爱旅行-系统架构设计及部署策略分析

爱旅行项目是一个综合性的旅游电商平台，并具备跨平台性。那么基于PC、移动端（Andriod、Ios）的考虑，以及该项目的非功能性需求（详见《爱旅行系统需求分析说明书.doc》）:安全性、可靠性、互操作性、健壮性、易用性、可维护性、可移植性、可重用性、可扩充性等等都是基于性能、安全等架构目标的，因此爱旅行项目系统架构的设计目标有以下7个：

1. 可靠性（reliable）
2. 安全性（secure）
3. 可扩展性（scalable/extension）
4. 可定制化（customizable）
5. 可维护性（maintainable）
6. 客户体验（customer experience）
7. 市场时机（time to market）

## 系统架构设计

爱旅行系统的整体技术架构如图1所示：

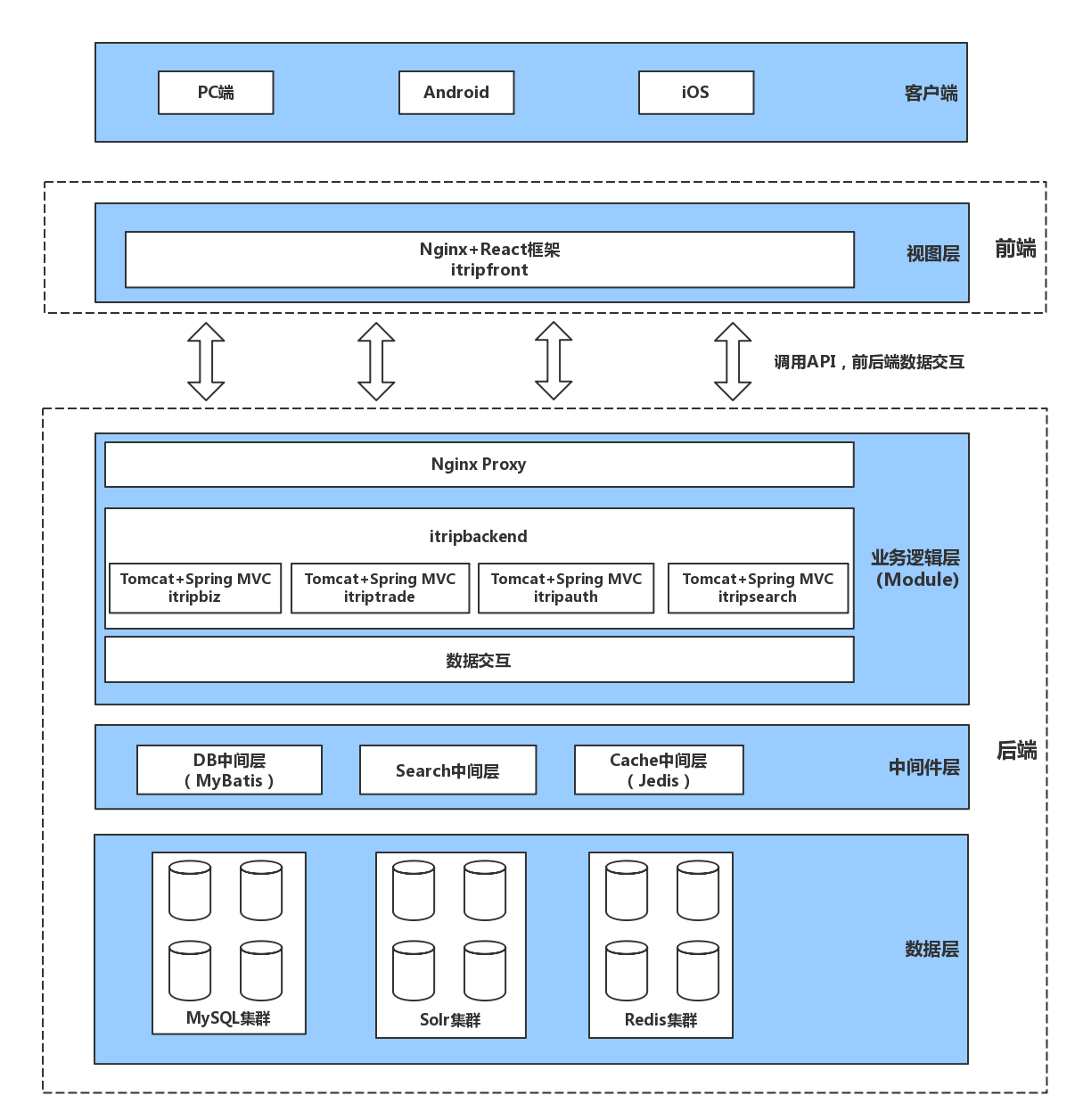


图1：爱旅行系统架构图

爱旅行架构从两个方向进行设计：前端架构和后端架构，将前端和后端完全分离，并采用分布式集群的架构设计，通过定义API，与前端进行数据交互，前端使用html进行数据展现，并可加入移动端（Andriod、Ios）实现，直接调用API即可。

前端架构采用React JavaScript 库实现组件化开发，并通过Nginx反向代理到后端项目，实现API的调用，进行前后端数据交互。

后端架构采用模块化开发，采用Maven进行多Module管理。根据业务进行模块划分，itripbackend分为以下四个Web模块：

1. 主业务模块（itripbiz）：包括酒店业务、旅游业务、机票业务、攻略业务
2. 搜索模块（itripsearch）：爱旅行项目中所有搜索功能（solr）
3. 认证模块（itripauth）：负责用户身份验证，生成并维护Token
4. 支付模块（itrptrade）：包括支付宝支付、微信支付

以及为Web模块提供服务的三个基础模块：itripdao、itriputils、itripbeans。

每个web模块均独立部署在单独的web服务器上，并通过Nginx反向代理到相应的应用服务器（tomcat）上：

|  |
| --- |
| upstream itripbiz\_server  {  server IP:port; # itripbiz模块部署的服务器IP及端口号  server IP:port backup; # itripbiz模块部署的服务器IP及端口号（备份服务器）  }  upstream itripsearch\_server  {  server IP:port; # itripsearch模块部署的服务器IP及端口号  }  upstream itripauth \_server  {  server IP:port; # itripauth模块部署的服务器IP及端口号  }  upstream itriptrade\_server  {  server IP:port; # itriptrade模块部署的服务器IP及端口号  } |

server {

…

location /biz {

…

proxy\_pass http://itripbiz\_server;

}

location /biz {

…

proxy\_pass http://itripbiz\_server;

}

location /biz {

…

proxy\_pass http://itripbiz\_server;

}

location /biz {

…

proxy\_pass http://itripbiz\_server;

}

## 1.2 系统部署架构设计

1、【强制】前端的健壮性开发，需要达到：在无API的情况下，前端一样能够完美奔跑

2、【推荐】前端采用mock开发