# 本期项目建设方案

## 应用支撑平台及应用系统建设

### 网信中台服务平台

#### 中台定位

中台是相对于前台后台的一个概念。

前台及是指直接面向用户的那一端。是由各类前台系统组成的前端平台。每个前台系统就是一个用户触点，即企业的最终用户直接使用或交互的系统，是企业与最终用户的交点。

后台是指由后台系统组成的后端平台。每个后台系统一般管理了一类核心资源（数据+计算）这类系统构成了企业的后台。

仅由前后台组成的系统在发展到一定规模之后都会遇到瓶颈。很多后台系统，在创建之初的目标，并不是主要服务于前台系统创新，而更多的是为了实现后端资源的电子化管理，解决企业管理的效率问题。这样的后台显然不能很好的支撑起前台，实现快速响应客户需求，高效创新。

为了机制化，产品化地解决这些问题，中台的思想逐渐成为共识。通过在前后台之间插入中台，成为前后台中间的配速器，解决前后台速率不一致的问题；另一方面，通过将（前台）通用业务的下沉，将早已臃肿不堪的前台系统中的稳定通用业务能力“沉降”到中台层，为前台减肥，恢复前台的响应力，通过将（后台）核心能力的聚合，将后台系统中需要频繁变化或是需要被前台直接使用的业务能力“提取”到中台层，赋予这些业务能力更强的灵活度和更低的变更成本，从而为前台提供更强大的“能力炮火”支援。

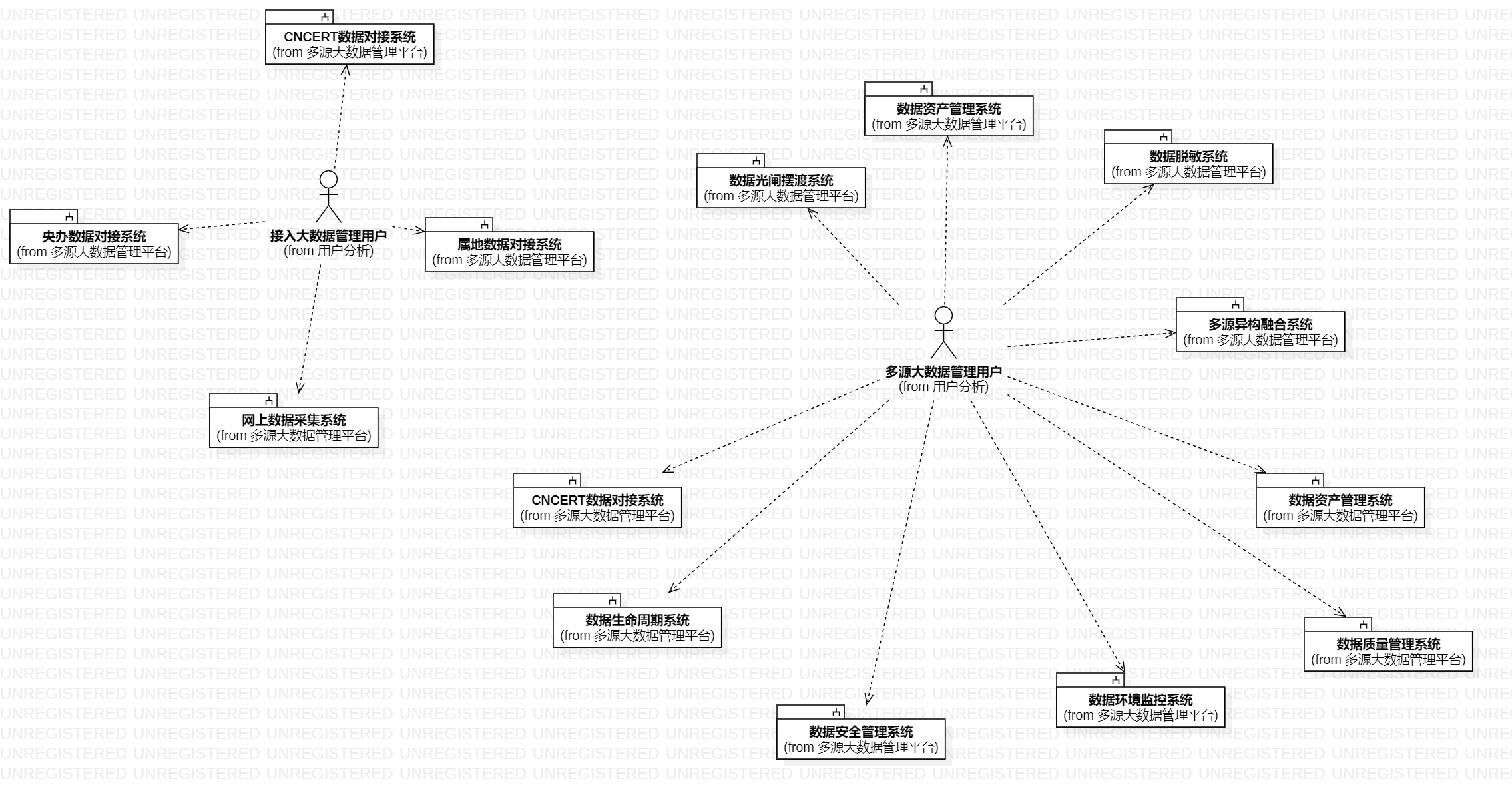
#### 中台组成

整个网信中台由业务中台、数据中台、技术中台3个中台外加支撑所有中台服务的中台底座，共计4部分组成。

**图 1.1‑2多源大户据管理平台-用户设计图**

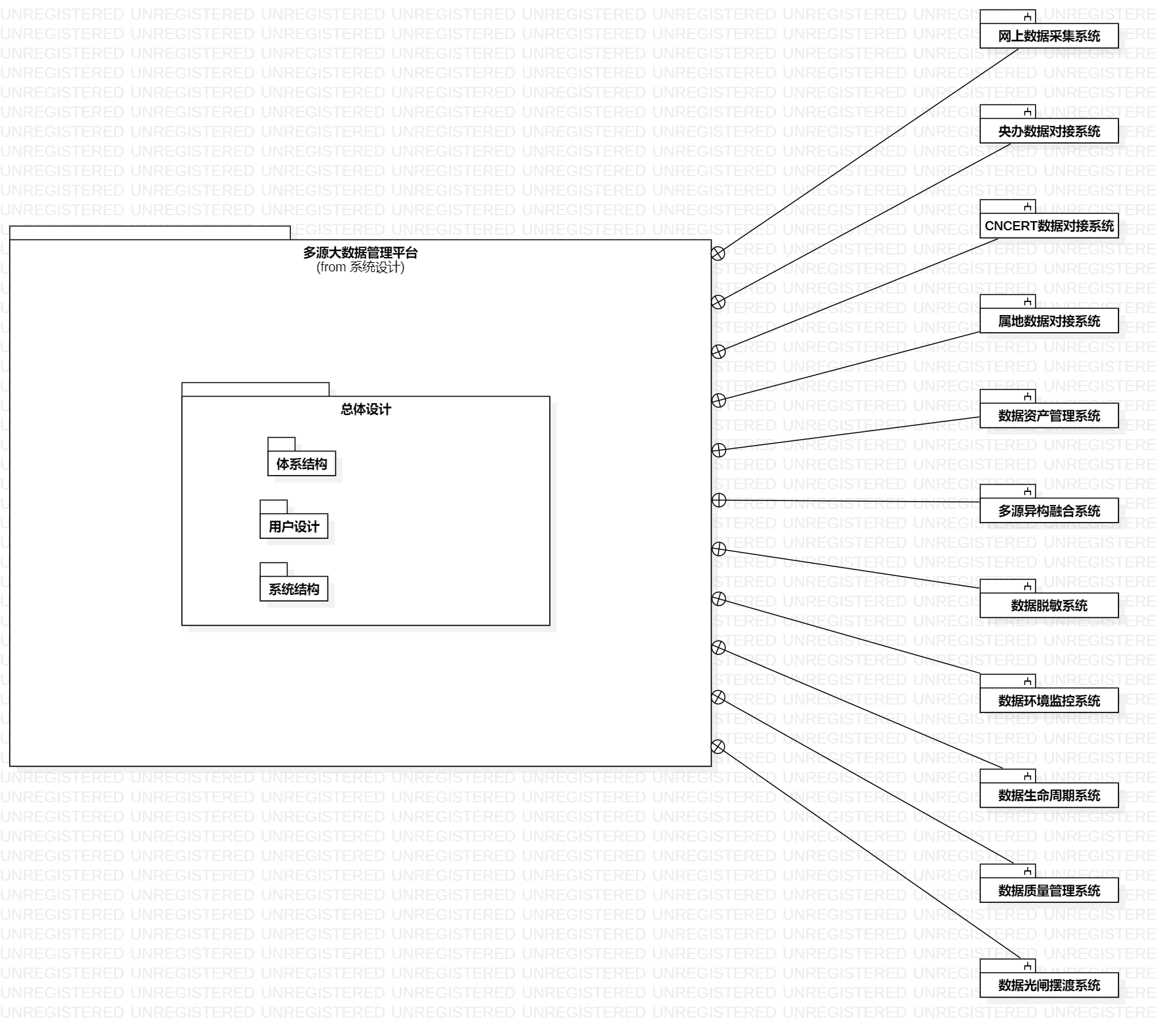
#### 用户设计

中台底座使用用户包括管理员和运营人员。



**图 1.1‑4多源大户据管理平台-用户与系统间关系设计图**

#### 中台体系结构设计

整个中台由聚合了后台系统支撑业务应用平台、多源大数据管理平台、网信智慧大脑三个平台19个系统的提供的能力和数据，并形成高复用性的中台服务层。

**图 1.1‑1多源大户据管理平台-平台组成图**

#### 中台总体设计

##### 用户与系统间关系设计

中台使用用户包括管理员和运营人员，具体用户分析说明：

1. 管理员

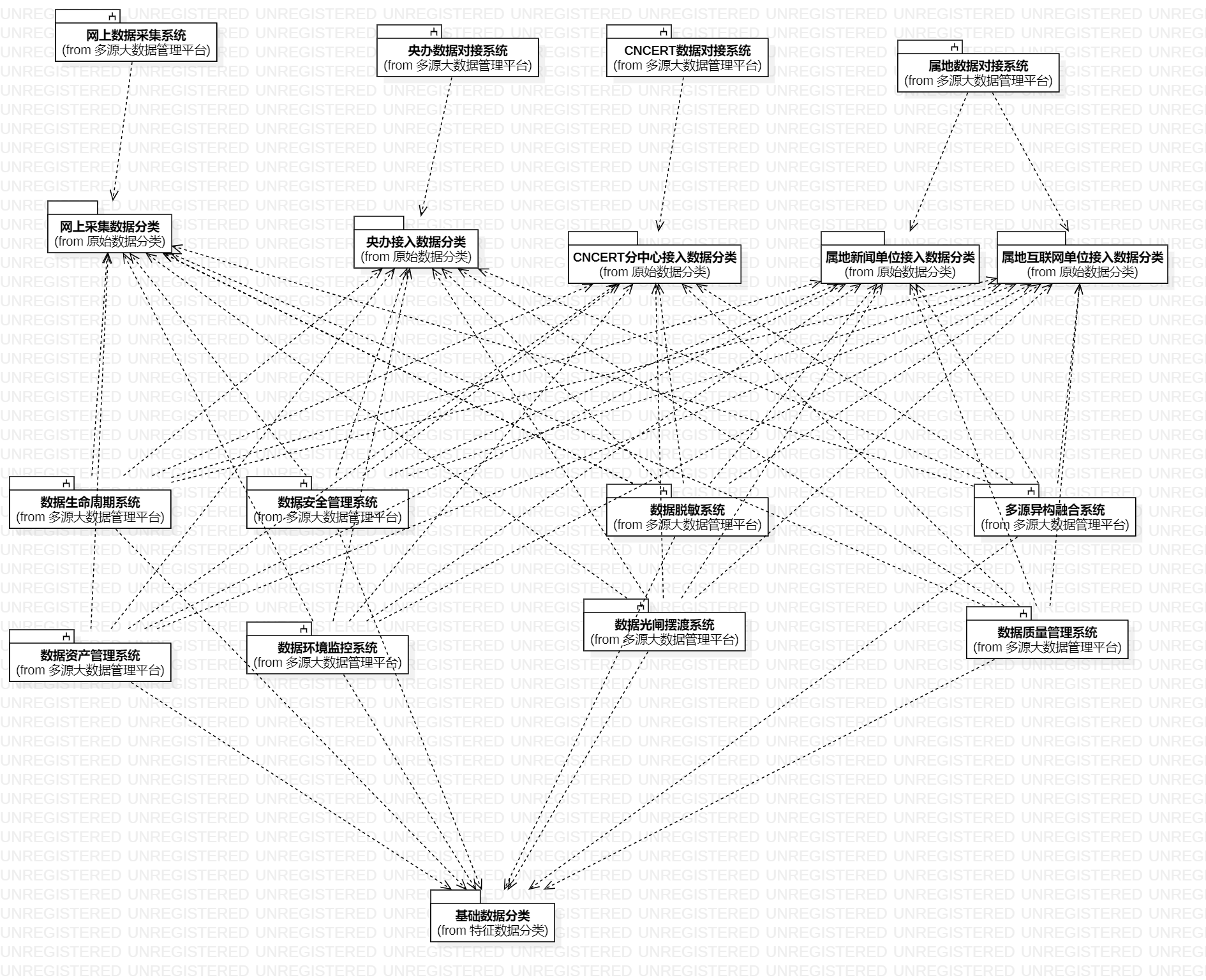
负责管理平台用户，服务使用情况等，同时监控各个主机的性能指标、日志信息等。

（二）运营人员

负责使用中台底座提供的能力管理、运营业务应用及各个中台服务，确保服务实例健康有效提供服务。

##### 数据与系统间关系设计

网信中台主要为前台网上传播监管系统、网上舆论引导系统、属地应急管控系统、应急研判指挥系统、应急有害发现系统这五大核心业务系统，以及未来的更多系统提供支撑。



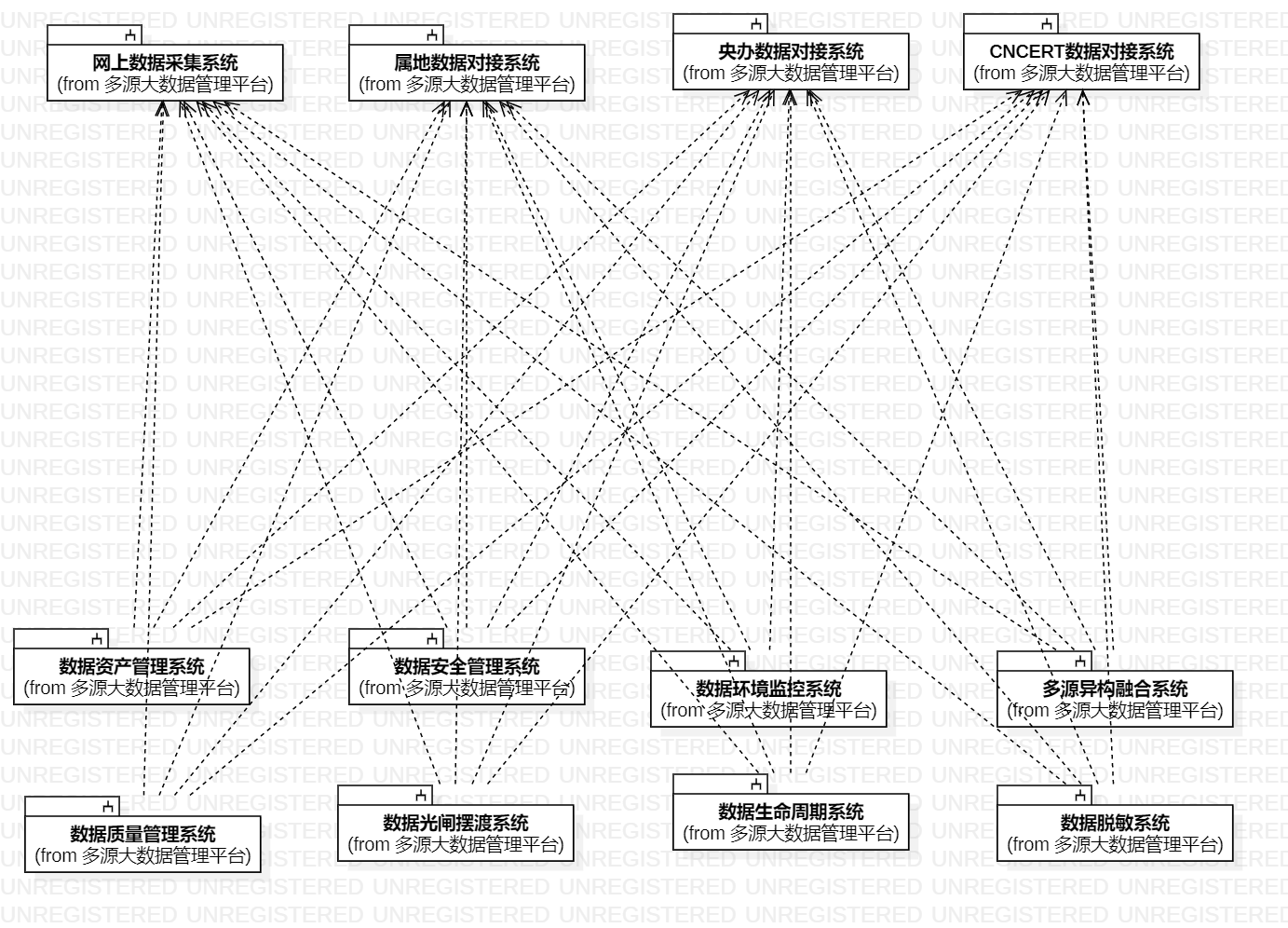
**图 1.1‑5多源大户据管理平台-数据与系统间关系设计图**

##### 各系统间关系设计（没有）

与多源大数据平台相关的内部系统包括网上数据采集系统。。。。。。、外部系统包括。。。。。。系统。关系如下：

发起系统，调用哪些相关系统，管理XXX东西；

1. 数据资产管理系统，与数据网上采集系统、属地数据对接系统、央办数据对接系统、CNCERT数据对接系统进行对接，管理数据资产；



**图 1.1‑6多源大户据管理平台-各系统间关系设计图**

#### 中台底座

##### 系统定位

中台是由大量服务组成，而中台底座即为所有中台服务提供管理、监控、审计、调度等全方位的支持的平台。

##### 技术路线

**1、容器运行时**

容器其实已有10来年的历史，它具有占用资源少、部署快、易迁移等特点。

由于容器的费用低、密度高，因而可在同一虚拟机或物理服务器内托管大量容器，这使得容器成了交付云应用的理想之选。而在当今市场上，Docker 已成为具有代表性的容器技术，它作为应用级别的容器引擎。具备更加全面的资源控制能力。

Docker是一个专为开发人员、系统管理员和DevOps设计的开源平台，它用于构建，传输和运行容器，使用Docker作为容器载体，让开发者打包自己的应用以及依赖包到一个可移植的容器中，利用中台管理门户和容器编排工具将这些应用发布到任何流行的 Linux 机器上，便可以实现虚拟化。

Docker改变了虚拟化的方式，使开发者可以直接将自己的成果放入Docker中进行管理。方便快捷已经是 Docker的最大优势，过去需要用数天乃至数周的任务，在Docker容器的处理下，只需要数秒就能完成。

**2、容器编排**

应用程序容器的部署和组织是通过容器编排工具完成的。一些流行的开源容器编排工具包括Kubernetes、Docker Swarm和LXC。结合公司技术架构的实际情况，我们选择使用kubernetes作为容器编排工具。kubernetes，简称K8s，它是一个开源的，用于管理云平台中多个主机上的容器化的应用，kubernetes的目标是让部署容器化的应用简单并且高效,Kubernetes提供了应用部署，规划，更新，维护的一种标准化机制。

使用K8s,可以实现：

1. 基于容器的应用部署、维护和滚动升级

2. 负载均衡和服务发现

3. 跨机器和跨地区的集群调度

4. 自动伸缩

5. 无状态服务和有状态服务

6. 插件机制保证扩展性

**3、服务网格**

服务网格是用于处理服务间通信的专用基础设施层。它负责通过包含现代云原生应用程序的复杂服务拓扑来可靠地传递请求。实际上，服务网格通常通过一组轻量级网络代理来实现，这些代理与应用程序代码一起部署，而不需要感知应用程序本身。

服务网格也被称为Service Mesh，它的功能包括应用程序间通讯的中间层、轻量级网络代理、应用程序无感知、解耦应用程序的重试/超时、监控、追踪和服务发现等等。服务网格的开源软件有 Istio、Linkderd、Envoy、SOFAMesh、Dubbo Mesh 等。

我们选用Istio作为微服务的服务网格，Istio 提供一种简单的方式来为已部署的服务建立网络，该网络具有负载均衡、服务间认证、监控等功能，而不需要对服务的代码做任何改动。它能够实现智能控制服务之间的调用流量，能够实现灰度升级、AB 测试和红黑部署等功能，自动为服务之间的调用提供认证、授权和加密，应用用户定义的 policy，保证资源在消费者中公平分配。，查看服务运行期间的各种数据，比如日志、监控和 tracing，了解服务的运行情况等等。

**4、API网关**

API网关是一个特殊的服务器，是系统的唯一入口。从面向对象设计的角度看，它与外观模式类似。API网关封装了系统内部架构，为每个客户端提供一个定制的API。它可能还具有其它职责，如身份验证、监控、负载均衡、缓存、请求分片与管理、静态响应处理。

API网关方式的核心要点是，所有的客户端和消费端都通过统一的网关接入微服务，在网关层处理所有的非业务功能。通常，网关也是提供REST/HTTP的访问API。服务端通过API-GW注册和管理服务。

选择Kong作为微服务的API网关，Kong 是在客户端和（微）服务间转发API通信的API网关，通过插件扩展功能，这些插件在 API 请求响应循环的生命周期中被执行。。它主要有两个组件组成，Kong Server和Apache Cassandra。Kong Server是基于nginx的服务器，用来接收 API 请求。而Apache Cassandra用来存储操作数据。Kong具有如下几个基础功能：HTTP 基本认证、密钥认证、CORS（ Cross-origin Resource Sharing，跨域资源共享）、TCP、UDP、文件日志、API 请求限流、请求转发以及 nginx 监控。

**5、采用分布式应用程序协调技术**

基于ZooKeeper项目进行系统集成。ZooKeeper是一个为分布式应用所设计的分布的、协调服务，它主要是用来解决分布式应用中经常遇到的一些数据管理问题，简化分布式应用协调及其管理的难度，提供高性能的分布式服务。ZooKeeper的目标就是封装好复杂易出错的关键服务，将简单易用的接口和性能高效、功能稳定的系统提供给用户。

**6、分布式监控**

Prometheus（普罗米修斯）是一套开源的监控&报警&时间序列数据库的组合.由SoundCloud公司开发。是为数不多的适合Docker、Mesos、Kubernetes环境的监控系统之一。

Prometheus基本原理是通过HTTP协议周期性抓取被监控组件的状态，任意组件只要提供HTTP接口就可以接入监控系统，不需要任何SDK或者其他的集成过程。

Prometheus的主要功能实现：

1. 一个多维数据模型（时间序列由指标名称定义和设置键/值尺寸）。

2. 非常高效的存储，平均一个采样数据占~3.5bytes左右。

3. 同时具备一种灵活的查询语言。

4. 不依赖分布式存储，单个服务器节点。

5. 时间集合通过HTTP上的PULL模型进行。

6. 可以通过中间网关支持推送时间。

7. 可以通过服务发现或静态配置发现目标。

8. 支持多种模式的图形和仪表板。

**7、日志聚合**

EFK是Elasticsearch、Fluentd、Kibana的简称。

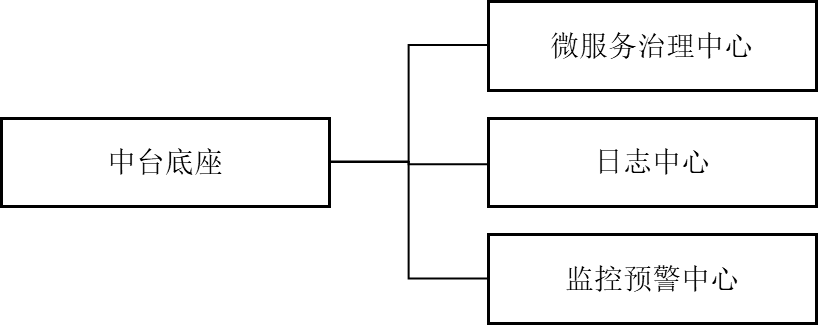
Elasticsearch 是一个分布式、可扩展、实时的搜索与数据分析引擎。它可以快速地储存、搜索和分析海量数据。维基百科、Stack Overflow、Github 都采用它。

Fluentd是一个开源的数据收集器，专为处理数据流设计，使用JSON作为数据格式。它采用了插件式的架构，具有高可扩展性高可用性，同时还实现了高可靠的信息转发。具备每天收集5000+台服务器上5T的日志数据，每秒处理50000条消息的性能。

Kibana是一个开源的分析与可视化平台，设计出来用于和Elasticsearch一起使用的。可以用kibana搜索、查看、交互存放在Elasticsearch索引里的数据，使用各种不同的图表、表格、地图等kibana能够很轻易地展示高级数据分析与可视化。

通过组合EFK工具，Elasticsearch负责日志保存和搜索，Fluentd负责收集日志，Kibana 负责界面，形成完整的日志聚合展示能力。

##### 系统组成



**图1.1‑10中台底座组成图**

##### 用户与角色设计

###### 用户设计

中台底座为技术中台、数据中台、业务中台提供支撑，主要被管理员和运营人员使用。

**图 1.1‑1网上数据采集系统-系统组成图**

###### 角色设计

系统主要使用人员情况如下表：

**表1.1‑1网上数据采集系统-用户与角色设计表**

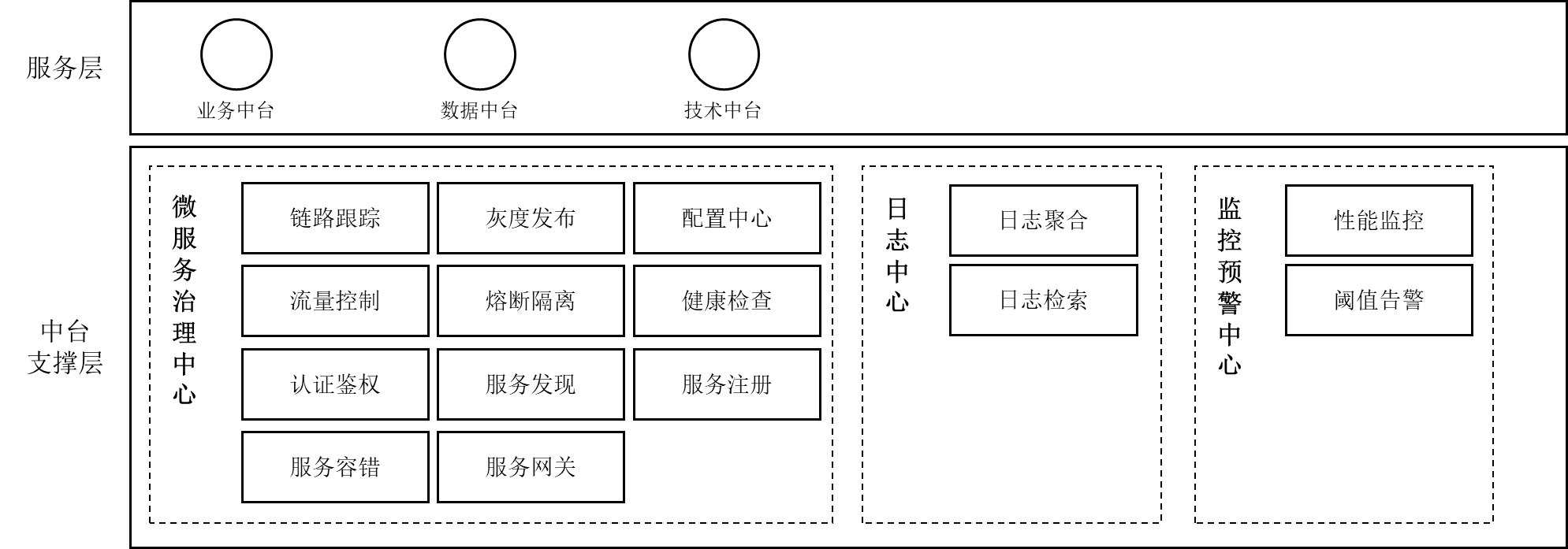
| **用户分类** | **系统角色** |
| --- | --- |
| 数据中心  管理人员 | 管理员、运营人员 |

具体用户分析说明：

**1、管理员：**负责管理平台用户，服务使用情况等，同时监控各个主机的性能指标、日志信息等。

**2、运营人员：**负责使用中台底座提供的能力管理、运营业务应用及各个中台服务，确保服务实例健康有效提供服务。

##### 系统体系结构设计

**

**图 4.1.1.4.3‑220 中台底座-系统体系结构设计图**

中台底座为上层中台服务提供支撑，底座由三部分构成：

**1、微服务治理中心**

微服务治理中心主要实现对容器化微服务的管理和调度。

**2、日志中心**

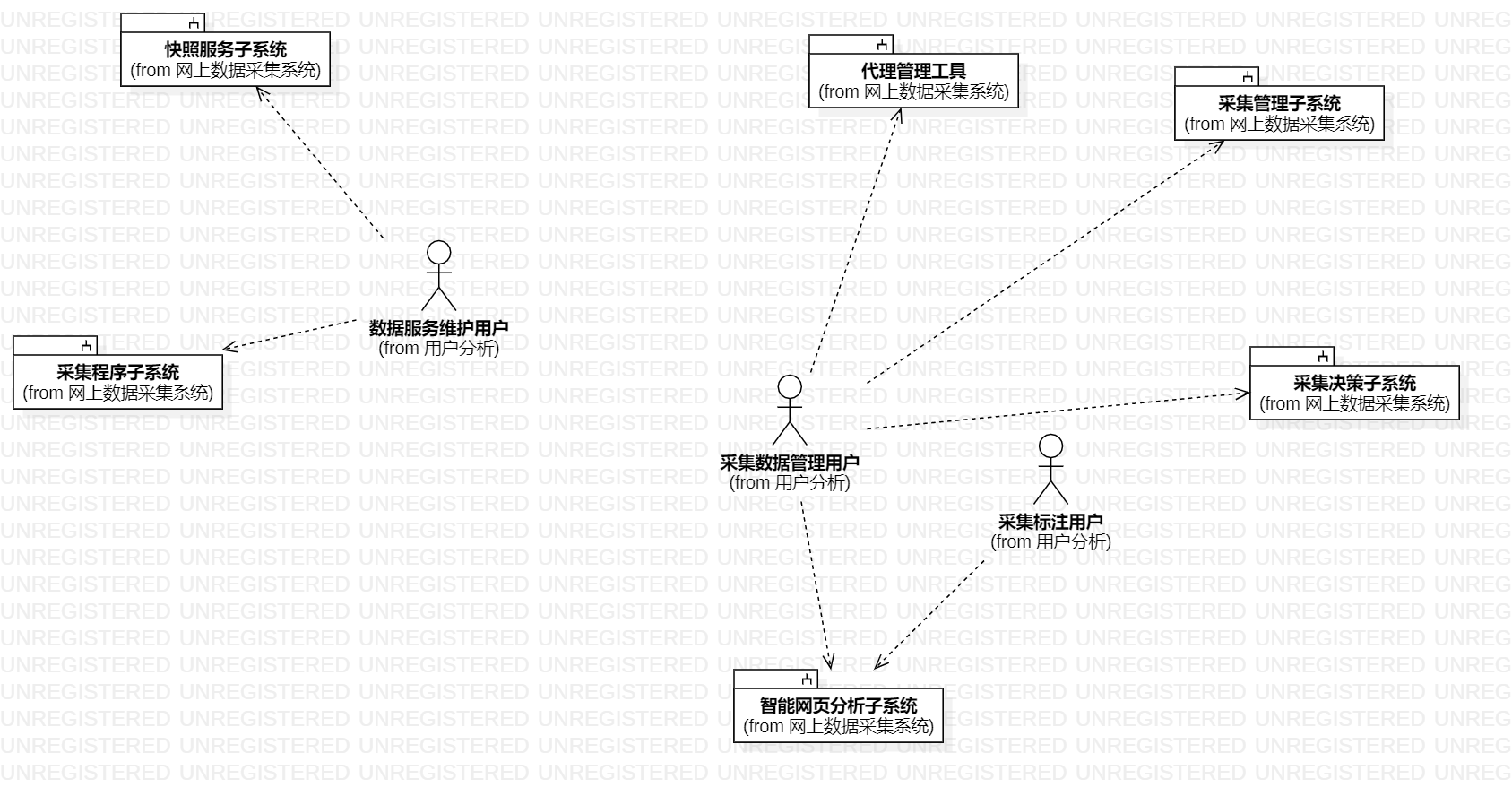
日志中心主要实现对不同主机不同容器的日志统一收集并提供检索功能。

**3、监控预警中心**

监控预警中心主要实现监控主机性能、服务实例资源占用等情况，并可设置阈值警报。

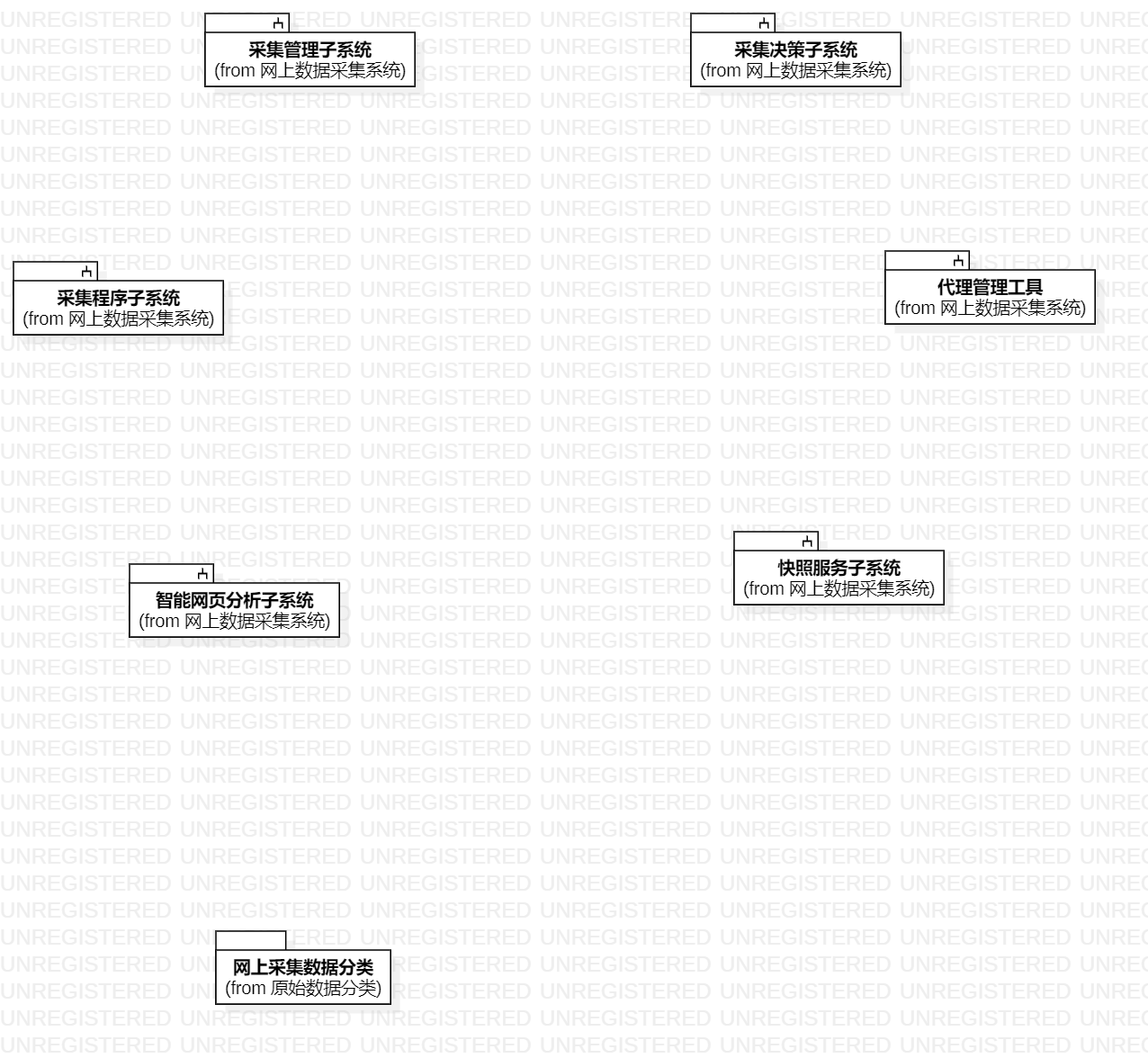
##### 系统总体设计（没有写）

###### 用户与系统间关系设计



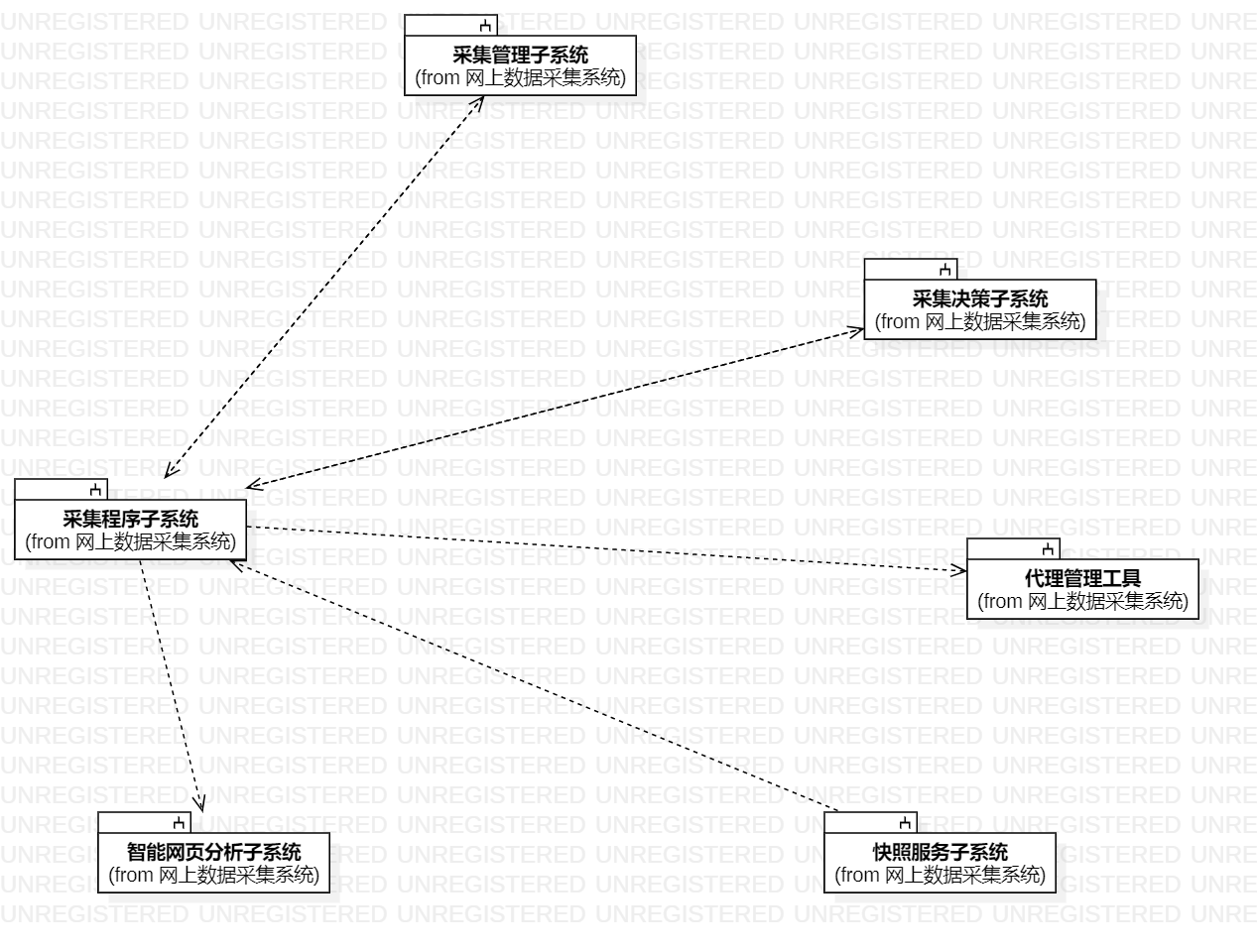
**图 1.1‑4网上数据采集系统-用户与系统间关系设计图**

###### 数据与系统间关系设计



**图 1.1‑5网上数据采集系统-数据与系统间关系设计图**

###### 各系统间关系设计



**图 1.1‑6网上数据采集系统-各系统间关系设计图**

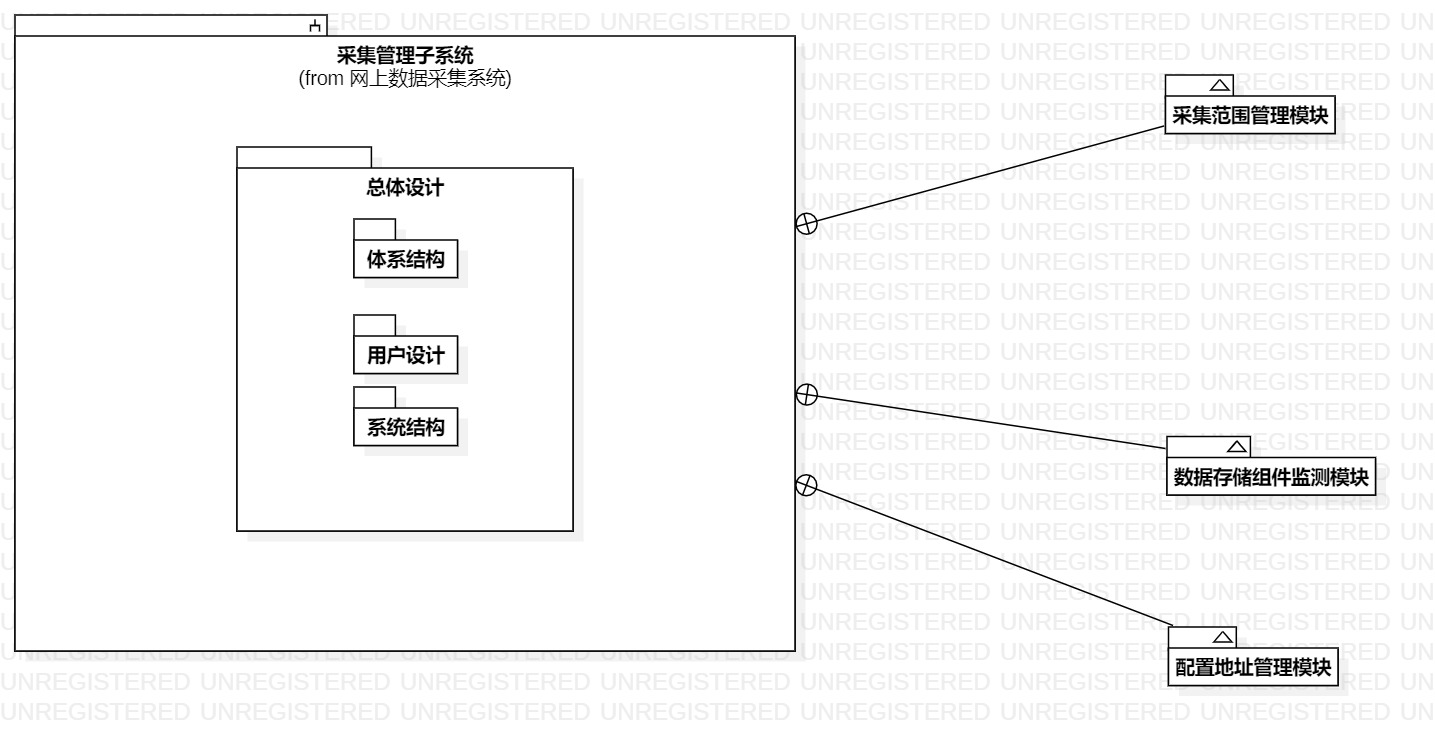
##### 微服务治理中心

###### 系统概述

通过微服务治理平台，可以调用配置中心提供的接口，动态地修改各种配置来实现服务的治理。治理包括限流、降级、切流量等功能，针对系统故障调用配置中心接口，修改非核心服务的限流阈值，通过降级一些非核心业务，减少非核心服务的调用，为核心服务留出充足的冗余度。是维持服务正常运行，维护系统安全的重要保障。

###### 系统组成

微服务治理中心由链路跟踪、灰度发布、配置中心、健康检查、流量控制、认证鉴权、熔断隔离、服务发现、服务注册、服务容错、服务网关共11个模块组成。



**图1.1‑14采集管理子系统-系统组成图**

###### 用户与角色设计

用户设计

微服务治理中心主要服务于管理用户，分为管理员和运营人员。



**图 1.1‑1采集管理子系统-用户设计图**

角色设计

系统主要使用角色情况如下表：

**表1.1‑2采集管理子系统-用户与角色设计表**

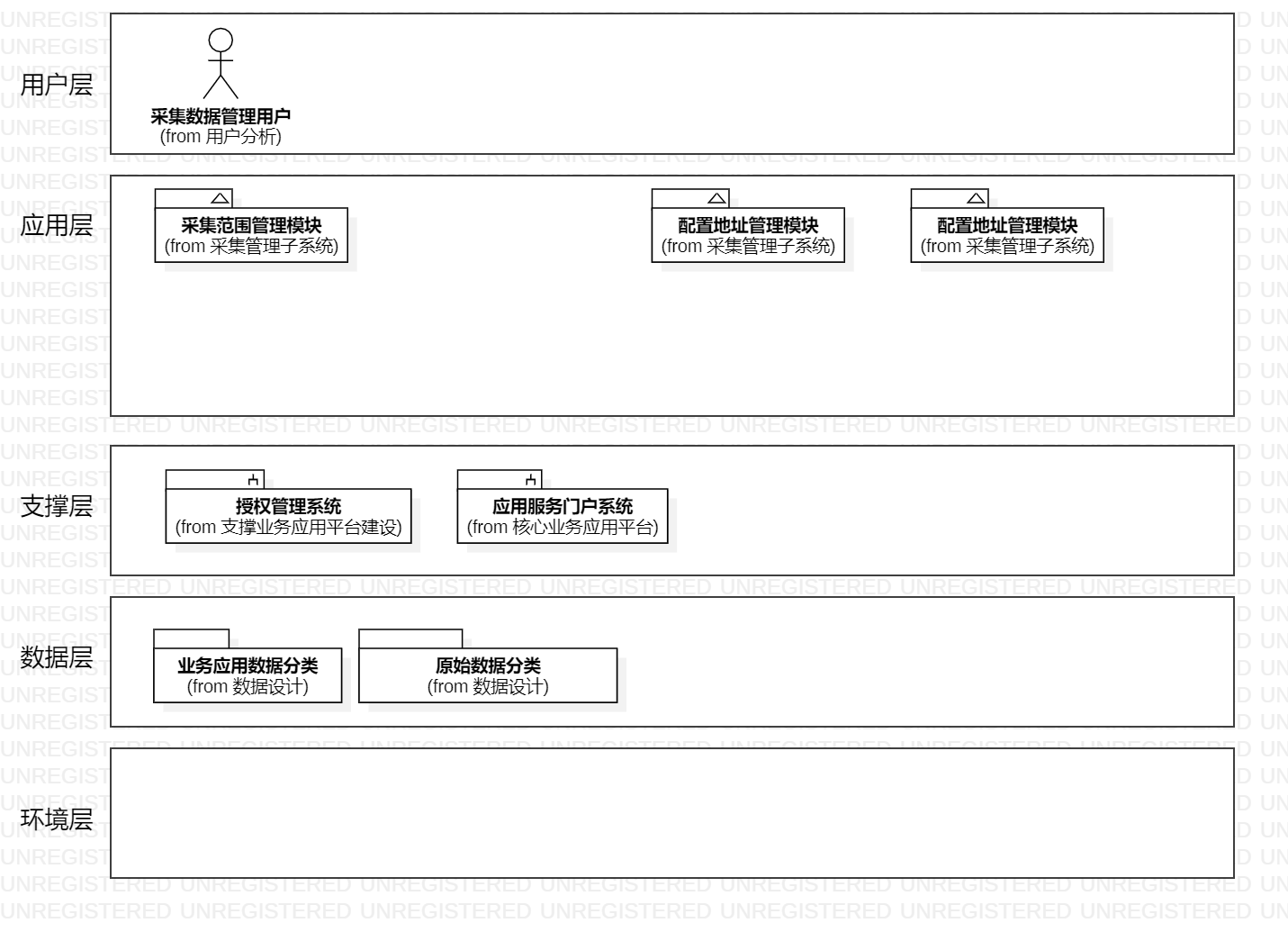
| **用户分类** | **系统角色** |
| --- | --- |
| 数据中心管理人员 | 管理员、运营人员 |

具体用户分析说明：

**1、管理员：**具备服务治理的全套指标，在服务查询界面管理所有的服务，具有查询服务提供者列表和服务状态的权限，同时可以监控各个主机的性能指标、代理池性能指标等。

**2、运营人员：**监控服务健康状态、资源占用、负载等信息，实现服务的灰度发布，水平伸缩，熔断隔离等情景。

###### 系统体系结构设计（没有写）



**图 1.1‑11 采集管理子系统-系统体系结构设计图**

**1、用户层**

主要是本平台主要服务用户，包含采集数据管理用户等。

**2、应用层**

应用层是在支撑层基础上构建的模块，主要为用户提供采集范围管理模块、配置地址管理模块、配置地址管理模块。

**3、支撑层**

支撑层在整个平台中承担着承上启下的关键作用，为业务应用系统提供技术支撑服务。支撑层由授权管理系统、应用服务门户系统等组成。

**4、数据层**

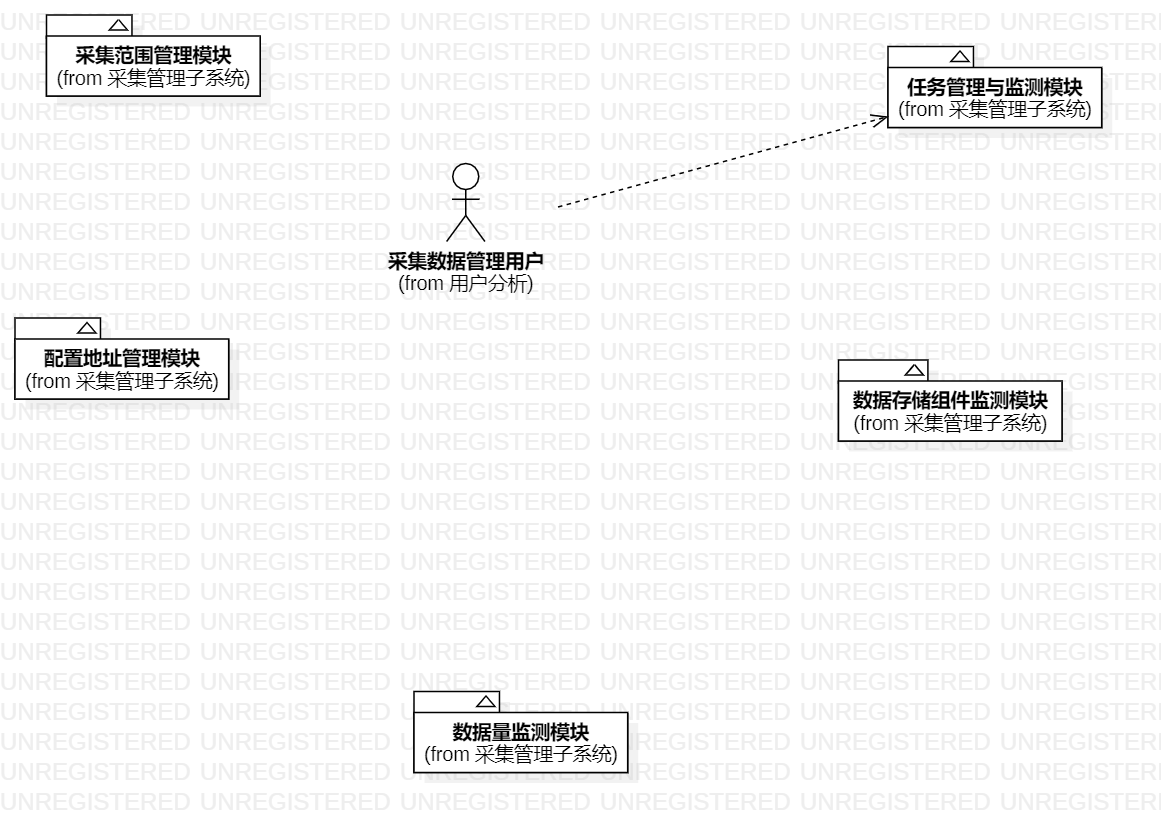
数据构建于环境层之上，并为上层的应用层、支撑层提供各种数据资源访问和存储服务，主要包括业务应用数据分类、原始数据分类等。

**5、基础设施层**

基础设施层为本项目的基础环境，基础设施层由基础网络、网络环境及主机、数据中心等部分共同组成。基础网络环境包含：互联网、网信专网；网络环境及主机包含：网络设备、安全设备、服务器、存储设备等；数据中心包含：IDC机房、核心数据机房等；

###### 系统总体设计（没有写）

用户与模块间关系设计



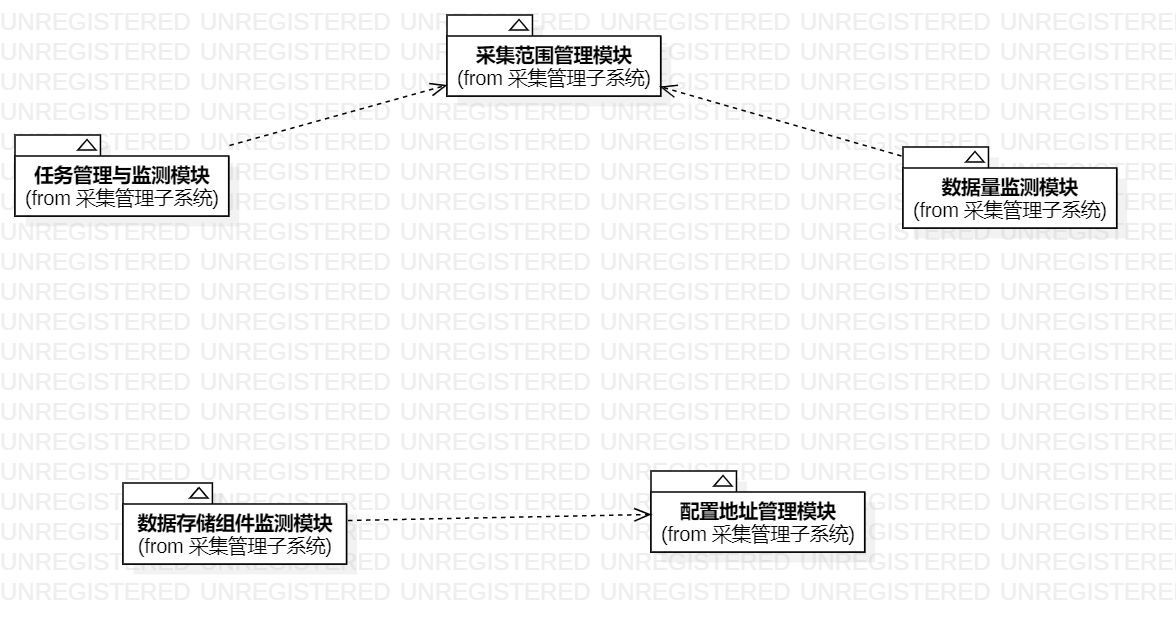
**图 1.1‑4采集管理子系统-用户与模块间关系设计图**

数据与模块间关系设计



**图 1.1‑4采集管理子系统-数据与模块间关系设计图**

各模块间关系设计



**图 1.1‑4采集管理子系统-各模块间关系设计图**

###### 链路跟踪模块

收集调用微服务的时序数据并集中展示方便度量调用链路的整体和局部性能，快速定位链路故障，展示微服务间拓扑关系，为后续链路 优化提供必要的数据支撑。

###### 灰度发布模块

支持按权重、用户特征等策略设定分 流规则，进行新老版本的验证测试 、支持查看监控指标、支持发布回滚、支持自定义规则灰度策。

###### 配置中心模块

统一管理各种应用配置的基础服务组件，能够集中管理微服务在不同环境的配 置信息重要配置项需要支持加密功能，比如数据库连接密码。

用户可以将分布式系统中用于不同模块的环境变量统一到一个对象中管理，利用环境变量和统一的配置文件来解决这些问题，能够实现统一管理各种应用配置的基础服务、集中管理微服务在不同环境的配置信息、重要配置项支持加密等功能。

###### 健康检查模块

支持主动和被动多种健康检查方式，如服务状态数据上报、log日志上报，主动监控数据采集 (队列长度、内存大小)。

###### 流量控制模块

支持多种流量控制策略，如请求量总并发，瞬时并发，时间窗口并发等，负载情况限流策略支持多种流控算法，如令牌桶，桶漏，计数器

###### 认证鉴权模块

微服务间服务调用的身份认证与访问鉴权，包括微服务与注册中心、配置中心之间的认证，以及微服务之间的认，token、session、cookies、（API级别认证和鉴权。

###### 熔断隔离模块

当服务的输入负载激增时，避免服务被迅速压垮导致雪崩效应，而对负载进行断路的一种方式支持多种熔断判断（熔断请求数，异常百分比）和熔断隔离方式：手动熔断、自动熔断、实例隔离。

###### 服务发现模块

支持服务节点在注册中心被增加或剔除后，能自动感知保证微服务节点数据的准确性和及时性。

###### 服务注册模块

服务在启动后自动提供自身的元数据以被请求方发现，在服务发生故障或不可用的情况下，自动对服务进行注册。

###### 服务容错模块

依赖的服务出现故障时候能够自动或者半自动的在应用层面实现平滑的失败处理，快速失败，失败重试，失败转移，并行调用，故障沉默，失败缓存。

###### 服务网关模块

服务网关是系统的唯一入口。API网关封装了系统内部架构，为每个客户端提供一个定制的API。除此之外，还具有身份验证、监控、负载均衡、缓存、请求分片与管理、静态响应处理等功能。

###### 接口设计

调用接口设计

本系统调用外部接口，详细如下所示：

**表 4.1‑146微服务治理中心-调用接口清单表**

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 接口名称 | 提供系统名称 | 描述 |
| 1 | 日志查询 | 日志中心 | 提供日志检索、日志聚合功能 |
| 2 | 监控查询 | 监控告警中心 | 提供性能监控、阈值告警功能 |

提供接口设计

本系统提供对外接口如下：

**表 4.1‑147 微服务治理中心-提供接口清单表**

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 接口名称 | 调用系统名称 | 描述 |
| 1 | 服务管理 | 中台底座 | 提供服务实例信息及管理功能 |
| 2 | 健康检查 | 中台底座 | 提供服务实例的运行/心跳信息 |
| 3 | 流量检查 | 中台底座 | 提供服务实例实时流量信息 |
| 4 | 配置管理 | 中台底座 | 提供配置管理能力 |

###### 功能与用户权限对应关系

**表4.1‑148微服务治理中心-功能与用户权限对应关系表**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 子系统名称 | 模块 | 使用用户 | 使用角色 | 备注 |
| 1 | 微服务  治理中心 | 链路跟踪模块 | 数据中心管理人员 | 运营人员 |  |
| 2 | 灰度发布模块 | 数据中心管  理人员 | 运营人员 |  |
| 3 | 配置中心模块 | 数据中心管  理人员 | 运营人员 |  |
| 4 | 健康检查模块 | 数据中心管  理人员 | 运营人员 |  |
| 5 | 流量控制模块 | 数据中心管  理人员 | 运营人员 |  |
| 6 | 认证鉴权模块 | 数据中心管  理人员 | 运营人员 |  |
| 7 | 熔断隔离模块 | 数据中心管  理人员 | 管运营人员 |  |
| 8 | 服务发现模块 | 数据中心管  理人员 | 运营人员 |  |
| 9 | 服务注册模块 | 数据中心管  理人员 | 运营人员 |  |
| 10 | 服务容错模块 | 数据中心管  理人员 | 运营人员 |  |

##### 监控预警中心

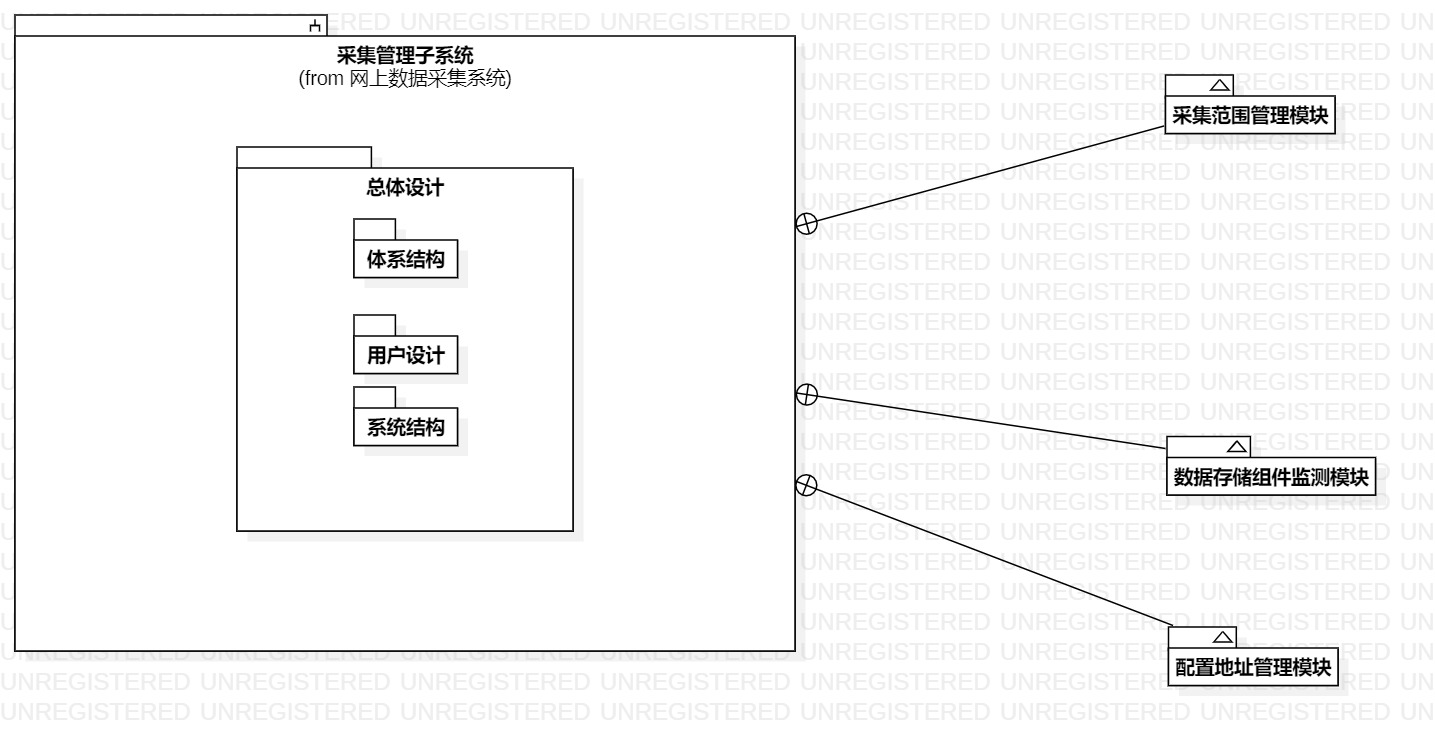
###### 系统概述

微服务治理中心虽然解决了目前诸多的架构层面的问题，但在分布式部署的环境中，必须要能够有效监控每一个服务，并及时发现系统中的问题，从而保证服务的安全。

在监控告警模块，通过分时点统计集群节点CPU、内存、 网络、磁盘等基础指标，获取容器和pod相关的性能指标数据,能准确反映微服务的实时及历史访问流量及健康状态，并且支持用户配置报警规则。可以实现快速定位问题，及时发现故障，在不影响应用处理能力的情况下尽可能的收集数据。

###### 系统组成

监控告警中心提供网站可用性检查、DNS解析、下载时间检查、内外链接存活检查等管理功能，可实时统计出当前服务中的各种性能指标，并可根据综合性能分析结果及管理策略，合理分配带宽资源，提高网站服务利用效率。在受管区通过Agent采集不同服务状态，监控平台会定期从Agent收集数据，并进行规则判定。如果满足告警规则的话，会实时往平台上发送告警。



**图1.1‑14采集管理子系统-系统组成图**

###### 用户与角色设计

用户设计

监控告警中心主要服务于管理用户，分为管理员和运营人员。



**图 1.1‑1采集管理子系统-用户设计图**

角色设计

系统主要使用角色情况如下表：

**表1.1‑2采集管理子系统-用户与角色设计表**

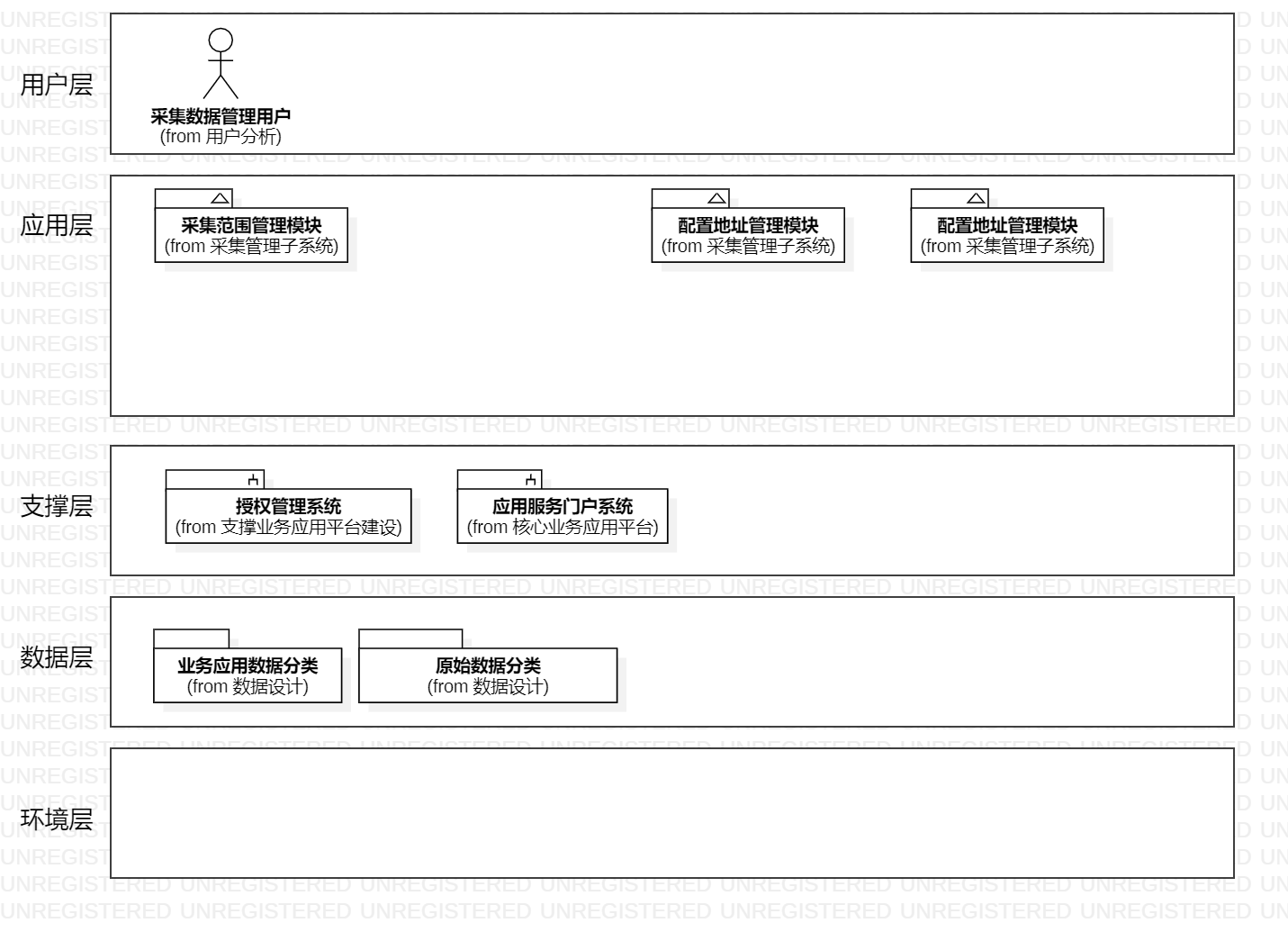
| **用户分类** | **系统角色** |
| --- | --- |
| 数据中心管理人员 | 管理员、运营人员 |

具体用户分析说明：

**1、管理员：**责管理性能监控和阈值告警模块，同时可以管理用户，监控各个主机的性能指标、代理池性能指标等。

**2、运营人员：**获取服务相关的性能指标数据，可以配置报警规则，快速定位问题，收集监控数据等。

###### 系统体系结构设计（没有写）



**图 1.1‑11 采集管理子系统-系统体系结构设计图**

**1、用户层**

主要是本平台主要服务用户，包含采集数据管理用户等。

**2、应用层**

应用层是在支撑层基础上构建的模块，主要为用户提供采集范围管理模块、配置地址管理模块、配置地址管理模块。

**3、支撑层**

支撑层在整个平台中承担着承上启下的关键作用，为业务应用系统提供技术支撑服务。支撑层由授权管理系统、应用服务门户系统等组成。

**4、数据层**

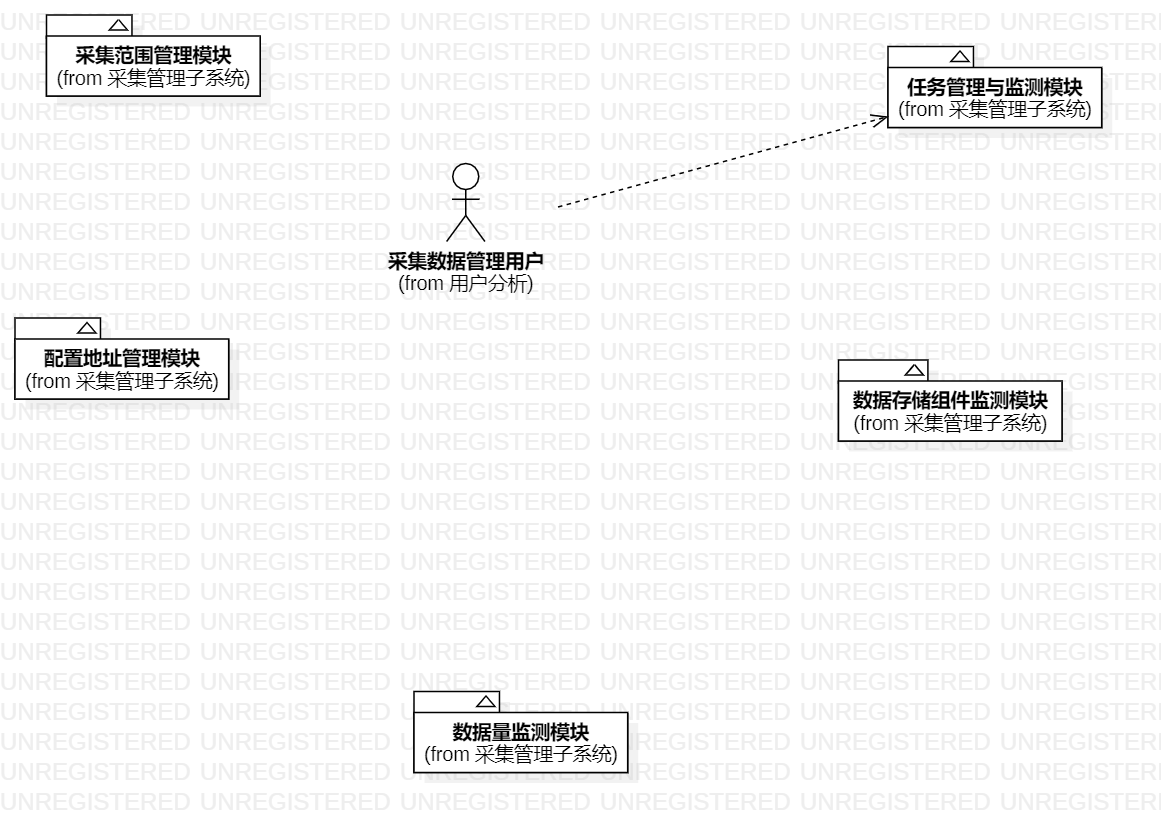
数据构建于环境层之上，并为上层的应用层、支撑层提供各种数据资源访问和存储服务，主要包括业务应用数据分类、原始数据分类等。

**5、基础设施层**

基础设施层为本项目的基础环境，基础设施层由基础网络、网络环境及主机、数据中心等部分共同组成。基础网络环境包含：互联网、网信专网；网络环境及主机包含：网络设备、安全设备、服务器、存储设备等；数据中心包含：IDC机房、核心数据机房等；

###### 系统总体设计（没有写）

用户与模块间关系设计



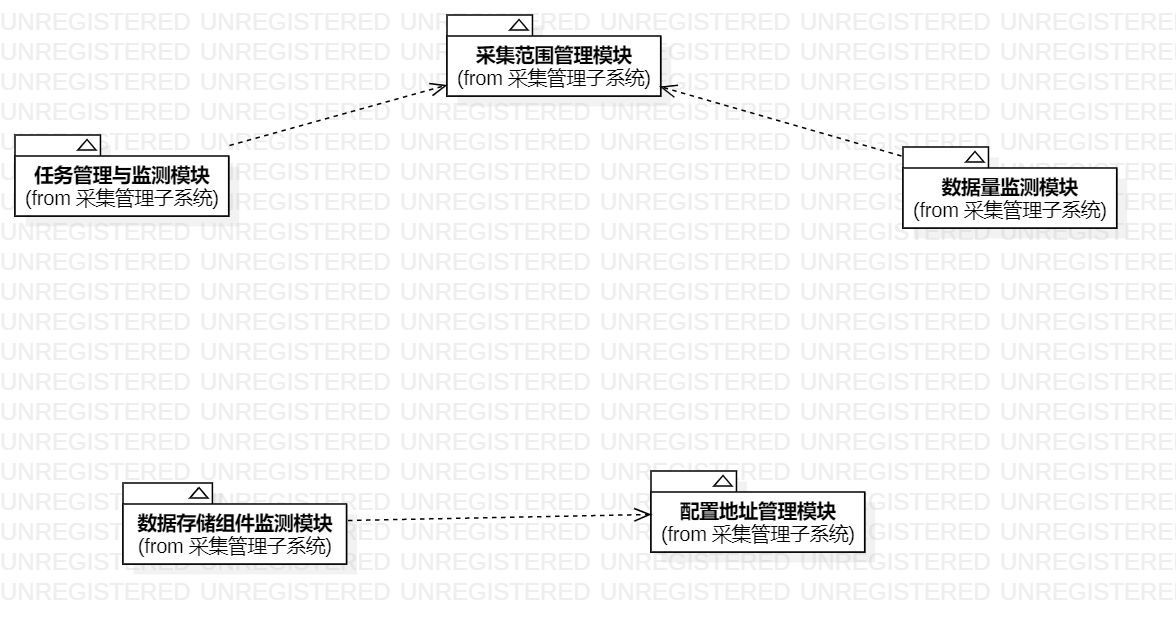
**图 1.1‑4采集管理子系统-用户与模块间关系设计图**

数据与模块间关系设计



**图 1.1‑4采集管理子系统-数据与模块间关系设计图**

各模块间关系设计



**图 1.1‑4采集管理子系统-各模块间关系设计图**

###### 性能监控模块

监控告警中心提供网站可用性检查、DNS解析、下载时间检查、内外链接存活检查等管理功能，可实时统计出当前服务中的各种性能指标，并可根据综合性能分析结果及管理策略，合理分配带宽资源，提高网站服务利用效率。在受管区通过Agent采集不同服务状态，监控平台会定期从Agent收集数据，并进行规则判定。如果满足告警规则的话，会实时往平台上发送告警。

###### 阈值告警模块

阈值告警模块可以对某些资源进行重点监控并在异常情况下及时响应时，在日常运维中对这些重点资源的指标创建阈值规则，可以第一时间发现异常并进行处理。在已设的连续周期内，当指标数据满足阈值条件时，系统会发送阈值告警；在已设的连续周期内，当没有指标数据上报时，系统会发送数据不足事件；当由于非工作时间或出差等原因导致不在现场，无法通过ServiceStage控制台即时查询阈值规则状态的变更信息时，可通过启用通知将该变更信息以短消息或邮件的方式自动发送给相关人员。

###### 接口设计

调用接口设计

本系统不调用外部接口。

提供接口设计

本系统为其他系统提供接口，详细如下所示：

**表 4.1‑150 监控告警中心-提供接口清单表**

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 接口名称 | 调用系统名称 | 描述 |
| 1 | 阈值告警 | 中台底座 | 为中台底座提供阈值告警功能 |
| 2 | 性能监控 | 中台底座 | 为中台底座提供性能监控功能 |
| 3 |  |  |  |

###### 功能与用户权限对应关系

**表4.1‑151监控告警中心-功能与用户权限对应关系表**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 子系统名称 | 模块 | 使用用户 | 使用角色 | 备注 |
| 1 | 采集范围管理子系统 | 阈值告警模块 | 数据中心  管理人员 | 运营人员 |  |
| 2 | 性能监控模块 | 数据中心  管理人员 | 运营人员 |  |

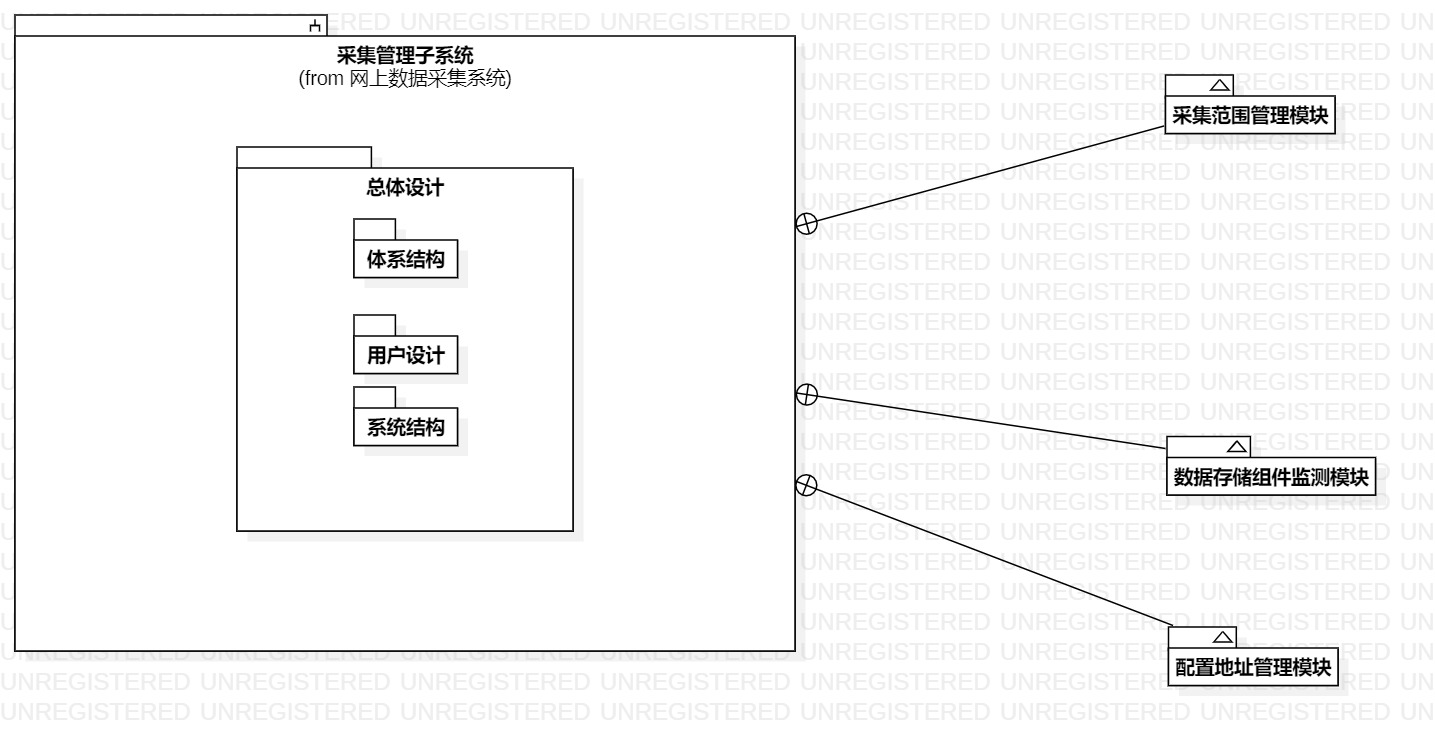
##### 日志中心

###### 系统概述

日志中心是保障高可靠服务的基础，不管是运维层面、业务层面、安全层面，还是系统监控、异常处理、安全、审计等都需要日志的配合协助。日志中心对日志进行实时采集、分析、展示等，日志内容涵盖基础设施层资源运行状态、调用链信息、以及微服务应用的运行日志等。整合分析日志信息，为平台链路跟踪、实时监控功能形成数据支撑等。

###### 系统组成

日志统计分析模块由日志聚合、日志检索两个模块组成，根据业务不同，将日志聚合，建立不同的索引并进行存储。



**图1.1‑14采集管理子系统-系统组成图**

###### 用户与角色设计

用户设计

日志中心主要服务于管理用户，分为管理员和运营人员。



**图 1.1‑1采集管理子系统-用户设计图**

角色设计

系统主要使用角色情况如下表：

**表1.1‑2采集管理子系统-用户与角色设计表**

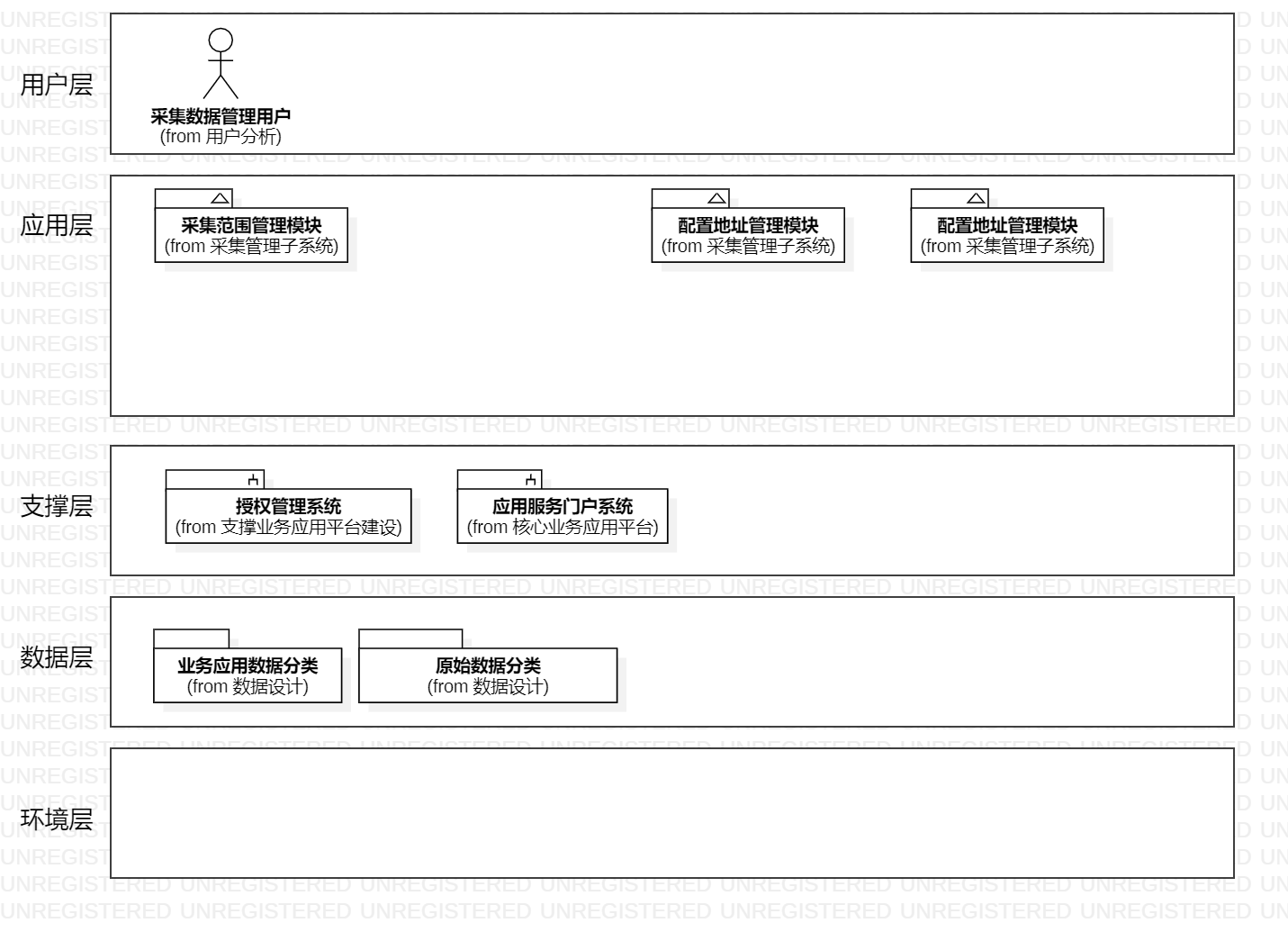
| **用户分类** | **系统角色** |
| --- | --- |
| 数据中心管理人员 | 管理员、运营人员 |

具体用户分析说明：

**1、管理员：**负责管理采集日志、日志存储、日志展示等，同时监控各个日志采集主机的性能指标、代理池性能指标等。

**2、运营人员：**使用日志搜索，对日志数据进行处理，通过日志展示查看服务运行情况。

###### 系统体系结构设计（没有写）



**图 1.1‑11 采集管理子系统-系统体系结构设计图**

**1、用户层**

主要是本平台主要服务用户，包含采集数据管理用户等。

**2、应用层**

应用层是在支撑层基础上构建的模块，主要为用户提供采集范围管理模块、配置地址管理模块、配置地址管理模块。

**3、支撑层**

支撑层在整个平台中承担着承上启下的关键作用，为业务应用系统提供技术支撑服务。支撑层由授权管理系统、应用服务门户系统等组成。

**4、数据层**

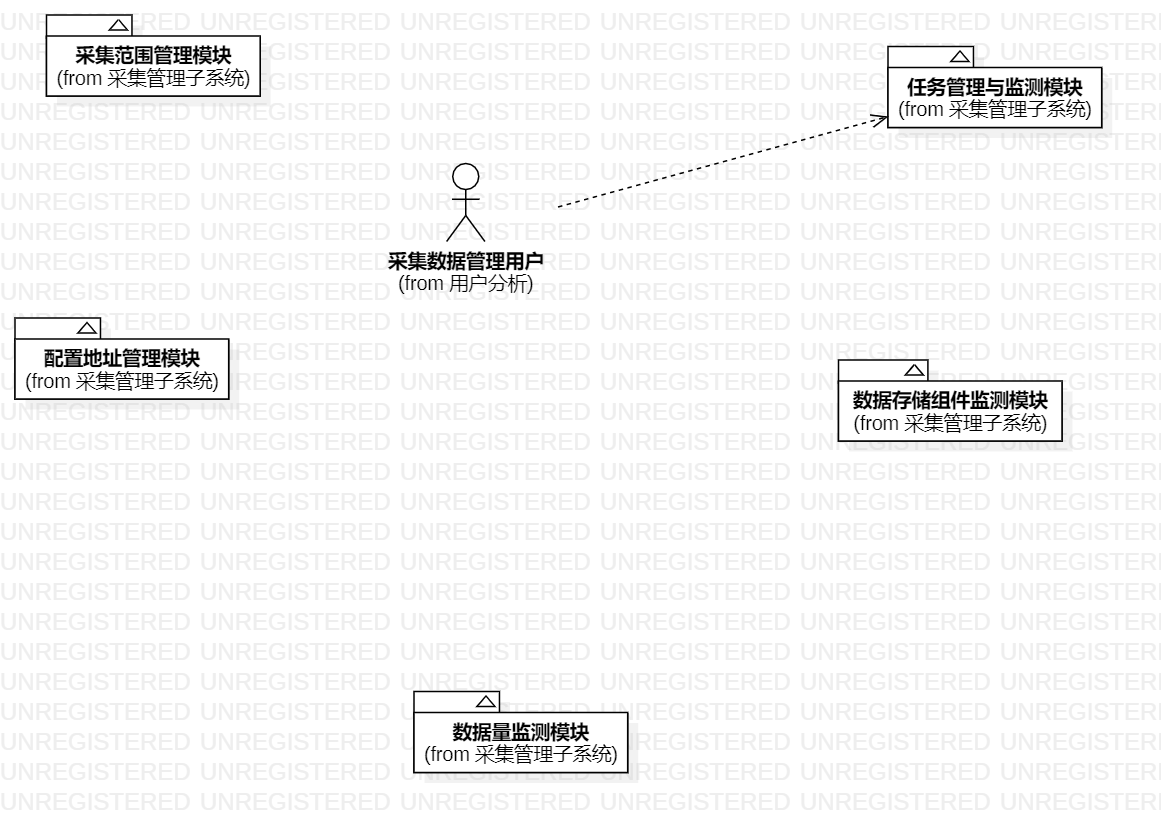
数据构建于环境层之上，并为上层的应用层、支撑层提供各种数据资源访问和存储服务，主要包括业务应用数据分类、原始数据分类等。

**5、基础设施层**

基础设施层为本项目的基础环境，基础设施层由基础网络、网络环境及主机、数据中心等部分共同组成。基础网络环境包含：互联网、网信专网；网络环境及主机包含：网络设备、安全设备、服务器、存储设备等；数据中心包含：IDC机房、核心数据机房等；

###### 系统总体设计（没有写）

用户与模块间关系设计



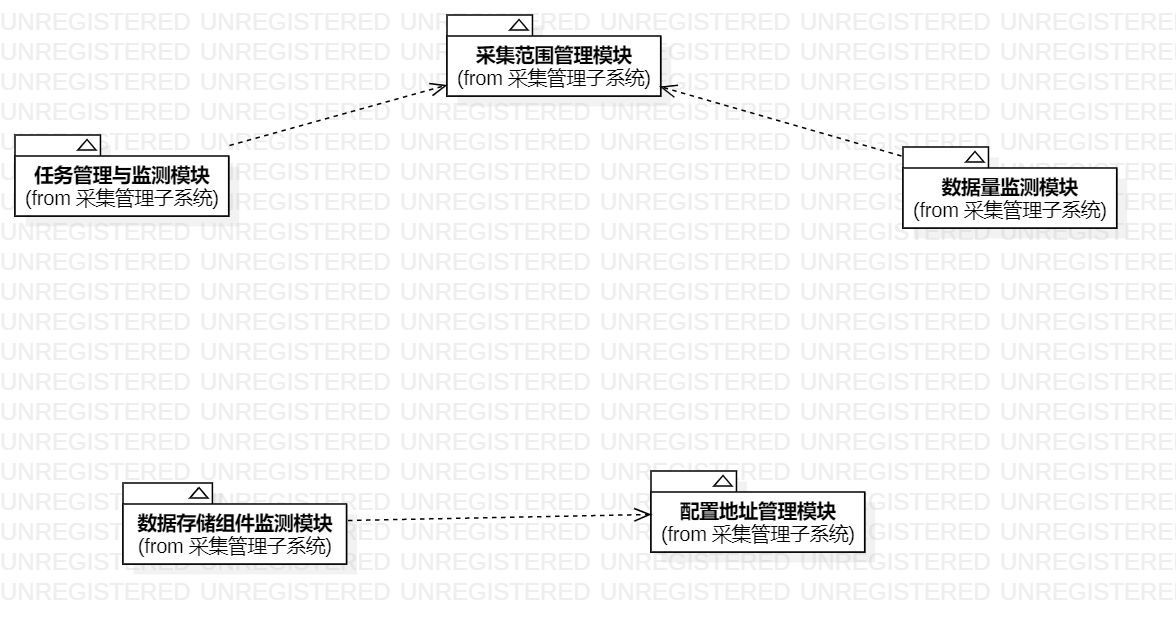
**图 1.1‑4采集管理子系统-用户与模块间关系设计图**

数据与模块间关系设计



**图 1.1‑4采集管理子系统-数据与模块间关系设计图**

各模块间关系设计



**图 1.1‑4采集管理子系统-各模块间关系设计图**

###### 链路跟踪模块

收集调用微服务的时序数据并集中展示方便度量调用链路的整体和局部性能，快速定位链路故障，展示微服务间拓扑关系，为后续链路 优化提供必要的数据支撑。

###### 日志聚合模块

根据业务实践，结合业务需求，日志聚合模块把来自不同服务器上不同应用程序产生的日志聚合起来，存放在单一的服务器，方便进行搜索和分析。包括跟踪日志、系统日志、部署日志、引擎日志、构件包日志等，通过对日志分类，在分析问题时明确要查找的位置，缩小查找范围提高效率。

###### 日志检索模块

日志检索模块可以处理海量数据，应对多种需求，并且拥有强大的搜索功能，通过系统、应用、应用实例分组、应用实例IP、关键字、日志级别、时间区间来检索所需要的日志信息。将收集到的日志汇总、分析，结合自定义检索内容生成图表，进行日志数据展示等。

###### 接口设计

调用接口设计

本系统不调用外部接口

提供接口设计

本系统为其他系统提供接口，详细如下所示：

**表 4.1‑153 日志中心-提供接口清单表**

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 接口名称 | 调用系统名称 | 描述 |
| 1 | 日志聚合 | 中台底座 | 为中台底座提供日志聚合功能 |
| 2 | 日志检索 | 中台底座 | 为中台底座提供日志搜索功能 |
| 3 |  |  |  |

###### 功能与用户权限对应关系

**表4.1‑154日志中心-功能与用户权限对应关系表**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 子系统名称 | 模块 | 使用用户 | 使用角色 | 备注 |
| 1、 | 日志中心 | 日志聚合模块 | 数据中心  管理人员 | 运营人员 |  |
| 2 | 日志搜索模块 | 数据中心  管理人员 | 运营人员 |  |

##### 接口设计

###### 调用接口设计

本系统不调用外部接口。

###### 提供接口设计

本系统不为其他系统提供接口。

##### 用户权限设计

**表4.1‑155中台底座-用户权限设计表**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 系统名称 | 子系统名称 | 使用用户 | 使用角色 | 备注 |
| 1 | 中台底座 | 微服务治理中心 | 数据中心  管理人员 | 运营人员 |  |
| 2 | 监控告警中心 | 数据中心  管理人员 | 运营人员 |  |
| 3 | 日志中心 | 数据中心  管理人员 | 运营人员 |  |
| 4 |  |  |  |  |

# 标题模板

## 二级

### 三级

#### 四级

##### 五级

###### 六级

七级

八级

九级

