Distributed email service

EDS: 1 bill users

~~Auth~~, send / receive, fetch, filter, search, anti-spam. ~~Attachment~~.

SMTP, POP, IMAP → HTTP

NFR: Reliability / Availability / Scalability / Extensibility

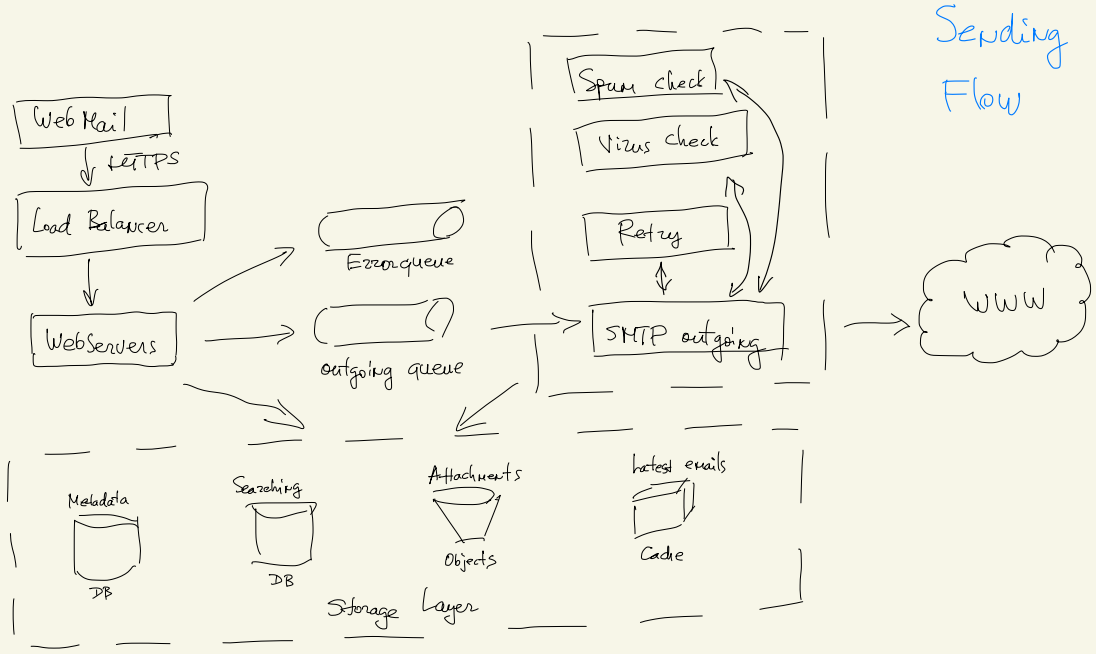
EST: $\frac{10^9 \times 10}{10^5} = 100.000$ QPS

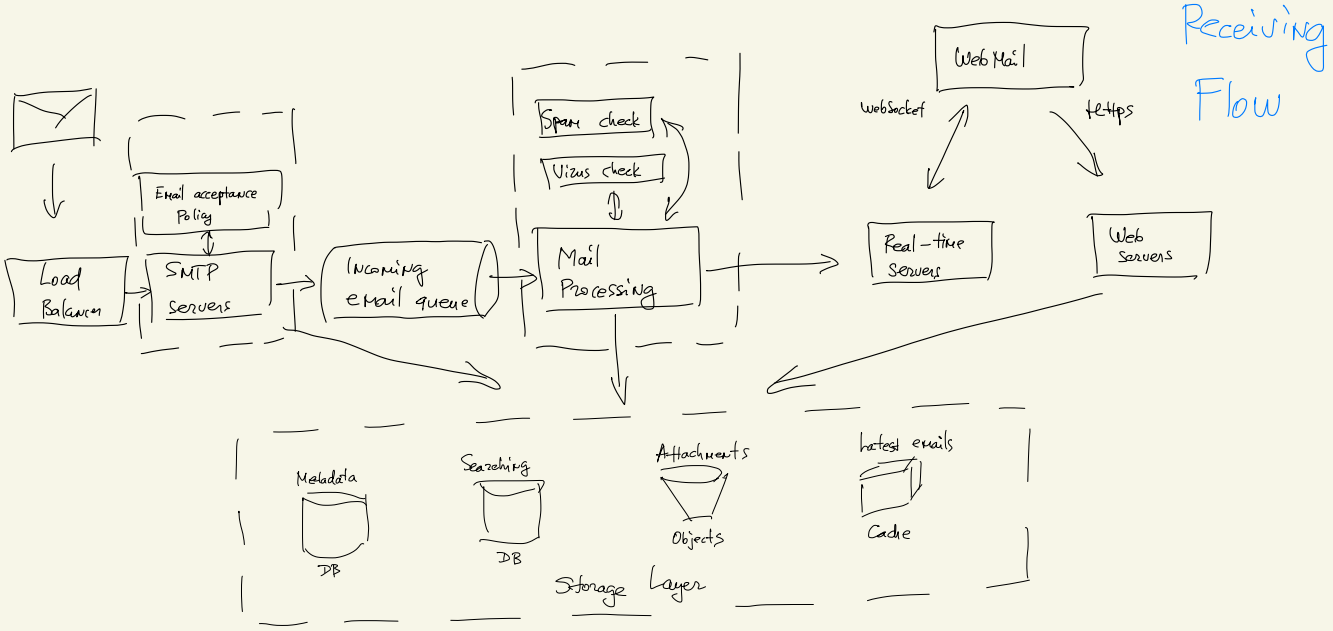
Metadata 1 bill x 40 emails x 365 x 50 KB = 730 PB.

20% contains attachment.

1 bill x 40 emails x 365 x 20% x 500 KB = 1,460 PB.

HLD:





DDD:

- Metadata Database
- $earch
- Deliverability
- $calability

82% of read queries are for data younger 16 days.
Data loss not acceptable.
Highly customized DB. (Reduce disk I/O IOPS)
with easy to create incremental backups.
user_id partition key. Data for one user stored on single shard.

- get all folders for a user.
- display all emails for a specific folder
- create / delete / get an email
- fetch all read or unread emails

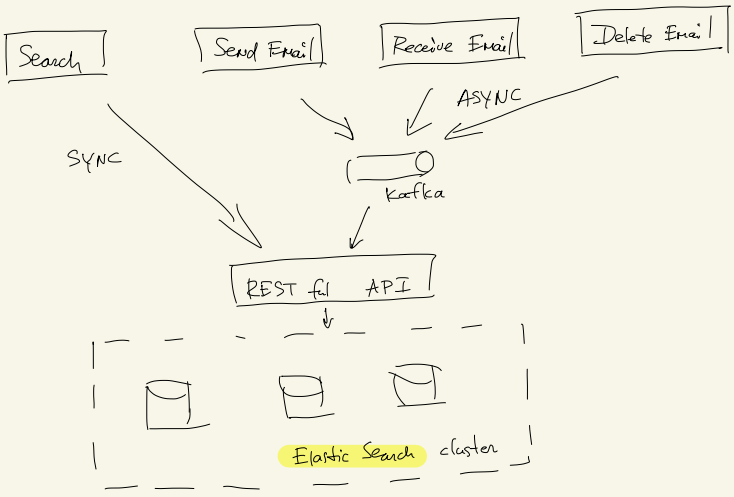
read / unread emails separate tables
Conversation threads : JWT algorithm

! Availability in favor of consistency. Sync/update until failover ends.
CAP - theorem.

Deliverability : spam folder!

- dedicated IP
- classify emails
- email sender reputation
- authentication

Sort by attributes, result accurate.
Elasticsearch / native datastore search. Easy to integrate.



Custom search solution:

LSM (Log-Structured Merged-Tree)
Sequential writes. (as in Cassandra, BigTable, Rock DB).
First emails → to Level ∅ in memory cache.

Availability → data replication. Servers in near DC.

Compliance. Legal regulations.
Europe. PII.
CA. GDPR.

Email security. Fishing protection / email encryption.

Optimization. Email to multiple recipients, some email attachments (S3 Amazon).