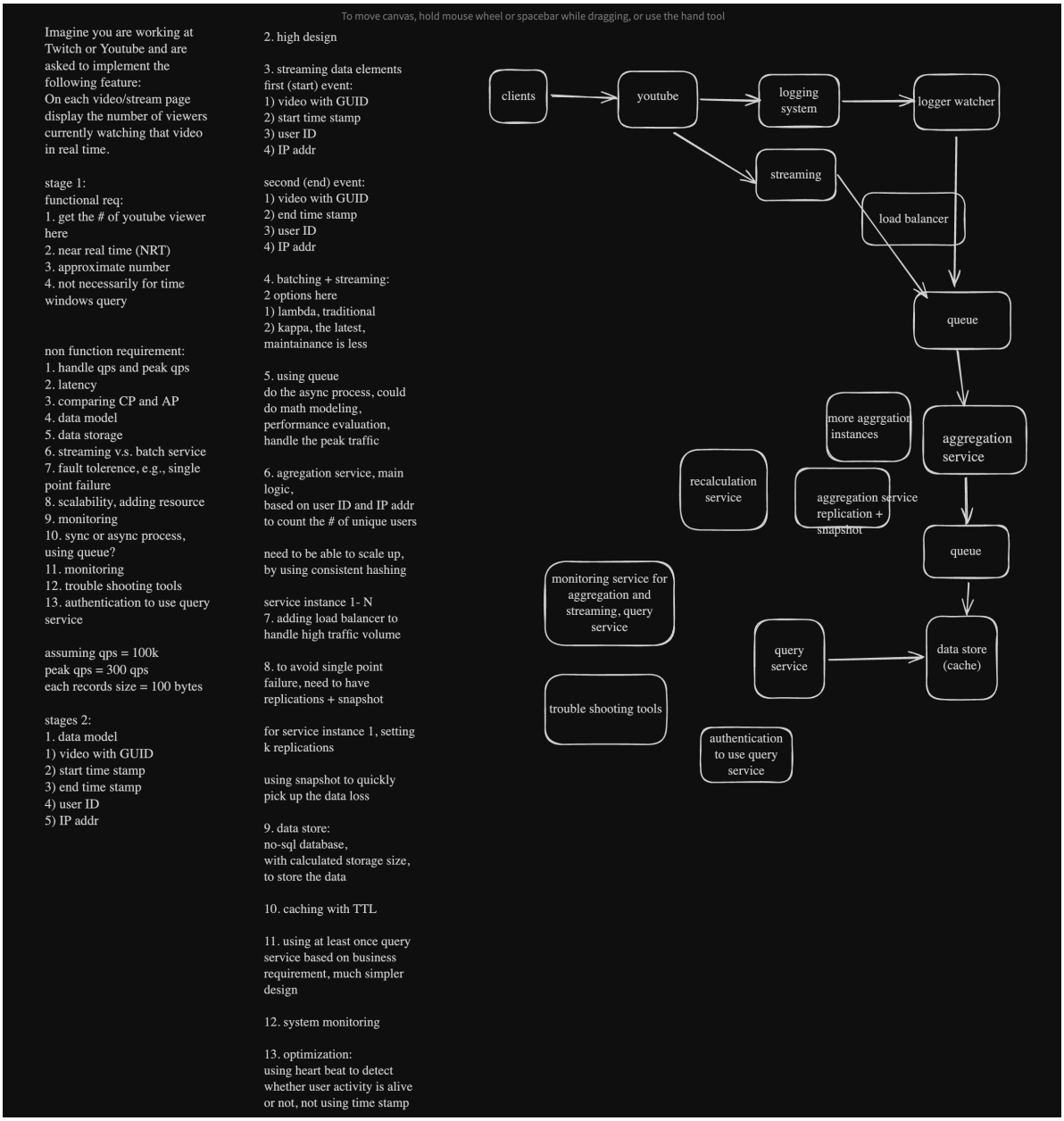
Liftoff system design



Imagine you are working at Twitch or Youtube and are asked to implement the following feature:
On each video/stream page display the number of viewers currently watching that video in real time.

stage 1:

functional req:
1. get the # of youtube viewer here
2. near real time (NRT)
3. approximate number
4. not necessarily for time windows query

non function requirement:
1. handle qps and peak qps
2. latency
3. comparing CP and AP
4. data model
5. data storage
6. streaming v.s. batch service
7. fault tolerence, e.g., single point failure
8. scalability, adding resource
9. monitoring
10. sync or async process, using queue?
11. monitoring
12. trouble shooting tools
13. authentication to use query service

assuming qps = 100k
peak qps = 300 qps
each records size = 100 bytes

stages 2:
1. data model
1) video with GUID
2) start time stamp
3) end time stamp
4) user ID
5) IP addr

2. high design

3. streaming data elements
first (start) event:
1) video with GUID
2) start time stamp
3) user ID
4) IP addr

second (end) event:
1) video with GUID
2) end time stamp
3) user ID
4) IP addr

4. batching + streaming:
2 options here

1) lambda, traditional
2) kappa, the latest, maintainance is less

5. using queue
do the async process, could do math modeling, performance evaluation, handle the peak traffic

6. agregation service, main logic,
based on user ID and IP addr to count the # of unique users

need to be able to scale up, by using consistent hashing

service instance 1- N
7. adding load balancer to handle high traffic volume

8. to avoid single point failure, need to have replications + snapshot

for service instance 1, setting k replications

using snapshot to quickly pick up the data loss

9. data store:
no-sql database,
with calculated storage size, to store the data

10. caching with TTL

11. using at least once query service based on business requirement, much simpler design

12. system monitoring

13. optimization:
using heart beat to detect whether user activity is alive or not, not using time stamp