

Service Discovery on top of Zookeeper

HPTS 2017

Arthur Zwiegincew, Salesforce

arthurz@salesforce.com

Service Discovery on ZK

Arthur Zwiegincew, Salesforce

arthurz@salesforce.com

salesforce

Objective

Present a simple distributed approach to service discovery on top of Zookeeper.

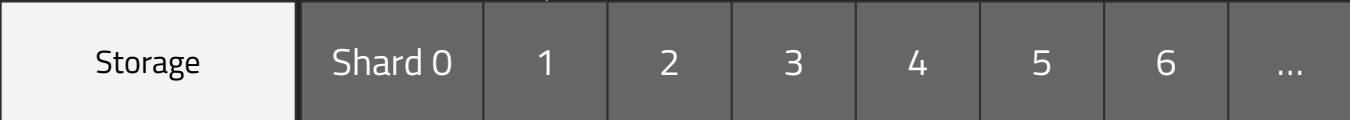
Goal: RPCs via logical addressing – not physical



Stateless tier – connections are load balanced

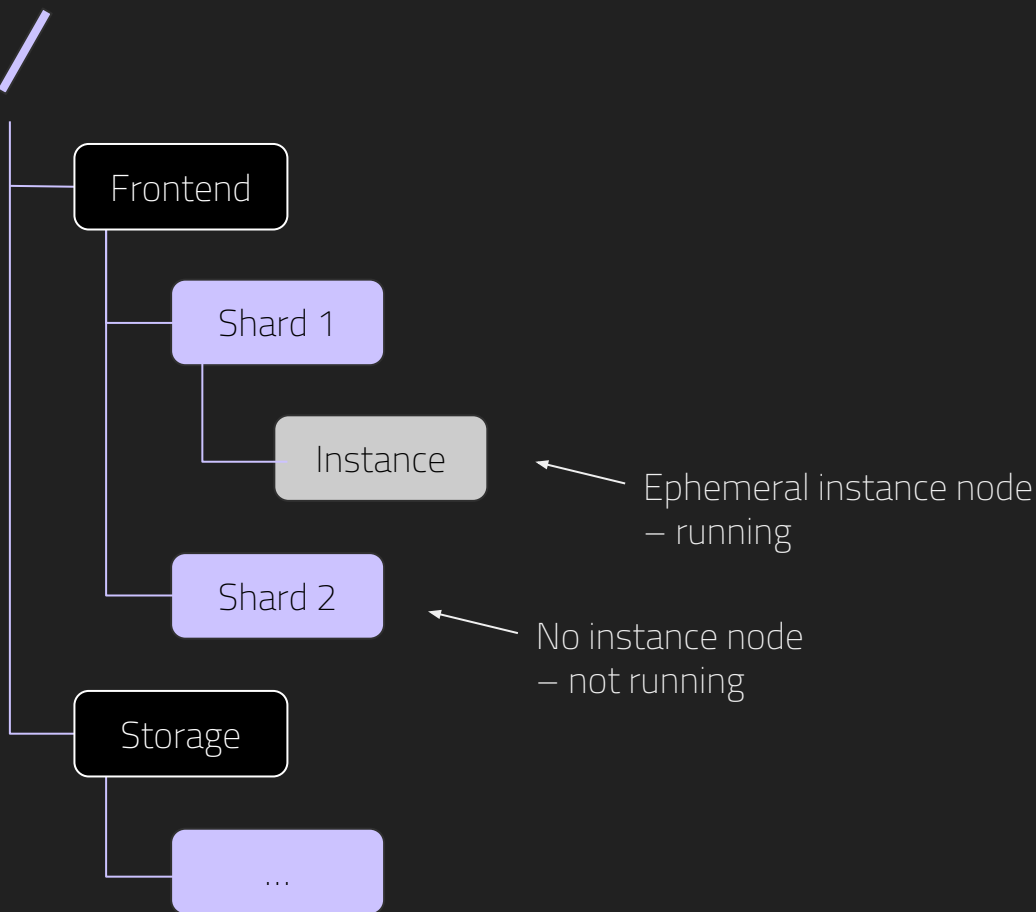


Stateful tier – BL2 explicitly talks to ST1



ZK schema

- Static role nodes.
- Ephemeral nodes representing a running instance.

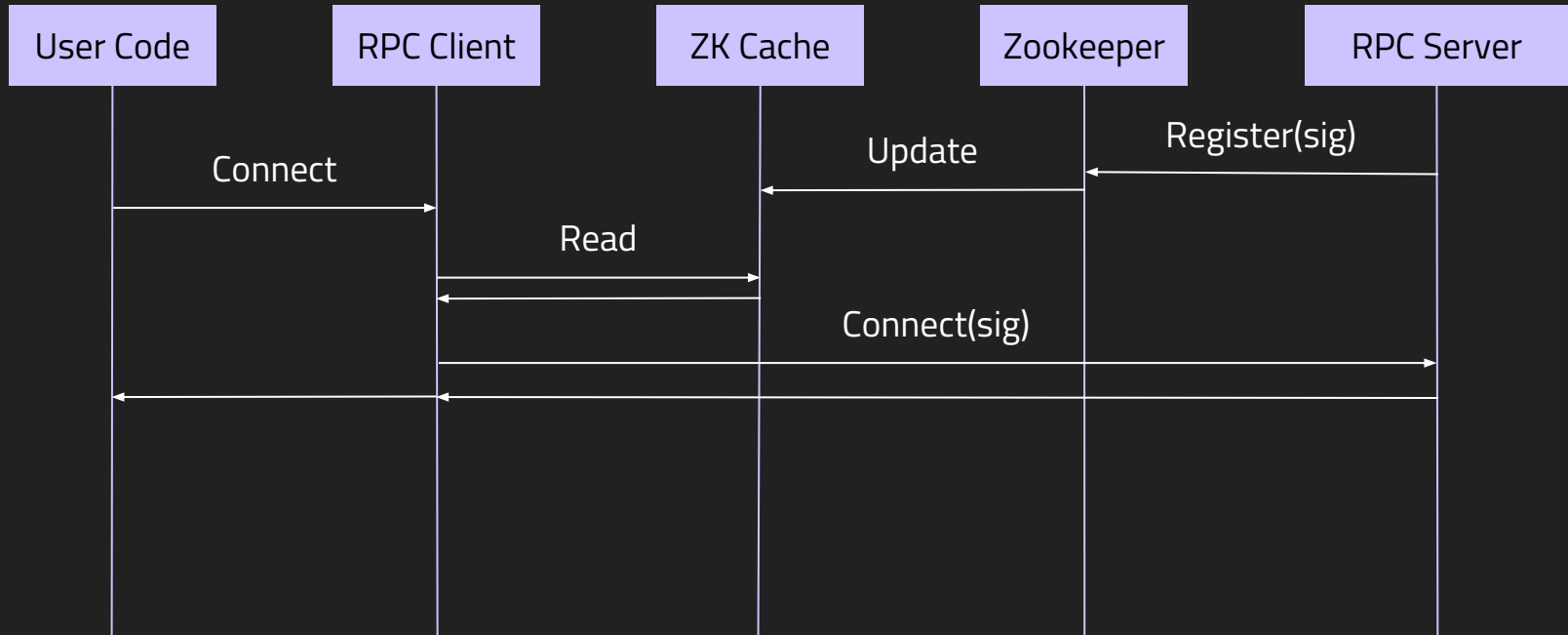


ZK schema – a little deeper

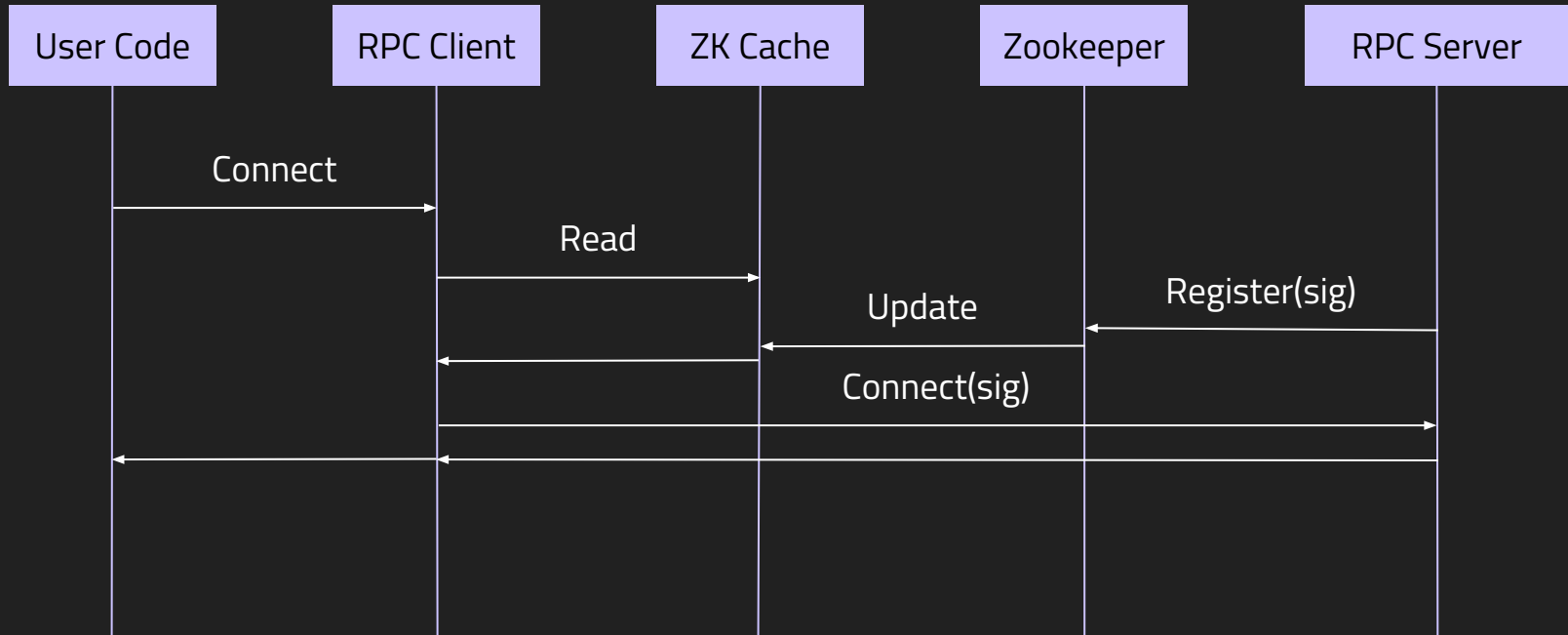
Role/shard	Address	Instance signature
ST42	10.0.1.152:12354	75944B8A-107A-4FAA-8ACE-9173CF679FE9
BL8	10.0.1.43:10240	C239ABF6-EF2C-4010-93B7-AC1F7753EFA4

If the signature retrieved from Zookeeper doesn't match that on the server, that means the ZK data was stale and thus the client needs to refresh.

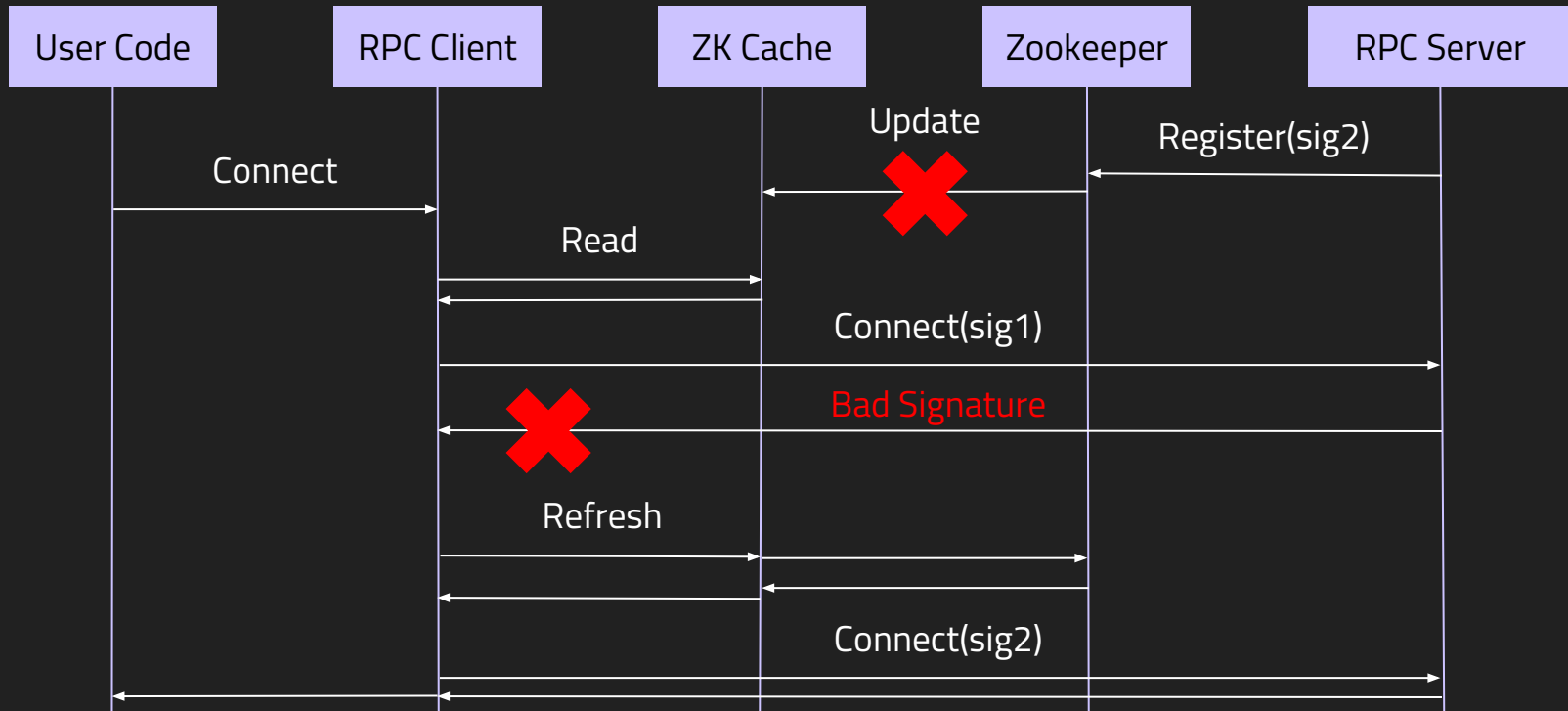
Typical flow



Server not running initially



Stale cache – e.g. network partitions, server restarts



Summary

- No centralized coördination outside of Zookeeper.
- Minimal load on Zookeeper – one tree read per process + refresh traffic.
- Resilient to cache staleness.
- Straightforward to replace Zookeeper with gossip.
- It's also easy to build a cluster scheduler on top of this design – talk to me offline.