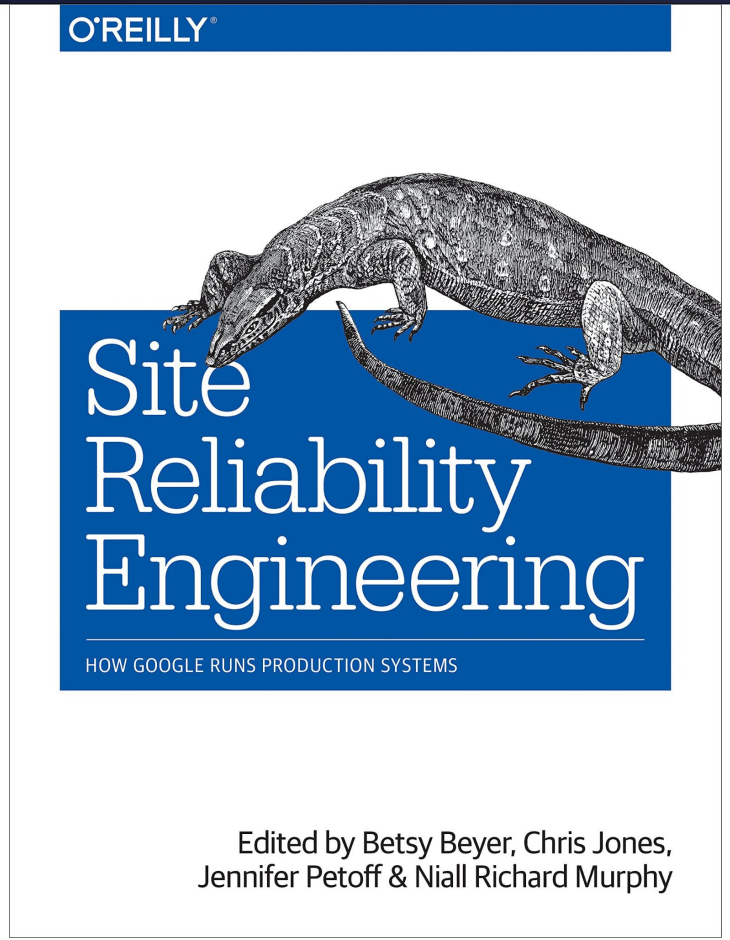

KubeCon
CloudNativeCon
North America 2022
BUILDING FOR THE ROAD AHEAD
DETROIT 2022
TiKV
TiKV
HELM

KubeCon CloudNativeCon North America 2022
BUILDING FOR THE ROAD AHEAD
DETROIT 2022

Lessons Learned From the etcd Data Inconsistency Issues

Marek Siarkowicz, Google
Benjamin Wang, VMware

Postmortem

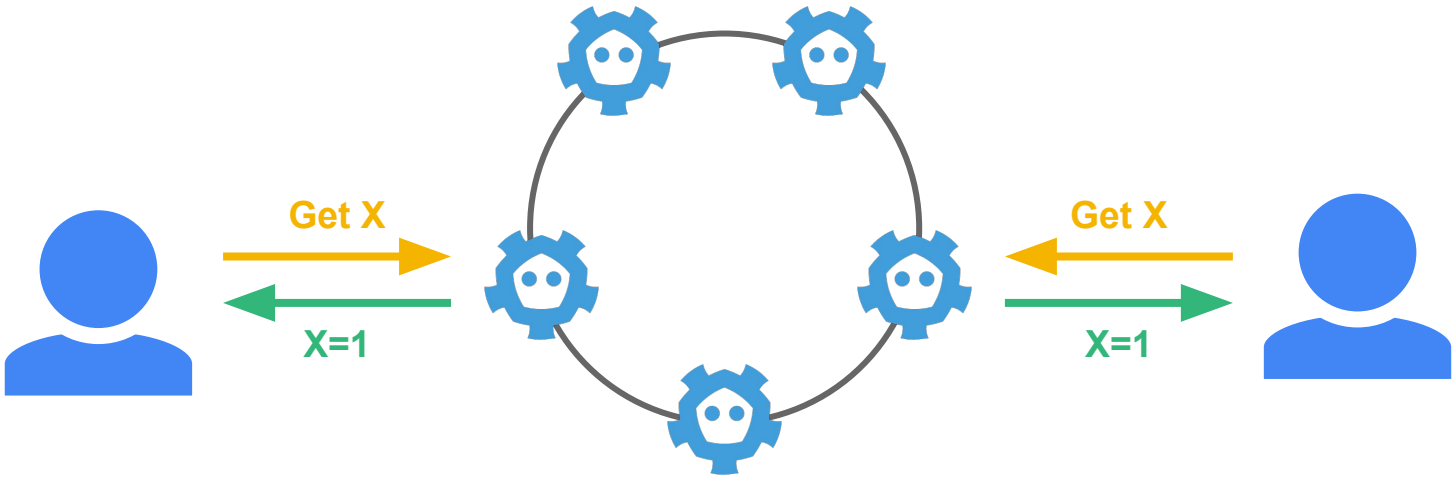

O'REILLY
Site Reliability Engineering
HOW GOOGLE RUNS PRODUCTION SYSTEMS
Edited by Betsy Beyer, Chris Jones, Jennifer Petoff & Niall Richard Murphy

Agenda

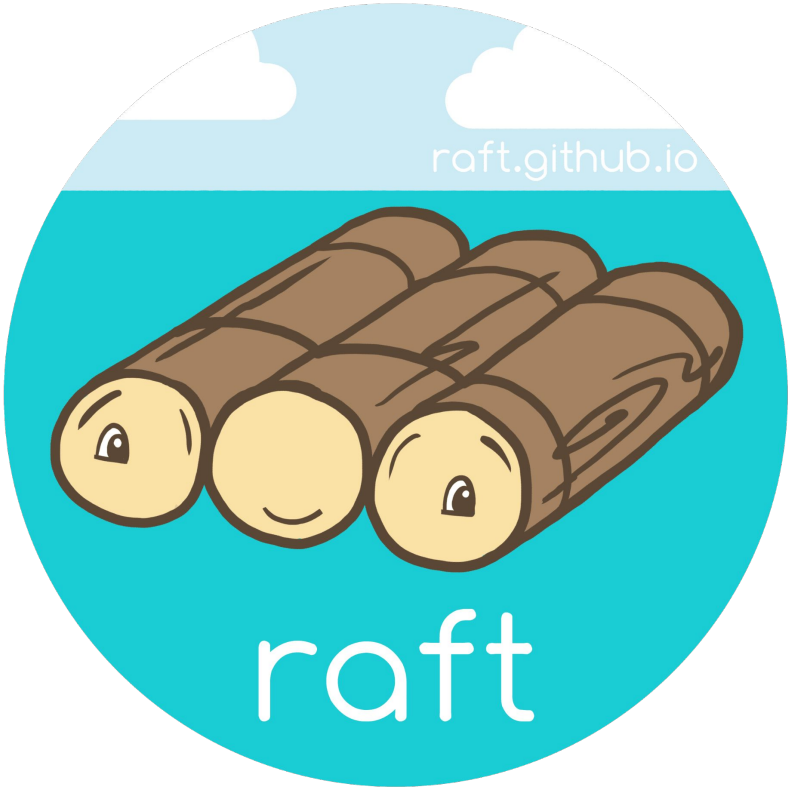
- Etcd introduction
- What is data inconsistency?
- etcd data inconsistency issue
- Lessons learned

Etcd introduction

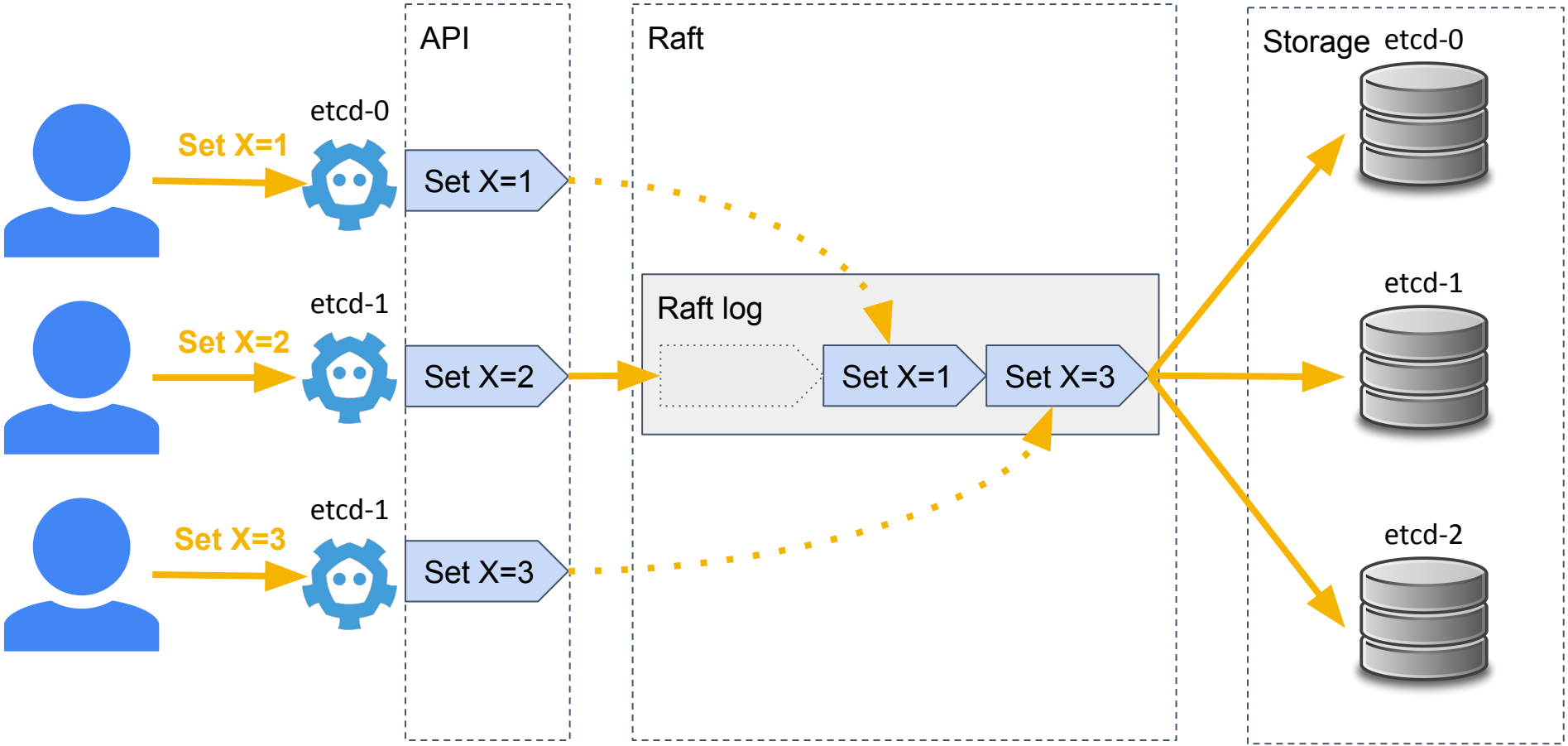
Distributed consensus



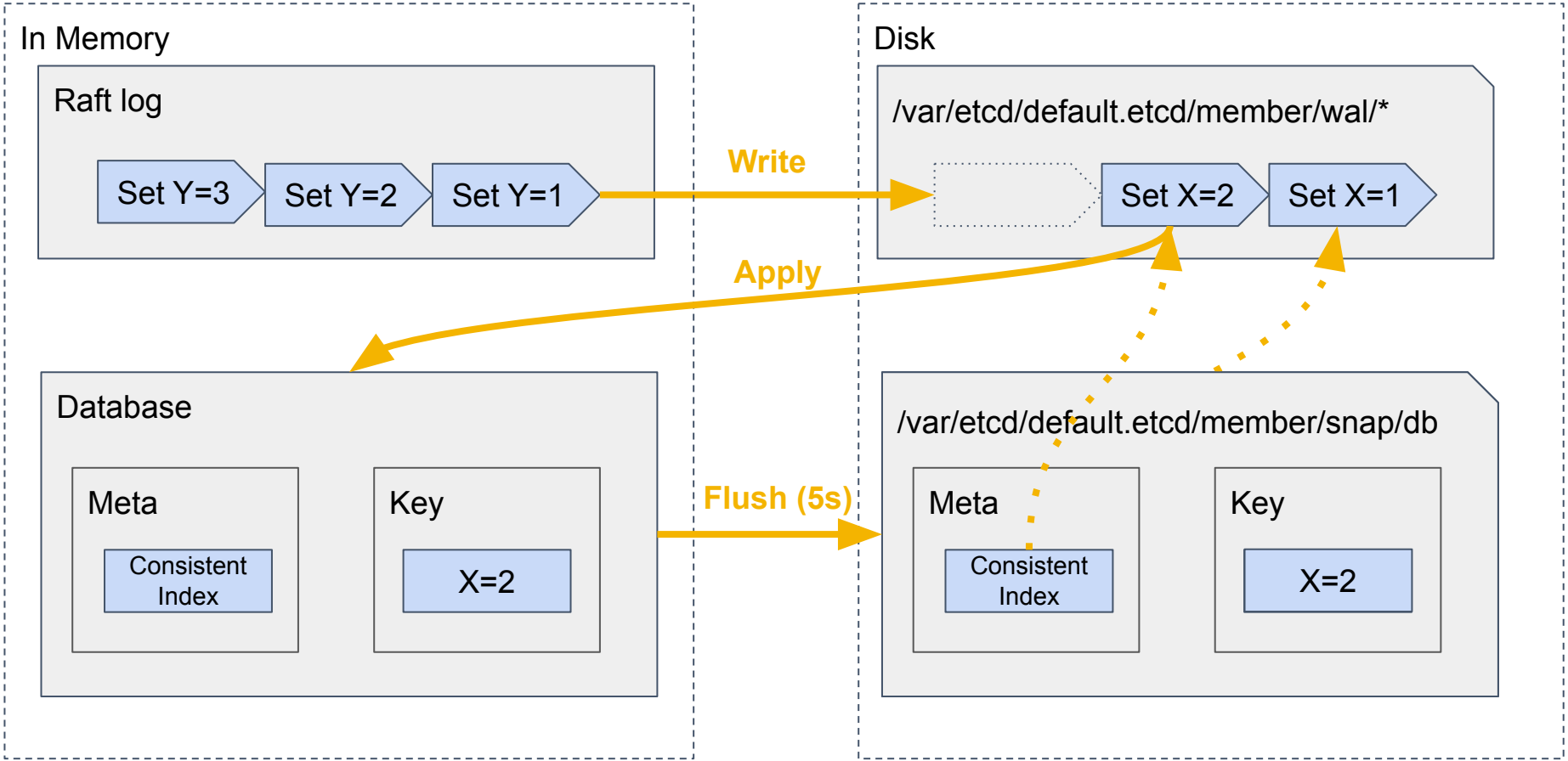
Raft



Raft log

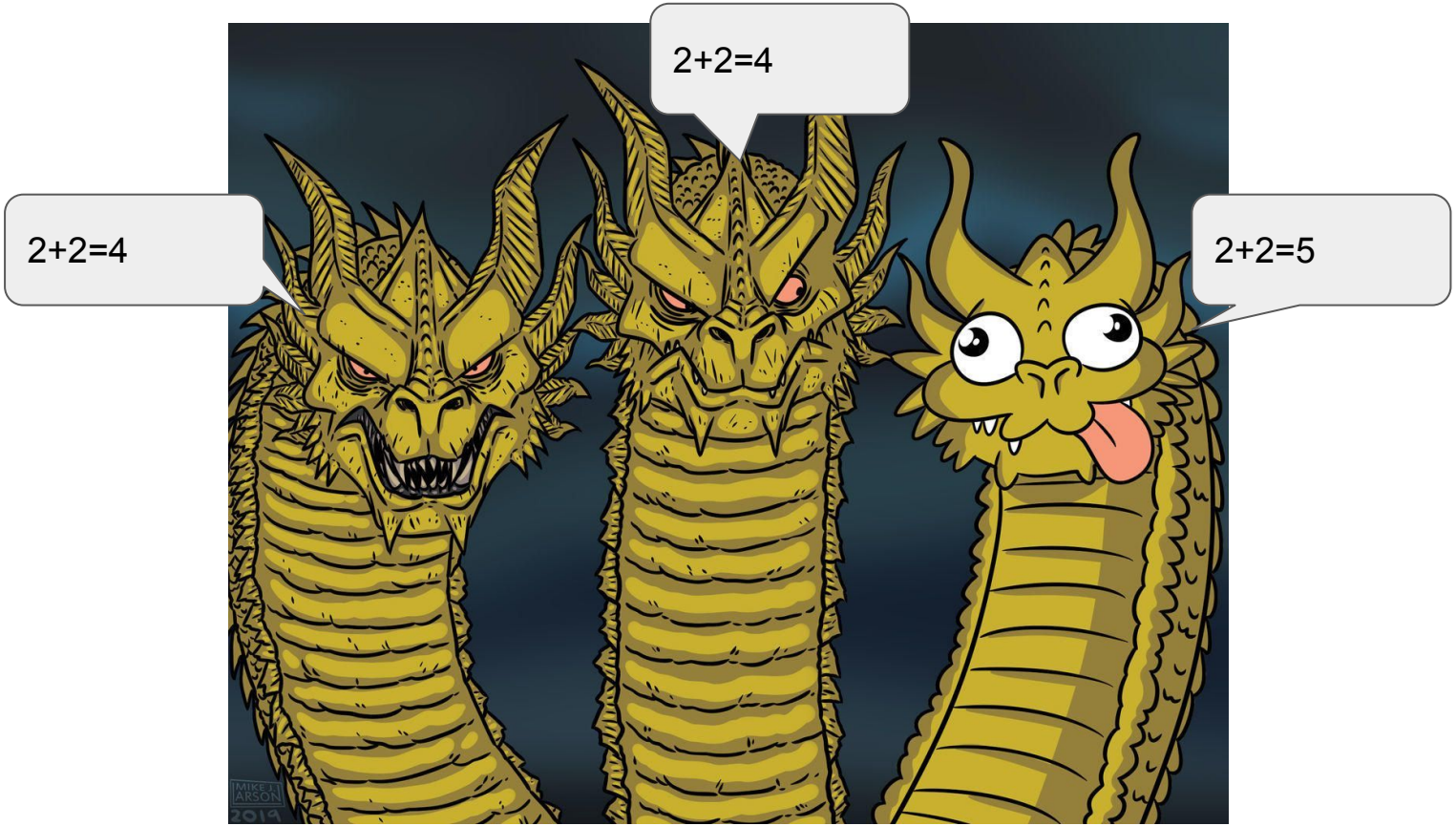

API
Raft
Storage
etcd-0
Set X=1
etcd-0
Set X=1
etcd-1
Set X=2
Raft log
Set X=2
Set X=1
Set X=3
etcd-1
etcd-1
Set X=3
Set X=3
etcd-2

Persisting etcd state



What is data inconsistency?

Data inconsistency

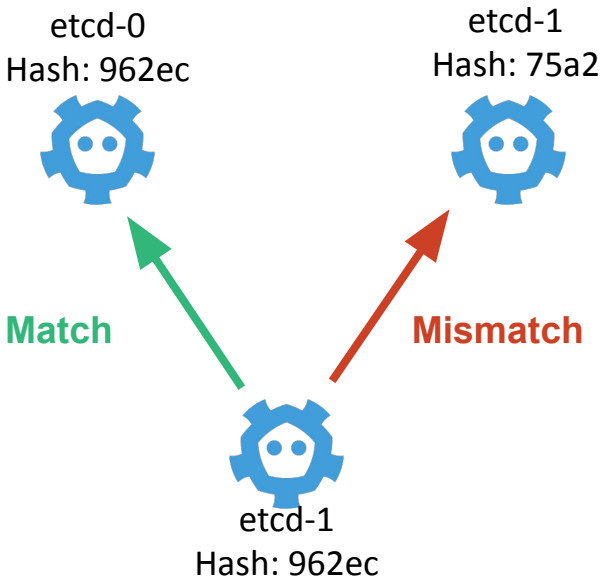


Causes

- Hardware failures
- Bugs
  - Bug in raft implementation
  - Raft log is not consistently applied
  - Database get desynchronized from raft log
- Incompatible etcd versions
  - Backward incompatible request
  - Downgrade
  - Two or larger etcd minor version skew

Detection (etcd v3.5.4 and older)

- Check database hash when member joins the cluster
- Periodic check of database hash



etcd data inconsistency issue

etcd data inconsistency issue

Issue

- https://github.com/etcd-io/etcd/issues/13766
- https://github.com/ahrtr/etcd-issues/tree/master/issues/13766

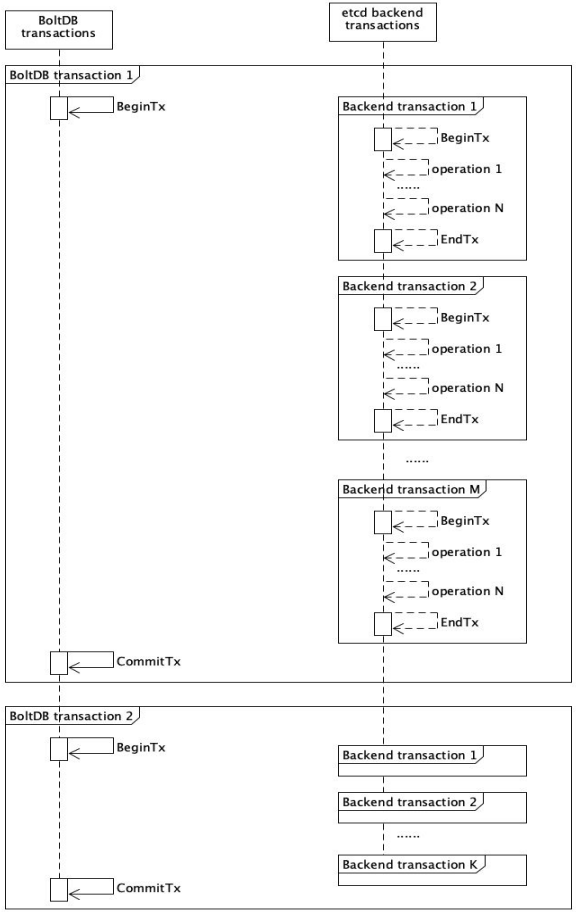
When

- etcd cluster is under high load;
- One member crashes;

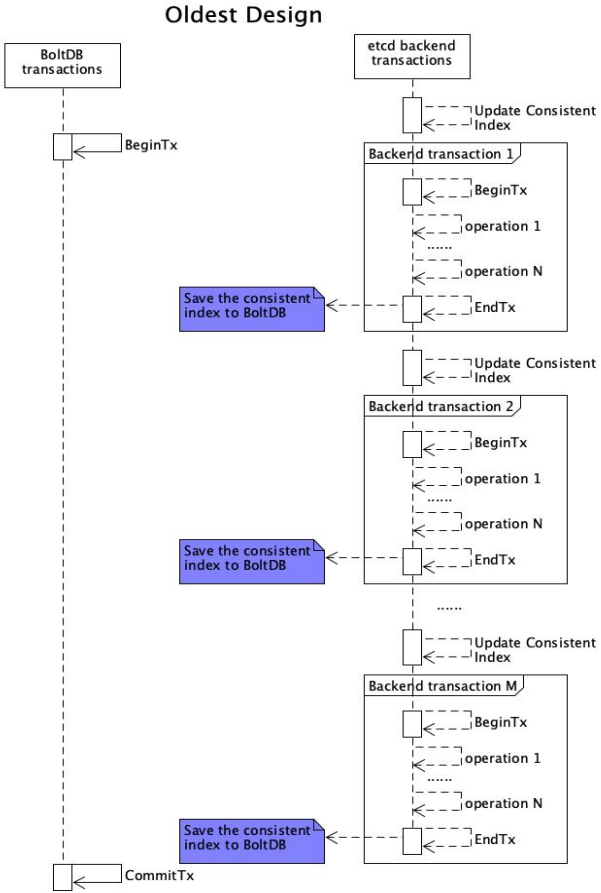
Then

- The member’s data might be inconsistent with other members.

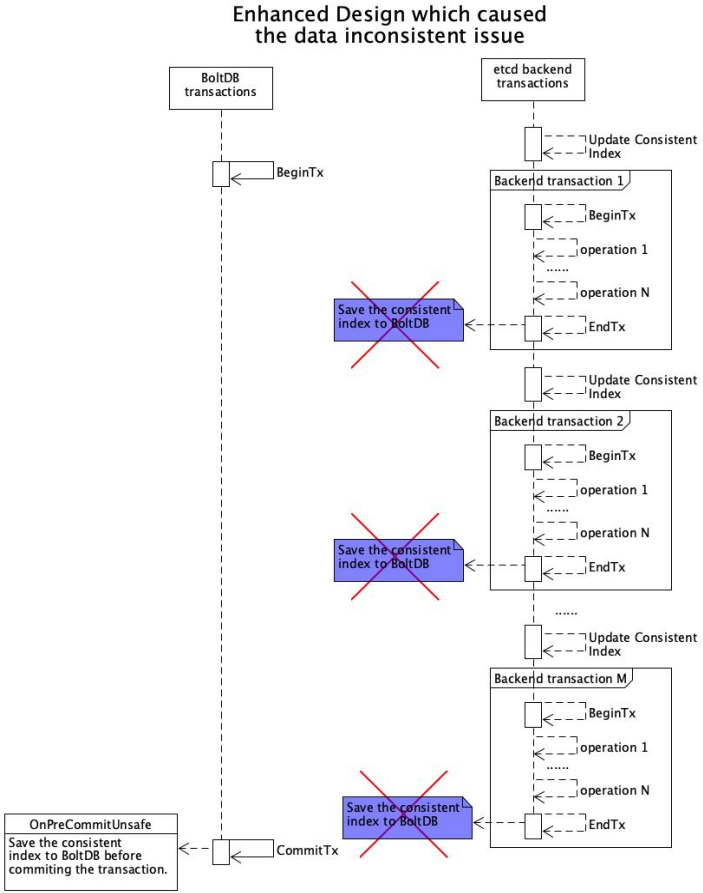
etcd transaction vs BoltDB transaction

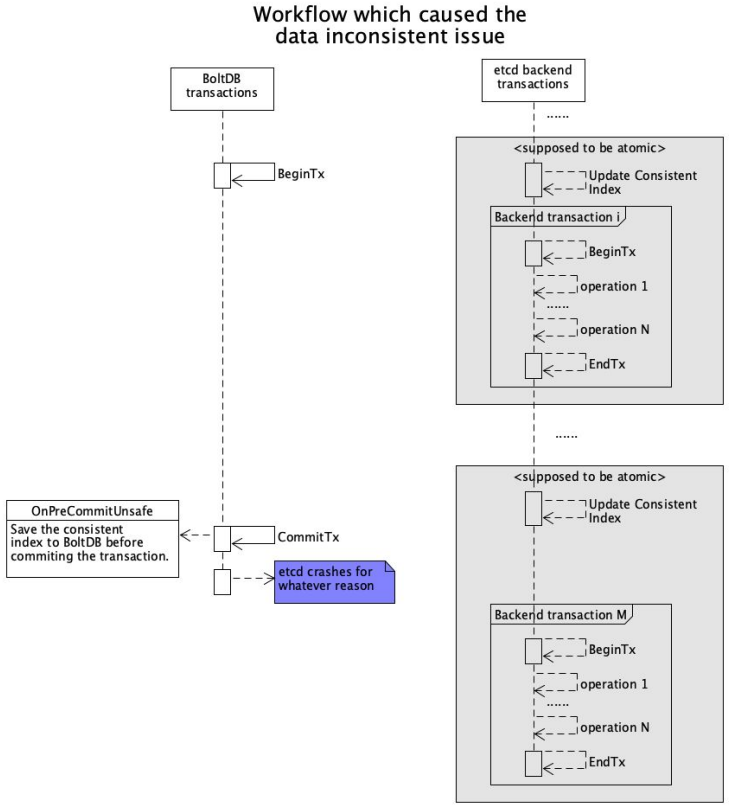


Original design before the issue

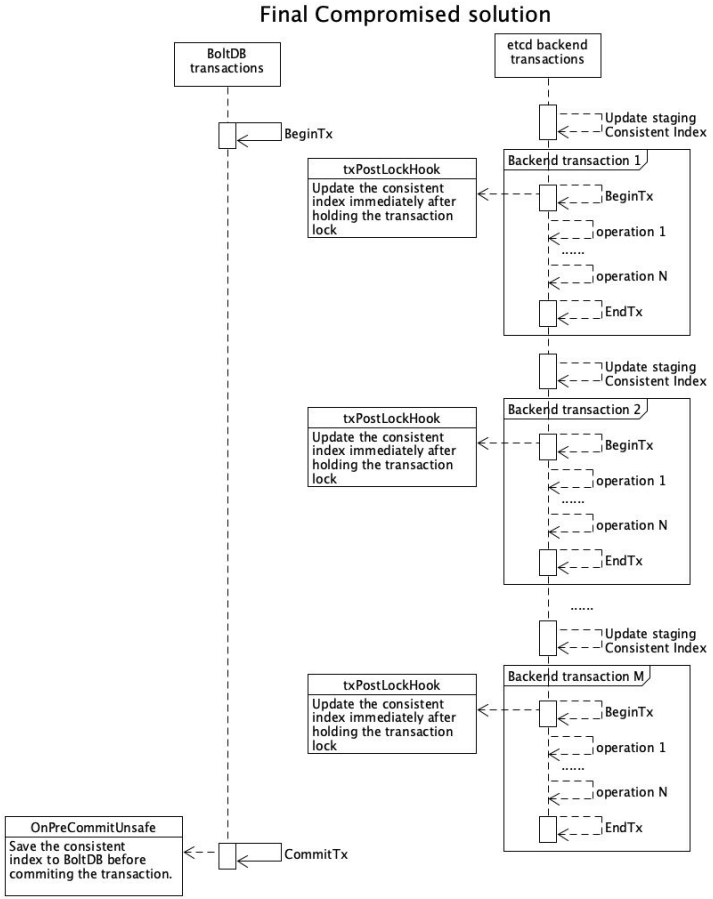


Enhanced design which caused the issue

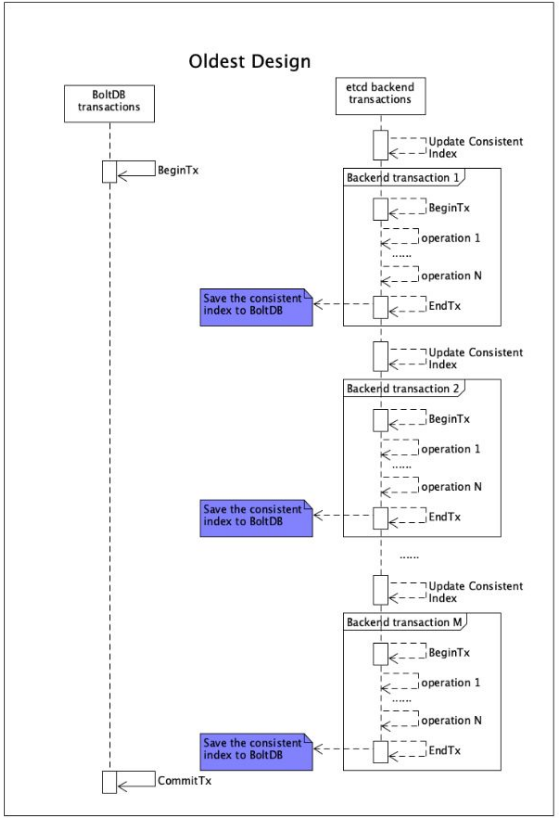


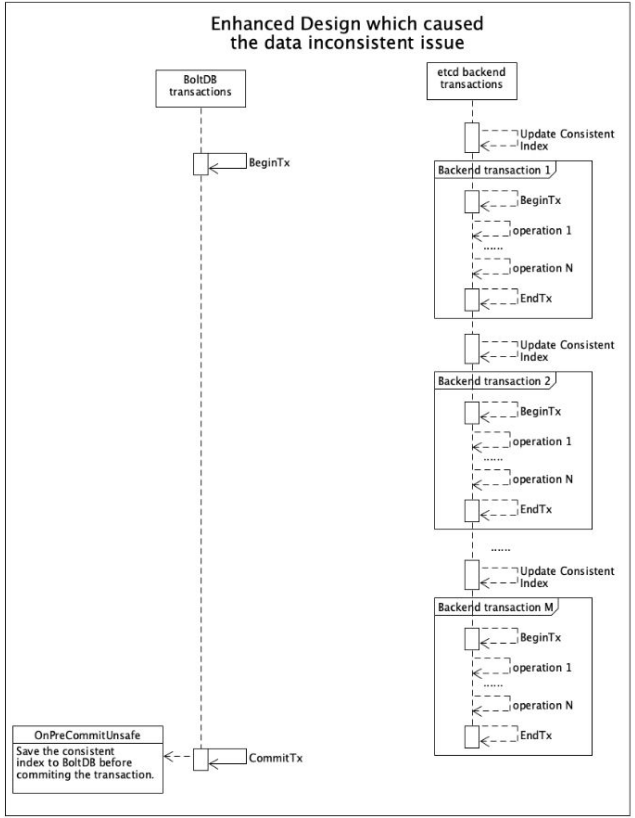


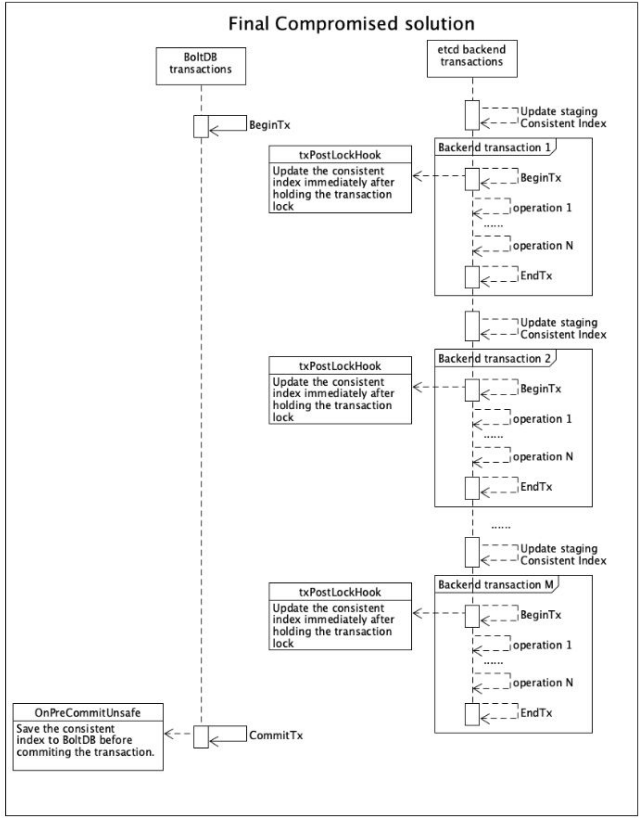
Final compromised solution



Three solutions comparison







The enhancement is unnecessary!

Long-term goal: get rid of all hooks.

Lessons learned

What went wrong

- Last minor etcd release was not properly qualified
- Missing of tests that could detect this class of issues
- No user adoption of data inconsistency detection as it was never graduated
- Fixing the issue took over 2 weeks and required multiple attempts

Action items

Types

- Prevent
- Detect
- Mitigate

Priority

- Critical
- Important
- Long term

Action items - Critical

Prevent

- Etcd testing can reproduce historical data inconsistency issues

Detect

- Etcd detects data corruption by default

Action items - Important

Prevent

- Etcd testing is high quality, easy to maintain and expand
- Etcd apply code should be easy to understand and validate correctness
- Critical etcd features are not abandoned
- Etcd is continuously qualified with failure injection

Detect

- Etcd can reliably detect data corruption
- Etcd check consistency of snapshots sent between leader and followers

Mitigate

- Etcd recovery from data inconsistency procedures are documented and tested

Action items - Long term

Mitigate

- Etcd can immediately detect and recover from data corruption

Status of work

- New data inconsistency check in v3.5.5
  - More reliable - handles slow followers
  - Much cheaper - can be run every couple of minutes
- Automated linearizability tests with failure injection
  - Easily reproduce historical issues

Personal learnings

- Contributor documentation is as (or more) important than user documentation
- Contributions rarely align with what is important for a project

BUILDING FOR THE ROAD AHEAD

DETROIT 2022

Thank you!

References

- https://github.com/etcd-io/etcd/blob/main/Documentation/postmortems/v3.5-data-inconsistency.md
- https://github.com/ahrtr/etcd-issues/tree/master/issues/13766

KubeCon | CloudNativeCon
North America 2022

BUILDING FOR THE ROAD AHEAD

DETROIT 2022

Please scan the QR Code above to leave feedback on this session