Infra Draft Analysis of Software Requirement

基础架构软件需求预分析

|  |  |  |  |
| --- | --- | --- | --- |
| Prepared by  拟制 | 陈远超 | Date  日期 | 2018-09-20 |
| Reviewed by  评审人 |  | Date  日期 |  |
| Approved by  批准 |  | Date  日期 |  |

Revision Record 修订记录

| Date  日期 | Revision Version  修订 版本 | CR ID / Defect ID CR号 | Sec No.  修改 章节 | Change Description  修改描述 | Author  作者 |
| --- | --- | --- | --- | --- | --- |
| 2018-09-20 | 1.0 |  |  | 初始化版本 | 陈远超 |
| 2018-10-08 | 1.1 |  |  | 补充决策分析 | 龙江 |
|  |  |  |  |  |  |

Catalog 目 录

1 Background Information 背景介绍 6

2 Title of The Requirement 需求标题 6

3 Detail Analysis of Requirements 需求的详细分析 6

3.1 Detail Description of Requirements 需求的详细描述 6

3.2 Application and Topologies of Requirement 需求的应用和组网情况 7

3.3 Information of Other Manufacturers 竞争对手的相关信息 7

3.4 Analysis of Technical 技术分析 8

3.4.1 Analysis of Technical Nodus 技术难点分析 8

3.4.2 Impact on Performance, Maintenance and Upgrade 性能、维护、升级影响分析 8

3.4.3 Technical Alternatives 技术备选方案 8

3.4.3.1 备选方案一： 8

3.4.3.2 备选方案二： 9

3.4.3.3 方案优劣对比 9

3.4.3.4 参与决策的CCB人员名单： 9

3.4.4 Business Alternatives 业务备选方案 9

3.4.4.1 备选方案一： 9

3.4.4.2 备选方案二： 10

3.4.4.3 方案优劣对比 10

3.4.4.4 参与决策的CCB人员名单： 10

3.5 Analysis of Issue Across Modules 跨模块问题分析 11

3.6 Analysis of Reuse 重用分析 11

3.7 Estimation 估计 11

3.8 Result of Draft Analysis 预分析结论 13

Table List 表目录

表1 代码规模估计 11

表2 风险估计 12

表3 开发工作量 14

表4 评审工作量 14

Figure List 图目录

**未找到目录项。**

Infra Draft Analysis of Software Requirement

Infra软件需求预分析

Keywords 关键词：用户管理、角色管理、分级授权、用户行为分析

Abstract 摘 要：

List of abbreviations 缩略语清单：

| Abbreviations缩略语 | Full spelling 英文全名 | Chinese explanation 中文解释 |
| --- | --- | --- |
| INFRA | Infrastructure | 本意基础架构，基础架构软件的代号 |
| TPS |  | 每秒钟request/事务 数量 |
|  |  |  |
|  |  |  |

# Background Information 背景介绍

紫光华智打造的智慧安防平台由多个产品通过多应用融合技术的技术整合而成，这些应用由不同的团队独立开发，开发过程中遵循一致的UI规范，用户最终看到的是一套风格样式完全一致的解决方案。但是各个应用产品之间存在一些共有的或通用的功能组件，比如用户管理、组织管理、角色权限管理等，传统的做法是每个应用产品各自实现了一套这些功能，为了解决这些通用组件的可重用性和工作重复性问题，为保证用户组织数据的一致性和角色权限的统一集中管控，迫切需要一套基础架构软件（简称INFRA）来统筹实现并管理这些功能。

# Title of The Requirement 需求标题

* 基础架构软件R版本用户管理
* 基础架构软件R版本组织管理
* 基础架构软件R版本角色管理
* 基础架构软件R版本权限管理
* 基础架构软件R版本用户操作日志
* 基础架构软件R版本用户行为分析

# Detail Analysis of Requirements 需求的详细分析

## Detail Description of Requirements 需求的详细描述

需求名：用户管理

说明：用户管理包括用户基础信息维护、用户查询接口和用户数据同步

目的：实现集中用户管理功能，提供统一的用户视图。

需求名：组织管理

说明：组织管理包括组织基础信息维护、组织查询接口和组织数据同步以及分级管控。

目的：实现组织管理功能，统一集中维护人员组织架构主数据，保证数据的准确性和一致性。

需求名：角色管理

说明：角色管理包括角色维护、角色成员维护、功能权限和数据权限赋予。角色管理要求支持分级授权和权限回收。

目的：作为统一的角色权限管理中心，集中维护角色权限。

需求名：权限管理

说明：对功能权限菜单目录树的维护。

目的：以便角色管理模块可以为用户选择功能授权。

需求名：用户操作日志管理

说明：用户操作日志管理包括操作日志存储和操作日志查看功能。

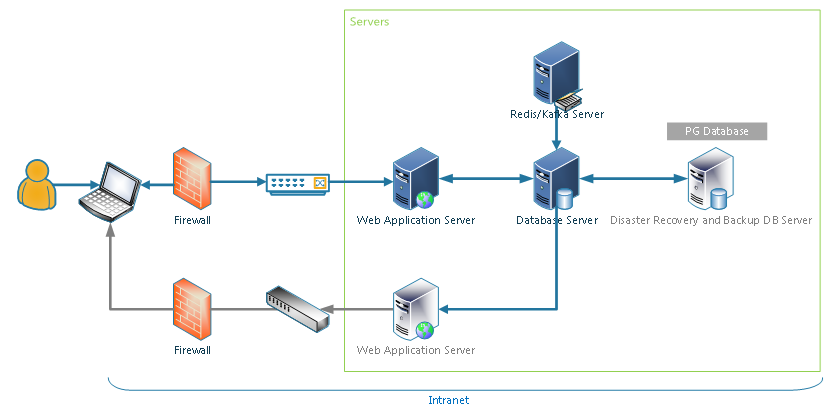
目的：实现统一的用户操作日志管理采集和呈现。

需求名：用户行为分析

说明：用户行为分析包括用户经常访问功能、页面停留时间、用户画像、用户行为路径、用户群组管理等功能的分析。

目的：实现用户行为分析，分析结果为产品优化、收藏、推荐、动态排序等功能提供支持。

## Application and Topologies of Requirement 需求的应用和组网情况



## Information of Other Manufacturers 竞争对手的相关信息

基础架构软件所提供的功能是非常通用和基础的服务，我们的设计目标是高度抽象和通用，降低前端应用的工作量。竞争对手海康威视也在这方面做了一定的提炼，他们目前只有基本的用户和权限管理，不支持分级管控和多级授权，没有用户行为分析，这些点是我们的优势所在。

## Analysis of Technical 技术分析

### Analysis of Technical Nodus 技术难点分析

需求名：用户管理

技术风险分析：1、用户管理核心接口高并发请求支持，为满足核心接口的并发性能需要使用缓存技术，使用缓存主要问题在缓存过期的处理，缓存过期有两种场景，一是缓存自动过期，另一种情况是由于资源的变动触发缓存的过期。2、用户信息、组织信息、权限信息的可扩展性，设计数据结构能够适用于不同系统的业务场景。

需求名：用户行为分析

技术风险分析：1、设计合理用户行为分析模型，模型需要保证可扩展性和计算结果的准确性以及能够适用于大多数业务场景。2、用户行为会有海量的数据，如何保证海量数据的计算效率。

### Impact on Performance, Maintenance and Upgrade 性能、维护、升级影响分析

用户数据量：约4.4万

性能分析：

1. 用户管理性能分析

获取用户详情接口满足 2000TPS 平均响应时间小于30ms

批量获取用户详情接口满足获取100条数据1000TPS平均响应时间小于50ms

1. 用户行为分析性能分析

用户行为分析埋点数据采集接口满足1000TPS平均响应时间小于20ms

1. 操作日志记录性能分析

操作日志记录接口满足2000TPS平均响应时间小于30ms

维护、升级分析：

Infra主要分为用户行为分析、用户操作日志记录、用户管理三个模块，项目采用微服务管理方式，每个模块作为一个单独的服务，具有很好的可维护性和性能的扩展性，但微服务方式对模块的管理和发布要求较高。

### Technical Alternatives 技术备选方案

#### 备选方案一：

Web前端技术栈：recate、nodejs、antd、dva

业务端技术栈：

框架：业务端基于SpringBoot、SpringMVC、Mybatis技术构建

五成分离：controller、service、manager、mapper、model，controller主要用于和前端进行交互，mapper用于数据库访问，manager用于构建缓存或调用其它业务接口。

缓存：Redis缓存

消息队列：kafka

#### 备选方案二：

Web前端技术栈：recate、nodejs、antd、dva

业务端技术栈：

框架：业务端基于SpringBoot、SpringMVC、Mybatis技术构建

四成分离：controller、service、mapper、model，controller主要用于和前端进行交互，mapper用于数据库访问。

#### 方案优劣对比

备选方案一优点：架构基于JavaEE，技术成熟稳定又能保证先进性。Redis缓存技术可能保证高频率访问接口的性能（如获取用户详情、获取用户权限列表接口），在一秒1000次的频率下也能保证效率；使用kafka消息队列进行错峰处理，可以避免同时间大量的日志写入数据库压力过大，100个线程，循环1000次，只要数据库连接够用，完全能够支撑下来。

备选方案一缺点：五层分离比较复杂，对代码规范的要求较高，各层之间相互依赖较小；依赖Redis、kafka组件增加了维护的成本，部署时需要部署redis、kafka、服务器端、前端四个服务，同时需同时保证四个服务的正常运行。

备选方案二优点：结构简单，依赖的第三方组件较少，开发和维护的工作难度较小，仅需关注自身服务器端、前段两个服务正常运行即可，无需额外分出1-2人处理第三方组件相关服务。

备选方案二缺点：缺少热点数据缓存，高并发情况下服务器的压力较大，TPS达到2000左右时，接口响应时间便会达到1-2秒左右。

|  |  |  |  |
| --- | --- | --- | --- |
|  | 代码量 | 可维护性 | 周边影响 |
| 方案一 | 10k | 使用部分第三方组件，维护所需资源增加；采用五层分离方式，代码耦合降低，由于多了一层架构，复用率较高，上层可相互调用下层代码，达到代码下沉的目的 | 各第三方组件确保接口效率，2000TPS平均响应时间小于30ms |
| 方案二 | 9k | 代码重复度较高，由于Service层不能相互调用的原则，各类重复代码均不能提出 | 大规模情况可能会影响接口效率，2000TPS平均响应时间达到1-2秒 |

经CCB成员评审，决定选择备选方案一：

五层分离更加灵活；使用redis缓存技术和kafka消息队列虽然增加了开发和维护的成本，但是有利于提高接口的性能。

#### 参与决策的CCB人员名单：

饶云、陈远超、谭刚、刘然、龙江、冯岭子、邓翔天、向妤、韦于评、毕芳娅、贺思思。

### Business Alternatives 业务备选方案

#### 备选方案一：

数据库：共application，dictionary，menu，organization，resource，role，user\_info，user\_role\_mapping，role\_menu\_mapping，role\_resource\_mapping十张关键数据表。

逻辑设计：严格按照数据库范式，将用户、角色、菜单、资源之间的关系进行表单级联。

#### 备选方案二：

数据库：共application，dictionary，menu，organization，resource，role，role\_mapping，user\_info八张关键数据表。

逻辑设计：

1. 通过取消用户与角色、用户与菜单、用户与资源、组织与角色、组织与菜单、组织与资源、角色与菜单、角色与资源之间的表单级联关系，转而集中在role\_mapping关系表中，同时对组织下的用户进行数据冗余。
2. role\_mapping表中每一条数据均存在一个parent\_id，用于保存数据来源，存储规则为数据来源parent\_id+数据来源id。

#### 备选方案三：

数据库：共application，dictionary，menu，organization，resource，role，role\_mapping，role\_owner\_mapping，user\_info九张关键数据表。

逻辑设计：在备选方案二的基础上，将role\_mapping表拆分成role\_mapping和role\_owner\_mapping两张表，role\_mapping仅维护角色和菜单、资源关系，role\_owner\_mapping仅维护角色和用户、组织关系。

#### 方案优劣对比

备选方案一优点：各种关系十分明显，易于理解，同时也比较符合数据库设计规范。

备选方案一缺点：权限回收操作十分复杂，需进行嵌套循环等数据库操作，10000人左右的权限回收会直接崩溃。

备选方案二优点：进行权限回收十分方便，所有关系均在同一张数据表中，无需额外数据表的连接，同时由于parent\_id字段的设计，能一次删除所有来源的数据。

备选方案二缺点：添加关系时操作较为复杂，需进行一定程度的数据冗余，假如为一个拥有100个权限的角色添加一个1000人的组织时，则需冗余1000\*100条数据。

备选方案三优点：role\_mapping表拆分成role\_mapping和role\_owner\_mapping两张表，这样便无需存储用户和菜单、资源关系，数据量锐减，假如为一个拥有100个权限的角色添加一个1000人的组织时，仅需增加一条数据。

备选方案三缺点：数据模型复杂，理解困难。

|  |  |  |  |
| --- | --- | --- | --- |
|  | 代码量 | 可维护性 | 周边影响 |
| 方案一 | 5k | 数据耦合度低，但是操作困难， | 数据量过大时，对权限回收性能影响较大，万级的操作会崩溃 |
| 方案二 | 4k | 加大数据关系表耦合度，增加了添加时的维护难度，但是减小了删除时维护成本，权限回收仅需一条删除语句 | 分配权限冗余过多，万级的操作接口延迟达到10秒左右 |
| 方案三 | 4k | 适当解耦，不在维护用户菜单关系，将数据量从（用户数\*菜单数）减少至1 | 略微降低权限回收效率，需多连接一张表，50000数据量的情况下性能降低5ms左右 |

经CCB成员评审，决定选择备选方案三：

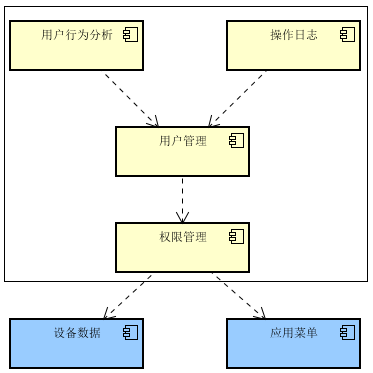
对于不同业务需要，数据库可进行一定程度的反范式设计，进行适度数据冗余，“备选方案三”相对来说操作还是简单了许多，同时大幅提高接口性能，各核心接口1000TPS平均响应时间小于20ms。

#### 参与决策的CCB人员名单：

饶云、陈远超、何健、王渝、谭刚、刘然、龙江、冯岭子、邓翔天、张冰、向妤、韦于评。

## Analysis of Issue Across Modules 跨模块问题分析

模块依赖关系图：



系统中主要分为四个模块，用户行为分析模块、操作日志模块、用户管理模块和权限管理模块，设备数据和应用菜单是其它业务系统数据。

问题分析：

1、用户行为分析和操作日志模块依赖用户管理问题分析

用户行为分析模块和用户操作日志模块都需要依赖用户管理模块的获取用户详情功能，由于用户操作日志模块的请求量很大，因此需要保证用户详情模块中获取用户详情接口的健壮性，支持高并发和批量用户查询等功能。为满足用户管理接口的高性能要求，需要使用Redis缓存和kafka消息队列技术，对Redis缓存一致性、缓存过期、缓存穿透、雪崩效应等问题的解决是影响接口稳定性的主要原因。

1. 权限管理依赖其它业务方数据问题分析

权限管理所管理的数据来源于两部分，第一部分由业务方根据自己的需求配置的菜单数据，第二部分来源于设备管理系统的设备数据。对于设备数据的依赖有两个问题需要关心，一、设备数据接口的稳定性，二、设备管理系统数据和权限管理数据一致性同步。

## Analysis of Reuse 重用分析

开源使用情况：Infra使用的是Spring和Mybatis开源框架，Mybatis框架作为数据库层的ORM框架提供了便捷的数据库操作，代码在不同数据库之间进行切换有很好的可移植性，但是根据不同数据库任然需要修改部分SQL和配置信息。

组件重用：Infra具有很好的通用性，提供的用户管理、权限管理、用户行为分析、用户操作日志等功能，不同的业务系统只需要通过指定的方式进行接入就能使用。

## Estimation 估计

1.代码规模：

1. 代码规模估计

| 分配需求号 | 分配需求名称 | 代码量估计（KLOC） | 说明 |
| --- | --- | --- | --- |
| 1 | 组织机构管理 | 0.5 |  |
| 2 | 用户管理 | 0.5 |  |
| 3 | 分级管控 | 0.5 |  |
| 4 | 用户组织关系维护 | 0.5 |  |
| 5 | 角色管理 | 2 | 角色管理工作量最大 |
| 6 | 功能权限 | 1 |  |
| 7 | 数据权限 | 0.5 |  |
| 8 | 分级授权 | 1 |  |
| 9 | 权限回收 | 1 |  |
| 10 | 菜单管理 | 0.5 |  |
| 11 | 用户操作日志 | 1 |  |
| 12 | 日志归档 | 0.3 |  |
| 13 | 统计报表 | 0.2 |  |
| 14 | 埋点规则 | 0 | 规则定义不涉及代码 |
| 15 | 行为统计 | 0.5 |  |

2.难度与复杂程度：

项目难度适中，主要复杂工作集中在角色管理、分级授权和权限回收三部分上。分级授权决定了，在没有越级授权的情况下，不会出现下级权限大于上级的 现象；权限回收则代表，当前用户失去某权限时，他曾授予的该权限均需回收。这两部分操作均需进行严格控制，防止出现越权与漏权的情况发生。

3.周边模块影响度；

各模块之间除了共用统一数据库，耦合度均比较低，可单独拿出来运行、使用，在接口入参与出参不变的情况下，也均可进行完美分离。如果需接入角色管理，则在分级授权与权限回收上有一定程度的功能依赖。

4.人力投入：

90人/天

5.测试工作量：

40人/天

6.风险：

1. 风险估计

|  |  |  |
| --- | --- | --- |
| **描述** | **规避措施** | **应急计划** |
| INFRA是公共服务组件，会被其他系统调用，在INFRA的需求设计过程中，需充分考虑系统的通用性 | 需求设计阶段在以通用性和可扩展性为前提的条件下，经量多的考虑各干系人的特定需求，提供系统原型，邀请各方人员进行评审 | 走变更控制流程 |
| Web前段框架对浏览器有要求，如果IE浏览器版本过低，存在系统界面不能正常交互的可能 | 提前确认IE浏览器版本 | 采用其他技术方案 |
| 项目组团队成员由新员工组成，普遍工作经验不足，可能会存在不能按时保质完成工作任务的情况，会影响项目进度 | 1.对项目组成员进行专业技能培训；2.做项目预算的时候预留额外缓冲时间； | 借用项目组外部专业技能较强的同事对项目架构及关键模块帮忙设计，解决技术难点 |

## Result of Draft Analysis 预分析结论

接纳。

附录A：

1. 开发工作量

|  |  |  |  |
| --- | --- | --- | --- |
| 开发用户 | 页数 | 页数范围 | 工作量（小时） |
|  |  |  |  |
|  |  |  |  |

1. 评审工作量

|  |  |  |  |
| --- | --- | --- | --- |
| 评审用户 | 页数 | 页数范围 | 工作量（小时） |
|  |  |  |  |
|  |  |  |  |

附录B：

参考资料清单：