

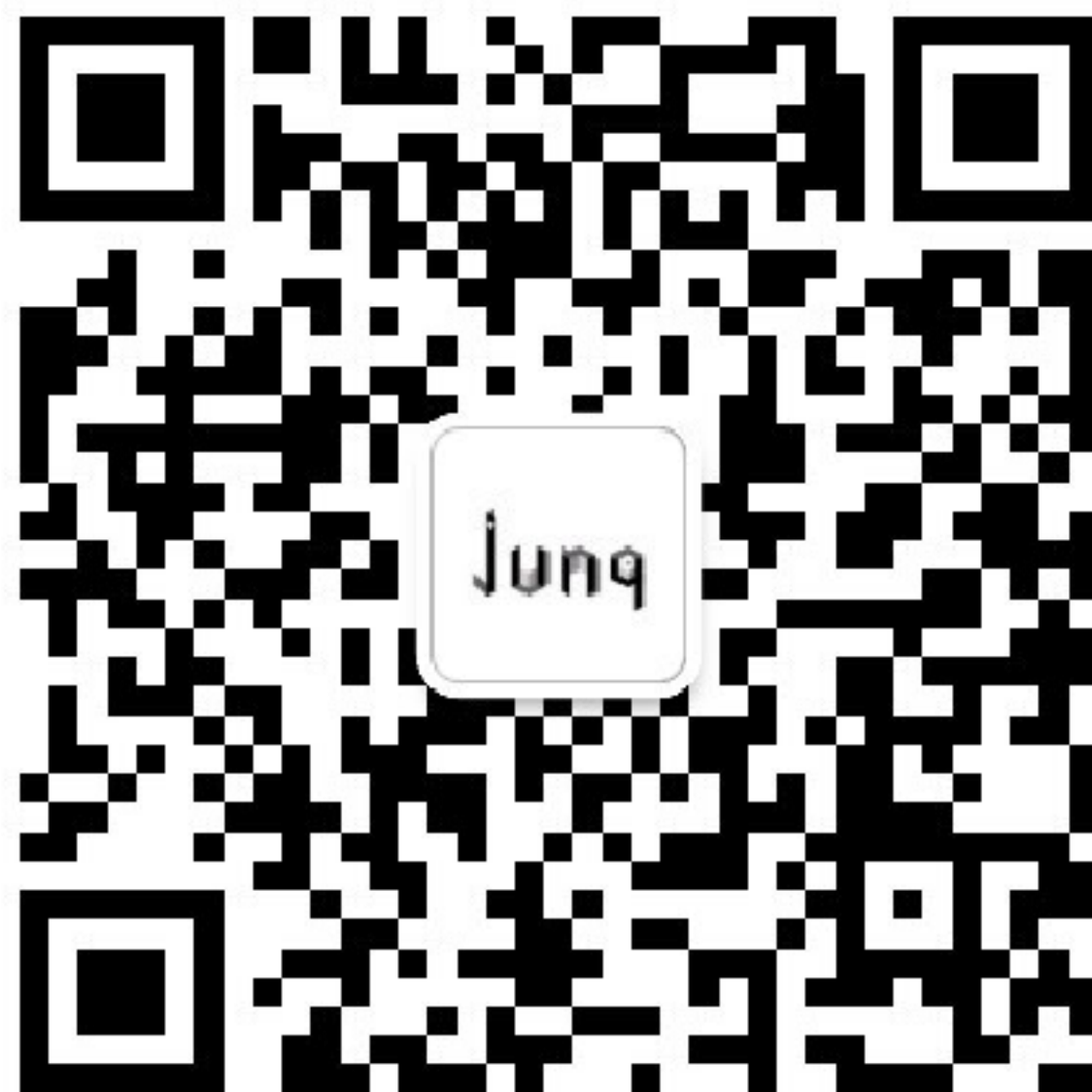


跟我做一个Java微服务项目

Week #8 微服务的生产环境实践 / 刘俊强



欢迎关注StuQ微信公众号



欢迎关注我的微信公众号 😊

大纲

- 微服务配置管理
- 弹性容错的微服务

微服务配置管理

常见配置方式

- 环境变量
- 配置文件
- 配置系统

环境变量

启动脚本 sample-start.sh:

```
#!/bin/sh
export SAMPLE_VARIABLE1="value 1"
export SAMPLE_VARIABLE2="value 2"

# 选项1 可执行jar
java -jar sample-service.jar

# 选项2 Servlet容器--Tomcat服务器
sh ${CATALINA_HOME}/bin/startup.sh
```

环境变量

Java读取环境变量：

```
Map<String, String> envMap = System.getenv();  
for (String envVariable : envMap.keySet()) {  
    System.out.format("%s=%s\n", envVariable,  
                      envMap.get(envVariable));  
}
```

强烈不建议使用!!!

配置文件-Spring Boot应用配置

- bootstrap.(yml|properties)
 - 在application.yml之前加载
 - 可理解为系统级配置，不变配置放置在此，如
`spring.application.name`
 - 使用Spring Cloud Config时，
`spring.application.name`和
`spring.cloud.config.server.git.uri`在此配置
- application.(yml|properties)
 - 可理解为应用级别、可动态替换
 - 发行包内、外部文件、Spring Cloud Config均可

配置文件-Spring Boot: bootstrap.yml

```
spring:
  application:
    name: cart-service

management:
  port: 9904
  context-path: /manage
  security:
    enabled: false

server:
  port: 9905

logging:
  level:
    root: INFO
    io.junq: DEBUG
```

配置文件-Spring Boot: application.yml

```
spring:
  application:
    instance_id: 001
  datasource:
    url: jdbc:mysql://cart-db/email?autoReconnect=true&use
    username: email
    password: emailPwd
    driver-class-name: com.mysql.jdbc.Driver
    schema: file:/email/data/schema-mysql.sql
    initialize: true
    continue-on-error: false
```

配置文件-Spring Boot应用小技巧

命令行指定配置属性：

```
java -jar sample-service.jar \  
--spring.application.name=sample-service
```

命令行指定配置文件：

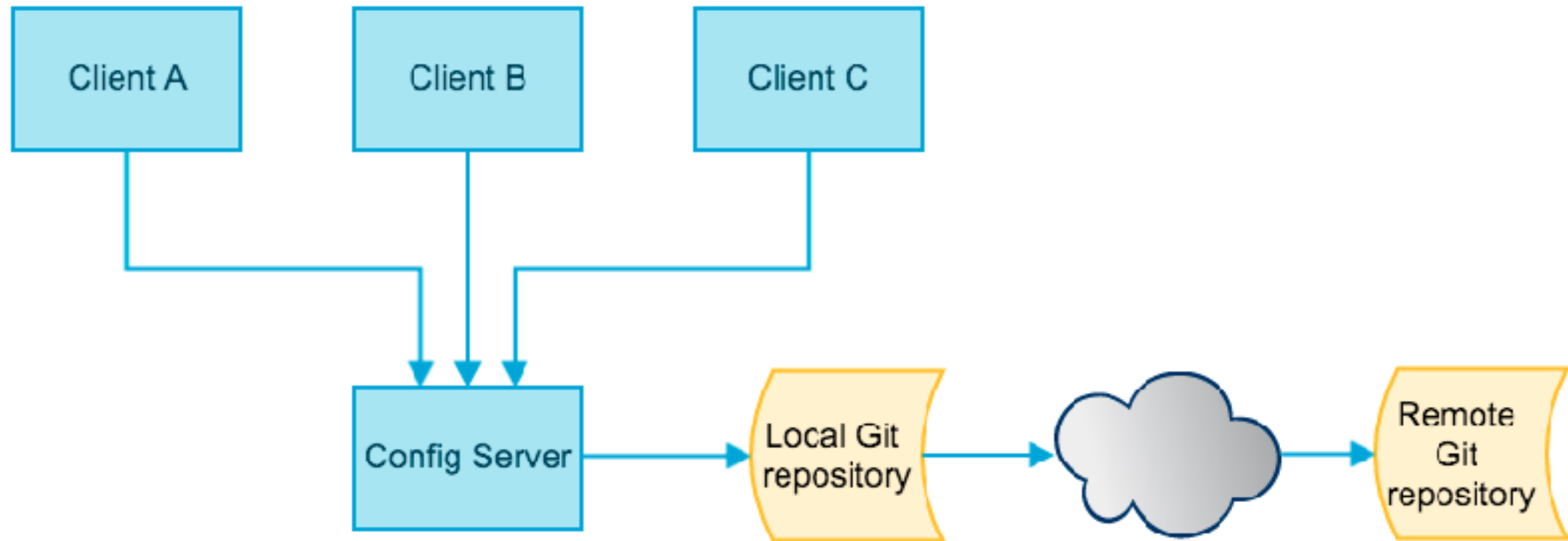
```
java -jar sample-service.jar \  
--spring.config.location=classpath:/override.properties
```

配置系统

常用配置系统

- Zookeeper
- etcd
- Consul
- Netflix archaius
- Spring Cloud Config

Spring Cloud Config



Config Server

资源访问格式:

- `/{{application}}/{{profile}}[/{{label}}]`
- `/{{application}}-{{profile}}.yml`
- `/{{label}}/{{application}}-{{profile}}.yml`
- `/{{application}}-{{profile}}.properties`
- `/{{label}}/{{application}}-{{profile}}.properties`

Config Server

资源访问参数:

- **application:** 应用名称, 一般对应应用的 `spring.application.name` 属性, 如 `order-service`
- **profile:** 环境, 一般对应应用的 `spring.profiles.active`, 如 `sandboxing`
- **label:** 版本或标签, 一般对应git仓库的 `commit id`、分支名称或 `tag`, 如默认为 `master`

Demo: Spring Cloud Config Server

Config Server Security: HTTP Basic Auth

pom.xml:

```
<dependency>  
  <groupId>org.springframework.boot</groupId>  
  <artifactId>spring-boot-starter-security</artifactId>  
</dependency>
```

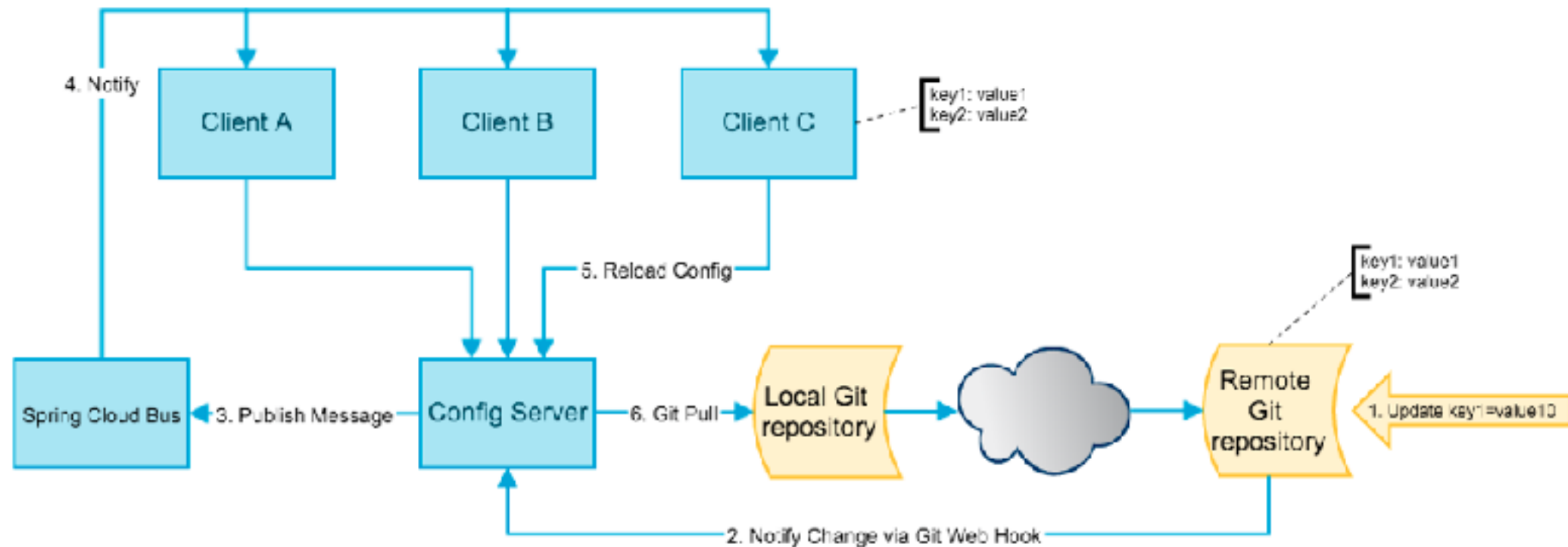
application.yml

```
security:  
  user:  
    name: root  
    password: c0nfIgS3cr3t
```

Config Server 加密

- 对称加密
- 非对称加密

Spring Cloud Config with Change Notification



弹性容错的微服务

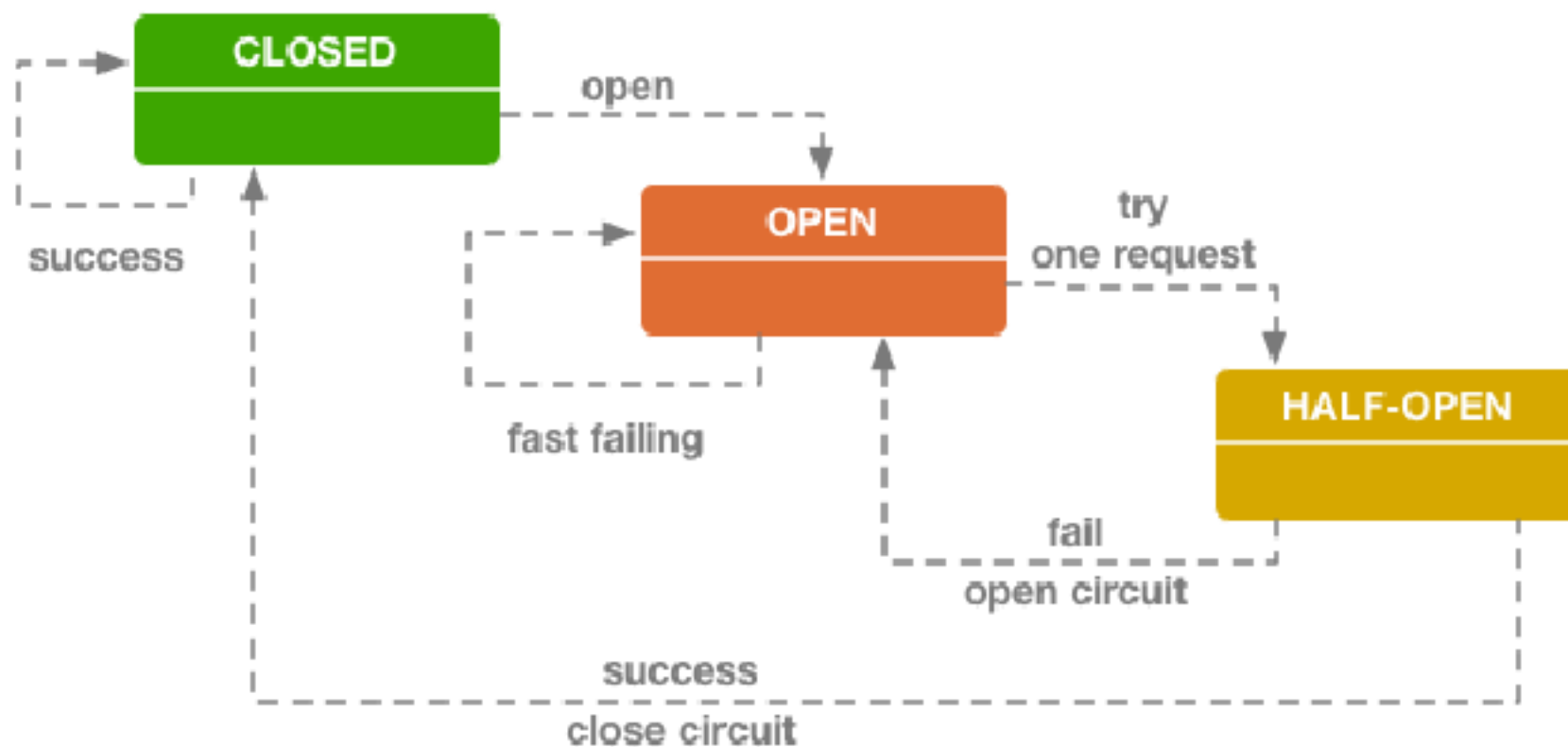
如何保证系统具备弹性？

- 快速超时
- 可降级
- 熔断器、断路器
- 壁仓
- 故障隔离

断路器模式

- 闭路/关闭状态：服务运行正常
- 开路/开启状态：服务运行故障
- 半开路/半开状态：检测服务状态

断路器模式示意图

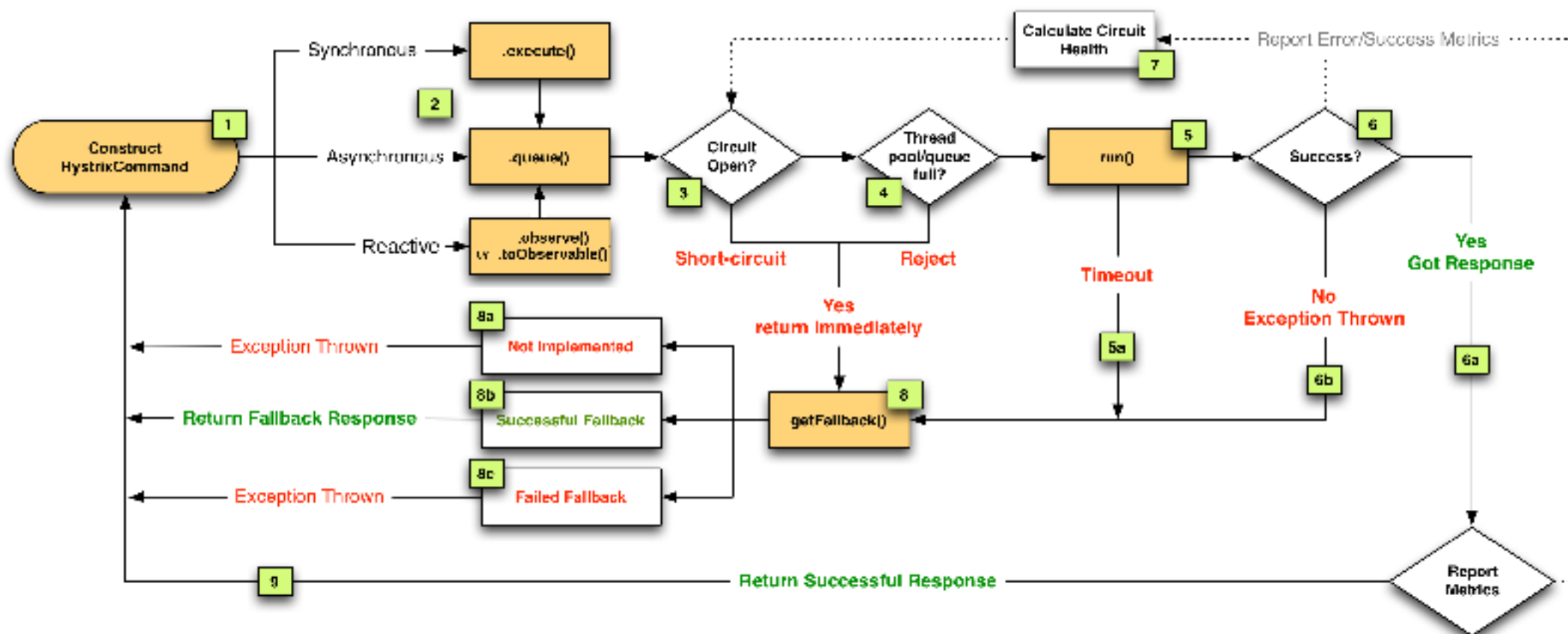


Circuit Breaker State Diagram

Hystrix

- 基于Java实现，Netflix贡献的开源项目
- 控制被依赖服务的延时和失败
- 防止在复杂系统中的级联失败
- 快速失败且快速恢复
- 可实现优雅降级
- 具有仪表盘，监控友好

Hystrix工作流程图

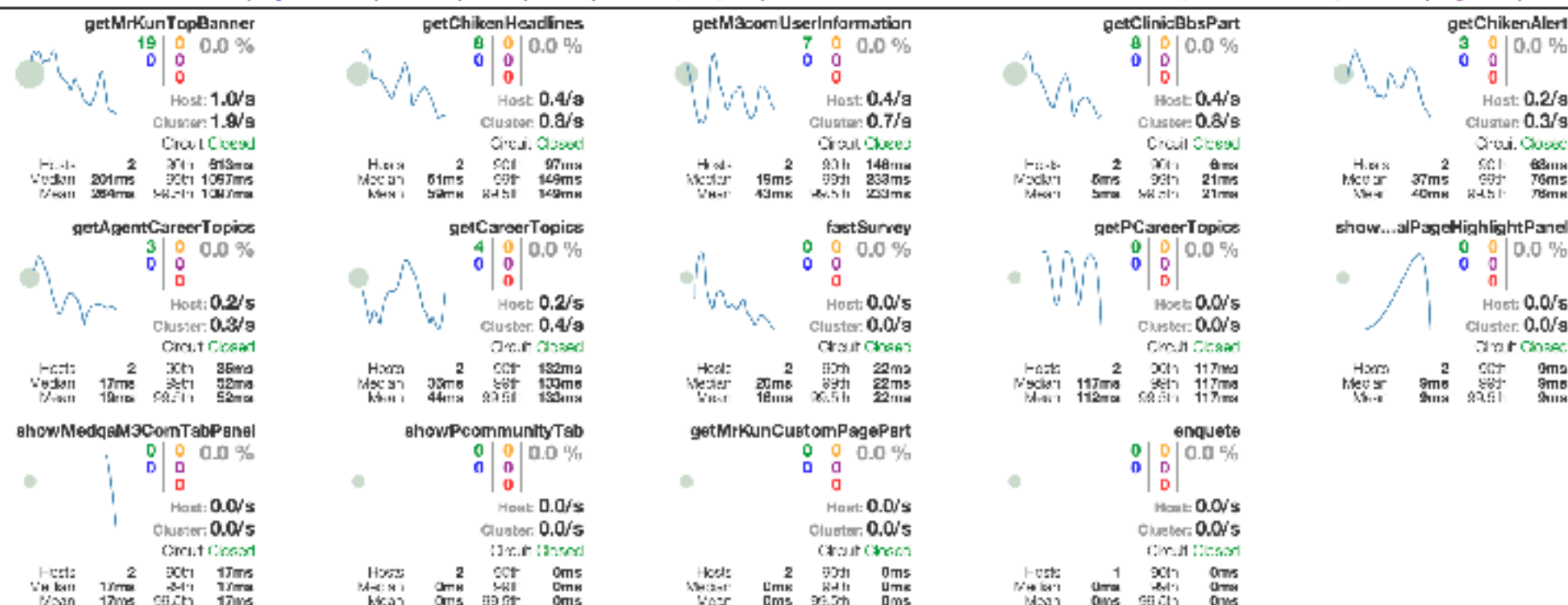


Hystrix Dashboard

Hystrix Stream: Octoparts



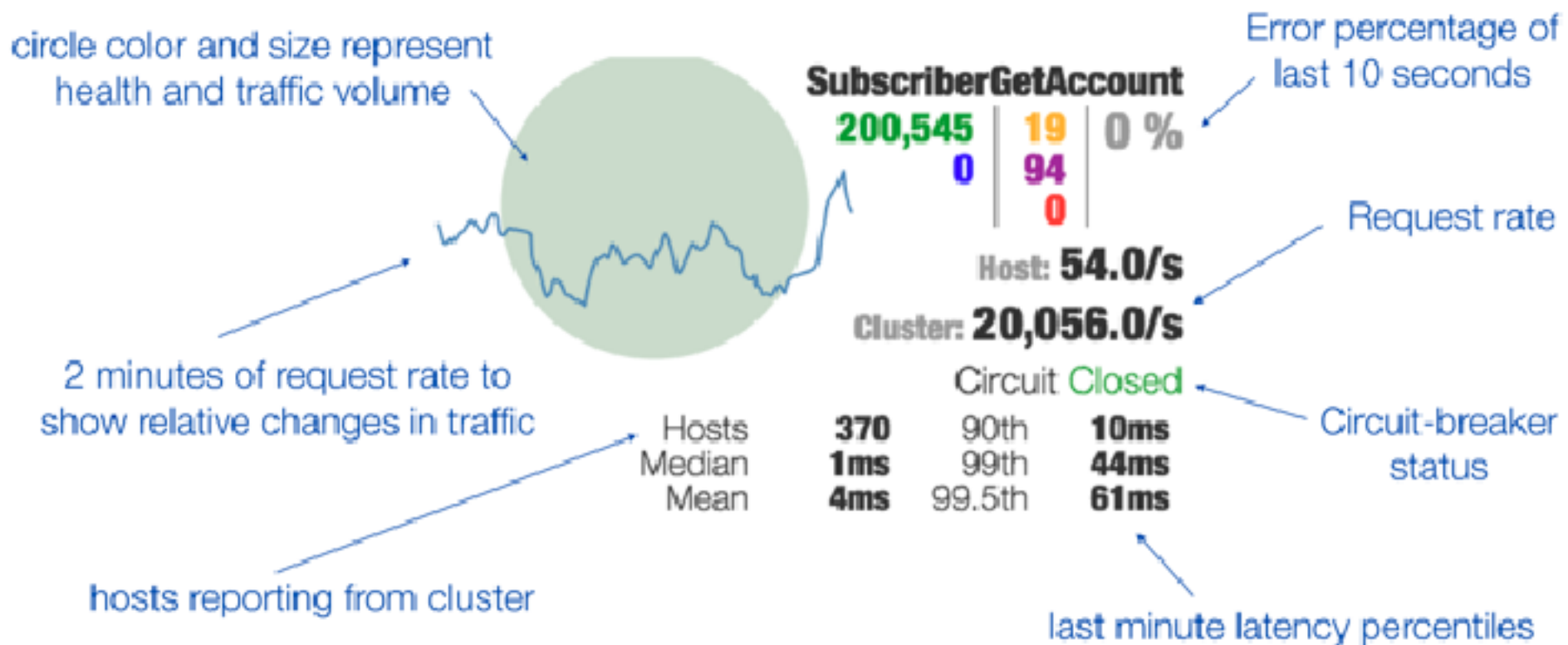
Circuit Sort: [Error](#) | [Error](#) | [Volume](#) | [Alphabetical](#) | [Volume](#) | [Error](#) | [Mean](#) | [Median](#) | 90 | 99 | 99.5 Success | Short-Circuited | Timeout | Rejected | Failure | Error %



Thread Pools



Hystrix Dashboard Detail



Demo: Hystrix



THANKS!



— 扫码了解更多 —