使用Zookeeper改造舊系統

Matt @ JCConf 2016

About me

- 何適宇 <Matt>
  - methodho@gmail.com
  - https://github.com/methodho
- 松凌科技 <http://www.softleader.com.tw/>

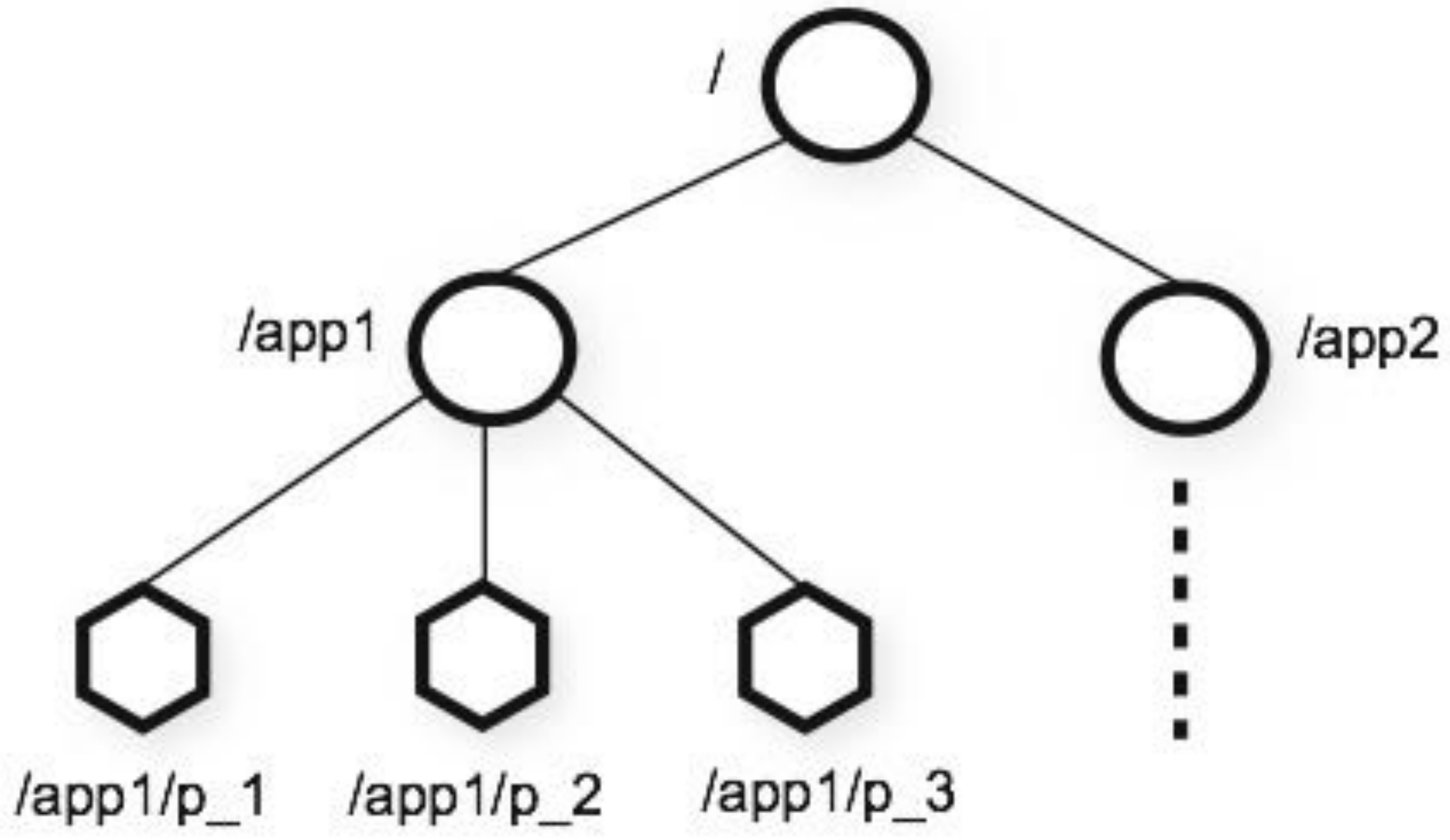
大綱

- ZooKeeper 介紹
- 使用 ZooKeeper 前的方法
- Java programming to ZooKeeper
    - o.a.z.ZooKeeper (3.4.9)
    - o.a.c.f.r.l.LeaderLatch (2.11.0)

Because coordinating distributed systems is a Zoo

Apache ZooKeeper wiki

WHAT IS ZOOKEEPER



https://zookeeper.apache.org/doc/current/zookeeperOver.html

ZOOKEEPER CLI

- 啟動server
  - /bin/zkServer.sh start
- 連結到 zookeeper server
  - /bin/zkCli.sh [-server localhost:2181]
- 關閉server
  - /bin/zkServer.sh stop

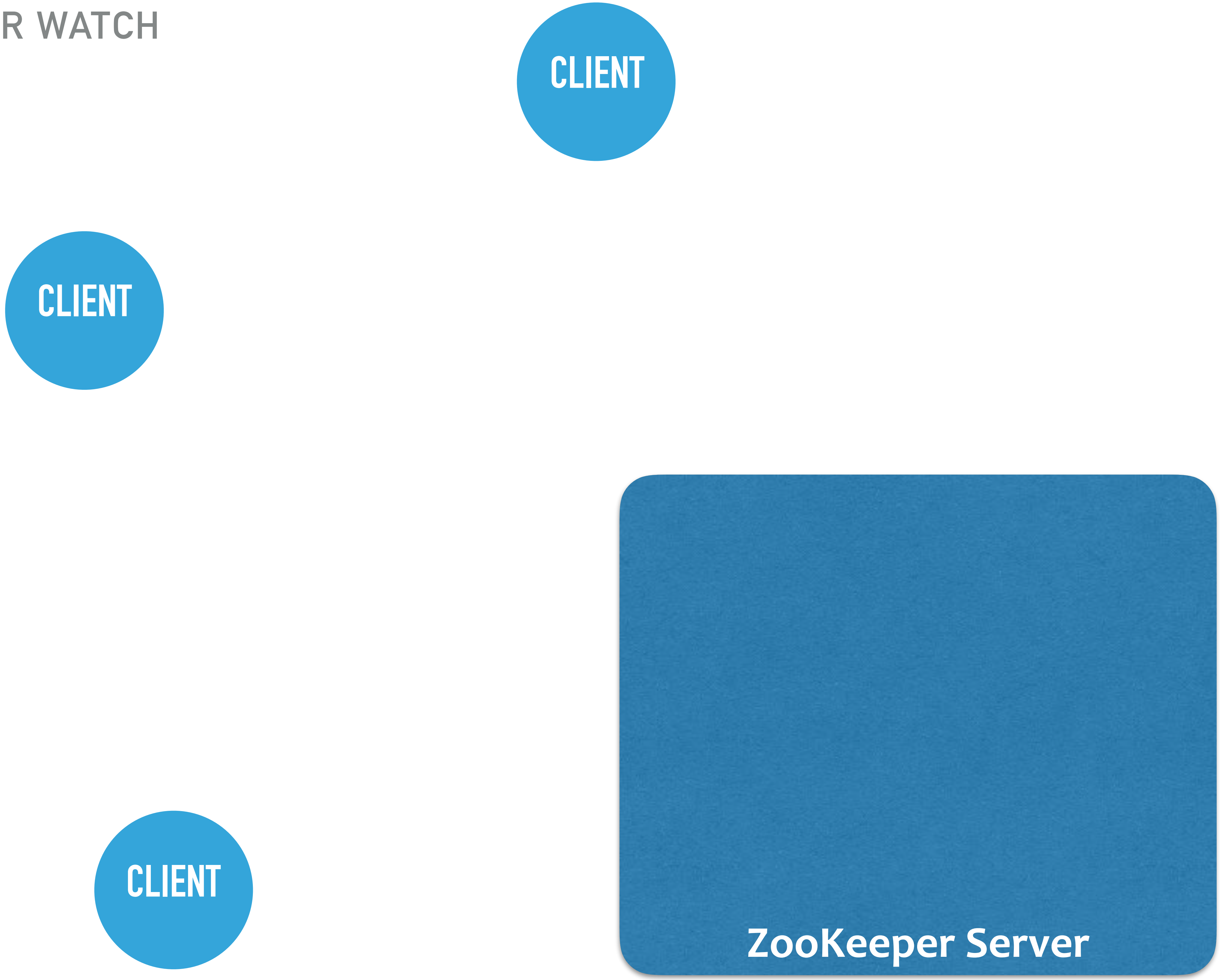
ZOOKEEPER CLI

- 新增 znode
  - create [-s] [-e] path data
- 刪除 znode
  - delete path
  - rmr path

ZOOKEEPER CLI

- 列出 znode 下的 children
  - ls path [watch]
  - ls2 path [watch]
- 讀寫 znode data (byte[])
  - set path data [watch]
  - get path [watch]

ZOOKEEPER WATCH



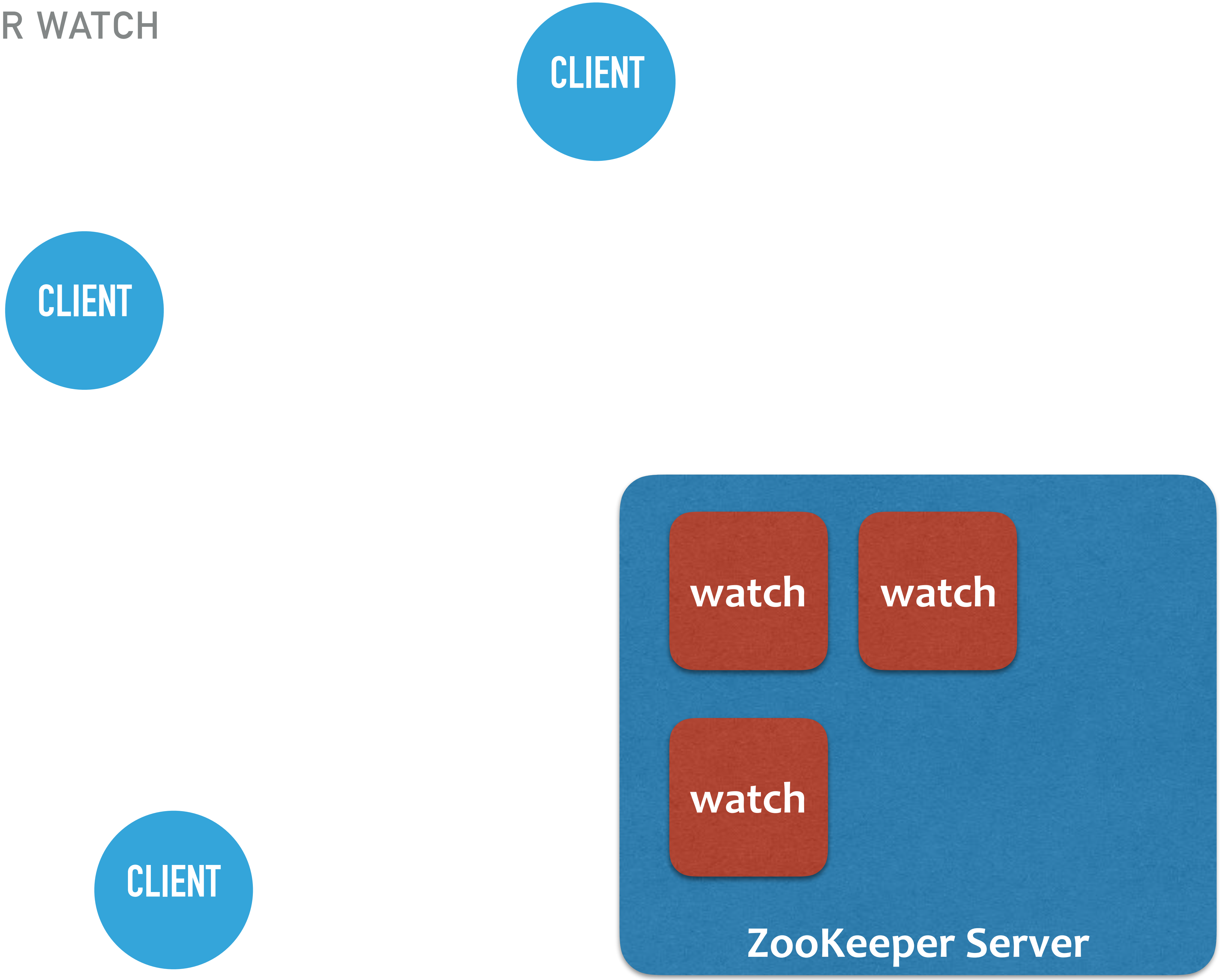
CLIENT

CLIENT

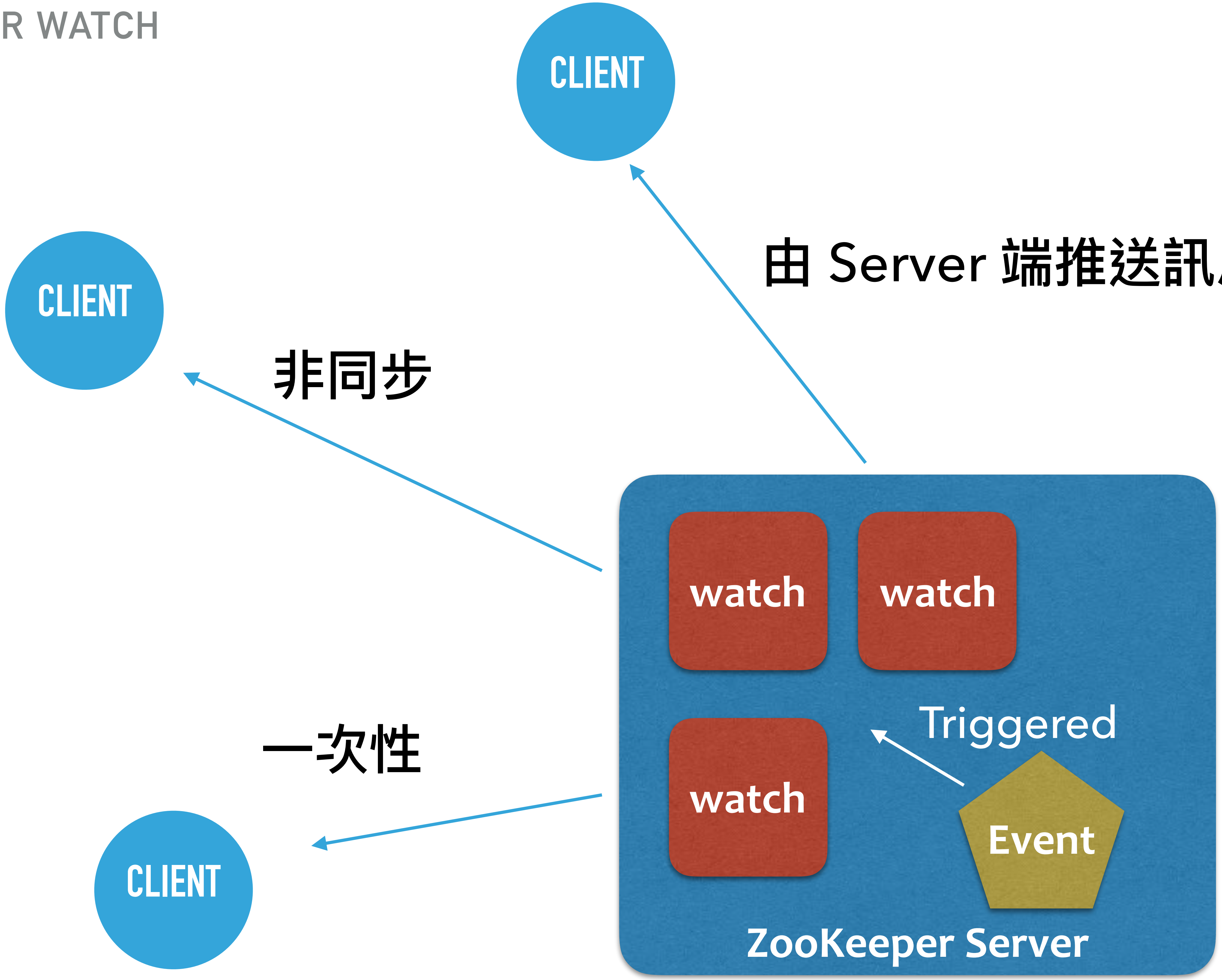
CLIENT

ZooKeeper Server

ZOOKEEPER WATCH



ZOOKEEPER WATCH

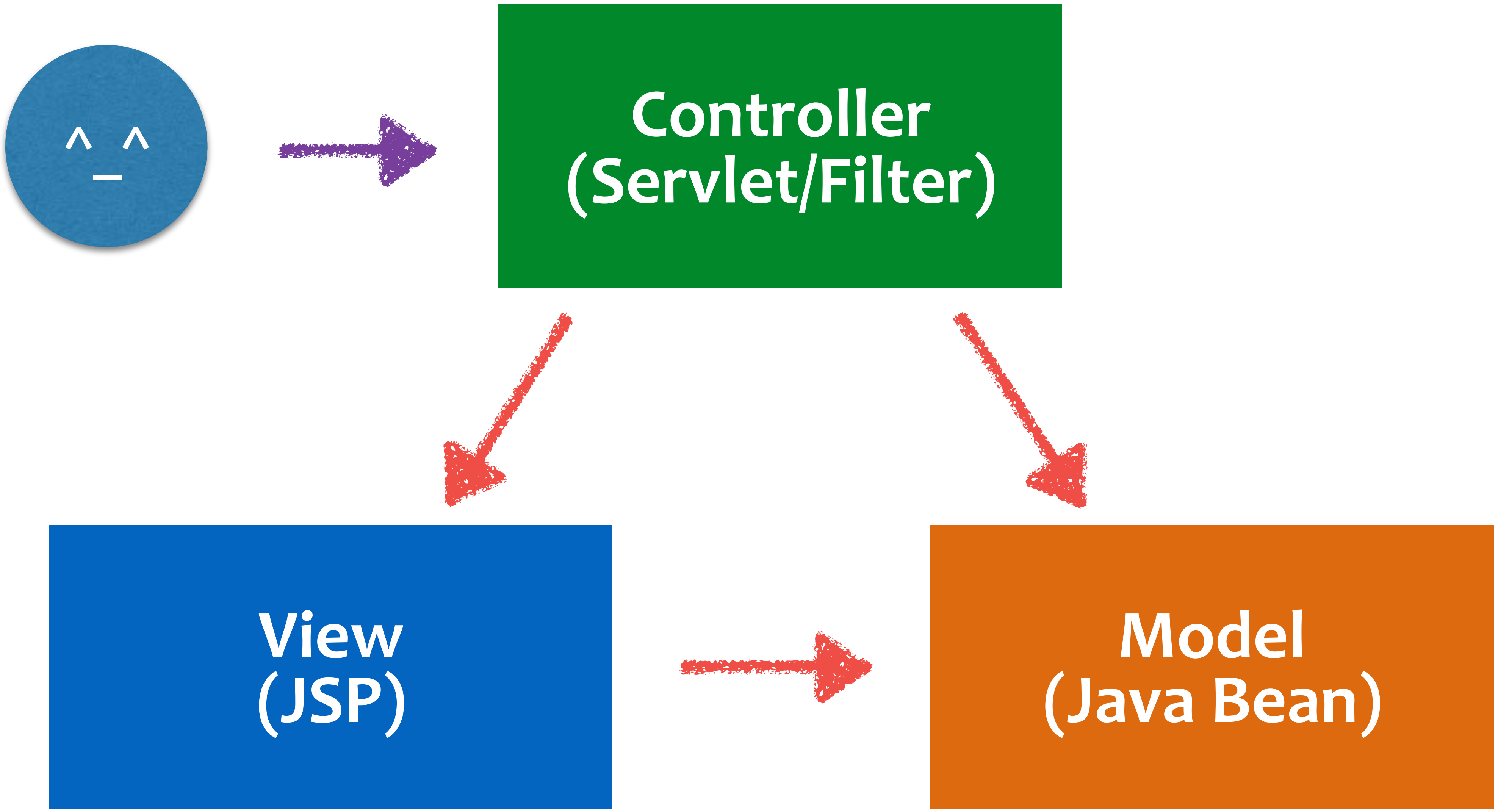


由 Server 端推送訊息

非同步

一次性

A MCV EXAMPLE (MODEL 2)

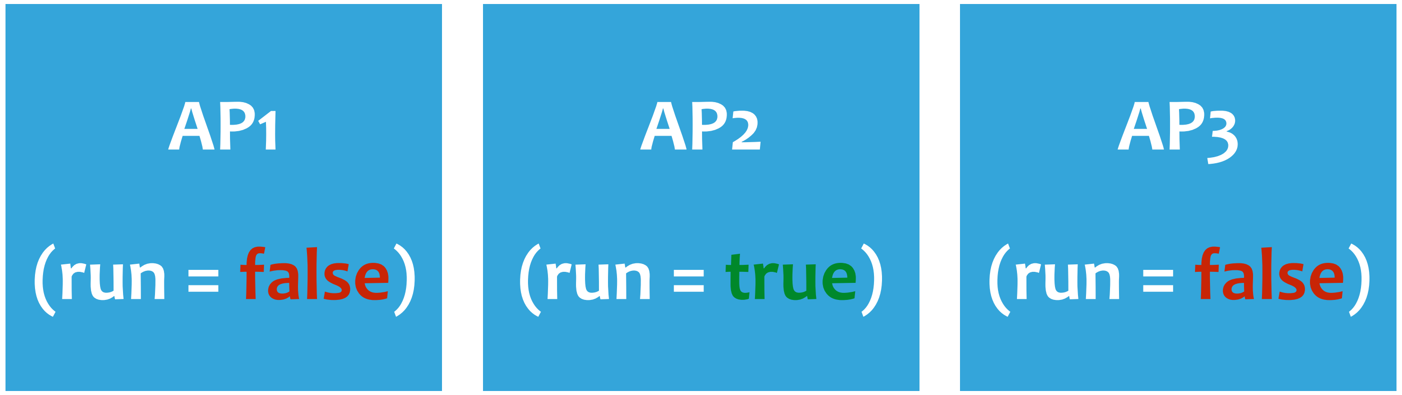


?

故事是這樣子的...

BEFORE ZOOKEEPER

- 由ap自己控制
  - hardcode
  - properties
  - passes argument …
- cons?



AP1

(run = false)

AP2

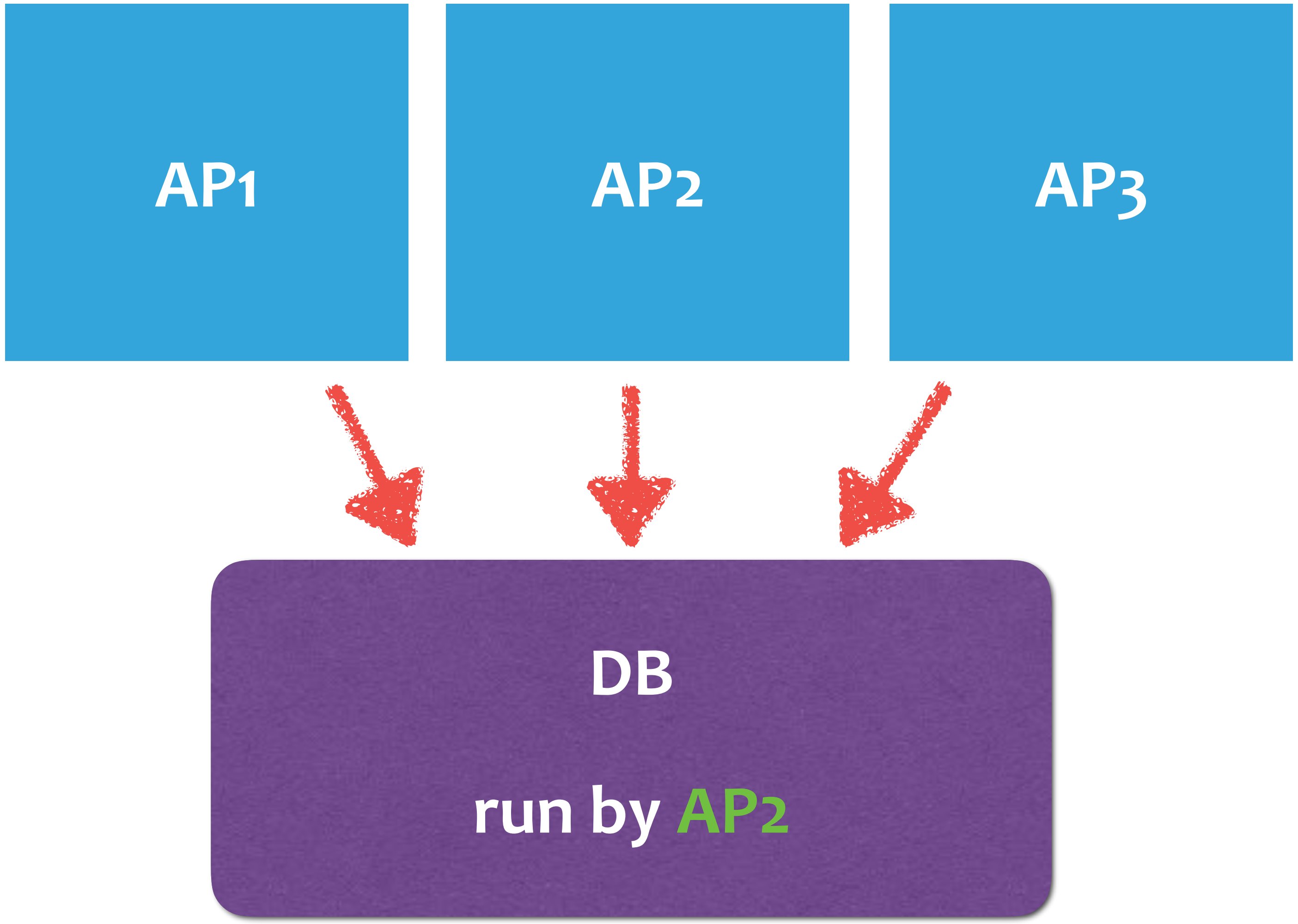
(run = true)

AP3

(run = false)

BEFORE ZOOKEEPER

- 將控制點抽離ap
  - db table control
  - db lock
- cons?



我們需要的是...

有一個方法可以實作自己的邏輯:

決定誰要執行

BEFORE ZOOKEEPER

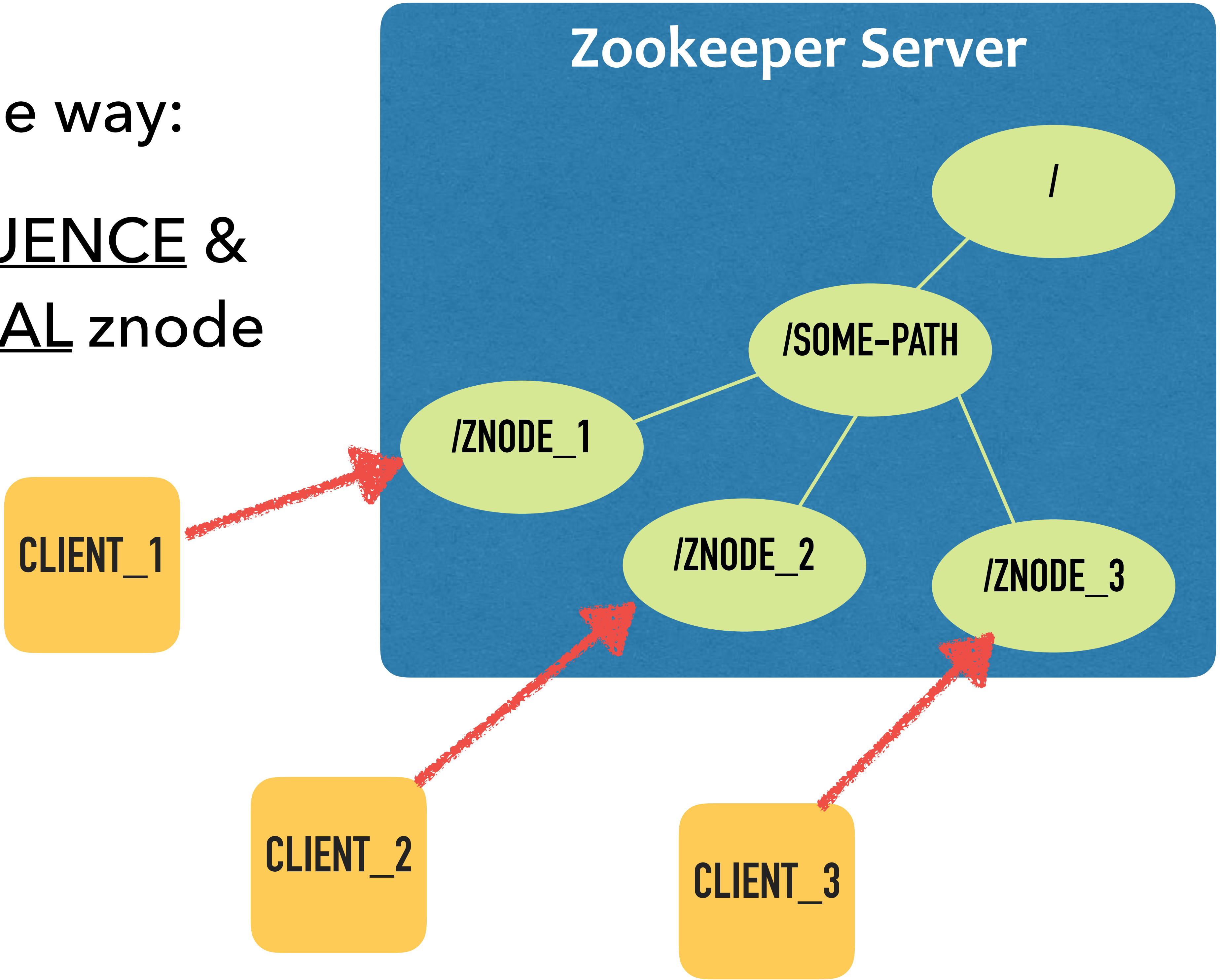
!

ZooKeeper Recipes

https://zookeeper.apache.org/doc/trunk/recipes.html

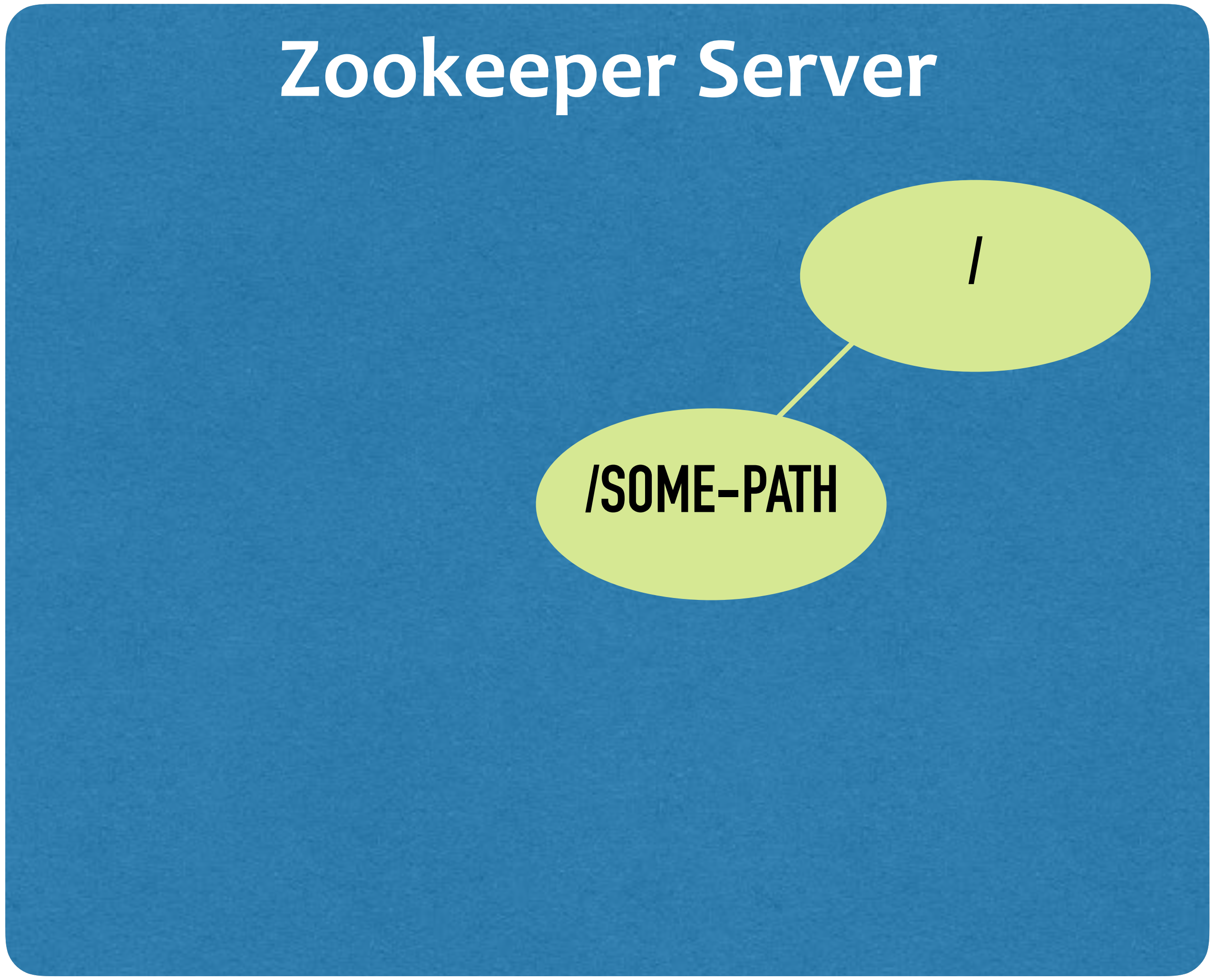
A simple way:

Use SEQUENCE & EPHEMERAL znode



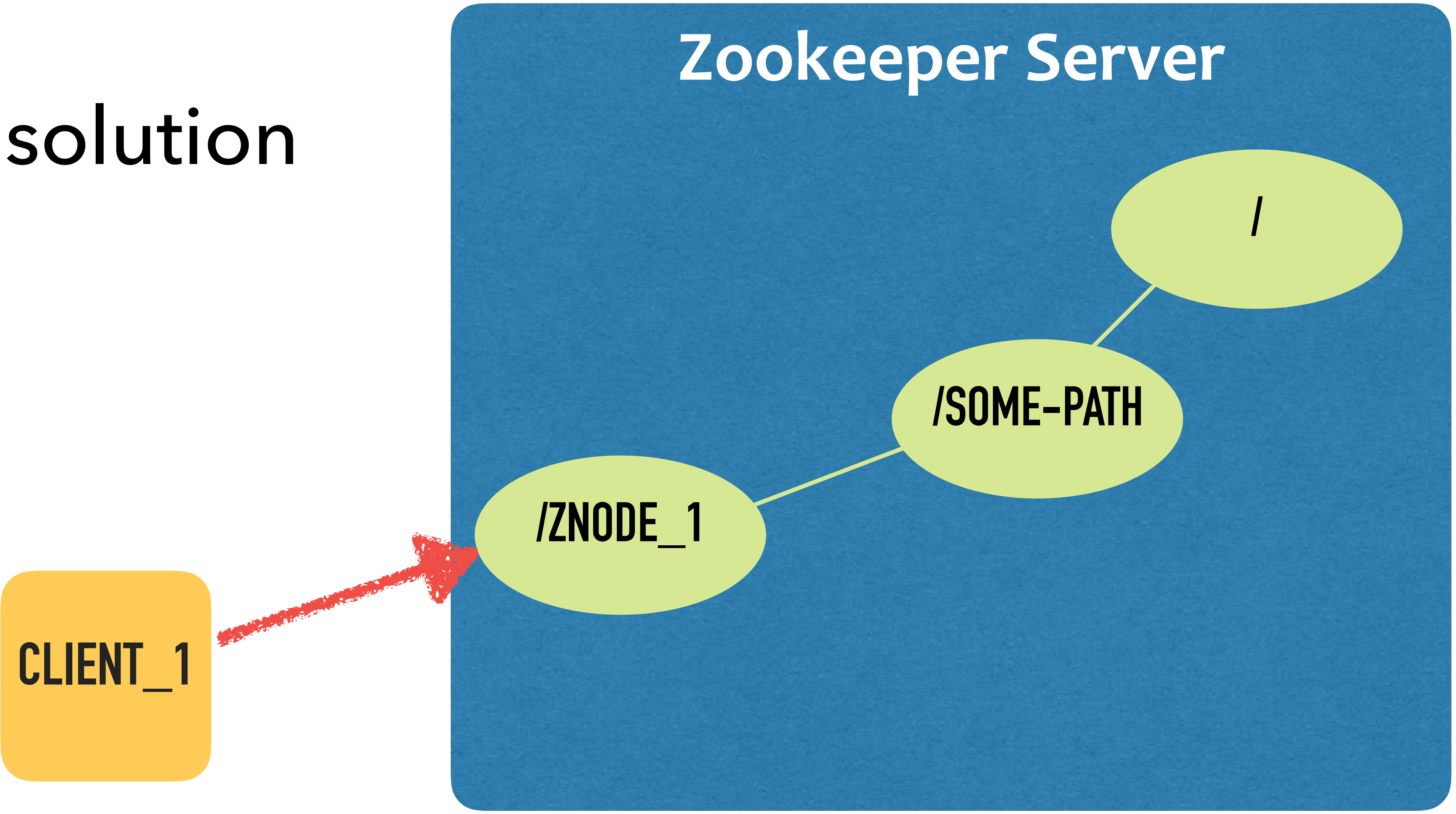
https://zookeeper.apache.org/doc/trunk/recipes.html#sc_leaderElection

Advance solution



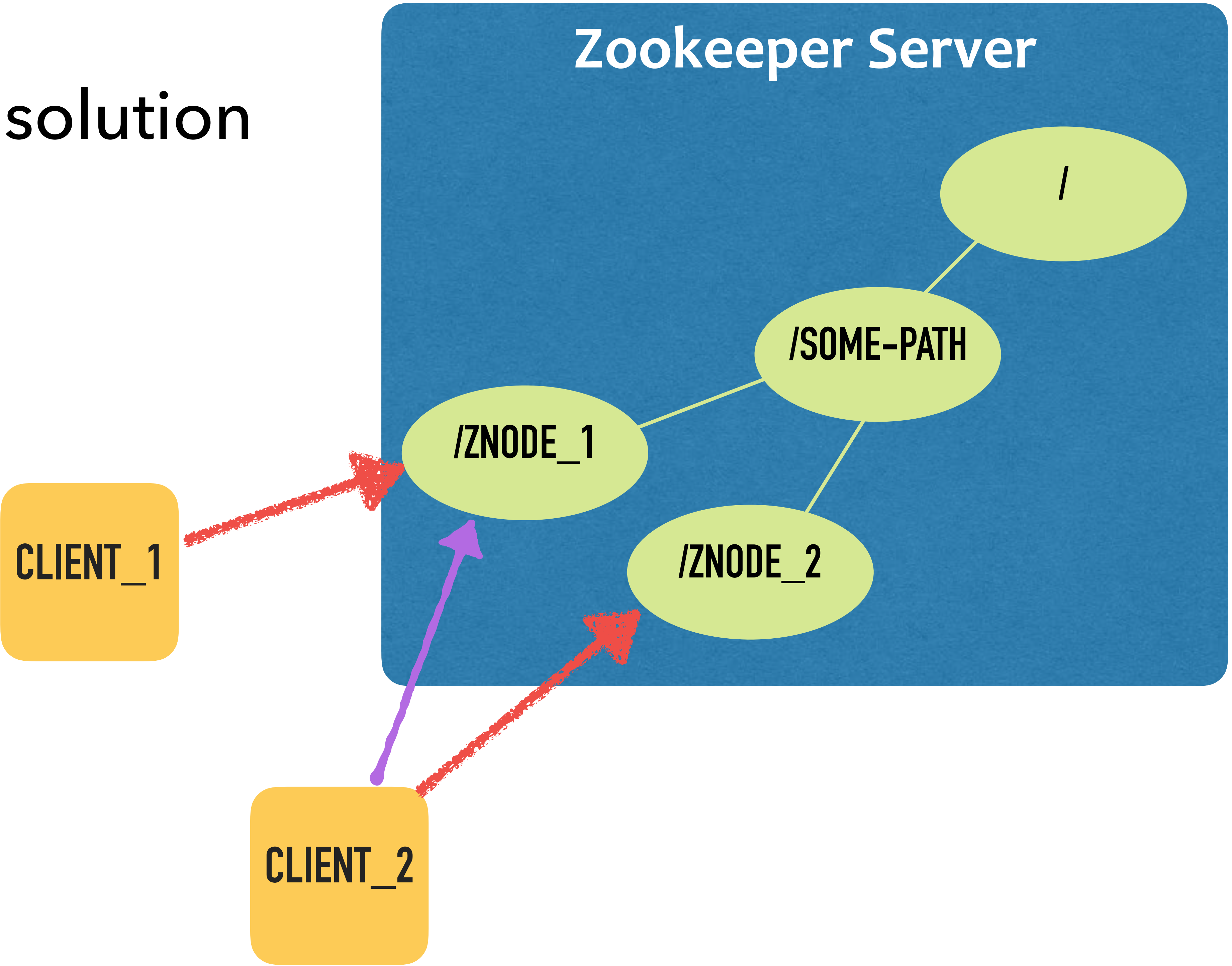
https://zookeeper.apache.org/doc/trunk/recipes.html#sc_leaderElection

Advance solution



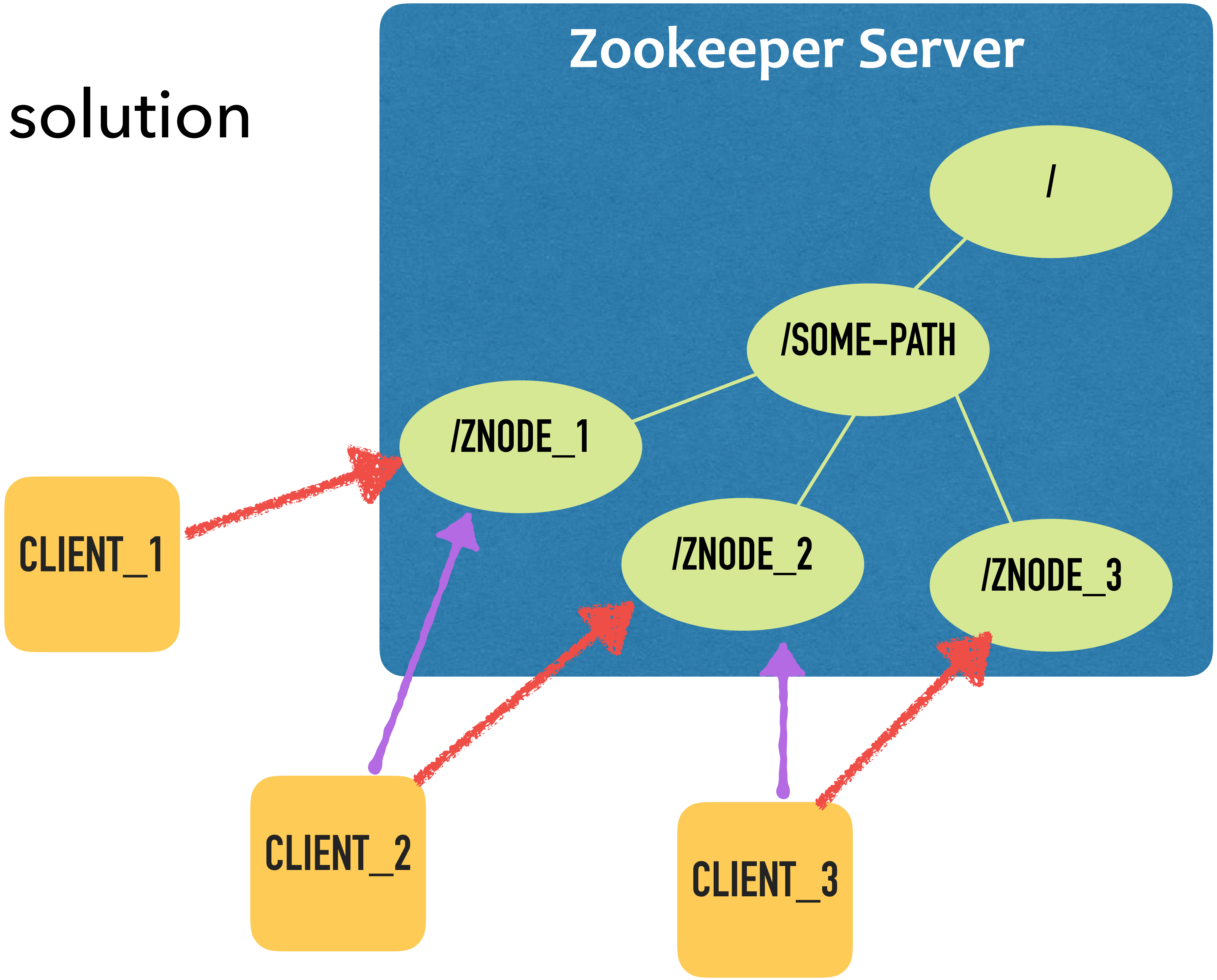
https://zookeeper.apache.org/doc/trunk/recipes.html#sc_leaderElection

Advance solution



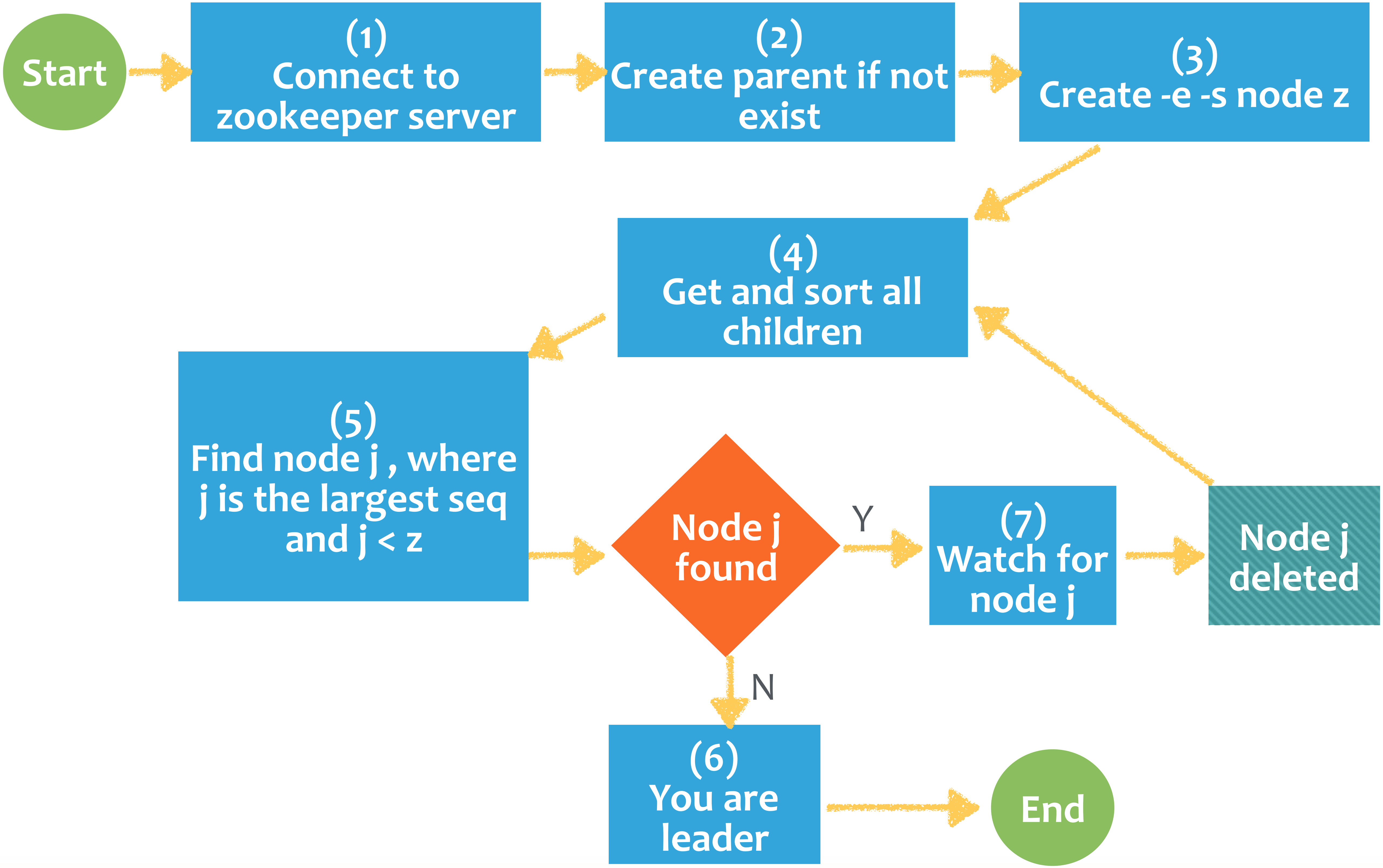
https://zookeeper.apache.org/doc/trunk/recipes.html#sc_leaderElection

Advance solution

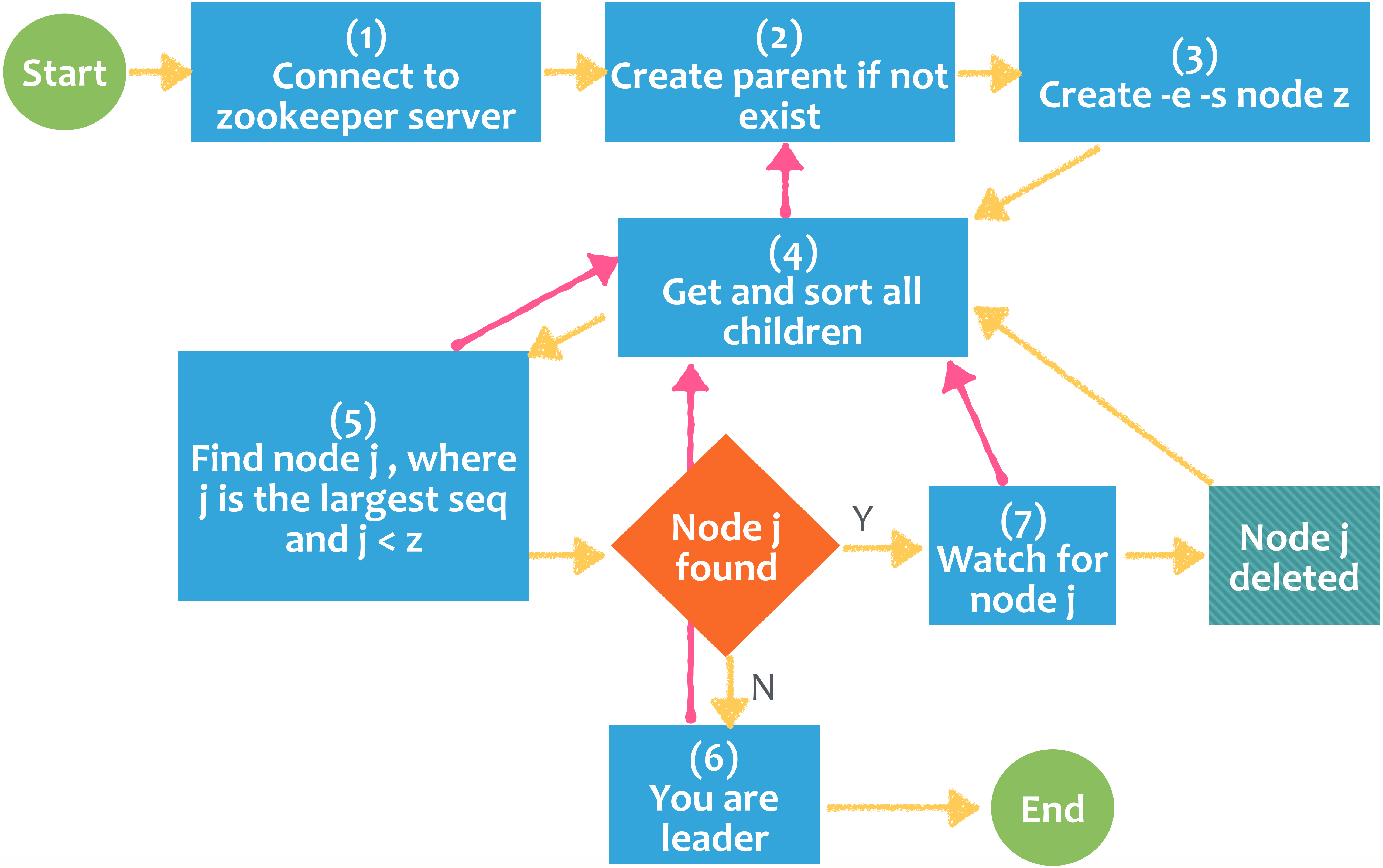


https://zookeeper.apache.org/doc/trunk/recipes.html#sc_leaderElection

IMPLEMENTING LEADER ELECTION


Start
(1)
Connect to zookeeper server
(2)
Create parent if not exist
(3)
Create -e -s node z
(4)
Get and sort all children
(5)
Find node j , where j is the largest seq and j < z
Node j found
Y
(7)
Watch for node j
Node j deleted
N
(6)
You are leader
End

IMPLEMENTING LEADER ELECTION


Start
(1) Connect to zookeeper server
(2) Create parent if not exist
(3) Create -e -s node z
(4) Get and sort all children
(5) Find node j , where j is the largest seq and j < z
Node j found
Y
N
(7) Watch for node j
Node j deleted
(6) You are leader
End

PROGRAMMING TO ZOOKEEPER

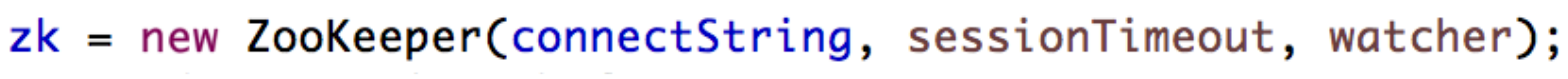
Curator

"Guava is to Java what Curator is to ZooKeeper"
Patrick Hunt, ZooKeeper committer

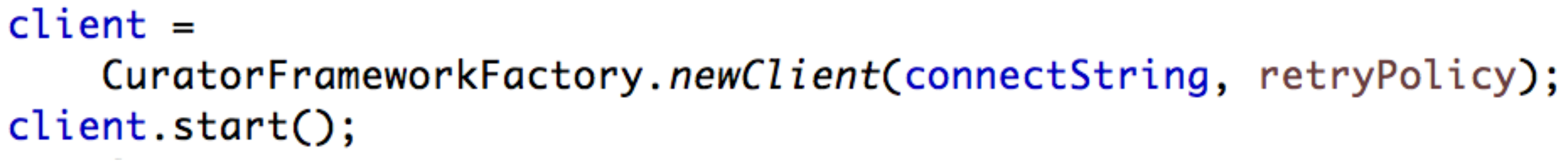
http://curator.apache.org/

PROGRAMMING TO ZOOKEEPER

(1)
Connect to
zookeeper server



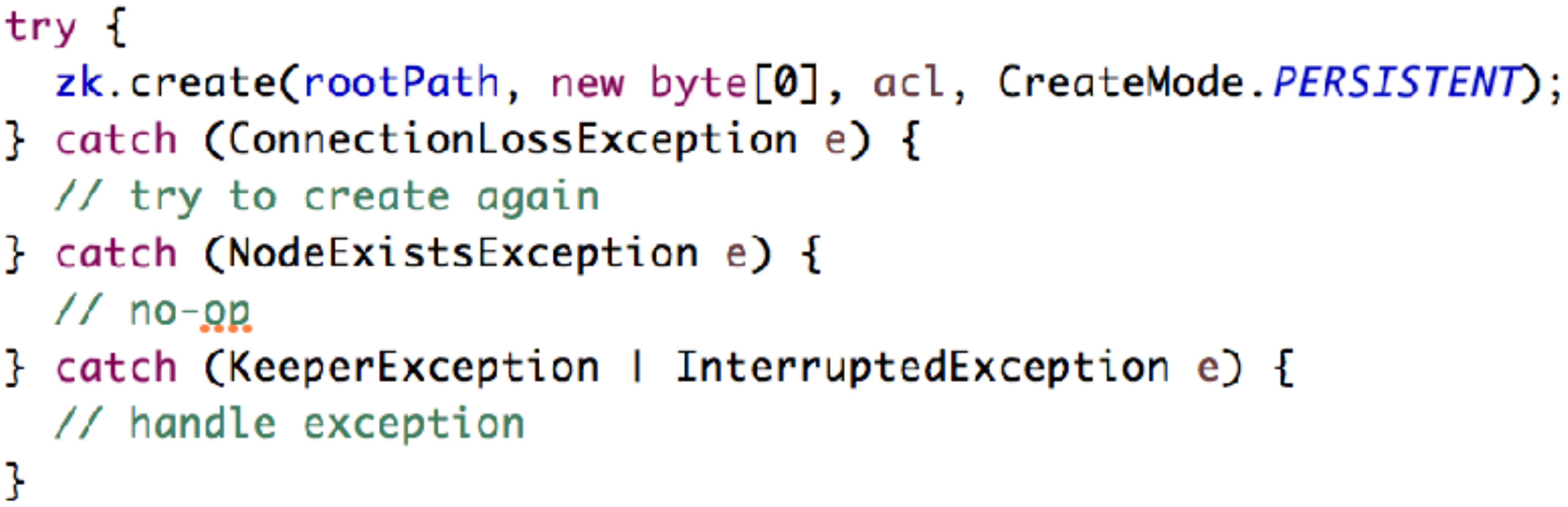
```
zk = new ZooKeeper(connectString, sessionTimeout, watcher);
```



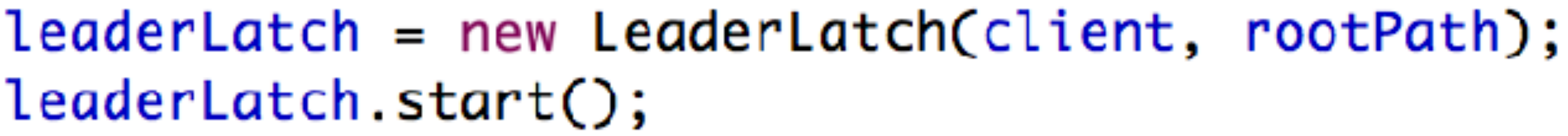
```
client =
    CuratorFrameworkFactory.newClient(connectString, retryPolicy);
client.start();
```

PROGRAMMING TO ZOOKEEPER

(2)
Create parent if not exist



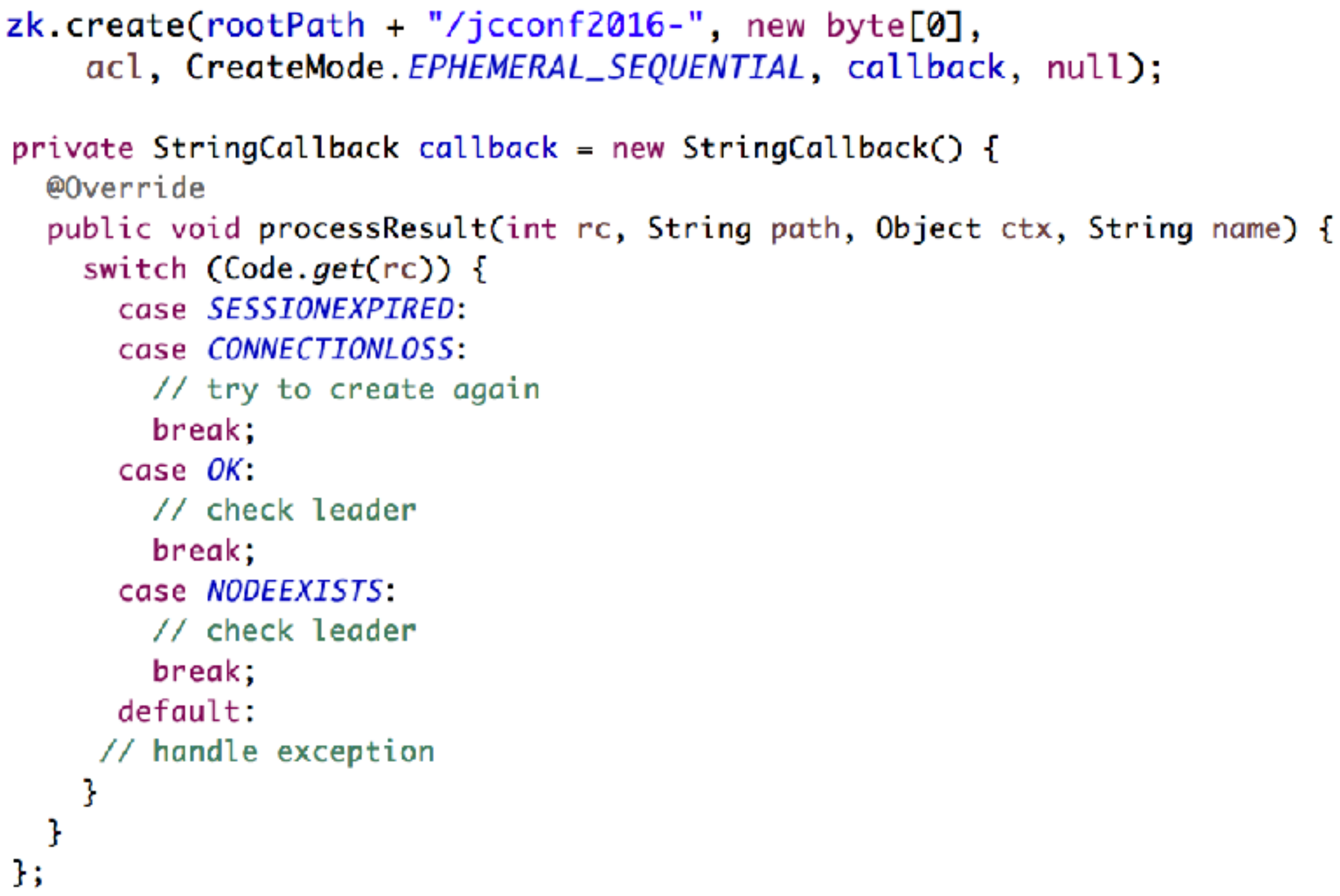
```
try {
  zk.create(rootPath, new byte[0], acl, CreateMode.PERSISTENT);
} catch (ConnectionLossException e) {
  // try to create again
} catch (NodeExistsException e) {
  // no-op
} catch (KeeperException | InterruptedException e) {
  // handle exception
}
```



```
leaderLatch = new LeaderLatch(client, rootPath);
leaderLatch.start();
```

PROGRAMMING TO ZOOKEEPER

(3)
Create -e -s node z

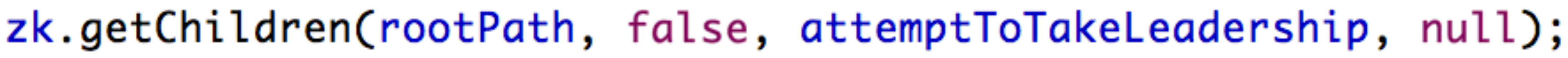


```
zk.create(rootPath + "/jcconf2016-", new byte[0],
    acl, CreateMode.EPHEMERAL_SEQUENTIAL, callback, null);

private StringCallback callback = new StringCallback() {
  @Override
  public void processResult(int rc, String path, Object ctx, String name) {
    switch (Code.get(rc)) {
      case SESSIONEXPIRED:
      case CONNECTIONLOSS:
        // try to create again
        break;
      case OK:
        // check leader
        break;
      case NODEEXISTS:
        // check leader
        break;
      default:
     // handle exception
    }
  }
};
```

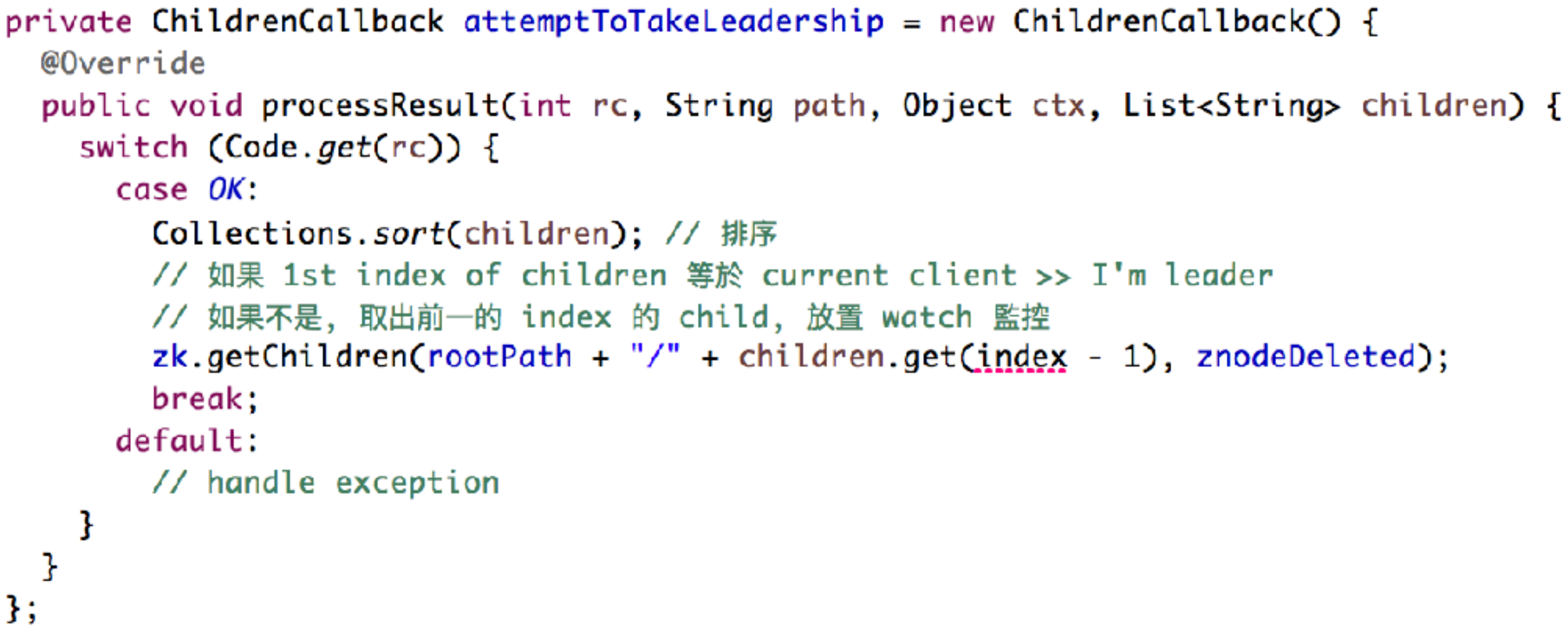
PROGRAMMING TO ZOOKEEPER

(4)
Get and sort all children

(5)
Find node j , where j is the largest seq and j < z



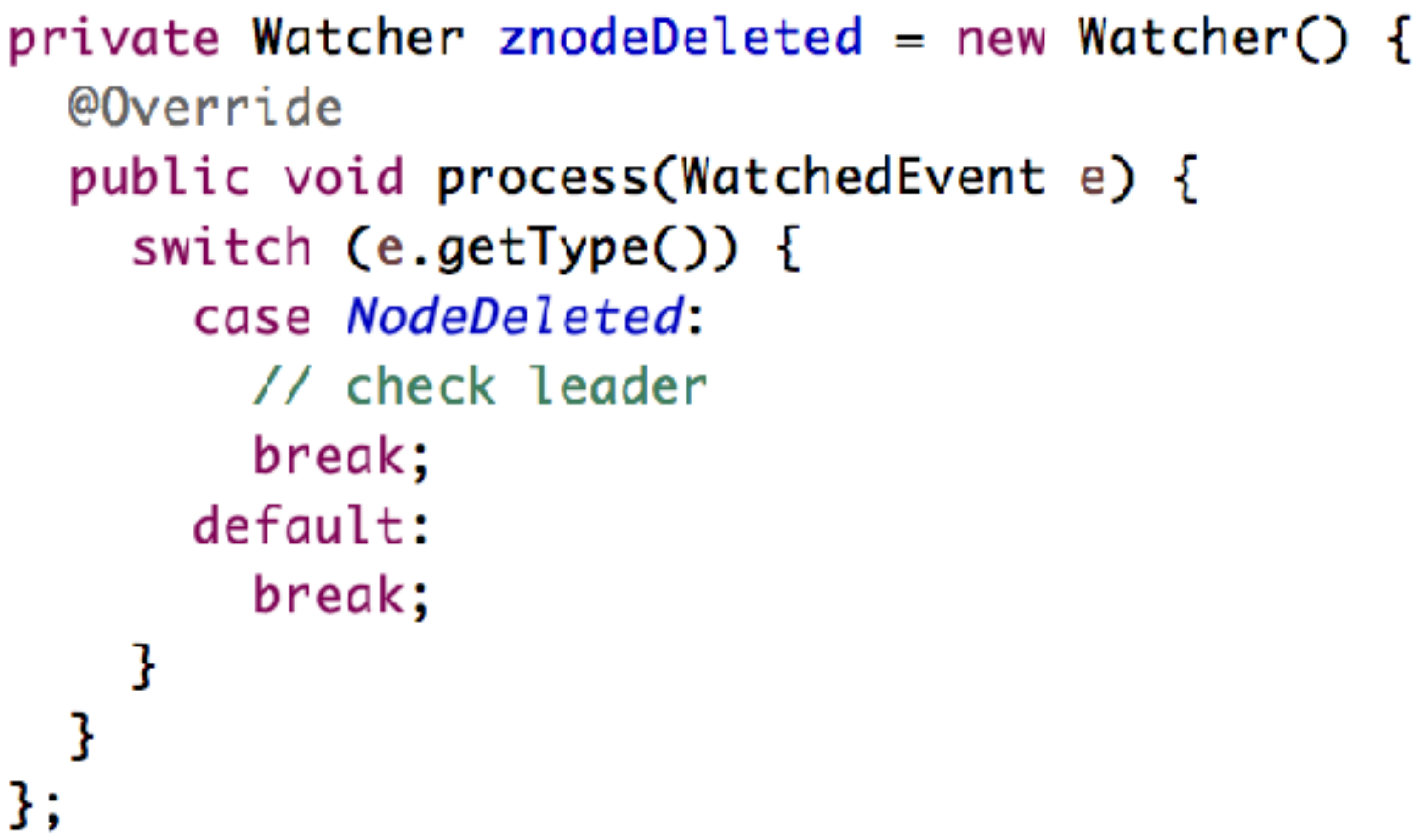
```
zk.getChildren(rootPath, false, attemptToTakeLeadership, null);
```



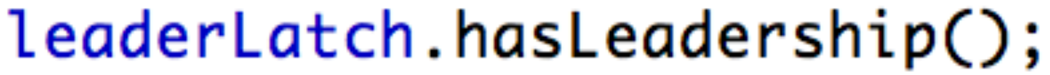
```
private ChildrenCallback attemptToTakeLeadership = new ChildrenCallback() {
  @Override
  public void processResult(int rc, String path, Object ctx, List<String> children) {
    switch (Code.get(rc)) {
      case OK:
        Collections.sort(children); // 排序
        // 如果 1st index of children 等於 current client >> I'm leader
        // 如果不是, 取出前一的 index 的 child, 放置 watch 監控
        zk.getChildren(rootPath + "/" + children.get(index - 1), znodeDeleted);
        break;
      default:
        // handle exception
    }
  }
};
```

PROGRAMMING TO ZOOKEEPER

(7)
Watch for
node j

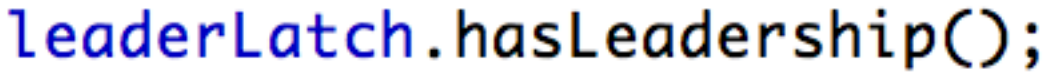


```java
private Watcher znodeDeleted = new Watcher() {
  @Override
  public void process(WatchedEvent e) {
    switch (e.getType()) {
      case NodeDeleted:
        // check leader
        break;
      default:
        break;
    }
  }
};
```



```
leaderLatch.hasLeadership();
```

PROGRAMMING TO ZOOKEEPER



```
leaderLatch.hasLeadership();
```

剛寫那麼多都是寫 ____ der...

?

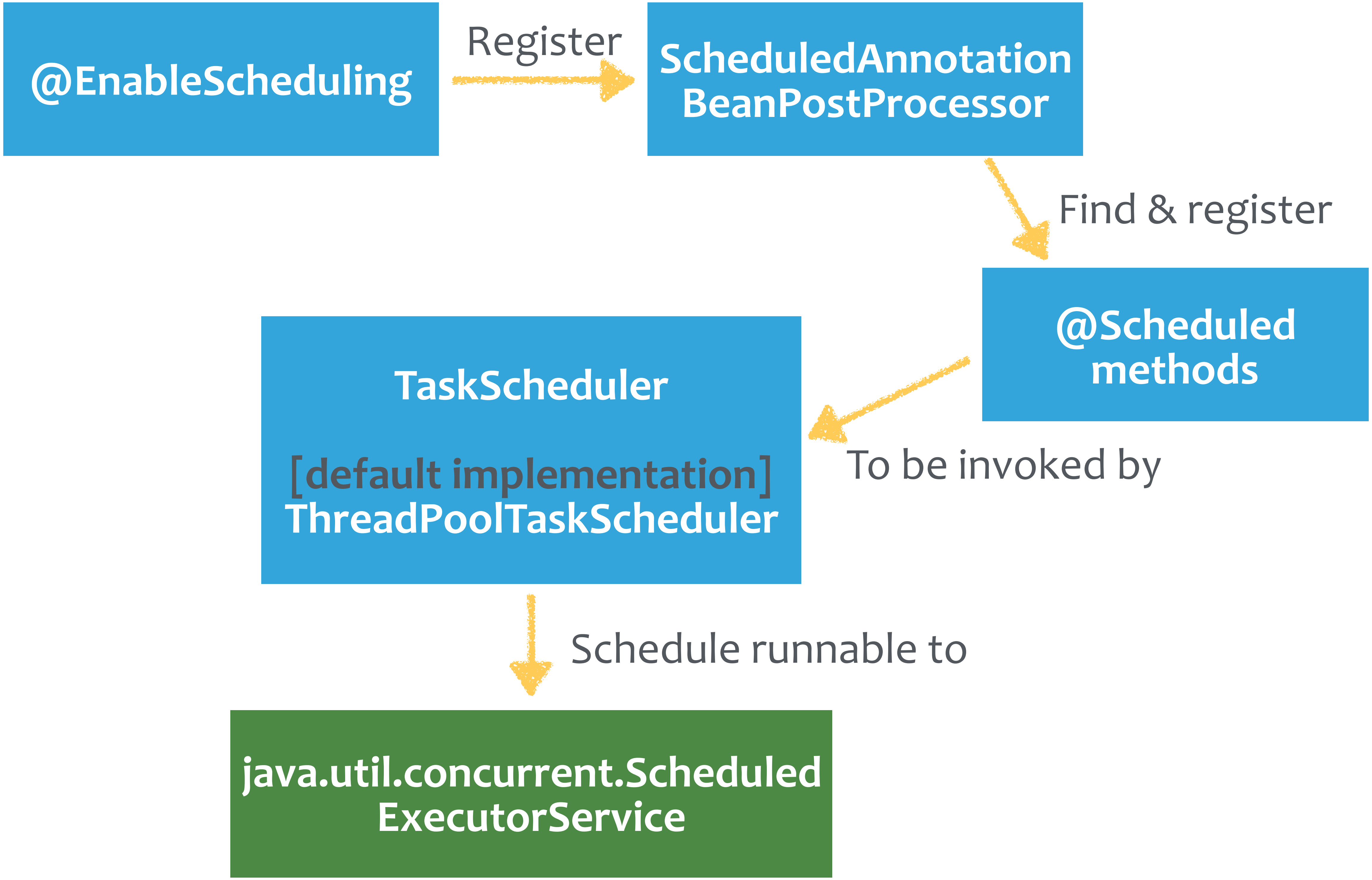
怎麼串接到排程中..?

ADAPTING INTO SCHEDULE FRAMEWORK

@EnableScheduling

@Scheduled methods

ADAPTING INTO SCHEDULE FRAMEWORK



@EnableScheduling

Register

ScheduledAnnotation BeanPostProcessor

Find & register

@Scheduled methods

To be invoked by

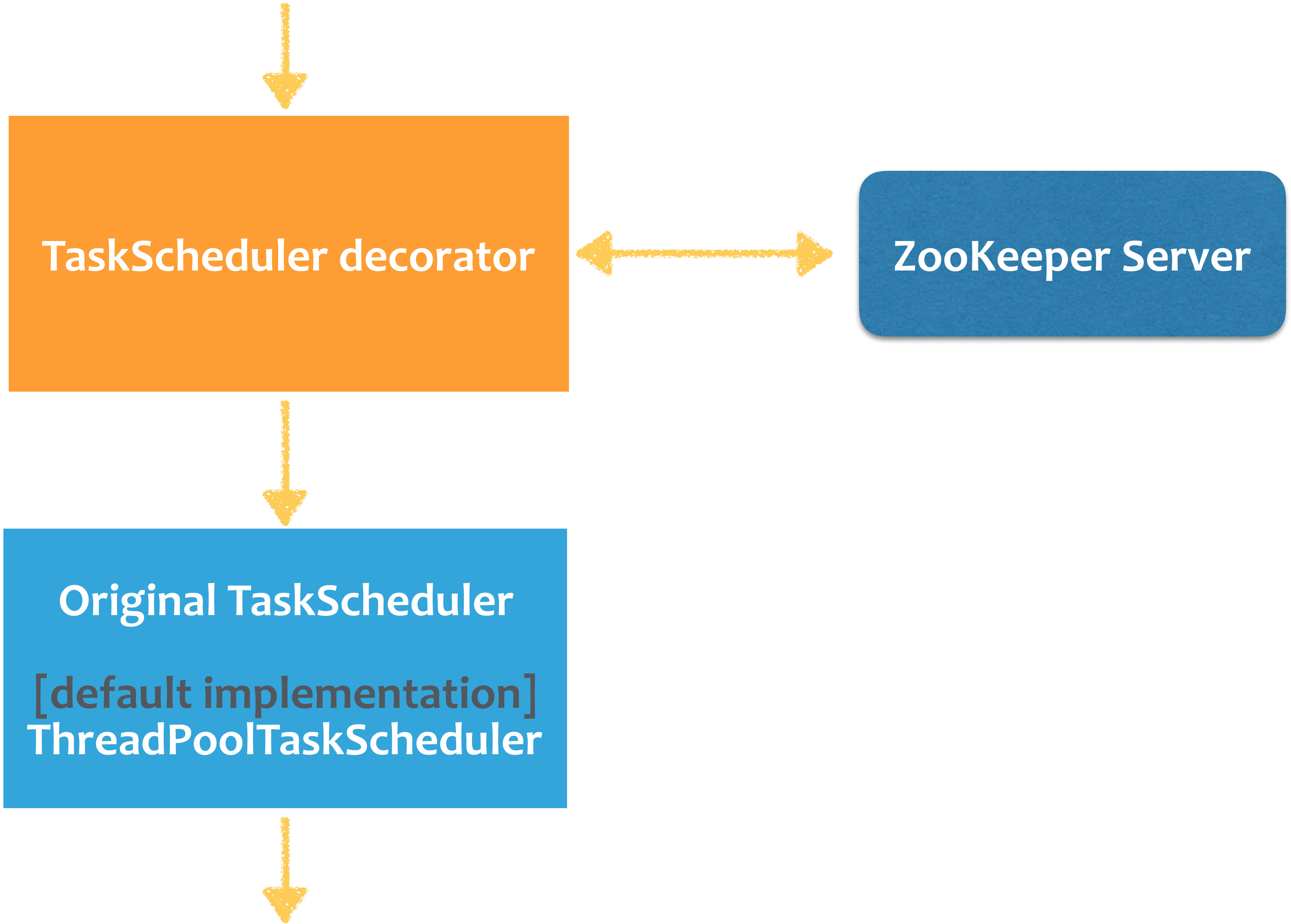
TaskScheduler

[default implementation] ThreadPoolTaskScheduler

Schedule runnable to

java.util.concurrent.Scheduled ExecutorService

ADAPTING INTO SCHEDULE FRAMEWORK



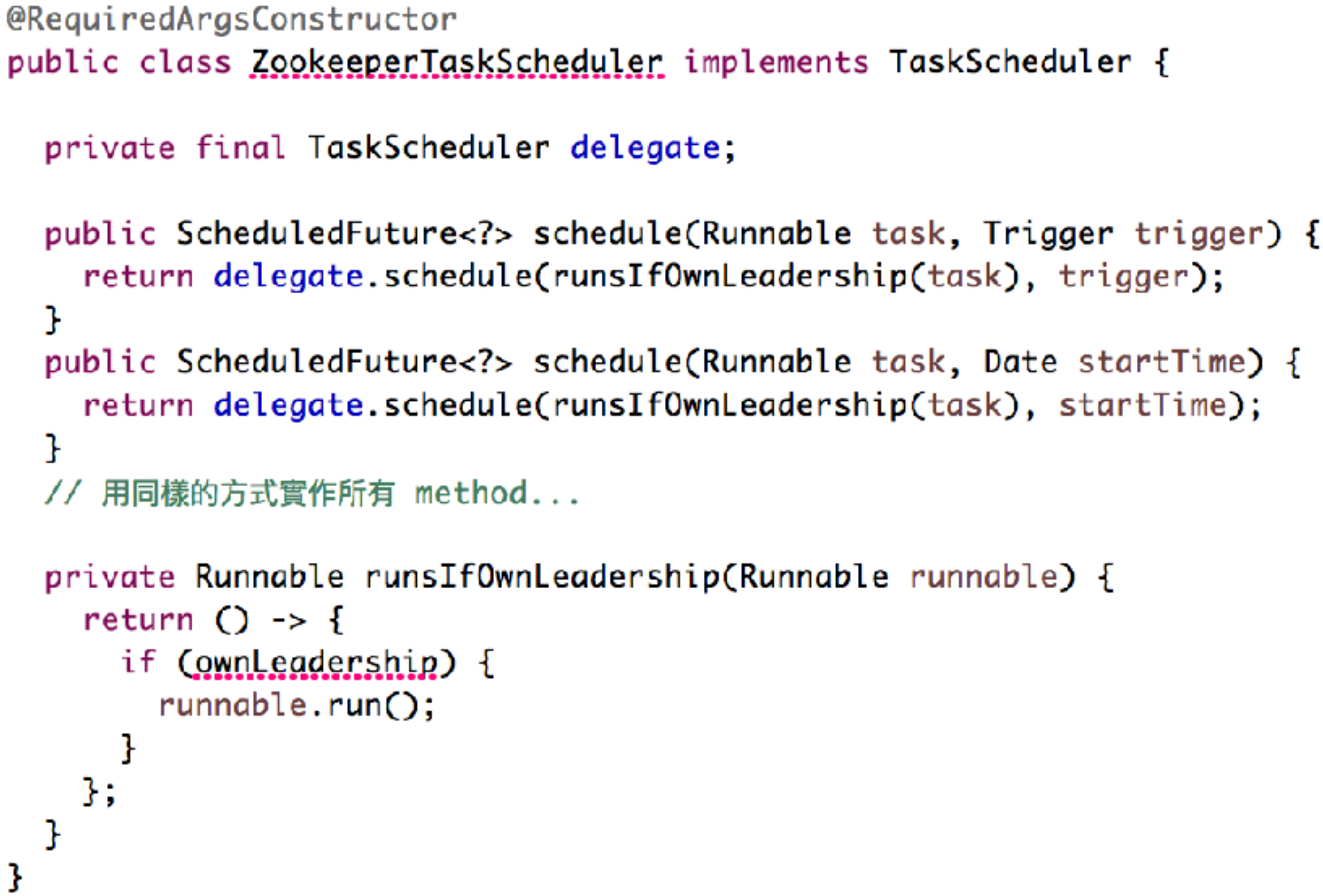
TaskScheduler decorator

ZooKeeper Server

Original TaskScheduler

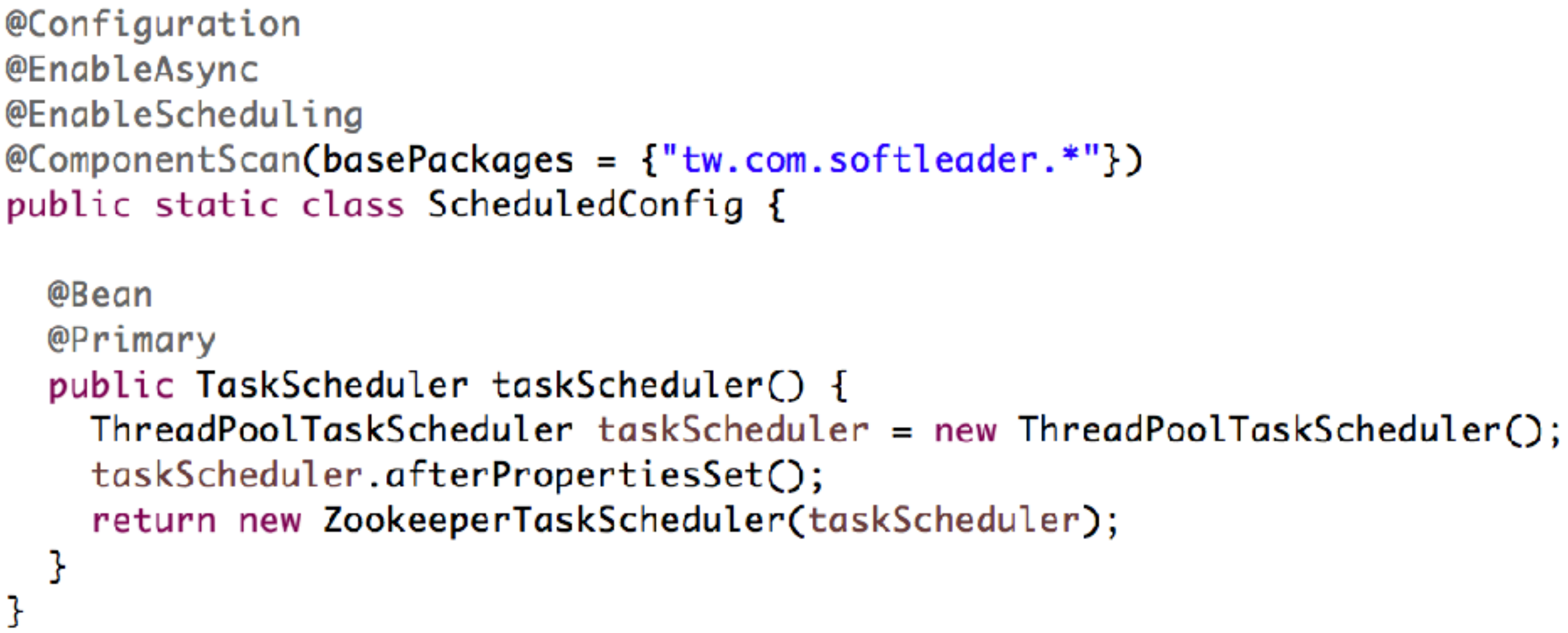
[default implementation] ThreadPoolTaskScheduler

PROGRAMMING TO SCHEDULE FRAMEWORK



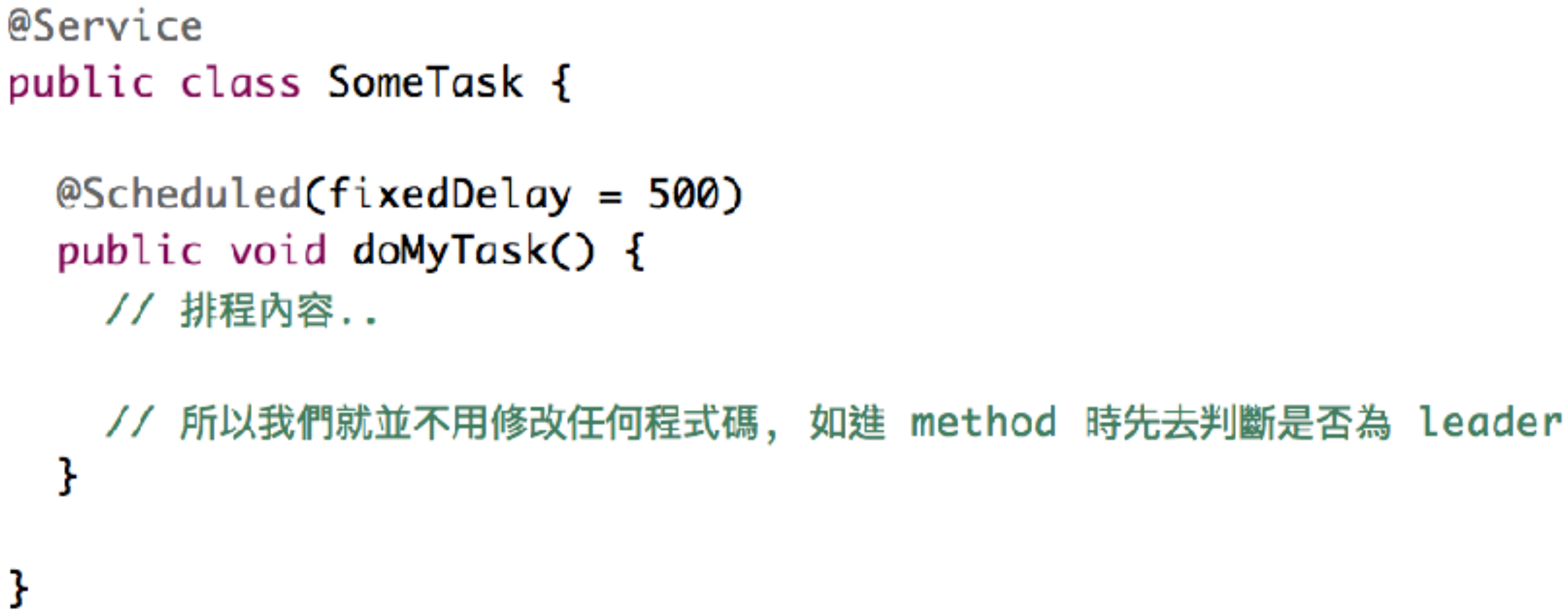
```java
@RequiredArgsConstructor
public class ZookeeperTaskScheduler implements TaskScheduler {

  private final TaskScheduler delegate;

  public ScheduledFuture<?> schedule(Runnable task, Trigger trigger) {
    return delegate.schedule(runsIfOwnLeadership(task), trigger);
  }
  public ScheduledFuture<?> schedule(Runnable task, Date startTime) {
    return delegate.schedule(runsIfOwnLeadership(task), startTime);
  }
  // 用同樣的方式實作所有 method...

  private Runnable runsIfOwnLeadership(Runnable runnable) {
    return () -> {
      if (ownLeadership) {
        runnable.run();
      }
    };
  }
}
```

PROGRAMMING TO SCHEDULE FRAMEWORK



```
@Configuration
@EnableAsync
@EnableScheduling
@ComponentScan(basePackages = {"tw.com.softleader.*"})
public static class ScheduledConfig {

  @Bean
  @Primary
  public TaskScheduler taskScheduler() {
    ThreadPoolTaskScheduler taskScheduler = new ThreadPoolTaskScheduler();
    taskScheduler.afterPropertiesSet();
    return new ZookeeperTaskScheduler(taskScheduler);
  }
}
```

PROGRAMMING TO SCHEDULE FRAMEWORK

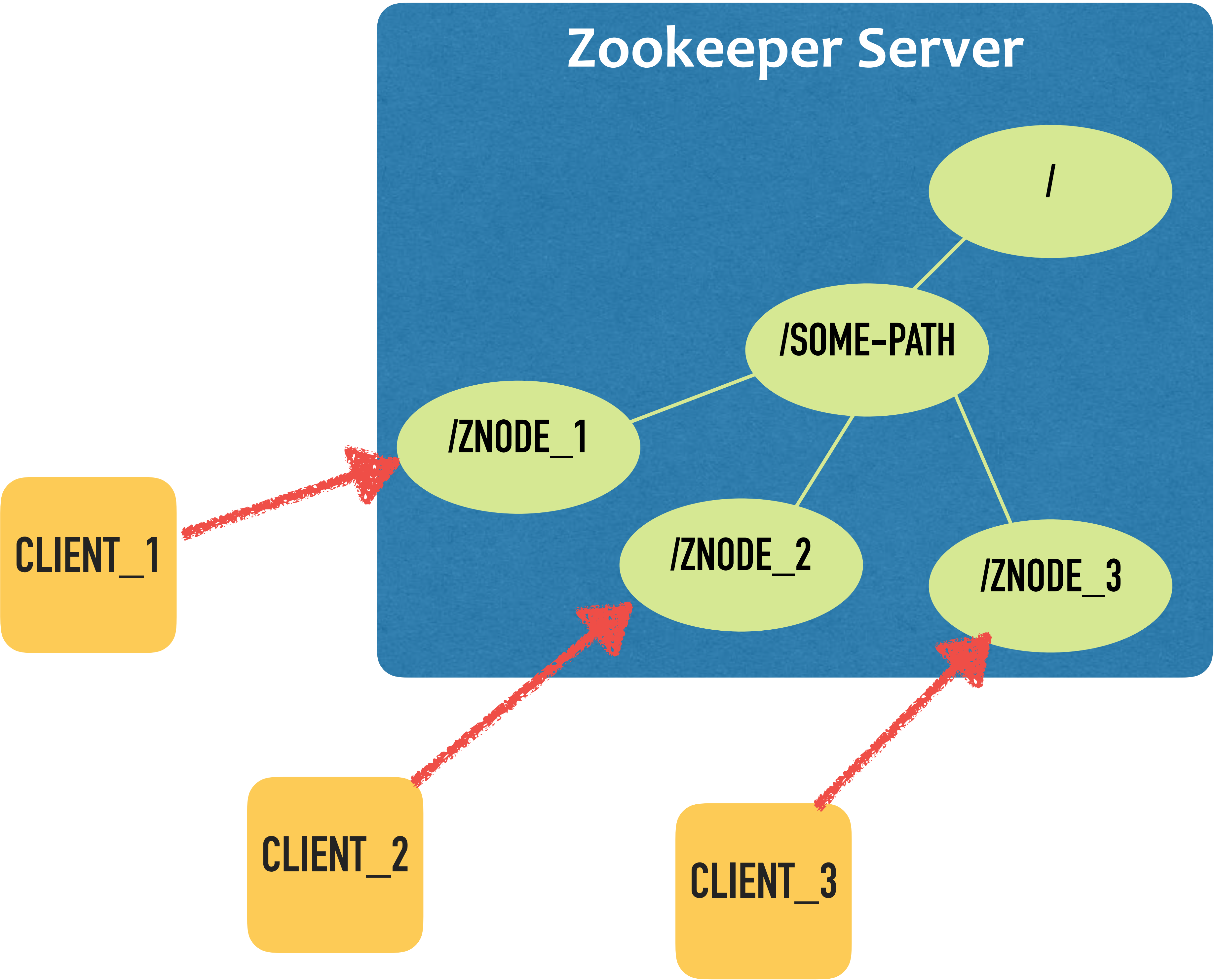


```java
@Service
public class SomeTask {

  @Scheduled(fixedDelay = 500)
  public void doMyTask() {
    // 排程內容..

    // 所以我們就並不用修改任何程式碼, 如進 method 時先去判斷是否為 leader
  }

}
```

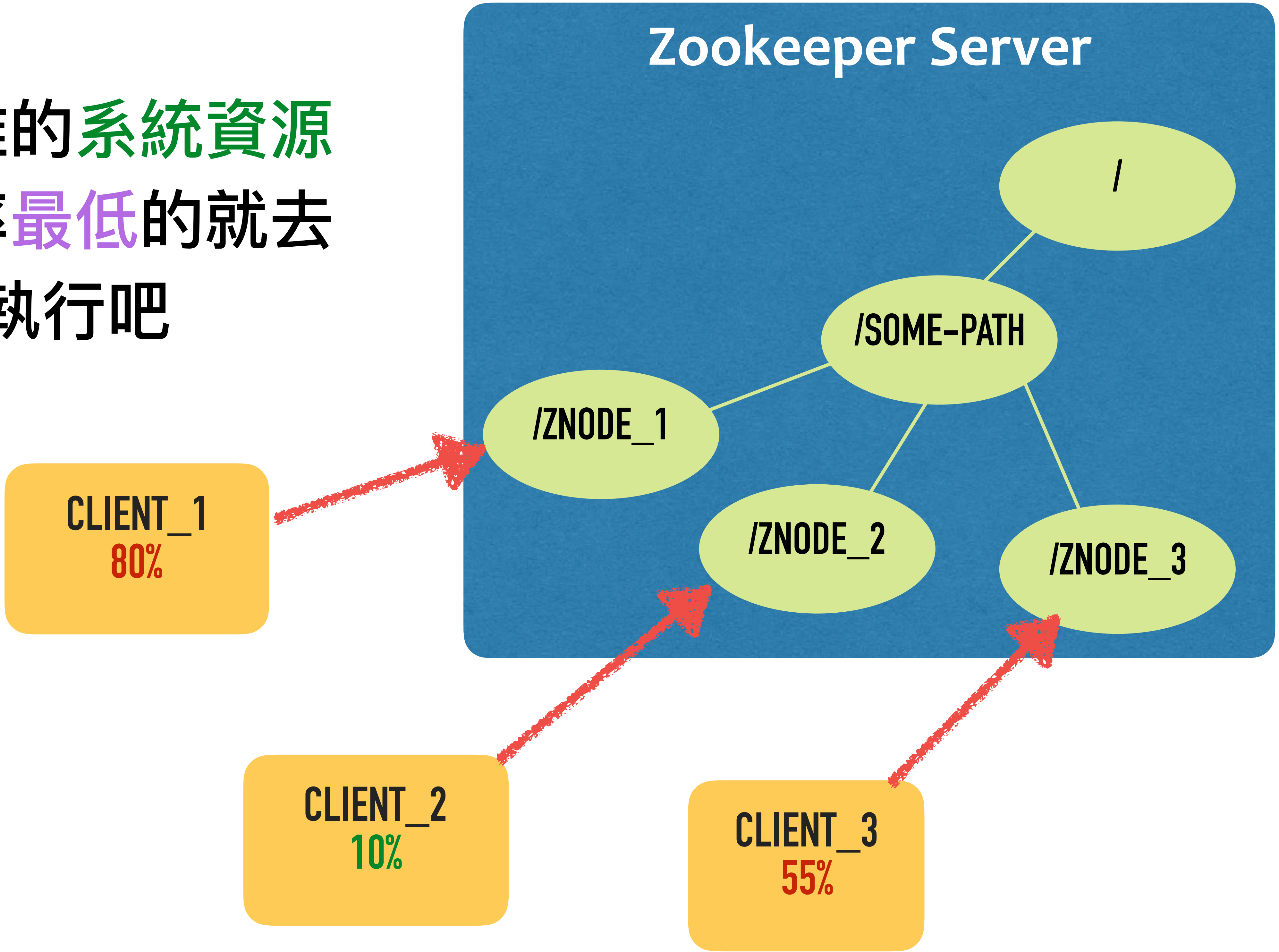
來看看實際的程式碼吧…

還可以做些什麼?

LEADER ELECTION



看看誰的系統資源使用率最低的就去執行吧



Please leave your comments ↓

https://github.com/methodho/jcconf2016-zookeeper

Please leave your comments ↓

https://github.com/methodho/jcconf2016-zookeeper

And we’re hiring!

http://www.softleader.com.tw/web/career.html