

课时4

服务容错保护-Hystrix

1. 服务雪崩产生原因及解决方案
2. Hystrix工作原理
3. Hystrix配置及监控方式

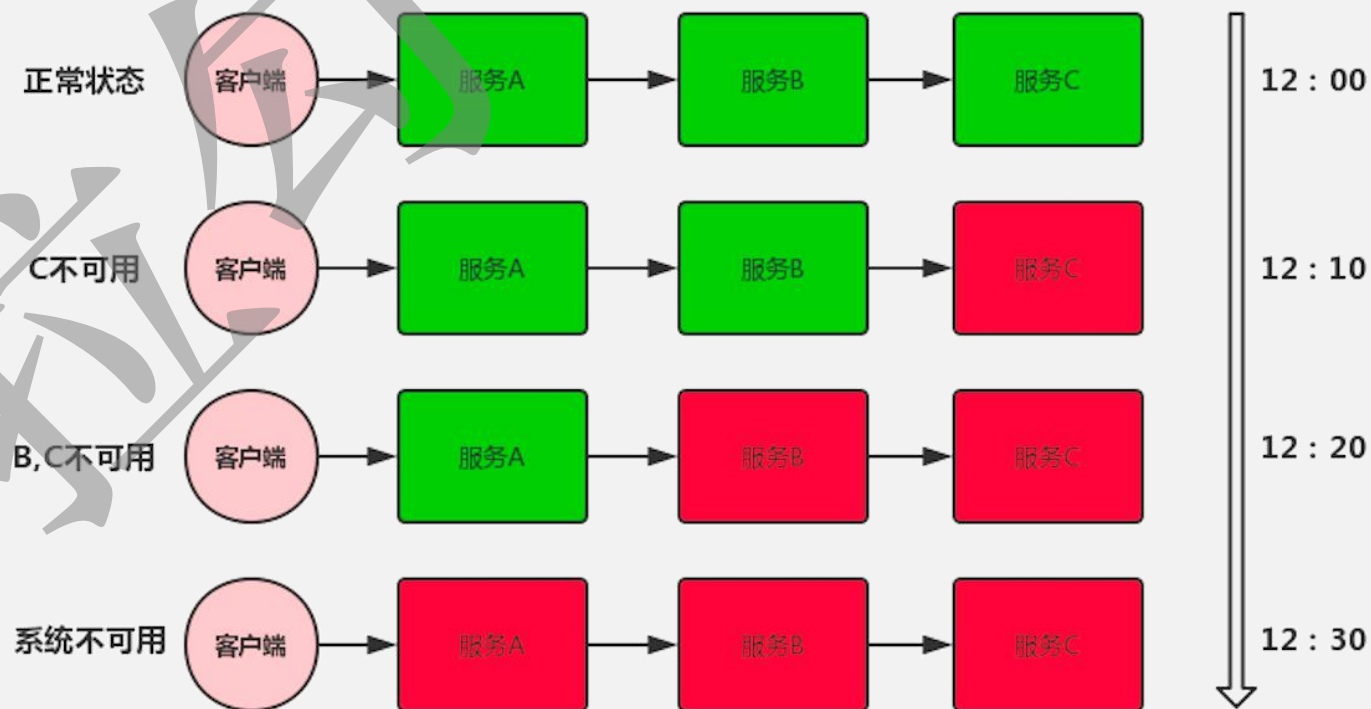
服务雪崩效应

微服务架构下，会存在服务之间相互依赖调用的情况

当某个服务不可用时，很容易因为服务之间的依赖关系使故障扩大

从而造成整个系统不可用的情况

这种现象被称为服务雪崩效应



服务雪崩产生原因

服务故障（提供者）

代码Bug

服务器硬件故障

慢SQL性能问题

请求量超出承受能力

服务故障（消费者）

同步调用等待结果导致资源耗尽

同时作为服务提供者被其他服务调用



服务雪崩解决方案



解决方案（提供者）

服务自动扩容，支持高并发
性能优化
流量控制

解决方案（消费者）

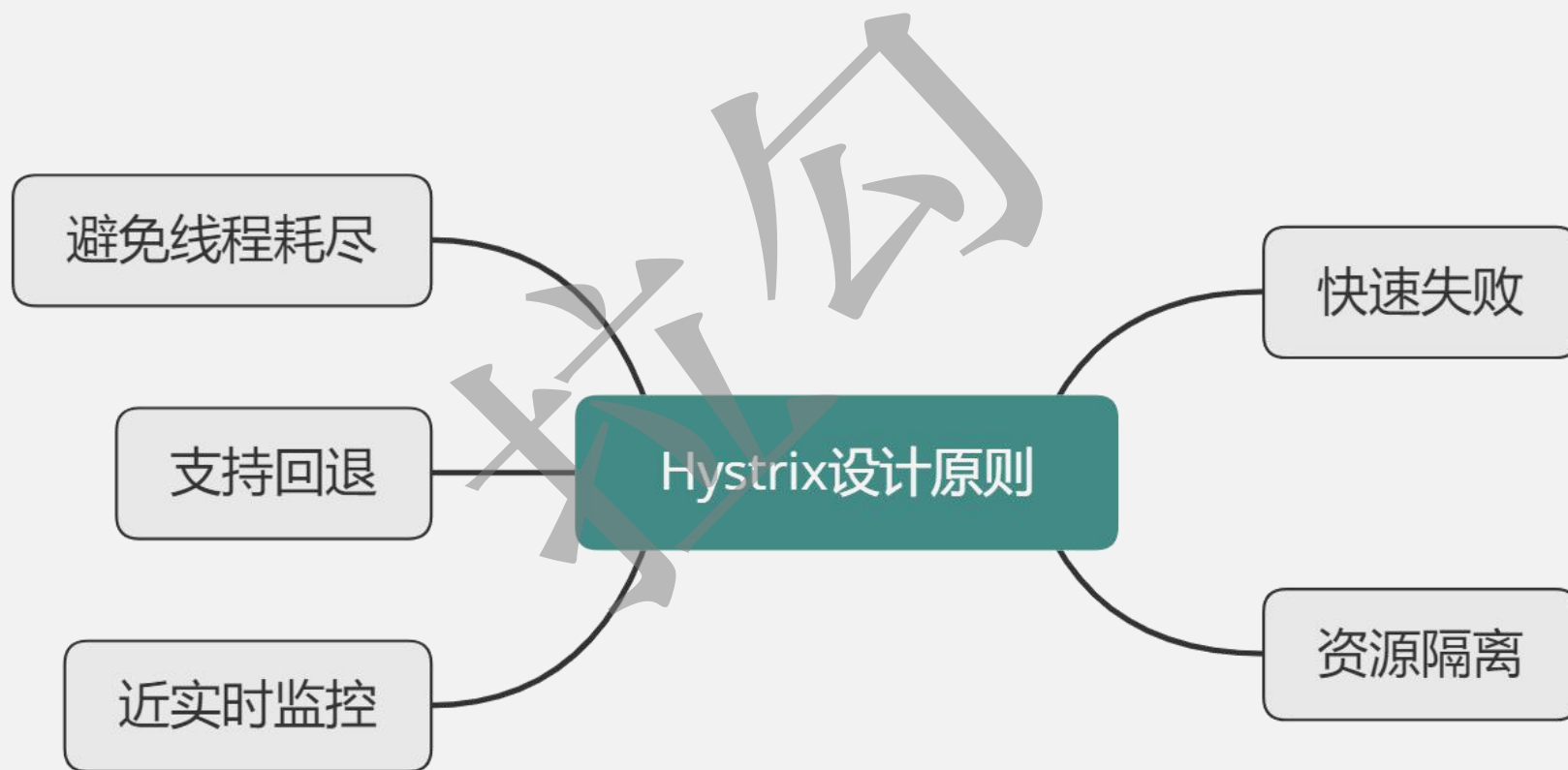
资源隔离，快速失败

Hystrix

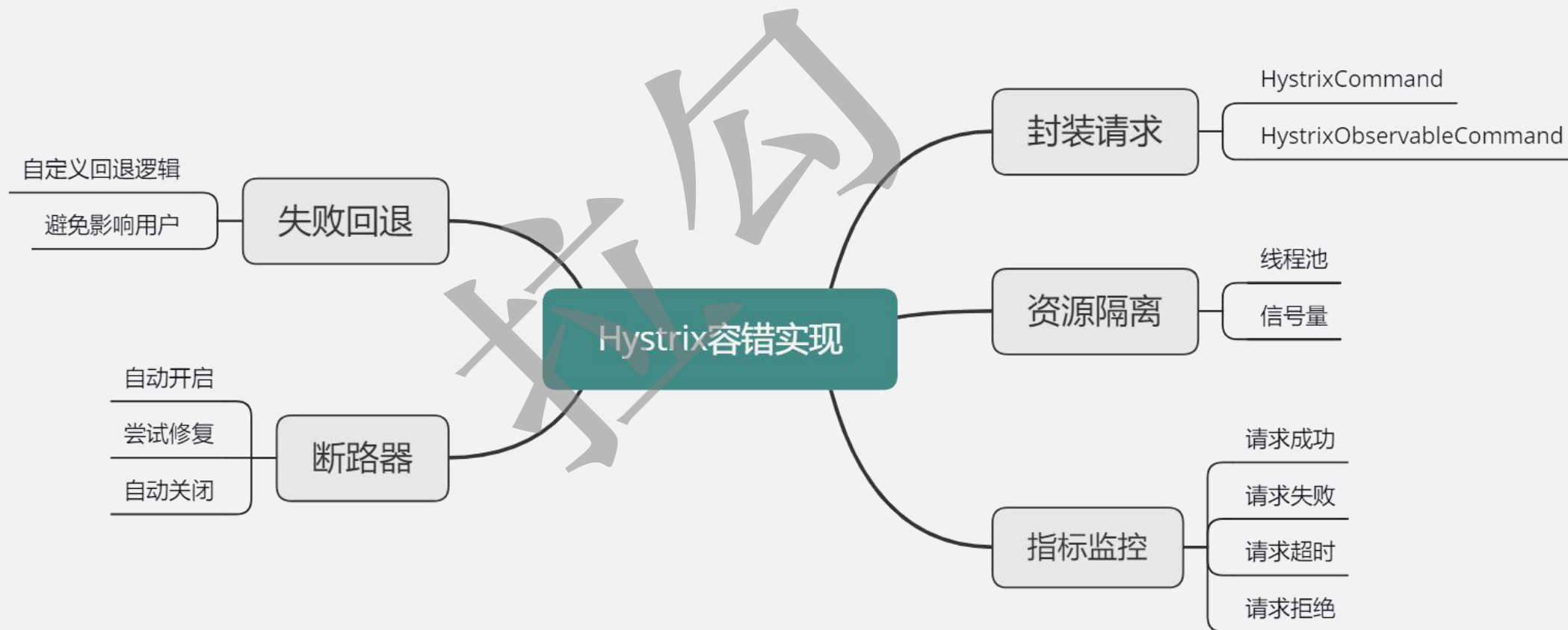
是Netflix针对微服务分布式系统的熔断保护中间件相当于电路中的保险丝

在微服务架构下，很多服务都相互依赖，如果不能对依赖的服务进行隔离，那么服务本身也有可能发生故障，Hystrix通过HystrixCommand对调用进行隔离，这样可以阻止故障的连锁反应，能够让快速失败并迅速恢复或者回退并优雅降级

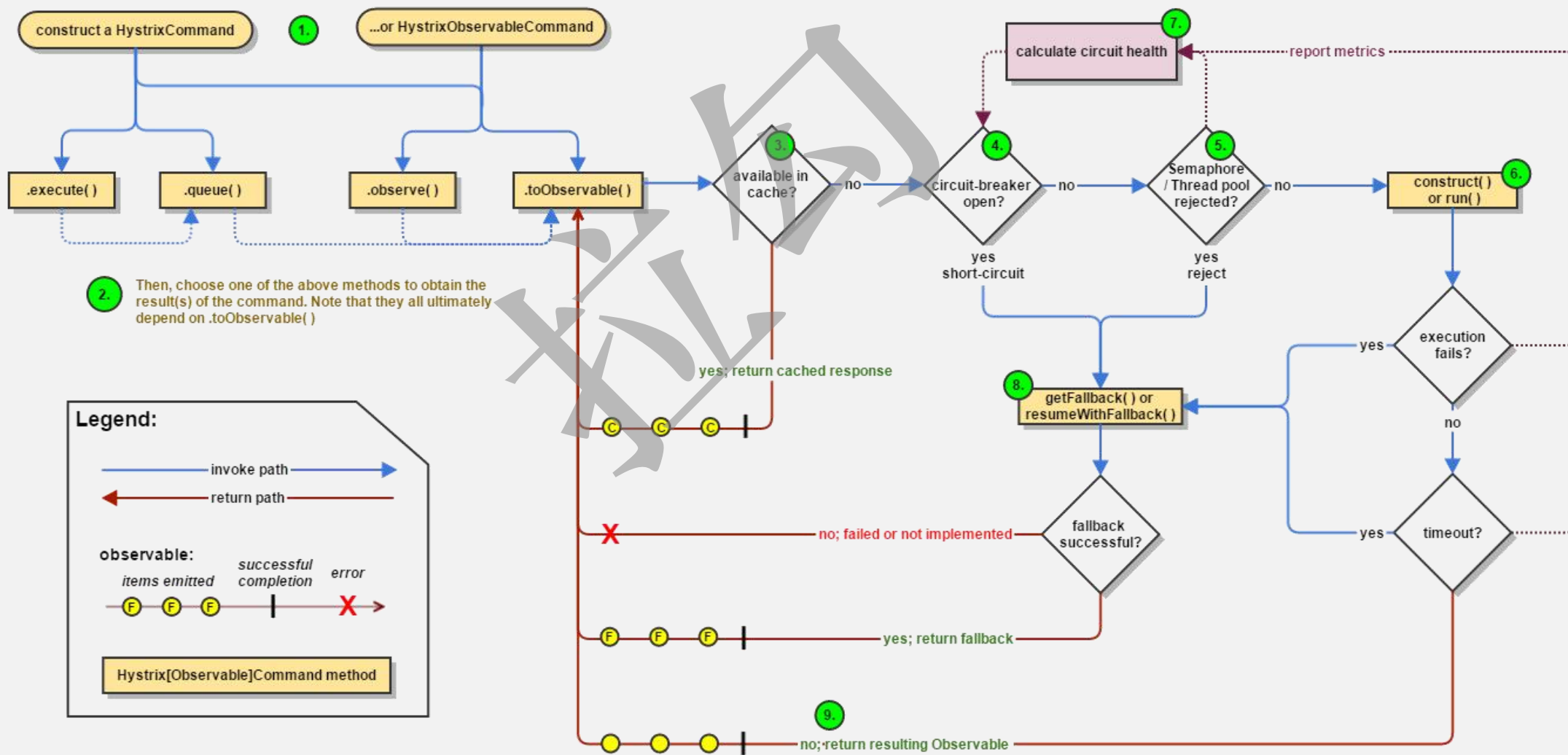




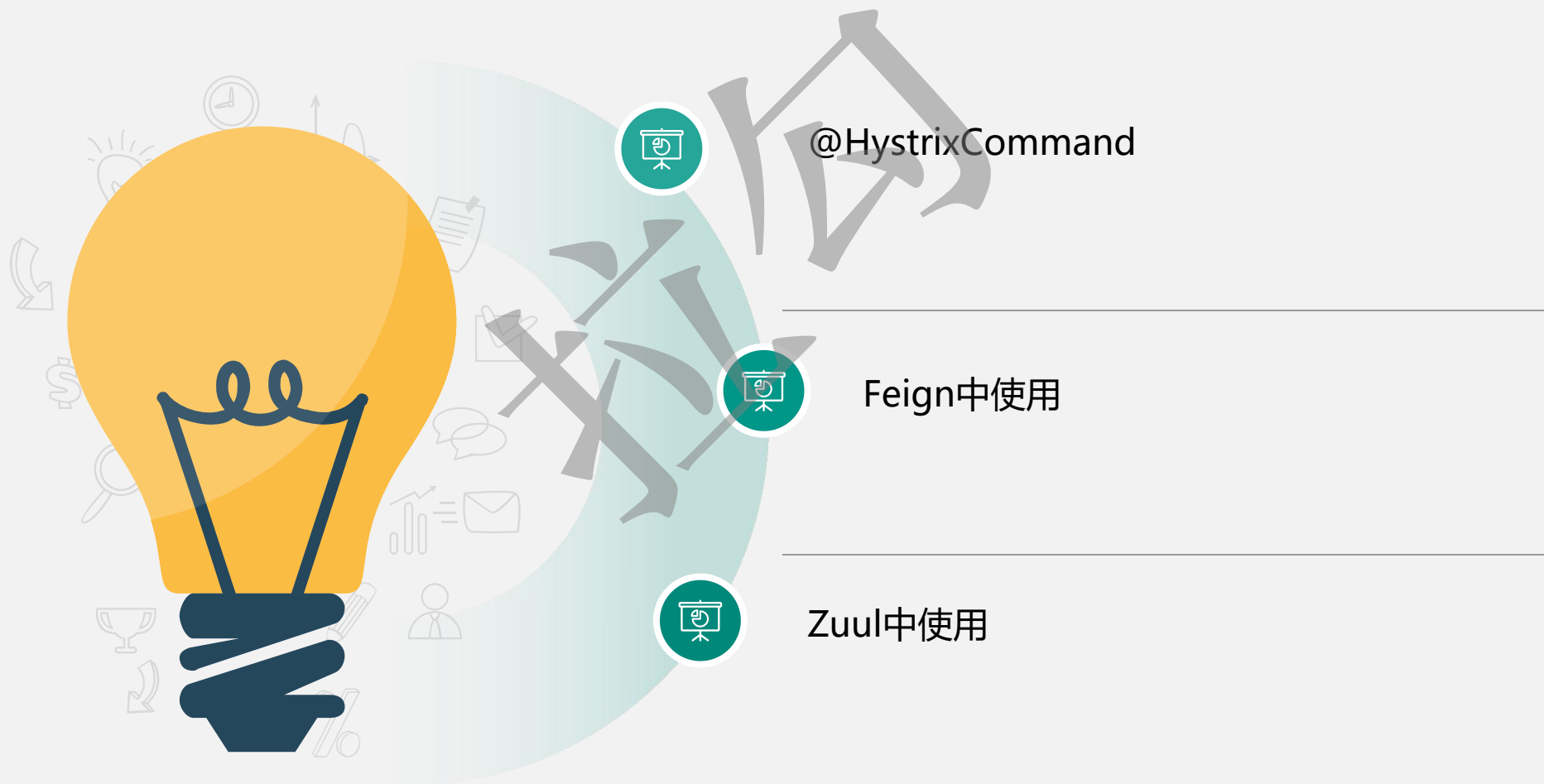
Hystrix容错实现



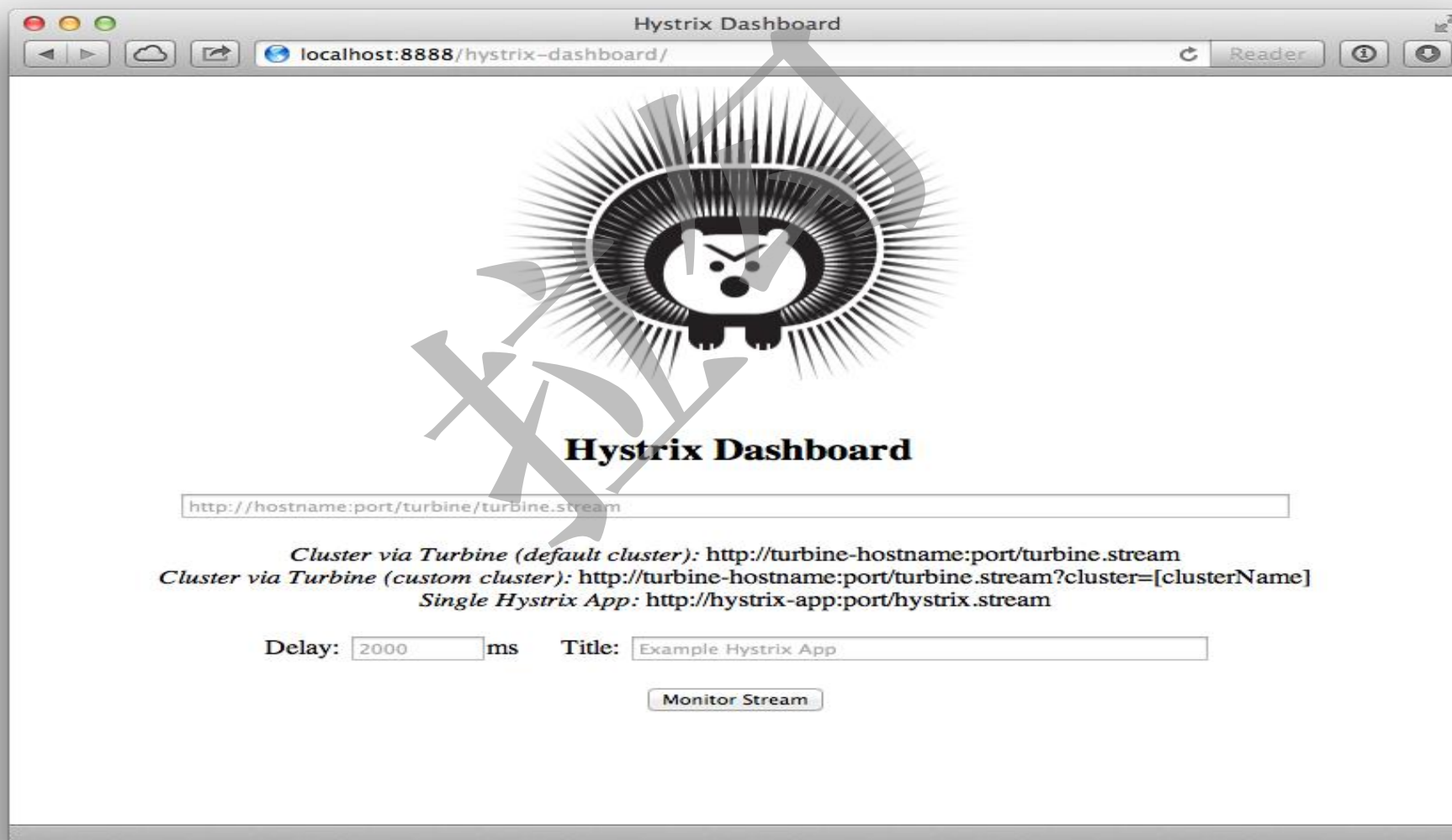
Hystrix工作原理

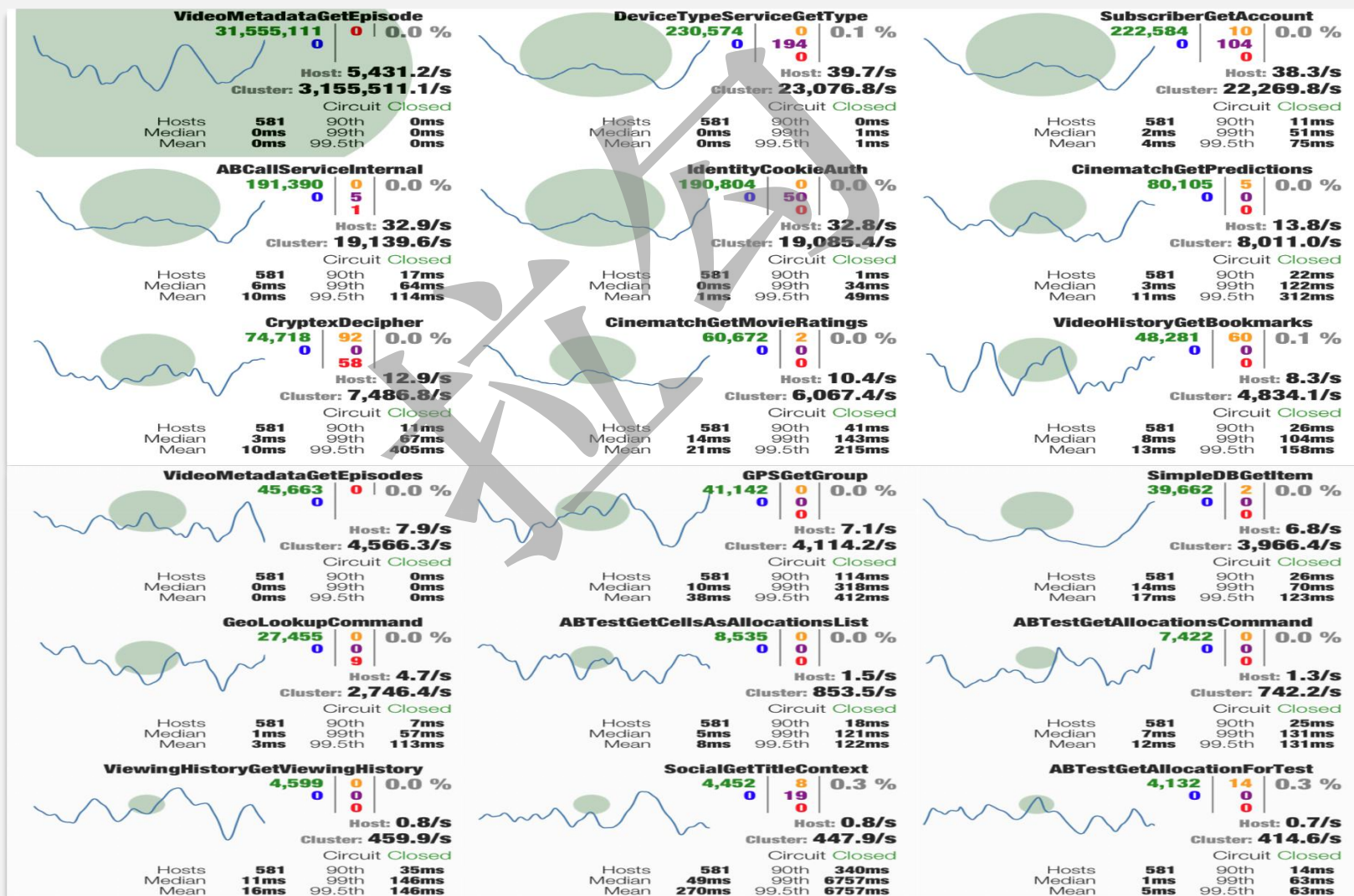


Hystrix使用



配置	作用
execution.isolation.strategy	隔离策略，可选Thread Semaphore
execution.isolation.thread.timeoutInMilliseconds	命令执行超时时间，默认1000ms
execution.isolation.semaphore.maxConcurrentRequests	最大并发请求数，默认10，信号量隔离有效
circuitBreaker.forceOpen	强制打开熔断器，如果打开这个开关，那么拒绝所有request，默认false
circuitBreaker.forceClosed	强制关闭熔断器 如果这个开关打开，circuit将一直关闭且忽略
coreSize	线程隔离下，并发执行的最大线程数，默认10
maxQueueSize	队列的大小，当设为-1，会使用SynchronousQueue，值为正时使用LinkedBlockingQueue。
queueSizeRejectionThreshold	达到queueSizeRejectionThreshold该值后，请求也会被拒绝





HystrixCommandAspect会切入增加了@HystrixCommand注解的方法
将其包装为HystrixCommand执行



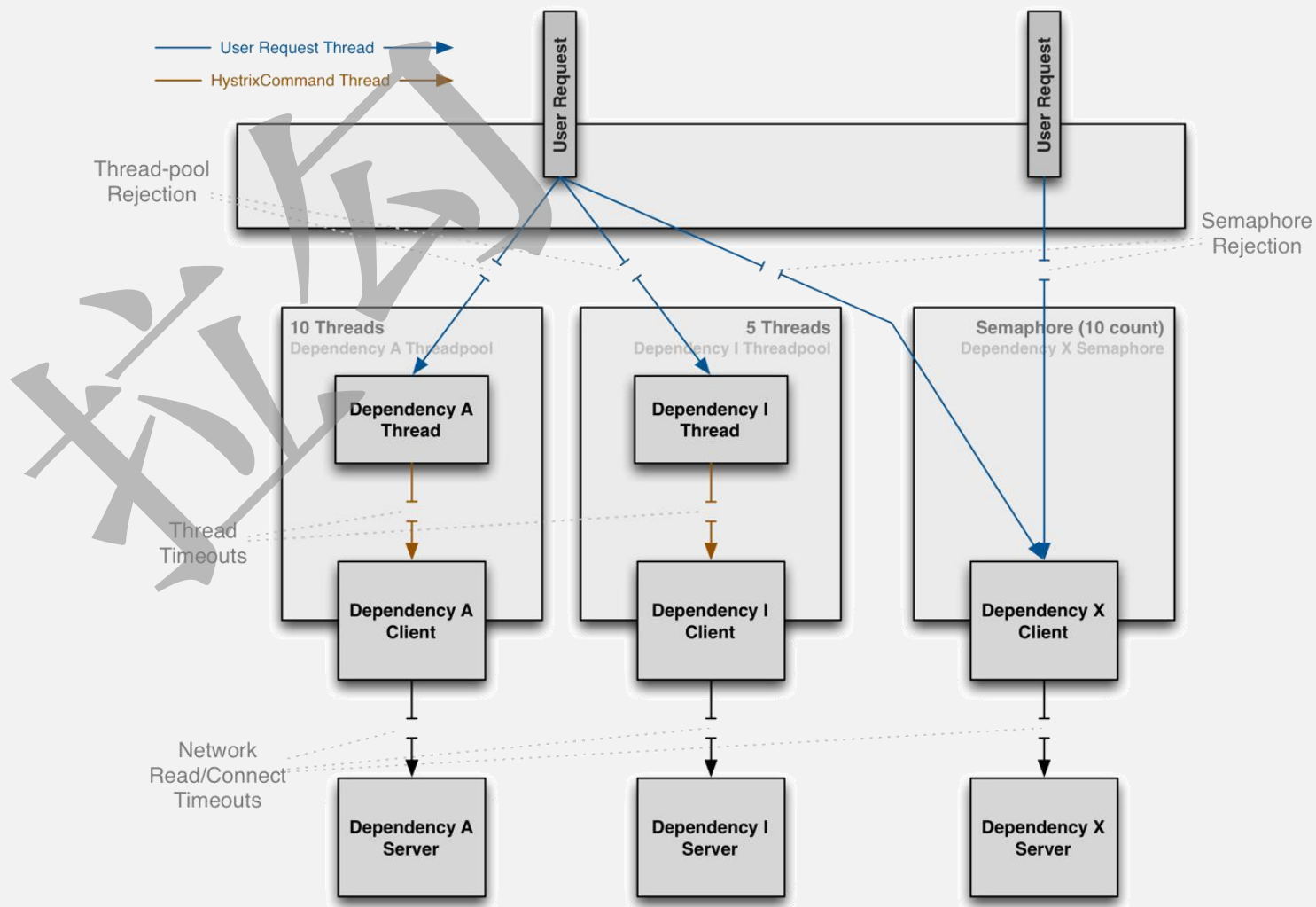
Hystrix隔离方式

线程池隔离

Command运行在单独线程中
支持超时

信号量隔离

Command是运行在调用线程中
不支持超时





- 配置可以对接配置中心进行动态调整
- 回退逻辑中可以手动埋点或者输出日志进行告警
- 用了线程池隔离模式再用ThreadLocal会有坑
- 网关中尽量用信号量隔离
- 插件机制可以实现很多扩展
- Hystrix各种超时配置方式
- commandKey, groupKey, threadPoolKey的使用

Next: 课时5《声明式服务调用-Feign》