LINUXCON
containercon
CLOUDOPEN
CHINA 2017
HUAWEI

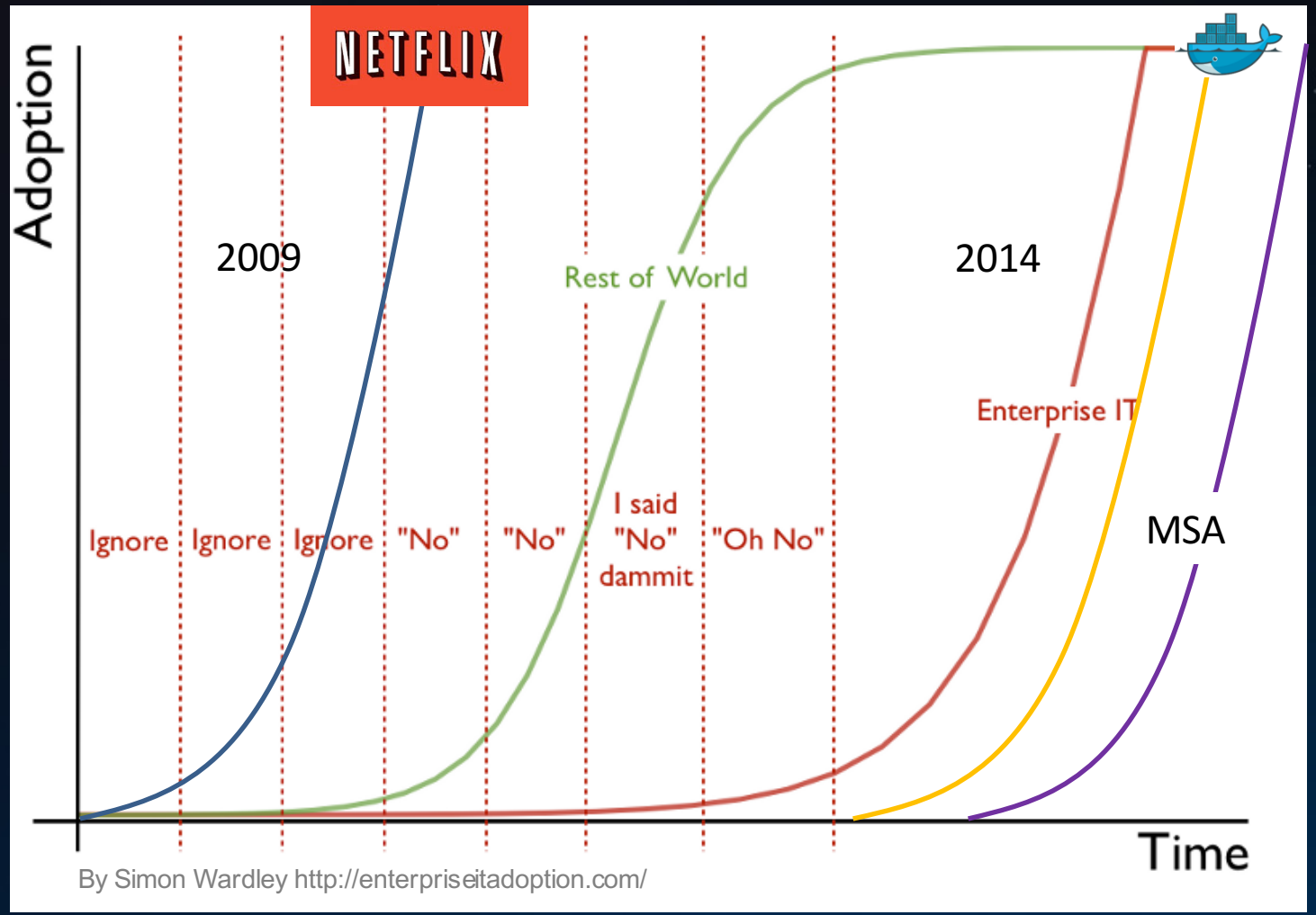
ServiceComb 的前生今世

张琦

议题

- 企业应用微服务化的趋势和动机
- 面临的问题
- 解决思路和设计方案

企业IT技术应用曲线



- 2014年被很多人认为是企业上云元年

- 2014年几乎企业应用都没有考虑的Docker在2015年就已经被所有人纳入应用计划

- 微服务在2016年成为仅次于物联网和认知计算的第三热门技术

传统企业应用开发模式

- **技术实现单一，需要想办法用一种技术解决所有问题**
- **只能按大颗粒系统发布版本，响应周期长**
  （小特性版本3-6个月，每年1个大版本）
- **无法做到永远在线，大版本升级时，要停机中断服务**

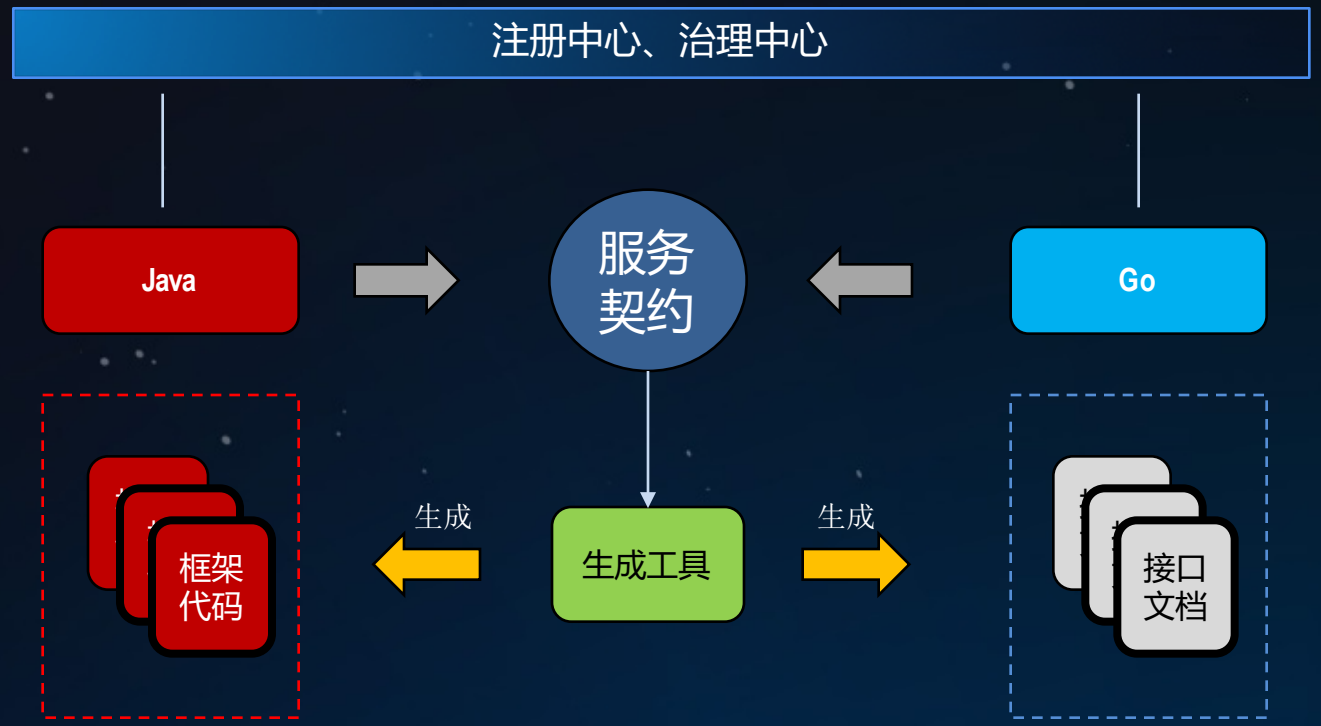
微服务化的应用构建和发布

Speed & Safety

面临的问题

- 企业应用和互联网应用的不同？
- 如何做到不同ISV的应用互联互通统一管理？
- 怎么才能加快微服务的开发？
- 微服务化后如何保证性能？
- 如何进行统一的路由控制？

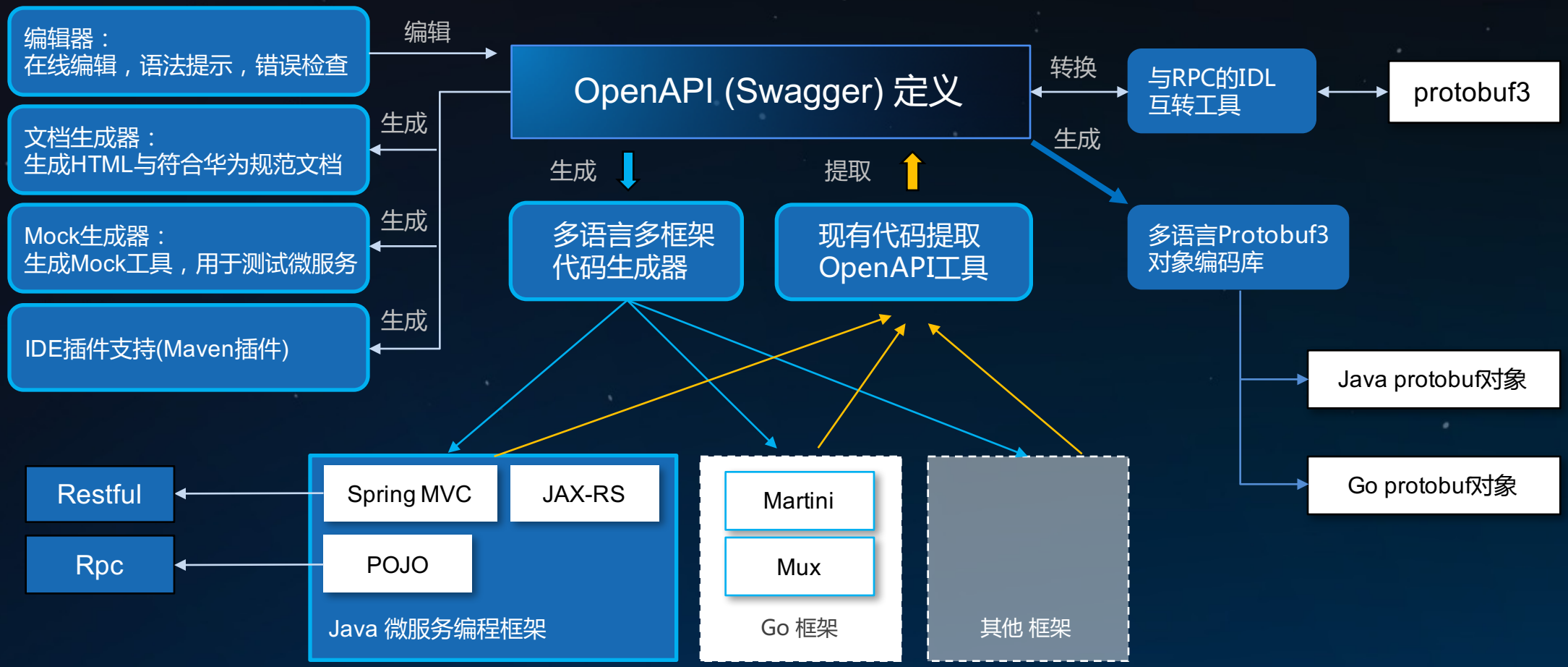
企业应用和集成



API First

- 面向契约而不是逻辑
- 解耦服务提供者和消费者的开发顺序
- 契约定义为语言中立
- 规范化系统接口，让实现与文档的同步成为必须
- 通过工具简化整个过程

增速微服务开发——工具支持



增速微服务开发——降低学习门槛

SpringMVC

Provider service:

```
import io.servicecomb.*;
import org.springframework.*;

@RestSchema(schemaId = "helloworld")
@RequestMapping(path = "/helloworld",produces = MediaType.APPLICATION_JSON)
public class HelloWorldProvider implements HelloWorld {

    @RequestMapping(path = "/sayHello",method = RequestMethod.GET )
    public String sayHello(@RequestParam("name") String name) {
        return "Hello " + name;
    }
}
```

Consumer service:

```
import io.servicecomb.*;
import org.springframework.*;

@Component
public class HelloWorldConsumer  {
        private static RestTemplate restTemplate = RestTemplateBuilder.create();

        public static void main(String[] args) {
       String result= restTemplate.getForObject("cse://springmvc/helloworld/syaHello?name={name}",String.class,"Ta
        }
}
```

JAXRS

Provider service:

```
import io.servicecomb.*;
import javax.ws.rs.*;

@RestSchema(schemaId = "helloworld")
@Path("/helloworld")
@Produces(MediaType.APPLICATION_JSON)
public class HelloWorldProvider implements HelloWorld {
    @Path("/sayHello")
    @GET
    public String sayHello(@Pathparam("name") String name) {
        return "Hello " + name;
    }
}
```

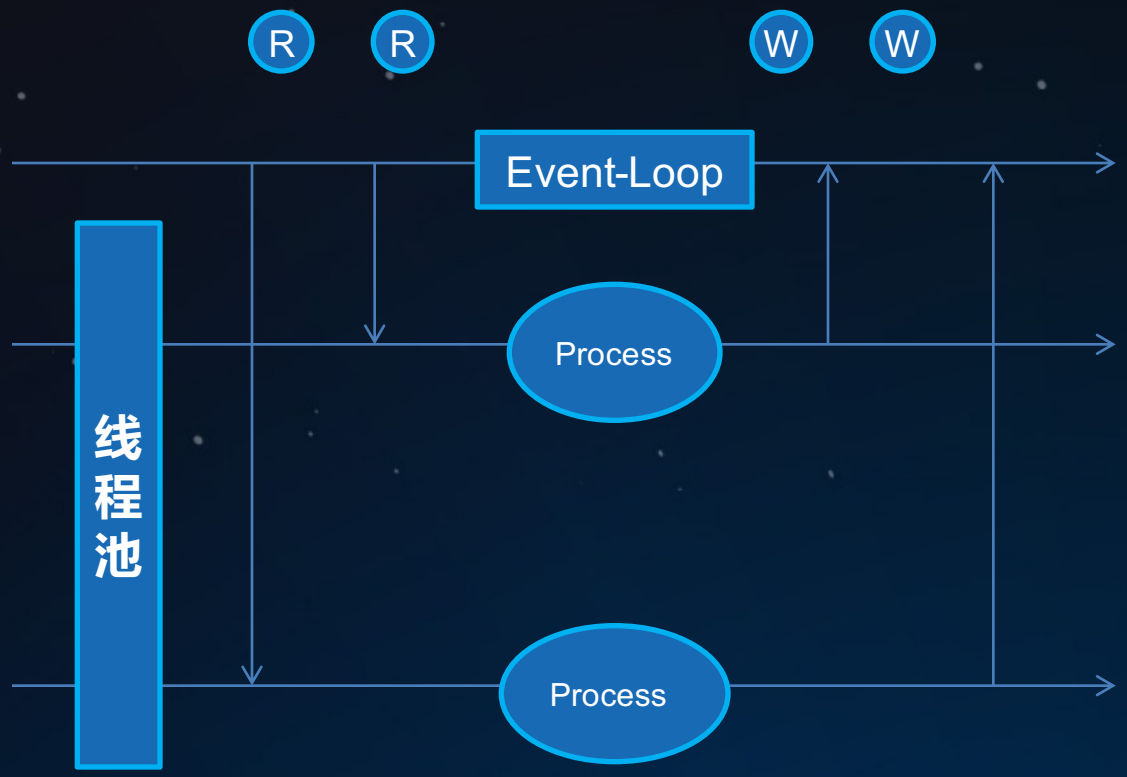
Consumer service:

```
import io.servicecomb.*;
import org.springframework.*;

@Component
public class HelloWorldConsumer  {
        private static RestTemplate restTemplate = RestTemplateBuilder.create();

        public static void main(String[] args) {
       String result= restTemplate.getForObject("cse://jaxrs/helloworld/syaHello?name={name}",String.class,"Tank");
        }
}
```

性能保证



- 异步
- 标准、开放、协议健壮性
- 开发框架的性能在于细节，而不仅仅是协议。

各主流微服务框架性能对比结果：

机型	协议	规格(CPU/内存/网卡)	EDAS	实例数	报文大小	调用线程数	TPS	时延(ms)	服
VM	RPC	8U16G1G	EDAS	1	1k	100	58431	1.73	2
物理机	RPC	24U64G1G	EDAS	1	1k	100	97586	1.01	3

机型	协议	规格(CPU/内存/网卡)	CSE SDK	实例数	报文大小	调用线程数	TPS	时延(ms)
VM	REST	8U16G1G	CSE SDK	1	1k	100	70669	1.414
物理机	REST	24U64G1G	CSE SDK	1	1k	100	107126	0.933

机型	协议	规格(CPU/内存/网卡)	Tars	实例数	报文大小	调用线程数	TPS	时延(ms)	服
VM	RPC?	8U16G1G	Tars	1	1k	100	75386	1.32	21

机型	协议	规格(CPU/内存/网卡)	SpringCloud	实例数	报文大小	调用线程数	TPS	时延(ms)
VM	REST	8U16G1G	SpringCloud	1	1k	100	22507	4.70
物理机	REST	24U64G1G	SpringCloud	1	1k	100	51382	2.84

更细致的服务路由管控

- 统一的路由策略管控
- 缓存以提升性能
- 支持pull/push两种模式监控实例变化
- 实例动态扩容，海量的长连接或者短连接
- 支持灰度发布、服务分组等高级管理特性

ServiceComb 开发框架

LINUXCON
containercon
CLOUDOPEN
CHINA 2017
HUAWEI

Thank You

- Github：https://github.com/ServiceComb
- 官网：https://www.servicecomb.io