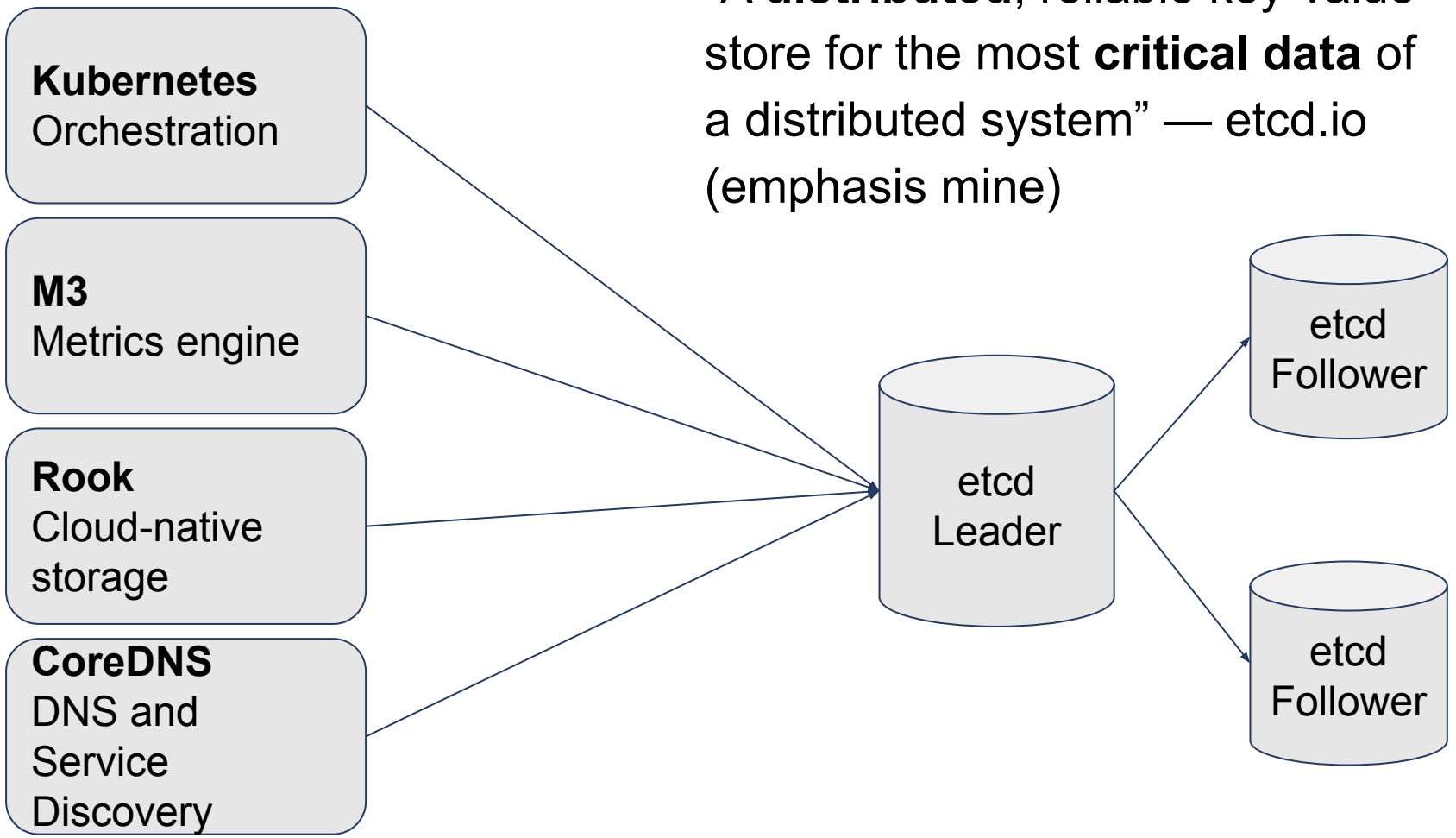
LSKV: Democratising Confidential Computing from the Core

Andrew Jeffery – University of Cambridge
andrew.jeffery@cst.cam.ac.uk

FOSDEM ‘23

Starting with etcd – the distributed key-value store

"A **distributed**, reliable key-value store for the most **critical data** of a distributed system" — etcd.io (emphasis mine)

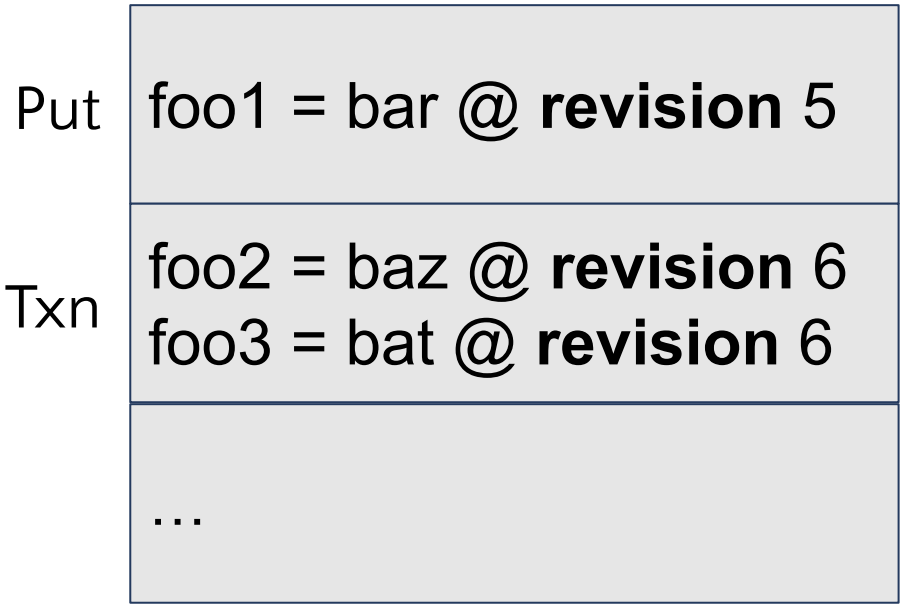


The core etcd API

- Put(key, value)
- Range(key, **range_end**, ?**revision**)
- DeleteRange(key, **range_end**)
- Txn(^)

- LeaseGrant(ttl)
- LeaseKeepAlive(id)
- LeaseRevoke(id)

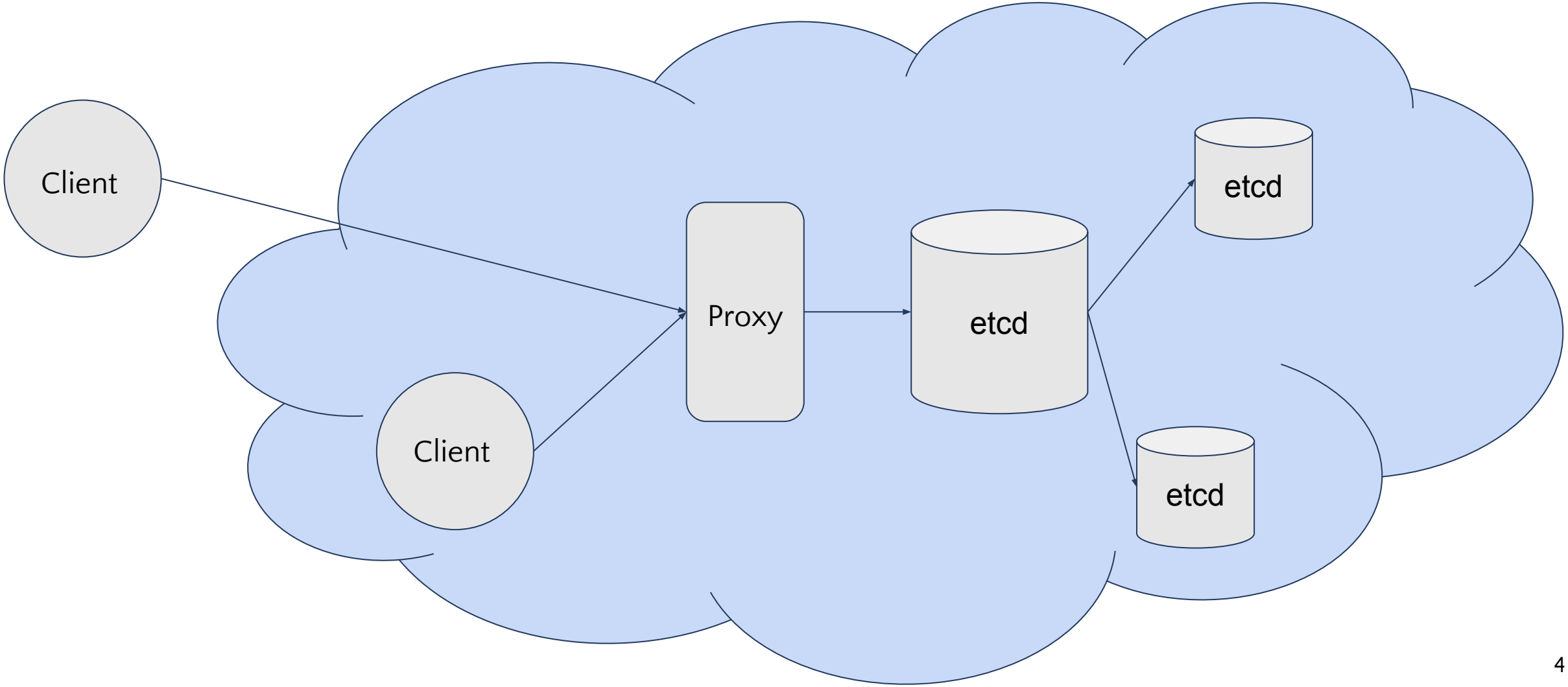
- Watch(key, range_end, ?**revision**)



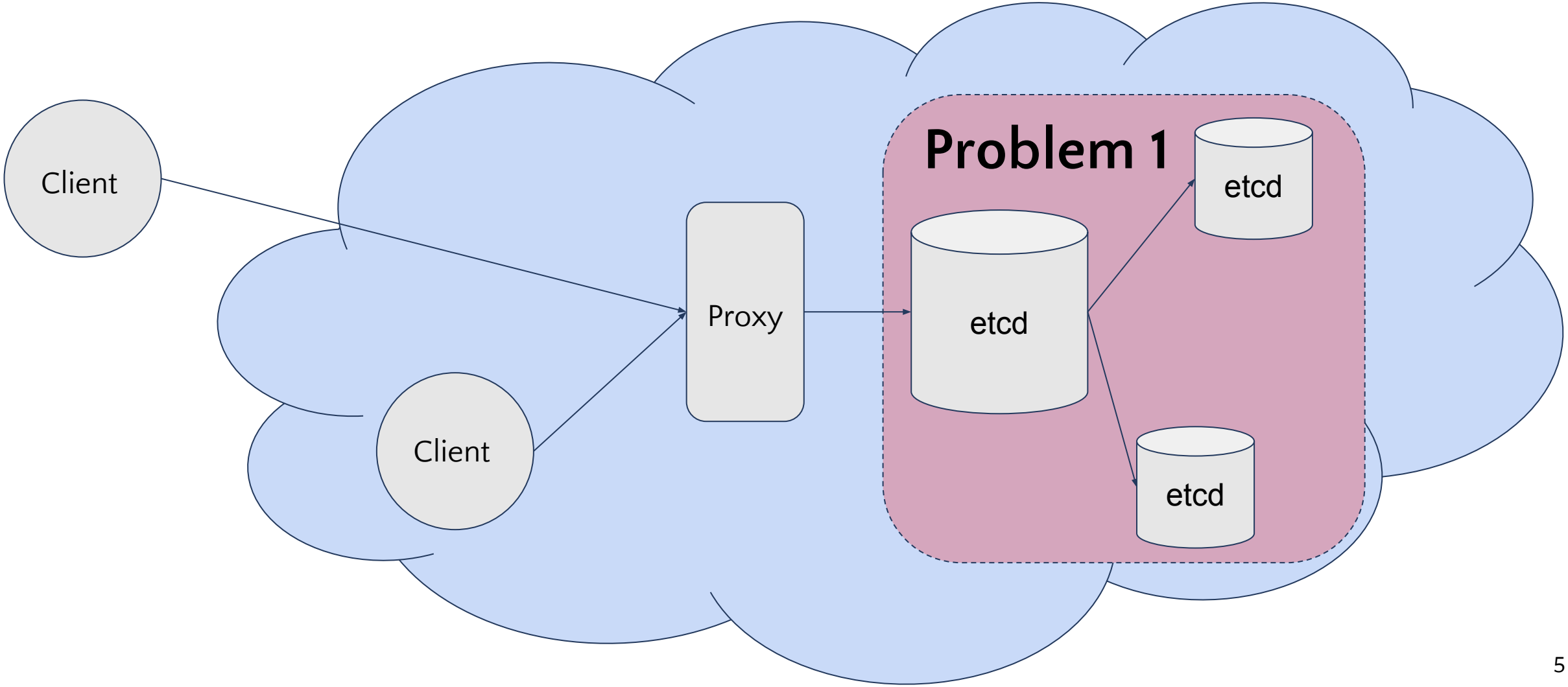
Range(foo1, foo4) = [foo1, foo2, foo3]

Range(foo1, foo4, **5**) = [foo1]

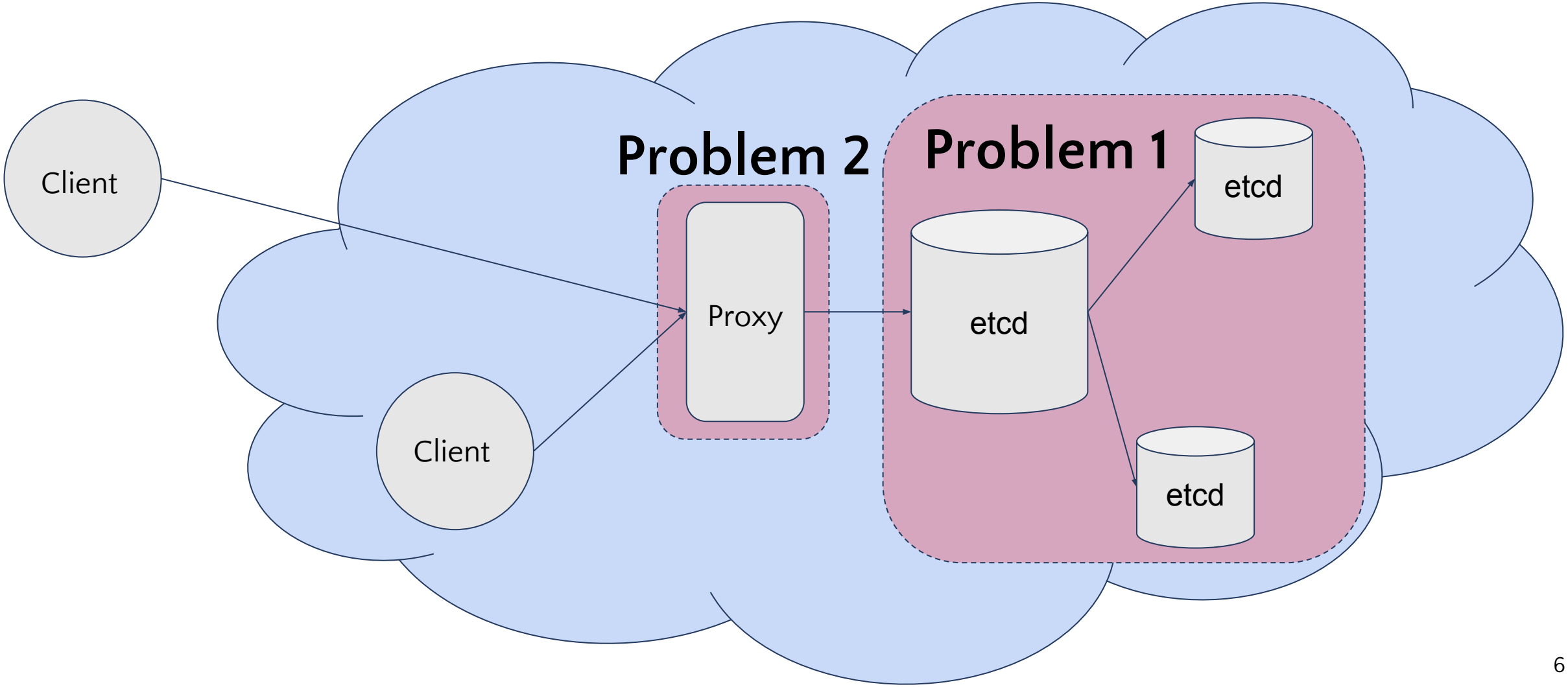
Datastores in the trusted cloud



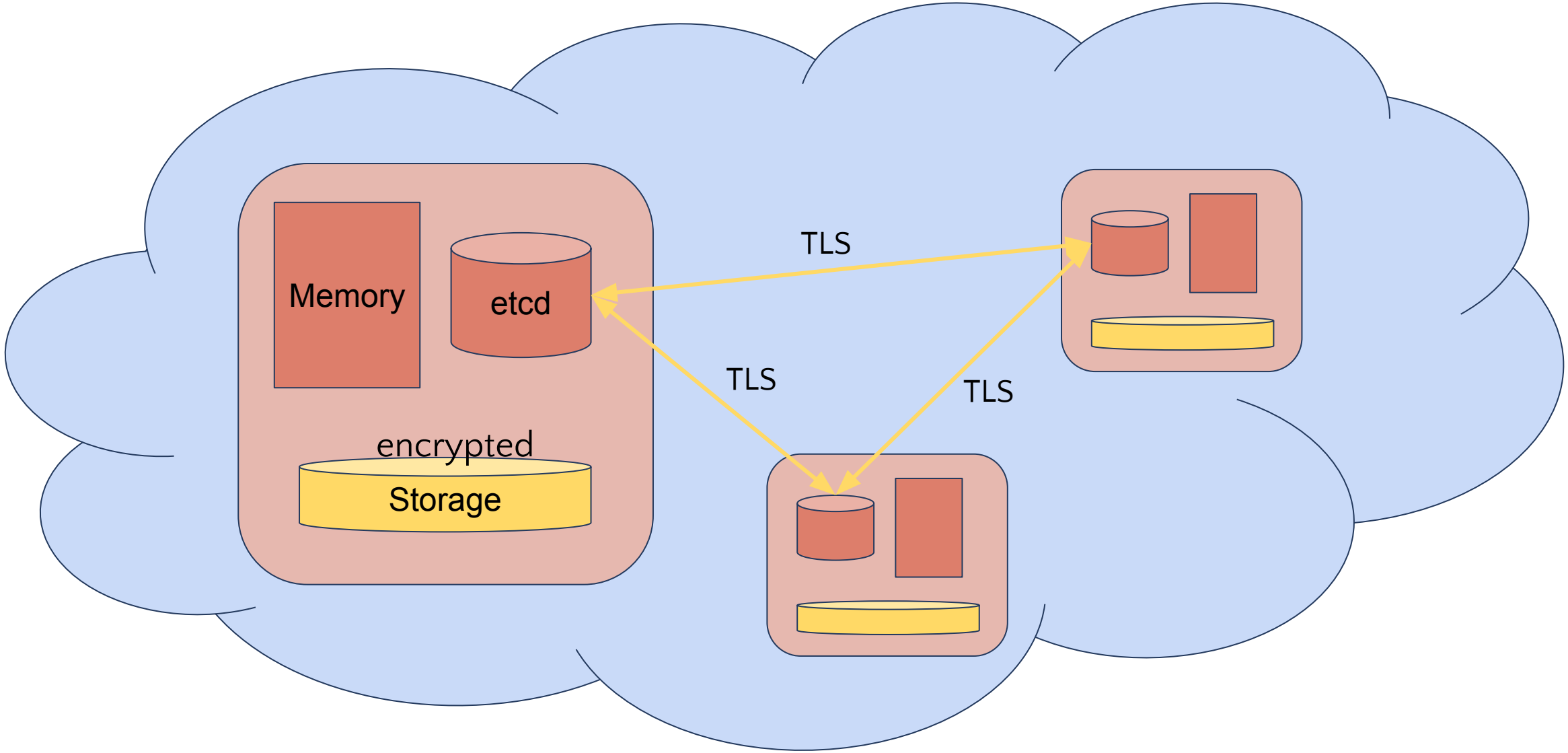
Problems in the untrusted cloud



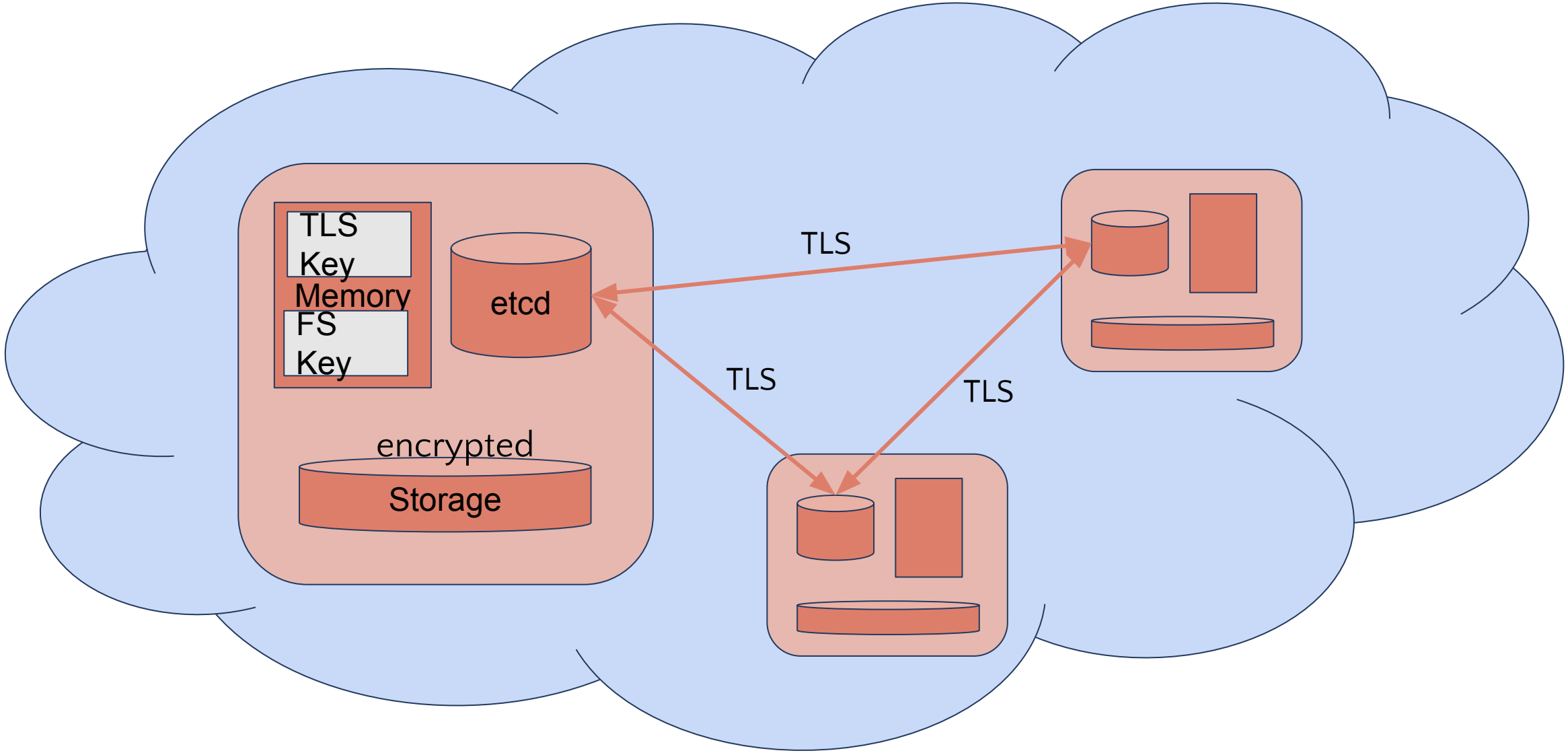
Problems in the untrusted cloud



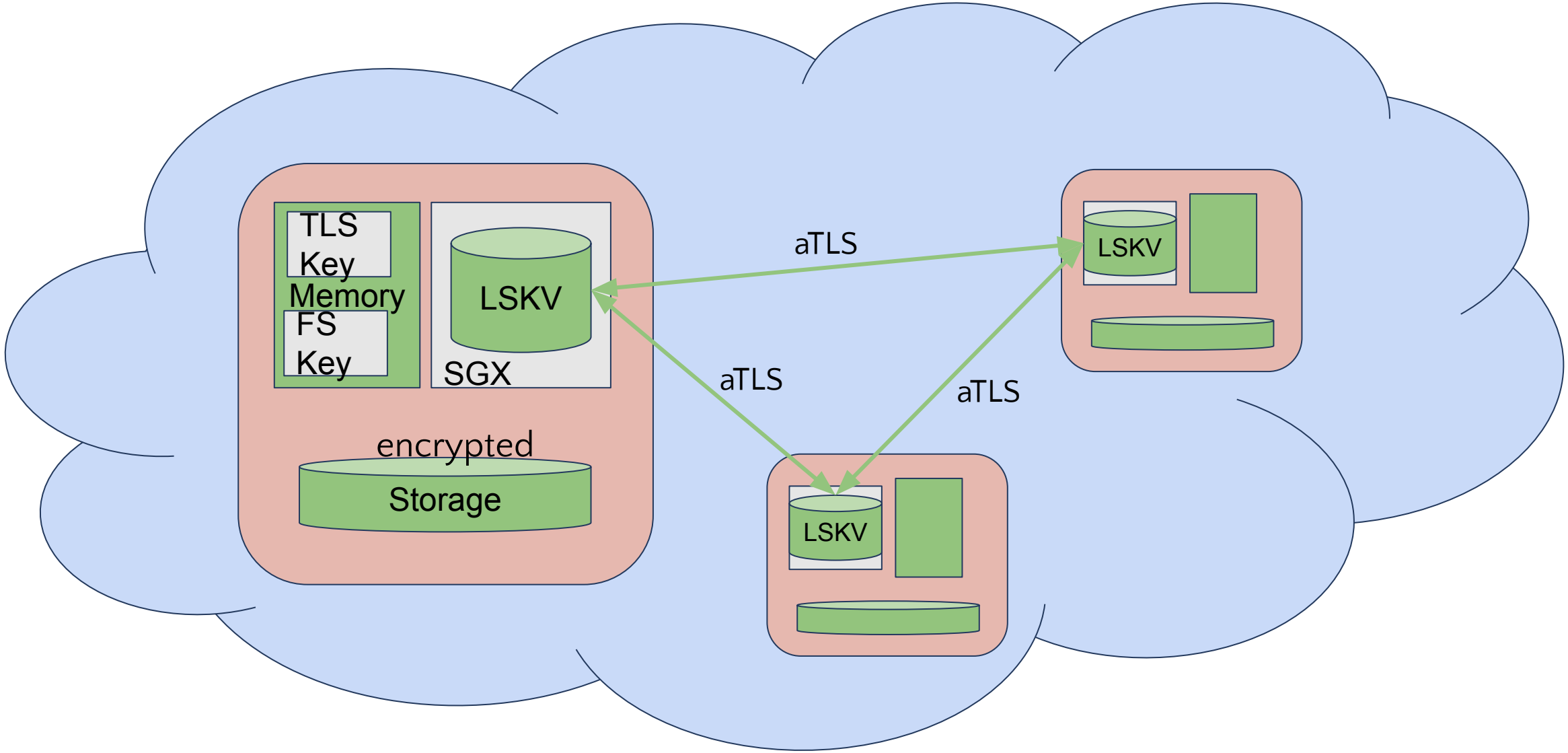
Problem 1 – Trusted Cloud?



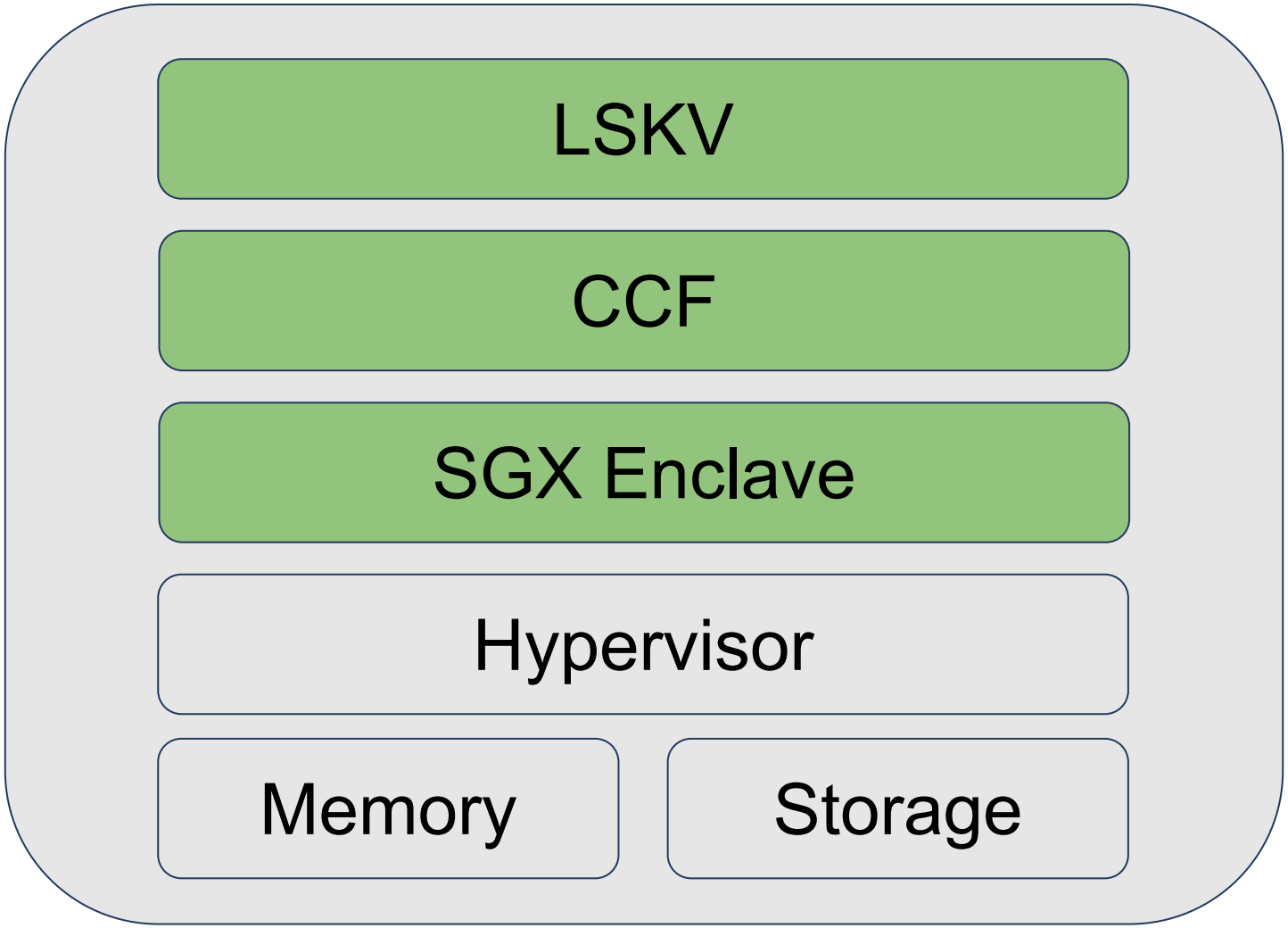
Problem 1 – Trusted Cloud?



Solution 1 – **Un**trusted Cloud



LSKV: The Ledger-backed Secure Key-Value store

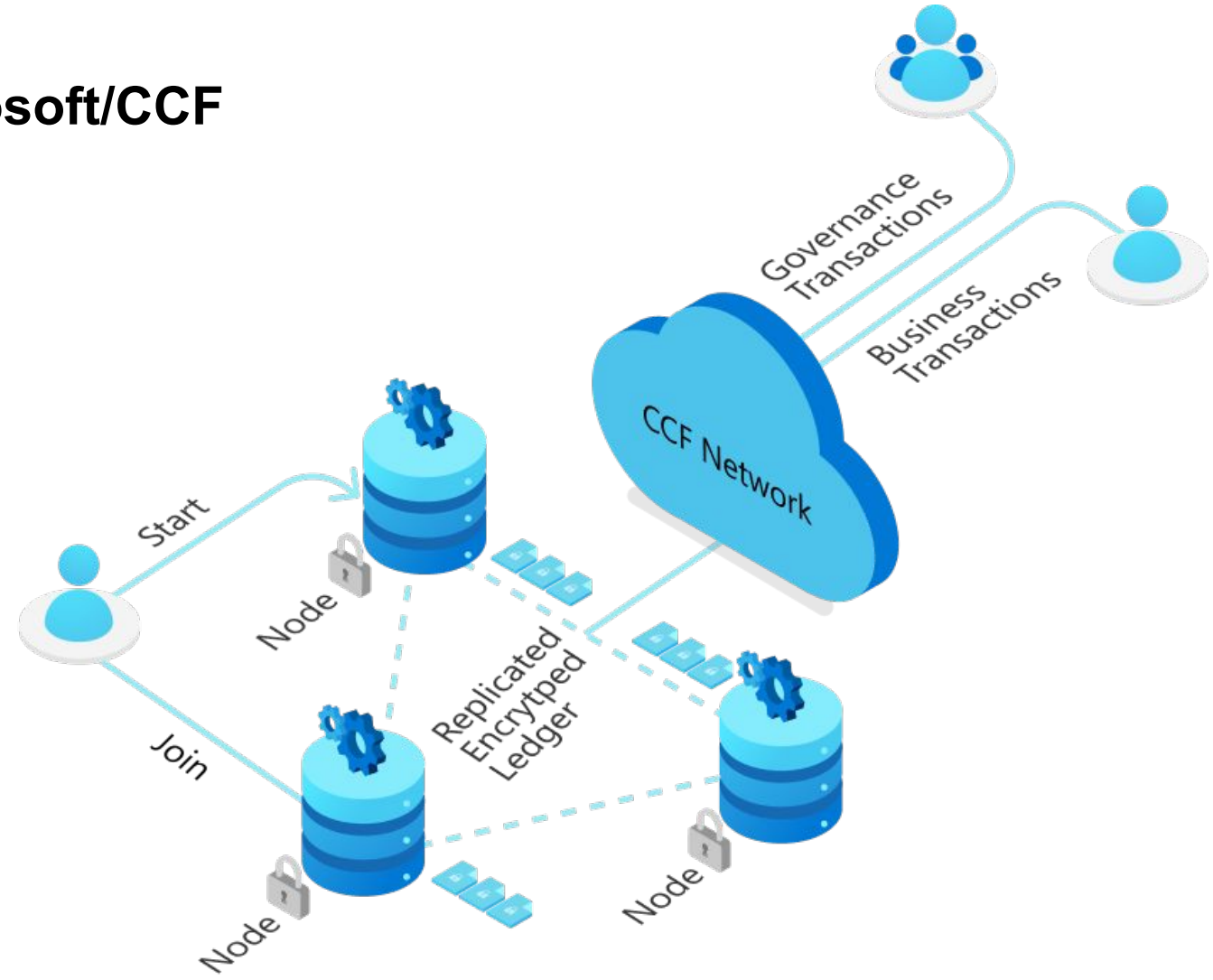


CCF: the Confidential Consortium Framework

github.com/microsoft/CCF

or

ccf.dev

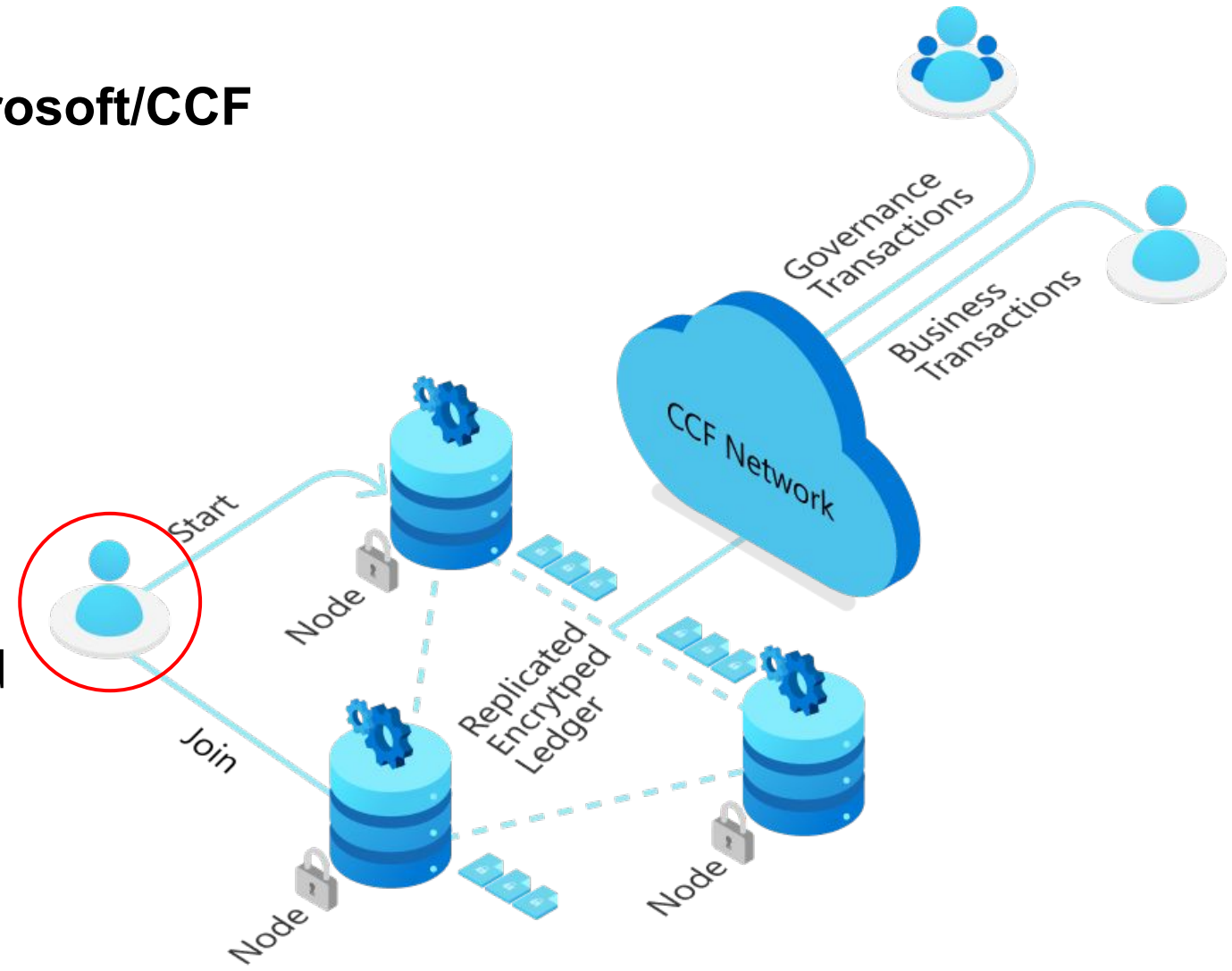


CCF: the Confidential Consortium Framework

github.com/microsoft/CCF

or

ccf.dev



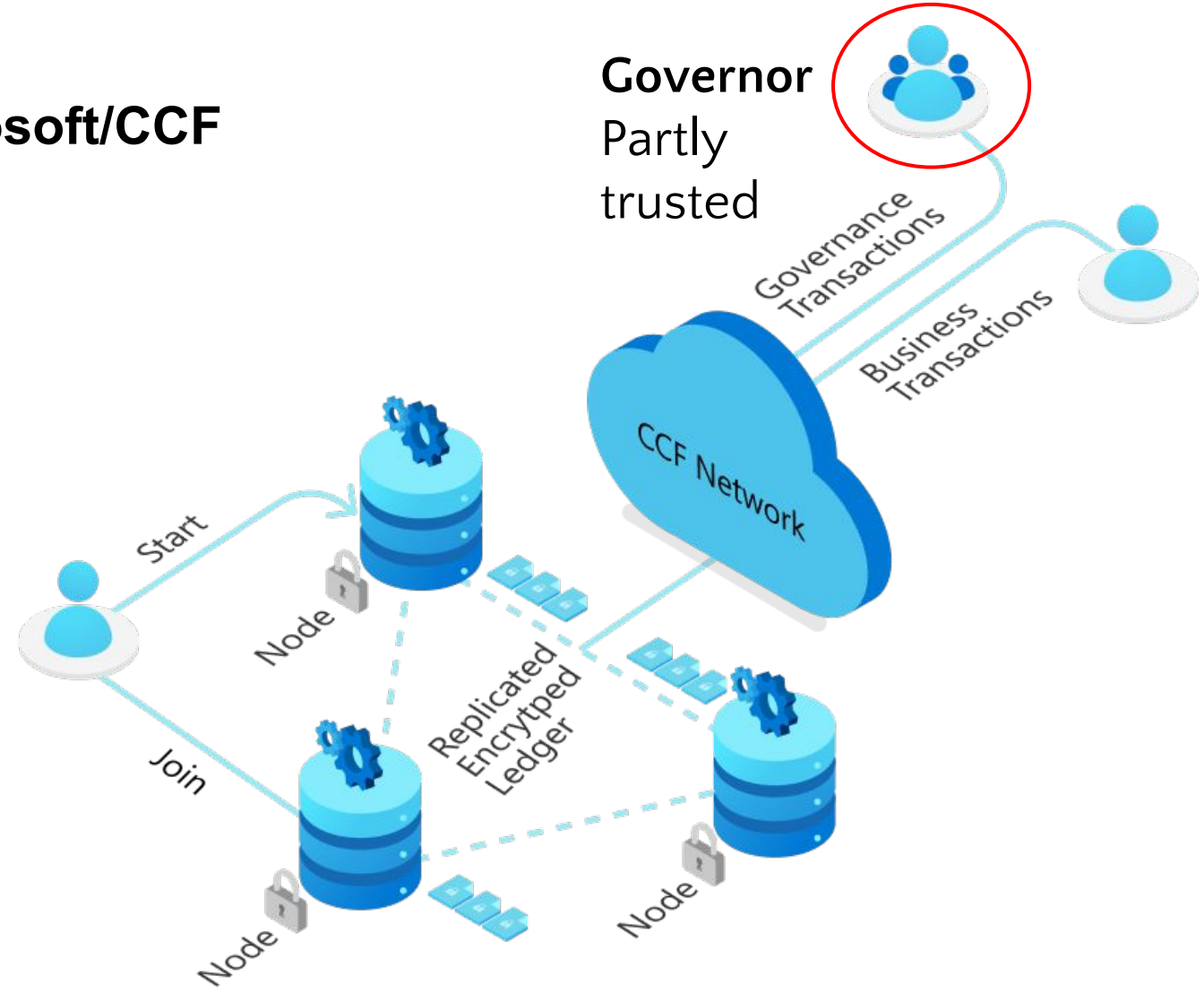
Operator
Untrusted

CCF: the Confidential Consortium Framework

github.com/microsoft/CCF

or

ccf.dev



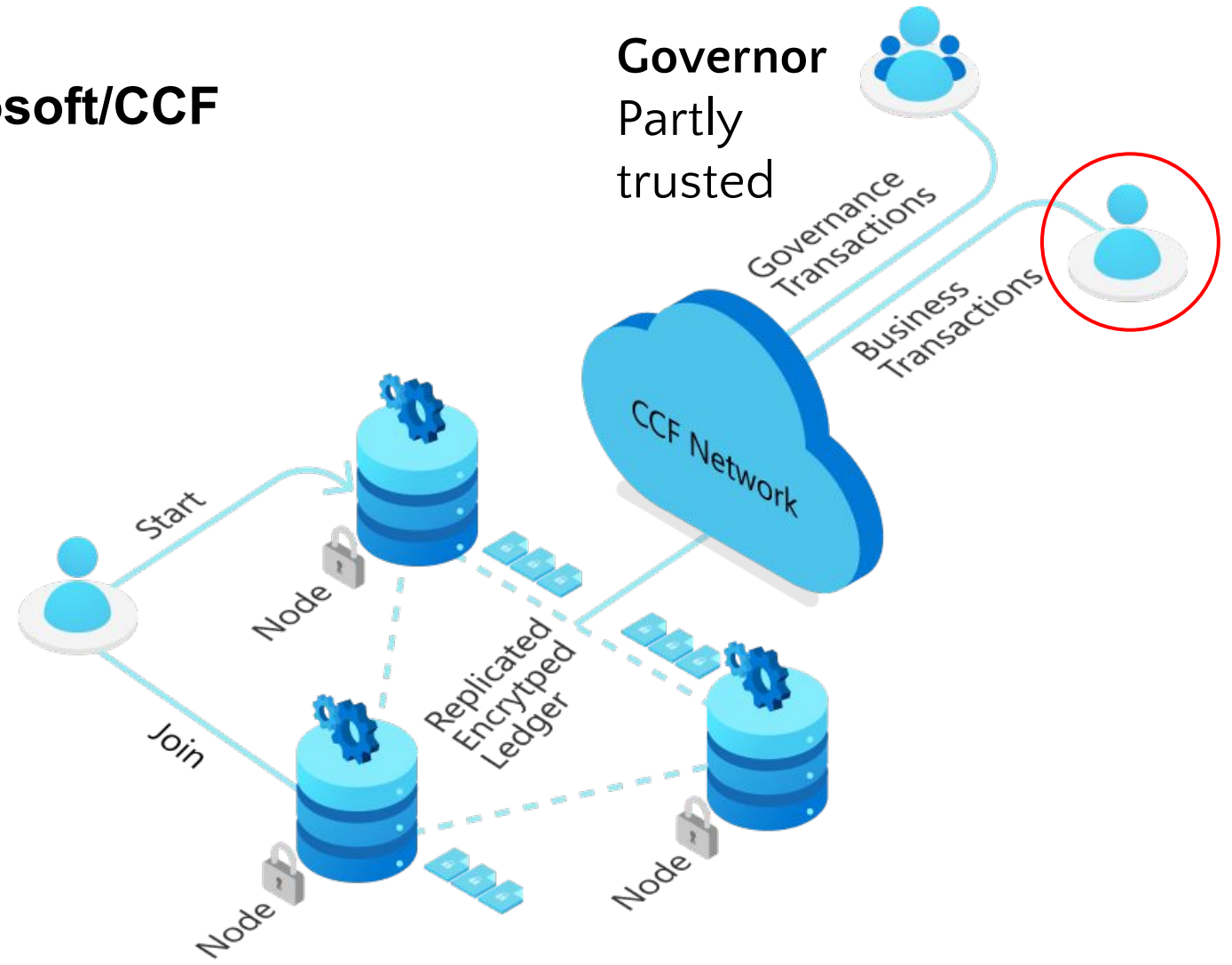
Operator
Untrusted

CCF: the Confidential Consortium Framework

github.com/microsoft/CCF

or

ccf.dev

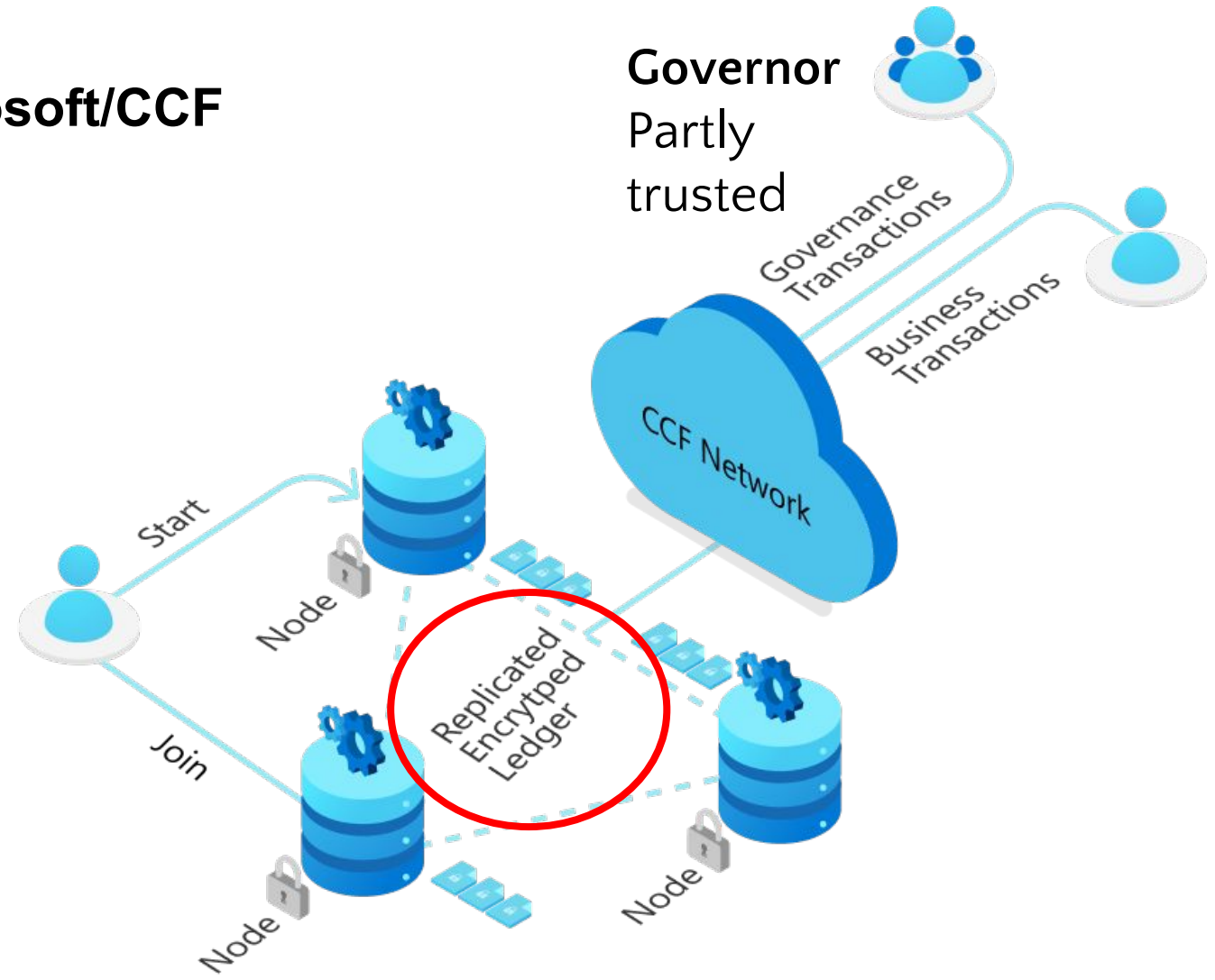


CCF: the Confidential Consortium Framework

github.com/microsoft/CCF

or

ccf.dev



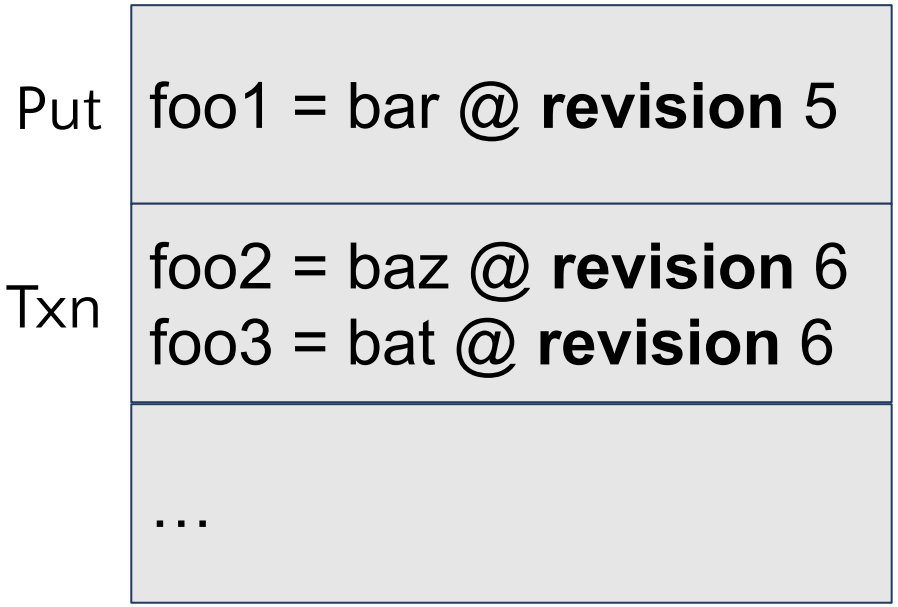
github.com/microsoft/LSKV

LSKV has an etcd-compatible API

- Put(key, value)
- Range(key, **range_end**, ?**revision**)
- DeleteRange(key, **range_end**)
- Txn(^)

- LeaseGrant(ttl)
- LeaseKeepAlive(id)
- LeaseRevoke(id)

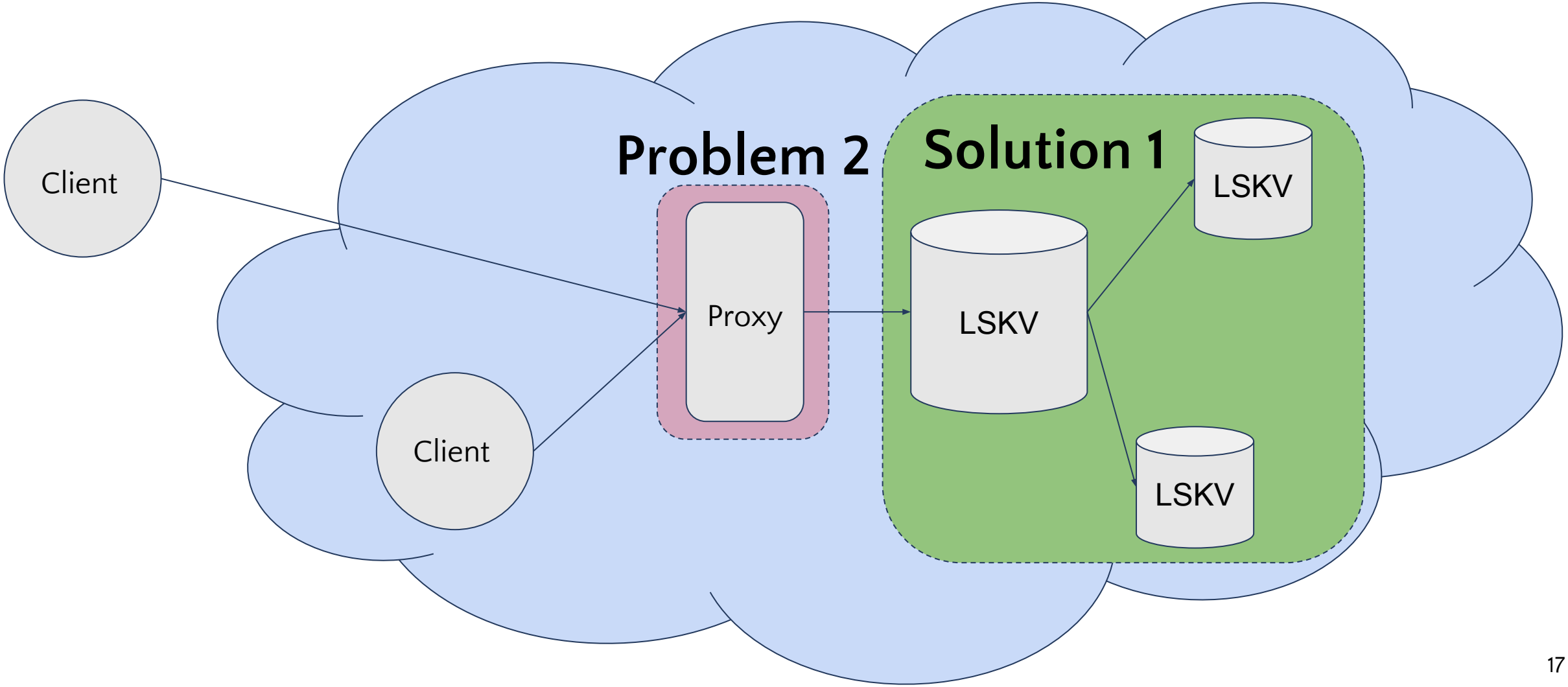
- Watch(key, range_end, ?**revision**)*



Range(foo1, foo4) = [foo1, foo2, foo3]

Range(foo1, foo4, **5**) = [foo1]

Solution 1 – Confidentiality with compatible API

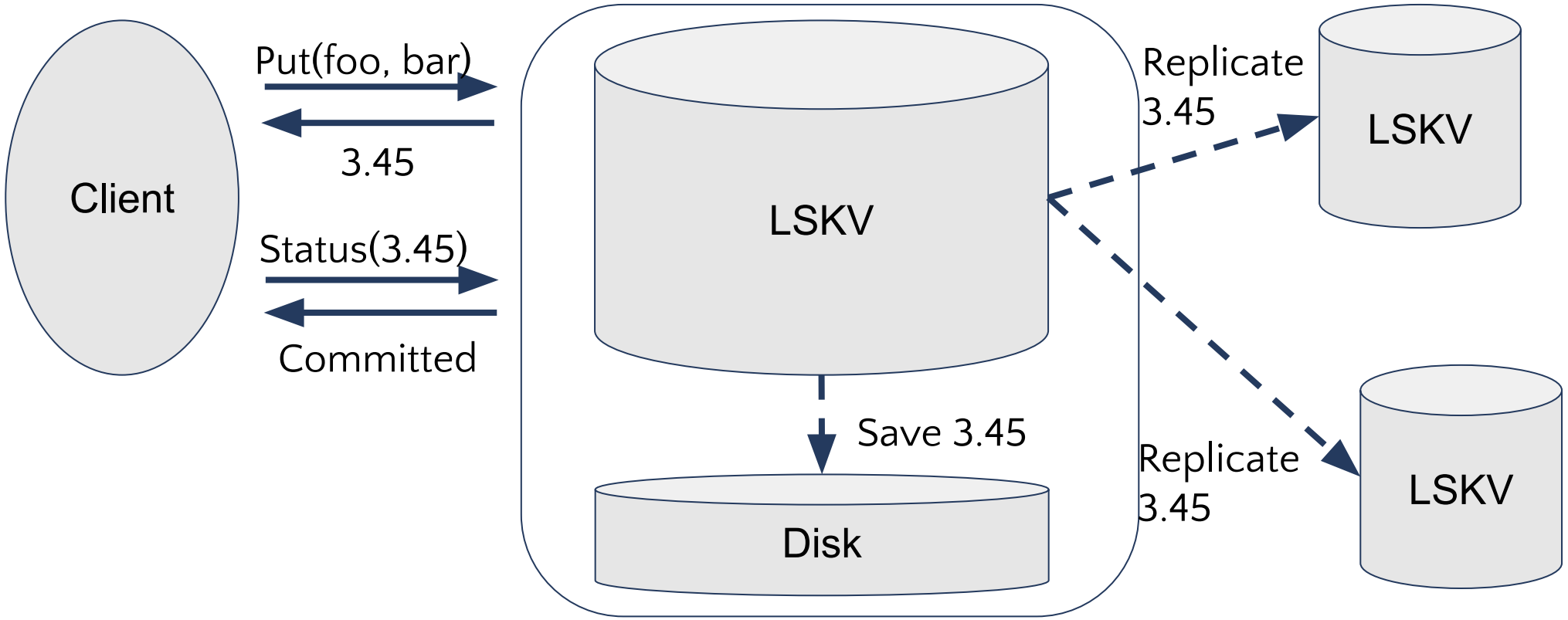


github.com/microsoft/LSKV

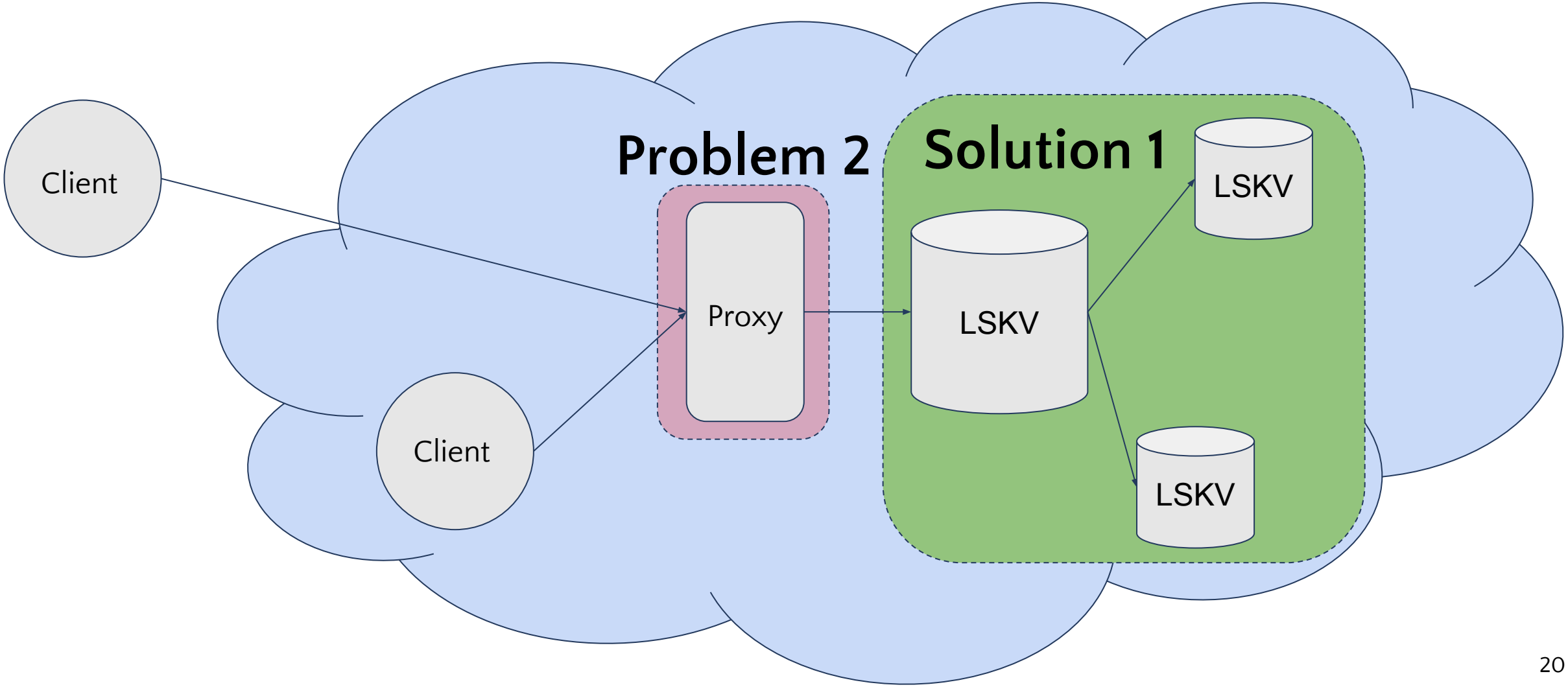
Trade offs

	etcd	LSKV
Consistency	Strong	**Optimistic**
Confidential	No	**Yes**
Management transparency	Missing	**Available** on the ledger
API	etcd API	**etcd API** + extras

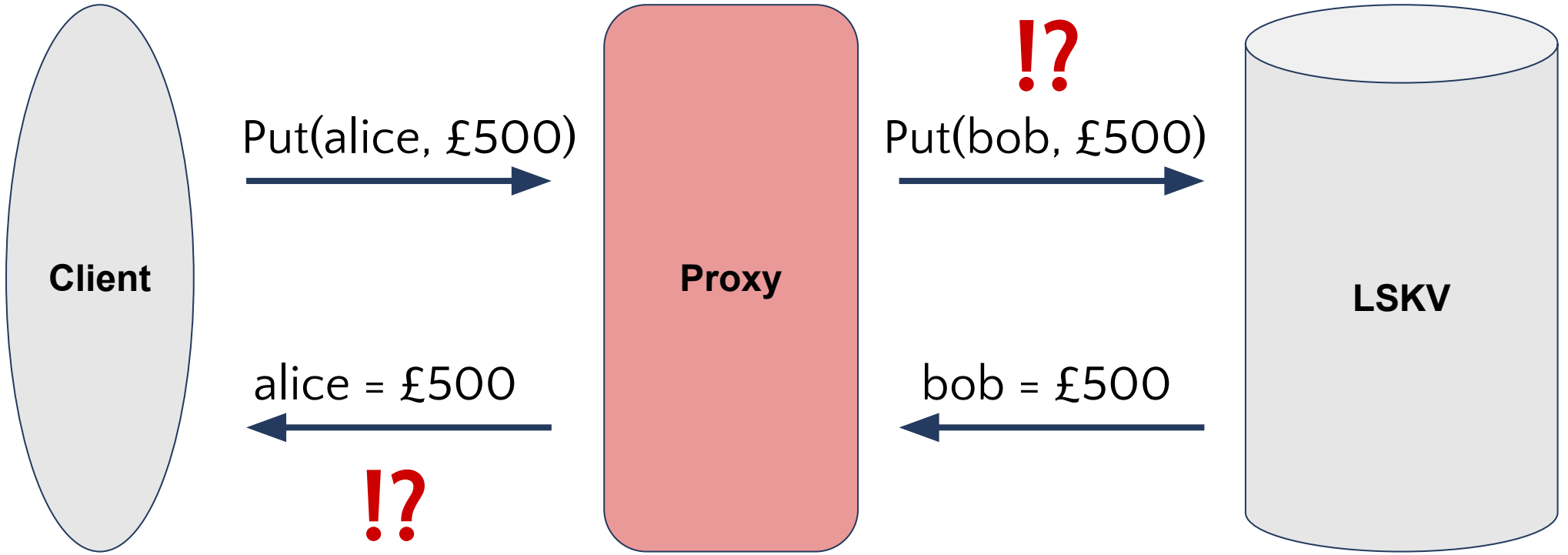
Technical interlude – Optimistic consistency



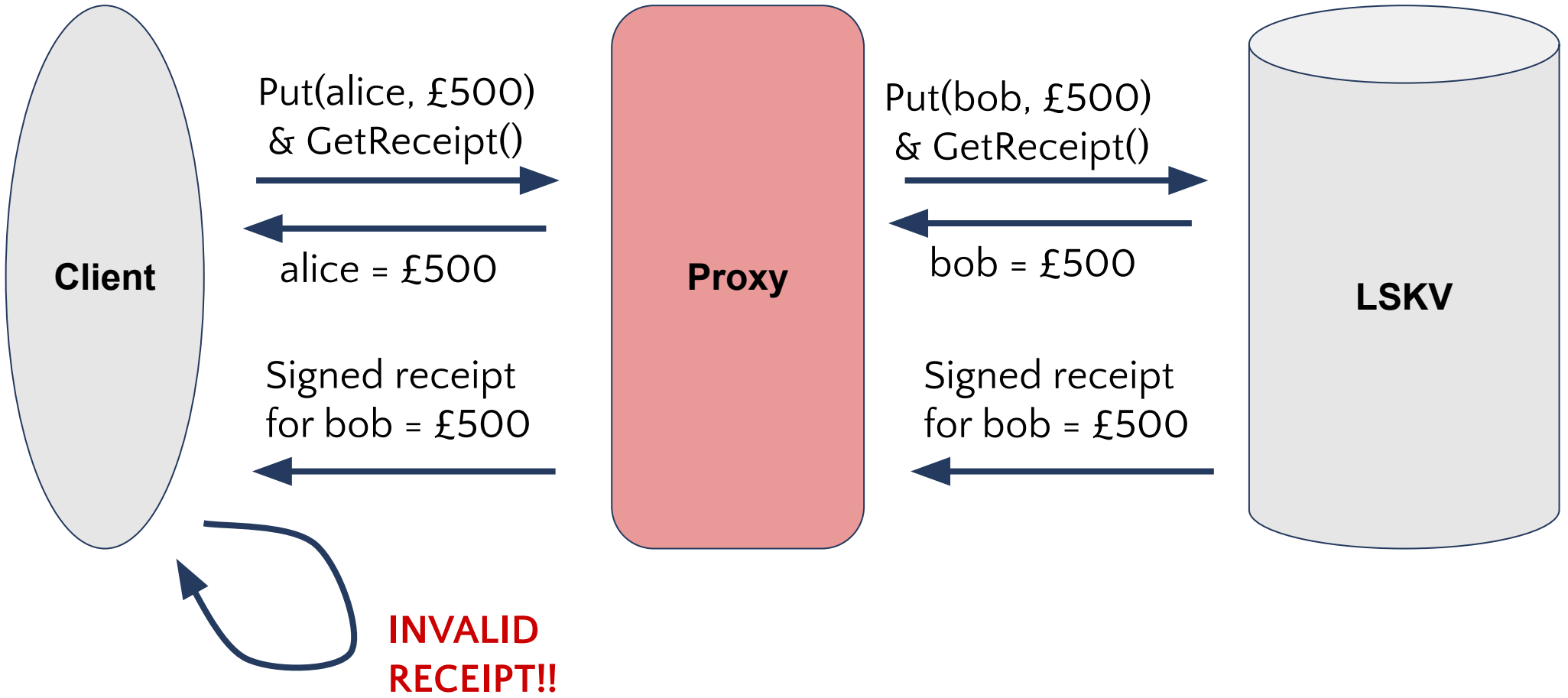
Onto problem 2! Mean proxies



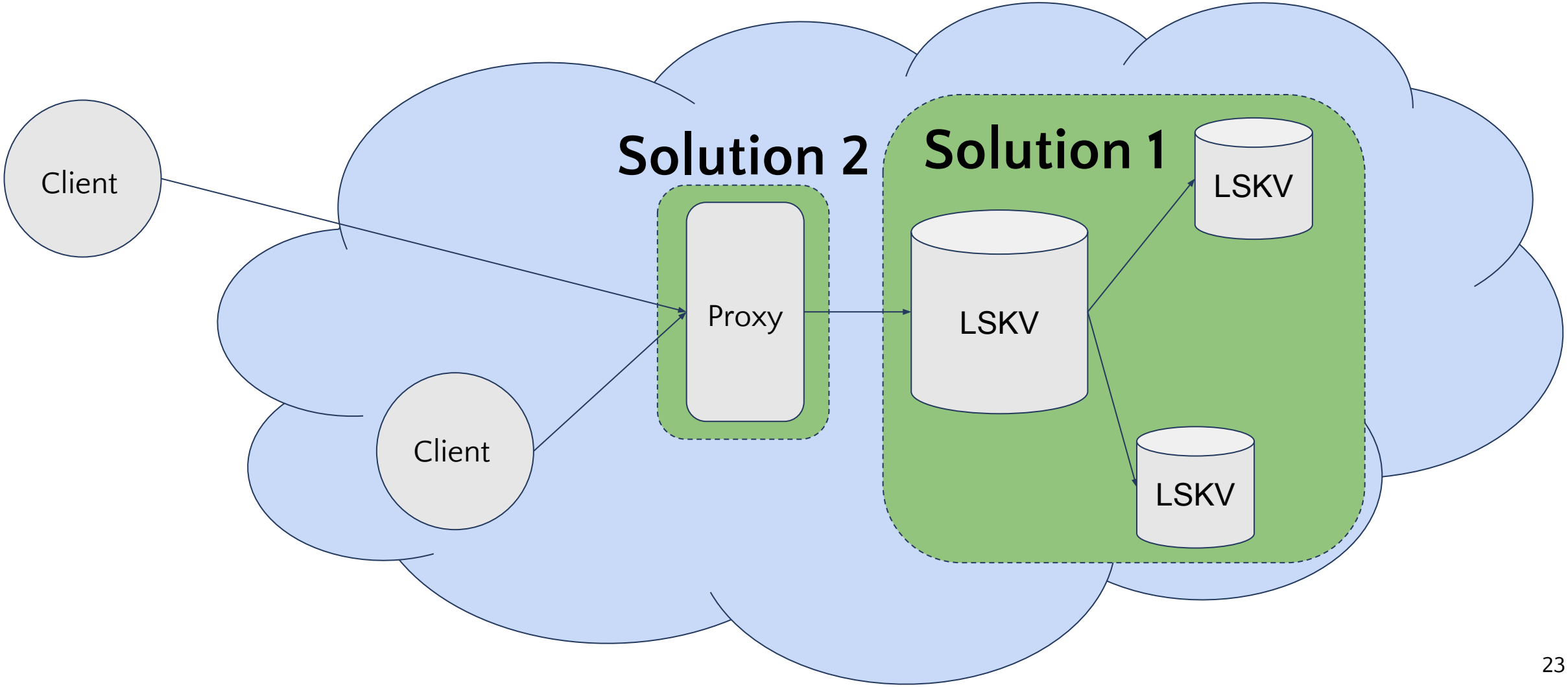
Problem 2 – Proxies can be mean



Solution 2: Don't trust the proxy – get a receipt



Solution 2 – Get receipts



github.com/microsoft/LSKV

Sorry, I missed that

- Current datastores aren't suited for confidential operation
- LSKV is a new **confidential datastore**, built on CCF with an etcd-compatible API
- LSKV can highlight **untrustworthy proxies** using receipts
- Oh, and it is fast: **3.5x throughput, 50% latency** vs etcd

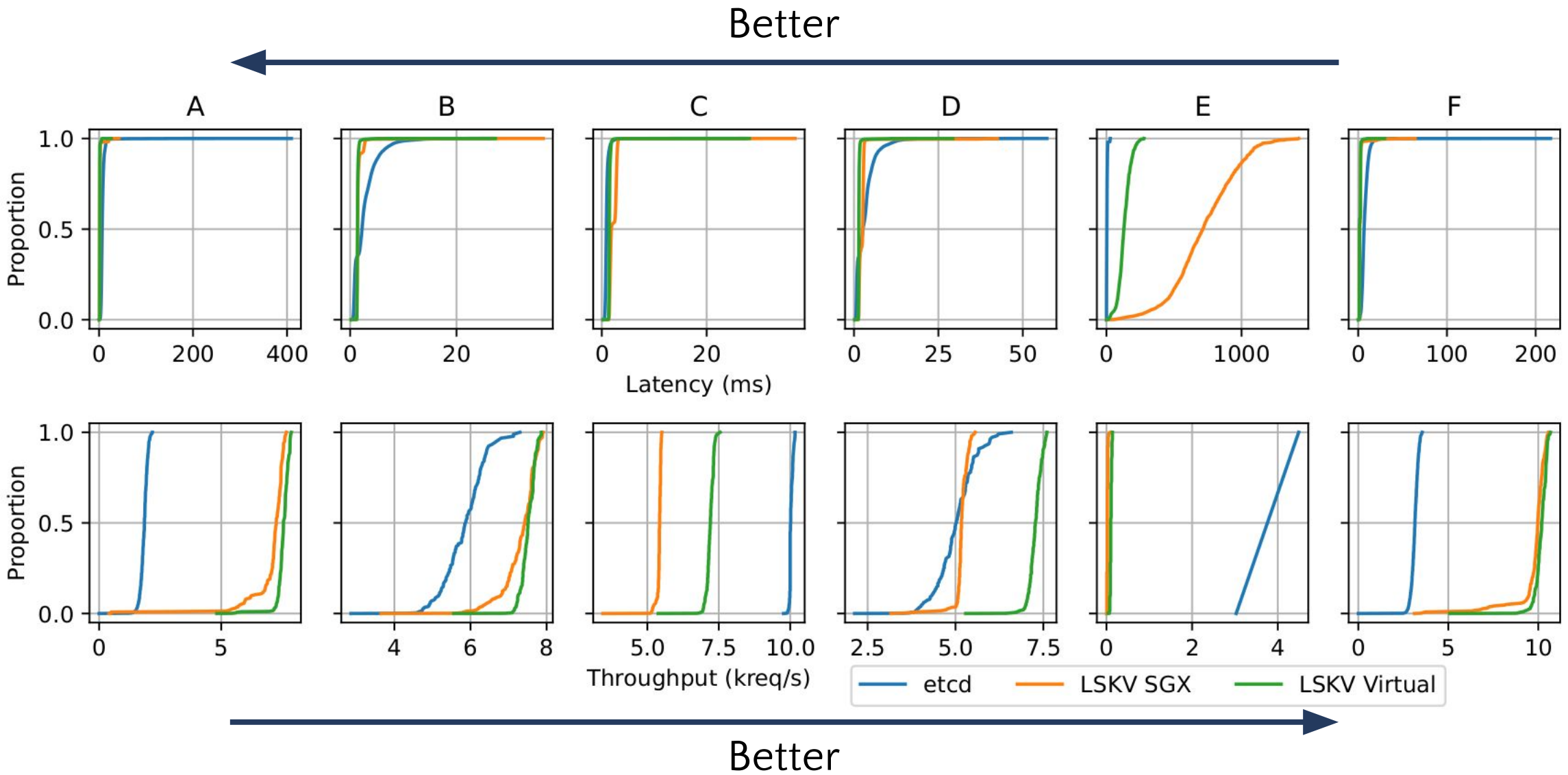
github.com/microsoft/LSKV

Andrew Jeffery – andrew.jeffery@cst.cam.ac.uk

Thank you

github.com/microsoft/LSKV

Oh, and its fast – YCSB workloads (3 nodes)



Ledger

- Operations are either public or private

- Private operations cannot be decrypted by the operator, governors have to combine key shards

- Responsibility of the operator to synchronise the ledger files to other nodes when joining new ones

- Ultimately used for disaster recovery

Optimistic checking

- Since all operations are optimistically acknowledged you may need to follow up if you want to check commit status

- Can also get this from responses to other requests

- Plans to have a watch channel for commit status

Historical staleness

- Specifying a revision acts on a historical copy of the store

- This can lead to observing stale data

- Watches are served from this

Durability

- Operations are persisted to disk lazily

- They also may not be available later, try to keep things in memory

- Stems from not trusting the host

Tackling untrusted proxies – read receipts

- Similar to write receipts but for read operations
- Processed in-application, at any node (not just the leader)
- May need to add a nonce-like field or minimum revision to range requests
  - Maybe use min revision fields in etcd range requests already