

适合自己的配置中心

造轮之路由此上

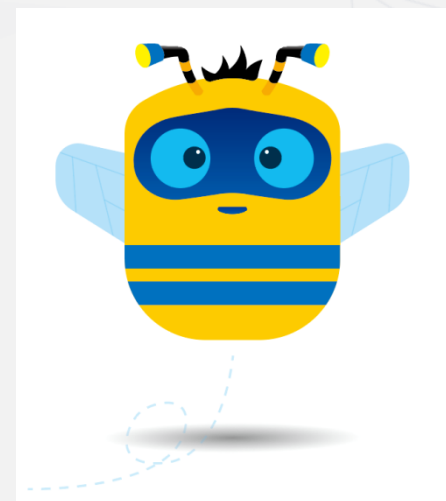
制作人：施正楚 2017/5/11



自我介绍

施正楚:

非学无以广才，非志无以成学。



为什么要做自己的配置中心

钱多人傻又有闲

基础服务

自定义需求

我们想要的



spring

在spring中方便的使用

直接引入一个jar的依赖，就马上可以使用。减少对业务系统的侵入。



预存配置

针对场景提前预设

在配置管理系统中提前设定好配置，一键应用配置。可用于回滚和应急。



可视化管理

Web管理

可以直接在线编辑配置，支持批量操作。

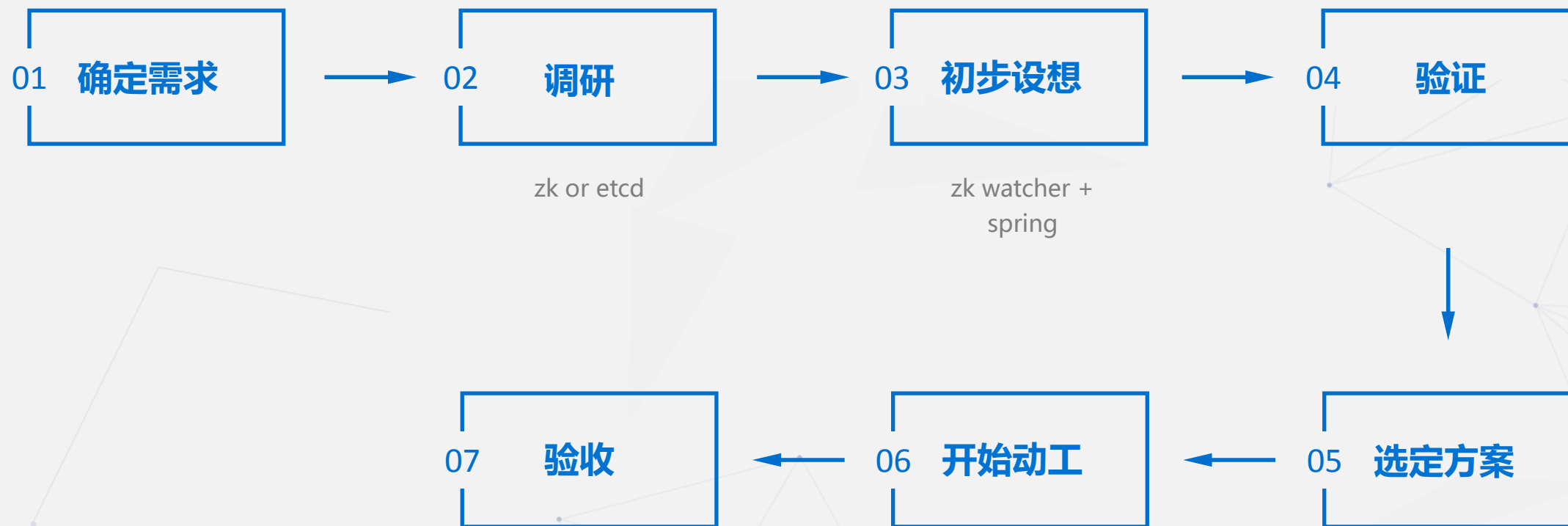


权限控制

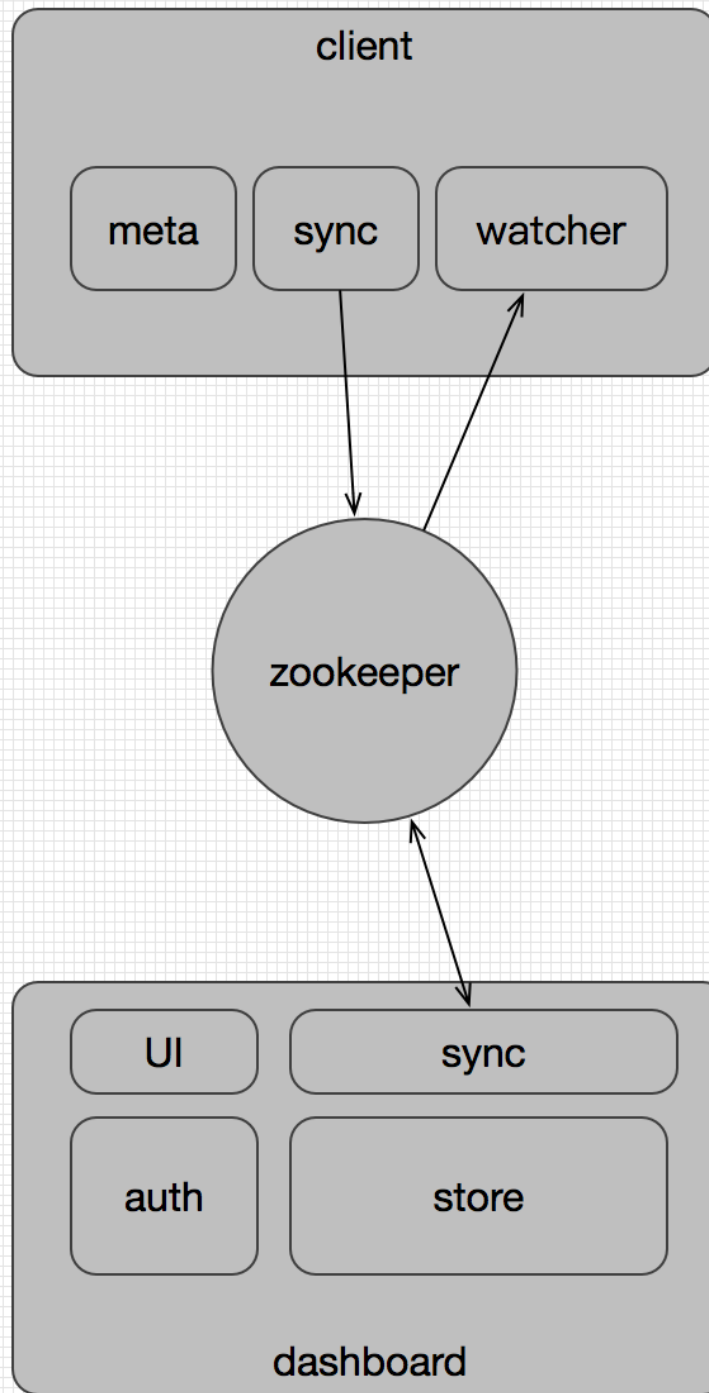
确保配置项安全

不同的用户有不同的查看和设置权限。

方案选择



鸟瞰

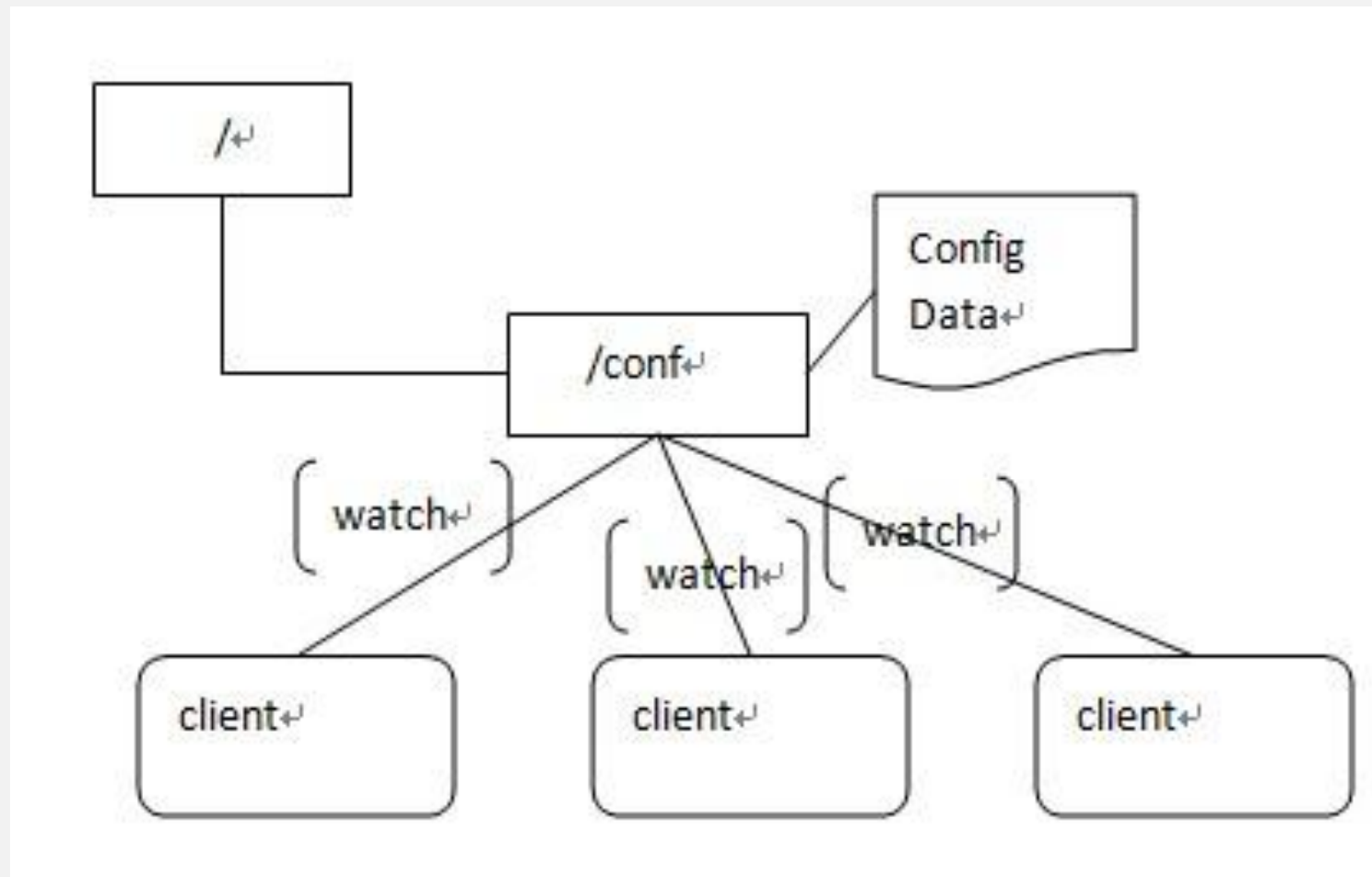


Zookeeper

树型结构

分布式

watch



Zookeeper

```
public byte[] getData(final String path, Watcher watcher, Stat stat);
```

```
abstract public void process(WatchedEvent event);
```

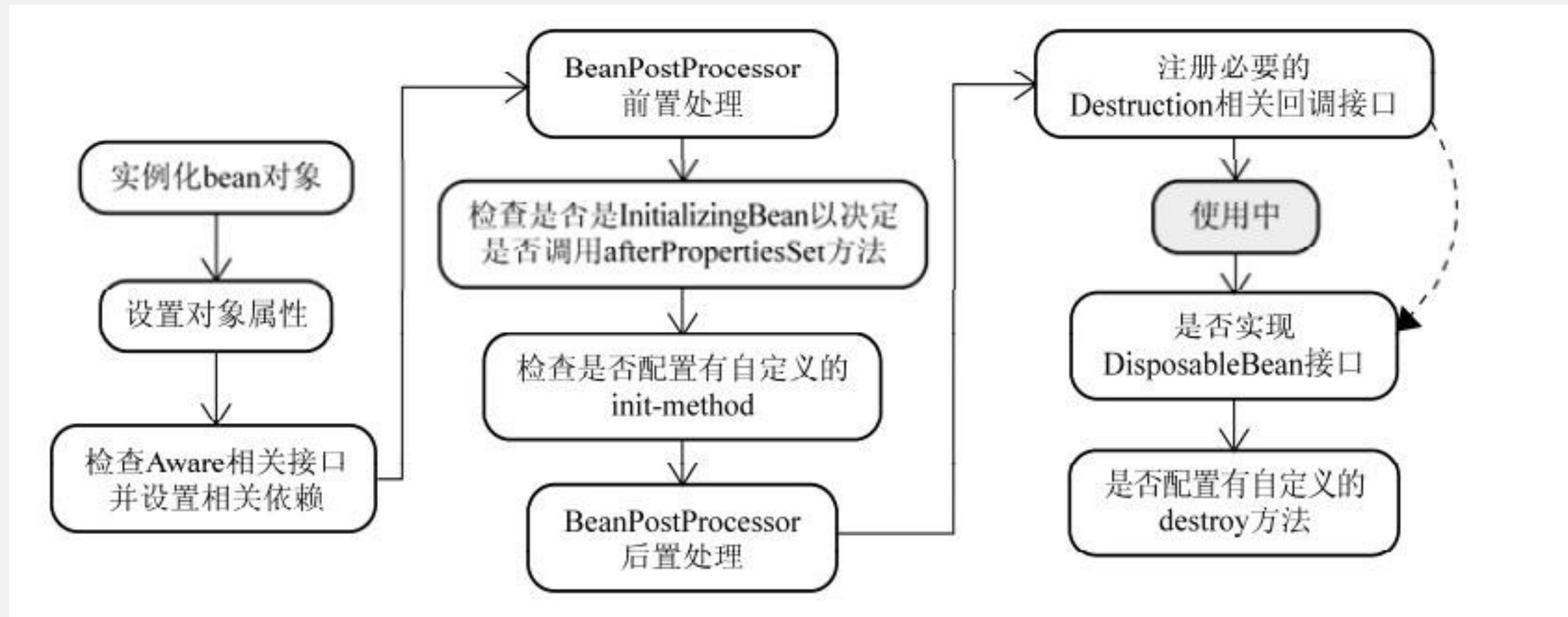
```
public class WatchedEvent {  
    final private KeeperState keeperState;  
    final private EventType eventType;  
    private String path;  
}
```

Zookeeper

KeeperState: Expired session 过期重连

EventType: NodeDataChanged 修改数据

Bean的启动过程



识别配置项

注解标记配置项ZkConfigAnnotation

- Key
 - Default value
-

BeanPostProcessor前置处理

识别配置项

- 第一步：识别ZkConfigAnnotation
 - 第二步：从zk读数据，并设置回bean
-

BeanPostProcessor.postProcessBeforeInitialization(Object bean, String beanName)

```
Field[] fields = bean.getClass().getDeclaredFields();  
field.getAnnotation(ZkConfigAnnotation.class)
```



识别配置项

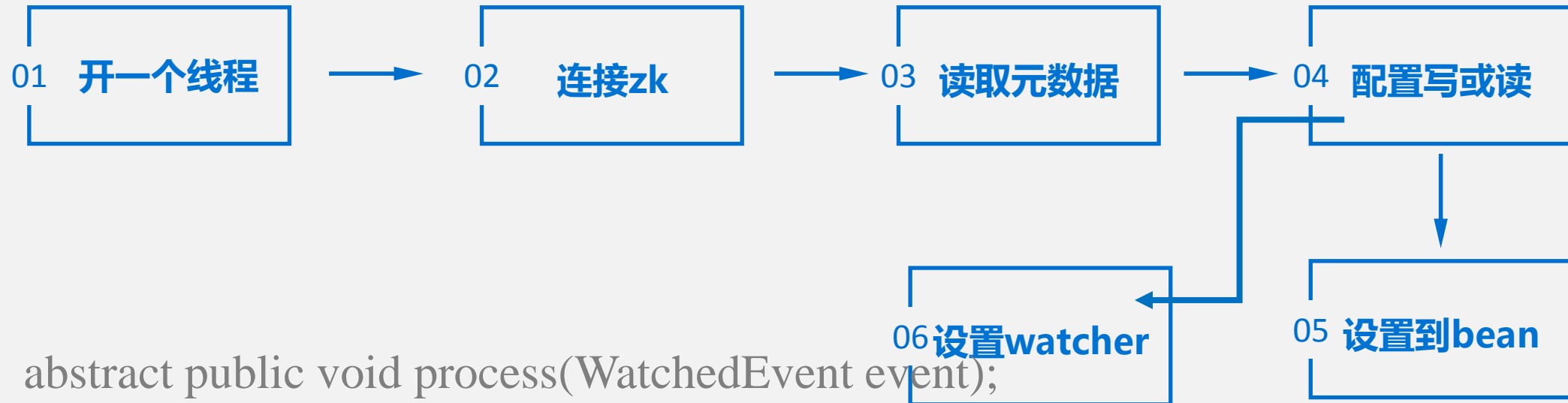
保存配置项元数据ConfigMeta

- Scene
 - Service
 - Path也就是key
 - Value也就是data
 - BeanName
 - Field
-

接收配置变更

在哪个地方启动接收数据变更呢？

ApplicationListener<ContextRefreshedEvent>



Watch处理

- Expired则关闭连接
 - 取path
 - 取元数据
 - 取data
 - 设置Field值
-

zk如何处理断线重连?

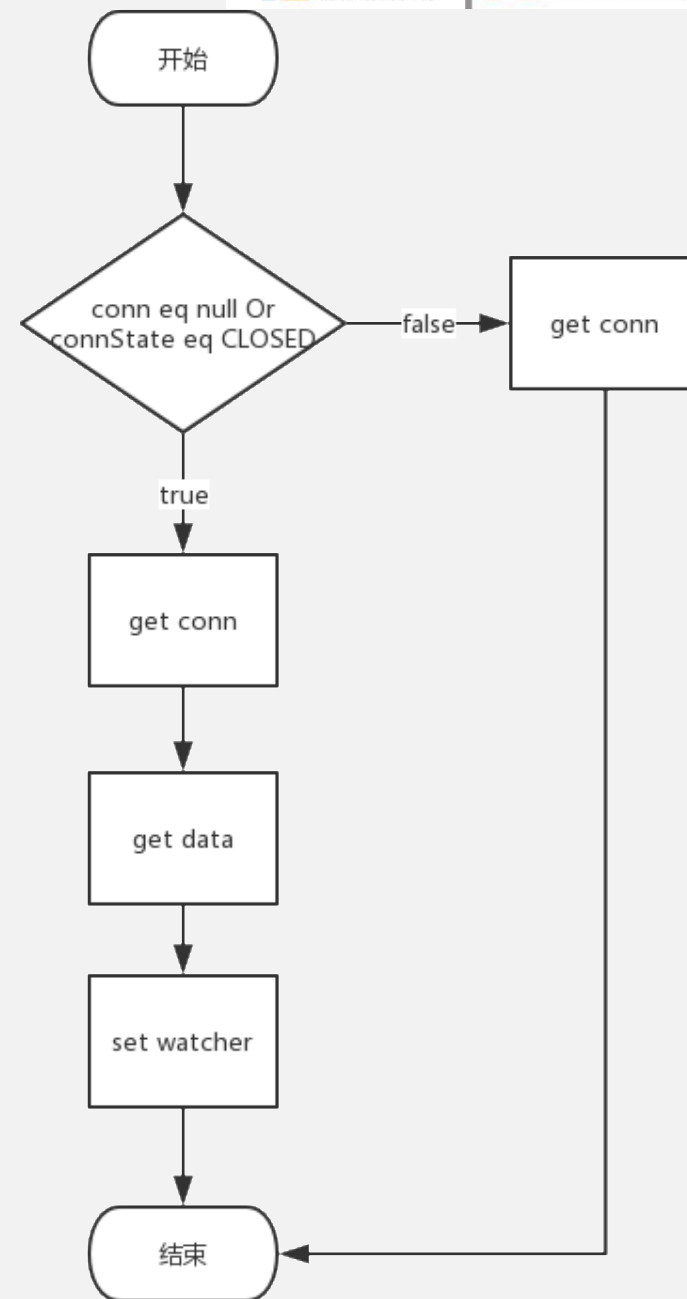
重连

判断是否断线 ZooKeeper.States.CLOSED

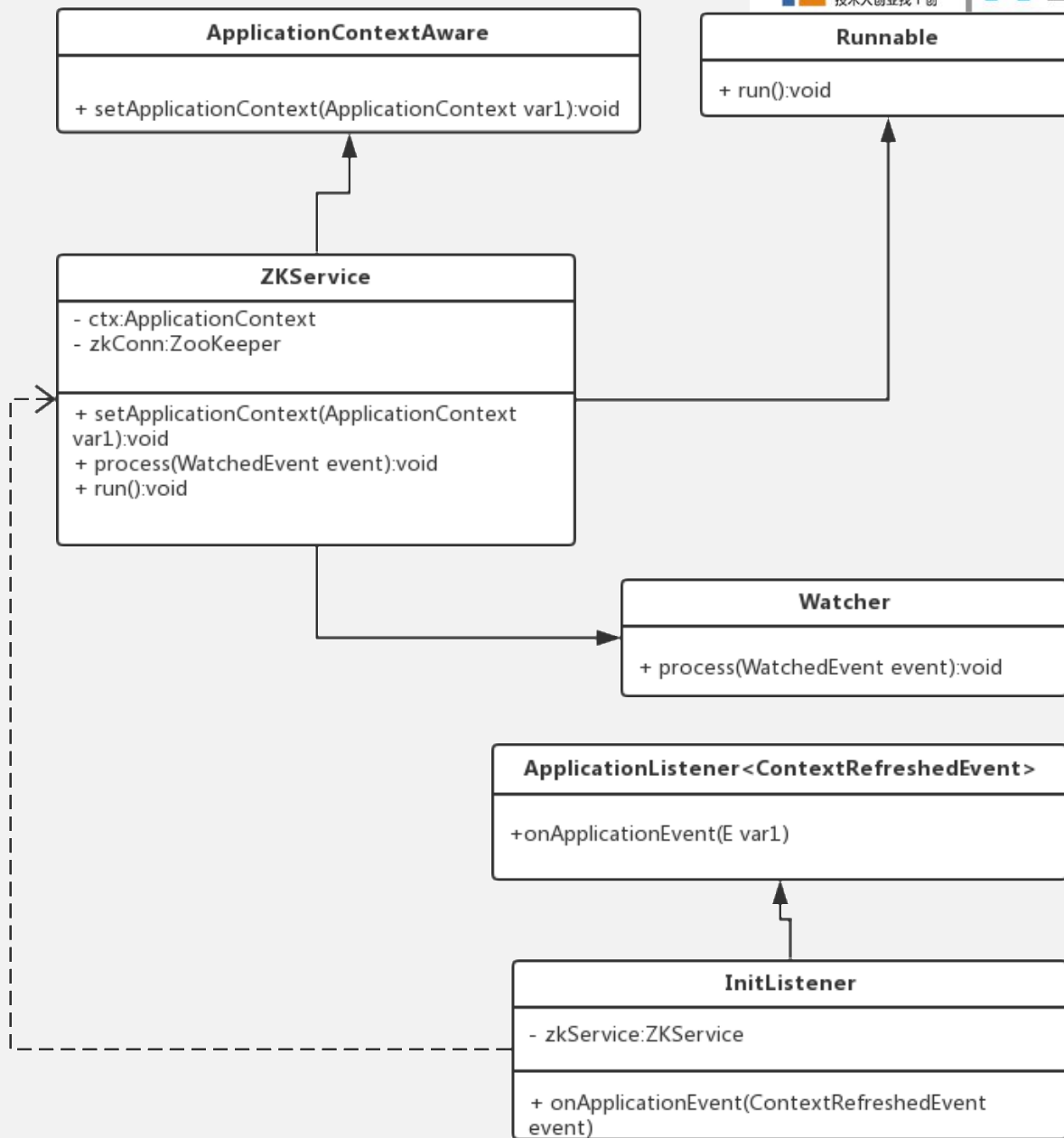
重连

重新读取线上配置

设置watch



Client端鸟瞰



PATH设计



/conf/prod/dashboard/redis/port----7777

关于Scene

不同的环境配置不同的场景

生产

预发

某活动

方便的复制存储

Scene和service可以方便的复制

修改后就是一个新的配置项

并配置给指定的服务

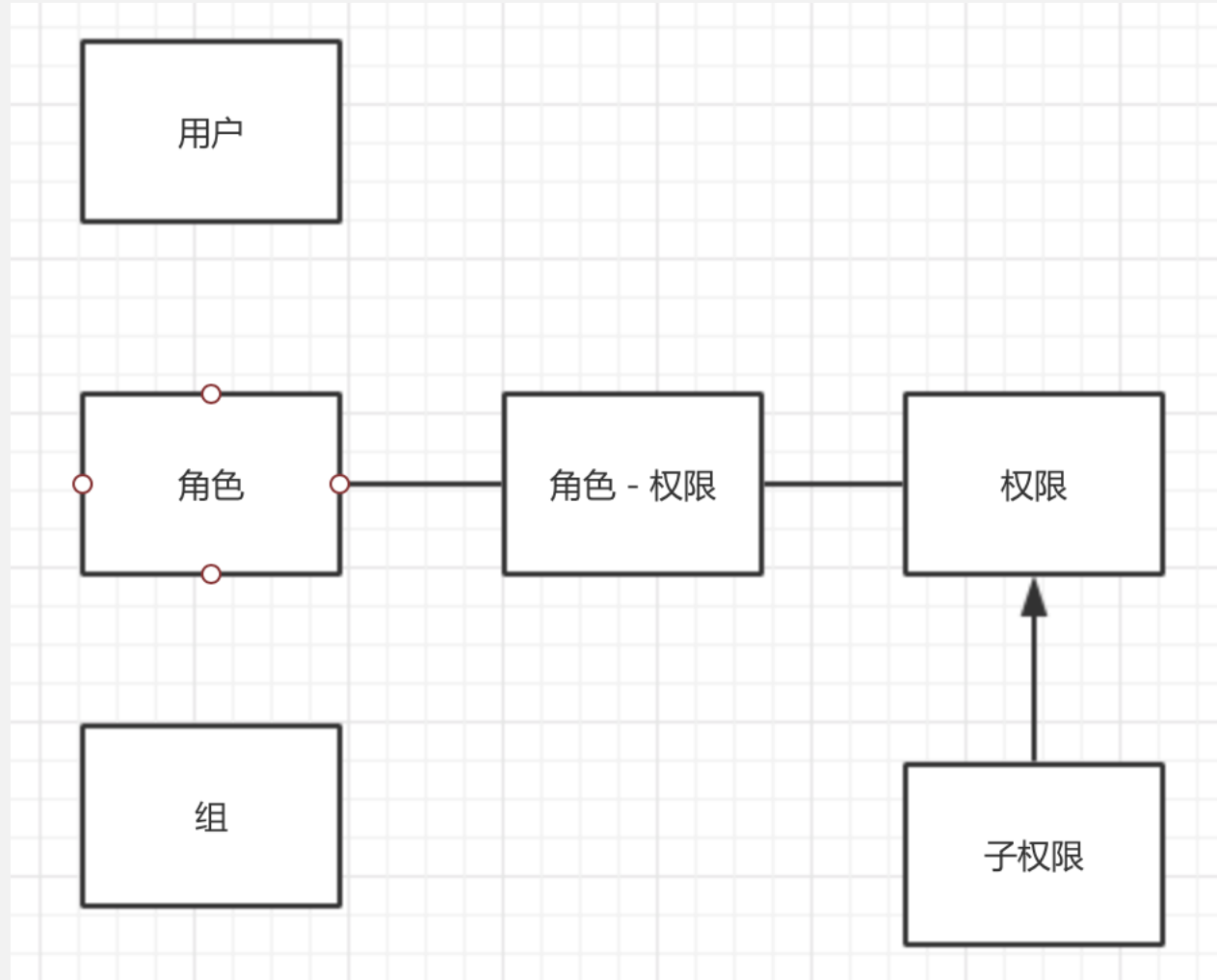
权限管理

暴雪降生丹妮莉丝 不焚者 龙之
母 弥林女王 七国全境守护者 卡
丽熙 打碎镣铐之人

RBAC: Role-Based Access Control



权限



谢谢

生活还有诗和远方

施正楚

么么哒~

