**HDI总体设计方案V1.0**

**广东以大供应链管理有限公司**

**编写人：罗荣熙**

**2019.03.27**

目录

[一、 引言 3](#_Toc2522)

[1、 需求背景 3](#_Toc2463)

[2、 业务场景 3](#_Toc25461)

[3、 存在问题 3](#_Toc6511)

[4、 名词术语 4](#_Toc64)

[二、 概述 5](#_Toc30721)

[1、 具体需求 5](#_Toc30931)

[1.1功能性需求 5](#_Toc14347)

[1.2后台服务需求 6](#_Toc1926)

[1.3非功能性需求 7](#_Toc15132)

[2、 建设目标 8](#_Toc7675)

[3、 建设规划 9](#_Toc4459)

[3.1第一期：系统基础建设 9](#_Toc25633)

[3.2第二期：数据中心建设 9](#_Toc8545)

[3.3第三期：数据分析统计 9](#_Toc21544)

[3.4第四期：大数据分析建设 9](#_Toc7214)

[三、 总体结构 10](#_Toc830)

[1、 总体架构图 10](#_Toc12315)

[1.1业务数据库层 11](#_Toc25870)

[1.2数据共享交互层 11](#_Toc11911)

[1.3数据仓库层 12](#_Toc2133)

[1.4扩展应用层 12](#_Toc19116)

[2、 核心流程图 13](#_Toc32015)

[四、 技术选型 13](#_Toc32669)

[1、 微服务架构 13](#_Toc22785)

[1.1特点 13](#_Toc24346)

[1.2优点 13](#_Toc13745)

[1.3缺点 14](#_Toc22578)

[2、 后台选用Spring Cloud 14](#_Toc6927)

[2.1为什么使用Spring Cloud 14](#_Toc15307)

[2.2Spring Cloud技术栈 15](#_Toc9296)

[3、 前端选用Vue 15](#_Toc13955)

[3.1响应的数据绑定/响应式编程 15](#_Toc22740)

[3.2组件化 15](#_Toc29325)

[4、 数据库选用MySQL 16](#_Toc16656)

[5、 缓存选用Redis 16](#_Toc26992)

[6、 消息中间件选用RabbitMQ 16](#_Toc16324)

[6.1RabbitMQ简介 16](#_Toc27866)

[6.2优点 16](#_Toc14870)

[7、 项目管理工具 16](#_Toc26333)

[8、 企业服务总线选用Mule ESB 17](#_Toc11585)

[9、 数据集成工具选用Kettle 17](#_Toc5465)

[五、 其他 17](#_Toc1784)

# 引言

本文档主要描述“HDI数据交互平台”软件的总体设计思路。首先从公司平台需求背景、业务场景、运行环境等方面概要描述系统的现状，其次分析存在的问题，再根据具体的需求从设计原则、功能设计、数据结构设计等方面描述系统的总体设计情况，然后进一步详细描述系统技术实现策略、确定技术选型的问题。

## 需求背景

随着公司业务的发展，合作的供应商以及医院越来越多，营模式也越来越复杂，目前供应商ERP数据和医院SPD相关业务数据呈离散分布，没有形成统一的数据中心，数据不流通，形成信息孤岛，如果整合业务数据、提供及时准确科学的决策、降低管理成本成为迫切的需求。为了让“HDI数据交互平台”能够更好地支撑未来业务的发展，满足多运营模式下供应商、集团、监管等多方的业务管理和数据展示等需求，对HDI进行了重新规划和设计，旨在把HDI建设成简洁易用的供应商管理平台、数据多维统计分析的数据展示平台。

## 业务场景

2.1供应商/运营商通过HDI能够便利地实现机构信息、商品信息等基础信息管理，相关机构信息、商品信息可以通过接口方式从ERP系统上传，并自动完成ERP、HDI、SPD等系统的机构信息和商品信息的匹对转化。

2.2供应商/运营商通过HDI能够实现商品采购计划、供货、退货、开票、结算等业务流程的管理，相关采购计划、供货、退货、发票、结算数据实现与ERP、SPD等系统的同步与交互。

2.3供应商/运营商通过HDI能够实现证照统一管理和证照预警，证照信息可以通过接口方式从ERP系统上传，也可以单独录入或批量导入，并可与商品、机构进行自动绑定。

2.4发票信息实现ERP/HDI/SPD等系统的互联互通，供应商/运营商通过HDI能够对发票信息进行维护管理，可进行退货开红冲发票和结算开票等操作。

2.5集团用户或监管用户可根据需要定制个性化、多维度的统计数据及报表数据。

2.6 HDI提供统一接口标准实现与ERP、SPD等外部系统的数据对接，系统对接高效可控，并可配置维护系统数据访问权限，数据传输独立安全。

## 存在问题

3.1目前的HDI系统数据规划不足以支撑集团管理和监管机构等多元化数据统计分析及呈现需求。

3.2各系统间的数据没有建立统一标准，例如医院商品信息和供应商商品信息等跨系统数据需要手工匹对，工作量大，工作效率低。

3.3 HDI没有建立对外统一的接口标准，导致系统对接工作重复繁琐，不适合未来业务急剧发展需要。

3.4 HDI没有统一的证照管理功能，证照不能实现和商品/机构的自动绑定，无法快捷分发到供应商合作的医院进行审核。

3.5 HDI没有发票管理功能，不能进行退货开红冲发票和结算开票操作，不能对发票信息记录进行维护管理。

3.6 HDI没有实现和ERP/SPD等上下游业务系统的全流程业务数据交互和监控，导致业务数据反馈不及时，不能集中管理。

3.7 HDI模块划分不够独立清晰，不具备可快速插拔重组特性，导致平台扩展性不强。

3.8平台用户接口和操作界面设计不够简洁易用，兼容性不够强。

## 名词术语

|  |  |
| --- | --- |
| **词语** | **解释** |
| ETL | 数据仓库技术：数据抽取、清洗、转换、装载 |
| DW | 可简写为DW或DWH。数据仓库，是为企业所有级别的决策制定过程，提供所有类型数据支持的战略集合。 |
| ODS | 操作数据存储，是数据仓库体系结构中的一个可选部分，ODS具备数据仓库的部分特征和OLTP系统的部分特征，它是“面向主题的、集成的、当前或接近当前的、不断变化的”数据。 |
| DM | 数据集市，DM是针对某一个业务领域建立模型，具体用户（决策层）查看DM生成的报表。 |
| DIM | 轻度汇总或维度数据 |
| MD | 元数据 |
| DWB/DWS | 轻度汇总、遵循企业业务事务处理的形式，将各个专业数据进行集中，将数据按分析的主题的形式存放，属于分析的公共资源 |
| 主题（Subject） | 主题是在较高层次上将企业信息系统中的数据进行综合、归类和分析利用的一个抽象概念，每一个主题基本对应一个宏观的分析领域。在逻辑意义上，它是对应企业中某一宏观分析领域所涉及的分析对象。 |

# 概述

## 具体需求

### 1.1功能性需求

#### 1.1.1机构管理

提供供应商、医院、厂商等机构信息的维护功能，包括查询、新增、修改、删除、批量导入/导出等操作功能，同时机构信息要实现与ERP、SPD系统的互联互通。

#### **1.1.2供应商商品管理**

提供供应商商品信息的维护功能，包括查询、新增、修改、删除、批量导入/导出等操作功能。商品信息可从ERP系统上传。商品信息按统一编码规格进行编码标识，并实现与ERP、SPD系统的商品信息进行自动匹对转化。

#### **1.1.3采购计划管理**

可查看SPD上传的采购计划（含采购详单），可转发采购计划到ERP并接收ERP回传的确认信息，同时系统提供查询、新增、修改、删除、批量导入/导出等采购计划维护功能，并可进行确认采购计划、生成供货记录等流程操作。

#### 1.1.4供货管理

供货信息实现ERP/HDI/SPD等系统的互联互通，提供供货记录的维护管理功能，包括查询、新增、修改、删除、批量导入/导出、打印供货清单、打印标签、开票等操作功能，同时可根据SPD回传的验收结果改变供货记录状态。

#### 1.1.5退货管理

退货信息实现ERP/HDI/SPD等系统的互联互通，可查看、审批医院提交的退货记录，提供退货记录的维护管理功能，包括查询、新增、修改、删除、批量导入/导出、退货确认、开红冲票等操作功能。

#### 1.1.6结算管理

结算数据实现ERP/HDI/SPD等系统的互联互通，提供结算记录的维护管理功能，包括查询、新增、修改、删除、批量导入/导出、结算单审阅、结算确认、结算开票等功能。

#### 1.1.7票据管理

票据信息实现ERP/HDI/SPD等系统的互联互通，提供发票记录的维护管理功能，包括查询、新增、修改、删除、批量导入/导出等功能。发票记录包括供货发票记录、退货红冲发票记录和结算发票记录。

#### 1.1.8证照管理

证照信息实现ERP/HDI/SPD等系统的互联互通，证照信息可从ERP上传，按证照所属机构或商品品种自动和所对应的机构/商品绑定，并根据供应商和医院的绑定关系把证照分发提交给医院进行审批，并根据SPD回传的审批结果改变证照记录审批状态。

提供证照信息的维护管理功能，包括查询、新增、修改、删除、批量导入/导出、提交审批等功能。

提供证照分类维护功能，包括增加、删除、修改证照分类，设置证照分类预警等功能。

提供证照近效期预警功能。

#### 1.1.9系统管理

提供账户管理、角色管理、权限配置（包括数据访问权限和功能操作权限）、系统参数设置、后台配置等功能。

### 1.2后台服务需求

1.2.1系统实时接收ERP上传的机构信息，并把机构信息按供应商、合作医院、厂商类型分别显示在供应商管理、合作医院管理、厂商管理页面信息列表。

1.2.2系统实时接收SPD上传的合作供应商信息，并根据供应商和医院的绑定关系自动完成SPD供应商信息和HDI供应商信息的自动匹对。

1.2.3系统实时接收ERP上传的供应商商品信息，把不符合统一编码规则的商品信息按系统商品统一编码规则进行编码，生成供应商商品信息记录显示在供应商商品信息列表。

1.2.4系统实时接收SPD上传的医院商品信息，把不符合统一编码规则的商品信息按系统商品统一编码规则进行编码，并根据商品编码自动实现供应商—医院商品的信息匹对转化。

1.2.5系统实时接收SPD上传的医院采购计划并自动转换成供应商商品采购计划（含采购详单），并可把供应商商品采购计划转发到ERP，以及接受ERP回传的采购计划确认信息。

1.2.6系统实时接收ERP上传的供货信息（含供货详单），生成供货信息记录显示在供货信息列表。

1.2.7系统实时接收SPD上传的退货信息（含退货详单），生成医院退货信息记录显示在退货信息列表。

1.2.8系统实时接收SPD上传的结算信息，生成医院结算信息记录显示在结算信息列表。

1.2.9系统实时接收ERP上传的证照信息，把证照信息按证照类型分别显示在供应商证照、商品管理、厂商证照信息列表，并按证照所属机构或商品品种自动和所对应的机构/商品绑定。

1.2.10证照提交审批时系统根据供应商和医院的绑定关系自动把证照分发给相应医院进行审批，并根据SPD回传的审批结果改变证照记录审批状态。

### 1.3非功能性需求

#### 1.3.1安全性需求

按照信息系统安全标准建设规范，采用可靠的安全防范技术和措施，使平台的网络安全、应用安全、数据安全等符合信息安全等保二级要求，确保系统安全、稳定可靠。应用系统投入运行后，不能出现数据库崩溃，数据丢失，系统不能正常运行等现象。

#### 1.3.2可扩展性需求

平台建设采用模块化的设计思路，通过应用支撑组件＋业务功能构件实现系统各项功能。模块化设计的原则是力求以少数模块组成尽可能多的产品，模块结构应尽量简单、规范，模块间的联系尽可能简单，使平台具备良好的可扩展性。

#### 1.3.3标准化和开放性需求

HDI平台需要和多个外部应用系统进行数据交互，在系统设计中要提高平台的标准性和开放性。系统信息的存储、管理、信息交换等方面都要采用标准化，支持标准协议、规范等，要做到平台无关性。同时，平台要能集成其他成熟的应用系统，接收和使用其他系统的信息和功能，提高平台的开放性。

#### 1.3.4易用性和兼容性需求

平台用户接口和操作界面设计要简洁直观，兼容现有主流操作系统、浏览器软件，具有较好的人机交互界面；支持当前主流的、先进的可视化、图形化的数据展示工具，实现查询、分析、统计等应用的可视化、图形化展现。

#### 1.3.5数据独立性需求

接入平台的各系统数据在传输、存储、访问时要确保相互独立性，做到数据隔离、严格授权、数据监控。

#### 1.3.6响应时间需求

页面响应时间达到1/3/5标准，1秒内为优秀值，3秒内为达标值，5秒为可接受值。

#### 1.3.7接口需求

提供独立的接口服务，利用标准化接口实现HDI和SPD/ERP等外部系统的数据采集和交互。接口服务与平台解耦，可独立运行，确保系统对接高效可控。

#### 1.3.8集团模式需求

平台要支持多机构多层级的集团管理模式，支持集团对下属机构进行统一业务流程管理和数据管理，支持集团多维度的数据统计分析展示需求。

#### 1.3.9数据迁移需求

新的HDI云平台启用后，需要把原来系统的生产数据迁移过来，并按照新平台的数据标准完成原有数据的清洗和转换。

## 建设目标

2.1建设数据中心，提供多维度的数据统计分析呈现，满足集团管理和监管机构等多元化数据需求。

2.2建立统一的数据标准，实现跨系统数据的自动匹对转化。

2.3对外建立统一的接口标准，方便外部系统接入及进行数据交互。

2.4提供证照管理功能，实现证照和商品/机构的自动绑定，快捷分发到供应商合作的医院进行审核。

2.5提供发票管理功能，对发票信息记录进行维护管理，增加退货开红冲发票和结算开票操作。

2.6实现和ERP/SPD等上下游业务系统的全流程业务数据交互和监控，集中管理相关业务数据。

2.7规划独立清晰的功能模块，通过应用支撑组件＋业务功能构件实现系统各项功能，使平台具备良好的可扩展性。

2.8设计简洁易用的用户接口和操作界面，兼容现有主流操作系统、浏览器软件以及图形化展示工具。

## 建设规划

### 3.1第一期：系统基础建设

需求评审、产品原型评审、UI设计评审、技术架构选型、开发框架搭建、库表设计、实现（机构信息、供应商信息、采购计划、供货信息、退货信息、结算信息、票据信息、证照信息、系统管理）等基础功能。

### 3.2第二期：数据中心建设

业务数据分析、任务调度、ETL前置机、ETL脚本、日志监控、数据汇总。

### 3.3第三期：数据分析统计

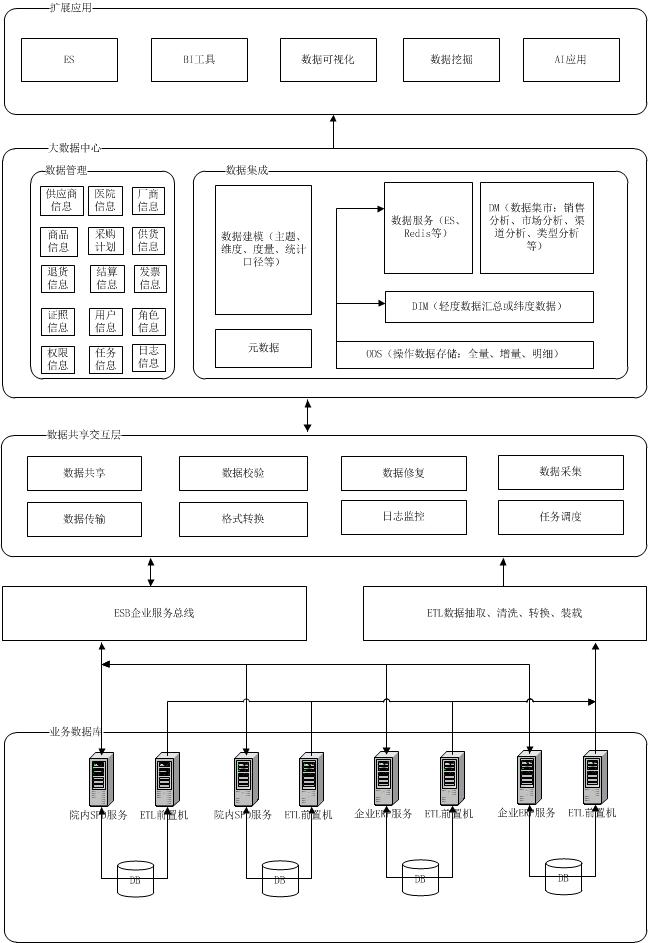
企业驾驶舱、医院驾驶舱、监管部门驾驶舱，实现可按主题、多维度进行数据分析统计。

### 3.4第四期：大数据分析建设

行业分析、数据可视化、数据挖掘、BI工具、AI应用等。

# 总体结构

## 总体架构图



### 1.1业务数据库层

业务数据主要来源于供应商ERP和医院SPD，包括药品信息、供应商信息、证照信息、出入库信息、采购信息、设备信息、耗材信息、配送信息、库存信息等相关数据。

### 1.2数据共享交互层

供应商ERP、医院SPD与HDI交互时，要求实时处理的数据，采用ESB定义统一的接口规范进行交互；对于不要求实时的数据采用ETL方式进行抽取。

#### 1.2.1数据同步

主要针对要求实时处理的数据，采用ESB企业服务总线提供统一的外部系统对接的接口标准，减轻对接的工作量和成本。

#### 1.2.2数据集成

ETL是构建数据仓库的重要基础性工作，在个业务系统部署ETL前置机，对数据进行实时增量抽取。ETL是数据流动的过程，即从不同异构数据源流向统一的目标数据的过程。其间，数据的抽取、清洗、转换和装载形成串行或并行的过程，ETL是数据仓库中的非常重要的一环，它是承前启后的必要的一步。

##### 1.2.2.1 ETL脚本

编写脚本的SQL语句时，我们可以只查询目标数据库表所需的数据，对数据进行处理，比如帅选、过滤、集合等操作，并添加一些校验的规则对数据进行清洗转换，这样数据要比原系统的数据少，还确保数据的合法有效性。

##### 1.2.2.2任务调度管理

ETL任务管理是ETL过程中的统一调度者和指挥者，它把复杂的数据处理过程中各个步骤整合成一个整体。在任务调度中配置ETL的执行时间、执行周期、执行过程等内容，可以实现自动化完成数据的采集、清洗及加载的过程。

##### 1.2.2.3全量抽取数据

即第一次抽取数据时，因为数据量比较大，所需要花的时间比较多。我们可以直接在Kettle的图形界面上进行操作，全量脚本需要包括步骤：输入表（对应SPD数据库的业务表）、输出表（对应数据仓库中的ODS存储表）、任务日志表（记录任务的执行情况：读取记录数、新增记录数、失败记录数、任务名称、执行时间、执行结果、耗时等）。

##### 1.2.2.4增量抽取数据

后续的数据增量更新，采用时间戳的方式进行处理。即系统在数据插入或更新时间记录操作的时间，这个时间往往是精确到秒。定时任务在执行ETL脚本时，将执行的起始和结束时间戳作为系统参数传到ETL脚本，由脚本的系统参数步骤进行接收。增量脚本需要包括步骤：系统参数（接收服务端传入的参数）、输入表（对应SPD数据库的业务表）、输出表（对应数据仓库中的ODS存储表）、任务日志表（记录任务的执行情况：读取记录数、新增记录数、失败记录数、任务名称、执行时间、执行结果、耗时等）。

### 1.3数据仓库层

#### 1.3.1 操作数据存储（ODS）

操作数据存储，是数据仓库体系结构中的一个可选部分，ODS具备数据仓库的部分特征和OLTP系统的部分特征，它是“面向主题的、集成的、当前或接近当前的、不断变化的”数据。用于支持企业对于即时性的、操作性的、集成的全体信息的需求。被作为数据仓库的数据处理的过渡，以降低直接进行数据处理的复杂度。

#### 1.3.2 轻度汇总或纬度数据（DIM）

轻度汇总层从多个纬度对数据进行汇总，如区域分布（省市区）、层级机构、用户属性聚合、费用汇总、清单汇总等。

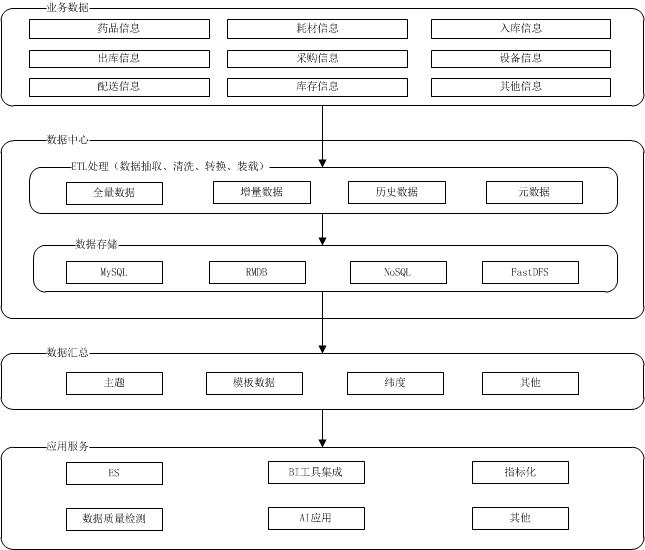
#### 1.3.3数据集市（DM）

数据集市是深度汇总层，作为商业智能系统的组成部分，从企业数据仓库中ODS层抽取相关数据并进行转换和装载，并根据应用需求形成的数据集合，支撑各个业务部门及人员进行建设专业化应用。

### 1.4扩展应用层

基于数据中心做大数据分析，扩展应用包括全文搜索引擎、BI工具、数据可视化（数据质量、指标分析）、数据挖掘、AI应用等。

## 核心流程图



# 技术选型

## 微服务架构

### 1.1特点

1.1.1将系统服务层完全独立出来，并将服务层抽取为一个一个的微服务。

1.1.2微服务遵循单一原则。

1.1.3微服务之间采用RESTful等轻量协议传输。

### 1.2优点

1.2.1服务拆分粒度更细，有利于资源重复利用，提高开发效率。

1.2.2可以更加精准的指定每个服务的优化方案，提高系统可维护性。

1.2.3微服务架构采用去中心化思想，服务之间采用RESTful等轻量协议通信，相比ESB更轻量。

1.2.4适用于互联网时代，产品迭代周期更短。

### 1.3缺点

1.3.1微服务过多，服务治理成本高，不利于系统维护。

1.3.2分布式系统开发的技术成本高（容错、分布式事务等），对团队挑战大。

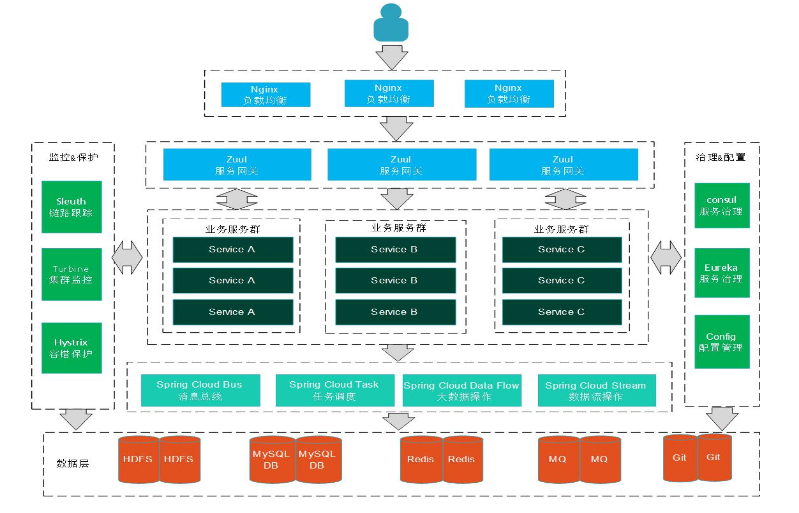
## 后台选用Spring Cloud

Spring Cloud为开发人员构建微服务架构提供了完整的解决方案，Spring Cloud是若干个架构的集合，它包括spring-cloud-config、spring-cloud-bus等近20个子项目，它提供了服务治理、服务网关、智能路由、负载均衡、断路器、监控跟踪、分布式消息队列、配置管理等领域的解决方案。

### 2.1为什么使用Spring Cloud

微服务架构的优点表明它可以提高我们的生产力，但是分布式系统本身的技术成本问题给互联网哪些创业型公司不少的挑战，阿里、百度等巨头所提供的微服务技术只是解决其中某个问题，而整合封装这些优秀的技术恐怕是Spring最擅长的领域了，Spring Cloud也正因为此而诞生。使用Spring Cloud来构建微服务架构可以省去你整合各家技术的成本，Spring Cloud为我们构建微服务架构提供了一站式的解决方案。

### 2.2Spring Cloud技术栈



Spring Cloud提供一站式的微服务架构解决方案

## 前端选用Vue

vue是一个构建用户界面的框架(库),它的目标是通过尽可能简单的api实现响应的数据绑定和组合的视图集合 vue自身不是一个全能框架的核心是只关注视图层,因此它非常容易学习，非常容易与其它库或已有项目整合 vue在与相关工具和支持库一起使用时, 也能完美地驱动复杂的单页应用。

### 3.1响应的数据绑定/响应式编程

Vue.js 的核心是一个响应的数据绑定系统，它让数据与 DOM 保持同步非常简单。在使用 jQuery 手工操作 DOM时，我们的代码常常是命令式的、重复的与易错的。 Vue.js 拥抱数据驱动的视图概念。通俗地讲，它意味着我们在普通 HTML 模板中使用特殊的语法将 DOM“绑定”到底层数据。一旦创建了绑定，DOM 将与数据保持同步。 每当修改了数据，DOM 便相应地更新。这样我们应用中的逻辑就几乎都是直接修改数据了，不必与 DOM更新搅在一起。这让我们的代码更容易撰写、理解与维护。

### 3.2组件化

组件（Component）是 Vue.js 最强大的功能之一。组件可以扩展 HTML 元素，封装可重用的代码。 在较高层面上，组件是自定义元素， Vue.js 的编译器为它添加特殊功能。在有些情况下，组件也可以是原生 HTML 元素的形式，以 is 特性扩展。

## 数据库选用MySQL

MySQL免费开源，持续更新维护，文档齐全，作为数据库的第二把交椅甚至直逼Oracle这个老大，在互联网开发中MySQL的应用是最广发的。

## 缓存选用Redis

Redis是Nosql数据库，是一个key-value存储系统。虽然redis是key-value的存储系统，但是redis支持的value存储类型是非常的多，比如字符串、链表、集合、有序集合和哈希。这些类型的数据存储不需要固定的模式，无需多余的操作就可以进行横向的扩展。相对于关系型数据库可以减少表和字段特别多的情况。也无型之间在架构的层面上带来了可扩展的能力。

## 消息中间件选用RabbitMQ

### 6.1RabbitMQ简介

是一个开源的消息代理和队列服务器，用来通过普通协议在完全不同的应用之间共享数据，RabbitMQ是使用Erlang语言来编写的，并且RabbitMQ是基于AMQP协议的。

### 6.2优点

1. 目前很多互联网大厂都在使用RabbitMQ；
2. RabbitMQ底层采用Erlang语言进行编写；
3. 开源、性能优秀、稳定性保障；
4. 与SpringAMQP完美整合、API丰富；
5. 集群模式丰富，表达式配置，HA模式，镜像队里模型；
6. 保证数据不丢失的前提做到高可用、可用性；
7. AMQP全称：Advanced Message Queuing Protocol；
8. AMQP翻译：高级消息队列协议。

## 项目管理工具

7.1私有库：Sonatype Nexus

7.2项目构建：Maven

7.3版本管理：SVN或GIT

7.4持续集成：Hudson或Jenkins

7.5代码质量：SonarQube

## 企业服务总线选用Mule ESB

Mule ESB是一个基于Java的轻量级企业服务总线和集成平台，允许开发人员快速便利地连接多个应用，并支持应用间的数据交换。Mule通过Transports/Connectors与外围的异构系统连接，提供Routing（路由）、Transaction Management（事务管理）、Transformation（转换）、Message Broker（消息代理）、Transportation Management（传输管理）、Security（安全）等核心模块。Mule可以单独使用，也可以架设在常用的应用服务器上。

## 数据集成工具选用Kettle

Kettle是一款国外开源的ETL工具，纯JAVA编写，通过提供图形化界面操作，快速实现转换脚本和工作流控制脚本编写，其脚本可与Quartz任务调度集成，方便脚本的统一管理。Quartz任务调度框架支持多线程，与Spring集成统一管理线程池，应对高并发。