000 tb (data per month)

High-level Architecture

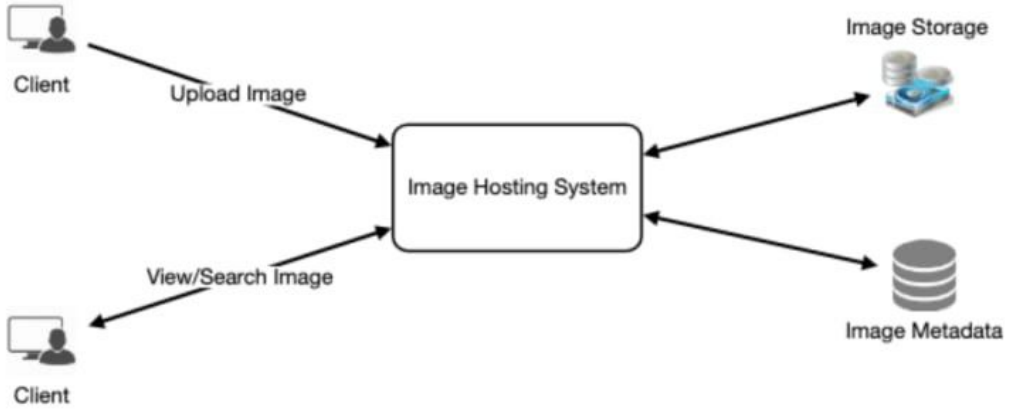


Components

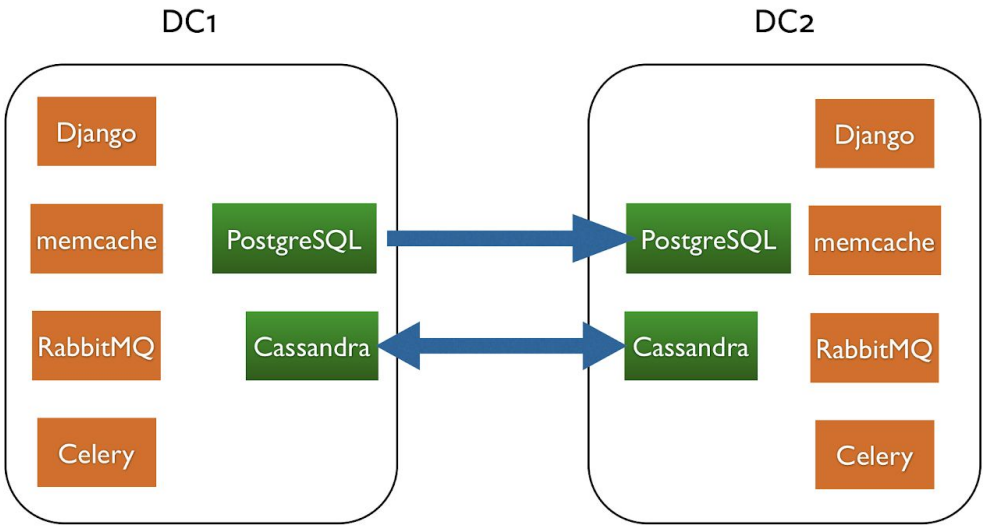
The following are the key system components for our platform:

- User Service: Provides API to manage users and interactions where a user follows or unfollows another user. A separate User Follower Service can be defined instead of making the follow/unfollow use case part of the User Service itself.
- Posts Service: Provides API to allows various to be created and stored in the data store. Also, it handles the ability for a user to add comments and like of posts as well. The Post Service can further be broken down into Post Comments and Post Like Services if required.
- Media Service: Provides API to allows storage and retrieval of photos and videos from its underlying data store.
- Messaging Service: The messaging service is a bus or a steaming event service. The various services like User Service, Posts Service, etc. are produces and post events like user liked a photo, user followed another user, etc. to relevant topics. On the other hand, various consumer services like User News Feed subscribe to these topics and take action based on the ents published by other services.

Media datastore considerations:



The actual Instagram stack



Possible Bottlenecks

1. Storage: due to the enormous amount of data, there would be a bottleneck for handling such datasets. Following patterns should be observed:
- Use additional object storage for content and a relational database for metadata + caching + CDN
- Use sharding
2. Load balancing of incoming read/writes
3. Caching, the actual story about handling caching on Instagram scale https://instagram-engineering.com/thundering-herds-promises-82191c8af57d

Possible SPOF

1. Infrastructure issues:
   DNS
   CDN
   Networking
   Compute
2. Data storage

Credits:
https://www.codercrunch.com/design/634265/designing-instagram#mcetoc_1dv10vl8s18
https://www.8bitmen.com/instagram-architecture-how-does-it-store-search-billions-of-images/