# 供应链金融

## 供应链金融



### 什么是供应链金融

供应链金融就是供应链中上下游中小微企业通过自身的真实交易和应收账款和订单等信息，来获取平台的授信，通过资金方获取资金，解决自身融资问题

### 系统角色

#### 资产方(卖方)

在供应链金融中处于弱势地位，因为要垫付资金导致资金周转有问题，需要通过应收账款或者订单或者货物或者信用等抵押发起融资

#### 核心企业(买方)

处于优势位置，但是通过平台获取资金，可以提高资金周转率，相当于变相增加了收入

#### 平台(技术和数据和资金服务等)

参与供应链贸易场景，提供技术支持数据服务资金服务等，主要是供应链金融技术和数据服务平台

#### 资金方

提供资金支持(包含自有资金和三方资金)等

### 业务流程

#### 平台入驻

供应链上下游企业入驻平台(注册登录，创建用户信息，提供客户信息和企业信息，创建账户)

#### 上传资料

供应链上下游企业上传资料(企业资料管理) ，尽调，授信，核定授信额度(授信管理)

#### 展业

供应链上下游企业选择对应的业务模式，金融产品(金融产品管理) ，开展业务

#### 下单

根据合作厂家的商品(商品管理) ，签署订单，产生订单凭证

#### 审核

融资方向平台申请融资服务(流程管理) ，平台进行核实

#### 合同

基于真实交易，签署对应的贸易服务合同(合同管理)

#### 资金管理

基于资金渠道(资金路由管理) ，提供对应的融资方案(资金管理)

#### 账单

根据支付凭证，产生对应的账单(账单管理)和票据(发票管理)和账务管理(账务管理)

#### 贷后

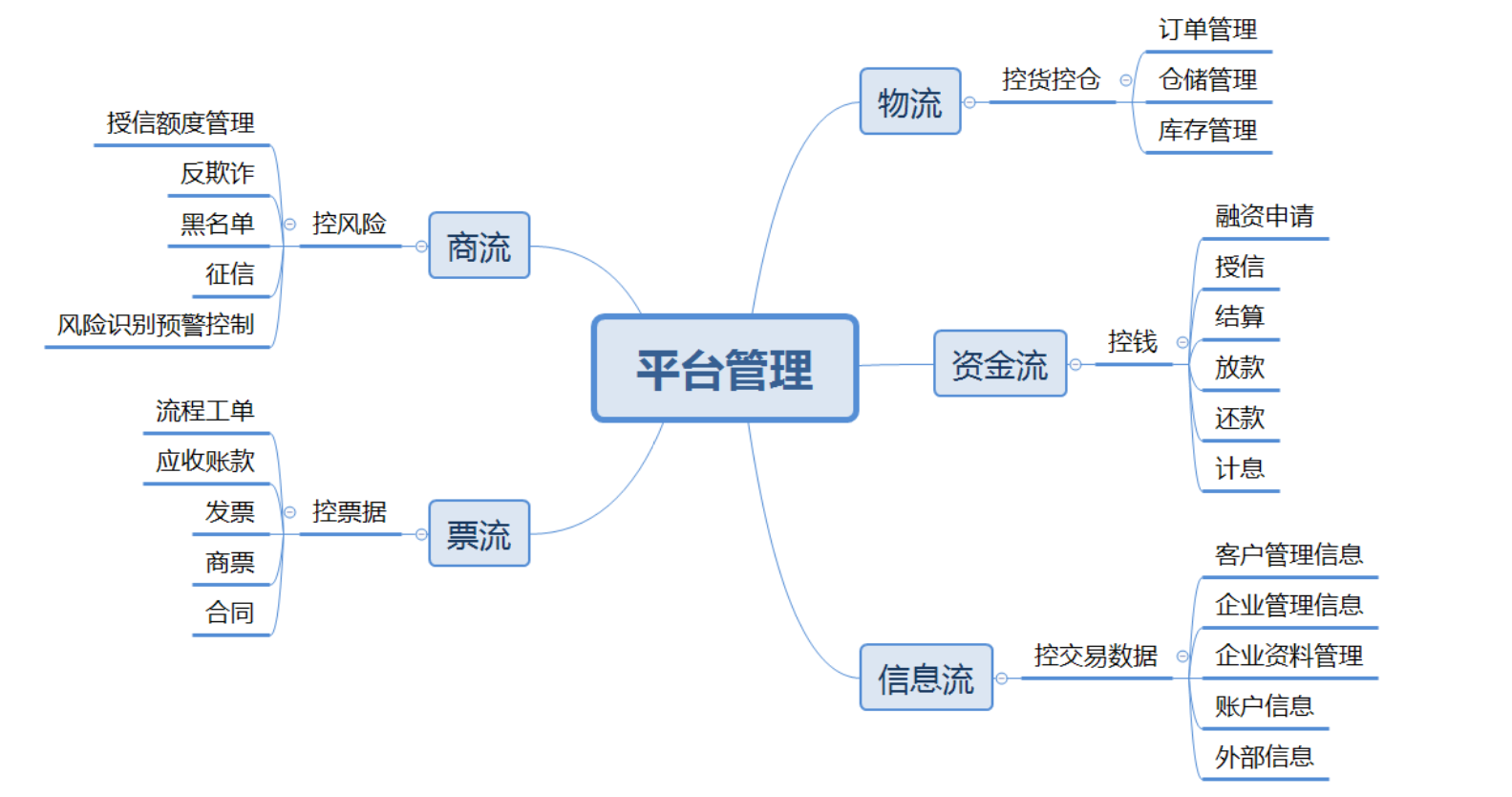
平台向融资方或者担保方进行催收，管理账款(贷后管理)(清结算服务)

#### 风控

风控贯穿全生命周期，提供风险识别，监控等服务

### 供应链金融平台

要从供应链金融管理的角度，搭建平台，要做到资金流，物流，信息流，商流，票据等的控制管理

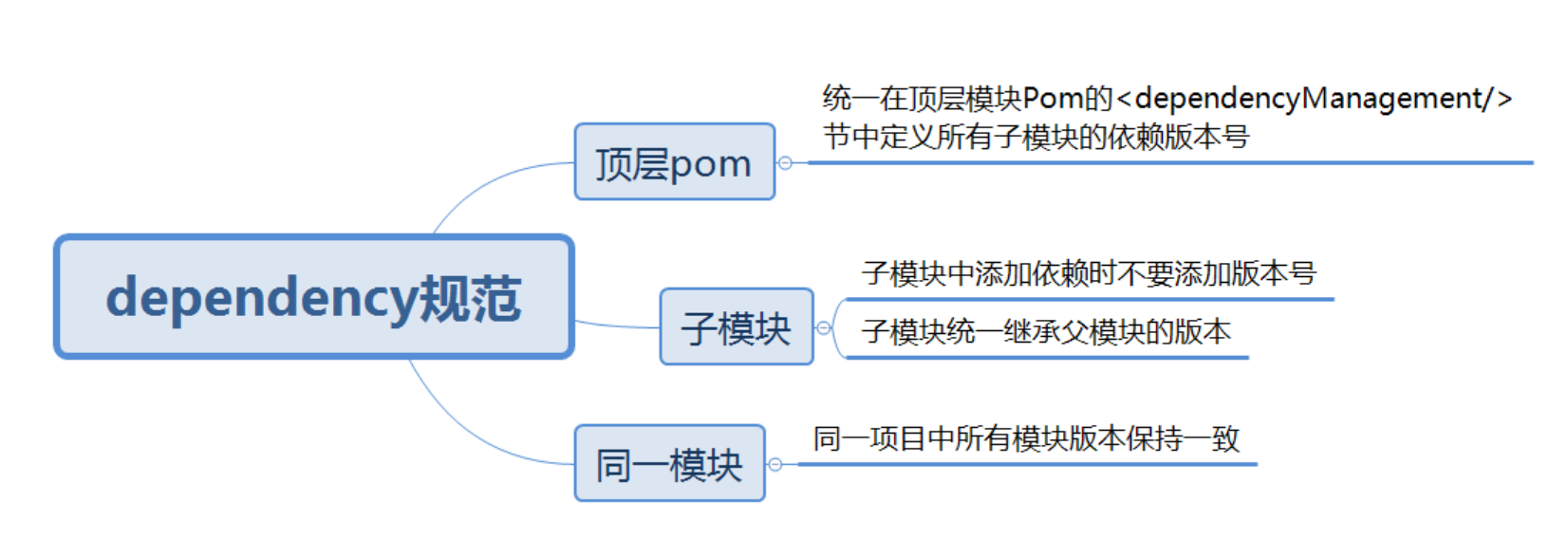


## 供应链金融规范设计

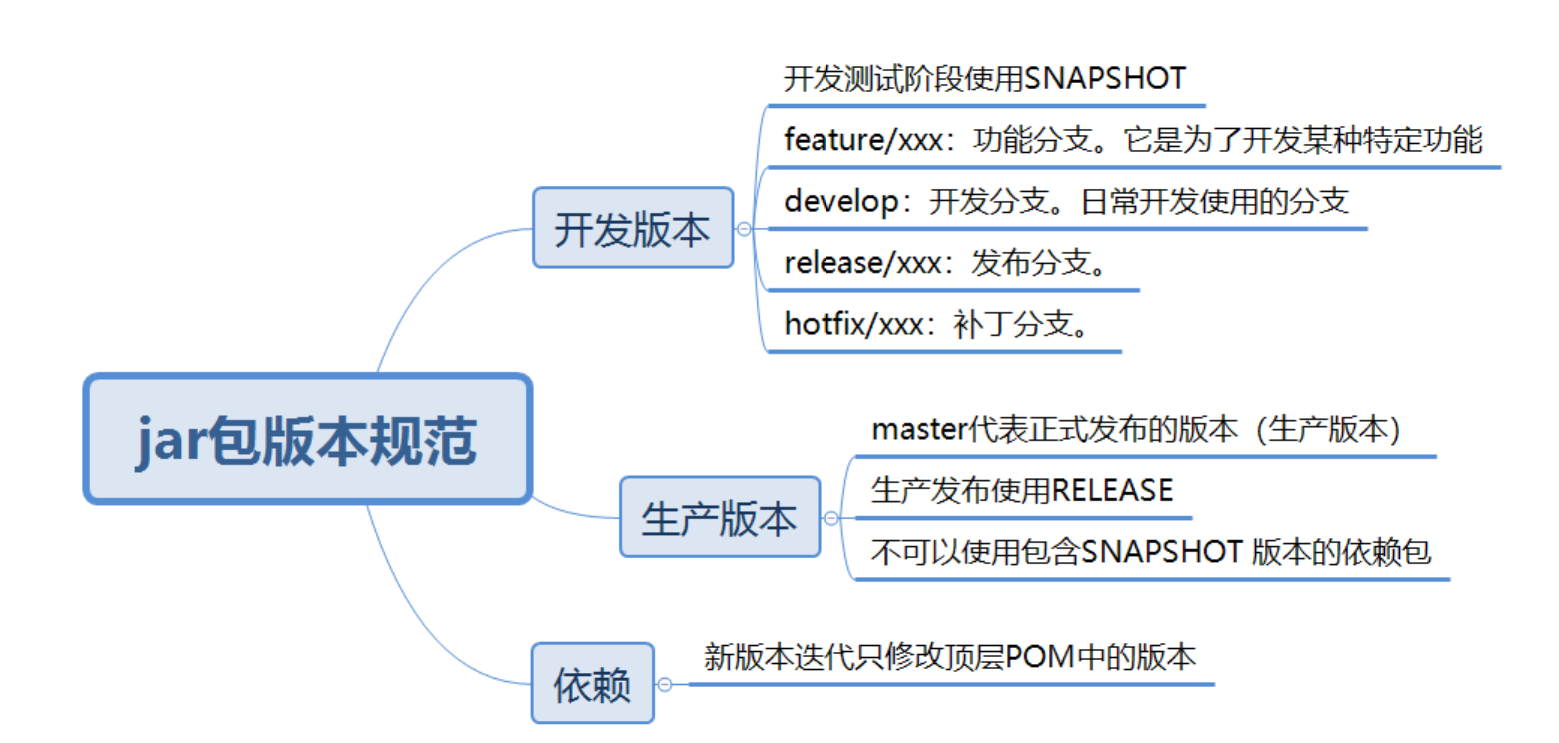
规范设计可以减少系统的复杂度，架构本身就是一种规范。

### 版本规范

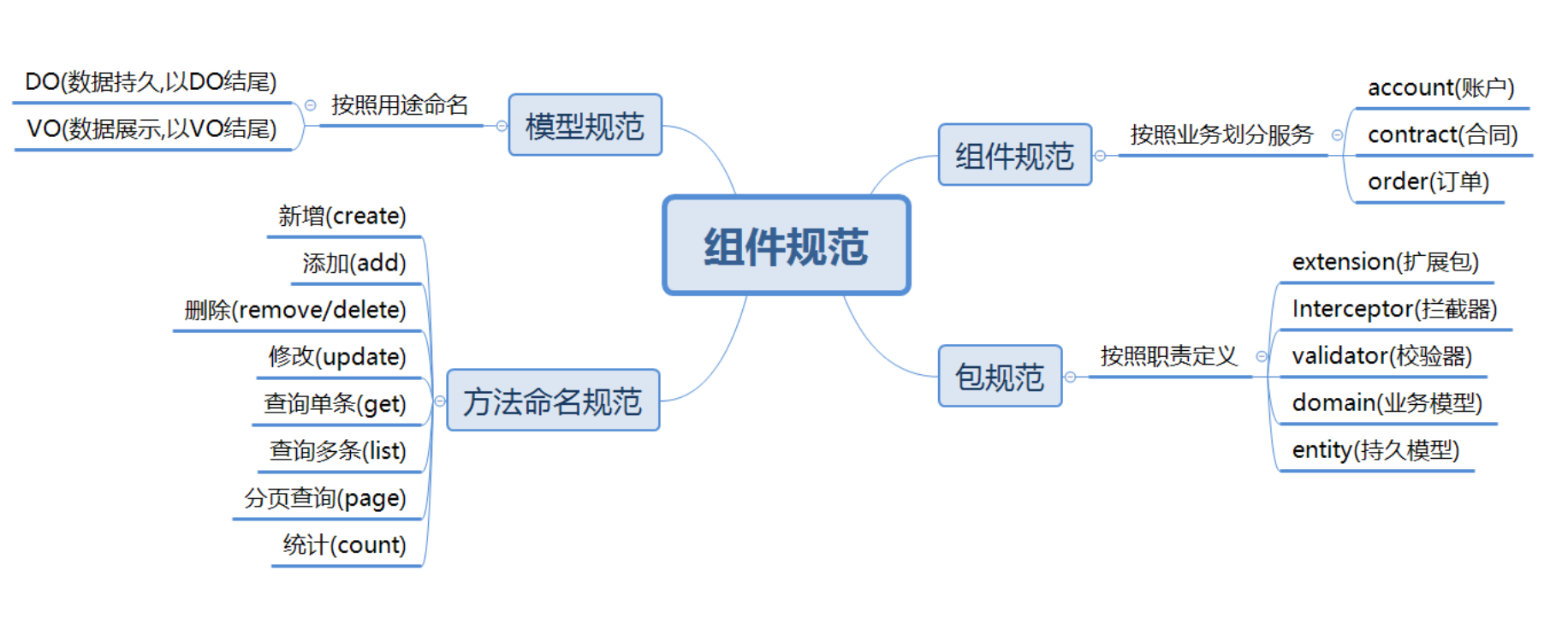
#### dependency版本规范



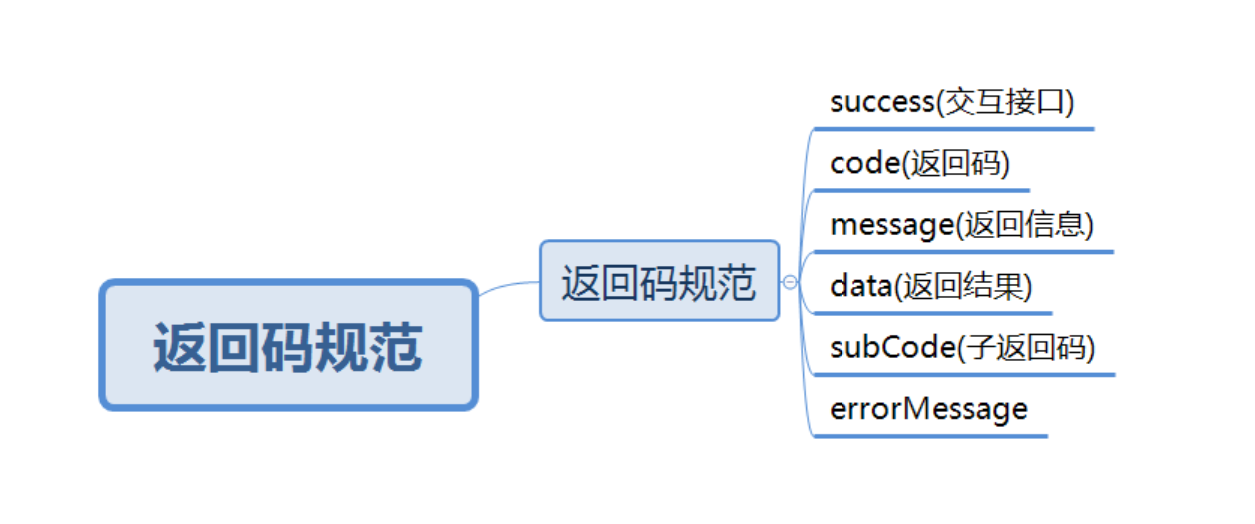
#### jar包规范



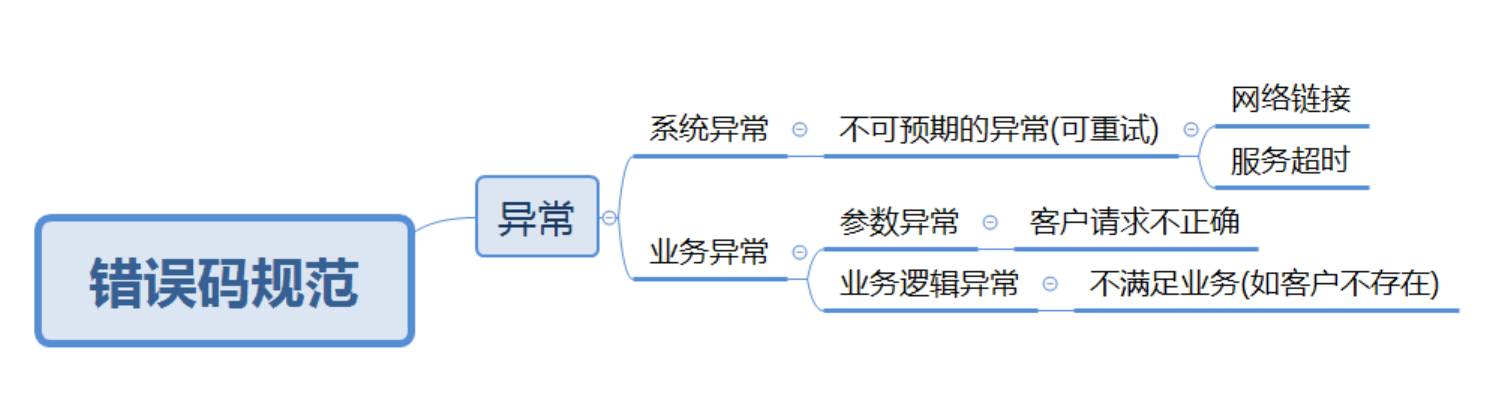
### 组件规范



### 返回码规范

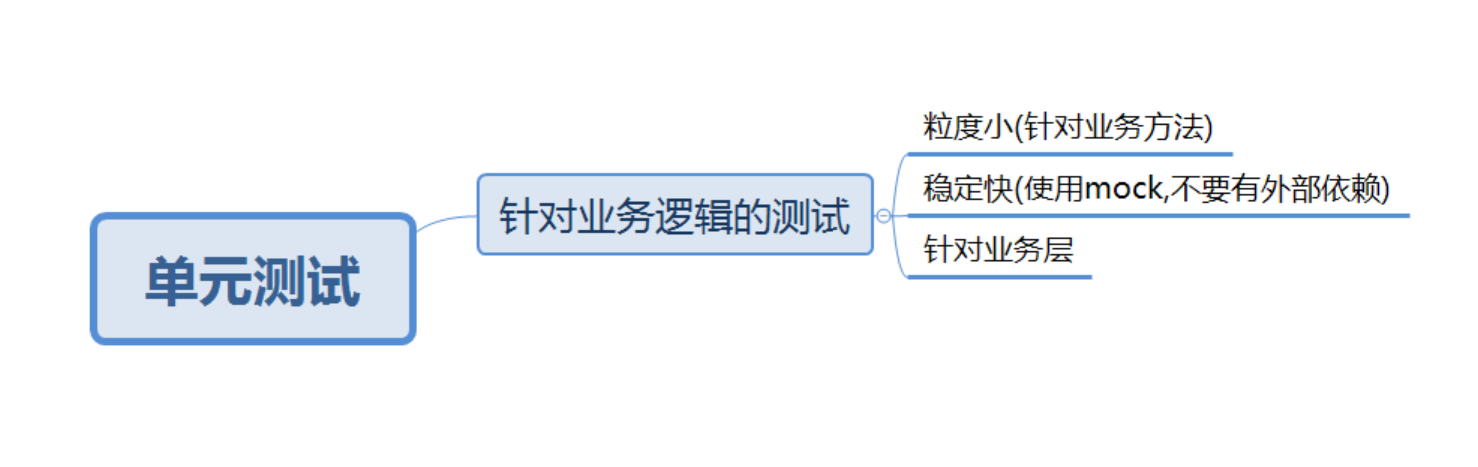


### 错误码规范



### 测试规范

#### 单元测试



#### 集成测试



## 供应链金融系统

### 基础服务

|  |  |
| --- | --- |
| id-service | ID生成器 |
| file-service | 文件服务器 |
| enum-service | 枚举服务 |
| sms-service | 信息发送服务 |
| activity-service | 流程引擎 |
| rule-service | 规则引擎 |
| config-service | 参数配置 |
| mq-service | 消息配置 |
| es-service | 查询管理 |
| report-service | 报表管理 |
| log | 日志追踪系统 |

### 业务服务系统

|  |  |
| --- | --- |
| sso-service | 单点登录系统 |
| login-service | 用户登录管理系统 |
| oauth-service | 权限管理系统 |
| corp-service | 企业管理系统 |
| document-service | 企业档案管理系统 |
| antifraud-service | 反欺诈系统 |
| data-service | 三方数据系统 |
| risk-service | 风控系统 |
| account-service | 账户管理系统 |
| credit-service | 授信管理 |
| product-service | 金融产品系统 |
| item-service | 商品管理系统 |
| order-service | 订单管理系统 |
| work-service | 工单系统 |
| bill-service | 账单系统 |
| transaction-service | 账务系统 |
| fund-service | 资金管理系统 |
| route-service | 渠道路由系统 |
| payment-service | 支付系统 |
| invoice-service | 开票系统 |
| fee-service | 费用系统 |
| after-service | 贷后系统 |
| alert-service | 告警系统 |
| vip-service | 会员管理系统 |
| marketing-service | 营销管理系统 |
| crm-service | 客户关系管理系统 |
| contract-servie | 合同协议管理系统 |
| portrait-service | 用户画像服务 |

### 外围服务系统

|  |  |
| --- | --- |
|  | 同盾数据 |
|  | 魔蝎数据 |
|  | 企查查数据 |
|  | 法海数据 |
|  | e签宝 |
|  | 鹏元征信 |
|  | 工商银行等银行 |

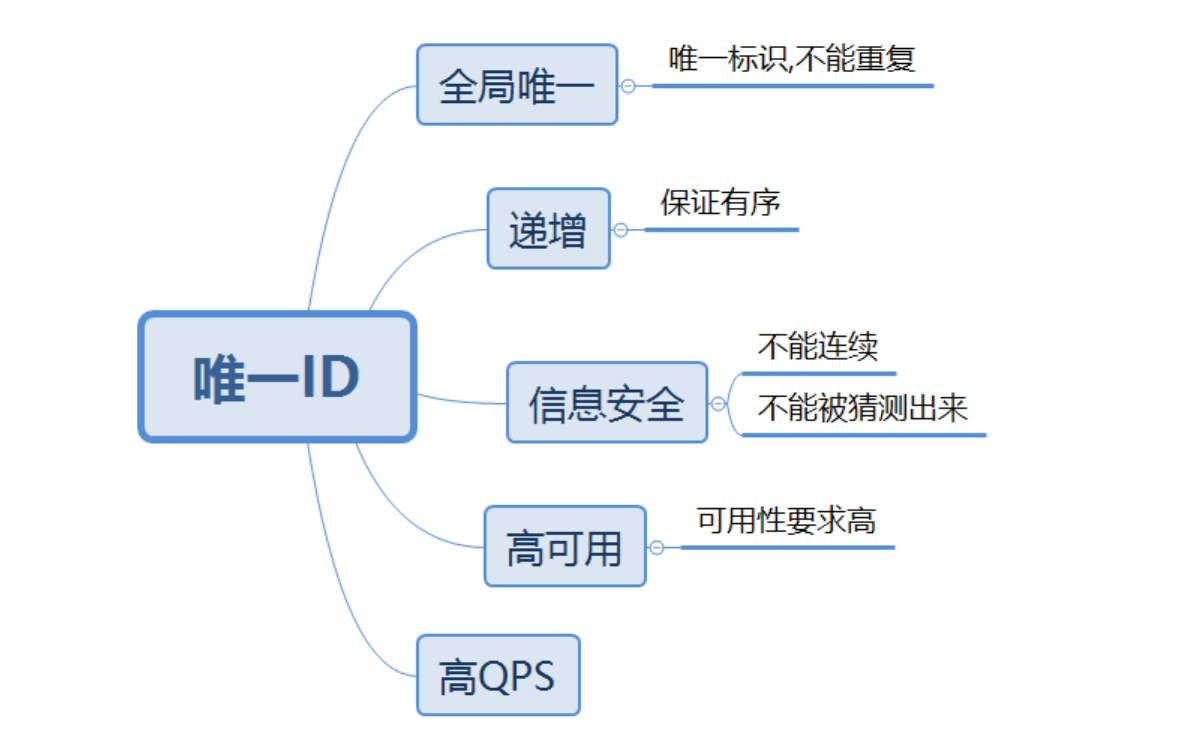
## 基础服务详情

### ID生成器(id-service)

#### 背景

在复杂的分布式系统中，需要对数据或者消息进行唯一标识，主要为了分库分表或者进行消息幂等等

#### 需求

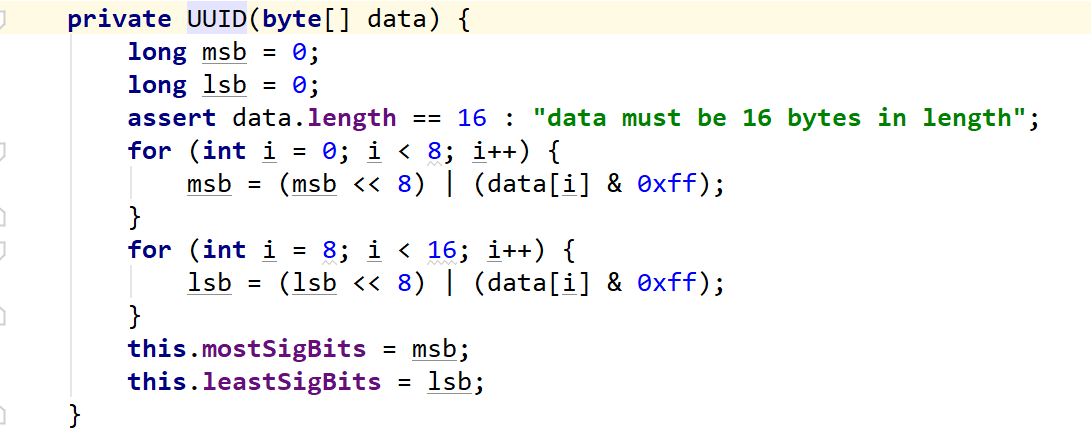


#### 实现

##### UUID

UUID的标准形式包含32个16进制数字，形式如下

550e8400-e29b-41d4-a716-446655440000



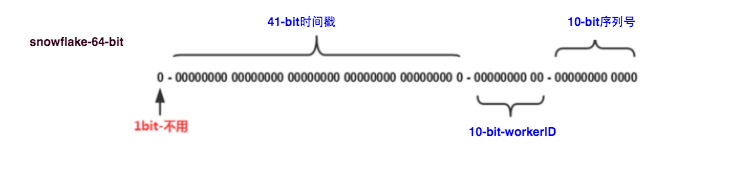
优点：

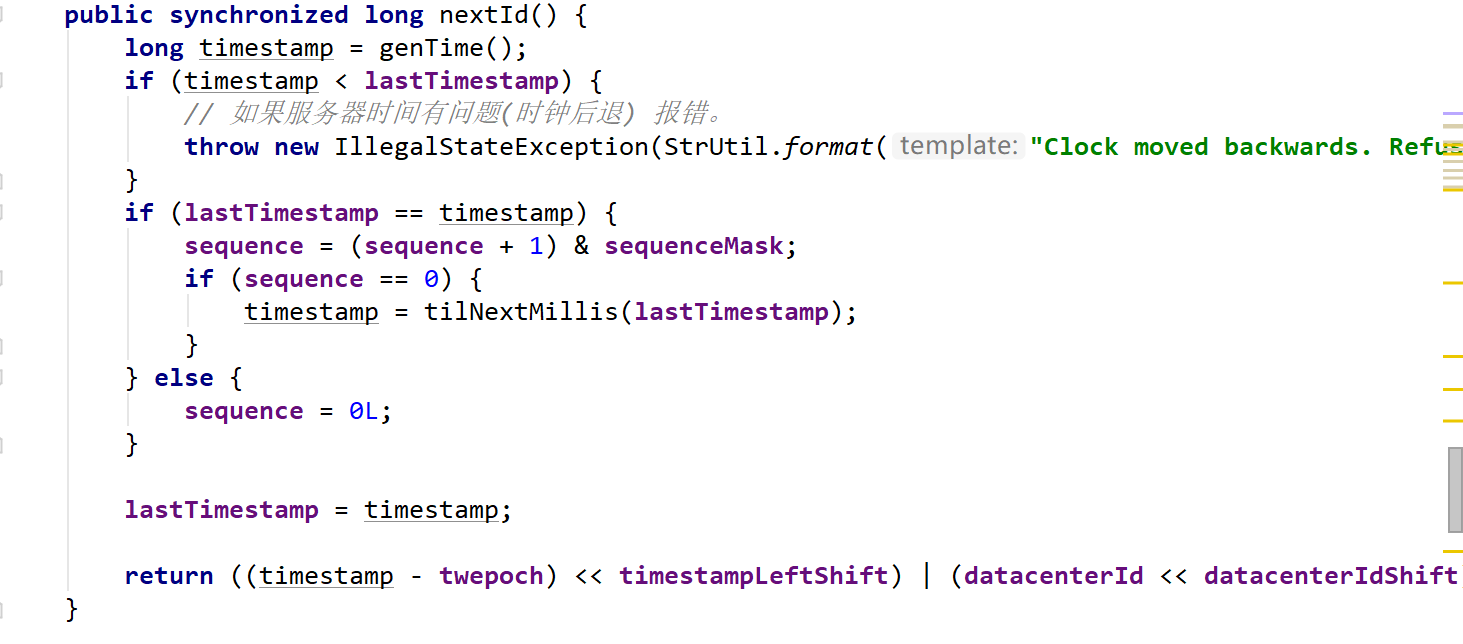
性能高

缺点：

不易存储：作为数据库主键不适合

##### 类snowflake



它给每台机器分配一个唯一标识，然后通过时间戳+标识+自增实现全局唯一ID。

Snowflake算法采用41bit毫秒时间戳，加上10bit机器ID，加上12bit序列号，理论上最多支持1024台机器每秒生成4096000个序列号

优点：

毫秒数在高位，自增序列在低位，整个ID都是趋势递增的

可以根据自身业务特性分配bit位

不依赖数据库等第三方系统，以服务的方式部署，稳定性更高

缺点:

强依赖机器时钟，如果机器上时钟回拨，会导致发号重复或者服务会处于不可用状态。

应用举例：Mongdb objectID

通过“时间+机器码+pid+inc”共12个字节，通过4+3+2+3的方式最终标识成一个24长度的十六进制字符。

##### 数据库

|  |
| --- |
| begin;  REPLACE INTO Tickets64 (stub) VALUES ('a');  SELECT LAST\_INSERT\_ID();  commit; |

优点：

非常简单，利用现有数据库系统的功能实现，成本小

ID号单调自增

缺点：

强依赖DB，当DB异常时整个系统不可用，属于致命问题

ID发号性能瓶颈限制在单台MySQL的读写性能

##### 如何实现id-service的高可用

使用Zookeeper持久顺序节点(PERSISTENT\_SEQUENTIAL)的特性完成对snowflake节点配置wokerID

1. 启动id-service,连接ZK,检查是否在id-service节点下注册
2. 如果注册过,取回自己的workerID,启动服务
3. 如果没注册过,如果在id-service节点下注册成功,取顺序号作为workerID

### ****文件服务器(file-service)****

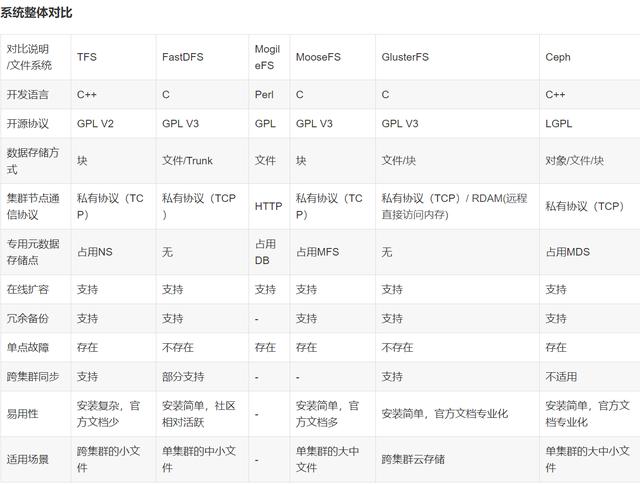
#### 背景

数据是数据服务型公司的最有价值的资源，分布式文件存储是应对数据爆炸的最好解决方案

#### 需求

那么该如何处理大数据的存储，计算分析呢？

#### 实现



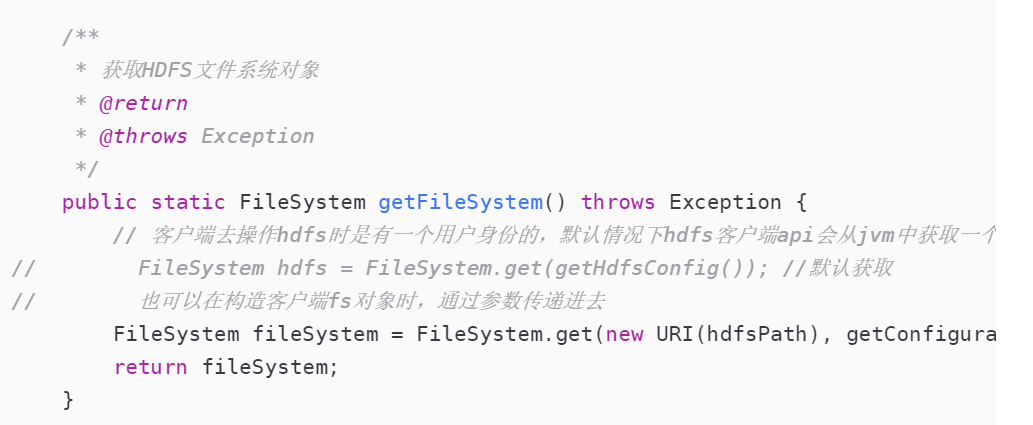
##### GFS（Google File System）

Google公司为了满足本公司需求而开发的基于Linux的专有分布式文件系统。没有开源。

##### HDFS

Hadoop 实现了一个分布式文件系统（Hadoop Distributed File System），简称HDFS。 Hadoop是Apache Lucene创始人Doug Cutting开发的使用广泛的文本搜索库。它起源于Apache Nutch，

后者是一个开源的网络搜索引擎，本身也是Luene项目的一部分。Aapche Hadoop架构是MapReduce算法的一种开源应用，是Google开创其帝国的重要基石。



##### TFS(Taobao FileSystem)

**3.TFS**

TFS（Taobao FileSystem）是一个高可扩展、高可用、高性能、面向互联网服务的分布式文件系统，主要针对海量的非结构化数据，它构筑在普通的Linux机器 集群上，可为外部提供高可靠

和高并发的存储访问。TFS为淘宝提供海量小文件存储，通常文件大小不超过1M，满足了淘宝对小文件存储的需求，被广泛地应用 在淘宝各项应用中。它采用了HA架构和平滑扩容，保证了整个文件系统的可用性和扩展性。同时扁平化的数据组织结构，可将文件名映射到文件的物理地址，简化 了文件的访问流程，一定程度上为TFS提供了良好的读写性能。

Google学术论文，这是众多分布式文件系统的起源，**HDFS和TFS都是参考Google的GFS设计出来的。**

典型的分布式文件存储的架构设计

我以hadoop的HDFS为例，毕竟开源的分布式文件存储使用的最多。

Hadoop分布式文件系统(HDFS)被设计成适合运行在通用硬件(commodity hardware)上的分布式文件系统。HDFS是一个高度容错性的系统，适合部署在廉价的机器上。HDFS能提供高吞吐量的数据访问，非常适合大规模数据集上的应用。HDFS放宽了一部分POSIX约束，来实现流式读取文件系统数据的目的。

**大规模数据集**

运行在HDFS上的应用具有很大的数据集。HDFS上的一个典型文件大小一般都在G字节至T字节。因此，HDFS被调节以支持大文件存储。它应该能提供整体上高的数据传输带宽，能在一个集群里扩展到数百个节点。一个单一的HDFS实例应该能支撑数以千万计的文件。

**简单的一致性模型**

HDFS应用需要一个“一次写入多次读取”的文件访问模型。一个文件经过创建、写入和关闭之后就不需要改变。这一假设简化了数据一致性问题，并且使高吞吐量的数据访问成为可能。Map/Reduce应用或者网络爬虫应用都非常适合这个模型。目前还有计划在将来扩充这个模型，使之支持文件的附加写操作。

**异构软硬件平台间的可移植性**

HDFS在设计的时候就考虑到平台的可移植性。这种特性方便了HDFS作为大规模数据应用平台的推广。

**Namenode 和 Datanode**

HDFS采用master/slave架构。一个HDFS集群是由一个Namenode和一定数目的Datanodes组成。

Namenode是一个中心服务器，负责管理文件系统的名字空间(namespace)以及客户端对文件的访问。

集群中的Datanode一般是一个节点一个，负责管理它所在节点上的存储。HDFS暴露了文件系统的名字空间，用户能够以文件的形式在上面存储数据。从内部看，一个文件其实被分成一个或多个数据块，这些块存储在一组Datanode上。

Namenode执行文件系统的名字空间操作，比如打开、关闭、重命名文件或目录。它也负责确定数据块到具体Datanode节点的映射。Datanode负责处理文件系统客户端的读写请求。在Namenode的统一调度下进行数据块的创建、删除和复制。



Namenode和Datanode被设计成可以在普通的商用机器上运行。这些机器一般运行着GNU/Linux操作系统(OS)。HDFS采用Java语言开发，因此任何支持Java的机器都可以部署Namenode或Datanode。由于采用了可移植性极强的Java语言，使得HDFS可以部署到多种类型的机器上。一个典型的部署场景是一台机器上只运行一个Namenode实例，而集群中的其它机器分别运行一个Datanode实例。这种架构并不排斥在一台机器上运行多个Datanode，只不过这样的情况比较少见。

分布式存储的未来

随着现代社会从工业时代过渡到信息时代，信息技术的发展以及人类生活的智能化带来数据的爆炸性增长，**数据正成为世界上最有价值的资源。**

根据物理存储形态，数据存储可分为集中式存储与分布式存储两种。集中式存储以传统存储阵列（传统存储）为主，分布式存储（云存储）以软件定义存储为主。

**传统存储**一向以可靠性高、稳定性好，功能丰富而著称，但与此同时，传统存储也暴露出横向扩展性差、价格昂贵、数据连通困难等不足，容易形成数据孤岛，导致数据中心管理和维护成本居高不下。

**分布式存储：**将数据分散存储在网络上的多台独立设备上，一般采用标准x86服务器和网络互联，并在其上运行相关存储软件，系统对外作为一个整体提供存储服务。。

总之，分布式文件存储，不仅提高了存储空间的利用率，还实现了弹性扩展，降低了运营成本，避免了资源浪费，更适合未来的数据爆炸时代场景。

### 信息服务系统(sms-service)

消息发送

(短信) (邮箱)(钉钉消息)等第三方消息

### 流程引擎(activity-service)

### 规则引擎(rule-service)

### 参数配置(config-service)

### 消息配置(mq-service)

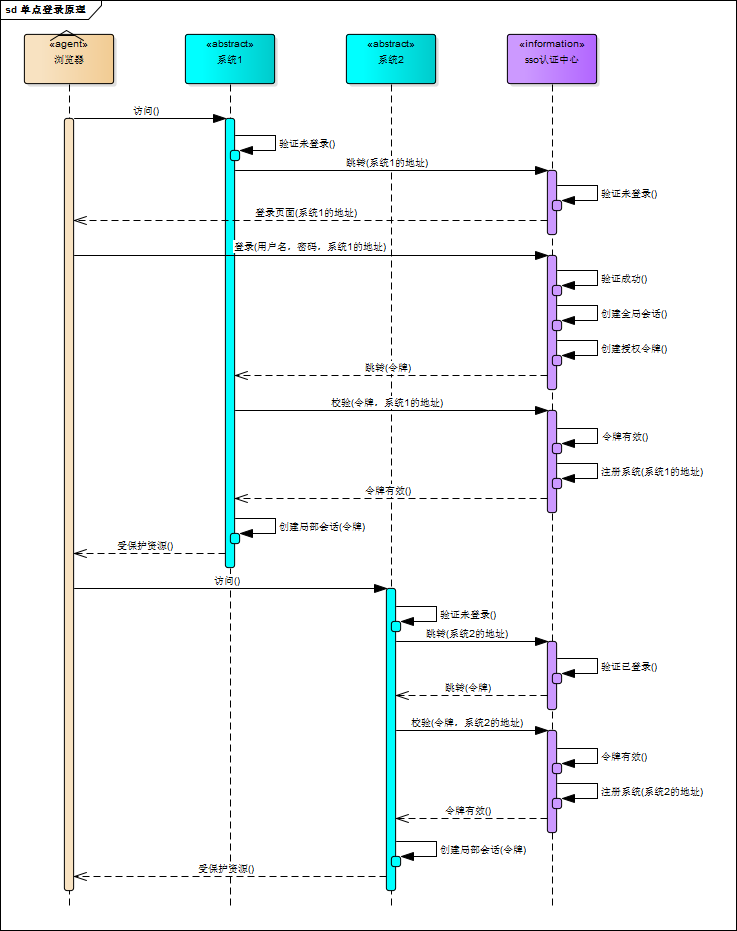
### 查询管理(es-service)

### 报表管理(report-service)

### 日志追踪(log)

## 业务服务系统详情

### 单点登录系统(sso-service)



什么是单点登录？单点登录全称Single Sign On（以下简称SSO），是指在多系统应用群中登录一个系统，便可在其他所有系统中得到授权而无需再次登录，包括单点登录与单点注销两部分

用户访问系统1的受保护资源，系统1发现用户未登录，跳转至sso认证中心，并将自己的地址作为参数

sso认证中心发现用户未登录，将用户引导至登录页面

用户输入用户名密码提交登录申请

sso认证中心校验用户信息，创建用户与sso认证中心之间的会话，称为全局会话，同时创建授权令牌

sso认证中心带着令牌跳转会最初的请求地址（系统1）

系统1拿到令牌，去sso认证中心校验令牌是否有效

sso认证中心校验令牌，返回有效，注册系统1

系统1使用该令牌创建与用户的会话，称为局部会话，返回受保护资源

用户访问系统2的受保护资源

系统2发现用户未登录，跳转至sso认证中心，并将自己的地址作为参数

sso认证中心发现用户已登录，跳转回系统2的地址，并附上令牌

系统2拿到令牌，去sso认证中心校验令牌是否有效

sso认证中心校验令牌，返回有效，注册系统2

系统2使用该令牌创建与用户的局部会话，返回受保护资源

　　用户登录成功之后，会与sso认证中心及各个子系统建立会话，用户与sso认证中心建立的会话称为全局会话，用户与各个子系统建立的会话称为局部会话，局部会话建立之后，用户访问子系统受保护资源将不再通过sso认证中心，全局会话与局部会话有如下约束关系

局部会话存在，全局会话一定存在

全局会话存在，局部会话不一定存在

全局会话销毁，局部会话必须销毁

　　你可以通过博客园、百度、csdn、淘宝等网站的登录过程加深对单点登录的理解，注意观察登录过程中的跳转url与参数

2、注销

　　单点登录自然也要单点注销，在一个子系统中注销，所有子系统的会话都将被销毁，用下面的图来说明

　　sso认证中心一直监听全局会话的状态，一旦全局会话销毁，监听器将通知所有注册系统执行注销操作

　　下面对上图简要说明

用户向系统1发起注销请求

系统1根据用户与系统1建立的会话id拿到令牌，向sso认证中心发起注销请求

sso认证中心校验令牌有效，销毁全局会话，同时取出所有用此令牌注册的系统地址

sso认证中心向所有注册系统发起注销请求

各注册系统接收sso认证中心的注销请求，销毁局部会话

sso认证中心引导用户至登录页面

---------------------

版权声明：本文为CSDN博主「Leon04095」的原创文章，遵循CC 4.0 by-sa版权协议，转载请附上原文出处链接及本声明。

原文链接：https://blog.csdn.net/lchq1995/article/details/79527490

sso采用客户端/服务端架构，我们先看sso-client与sso-server要实现的功能（下面：sso认证中心=sso-server）

　　sso-client

拦截子系统未登录用户请求，跳转至sso认证中心

接收并存储sso认证中心发送的令牌

与sso-server通信，校验令牌的有效性

建立局部会话

拦截用户注销请求，向sso认证中心发送注销请求

接收sso认证中心发出的注销请求，销毁局部会话

　　sso-server

验证用户的登录信息

创建全局会话

创建授权令牌

与sso-client通信发送令牌

校验sso-client令牌有效性

系统注册

接收sso-client注销请求，注销所有会话

接下来，我们按照原理来一步步实现sso吧！

### 用户登录管理系统(login-service)

用户多种方式登录系统

### 用户权限管理系统(oauth-service)

系统的使用者通常称为用户，用户登录后，系统通过各种标识赋予用户操作和查看权限。

后台需要对用户账户、操作权限和数据权限等进行管理。用户管理贯穿业务各个环节，是支撑业务运营的核心部分。

本文对后台用户管理进行总结，期望对大家有所帮助。

一、外部用户

系统用户一般分为内部用户和外部用户。这种区分是从平台运维角度出发，平台运维公司的员工称为内部用户，其他用户称为外部用户。

外部用户隶属关系比较复杂，有的挂靠在某公司之下，无需注册，由系统管理员分配账户。

有的则以独立个体存在，例如：钉钉，以独立个体注册，公司邀请员工，员工同意后才有了归属。

第三种情况是用户以独立个体注册时选择所属公司，这种方式存在两个问题：

1. 用户随意选择企业，不能保证准确性；
2. 是否要在后台维护企业基础数据，不维护的话，无法统一数据源，维护的话，不能覆盖全部注册用户的企业。

选择哪种方式，需要根据实际业务情况进行设计。

目前平台的服务对象是门店销售员，由于业务冲突，平台无法获得企业级的合作，前两种情况需要隶属公司在平台上进行操作，只能选择第三种方式，用户以独立个体注册平台，并填写所属企业。

为避免第三种方式存在的问题，做如下处理：

1. 用户注册填写企业，无需选择基础数据；
2. 后台维护企业基础数据；
3. 用户审核时，有平台客服人员选择企业基础数据。

这样既减少了用户操作的繁琐，平台又能获得精准的数据。

二、内部用户

内部用户也有归属，旧有的平台添加员工时选择归属，在员工管理菜单下展示为员工列表，这样做需要提前维护组织机构，频繁切换菜单。

像钉钉、企业微信等，在维护员工时，按照公司的组织架构进行维护，在同一页面完成部门和员工的添加，层级关系清晰（见下图）：

内部用户一般无需注册，由系统管理员分配。分配的账户可以是字符串，也可以是员工的手机号码。



三、角色与权限

每个账号，都被赋予了特定的角色；而每个角色的背后，都有其对应的权限信息。

通过RBAC权限管理模型，用户可按照实际业务的需要分配不同的角色和权限，在共享一个软件平台的基础上，实现不同用户的不同功能。

权限的赋予分为自动赋权和手动赋权两种。

外部用户的自动赋权一般是账户有默认的基础权限，要想获得更多的权限需要另外开通；内部用户的自动赋权，一般是把角色与行政关系下的部门建立绑定关系，用户进入到该部门后，账户自动被加入到对应的角色中，并且拥有该角色所有的权限。

手动赋权无法通过用户行政关系自动绑定来实现，需要手动建立角色赋予给账户。

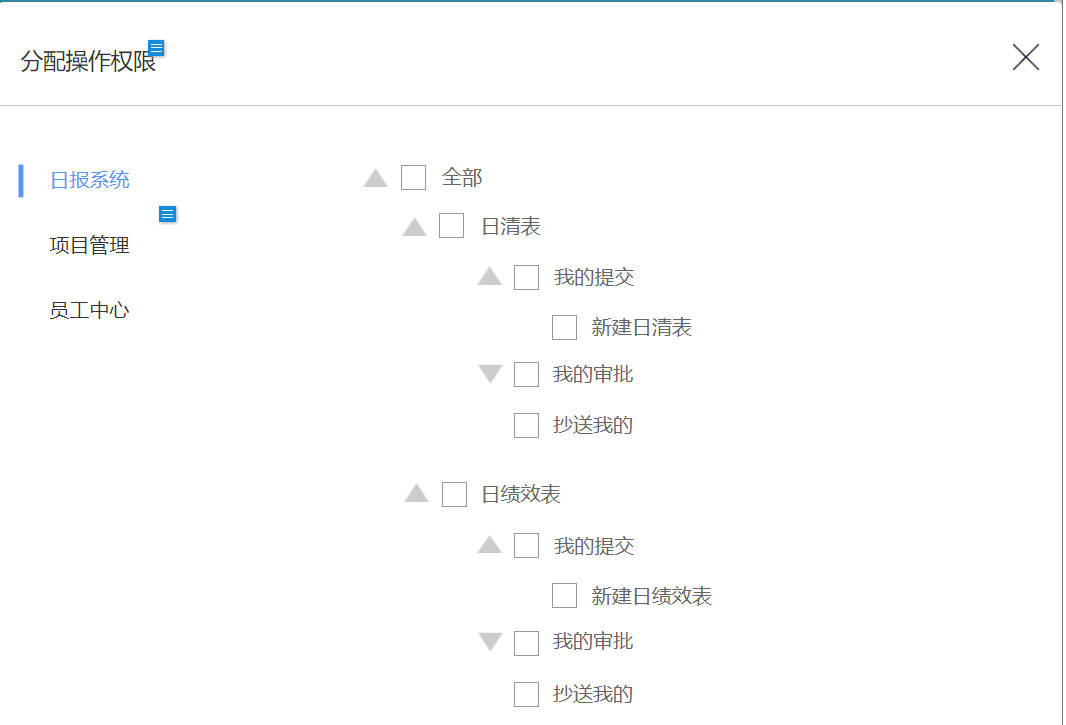
例如：要求客服专员只能看分到自己名下的客户，上级领导看全部下级的客户，通过两种方式实现：

1. 一是通过组织机构的上下级归属自动赋予数据权限；
2. 另外一种用数据角色，实现跨部门查看数据。

另外，角色可以继承，子账户继承父系角色的权限。例如：赋予某企业级外部用户admin某些权限，该企业下的其他子账户只能从admin权限中选择可用权限。

权限依据属性可分为操作权限和数据权限。

操作权限会从功能菜单、功能操作两方面考虑，角色勾选上菜单和功能，拥有该角色的账户即可操作相应的菜单和菜单下的操作（例如新、增、改、查），分配操作权限页面见下图：



数据权限会从菜单、列表及数据字段三个颗粒度考虑：

1. 菜单是最粗颗粒度的数据权限，获得授权即可查看该菜单下的全部数据；
2. 列表是较细颗粒度的权限，不同角色的用户进入同一菜单页后，查看的列表字段相同，但量级不同，例如上面描述的员工看自己的数据，领导看全部员工数据；
3. 数据字段是最细颗粒度的权限，不同角色的用户进入同一菜单页后，可见的数据字段有差异，这些字段共存在一个菜单页中，只是受限于不同的角色权限。

四、单点登录

平台会涉及多个应用系统，例如：合同管理、资金管理、小程序管理、兑换商城等，可通过单点登录（SSO）。

在多个应用系统中，实现登录一次就可以访问所有相互信任的应用系统。

应用系统加入了单点登录协议，管理用户帐号的负担就会减轻，实现应用系统的用户、角色和组织机构统一化管理。

要实现SSO，需要：

1. 统一集中的用户身份管理系统。所有应用系统共享一个身份认证系统，用户登录时，信息和用户信息库相比较，对用户进行登录认证；认证成功后，认证系统生成统一的认证标志（ticket），返还给用户。另外，认证系统还应该对ticket进行效验，判断其有效性。
2. 所有应用系统能够识别和提取ticket信息。应用系统对ticket进行识别和提取，通过与认证系统的通讯，自动判断当前用户是否登录过，从而完成单点登录功能。通过基于互联网协议的单点登录、登出体系实现用户不同应用系统间身份一致性，实现信息空间身份一致性。

不论是外部用户，还是内部用户管理，都涉及到账户、角色和权限的管理。

本文从自身经验出发，总结了各个环节需要注意的地方，另外单点登录涉及到用户管理，贯穿整个平台，此部分技术人员参与的比较多，本文只做了简要的总结。

用户管理系统最难的是后续的维护，系统中往往会出现很多冗余的角色，这都需要慎重思考，和业务沟通清楚后再做调整。

### 企业管理系统(corp-service)

#### 企业信息管理

### 企业档案管理系统(document-service)

#### 何为企业档案管理

文档是一个企业重要的资源,保存着公司很多重要的内容信息.在企业中,一般以电子文档的形式存在.从格式上来说分为doc文档PDF文档 txt文档png图片等从内容上，可能是商务合同、会议记录、产品手册、客户资料、设计文档、推广文案、竞争对手资料、项目文档、经验心得等。这些文档可能是过程性质的，也可能是公司正式发布的文档，可能处在编写阶段，也可能是已经归档不能再修改的。文档的状态包括草稿、正式、锁定、作废、归档、删除等。文件管理就是指这些文档、电子表格、图形和影象扫描文档的存储、分类和检索。文档管理的关键问题就是解决文档的存储、文档的安全管理、文档的查找、文档的在线查看、文档的协作编写及发布控制等问题。

文档分类 文档上传 文档下载 文档解析 文档查找 文档在线查看

文档版本控制

文件分类 商务合同 会议记录 客户资料 设计文档

文件存储 文件系统 FileId

公司文件 分类 Company version cat file(每次取版本号最大的文件)

各种paser写法 解析成需要的数据格式

### 反欺诈系统(antifraud-service)

## 二、反欺诈服务

不同的金融应用场景，有着不同的业务流程和环节，需要设计不同的风险检查环节和风控策略，构建完成基于场景、事件和规则驱动下的欺诈风险判别服务，通过灵活配置就能满足不同场景下、不同业务环节欺诈风险判定的能力。

* 理财应用：激活事件、注册事件、登录事件、认证事件、绑卡事件、活动时间等场景；
* 借贷应用：注册事件、登录事件、绑卡事件、充值事件、授信事件、放款提现事件等场景。

通过反欺诈模型、用户行为分析、风险信息库和反欺诈规则库判定客户的信用度。

### ****2.1 评估维度****

**（1）身份评估模块**

* **设备异常评估：**设备绑定认证、设备相似度评估、跨机构设备注册。
* **地理位置评估：**GPS ip评估、陌生交易地区评估、跨地区跨机构、GPS定位与申请地址不符；
* **客户环境异常：**IOS越狱、安卓ROOT、公共WIFI、疑似木马应用；
* **习惯相似度评估：**大小写习惯、阅览时间、客户输入法行为；
* **客户身份核查：**姓名、身份证、手机号、人体活体验证。

**（2）信贷交易行为评估**

* 交易行为：交易时间习惯、交易频率、短时间交易数量、交易金额；

**（3）信用评估**

* 欺诈信息库：设备欺诈库、IP欺诈库、账号欺诈库；
* 失信信息库：信贷逾期名单、司法不良名单、多头申请多头负债名单、团伙欺诈名单；
* 欺诈关联图谱：欺诈关联图谱。

### 2.2 反欺诈策略

**（1）七大策略**



**（2）信贷全流程**

* **用户注册：**三要素核身、客户端环境检测、注册设备是否关联多用户；
* **登录：**异常设备登录检测、异常登录地检测、异常登录IP检测、登录异常人脸识别
* **开户绑卡：**四要素核身、人脸识别、设备是否绑定多银行卡开户、银行卡是否涉及欺诈；
* **申请授信：**欺诈名单对比、失信名单对比、多头申请贷款对比、多头逾期负债名单对比、关联人欺诈名单对比；
* **确认用信：**虚假用信效验、信用行为记录；
* **提现：**同卡同出监控。

## 贷前风险情况详细介绍

第三部分贷前风险情况是同盾报告的核心部分，包括5项核查内容，分别是个人基本信息核查、风险信息扫描、多平台借贷申请检测、关联人信息扫描、客户行为检测。

### ****1、个人基本信息核查****

主要是核验身份证号、手机号格式是否正确，身份证姓名和号码是否对应，是否伪冒身份，身份证和手机号归属地是否是风险集中地区，

### ****2、风险信息扫描****

是将客户的基本信息与系统的风险数据库比对，核查该客户是否已经进入系统的风险关注名单数据库，一般通过身份证号码或手机号进行信息匹配。

常见的风险关注名单类型有：异常借款、信用异常、机构代办、垃圾注册、信贷逾期、法院结案、失信被执行人等。

**（1）异常借款**

命中异常借款风险名单的人数最多，一般是由于操作不当，违反了同盾系统的反欺诈规则。目前，同盾系统的反欺诈规则多达上万条，稍不注意就会被系统纳入到异常借款风险关注名单。

比较常见的违规情况有：

1. 填写的个人信息多变；
2. 频繁变换申请设备或网络环境（用wifi申请网贷是大忌）；
3. 同时申请多家平台；
4. 手机号正常使用不足3个月；
5. 手机通讯录、通话记录有网贷黑名单人员；
6. 手机短信中有多个小贷公司发的信息；
7. 有浏览博彩等不良网站的记录；
8. 夜晚12:00至早上6:00申请借款；
9. 用“复制+粘贴”填写个人信息；
10. 使用技术性欺诈手段。

**（2）信用异常**

信用异常是指客户存在影响正常还款的异常信息，比如，以前有过信贷逾期记录、网贷待还款负债过高、工作频繁变动、收入不稳定、存在借东墙补西墙嫌疑等情况。

**（3）机构代办**

有些人缺乏网贷申请经验或为省事，会找专业的借款中介帮忙申请网贷。目前，市场上已经出现很多家贷款中介服务平台。同盾大数据风控系统针对机构代办情况，设置了专门的风控规则。

例如，同盾系统在某一被拒绝的用户中，关联出来了一个失信的身份证和设备，而且发现其设备有较多的申请行为，系统会认定存在机构代办嫌疑，那么，这个被关联出来的用户或将需要严格的人工审核，甚至可以直接拒绝。

**（4）垃圾注册**

垃圾注册是指恶意借款人利用自动化脚本、计算机程序或雇佣自然人的方式，模拟正常合法的用户在网站上进行注册，给网站的正常运营带来恶劣影响。同盾科技通过对注册行为进行监测，结合同盾的设备指纹技术，发现异常行为，保证网站的正常运营。

**（5）贷款逾期**

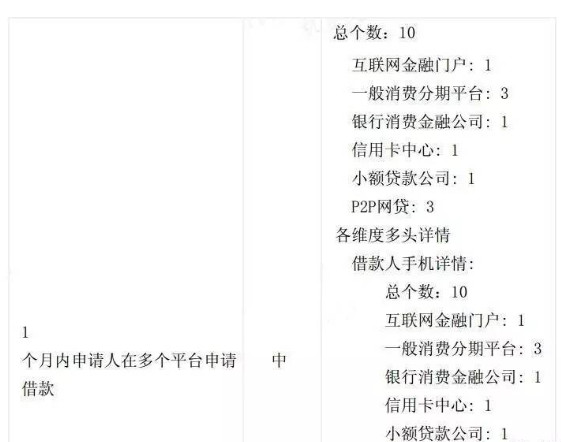
显示信贷逾期记录是大数据风控报告的重要功能之一。目前，同盾大数据风控系统主要用于互联网金融领域，主要记录网贷逾期情况，包括逾期发生的时间、金额范围、逾期天数以及逾期次数。

**（6）法院结案或失信被执行人**

目前，法院结案及失信被执行人信息都是公开的，同盾大数据风控系统可以利用网络爬虫技术从互联网收集到，并与信贷申请客户进行比对。

### ****3、多平台借贷申请检测****

此部分是同盾报告主体部分，体现了同盾科技跨行业跨平台联防联控的风控理念。一般根据近7天、1个月、3个月、6个月、12个月、18个月、24个月、60个月这八个时段，显示借款人在多个平台申请借款的记录，并依据不同时段借款次数的多少，划分高、中、低三个档次的风险等级，一般近7天内有大于等于两次申请记录就属于高风险，同时，还会显示申请借款的机构类型；如下图。



### ****4、关联人信息扫描****

主要核查借款人重要关联人的信用状况。一般借款都需要填写1-2个重要联系人，一般是配偶或直系亲属，配偶或直系亲属的信用状况对借款人信用也有很大影响，如果重要关联人的信用状况非常糟糕，也直接影响借款的借贷申请。

### ****5、客户行为检测****

就是异常行为监测。常见的异常行为如下：

* 3个月内身份证关联多个申请信息，一般3个月内，在不同放贷机构身份证关联的手机号、家庭地址、邮箱数大于等于两个，系统就显示异常行为信息；
* 3个月内身份证关联工作单位地址大于等于2，工作单位信息与借款人还款能力密切相关，这部分异常行为信息，系统会重点显示；
* 3个月内申请信息关联多个身份证，特别是同一个手机号关联多个身份证，有中介代办的嫌疑。

# 大数据反欺诈技术架构

[](https://www.jianshu.com/u/32df81e02fc2)

[Java大生](https://www.jianshu.com/u/32df81e02fc2) 关注

 0.2 2019.01.21 11:20 字数 2889 阅读 519评论 0喜欢 2

一年多以前，有朋友让我聊一下你们的大数据反欺诈架构是怎么实现的，以及我们途中踩了哪些坑，怎么做到从30min延迟优化到1s内完成实时反欺诈。当时呢第一是觉得不合适，第二也是觉得场景比较局限没什么分享的必要性。

时间也过了很久了，最近看到圈里一些东西，发现当时的这套架构并未落伍，依然具有很大的参考价值，所以今天跟大伙聊聊关于大数据反欺诈体系怎么搭建，主要来源是来自于我工作的时候的实践，以及跟行业里的很多大佬交流的实践，算是集小成的一个比较好的实践。

这套架构我做的时候主要领域是信贷行业的大数据反欺诈，后来也看过电商的架构，也看过金融大数据的架构，发现其实大家使用的其实也差不多是这个套路，只是在各个环节都有不同的细节。

大佬说的，能用图的，尽量不要打字，那我就打少点字，多做点图。其实大数据不外乎这么几个步骤。数据源开拓、数据抽取、数据存储、数据清洗和处理、数据应用，且听我一个一个说。

数据源

数据源是一个比较重要的点，毕竟如果连数据源都是垃圾，那么毫无疑问可以预见，最终产出的一定是垃圾，所以挑选数据源和对接数据源的时候都要关注，该机构产出的数据是不是都是质量比较高的数据。

比如人行征信数据就是一个质量非常非常高的数据，主要涉及信用卡、银行流水、老赖、失信、强制执行信息等，都非常核心，任何一个点都可能是一笔坏账的苗头。以及各种行政机构提供的付费机密数据。

比如运营商通讯数据、比如大型电商的行为数据、比如各种保险数据，以及各个机构贷款记录的互相沟通，这些数据源，都非常核心也都非常值钱，是现在反欺诈非常核心的数据。

当然也有更加粗暴更加高效的做法，就是直接购买外部的黑名单数据，这让反欺诈变得更加简单，遇到就直接拒，可以减少非常的人力物力成本去做其他的核查。

数据抽取

有了高质量的数据源后，当然就是怎么抽取的问题了，各个机构所提供的数据格式是多种多样的，其中包括 http 接口的json、xml，内部其他数据源的 etl、定时人工上报的 Excel，以及 sqoop+ozzie 这两个直接数据抽取通道，这个过程只需要保证通道稳定，数据服务幂等即可，没什么特殊的地方。

数据存储

数据存储其实就是建立一个数据仓库和一个实时仓库，数据仓库用于存储来自各大数据源的原始数据，实时仓库用于业务系统的核心作业，数据仓库的数据量一般都以 T 为单位，实时仓库以 M 和 G 为单位。

离线计算&实时计算

数据保证了，那么计算就是这套架构的核心之处，从大的角度来看可以分成离线计算和实时计算。

离线计算主要会做两件事情。Hive 、Spark 数据整合与清洗和离线数据建模。Hive 数据整合主要做的事情就是把各个数据库里面的东西，进行清洗和过滤，最终写到我们定义的标准表里边，提供给下游的计算使用。如果是非常复杂的数据清洗，我们会使用 Spark 写程序来做，毕竟有一些操作不是 Hive 这种标准 SQL 能解决的。离线数据建模，就是对于这批数据进行建模，以便后续用于实时计算和应用中，算法嘛，精通两个基本就稳了，LogisticRegression & 决策树，不要问我为什么。

实时计算又会做些什么事情？SparkStreaming和Flink用于实时流计算，主要是用于一些统计类的事情，以及多个数据流的 join 操作。在这里我们希望做到什么事情呢？就是希望服务可以准实时，什么叫准实时呢？就是在一个可以接受的范围内，我允许你有一定的延迟，这块我们一开始的任务延迟是 30 min。

我们踩过哪些坑呢？

一开始我们希望使用流批次计算来实现实时计算，实践下来准实时跟实时还是区别很大的，一个业务通常是允许不了分钟级别的延迟的，然而 Spark 的 GraphX 必然有分钟级别的延迟，只适合离线计算。

Hive + Ozzie 处理离线批量处理是一个非常大的利器，很多人都以为Hive数据清洗不就写写几行 SQL？几百张乃至几千张表背后的复杂的数据清洗规则，任务依赖，任务重跑，数据质量，数据血缘关系维护。相信我，要是没有细心和工具，这些能把你搞崩溃。

ElasticSearch 集群多个机器的负载吞吐量，比单台机器高性能的要高，毕竟网络卡在那。

我们趟了很多的坑，摸了很多的时候，最终决定把所有的实时操作都架构在 ElasticSearch 和 Neo4j 上，因为我们不仅仅需要实时的全文本全字段社交关系生成，更是需要实时搜索多维度多层社交关系并反欺诈分析，而这个关系可能是百万级别的，根据六度理论，决定了我们选取的层次不能太多，所以最终我们只抽取其中三层社交关系。最终确定这个架构，这很核心地确定了我们的响应时间，并最终决定了我们服务的可用性。

很多地方产生的结果数据只是整个决策链上的一个细节，所以我们还需要 Drools 这类规则引擎帮助做一个最终决策。

业务应用

最终业务系统应该怎么使用，怎么对外提供服务？这也是一个非常核心的问题，因为这部分要求非常非常稳定，以及非常非常高效，一般来说不允许有太高的延迟，同时还要求非常高的并发量。这就要求了我们第一要尽量提高计算效率，第二要求我们对于系统的架构要有非常高的保障。

计算效率要高效，有什么技巧呢，保证各个系统之间的交互都是聚合、加工、计算后的结果，而不是原始数据，毕竟网络传输是需要很高成本的在目标数据量非常大的场景下。比如一次性要加载几十万条数据，那全部拉回来再重新计算是不是就显得很蠢了？为什么不在目标系统里以数据服务的形式提供呢？

技术架构保障，其实大部分都是基础架构的事情了，比如动态负载均衡、一主多从、异地多机房容灾、断网演练、上游服务故障预案等等。

建模之社交网络

很久以前就已经介绍了各种社区发现算法，这里就不再赘述，有兴趣的自己点进去细致了解一下。

这里聊聊一个知识图谱的标准建立过程。

1、主体确认

2、关系建立。

3、逻辑推理建立。

4、图谱检索

主体确认，从图的角度来看就是确认各个顶点，这些顶点有自己的属性，代表着网络里的一个个个体。

关系建立，从其他资料关系得来，也可以根据第三步的逻辑推理得来，从图的角度来看就是确认各个边，这些边有起点有终点也有自己的属性，代表着网络里各个个体的关联。

逻辑推理建立，这是非常重要的一个部分，比如姚明的老婆的母亲，就是姚明的岳母，这种先验知识的推理可以在图谱的帮助下，为我们解决很多的实际问题。

图谱检索，有了图谱我们就开始使用，我们有四件套，主体属性搜索，关系属性搜索，广度优先搜索，宽度优先搜索。我们一般的使用策略都是，优先确定一个顶点比如目标人物，然后向外扩散，直到找到所有符合条件的个体。

这里我们踩了什么坑做了什么优化呢？我们一开始是把整个搜索结果拉到本地再进行计算，而图谱搜索后的结果总是很大，毕竟我们找了很多维的关系，所以总是卡在网络这块。经过探索和咨询，最终确认了 Neo4j 这种图数据库不仅仅提供数据查询服务，还做了基于定制化的社交网络分析的插件化开发，把我们的反欺诈服务以插件化的形式部署到服务器中，这就减少了非常多的网络开销，保障了我们服务的秒级响应。

完整架构图

从数据来源、获取、存储、加工、应用，一步到位，万一有点帮助那就更好了，如果还心存疑虑，这篇文章从下往上，再看一遍。

### 三方数据系统(data-service)

### 风控系统(risk-service)

风险管理的程序：

第一步是进行风险识别，识别存在怎样的风险；

第二步是进行风险估测，可能会造成多大的损失；

第三步是风险评价，发生概率有多大，是否是主要风险；

第四步是风险控制；第五步是对风控措施的效果评价，需要定期用标准去检验自己的风控措施是否合适，是否需要调整。

**互联网金融与传统金融的存在一定的差异，互联网金融有两重风险，第一重是互联网的风险，第二重是金融的风险。**

互联网里面常见风险主要有：信用风险，操作风险，政治性风险，流动性风险。

金融的业务流程风险管控整体流程有三个部分，贷前审核，贷中预警，贷后催收。



大数据平台的产品架构图：



风控产品主要有四大系统：

1、大数据征信系统：偏线下的自主查询大数据征信系统，通过各种数据维度对用户进行信用评估，标识风险信息。相比于人工筛选用户效率会更高。

2、风控决策系统：灵活配置的风控决策系统，满足互联网金融风控实时性、灵活性的要求；

3、反欺诈系统：基于关系图谱和图结构数据的离线反欺诈分析系统，发现风险用户的欺诈特征，并在风控决策系统中一键生成，打造风控闭环；

4、用户标签系统：挖掘用户的标签，包括实时定义标签和离线定义标签，从而让信用审核人员能够根据标签进行审单，提高效率，减少主观判断。

**个人服务是现金贷，消费金融，社区金融等**，有交易有增值，底层都是支付系统和统一用户体系，上层的功能业务，系统后台等都依托底层支付系统和统一用户体系来做规划。国内的风控模型核心的还是银行的风控模型，国内的投融资公司做的所谓风控模型也就是把目前的客户群体筛选出来，**把客户群体类型列出来，用几个维度分析：基本信息维度、身份维度、社会行为属性维度、行为偏好贡献值和价值属性这五个维度。**有了这些信息后台根据规则引擎计算之后才能做一个精准的推送。

创新金融产品的核心还是传统金融产品，只是在传统金融产品的基础上进行了包装，看起来像是创新金融产品，本质是不变的。

### 账户管理系统(account-service)

#### 何为账户

账户体系设计首先要区分两个概念，支付账户和登录账号。 这是两个不同业务领域的概念。 **支付账户** 指用户在支付系统中用于交易的资金所有者权益的凭证。 **登录账号** 指用户在系统中的登录的凭证和个人信息。 一个用户可以有多个登录账户，一个登录账户可以有多个支付账户，比如零钱账户，储值卡账户等。 一般来说，支付账户不会在多个登录账户之间共用。如果没有特殊说明，下文中的账户，都默认指支付账户。

#### 账户的需求设计

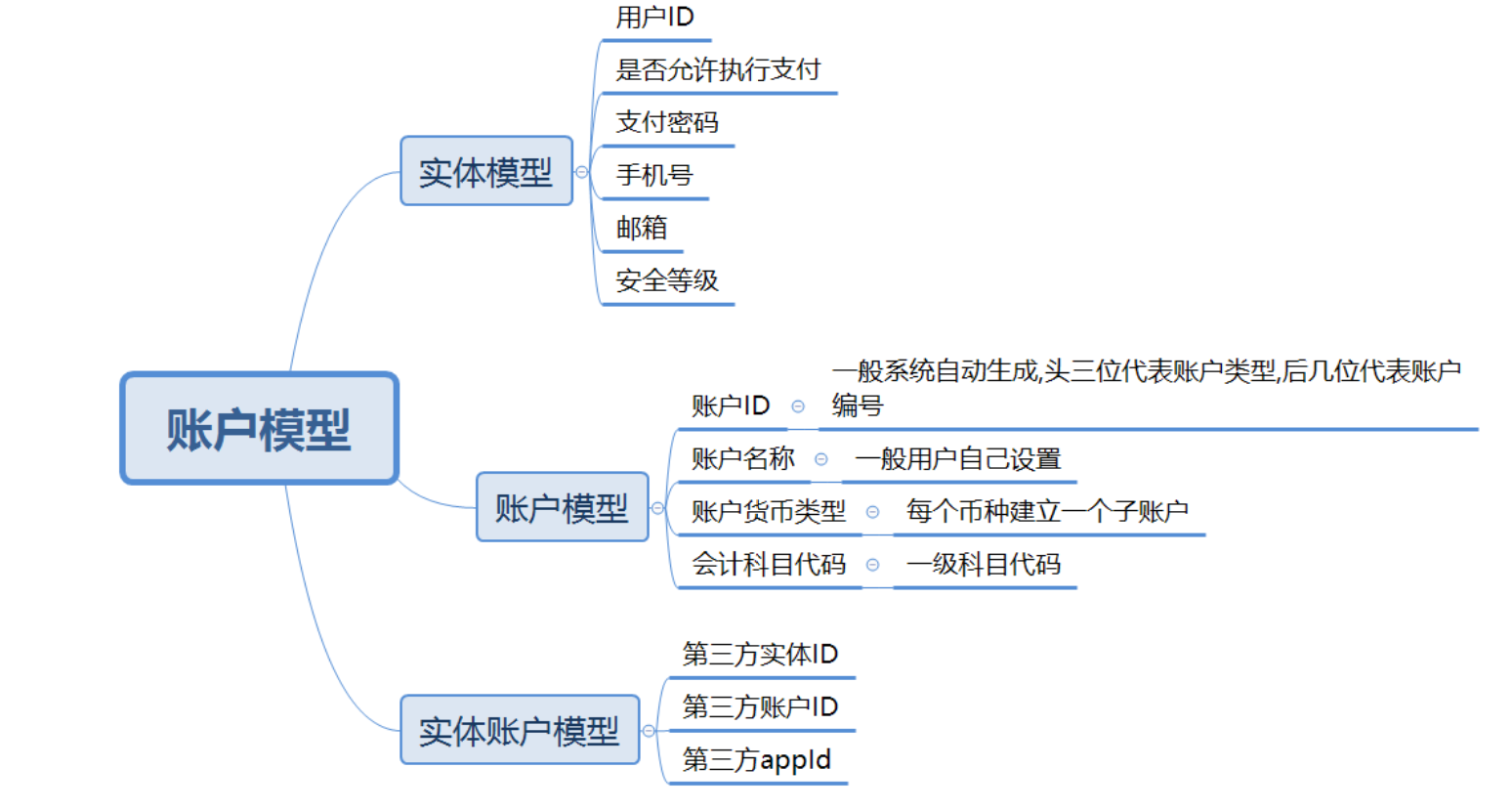


首先账户分为用户账户和业务账户。业务账户初始值为0，但是可以不断透支。一个用户完成某个动作，产生的一条从对应的业务账户到该用户对应账户的交易记录。至于具体是哪个账户，可以按照需求增加，比如积分账户、红包账户、人民币账户、提现账户等等。同理，交易记录也可分为红包、充值、转账、提现等等。

而交易记录根据是否需要审核，决定是直接生成流水记录，还是通过审核记录来生成流水记录。

#### 账户建模

账户建模本身包括三大核心模型：实体模型、账户模型和交易模型。 从交易模型中可以衍生出针对各个角色的账户流水，即明细模型，用于支持对账。



#### 交易设计

交易记录，交易流水，账户流水，交易台账，这三个容易混淆的概念，从数据上来说，却并不复杂，它们的核心是交易流水，账户流水是从账户视角的交易流水。那对一笔交易，涉及到的方方面面内容很多，有哪些需要记录的呢？考虑到交易记录将被用于风控和信用分析，能收集到的信息是越全面越好。



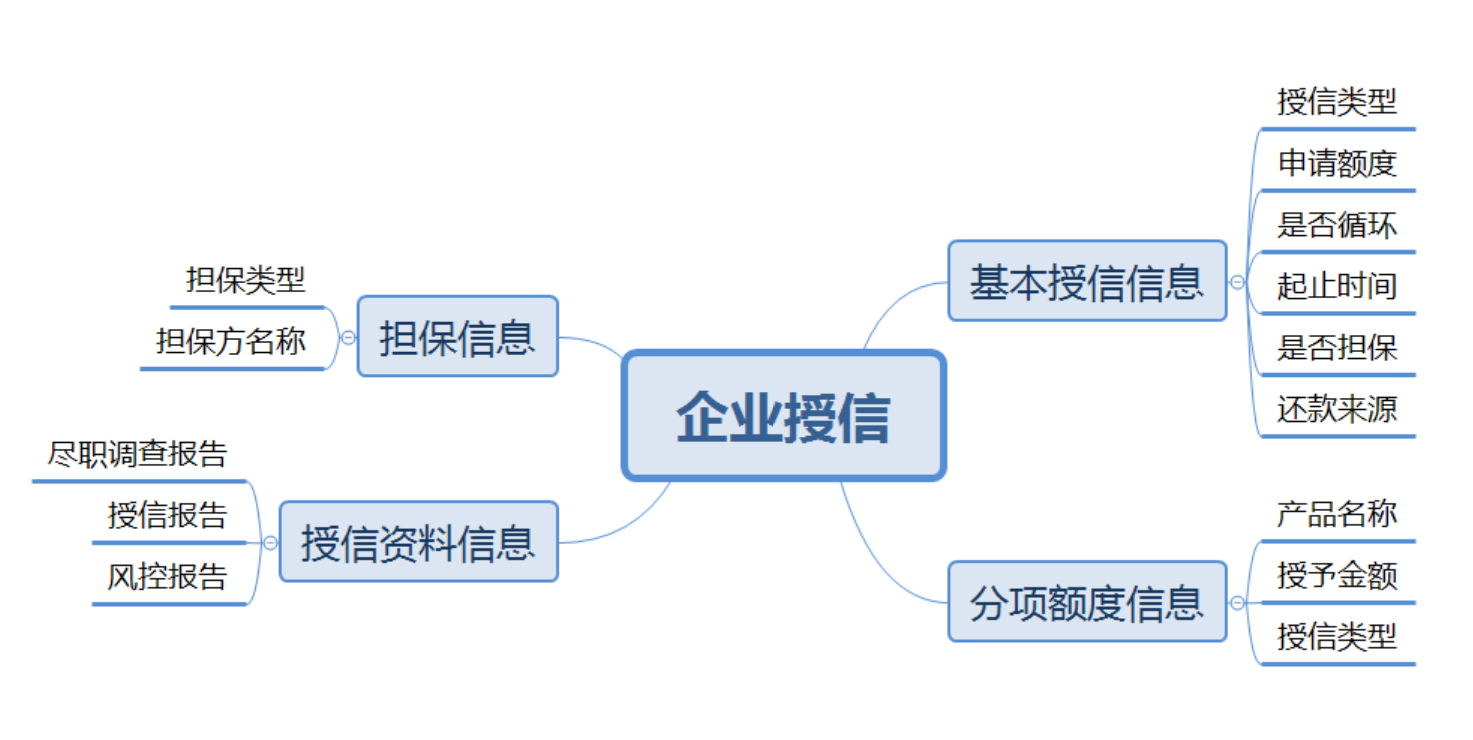
### 授信管理(credit-service)

#### 何为企业授信

简单来说，企业授信是指资金方(资金来源)向资产方(资金使用方)直接提供资金支持，授信不能等同于贷款，授信是一种风险控制的总的概念。当融资企业需要贷款时，贷款金额不能超过其企业的授信金额。

#### 授信内容概述

企业授信涉及到单个企业授信和集团企业授信。



#### 授信内容详解

##### 基本授信信息

**额度类型**:

新贷:企业第一次办理授信

增贷:本年内业务变更需要提额

年审:每年重新发起一次审核

变更:条件变更需要修改额度信息

**申请的额度**:

本次授信的金额(一般根据尽职调查报告等授信资料获取额度)

**是否循环**:

额度是否可以循环使用

循环授信与非循环授信的区别：

**循环授信：**授信500W，用信200W后，剩余额度为300W，当核心企业还款后，剩余额度变为500W。

**非循环授信：**授信500W，用信200W后，剩余额度为300W，当核心企业还款后，剩余额度为300W。

**额度的起止时间**:额度生效时间与额度到期时间

**是否担保**:授信过程中可以不提供担保企业或者担保人

**还款来源**:描述信息

##### 分项额度信息

分项额度指资金方针对每一项基础产品来对企业授信，一般情况下，资金方会有多种产品组合。

**产品名称**:基础产品

**授予金额**:该项产品的授予金额

**额度类型**:额度共享和额度独立

**额度共享：**指不同的产品授予金额可以小于等于企业的申请额度。

**额度独立：**指所有的产品授予总金额不能大于企业的申请额度。

企业申请额度为100W，额度共享指所有产品都可以是100W.额度独立指的是一个产品如果是50W，那么其他额度共享的产品只能是50W,了。相当于从100W中单独分配了50W给这个额度独立的产品

##### 担保信息

若申请信息有担保，则担保信息必填；若申请信息无担保，则担保信息可不填。

**担保类型**：企业担保/自然人担保

**担保方名称**:企业名称/法人名称

##### 授信资料

授信资料直接影响了整个业务环节的授信情况。

#### 授信功能设计

### 金融产品系统(product-service)

### 商品管理系统(item-service)

### 订单管理系统(order-service)

### 工单系统(work-service)

### 账单系统(bill-service)

### 账务系统(transaction-service)

### 资金管理系统(fund-service)

### 渠道路由系统(route-service)

### 支付系统(payment-service)

### 开票系统(invoice-service)

### 费用系统(fee-service)

### 贷后系统(afterloan-service)

### 告警系统(alert-service)

### 会员管理系统(vip-service)

### 营销管理系统(marketing-service)

### 客户关系管理系统(CRM-service)

### 合同协议管理系统(contract-service)

### 用户画像服务(portrait-service)

互联网金融的的核心在于风控系统，这篇文章就给大家讲下风控模块里边的用户信用画像构建、反欺诈服务。



## ****一、用户信用画像构建****

说到用户信用画像的构建对于整个风控体系的作用毋庸置疑，不同的金融平台可以根据自身的业务场景以及能力构建自己用户画像，毕竟有些画像的数据自身没有也很难从其他的三方平台获取，所以构建的时候要根据自身的业务场景和公司情况量力而行。

用户信用画像的组成体系包括但不限于以下几点：用户身份信息、婚恋社交数据、芝麻信用、用户认证数据、消费收支数据、用户行为数据、人行征信报告、互金和银行黑灰名单、设备相关数据等。



### 1.1 用户身份信息

该信息的获取是这九大数据里边比较容易获取到的数据，其中包括用户的三身或四身数据、居住地、婚姻状况、子女情况、工作单位、职位、房产、收入、联系人数据等，该数据用途多用于贷前信用评估、用户平台入驻前的认证。

### 1.2 婚恋社交数据

婚恋社交数据在整个信用体系里边承担着一个什么样的角色呢？

其他数据类型可以看出一个人的信用水平、还贷能力，消费水准、用户行为等等，但是涉及到人格品质方面的评估就略显乏力了，一个人在家庭的的责任、社交的言论、人生的规划、社交圈子等种种产生的行为的数据，更能反映一个在当下社会主流的人格品质范畴属于什么角色定义。

在婚恋社交情况里边有几点数据尤为重要：家庭情况、房产情况、学历情况、生活作息、爱情规划、爱情账户等级、婚恋社交信用度、社交人脉圈等。

### 1.3 芝麻信用

芝麻信用在现有的信用体系里边也是占据了一定的地位，很多涉及到资金一块的信用认证多数会将芝麻信用列为评判标准之一。

其中主要包括信用评分、行业关注名单、申请欺诈评分、欺诈信息验证、欺诈关注清单、企业信用评分。

### 1.4 用户认证数据

现在用户的认证数据主要有几块，公积金社保、运营商通信、学历数据（学信网）、职业数据（脉脉、猎聘、BOSS等职业招聘平台的认证数据）

### 1.5 消费收支数据

消费数据在几个数据里边是比重比较重的一个，关系到个人的经济水平、偿还能力、以及消费行为的判断，消费收支数据主要包括以下几点：线上电商和线下消费、银联消费、银行卡收支、航旅出行数据等。

### 1.6 用户行为数据

用户行为数据可以将申请表单的填写时间和借款协议页面停留时间，作为参考数据之一。

### 1.7 人行征信报告

人行征信的数据就不用多说了，是金融征信取证的重要依据之一，包括用户的贷款信息、信贷交易信息、个人公共信息。

### 1.8 互金和银行黑灰名单

互金和银行的黑灰名单可以作为搭建风控里边黑灰名单的一个重要依据，黑灰名单里边包含信贷逾期名单、司法不良名单、多头申请和多头负债名单、团队欺诈名单等。

## 信贷反欺诈的常用手段：名单库、专家策略、机器学习

欺诈从本质上来看是操作风险的一种，而信贷反欺诈的常用手段有名单库、专家策略、机器学习三种。信贷反欺诈的手段虽多，但是都必须建立在深入理解平台业务的基础之上。

低廉的造假成本和层出不穷的欺诈手段，给一个信贷机构带来的不仅仅是风险损失，更给信贷机构带来极大的挑战。

在整个信贷流程中，如何在贷前申请中准备快速地识别欺诈风险，将欺诈群体拒之门外是业务的重中之重。

## 一、信贷欺诈的类别

欺诈从本质上来看是操作风险的一种。

在信贷行业，据悉70%以上的风险来自欺诈风险，而欺诈形式多种多样，如身份造假、中介黑产、内外勾结等等。从欺诈主体来看，可以分为第一方欺诈、第二方欺诈、第三方欺诈。

第一方欺诈，主要是申请贷款本人恶意骗贷、还款意愿极低、拒绝还款等；第二方欺诈是指内部欺诈或内外勾结；而第三方欺诈主要是盗用冒用他人身份、他人账号以及团伙欺诈等。

这其中：团伙欺诈已形成一个黑色产业链，黑中介通过购买个人信息、和客户联合等手段进行欺诈。

所以，信贷反欺诈就是和欺诈人员斗智斗勇的过程：欺诈分子一直在寻找业务的漏洞，而反欺诈人员则需要在不断变化的漏洞中打上一个个“补丁”。

那么，该如何在贷前阶段做好申请反欺诈呢？

其实要解决的问题无非就是判断申请借款的是人还是机器？是本人还是他人？是价值用户还是无效用户？

目前贷前反欺诈常用的手段有名单库、专家策略、机器学习等。

## 二、名单库筛选

名单库筛选就是我们常说的黑白名单。名单库一般通过平台内部进行积累，或与其他合作机构合作进行获取。黑名单在很大程度上避免了重复欺诈行为的发生，也是一种逻辑简单、成本较低的反欺诈手段。

当然，黑名单覆盖群体较小、需要时间积累，也存在准确率较低、名单库易污染等缺点，但是可以作为反欺诈的第一道过滤。

同理，白名单一般指平台内部的优质客户列表，建立白名单库可以有效且降低公司的成本和信用风险，提高放款效率。

## 三、专家策略

贷前反欺诈一般都是先有专家策略进行冷启动，等数据积累到一定程度的时再慢慢地对数据进行挖掘，并对策略进行调优或者构建模型。

很多人都觉得专家策略不过是“拍脑袋”，其实反欺诈策略往往基于策略人员以往的经验和踩过的“坑”，并以研究欺诈者的行为和心理为基础而制定。

而且，目前的信贷反欺诈手段中，专家策略比较常用且较为成熟。当借款人的操作请求和操作行为触发反欺诈规则、并达到一定的程度时，即被认定为欺诈行为。合作方可以启动拦截，或进行人工审核，如客户的行为异常监测策略、设备类异常策略、聚集度策略等。

现在欺诈手段日新月异，欺诈人员和策略人员处于攻与防的角色，如果无法在第一时间做出反应，需要事后进行大量的数据分析和挖掘后才能提取新的特征和规则。

而专家策略往往存在一定程度上的误杀率，而误杀率的高低取决于策略人员的经验水平，不同的策略人员制定的专家策略也会存在较大的区别，呈现不同的效果。

此外，策略需要不定期进行更新，并要严格保密，一旦泄露将对平台造成不可挽回的损失。因此，专家策略实现简单，可解释性强，但会存在滞后性。

顶象技术在贷前反欺诈策略方面有较为丰富的策略模板，通过对客户个人信息、设备指纹、操作行为、位置等各个维度进行欺诈识别，帮助合作方识别出风险较高的客户，力争将合作方的欺诈风险降至最低。

## 四、机器学习

近年来，机器学习在反欺诈方面的应用越来越广。

常见的机器学习反欺诈分为有监督和无监督两种，它们通过机器学习方法，收集客户各个维度的数据，结合当前用户特征，与欺诈建立起关联关系，实时识别用户欺诈行为。

有监督机器学习反欺诈是目前机器学习反欺诈中较为成熟的一种方法，它通过大量客户的历史表现数据，进行标签化，并利用相关算法，提取特征，发现欺诈行为的共同点，进行识别。

而无监督机器学习反欺诈则相对较新，只是通过对用户的各纬度数据特征的聚类，找出与大多数用户和行为差异较大的用户和操作请求，并予以拦截。采用无监督机器学习，可以有效地识别团伙欺诈行为，让欺诈团伙无处遁行。

俗话说：“不管黑猫白猫，能抓到老鼠的就是好猫”，无论是专家策略、还是机器学习，无论是有监督还是无监督，都有其优劣性，没有所谓的好坏之分，只有合适与否。总之，反欺诈的方法虽然很多，但信贷反欺诈必须建立在深入理解平台业务的基础之上。

### 同盾

## 同盾分

同盾贷前审核报告中，同盾分范围是0-100分。评分分三个档次，0-20低风险区，系统建议通过审核；20-80分，较大风险区，系统建议进行人工审核；80-100分，系统检测出高危风险，建议直接拒绝。

虽然通常80-100分区域的客户被称为网贷黑名单客户（网贷黑户），但很多网贷公司实践中已经把“黑户”标准减低到70分甚至60分，也就是说，如果某客户同盾分虽然不到80分，但是如果超过60分，实际上也是很难下款的。

同盾贷前审核报告主要包括三大部分：

1. **扫描建议。**包含：同盾评分及风险建议
2. **基本信息。**包含：客户姓名、身份证号、手机号、归属地
3. **贷前风险情况。**是同盾报告的核心部分，包括5项核查内容：个人基本信息核查、风险信息扫描、多平台借贷申请检测、关联人信息扫描、客户行为检测

我们看下同盾贷前审核报告的例子：



## 四、贷前风险情况详细介绍