

ServiceComb设计揭秘：标准与开发

华为CloudBU PaaS微服务引擎开发团队

代码阅读

- 一、系统架构解读
- 二、系统模块解读
- 三、框架的启动与停止
- 四、框架运行时请求发送与处理过程

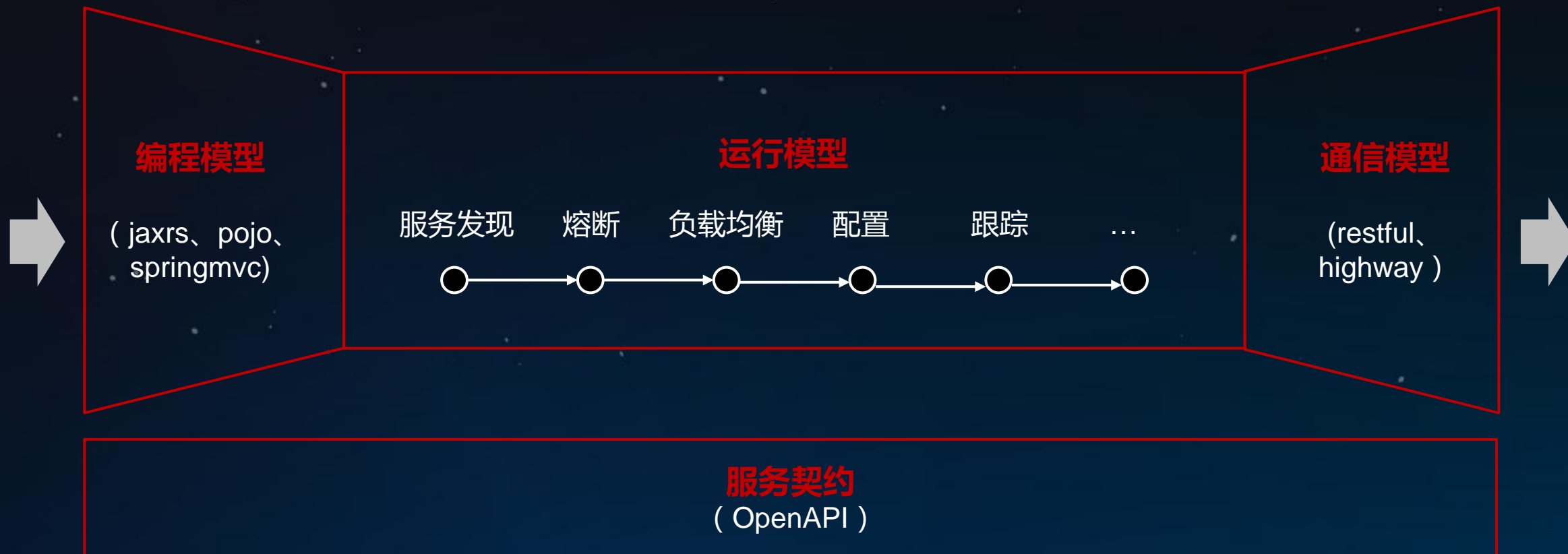
ServiceComb 开发框架

LINUXCON

containercon

CLOUDOPEN

CHINA 2017



基本概念

- SchemaMeta

服务接口定义元数据（服务契约）：一个微服务可以拥有多个schema文件，在同一个微服务当中每个schema文件都有唯一Schema-Id与之对应；

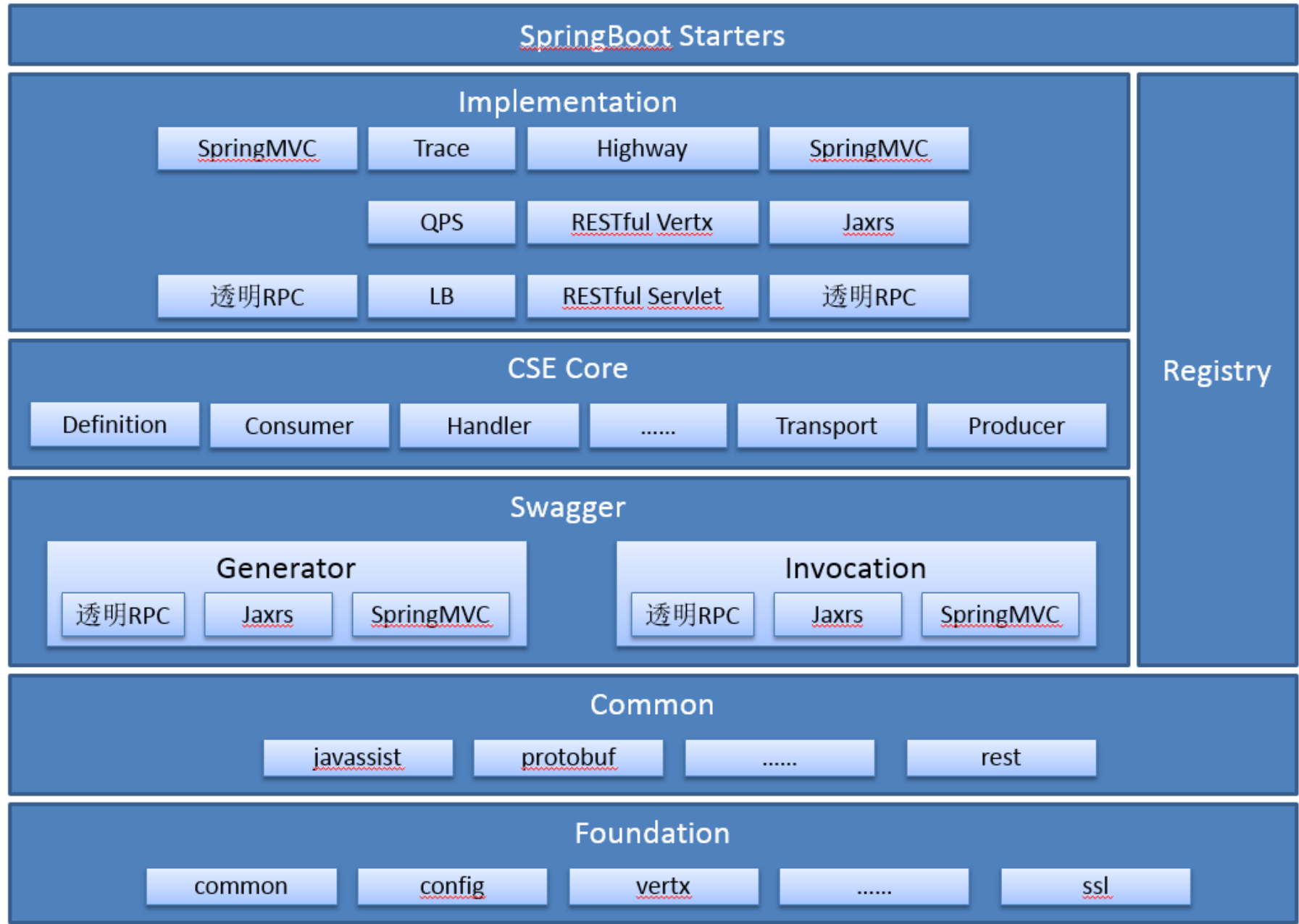
- MicroService

微服务元数据：包括应用名、微服务名称、微服务Id、版本、描述、Schema-id等；

- MicroServiceInstance

微服务实例：一个独立的拥有自IP端口的微服务实例（通常为进程），与service id的关系为n:1，即Service ID可以拥有多个微服务实例；

系统模块图



框架的启动与停止

pkg

Log4jUtils

```
+ init() : void  
+ init(locationPattern : String) : void  
+ init(locationPatterns : List) : void
```

- 1、框架使用 slf4j，支持对接 log4j/log4j2/logback；
- 2、使用框架的 log4j 的话支持 properties 文件扩展；
- 3、默认加载 config/log4j.properties 以及 log4j.*.properties 文件，对所有文件进行 merge 后输出到日志配置文件列表最后一个的父目录（如果该目录可写）

BeanUtils

```
+ init() : void  
+ init(configLocations : String) : void  
+ getBean(name : int) : T<T>  
+ getImplClassFromBean(bean : obeit) : Class
```

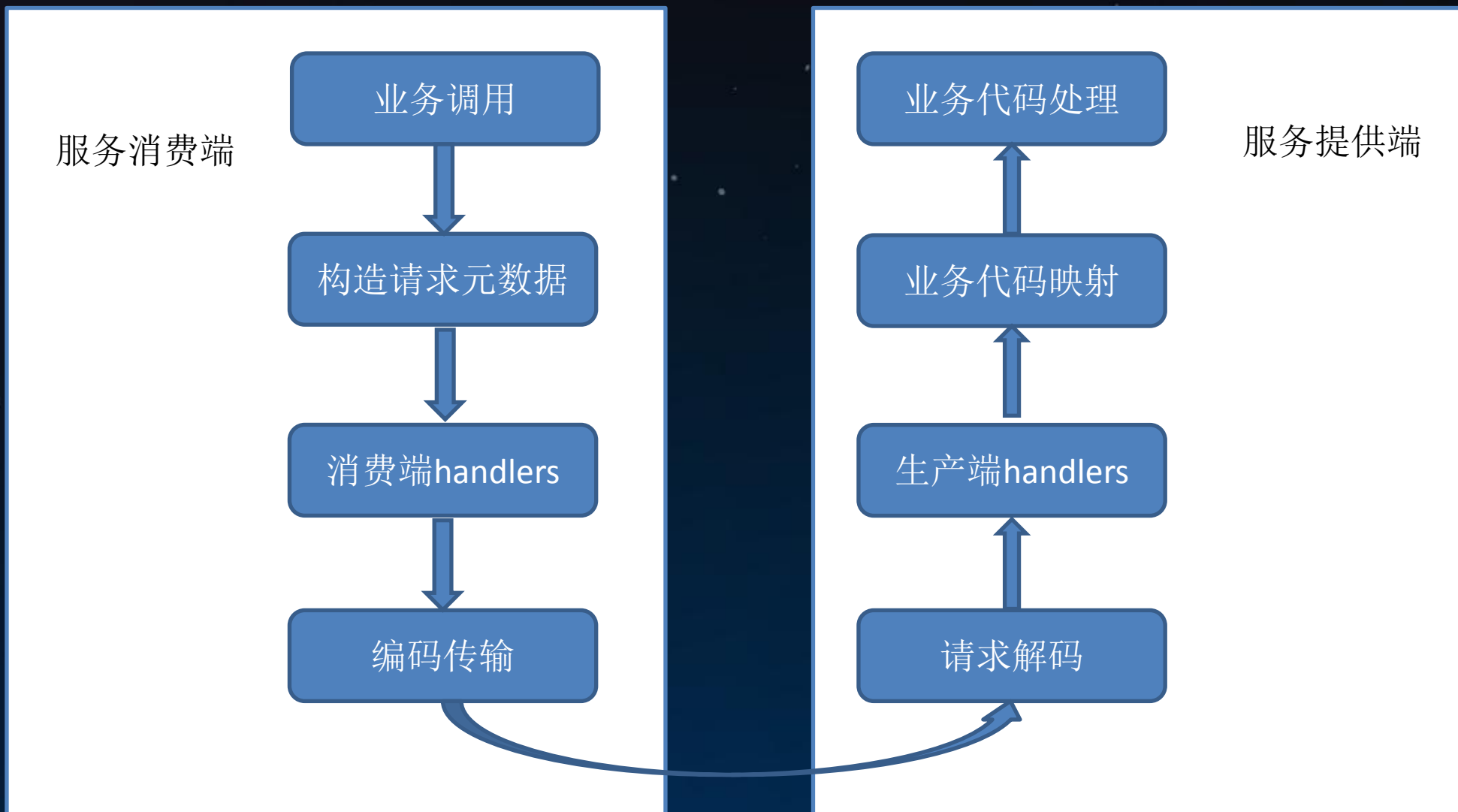
- 1、通过 ClassPathXmlApplicationContext 读取 spring 的 bean 配置文件；
- 2、默认读取 META-INF/spring/*.bean.xml

CseApplicationListener

```
- producerProviderManager : ProducerProviderManager  
- consumerProviderManager : ConsumerProviderManager  
- transportManager : TransportManager  
- schemaListenerManager : SchemaListenerManager  
- applicationContext : ApplicationContext  
  
+ onApplicationEvent(event : ApplicationEvent) : void
```

- 1、handles 相关配置文件加载
- 2、服务提供端接口定义文件的加载
- 3、发布微服务实例监听的 endpoints 到微服务实例数据当中
- 4、启动服务注册发现线程
- 5、优雅停机，卸载本微服务实例信息

框架运行时请求发送与处理过程



如何参与到ServiceComb社区

• 线上：

- ✓ 关注微信公众号获取信息
- ✓ 加入微信群进行交流
- ✓ 通过邮件列表参与讨论
- ✓ 通过Github发起PR

• 线下：

- ✓ 月度Meetup
- ✓ 不定期沙龙探讨



微服务引擎CSE

微服务引擎(Cloud Service Engine)是企业级微服务应用管理平台,包含微服务SDK,服务注册、配置和治理中心,帮助用户实现微服务应用的快速构建、实时监控和高可用保障。兼容主流开源生态,不绑定特定开发框架和平台,支持已有应用业务代码零修改接入

免费

立即使用



低门槛

微服务框架开放无绑定,开箱即用;
微服务流水线一键式构建、部署和上线



零改造

提供非侵入式方案,让遗留应用零改造,轻松上云;兼容SpringCloud主流开源社区,与业界生态能力互通



高可靠

三十年华为电信级高可靠经验积累,历经华为终端云亿级用户考验



专业服务

20+ISV行业生态合作伙伴;30+微服务高级咨询师



微服务引擎官方论坛



答疑求助
微服务技术探讨
教程攻略
干货分享
禅与摩托车
颈椎康复...
人生感悟...