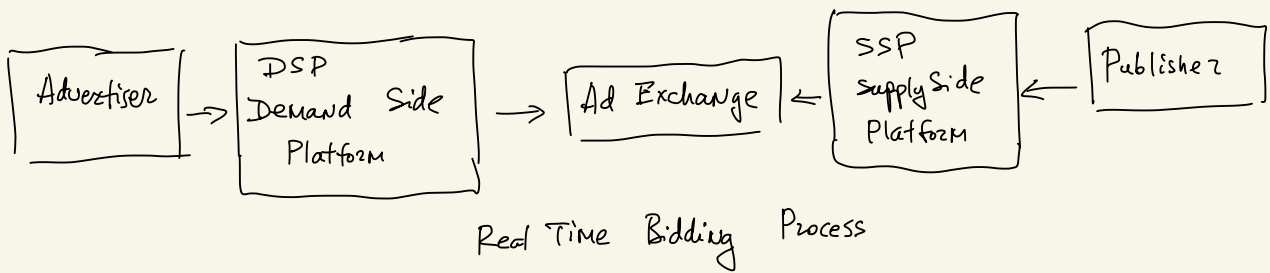


Ad Click Event Aggregation



CTR click through rate

CVR conversion rate

- log file on servers, appended to end of file. ad-id, timestamp, user-id, ip, country,
- 1 bill clicks per day, 2 mill ads. Growth 30% year over year.
- Queries :
 - number of click of the last M minutes
 - top 100 most clicked in the past 1 minute
 - support data filtering
- Edge : events arrive late ; duplicate events ; system down - recovery

FR

NFR :
correctness of the aggregation results
properly handle delayed / duplicate
robustness - resilient to partial failures
latency end-to-end few minutes

EST :
1 bill DAU
1 bill ad click events / day
$$QPS = \frac{10^9 \text{ events}}{10^5 \text{ sec/day}} = 10,000$$

Peak QPS = 50,000 QPS
Storage : 0.1 KB x 1 bill = 100 GB . 3TB monthly.

HLD : API, Data, HL.

Client-Server agreement.

user ^{query} → Aggregation data.

GET /v1/ads/{ad-id}/aggregated-count

Req: from
to
filter

Resp: ad-id
count

return aggregated,
event for given
ad-id

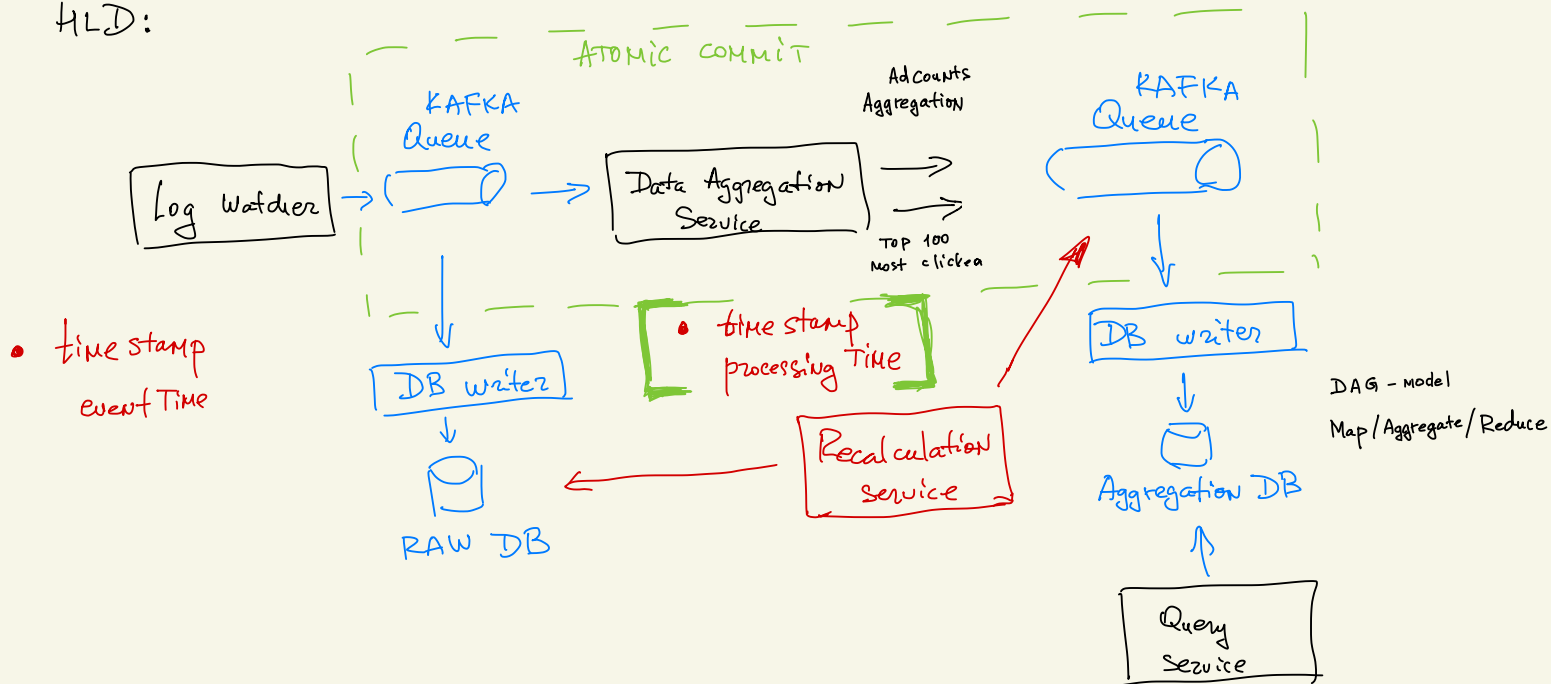
GET /v1/ads/popular-ads

Req: count
window
filter

Resp: ad-ids

- Data: RAW, aggregated (smaller data set)
 - ↳ recovery perspective / backup
- DB: write-heavy, read volume is low.
 - Relational: scaling write challenging.
 - Cassandra, Influx DB. Storing in \$3.

HLD:



DDD:

Stream / Batch processing.

- ↳ backup
- ↳ data arrive, aggregation

} Lambda architecture
"Kappa" architecture

ROI watermark isn't reasonable for low probability events.

Sliding window (specific interval)

Kafka: three delivery semantics:

1. at-most once
2. at-least once
3. exactly once

SCALE :

• Message Queue

Producers +
Consumers rebalancing
during off-peak hours
Brokers: hashing key,
topic sharding

/ • Aggregation Service

Map/Reduce
adding/removing nodes
Deploy to Apache Hadoop

/ • DB.

Cassandra support HZ
consistent hashing