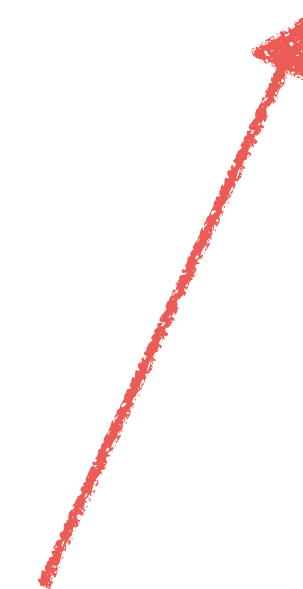
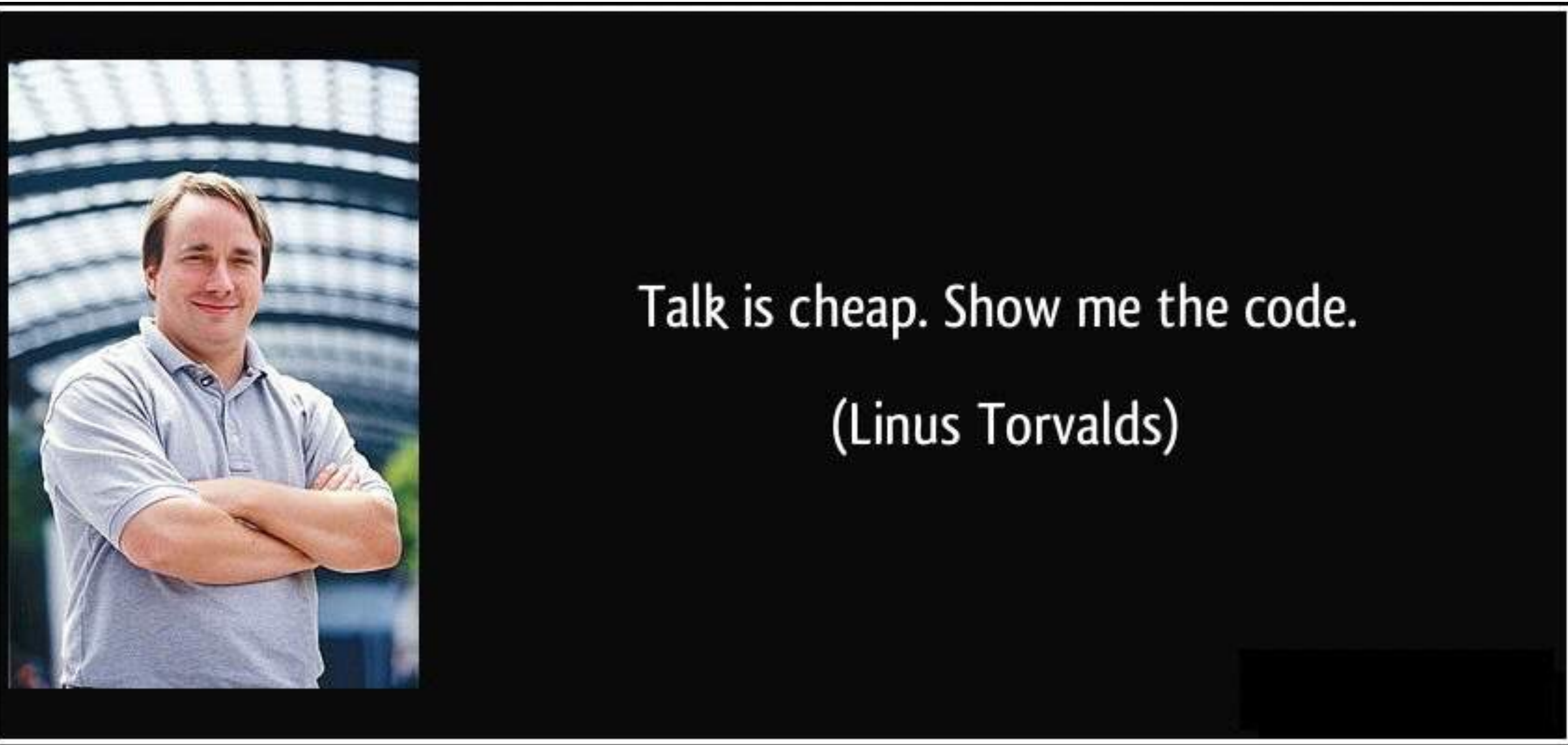


OAuth 2.0的授权模式：

- 授权码模式
- 简化模式
- 密码模式
- 客户端模式



假设是内部服务，用客户端模式举例吧



现场演示时间

e.g. OAuth 2.0认证服务端

声明提供OAuth 2.0认证服务

```
@Configuration
@EnableAuthorizationServer
public class OAuthConfiguration extends AuthorizationServerConfigurerAdapter {

    @Override
    public void configure(AuthorizationServerSecurityConfigurer securityConfigurer)
        throws Exception {
        securityConfigurer.checkTokenAccess("isAuthenticated()");
    }

    @Override
    public void configure(ClientDetailsServiceConfigurer clients) throws Exception {
        clients.inMemory()
            .withClient("digitalsonic").secret("password")
            .scopes("vip")
            .authorizedGrantTypes("client_credentials");
    }
}
```

仅支持客户端模式

e.g. 服务端与客户端

声明这是资源服务器

```
@SpringBootApplication
@EnableDiscoveryClient
@EnableResourceServer
public class SpringCloudDemoServerApplication {
    public static void main(String[] args) {
        SpringApplication.run(SpringCloudDemoServerApplication.class, args);
    }
}
```

资源服务器&客户端相关配置

```
@EnableOAuth2Client
@RestController
public class SpringCloudDemoSimpleClientApplication {
    @Autowired
    private RestTemplate restTemplate;
}
```

```
security.oauth2.resource.userInfoUri=http://localhost:8888/user
security.oauth2.resource.tokenInfoUri=http://localhost:8888/oauth/check_token

security.oauth2.client.clientId=digitalsonic
security.oauth2.client.clientSecret=password
security.oauth2.client.authorizedGrantTypes=client_credentials
security.oauth2.client.scope=vip
security.oauth2.client.accessTokenUri=http://localhost:8888/oauth/token
```

声明OAuth 2.0客户端
配置对应RestTemplate

```
@Bean
public static OAuth2RestTemplate oauth2RestTemplate(ClientCredentialsResourceDetails details,
                                                    OAuth2ClientContext oauth2ClientContext) {
    return new OAuth2RestTemplate(details, oauth2ClientContext);
}
```


微服务的防护

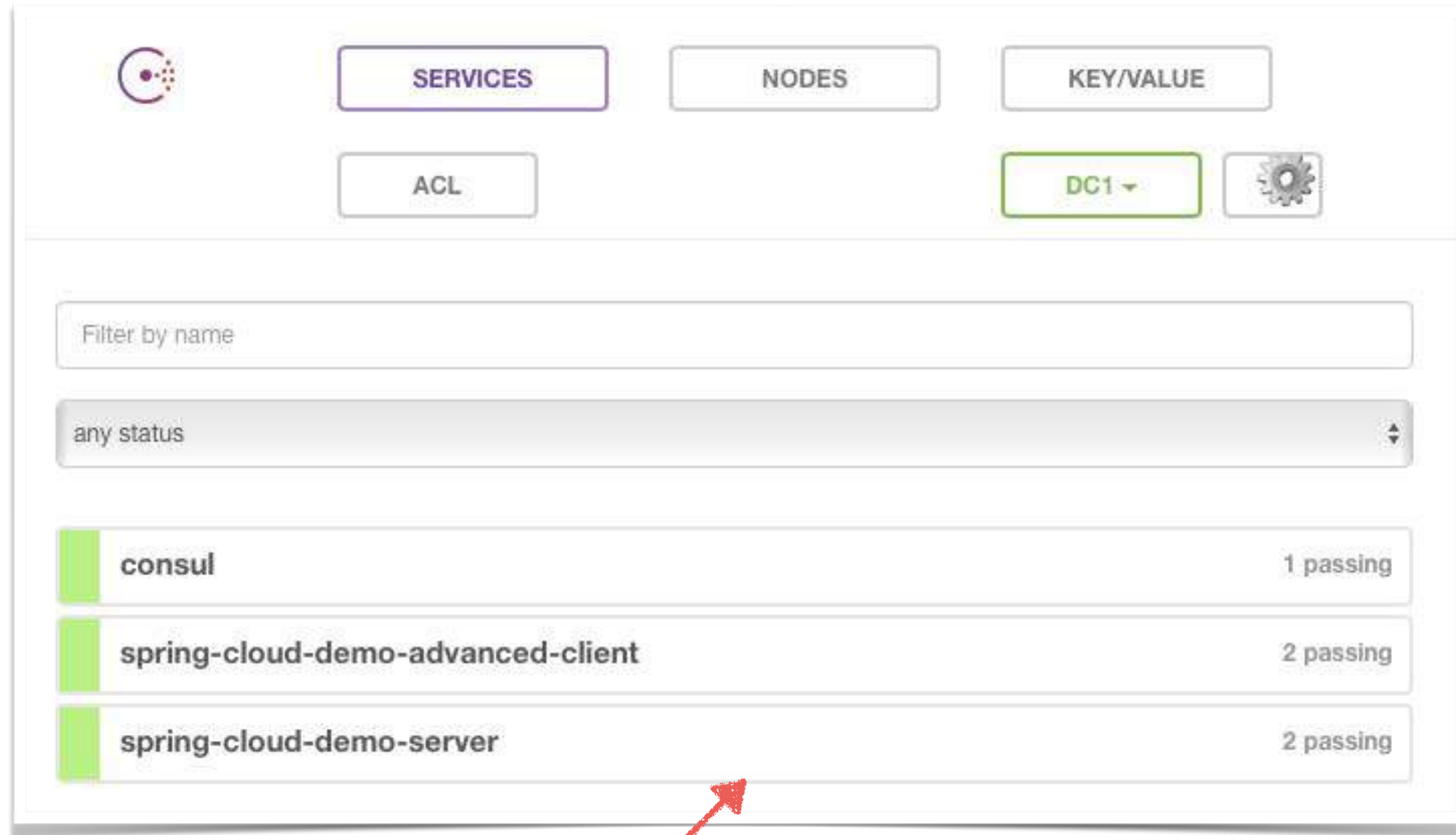
- 你可能还会想到
 - 安全框架——Spring Security、Apache Shiro
 - 签名&摘要工具——Apache Commons Codec
 - 加解密工具——the Legion of the Bouncy Castle

服务可视化

- 为什么要把服务可视化？
- 把哪些内容可视化？
- 有些什么方便的工具么？



e.g. 服务的发现与注册

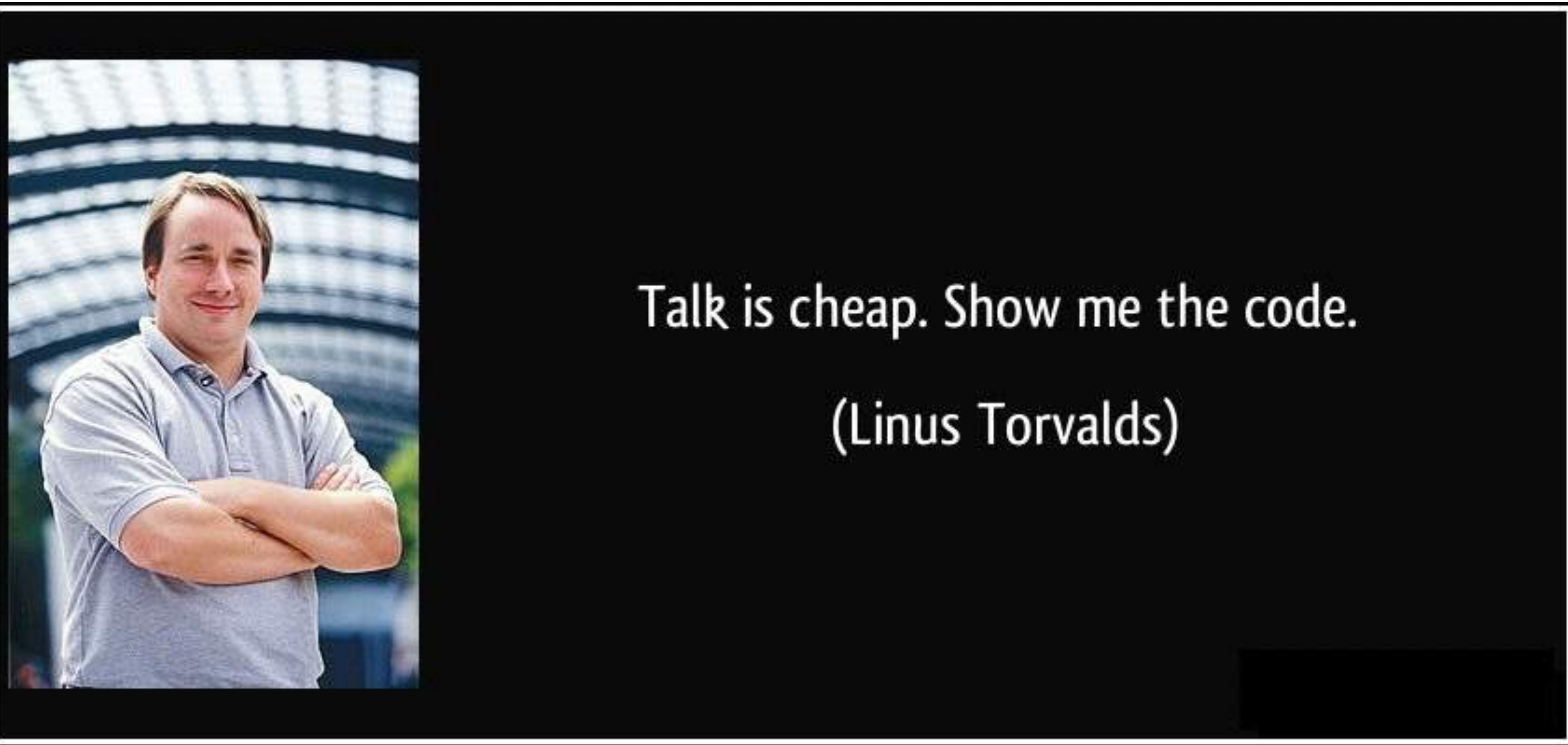


类似的东西还有：

- Eureka
- Dubbo

Consul Dashboard

虽然不是Java写的，但配合Spring Cloud Consul还挺好用
我总得演示些默认之外的东西吧:-)



现场演示时间

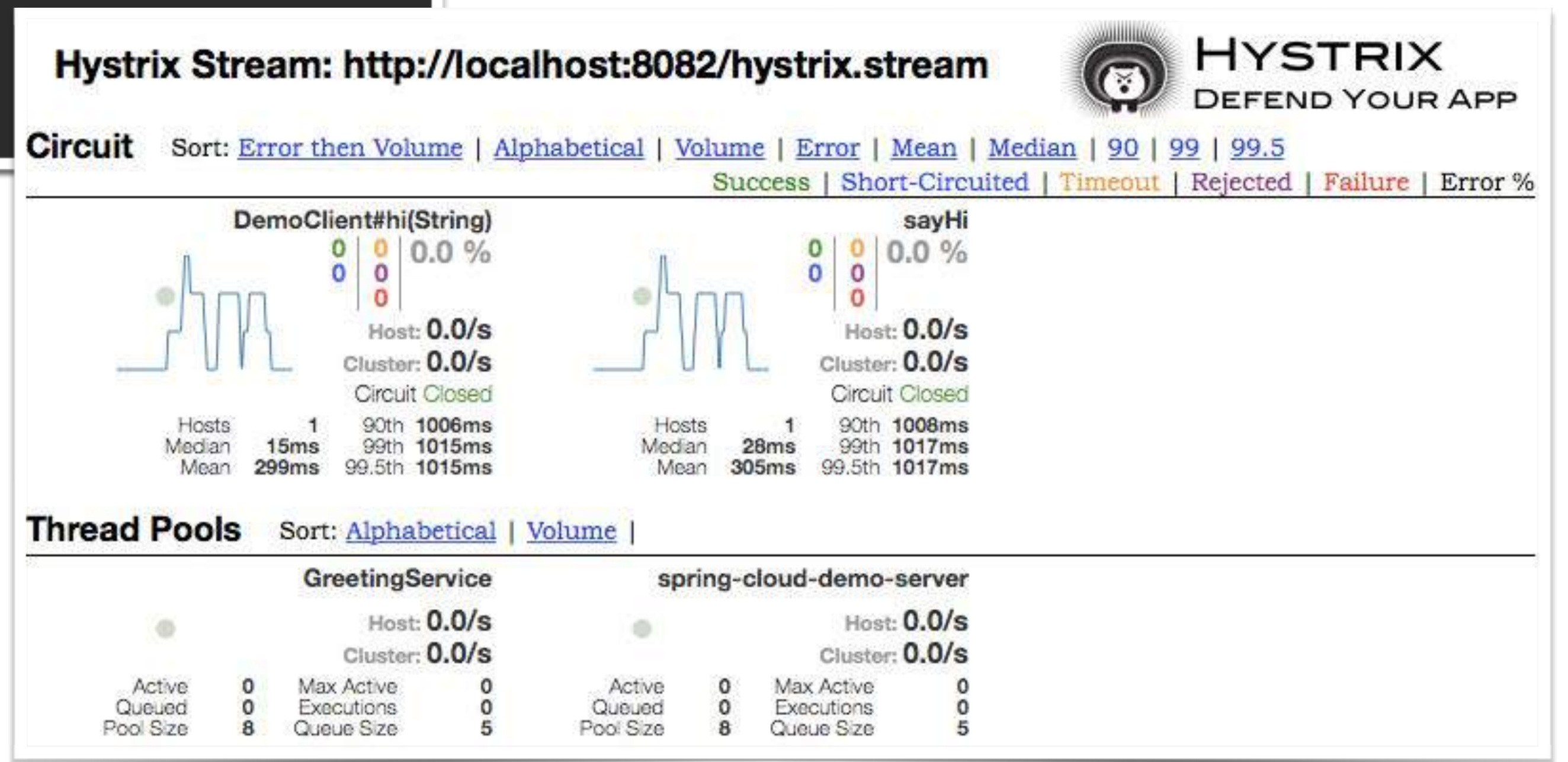
e.g. 服务的运行情况

```
@SpringBootApplication
@EnableDiscoveryClient
@RestController
@EnableFeignClients
@EnableCircuitBreaker
@EnableHystrixDashboard
public class SpringCloudDemoAdvancedClientApplication {
    @Autowired
    private GreetingService greetingService;

    public static void main(String[] args) {
        SpringApplication.run(SpringCloudDemoAdvancedClientApplication.class, args);
    }

    @RequestMapping("/")
    public String remoteHi() {
        return greetingService.sayHi();
    }
}
```

我要有个DashBoard



P.S.
CAT是个好东西!
这个例子为啥不用CAT呢?

e.g. 服务的链路分析

Zipkin Investigate system behavior Find a trace Dependencies Go to trace

spring-cloud-demo-simple-clie... all

Start time 12-04-2016 17:10 End time 12-11-2016 17:20

Duration (µs) >= Limit 10 Find Traces

Annotations Query (e.g. "finagle.timeout", "http.path=/foo/bar/ and cluster=foo and cache.miss")

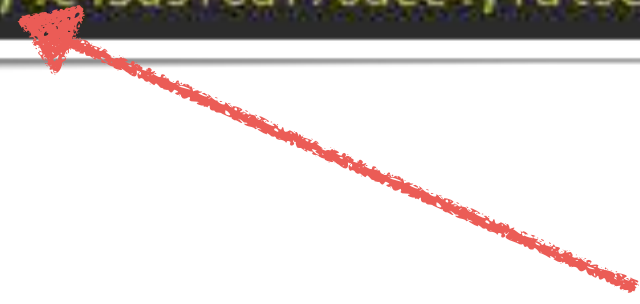
Showing: 2 of 2 Services: spring-cloud-demo-simple-client-c5114714da8e3ad8190ee2adac654197 Sort: Longest First

2.033s 1 spans
spring-cloud-demo-simple-client-c5114714da8e3ad8190ee2adac654197 100%
spring-cloud-demo-simple-client-c5114714da8e3ad8190ee2adac654197 x1 2032ms 8 minutes

24.950ms 1 spans
spring-cloud-demo-simple-client-c5114714da8e3ad8190ee2adac654197 100%
spring-cloud-demo-simple-client-c5114714da8e3ad8190ee2adac654197 x1 24ms 9 minutes ago

```
<dependency>  
  <groupId>org.springframework.cloud</groupId>  
  <artifactId>spring-cloud-starter-sleuth</artifactId>  
</dependency>  
<dependency>  
  <groupId>org.springframework.cloud</groupId>  
  <artifactId>spring-cloud-sleuth-zipkin</artifactId>  
</dependency>
```

```
INFO [spring-cloud-demo-simple-client,3e500dfff70b444a,fc0eff57117283b9,false] 51921 --- [nio-8081-exec-7] s.SpringCloudDemoSimpleClientApplication : Going to call  
INFO [spring-cloud-demo-simple-client,4042579290fa7cc8,4c45d9f6af76acc4,false] 51921 --- [nio-8081-exec-8] s.SpringCloudDemoSimpleClientApplication : Going to call
```



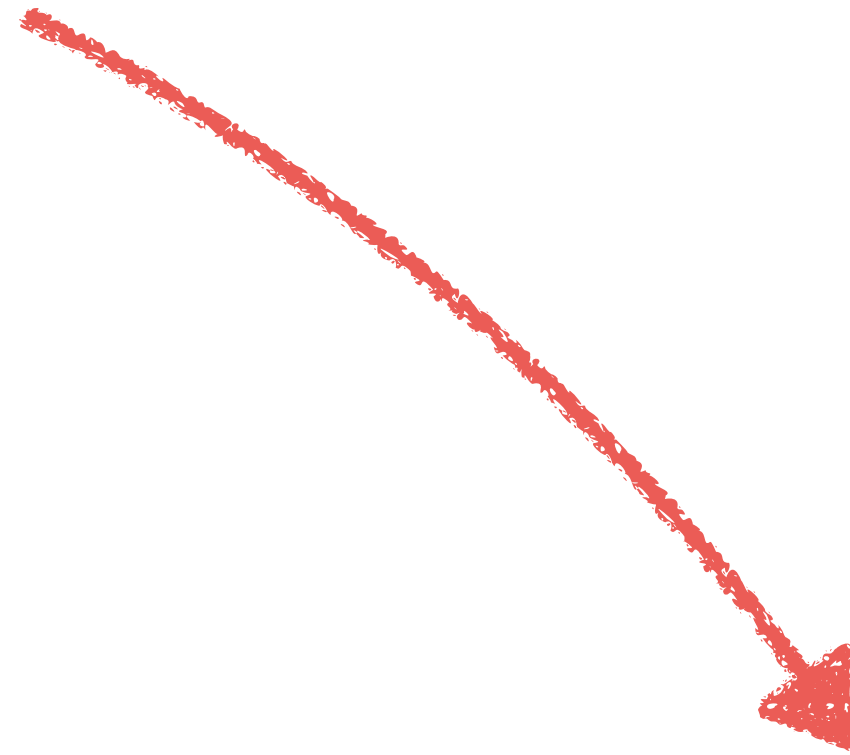
这些ID很重要

少些幻想

- 不要指望开源软件拿来就能用
- 整合的工作是难免的
- 总会出现定制的需要



如果.....你不记得今天我说了什么



- ① 对微服务而言，设计合理的业务模型很重要
- ② 微服务与语言没有太大的关系
- ③ 完整的生态环境，能帮助你快速地将微服务落地



GIAC | BEIJING
Dec.12.16-17

技术架构未来

架構
ARCHNOTES
高 可 用 架 构



Thank you!



微博: @DigitalSonic



邮箱: digitalsonic@sina.com