背景：

关键字：平台化、分步式服务自治架构、运维成本、微服务、技术迭代。

框架：

ZooKeeper (路由和负载)

RMI（远程服务调用）

OSGI（服务实现和发布）

SpringDM[暂定]（服务上下文的管理）

架构图：

client

regsity

gateway

Container

Provider1

Service1

Service2

Provider1

Service1

Service2

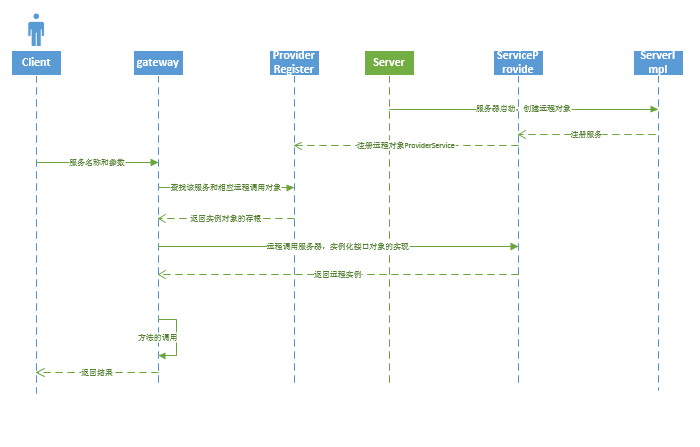
Provider2

Service1

Service2

Monitor

时序图：



模块：

1. Gateway

调用服务的消费方。负责对从registry获取服务的信息、接口的认证、安全过滤、日志记录、远程接口调用、实例化远程服务。

1. Registry

Provider的注册、负载均衡、保存服务信息

1. Monitor

监控服务被调用次数和时间、检测ServiceProvder的心跳。

1. service-provider

负责对外暴露服务接口、服务的管理（添加、更新、卸载等）

1. Server（Container）

服务器启动的时候ServiceImpl向Provider注册服务并通过ProviderService注册到Registy中。。