# 微服务框架技术选型

## 微服务框架常见使用场景

微服务是一种软件架构风格，它将一个大型应用程序拆分成一组小型、独立的服务，每个服务都可以独立部署、扩展和维护。每个服务都有自己的业务逻辑和数据存储，它们通过轻量级的通信机制（如REST API）相互通信。

对于企业来说，微服务架构可以带来以下价值：

1. 更快的应用程序开发和部署：由于每个微服务都是独立的，开发人员可以同时工作，加快了应用程序的开发和部署速度。
2. 更高的可扩展性和可维护性：微服务架构可以让企业更容易地扩展和维护应用程序，因为每个微服务都是独立的，可以单独扩展或维护，而不会影响整个应用程序。
3. 更好的容错性和可靠性：微服务架构可以让企业更容易地处理故障，因为每个微服务都是独立的，可以在一个服务出现故障时继续运行其他服务。
4. 更好的安全性：微服务架构可以让企业更容易地实现安全措施，因为每个微服务都可以有自己的安全措施，如身份验证、授权等。
5. 技术多样性：微服务架构可以让企业使用不同的技术栈和编程语言来开发不同的服务，这可以根据不同的需求和场景来选择最合适的技术。

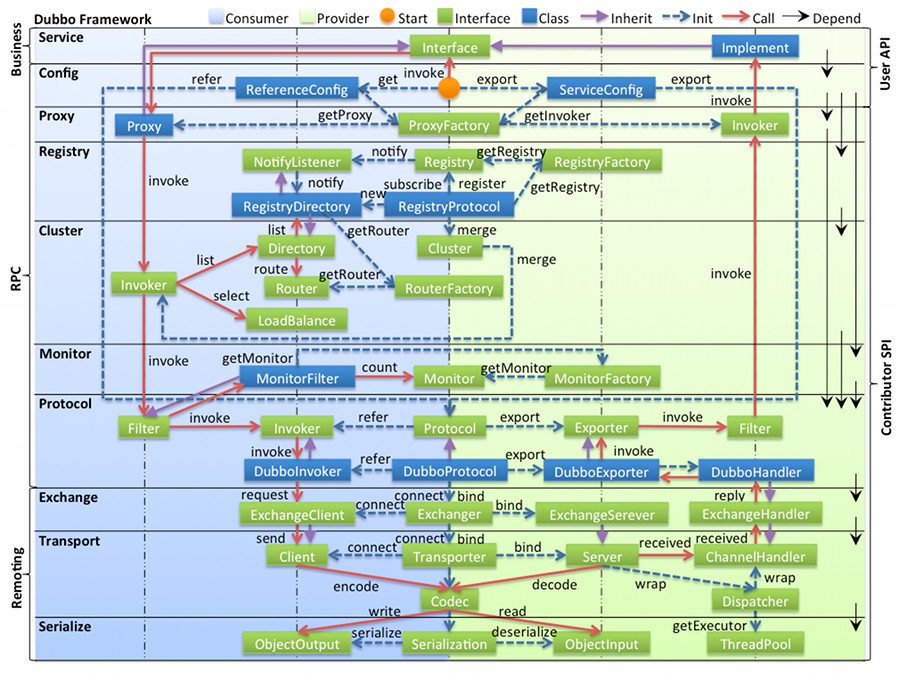
总之，微服务架构可以帮助企业更快、更容易、更可靠地开发和维护应用程序，提高业务效率和客户满意度。

微服务技术体系目前已经是大家开发分布式系统的首选方案，但是常见的微服务框架都是通用的各类单独技术工具，如何整合这些不同的框架，形成一套完善的微服务技术平台，并且有效的支持银行复杂的业务落地，是微服务技术选型的最重要考量点。

## 主流微服务框架

1. **Dubbo**

1）架构图

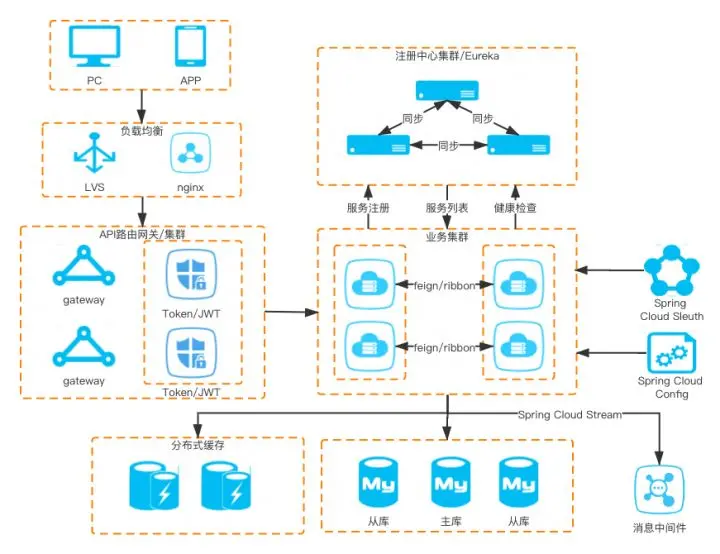


1. 核心功能介绍

[Dubbo](https://so.csdn.net/so/search?q=Dubbo&spm=1001.2101.3001.7020" \t "https://blog.csdn.net/ChenRui_yz/article/details/_blank)是一款Java RPC框架，致力于提供高性能的RPC远程服务调用方案。Dubbo 作为主流的微服务框架之一，为开发人员带来了非常多的便利。

* 高性能RPC调用：提供高性能的基于代理的远程调用能力，为开发者屏蔽远程调用底层细节
* 智能负载均衡：内置多种负载均衡策略，智能感知下游节点健康状况，显著减少调用延迟，提高系统吞吐量
* 服务自动注册与发现：支持多种注册中心服务、服务实例上下线实时感知
* 高度可扩展能力：遵循微内核+插件的设计原则，所有核心能力均被设计为扩展点，平等对待内置实现和三方实现
* 运行期流量调度：内置条件、脚本等路由策略，通过配置不同的路由规则轻松实现灰度发布、同机房优先等功能
* 可视化的服务治理与运维：提供丰富的服务治理、运维工具，随时查询服务元数据、服务健康状态及调用统计，实时下发路由策略，调整配置参数

1. **SpringCloud**
2. 架构图



1. 组件介绍

Spring Cloud是一种最常见的分布式系统模式，为开发者提供了一种简单并且易于接受的编程模型，帮助开发人员构建有弹性的、可靠的、协调的程序软件，轻松实现微服务项目的构建。

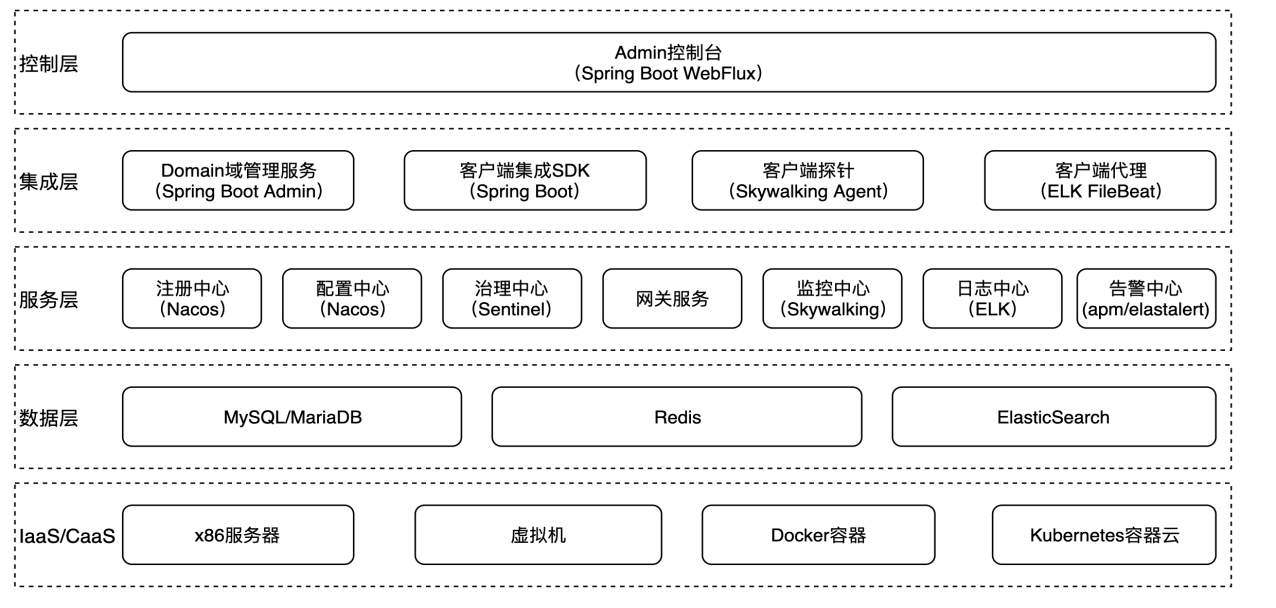
* SpringCloudGateway：网关路由服务，提供代理访问转发，保证服务安全
* SpringCloudEureka：注册中心服务，提供服务注册与发现能力
* SpringCloudFeign：服务代理，提供代理调用服务
* SpringCloudRibbon：负载均衡器，提供多种负载均衡算法
* SpringCloudConfig：服务配置，提供统一配置功能
* SpringCloudSleuth：分布式链路追踪，将一次分布式请求还原成调用链路，进行日志记录，性能监控并将一次分布式请求的调用情况集中展示。
* SpringCloudStream：为微服务应用构建消息驱动能力的框架

1. **EDSP（长亮科技企业级分布式服务平台）**
2. 核心功能介绍

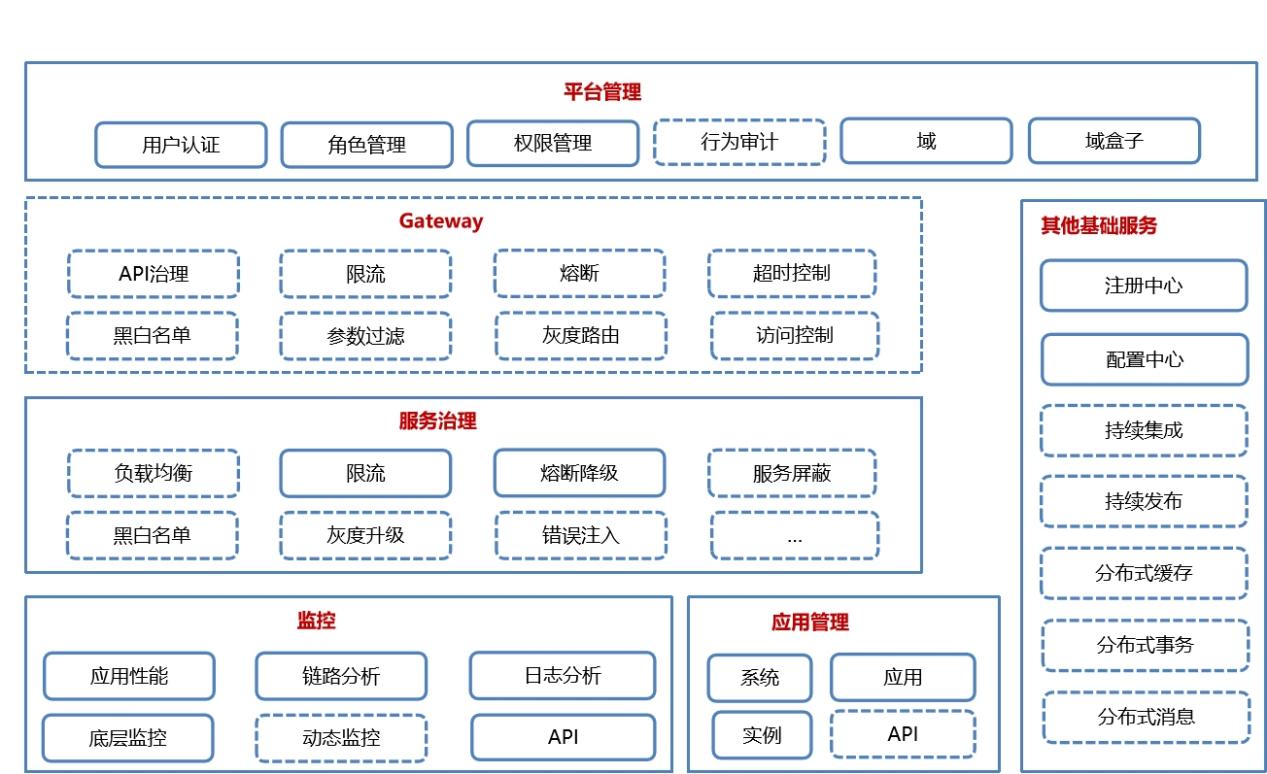
长亮科技微服务框架（EDSP）是一个围绕微服务应用的gPaaS平台，提供应用全生命周期管理、数字化运维、立体化监控、细粒度服务治理等功能；拥抱Spring Boot和Spring Cloud开源社区，帮助企业客户解决传统集中式架构转型的困难，打造大规模高可用的金融级分布式系统架构，实现业务和产品的快速落地。

* 拥抱开源：拥抱Spring Boot/Spring Cloud开源社区，提供高可用、可扩展、灵活的微服务技术中台架构支持，降低用户迁移到微服务架构的时间和人力成本。
* 全生命周期管理：提供从创建应用到运行应用的全生命周期管理，支持创建、部署、扩容、回滚、下线、启动和停止，支持虚拟机和容器两种模式，满足不同的客户使用需求。
* 细粒度服务治理：提供服务和API级别的服务治理能力，支持控制台上配置服务路由、限流策略、鉴权规则等，支持分布式配置管理。
* 全面的基础组件：集成了注册中心、配置中心、治理中心、日志中心、链路中心、分布式缓存、分布式消息队列、分布式锁、流量网关、治理网关等微服务架构所需的基础组件，并对应用层保持透明性，解决企业级架构管理问题。
* 全方位运维管理：支持应用和组件级监控告警、应用和组件级日志分析、交易链路和依赖拓扑图等多种维度的运维管理，满足不同维度的运维需求。
* 两地三中心：支持同城跨可用域容灾和就近路由，规避单可用域可能存在的不可抗力风险，提高服务的可用性和容灾能力。

1. 组件架构



1. 功能架构



## 各微服务选型对比

|  |  |  |  |
| --- | --- | --- | --- |
| **核心要素** | **Dubbo** | **SpringCloud** | **EDSP** |
| 开发难度 | 高 | 低 | 低 |
| 开源 | 开源 | 开源 | 可交付源码、可定制化、满足自主可控 |
| 成熟度 | 高 | 高 | 高 |
| 权限控制 | 无 | 无 | 完善的用户权限体系、细粒度权限控制、操作审计 |
| 通讯协议 | RPC | HTTP/REST | HTTP/REST |
| 服务网关 | 无 | Zuul/Gateway | edsp-gateway |
| 观测性 | 无，一般需要手工集成第三方工具 | Sleuth集成Zipkin进行链路展示 | 基于skywalking增强，定制化链路、性能分析，与日志平台高度联动 |
| 统一管控 | 管控功能较弱 | 管控界面松散，体验性差 | 统一管控，体验性强 |
| 银行案例 | 邮储银行深度定制 | 未知 | 交通银行信用卡核心、兴业银行、渤海银行、民生银行，邮储直销银行 |
| 售后服务 | 企业运维团队支持 | 企业运维团队支持 | 提供稳定及时的售后服务，强大的技术、运维团队支持 |
| 技术成熟度 | 成熟 | 成熟 | 成熟/完全兼容SpringCloud体系 |

## 微服务框架选型建议

基于以上分析对比，常见的各类微服务框架技术汇总如下：

1）Dubbo作为国内微服务框架开源的代表作，使用方便上手简单。以及早期的金融微服务场景，多用于互联网行业，存量用户较多。国内银行领域使用的案例较少。

2）SpringCloud作为后起之秀，目前在新的微服务项目中使用的比较多，但是其作为一系列零散的通用框架组合，缺少一整套的微服务解决方案和最佳落地实践支持。

3）长亮企业级分布式服务平台EDSP作为一套完全兼容SpringCloud的微服务技术体系，提供应用全生命周期管理、数字化运维、立体化监控、细粒度服务治理等功能；帮助企业客户解决传统集中式架构转型的困难，打造大规模高可用的金融级分布式系统架构，实现业务和产品的快速落地。在过去的几年，ESDP拥有大量的银行客户案例，成功帮助了几十家银行实现数字化转型，完成核心系统的分布式微服务技术实施，深得广大用户的一致好评。

微服务框架选型建议采用EDSP（长亮企业级分布式服务平台）。