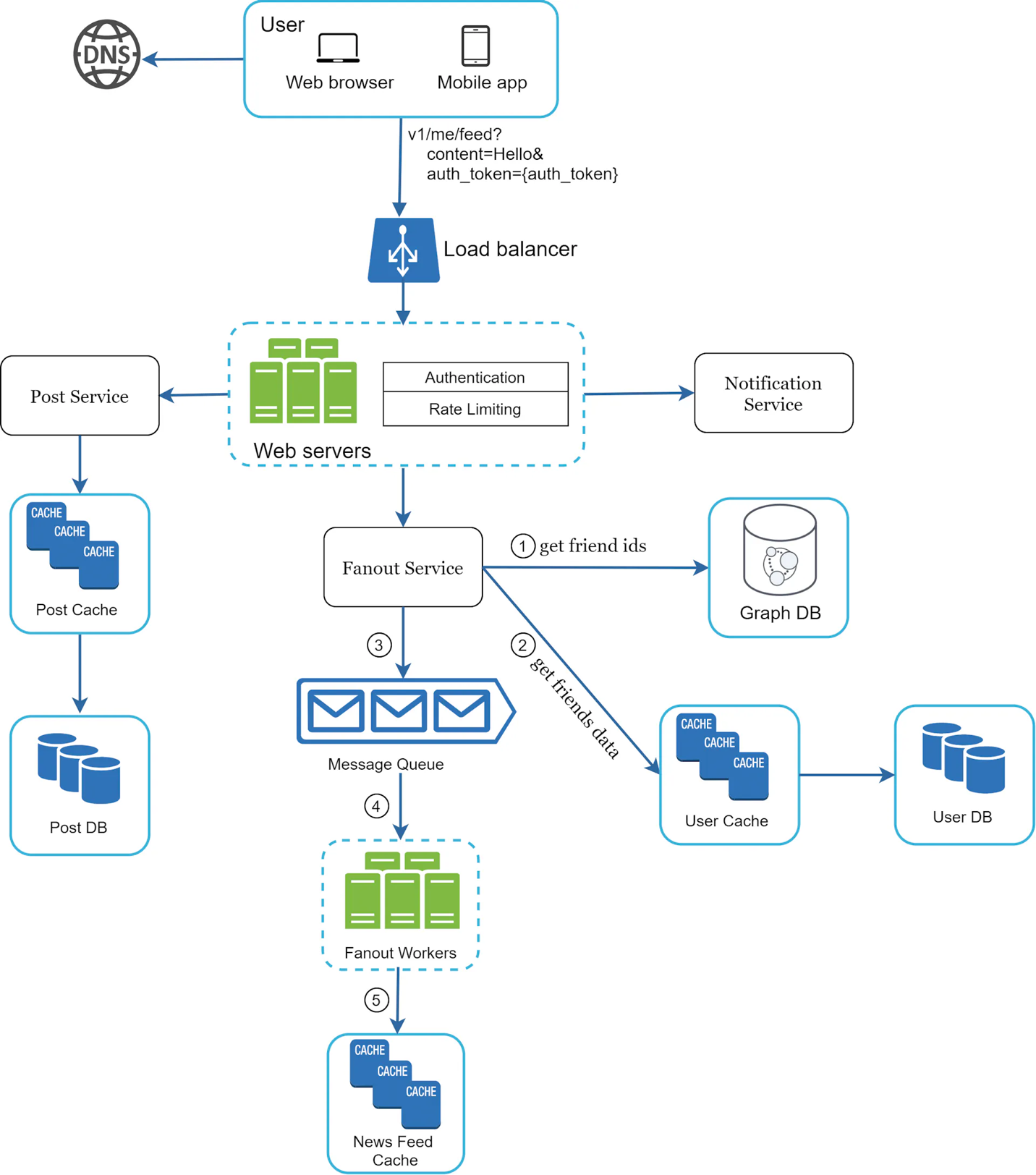
First part of tis question :

was

What to add for posting content POST /v1/me/feed

What to add for news retrieval?



**What's a fan out service?**

Basically in our case, process of delivering a post to all the friends that we have

A fan-out system is a method of distributing data to multiple consumers in a highly efficient and scalable manner. One example of a fan-out system in action is on the popular social media platform, Twitter.

When a user tweets, the tweet is broadcast to all of the user’s followers in near real-time. This is an example of a fan-out system because the tweet is distributed to multiple consumers (the user’s followers) simultaneously.

**What are the fan out on write approach and fan out on read here?**

**0. Fan out on write: (push model)**

**pro:**  With this approach, news feed is pre-computed during write time. A new post is delivered to friends’ cache immediately after it is published.

Con:

* If a user has many friends, fetching the friend list and generating news feeds for all of them are slow and time consuming. It is called hotkey problem.

Type 2 fan out here:

1. **Fanout on read** (pull model)

The news feed is generated during read time. This is an on-demand model. Recent posts are pulled when a user loads her home page.

Cons:

* Fetching the news feed is slow as the news feed is not pre-computed.

What's the ideal solutino here?

So basically we decide to use a hybrid apporahc model for ceebrities (use pull). Use consistent hashing to solve the hotkey problem.

**The structure of the fan out service here?**

**A diagram of a computer network

Description automatically generated**

And then here

**How does the fan out service work? Step by step**

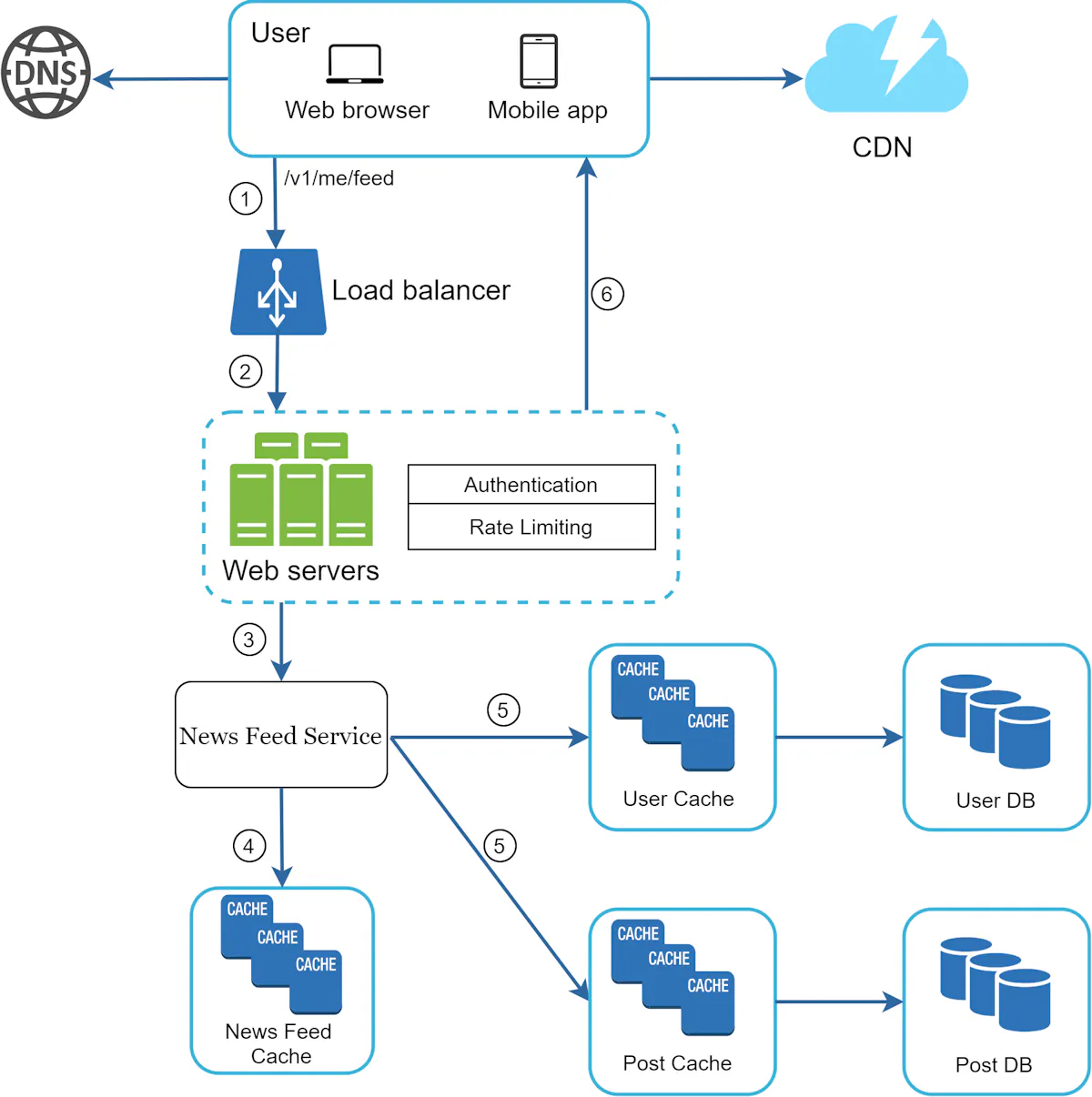
1. Fetch friend ids from the grpah db

2. Get freinds data from the cache in the database (this includes friend lists and post id)

3. Send friends lists and new post ids to the msg q

4. We will then have a <post\_id , user\_id> table that shows, only ids are stored, otherwise get too big

**And then next aprt is the news feeed retrieval service here**



Below is the diagrm for this:

1. And then next we can look for sth else here

2.

How would we load the 3 new tweets here from tiem line as shown in the database?

A blackboard with colorful writing

Description automatically generated

Cache

Need sth push based towards lcient

1. Use websocket to push from server to client here,

**How does it work in the memCache?**

2. It basically means that we need to evict after 5 days of caching in the list above.

3. 100 tweets basically, so not too many here

4. This will then improve our read speed here

**Fan out exchange**

1. Done using kafka or sth at the very end, and then can produce messages here

2. So basically when Mary writes request goes to Kafka and then goes to Bob's read api

3. The read api notifies Bob of new push msg and then Bob gets it here ,

4. Bob reads from the read api here

5. Also important to know what kind of db supprots sharding like above here

Follow up questions

How many read instances do you need from here?

2. How costly is the load balancer going to be to persist websocket connectsions here