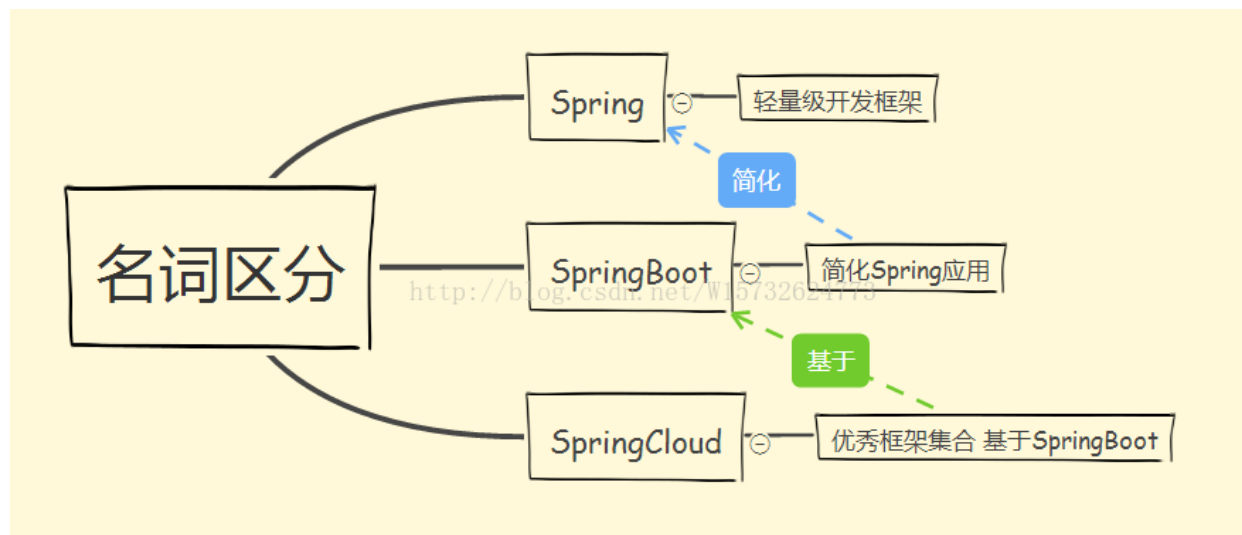


# SpringCloud之一

## 说在前面

相信很多人学习SpringCloud的时候都会想Spring、SpringBoot、SpringCloud的关系，如下图所示，希望有助于理解：



- **Spring**是一个轻量级的Java开发框架，它能使用基本的JavaBean代替EJB。
- **SpringBoot**是由Pivotal团队提供的全新框架，用来简化新Spring应用的初始搭建和开发过程。开发人员无需定义样板化配置。
- **SpringCloud**是一系列框架的有序集合，它把好的东西集合到一起，这就是所谓的集大成者。同时它利用SpringBoot的开发便利性巧妙的简化了分布式系统基础设施的开发。

## 一、spring cloud简介

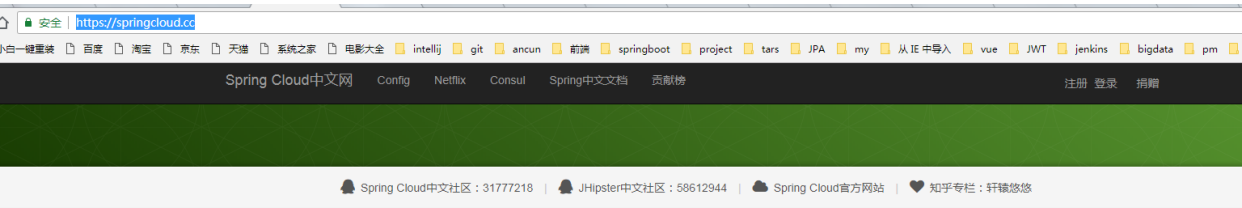
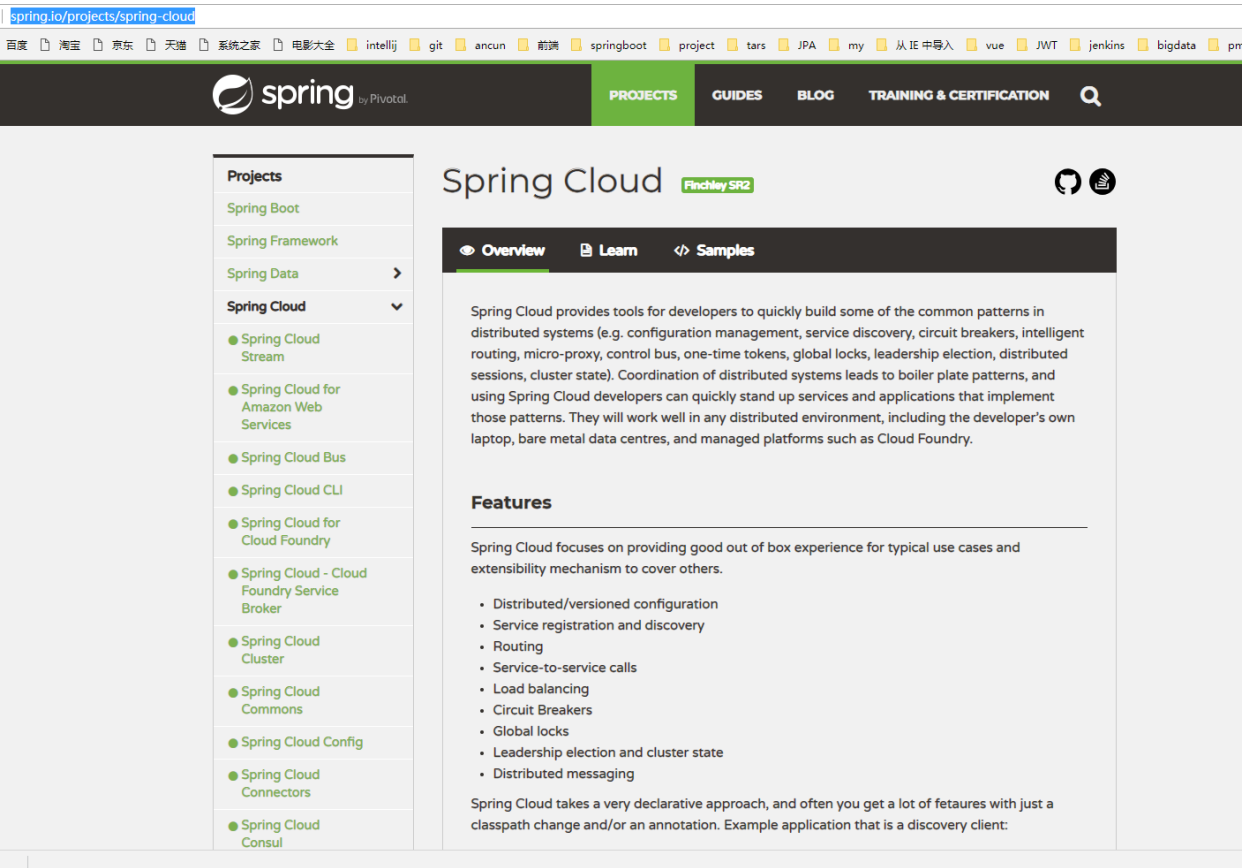
目前支持的版本为Spring Boot版本2.0.3.RELEASE, Spring Cloud版本为Finchley.RELEASE。

Finchley版本的官方文档如下：

<http://cloud.spring.io/spring-cloud-static/Finchley.RELEASE/single/spring-cloud.html>

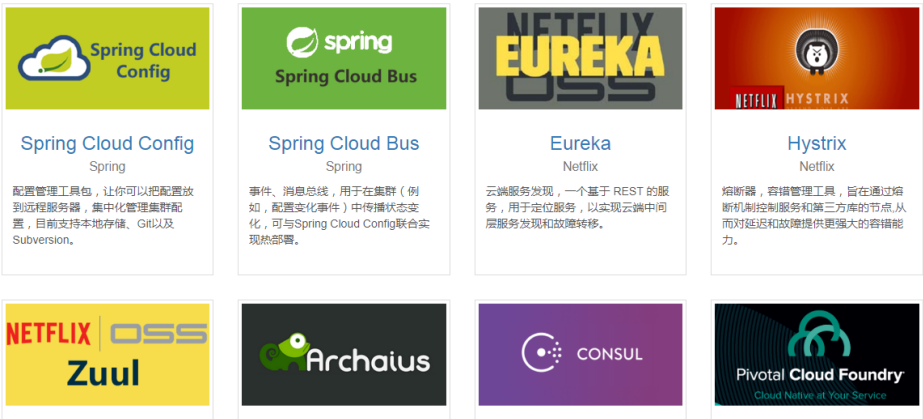
spring cloud 为开发人员提供了快速构建分布式系统的一些工具，包括配置管理、服务发现、断路器、路由、微代理、事件总线、全局锁、决策竞选、分布式会话等等。它运行环境简单，可以在开发人员的电脑上跑。另外说明spring cloud是基于springboot的，所以需要开发中对springboot有一定的了解，如果不了解的话可以查看springBoot官网或其他论坛。另外对于“微服务架构”不了解的话，可以通过搜索引擎搜索“微服务架构”了解下。

SpringCloud官网界面如下：



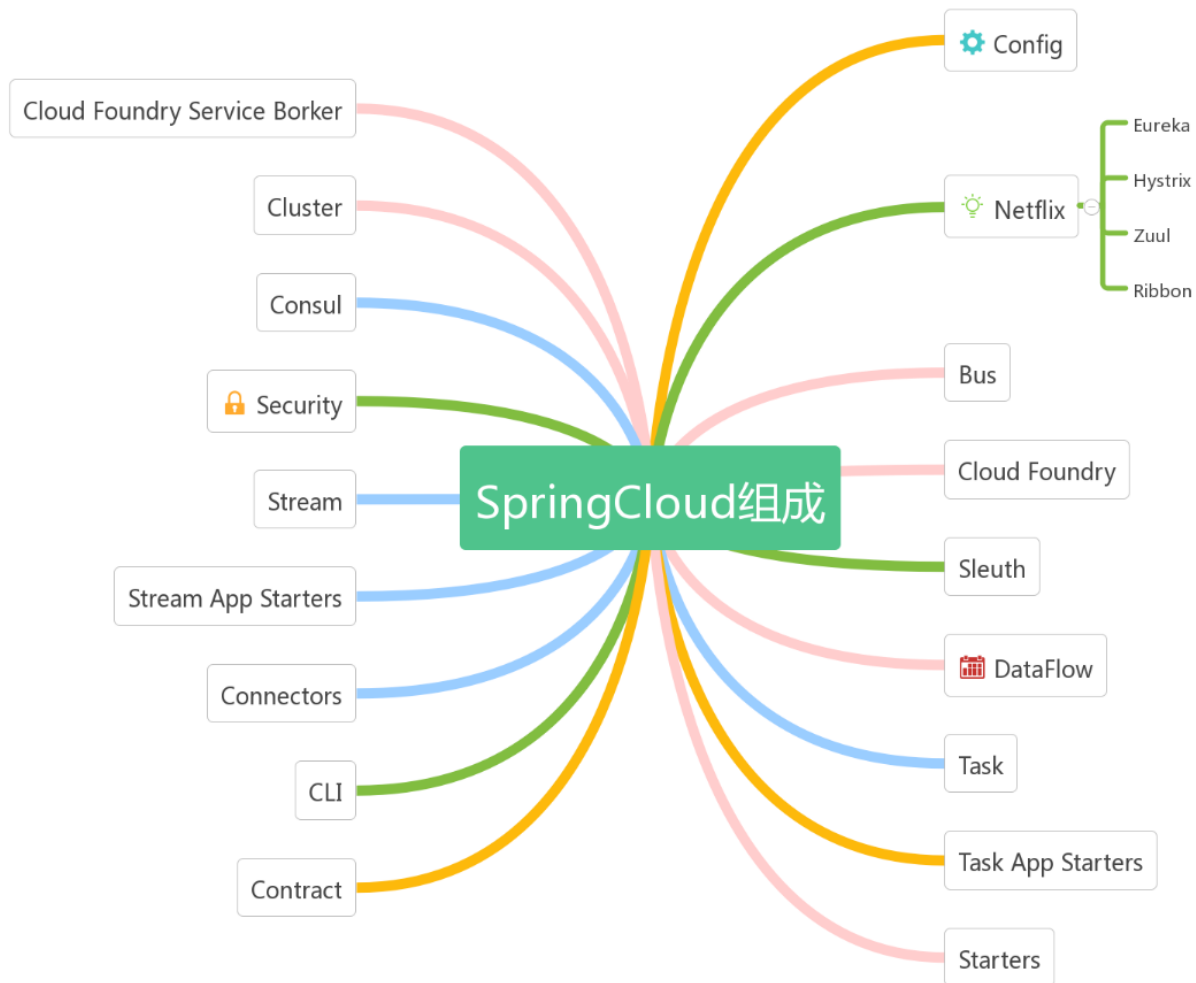
## Spring Cloud集成相关优质项目推荐

这些项目是Spring Cloud官方项目或是对Spring Cloud进行了有益的补充以及基于Spring Cloud最佳实践。



## 二、SpringCloud生态圈

参考官网，列举如下：

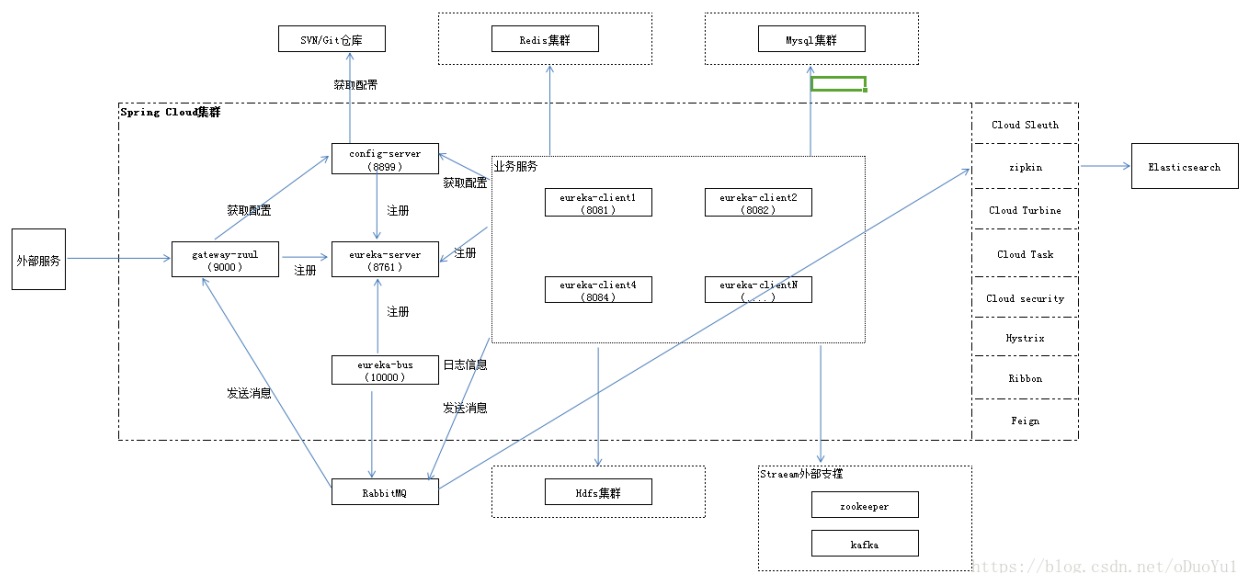
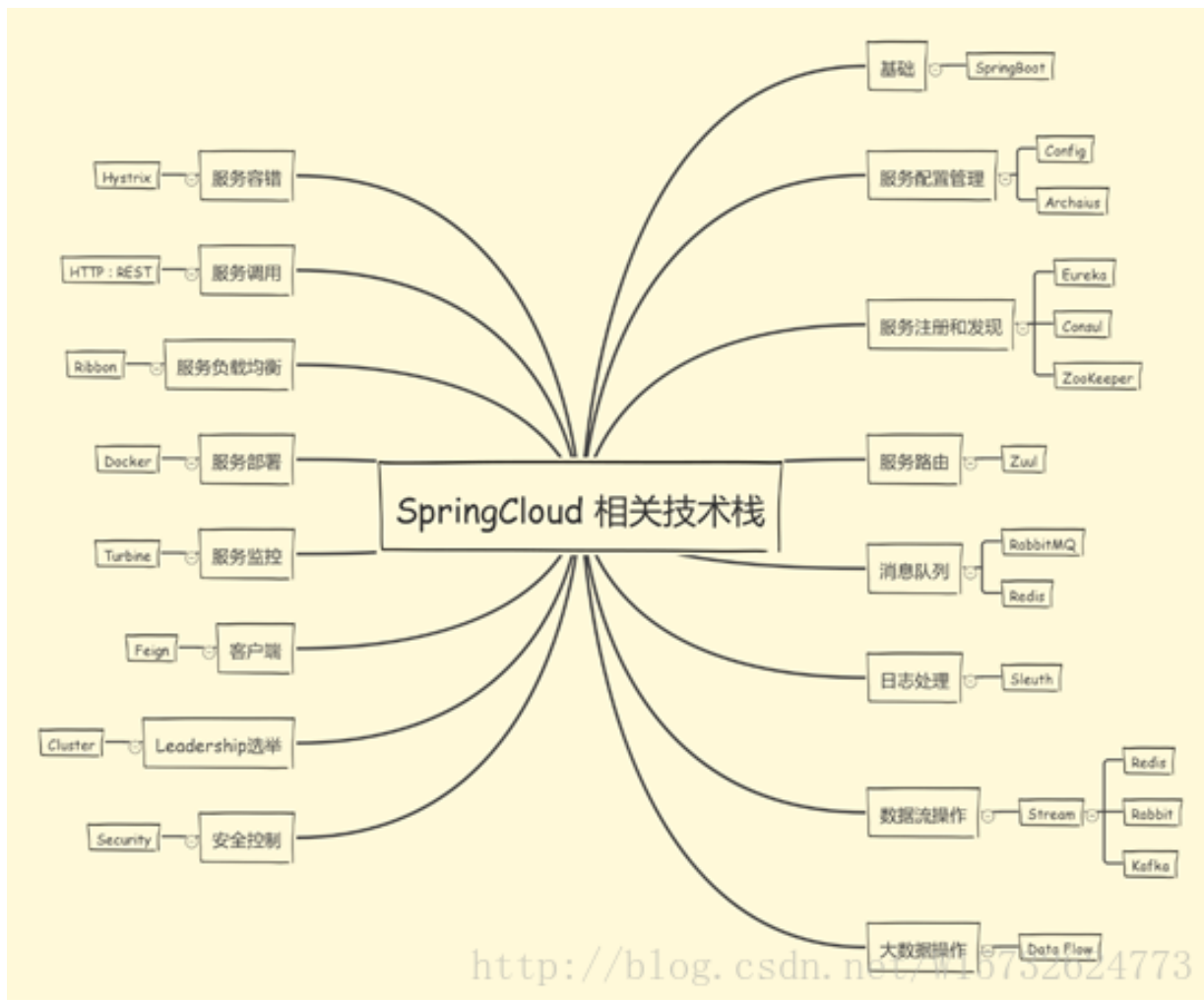


## 常用项目简介：

- 1. Spring Cloud Config：**配置管理工具包，可以将配置放到远程服务器，集中化管理集群配置，目前支持本地存储、Git以及Subversion。
- 2. Spring Cloud Bus：**事件、消息总线，用于在集群（例如，配置变化事件）中传播状态变化，官方集成了RabbitMQ，可与Spring Cloud Config联合实现热部署。
- 3. Spring Cloud Netflix：**多种Netflix组件提供的开发工具包，其中包括Eureka、Hystrix、Zuul、Ribbon等。
- 4. Spring Cloud Netflix?Eureka：**这是 Spring Cloud Netflix 微服务套件中的一部分，主要负责完成微服务架构中的服务治理功能，包括服务注册中心、服务注册与服务发现机制的实现，同时实现负载均衡和中间层服务器的故障转移。**Eureka** 云端服务发现，一个基于 REST 的服务，用于定位服务，以实现云端中间层服务发现和故障转移。
- 5. Spring Cloud Netflix Hystrix：**熔断器，容错管理工具，旨在通过熔断机制控制服务和第三方库的节点,从而对延迟和故障提供更强大的容错能力。

- 6. Spring Cloud Netflix Zuul** : 网关, 提供动态路由、监控、弹性、安全等边缘服务的框架。Zuul 相当于是设备和 Netflix 流应用的 Web 网站后端所有请求的前门。
- 7. Spring Cloud Netflix Ribbon** : 提供客户端的负载均衡, 有多种负载均衡策略可供选择, 可配合服务发现和断路器使用。
- 8. Spring Cloud Netflix?Turbine** : 聚合服务器发送事件流数据的一个工具, 用来监控集群下hystrix的metrics情况。
- 9. Spring Cloud Netflix?Archaius** : 配置管理API, 包含一系列配置管理API, 提供动态类型化属性、线程安全配置操作、轮询框架、回调机制等功能。
- 10. Spring Cloud?Feign** : 声明式、模板化的服务调用组件。
- 11. Spring Cloud Sleuth** : 日志收集工具包, 封装了Dapper和log-based追踪以及Zipkin和HTrace操作, 为SpringCloud应用实现了一种分布式追踪解决方案。
- 12. Spring Cloud Security** : 基于spring security的安全工具包, 提供应用安全控制。
- 13. Spring Cloud Zookeeper** : 操作Zookeeper的工具包, 用于使用zookeeper方式的服务发现和配置管理。
- 14. Spring Cloud Stream** : 数据流操作开发包, 封装了与Redis,Rabbit、Kafka等发送接收消息。
- 15. Spring Cloud CLI** : 基于 Spring Boot CLI, 可以命令行方式快速建立云组件。
- 16. Spring Cloud Task** : 提供计划任务管理、任务调度。

下面是网上看到的SpringCloud的技术栈图, 希望能有帮助:



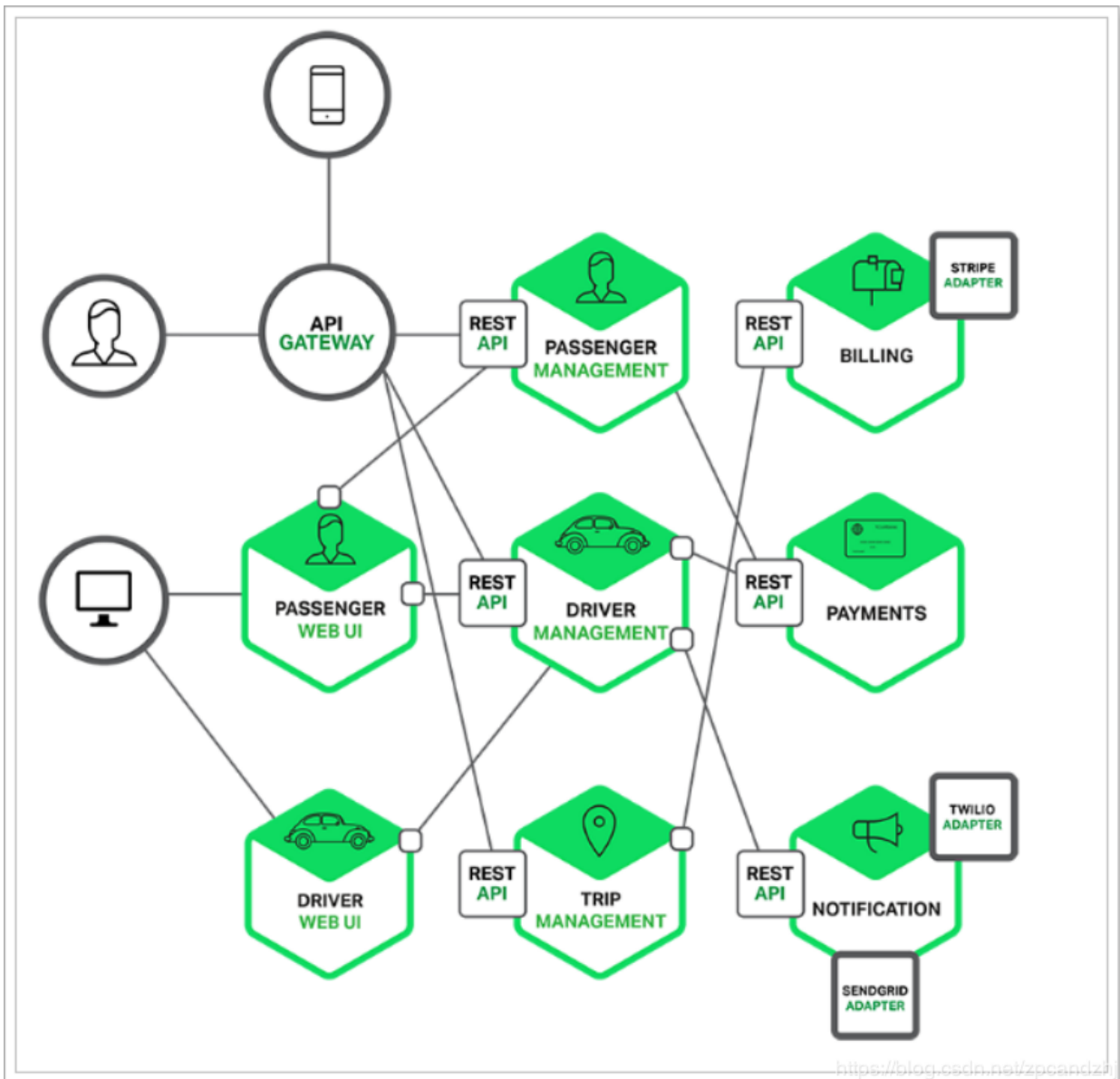
附带：

关于SpringCloud不得不说一下微服务，

- 每个微服务可独立运行在自己的进程里。
- 一系列独立运行的微服务共同构建起整个系统。
- 每个服务为独立的业务开发，一个微服务只关注某个特定的功能，例如订单管理、用户管理等。
- 微服务之间通过一些轻量的通信机制进行通信，例如通过 RESTful API 进行调用。
- 可以使用不同的语言与数据存储技术。
- 全自动的部署机制。

<https://blog.csdn.net/zpcandzj>

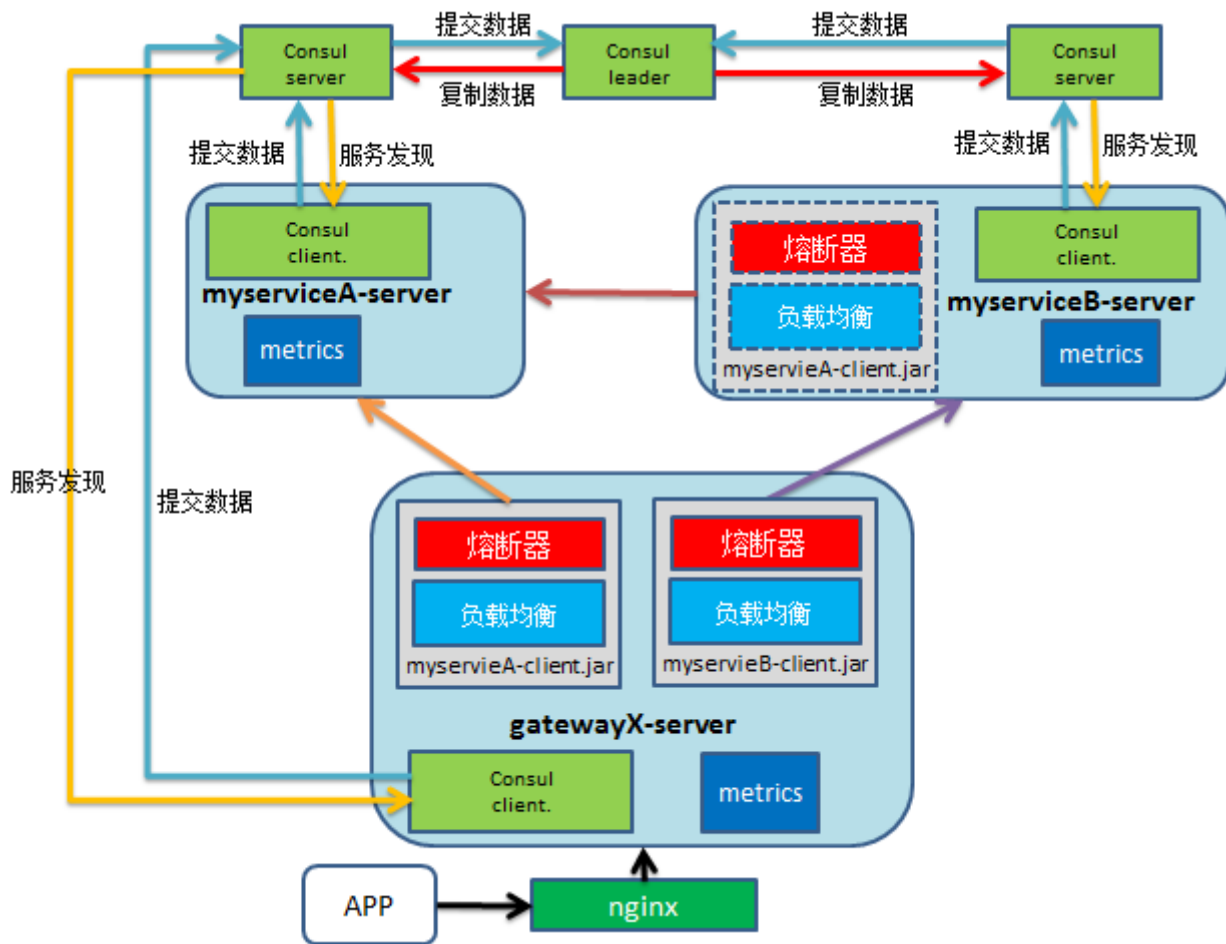
如图示例：每一个应用功能区都使用微服务完成。



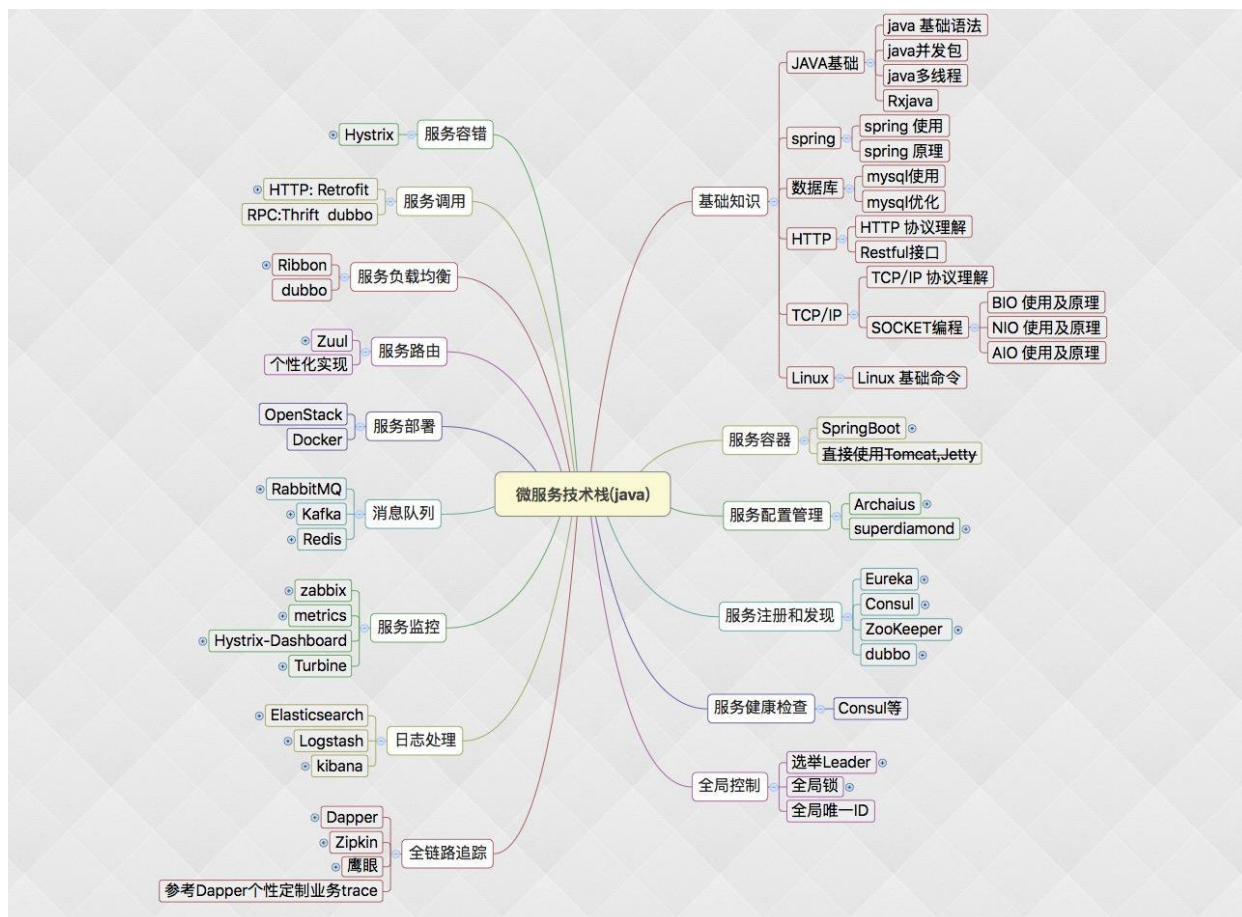
<https://blog.csdn.net/zpcandzj>

以下是springCloud的微服务架构图：





技术选型如下图：



## 参考资料

SpringCloud 官网：<http://spring.io/projects/spring-cloud>

SpringCloud 开源代码：<https://github.com/spring-cloud>

Spring Cloud技术栈简述：<https://blog.csdn.net/oDuoYu1/article/details/82494321>

Spring Cloud 中文官方文档：<https://springcloud.cc/>

微服务之springcloud技术栈：

<https://blog.csdn.net/xiangxizhishi/article/details/79434775>

Spring Cloud教程1：

[https://blog.csdn.net/hellozpc/article/details/83692496#96fallback\\_2064](https://blog.csdn.net/hellozpc/article/details/83692496#96fallback_2064)

<https://blog.csdn.net/tudou201601/article/details/79123912>