

【基于用户画像的智能推荐构件】

设计方案

|  |  |  |  |
| --- | --- | --- | --- |
| 作者 | 胡双双 | 审核人 |  |
| 修改日期 | 2019-06-22 | 审核日期 | YYYY-MM-DD |

中汇信息技术（上海）有限公司

变更履历

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **版本号** | **变更时间** | **变更范围** | **变更简要说明** | **变更**  **类别** | **变更人** |
| **V0.1** | **2019-06-08** | **全文** | **初始化创建文档** | **N** | **胡双双** |
| **V1.0** | **2019-06-22** | **全文** | **修改** | **M** | **胡双双** |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

注：N-New 新建；M-Modify 修改；A-Add 新增；D-Delete 删除；

目录

目录

[1 基本概述 5](#_Toc12371030)

[1.1 文档描述 5](#_Toc12371031)

[1.2 术语说明 5](#_Toc12371032)

[1.3 参考资料 5](#_Toc12371033)

[1.4 局限性 6](#_Toc12371034)

[2 需求描述 7](#_Toc12371035)

[2.1 需求概述 7](#_Toc12371036)

[2.2 功能需求 7](#_Toc12371037)

[2.3 非功能需求 7](#_Toc12371038)

[3 技术体系 8](#_Toc12371039)

[3.1 总体技术架构 8](#_Toc12371040)

[3.1.1 总体架构 8](#_Toc12371041)

[3.1.2 模块描述 8](#_Toc12371042)

[3.1.3 构件复用 16](#_Toc12371043)

[3.2 关键技术说明 16](#_Toc12371044)

[3.3 典型场景描述 16](#_Toc12371045)

[3.3.1 实时数据流计算 16](#_Toc12371046)

[3.3.2 定时跑批计算 17](#_Toc12371047)

[3.3.3 交易要素画像管理 17](#_Toc12371048)

[3.4 安全性设计 18](#_Toc12371049)

[领域 18](#_Toc12371050)

1. 基本概述

用户画像是一种将用户信息标签化，从而获得更深刻的用户行为洞察的技术。用户画像技术作为数据洞察的基石，已广泛深入的应用于各个行业，为上层应用，如智能推荐、营销策略制订、市场研判等提供辅助决策支撑。将用户画像技术运用在金融交易领域，通过收集和分析与金融交易要素，如交易机构、交易产品、交易员等相关的各类交易数据、行为数据，来构建交易要素画像。目前中心已建成了数据仓库系统，但尚未基于数据仓库系统中的数据构建中心自己的用户画像系统。另一方面，目前使用的智能推荐算法相对比较简单，更多是基于规则进行推荐，没有从交易要素画像中提取可用于辅助推荐的相关决策信息，如交易机构的相似度、交易机构对交易产品的偏好、交易机构近期的交易策略等。

* 1. 文档描述

本文档描述了架构资产的功能、设计及属性方法，阐述其关键组成部分的功能及具体实现，约定其的前置条件、需求和规划，为设计、开发、测试、发布提供参考依据和标准。

文档阅读人员：架构设计人员、开发人员、测试人员、运维人员。本文档为需求、设计、开发、测试、发布和维护人员提供基础的理论依据文档。

* 1. 术语说明

| 术语 | 解释 |
| --- | --- |
| 用户画像 | 用户画像是一种勾画目标用户、洞察用户的大数据技术。 |
| 流计算 | 相对于传统的跑批计算，流计算的数据源是实时，连续的数据流，计算延迟一般要求实时或准实时。 |

* 1. 参考资料

[列举相关的通用规范或项目文档]

| 文档名称 | 备注 |
| --- | --- |
|  |  |
|  |  |

* 1. 局限性
* 此构件中将包含若干用于说明交易要素画像生命周期管理过程的标签，但考虑到中心标签体系建设是一个渐进式的业务过程，而非一个阶段性的开发过程，所以本期项目中不会覆盖中心与金融交易要素画像相关的所有标签。
* 此构件中将集成包括Recommender，SKLearn，TensorFlow，Pytorch在内的各种主流机器学习算法框架，并提供用于提升智能推荐算法开发效率的模板，但考虑到智能推荐算法需要和具体的业务场景相结合，所以作为通用型构件，本期项目不会实现针对某个具体业务场景的智能推荐算法。

1. 需求描述
   1. 需求概述
   2. 功能需求

* 智能推荐构件支持自定义规则和智能推荐算法，
* 提供对数据源进行数据加工处理，根据标签定义及规则生成交易要素画像的功能；
* 根据设置推荐算法或规则，生成推荐数据；
* 提供交易要素画像创建、编辑、审批、部署的流程以及查询、存储、分析功能；
* 提供实时数据流处理及定时数据处理两种模式的数据处理方式；
  1. 非功能需求
* 系统中不存在单点故障问题，所有组件通过主备或多活方式实现高可用；
* 计算组件和数据存储组件可通过水平扩展方式提高整体吞吐能力；
* 提供多种机器学习算法框架，为开发人员提供更多样化的选择；
* 支持两种部署方式：基于容器化底座部署，和直接物理机部署；

1. 技术体系
   1. 总体技术架构
      1. 总体架构



图1 总体架构图

* + 1. 模块描述
       1. 容器化底座

此构件包含一套基于容器的底层基础设施，用于简化环境配置、开发、测试、生产以及部署的流程，同时为整个系统提供若干高级非功能特性，包括负载均衡、自动故障转移、水平扩展以及基于容器的插件式扩展机制等。

图片包含 文字, 地图

描述已自动生成

图2 容器化底座的基础架构图

容器化底座架构主要包含以下组件：

* Docker：容器化底座的核心实现技术，此构件中使用的Docker版本为18.06.1-ce。
* Kubernetes（K8S）：K8S是一个基于Docker的自动化容器操作平台，包括部署，调度和节点集群间扩展。此构件中使用的K8S版本为1.13.4。
* Rancher：Rancher在K8S基础上提供了基于Web页面的容器管理界面，此构件中使用的Rancher版本为2.2.0 RC13。
* 图片包含 屏幕截图

  描述已自动生成

图3 Rancher用于管理容器集群的Web界面

* + - 1. 分布式内存数据库

在本构件中，使用Ignite作为分布式内存数据库，使用版本为2.7.0。

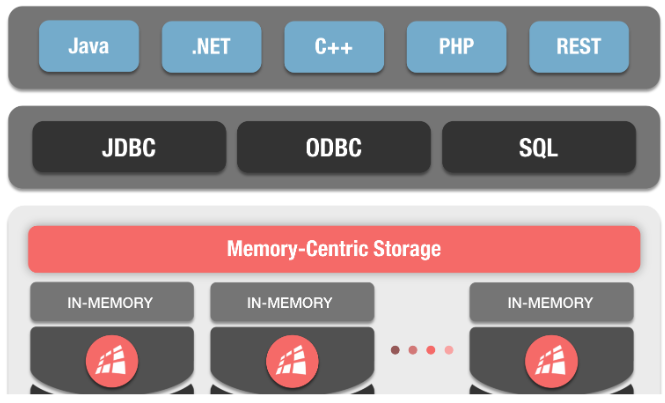


图4 Ignite整体架构图

Ignite作为分布式内存数据库，其主要特性包括：支持超大规模（PB级）数据集，内存及磁盘混合保存数据，同时支持键-值对及SQL，支持ACID事务，支持并置处理和机器学习等。

在此构件中，Ignite主要用于保存交易要素画像的相关数据。

* + - 1. 计算模块

在本构件中，使用Flink作为计算引擎，使用版本为1.7.2。

图片包含 屏幕截图

描述已自动生成

图5 Flink整体架构图

在此构件中，Flink用于支持如下两种计算模型：

* 流计算。场景包括：推荐交易要素，异常检测，跨系统数据处理等
* 跑批计算。主要场景为定期更新交易要素画像。
  + - 1. 数据端口

数据端口模块是一个从外部消息总线（DEP接口或IMTAPI接口）接收控制消息和业务数据流，业务逻辑处理完毕后再将回复消息回写到消息总线中。数据端口由Java开发，以独立进程的形式运行，数据端口模块使用ZooKeeper实现负载均衡，高可用等特性。

* + - 1. 事件流分发

此构件使用Kafka作为内部事件流的分发组件。数据端口接收到消息总线推送过来的数据流消息和指令消息后，会将消息置入Kafka的消息管道，后续处理模块监听到指定的消息管道中的消息，会将消息取处出来进行处理，处理完毕后，再将回复消息置入Kafka中。

此构件中使用的Kafka的版本为2.12，使用ZooKeeper实现负载均衡，高可用等特性。

* + - 1. 机器学习模块

此模块提供机器学习算法库和框架，用于实现交易要素画像的更新和智能推荐算法。提供的算法模型包括：

|  |  |  |  |
| --- | --- | --- | --- |
| 算法模型 | 协同过滤 | 基于内容 | 混合 |
| 非深度学习 | 1.Alternating Least Squares (ALS)  2.Singular Value Decomposition (SVD)  3.Smart Adaptive Recommendations (SAR)  4. FastAI Embedding Dot Bias (FAST) | 5.LightGBM/Gradient Boosting Tree | 6.Vowpal Wabbit (VW) |
| 深度学习 | 7. Neural Collaborative Filtering (NCF)  8. Restricted Boltzmann Machines (RBM) | 9.Deep Knowledge-Aware Network (DKN) | 10. Wide and Deep  11.Extreme Deep Factorization Machine (xDeepFM) |

所有算法模型统一封装到一个引擎中。针对不同的业务域的不同业务场景，可运行不同的引擎实例。每个业务场景对应的具体算法封装为运行在引擎实例中的一个进程，对外提供Restful API接口。

* + - 1. 交易要素画像分析&管理

##### 分析&管理功能

此模块用于对交易要素画像的生命周期进行管理，具体功能包括：

* 分析
  + 单个交易要素画像分析
  + 交易要素标签历史轨迹跟踪
  + 多个交易要素画像对比分析
  + 交易要素画像查询
* 管理
  + 标签定义
  + 标签查询
  + 标签编辑
  + 标签审批&发布

##### 画像类型

此构件中包含的交易要素画像类型为：

* 交易机构
* 交易员
* 交易产品

##### 标签类型

交易要素画像的标签类型分为如下四种：

* 统计型标签，标签产生来源于对历史数据的SQL计算获得。例如，机构类型标签、活跃度标签、交易相关指标等
* 事件型，由于特定的事件产生而触发标签产生。例如，A机构有今日提交5笔报价，成交2笔
* 分析型，基于特定业务算法或智能模型处理得到。例如，A机构对B产品的偏好高于全市场基准50个%，A机构更偏好短期交易行为
* 手工标注型，对无法从数据源中抽取的信息进行补全。例如，将A，B机构列入重点关注名单

##### 标签设计原则

此构件的交易要素画像的标签体系依据如下原则进行设计：

* 不同系统有不同的数据统计口径，如本币，外币等。所以标签也需要按照不同系统进行划分，例如对应本币系统，有一批标签；对于外币系统有一批标签
* 在某个系统里，根据系统特性做进一步细分。例如，针对本币系统，按照市场划分若干口径：现券，SL，或者买，卖等
* 标签需要按时段划分，以更准确的反映在长期，中期，和短期的变化趋势。比如年，半年，季度，月，周等
* 对于排名类标签，带上TOP数，以更全面的刻画排名，例如：A机构在本币系统现券市场本月买入对手方交易金额排名TOP5：B，C，D，E，F
* 除细分标签之外，需要上层汇总的标签（上层汇总标签更符合人的认知，有利于全局把握），如本币全市场活跃度TOP5

##### 标签体系

下表显示了基于本币交易系统现券市场构件的交易机构画像的部分标签。

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **标签名称** | **标签解释** | **标签主题** | **一级标签** | **二级标签** | **三级标签** |
| 名称 |  | 基本属性 |  |  |  |
| 编号 |  | 基本属性 |  |  |  |
| 注册省份 |  | 基本属性 |  |  |  |
| 注册城市 |  | 基本属性 |  |  |  |
| 注册交易员数量 |  | 基本属性 | 交易员数量 |  |  |
| 活跃交易员数量（年） |  | 基本属性 | 交易员数量 |  |  |
| 活跃交易员数量（季度） |  | 基本属性 | 交易员数量 |  |  |
| 活跃交易员数量（月） |  | 基本属性 | 交易员数量 |  |  |
| 活跃交易员数量（周） |  | 基本属性 | 交易员数量 |  |  |
| 做市商 | 该机构是否在某市场中是做市商 | 基本属性 |  |  |  |
| 评级A | 示例，该机构的评级信息，机构可以有多个不同的评级 | 基本属性 | 机构评级 |  |  |
| 评级B | 基本属性 | 机构评级 |  |  |
| 评级C | 基本属性 | 机构评级 |  |  |
| 本币\_现券\_买入\_金额\_年 |  | 本币市场 | 交易刻画 | 现券市场 | 交易金额 |
| 本币\_现券\_买入\_金额\_半年 |  | 本币市场 | 交易刻画 | 现券市场 | 交易金额 |
| 本币\_现券\_买入\_金额\_季 |  | 本币市场 | 交易刻画 | 现券市场 | 交易金额 |
| 本币\_现券\_买入\_金额\_月 |  | 本币市场 | 交易刻画 | 现券市场 | 交易金额 |
| 本币\_现券\_买入\_金额\_周 |  | 本币市场 | 交易刻画 | 现券市场 | 交易金额 |
| 本币\_现券\_卖出\_金额\_年 |  | 本币市场 | 交易刻画 | 现券市场 | 交易金额 |
| 本币\_现券\_卖出\_金额\_半年 |  | 本币市场 | 交易刻画 | 现券市场 | 交易金额 |
| 本币\_现券\_卖出\_金额\_季 |  | 本币市场 | 交易刻画 | 现券市场 | 交易金额 |
| 本币\_现券\_卖出\_金额\_月 |  | 本币市场 | 交易刻画 | 现券市场 | 交易金额 |
| 本币\_现券\_卖出\_金额\_周 |  | 本币市场 | 交易刻画 | 现券市场 | 交易金额 |
| 本币\_现券\_买入\_对手方\_年 | 本方买入，交易金额最大的对手方TOP5（对手方偏好） | 本币市场 | 交易刻画 | 现券市场 | 对手方 |
| 本币\_现券\_买入\_对手方\_半年 |  | 本币市场 | 交易刻画 | 现券市场 | 对手方 |
| 本币\_现券\_买入\_对手方\_季 |  | 本币市场 | 交易刻画 | 现券市场 | 对手方 |
| 本币\_现券\_买入\_对手方\_月 |  | 本币市场 | 交易刻画 | 现券市场 | 对手方 |
| 本币\_现券\_买入\_对手方\_周 |  | 本币市场 | 交易刻画 | 现券市场 | 对手方 |
| 本币\_现券\_卖出\_对手方\_年 |  | 本币市场 | 交易刻画 | 现券市场 | 对手方 |
| 本币\_现券\_卖出\_对手方\_半年 |  | 本币市场 | 交易刻画 | 现券市场 | 对手方 |
| 本币\_现券\_卖出\_对手方\_季 |  | 本币市场 | 交易刻画 | 现券市场 | 对手方 |
| 本币\_现券\_卖出\_对手方\_月 |  | 本币市场 | 交易刻画 | 现券市场 | 对手方 |
| 本币\_现券\_卖出\_对手方\_周 |  | 本币市场 | 交易刻画 | 现券市场 | 对手方 |
| 本币\_现券\_债券类型\_年 | 交易过的债券类型占比TOP5，包括类型和占比（债券类型偏好） | 本币市场 | 交易刻画 | 现券市场 | 债券 |
| 本币\_现券\_债券类型\_月 |  | 本币市场 | 交易刻画 | 现券市场 | 债券 |
| 本币\_现券\_债券类型\_季 |  | 本币市场 | 交易刻画 | 现券市场 | 债券 |
| 本币\_现券\_债券类型\_周 |  | 本币市场 | 交易刻画 | 现券市场 | 债券 |
| 本币\_现券\_债券\_年 | 交易过的债券占比TOP5，包括名称和占比（债券偏好） | 本币市场 | 交易刻画 | 现券市场 | 债券 |
| 本币\_现券\_债券\_月 |  | 本币市场 | 交易刻画 | 现券市场 | 债券 |
| 本币\_现券\_债券\_季 |  | 本币市场 | 交易刻画 | 现券市场 | 债券 |
| 本币\_现券\_债券\_周 |  | 本币市场 | 交易刻画 | 现券市场 | 债券 |
| 本币\_现券\_机构相似度 | 在现券市场上的机构相似度TOP5，包括名称和相似度值 | 本币市场 | 交易刻画 | 现券市场 | 对手方 |
| 本币\_现券\_机构关联度 | 在现券市场上与本方交易最频繁的机构TOP5，包括名称和关联度值 | 本币市场 | 交易刻画 | 现券市场 | 对手方 |
| 本币\_现券\_活跃度\_年 | 根据在本币市场上交易金额，按5档划分统计活跃度 | 本币市场 | 交易刻画 | 现券市场 | 活跃度 |
| 本币\_现券\_活跃度\_季 |  | 本币市场 | 交易刻画 | 现券市场 | 活跃度 |
| 本币\_现券\_活跃度\_月 |  | 本币市场 | 交易刻画 | 现券市场 | 活跃度 |
| 本币\_现券\_活跃度\_周 |  | 本币市场 | 交易刻画 | 现券市场 | 活跃度 |
| 对手方偏好 | 根据历史交易记录计算对某种交易要素的偏好，包括对手方，债券类型，债券 | 本币市场 | 交易刻画 | 现券市场 | 交易偏好 |
| 债券类型偏好 | 本币市场 | 交易刻画 | 现券市场 | 交易偏好 |
| 债券偏好 | 本币市场 | 交易刻画 | 现券市场 | 交易偏好 |
| 长期稳健 | 根据算法（尚未确定），尝试抽取机构的交易策略，并贴上对应的标签。只有当策略比较明显时才会贴上标签，否则不用贴 | 本币市场 | 交易刻画 | 现券市场 | 交易策略 |
| 短期逐利 | 本币市场 | 交易刻画 | 现券市场 | 交易策略 |
| 波段操作 | 本币市场 | 交易刻画 | 现券市场 | 交易策略 |

##### 标签存储模型

以下为用于保存交易要素画像对应标签的核心存储模型。

标签数据表（历史）

|  |  |  |  |
| --- | --- | --- | --- |
| 编号 | 名称 | 类型 | 备注 |
| 1 | EntityId | str | 需要附加标签的实体编号 |
| 2 | TagId | str | 标签编号 |
| 3 | TagValue | obj | Tag的值，如果有；如果么有则为null |
| 4 | FromTime | datetime | 标签有效起始时间 |
| 5 | ToTime | datetime | 标签有效结束时间，用于保留历史痕迹 |
| 6 | ExpiryTime | datetime | 标签失效时间 |

标签表（最新）

|  |  |  |  |
| --- | --- | --- | --- |
| 编号 | 名称 | 类型 | 备注 |
| 1 | Key | str | 画像对象编号 |
| 2 | Value | json | {updateTime:'2019-04-11',  tags:{ ‘tag1’:value1, 'tag2':'value2',...}} |

标签元数据表

|  |  |  |  |
| --- | --- | --- | --- |
| 编号 | 名称 | 类型 | 备注 |
| 1 | TagId | str | 标签编号 |
| 2 | Name | str | 标签中文名称 |
| 3 | TopicId | str | 标签对应的主题编号 |
| 4 | Type | int | 标签类型：统计/分类/算法 |
| 5 | DevMode | int | 标签开发模式：跑批/算法 |
| 6 | UpdateFrequency | int | 标签更新频率：每小时/天/周/月 |
| 7 | Description | str | 标签描述性内容 |

主题数据表

|  |  |  |  |
| --- | --- | --- | --- |
| 编号 | 名称 | 类型 | 备注 |
| 1 | TopicId | str | 主题编号 |
| 2 | Name | str | 主题中文名称 |
| 3 | ParentId | str | 主题的父主题编号 |
| 4 | Description | str | 主题描述性内容 |

* + - 1. 开发、运维管理框架/模板

提供一个CLI工具，内置各类指令可用于开发、运维管理。具体功能包含如下：

|  |  |
| --- | --- |
| 类别 | 实现功能 |
| 开发 | 提供指令根据项目模版创建数据处理任务及算法模版。 |
| 运维 | 按照模块，提供不同指令，可以查看当前模块的运行状态以及执行任务。 |

* + 1. 构件复用

无

* 1. 关键技术说明

| 技术名称 | 所属类别 | 技术选型说明 |
| --- | --- | --- |
|  |  |  |

* 1. 典型场景描述
     1. 实时数据流计算

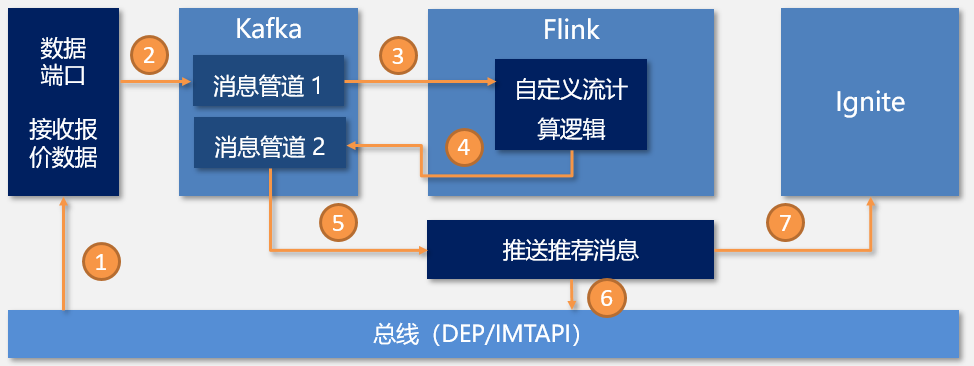


图6 实时数据流计算的流程图

流程说明：

1. 数据端口模块从总线接收到报价消息
2. 数据端口模块解析报价消息，根据预先配置的转发逻辑，将报价消息推送到对应的Kafka消息管道
3. Flink中的自定义流计算逻辑会根据报价消息，调用相应的推荐算法（基于智能或基于规则），计算出需要推荐的交易要素
4. 自定义流计算逻辑计算完毕后，将回复消息推送回Kafka的消息管道
5. 推送推荐消息模块从消息管道收到回复消息
6. 推送推荐消息模块将回复消息转换格式后，提交给总线路由到下游环节
7. 推送推荐消息模块将回复消息回写到数据库，供后续审计和分析使用
   * 1. 定时跑批计算

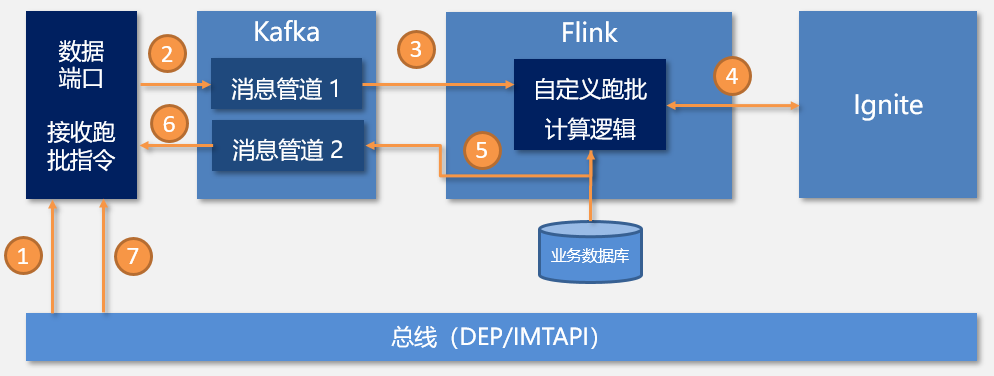


图7 定时跑批计算的流程图

流程说明：

1. 数据端口模块从总线接收到包含跑批指令的消息
2. 数据端口模块解析报价消息，根据预先配置的转发逻辑，将跑批指令消息推送到对应的Kafka消息管道
3. Flink中的自定义跑批计算逻辑收到消息后，会根据预先配置的跑批流程，调用相应的跑批处理逻辑，执行的跑批处理逻辑主要是基于当天的交易及用户行为数据，更新交易要素画像。在更新的过程中，跑批处理逻辑需要与业务数据库和用于保存当前画像数据的Ignite数据库进行交互
4. 自定义跑批计算逻辑计算完毕后，将更新的交易要素画像写入Ignite数据库
5. 自定义跑批计算逻辑推送一条消息至Kafka的消息管道，指示跑批计算完成
6. 数据端口模块从Kafka的消息管道接收到包含跑批计算完成的消息
7. 数据端口模块推送一条指示跑批计算完成的消息至总线
   * 1. 交易要素画像管理

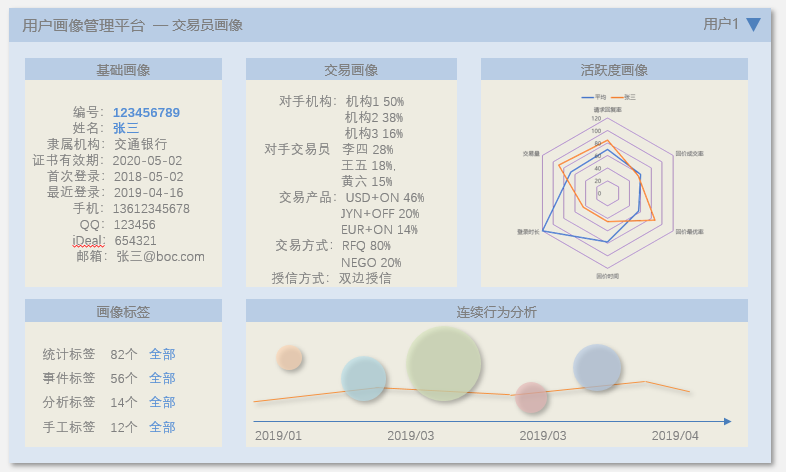


图8 交易要素画像管理

* 1. 安全性设计

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 领域 | 编号 | 要求级别 | 安全要求 | 适用性 | 备注 |
| 客户端安全 | C.0.1 | 二级 | 采用工具对客户端程序进行代码混淆 | 选择一项。 | [例：详见概要设计xxx节] |
| C.0.2 | 二级 | 客户端程序进行代码数字签名，程序启动时，由“客户端模块静态视图”章节所述的模块根据数字签名对程序完整性进行校验 | 选择一项。 |  |
| C.0.3 | 二级 | 客户端由“客户端模块静态视图”章节所述的模块使用控件处理用户密码输入 | 选择一项。 | [例：不适用-本系统无用户登录功能] |
| C.0.4 | 二级 | 客户端使用控件对敏感信息和重要信息输入进行防键盘窃听保护，涉及敏感信息和重要信息输入的模块包括“客户端模块静态视图”章节所述的模块 | 选择一项。 |  |
| C.0.5 | 二级 | 客户端使用控件对敏感信息输入进行加密保护，涉及敏感信息输入的模块包括“客户端模块静态视图”章节所述的模块 | 选择一项。 |  |
| C.0.6 | 二级 | 客户端涉及敏感信息处理的包括“客户端模块静态视图”章节所述的模块，敏感信息在使用后，不以任何形式存储于用户终端 | 选择一项。 |  |
| C.0.7 | 二级 | 客户端涉及重要信息处理的包括“客户端模块静态视图”章节所述的模块，是否有须在客户端持久化（如临时文件、本地数据库等）保存的重要数据： □否 □是，需保存的信息保护措施. | 选择一项。 |  |
| C.0.8 | 二级 | 客户端程序使用图形验证码组件对人机操作进行识别，在“客户端模块静态视图”章节所述的 模块中进行调用 | 选择一项。 |  |

网络通讯安全设计

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 领域 | 编号 | 要求级别 | 安全要求 | 适用性 | 备注 |
| 网络通信安全 | N.0.1 | 二级 | 采用第种方式,实现对与交易中心外部通信传输的敏感信息和重要信息的加密保护： 1.使用web/中间件服务器的SSL组件对所有传输数据加密 2.使用中心框架的SSL组件对所有传输数据加密 3.自行设计实现，由客户端“客户端模块静态视图”章节所述的模块对所有传输数据加密 4.由客户端“客户端模块静态视图”章节所述的模块对敏感信息和重要信息加密后传输 5.其他方式：。 | 选择一项。 |  |
| N.0.2 | 三级 | 采用第种方式,实现对与交易中心外部通信传输的所有数据的加密保护： 1.使用web/中间件服务器的SSL组件对所有传输数据加密 2.使用中心框架的SSL组件对所有传输数据加密 3.自行设计实现，由客户端“客户端模块静态视图”章节所述的模块对所有传输数据加密 4.其他方式：。 | 选择一项。 |  |
| N.0.3 | 二级 | 采用第种方式,实现对重放攻击的防范： 1.基于或参考SSL/TLS机制实现，使用动态会话密钥对所有通信数据进行加密 2.由客户端“客户端模块静态视图”章节所述的模块，在应用层数据包中加入（如序列号/时间戳/挑战应答机制） 3.业务逻辑限制:; 4.其他方式:。 | 选择一项。 |  |
| N.0.4 | 二级 | 采用第种方式,实现对敏感信息和重要信息的完整性校验： 1.基于或参考SSL/TLS机制实现，对所有通信数据进行完整性校验 2.由客户端“客户端模块静态视图”章节所述的模块，在应用层数据包中加入对敏感信息和重要信息的完整性校验 3.其他方式:。 | 选择一项。 |  |

服务器端安全设计

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 领域 | 编号 | 要求级别 | 安全要求 | 适用性 | 备注 |
| 系统架构安全 | S.1.3 | 二级 | 系统通过端口向外部用户提供服务; 系统通过端口向内部管理用户提供管理功能; 系统采用第种方式以限制仅能通过内部网络进行系统管理：  1.管理端口与对外服务端口分离，管理端口不对外开放  2.应用层面限制仅内部IP可访问管理端口  3.其他方式:. | 选择一项。 |  |
| S.1.4 | 二级 | 1.处理用户提交数据的服务器端模块应使用统一的过滤器对特殊字符进行处理，并在第6章相应模块的设计中说明对用户所提交数据字段的校验规则，如数据类型、长度和格式等，第6章中涉及相关模块的设计章节包括:   ;  2.处理用户操作的服务器端模块，应在第6章相应模块的设计中说明对用户操作进行身份和权限校验的逻辑，第6章中涉及相关模块的设计章节包括:  ; | 选择一项。 |  |
| 身份鉴别安全 | S.2.1 | 三级 | 三级系统采取双因素认证方式，由服务器端“服务器端模块静态视图”章节所述的模块对用户进行双因素认证 | 选择一项。 |  |
| S.2.2 | 二级 | 由服务器端“服务器端模块静态视图”章节所述的 模块对用户密码强度进行检查 密码长度要求位以上(应不低于8位) 密码复杂度度要求包含（应至少要求含数字和字母的组合） | 选择一项。 |  |
| S.2.3 | 二级 | 由服务器端“服务器端模块静态视图”章节所述的 模块对用户密码周期进行管理，并由模块在用户登录时进行校验 密码有效期为天（建议为90～180天），密码到期后处理机制为以下第种： 1. 强制用户修改 2. 提示用户修改 3. 其他：. | 选择一项。 |  |
| S.2.4 | 二级 | 系统是否有初始密码设置 □否 □是，系统初始密码由符合密码强度要求的随机字符组成，并由服务器端“服务器端模块静态视图”章节所述的 模块强制客户首次登录时修改初始密码 | 选择一项。 |  |
| S.2.5 | 二级 | 由服务器端“服务器端模块静态视图”章节所述的 模块对用户密码历史进行管理，禁止用户使用和前个密码相同的密码（应不少于4个） | 选择一项。 |  |
| 访问控制安全 | S.3.1 | 二级 | 1.系统是否有默认用户 □否 □是，默认用户仅有必要的最小权限，权限包括：.  2.系统用户管理和用户授权权限的设置方式是 □由不同用户承担，用户管理由用户承担，用户授权由用户承担 □权限可配置，且用户管理和用户授权不能授予同一用户 □其他：.  3.应在第6章相应权限管理模块的设计中描述本系统的用户权限模型，并结合用户权限清单、角色权限矩阵等方式进行说明。 | 选择一项。 |  |
| S.3.3 | 二级 | 由服务器端“服务器端模块静态视图”章节所述的 模块对用户登录进行控制 用户连续次登录失败(应不超过5次)后，对账户采取第 种锁定策略： 1. 固定锁定分钟（应不低于10分钟） 2. 动态锁定，首次触发锁定分钟，后续锁定时间按照 方式递增 3. 其他方式：. | 选择一项。 |  |
| S.3.4 | 二级 | 由服务器端“服务器端模块静态视图”章节所述的 模块对用户空闲超时进行控制 结合具体业务需求设置用户空闲超时机制，用户会话空闲超过分钟后（建议不超过30分钟），采取第 种方式进行控制  1. 注销用户登录  2. 锁定用户屏幕  3. 其他方式：. | 选择一项。 |  |
| S.3.5 | 二级 | 对于用户敏感信息的修改，采取第种方式，由服务器端“服务器端模块静态视图”章节所述的模块对用户身份进行二次认证 1.用户密码； 2.手机短信验证 3.其他：. | 选择一项。 |  |
| 数据交互安全 | S.4.2 | 二级 | 由服务器端“服务器端模块静态视图”章节所述的 模块对上传文件进行安全处理 1.系统允许上传的文件类型包括：; 2.如无必要，应尽量避免上传文件被用户直接访问，本系统上传文件是否须被用户访问： □否,对上传文件的安全处理措施:; □是,文件类型:; 对用户可访问的上传文件的安全处理措施:. | 选择一项。 |  |
| S.4.3 | 二级 | 由服务器端“服务器端模块静态视图”章节所述的 模块对文件下载进行控制，并将用户文件下载路径严格限制在服务器端的指定目录下 | 不适用 |  |
| 数据存储安全 | S.5.1 | 二级 | 服务器端涉及用户敏感信息存储的位置包括*[如数据库]*，敏感信息采用以下第种方式加密存储：  1.采用带salt的多轮哈希算法处理（推荐）  2.采用可逆算法加密处理，密钥妥善保存 |  |  |
| 资源控制安全 | S.6.1 | 二级 | 对单个用户的请求进行限制，包括： □限制用户并发会话数，允许的并发会话数为； □对消耗较多系统资源的用户操作进行频率控制（可自行添加）  操作1：； 频率：；  操作2：； 频率：； □其他：； | 不适用 |  |
| S.6.3 | 二级 | 系统中需对使用情况进行监控的重要资源及监控措施包括（可自行添加）： 资源1：； 监控措施1：； 资源2：； 监控措施2：； 资源3：； 监控措施3：； | 不适用 |  |
| 日志审计安全 | S.7.1 | 二级 | 系统由服务器端“服务器端模块静态视图”章节所述的 模块记录审计日志并提供统计分析功能，审计日志以（如文件/数据库）方式保存，采用第种方式对审计日志进行完整性保护 1.以SYSLOG消息方式，将日志发送至运行部日志服务器，在日志服务器上保存审计日志 2.由日志服务器通过部署agent或远程数据库访问方式获取日志，在日志服务器上保存审计日志 3.本系统自行对审计进行完整性保护，保护措施:  ; | 不适用 |  |
| 系统集成安全 | I.2.2 | 二级 | 应用系统使用的操作系统、数据库、中间件等基础系统软件的用户密码采用第种方式保存 1.由第三方基础软件加密保存 2.保存于配置文件中，采用算法进行加密 3.其他：. | 不适用 |  |
| 编码安全 | P.0.4 | 二级 | 本系统使用的密码算法包括 国产密码算法：. 国际公开密码算法：. | 不适用 |  |