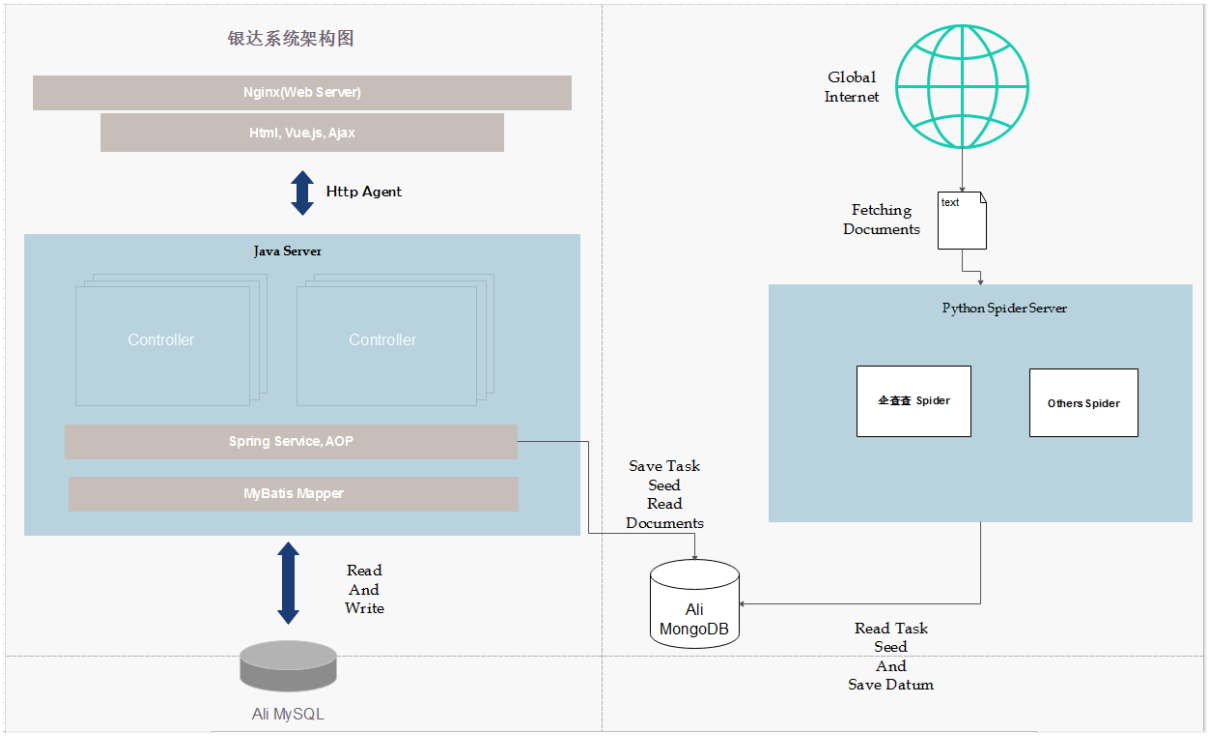
系统总体设计说明书

方楠

- 《保函业务智能查询监控平台》系统主要由以下三个部分组成，分别是：前端部分、后端部分和爬虫部分。
- 前端部分主要基于vue.js开发。根据需求分为了用户管理、大区管理、监控管理、预警管理、授权账号管理和审核列表模块。审核列表是系统的核心功能，通过审核列表中的新增信息功能添加业务单，在列表中查询采集进度和处理，在详情页中可点击公司名称进行数据立体化展现。
- 后端部分主要基于java语言开发，spring-boot框架编写。依赖包基于maven进行管理，并采用了Lucene底层进行模糊查询，接口共有约80个左右controller进行交互。对每一条任务都下发了唯一ID进行生命周期管理，并根据设计了用户权限功能来保护隐私和方便管理
- 爬虫部分主要基于python语言开发，scrapy框架和webdriver框架进行编写。基于不同的渠道采集难度使用了不同的采集技术。包括UA伪造、高匿IP池突破、验证码突破、浏览器模拟操作等。爬虫基于分布式部署，不同渠道爬虫遵循着相同的规约，消费共同的种子队列、在共同的地方管理自身状态、并且返回尽可能一致的数据结构。
- 系统架构图

- 任务全生命周期流程图



- 首先客户在前端下发一个任务，填写担保人、受益人等特征。
- 前端调用后端下发接口，赋予任务ID(missionID)，做正则校验正常后后推送任务至任务消费队列，里面带有全部特征。
- 任务消费队列基于MongoDB存储。后端下发默认所有爬虫的采集状态为0（未采集）。爬虫通过嗅探表的增量数据，发现存在采集任务。
- 每个爬虫会从种子中寻找自己需要的数据特征，进行采集任务。成功则会更新采集状态为1，失败且重试超过指定次数后则会给予-1，正在采集则会给予0.5.
- 爬虫的数据将会存在mongoDB里，爬虫采集的数据建立了missionID的外键，便于后端检索。
- 一旦状态信息为1，列表页中对应的渠道采集进度将会打勾，采集进度百分比会递增。
- 点击详情页，后端便会对mongoDB利用missionID进行查询。反馈到前端就会对相应数据展示。如果该渠道还没有采集完成，那对应的地方将会为空。

- 若遇到需要更新的数据，后端会将对应mongoDB的爬虫采集状态重置成0，这样就会重新采集。若遇到修改的单据，后端将会删除原missionID对应的任务，重新下发一个新missionID的任务。