zalando

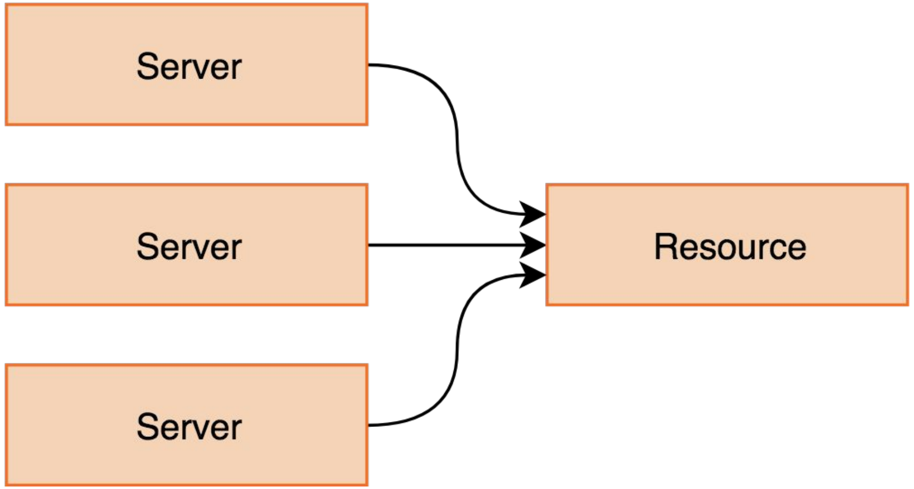
Raft Consensus Algorithm and Kubernetes

MAGNUS BOYE

27-02-2019

Distributed consensus

High availability



zalando

Raft in short

- Strong leader
- Strong consistency
- Designed for ease of understanding and implementation

Raft concepts

Leader election

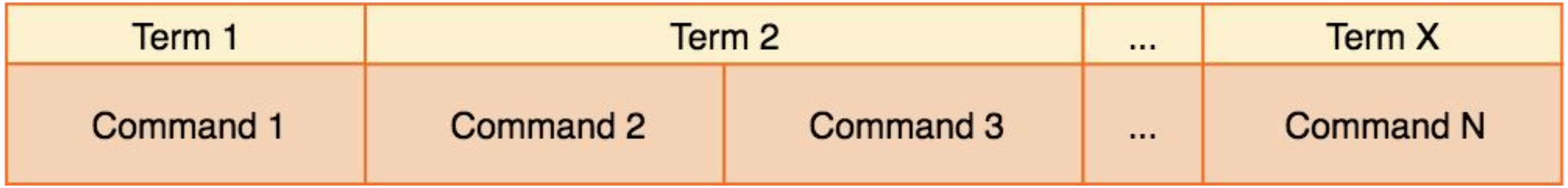
Log replication

Log compaction

Membership changes

Leader election

Raft log



Term 1	Term 2		...	Term X
Command 1	Command 2	Command 3	...	Command N

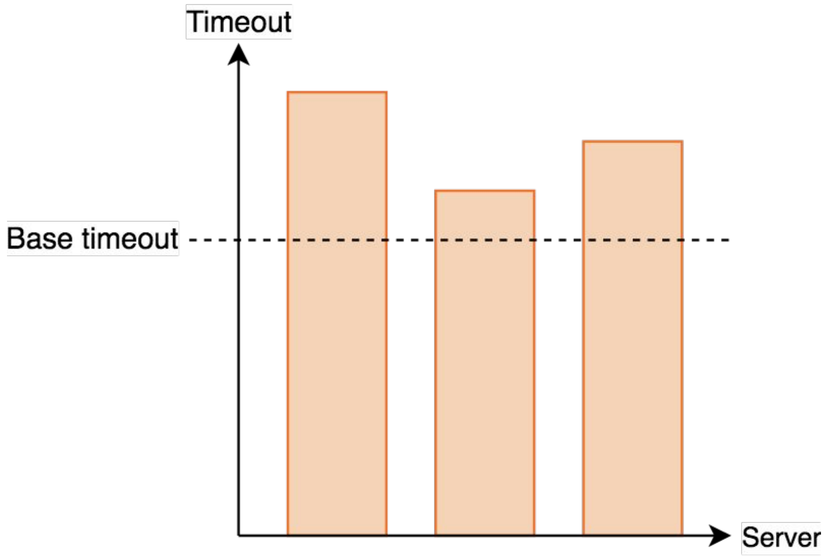
Server states

Leader

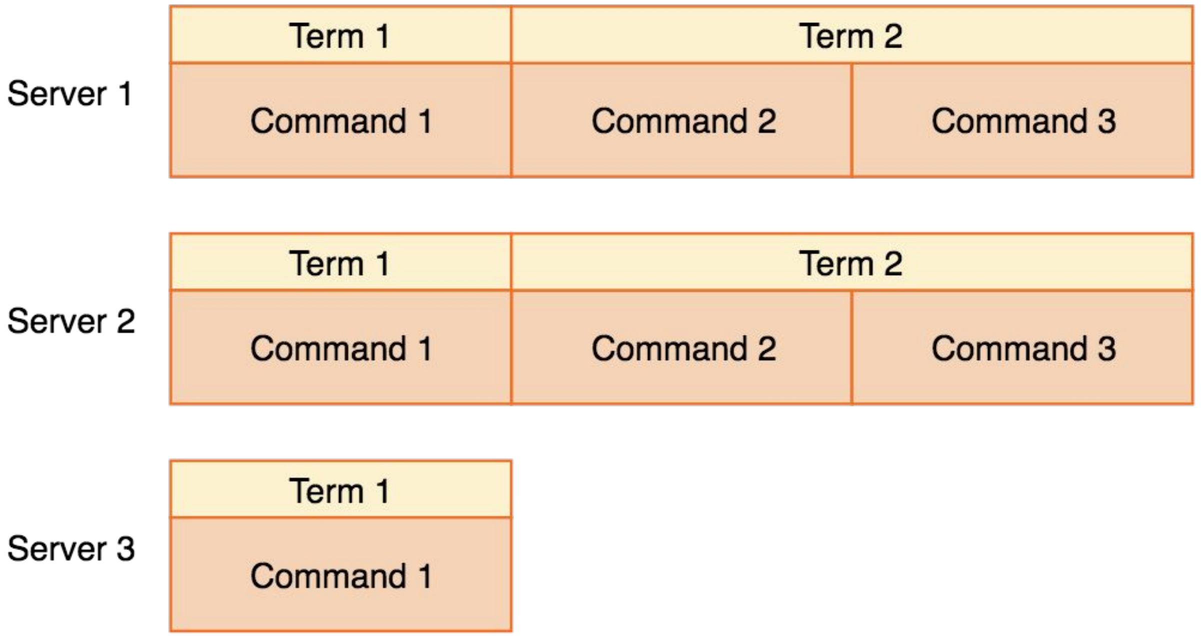
Follower

Candidate

Election timeout

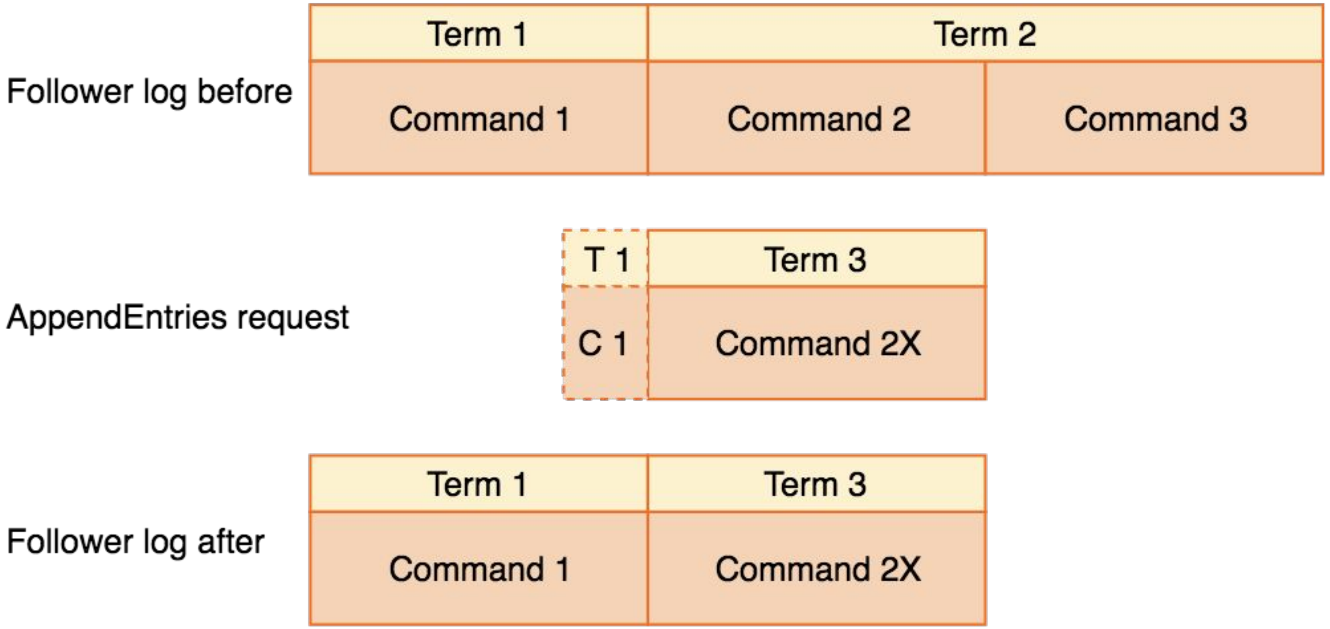


Voting criteria



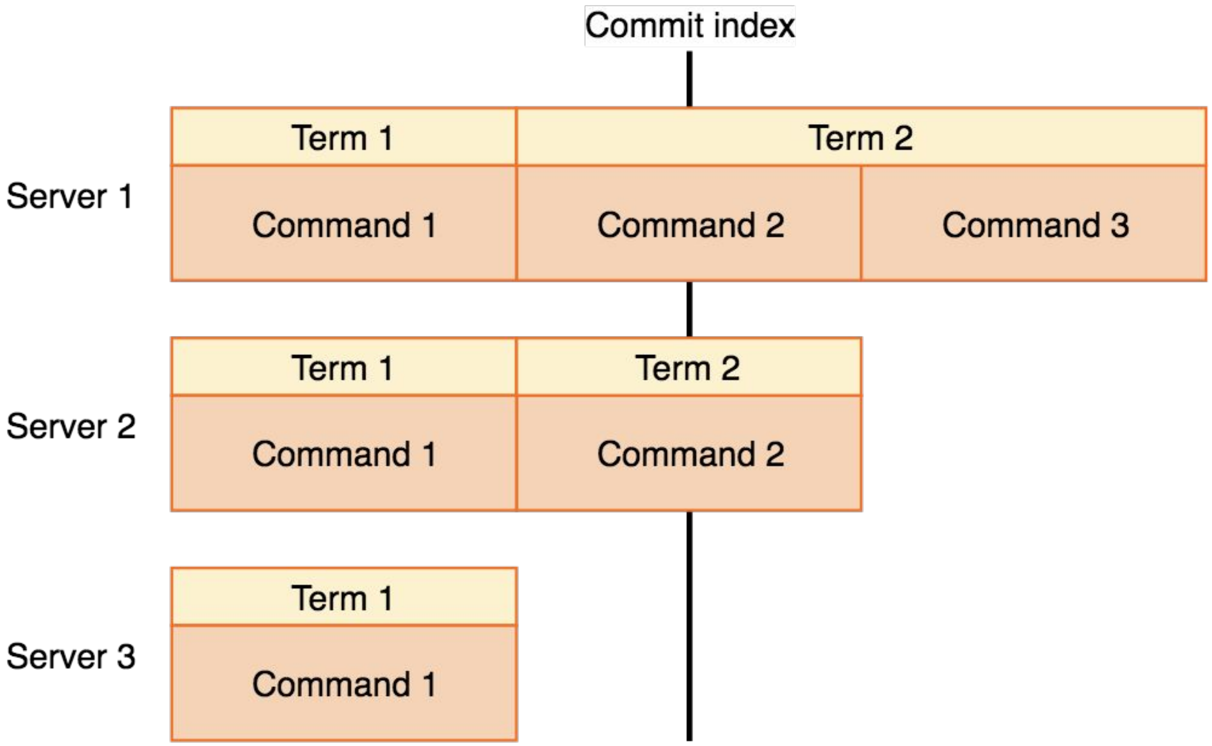
Log replication

Appending log entries



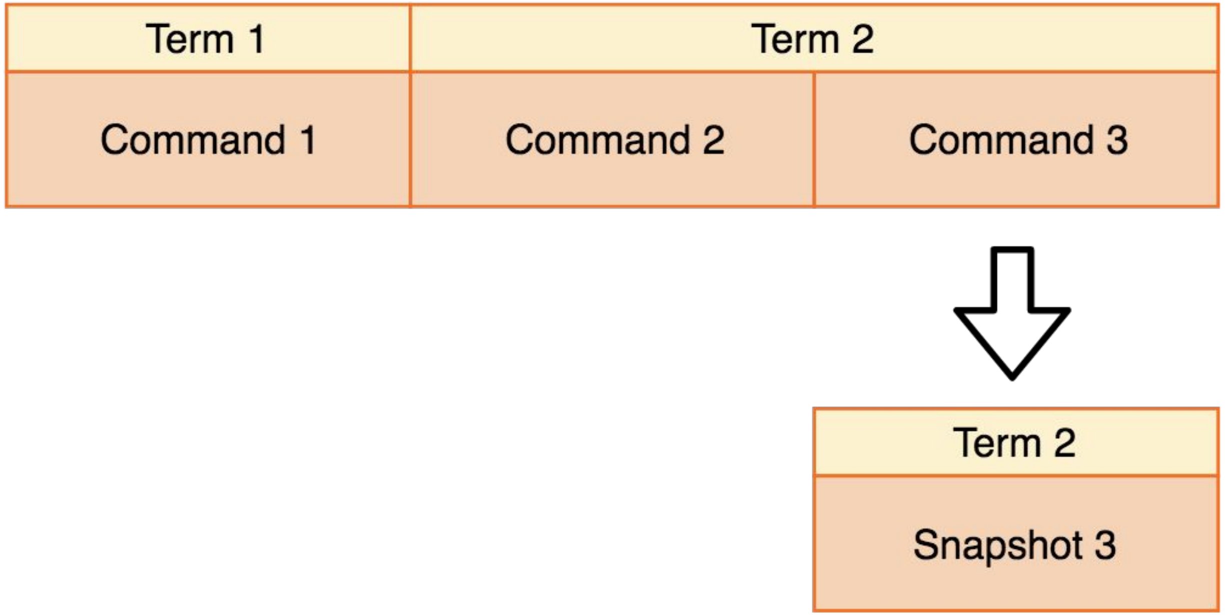
zalando

Committing log entries



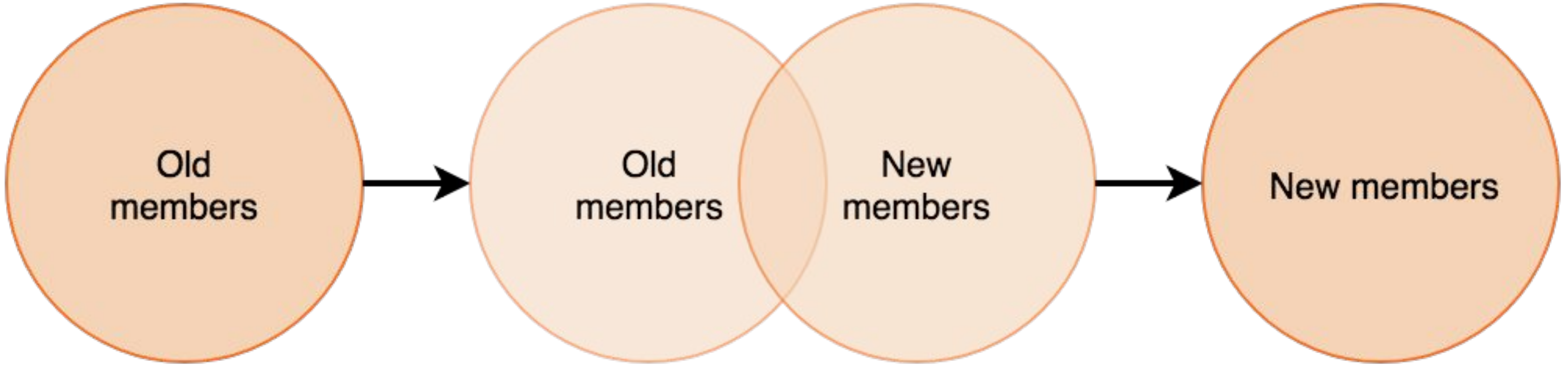
Log compaction

State machine snapshot



Membership changes

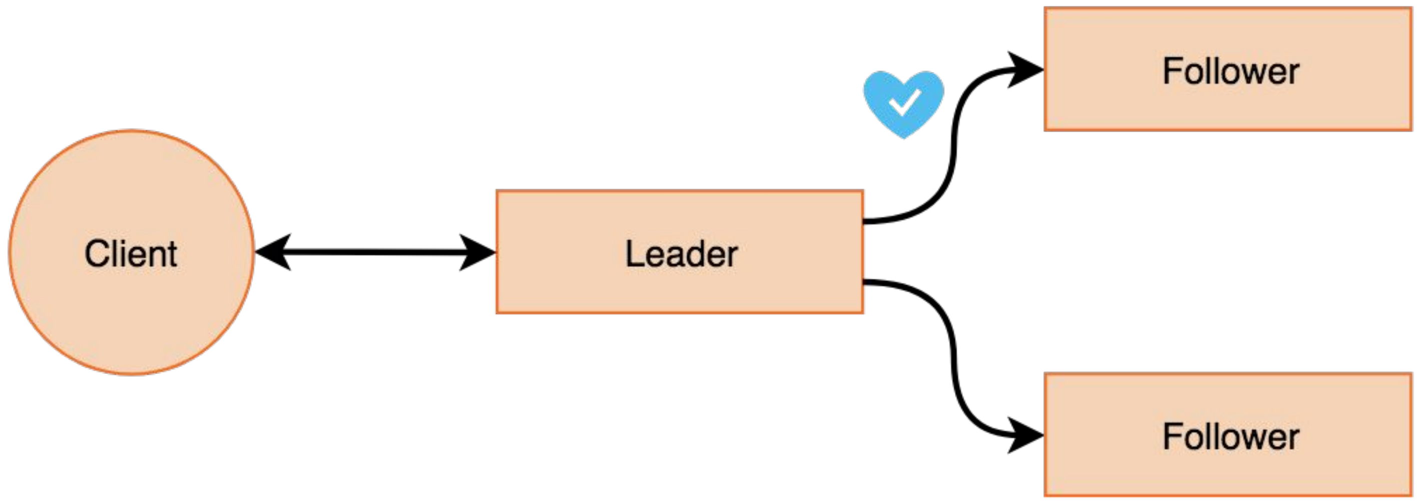
Joint consensus



Client access

zalando

Strong reads



Etcd

Features

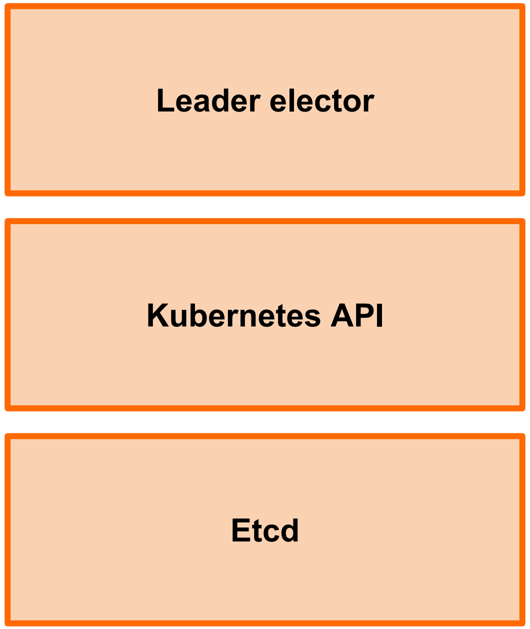
Key-value store

Transactions

Change notification

Leader Election for Applications in Kubernetes

Building on strong foundation



zalando

Endpoint annotations

```
control-plane.alpha.kubernetes.io/leader: '{
  "holderIdentity": "mboye-leader-election-demo-54df6685cf-n5bm4",
  "leaseDurationSeconds": 10,
  "acquireTime": "2019-02-26T15:43:53Z",
  "renewTime": "2019-02-26T15:57:11Z",
  "leaderTransitions": 0
}'
```

Demo

Questions

Thanks for listening

- Raft: https://raft.github.io
- Etcd: https://github.com/etcd-io/etcd
- Demo application: https://github.com/mboye/raft-talk
- Leader election for applications in Kubernetes