# 简介

Spring Boot来简化Spring应用开发，约定大于配置，去繁从简，just run就能创建一个独立的，产品级别的应用；整个Spring技术栈的一个大整合

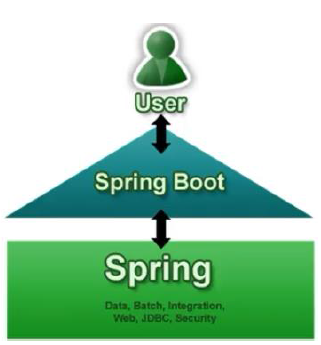
## 简介

### 背景：

J2EE笨重的开发、繁多的配置、低下的开发效率、复杂的部署流程、第三方技术集成难度大。

解决：“Spring全家桶”时代。

1.Spring Boot 解决J2EE一站式解决方案 2.Spring Cloud解决分布式整体解决方案



### 优点：

–快速创建独立运行的Spring项目以及与主流框架集成

–使用嵌入式的Servlet容器，应用无需打成WAR包

–starters自动依赖与版本控制

–大量的自动配置，简化开发，也可修改默认值

–无需配置XML，无代码生成，开箱即用

–准生产环境的运行时应用监控

–与云计算的天然集成

## 微服务

### 单体应用

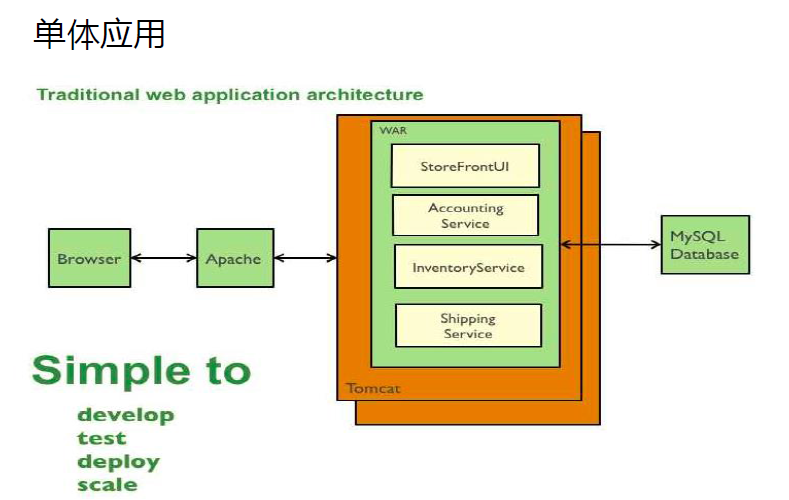
目前的所有传统项目

好处：

1. 开发测试简单：所有服务都在一个应用，不必联调其他应用
2. 部署也简单：只有一个应用
3. 负载也方便：直接复制应用放在10几个tomcat里面

弊端：

1. 修改部署麻烦：修改一点地方，这个应用需要全部重新部署
2. 注定不可能成为大型应用：后期维护麻烦



### 微服务

微服务：架构风格（服务微化）

一个应用应该是一组小型服务；可以通过HTTP的方式进行互通；



优点：

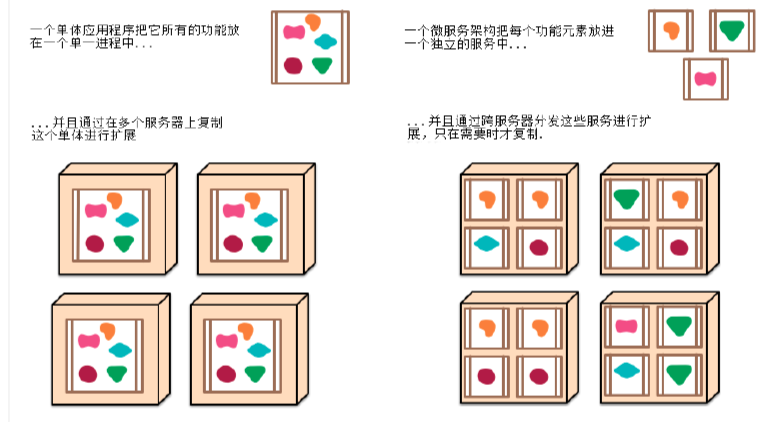
1. 节约资源：拆分功能元素，有需要时才复制；区别单体应用的全部复制
2. 可替换性强：每一个功能元素最终都是一个可独立替换和独立升级的软件单元；

微服务文档](https://martinfowler.com/articles/microservices.html#MicroservicesAndSoa)

缺点：

部署麻烦：功能单元太多

### 对比

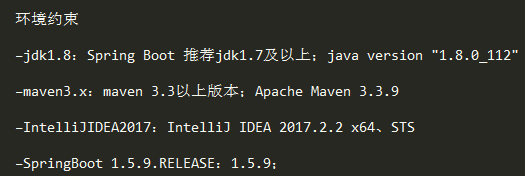


### 联系



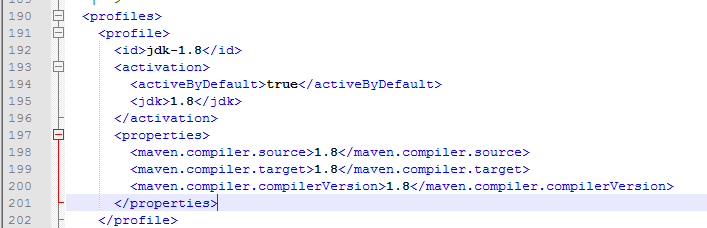
# 环境

## 统一配置



## MAVEN设置；

给maven 的settings.xml配置文件的profiles标签添加



<profile>

<id>jdk-1.8</id>

<activation>

<activeByDefault>true</activeByDefault>

<jdk>1.8</jdk>

</activation>

<properties>

<maven.compiler.source>1.8</maven.compiler.source>

<maven.compiler.target>1.8</maven.compiler.target>

<maven.compiler.compilerVersion>1.8</maven.compiler.compilerVersion>

</properties>

</profile>

## IDEA设置

