



基于ServiceComb的智能城市微 服务改造历程

软通动力智慧城市科技有限公司

鲍永伟

AGENDA

A circular graphic for the first agenda item. It features a yellow-to-green gradient background with a blurred image of a person in a suit. The number "01" is prominently displayed in white in the center.

01

智慧城市与微服务

A circular graphic for the second agenda item. It features a blue background with a blurred image of a man in a suit and glasses. The number "02" is prominently displayed in white in the center.

02

微服务ServiceComb改造

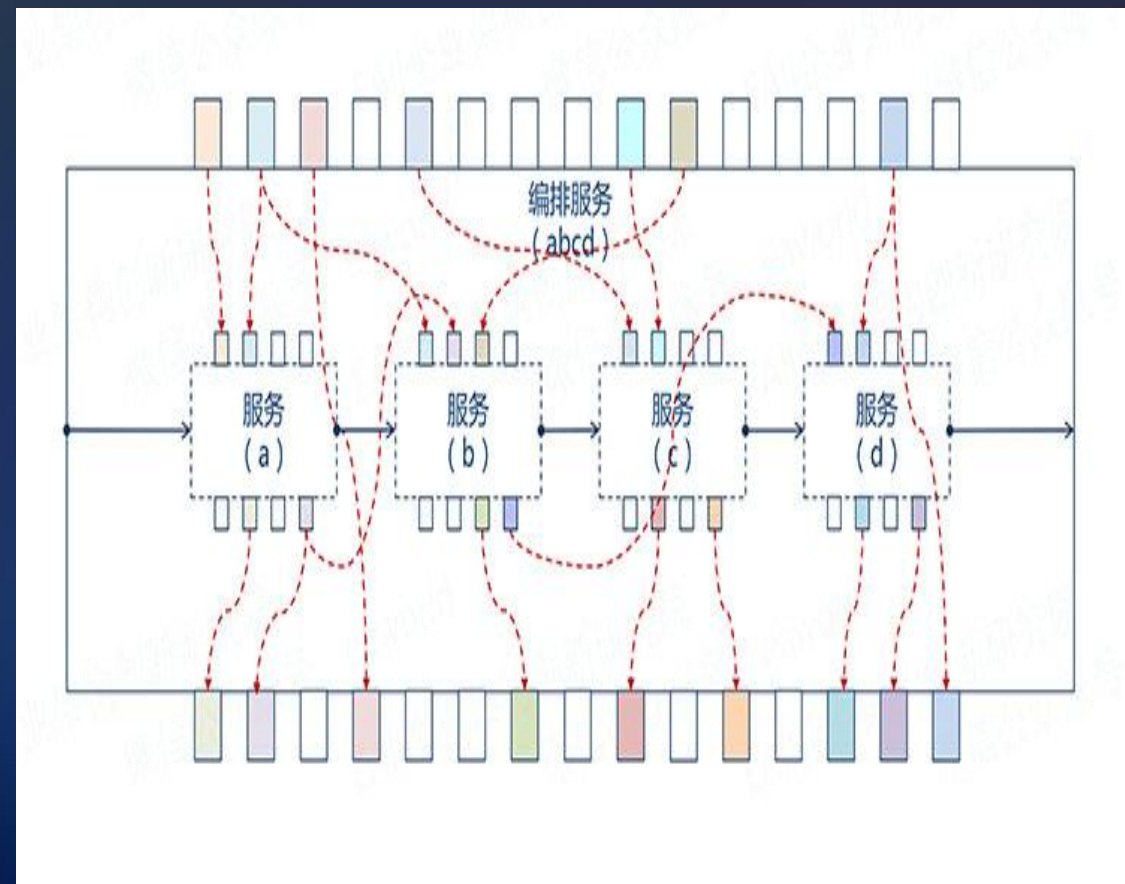
A circular graphic for the third agenda item. It features a blue background with a blurred image of server hardware. The number "03" is prominently displayed in white in the center.

03

基于ServiceComb智慧城市
微服务平台



智慧城市与微服务





ServiceComb在软通动力的应用

- 老系统基于Spring Cloud的微服务迁移
- 老系统基于PHP、.Net等应用API
- 新开发直接使用ServiceComb微服务

为什么要做迁移改造？

Website: <http://servicecomb.incubator.apache.org/>

Gitter: <https://gitter.im/ServiceCombUsers/Lobby>



从Spring服务改造到ServiceComb 5步

- 添加pom.xml依赖
- 添加microservice.yaml文件以及配置
- 启动类添加@EnableServiceComb注解
- REST调用，service层添加
@RestSchema(schemaId = "xxx")注解
- rpc调用，service层添加@RpcSchema(schemaId =
"XXX")注解



改造中涉及的一些关键问题

- `HttpServletRequest`是J2EE(Servlet)协议定义的对象。ServiceComb支持在Servlet协议上、HTTP协议以及其他协议上提供REST服务，因此不支持特定技术框架的对象。需要将接口定义修改为平台无关的原型，这方面我们花费很多时间调整

以下面接口为例：

```
@RequestMapping(value = "/auth", method = RequestMethod.POST)
public ResultResponse createAuthenticationToken(HttpServletRequest request,
    @RequestBody JwtAuthenticationRequest authenticationRequest) throws
AuthenticationException{
    String type = authenticationRequest.getType();
    String appCode = request.getHeader(BaseTypeConstants.HEADER_APP_CODE);
    String appType = request.getHeader(BaseTypeConstants.HEADER_APP_TYPE);
```

修改后：

```
@RequestMapping(value = "/auth", method = RequestMethod.POST)
public ResultResponse createAuthenticationToken(@RequestHeader(name=
BaseTypeConstants.HEADER_APP_CODE) String appCode,
    @RequestHeader(name= BaseTypeConstants.HEADER_APP_TYPE) String appType),
    @RequestBody JwtAuthenticationRequest authenticationRequest) throws
AuthenticationException{
    String type = authenticationRequest.getType();
```

Website: <http://servicecomb.incubator.apache.org/>

Gitter: <https://gitter.im/ServiceCombUsers/Lobby>



改造中涉及的一些关键问题

- ServiceComb提供了比Feign更加容易使用的客户端RPC方式，客户端不需要像Feign一样声明REST映射关系。我们在具体应用中比较发现ServiceComb的方式更方便

改造前：

客户端接口声明

```
@FeignClient("helloprovider")
```

```
@RequestMapping(path = "/hello")
```

```
public interface Hello {
```

```
    @RequestMapping(path = "/sayhi", method = RequestMethod.GET)
```

```
    String sayHi(@RequestParam(name = "name") String name);
```

```
}
```

客户端使用

```
@Autowired
```

```
Hello client;
```

改造后：

客户端接口声明

```
public interface Hello {
```

```
    String sayHi(String name);
```

```
}
```

客户端使用

```
@RpcReference(microserviceName="helloprovider", schemaId="hello")
```

```
Hello client;
```

对于习惯RPC编程的开发人员，可以直接在服务定义的时候，就声明接口，然后作为API发布。这样客户端就可以避免重复写代码了

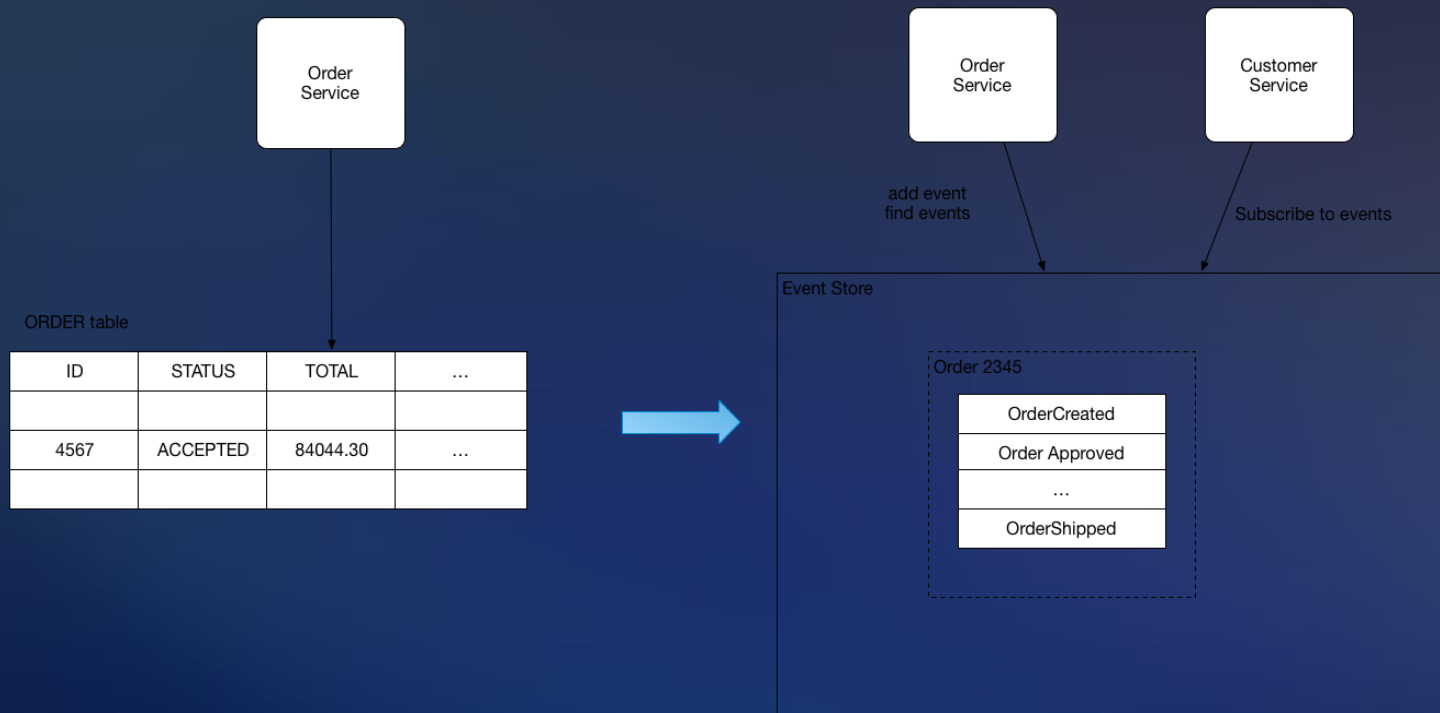
Website: <http://servicecomb.incubator.apache.org/>

Gitter: <https://gitter.im/ServiceCombUsers/Lobby>



改造中涉及的架构设计问题

- 微服务的粒度
- 数据库的同步
- PaaS与虚拟机模式
- SaaS化程序改造





基于ServiceComb智慧城市微服务平台



行业级应用PaaS平台，将物联网与视频做为应用服务的核心，提供微服务及容器使能技术服务、大数据环境服务、IOT环境技术服务、视频云技术服务、API网关路由服务及通用技术支撑服务，底端融合Devops、开发云、合作伙伴分账等第三方功能接口

Website: <http://servicecomb.incubator.apache.org/>

Gitter: <https://gitter.im/ServiceCombUsers/Lobby>



平台部分页面



API产业链
旅游 教育 环保

王者荣耀



获取开发商列表

推荐API



官方网站

超级管理员

仪表盘 用户管理 导航管理 编辑is_menu 行业管理 在线人员 模块管理 组织管理 系统管理

is_menu管理 查询 新增

	id	menu_name	pid	menu_t	menu_u	status	前端样式	remark	前端路由	前端调用方法	参数	create_time	ba
1	1b6c...	调用线路图	9a55...			正常						2018...	
2	24ed...	测试查看	9a55...			正常						2018...	
3	278e...	性能APM	9a55...			正常						2018...	
4	31e6...	more	345b...			正常						2018...	
5	3f69...	API开发测试	0			正常	part3					2018...	
6	5db1...	测试原始记录	9a55...			正常						2018...	
7	791b...	应用编排类	345b...			正常						2018...	

发测试

原生测试

测试管理



获取开发商列详细...





Thank You.



Website: <http://servicecomb.incubator.apache.org/>
Gitter: <https://gitter.im/ServiceCombUsers/Lobby>