高能物理开放元数据管理框架设计方案

科研大数据元数据管理软件框架关键技术研究与开发项目组

二〇二〇年三月

目录

引言

科学数据一般是指在自然科学、工程技术科学等领域，通过基础研究、应用研究、试验开发等产生的数据，以及通过观测监测、考察调查、检验检测等方式取得并用于科学研究活动的原始数据及其衍生数据，贯穿于科技创新活动的全过程。一般由大科学装置和大科学项目产生的科学数据又被称为科学大数据。

基于同步辐射和中子技术的大科学装置是国家投资建设的大型科研设施，属于多学科交叉前沿的公共实验类研究平台，如同步辐射光源、自由电子激光装置和中子散裂中子源。大科学装置每年会产生 TB 级或者 PB 级甚至数十 PB 级的海量科学大数据，这些数据具有不可重复性、高维性、高度计算复杂性和高度不确定性等特征。高能同步辐射光源（High Energy Physics Photon Source, HEPS）作为我国服务于超高空间分辨、时间分辨、能量分辨的高通量同步辐射实验等多学科综合研究、为国家的重大战略需求和前沿基础科学研究提供技术支撑的平台类重大科技基础设施，建成后将是我国第一台也是世界上亮度最高的第四代高能量同步辐射光源。

高能同步辐射光源（High Energy Physics Photon Source, HEPS）是我国“十三五”期间优先建设的、为国家的重大战略需求和前沿基础科学研究提供技术支撑平台的国家重大科技基础设施，位于怀柔科学城北部核心区域。建成之后，HEPS 将是我国第一台高能量同步辐射光源，也将是世界上亮度最高的第四代同步辐射光源。

HEPS 建成之后将是一个开放的大型科学实验平台，年均面向全球用户提供实验机时将在 5000 小时以上，开展 X 射线衍射、散射、成像和谱学等同步辐射实验。作为第四代同步辐射装置，HEPS 产生的 X 射线能够获得毫电子伏的能量分辨、纳米的空间分辨率和飞秒的时

间分辨，相对于现有光源有数量级的提升。同时，随着光学、电子学技术的发展，以及先进的光学仪器、探测器的使用，用户实验过程中的实验数据将呈现爆发性增加，海量的实验原始数据、实验元数据需要高效、安全地进行采集、传输、存储、分析和共享，以满足装置、光束线实验站、用户等各方面的需求，促进依托 HEPS 的科研实验产出。

国家高能物理科学数据中心采用“一平台多中心”的发展思想，持续统筹推进国家高能物理科学数据中心、国家高能物理科学数据中心大湾区分中心两大区域中心能力建设，同时以高速网络连接北京、东莞、稻城、济南等多个大型高能物理实验装置，聚合山东大学、中国科技大学等众多合作组成员单位资源，为国内及全球上千名高能物理以及其它领域的科研人员提供形式多样、稳定可靠数据访问服务。国家高能物理科学数据中心持续实时关注国际同领域数据中心相关的政策、技术、平台等发展动态。高能物理领域的大型高能物理实验采用全球大科研工作模式，国际上高能物理研究机构均建立了同科研活动紧密结合的科学数据中心平台，并有完善的资源组织模式、运行维护体系以及技术发展方向。国际高能物理领域采用全球范围内的科研数据与计算系统融合技术，形成大规模数据平台服务模式，可以大幅提高数据资源价值，服务于依托重大科技基础设施开展的多学科、多领域科学数据分析需求。

HEPS 作为我国面向多学科综合研究平台类重大科技基础设施，其运行和科学数据处理处理效率以及用户体验至关重要。HEPS 科学数据处理平台作为设施的重要支撑，需要提供依托 HEPS 的科学实验过程、实验数据分析与共享、科学成果发布与管理等科研活动过程的自动化。

建成之后的 HEPS 将面向全球用户为其提供开展 X 射线衍射、散射、成像和谱学等同步辐射实验年均 5000 小时以上的实验机时，海

量的实验原始数据和光束线实验站预计平均每天产生的 200TB 的原始实验元数据需要高效、安全地进行采集、传输、存储、分析和共享。科学数据管理系统将实现从实验不同阶段从控制系统、用户服务系统、数据分析系统获取数据和元数据存放进以元数据管理软件框架为关键技术的科学元数据目录管理架构中以保证科学数据的完整性和可追溯性，并制定相应的科学数据管理标准与规范，实现在数据管理制度和规范下科研人员、工程技术人员以及用户对数据的可查看、可下载、可共享和可利用高效便捷的用户数据服务。

科学数据管理系统是 HEPS 科学数据处理平台的重要组成部分，科学数据管理系统对 HEPS 实验产生的所有科学数据的在数据获取、传输、存储、分析和数据成果发布各个阶段进行全生命周期的管理和跟踪，是科学数据管理平台实现从科学实验用户视角保障用户实验全过程的信息化、自动化和便利化，从 HEPS 设施运行视角实现实验运行过程的数字化和智能化，实现 HEPS 的计算平台硬件架构异构化、软件功能模块化、功能接口及数据管理标准化的重要支撑，对推动 HEPS 科学数据处理平台顺利成为依托于国家重大科技基础设施开展科学研究和科学实验的重要支撑系统，未来满足光学仪器、探测器等技术快速发展对科学数据和计算系统在功能、性能等各方面的需求具有重要意义。

HEPS 科学数据处理平台需要以服务依托 HEPS 设施开展科学实验的用户和科学家需求，围绕科学实验数据全生命周期开展支撑服务。依托信息化基础设施，结合信息化业务系统软件，在整个科学实验的前期准备阶段、实验开展阶段以及试验后阶段均提供相应的融合的科研信息化综合服务。

元数据管理软件框架是科学数据管理系统的关键技术，元数据管理软件框架需要实现对科学数据进行全生命周期的管理。元数据提取器从消息队列消费元数据，并将元数据通过元数据管理 API 存放至元

数据目录数据库中。用户通过提供的数据 web 服务界面在元数据目录数据库中搜索查找数据集。同时，用户可以将数据发送至现有计算资源或超算平台进行分析和处理，数据相关分析和处理作业信息被保存到元数据库，实现对数据处理过程的跟踪。数据分析的结果被保存到数据中心存储，相关分析结果元数据和数据发布信息最终被保存至元数据库。科学数据管理系统是 HEPS 科学数据处理平台的重要组成部分，需要依托元数据管理软件框架对实验数据进行组织和管理。

1. 研究背景、目标及内容

HEPS 科学数据处理平台需要以服务依托 HEPS 设施开展科学实验的用户和科学家需求，围绕科学实验数据全生命周期开展支撑服务。依托信息化基础设施，结合信息化业务系统软件，在整个科学实验的前期准备阶段、实验开展阶段以及试验后阶段均提供相应的融合的科研信息化综合服务（图 1）。

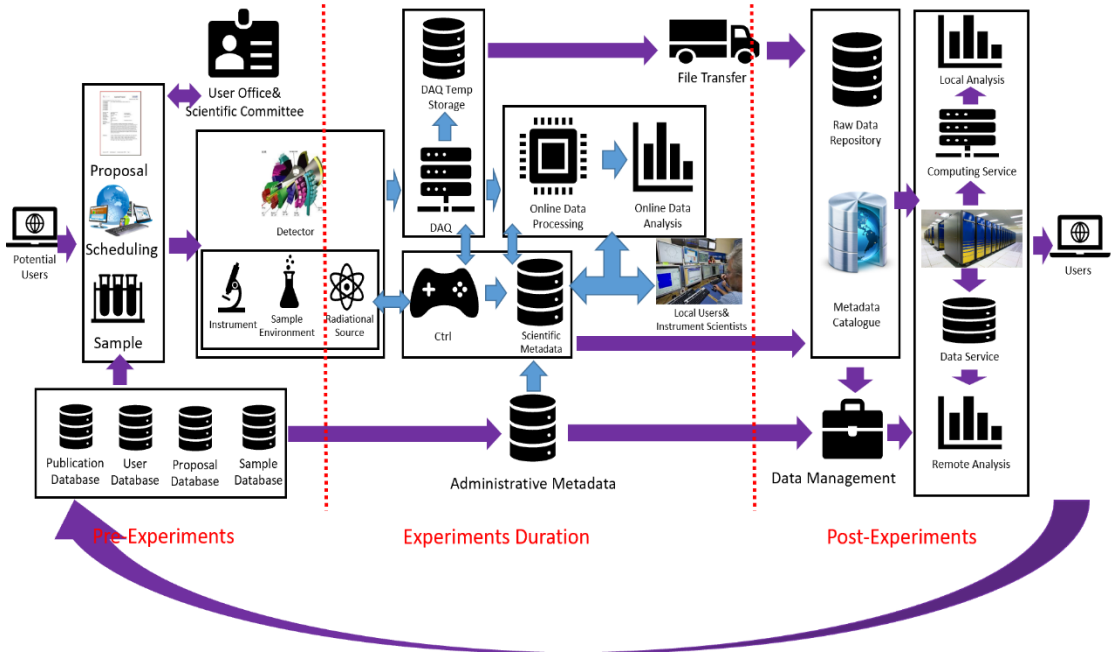

图 1　HEPS 科学实验过程与科学数据处理系统

HEPS 一期工程包括 B1-BE 共 14 个线站和 1 个测试线站，预计平均每天平均产生 200TB 的原始实验数据（峰值可达每天 500TB）（表 1）。但这些线站中不同线站数据产生速率差异较大。其中 B7 线站的原始数据产生速率约为 10GB/s，后续有可能继续增加到 30GB/s，预计每天产生的数据量为 100TB-500TB。原始数据会通过对探测器像素的分块采集，由多流写入存储系统。B2、BA 和 BE 线站的原始数据产生速率约为 500 MB/s-1.5 GB/s，预计每天产生的数据量为在 15TB-

40TB 之间。其余线站的数据产生速率均小于 200MB/s，每天产生的数据量小于 4TB。三种线站的数据写入吞吐率之间各相差了一个数量级，B7 线站的存储空间需求和数据读写吞吐率需求超过其他线站的总和。

表 1　HEPS 各实验站原始数据统计表

线站编号	每天平均产生数据量（Byte/Day）	每天峰值产生数据量（Byte/Day）
B1	820G	4T
B2	14T	20T
B3	112.5G	1.4T
B4	2T	2T
B5	6.8G	10G
B6	20G	50G
B7	95T	520T
B8	10.32G	30G
B9	5G	10G
BA	35T	35T
BB	91.08G	165.6G
BC	1T	1T
BD	275M	500M
BE	11.2T	25T
合计	159.27T	608.67T

1.1 项目背景及基本情况

线站产生的科学实验数据均需要得到及时、快速的处理、分析、存储和共享，同时需要提供实验数据的实时分析和快速反馈，以便为实验站用户提供决策以指导和修正实验过程。

从光束线实验站科学家角度出发，为了线站的高效运行，需要足够的数据存储资源并提供满足不同计算需求（CPU、GPU、FPGA 等）的在线分析环境。部分实验站产生的实验数据产生速率较大（如 B7 线站未来采用高分辨面探测器，数据产生速率可达 50GB/s），需要具有融合数据、计算、网络、软件环境等多种功能并且优异性能的科学数据处理平台作为支撑，实现科学实验过程和科学成果获取所需实验数据的及时处理、反馈和利用。从实验用户角度来说，希望能在短时间内完成实验数据的分析、获得理想的实验成果并获取科研成果，因此需要 HEPS 能够提供快捷、便利、易用和友好的支撑环境，方便用户跟踪、访问、下载实验数据和成果，同时保证实验数据的长期保存、安全和完整。部分实验数据由于特别庞大（如衍射断层扫描实验，一次实验可能产生高达 50TB 的数据），一方面需要优质的网络性能实现实验数据的快速传输，另一方面希望 HEPS 能够提供强大的计算平台和软件环境进行实验数据分析。

从整个 HEPS 装置来说，对平均每天 200TB、文件大小不一（KB~GB）、格式不一（ASCII、图像、HDF5）、来源不一（不同实验技术、光束线、前端等）的实验数据进行统一化管理是非常重要的工作。涉及科学数据库（实验数据库、元数据库、仪器设备数据库等）、数据管理规范、数据标准化、数据接口等一系列工作，为科学数据利用和共享提供支撑和保障。

此外，HEPS 总体规划中，未来将有超过 90 个光束线站运行，届时每天产生实验数据将会有更大幅度的提升，这对科学数据存储、分析、管理和共享带来更大的压力，其科学数据处理平台在规划时需要

考虑架构、技术上的扩展性。

HEPS 科学数据处理平台将为 HEPS 装置、束线科学家、工程技术人员以及用户提供包括设数据传输、数据存储、数据分析、数据共享、科研协同等在内的网络、计算、存储等基础设施能力，以及提供科学软件、通用软件、通用信息系统和网络信息安全服务等。同时，在平台整体规划和建设时，统筹和设计同数据相关的多个系统之间的标准化和规划化接口。根据科学数据处理平台系统功能，具体系统包括支撑平台运行的基础设施、网络系统、存储系统、计算系统、公共信息系统、网络安全系统、智能化运维保障系统以及服务于科学数据全生命周期的的软件系统、数据管理系统（图 2）。

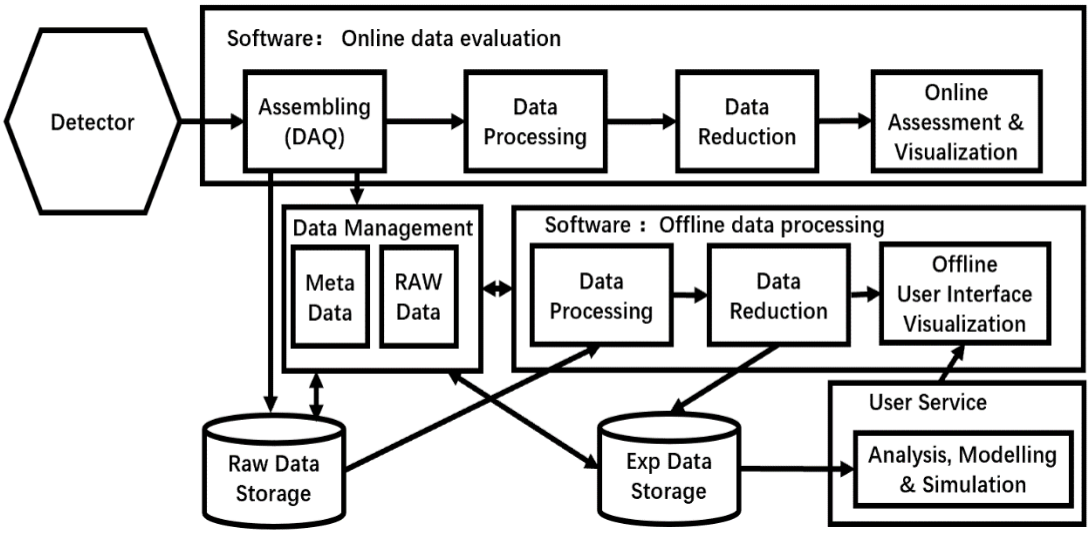

图 2　科学数据流程

科学数据处理系统是 HEPS 科学数据处理平台的重要组成部分。元数据管理软件框架对实验元数据进行组织管理，实现科学数据高效地组织和利用，实现科学数据生命周期的跟踪与管理，提高前沿交叉学科基础研究、基础应用研究成果的产出效率。科学实验过程数据处理流程如图 3 所示，实验数据通过探测器被收集并存储到 DAQ Database、HDFS 文件块、磁带等存储介质中，Metadata Creator（元数据提取器）从用户服务系统获取提案、用户、样本相关信息，从实验参数文件获取部分关键实验元数据，并存储到元数据目录数据库 MongoDB，用于实验数据的查找、搜索和共享。实验数据的组织和管理需要依赖于实验元数据，将实验元数据存储到元数据目录数据库，能够实现实验数据生命周期的跟踪与管理。

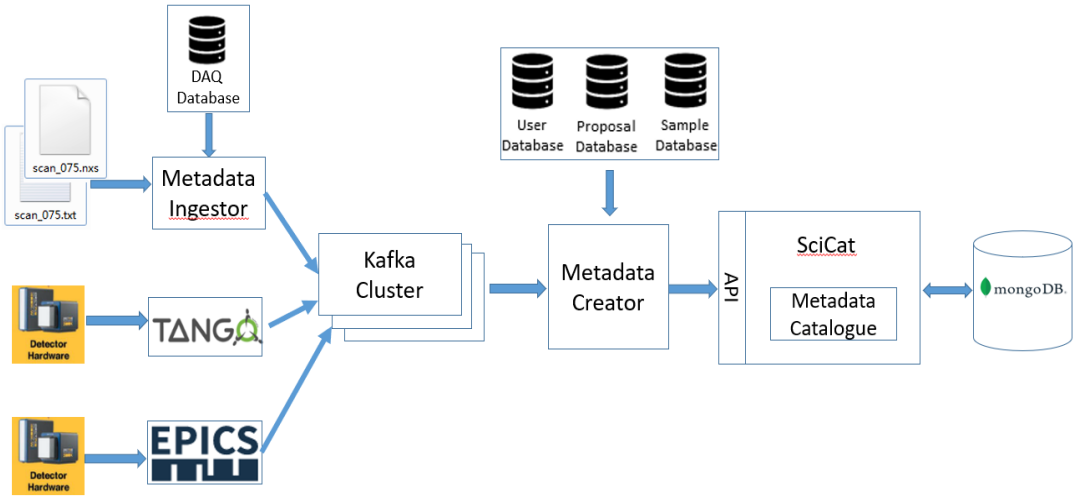

图 3　HEPS 科学实验过程数据处理流程图

从实验用户角度来说，科学数据是设施实验的核心，贯穿于整个科研活动过程中，在实验过程的不同阶段中会产生相应的元数据和原始数据。实验过程中，用户需要实时查看实验产生的数据，对数据进行快速分拣和提炼，对实验过程中数据质量检查及快速分析；在实验完成后，用户希望根据不同的参数进行快速查找数据，查看数据，下载数据，发布数据。

从 IT 角度出发，整个科学数据管理流程包括：数据获取、数据

格式标准化、数据存储，数据目录索引（catalogue），数据服务。首先从线站获取原始数据和元数据，对数据进行封装，并转化成标准格式 HDF5 文件，将原始数据和标准化后数据文件传输到数据中心存储。元数据管理软件框架负责存储元数据提取器收集到的实验元数据（如：用户系统的用户提案数据、DAQ 系统的实验样本数据、HDF5 文件系统的文件存储信息数据）。

1.2 研究目标及内容

需要实现对 HEPS 实验产生的所有科学数据的在数据获取、传输、存储、分析和数据成果发布各个阶段进行全生命周期的管理。需要实现的目标包括：制定科学数据管理标准与规范；研究和设计科学元数据目录管理架构，实现对科学数据全生命周期的管理，保证科学数据的完整性和可追溯性；实现从实验不同阶段从控制系统、用户服务系统通、数据分析系统获取数据和元数据；提供标准接口，满足其他各系统之间的协作与通讯；提供高效便捷的用户数据服务，实现数据管理制度和规范下对数据的可查看、可下载、可共享和可利用。

（1）数据管理标准与规范

为了提升科学数据开放和共享的效率，大科装置需要研究并设计统一的格式对科学数据进行管理，科学数据的管理需要从科学数据组成、数据操作权限和数据标识符三个角度来进行研究。

科学数据的特点是数据量大且构成复杂，包括了实验原始数据、元数据和实验分析数据，需要根据不同类型科学数据的特点，研究设计相应数据的管理格式。实验原始数据和元数据直接依赖于实验设施及其运行，其产生代价高昂，且是所有后续科研活动的起点。因此，科学数据的安全尤为重要，需要明确数据的操作权限以保证科学数据的安全。

缺乏统一的数据标识会导致科学数据的独立出版和引用问题，这

也是科学数据管理中需要解决的问题，在科学数据管理中需要对科学数据集的唯一永久标识符(Persistent Identifier,PID)进行研究。目标是制定符合国家规范、符合领域规范、得到用户认可的科学数据管理标准规范。

（2）数据索引

整个 HEPS 装置平均每天能产生 200TB、文件大小不一（KB-GB）、格式不一（ASCII、图像、HDF5）、来源不一（不同实验技术、光束线、前端等）的实验数据。借助科学数据处理软件系统，把实验采集的测量数据处理成为具有明确物理意义的科学数据，并通过对科学数据的分析获得对实验对象的科学认识。在整个处理流程中，需要对所有科学数据的在数据获取、传输、存储、分析和数据成果发布各个阶段进行全生命周期的管理和跟踪。因此需要研究和设计出科学元数据目录管理架构，实现对原始数据和衍生数据的 catalogue，即实现对科学数据全生命周期的管理，保证科学数据的完整性和可追溯性。

（3）开放接口

根据科学数据处理平台系统功能，具体系统包括支撑平台运行的基础设施、网络系统、存储系统、计算系统、公共信息系统、网络安全系统、智能化运维保障系统以及服务于科学数据全生命周期的的软件系统、数据管理系统。

（4）元数据获取

科学数据管理系统管理的数据包括整个 HEPS 装置产生的实验数据、由科学数据处理软件系统处理生成的科学数据，以及由平台其他系统产生的元数据和数据。为了科学数据管理系统能统筹管理这些数据，根据用户或实验的需要，将数据从存储中心拿出利用，需要研究数据和元数据的获取方法，实现从实验不同阶段从控制系统、用户服务系统、数据分析系统获取数据和元数据。

（5）数据服务

可访问是科学数据开放共享需要满足的基本原则。科学数据的访问对象范围较广，如实验用户、装置相关工作人员、普通用户等。实验用户是数据的生产者，需要掌握科学数据的所有相关信息。装置相关人员负责科学数据的管理，需要被赋予部分科学数据类型的访问权限。对于普通用户，大科学装置可以在一定条件下公开相关科学数据类型的访问权限。访问期限也是科学数据访问权限中需要研究的问题，设置合理的数据保护期以保障参与数据生产的用户在数据使用上的优先权。另外，不同类型科学数据的访问权限也应该有所不同。在实验数据未被公开发表之前，科研人员可以独自使用已经得到确证或有效的数据。一旦科研人员将实验结果公开发表，其他人就可以自由地获取实验涉及到的所有数据，包括最终结果，以便于检验和使用。对于这些数据如何能被其他科研人员重复使用，在开放数据领域，国外提出科学数据的 FAIR 原则，包括数据的可发现（Findable）、可访问（Accessible）、可互操作（Interoperable）和可重用（Reusable）。综上，可从用户类型、数据访问期限和科学数据类型、FAIR 原则等角度对科学数据的开放共享进行研究，实现提供高效便捷的用户数据服务，实现数据管理制度和规范下对数据的可查看、可下载、可共享和可利用的目标。

1.3 拟解决的关键技术问题

（1）对不确定性元数据项的存储

科研元数据的字段项具备不确定性：字段数量不确定、字段间关系不确定、字段内部嵌套深度不确定，针对这种情形，数据模型无法确定 schema，导致传统存储方式难以完成元数据的存储。本课题拟设计一种新的存储策略，能够让一个数据模型同时拥有多个 schema，不同数据实例只需要对应不同的 schema 即可。

（2）面向同步辐射实验线站数据管理员的接口定义方式

线站的实验数据通常由实验人员的实验方案决定，不同的实验方案可能会产生不同的数据接口需求，需求的变化往往会引起数据接口的变动，接口变更需要进行编码工作，这种方式会影响整个科研数据处理平台的效率。本课题拟设计一种能够让数据管理员动态定义接口的方案。

（3）配置规范

配置规范包含：①数据模型的定义规范，即通过数据模型配置定义表结构；②数据接口的定义规范，即通过接口配置定义后端数据接口。本课题拟设计一套实验线站数据管理员和框架均能读懂的简易配置规范。

通过背景、目标及相关关键技术分析，结合项目实际需求和研讨，确定主要需求包括：

（1）为数据管理开发人员提供 WEB 界面能简单、便捷地配置元数据模型及模型之间的关系。

（2）通过简单的元数据模型配置，自动生成 API，满足其他各系统与数据管理软件之间的交互需求。

（3）不同的元数据需要关系/非关系型数据库的支持。

（4）兼容已有的数据管理元数据目录，支持从已有数据库生成元数据模型。

（5）通过数据管理软件框架可快速开发和部署数据管理软件。

项目目标：实现数据管理软件快速、便捷的开发和部署。

（1）通过 web 界面实现元数据模型及模型关系配置。

（2）通过元数据模型及模型关系配置快速生成 API。

（3）实现 API 对元数据模型及模型关系的验证。

（4）支持基于关系型和非关系型数据库建立元数据模型。

（5）预置 HEPS 通用的元数据模型，可实现元数据管理目录 API 快速部署。

（6）实现由已有数据源生成元数据模型的逆向工程。

2 方案总体架构

科学数据管理系统需要实现对 HEPS 实验产生的所有科学数据的在数据获取、传输、存储、分析和数据成果发布各个阶段进行全生命周期的管理和跟踪。需要实现的目标包括：制定科学数据管理标准与规范；研究和设计科学元数据目录管理架构，实现对科学数据全生命周期的管理，保证科学数据的完整性和可追溯性；实现从实验不同阶段从控制系统、用户服务系统通、数据分析系统获取数据和元数据；提供标准接口，满足其他各系统之间的协作与通讯；提供高效便捷的用户数据服务，实现数据管理制度和规范下对数据的可查看、可下载、可共享和可利用。

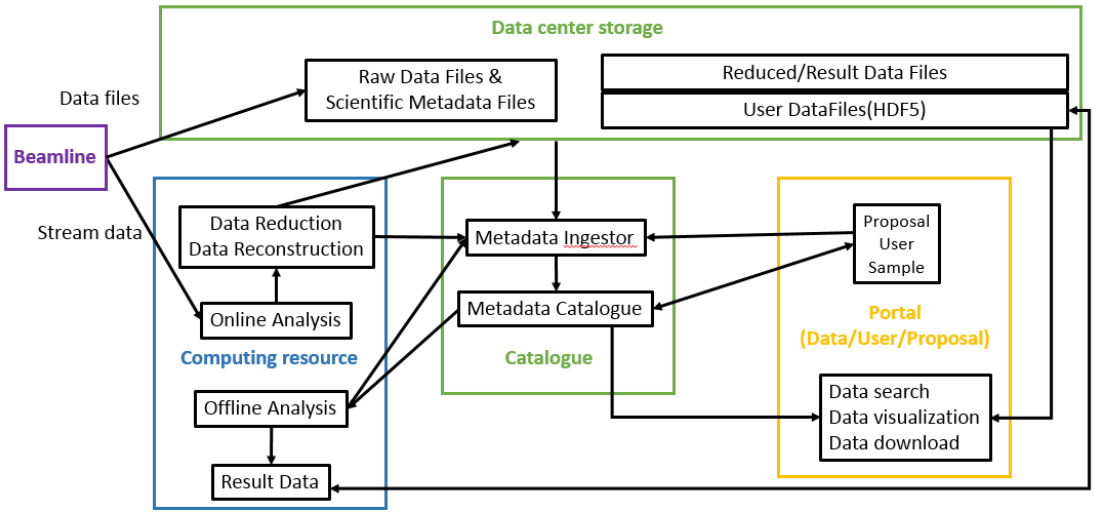

图 4　科学数据管理系统数据流图

科学数据管理系统数据流图（图 4）描述了整个实验过程中所有数据流向过程。具体包括：从线站接收的原始数据和科学元数据以文件的形式保存到数据中心存储。另一方面，原始数据流经过在线分析系统得到分析结果数据同样也保存到数据中心存储。同时，元数据提取器从用户服务系统获取提案、用户、样本相关信息，从实验参数文件获取部分关键实验元数据，并存储到元数据目录数据库，用于实验

数据的查找、搜索和共享。如果从探测器获取的原始数据文件不是标准 HDF5 格式，需要对数据进行格式转换和数据封装（包括原始数据和所有元数据），形成标准 HDF5 文件，注册元数据并长期保存。用户通过数据服务 portal 可以对数据进行搜索、查看、下载和离线分析，同时，被经过离线计算分析得到的结果数据同样会被存放到中心存储文件系统中长期保存。

2.1 元数据分类

元数据的分类及定义如下：

1）管理元数据（administrative metadata）：管理元数据包括与数据相关的提案信息、用户信息、实验类型、线站信息等，来自提案系统和用户系统。

2）科学元数据（scientific metadata）：包括与数据相关的样本信息和实验环境参数等相关信息，从控制系统获取。由于科学元数据会被用于数据分析和数据目录索引，它需要进行两部分存储：与原始文件一起进行数据封装标准化，生成 HDF5 文件，同时需要被存储到元数据库。

3）其他元数据（other metadata）：其他元数据主要来自传输或者分析应用，包括该数据传输中完整性校验信息（checksum）、分析软件名称版本信息、数据更新时间等。

2.2 元数据管理框架

SciCat 是由 PSI、ESS 和 MaxIV 合作研发的开源的元数据目录管理框架，通过微服务的架构对科学数据进行全生命周期的管理。SciCat 架构（图 5）采用了弹性框架，具有可扩展性，不受数据类型和数据量变化的限制；使用 json 文件定义元数据模型，并且可由元数据模型直接生成 RESTful API，供其他系统调用；后端采用非关系型文档数据库 MongoDB，支持高并发元数据读写，同时数据结构灵活，能符合各线站元数据不一致的需求；支持可视化编程的数据流处理，高效地匹配每个线站前期元数据处理和整理的流程；同时该架构集成基于 web 的可视化界面，支持元数据检索功能，简化后期 web 界面开发过程。

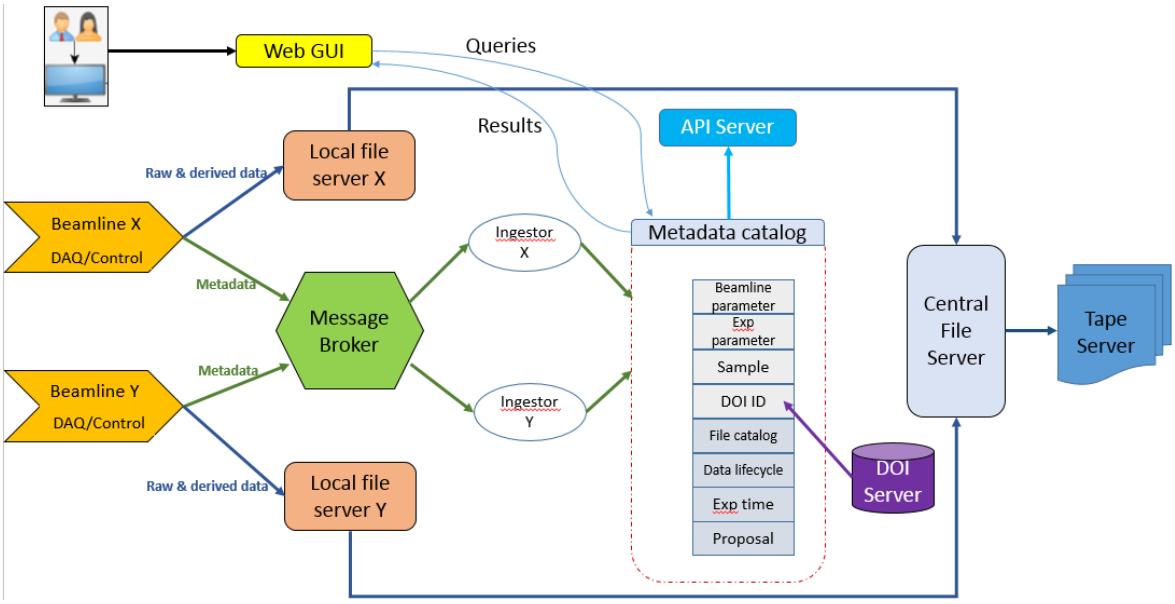

图 5　SciCat 元数据管理架构

该架构采用消息队列的方式接收从线站收集数据集的相关元数据，元数据提取器从消息队列消费元数据，并将元数据通过元数据管理 API 存放至元数据目录数据库中。用户通过提供的数据 web 服务界面在元数据目录数据库中搜索查找数据集。同时，用户可以将数据发

送至现有计算资源或超算平台进行分析和处理，数据相关分析和处理作业信息被保存到元数据库，实现对数据处理过程的跟踪。数据分析的结果被保存到数据中心存储，相关分析结果元数据和数据发布信息最终被保存至元数据库。该架构可实现数据集从提案、数据获取、数据存储、数据分析、数据发布整个过程进行全生命周期的记录和跟踪。

2.3 元数据管理软件框架架构

元数据管理软件从处理流程上来讲，用户在前端定义数据模型和接口，接口供线站调用。后端不仅负责系统服务，还要提供接口自动化创建的功能，创建好的接口能够被即时调用。处理流程如图 6 所示。数据库的实体表分为两类：系统级别、用户级别。使用 API 的有：线站 DAQ 系统、存储系统、数据传输系统、数据服务系统。

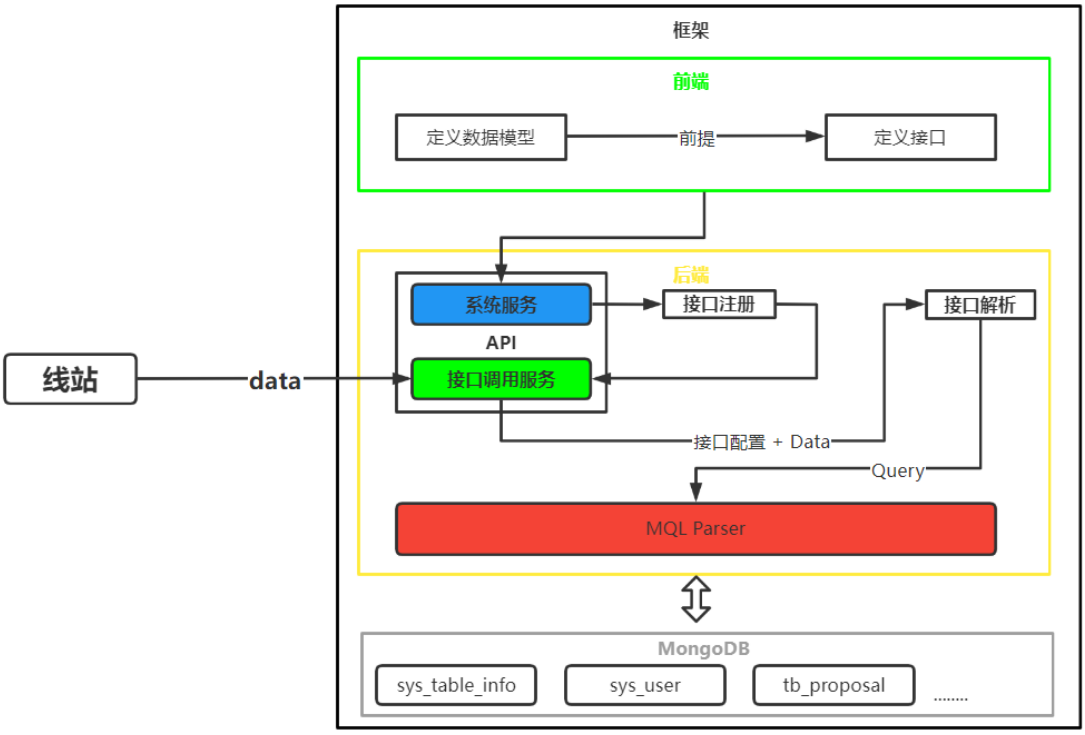

图 6　元数据管理软件处理流程

元数据管理软件框架架构如图 7 所示，架构采用弹性框架，具有可扩展性，不受数据类型和数据量变化的限制；使用 json 配置文件定义元数据模型，并且可由元数据模型直接生成 RESTful API，供其他系统调用；后端采用关系型数据库 Mysql 和非关系型文档数据库 MongoDB 结合方式，Mysql 负责存储结构化数据，包括系统级别的运行数据、用户登录数据、数据模型定义数据、动态接口对象数据等；MongoDB 负责存储业务数据（即实验元数据），它支持高并发元数据读写，同时数据结构灵活，能符合各线站元数据不一致的需求；同时该架构集成基于 web 的可视化界面，支持元数据检索功能，简化后期 web 界面开发过程。

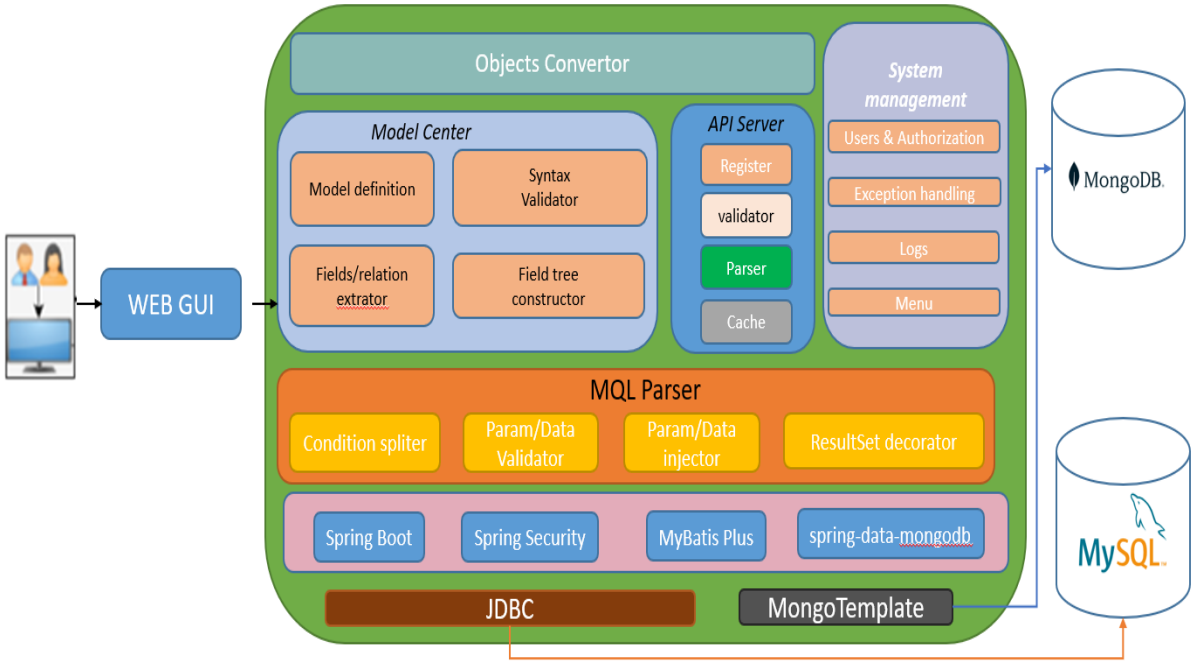

图 7　元数据管理软件框架架构图

Objects Convertor：配置转换器，负责将 WEB GUI 交互结果的 json 配置文件转换为 Java 实体对象。

Model Center：模型中心，负责 WEB GUI 用户数据模型的定义过程。包括模型的合法校验、字段项及字段树提取、模型关系提取、创

建模型、修改模型、删除模型、字段库检索、字段树检索、模型库检索。

API Server：动态接口服务，为 WEB GUI 用户提供可靠的动态接口配置服务，目前仅支持 post 请求方式的接口。用户在 WEB 界面为需要传输的数据模型创建动态接口，定义动态接口的过程主要包括接口注册、接口解析、接口校验、接口缓存。

System Management：系统服务中心，为整个系统的运行提供可靠的数据服务。系统级别的业务需要依赖 System Server，如用户管理、权限管理、登录管理、菜单管理、日志、异常处理、模型管理、接口管理等,这些系统业务所产生的数据存储在关系型数据库 Mysql。

MQL Parser：MongoDB 语句解析器，负责将接口对象解析为可执行的 MongoExecutor（MongoTemplate 组件能够将其转换为 MongoDB 语句）。接口被调用时，MQL Parser 在解析 API 对象过程中，会结合 API 对象的约束条件（如查询条件、分页、排序），将请求体 body 中的数据作为参数封装为 MongoExecutor 对象。

3 详细设计方案

3.1 元数据模型管理

元数据模型管理通过微服务的架构对科学数据进行全生命周期的管理。元数据管理软件框架使用关系型数据库 Mysql 记录元数据模型的结构信息，非关系型数据库 MongoDB 记录元数据。

3.1.1 元数据

元数据是科学数据管理的基础，在光源类设施的现实应用背景下，它包含了数据在连接科研活动方方面面的信息：从提案到数据发布，贯穿整个实验和数据分析过程。图 8 从逻辑上展示了科学元数据结构。

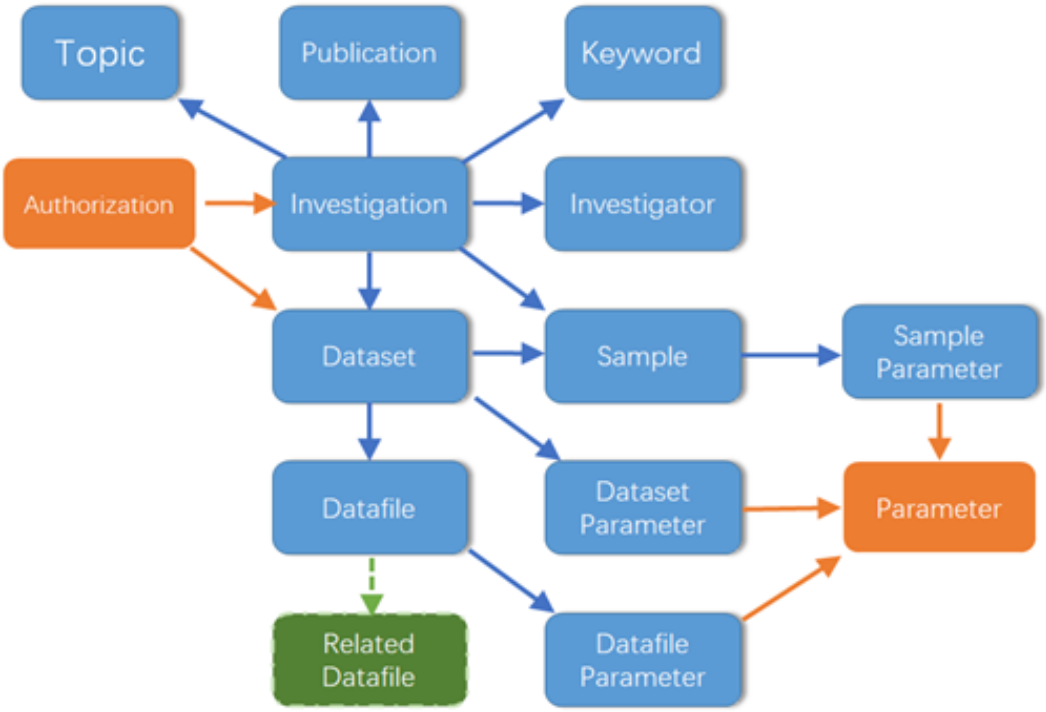

图 8　科学元数据结构示意图

从图 8 可以看出，科学元数据的结构围绕着“科学研究”而展开，一项“科学研究”关联着实验主题、实验数据集、实验样本、实验者和关键词、数据发布等信息；实验数据集关联样本信息、数据文件和

数据集参数；数据文件关联数据文件参数和其他相关数据文件。

如前文所述，元数据分为三类：管理元数据（administrative metadata）、科学元数据（scientific metadata）、其他元数据（other metadata）。管理元数据包括与数据相关的提案信息、用户信息、实验类型、线站信息等，来自提案系统和用户系统，会被存储到元数据库，用于元数据目录管理。科学元数据包括与数据相关的样本信息和实验环境参数等相关信息，从控制系统获取。由于科学元数据会被用于数据分析和数据目录索引，它需要进行两部分存储：与原始文件一起进行数据封装标准化，生成 HDF5 文件，同时一部分科学元数据需要被存储到元数据库。其他元数据主要来自传输或者分析应用，包括该数据传输中完整性校验信息（checksum）、分析软件名称版本信息、数据更新时间等，会被存储到元数据库，用于元数据目录管理。各类元数据获取来源详细示意如图 9 所示。

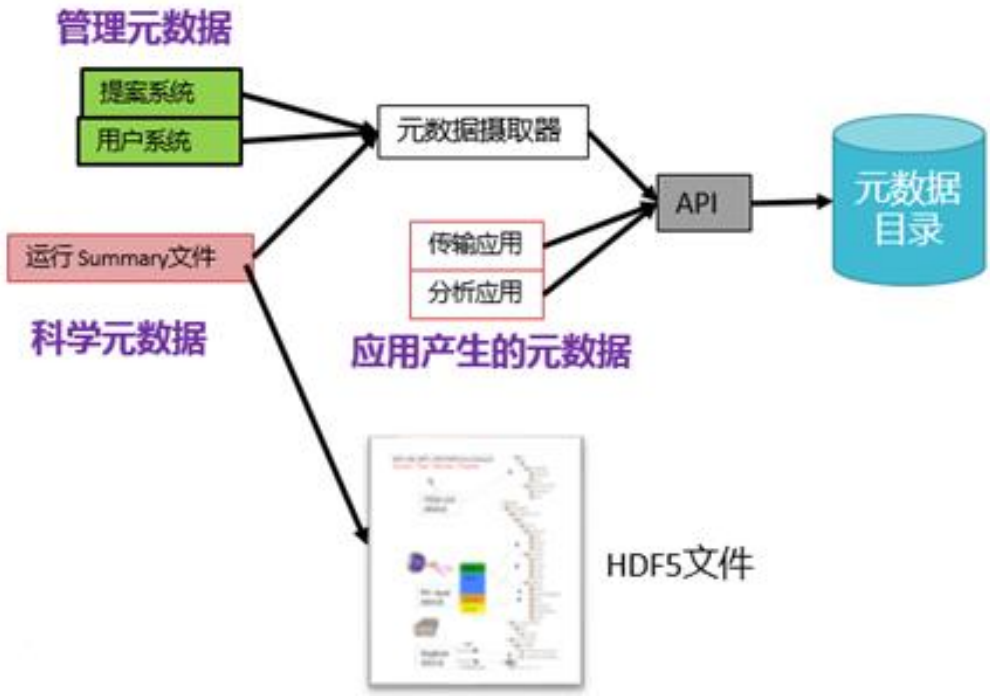

图 9　元数据获取来源

元数据管理软件框架提供基于 json 格式的元数据项和数据格式接口，分别接收元数据提取器、各类第三方应用中的元数据，具体如图 10 所示。

图 10　基于 json 格式的元数据项和数据格式

3.1.2 元数据模型结构设计

1）逻辑结构

元数据模型是对元数据特征的抽象，是一种面向用户、面向客观世界的模型，主要用来描述元数据的概念化结构。它由模型名称、模型属性、模型之间引用关系等组成。元数据模型逻辑结构如图 11 所

示。

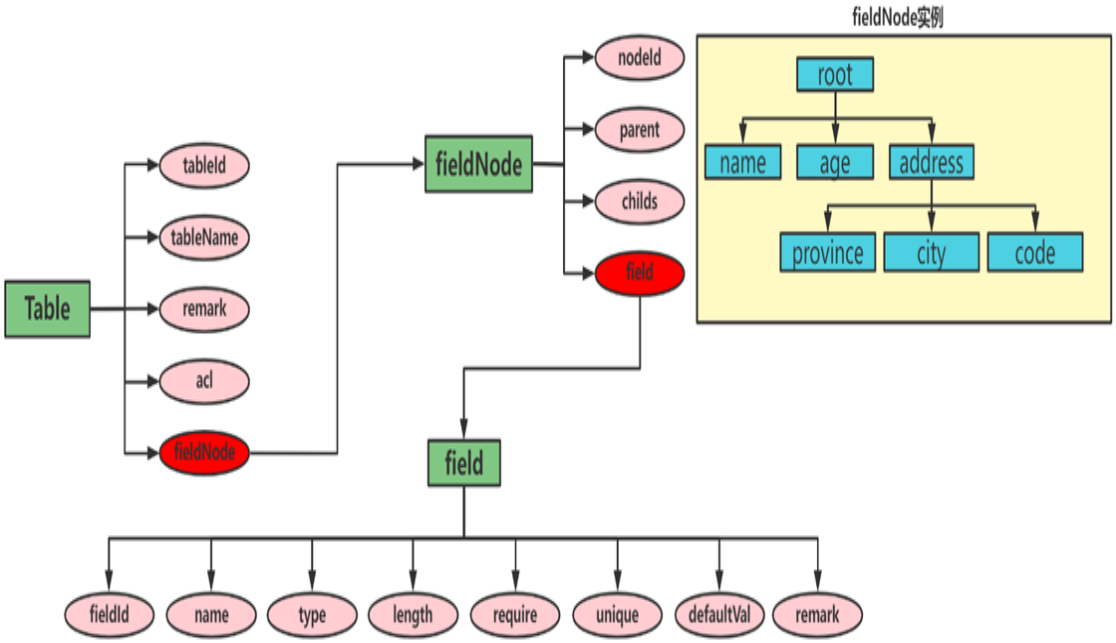

图 11　元数据模型逻辑结构

2）物理结构

id	model_name	field	remark
1	user	1	用户模型

id	parent_id	field_id	default_name
1	-1	-1	root
2	1	100	user_name
3	1	101	
4	1	102	

id	field_name	type	length	require	unique	default_value	remark
100	name	string	20	1	0		姓名
101	sex	int	1	0	0	1	1男0女
102	age	int	3	0	0		年龄

图 12　元数据模型物理结构

元数据模型的结构信息属于结构化数据，适合用关系型数据库进

行组织管理，如图 12 所示，利用关系型数据库存储元数据模型的结构信息，由模型表、字段项表、字段树表将模型的结构信息进行组织。模型表记录元数据模型的基本信息，字段项表记录每一个字段的详细信息，字段树表记录模型中字段之间的组织结构信息。

从图 12 可以看出，三张表可以确定一个数据模型的结构，字段项表记录了每一个字段的信息，字段树表的字段 field_id 可以关联字段项表的字段 id，parent_id 可以组织字段项之间的结构，从而形成字段树，一棵字段树有且仅有一个根节点，模型表字段 field 只需要记录这个根节点 id，能够实现模型和字段集的一一对应关系。

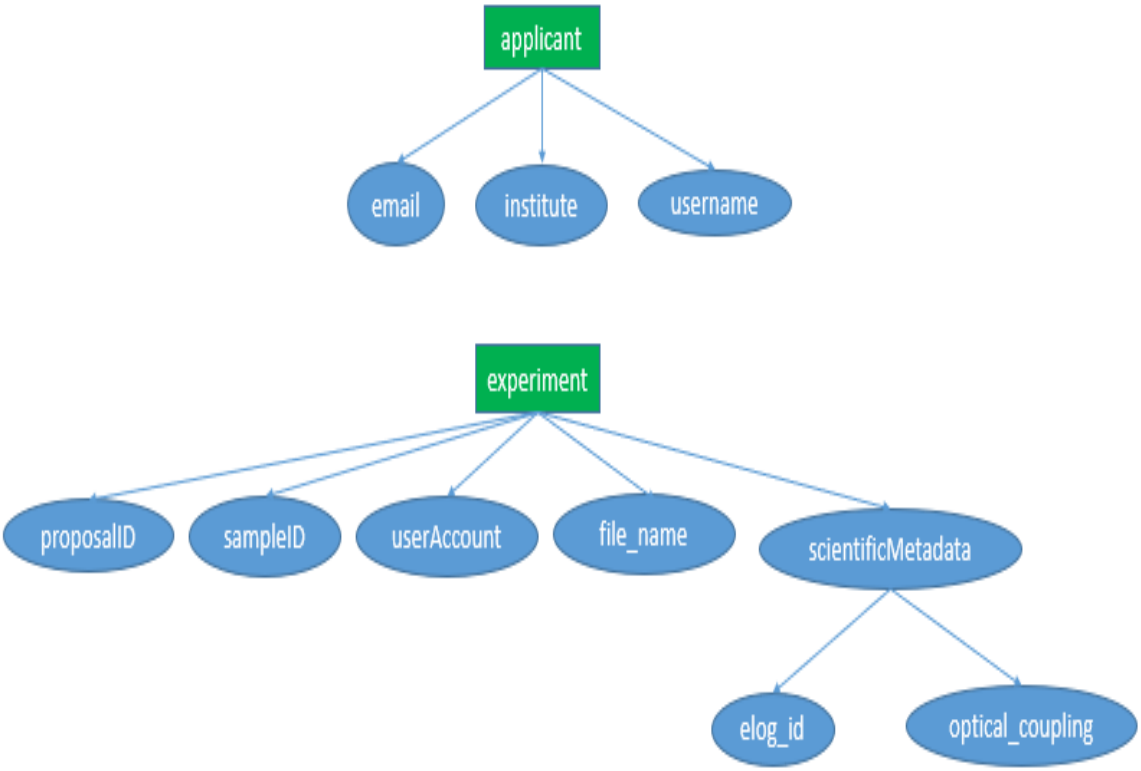

图 13　元数据模型的两类字段集结构

科学元数据按照 json 格式传输到元数据管理软件框架，不同类型的元数据结构存在较大差异。如图 13 所示，字段集结构可分为两类：线性结构的字段集、树形结构的字段集，元数据模型的字段集如果是线性结构，可以采用同级树形节点存储字段结构（即兼容关系型

数据库的字段结构)；如果是树形结构，可以采用多级树形节点存储字段结构（即兼容非关系型数据库的字段结构)。

3.1.3 元数据模型的定义

元数据模型的定义过程本质上是一个创建实体表的过程。用户通过 WEB 页面创建模型信息，前端负责将用户交互结果组织为 json 格式的模型定义数据（如图 14 所示)，后端接收到模型 json 数据后会依次进行配置校验、字段及字段树的提取、字段及字段树的存储、模型关系提取、模型创建。


```
{
    "modelName": "user",
    "remark": "描述说明......",
    "fields": [
        {"id": 1, "nodeType": 1, "parentId": -1, "fieldInfoId":-1, "defaultName": "root"},
        { ...
        },
        {"id": 3, "nodeType": 2, "parentId": 1, "fieldInfoId": -1, "defaultName": "address"},
        {
            "id": 4,
            "nodeType": 0,
            "parentId": 3,
            "fieldInfoId": -1,
            "defaultName": "",
            "fieldName": "province",
            "fieldType": "string",
            "length": 11,
            "isRequire": 0,
            "isUnique": 0,
            "defaultValue": "",
            "remark": "省份"
        },
        { ...
        },
        { ...
        }
    ],
    "relations": [
        {"type": 1, "modelId": 5, "fieldId": "6", "referedModelId": 5, "referedFieldId": "11"}
    ]
}
```


图 14　元数据模型的定义

1）配置校验

配置校验负责校验元数据模型定义配置是否具备合法性，如下图所示，校验的规则包含：①模型名称校验，是否命名合法、已经存在重名模型 ②引用字段项校验，字段集中引用的字段项 ID 是否真实存在 ③字段树结构校验，配置中组织的字段结构是否符合树形结构 ④

模型关系校验，模型关系中引用的模型 ID、字段 ID 是否存在。

2）字段及字段树提取

在元数据模型配置信息中，fileds 内容包含了数据项和数据项之间的结构信息，字段间的结构关系通过前端组件的序号方式进行组织，字段和字段树是模型建立的基础。因此，在框架创建模型前需要从配置信息中提取出所有字段项和仅有的一棵字段树。这个过程负责将 json 配置中的 fields 内容转换为内存中的字段项对象、字段树对象。

3）字段及字段树的存储

从元数据模型的物理结构可以看出，模型依赖于字段树、字段树依赖于字段项。因此，框架将字段项和字段树信息提取并转换为内存中的对象，为模型创建做好准备工作。为了组织字段项、字段树、模型之间的引用关系，必须先存储字段项，在字段项存储成功后返回对应的字段项 ID，再创建字段树，字段树需要引用字段项 ID，最后，字段树存储成功后会将其 ID 返回给模型，并与模型建立对应关系。

字段项存储时需要考虑字段重复冗余问题，按照框架的设计要求，字段名称、类型、长度、是否必填、是否唯一、默认值均不同，才能保证两个字段项不同。如图 15 所示，字段名称都为 name，但由于字段长度不同，字段 1 和字段 2 属于两个不同字段项。

图 15　不同字段项的比较

4）模型创建

字段和字段树创建成功后会返回字段树根节点的 ID，模型只需要关联字段树根节点 ID，即可对应本模型的数据结构信息。

5）模型关系提取

元数据模型之间存在主外键关系，用户在创建模型时会考虑是否本模型存在外键，主外键关系通过模型配置中的 relations 内容进行约束，约束内容包括：关系类型（1：表示主外键关系 2：表示继承关系，暂时只考虑主外键关系）、被参照模型的 ID、被参照模型的字段 ID、本模型的参照字段 ID。模型创建成功后会返回模型 ID、字段 ID，框架从配置中提取出 relations 内容，将模型 ID、字段 ID 绑定到模型关系对象，存入模型关系表。

3.1.4 元数据模型创建过程

元数据模型的创建过程如图 16 所示，经过配置校验、字段提取、字段创建、提取字段树结构、创建字段树、提取及创建模型关系等阶段，直至创建模型。

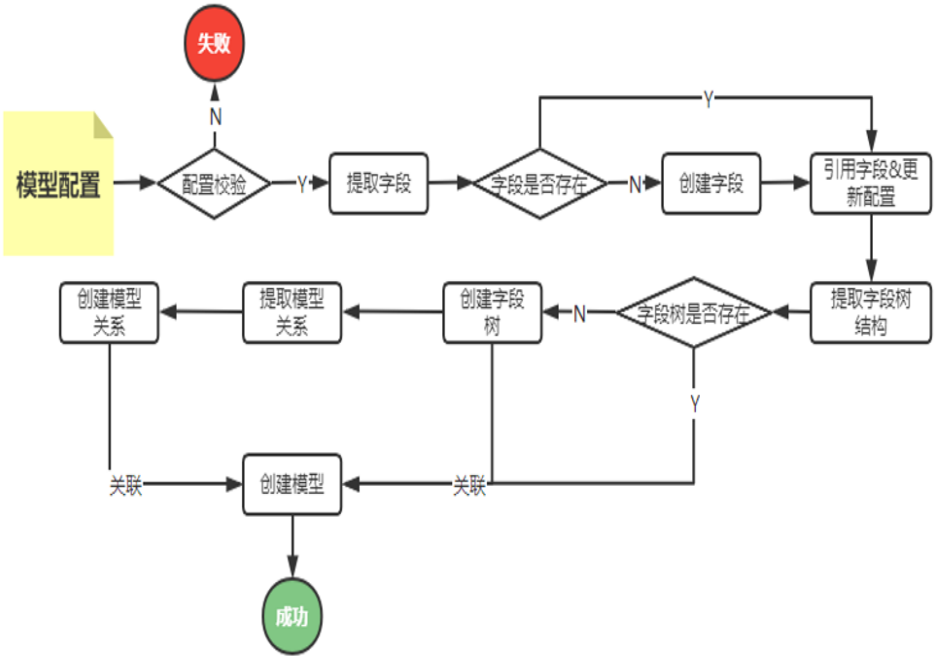

图 16　元数据模型创建过程

3.2 接口管理

元数据管理软件框架是数据处理平台的中枢大脑，平台内所有系统将元数据按照 json 格式转发到元数据管理软件框架，框架对所有元数据进行组织、管理，从而提高科学实验数据的利用率，提供更加高效、便捷的数据管理服务。框架基于微服务架构对外提供接口服务，由于数据源种类多、元数据项具有多样不确定性，传统的静态接口（硬编码方式创建固定接口）难以适应用户需求，用户希望基于配置、零代码的方式去创建接口，为数据管理过程提供高效、便捷的接口服务。

1）静态接口与动态接口

目前 WEB 应用基本上都是采用前后端分离的方式，前端负责在页面渲染后端交互的数据，后端负责为前端提供数据接口，数据接口是需要后端开发人员进行编码完成，重启服务后才能生效，即静态接口；动态接口是基于用户对框架需求提出的新概念，即用户可以在前端仅通过页面交互方式创建一个接口，接口的创建过程由框架自动完成，不再需要开发人员进行编码实现。

2）接口注册

接口是操作数据的工具，数据需要依赖于接口才能进入数据库，在注册接口时必须先确定当前接口需要操作的数据模型对象。接口注册是将接口配置信息（如图 17 所示）存入系统数据库的过程，即创建接口。接口创建成功后会返回接口 ID，接口 ID 能唯一确定一个接口对象。图 18 是一个创建动态接口的过程，通过前端配置即可迅速创建动态接口。

```
{
    "title": "添加用户接口",
    "opType": "get", //接口数据操作类型 add update delete get
    "httpType": "post", //post get
    "modelIds": [11,13], 11是user的id 13是dept的id
    "condition": "user.dept_id = dept.id and user.age > 20",
    "sorts": "age: -1, id: 1",  // -1降序  1升序
    "status": 1, //接口状态：1正在使用 0禁用
    "remark": "这个接口是用来添加新用户",
    "createUser": 100000
}
```

图 17 接口配置文件

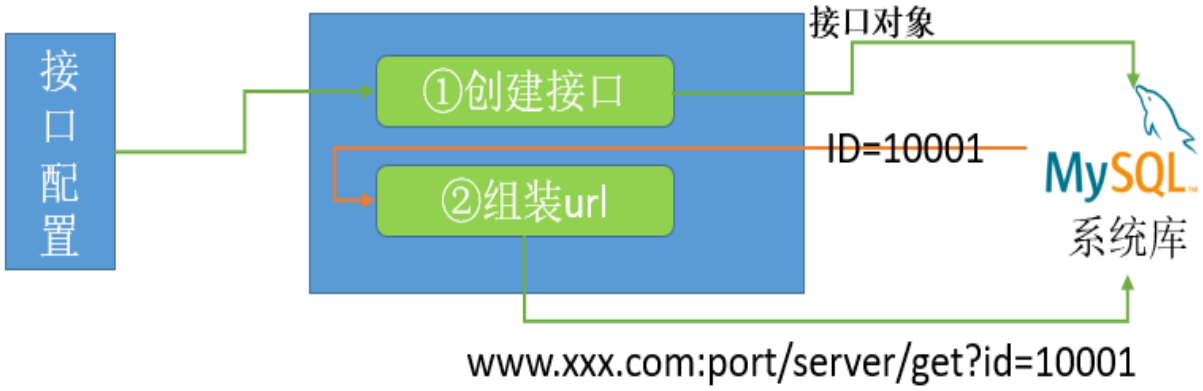

图 18 动态接口注册过程

3）接口校验

接口校验是对接口配置文件的合法性进行校验，对接口配置进行的合法性验证过程。触发时机：创建或者修改接口对象，接口配置文件中存在引用模型 ID、操作条件 condition、排序设置 sort、创建接口用户等信息，在接口创建之前需要对这些数据进行合法校验。

4）接口解析

接口注册成功会立即生成 url 地址，供调用者使用，调用者按照接口的说明（数据开发管理人员约定）进行接口调用。下图所示，接口解析是将配置信息转换为可执行对象的过程，可执行对象本质上是存储配置信息的数据结构（从数据库中把配置信息封装到内存对象中），它存储在内存中。接口被频繁调用的情况下，只需在创建接口时解析接口，生成可执行对象，调用接口时不需要执行接口解析操作。接口解析过程如图 19 所示。

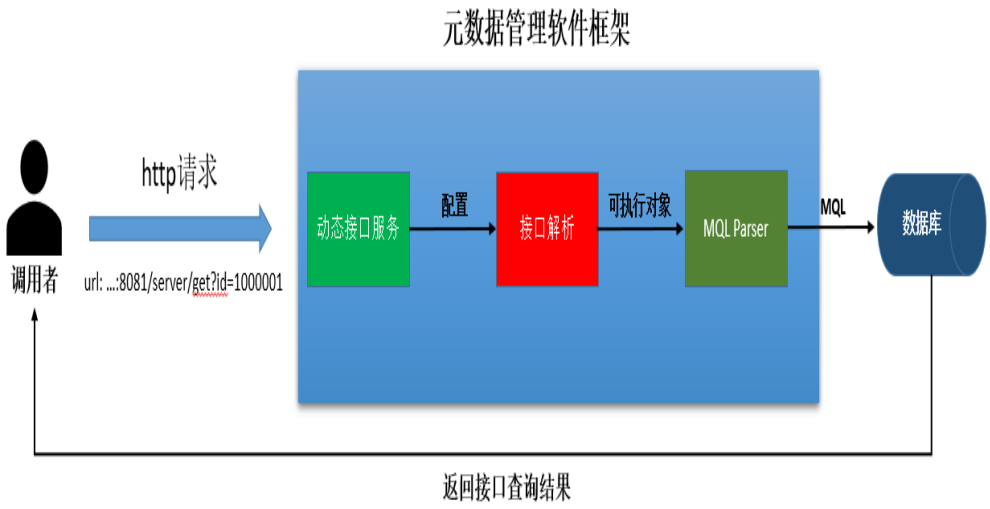

图 19　动态接口解析过程

5）数据校验

接口按照数据操作方式可以分为增加、删除、修改、查询四类，对于查询和删除类型的接口，数据模型对应的 MongoDB 集合不会受到脏数据的污染；对于增加和修改类型的接口，元数据需要校验通过后才能入库，否则会破坏数据的一致性。数据校验规则策略如图 20 所示。

获取接口中操作的数据模型结构，按照数据模型中字段的存储结构遍历，对比数据实例

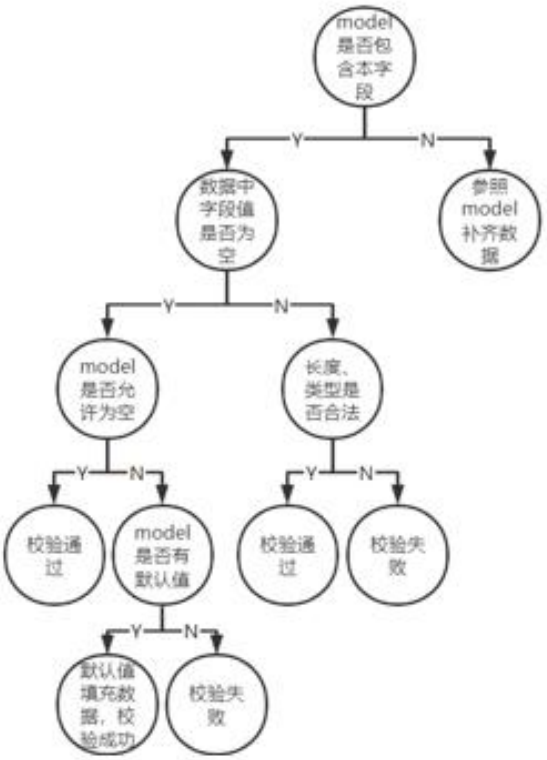

图 20　数据校验策略

4 开发与实现建议

开发与实现过程基本技术方法和路线：通过对现有系统分析，结合用户及业务需求，进行系统需求调研；对框架架构、功能等进行系统性设计；对框架部分进行编码实现，完成相关功能点；分别在本地环境和测试床环境完成相关测试；对框架进行部署，并与其他子系统进行联调。

项目的开发过程应包括：通过对现有系统分析，结合用户及业务需求，进行系统需求调研；在对框架架构、功能等进行系统性设计等工作的基础上，框架开发与实现将在框架设计方案的基础上进行有序推进。

鉴于用户需求、系统功能、相关环境、接口、关联关系等存在调整变化因素，系统需求分析及设计也将随之调整，但整体目标将围绕有利于整个系统的开发和使用进行增量迭代式展开和推进。