



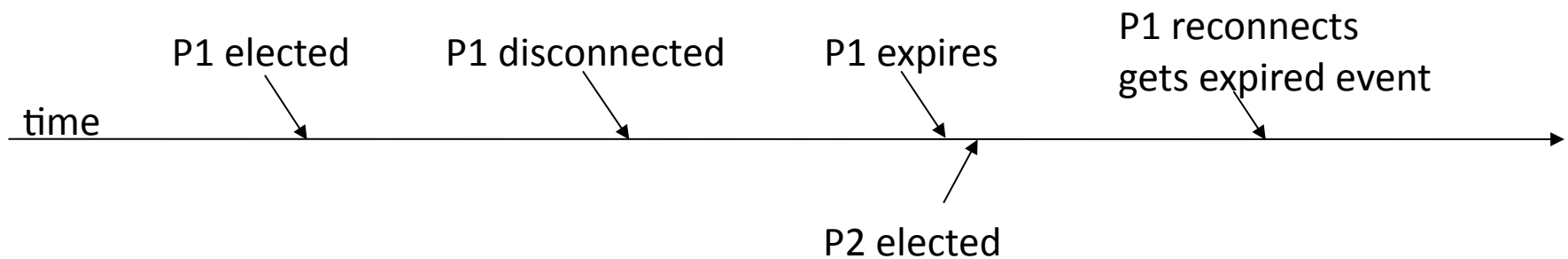
# ZooKeeper Tutorial

Part 4

*Caveat Emptor*

# Revisit FLP and CAP

- What should a master do when disconnected?
  - What is the consequence of acting as a master while disconnected?



# Revisit FLP and CAP

- What happens if master election gets a “`ConnectionLossException`” after the create?
  - How do you fix it?
  - How do you test it?



# Guidelines to `ConnectionLoss`

- A process will not see state changes while disconnected
- Masters should act very conservatively, they should not assume that they still have mastership
- Don't treat as if it's the end of the world. The client library will try to recover the session



# Other issues

- Watch out for `SEQUENTIAL` | `EPHEMERAL`!
- Problems resetting the ZooKeeper state
  - What happens when you clear server state while clients are running?
  - What happens when you clear some servers but not others?



# Writing a test

- Use JUnit
- Use QuorumBase
  - In setup call `QuorumBase.setup()`
  - In tearDown call `QuorumBase.tearDown()`
- Write a simple test
  - Use `QuorumBase.hostPort` to initialize the ZooKeeper object in the tests
  - Startup a master and a backup.
  - Kill the master and make sure backup takes over



# Guidelines for SessionExpiration

- It is the end of the world!
- Should be rare.
- The session handle is dead, so you need a new one.
- It is dangerous to try to transparently recover by creating a new session. Usually there is some cleanup and setup that needs to be done





Code on your own or follow  
together



# Summary

- When used properly ZooKeeper can make it easy to build distributed applications.
- ZooKeeper is a tool to help you deal with the chaos of distributed systems. It isn't magic.
  - Don't try to shortcut the API
  - Think about the consequences of `ConnectionLoss` and `SessionExpiration`
  - Make sure you test
- Checkout the developer resources

<http://zookeeper.apache.org>

