

# SpringBoot在链家网实践

程天亮

链家网平台策略部

自我介绍

SpringBoot实践

困难与挑战

# 目录

# PART 1

---

## 自我介绍



## 程天亮-9年互联网研发的老兵



毕业9年

高级研发工程师



SONY  
make.believe

架构师/研发主管

LIANJIA.链家



## 链家网平台策略部

- 数据挖掘/机器学习平台
- NLP处理平台
- 检索API平台
- API网关平台
- 用户中心平台
- 品控运营平台

- 检索每日提供**10亿+**服务
- 数据挖掘/机器学习超过**90%**的预估准确率
- 用户中心每日提供用户信息服务高达**4亿+**服务

# PART 2

SpringBoot实践



# Spring WAR 是什么？

简化开发

内置web容器

自动配置

自带内部监控

Spring4+生态

## 链家网平台策略技术架构





## 链家网平台策略部发展遇到的问题

### 服务配置复杂

基础服务多，服务的资源配置复杂。传统方式管理服务复杂

01  
OPTION

02  
OPTION

### 服务之间调用复杂

检索服务、用户中心服务等，其之间的调用复杂，依赖多，需要微服务化

### 服务监控难度大

服务比较多，机器部署复杂  
服务存活监控、业务是否正常监控尤为重要

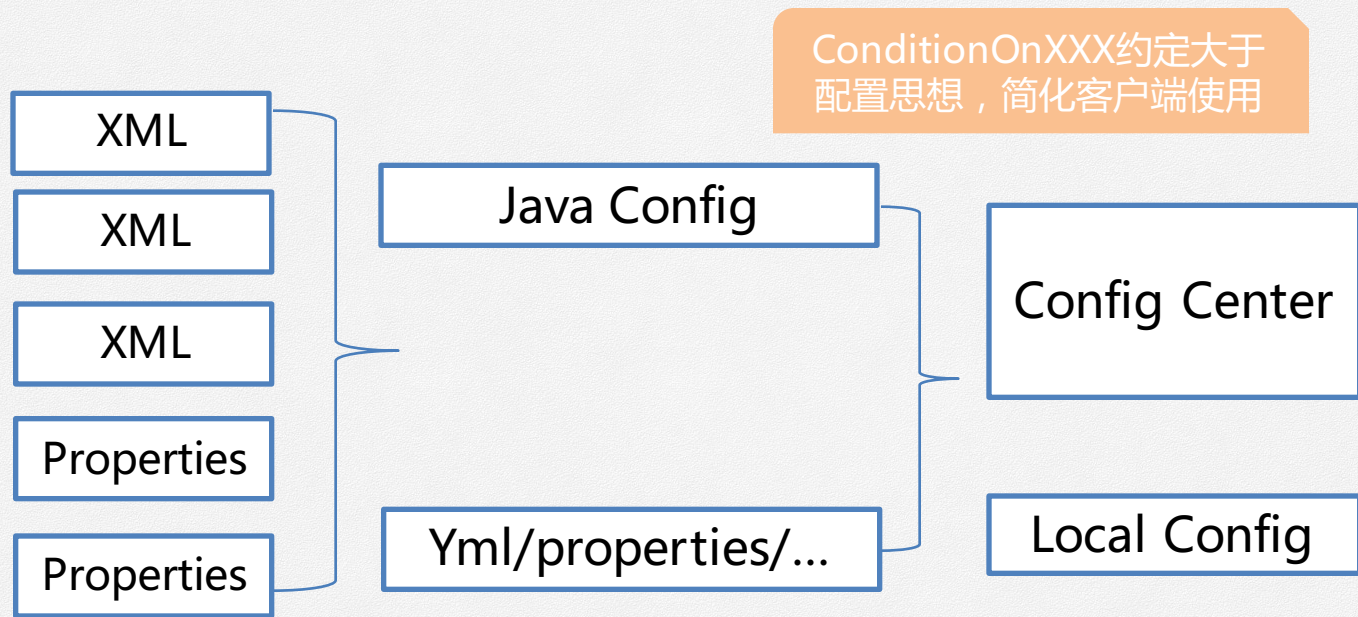
03  
OPTION

04  
OPTION

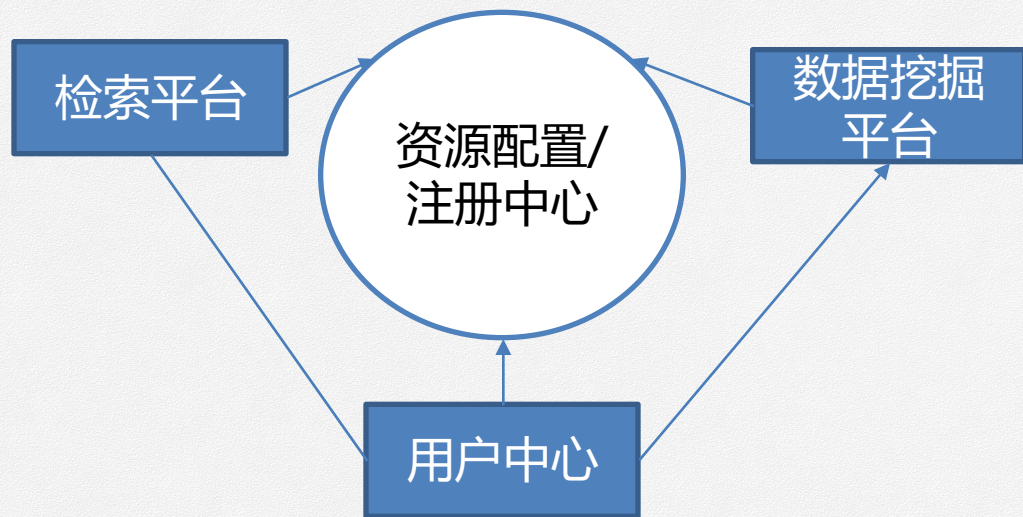
### 服务化测试问题

服务依赖性比较大，测试一个小的功能，周边服务也需要启动。

# SpringBoot实践之旅-解决配置凌乱 ( 1/2 )



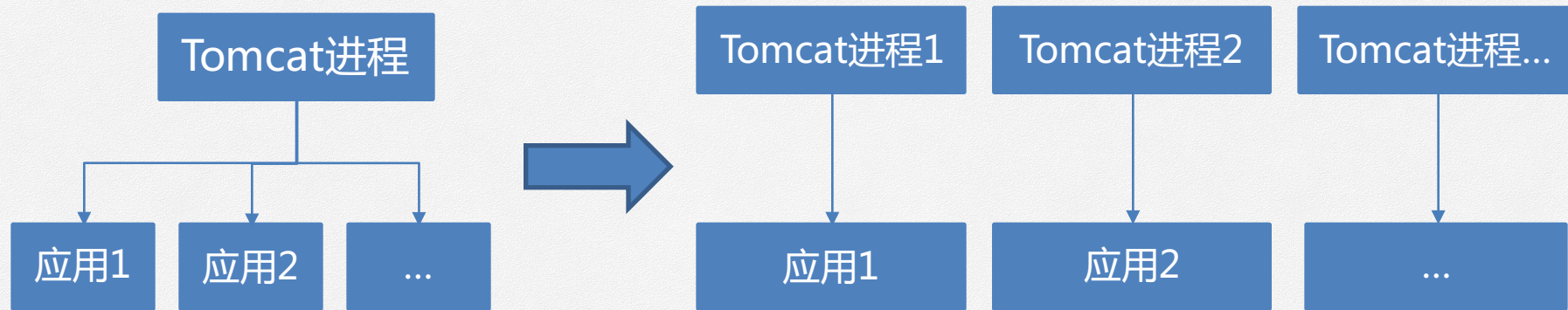
# SpringBoot实践之旅-解决配置凌乱 ( 2/2 )



- 服务平台扩展  
AdviceModeImportSelector进而实现  
EnableAutoRefreshConfig，自动接收配置中心配置更新指令
- 资源配置中心指令发送

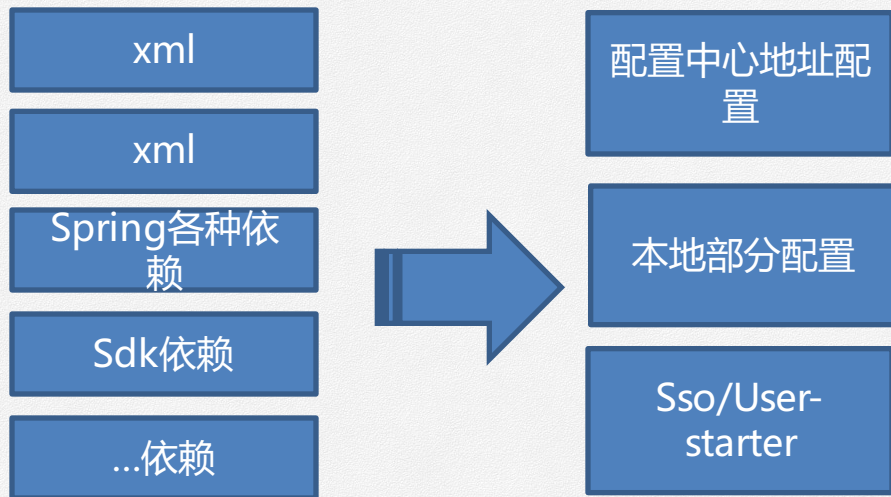


# SpringBoot实践之旅-解决单进程多应用低吞吐量



- 单应用单处理，使用SpringBoot fat jar模式，将tomcat等容器内嵌
- 每个tomcat只管理一个应用，提高应用的吞吐量
- 服务的管理更加清晰化，打包工程更加流程化
- 服务内嵌tomcat，测试更加容易控制进程，更容易实现自动化测试

# SpringBoot实践之旅-SSO/权限管理

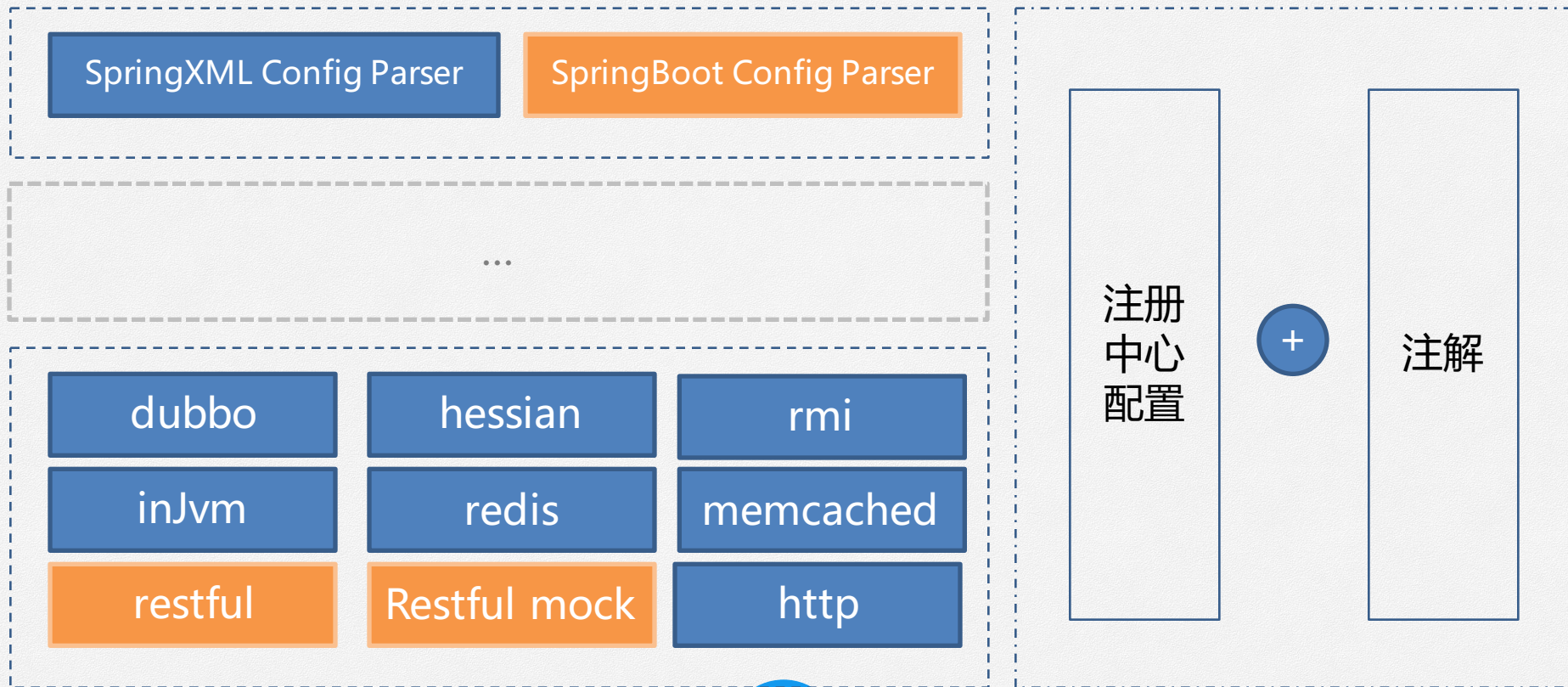


- SpringBoot Starter
- SpringBoot扩展注解  
EnableAutoRefreshConfig/EnableSSO/EnableAutoDiscovery

```
@SpringBootApplication
@EnableAutoRefreshConfig
@EnableSso
public class FlushApplication {

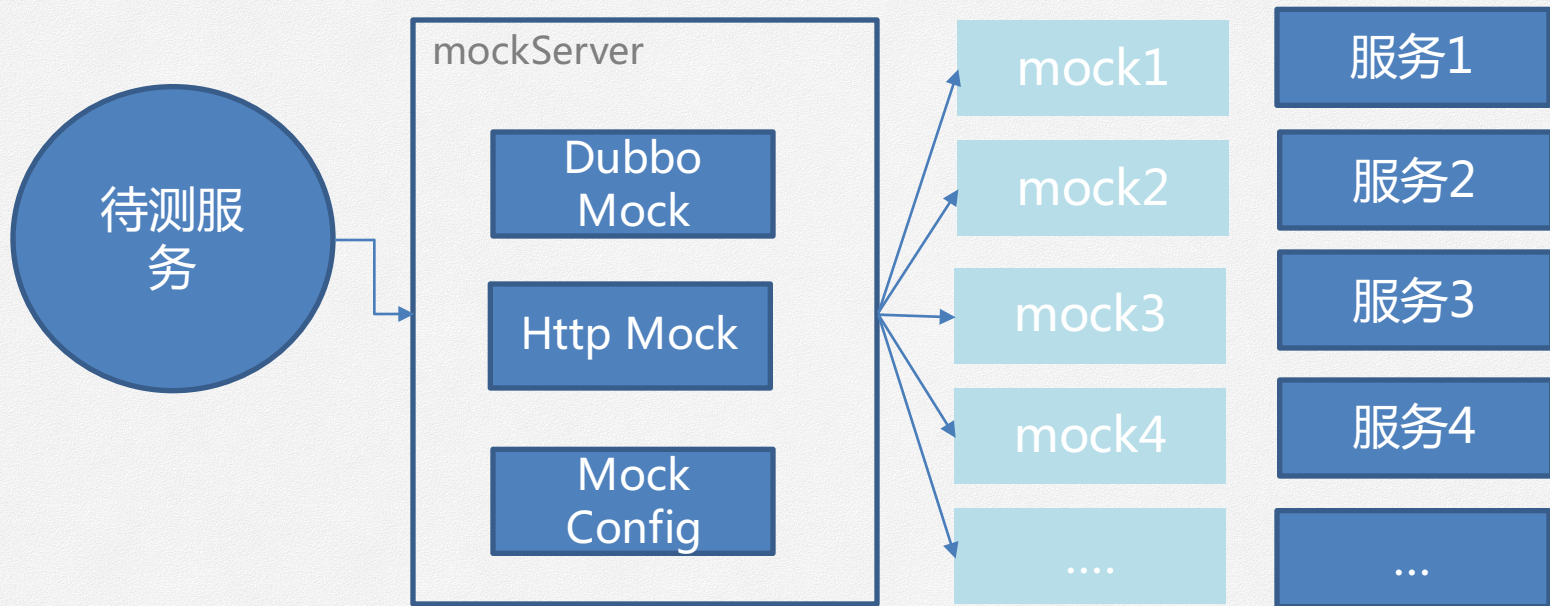
    public static void main(String[] args) {
        SpringApplication.run(FlushApplication.class, args);
    }
}
```

# SpringBoot实践之旅-dubbo ( RPC ) 改造

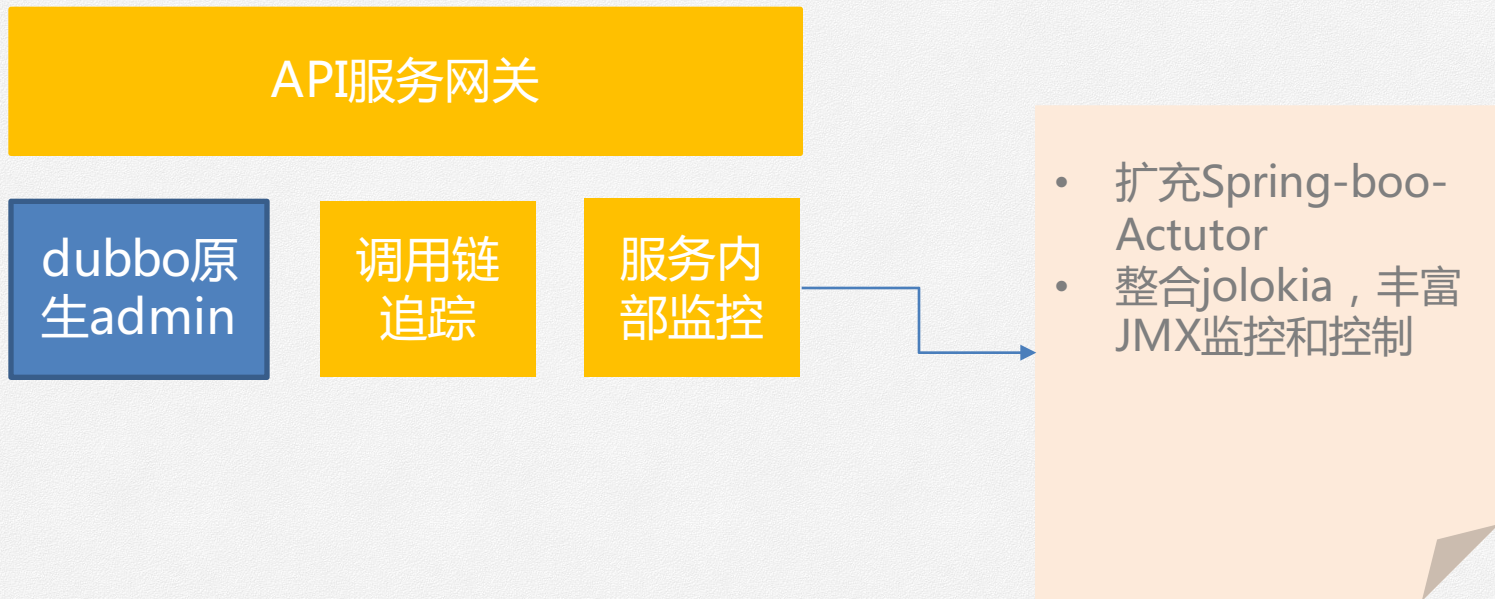




## SpringBoot实践之旅-微服务mock测试



## SpringBoot实践之旅-微服务治理



# PART 3

## 困难与挑战

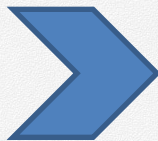


## 线上运维问题

单Tomcat

标准启停脚本

Tomcat配置Op  
统一



多Tomcat

脚本随意化

Tomcat配置下放  
到应用

## 编码上的问题

### 01

#### 代码阅读难度加大

- 1、代码的入口更加隐藏，不如web.xml和spring.xml逻辑清晰
- 2、要求更高的编程规范

### 02

#### 学习成本加大

- 1、新的技术都有学习成本
- 2、SpringBoot入门简单，用好或扩展需要更深的学习或源码学习

### 03

#### 入门级微服务框架

- 1、SpringBoot主要还是对Spring的在封装，解决Spring易用
- 2、SpringBoot作微服务，还需要对周边进行丰富（如SpringCloud）

**Q & A**





