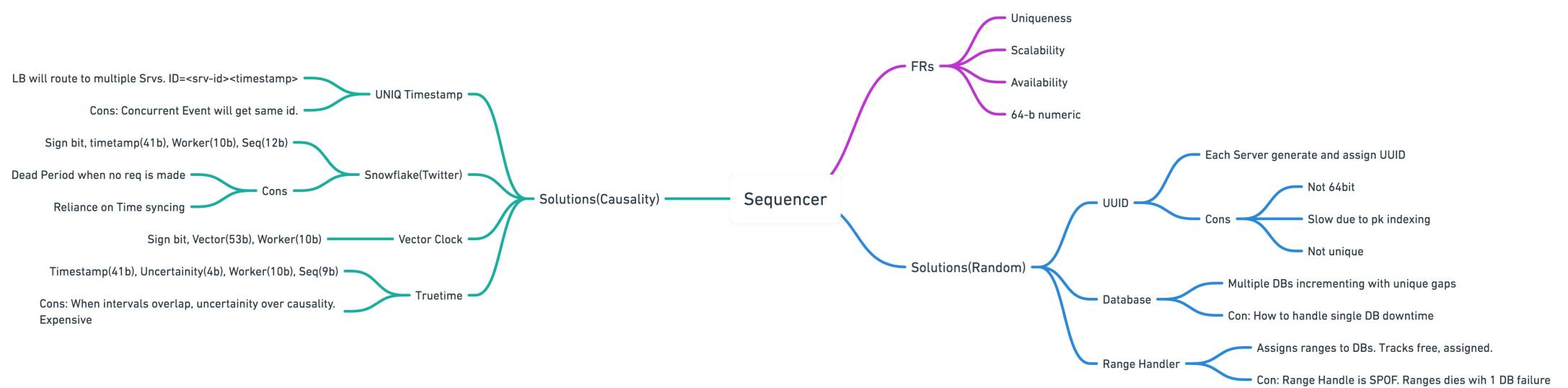
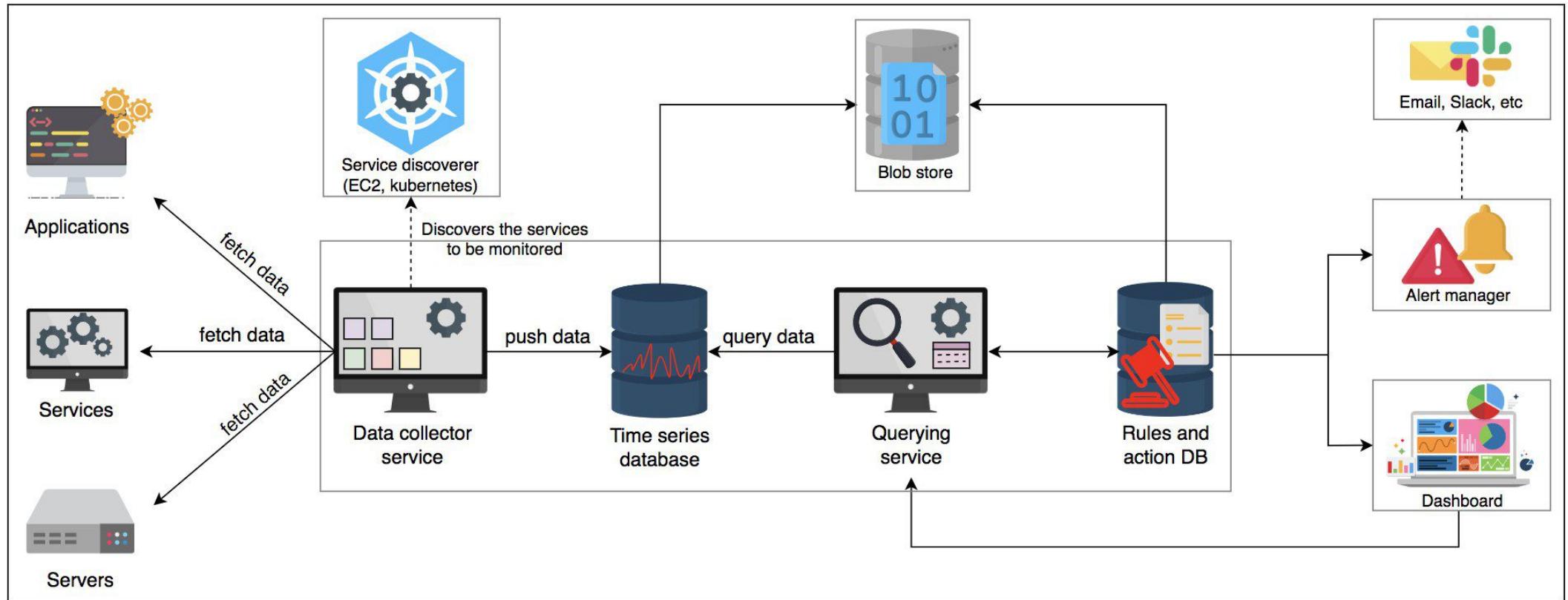


represents periodic activity



represents operation when content is not in the proxy server and content needs to be fetched from the origin servers.



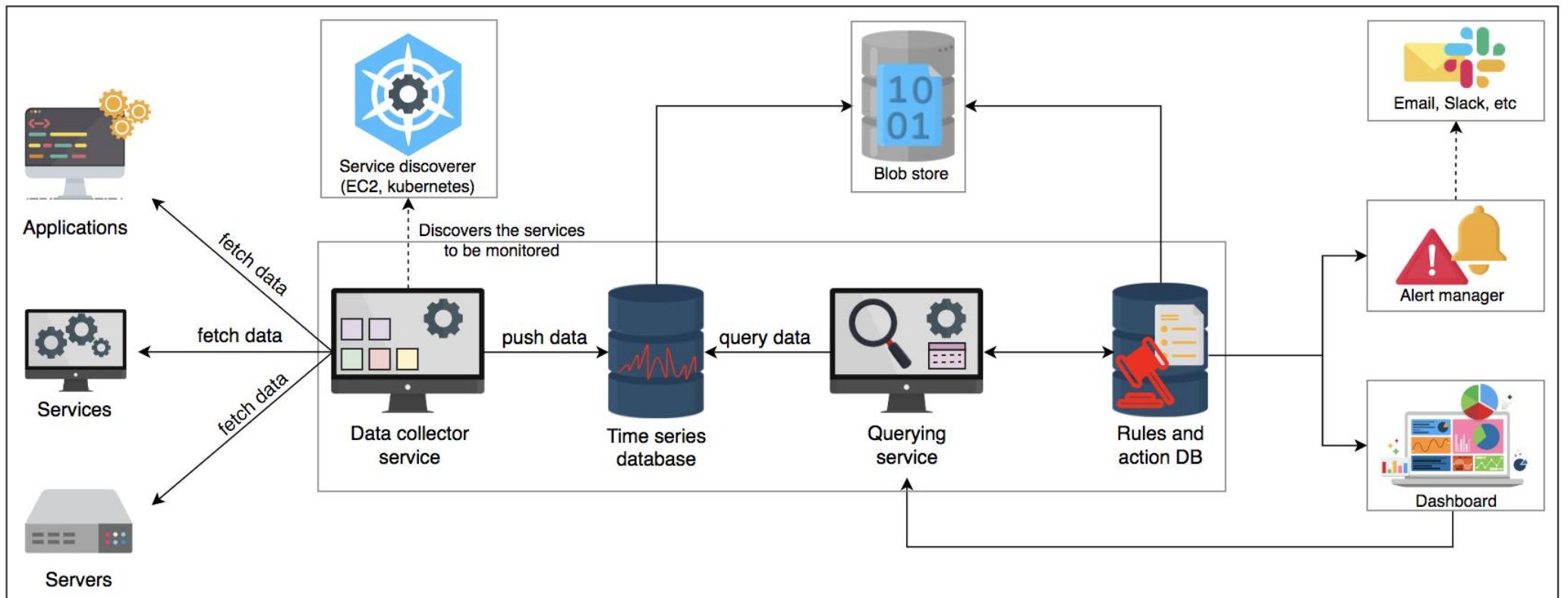


Detailed design of monitoring system

Requirements Fulfilled by Each Approach

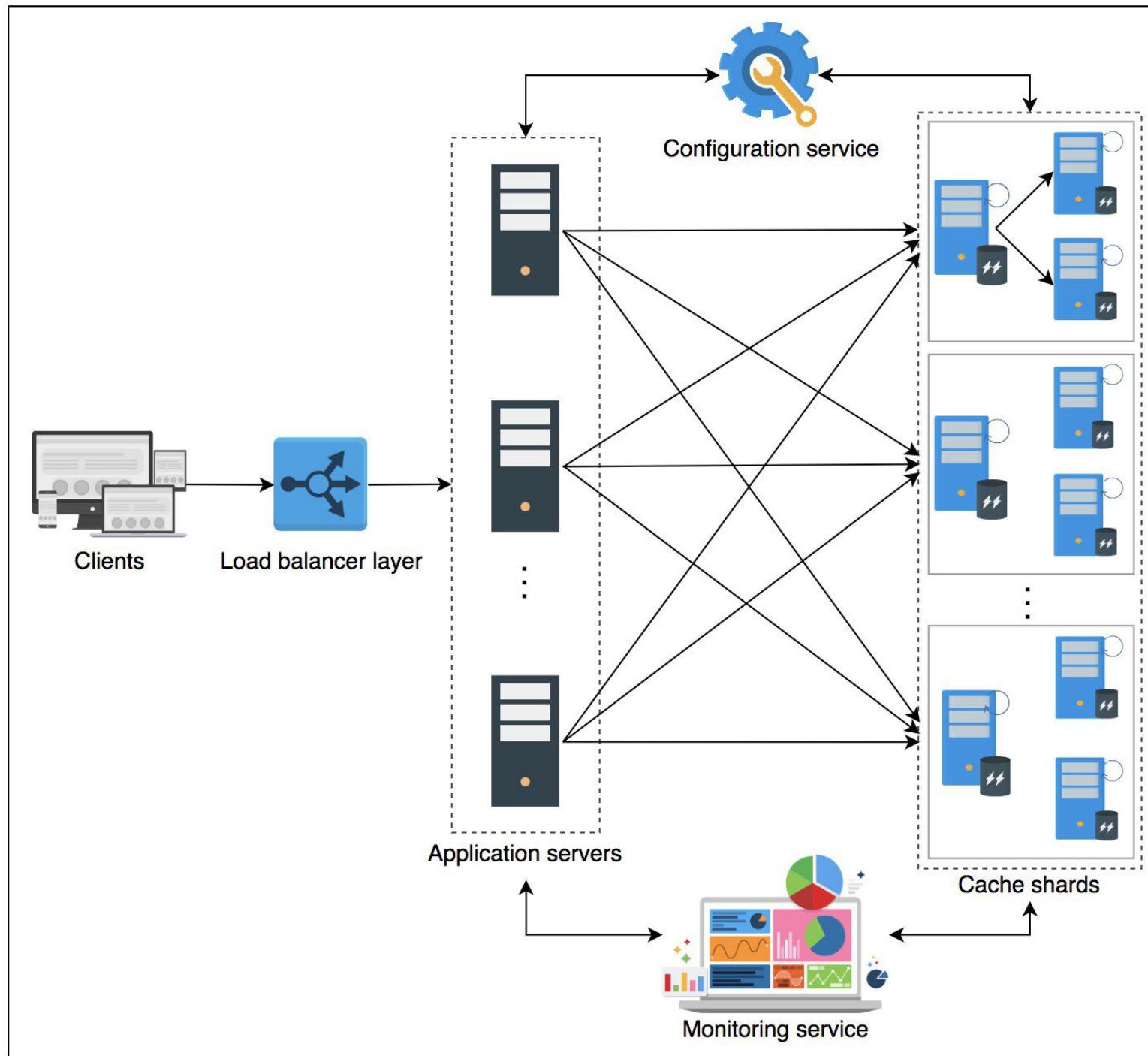
	Unique	Scalable	Available	64-bit numeric ID	Causality maintained
Using UUID	✗	✓	✓	✗	✗
Using a database	✗	✗	✓	✓	✗
Using a range handler	✓	✓	✓	✓	✗
Using UNIX time stamps	✗	weak	✓	✓	weak
Using Twitter Snowflake	✓	✓	✓	✓	weak
Using vector clocks	✓	weak	✓	can exceed	✓
Using TrueTime	✓	✓	✓	✓	✓



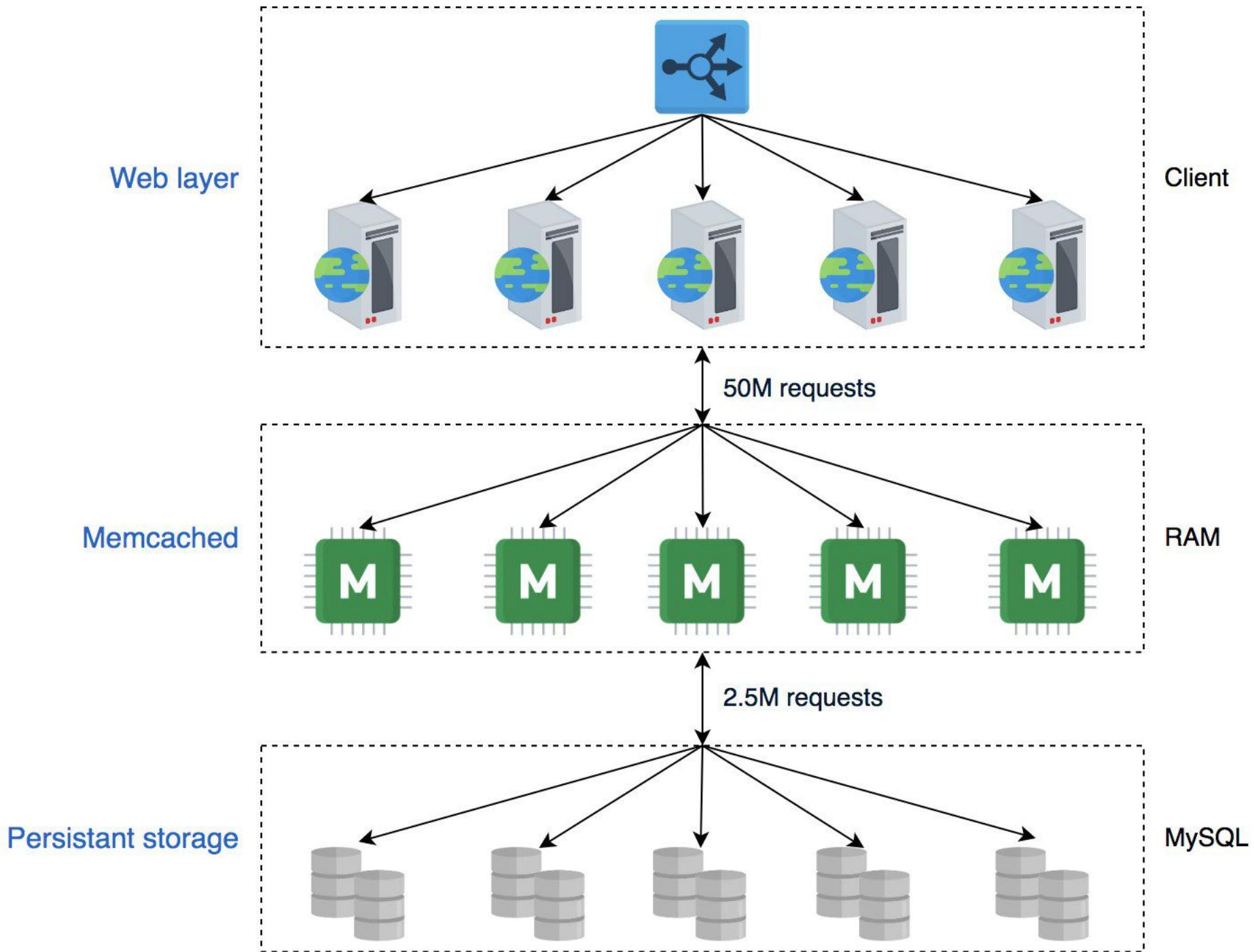


Detailed design of monitoring system



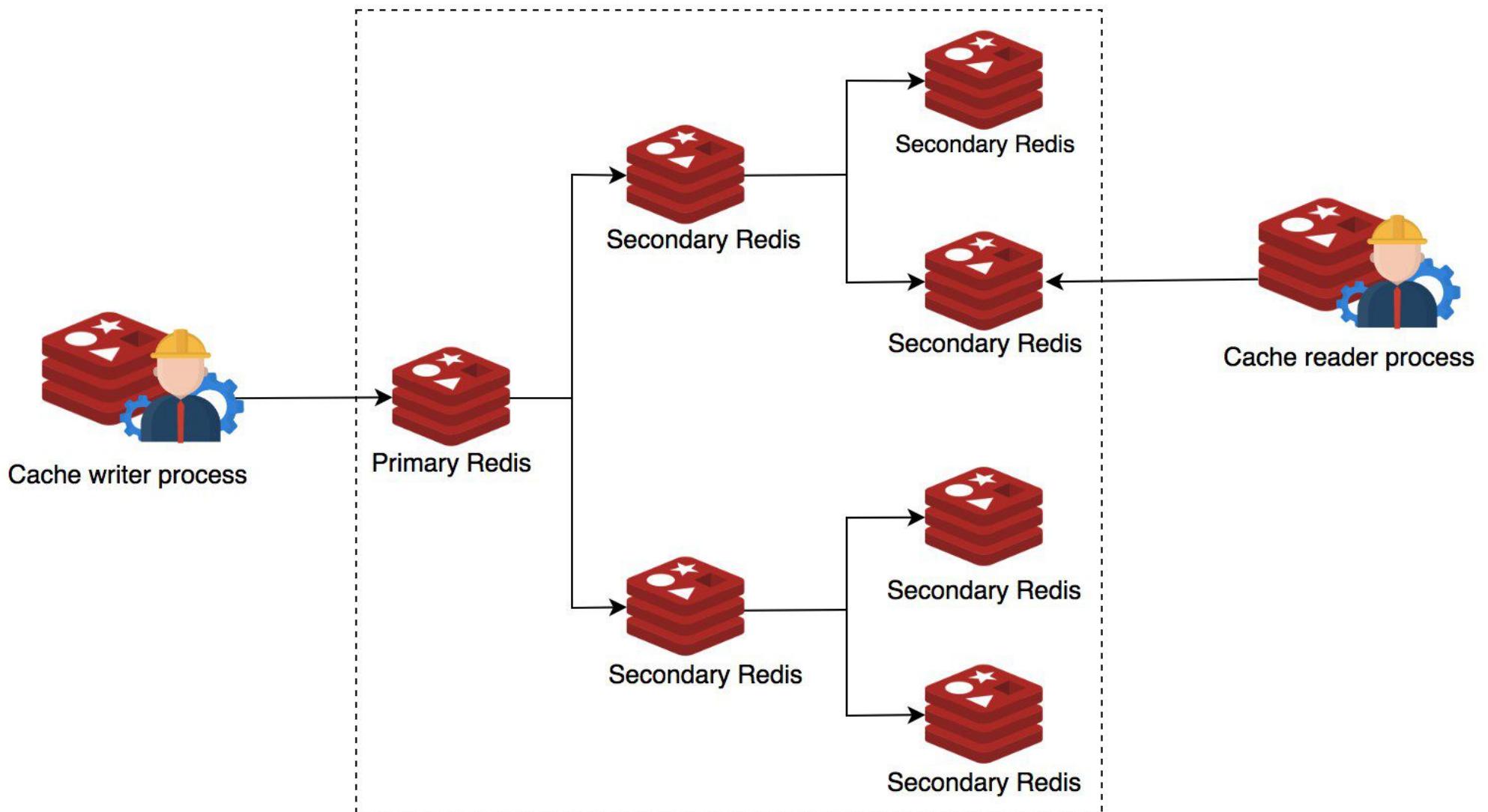


Detailed design of a distributed caching system



Redis cluster

- supports automatic failover, if primary fails
- supports tiered storage

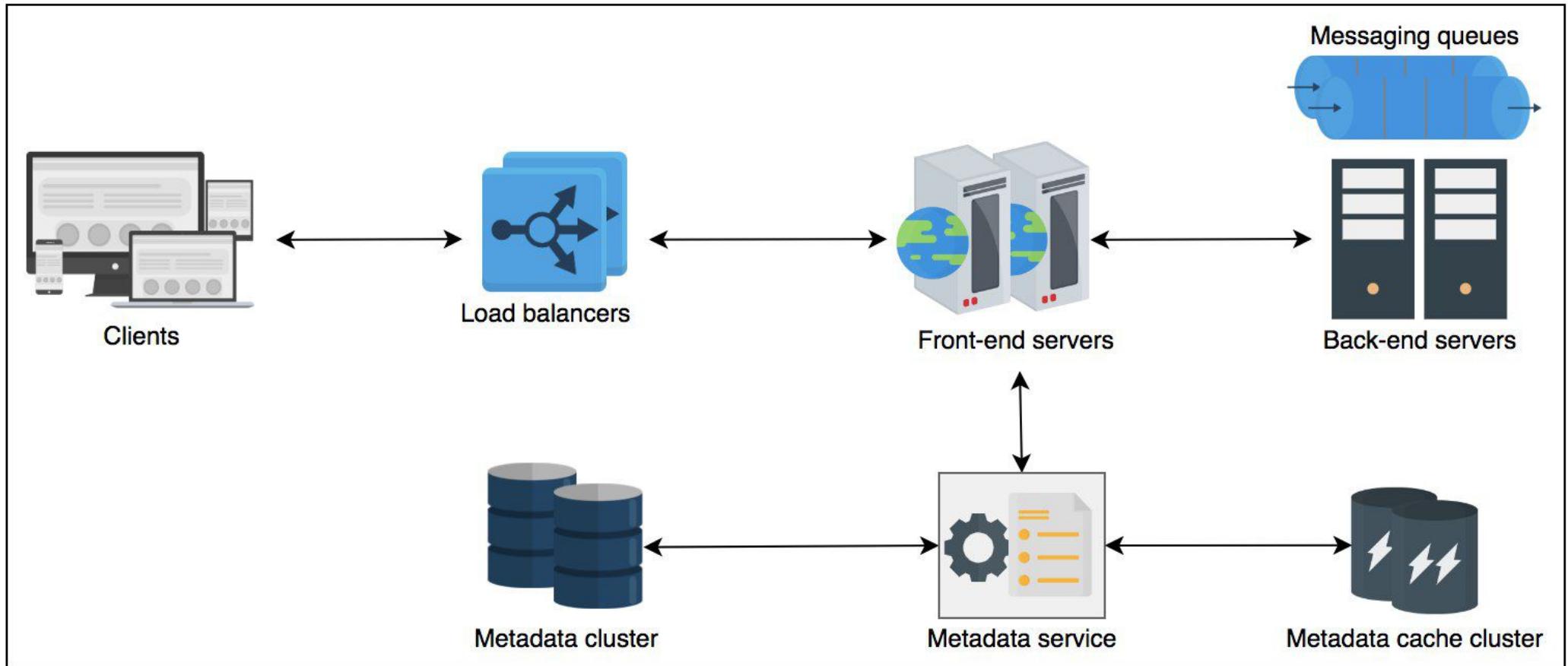


Redis structure supporting automatic failover using redundant secondary replicas

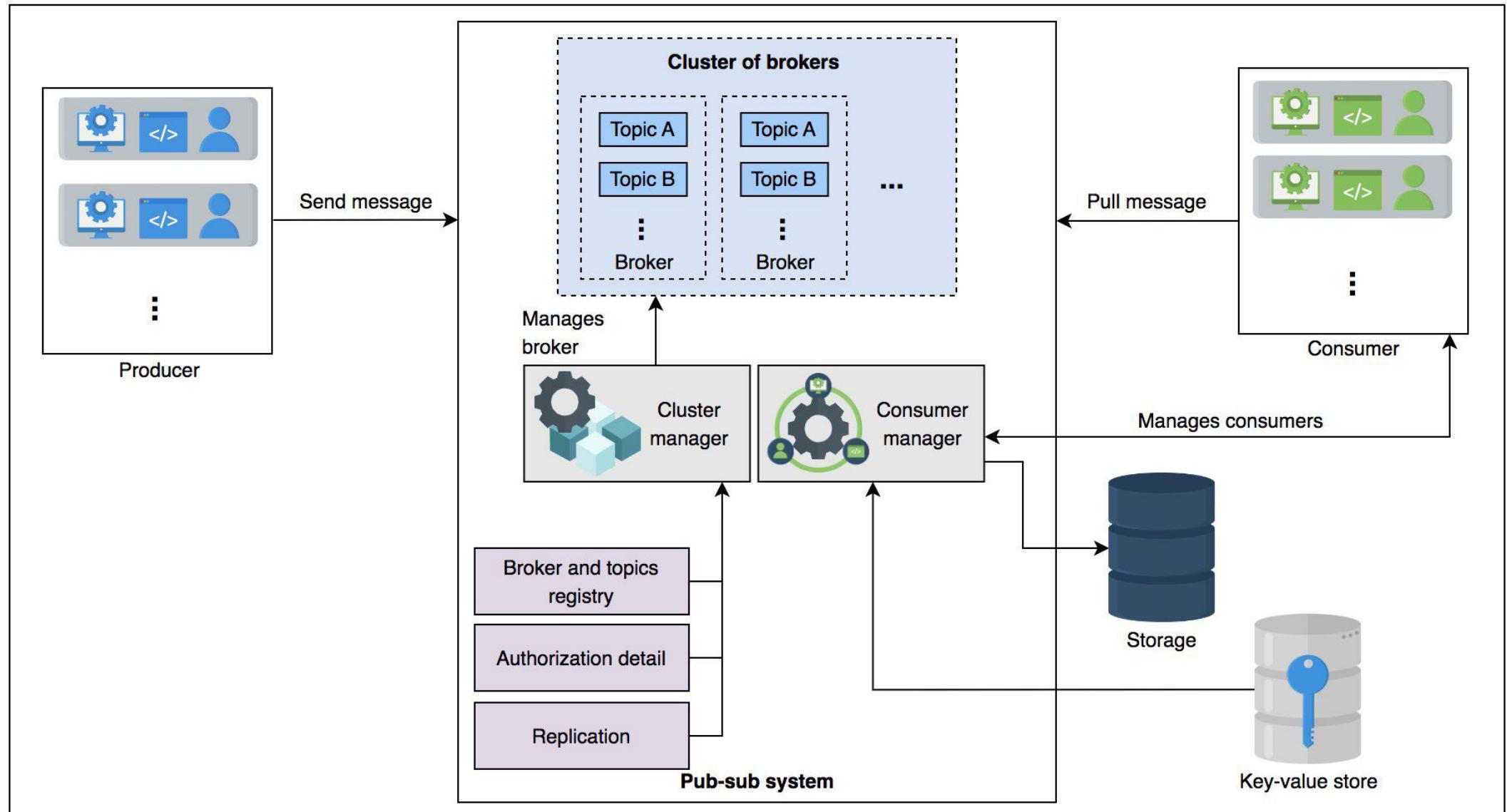
Features Offered by Memcached and Redis

Feature	Memcached	Redis
Low latency	Yes	Yes
Persistence	Possible via third-party tools	Multiple options
Multilanguage support	Yes	Yes
Data sharding	Possible via third-party tools	Built-in solution
Ease of use	Yes	Yes
Multithreading support	Yes	No
Support for data structure	Objects	Multiple data structures
Support for transaction	No	Yes
Eviction policy	LRU	Multiple algorithms
Lua scripting support	No	Yes
Geospatial support	No	Yes









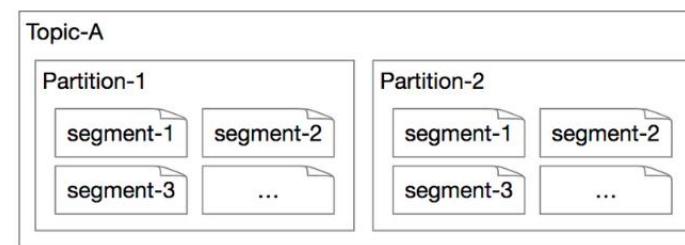
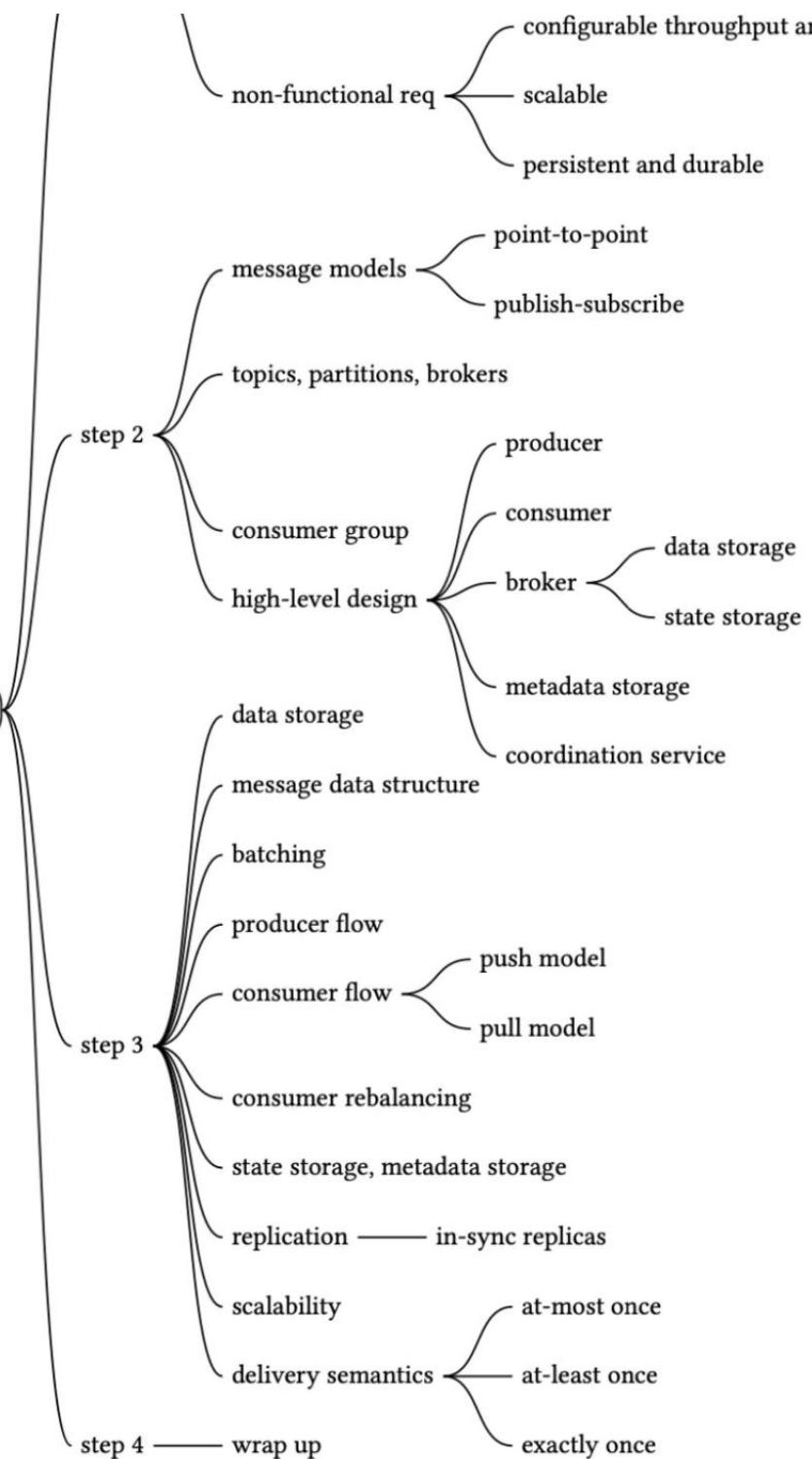


Figure 10 Data segment file distribution in topic partitions

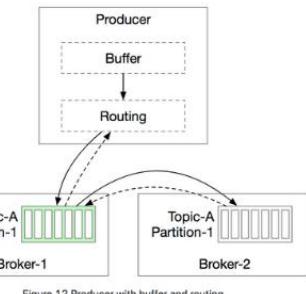


Figure 12 Producer with buffer and routing

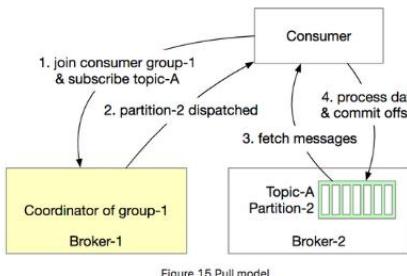


Figure 15 Pull model

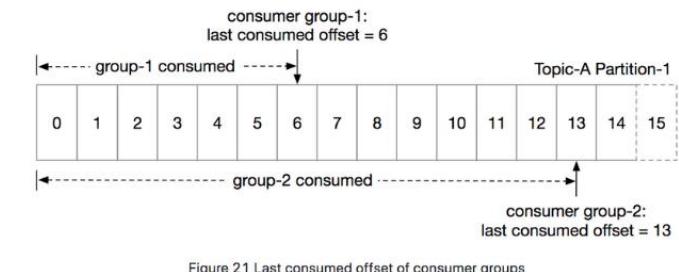


Figure 21 Last consumed offset of consumer groups

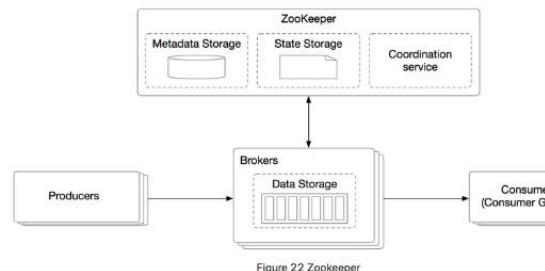


Figure 22 Zookeeper

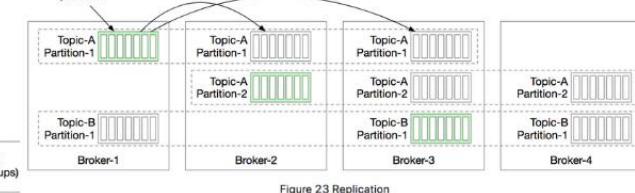


Figure 23 Replication

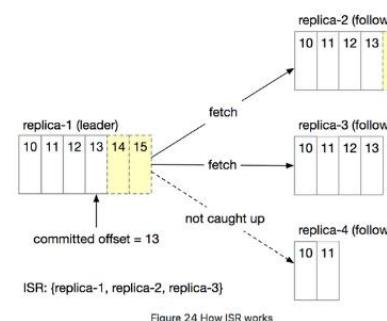


Figure 24 How ISR works

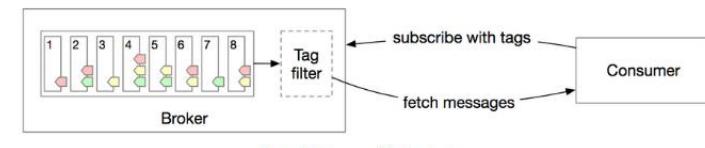


Figure 35 Message filtering by tags

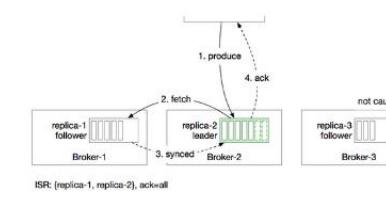


Figure 25 ack-all

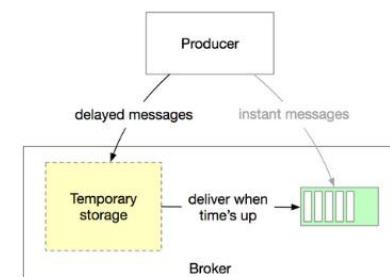
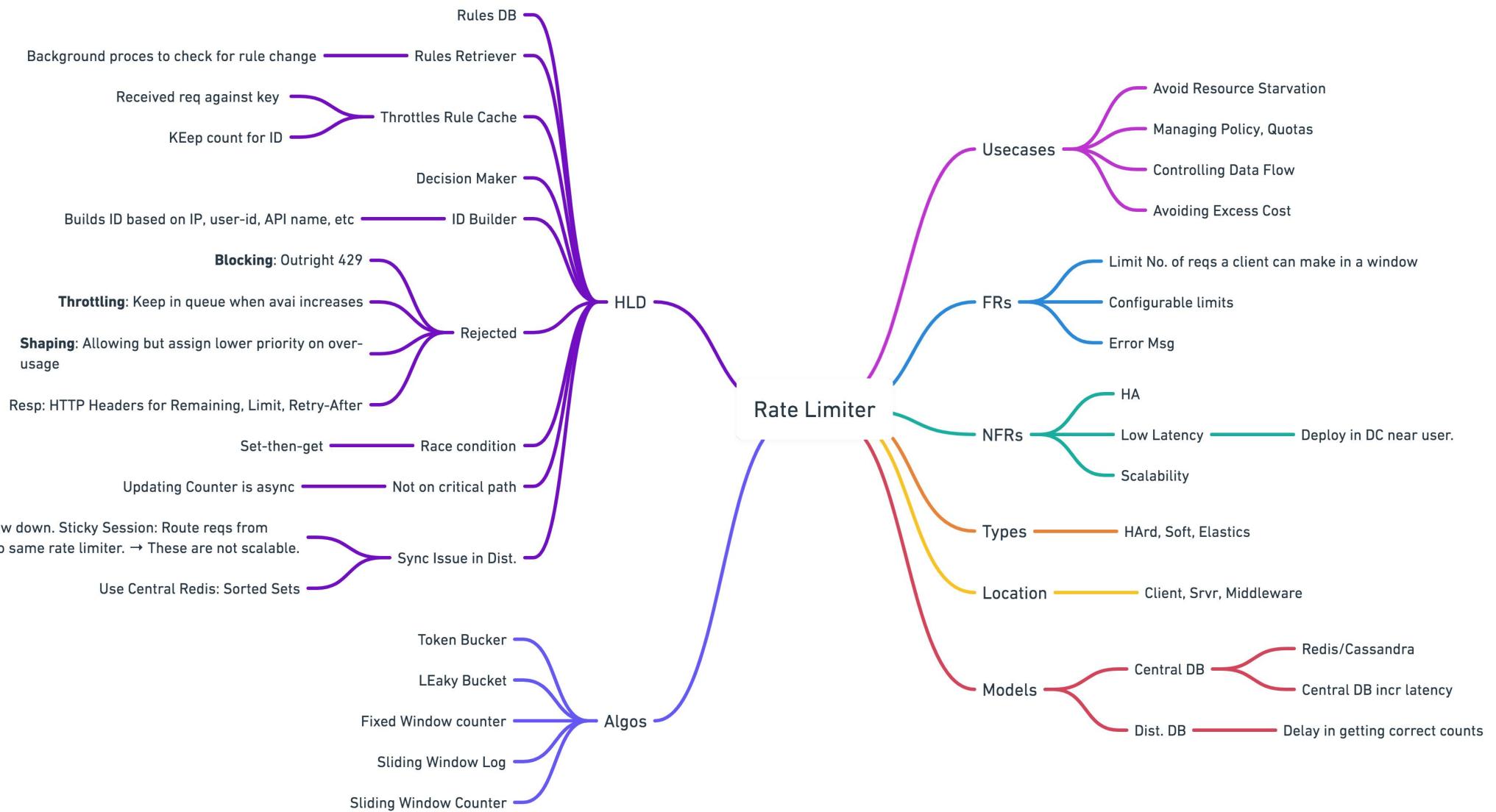
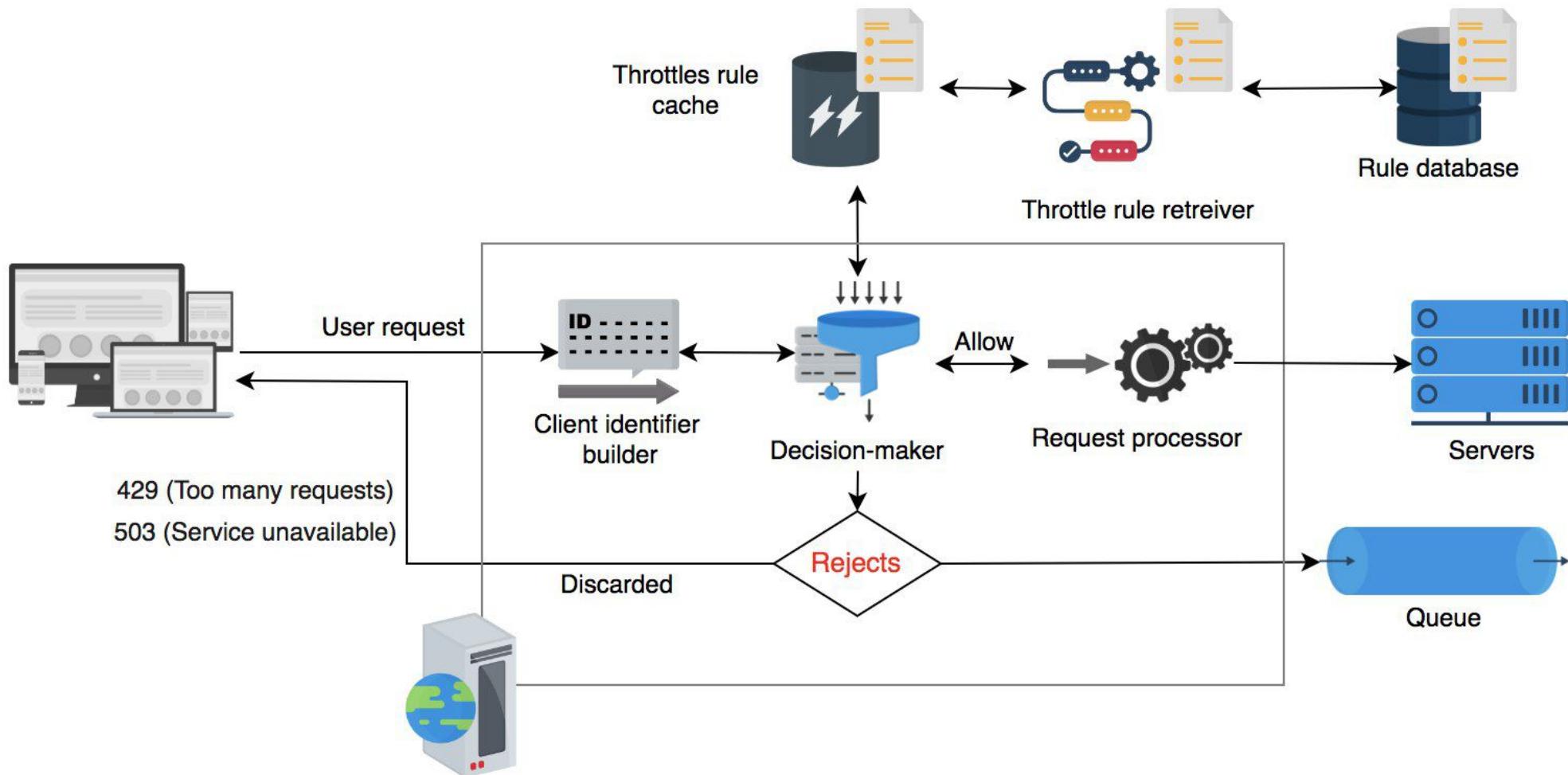


Figure 36 Delayed messages

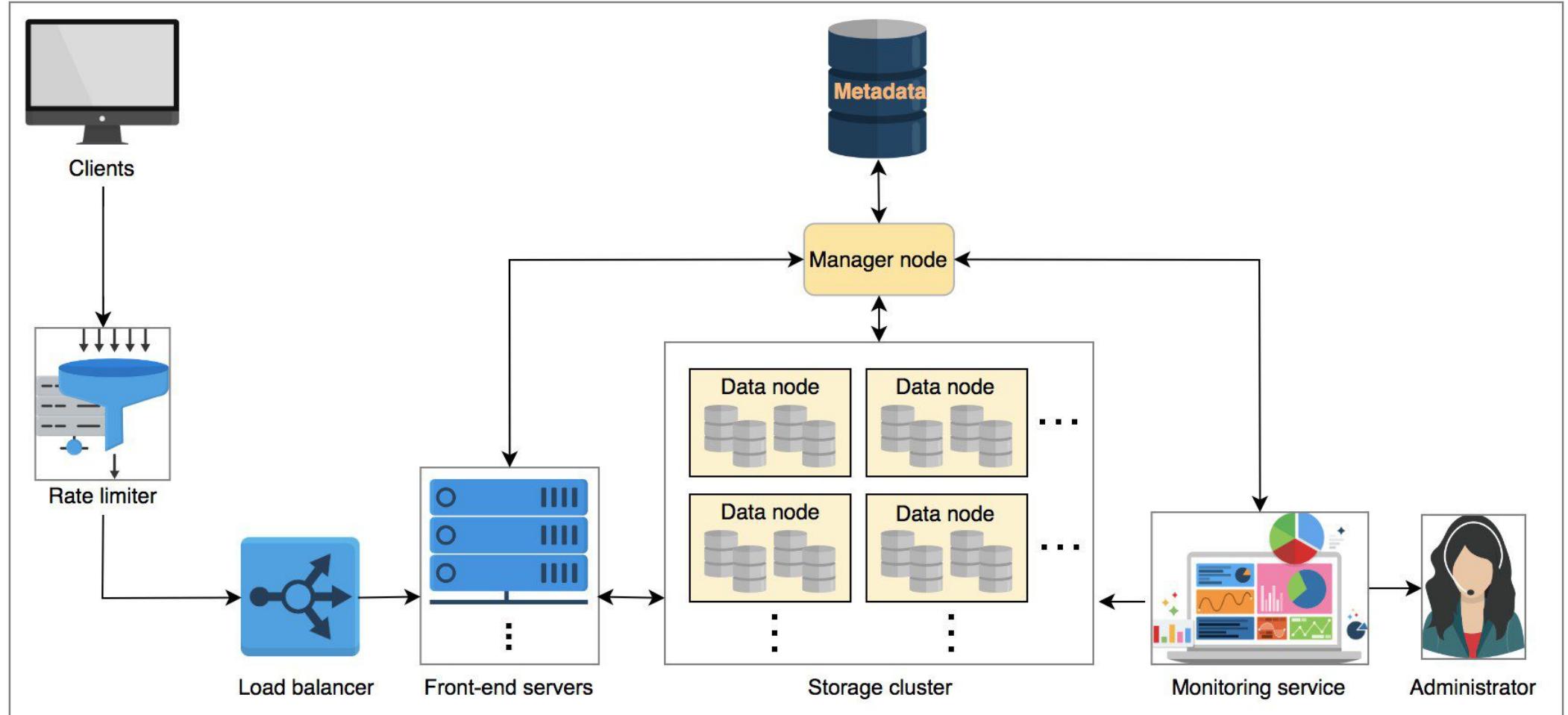




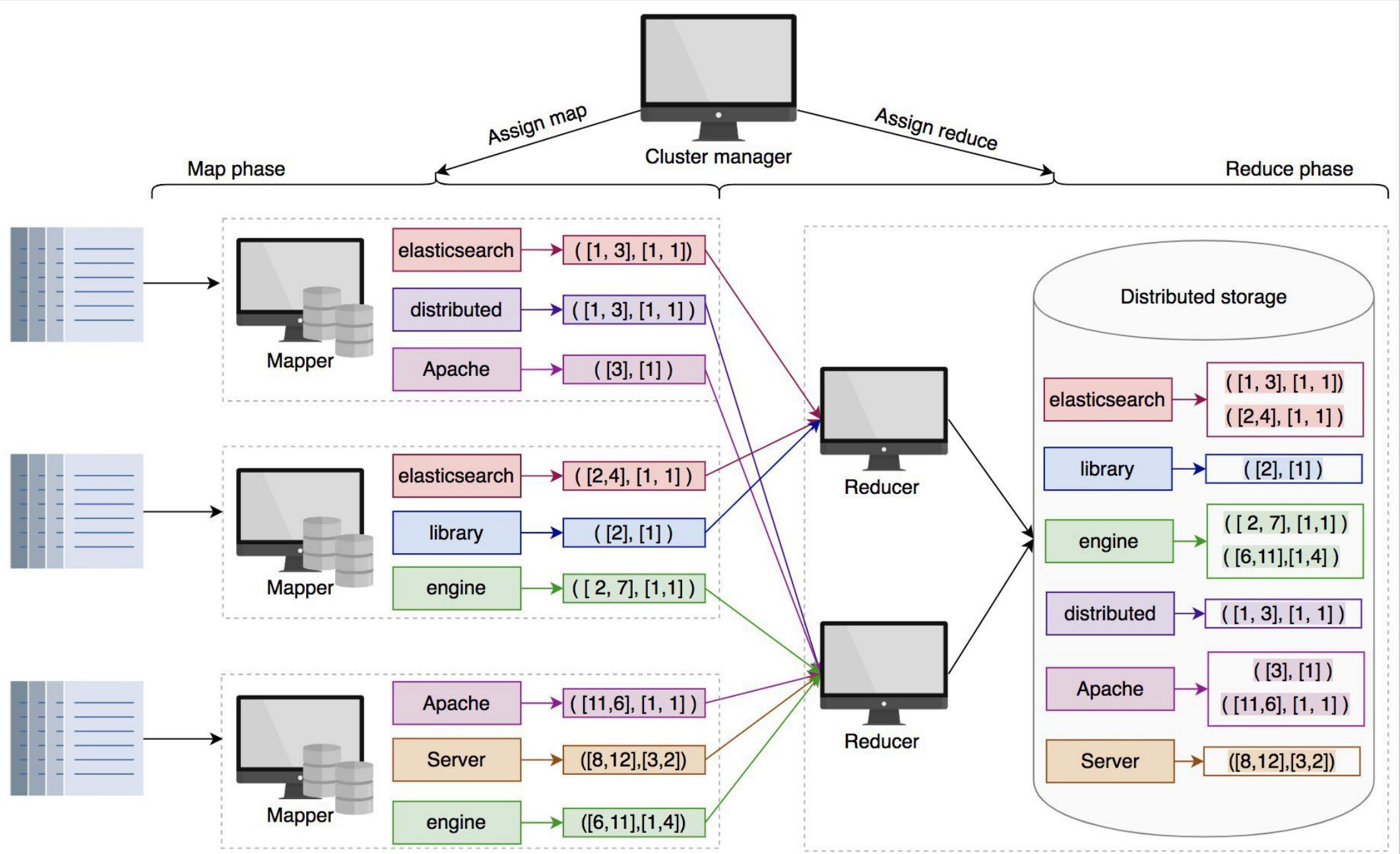
A Comparison of Rate-limiting Algorithms

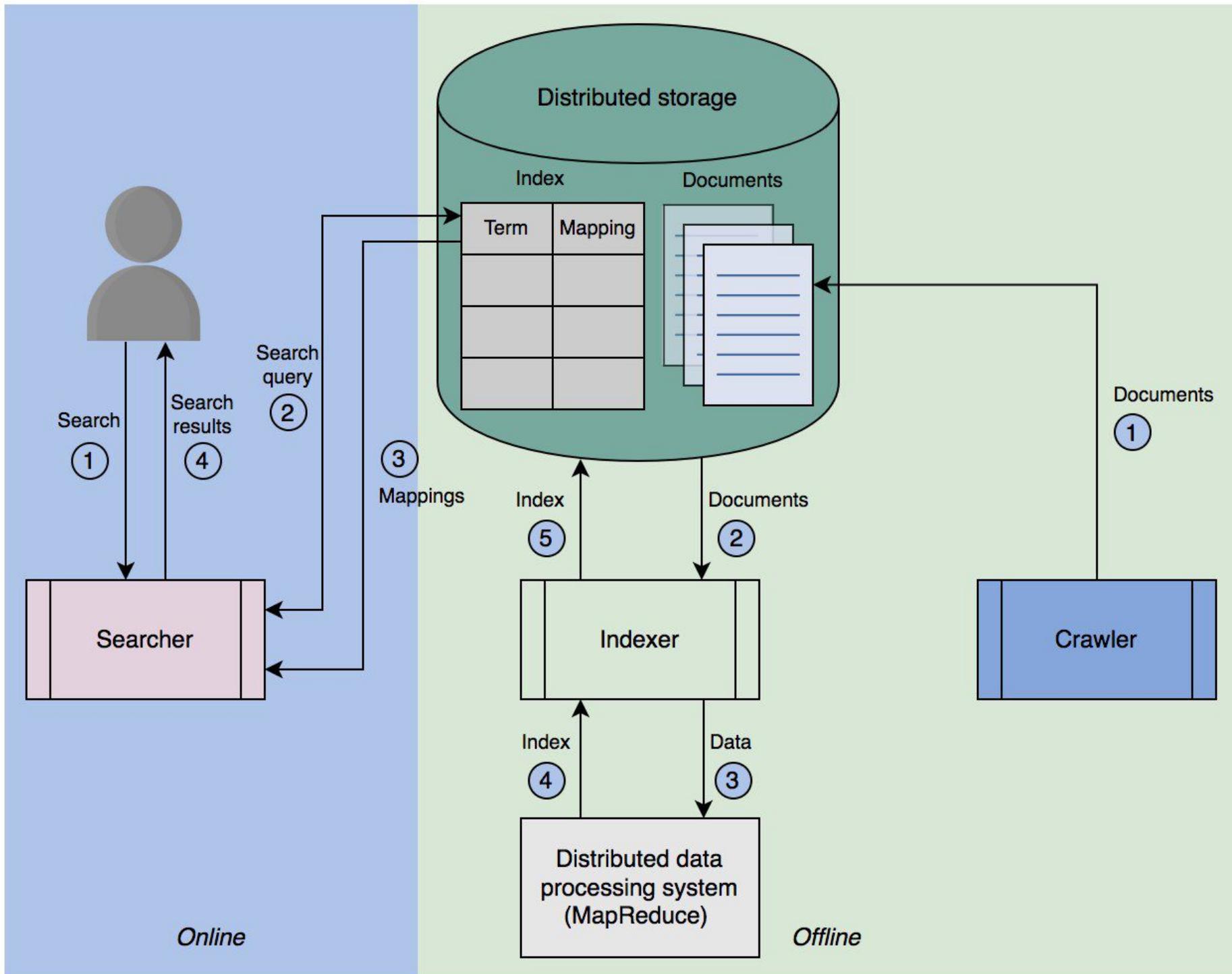
Algorithm	Space efficient	Allows burst?
Token bucket	Yes	Yes, it allows a burst of traffic within defined limit.
Leaking bucket	Yes	No
Fixed window counter	Yes	Yes, it allows bursts at the edge of the time window and can exceed the defined limit.
Sliding window log	No, maintaining the log requires extra storage.	No
Sliding window counter	Yes, but it requires relatively more space than other space efficient algorithms.	Smooths out the burst



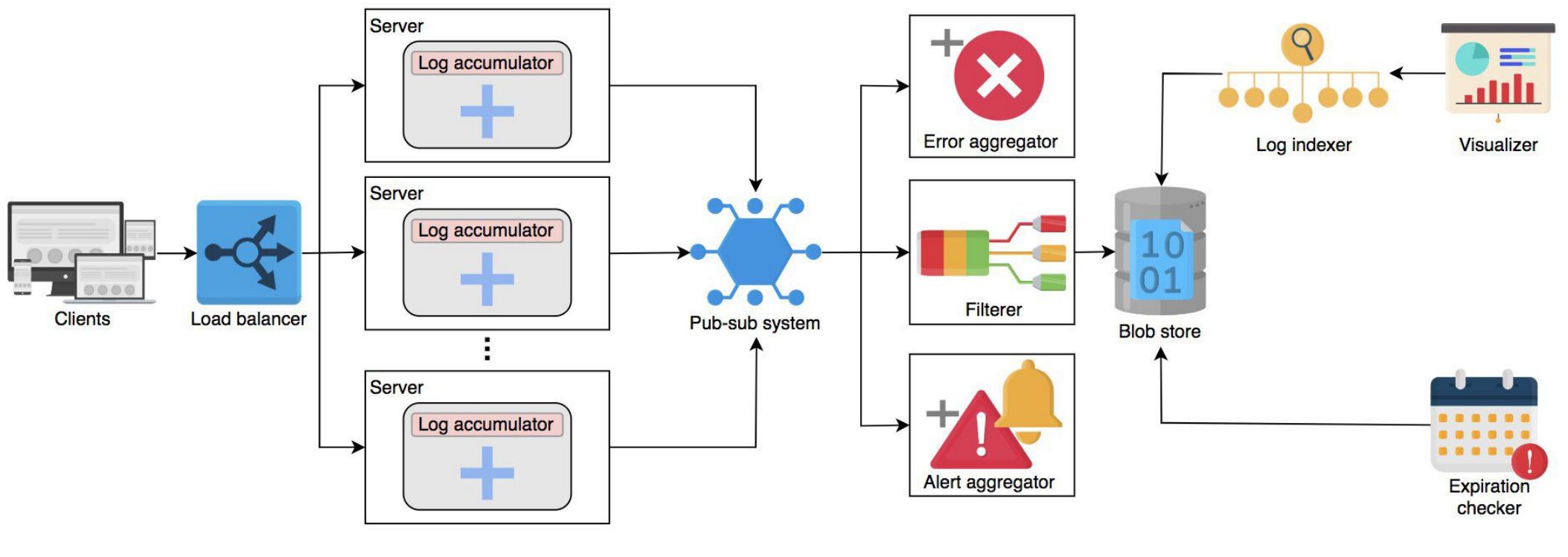


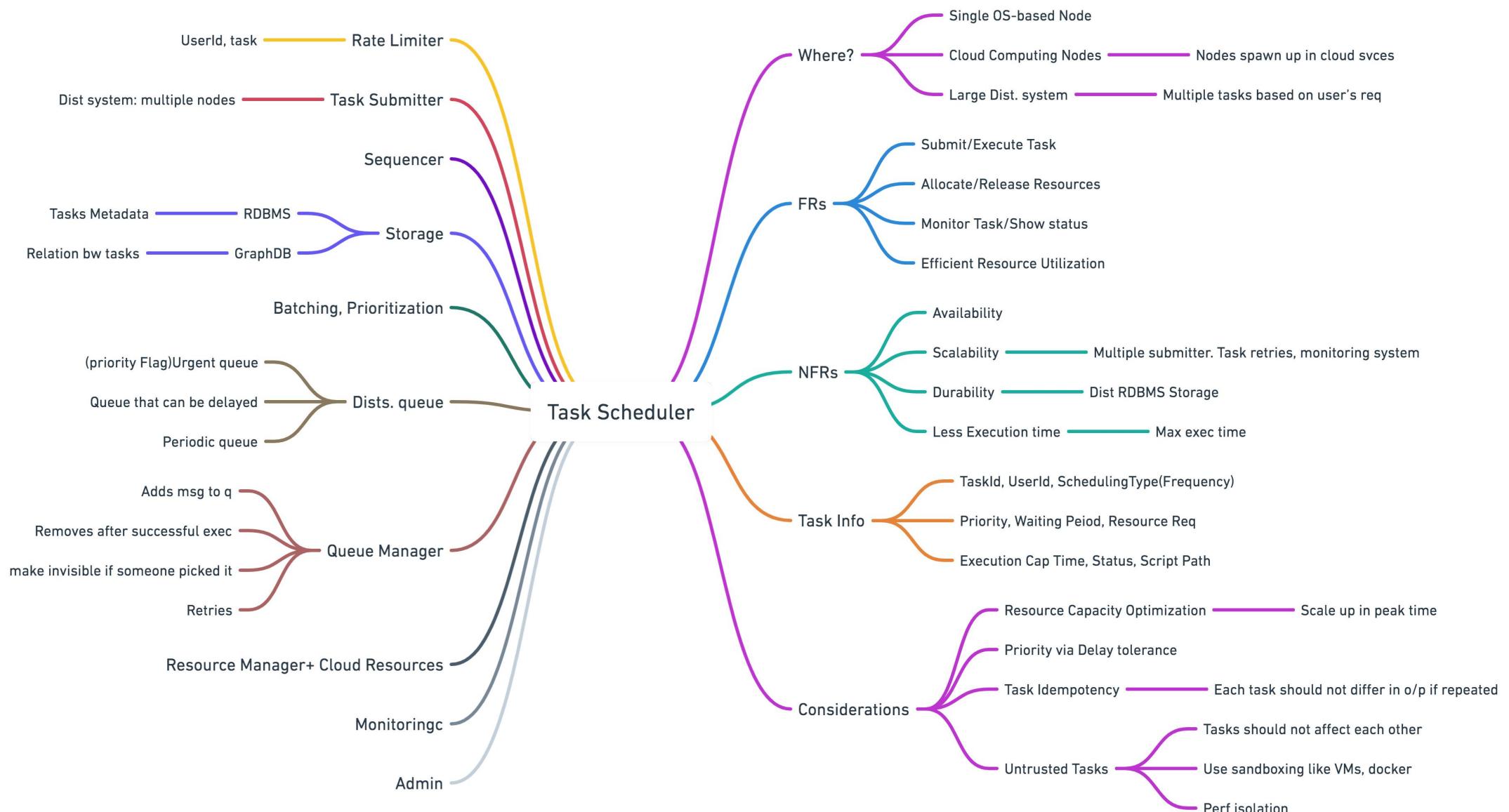


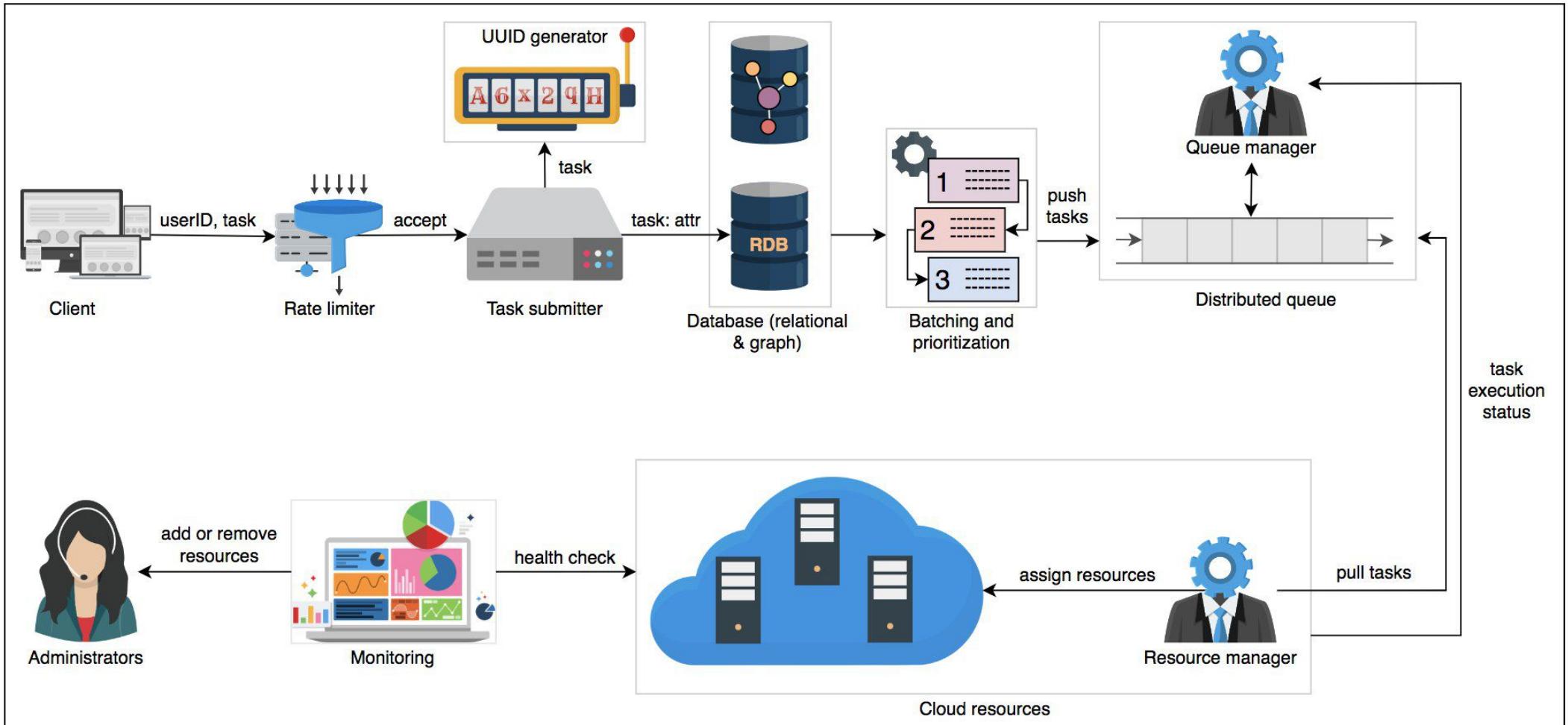




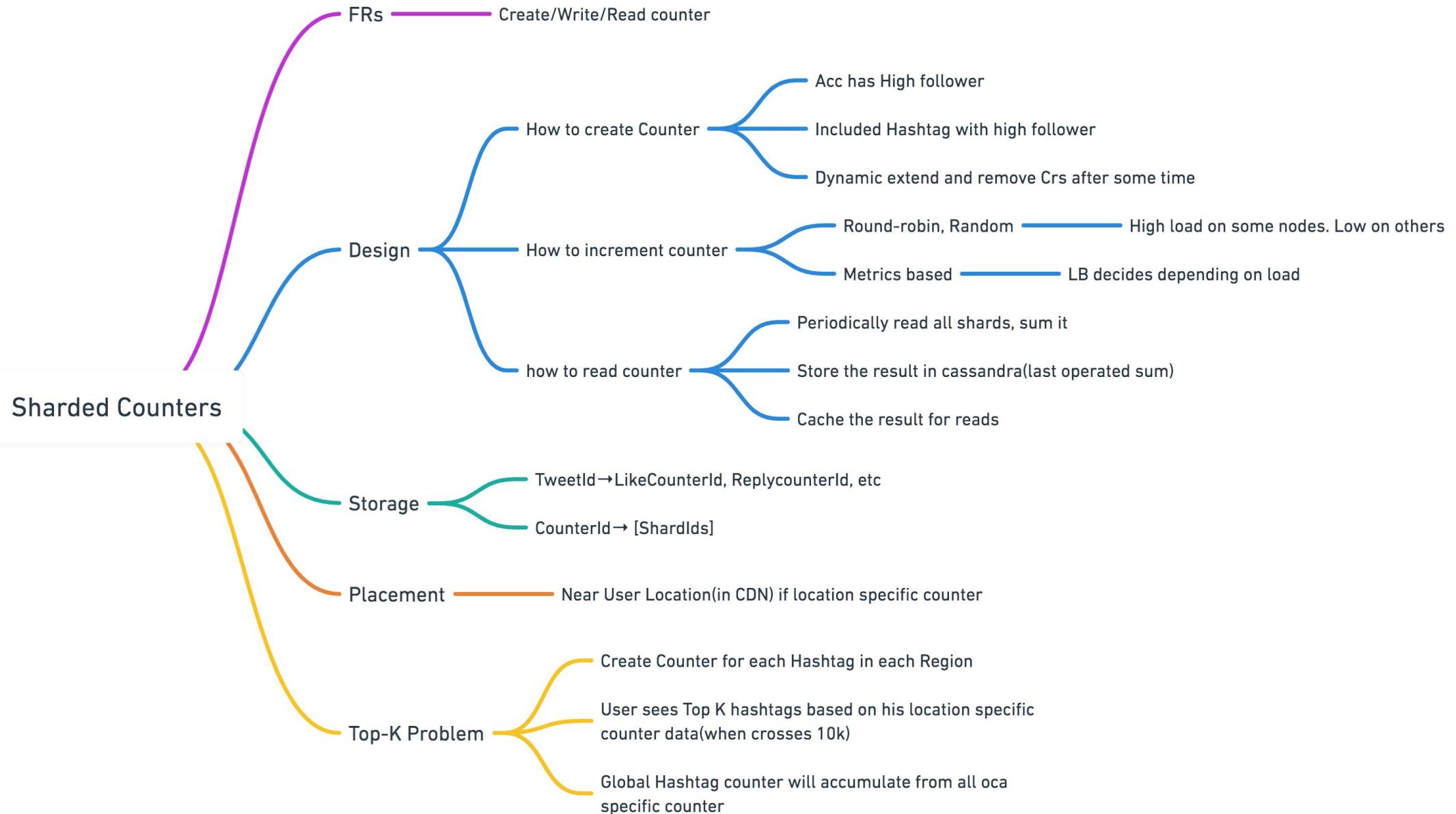


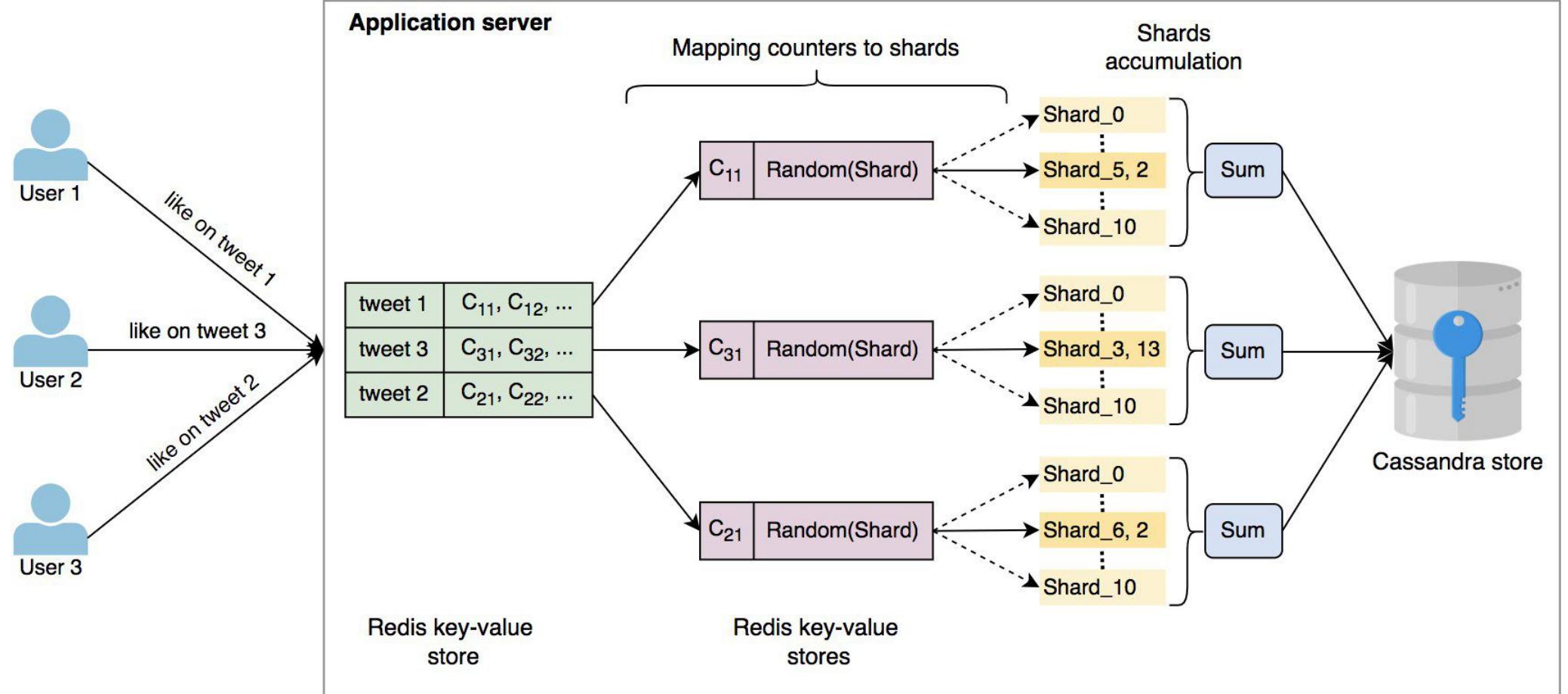


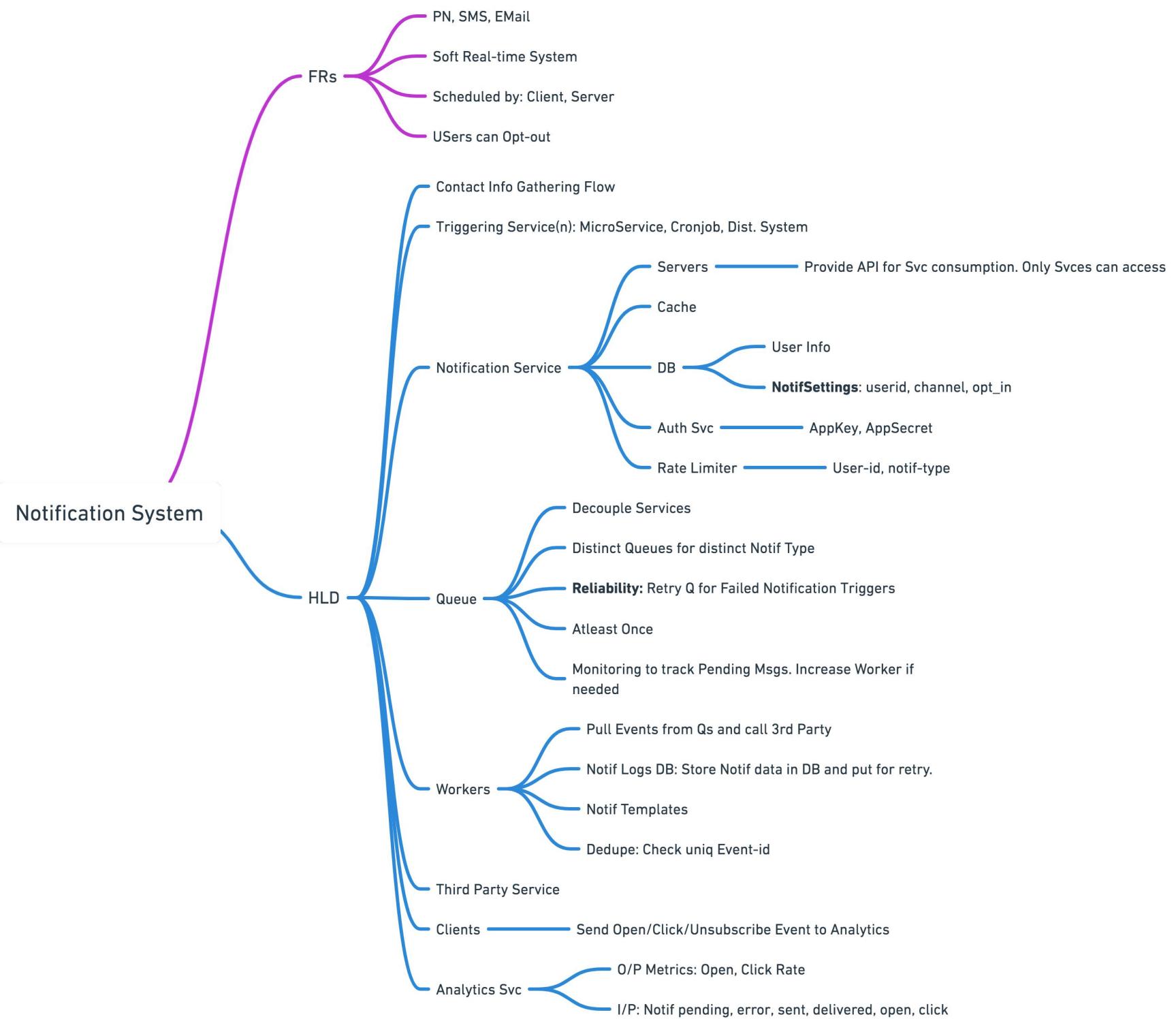




The design of task scheduler







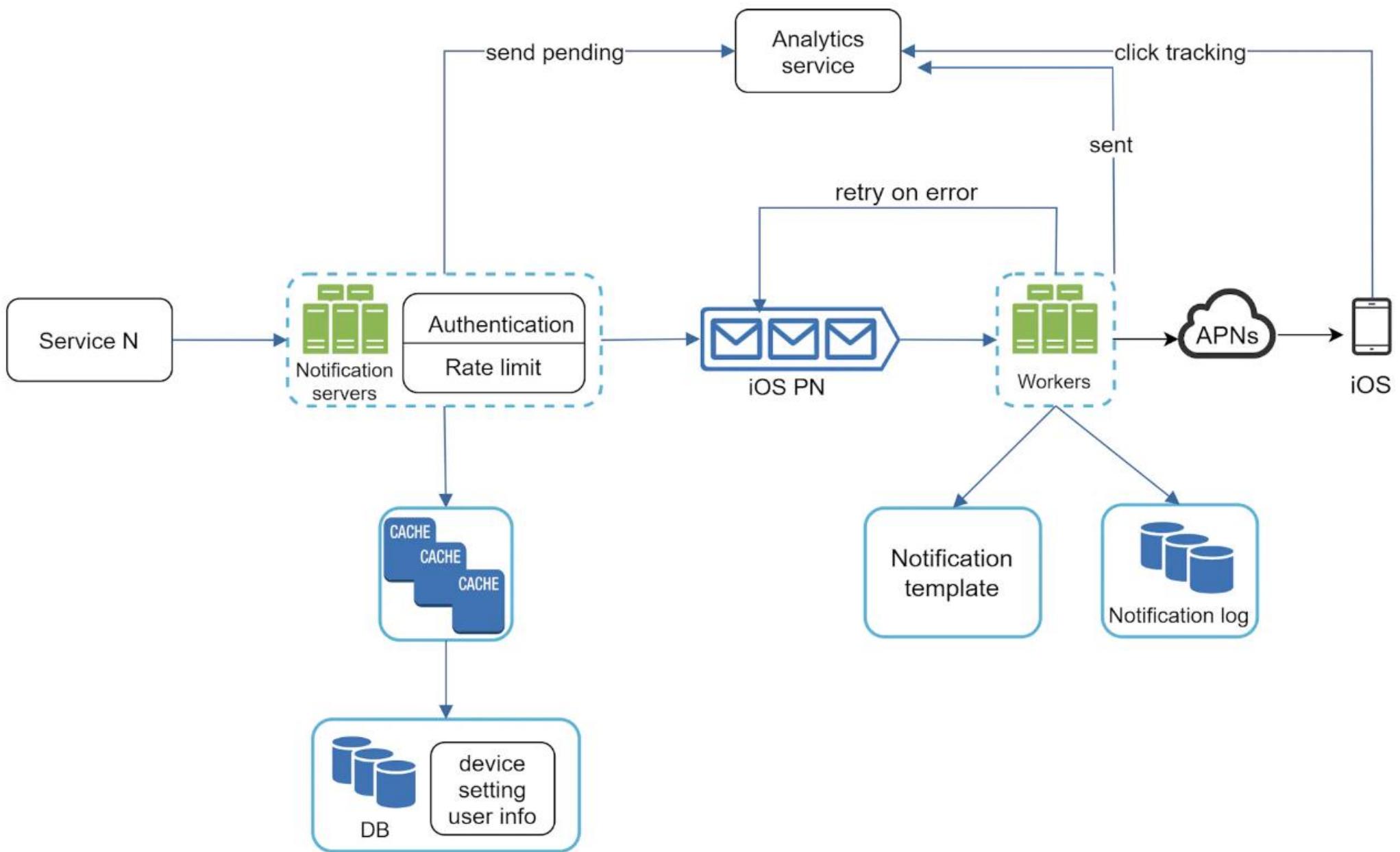


Figure 14

