热线产品化设计方案

2019年10月

目录

[一、 说明 3](#_Toc21461056)

[二、 建设背景 3](#_Toc21461057)

[三、 建设内容 3](#_Toc21461058)

[四、 详细建设内容 4](#_Toc21461059)

[4.1. 基础支撑部分 4](#_Toc21461060)

[4.1.1. 认证中心 4](#_Toc21461061)

[4.1.2. RBAC用户权限管理系统 5](#_Toc21461062)

[4.2. 基础业务部分 6](#_Toc21461063)

[4.2.1. 受理系统 6](#_Toc21461064)

[4.2.2. 微平台 7](#_Toc21461065)

[4.2.3. 转办系统 7](#_Toc21461066)

[4.2.4. 办理系统 7](#_Toc21461067)

[4.2.5. 回访系统 7](#_Toc21461068)

[4.2.6. 督办系统 7](#_Toc21461069)

[4.3. 辅助决策部分 7](#_Toc21461070)

[4.3.1. 统计系统 7](#_Toc21461071)

[4.3.2. 预警系统 7](#_Toc21461072)

[4.3.3. 考核系统 8](#_Toc21461073)

[4.4. 智能化部分 8](#_Toc21461074)

[4.4.1. 语音转文字 8](#_Toc21461075)

[4.4.2. 智能转办 8](#_Toc21461076)

[4.4.3. 智能分类 8](#_Toc21461077)

[4.4.4. 智能知识库（拟建设） 8](#_Toc21461078)

# 说明

此文档适用于济南热线国产化版本，用于开发阶段的指导性文件，文档根据子项目进行划分。文档将由多人修订完成，所有修订人员需要在修订记录中进行记录。

**适用人员：**热线国产化开发人员、客户方、管理人员。

**保密级别：**内部保密（仅上述适用人员使用，不得随意发送）。

# 建设背景

为适配政务行业国产化要求，建设自主可控的系统，提升现有系统效率，改善用户体验，开发新版的国产化热线系统。

现有热线系统受限于底层架构设计，在系统升级和扩展方面能力受限，业务功能扩展难度较高，复杂且长久的需求更新导致数据库性能愈发降低，在考核报表、统计等功能方面无法满足性能要求。

# 建设内容

为快速敏捷的开发可用产品，将热线分为五部分进行建设,包括：基础支撑部分、基础业务部分、领导决策辅助部分、智能化部分、热线生态周边等功能及模块。

# 设计原则

## 设计原则

本项目设计以现有系统为基础，以用户需求为导向，采用模块化、组件化的设计思想，采用“总体规划，分布实施”的策略，建设一套高扩展性，高复用性，安全可靠的热线业务系统，充分考虑和利用智能化组件,打造热线业务标准，构筑热线生态。

系统采用时下最新的信息技术，使用前后端分离的思想，既能满足当前系统功能需求，也能为将来的系统功能拓展，服用提供基础条件，保证系统在5-10年内不落后。

实用性：充分考虑目前业务流程和业务需求，对系统功能角色进行重构，从实用角度出发，明确角色划分，重构模块组合，达到学习成本低，使用效果强的实用性。

标准化：在系统设计过程中，对数据库、字段进行标准化，避免无意义的冗余，主要业务流程字段的标准化，在系统建设过程中，对接口标准，数据交互方式等采用业内标准接口，以实现全套系统的标准化。

安全和保密：系统设计时充分考虑安全性，系统通信采用HTTPS加密协议，在保密工单的处理中采用非对称加密的处理，保证数据在存储、流转过程中全部以加密状态流转。

高可用性：系统应用采用负载均衡，多机热备，保障软件在最大化的可用性，系统数据库采用集群化部署，并采用读写分离的机制，达到数据库的永不离线。模块轻耦合，隔离化，实现业务环节的分离。

轻量化：系统中文件存储使用FastDFS系统进行分布式部署，减少数据库存储和备份的体量，在文件传输过程中由文件传送改为地址传送，减少业务流转过程中的网络流量压力。

## 系统先进性

### 系统构建方式-模块化、服务化

随着系统复杂度的提升，以及对系统扩展性的要求越来越高，独立运行可以带来两个比较明显的好处：

第一个就是可扩展性。我们可以快速地添加服务集群的实例，提升整个模块集群的服务能力，如办理模块，可以独立部署，单独拓展集群，而不需整体升级，既不影响其他模块和服务的运行，也不会影响原有系统用户的体验。

第二个好处，即隔离性，隔离性实际上是可扩展性的基础，当我们将每个模块都隔离为独立的运行单元之后，任何一个或者多个模块的失败都将只影响自己或者少量其他模块，而不会大面积地波及整个服务运行体系，如交办模块与办理模块，模块间交互采用队列分离，当办理模块全部集群宕机时（在集群部署的模式下几乎不可能发生），转办仍然可以通过队列进行转办，当办理模块启动时由队列分发请求，转办请求将陆续进入办理系统。

在架构设计上有一种实践模式，即隔板模式（Bulkhead Pattern），这种架构设计模式的首要目的就是为了隔离系统中的各个功能单元和实体，使得系统不会因为一个单元或者服务的失败而导致整体失败。

### 消息通知

传统的系统通知、界面数量提醒等采用轮询或长轮询的机制进行，客户端以一定的时间间隔向服务端发出请求，以频繁请求的方式来保持客户端和服务器端的同步。这种同步方案的最大问题是，当客户端以固定频率向服务器发起请求的时候，服务器端的数据可能并没有更新，这样会带来很多无谓的网络传输，所以这是一种非常低效的实时方案。

轮询(Polling)是指不管服务器端有没有更新，客户端(通常是指浏览器)都定时的发送请求进行查询，轮询的结果可能是服务器端有新的更新过来，也可能什么也没有，只是返回个空的信息。不管结果如何，客户端处理完后到下一个定时时间点将继续下一轮的轮询。

长轮询是对定时轮询的改进和提高，目地是为了降低无效的网络传输。当服务器端没有数据更新的时候，连接会保持一段时间周期直到数据或状态改变或者时间过期，通过这种机制来减少无效的客户端和服务器间的交互。当然，如果服务端的数据变更非常频繁的话，这种机制和定时轮询比较起来没有本质上的性能的提高。

现有系统采用WebSocket推送机制，通过构建单独的消息服务器，向用户端定向推送，浏览器通过 JavaScript 向服务器发出建立 WebSocket 连接的请求，连接建立以后，客户端和服务器端就可以通过 TCP 连接直接交换数据，因为 WebSocket 连接本质上就是一个 TCP 连接，所以在数据传输的稳定性和数据传输量的大小方面，和轮询以及 Comet 技术比较，具有很大的性能优势。

典型的业务应用如现有的公告信息，转办退受理通知等消息，使用WebSocket替代轮询后可以减少大量的数据库查询，减少数据库及网络压力。

### 文件及静态资源的优化

使用分布式文件系统对系统中的附件、静态资源进行存储，避免因为附件导致的数据库过大，占用空间大、异地备份难的问题，解决附件在系统间传输的安全问题，大小限制问题，并且为系统一定的扩展能力和高可用性:。

在分布式文件系统中，高可用性包含两层，一是整个文件系统的可用性，二是数据的完整和一致性；

支持弹性存储: 可以根据业务需要灵活地增加或缩减数据存储以及增删存储池中的资源，而不需要中断系统运行。

### 统计、考核时效性提升

系统在设计初期充分考虑当前统计、考核需求以及未来的数据增量，通过在数据库采用集群部署、读写分离以及业务表状态的划分的形式尽量达到数据库资源的最优化使用，在统计考核系统设计中采用定时任务技术固化报表，引入大数据平台的形式做到最大限度的优化，避免统计、考核系统中因为响应时间导致的异常，提升使用体验。

### 解决办理单位数据管理成本高、统计困难

目前办理单位数据库分布在多个数据库中，造成办理单位数据无法统计或通过变通方式统计造成的数据不精准问题，通过数据库合并，对数据进行整合，达到统一的数据管理，完善的全流程查询，精准且高效的数据统计。

# 系统架构设计

## 技术架构说明

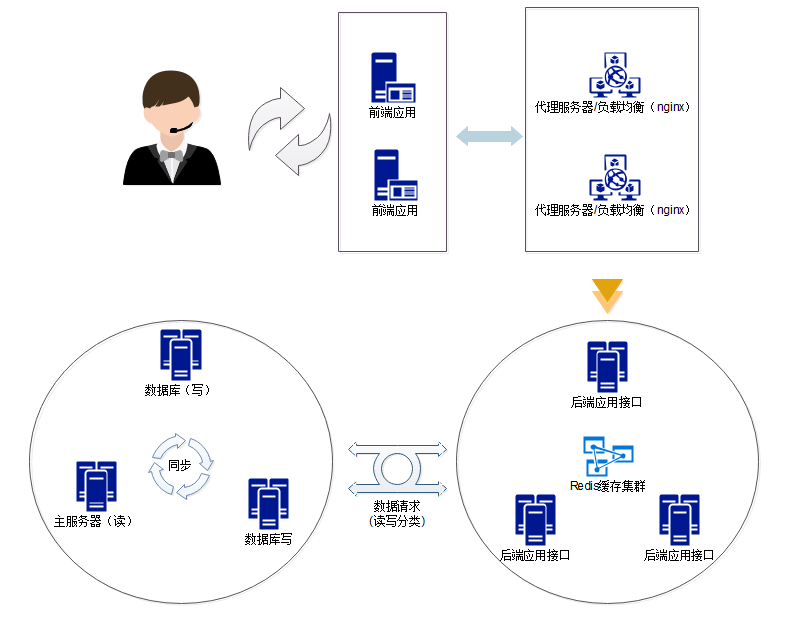
系统采用前后端分离的模式进行开发，前端使用Vue+Vuex+VueRouter框架，搭配antd-design-vue 组件库进行前端构建，支持HTML5，CSS3规范。

后端采用Java语言的Spring Boot框架，配合缓存框架 Redis，定时任务Quartz,消息队列ActiveMQ，进行构建，采用通用的标准化Restful风格接口与前端通信，通信过程中使用JSON Web Token（JWT）进行Token管理和身份认证。

数据访问层使用Mybatis框架，配合Druid数据连接池构建持久化层，在数据库方面采用瀚高集群化部署。

系统部署中间件选用Nginx进行应用负载均衡和跨域支持，使用Tomcat作为后端服务器。

应用部署图如下：



### VUE前端框架

Vue是一个构建用户界面的UI框架。Vue 的核心库只关注视图层，不仅易于上手，还便于与第三方库或既有项目整合。通过简单的api实现响应式的数据绑定和组合视图组件，Vue的核心是一个数据绑定系统。Vue的特性如下：

1. 轻量级的框架（压缩后只有几十kb）
2. 支持指令和自定指令
3. 双向数据绑定
4. 可复用性（组件）
5. 解耦数据与视图（数据驱动视图）
6. 虚拟Dom（通过Dom Diff算法提升性能）
7. 前端路由
8. 状态管理

### SPRING BOOT 后端框架

Spring框架是Java平台上的一种开源应用框架，提供具有控制反转特性的容器。尽管Spring框架自身对编程模型没有限制，但其在Java应用中的频繁使用让它备受青睐，以至于后来让它作为EJB（EnterpriseJavaBeans）模型的补充，甚至是替补。Spring框架为开发提供了一系列的解决方案，比如利用控制反转的核心特性，并通过依赖注入实现控制反转来实现管理对象生命周期容器化，利用面向切面编程进行声明式的事务管理，整合多种持久化技术管理数据访问，提供大量优秀的Web框架方便开发等等。Spring框架具有控制反转（IOC）特性，IOC旨在方便项目维护和测试，它提供了一种通过Java的反射机制对Java对象进行统一的配置和管理的方法。Spring框架利用容器管理对象的生命周期，容器可以通过扫描XML文件或类上特定Java注解来配置对象，开发者可以通过依赖查找或依赖注入来获得对象。Spring框架具有面向切面编程（AOP）框架，SpringAOP框架基于代理模式，同时运行时可配置；AOP框架主要针对模块之间的交叉关注点进行模块化。Spring框架的AOP框架仅提供基本的AOP特性，虽无法与AspectJ框架相比，但通过与AspectJ的集成，也可以满足基本需求。Spring框架下的事务管理、远程访问等功能均可以通过使用SpringAOP技术实现。Spring的事务管理框架为Java平台带来了一种抽象机制，使本地和全局事务以及嵌套事务能够与保存点一起工作，并且几乎可以在Java平台的任何环境中工作。Spring集成多种事务模板，系统可以通过事务模板、XML或Java注解进行事务配置，并且事务框架集成了消息传递和缓存等功能。Spring的数据访问框架解决了开发人员在应用程序中使用数据库时遇到的常见困难。它不仅对Java:JDBC、iBATS/MyBATIs、Hibernate、Java数据对象（JDO）、ApacheOJB和ApacheCayne等所有流行的数据访问框架中提供支持，同时还可以与Spring的事务管理一起使用，为数据访问提供了灵活的抽象。

### Redis 缓存框架

redis是一个key-value存储系统。和Memcached类似，它支持存储的value类型相对更多，包括string(字符串)、list(链表)、set(集合)、zset(sorted set --有序集合)和hash（哈希类型）。这些数据类型都支持push/pop、add/remove及取交集并集和差集及更丰富的操作，而且这些操作都是原子性的。在此基础上，redis支持各种不同方式的排序。与memcached一样，为了保证效率，数据都是缓存在内存中。区别的是redis会周期性的把更新的数据写入磁盘或者把修改操作写入追加的记录文件，并且在此基础上实现了master-slave(主从)同步。

Redis 是一个高性能的key-value数据库。 redis的出现，很大程度补偿了memcached这类key/value存储的不足，在部 分场合可以对关系数据库起到很好的补充作用。它提供了Java，C/C++，C#，PHP，JavaScript，Perl，Object-C，Python，Ruby，Erlang等客户端，使用很方便。 [1]

Redis支持主从同步。数据可以从主服务器向任意数量的从服务器上同步，从服务器可以是关联其他从服务器的主服务器。这使得Redis可执行单层树复制。存盘可以有意无意的对数据进行写操作。由于完全实现了发布/订阅机制，使得从数据库在任何地方同步树时，可订阅一个频道并接收主服务器完整的消息发布记录。同步对读取操作的可扩展性和数据冗余很有帮助。

在日常对数据库的访问中，读操作的次数远超写操作，比例大概在 1:9 到 3:7，所以需要读的可能性是比写的可能大得多的。当我们使用SQL语句去数据库进行读写操作时，数据库就会去磁盘把对应的数据索引取回来，这是一个相对较慢的过程。

如果把数据放在 Redis 中，也就是直接放在内存之中，让服务端直接去读取内存中的数据，那么这样速度明显就会快上不少，并且会极大减小数据库的压力，但是使用内存进行数据存储开销也是比较大的，限于成本的原因，一般只是使用 Redis 存储一些常用和主要的数据，比如用户登录的信息等。

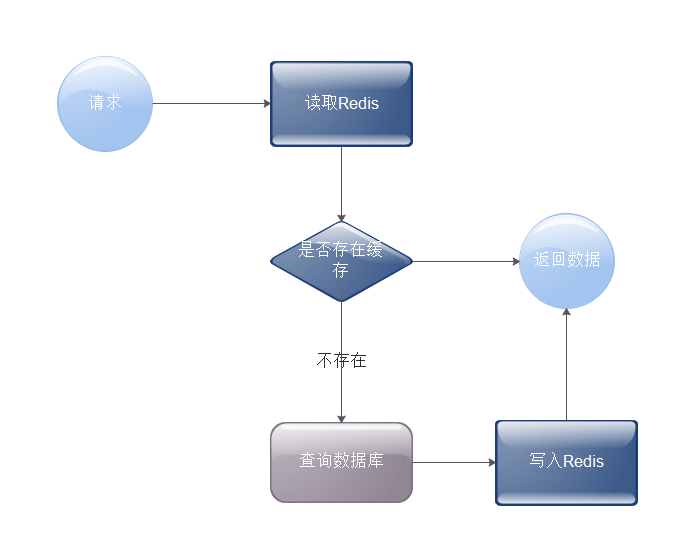
实际业务场中在使用 Redis 进行存储的时候，我们需要从以下几个方面来考虑：

业务数据常用吗？命中率如何？如果命中率很低，就没有必要写入缓存；

该业务数据是读操作多，还是写操作多？如果写操作多，频繁需要写入数据库，也没有必要使用缓存；

业务数据大小如何？如果要存储几百兆字节的文件，会给缓存带来很大的压力，这样也没有必要；

使用 Redis 作为缓存的读取逻辑如下图所示：

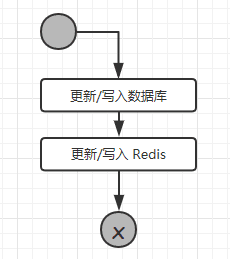


当第一次读取数据的时候，读取 Redis 的数据就会失败，此时就会触发程序读取数据库，把数据读取出来，并且写入 Redis 中；

当第二次以及以后需要读取数据时，就会直接读取 Redis，读到数据后就结束了流程，这样速度就大大提高了。

使用 Redis 来处理日常中需要经常读取的数据，速度提升是显而易见的，同时也降低了对数据库的依赖，使得数据库的压力大大减少。

**写操作的流程：**

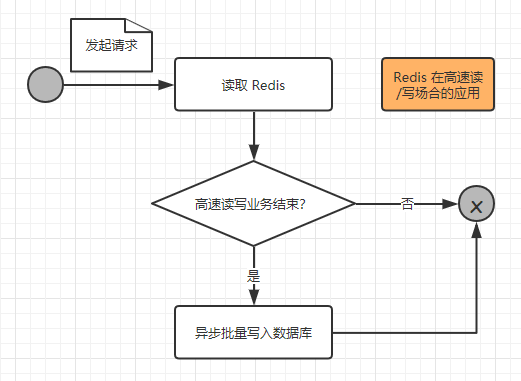


更新或者写入的操作，需要多个 Redis 的操作，如果业务数据写次数远大于读次数那么就没有必要使用 Redis。

**高速读/写的业务场合**

在热线业务中存在高并发的情况，比如下班前的回复，话务员的录入工单等，这些场合都是在某一个瞬间或者是某一个短暂的时刻产生大量请求到达服务器，如果单纯的使用数据库来进行处理，就算不崩，也会很慢的，轻则造成用户体验极差，重则数据库瘫痪，服务宕机，而这样的场合都是不允许的！

所以需要使用 Redis 来应对这样的高并发需求的场合，我们先来看看一次请求操作的流程图：



当一个请求到达服务器时，只是把业务数据在 Redis 上进行读写，而没有对数据库进行任何的操作，这样就能大大提高读写的速度，从而满足高速响应的需求；

但是这些缓存的数据仍然需要持久化，也就是存入数据库之中，所以在一个请求操作完 Redis 的读/写之后，会去判断该高速读/写的业务是否结束，这个判断通常会在大量录入，大量回复时成立，如果不成立，则不会操作数据库；如果成立，则触发事件将 Redis 的缓存的数据以批量的形式一次性写入数据库，从而完成持久化的工作。

### Qurtz 定时任务

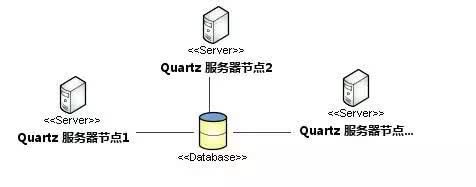
Quartz是Java领域最著名的开源任务调度工具，是一个任务调度框架，通过触发器设置作业的定时运行规则，来执行定时任务。其中quartz集群通过故障切换和负载平衡的功能，能给调度器带来高可用性和伸缩性。

Quartz提供了极为广泛的特性如持久化任务，集群和分布式任务等,其特点如下：

完全由Java写成，方便集成(Spring)、伸缩性、负载均衡、高可用性；

* 典型的使用场景，主要用来执行定时任务，例如：
* 定时发送信息
* 定时生成报表
* 自动更新静态数据

**1.Quartz集群部署：**



Quartz集群中的每个节点是一个独立的Quartz应用，它又管理着其他的节点。该集群需要分别对每个节点分别启动或停止，不像应用服务器的集群，独立的Quartz节点并不与另一个节点或是管理节点通信。Quartz应用是通过数据库表来感知到另一应用。只有使用持久的JobStore才能完成Quqrtz集群。

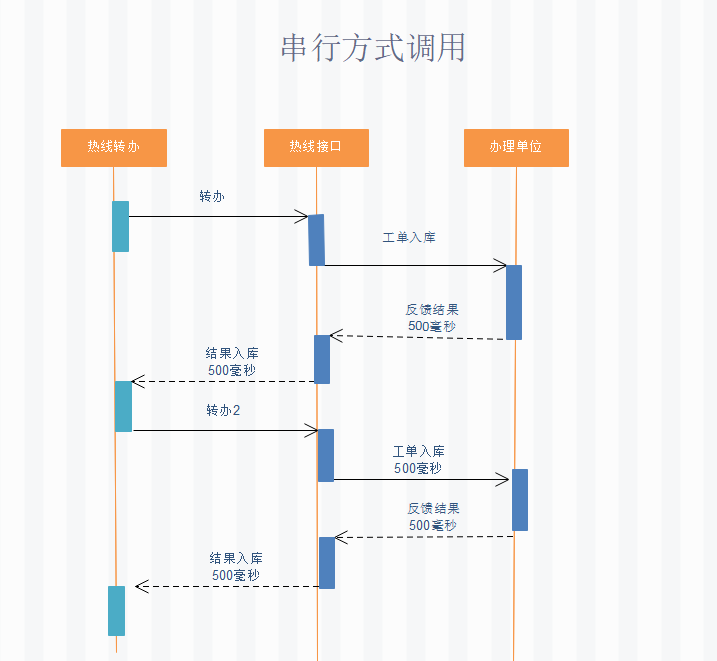
在热线业务中存在大量的数据抓取，定时服务，如：考核数据抓取，报表静态化等，使用Quartz来管理所有的定时任务，保障业务数据的准时、准确的抓取。

### MQ消息队列-ActiveMQ实现

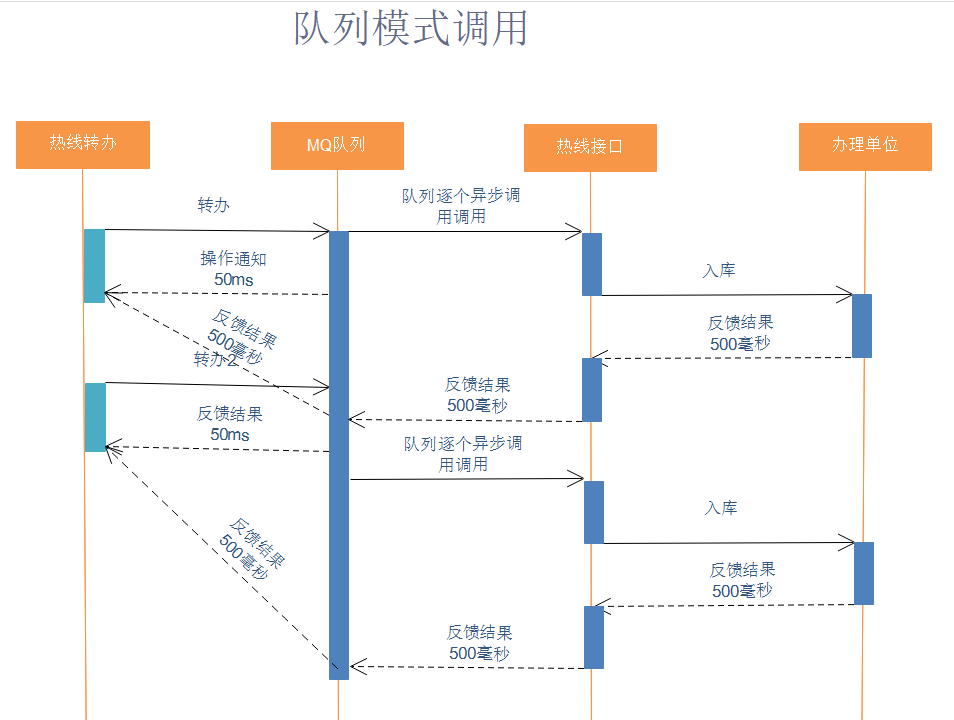
MQ全称为Message Queue,消息队列(MQ)是正确而又完整的 JMS 实现,消息队列（MQ）是一种应用程序对应用程序的通信方法。应用程序通过写和检索出入列队的针对应用程序的数据（消息）来通信，而无需专用连接来链接它们。消息传递指的是程序之间通过在消息中发送数据进行通信，而不是通过直接调用彼此来通信，直接调用通常是用于诸如远程过程调用的技术。

场景说明:转办人员通过转办功能，向办理单位转办。

串行方式：点击转办按钮调用转办接口向办理单位写入下发转办数据，调用办理单位接口入库需耗时500毫秒，转办结果反馈入库需要耗时500毫秒



使用消息队列：点击转办按钮调用消息队列接口向办理单位写入下发转办数据，消息队列反馈通知消息耗时500毫秒，此时转办人员直接开始下一张工单转办，由队列异步调用办理单位接口入库需耗时500毫秒，转办结果由队列反馈入库需要耗时500毫秒。用户只需等待转办提交到队列的500毫秒即可开始后续的工作，当业务量大时该方式将体现出巨大的优势。



2. 应用解耦

场景说明:转办系统与办理系统，办理系统与回访系统直接调用接口进行交互，系统质检相互依存性较高。

1.与其他系统过度耦合

2.其中某一个系统出现问题，影响主业务

如果使用消息队列

优点:

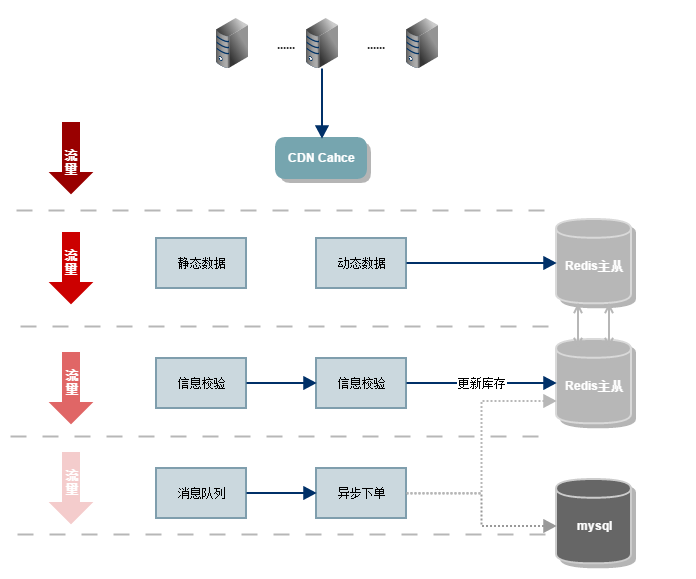
1.交互到达消息队列则返回成功。

2.某系统出现问题不会导致其他系统业务无法流转，实现解耦。

3. 大数据量并发方案

场景说明:即将下班时所有单位共同发送未办结回复或回退。

在此时段将在较短时间内产生比平时大数十倍页面访问流量和请求流量，会对数据库产生大量的行锁，导致系统卡顿。



利用读写分离 Redis 缓存拦截流量(开始时拦截大部分动态数据请求)

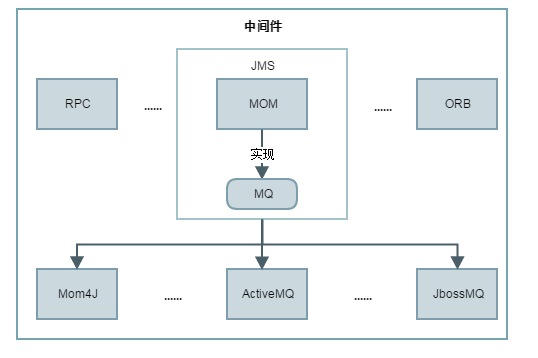
成功回复后，进入下层服务，开始进行信息校验。为了避免直接访问数据库，我们使用主从版 Redis 来进行状态记录。

如果还有大量并发的请求则利用消息队列组件，将回复信息写入消息队列后，即可认为回复完成，避免直接操作数据库。

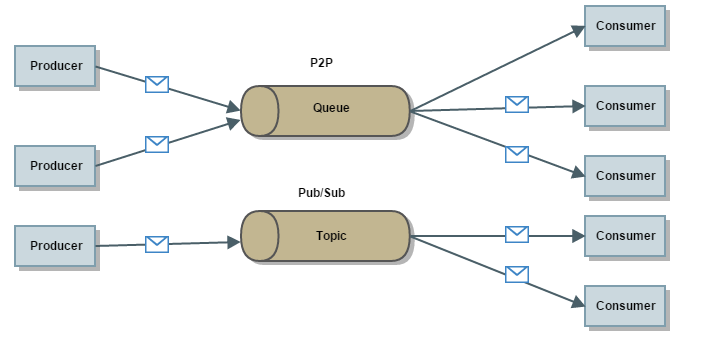
五.JMS实现--ActiveMQ

ActiveMQ是Apache软件基金下的一个开源软件，它遵循JMS1.1规范（Java Message Service），是消息驱动中间件软件（MOM）。它为企业消息传递提供高可用，出色性能，可扩展，稳定和安全保障。

5.1 中间件、JMS、MQ、ActiveMQ之间的关系



5.2 ActiveMQ的消息传递模式



P2P （点对点）消息域使用 queue 作为 Destination，消息可以被同步或异步的发送和接收，每个消息只会给一个 Consumer 传送一次。

Pub/Sub（发布/订阅，Publish/Subscribe）消息域使用 topic 作为 Destination，发布者向 topic 发送消息，订阅者注册接收来自 topic 的消息。发送到 topic 的任何消息都将自动传递给所有订阅者。接收方式（同步和异步）与 P2P 域相同

### Web socket 通信技术

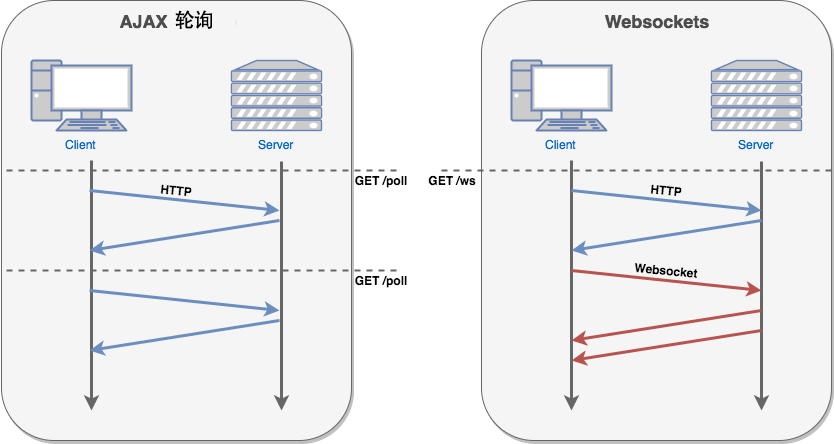
WebSocket 是 HTML5 开始提供的一种在单个 TCP 连接上进行全双工通讯的协议。

WebSocket 使得客户端和服务器之间的数据交换变得更加简单，允许服务端主动向客户端推送数据。在 WebSocket API 中，浏览器和服务器只需要完成一次握手，两者之间就直接可以创建持久性的连接，并进行双向数据传输。

在 WebSocket API 中，浏览器和服务器只需要做一个握手的动作，然后，浏览器和服务器之间就形成了一条快速通道。两者之间就直接可以数据互相传送。

现在，很多网站为了实现推送技术，所用的技术都是 Ajax 轮询。轮询是在特定的的时间间隔（如每1秒），由浏览器对服务器发出HTTP请求，然后由服务器返回最新的数据给客户端的浏览器。这种传统的模式带来很明显的缺点，即浏览器需要不断的向服务器发出请求，然而HTTP请求可能包含较长的头部，其中真正有效的数据可能只是很小的一部分，显然这样会浪费很多的带宽等资源。

HTML5 定义的 WebSocket 协议，能更好的节省服务器资源和带宽，并且能够更实时地进行通讯。

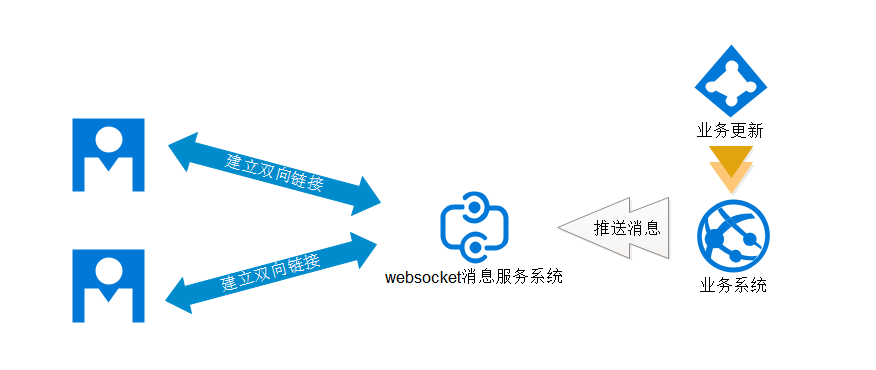


浏览器通过 JavaScript 向服务器发出建立 WebSocket 连接的请求，连接建立以后，客户端和服务器端就可以通过 TCP 连接直接交换数据。

**在业务中的应用**

在热线业务中，常常存在大量的轮询操作，如列表数量统计，并以角标显示，对于话务员来说转办退受理、转办反受理的工单提醒，对于转办人员来说，部门回退的工单，公告的通知推送等信息，传统业务中使用大量的定时轮询来访问后台拿取数据，不仅造成数据库压力过大，同时刷新周期不易控制，间隔短则数据库压力过大，间隔太长则数据时效性较差。

采用websocket推送机制可以有效的避免上述的问题，当业务系统启动时创建websocket通道，待有后台业务更新时查找用户的标志，并给指定用户推送消息和提醒。



### NGINX 代理服务

Nginx以事件驱动的方式编写，所以有非常好的性能，同时也是一个非常高效的反向代理、负载平衡。其拥有匹配Lighttpd的性能，同时还没有Lighttpd的内存泄漏问题，而且Lighttpd的mod\_proxy也有一些问题并且很久没有更新。

现在，Igor将源代码以类BSD许可证的形式发布。Nginx因为它的稳定性、丰富的模块库、灵活的配置和低系统资源的消耗而闻名．业界一致认为它是Apache2.2＋mod\_proxy\_balancer的轻量级代替者，不仅是因为响应静态页面的速度非常快，而且它的模块数量达到Apache的近2/3。对proxy和rewrite模块的支持很彻底，还支持mod\_fcgi、ssl、vhosts ，适合用来做mongrel clusters的前端HTTP响应。

nginx做为HTTP服务器，有以下几项基本特性：

处理静态文件，索引文件以及自动索引；打开文件描述符缓冲．

无缓存的反向代理加速，简单的负载均衡和容错．

FastCGI，简单的负载均衡和容错．

模块化的结构。包括gzipping, byte ranges, chunked responses,以及 SSI-filter等filter。如果由FastCGI或其它代理服务器处理单页中存在的多个SSI，则这项处理可以并行运行，而不需要相互等待。

支持SSL 和 TLSSNI．

Nginx专为性能优化而开发，性能是其最重要的考量,实现上非常注重效率 。它支持内核Poll模型，能经受高负载的考验,有报告表明能支持高达 50,000个并发连接数。

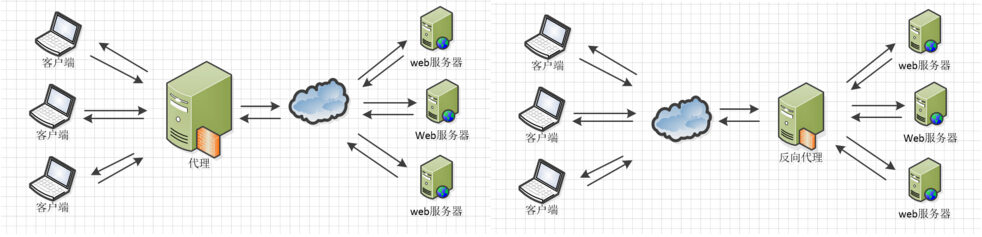
Nginx具有很高的稳定性。其它HTTP服务器，当遇到访问的峰值，或者有人恶意发起慢速连接时，也很可能会导致服务器物理内存耗尽频繁交换，失去响应，只能重启服务器。例如当前apache一旦上到200个以上进程，web响应速度就明显非常缓慢了。而Nginx采取了分阶段资源分配技术，使得它的CPU与内存占用率非常低。nginx官方表示保持10,000个没有活动的连接，它只占2.5M内存，所以类似DOS这样的攻击对nginx来说基本上是毫无用处的。就稳定性而言,nginx比lighthttpd更胜一筹。

Nginx支持热部署。它的启动特别容易, 并且几乎可以做到7\*24不间断运行，即使运行数个月也不需要重新启动。你还能够在不间断服务的情况下，对软件版本进行进行升级。

Nginx采用master-slave模型,能够充分利用SMP的优势，且能够减少工作进程在磁盘I/O的阻塞延迟。当采用select()/poll()调用时，还可以限制每个进程的连接数。

**Nginx的应用**

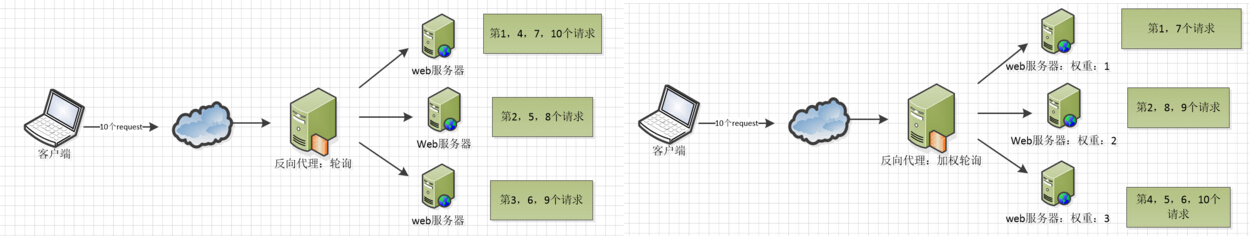
1、Http代理，反向代理：作为web服务器最常用的功能之一，尤其是反向代理。



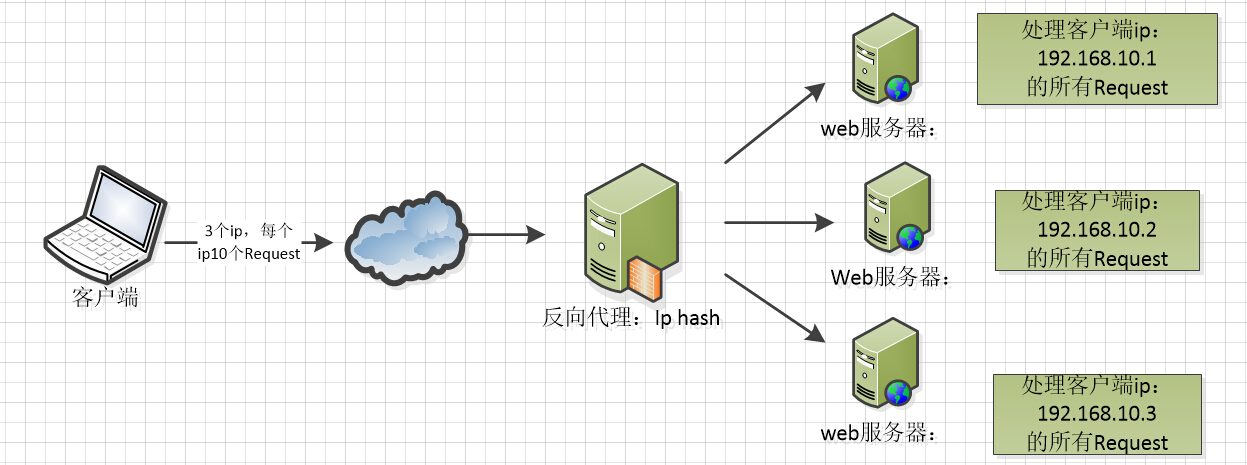
Nginx在做反向代理时，提供性能稳定，并且能够提供配置灵活的转发功能。Nginx可以根据不同的正则匹配，采取不同的转发策略，比如图片文件结尾的走文件服务器，动态页面走web服务器，只要你正则写的没问题，又有相对应的服务器解决方案，你就可以随心所欲的玩。并且Nginx对返回结果进行错误页跳转，异常判断等。如果被分发的服务器存在异常，他可以将请求重新转发给另外一台服务器，然后自动去除异常服务器。

2、负载均衡

Nginx提供的负载均衡策略有2种：内置策略和扩展策略。内置策略为轮询，加权轮询，Ip hash。



Ip hash算法，对客户端请求的ip进行hash操作，然后根据hash结果将同一个客户端ip的请求分发给同一台服务器进行处理，可以解决session不共享的问题。



3、web缓存

Nginx可以对不同的文件做不同的缓存处理，配置灵活，并且支持FastCGI\_Cache，主要用于对FastCGI的动态程序进行缓存。配合着第三方的ngx\_cache\_purge，对制定的URL缓存内容可以的进行增删管理。

### Apache TOMCAT 服务器中间件

Tomcat 服务器是一个免费的开放源代码的Web 应用服务器，它是Apache 软件基金会（Apache Software Foundation）的Jakarta 项目中的一个核心项目， Tomcat 技术先进、性能稳定，是目前比较流行的Web 应用服务器。

Tomcat服务器是一个免费的开放源代码的Web 应用服务器，属于轻量级应用服务器，在中小型系统和并发访问用户不是很多的场合下被普遍使用，是开发和调试JSP 程序的首选。

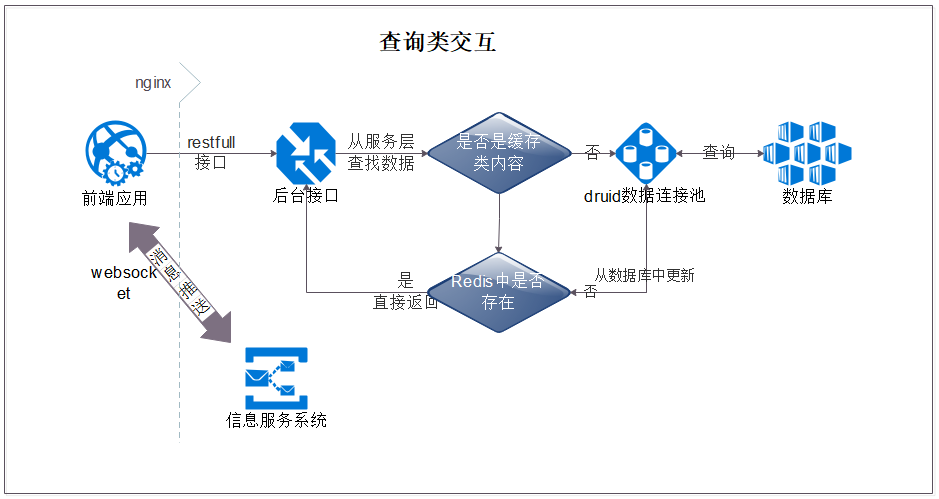
在业务应用中，系统后台使用tomcat作为系统运行环境，为java提供虚拟机。

## 前后端数据交互设计

前后端交互需要基于Nginx的反向代理功能，主要为解决跨域访问问题，前后端交互分为两种：WebSocket长连接和Restful接口调用。

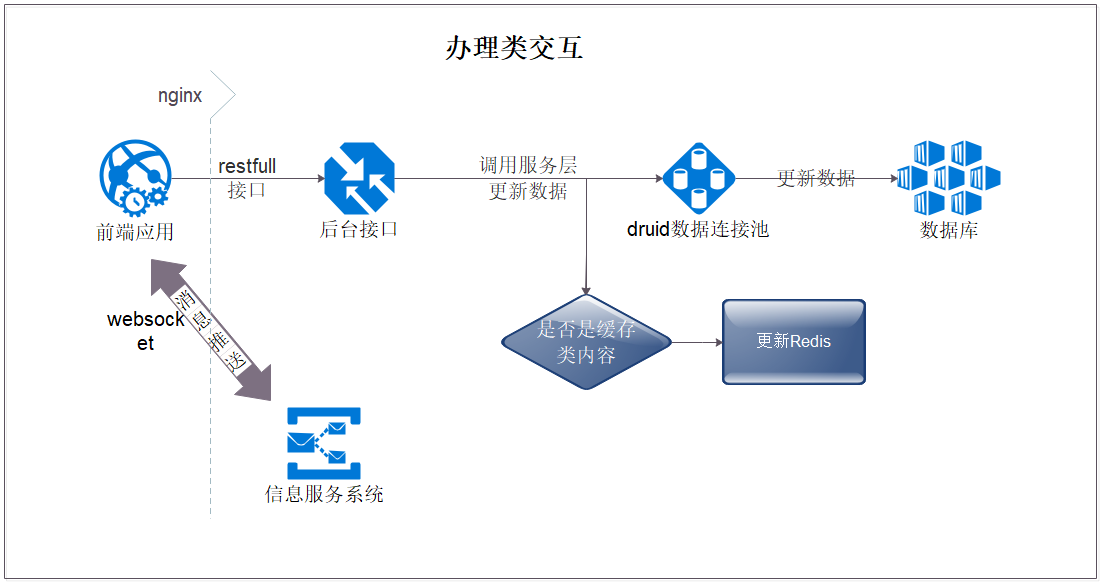
其中消息类、须推送类内容由前端与消息服务创建WebSocket进行双向通信。业务查询、办理类操作通过Restful接口进行交互。

查询类的前后端交互图如下：

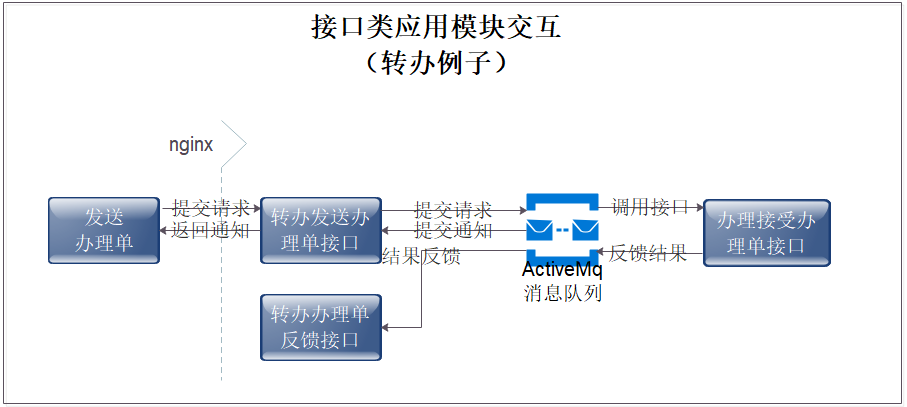


前端通过Nginx反向代理访问后台接口，后台接口调用服务层，如果是缓存类数据，则进入Redis中查找是否存在，如果存在直接从缓存中返回，如果缓存中不存在（第一次加载），则通过Druid数据库连接池，从数据库中查找数据并更新Redis，同时返回到前端。

如果不是缓存类数据，则服务层直接调用Druid数据连接池从数据库中查询数据返回给前端。



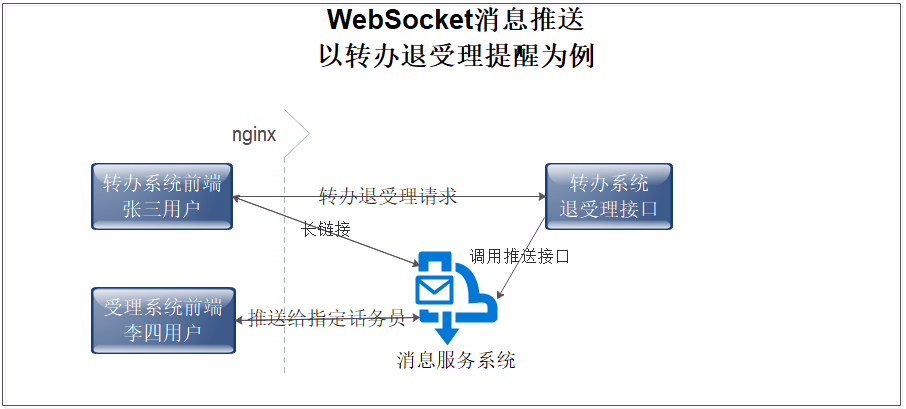
## 模块间（子系统）数据交互设计



模块间数据交互采用ActiveMq作为中间件进行队列消息管理，以转办为例，当转办人员发送办理单时，前端发起请求调用转办系统的办理单发送接口，转办系统发送接口调用ActiveMq队列，由队列接管请求，并将通知消息返回给办理单发送接口，由发送接口通知前端已经发送结束，消息队列异步的将请求发送给办理系统。

此设计方式的优点在于高可用性，即便办理端离线，也不影响转办人员继续发送办理单，当办理单恢复正常后仍旧可以接收到办理单，不会因为系统中某一个环节宕机导致整条业务线的无法流转，并且在业务高峰期时结合Redis缓存可以实现顺畅的用户体验和高效的系统性能。

## 消息推送机制设计



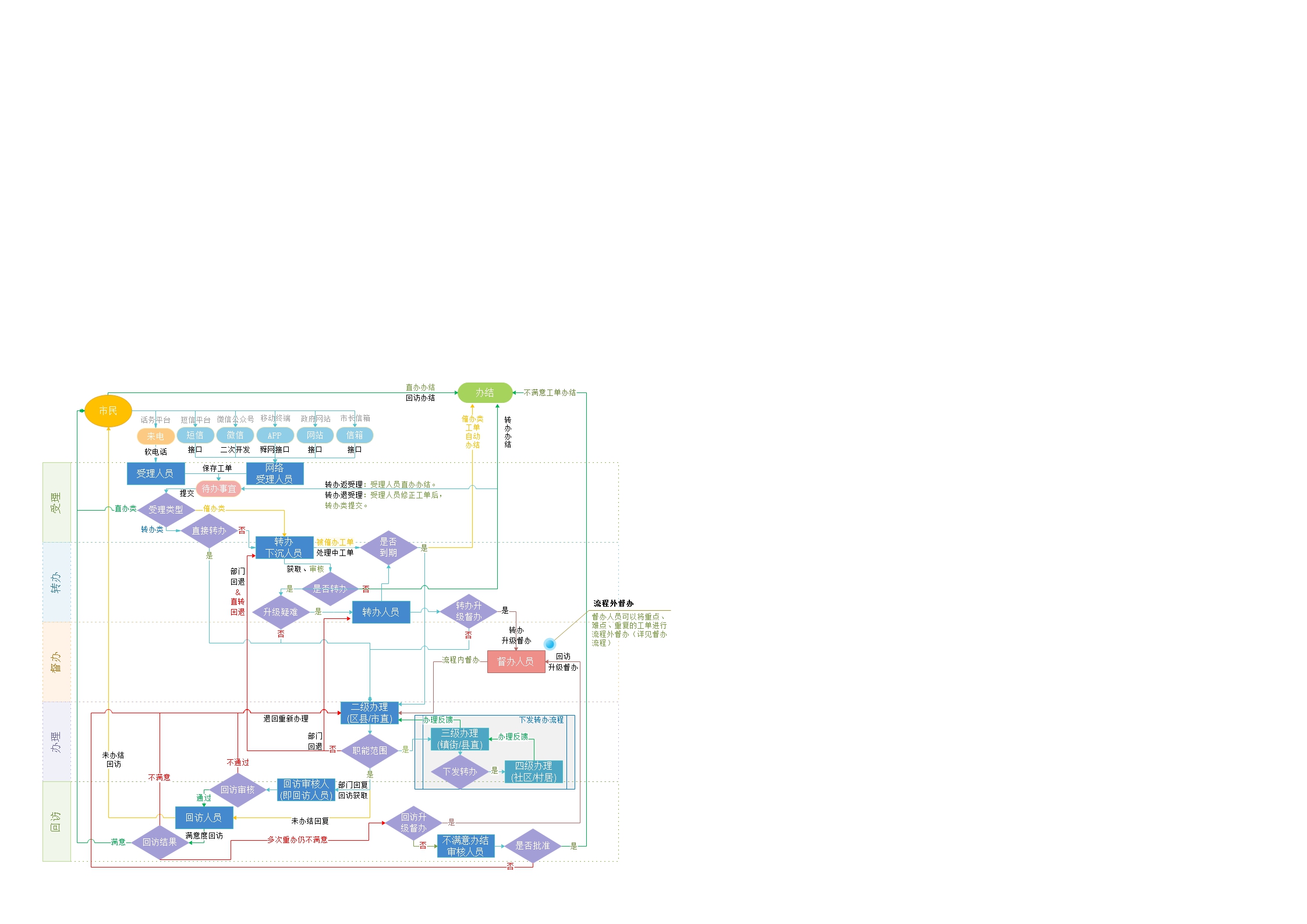
消息推送机制贯穿于各个模块间，通过WebSocket长链接的形式时限消息、通知、公告的实时推送.

以转办退受理提醒为例，当转办人员张三使用转办系统发起退受理请求，转办系统后台接口在变更工单状态，插入流转信息后调用消息服务系统，将通知信息推送给工单的受理人员李四。

此设计方案的优势在于可以用于系统中需要实时展现的信息，如公告，列表数量等，替代传统的Ajax轮询操作，极大的减轻数据库的压力。

# 业务流程分析

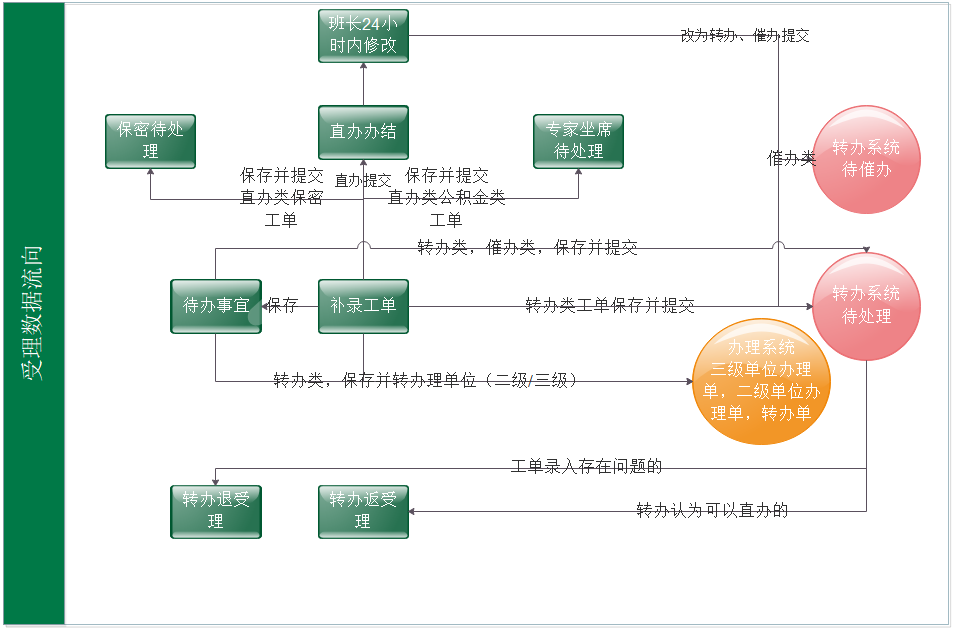
## 总体业务流程图



## 系统用户划分

|  |  |  |
| --- | --- | --- |
| 功能分区 | 角色名称 | 角色说明 |
| 受理中心 | 受理人员 | 受理系统访问权限，使用受理系统接听来电，录入工单。 |
| 转办下沉人员 | 隶属受理中心，使用转办系统权限，转办本班组内的受理人员录入的工单，此角色工单按班组获取，包含受理人员所有权限 |
|  | 网络坐席受理人员 | 微平台访问权限，使用微平台受理互联网、APP、微信、短信、信箱及其他拓展渠道反映的诉求。 |
|  | 受理班长 | 受理系统权限，软电话强插、软电话监听，修改24小时内话务员直办办结的工单（修改转办类未获取工单，修改为直办办结），包含受理人员所有权限，包含网络坐席受理人员角色。 |
| 转办中心 | 市级转办（省级转办） | 转办系统转办权限，转办受理和微平台提交的转办类、催办类工单，此角色转办人员按查询条件获取工单，包含受理人员所有权限，可以处理疑难转办升级疑难工单，升级督办。 |
| 转办班长 | 转办系统，转办升级督办权限，包含转办所有人员所有权限，可以查看并处理所有转办人员（转办下沉人员）工作界面工单 |
| 回访中心 | 初审人员 | 回访系统回访初审权限 |
| 回访人员 | 回访系统回访功能权限 |
| 回访审核人员 | 审核不满意办结工单和不再办理工单，工单办结权限 |
| 回访班长 | 回访升级疑难工单处理权限，包含初审、回访人员、回访审核人员所有权限。 |
| 保密回访专席 | 保密工单回访权限（优先处理保密工单，也可以处理非保密工单） |
| 督办中心 | 督办人员 | 督办系统使用权限 |
| 督办班长 | 督办人员所有权限，访问所有督办人员督办记录 |
| 质检中心 | 质检人员 | 质检系统使用权限，个人质检记录查看，个人工作量统计 |
| 质检班长 | 包含质检人员所有权限，访问所有质检人员的质检记录和质检结果统计、分析 |
| 知识库 | 知识库班长 | 知识库建议、知识库更新审核权限 |
| 采编人员 | 知识库添加、维护权限 |
| 办理单位 | 转办人员 | 办理系统下发转办、办理单等权限 |
| 回访人员 | 办理系统回访权限 |
| 统计人员 | 办理单位统计报表访问权限 |

## 受理系统业务流程说明



### 外呼处理

系统基础业务流程由受理或微平台开始，以受理系统为例，市民来电时，话务员接听电话，并录入工单，录入工单时分为直办、转办、催办三类受理类型。

当受理类型为直办时，系统允许话务员提交和保存，话务员提交直办类工单则直接办结，点击“保存”时则工单进入待办事宜，可以在电话结束后进行编辑完善。

当受理类型为转办时，可以使用“保存”、“保存并提交”或“保存并转办理单位”，“保存”功能与直办中保存一致，如果使用“保存并提交”则工单进入转办系统，由转办下沉人员进行处理，如果使用“保存并转办理单位”则可以将工单直接转办到办理单位。

当受理类型为催办时,可以使用“保存”或“保存并转办理单位”，保存功能与直办中保存一致，如果使用“保存并提交”功能，则工单进入转系统中的待催办工单，由转办人员办理。

待办事宜中已保存的工单可以进行编辑，在编辑时也可以修改工单的受理类型，根据受理类型不同可以执行不同的操作，操作的控制与录入工单时一致。

在受理系统中，受理班长可以对24小时内话务提交的工单进行修改，如将工单受理类型的直办改为转办，此时工单状态将变为待获取，由转办人员获取。待获取的工单也可以将转办变为直办，此时工单由待获取状态变为直办办结。

### 转办退受理

受理人员提交的转办工单，如果转办人员认为存在问题的工单，转办人员可以使用退受理功能，退回到受理人员，退回的工单可以进行修改后继续提交。

### 转办返受理

受理人员提交的转办工单，如果转办人员认为可以直办的工单，转办人员可以使用返受理功能，返回给受理人员，此时受理人员可以进行直办办结。

### 保密工单

对于保密类工单，当市民要求保密时，话务员使用受理类型需要选择“直办”，并提交，此电话将转给保密坐席，如果保密坐席全忙则进入队列等待。

保密坐席接听市民电话时系统读取该联系人历史工单，并打开最近的一张保密工单，由保密坐席继续为市民服务。

当市民在电话转接过程中挂断时，保密坐席可以在保密待处理中获取未成功受理的保密工单，通过外拨电话主动联系市民继续服务。

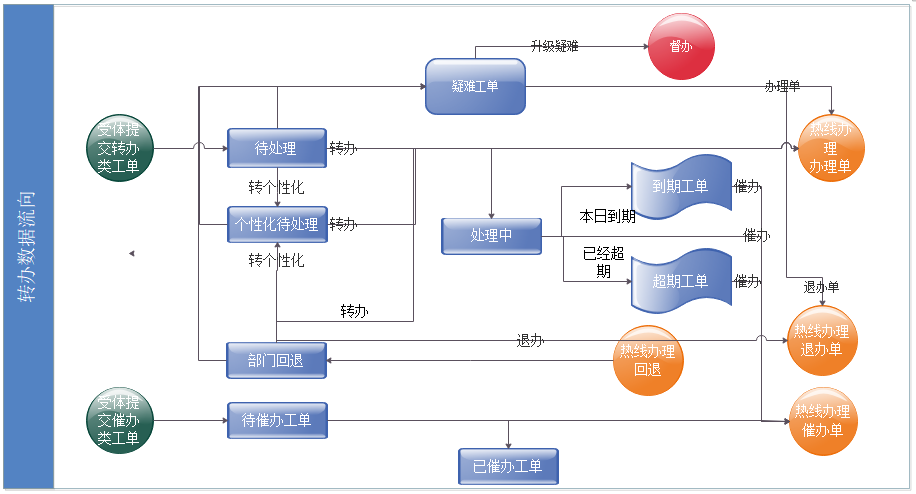
### 专家坐席

当问题分类选择了需要专家坐席受理的类别时（目前是公积金类），话务员可以选择“转专家坐席受理”选项，受理类型需要选择“直办”，并提交，此电话将转给专家坐席，如果专家坐席全忙则进入队列等待。

专家坐席接听市民电话时系统读取该联系人历史工单，并打开最近的一张问题分类为专家坐席类的工单，由专家坐席继续为市民服务。

当市民在电话转接过程中挂断时，专家坐席可以在专家坐席待处理中获取未成功受理的专家坐席工单，通过外拨电话主动联系市民继续服务。

## 转办系统业务流程说明



转办人员人员使用转办系统，对受理人员提交的受理类型为转办类、催办类的工单进行转办和催办。

### 工单转办

**工单获取**

受理系统提交的所有工单由转办人员在待处理中获取，已经获取的工单将由个人进行处理，班长用户可以看到所有工单。

**工单释放**

对于已获取的工单，如果无法继续处理，则可以进行释放，被释放的工单可以被其他转办人员获取到。

**工单转办**

对于已获取的工单，提供转办功能，可以向办理单位发送办理单，交由办理单位进行办理。

### 转个性化待处理

转个性化待处理相当于工单标记功能，可以将工单转入个性化待处理，其中的功能与待处理一致。

### 转办退受理

受理人员提交的转办工单，如果转办人员认为存在问题的工单，转办人员可以使用退受理功能，退回到受理人员。

### 转办返受理

受理人员提交的转办工单，如果转办人员认为可以直办的工单，转办人员可以使用返受理功能，返回给受理人员，此时受理人员可以进行直办办结。

### 部门回退

对于部门认为非本单位职责内工单可以进行回退，回退的工单继续回到转办系统，由对应的转办人员继续处理，此时转办人员可以发送退办单，由原单位继续处理，也可以发送办理单将工单交给别的单位处理。

### 工单催办

对于受理人员提交的催办类工单，转办人员可以在待催办工单中查看（系统自动根据被催办工单派发到被催办工单的转办人员），转办人员可以在工单中发送催办单。

### 到期工单

对于已经转办并且本日超期的工单，将自动在到期工单中进行提醒，转办人员可以发送催办单进行催办。

### 超期工单

对于已经转办并且已经超期的工单，将自动在超期工单中进行提醒，转办人员可以发送催办单进行催办。

### 升级疑难

对于转办人员待处理、个性化待处理、部门回退状态工单，可以使用升级疑难将无法处理的工单转为疑难工单交给班长处理。

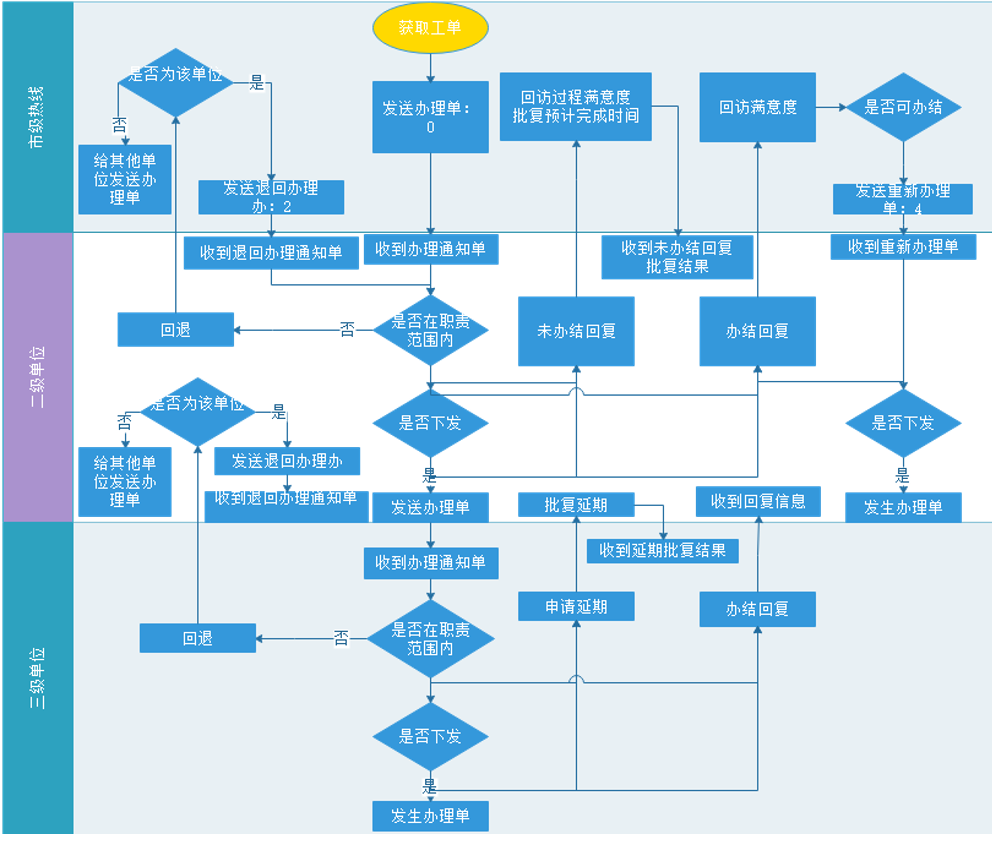
### 撤销疑难

班长对与已经升级为疑难的工单可以撤销疑难，工单将回到对应的转办人员升级疑难前的列表中。

### 升级督办

班长用户可以将无法处理的普通工单和升级疑难的工单发送给督办系统，交由督办人员处理，当督办人员撤销督办时，工单回到班长对应的升级督办前工单列表。

## 办理系统业务流程说明



办理单位接受市级热线转办发送的办理单、催办单、退办单、回访系统发送的重新办理单、督办系统发送的督办单，并对除催办单外的其他类型工单进行办理，二级单位具有回复、回退、下发转办的功能，其下级具有回复、回退、申请延期的功能。

### 回复

二级办理单位回复包含办结回复、未办结回复两种。

未办结回复反馈回访系统，进入初访环节，初访通过后根据未办结回复的预计完成时间更新办理单的办结时限，审核不同则可以延期两天，每个工单可以发送两次未办结申请，仅能申请通过一次，如果第一次审核通过，则无法申请第二次，如果第一次申请被驳回则可以申请第二次。

办结回复反馈回访系统，进入回访初审环节，审核通过则进入回访，审核不通过则发送重新办理单进入重办，当审核通过时由回访人员进行回访，回访办理过程和办理结果都满意时则可以办结，回访不满意则由回访人员发送重新办理单，由办理单位重新办理。

### 二级单位回退

对于不属于本单位业务范围内的工单可以使用回退功能，将工单退回到转办人员，当转办人员判定该工单属于本单位时，可以发送退办单由本单位继续办理，当判定不属于本单位时，则可以发送办理单给别的单位。

### 下发转办

当上级单位认为工单诉求涉及到具体下级时，可以将工单下发转办，发送办理单交由下级单位办理。

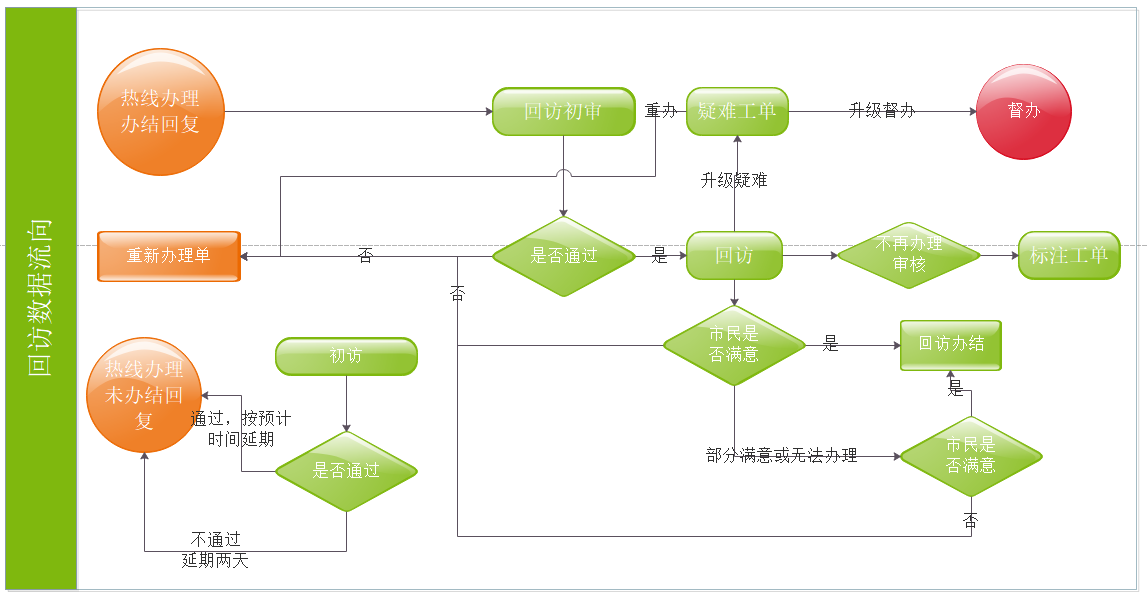
### 三级、四级…N级单位回退

回退操作同二级单位回退，只是审核人由转办人员变为上级单位。

### 申请延期

三级、司机…N级单位具有申请延期的功能，可以向上级发送申请延期申请，填写延期期限，如果上级审核通过则将工单办结时限延长至延期期限。

## 回访系统业务流程说明



### 初访

当办理单位进行未办结回复时，进入初访环节，初访通过后根据未办结回复的预计完成时间更新办理单的办结时限，审核不同则可以延期两天，每个工单可以发送两次未办结申请，仅能申请通过一次，如果第一次审核通过，则无法申请第二次，如果第一次申请被驳回则可以申请第二次。

### 回访初审

单办理单位进行办结回复时，进入回访初审环节，审核通过则进入回访，审核不通过则发送重新办理单进入单位重办。

### 回访

当回访初审审核通过时进行回访，回访办理过程和办理结果都满意时则可以办结，回访不满意则由回访人员发送重新办理单，由办理单位重新办理。

### 不满意审核

当办理单位多次办理办理过程满意度或办理结果满意度不能达到满意时，可以转入不满意审核环节，当审核通过时则办结工单，当审核不通过时交由办理单位重新办理。

### 不再办理审核

当办理单位回复内容中选定不在办理审核时，回访人员（回访审核，同不满意办结权限）可以审核不在办理诉求，审核通过则该工单状态变为不在办理，并且在受理系统中历史工单以特殊颜色标注，并排在第一位，提醒受理人员该工单级相关工单不在办理。

### 升级疑难

回访人员对于无法处理的工单可以使用升级疑难功能，将工单升级为疑难工单，由班长进行处理。

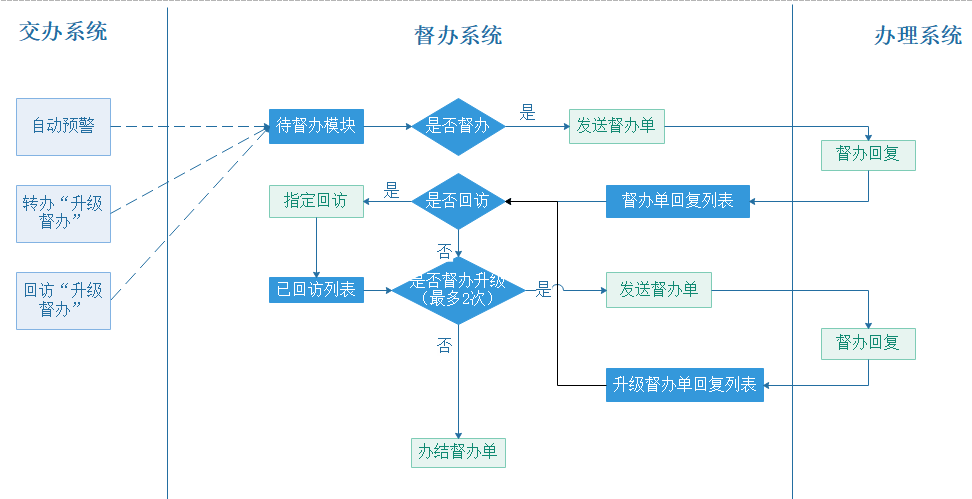
### 撤销疑难

班长用户可以将已升级的疑难工单进行撤销，撤销后的工单回到原工单列表。

### 升级督办

班长用户可以将无法处理的工单升级督办，转到督办系统由督办人员处理，如果督办人员撤销督办，工单回到原工单列表。

## 督办系统业务流程说明



将交办系统的自动预警模块系统自动抓取的数据，和转办升级、回访升级的数据，在此系统中按新督办流程流转处理，提供多种抓取功能：

1. 同一联系人不满意：
2. 同一事发地不满意：
3. 备注预警
4. 超期预警
5. 承办单位不满意
6. 台账预警

督办系统分为流程内督办和流程外督办，工单在未办结状态，转办升级督办和回访升级督办为流程内督办；督办人员根据业务系统预警的重点、难点、重复工单，发起的督办流程为流程外督办，流程外督办将由督办人员根据现有事件创建一张工单，并对新建工单督办，不影响历史工单。

流程外督办与流程内督办的督办流程一致，当存在需要督办的工单时，督办人员向办理单位发送督办单，办理单位回复后进入督办回复状态，督办人员可以对已回复督办单指派回访人员进行回访，需要区分于回访系统中的普通工单回访，回访的结果返回督办系统，督办人员根据回访结果发送督办重办单或督办办结。

# 性能需求分析

## 信息量分析与预测

目前热线拥有工单3300万条数据，办理单数量1500万条，回复量1900万，回访量1300万，2016年工单量3196425，2017年工单量3775168，2018年工单量5761697，2019年截至10月14工单量6160120，预计2019全年工单量7300000，年增长量约25%，按照此增量预计五年后数据总量将达到1.1亿。

当工单数量达到亿级，办理单数量约5000万，回复记录约7000万，回访记录约6000万。按照此量级统计、分析的无法满足要求，综合查询（工单管理、列表查询）等效率在关系型数据库中的效率也将急剧下降。

为满足系统要求，对于统计分析类业务采用大数据平台（Aache Kylin或spark）+固化报表的结合形式，针对综合查询类需求，通过划分备案表与实时状态表区分的方式，结合历史查询与实时查询提供高效查询。

## 性能设计

可支持系统最大用户数不限，支持同时在线人数3000人，最大并发操作数500。

# 详细建设内容

## 系统架构设计

全部系统模块架构图如下：



其中需要在门户中展示的内容包括：

用户权限管理系统：用于管理用户账号、授权等信息；

日志审计系统：用于查询系统访问日志、业务操作日志、保密工单查看日志等。

信息服务：提供知识库、公告、系统消息推送、通知等。

系统门户：提供统一的用户工作台，提供各模块快捷访问方式。

即时通讯：提供局域网在线聊天，群组聊天等功能。

受理系统：话务员使用的诉求受理系统；

微平台：互联网受理渠道拓展系统；

市级转办系统：济南市热线转办系统，用于济南市内工单的转办、催办类工单的转办和催办；

省级转办系统：山东省热线工单转办系统，用于山东内工单的转办、催办类工单的转办和催办；

市级办理系统：济南市热线办理系统，用于办理济南市内市民诉求；

省级办理系统：济南市热线办理系统，用于办理山东省内市民诉求；

市级回访系统：济南市内热线回访系统，用于审核和回访市办理办理系统已回复的工单；

省级回访系统：山东省热线回访系统，用于审核和回访省热线办理系统已回复的工单；

市级督办系统：用于督办市级热线工单办理，抓取重点工单进行督办；

省级督办系统：用于督办省级热线工单办理，抓取重点工单进行督办；

考勤系统：用于对受理人员、转办人员、回访人员进行考勤、请假等管理功能；

考核系统：用于对办理单位进行考核，生成考核报表，评分等。

质检系统：用于对话务人员、回访人员、转办人员进行质检、评分。

民意调查系统：用于民意调查业务，针对各个单位，生成问卷进行民意调查。

统计系统：对受理、转办、统计、办理等业务进行统计，生成报表。

预警系统：基于智能分词对热点、地图分布、事发地、反应问题的同比反比行政预警信息。

数据共享系统：基于热线数据库，形成单位图标，使办理单位可以共享市热线数据，保障数据的真实性。

大屏展示：提供大屏展示功能，包括热线大屏展示，地图分布，民调大屏展示内容。

移动办理：提供移动办理APP的下载说明和二维码，供单位下载。

社区眼神：用于社区、街道等没有行政服务域网络的单位使用的办理系统，不在门户首页中展示。

办理拓展接口：提供统一的接口文档，供外单位独立办理系统调用，进行获取待办诉求、回复、回退工单，不在门户首页中展示。

地图拓展：支持多方地图，百度离线地图，天地图等，不在门户首页中展示。

话务平台：提供话务平台的链接展示，可以跳转到话务平台管理界面。

智能知识库：集成智能知识库功能（第三方），提供链接进行跳转。

智能分类、智能分办：提供工单录入时的智能分类，受理和转办系统的智能分办功能，不在门户首页中展示。

材料分析系统：基于智能分词，为材料分析人员提供可靠的诉求分类、分类功能，为材料分析人员提供统计图、表等。

坐席助手、语音转写：在受理系统中集成坐席助手、语音撰写功能（第三方），进行实时语音转写，知识推荐等功能，不在门户首页中展示。

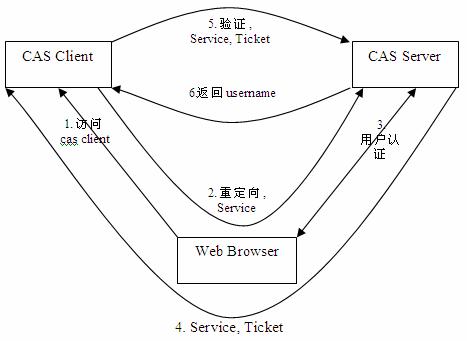
智能问答机器人：集成智能问答机器人（第三方），基于智能知识库集成问答机器人，提供在线问答功能，不在本期建设内容中，不在门户首页中展示。

信息服务包括知识库、公告、固定公告、通知消息服务等，用于信息展示，通知类业务的消息管理，基于WebSocket 协议的通信、通知，为所有的子系统提供消息通知，对外提供接口，为其他子系统的业务消息推送，公告通知，系统消息等信息推送等服务。

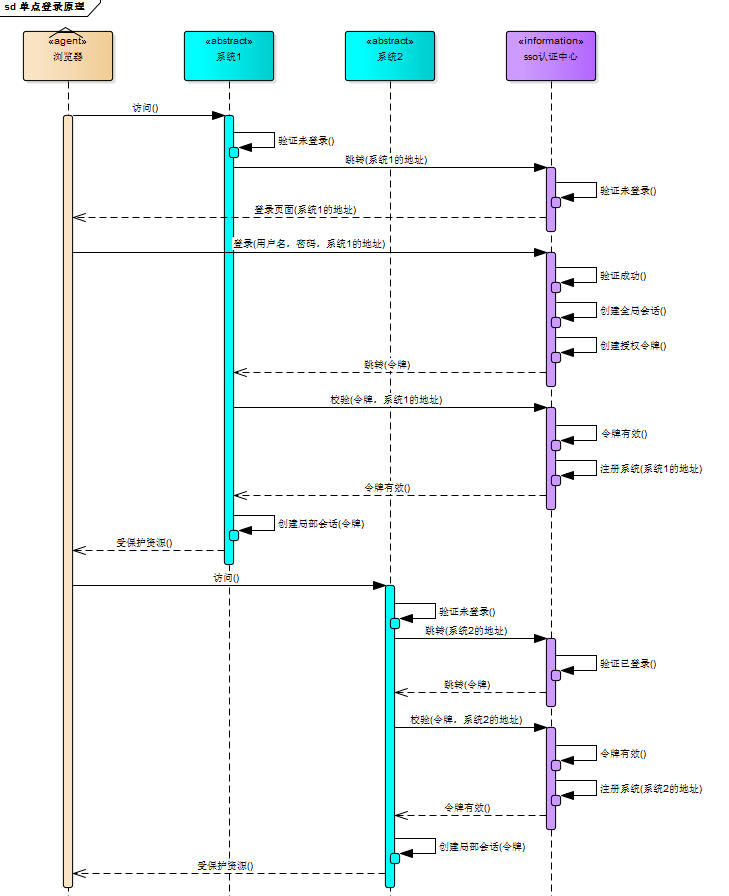
## 基础支撑部分

### 认证中心

包括统一认证系统，统一认证系统为系统提供SSO单点登录，基于Apache Cas 开源项目开发，采用OAuth2 协议进行认证。

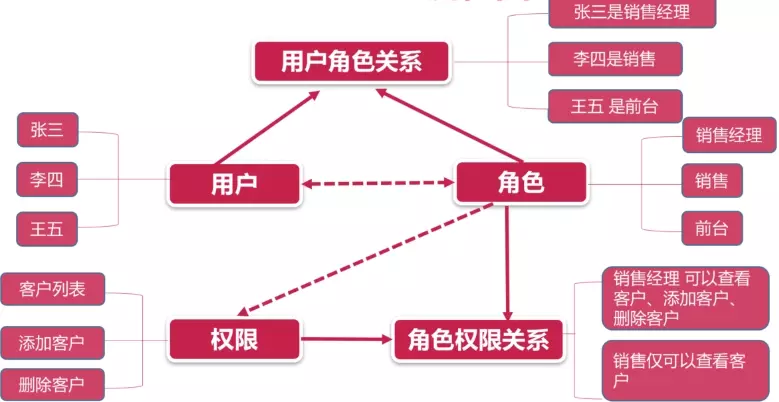


CAS单点登录原理如下：

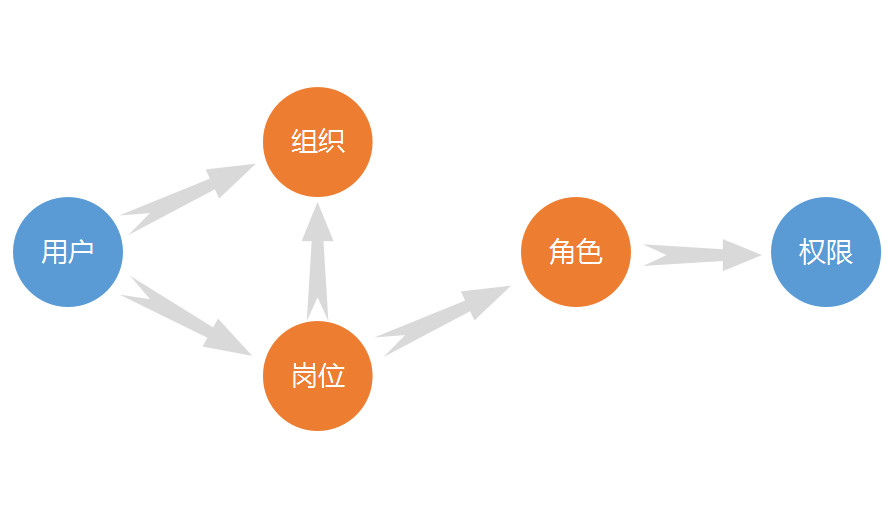


### RBAC用户权限管理系统

RBAC基于角色的权限访问控制（Role-Based Access Control）是商业系统中最常见的权限管理技术之一。在RBAC中，权限与角色相关联，用户通过成为适当角色的成员而得到这些角色的权限。这就极大地简化了权限的管理。在一个组织中，角色是为了完成各种工作而创造，用户则依据它的责任和资格来被指派相应的角色，用户可以很容易地从一个角色被指派到另一个角色。角色可依新的需求和系统的合并而赋予新的权限，而权限也可根据需要而从某角色中回收。



基于RBAC设计模型的基础上，增加组织、岗位的设定，避免因为人员的调动造成的权限混乱。



### 系统门户

系统用户作为用户登录后的工作台，根据角色划分功能，系统门户采用类window桌面的形式，提供功能模块展示，公告通知，消息推送等功能。基础业务部分

### 受理系统

话务员使用的基础业务系统，受理系统作为7\*24小时运行的系统，作为热线业务的核心业务系统，秉承模块化，低耦合的原则，尽量精简受理系统，仅保留话务员话务相关的业务内容，包括：待办事宜，外呼处理、联系人管理、坐席状态、保密坐席、专家坐席。

受理系统中问题分类、专项分类、转办单位、常用字典项采用本地存储的形式进行临时缓存，在系统初始化时同步加载，避免补录工单时的频繁加载。

#### 系统通知

基于信息服务系统的全局系统消息通知，对于业务提醒，通知公告、知识库审核信息进行提醒，其中公告信息需要在页面底部滚动展示。

现有系统共包含：未处理已标记号码工单（数量）、已回答未更新知识库建议（数量）、通知公告、紧急事件通知、紧急上报预警工单提醒、签到信息等提醒内容。

#### 待办事宜

作为话务员最常用的模块，使用Tab页加角标提醒的的形式记录话务员所有的已保存未提交工单、转办退受理、转办反受理、记事本等信息，待办角标基于信息服务进行推送。

#### 工单管理

工单管理功能放在交办系统中，采用超链接的形式引入。

#### 联系人管理

联系人管理用于关于所有来电人信息，号码、姓名等可能存在重复。

联系人管理提供基于所有联系人信息的查询功能。

### 微平台

### 转办系统

### 办理系统

### 回访系统

### 督办系统

## 辅助决策部分

### 统计系统

### 预警系统

### 考核系统

## 智能化部分

### 语音转文字

### 智能转办

### 智能分类

### 智能知识库（拟建设）

## 数据库设计说明

### 业务设计

### 数据库设计规范

数据库中所有表中统一使用Guid作为主键，针对需要Code进行统计、层级查询等需求在主表冗余存储Code。

数据库中所有表添加最后更新时间字段，用于避免脏数据。

系统中所有字段设计采用英文加下划线的形式设计：xxxx\_xxxx。

## 项目建设计划