



主讲人：付磊

主页：<http://carlosfu.iteye.com/>

## 8-1.Redis Sentinel(哨兵)



# Redis Sentinel

主从复制？

架构

安装配置

客户端连接

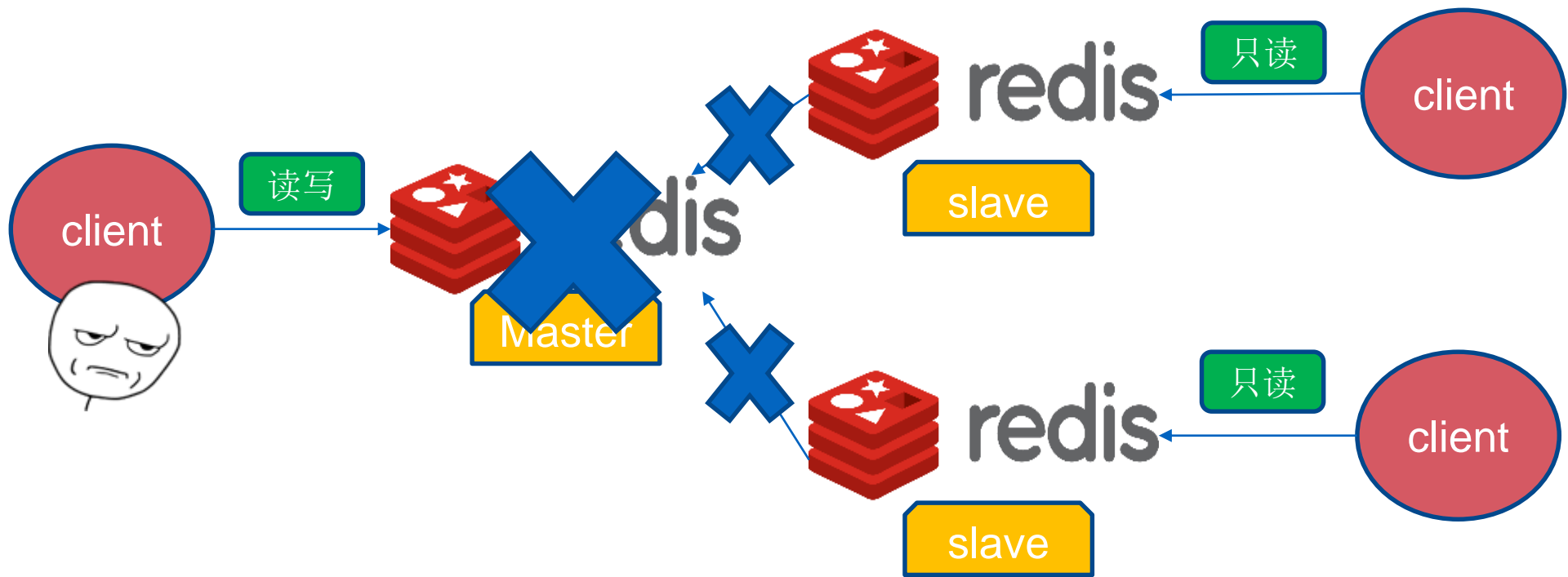
# 1.主从复制

# 主从复制问题

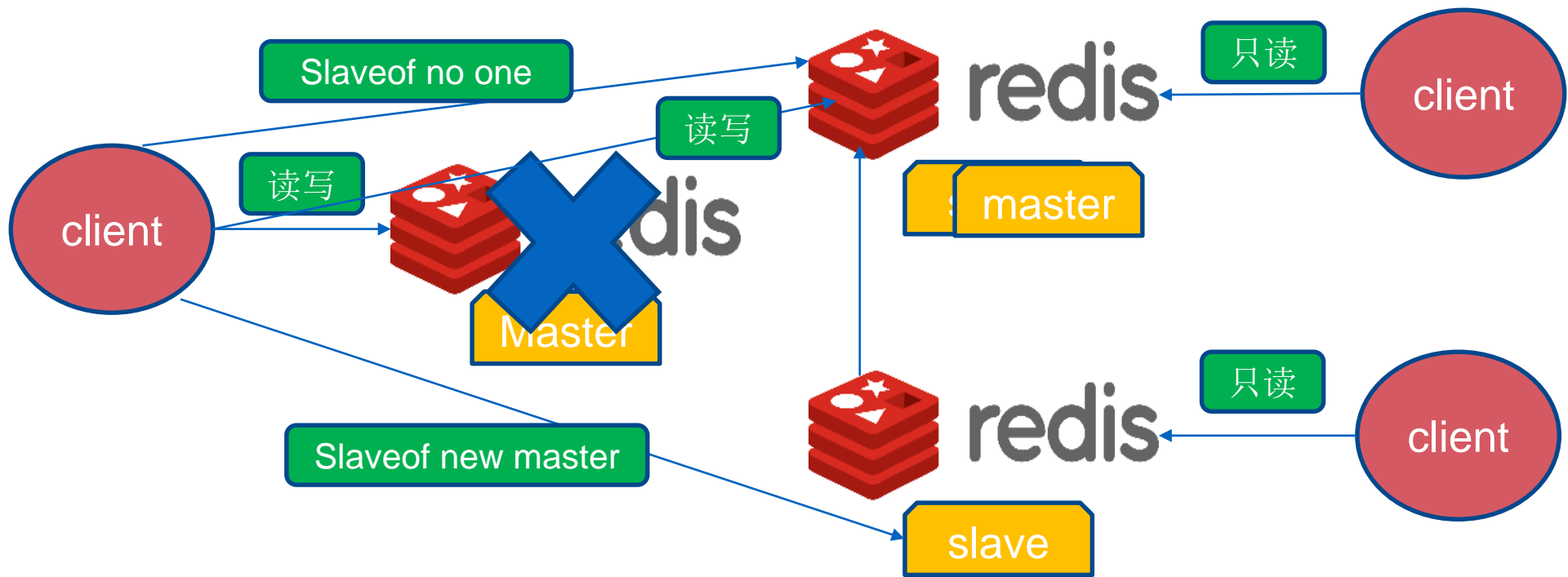
手动故障转移

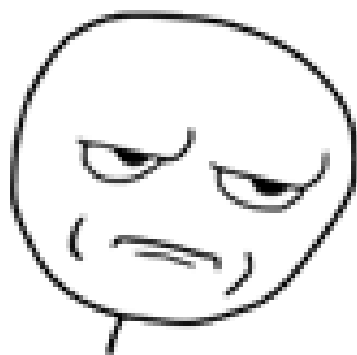
非高可用

## 主从复制-Master宕掉



# 主从复制- Master宕掉故障处理

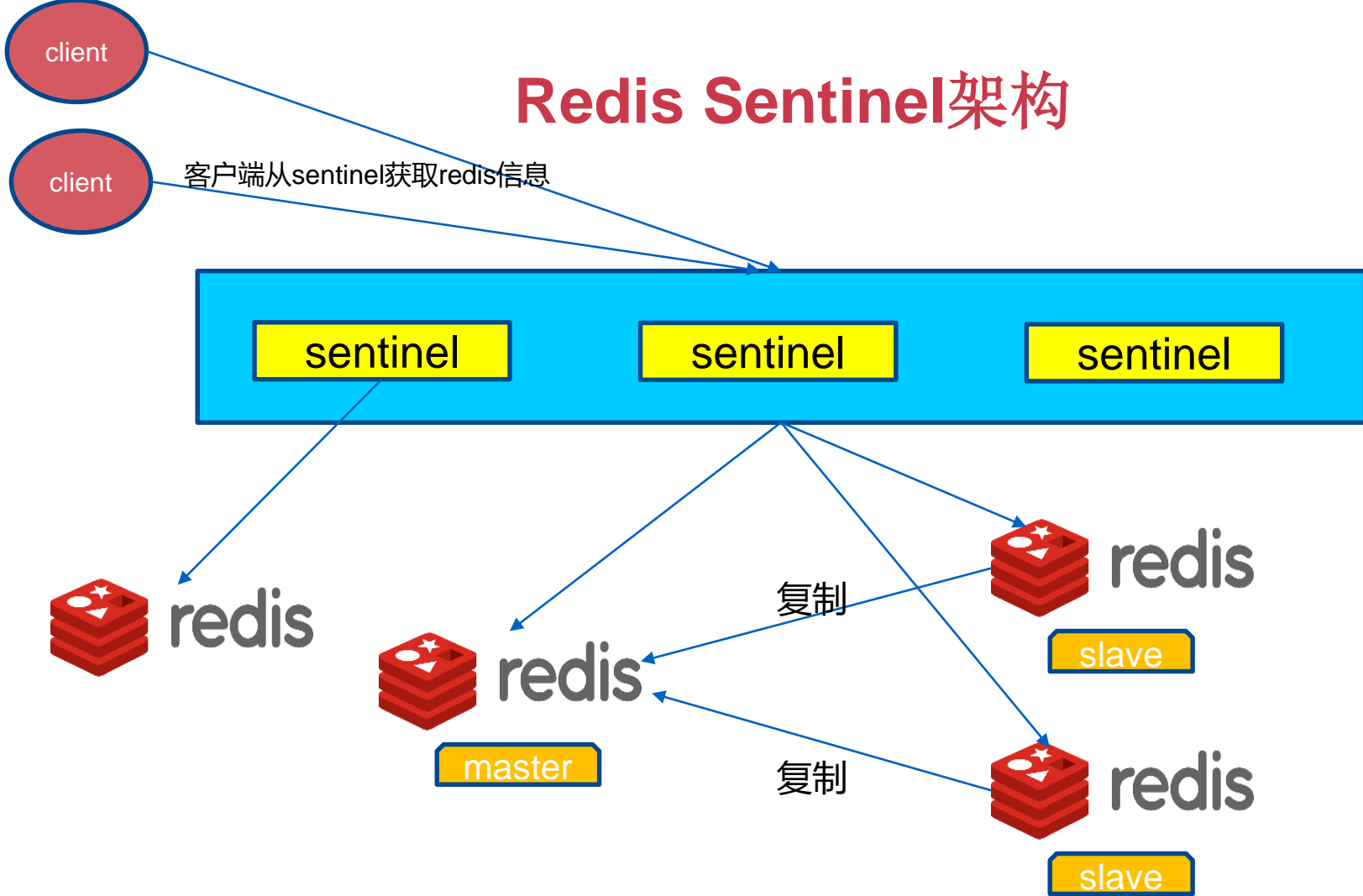




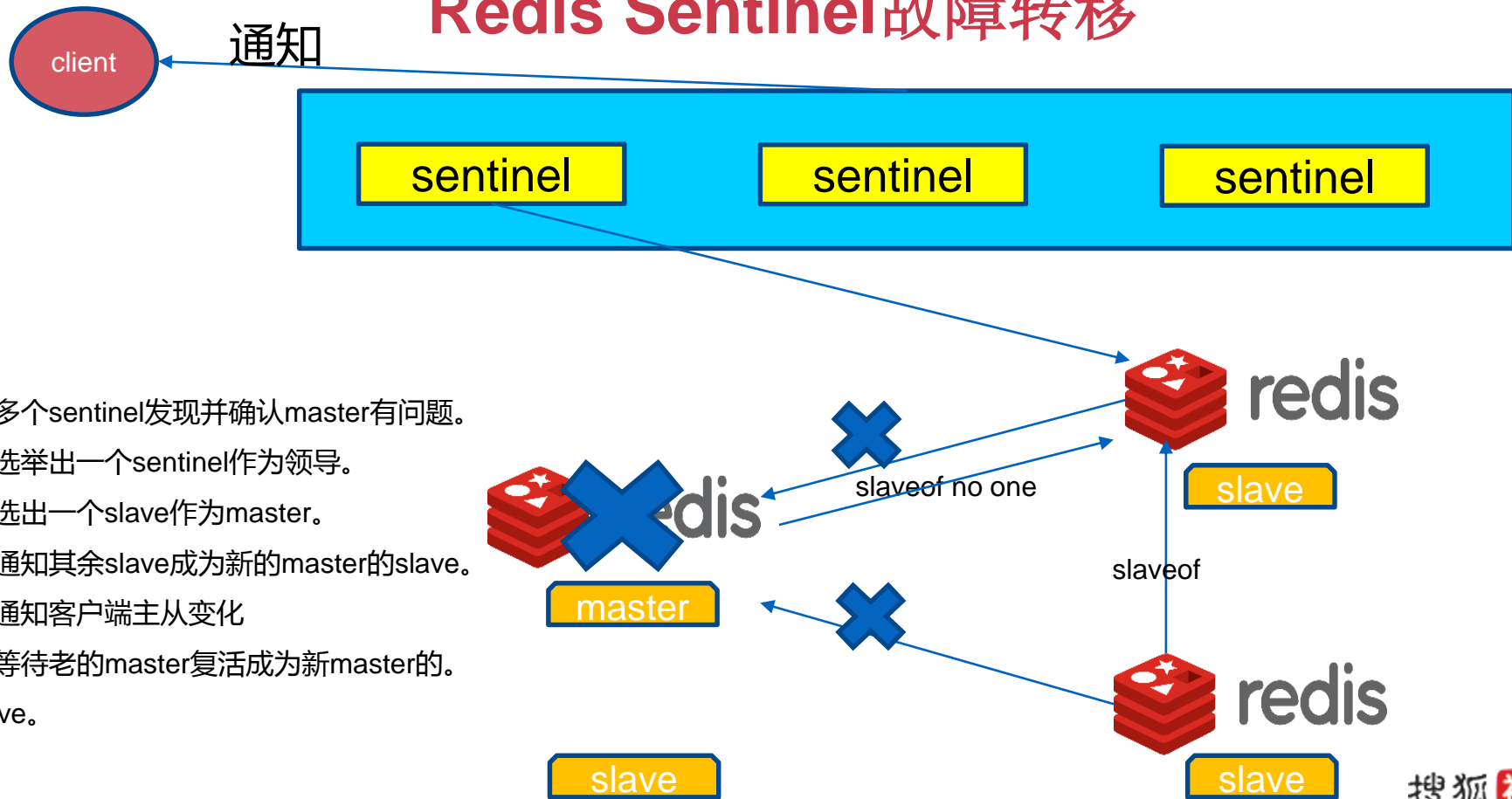


## 2. Redis Sentinel架构

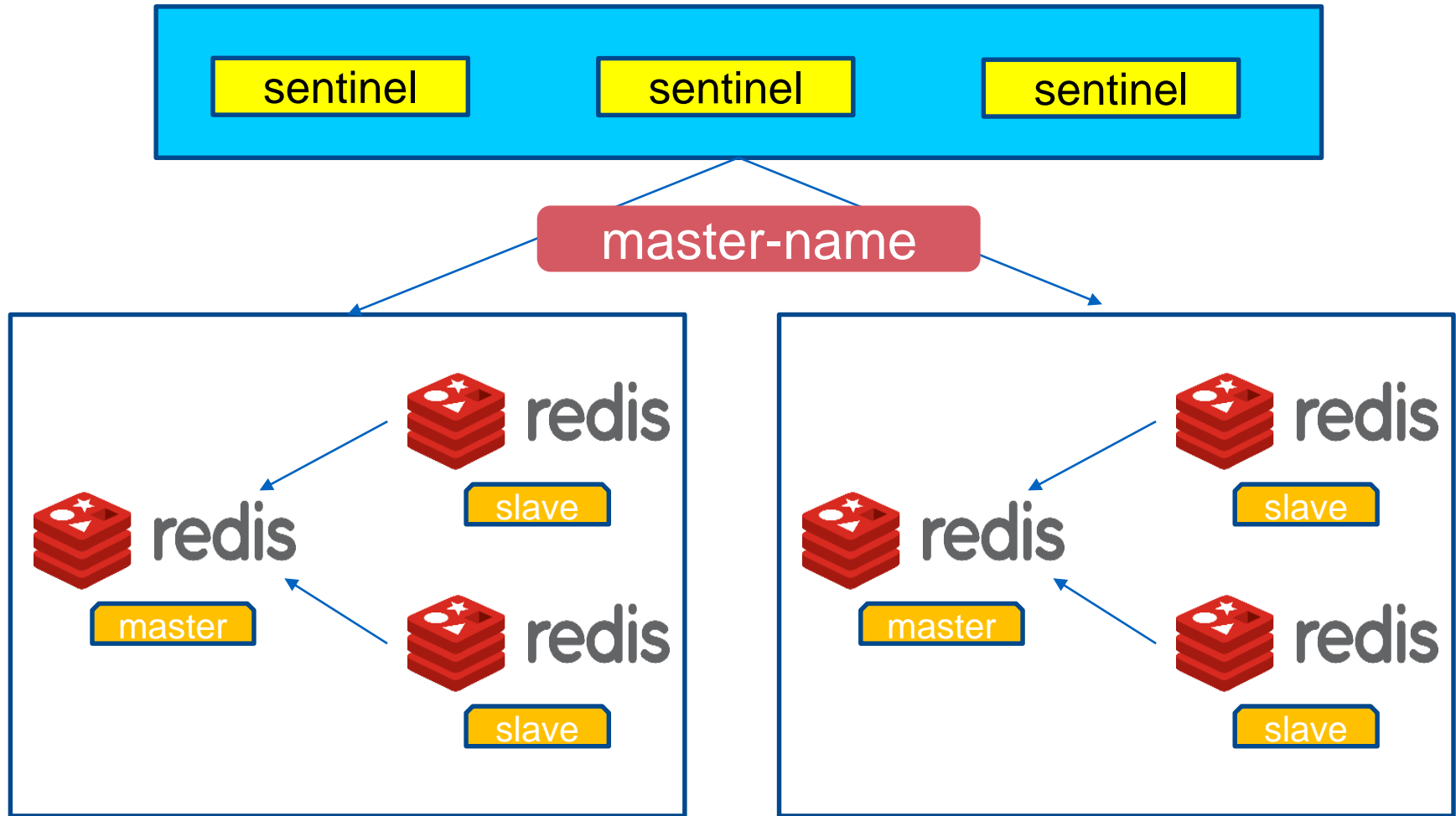
# Redis Sentinel架构



# Redis Sentinel故障转移



1. 多个sentinel发现并确认master有问题。
2. 选举出一个sentinel作为领导。
3. 选出一个slave作为master。
4. 通知其余slave成为新的master的slave。
5. 通知客户端主从变化
6. 等待老的master复活成为新master的slave。





Redis Sentinel failover automatically

### 3. Redis Sentinel安装配置

# 安装与配置

<http://carlosfu.iteye.com/blog/2275021>

# 安装与配置

1. 配置开启主从节点
2. 配置开启sentinel监控主节点。(sentinel是特殊的redis)
3. 实际应该多机器
4. 详细配置节点



# Sentinel主要配置

port 26379

dir /bigpath

sentinel monitor mymaster 127.0.0.1 6379 2

sentinel down-after-milliseconds mymaster 30000

sentinel parallel-syncs mymaster 1

sentinel failover-timeout mymaster 180000

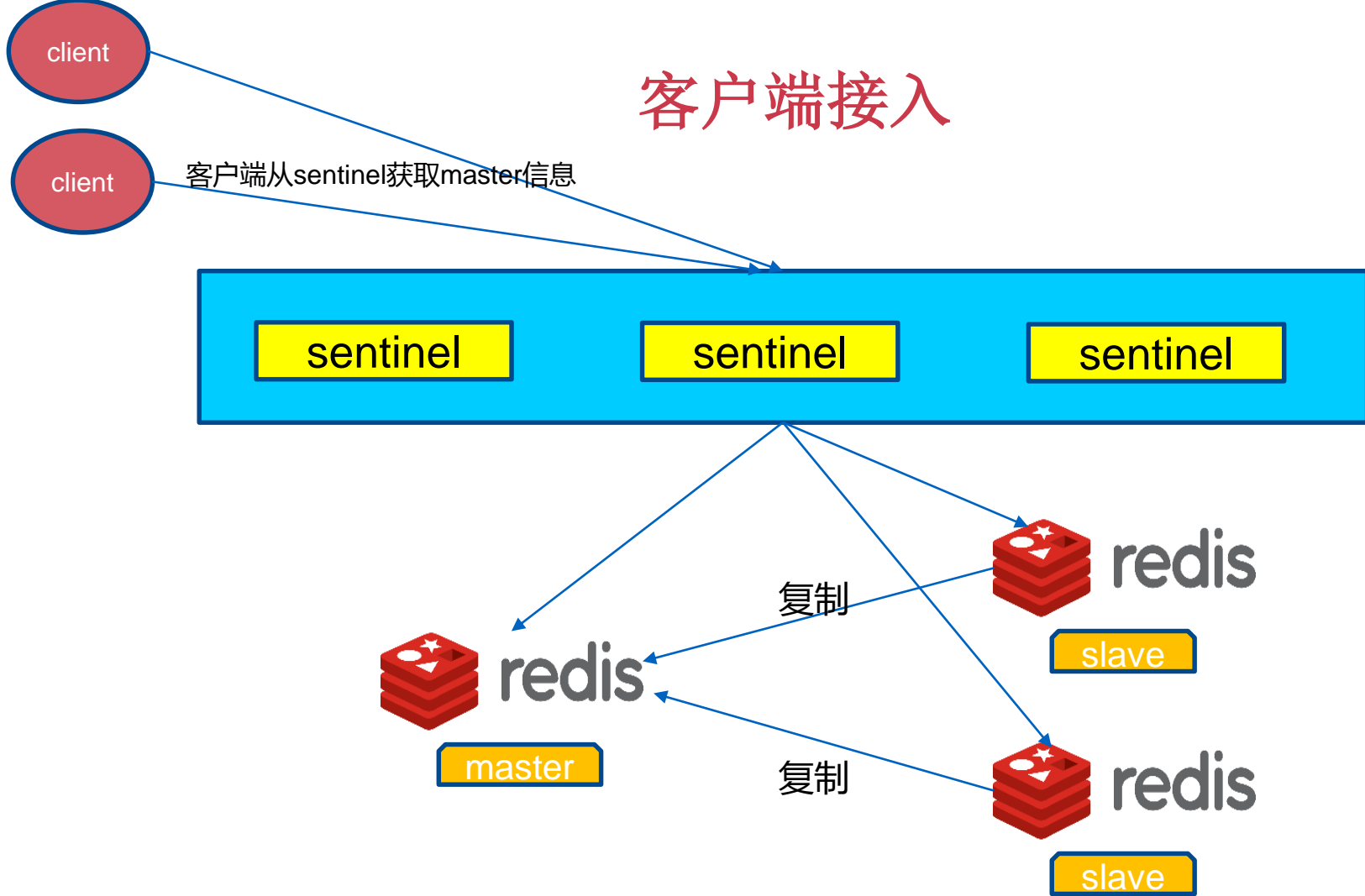
## 4. 客户端

# 客户端

请求响应流程

Jedis

# 客户端接入



# 客户端接入流程

1. Sentinel地址
2. masterName
3. 不是代理模式

# jedis

```
JedisSentinelPool sentinelPool = new JedisSentinelPool(masterName, senti
nelSet, poolConfig, timeout);
Jedis jedis = null;
try {
    jedis = redisSentinelPool.getResource();
    //jedis command
} catch (Exception e) {
    logger.error(e.getMessage(), e);
} finally {
    if (jedis != null)
        jedis.close();
}
```

# 实验

安装

故障转移

配置

客户端



# 实现原理

如何监控

领导者选举

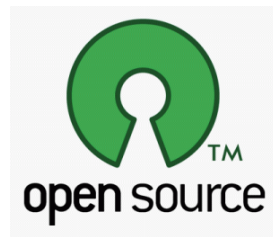
主客观失败

协议









搜狐视频Redis私有云平台开源了！！

Github主页：<https://github.com/sohutv/cacheccloud>

QQ群：534429768