# 本期项目建设方案

## 应用支撑平台及应用系统建设

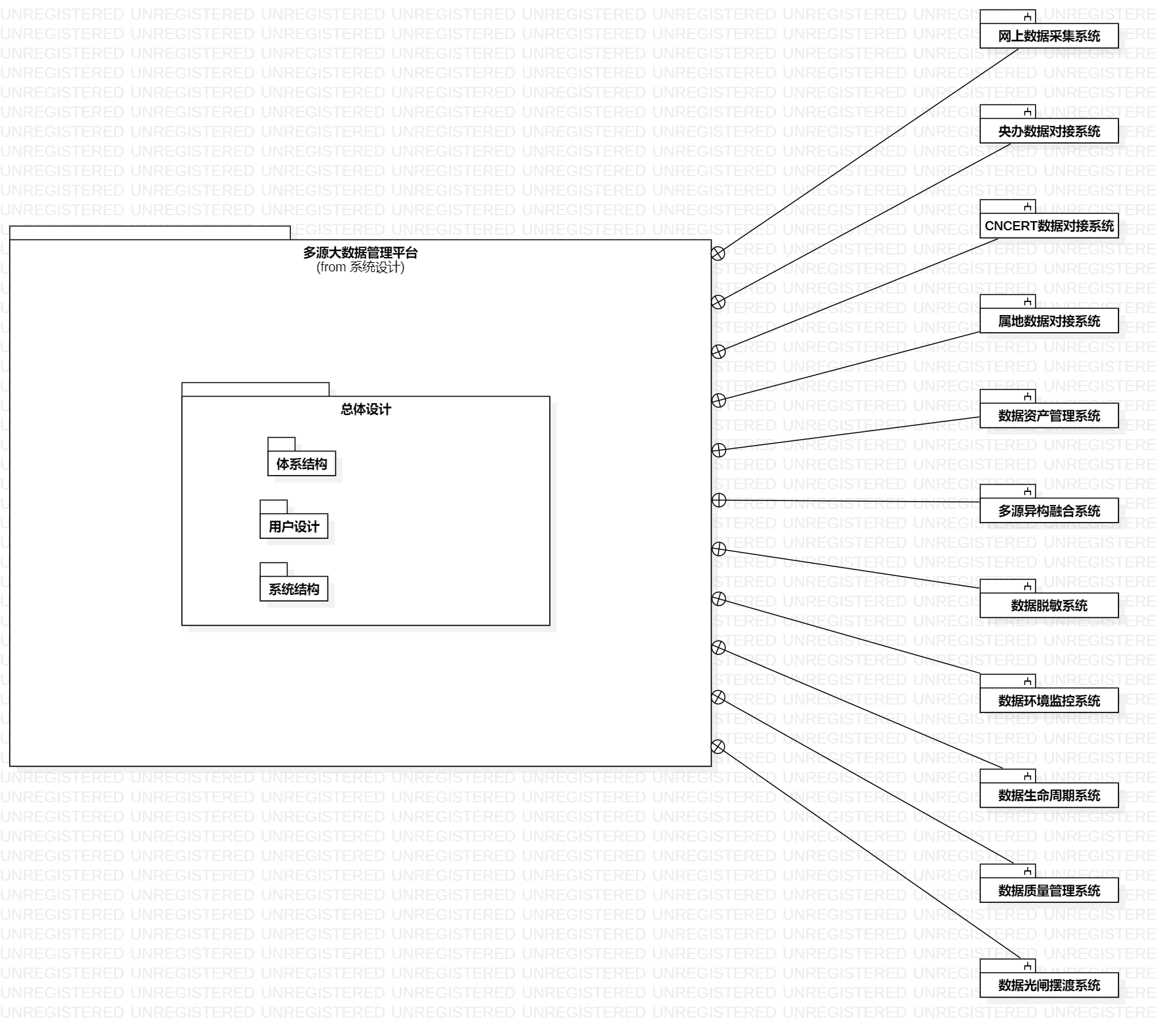
### 多源大数据管理平台

#### 平台定位

建设多源大数据管理平台，实现对央办、属地、CNCERT、互联网公共平台等涉粤涉港澳台多源异构互联网内容数据进行采集、共享交换、计算融合，实现数据“存、通、用”全生命周期管理。建设范围包括网上数据采集系统、属地数据对接系统、数据光闸摆渡系统、央办数据对接系统、多源异构融合系统、数据换南京监控系统、数据生命周期管理系统、数据质量管理系统、数据安全管理系统、数据脱敏系统、数据资产管控系统、多源搜索引擎系统、CNCERT数据接入系统等。

#### 平台组成

多源大数据管理平台由网上数据采集系统、央办数据接入系统、属地数据接入系统、CNCERT数据接入系统、数据光闸摆渡系统、数据多源异构融合系统、数据环境监测系统、数据生命周期管理系统、数据质量管理系统、数据安全管理系统、数据脱敏系统、数据资产管控系统等12个系统组成。



**图 1.1‑1多源大户据管理平台-平台组成图**

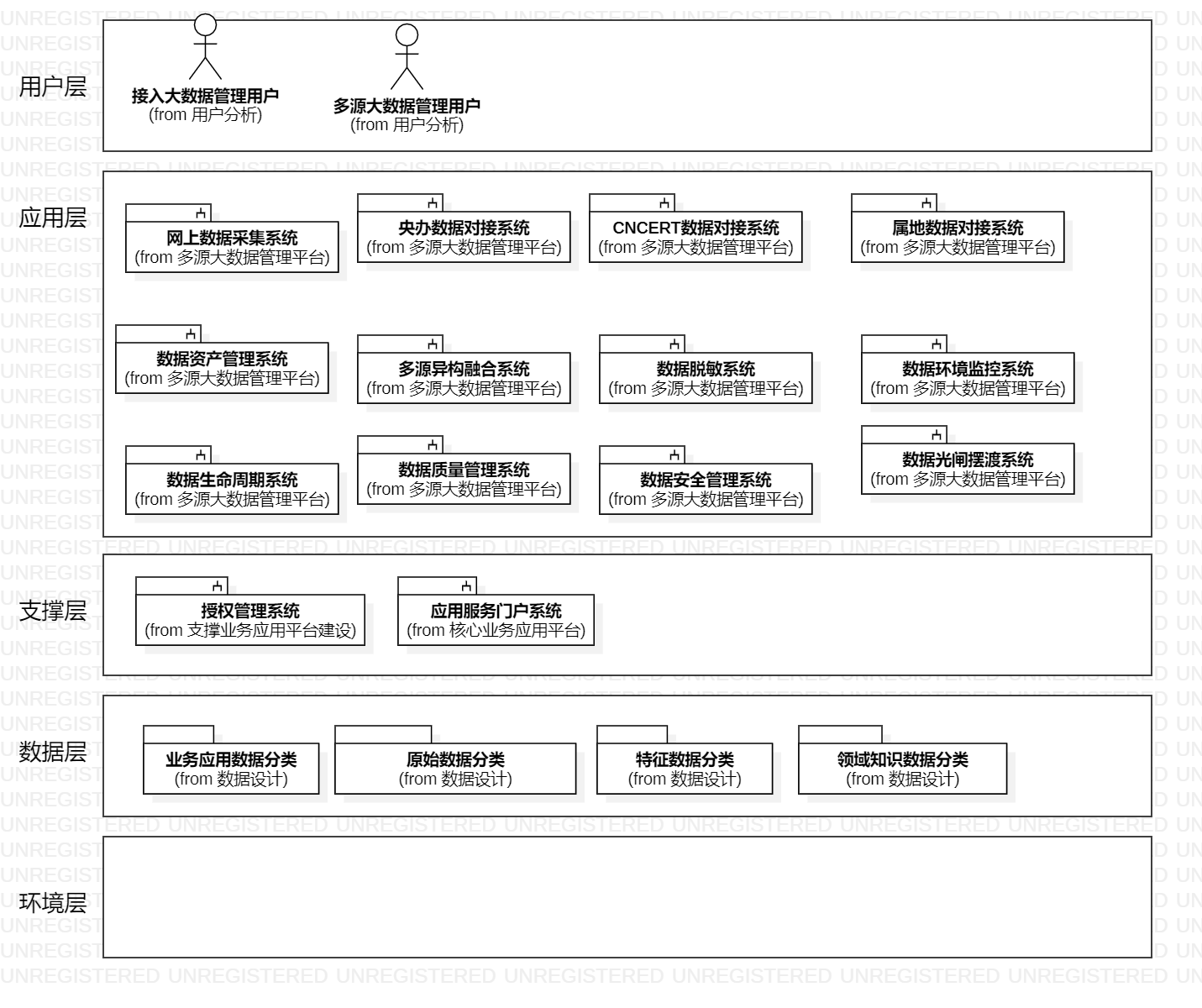
#### 用户设计

多源大数据管理平台主要服务于技术管理用户，分为接入大数据管理用户、多源大数据管理用户。

**图 1.1‑2多源大户据管理平台-用户设计图**

#### 平台体系结构设计

多源大数据管理平台体系结构设计分为5层，含用户层、应用层、支撑层、数据层、环境层。总体体系结构如下：



**图 1.1‑3多源大户据管理平台-平台平台体系结构设计**

**1、用户层**

主要是本平台主要服务用户，包含接入大数据管理用户、多源大数据管理用户等。

**2、应用层**

应用层是在支撑层基础上构建的系统，主要为用户提供网上数据采集系统、央办数据接入系统、属地数据接入系统、CNCERT数据接入系统、数据光闸摆渡系统、数据多源异构融合系统、数据环境监测系统、数据生命周期管理系统、数据质量管理系统、数据安全管理系统、数据脱敏系统、数据资产管控系统。

**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、特征数据分类、领域知识数据分类等。

**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；

#### 平台总体设计

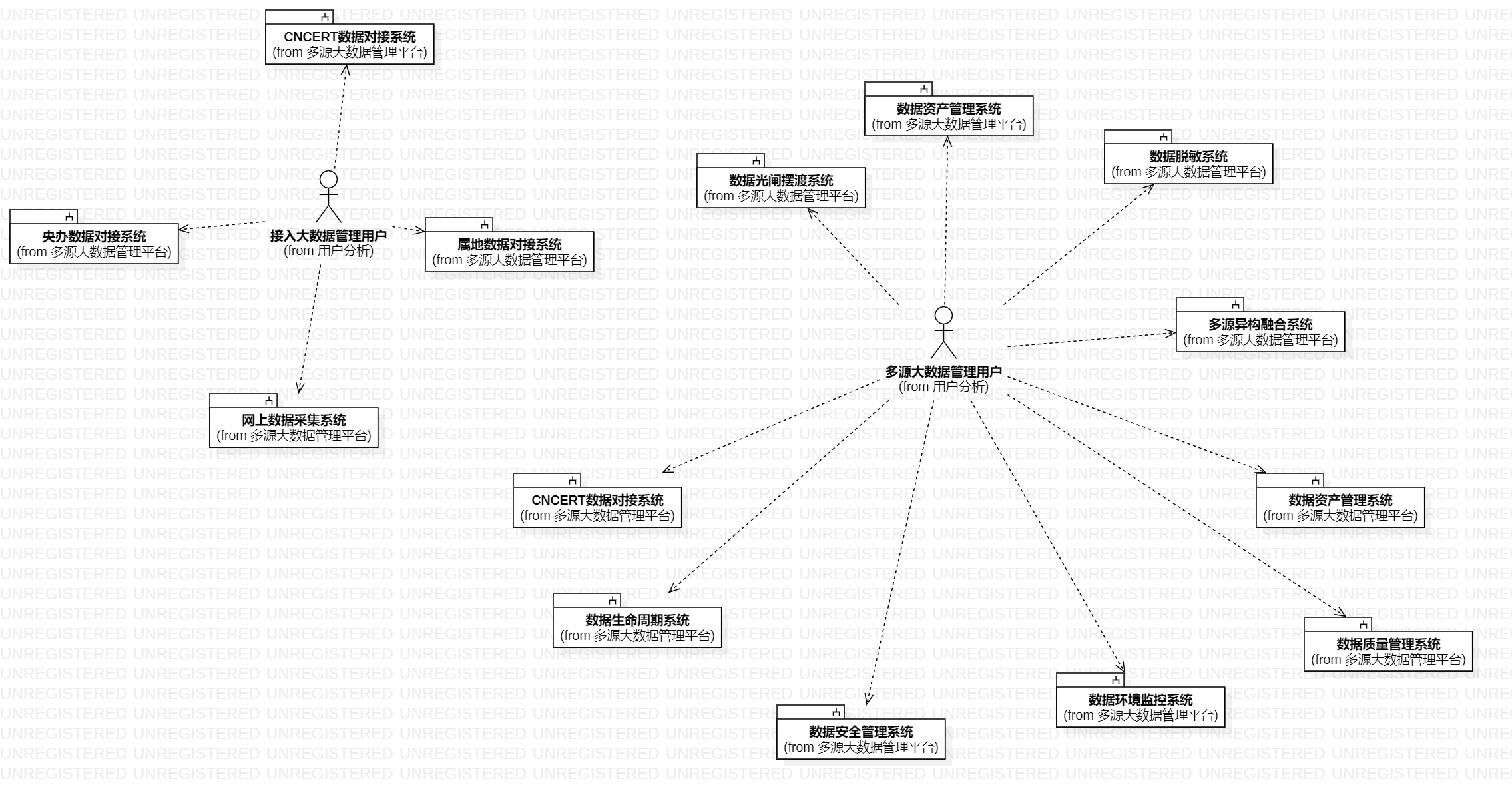
##### 用户与系统间关系设计

多源大数据管理平台使用用户包括接入大数据管理用户、多源大数据管理用户。用户与系统间关系如下：

1. 接入大数据管理用户

接入大数据管理用户使用系统包括中央数据对接系统、网上数据采集系统、网络数据采集系统、属地数据对接系统。

1. 描述原则：什么用户，使用什么系统，实现什么业务需求。
2. 接入大数据管理用户，使用中央数据对接系统，实现中央数据对接规则维护、数据交换质量验证、监控整个数据交换过程；
3. 接入大数据管理用户，使用网上数据采集系统，实现采集范围维护。。。



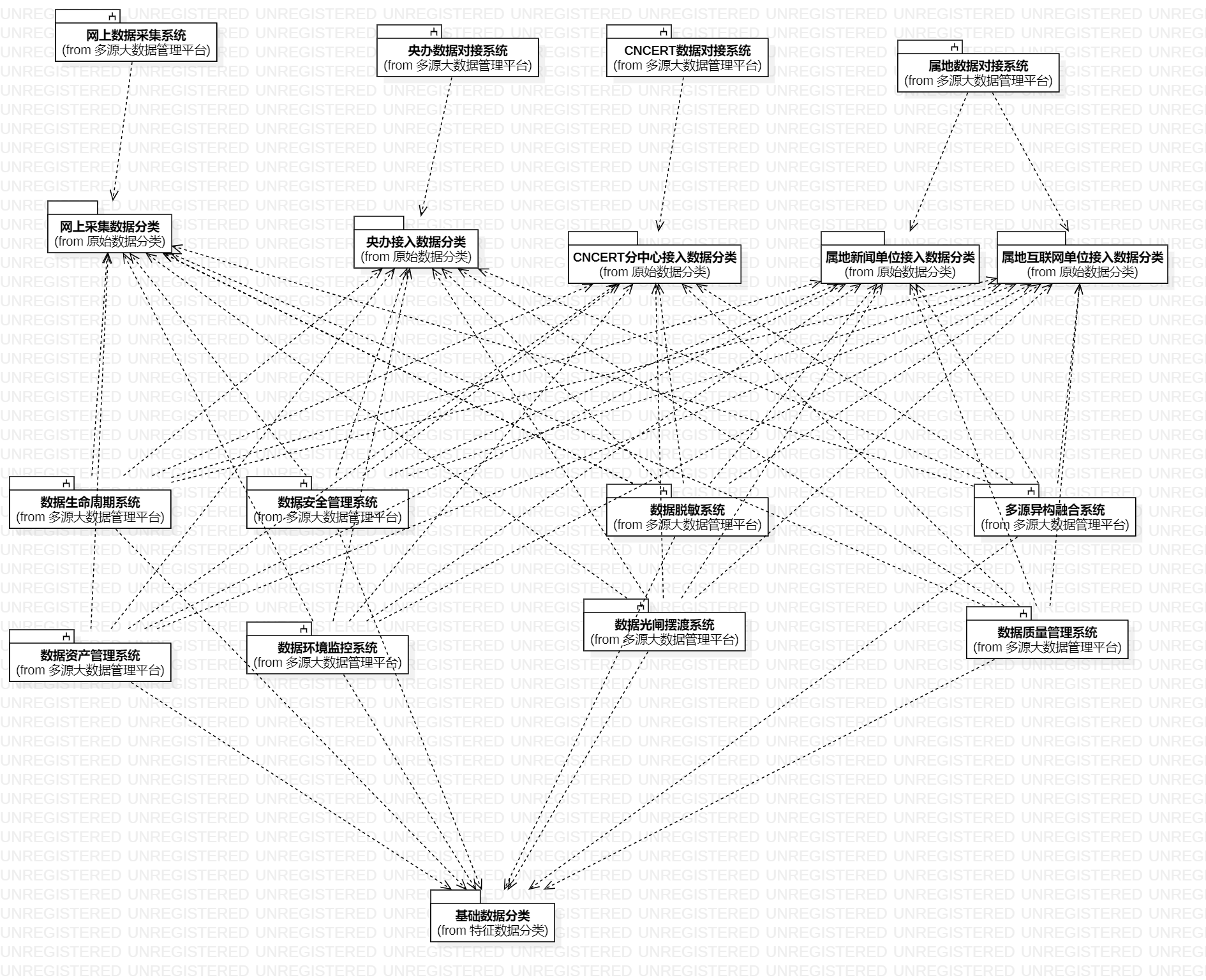
**图 1.1‑4多源大户据管理平台-用户与系统间关系设计图**

##### 数据与系统间关系设计

多源大数据管理平台包括网上采集数据分类、央办数据接入分类、CNCERT分中心接入数据分类、属地新闻单位接入数据分类、属地互联网单位数据接入分类、基础数据分类。数据与系统间关系如下：

什么系统维护什么数据，实现什么操作；

1. 网上采集系统，维护网上数据采集，实现将采集数据实时存入。。。。采集数据；
2. 数据生命周期系统，负责维护数据生命周期，负责将网上采集数据、央办接入数据、。。。。的生命周期进行管理；维护安全，负责。。。。。。。



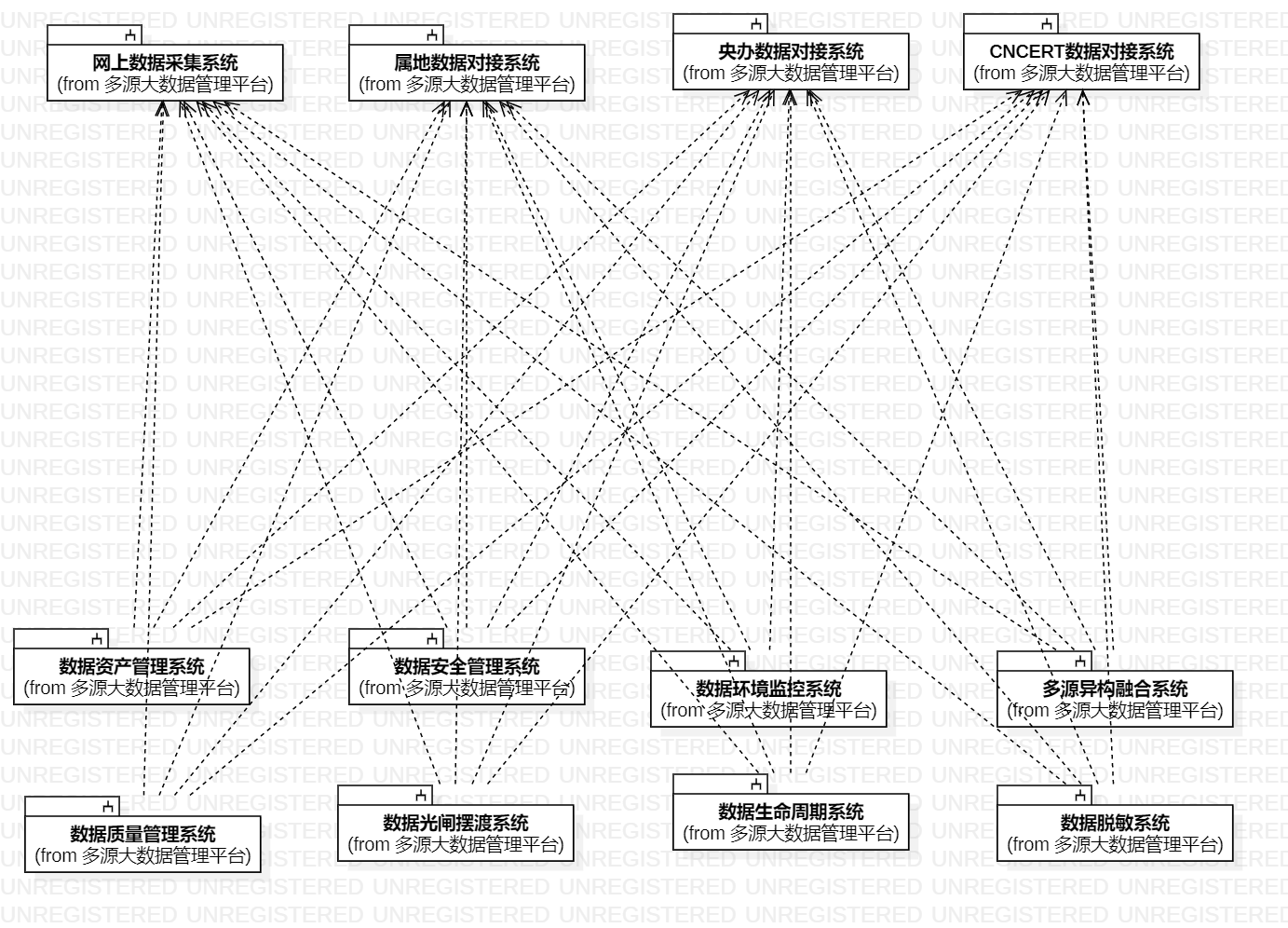
**图 1.1‑5多源大户据管理平台-数据与系统间关系设计图**

##### 各系统间关系设计

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**

#### 网上数据采集系统

##### 系统定位

网上数据采集系统主要面向用户为数据中心管理人员数据管理员、系统（工具）管理员和业务应用员。利用多种互联网采集技术，结合隐藏身份技术和手机群控技术，通过结构化解析识别实现对海量互联网数据的采集和持久化存储。为上层应用业务提供海量基础数据支撑。

##### 技术路线

**1、采用基于J2EE的B/S架构设计**

系统从应用性、安全性、可扩展性与可维护性的角度考虑，采用了基于J2EE的轻量级B/S架构体系。

J2EE体系结构提供中间层集成框架用来满足无需太多费用而又需要高可用性、高可靠性以及可扩展性的应用的需求。通过提供统一的开发平台，J2EE降低了开发多层应用的费用和复杂性，同时提供对现有应用程序集成强有力支持，完全支持Enterprise JavaBeans，有良好的向导支持打包和部署应用，添加目录支持，增强了安全机制，提高了性能。

B/S结构是一种以Web技术为基础的新型系统平台模式。B/S架构的主要工作原理是，操作时用户通过客户端浏览器向WEB服务器发出请求信息，WEB服务器再向数据库服务器请求，数据库服务器完成WEB服务器的请求后，向WEB服务器发出回复信息。然后WEB服务器将数据库服务器提供的信息传回浏览器，这样用户可以在浏览器看到自己请求的信息。

**2、采用vue.js框架**

Vue.js是一套构建用户界面的渐进式框架。与其他重量级框架不同的是，Vue 采用自底向上增量开发的设计。Vue 的核心库只关注视图层，容易与其它库或已有项目整合。另一方面，Vue 完全有能力驱动采用单文件组件和Vue生态系统支持的库开发的复杂单页应用。

**3、采用Nginx负载技术**

Nginx是一个高性能的HTTP和反向代理服务器，也是一个IMAP/POP3/SMTP服务器。Nginx是一款轻量级的Web 服务器/反向代理服务器及电子邮件（IMAP/POP3）代理服务器，并在一个BSD-like 协议下发行。其特点是占有内存少，并发能力强，事实上nginx的并发能力确实在同类型的网页服务器中表现较好，中国大陆使用nginx网站用户有：百度、京东、新浪、网易、腾讯、淘宝等。在连接高并发的情况下，Nginx能够支持高达 50,000 个并发连接数的响应。Nginx代码完全用C语言从头写成，已经移植到许多体系结构和操作系统，包括：Linux、FreeBSD、Solaris、Mac OS X、AIX以及Microsoft Windows。Nginx是安装简单、配置文件简洁、Bug少的服务器。Nginx 启动方便，7\*24不间断运行，可以在不间断服务的情况下进行软件版本的升级。

**4、采用分布式数据存储与计算技术**

基于Hadoop项目进行系统集成和开发，用于采集的网络数据的存储，并对采集数据进行分析和挖掘。用户可以在不了解分布式底层细节的情况下，开发分布式程序。充分利用集群的威力进行高速运算和存储。Hadoop实现了一个分布式文件系统（Hadoop Distributed File System），简称HDFS。HDFS有高容错性的特点，并且设计用来部署在低廉的（low-cost）硬件上；而且它提供高吞吐量（high throughput）来访问应用程序的数据，适合那些有着超大数据集（large data set）的应用程序。HDFS放宽了（relax）POSIX的要求，可以以流的形式访问（streaming access）文件系统中的数据。Hadoop框架最核心的设计就是：HDFS和MapReduce。HDFS为海量的数据提供了存储，而MapReduce则为海量的数据提供了计算。

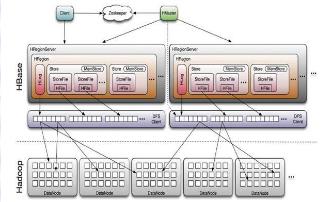
基于FastDFS项目实现图片的存储和显示。FastDFS是一个高可扩展、高可用、高性能、面向互联网服务的分布式文件系统，主要针对海量的非结构化数据，它构筑在普通的Linux机器集群上，可为外部提供高可靠和高并发的存储访问。它采用了HA架构和平滑扩容，保证了整个文件系统的可用性和扩展性。同时扁平化的数据组织结构，可将文件名映射到文件的物理地址，简化了文件的访问流程，一定程度上为FastDFS提供了良好的读写性能。

Apache Hadoop YARN （Yet Another Resource Negotiator，另一种资源协调者）是一种新的 Hadoop 资源管理器，它是一个通用资源管理系统，可为上层应用提供统一的资源管理和调度，它的引入为集群在利用率、资源统一管理和数据共享等方面带来了巨大好处。

HBase是一个分布式的、面向列的开源数据库。该技术来源于FayChang所撰写的Google论文“Bigtable：一个结构化数据的分布式存储系统”。就像Bigtable利用了Google文件系统（FileSystem）所提供的分布式数据存储一样，HBase在Hadoop之上提供了类似于Bigtable的能力。HBase是Apache的Hadoop项目的子项目。HBase不同于一般的关系数据库，它是一个适合于非结构化数据存储的数据库。另一个不同的是HBase基于列的而不是基于行的模式。HBase作为分布式数据存储优势在于：

1、采用独立的zookeeper，分布式应用程序协调服务，协调hbase集群各个节点，稳定有序工作。

2、数据多次冗余提高hbase集群读写速度，保证数据完整性，防止数据丢失。



**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。

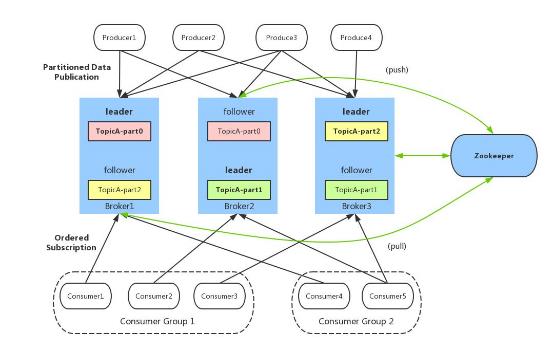
**6、采用分布式并行计算技术**

基于Spark项目进行系统集成和开发Spark 是一种与 Hadoop 相似的集群计算环境，但是两者之间还存在一些不同之处，这些有用的不同之处使 Spark 在某些工作负载方面表现得更加优越，换句话说，Spark 启用了内存分布数据集，除了能够提供交互式查询外，它还可以优化迭代工作负载。Spark 是在 Scala 语言中实现的，它将 Scala 用作其应用程序框架。与 Hadoop 不同，Spark 和 Scala 能够紧密集成，其中的 Scala 可以像操作本地集合对象一样轻松地操作分布式数据集。

**7、采用分布式消息中间件技术**

Kafka是一种高吞吐量的分布式发布订阅消息系统，它可以处理消费者规模的网站中的所有动作流数据。这种动作（网页浏览，搜索和其他用户的行动）是在现代网络上的许多社会功能的一个关键因素。这些数据通常是由于吞吐量的要求而通过处理日志和日志聚合来解决。对于像Hadoop的一样的日志数据和离线分析系统，但又要求实时处理的限制，这是一个可行的解决方案。Kafka的目的是通过Hadoop的并行加载机制来统一线上和离线的消息处理，也是为了通过集群来提供实时的消费。Kafka作为分布式消息中间件技术选型，具备以下优势：

1. 通过O(1)的磁盘数据结构提供消息的持久化，这种结构对于即使数以TB的消息存储也能够保持长时间的稳定性能。
2. 高吞吐量：即使是非常普通的硬件Kafka也可以支持每秒数百万的消息。
3. 支持通过Kafka服务器和消费机集群来分区消息。
4. 支持Hadoop并行数据加载。



**8、采用离线计算引擎技术**

离线计算引擎通过对映射规约规约编程框架进行封装，为上层应用提供更易使用的类SQL数据分析高级语言，实现对海量数据的高效离线分析处理。

离线计算框架包括查询引擎、计算引擎和存储引擎，支持过滤、投影、关联、聚集等多种操作算子。

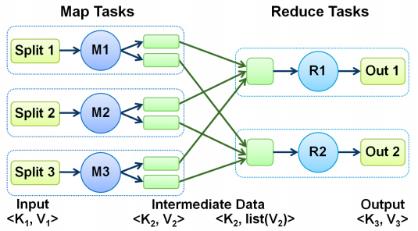
基于映射规约规约的离线计算框架如下图所示，主要分为三个层次：查询引擎、计算引擎和存储引擎。

查询引擎负责对外提供编程API，当用户提交查询语句时，查询引擎对该查询语句进行词法分析、语法分析、语义分析、查询优化、生成可执行的计划，最终转化为一个或多个作业，交由计算引擎执行；

计算引擎负责对海量数据提供并行计算服务，调用存储引擎API来读取数据，并将最终作业执行结果返回给用户；

存储引擎则负责对海量数据可靠存储，并提供统一的访问接口和高效的文件访问服务，是支撑上层应用的基础。

映射规约规约处理海量数据主要包括两阶段，映射阶段和规约规约阶段。具体流程如图所示：



映射阶段的主要操作包括：首先，读取分片数据，并将它们转换为键值对的形式，然后，用户自定义的映射函数作用在该数据分片包含的键值对上，生成中间结果并输出到内存缓冲区中，当缓冲区满的时候则将中间结果写入本地磁盘，这些文件最终会合并成一个排序文件；若映射任务输出的中间结果包含大量重复值，则执行联合操作，对相同key进行归并，减少后续规约规约阶段的输入数据量，该阶段是可选的。

规约规约阶段的主要操作包括：首先，按照指定的键值对映射产生中间结果进行重划分，默认采用哈希划分；然后，按照key值进行排序，使得具有相同key值的数据聚合在一起，并根据用户自定义的规约函数执行相应的数据处理操作；最后，将结果输出到分布式文件系统中。在一个映射规约作业中可能并不包含上述所有阶段，可能包括其中几个阶段，只有映射阶段是必须存在的。

**9、采用网络爬虫技术**

该系统基于网络爬虫技术，基于微服务架构风格设计，以成熟的开源框架为核心分布式部署，实现分布式爬虫，以针对不同采集源使用不同的采集方式（深度优先，广度优先）完成互联网数据采集以及分析存储过程。网络采集过程主要包含网页下载器，网页解析器，分类下载器以及去重过滤器。基于领域划分，对采集过程进行服务拆解，这样增强了服务的复用性，同时便于监控与管理。此外爬虫技术还依赖代理建设，基于反爬虫技术的升级，多次爬取需要利用特殊的反反爬虫技术。其代理技术是每个爬虫的必备手段，利用第三方模块，可再互联网上获取代理ip。增加数据覆盖范围，覆盖区域，增快数据获取速度。爬虫过程依赖数据的高并发处理能力。多线程，多进程，协程技术。多线程，多进程以及协程技术，可以最大化利用服务器资源，如cpu,内存等，实现多并发，多并行，多分布式的部署爬虫，以达到更快速准确的获取目标源数据。

**10、采用网页搜索算法技术**

网络爬虫需要对搜索目标的主题进行描述，主题描述方式可以分为基于目标网页特征，基于目标数据模式以及基于领域概念三种。

基于目标网页特征的描述方式中，采集的搜索对象通常是网页，网页特征可以是网页的内容特征，或为网页的链接结构特征等。种子样本获取方式可以分为以下三种：

1. 预先级定的初始种子样本
2. 预先给定的网页分类目录以及分类目录对应的种子样本
3. 通过用户行为确定的搜索目标样例，通过日志挖掘得到访问模式以及相关样本

基于目标数据模式的描述方式中，采集的搜索对象是网页上的数据，网页数据符合一定的数据模式或能够映射目标数据的模式。

在基于领域概念的描述方式中，通过建立目标领域的词典，从语义角度分析不同特征在目标主题的重要程度。

**11、采用模拟浏览器控制技术**

针对反爬机制较为完善的网站，需要用过的浏览器模拟技术，如 Mozilla Firefox，Safari，Google Chrome，Opera等浏览器，进行模型行为操作，进行数据获取。

Selenium 测试直接在浏览器中运行，就像真实用户所做的一样。Selenium 测试可以在 Windows、Linux 和 Macintosh上的 Internet Explorer、Chrome和 Firefox 中运行。其他测试工具都不能覆盖如此多的平台。使用 Selenium 和在浏览器中运行测试还有很多其他好处。

下面是主要的两大好处：

通过编写模仿用户操作的 Selenium 测试脚本，可以从终端用户的角度来测试应用程序。通过在不同浏览器中运行测试，更容易发现浏览器的不兼容性。Selenium 的核心，也称browser bot，是用 JavaScript 编写的。这使得测试脚本可以在受支持的浏览器中运行。browser bot 负责执行从测试脚本接收到的命令，测试脚本要么是用 HTML 的表布局编写的，要么是使用一种受支持的编程语言编写的。

**12、采用手机群控与协议分析技术**

使用群控技术操作手机，进行拟人的操作，并且该系统可以自动重启，清理手机缓存，内存等。利用抓包软件获取，机APP软件操作时产生的数据包。方便采集程序对数据包解析，获取目标数据。

通过向网络上传和从网络下载一些数据包来实现数据在网络中的传播。通常这些数据包会由发出或者接受的软件自行处理，普通用户并不过问，这些数据包一般也不会一直保存在用户的计算机上。抓包工具可以帮助我们将这些数据包保存下来，如果这些数据包是以明文形式进行传送或者明确其加密方法，、可以分析出这些数据包的内容以及它们的用途。

**13、采用进线程及协程调度技术**

linux中的进程主要有三种调度策略：

优先级调度：将进程分为普通进程和实时进程；

先进先出（队列）调度：实时进程先创建的先执行，直到遇到io或主动阻塞。

轮转调度（时间片）：达到一定的CPU执行时间后强制切换；

多进程程序的调度其实还是线程的调度，线程才是CPU调度的基本单位；在同一个进程内线程切换不会产生进程切换，由一个进程内的线程切换到另一个进程内的线程时，将会引起进程切换。

引起进程or线程调度的原因是正在执行的进程执行完毕；执行中进程发生阻塞；（如调用sleep）执行中进程调用了P原语操作，从而因资源不足而被阻塞；或调用了v原语操作激活了等待资源的进程队列；执行中进程提出I/O请求后被阻塞；CPU分配的时间片用完；（默认10ms）就绪队列中的某进程的优先级变得高于当前执行进程的优先级，引发强制切换；

linux下python进程or线程调度，如果我们使用python创建了多进程或多线程，可以认为这几个进程或线程是在公平队列（即优先级相同）的实时进程，那么其调度策略是FIFO和RR。如现在有一个单核的CPU，python程序创建了5个线程，这五个线程会按创建的时间先后进入到一个公平队列中，CPU按先进先出原则开始执行第一个线程，如果遇到IO操作或休眠，或者执行这个线程的时间超过10ms；CUP就会停止当前线程，切换到第二个线程执行直到第五个线程；然后又从第一个线程开始循环，直到所有的线程执行完毕资源被操作系统回收。切换进程或线程也需要付出代价的，进程切换的代价大于线程。

进程、线程和协程的资源比较

进程：创建一个进程后，每个进程拥有自己独立的内存地址空间，代码段，数据段，BSS段，堆，栈等所有用户空间的信息；多进程中，子进程复制主进程的几乎所有信息，除了pid等特殊信息；

线程：一个进程下多个线程，多个线程共享进程的进程代码段，进程的公有数据（堆），进程的所拥有其他辅助资源；各个线程独立拥有的资源包括：线程id，程序计数器，一个栈，计数器寄存器和栈用来保存线程的执行历史和执行状态。

协程：协程可以看做轻量级的线程，即协程是在线程下开启，多协程在单线程下实现并发，而操作系统最多只能感知到线程，也就是说协程的切换对于操作系统来说是无感知的，属于程序级别的切换；多个协程共享单线程的代码段、公有数据（堆）等；每个协程拥有自己的栈来保存上下文状态，协程的切换开销更小，对操作系统来说，会认为一个开启了多协程的线程一直在计算；协程的优势在于切换的代价更小，因此CPU的有效利用率得到了提高。

python的协程主流通过gevent和asyncio模块实现，它们的核心原理都是底层用代码创建事件循环来对多个协程的上下文进行调度。

**14、采用自动化网页解析技术**

网络爬虫会将互联网上的非结构化信息抓取至本地，利用自动化网页解析技术对其进行结构化分析，通过人工辅助定义结构化信息标识，最终生成规则数据，完成对网页的结构化解析动作。

HTML即超文本标记语言。它是标准通用化标记语言SGML的应用。用HTML编写的超文本文档称为HTML文档，它能独立于各种操作系统平台(如UNIX， Windows等)。使用HTML语言，将所需要表达的信息按某种规则写成HTML文件，通过专用的浏览器来识别，并将这些HTML文件“翻译”成可以识别的信息，即现在所见到的网页。

借助HTMLParse，对HTML页面结构进行预处理，包含以下核心处理方式：

HTMLParser.feed（数据），将文本输入解析器。它由完整元素组成的范围内进行处理；缓冲不完整的数据，直到馈送或close()调用更多数据为止 。 数据可以是unicode或 str，但unicode建议传递。

HTMLParser.get\_starttag\_text（）返回最近打开的开始标签的文本。在处理“已部署”的HTML或以最小的更改重新生成输入（可以保留属性之间的空白等）。

HTMLParser.handle\_starttag（tag，attrs ）调用此方法来处理标签的开头（例如）。<div id="main">的标签参数是标记的名称转换为小写。该ATTRS 参数是一个列表包含标签的发现里面的属性对括号。该名称将被转换为小写，并在报价价值已经被删除，字符和实体引用已被替换。(name, value)<>例如，对于标记，此方法将称为。<A HREF="https://www.cwi.nl/">handle\_starttag('a', [('href', 'https://www.cwi.nl/')])

HTMLParser.handle\_startendtag（tag，attrs ）与handle\_starttag()相似，但在解析器遇到XHTML样式的空标记（）时调用。该方法可能被需要此特定词法信息的子类覆盖；默认实现只是调用和。<img ... />handle\_starttag()handle\_endtag()

HTMLParser.handle\_comment（数据）遇到评论时（例如<!--comment-->）调用此方法。例如，注释将导致此方法与arguments一起调用。<!-- comment -->' comment 'Internet Explorer条件注释（condcoms）的内容也将发送到此方法，因此，此方法将接收。<!--[if IE 9]>IE9-specific content<![endif]-->'[if IE 9]>IE9-specific content<![endif]'

使用上述功能等方式初步解析页面，借助人为分析网页的核心内容，包含标题，正文，发布时间等。完成自动化网页解析全过程。

##### 系统组成

网上采集系统由网采集管理子系统、采集决策子系统、智能网页分析子系统、代理管理工具、采集程序子系统、快照服务子系统等6个系统组成。



**图1.1‑10网上数据采集系统-系统组成图**

##### 用户与角色设计

###### 用户设计

网上书采集系统主要服务于接入大数据管理用户，分为采集标注用户、采集数据管理用户、数据服务维护用户。

**图 1.1‑1网上数据采集系统-系统组成图**

###### 角色设计

系统主要使用角色情况如下表：

**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 | 专业标注员、数据管理员、多源异构融合系统、应急有害发现系统 |

具体用户分析说明：

**1、专业标注员：**负责在子系统中分配的标注任务进行处理，执行指定操作，提交上报，审核提交的标注结果等。

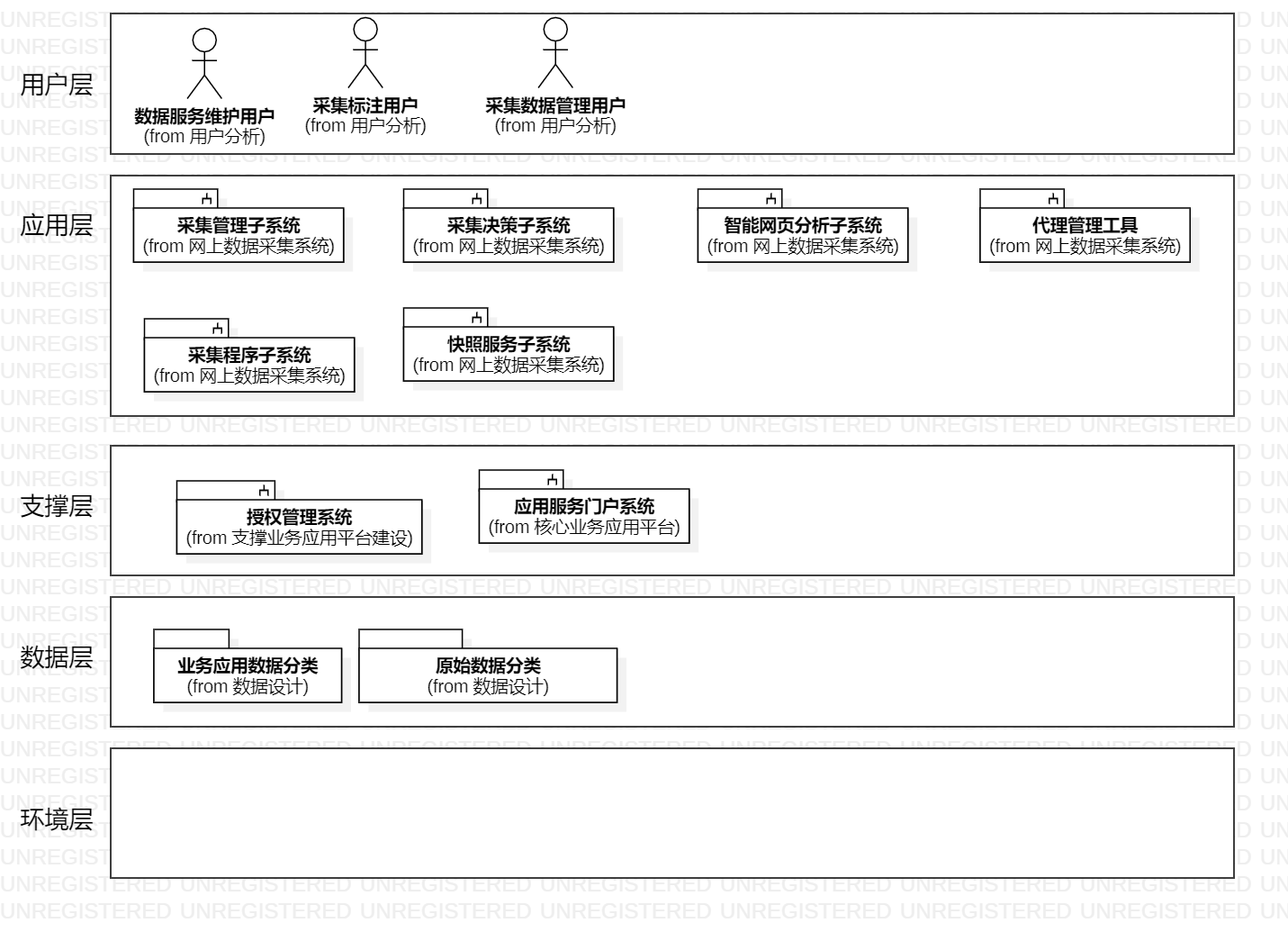
**2、多源异构融合系统：**与网上数据采集系统建立数据共享与数据支撑。

**3、数据管理员：**负责采集相关数据及业务子系统中数据管理。

**4、应急有害发现系统：**与网上数据采集系统建立数据传输，使用服务实现自身功能。

##### 系统体系结构设计

网上数据采集系统体系结构设计分为5层，含用户层、应用层、支撑层、数据层、环境层。总体体系结构如下：



**图 1.1‑7 网上数据采集系统-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，包含采集标注用户、采集数据管理用户、数据服务维护用户等。

**2、应用层**

应用层是在支撑层基础上构建的系统，主要为用户提供网采集管理子系统、采集决策子系统、智能网页分析子系统、代理管理工具、采集程序子系统、快照服务子系统。

**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**

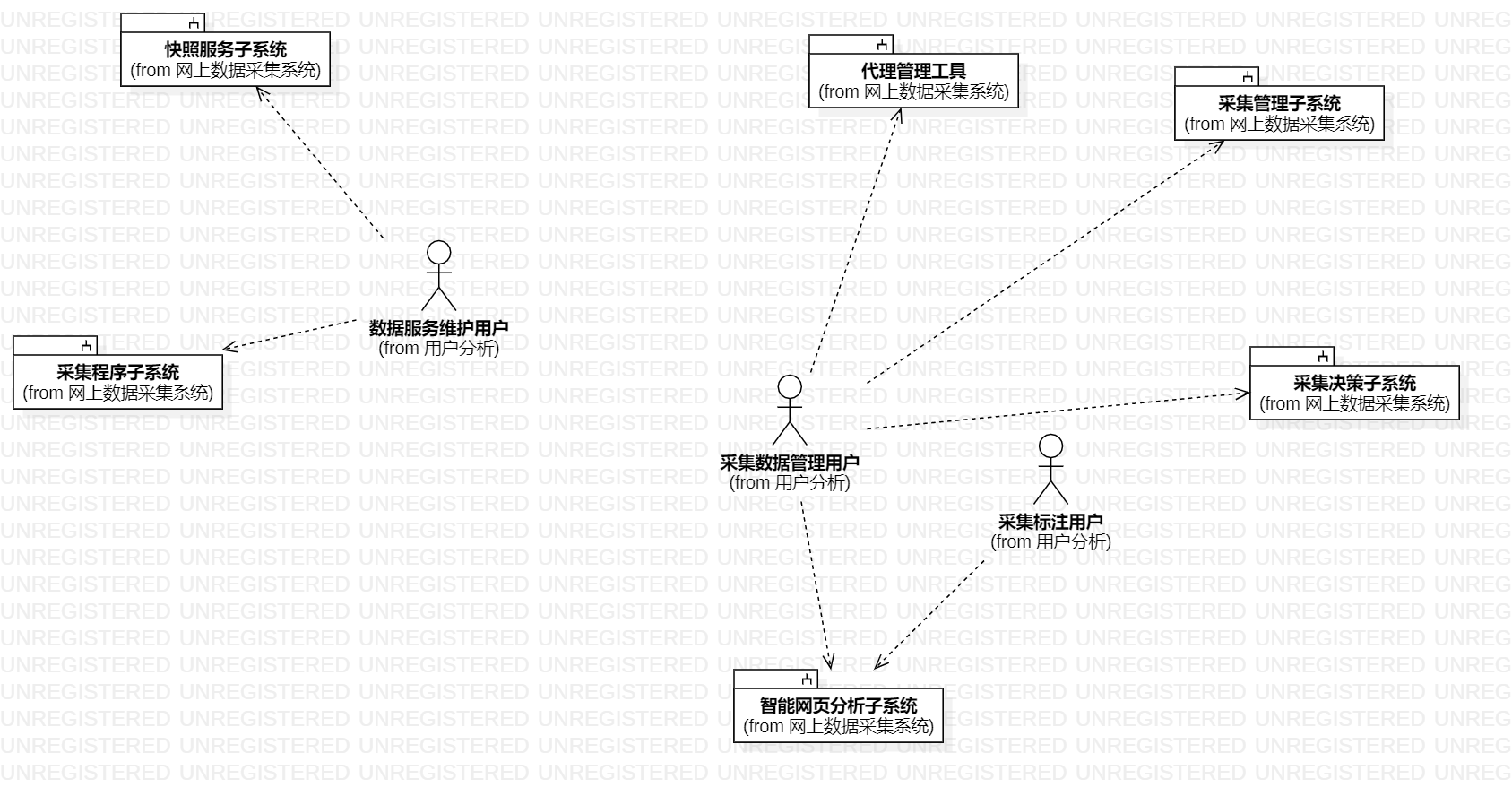
数据构建于环境层之上，并为上层的应用层、支撑层提供各种数据资源访问和存储服务，主要包括业务应用数据分类、原始数据分类等。

**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；

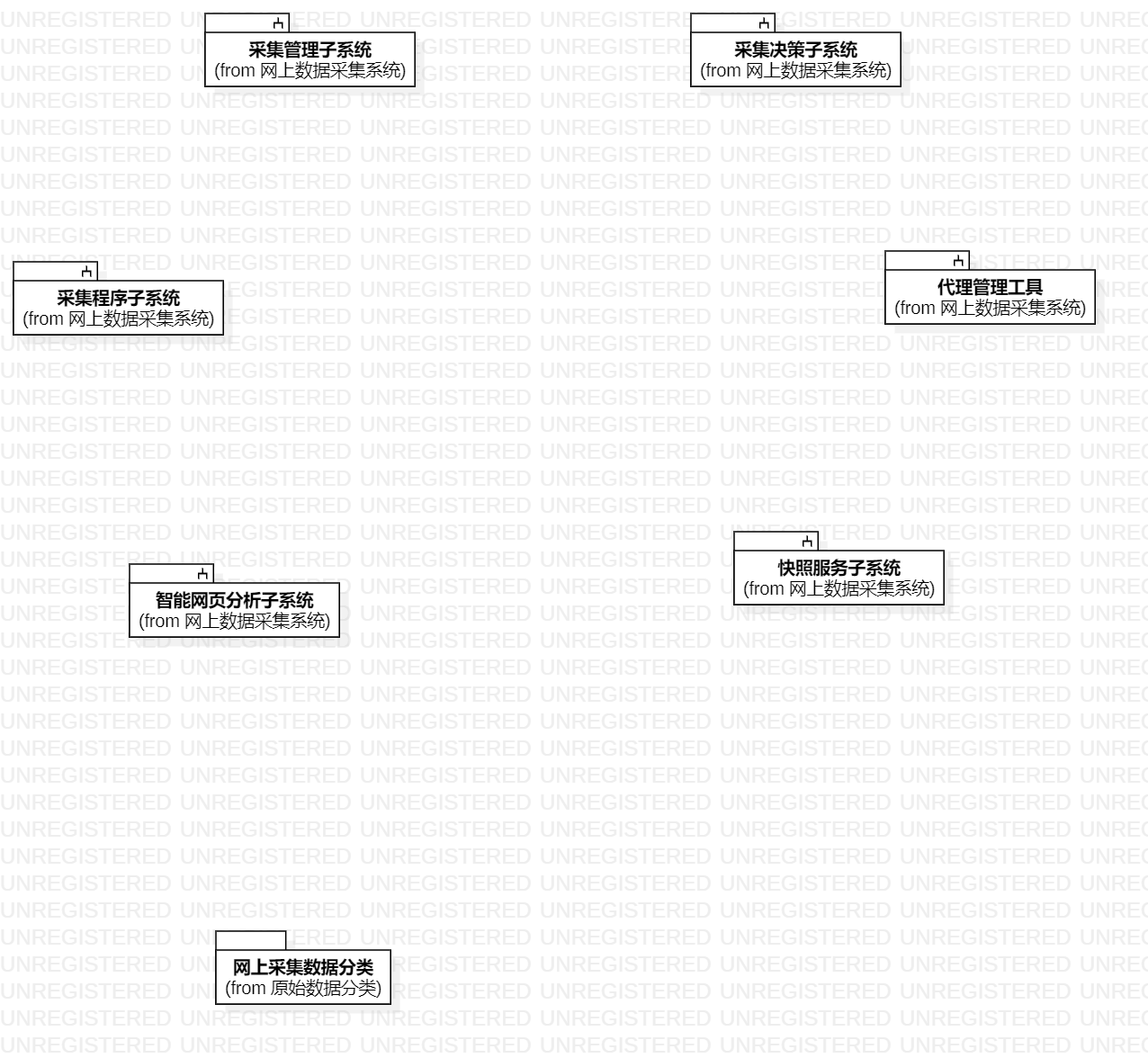
##### 系统总体设计

###### 用户与系统间关系设计



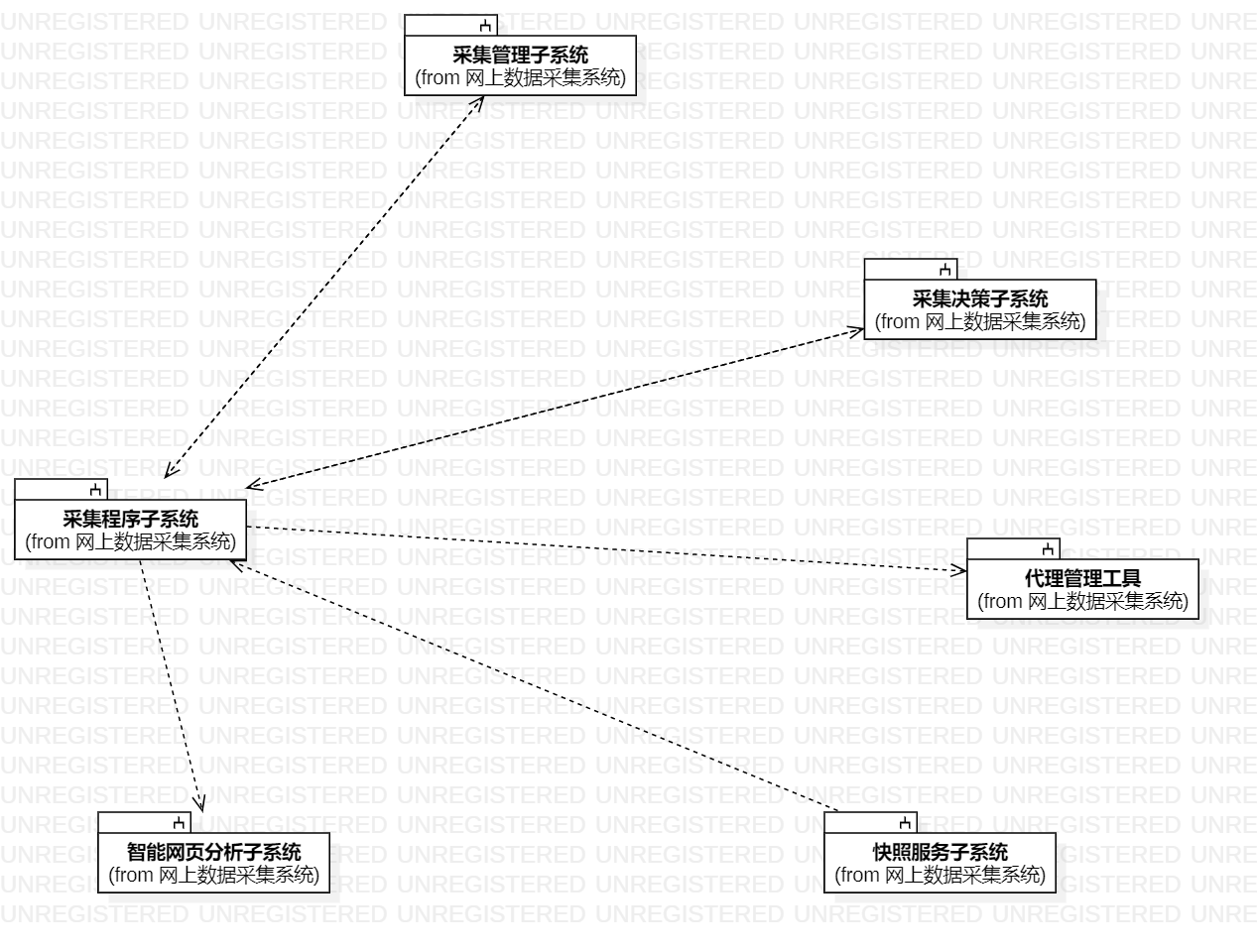
**图 1.1‑4网上数据采集系统-用户与系统间关系设计图**

###### 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**

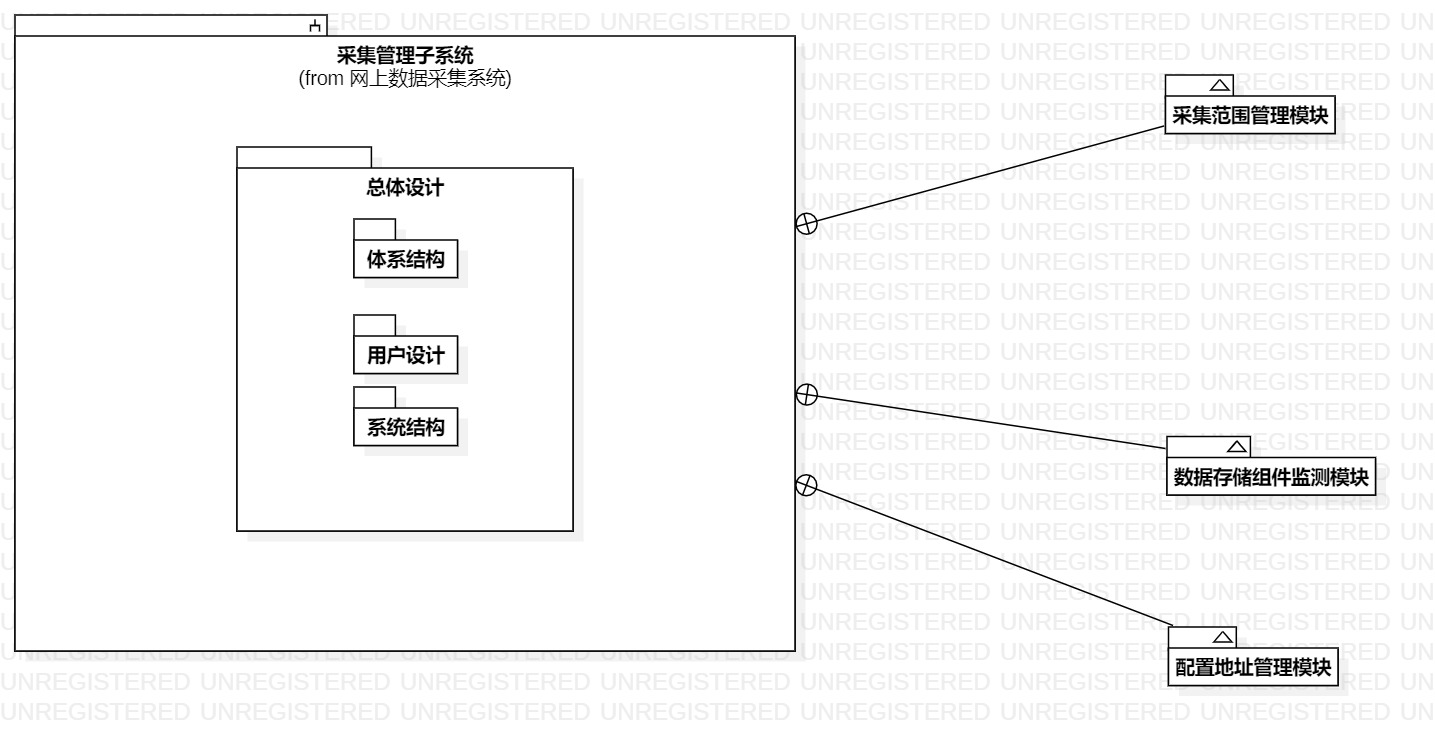
##### 采集管理子系统

###### 系统概述

采集管理子系统面向用户为数据管理员、采集程序子系统、采集决策子系统，通过数据同步机制，实现配置化采集流程，以及采集流程可视化。主要实现对采集范围的管理、对采集任务的集中调度管理，采集配置地址的信息维护，针对采集流程中重要节点进行实时监测分析等，是高质量数据的重要保障。

###### 系统组成

采集管理子系统由采集范围管理、数据量监测、任务管理与监测、数据存储组件监测、配置地址管理等5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

采集管理子系统主要服务于接入大数据管理用户，分为采集数据管理用户。



**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 | 数据管理员、采集程序子系统、采集决策子系统 |

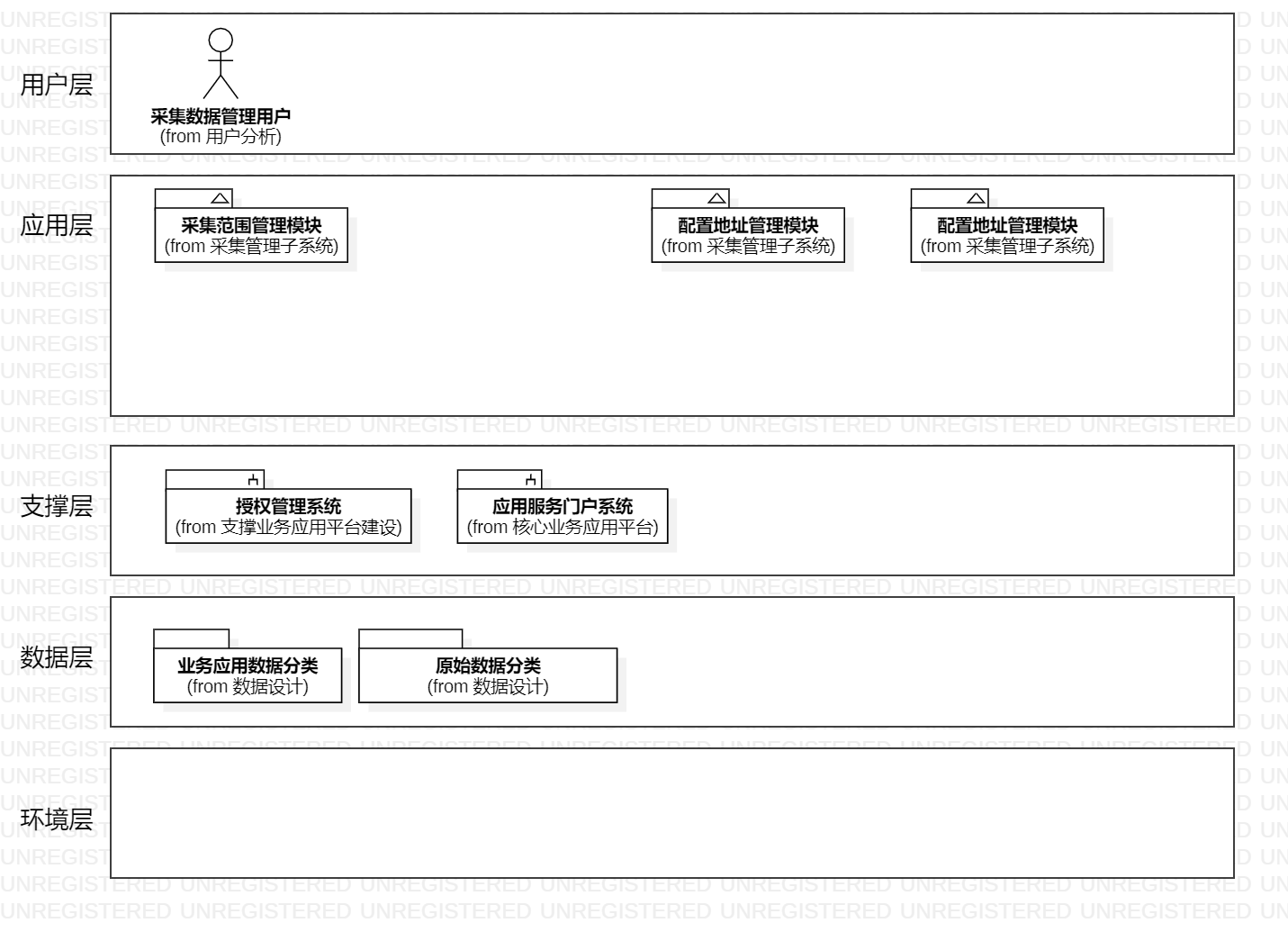
具体用户分析说明：

**1、数据管理员：**负责对管理调度、地址配置和监测数据及组件状态，范围管理等。

**2、采集决策子系统：**使用任务管理数据作为自身系统支撑。

**3、采集程序子系统：**使用范围管理数据、任务管理数据和配置地址数据作为自身系统支撑。

###### 系统体系结构设计



**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**

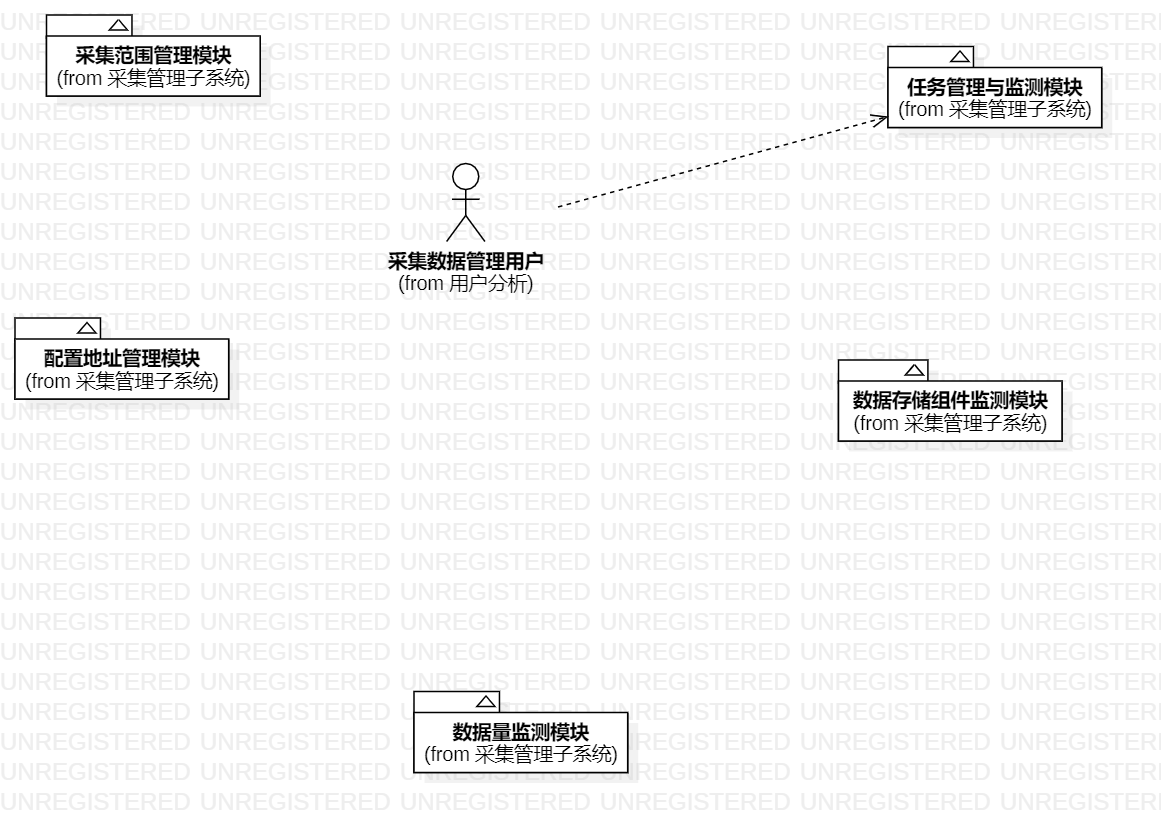
数据构建于环境层之上，并为上层的应用层、支撑层提供各种数据资源访问和存储服务，主要包括业务应用数据分类、原始数据分类等。

**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

用户与模块间关系设计



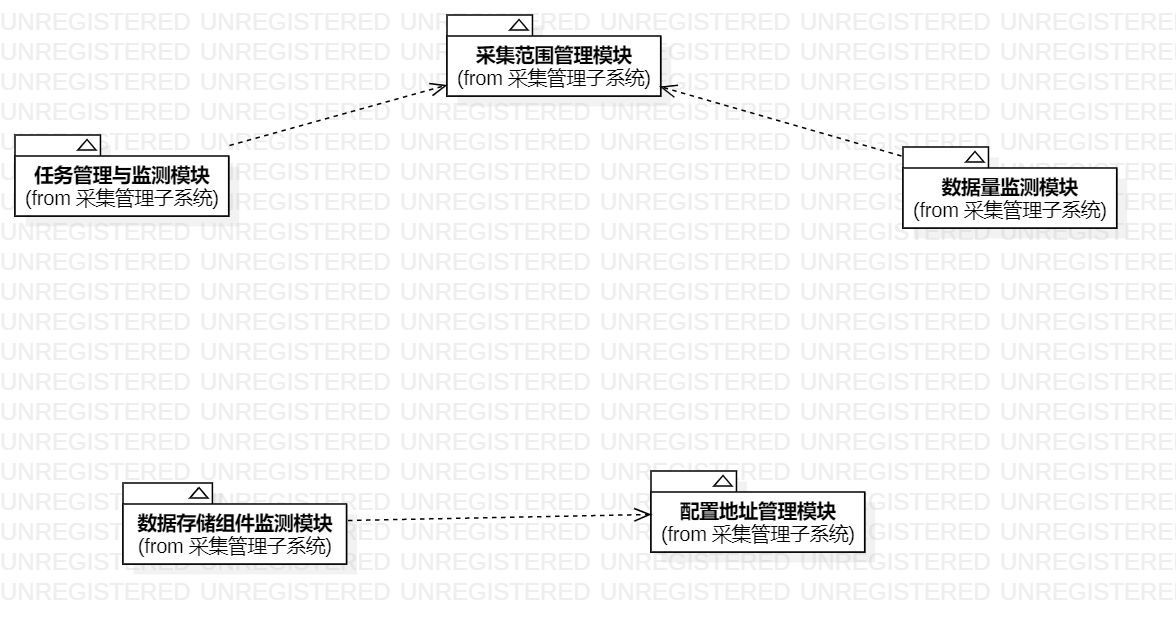
**图 1.1‑4采集管理子系统-用户与模块间关系设计图**

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**

###### 采集范围管理模块

采集范围管理模块面向用户为系统管理员，将采集范围统一管理把控数据最初入口，是采集程序采集目标的范围基础支撑，基于B/S结构建设，实现对采集程序子系统涉及采集范围定义，统一管理范围避免多库管理产生数据信息落差，为后续采集流程中的调度管理提供数据支撑，是整个采集流程中的重要部分之一。通过范围管理结合数据同步机制实现了采集范围可配置化。

采集范围管理针对数据源进行不同类型范围管理，包含关键词管理、账号管理和上报管理。

###### 数据量监测模块

数据量监测可直观查看采集结果，是采集能力的体现。数据监测模块的主要功能是将指定数据源采集数据进行统计实现数据可视化，对数据进行实时监测，若超出设定阈值将提前预警及时报警。它主要分为数据量监测功能和数据量预警功能，数据量监测功能指的是对多数据源（如：新浪微博、新闻媒体、属地网站、论坛博客等）的数据量进行统计，数据量预警功能指的是当数据低于阈值，启用预警功能，通知相关的开发管理者进行错误的排查与管理。

###### 任务管理与监测模块

任务管理与监测监测模块面向用户是系统管理员，是对采集调度的管理和数据通道的监测，有效把控调度重要参数，实时反馈数据处在采集流程中的存亡状况。任务管理与监测模块由调度管理，任务监测和计数器监测组成，采集调度是采集策略的体现，与采集范围数据对接统一下发调度任务，透明调度配置信息减少开发工作，明晰项目中调度下发配置信息；任务监测即针对已下发调度任务中的范围进行实际采集请求的监测，明确采集时间及采集准确目标信息；计数器如采集流程中的监视器存在于每一个数据交换点，有效监控采集中细小变化，对数字敏感做到提前预警。此模块由调度管理、采集任务监测和计数器监测组成。调度管理实现对采集流程中范围调度的统一管理。

###### 数据存储组件监测模块

数据存放组件监测模块面向用户为系统管理员，包含Redis监测，Hbase监测和FastDFS监测。该模块以监测数据存储组件运行和使用情况为主，实时获取最新组件数据，进行可视化展示并配有完善的预警报警机制，可指定内容和发送人将数据按预警或报警等级进行消息发送，及时发现问题并提醒处理，避免发生数据流失或堵塞状况，造成无人监管导致的数据中断，保证应用系统正常使用数据，

###### 配置地址管理模块

配置地址管理模块面向用户为系统管理员，将采集所需要配置地址的全部信息，进行统一管理实现一键式同步配置信息减少开发工作，其中包含Redis、Hbase、FastDFS，ES和MySQL配置地址管理。此模块为采集程序提供配置信息的数据支撑，不同存储地址根据需求产生不同数据结构和内容，包容性强，适用广泛，操作简单，可支持定制化加入子模块进行地址管理。

###### 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 1.1‑3采集管理子系统-调用接口清单表**

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 接口名称 | 提供系统名称 | 描述 |
| 1 | 专题关键词获取接口 | 应急有害发现系统 | 获取舆情运营专题关键词 |
| 2 | 账号信息查询接口 | 应急有害发现系统 | 获取各账号的信息，关注状态等 |

提供接口设计

本系统未对外提供接口。

###### 功能与用户权限对应关系

**表1.1‑4采集管理子系统-功能与用户权限对应关系表**

| 序号 | 子系统 | 模块 | 使用用户 | 备注 |
| --- | --- | --- | --- | --- |
| 1 | 采集管理子系统 | 采集范围管理模块 | 数据管理员 |  |
| 2 | 数据量管理模块 | 数据管理员、采集决策子系统 |  |
| 3 | 任务管理与监测模块 | 数据管理员、采集程序子系统、采集决策子系统 |  |
| 4 | 数据存储组件监测模块 | 数据管理员、采集程序子系统 |  |
|  | 配置地址管理模块 | 数据管理员、采集程序子系统 |  |

##### 快照服务子系统

##### 快照服务子系统

##### 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 1.1‑5网络采集系统-调用接口清单表**

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 接口名称 | 提供系统名称 | 描述 |
| 1 | 专题关键词获取接口 | 应急有害发现系统 | 获取舆情运营专题关键词 |
| 2 | 光栅数据换入接口 | 属地数据对接系统 | 光栅数据换入交换功能。 |
| 3 | 属地数据对接接口 | 属地数据对接系统 | 厂商属地数据对接数据查询功能 |
| 4 | 语种接口 | 多语种翻译引擎、网信数据中台 | 提供文本语种信息 |

###### 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 1.1‑6 网络采集系统-提供接口清单表**

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 接口名称 | 调用系统名称 | 描述 |
| 1 | 有害数据上报接口 | 应急有害发现系统 | 对有害数据判别信息更新数据存储 |
| 2 | 存储数据更新接口 | 应急有害发现系统 | 对已有数据字段进行更新 |
| 3 | 快照碎片查询接口 | 应急有害发现系统 | 快照碎片信息查询 |
| 4 | 快照碎片快速补充接口 | 应急有害发现系统 | 快照碎片信息快速补全 |
| 5 | 快照语种查询接口 | 应急有害发现系统 | 快照语种信息查询 |

##### 用户权限设计

**表1.1‑7网上数据采集系统-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 | 网上数据采集系统 | 采集管理子系统 | 数据中心管理人员 | 数据管理员、采集程序子系统、采集决策子系统 |  |
| 2 | 采集决策子系统 | 数据中心管理人员 | 数据管理员、采集管理子系统 |  |
| 3 | 代理管理工具 | 数据中心管理人员 | 数据管理员、采集程序子系统 |  |
| 4 | 智能网页分析子系统 | 数据中心管理人员 | 数据管理员、专业标注员、采集程序子系统、多源异构融合子系统 |  |
| 5 | 采集程序子系统 | 数据中心管理人员 | 多源异构融合系统 |  |
| 6 | 快照服务子系统 | 数据中心管理人员 | 多源搜索子系统、应急专项分析子系统、短视频有害发现子系统、重点目标布控子系统 |  |

# 标题模板

## 二级

### 三级

#### 四级

##### 五级

###### 六级

七级

八级

九级

