

Dubbo 过去、现在以及未来

马昕曦 (小马哥)
阿里巴巴

自我介绍

姓名：马昕曦

花名：桃谷

昵称：小马哥

阿里巴巴技术专家，Dubbo 维护者之一，负责阿里巴巴微服务架构推广、技术实施、架构衍进、基础设施构建等。

GitHub：<https://github.com/mercyblitz>

微信/微博：mercyblitz

议程

基本介绍

功能架构

微服务支持

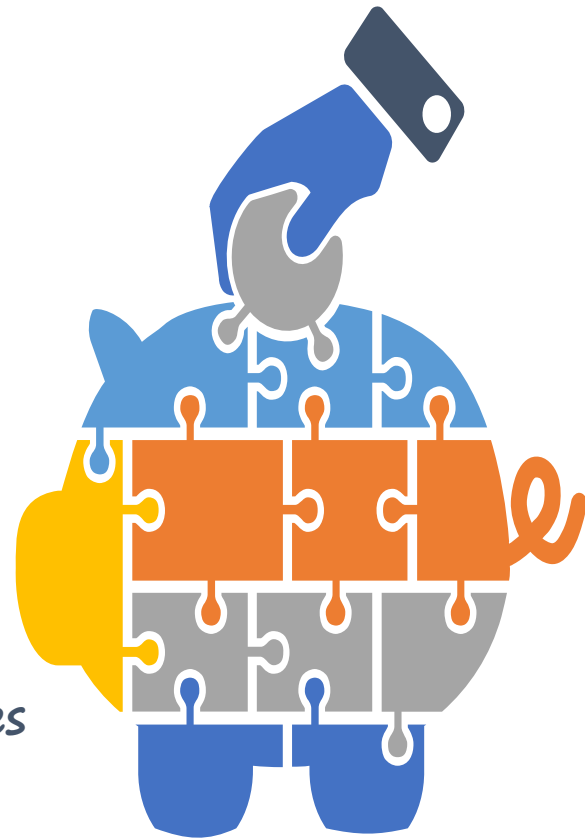
路线规划

问答互动

基本介绍

Dubbo 介绍

- ✓ *Open sourced by Alibaba at 2011*
- ✓ *Production-tested and Proven At Scale*
- ✓ *A high performance RPC Framework*
- ✓ *Beyond RPC: Rich features for micro-services*



当前情况

1

社区活跃

- 14 K + starts
- 11 K + forks
- 100+ issue & PR / 每周

2

服务化基石

- 阿里内部微服务支持
- 生产环境运行多年

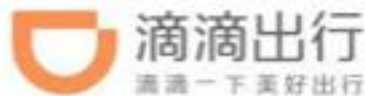
3

广泛运用

- 多家大规模公司使用
- 多种衍生框架

The screenshot shows the GitHub repository page for `alibaba/dubbo`. At the top, it displays the repository name and navigation options: Unwatch (2,577), Unstar (14,771), and Fork (11,346). Below this, there are tabs for Code, Issues (401), Pull requests (30), Projects (0), Wiki, Insights, and Settings. The main content area features a description: "Dubbo is a high-performance, java based, open source RPC framework" with a link to <http://dubbo.io>. Below the description are several topic tags: `dubbo`, `rpc`, `rpc-framework`, `java`, `high-performance`, `soa`, `service-oriented`, `service-registration`, `distributed-systems`, `service-discovery`, `cluster`, `spring`, `service-consumer`, `service-provider`, `service-registry`, `microservices`, and `Manage topics`. At the bottom, a summary bar shows: 2,023 commits, 4 branches, 47 releases, 45 contributors, and Apache-2.0 license.

谁在使用？



更多：<https://github.com/alibaba/dubbo/issues/1012>

功能架构

核心功能

透明 RPC

本地接口代理实现 RPC 透明调用



版本 & 分组管理

分组管理不同服务实现；
同服务多版本管理

服务注册 & 发现

服务自动注册和发现，多种注册中心支持



多协议支持

通讯协议: dubbo, hessian, thrift, RMI ;
序列化协议: hessian2, json ...

实时流量管控

自定义路由规则, 流量管控策略, failover 以及故障检测



负载均衡

内建随机、一致性 Hash、轮训等多种负载均衡组件，支持自定义扩展

SPI 扩展

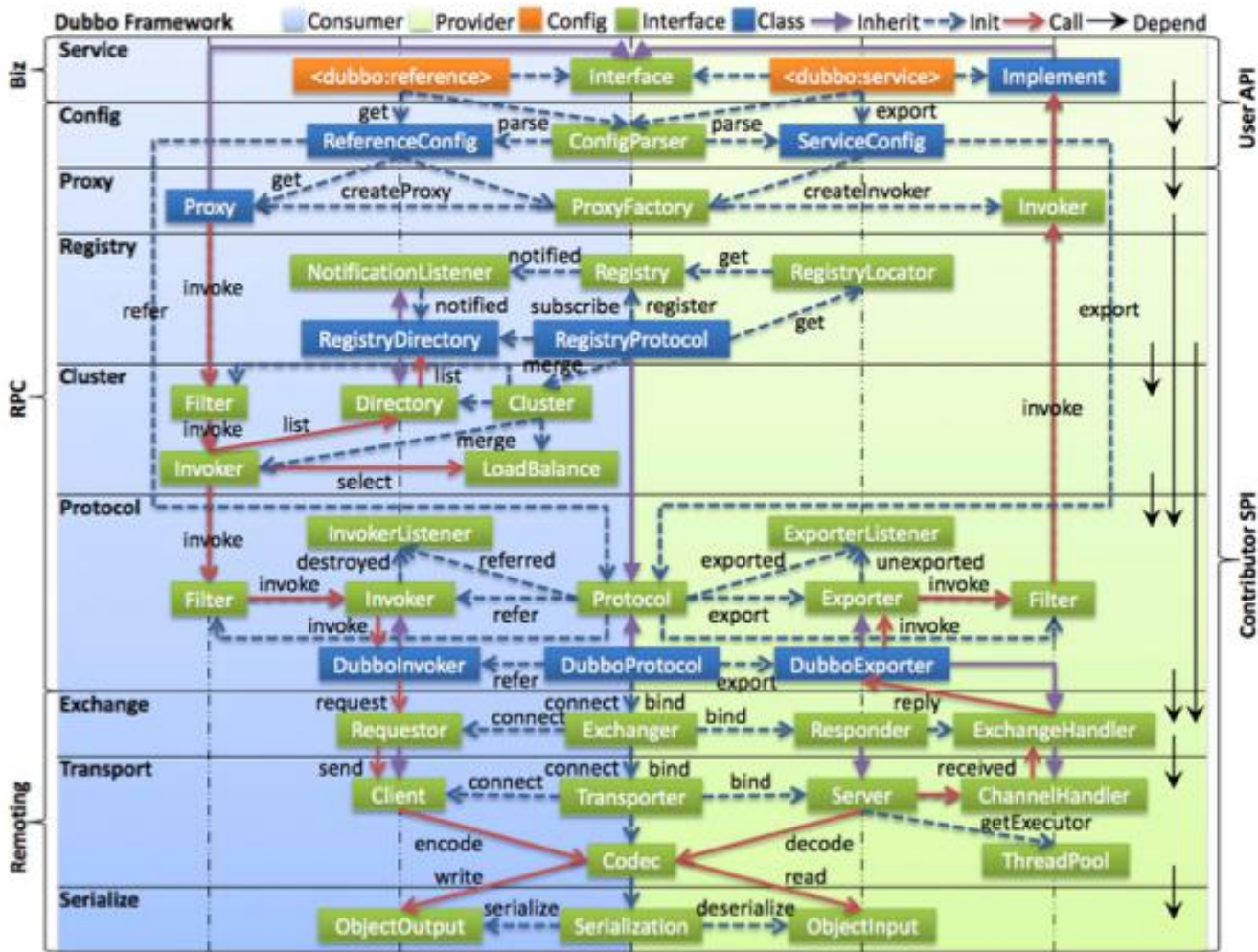
丰富的SPI 扩展点：协议、集群、路由、负载均衡、注册中心等



Production-Ready

运行时动态配置、dubbo-admin OPS, telnet 命令支持等

整体架构



工作原理

0

服务导出

服务提供方导出服务，监听端口

1

服务注册

服务提供方注册服务信息到注册中心

2

服务订阅

服务消费方订阅关注的服务

3

服务发现

注册中心通知服务消费端，当服务地址发生变更

4

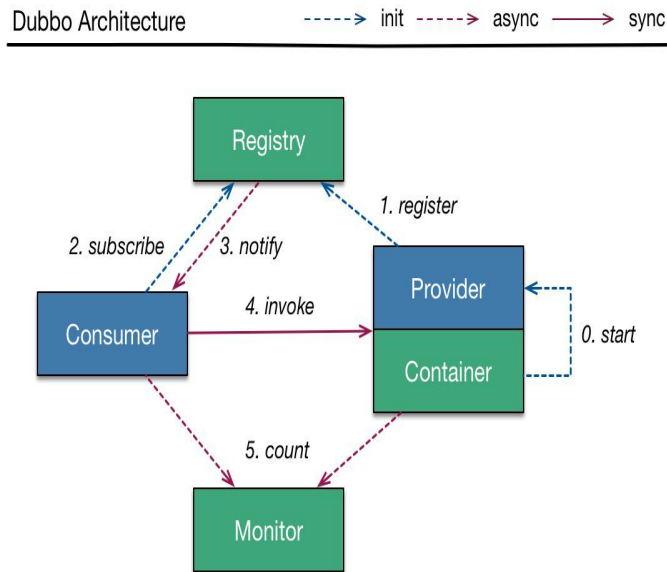
服务调用 (RPC)

根据负载均衡以及路由规则选择服务地址，直接调用

5

监控

监控器收集和展示服务提供方以及服务提供方之间的统计信息

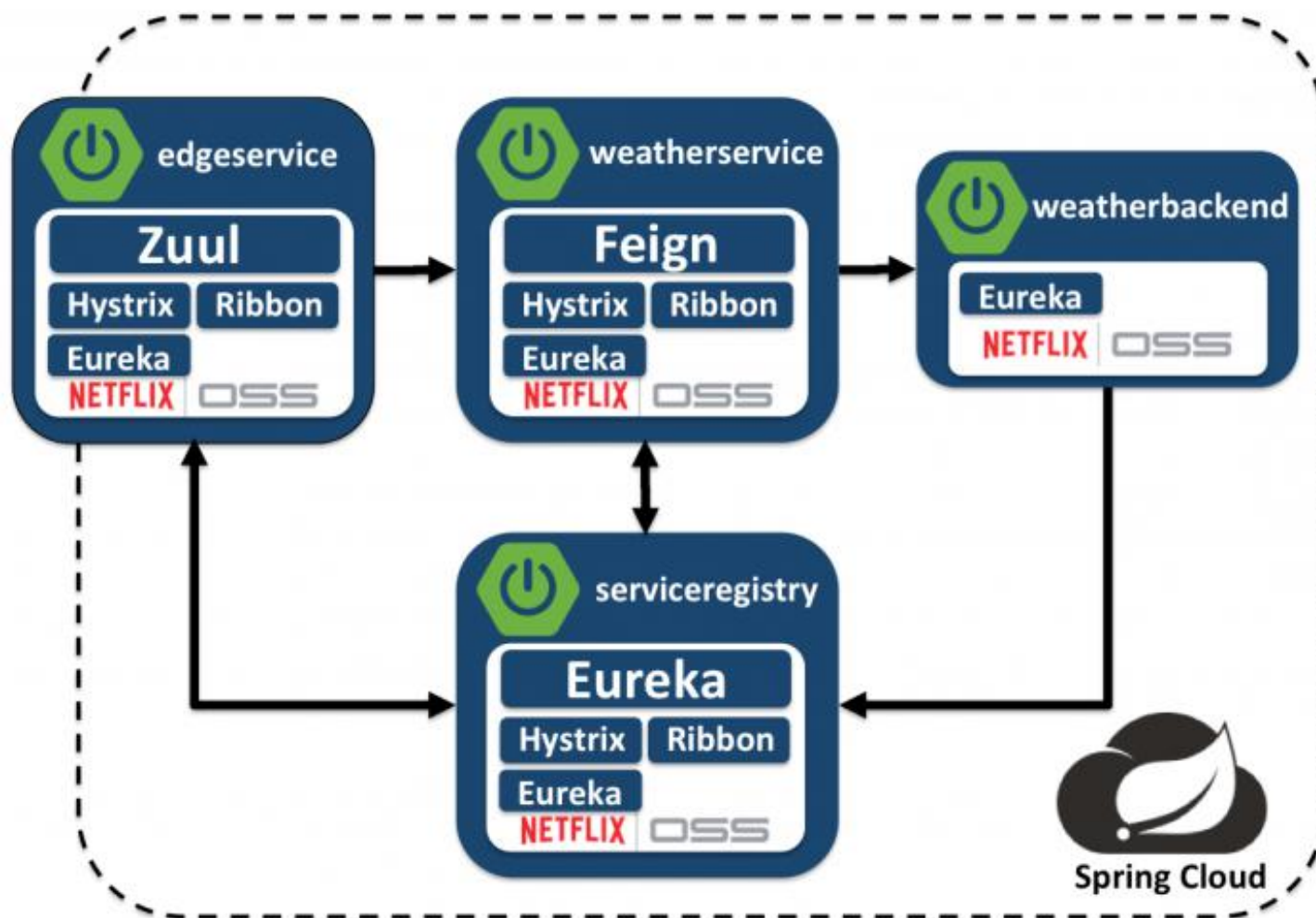


微服务支持

微服务关注

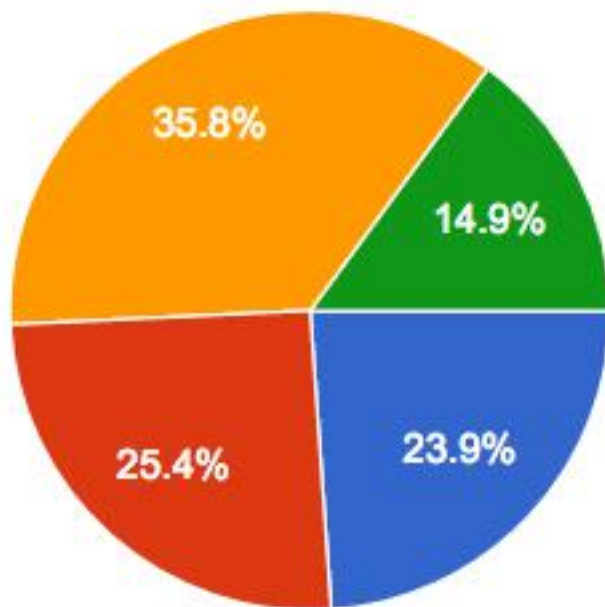


Spring Cloud + Netflix OSS



微服务分布

Do you use Microservices?

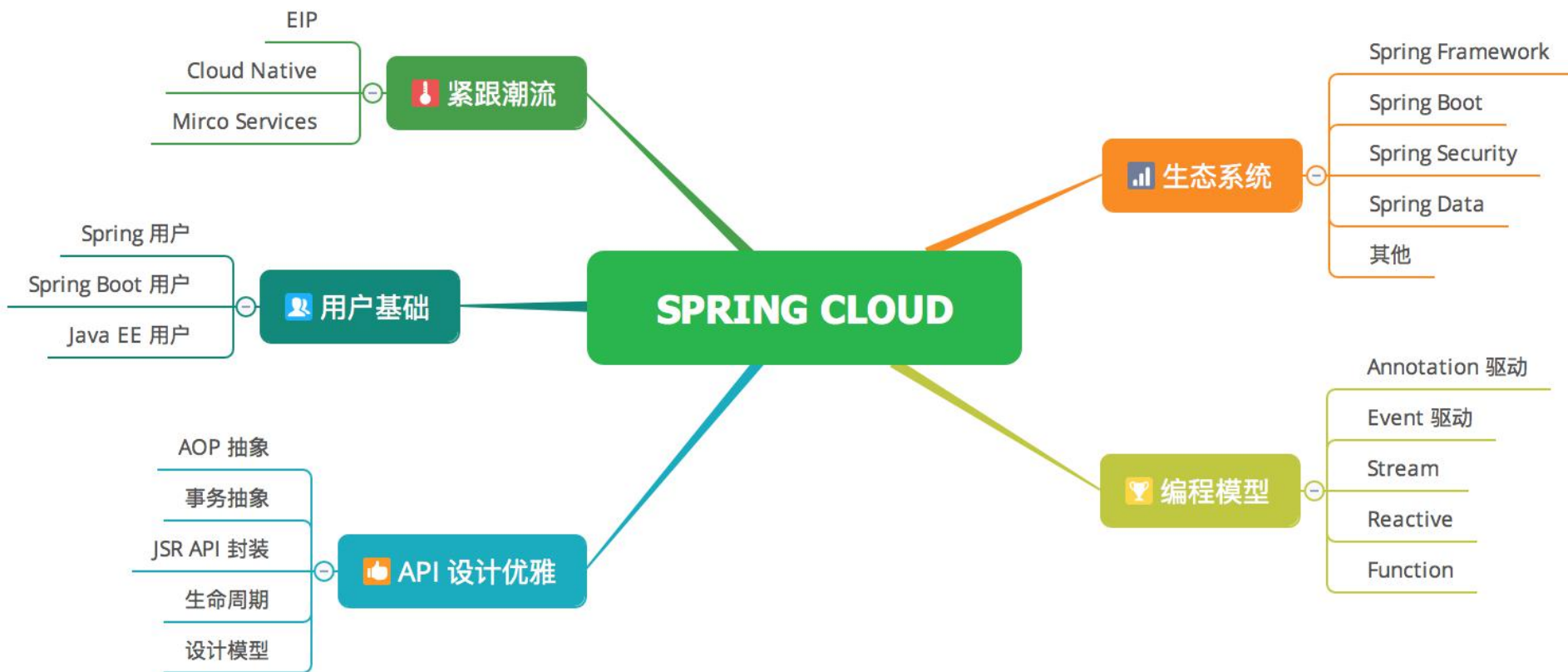


- Microservices are not an option for us.
- We intend to transform our existing software monolith into Microservices.
- We build new components in a microservice style while maintaining our software m...
- We use microservices a lot in our main application.

放弃 Spring Cloud ?



Spring Cloud 核心优势

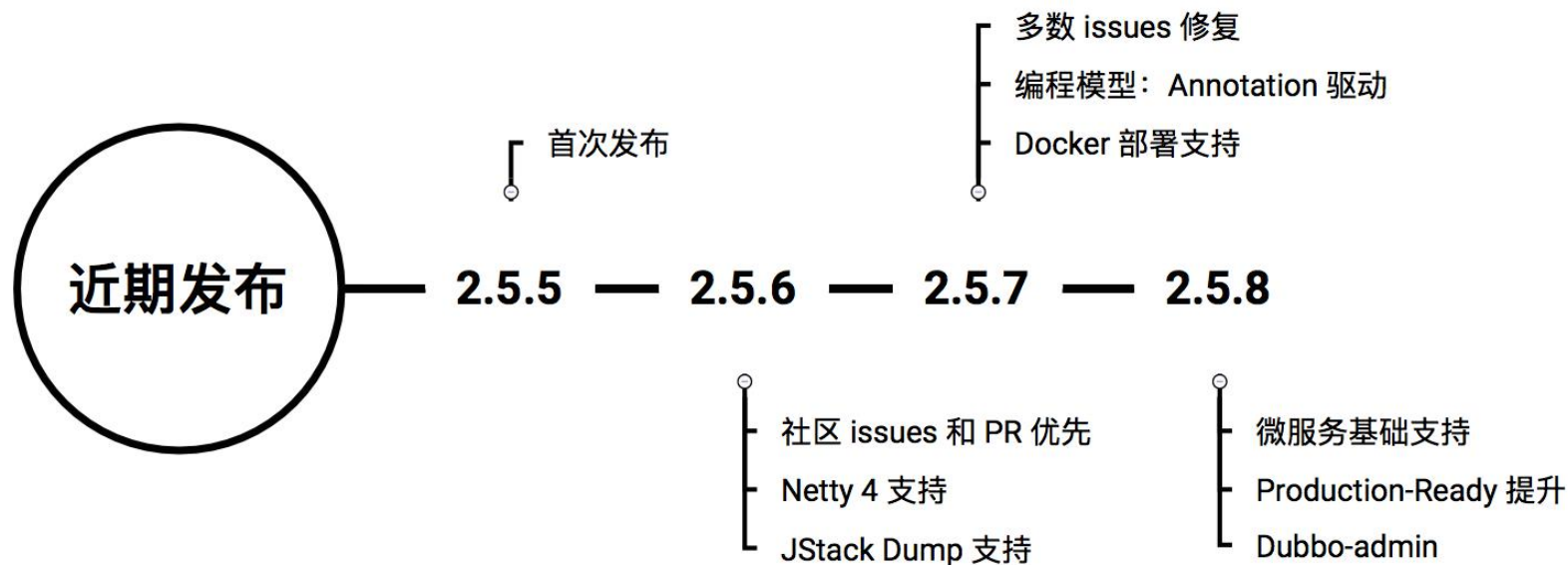


Spring Cloud VS Dubbo

RPC 相关比较

微服务技术	Spring Cloud	Dubbo
分布式管理	Git、Consul、Zookeeper	Zookeeper、Redis
服务发现	Eureka、Consul、Zookeeper	Zookeeper、Redis、Multicast
路由	Zuul	Router
服务调用	Feign / REST	多种实现（长连接）
负载均衡	Zuul、Ribbon	LoadBalance
服务熔断	Hystrix	Fallback, Failover
服务安全	Security	Token
分布式跟踪	Zipkin	无

Dubbo 微服务准备



Dubbo 微服务准备 - 依赖升级

Dependencies Upgrade

dependency	current	target	influences
spring	3.2.16.RELEASE	4.3.10.RELEASE	schema parse; Http RPC
zookeeper	3.3.3	3.4.9	registry
zkclient	0.1	0.10	zookeeper client framework
curator	1.1.16	2.12.0	zookeeper client framework
commons-logging	1.1.1	1.2	Logger
hessian	4.0.6	4.0.38	hessian RPC
jedis	2.1.0	2.9.0	redis registry; redis RPC
httpclient	4.1.2	4.5.3	hessian uses http connection pool
validator	1.0.0	1.1.0.Final	java validation
cxfr	2.6.1	3.0.14	webservice
jcache	0.4	1.0.0	jcache

Dubbo 微服务准备 - Annotation 驱动

```
@DubboComponentScan(basePackages = "com.alibaba.dubbo.config.spring.context.annotation.provider")
@PropertySource("META-INF/default.properties")
@EnableTransactionManagement
public class ProviderConfiguration {

    /**
     * Current application configuration, to replace XML config:
     * <prev>
     * &lt;dubbo:application name="dubbo-annotation-provider"/&gt;
     * </prev>
     *
     * @return {@link ApplicationConfig} Bean
     */
    @Bean("dubbo-annotation-provider")
    public ApplicationConfig applicationConfig() {
        ApplicationConfig applicationConfig = new ApplicationConfig();
        applicationConfig.setName("dubbo-annotation-provider");
        return applicationConfig;
    }
}
```

Dubbo 微服务准备 - 外部化配置

```
@EnableDubboConfig(multiple = true)
@PropertySource("META-INF/config.properties")
private static class TestMultipleConfig {

}
```

```
@EnableDubboConfig
@PropertySource("META-INF/config.properties")
private static class TestConfig {

}
```

```
application.prefix = dubbo.application.
# single bean definition
```

```
## application
dubbo.application.id = applicationBean
dubbo.application.name = dubbo-demo-application
```

```
## module
dubbo.module.id = moduleBean
dubbo.module.name = dubbo-demo-module
```

```
## registry
dubbo.registry.address = zookeeper://192.168.99.100:32770
```

```
## protocol
dubbo.protocol.name = dubbo
dubbo.protocol.port = 20880
```

```
## monitor
dubbo.monitor.address = zookeeper://127.0.0.1:32770
```

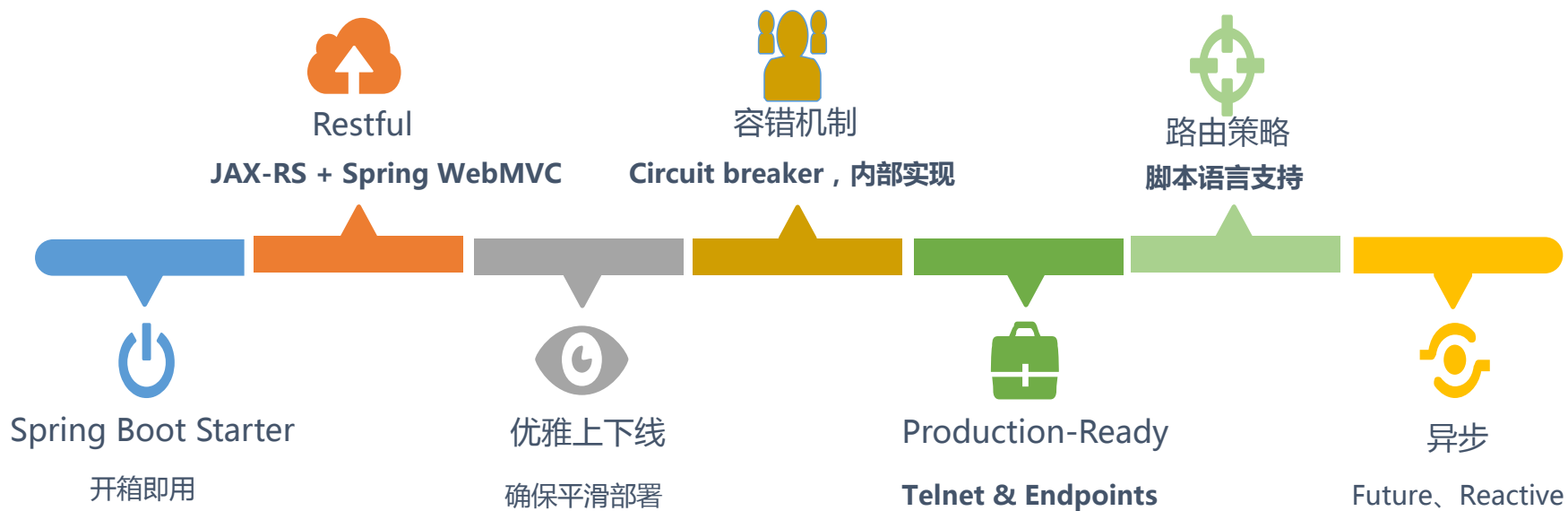
```
## provider
dubbo.provider.host = 127.0.0.1
```

Dubbo 微服务准备 – 组件装配

```
@EnableDubbo(scanBasePackages = "com.alibaba.dubbo.config.spring.context.annotation.provider")  
@ComponentScan(basePackages = "com.alibaba.dubbo.config.spring.context.annotation.provider")  
@PropertySource("META-INF/dubbb-provider.properties")  
@EnableTransactionManagement  
public static class TestProviderConfiguration {
```

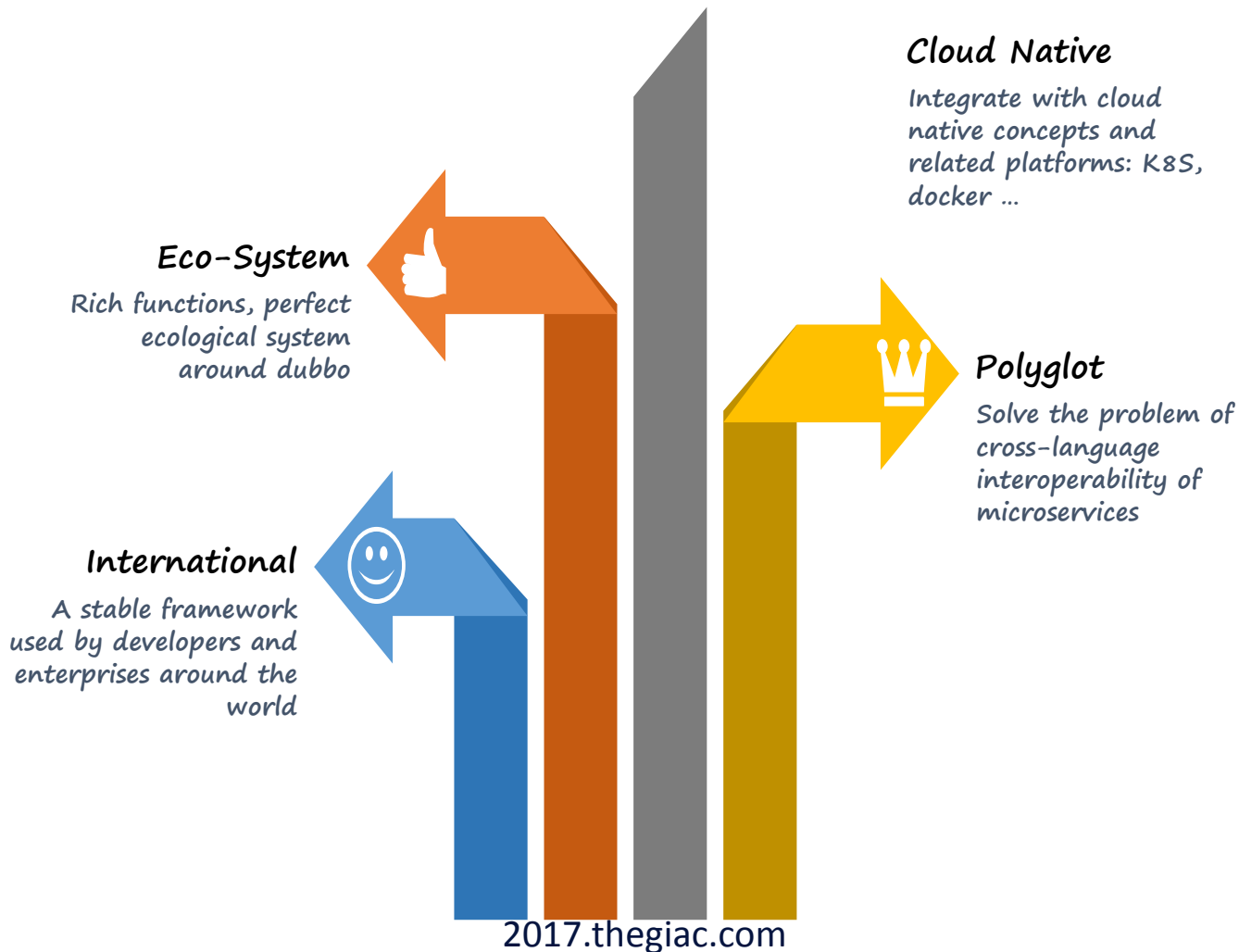
@EnableDubbo = @EnableDubboConfig + @DubboComponentScan

Dubbo 微服务进行时

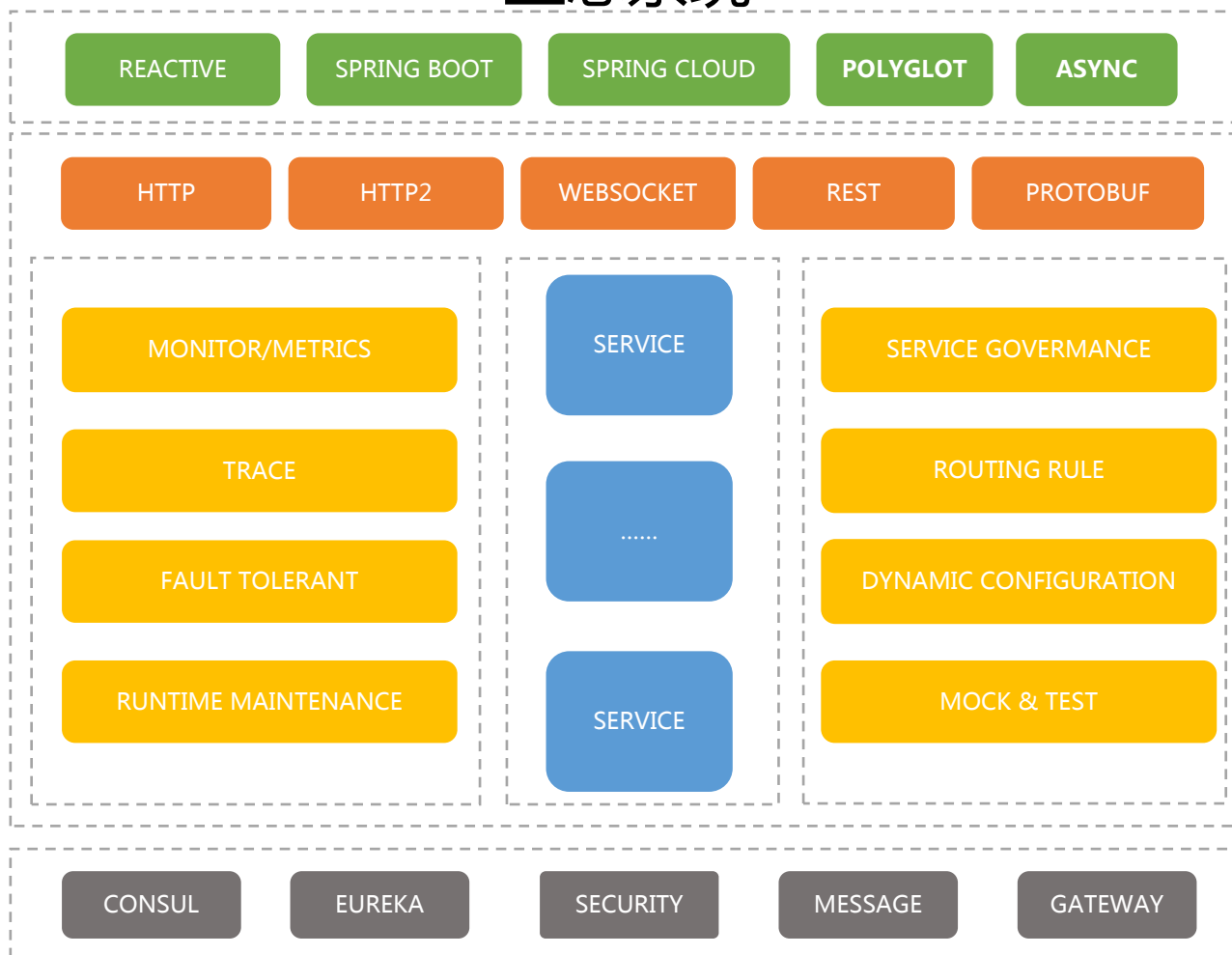


路线规划

长期计划



生态系统



需要您的力量



COMMUNITY

Github Issue: <https://github.com/alibaba/dubbo/issues>

Gitter: <https://gitter.im/alibaba/dubbo>

Google 邮件组: dubbo@googlegroups.com



TRANSLATION

文档: <https://github.com/alibaba/dubbo/issues/948>

Code Base: <https://github.com/alibaba/dubbo>

Blogs: <http://www.baeldung.com/dubbo>



CODE & DESIGN

Pull Request: <https://github.com/alibaba/dubbo/pulls>

《Spring Boot 编程思想》

上市时间：2018 年

特点：

系统学习，拒绝浅尝辄止
重视规范，了解发展趋势
场景分析，掌握技术选型
源码解读，理解设计思想
实战演练，巩固学习成果

谢谢

GIAC | 全球互联网架构大会
GLOBAL INTERNET ARCHITECTURE CONFERENCE

GIAC

全球互联网架构大会

GLOBAL INTERNET ARCHITECTURE CONFERENCE



扫码关注GIAC公众号

2017.thegiac.com