## 全行业清算平台技术架构调研

## 编写目的

本文档用于描述全行业清算平台的应用层架构设计。通过视图流程图和文字的方式，提供系统的设计思路，以及系统运用到的工具包和相关的技术。

文档的目标受众，包括: 需求人员、架构人员、开发人员、测试人员、应用维护人员。

## 背景

       互联网技术，尤其是移动互联网技术的广泛应用,推动着清算平台的不断变革，新的商业模式和平台不断推出。清算部部门以目前的新清算系统为基础，为方便发布对外报表信息以及收集整合方数据信息，方便全行业的清算，特对目前系统进行整合扩展，以达到各个平台用户的需求。

项目建设的必要性：

* 现有结算系统的功能需要进一步增强和整合
* 现有系统的架构和开发技术亟待升级

## 参考文档

## 术语说明

# 技术架构

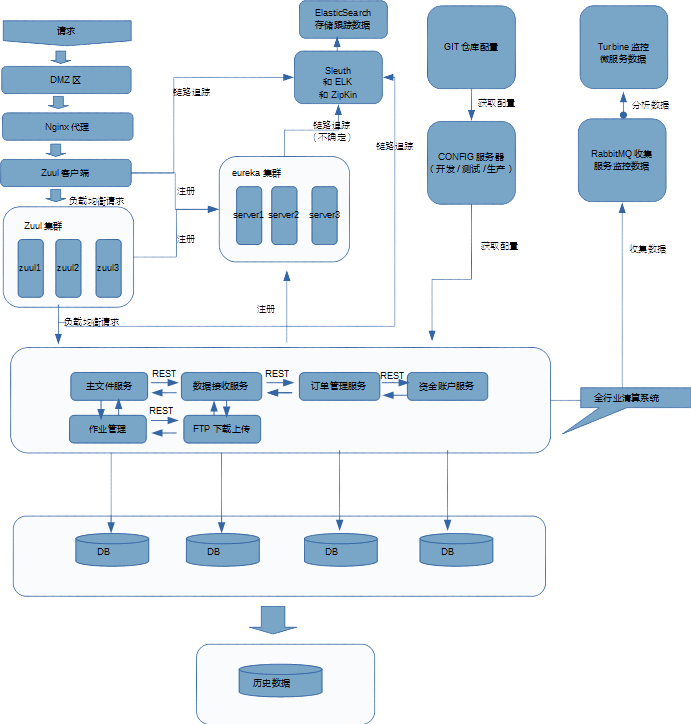
## 逻辑架构

### **概述**

全行业清算平台预采用微服务架构模式，将各个模块的业务功能拆分为多个子系统。各个子系统独立灵活，能够单独部署，而不影响其他子系统组件。不再像以前一样需要服务器效率很高，内存大和CUP核数多来提高系统效率。这种架构风格，将一个单一的应用程序开发为一组小型服务的方法。每个服务运行在自己的进程中，服务间采用轻量级通信机制（通常用HTTP资源API）。这些服务共用一个最小型的集中式管理，每个微服务可以用不同的开发语言。使用不同的数据存储技术，这样使得每个微服务很容易按需扩展。微服务之间是松耦合的，微服务内部是高内聚的。服务之间通讯采用REST风格。

目前的微服务拆分是按业务域拆分，例如：主文件服务、数据接收配比服务、订单管理服务、资金账户服务、作业管理、FTP下载上传。

应用架构图如下：



### **DMZ区：**

外部https或者http请求首先进入DMZ区。DMZ可以为主机环境提供网络级的保护,能减少为不信任客户提供服务而引发的危险,是放置公共信息的最佳位置。我们可以将需要保护的Web应用程序服务器和数据库系统放在内网中,把没有包含敏感数据、担当代理数据访问职责的主机放置于DMZ中,为应用系统安全提供保障。DMZ使包含重要数据的内部系统免于直接暴露给外部网络而受到攻击,攻击者即使初步入侵成功,还要面临DMZ设置的新的障碍。

**Nginx代理**

Nginx是一个轻量级、高性能、稳定性高、并发性好的HTTP和反向代理服务器。也是由于其特性，其应用非常广。

正向代理：某些情况下，代理我们用户去访问服务器，需要手动的设置代理服务器的ip和端口号。

反向代理：是用来代理服务器的，代理我们要访问的目标服务器。代理服务器接受请求，然后将请求转发给内部网络的服务器(集群化)，并将从服务器上得到的结果返回给客户端，此时代理服务器对外就表现为一个服务器。

通过对http请求地址的解析，根据地址规则进行二次转发，通过Nginx服务器找到真正处理该业务的服务器。

### **Zuul网关**

有时完成一个业务需求，可能调用多个微服务接口。如果直接让客户端和各个微服务通信，会有以下问题：

客户端会多次请求不同的微服务，增加了客户端的复杂性。

存在跨域请求，在一定场景下处理相对复杂。

认证复杂，每个服务都需要独立认证。

难以重构，随着项目的迭代，可能需要重新划分微服务。可能将多个微服务合并成一个或者将一个拆分成多个。如果客户端直接与微服务通信则很难实施。

引入了微服务网关，它介于客户端和服务端之间的中间层，所有的外部请求都要经过微服务网关。客户端只需要和网关打交道。

Zuul网关配置多个或者集群

Zuul网关的优点：易于监控、易于认证。认证后的请求，在后端的每个微服务之间不再进行认证。

Zuul的核心：

1、身份认证与安全：未登录：生成token，已登录：token进行刷新获取；

2、动态路由：动态的将请求路由到不同的后端集群

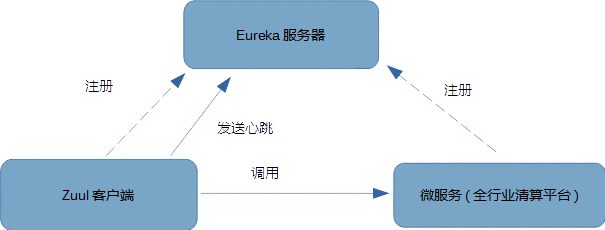
3、压力测试：逐渐增加指向集群的流量，以了解系统的速度性能，吞吐量。

4、负载分配：根据随机、轮询、hash等方法，设定限定值，均衡的分配请求服务器。

5、静态响应处理：在边缘位置直接建立部分响应，从而避免其转发到内部集群。

### **Eureka服务注册与发现**

对以上图详细分析Eureka服务器的原理如下图：



工作流程：各个微服务：Zuul微服务网关，全行业清算平台的各个子微服务模块：订单管理模块、资金管理模块等等，在第一次启动时，将自己的网络地址注册到Eureka服务器注册表上。多个集群之间，以复制的方式来实现注册表中数据的同步。注册时候分为客户端和服务端进行分别注册保存，这样服务器列表里可以想象出是保存着一对一，一对多或者多对多的对应关系地址，Eureka对这些地址进行维护。

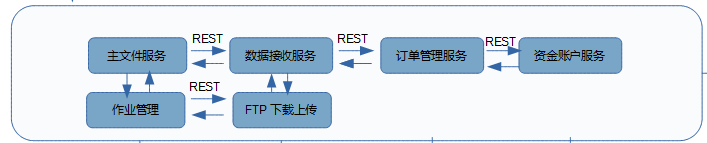
当请求真正来时，Zuul客户端从Eureka服务器上查询微服务（全行业清算平台）的网络地址，比如：资金管理模块的网络地址，拿到地址后，使用该地址调用资金管理模块的接口，进行数据处理和返回。

Zuul客户端，和全行业清算平台的各个微服务，与Eureka保持一定机制的心跳通信，比如30秒未收到返回信息，则会注销该实例，认为该微服务未启动。

提供这种服务注册与发现的组件还有很多：比如Consul，Zookeeper，NACOS等。

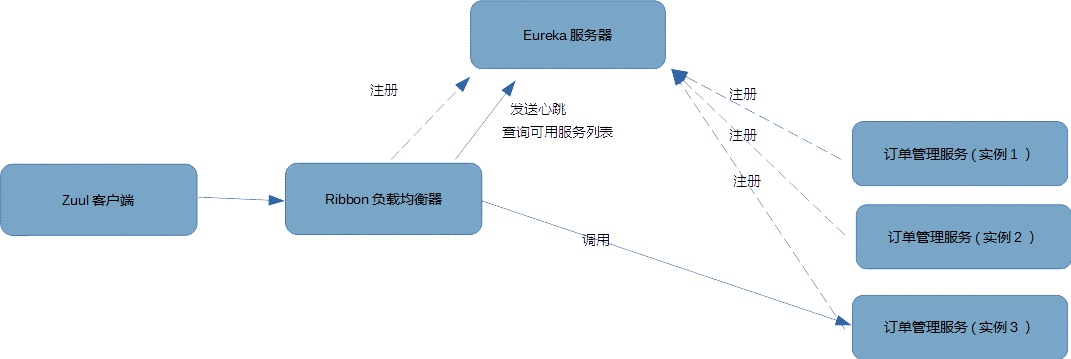
### **服务端**

所有的各个微服务组件共同组成了服务端。如下图所示：



服务端也可以部署集群模式，如果某一个模块用户访问过多，吞吐量大，可以部署多个微服务，多个微服务之间进行负载均衡。Ribbon可以提供这种负载均衡算法，根据：轮询、随机等等方法，找到对应的微服务来处理请求。

Ribbon和Eureka配合使用如下图所示：



### **HTTP链路追踪**

有对于请求边缘位置的有意义的数据和统计结果，进行追踪，可以进行产生精确的生成视图。

链路追踪的目的是为了记录日志，追踪请求的每一步。

1、Sleuth和ELK整合，

2、Sleuth和ZipKin整合

其他追踪的方式有：SpringBoot Actuator、Hystrix、HystrixCommand等。

**消息中间件链路追踪**

上一步链路跟踪是使用HTTP直接收集跟踪数据，也可以使用消息中间件收集追踪数据，RabbitMQ。

优点：

1、微服务与Zipkin Server解耦，微服务无需知道Zipkin Server的网络地址。

2、一些场景下，Zipkin Server与微服务网络可能不通，使用HTTP直接收集的方式可能无法工作，此时借助消息中间件实现数据收集。

### **CONFIG微服务配置管理**

常用的单体应用，使用配置文件管理所有配置。如果切换环境，设置多个profile，在启动时候指定profile。

在微服务中对配置进行集中管理，微服务架构的应用系统微服务比较多，可能几十个上百个，将所有的配置按照名称，环境进行区分，放入GIT仓库统一管理。

好处：

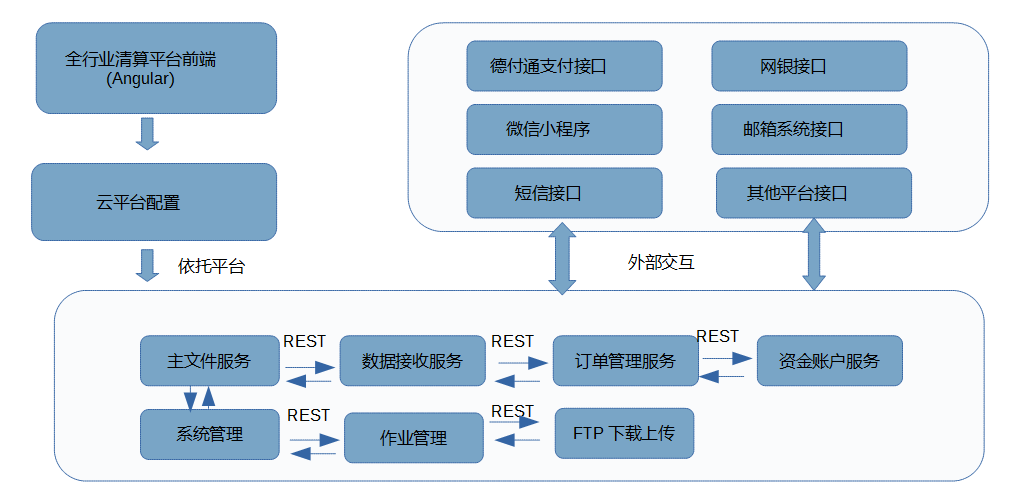
1、不同环境不同配置。区分：开发、测试、预发布、生产。

2、运行期间可以动态调整，而不需要停止微服务。

3、配置更改后，可自动更新。

### **全行业清算系统各个模块功能说明**

全行业清算系统和其他系统的关系如下图所示：



### **1、主文件服务**

系统中主要的基本新：系统中清算成员信息、清算成员组别、映射信息、归集信息、比价信息、清算日历、手续费协议等主文件。

### **2、系统管理服务**

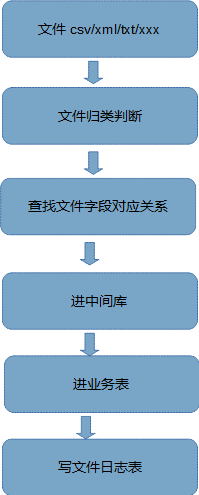
用户管理，角色管理，权限管理，数字字典，系统配置等公共的服务。

**3、数据接收服务**

对各种数据文件的接收解析：xml,csv,txt,以及加密文件等等，按照协议要求进行解析接入进库。以及对数据的清洗、去重、预处理、配比、错误处理、忽略、不符合条件数据删除等操作。

3.1、标准文件的接口：

各种文件的接收流程如下图所示：



文件类型表1：新增加一种接收的文件类型，在该表维护一条记录，包括文件类型、文件别称、解析方式：按节点、按列、按开始位置截止位置截取。

|  |  |  |  |
| --- | --- | --- | --- |
| 文件别名 | 文件类型 | 解析方式 | 匹配文件正则表达式 |
| Type1 | csv/xml/txt | 按节点、按列、按开始位置截止位置 | 符合匹配上的文件名，属于该类文件设置的接收方式，可以多种匹配方式 |

存储接收数据的中间表2：所有走标准接口设置的文件，都会存入该中间库。预先设计好标准表（中间表）的字段，各种数据类型String、Number等等类型的字段可以初始给5个，后期如果哪种类型字段不够用，可以再动态增加。

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Field1 | Field2 | Field3 | Field4 | Field5 | Field6 | Field7 | Field8 | Field9 |
| varchar | varchar | varchar | varchar | varchar | number | number | number | number |

字段关系设计表3：增加一种类型的文件，该表维护一套该文件的标准，前台动态配置调整对应关系，程序根据该表的对应关系，将文件中的数据存入的对应的表中。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 文件类型(表1字段） | 文件中节点(全路径) | 存入字段（表2字段） | 进业务表名 | 进业务表字段 |
| 20200101FMT.XML | InvoiceHeader/InvoiceNumber | Field1 | CLR\_BILL\_P | CLM |
| 20200101FMT.XML | InvoiceHeader/Number | Field2 | CLR\_BILL\_P | PRD |
| 202001.CSV | 0 | Field1 | CLR\_INVO\_P | CLM |
| 202001.CSV | 1 | Field2 | CLR\_INVO\_P | PRD |
| DEBT202001.XXX | （0,4） | Field1 | CLR\_BASE | BASECLM |
| DEBT202001.XXX | （4,6） | Field2 | CLR\_BASE | BASEPRD |

字段关系设计表4：接入的文件日志表：记录每个导入文件的详细日志：

导入文件的名称，导入标志，导入的服务器地址、导入时间、导入总行数、导入开始时间、导入结束时间、导入的文件类型。

**配比、错误处理、忽略**

错误处理：增加错误记录维护表，错误的标准定义，以及出现错误的处理方式。

配比：根据业务规则，根据错误记录标识去核对，按错误处理方式设置进行错误数据处理，包括忽略、删除、进业务表。

**4、作业管理服务**

作业管理子系统是批处理作业管理的子系统。提供了作业配置，作业计划定义，作业监控等服务。其他模块中有批处理作业的需求，可以通过在作业管理子系统中定义作业，以及作业的定时执行计划，实现作业执行与监控的需求。

**5、FTP文件下载上传**

文件管理子系统是为整个系统提供上传，下载的统一入口服务。其他各模块的文件上传服务都由此执行，并通过作业子系统自动触发作业的导入工作。文件子系统还提供附件管理，缩略图管理等功能。

**6、对外接口服务**

实现和外部系统的对接，分为：1、消息通知邮件通知。2、短信提醒；3、支付接口：支持调用德付通接口进行资金交付；4、微信小程序：显示系统主要信息。5、爬虫技术爬取数据。

6.1、邮件通知。通过配置，调用ACCA内部邮箱服务器的接口，实现系统实时和重要数据的提醒。

6.2、短信提醒。实现短信提醒需要以下步骤：

注册接口账号、创建应用、创建短信模板、创建测试号码、下载SDK、修改demo类参数

、测试短信发送、审核、申请短信项目上线。

6.3、德付通支付接口：调用德付通提供的支付接口，根据返回的交易信息，保存单号，交易金额，交易详细信息，交易成功标识等等信息。

6.4、微信小程序对接：

首先登录微信公众平https://mp.weixin.qq.com，注册开通微信小程序与微信支付商户，接下来具体步骤如下：

步骤一：选用合适模板

可以套用模板，通过自由拖拽组件的方式快速搭建自己的店铺，操作起来非常的方便快捷，在里面，可以使用木鱼小铺平台（https://www.muyu007.cn/）的工具，里面有各行各业的模板，

步骤二：利用组件，快速搭建个性化页面

在后台利用基础组件和插件快速搭建界面，将组件直接做拖拽到手机移动端里，上传图片和链接，点击提交即可。

步骤三：一键发布

组件设置完成之后，点击保存并发布，发布完成之后，再点击一键发布，输入您的微信支付等信息，点击提交，给微信官方审核，审核通过之后，就可以在微信上搜索小程序了

6.5.爬虫技术实现数据爬虫。

根据提供的爬虫地址，分析网页源码数据，获取有价值的信息数据。

**6、订单管理**

1、用户区分：航司、机场、一级货代、非航供应商。

2、对账号进行分级、监控、权限控制。

3、计算手续费、设置支付方式、轧差规则。

4、订单拆分、订单归集、订单监控。

5、风控规则、风控监控、黑名单管理、风险评级、授信额度。

6、提供交易查询、资金查询等。

7、短信接口、邮件接口、微信小程序接口、API账号查询。

8、定时作业，作业异常处理调用作业管理服务模块。

**资金管理服务**

1、用户区分：航司、机场、一级货代、非航供应商。

2、对资金流水，手续费、资金异常进行处理。

3、支付凭证、发票、订单查询、资金查询。

4、支付方式设置、外币管理、保证金/预付款处理。

5、系统信息流文件禹银行的信息流文件勾兑。

6、虚拟账户数据与资金账户数勾兑。

7、科目管理、账卡管理、试算平衡。

8、定时作业，作业异常处理调用作业管理服务模块。

**历史数据**

1、数据区只存放两年的数据。

2、两年以上的数据存放于历史数据库，对实时数据区的数据进行搬移处理。

3、报表库结合的从历史数据出。