Spring Cloud 官方新选择 — Spring Cloud Alibaba

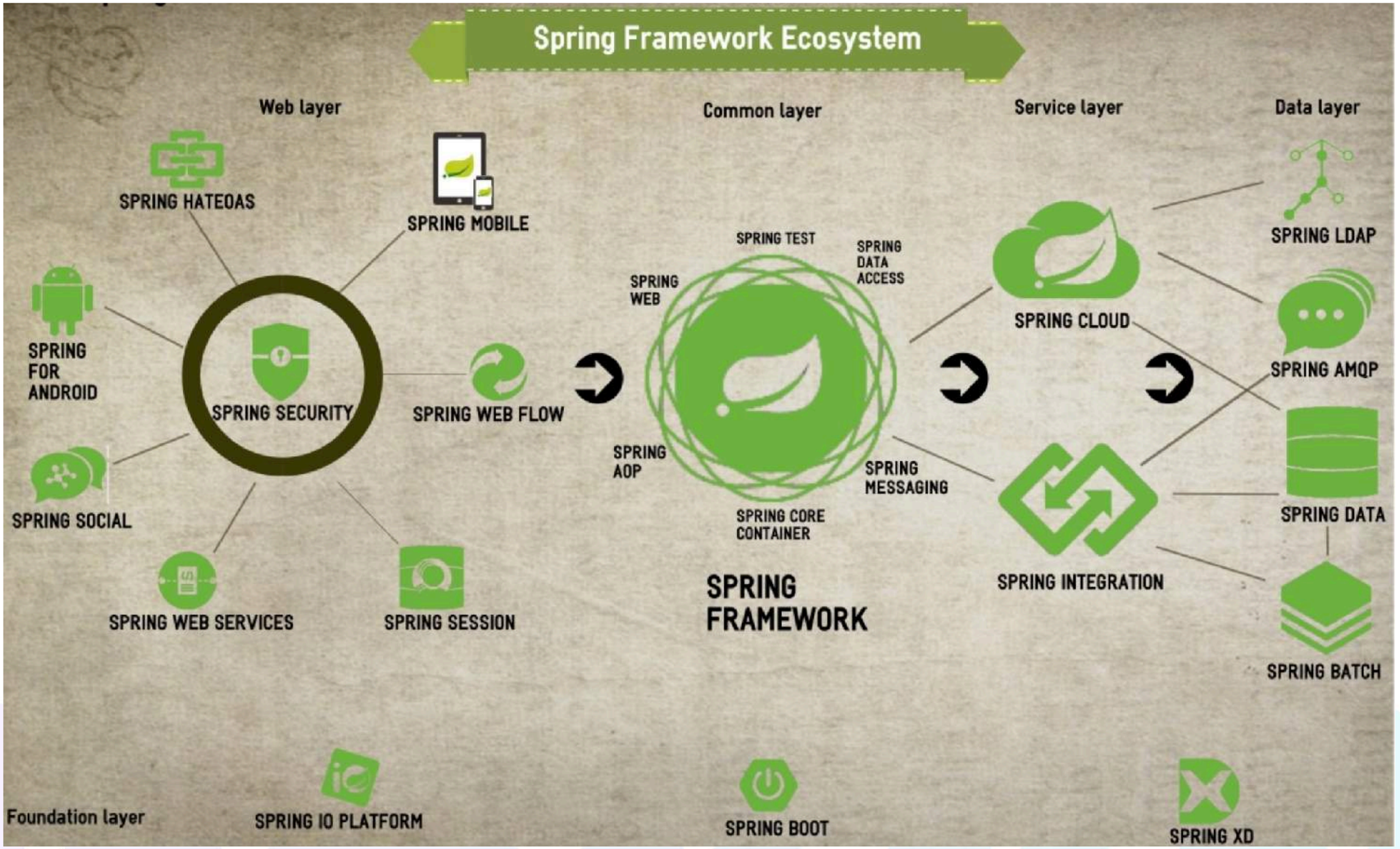
方剑@fangjian0423

Spring Cloud Alibaba PMC, 阿里云 EDAS 产品开发

大纲

- Spring Ecosystem 简介
- Spring Cloud Alibaba
  - Spring Cloud Sentinel
  - Spring Cloud Nacos Config & Discovery
  - Spring Cloud RocketMQ Binder & Bus
  - Spring Cloud Dubbo
  - Spring Cloud Seata
- Demo – Spring Cloud Alibaba 微服务应用开发

Spring Framework Ecosystem



Spring Framework & Boot & Cloud

Core of All Spring Projects

DI, Events, Data Access, MVC, WebFlux, Tx, AOP, Messaging

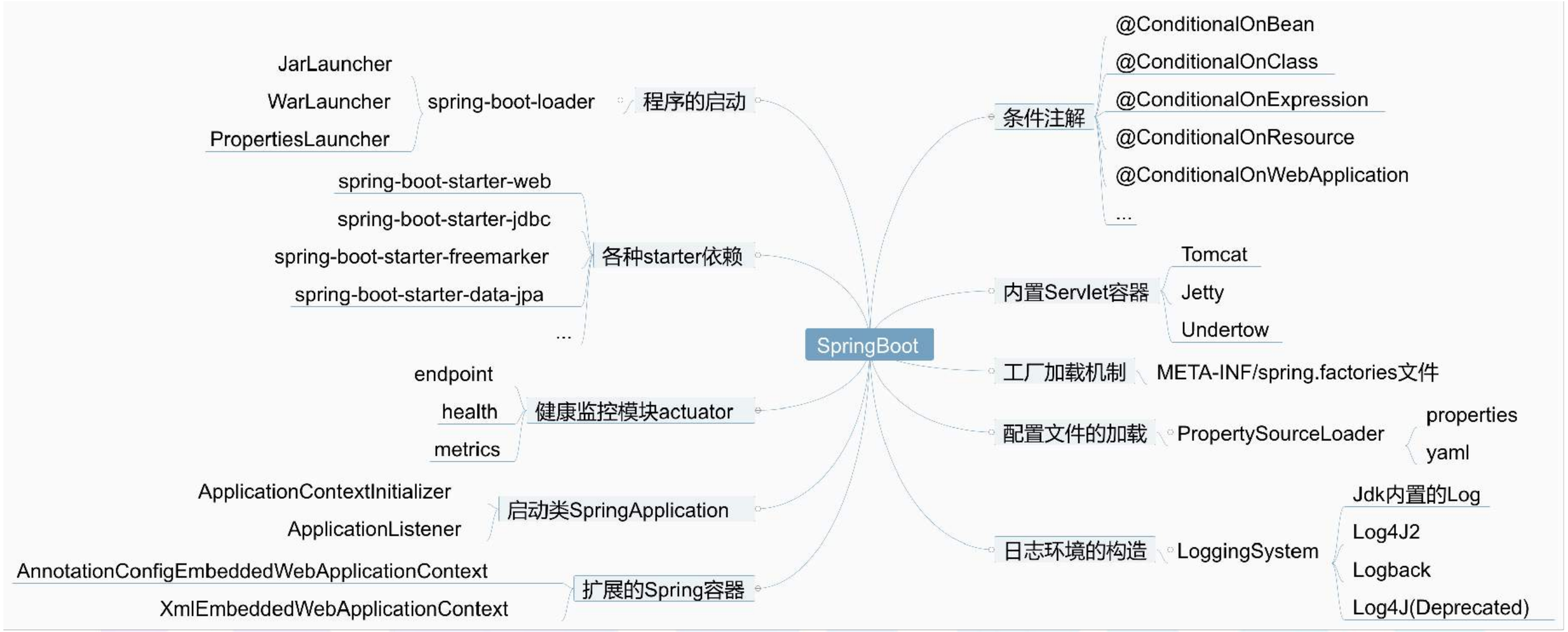
Build Anything

外部化配置，自动化配置，工厂加载机制，内置容器，条件注解

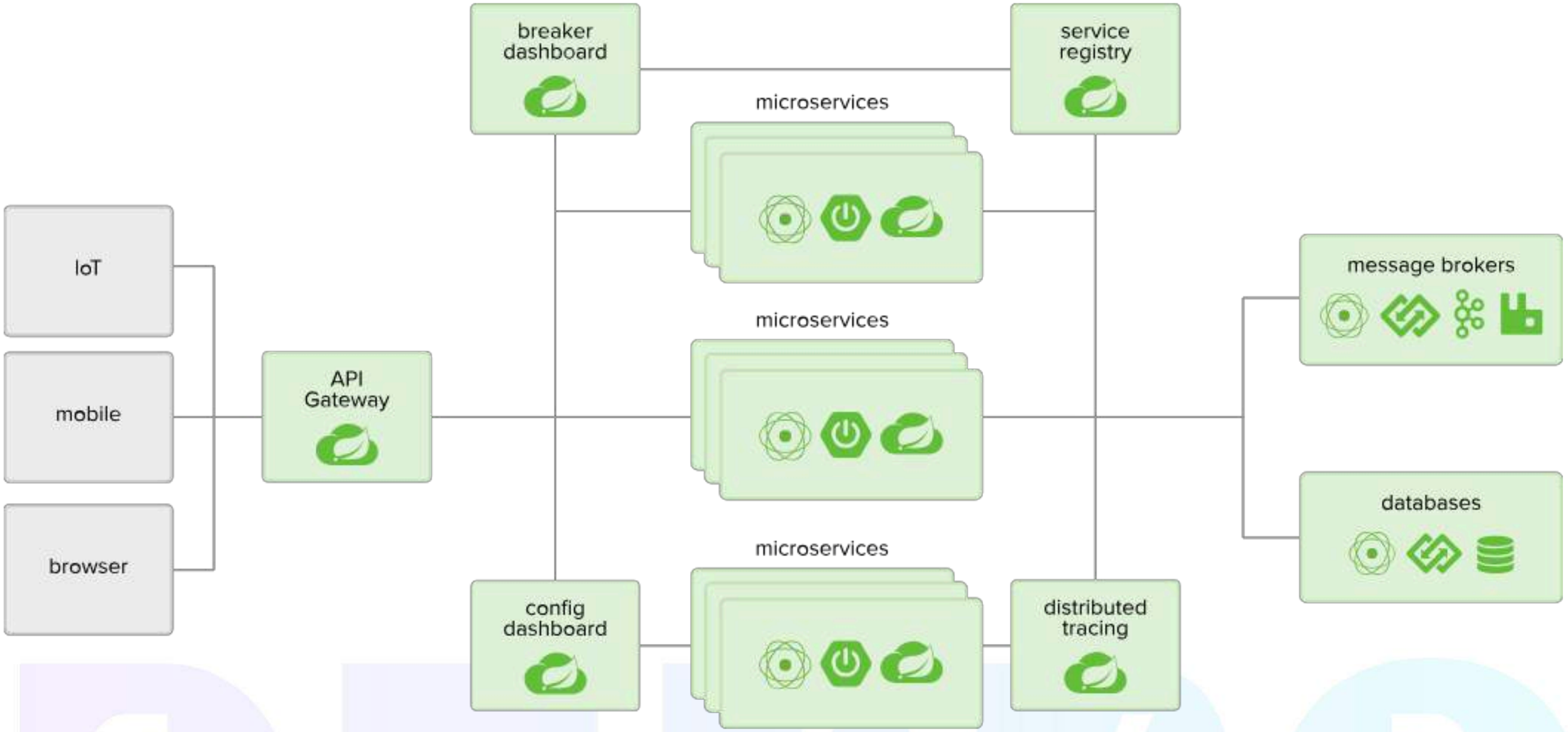
Coordinate Anything

分布式配置，服务注册与发现，服务调用，负载均衡，熔断，分布式消息

Spring Boot 简介



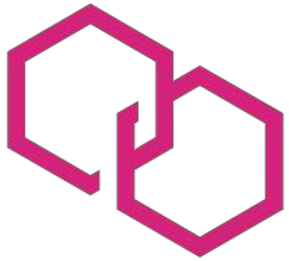
Spring Cloud 简介



Spring Cloud Alibaba 介绍

+

阿里云



阿里巴巴开源中间件以及阿里云服务
跟 Spring 技术栈的整合

引入 Starter 开箱即用

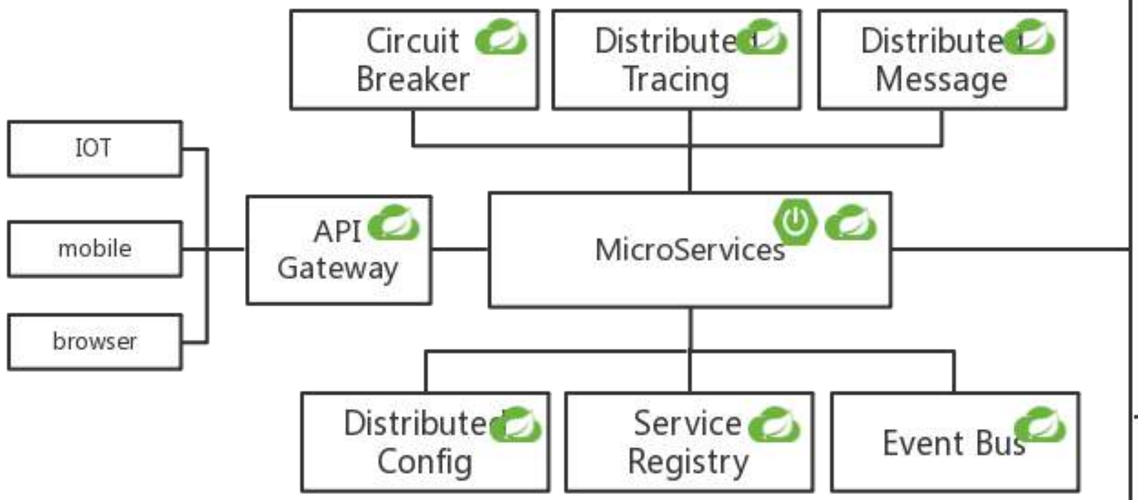
致力于成为微服务开发的一站式解决方案

Pivotal

Alibaba & Pivotal 合作项目(即将毕业)

https://github.com/spring-cloud-incubator/spring-cloud-alibaba

Spring Cloud Alibaba 体系



Spring Cloud Alibaba

- Spring Cloud Circuit Breaker with Sentinel
- Spring Cloud Dubbo
- Spring Cloud Stream with RocketMQ
- Spring Cloud Seata
- Spring Cloud Config with Nacos Config
- Spring Cloud Service Registry with Nacos Naming

Spring Cloud AliCloud

- EDAS
- OSS
- SchedulerX
- SMS

Spring Cloud Alibaba 组件

		 + 阿里云
分布式配置	Spring Cloud Config(依赖 Bus) Consul Zookeeper	**Nacos**
服务注册与发现	Eureka(维护模式) Consul Zookeeper	**Nacos**
服务熔断	Hystrix(维护模式)	**Sentinel**
服务调用	Open Feign RestTemplate	**Dubbo + RestTemplate** **Dubbo + Open Feign** **Dubbo**
分布式消息/消息总线	Kafka RabbitMQ	**RocketMQ**
分布式事务	无	**Seata**
存储	Spring Resource	**OSS**

Spring Cloud Alibaba 版本

Spring Cloud Alibaba	Spring Boot	Spring Cloud
0.9.X	2.1.X	Greenwich
0.2.X	2.0.X	Finchley
0.1.X	1.5.X	Edgware

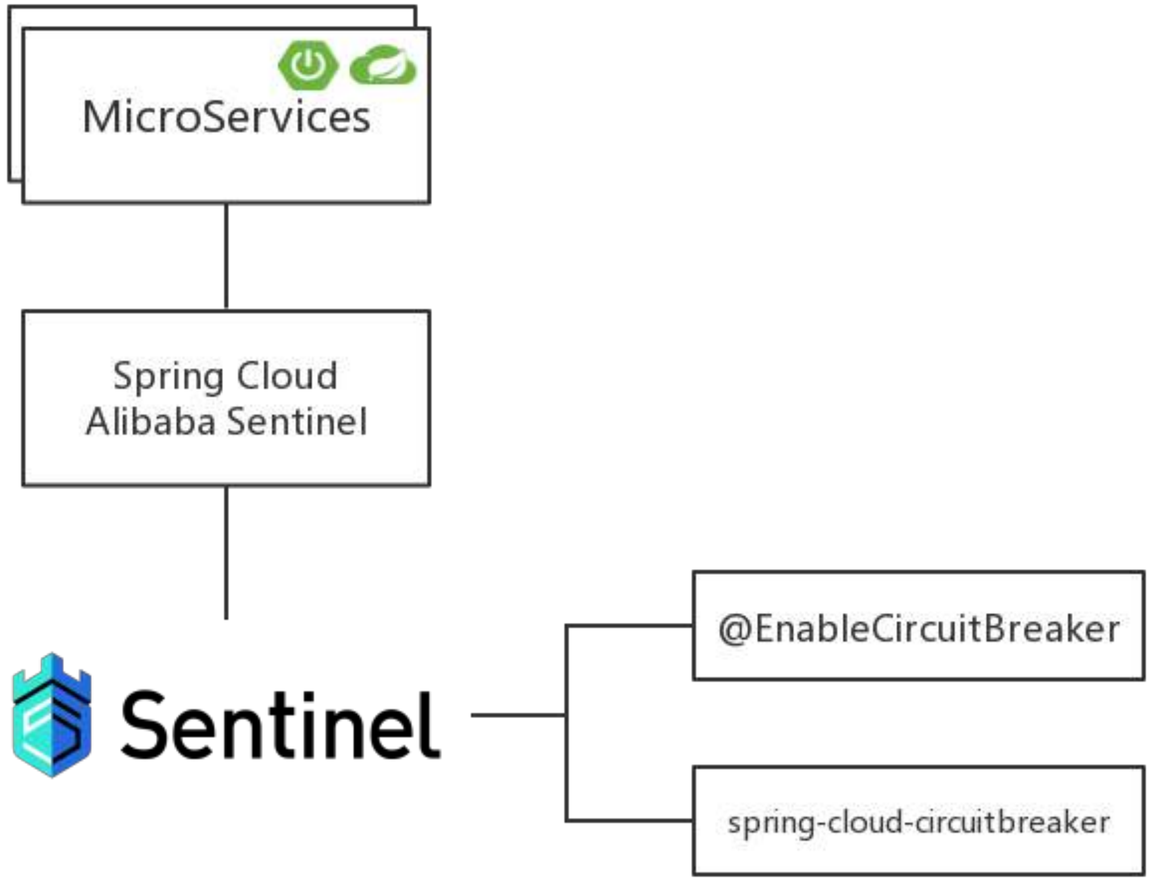
一定要使用正确的版本
Spring Boot/Cloud 版本兼容性比较差
没对应上可能会报错!

Central (7)

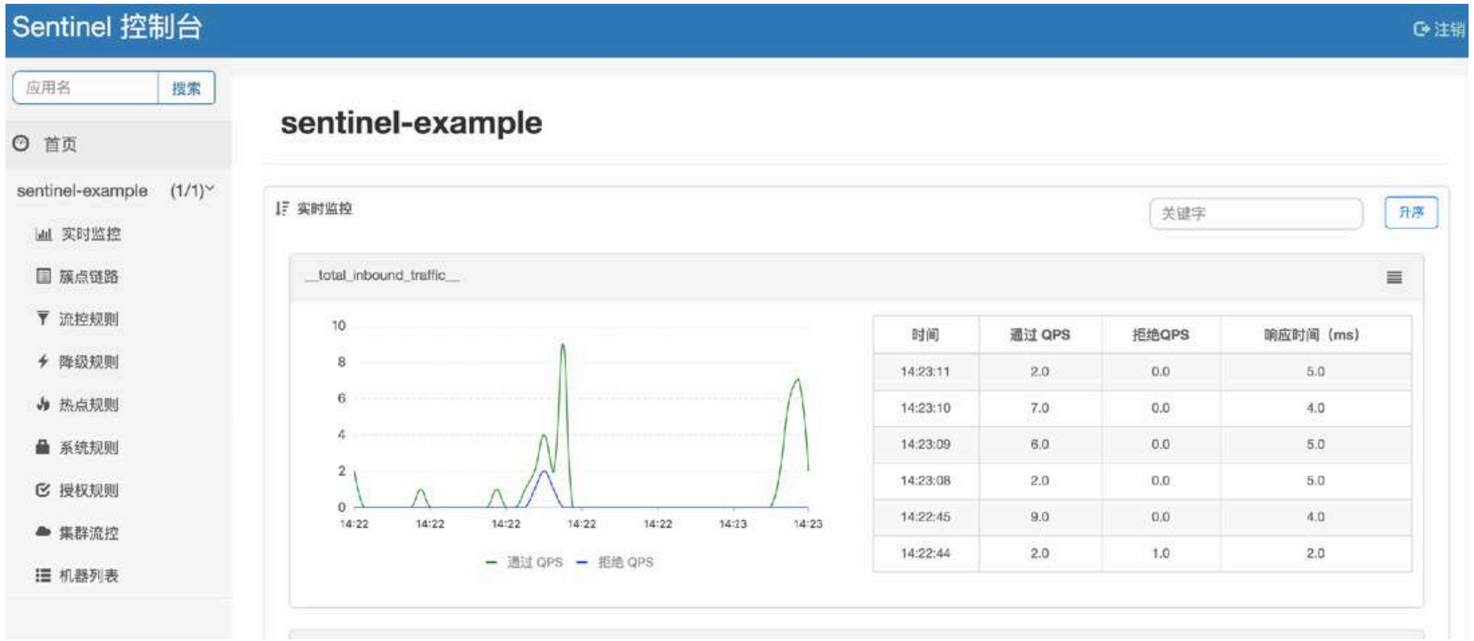
	Version
0.9.x	0.9.0.RELEASE
0.2.x	0.2.2.RELEASE
	0.2.1.RELEASE
	0.2.0.RELEASE
0.1.x	0.1.2.RELEASE
	0.1.1.RELEASE
	0.1.0.RELEASE

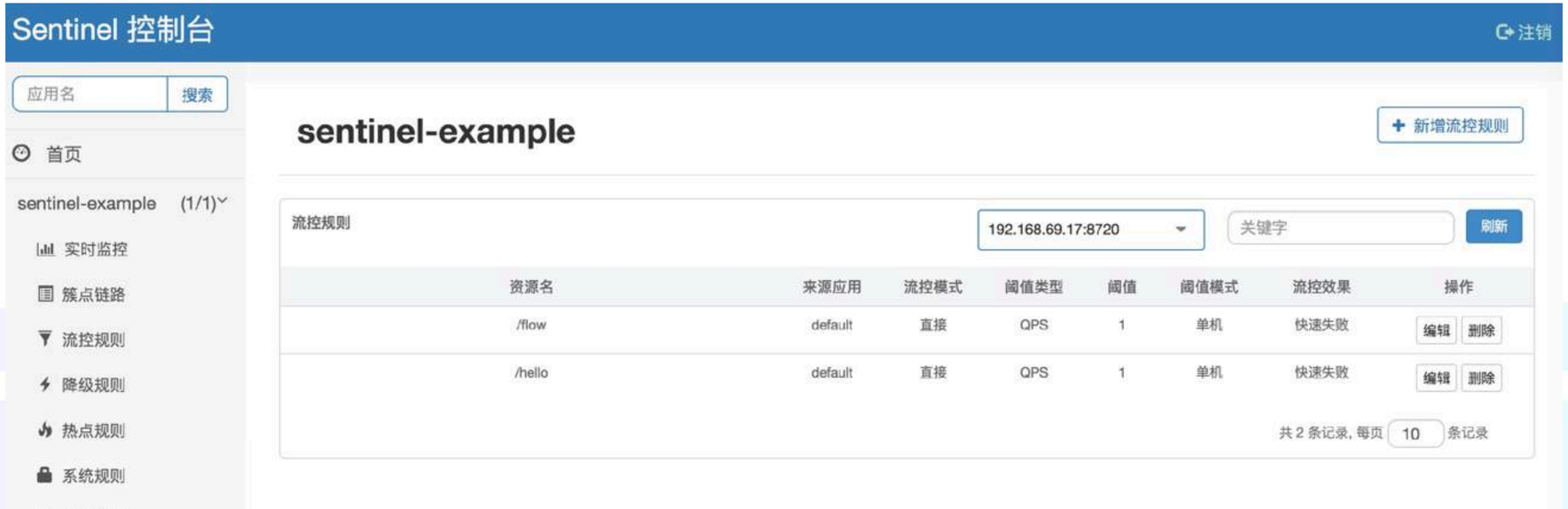
Spring Cloud Sentinel

- 限流 Flow Control
- 降级 Circuit Breaking
- 系统保护 System adaptive protection
- 实时监控 Real-time monitoring
- 支持 Spring Cloud Stack
  - WebServlet & WebFlux
  - FeignClient
  - RestTemplate
  - Zuul
  - Spring Cloud Gateway
- AutoConfiguration



Spring Cloud Sentinel





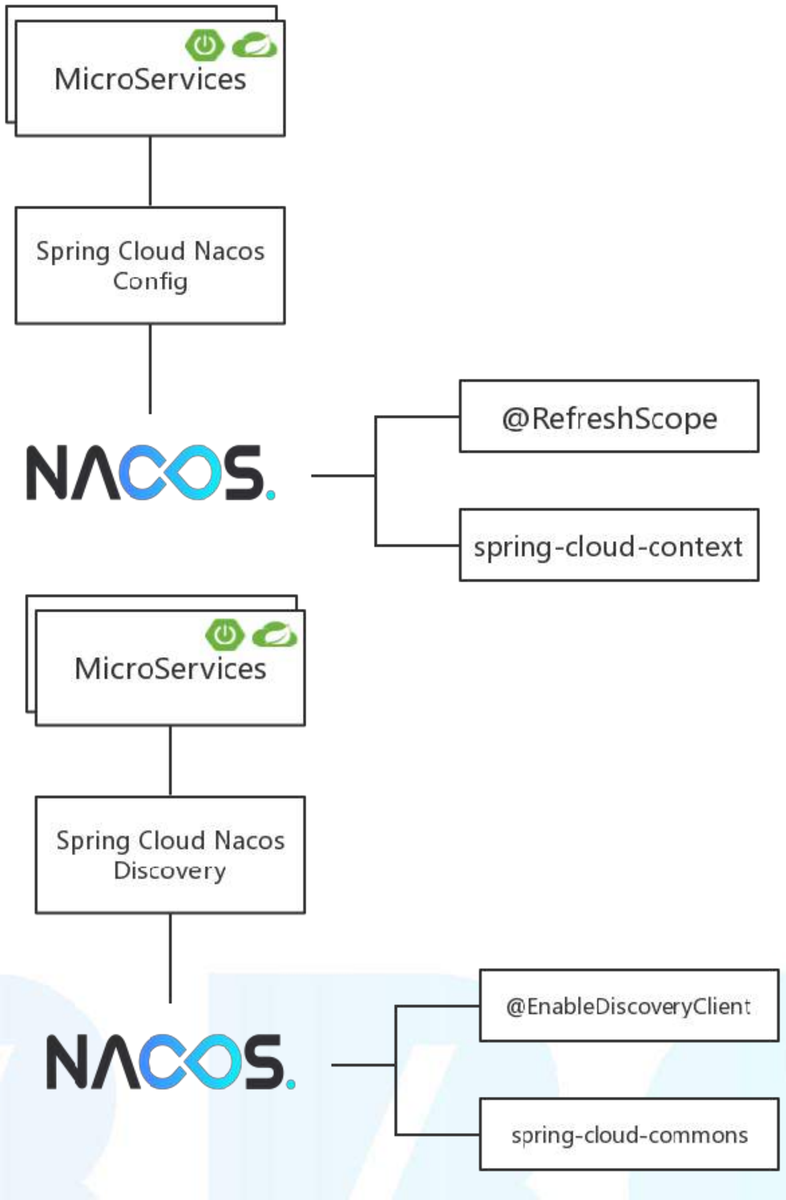
Spring Cloud Nacos Config & Discovery

Nacos Config

- 实时推送
- 推送状态可查
- 支持多种数据格式、支持中文
- 支持共享配置
- 支持历史配置查询、回滚
- 实现 spring-cloud-context 规范
- AutoConfiguration

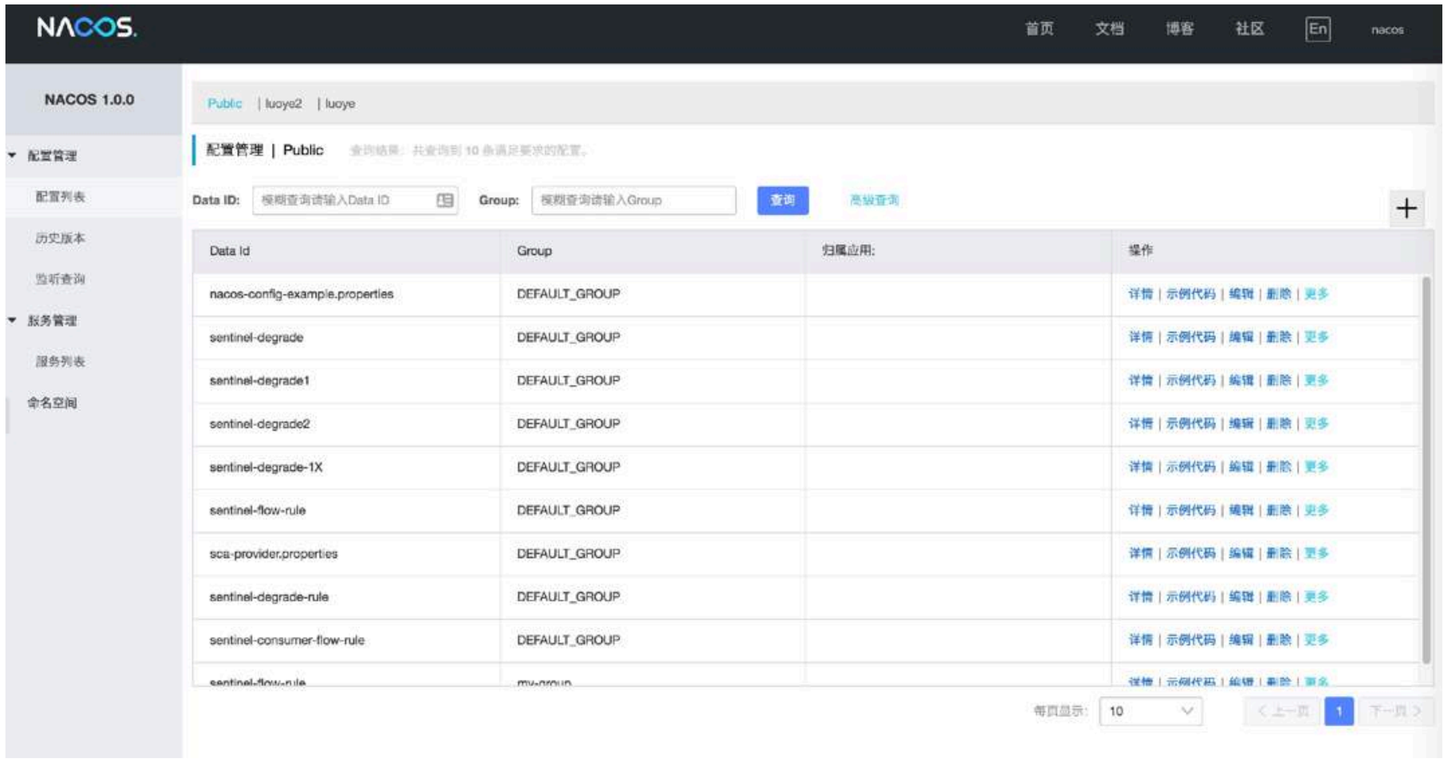
Nacos Discovery

- 实时发现
- 高可用，服务中心宕机不影响调用
- 性能强大
- 实现 spring-cloud-commons 规范
- AutoConfiguration

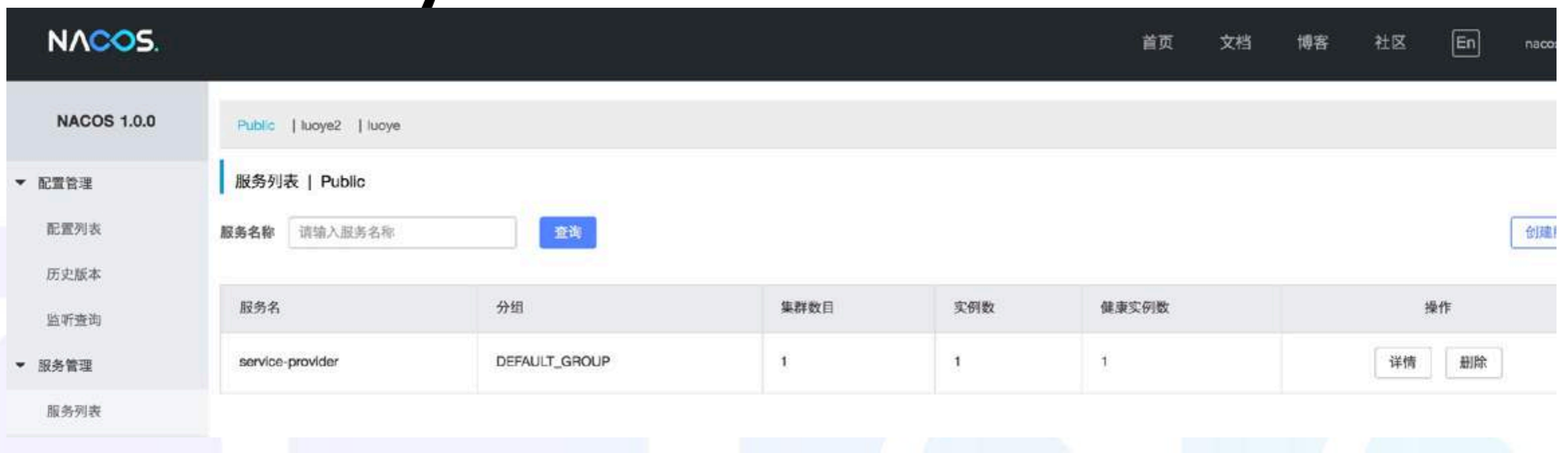


Spring Cloud Nacos Config & Discovery

Nacos Config



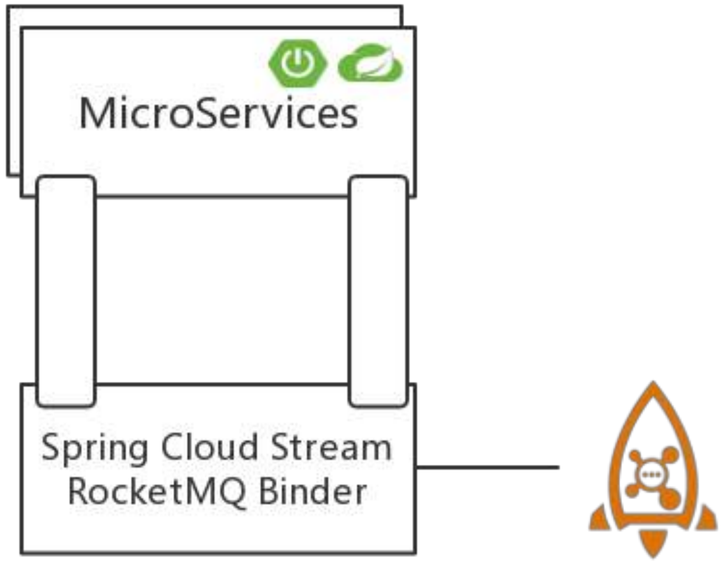
Nacos Discovery



Spring Cloud RocketMQ Binder & Bus

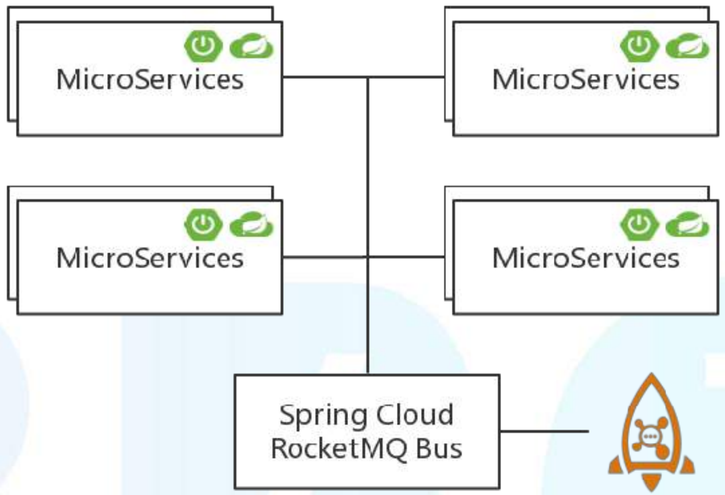
RocketMQ Binder

- 依赖 RocketMQ-Spring
- 支持 Polled Consumer 拉取消息
- @RocketMQTransactionListener 事务消息
- 额外的配置
  - Producer
  - Consumer
- AutoConfiguration



RocketMQ Bus

- 引入 starter 可发送远程消息
- 引入 starter 所有节点都可发送/接收远程消息



Spring Cloud RocketMQ Binder & Bus

RocketMQ Binder

```java
@Component
public class MessageReceiver {

    @StreamListener(value = Sink.INPUT, condition = "headers['type']=='user'")
    public void receiveUser(@Payload User user) { log.info("receiveUser: {}", user); }

    @StreamListener(value = Sink.INPUT, condition = "headers['type']=='string'")
    public void receiveString(String str) {
        log.info("receiveString: {}", str);
    }

    @StreamListener(value = Sink.INPUT)
    public void receiveAll(Message message,
                           @Header(value = "type", required = false) String type,
                           @Header(value = "test", required = false) String test) {
        log.info("receiveAll msgs, payload: {}, type header: {}, test header: {}",
            message.getPayload(), type, test);
    }

}
```

RocketMQ Bus

Node1

```java
@Component
public class EventReceiver {

    @Value("${spring.application.name}")
    private String applicationName;

    @Value("${server.port}")
    private Long serverPort;

    @EventListener(classes = EchoRemoteEvent.class)
    public void receiveEvent(EchoRemoteEvent echoRemoteEvent) {
        log.info("{}-{} get remoteEvent: {}",
            applicationName, serverPort, echoRemoteEvent.getEchoMessage());
    }

}
```

Node2

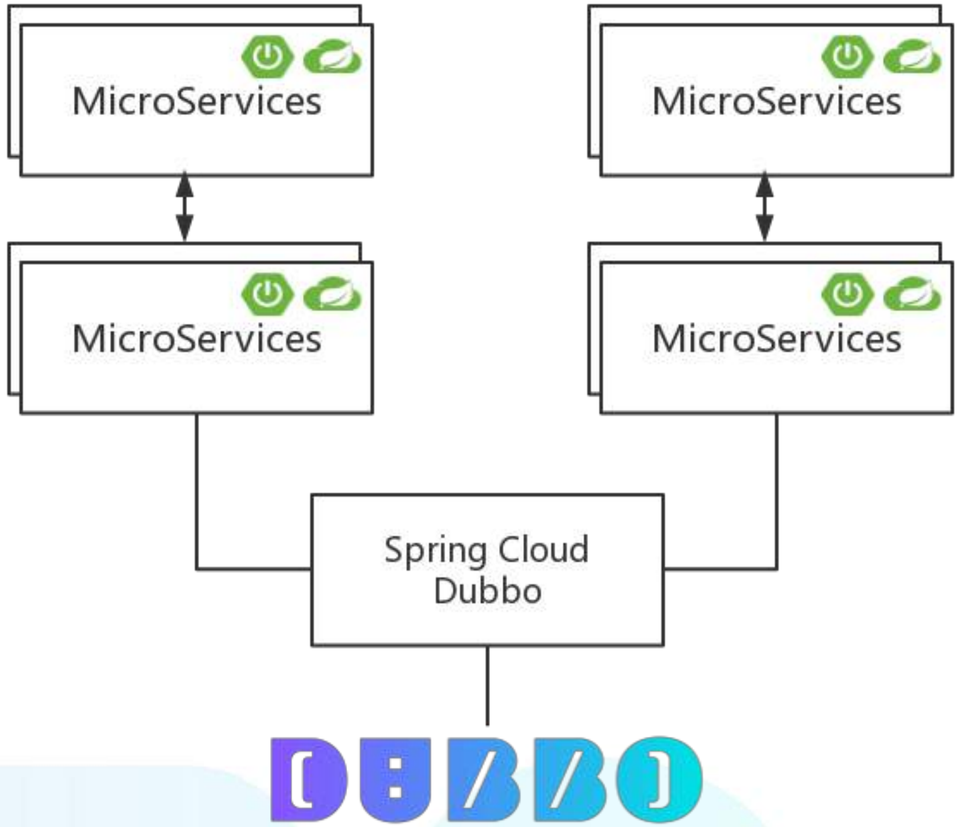
```java
@Component
public class EventReceiver {

    @Value("${spring.application.name}")
    private String applicationName;

    @Value("${server.port}")
    private Long serverPort;

    @EventListener(classes = EchoRemoteEvent.class)
    public void receiveEvent(EchoRemoteEvent echoRemoteEvent) {
        log.info("{}-{} get remoteEvent: {}",
            applicationName, serverPort, echoRemoteEvent.getEchoMessage());
    }

}
```

Spring Cloud Dubbo

- Spring Cloud 注册中心: spring-cloud://localhost
- 作为 Spring Cloud 服务调用的第三种选择

```java
@FeignClient("rest-service")
@DubboTransported(protocol = "dubbo")
public interface DubboFeignRestService {

    @GetMapping(value = "/param")
    String param(@RequestParam("param") String param);

    @PostMapping("/params")
    String params(@RequestParam("b") String paramB, @RequestParam("a") int paramA);

}
```

```java
@Bean
@LoadBalanced
@DubboTransported
public RestTemplate restTemplate() {
    return new RestTemplate();
}
```



Spring Cloud Dubbo

Dubbo & SpringMVC

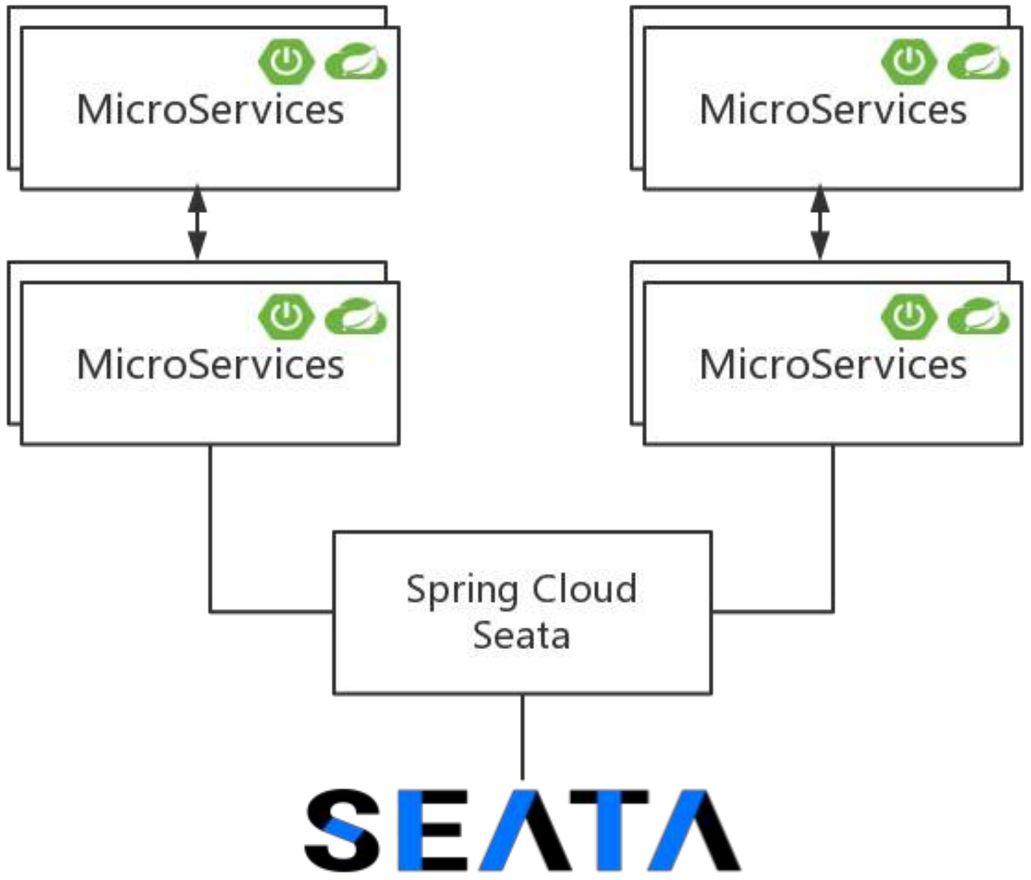
```java
@Service(version = "1.0.0", protocol = "dubbo")
@RestController
public class SpringRestService implements RestService {

    private static final Logger log = LoggerFactory
        .getLogger(SpringRestService.class);

    @Override
    @GetMapping("/param")
    public String param(@RequestParam String param) {
        log.info("param: " + param);
        return "param: " + param;
    }

    @Override
    @PostMapping("/params")
    public String params(@RequestParam int a, @RequestParam String b) {
        log.info("params: " + a + ", " + b);
        return "params: " + a + ", " + b;
    }

}
```

Dubbo & JAX-RS

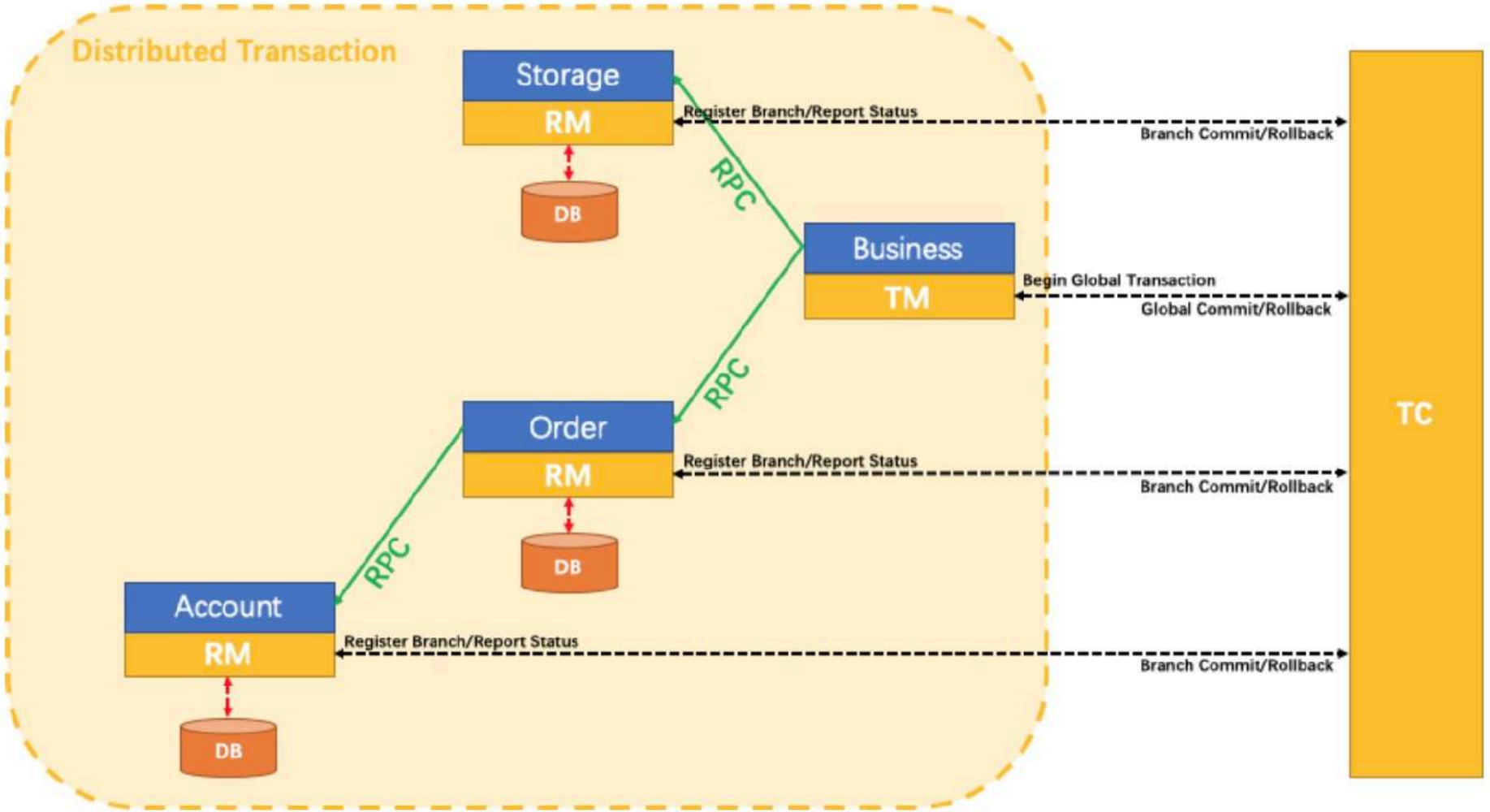
```java
@Service(version = "1.0.0", protocol = {"dubbo", "rest"})
@Path("/")
public class StandardRestService implements RestService {

    private Logger logger = LoggerFactory.getLogger(getClass());

    @Override
    @Path("param")
    @GET
    public String param(@QueryParam("param") String param) {
        log( url: "/param", param);
        return param;
    }

    @Override
    @Path("params")
    @POST
    public String params(@QueryParam("a") int a, @QueryParam("b") Strin
        log( url: "/params",  result: a + b);
        return a + b;
    }

}
```

Spring Cloud Seata

- 适配 AT 模式
- WebServlet 环境自动还原 Seata 上下文
- RestTemplate 和 Open Feign 自动传递 Seata 上下文



Spring Cloud Seata



Demo – Spring Cloud Alibaba 微服务应用开发

全家桶体验

- Spring Cloud Alibaba: https://github.com/spring-cloud-incubator/spring-cloud-alibaba

- 扫码加入 Spring Cloud Alibaba 钉钉交流群

第五届
中间件性能挑战赛

挑战双11零点流量洪峰

阿里云

扫描二维码立即参赛

邀请新人参赛还有惊喜奖品

Thank you !