

数据组件：一种面向数据跨域流通可信抽象和封装方法

林绍福¹, 李云江¹, 王宇航², 贾晓丰²

(1. 北京工业大学 计算机学院, 北京 100124; 2. 北京市大数据中心, 北京 100193)

摘要: 数据作为关键生产要素, 具有自由流动、动态组合、跨域融合和使用增值等特征, 但在跨域流通、数据孤岛连通与标准化处理等方面仍面临挑战。从软件系统视角提出数据组件抽象模型和封装方法, 将数据要素抽象为独立、轻量、可复用的最小单元即数据组件, 具备可寻址、可交换、可操作与可管控特性; 设计实现数据组件注册和发布机制, 以规范化协议支撑数据组件发现与互操作, 提供统一描述、封装发布、发现获取、动态组合与可信交互等能力, 支撑数据要素跨域可信高效流通。以健康养老数据共享场景为例, 设计实现数据组件共享原型系统进行验证, 实验表明, 相较传统微服务模式, 跨域传输成功率提升 13%, 访问成功率提升 11%, 检索命中率提升 15%, 调度成功率提升 20%。

关键词: 数据组件; 数据解耦; 可信抽象; 标准化封装; 互操作机制

中图分类号: TP311.13; TP393.09 **文献标志码:** A **DOI:** 10.19358/j.issn.2097-1788.2026.04.003

中文引用格式: 林绍福, 李云江, 王宇航, 等. 数据组件: 一种面向数据跨域流通可信抽象和封装方法[J]. 网络安全与数据治理, 2026, 45(4): 17-26.

英文引用格式: Lin Shaofu, Li Yunjiang, Wang Yuhang, et al. Data components: a trusted abstraction and encapsulation method for cross-domain data flow[J]. Cyber Security and Data Governance, 2026, 45(4): 17-26.

Data components: a trusted abstraction and encapsulation method for cross-domain data flow

Lin Shaofu¹, Li Yunjiang¹, Wang Yuhang², Jia Xiaofeng²

(1. Faculty of Information Technology, Beijing University of Technology, Beijing 100124, China;

2. Beijing Big Data Center, Beijing 100193, China)

Abstract: Data, as a key production factor, has characteristics such as free flow, dynamic combination, cross-domain integration, and value-added utilization. However, it still faces challenges in cross-domain circulation, connecting data silos, and standardized processing. This paper proposes a data component abstraction model and encapsulation method from a software system perspective, abstracting data elements into independent, lightweight, and reusable minimum units—data components—possessing addressable, exchangeable, operable, and manageable characteristics. A data component registration and publishing mechanism is designed and implemented, using standardized protocols to support data component discovery and interoperability, providing capabilities such as unified description, encapsulation and publishing, discovery and acquisition, dynamic combination, and trusted interaction, supporting the reliable and efficient cross-domain flow of data elements. Taking a health and elderly care data sharing scenario as an example, a prototype system for data component sharing is designed and implemented for verification. Experimental results show that compared to the traditional microservice model, the cross-domain transmission success rate increases by 13%, the access success rate increases by 11%, the retrieval hit rate increases by 15%, and the scheduling success rate increases by 20%.

Key words: data component; data decoupling; trusted abstraction; standardized encapsulation; interoperability mechanism

0 引言

互联网以开放式体系结构和标准化协议在物理网络之上构建了一个由主机组成的虚拟数据传输网络, 从而屏蔽了下层物理网络的通信细节, 在异构的物理

网络之上形成了全球一体化的网络空间。随着互联网的发展和大数据时代的到来, 数据成为互联网中最具价值的核心资源。然而, 在现有架构中, 数据高度依赖于具体的应用系统和底层平台, 未能实现真正的解

耦, 导致数据在访问、共享和流通过程中面临诸多障碍。数据不解耦造成的问题主要体现在以下几个方面:

(1) 数据高度分散在各类应用系统中。在当前复杂的数据生态中, 数据分散存储与异构化是难以完全消除的客观存在, 传统的数据管理机制使数据在跨系统调用时存在反复的格式转换和接口适配, 使得数据的检索、流通和访问使用效率较低。亟需更加优化、高效的解决方案通过标准化的封装与接口机制, 来降低跨系统数据访问与整合的开销。

(2) 数据结构、语义和接口标准高度异构。在不同应用场景下, 数据格式多样、语义难以统一、接口规范缺失, 使系统之间的互操作性始终面临挑战。当前主流的数据管理机制依然存在因异构造成的效率瓶颈, 需要研究如何有效化解数据异构性衍生的数据互操作效率。

(3) 数据深度依赖具体的计算任务和运行环境。数据的这一客观特性, 使得数据仅在计算任务生命周期内是有效“存在”和使用的, 而原本零散、短暂的数据难以被追溯和利用, 需要研究如何降低数据与计算任务和运行环境的依赖性, 以实现数据跨任务生命周期、跨系统环境的有效流转和利用, 提高数据可追溯性和利用效率。

(4) 数据紧耦合于封闭系统, 外部主体难以有效访问或调用。在面向对象、面向服务等解耦和封装技术基础上, 如何更加有效地降低跨系统、跨行业数据复用的难度, 使数据更易发现、获取和利用, 对于充

分释放数据要素价值、发挥赋能作用具有重要意义。

因此, 为实现数据在互联网空间中的高效流通与治理, 必须从根本上改变“数据附属于计算”的技术逻辑, 推进数据的抽象、封装与解耦。由此, 数据空间应运而生, 作为互联网架构从“以计算为中心”向“以数据为中心”演进的关键形态, 其核心在于将数据从计算过程的附属角色中解放出来, 确立其作为一阶实体的独立地位。围绕这一理念, 数据不再仅作为计算任务中的输入输出, 而是具备自主标识、独立生命周期和标准交互能力的基础资源。数据解耦的推进过程具有阶段性特征, 主要经历了三个发展阶段, 每个阶段都对应着不同的目标、抽象方式和技术机制。第一次解耦是数据与应用程序的解耦, 其目标是屏蔽数据访问的复杂性, 降低应用系统的开发门槛, 该阶段数据基本抽象是 ER 模型, 核心系统形态有数据库和数据仓库。第二次解耦是数据与业务系统的解耦, 目标是屏蔽数据汇聚分析的复杂性, 降低企业级系统的开发门槛, 数据基本抽象是“键-值对”KV 模型, 核心系统形态有数据湖。第三次解耦是数据与生产和消费主体的解耦, 目标是屏蔽数据流转与使用的复杂性, 降低数据要素社会化供给、流通与应用的门槛。此次解耦的相关理论和技术还在研究发展中。

基于以上问题以及数据解耦的发展, 相关组织及学者开展数据要素基础设施技术研究, 旨在探索出一条安全、高效的国家数据流通利用基础设施的有效技术路径。当前正在研究的主要技术路线见表 1。

表 1 技术路线综合分析表

技术路线	技术原理	重点、难点	关键技术	适用场景
数场 ^[1]	构建统一的数据流通利用基础设施	数据分散存储于多环境, 技术栈庞杂, 整合成本高	统一身份登记、标识与接口技术	广域数据流通、低门槛接入场景
可信数据空间 ^[2]	构建互联数据生态系统, 同时基于使用控制技术保障数据的访问、共享和使用权限	技术供给不足、标准规范难统一、商业模式未成熟	访问控制技术、分布式架构技术	跨组织/行业数据安全流通 (如金融数据汇聚、跨境数据监管)
数据网 ^[3]	在不可信的互联网之上, 基于专有通信协议用于企业通信的一种网络。是实现可信可控数据互联互通、数据快速检索发现的新型基础设施	标识技术栈繁多, 解析互通性差; 垂直节点未完全互通	数据对象接口协议、数字对象寻址协议	数据注册、发现与搜索场景
数据元件 ^[4]	对数据脱敏处理后, 根据需要由若干字段形成的数据集或数据的关联字段, 通过建模形成的数据特征	确权、定价、计量难题与安全风险	不可逆算法安全技术、规模化加工技术	需安全且规模化加工的场景
数据件 ^[5]	通过标准化封装异质多源数据的语义、结构与操作, 实现数据要素与主体、应用解耦	异质多源数据的标准化封装与跨系统兼容性保障	元数据智能映射、动态适配接口、轻量化封装协议	数据跨主体/系统高效流转与使用场景 (如社会化供给、数据场系统)

通过对万维网^[6]和数字对象架构（Digital Object Architecture, DOA）的分析，面向数据要素治理应用场景和数据加工利用的第三次解耦趋势，本文以构建数据要素高效流通的基础理论模型和技术机制为中心，聚焦数据组件的理论模型构建和技术实现，厘清其作为数据要素流通最小单元的基本特征和可寻址、可交换、可操作与可管控等关键特性，开展数据组件模型、标识机制、互操作协议及注册发布发现等核心机制的研究，旨在建立一套具有规范性、可扩展性和可操作性的技术框架，实现数据组件在多主体、多系统间的发现、获取、流转与共享利用，推动数据要素在可信环境下高效、安全、有序地流通。

1 数据组件模型

1.1 数据组件的理论基础

从计算机体系结构视角看，冯·诺依曼体系以数据、运算和控制的紧耦合为核心，数据流动受限于指令线性过程；数据库三级模式首次实现逻辑与物理分离；GPU、存算一体芯片等进一步弱化数据与计算的绑定。数据组件借鉴内存分段与虚拟化技术，将数据封装为独立的可寻址单元，实现与计算逻辑的动态绑定与解绑，为数据自由流动奠定基础^[7]。

在计算机网络视角下，OSI/TCP-IP 模型通过协议分层推动数据跨域传输，IP 分组的头部/载荷分离及 HTTP/URI 机制实现资源标识与数据内容的解耦，NDN 等内容中心网络以数据命名替代主机寻址，强化数据独立性。数据组件继承网络分层思想，采用唯一标识与解析协议，支持跨域数据寻址与动态路由，形成与网络协议兼容的封装范式^[8]。

从软件系统演化看，数据与处理逻辑的解耦不断深化：从面向过程的数据和函数分离，到面向对象的对象封装，再到组件化和微服务以标准接口实现跨系统协作。数据组件继承“高内聚、低耦合”原则，通过标准化接口与元数据描述，将数据封装为可独立部署、动态组合的原子单元，推动数据从绑定于过程、对象到完全自治的范式转型^[9]。

面向对象思想与数字对象架构为数据组件的结构化设计提供方法论。封装与继承机制将数据属性与操作整合为对象，Dublin Core 元数据模型构建分类体系；DOA 进一步将数据抽象为“标识—元数据—数据体”三元组，通过 Handle 系统实现全球唯一解析^[10]。数据组件扩展此框架，增加治理维度，在兼容既有标准的同时满足数据要素治理中的合规性需求。

软件组件与微服务架构直接指导数据组件的工程

化实现。基于接口定义语言（Interface Definition Language, IDL）定义输入/输出规范，依赖注入与服务发现实现运行时动态绑定。

通过从计算机体系架构、网络架构以及软件设计等多角度，探讨数据与计算解耦的理论与实践，结合对传统紧耦合模式及解耦理论和技术的分析与借鉴，提出“数据组件”这一新的抽象封装方式，基于现代计算架构，设计和实现灵活的数据描述与封装，并通过标准化接口与动态组合机制，实现跨主体、跨平台、跨系统的数据互操作，支撑数据资源化、要素化的管理和开发利用。

1.2 数据组件的定义和封装规则

数据组件是数据要素流动与使用的基本单位，通过对异质多源数据的语义和结构等进行标准化封装，使得数据要素与数据主体^[11]、数据应用“解耦”，让数据在不同主体、不同应用系统间高效地流转与使用。

从性质与目标来看，数据组件不仅是数据封装的技术单元，更是数据治理与价值释放的操作载体。通过对异构数据的语义、结构与操作方式进行标准化处理，数据组件使数据具备可识别、可管理和可操控的特性，满足“可用、可控、可信、可交易”的要求，从而为数据要素作为新型生产要素的实现与流通奠定机制基础。

在外部表现层面，数据组件通过标准化接口体现封装边界，支持查询、调用、更新、组合等操作模式，为其与数据平台、应用系统及其他组件的互操作提供统一入口。内部结构上，数据组件由标识、描述、数据路由与接口组成，形成具备自描述与服务能力的数字实体单元。元数据和权属描述用于记录组件属性、使用边界，并支撑其发现、检索与生命周期管理。

数据组件具备良好的可扩展性与多层级功能体系，如图 1 所示。通过对异构多源数据进行语义、结构与操作的标准化封装，形成边界清晰的数据单元，并提供注册、调用与组合等统一操作；在此基础上支持预处理与质量校验等扩展能力，进一步实现组件的动态组装、按需复用与跨域流通，支撑数据要素的可信治理与高效利用。

在封装方法上，数据组件遵循统一模型与技术规范，通过语义一致性建模、功能抽象接口与生命周期治理实现语义、行为与权属的一体化绑定，保障数据组件的自治性与可控性。相较之下，第一次解耦以结构化模型实现数据与应用分离，侧重静态存取管理；第二次解耦以数据湖为代表，实现数据与业务逻辑分

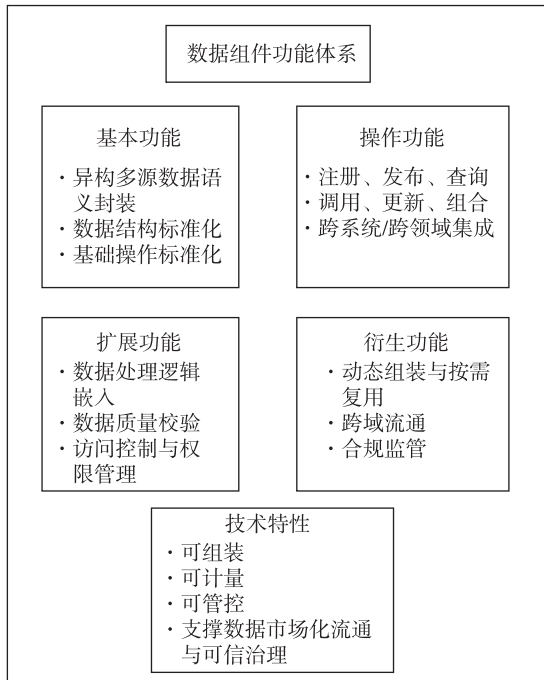


图1 数据组件多层次功能体系图

离,引入面向对象的数据抽象,支持跨场景复用并初步体现语义多样性。

数据组件将数据从计算任务的附属角色中解放出来,并赋予其独立的标识,确保数据能够在不同系统间进行自由交互与组合。这一方法的根本区别在于数据不仅是存储与访问的对象,更是可以作为独立资源进行管理、调用和复用的实体。

第一、二、三次解耦封装方式的分析概要见表2。

1.3 数据组件描述模型

数据组件的描述模型可以从标识、描述、数据路由、接口四个维度进行设计,同时结合可组装、可计量、可管控的特性,构建一个完整的规范化描述框架,如图2所示。

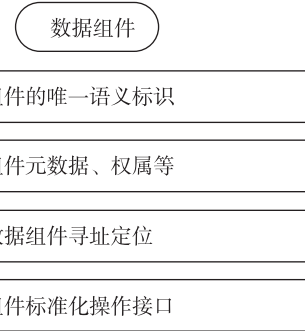


图2 数据组件标准化模型

“标识”部分为数据组件提供全局唯一的语义化标识,确保其在流通与使用中的可识别性;“描述”部分通过元数据信息和权属关系详细记录数据组件的基本属性,支持数据组件的发现与管理;“数据路由”定义数据组件在网络空间中的定位、寻址与访问路径,支撑数据流通;“接口”部分提供规范化的操作接口,支持数据组件的查询、更新、删除等操作。这四个维度共同构建了一个完整的数据组件标准化封装框架,确保数据组件在发现、获取、流转、共享和利用的一致性、可操作性与安全性。

数据组件具备可组装、可计量、可管控三项核心特性:通过接口兼容与依赖约束支持组件的动态组合与复用;通过使用计量模型与智能合约实现按需计费与结算;通过访问控制与生命周期策略实现全流程安全管控,从而保障数据流通与使用的灵活性、透明性与安全性。

基于数据组件标准化模型的设计,一个标准化的数据组件可以采用如图3所示的JSON格式进行封装设计,涵盖标识、描述、数据路由、接口四个核心维度,并体现可组装、可计量、可管控的特性。

表2 三次解耦封装方式的分析要点

解耦封装阶段	目标	数据关系	封装形式	主要特征
第一次解耦封装	屏蔽数据访问的复杂性,减少开发负担	数据与应用程序紧耦合	静态结构设计,如ER模型	数据存储与访问分离,应用程序简化
第二次解耦封装	数据与业务逻辑解耦,支持多元应用场景	数据与业务系统解耦,支持多种数据结构和语义	面向对象的抽象,如数据湖	支持不同数据源的灵活集成与流动
第三次解耦封装	实现数据自治,支持跨系统、跨域、跨主体流通	数据与计算、平台、生产流通消费主体、要素治理全生命周期应用场景解耦,具备独立性	独立、轻量、可复用的最小单元,具备标准接口与独立能力	数据具备可组合、可追溯、可治理特性,支持动态绑定与解绑,实现跨平台、跨域、跨主体的自主流通与可信协同

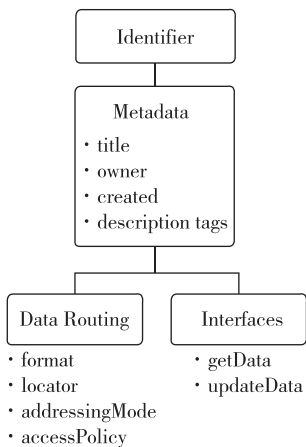


图3 数据组件 JSON 格式

这个 JSON 格式包括：

- (1) identifier：组件唯一标识；
- (2) metadata：基本元数据描述；
- (3) dataRouting：支撑高效定位与流通，包含位

置定位、访问策略、存储映射、寻址规则；

(4) interfaces：提供数据访问的 API。

1.4 数据组件与其他相关概念的辨析

数据组件与微服务、数据湖等现有数据治理技术存在本质差异，见表3。相较于微服务聚焦业务逻辑的拆分与协同^[12]，数据组件以数据要素为核心，通过权属管理、计量模型等治理维度的封装，实现数据作为资产的独立流通，其创新并非替代微服务，而是在数据要素化场景中补足现有技术栈缺失的“数据治理语义层”。相较于数据湖^[13]，数据组件则通过标识解析与元数据标准化，构建跨域数据的语义一致性与自治能力。相较于数据即服务（Data as a Service, DaaS）^[14]，其突破了“服务依赖特定平台”的局限，通过 Handle 标识与 DOIP 协议（数字对象接口协议）^[15]实现跨生态的可信交互，实现数据与平台和系统的松耦合。

表3 数据组件与现有数据治理技术区别

对比维度	微服务 (Microservices)	数据湖 (Data Lake)	DaaS (Data as a Service)	数据组件 (Data Component)
核心对象	业务功能/服务	数据集合	数据服务接口	数据要素本身
关注重点	业务拆分与协同	异构数据集中存储	平台化数据供给	数据治理与流通语义
解耦对象	服务与服务	数据与分析工具	服务与调用方	数据与平台/应用/主体
标识与寻址	服务实例地址	表/文件路径	平台内部接口	全局唯一标识 (Handle)
跨域互操作	有限 (依赖网关)	较弱	受平台限制	强 (DOIP + 标识解析)
治理能力	不涉及权属	存储级治理	平台策略	权属、计量、追溯、管控
数据流通形态	隐式	集中式	平台内服务	去中心化、可组合流通

2 核心机制设计

2.1 数据组件核心机制组成框架

如图4所示，数据组件核心机制框架采用分层化设计，由封装建模、注册与标识、访问与互操作以及治理与管控四个层次构成。通过标准化封装实现数据与平台、应用的解耦；通过注册与标识解析机制支撑数据组件的可发现与可定位；依托统一互操作协议实

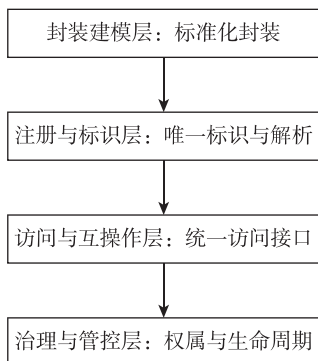


图4 数据组件核心机制框架

现跨系统透明访问；通过权属、计量与生命周期管理实现可管控与可追溯，形成支撑数据要素跨域流通与合规利用的基础技术体系。

2.2 数据组件注册索引机制

数据资源经规范化封装后需完成标准化注册，纳入数据组件注册库实现统一管理，这是数据组件实现可信流通与共享利用的基础。基于标准化描述模型的注册机制围绕“标识—描述—数据路由—接口”四个维度构建，通过为组件分配唯一标识并关联元数据，形成可检索的索引体系，以支持大规模场景下的高效定位与一致性管理。

在服务能力建设上，设计并实现面向第三方系统的数据组件注册接口，遵循 DOIP 规范，涵盖注册提交、元数据更新、数据组件注销、索引查询等功能。

2.3 数据组件标识解析机制

2.3.1 标识解析技术路线

数据组件在注册发布后，可通过静态或动态方式

被检索、定位与调用, 其中的核心在于标识解析机制。该机制以“标识即入口”为理念, 通过全局唯一标识实现数据组件在跨域环境中的唯一定位与动态访问。

每个组件在注册后获得唯一标识, 并与其存储位置、访问接口及元数据建立映射关系, 写入标识解析表以支持统一解析, 如图 5 所示。

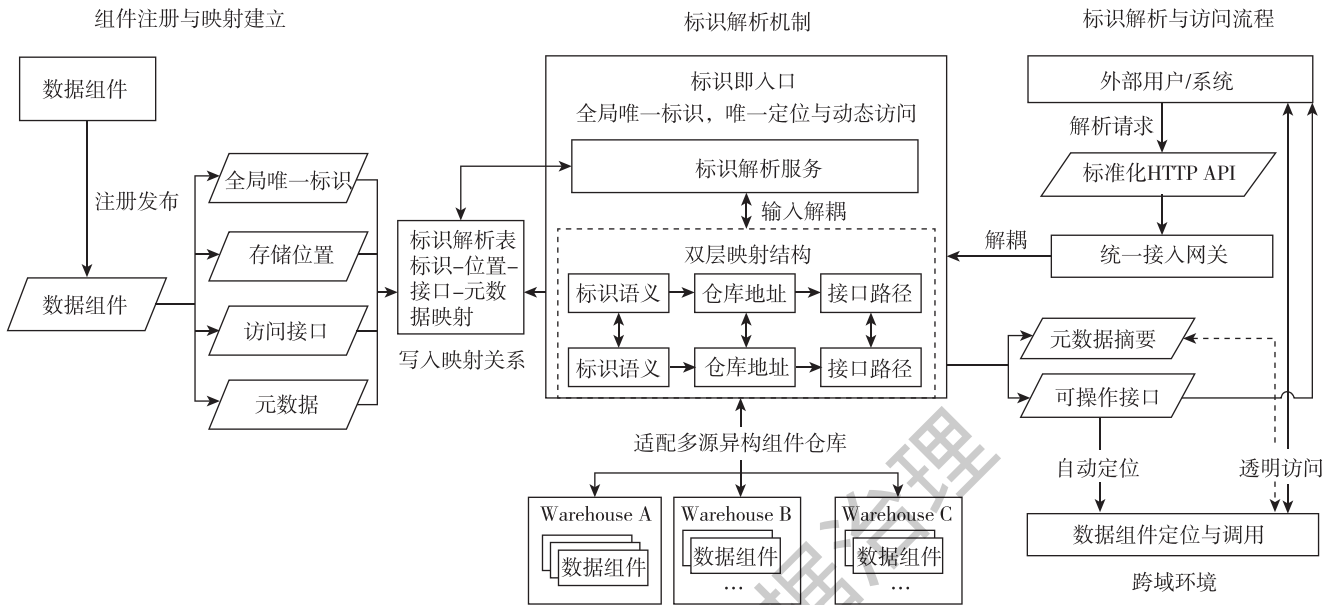


图 5 数据组件标识解析技术路线图

解析服务采用双层映射结构, 将标识语义与仓库地址、接口路径解耦, 使其能够适配多源异构的组件仓库。在接口层, 解析机制通过标准化 HTTP API 提供服务, 解析请求经由网关统一接入, 并返回目标数据组件的元数据摘要及可操作接口, 实现自动定位与透明访问。

2.3.2 解析机制架构

数据组件标识系统采用树状层级结构对标识进行组织与分配, 并向外提供统一的标识解析能力。体系由根节点、域节点和端节点构成。根节点位于最上层, 可由多个节点组成根节点网络, 负责管理下属域节点的增删改及其前缀、IP 地址、注册量、解析量等信息, 并与其他根节点同步前缀与地址数据。

域节点隶属于单一根节点, 负责管理其下的端节点, 同时需将端节点的标识注册量与解析量等运行信息上报至根节点。端节点作为叶子节点, 是标识注册、解析、更新与删除的直接执行者, 并承担访问权限管理与标识数据存储。所有标识解析请求最终由端节点响应, 完成标识到具体数据组件的映射与定位, 如图 6 所示。

3 基础协议架构设计与实现

3.1 基础协议架构设计与标准协议选择

标准化的协议是数据组件核心机制技术实现和访

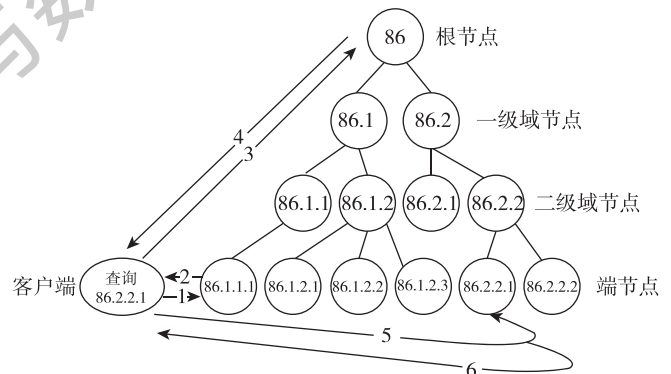


图 6 根节点标识解析策略路由流程示意图

问、使用数据组件的统一规则和约定。在生成数据组件之后, 使用 DOIRP 协议注册到数据组件解析系统, 使用 DOIP 协议注册到数据组件仓库, 并且该数据组件仓库还能够根据元数据进行搜索。DOIP 是现代、更全面的协议, 正在成为可发现、可获取、可互操作、可重用 (Findable, Accessible, Interoperable, Reusable, FAIR) 数据框架^[16]中重要的一环。而 DOIRP 更像是前置解析系统, 用于支持发现和定位。

表 4 展示了关于数据组件处理相关协议的对比, 涵盖了目前主流的几个协议, 包括 DOIP、DOIRP、Handle System、REST API^[17], 从多个维度来对比它们的设计目标、通信方式和适用场景。

表4 数据组件处理相关协议对比表

协议名称	全称	核心功能	通信方式	数据格式	典型使用场景
DOIP	Digital Object Interface Protocol	与数字对象进行交互 (读/写/执行)	TCP/HTTP + JSON	JSON/Binary	数据服务、FAIR 数据共享
DOIRP	Digital Object Interface Resolution Protocol	解析数字对象标识符, 返回元信息	HTTP/Handle Protocol	XML/JSON	标识符解析、发现对象位置
Handle	Handle System	提供唯一标识符解析服务	TCP + Binary Protocol	Binary	DOI 服务底层支持
REST API	Representational State Transfer	通过 URI 操作资源	HTTP	JSON/XML	各类 Web 服务、传统数据接口
GraphQL (可选)	Graph Query Language	客户端自定义查询接口	HTTP	JSON	灵活的数据访问和接口聚合

实现数据组件在跨平台、跨系统环境中的协同需求构建统一的基础协议架构。该架构由两类核心协议组成：基于 DOIP 的数据组件互操作协议，用于建立跨异构系统的标准化访问与语义一致性；基于 Handle 的标识解析协议，实现组件的唯一标识、定位与持久引用。

3.2 数据组件互操作协议设计与实现

数据组件互操作协议 (Data Component Interoperability Protocol, DCIP) 为分布式环境中的数据组件提供统一、可扩展的通信机制，以解决多源异构系统接口不一致与交互复杂等问题。DCIP 以数据组件为独立可寻址实体，通过标准化接口、操作语义与通信流程实现跨系统、跨域的无缝互操作，确立“以数据为中心”的交互范式，保障数据要素的自治流通能力。

在设计上，DCIP 构建面向数据组件的操作模型，涵盖访问、更新、查询与传输等核心指令；通过将数据组件抽象为具备唯一标识和元数据的数字对象，屏蔽底层存储差异，使调用方以统一方式实现跨系统交互。协议兼容多类型数据结构与元数据标准，并支持对多域、多架构环境的灵活适配。

在实现层，DCIP 以模块化方式嵌入数据组件服务框架，运行时由中间件完成消息解析、路由与响应，形成透明通信通道。协议同时集成身份认证与访问控制机制，在保证跨域互联的同时确保安全与合规，为数据要素治理提供关键的互操作支撑。

3.3 数据组件标识解析设计与实现

数据组件标识解析机制 (Data Component Handle Resolution, DCHR) 基于全局唯一、持久可用的前缀-后缀式标识体系，为每个数据组件赋予独立的可寻址与可解析能力。前缀表示注册命名空间，后缀对应具体对象，标识与元数据深度绑定，使解析结果同时携

带访问地址、版本、权属等治理属性，从而超越传统依赖路径或 URI 的单一寻址模式。

在解析架构上，DCHR 采用分布式部署与本地缓存相结合的方式，并通过与根节点的同步和多层映射结构实现跨域场景下的快速定位与灵活寻址。系统对外提供标准化 REST 接口，支持标识的注册、查询、更新与撤销，便于自动化集成与跨系统调用。

3.4 数据组件访问获取机制设计与实现

数据组件访问获取机制 (Data Component Access & Get, DCAG) 以统一标识和标准化操作语义为基础，为数据组件提供一致、可预测的跨域访问入口。不同于依赖平台服务的传统调用方式，DCAG 将数据组件视为独立可寻址资源，通过唯一标识与分层 URI 结构实现检索、注册与更新等全生命周期操作，使其具备动态可操作性。

在交互层，DCAG 通过协议化路由将外部请求映射至对应的组件操作模块，在统一接口下屏蔽底层存储与运行环境差异。组件的持久化访问由抽象层透明处理，使同一操作在异构系统中保持一致行为。所有返回结果均采用结构化响应形式，并结合可扩展序列化与字段级可见性控制，使调用方可根据权限与需求灵活获取不同粒度的数据内容。图7展示了 DCIP、DCHR、DCAG 三大机制的整体框架图。

4 实验验证

4.1 实验环境

为验证数据组件在标识唯一性、元数据描述、数据路由与接口统一性四个维度的技术可行性及其相对于传统紧耦合数据治理模式的性能优势，本文设计了对比实验方案。实验以智慧医疗健康科研数据共享场

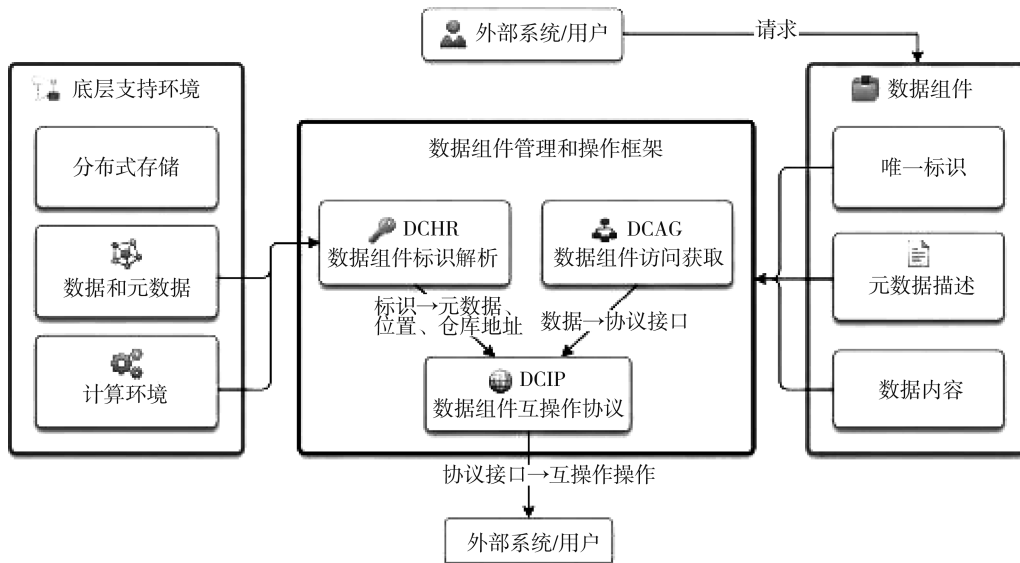


图7 DCIP、DCHR、DCAG 三大机制整体框架图

景为例，构建传统紧耦合模式与基于数据组件的松耦合治理模式两类仿真环境。

(1) 传统紧耦合模式环境：采用基于微服务架构的现代数据共享系统，使用 Spring Cloud 构建服务集群，集成 API 网关与分布式缓存，底层存储采用 MySQL 集群与 MongoDB 分片，模拟企业级数据中台的典型架构。数据访问通过服务间 RESTful API 交互，元数据采用自定义格式存储。

(2) 基于数据组件的松耦合模式环境：基于 MinIO 对象存储搭建数据资源池，辅以自建的数据组件注册服务与标识解析服务。各系统通过标准化 API 接口交互，支持数据的独立封装、统一注册、解析访问与动态调度，整体上构建了一个松耦合、自治化的数据要素流通环境。

(3) 硬件配置：实验组与对照组均部署于由 4 个节点组成的分布式集群，每个节点配置为 16 核 CPU、64 GB 内存、1 TB SSD 存储，排除硬件差异对实验结果的影响。

(4) 网络环境：模拟广域网与局域网场景，WAN 环境下网络带宽为 100 Mb/s，平均延迟 50 ms；LAN 环境下带宽为 1 Gb/s，延迟低于 1 ms。

(5) 基线系统：选择基于微服务架构 + API 网关的现代数据共享平台。

(6) 数据集：使用公开智慧医疗健康数据集，每个组件包含标识、元数据、原始数据及 RESTful API 接口。

4.2 实验结果与分析

数据组件模式通过标准化接口、解耦架构与优化

路由，显著提升了数据流通效率、访问性能、检索效率和调度能力，如表 5 所示。在数据流通效率方面，平均数据传输延迟由 1 200 ms 降至 350 ms，降低 70.8%，跨域传输成功率由 85% 提升至 98%，提升 13 个百分点；数据访问测试表明，平均访问响应时间由 950 ms 降至 230 ms，降低 75.8%，访问成功率由 88% 提升至 99%，访问失败率降至 1%；在数据检索方面，平均检索时间由 800 ms 降至 220 ms，降低 72.5%，检索命中率由 82% 提升至 97%；数据调度能力方面，平均调度时间由 1 000 ms 降至 320 ms，降低 68%，调度成功率由 75% 提升至 95%，调度失败率降至 5%。这些数据表明，数据组件模式通过技术创新有效提升了数据流通的效率、稳定性与可扩展性，推动了跨域数据的可信流通。

数据组件模式在数据要素可信流通中的关键优势机理体现在其通过标准化接口与解耦架构实现了高效的数据流通。通过统一的接口与标识解析，减少了跨域传输的适配需求，降低了延迟并提高了成功率。解耦架构使得系统间的依赖关系最小化，从而减少了访问和检索过程中的延迟和失败率。此外，数据组件模式通过优化的路由与并行处理技术，显著提高了数据调度的稳定性与效率，确保了在高负载条件下的可靠性。整体而言，数据组件模式通过技术创新实现了跨系统、跨平台的数据流通优化，保证了数据在不同环境中的高效、稳定与可信流通。

在智慧医疗与政务数据共享场景中，传统模式因语义冲突、接口不兼容与权限模型不一致而导致高失败率：38% 源于元数据语义差异，27% 来自接口版本

表5 实验结果分析表

评估维度	关键指标	传统紧耦合模式	基于数据组件的松耦合模式	性能提升/变化	核心优势体现
数据流通效率	平均数据传输延迟/ms	1 200	350	降低 70.8%	标准化接口减少转换开销，标识解析优化路由
	跨域传输成功率/%	85	98	提升 13 个百分点	统一的认证与解析机制增强稳定性
数据访问性能	平均访问响应时间/ms	950	230	降低 75.8%	接口标准化与解耦架构降低访问复杂度
	访问成功率/%	88	99	提升 11 个百分点	有效解决接口版本不兼容与认证失败问题
数据检索效率	平均检索时间/ms	800	220	降低 72.5%	标准化元数据与多维索引大幅提升检索速度
	检索命中率/%	82	97	提升 15 个百分点	统一的元数据描述极大提高了发现准确性
数据调度能力	平均调度时间/ms	1 000	320	降低 68%	组件化封装使动态组合与匹配更高效
	调度成功率/%	75	95	提升 20 个百分点	有效解决协议、格式不统一等互操作难题

问题，15%来自权限校验冲突。引入数据组件后，通过语义标准化、接口版本管理与统一权限模型，上述问题得到系统性消解。综合多维测试结果，数据组件在流通延迟、访问性能、检索效率与调度稳定性上均显著优于传统模式，特别在跨域与异构环境中展现出更高的自治性、可扩展性与可靠性。

5 结论

数据组件的标准化封装与系统实现推动数据治理从紧耦合走向松耦合、自治化与可组合的新范式。一方面，数据组件构建独立、轻量、可复用的数据单元，确保跨域环境下的统一描述、可信交互与动态组合，实现数据的自治性与可移植性；另一方面，通过可寻址、可交换、可操作、可管控的结构模型，为异构系统中的封装、注册发布、发现获取与调度分发提供底层支撑。统一语义标识与标准化操作接口突破了跨平台共享瓶颈，使数据在异构环境中实现可寻址、可操作与可组合的灵活流通。实验结果显示，数据组件机制通过系统性解耦显著提升跨域流通效率与协同治理能力，在访问延迟、检索性能与复用灵活性方面均优于传统模式，验证了该范式在数据要素治理中的可行性与优势。

尽管本文已对封装建模、接入与检索机制进行验证，但注册索引与标识解析机制仍需深入评估。鉴于其对精准寻址、动态路由与可信访问的关键作用，后续研究将围绕元数据生成质量、注册检索效率、解析

延迟与路由准确性等指标开展系统实验，以进一步验证数据组件在大规模、多域环境下实现自主流动与可信交互的能力，为数据要素治理体系的工程化与制度化落地提供技术支撑。

参考文献

- [1] 吴曼青, 洪日昌, 王佐成, 等. 数据要素场技术体系及工程实践[J]. 中国工程科学, 2025, 27 (1): 51-62.
- [2] 樊振佳, 刘亭孝. 可信数据空间: 构筑文化数据流通与价值共创的新平台[J]. 图书馆建设, 2025 (2): 29-31, 48.
- [3] 黄罡. 数联网: 数字空间基础设施[J]. 中国计算机学会通讯, 2021, 17 (12): 60-62.
- [4] 韩光, 周崇毅, 郝东林, 等. 基于数据元件与数据金库打造安全可信金融数据空间[J]. 网络安全与数据治理, 2023, 42 (9): 1-5.
- [5] 罗超然, 马郗, 景翔, 等. 数据空间基础设施的技术挑战及物联网解决方案[J]. 大数据, 2023, 9 (2): 110-121.
- [6] BERNERS-LEE T, CAILLIAU R, LUOTONEN A, et al. The world-wide web[J]. Communications of the ACM, 1994, 37 (8): 76-82.
- [7] 梅宏, 曹东刚, 谢涛. 泛在操作系统: 面向人机物融合泛在计算的新蓝海[J]. 中国科学院院刊, 2022, 37 (1): 30-37.
- [8] CHEN Y, ZHENG Z B, LUO S, et al. Research on application of handle system in the industrial Internet[C]//Proc. of the 4th Intl Conf. on Intelligent Robotics and Control Engineering (IRCE). IEEE, 2021: 124-128.
- [9] 吴家菊, 刘刚, 席传裕. 基于 Web 服务的面向服务 (SOA)

- 架构研究[J]. 现代电子技术, 2005, 28 (14): 1-4, 7.
- [10] ELWENSPOEK M, LAMMERINK T S J, MIYAKE R, et al. Towards integrated microfluid handling systems[J]. Journal of Micromechanics and Microengineering, 1994, 4 (4): 227-245.
- [11] 陆志鹏. 数据主体主导的数据流通逻辑与实现路径研究[J]. 网络安全与数据治理, 2022, 41 (7): 3-8, 29.
- [12] 李春阳, 刘迪, 崔蔚, 等. 基于微服务架构的统一应用开发平台[J]. 计算机系统应用, 2017, 26 (4): 43-48.
- [13] 蔡珉官, 王朋. 数据湖技术研究综述[J]. 计算机应用研究, 2023, 40 (12): 3529-3538.
- [14] TERZO O, RUIU P, BUCCI E, et al. Data as a service (DaaS) for sharing and processing of large data collections in the cloud [C]//2013 Seventh International Conference on Complex, Intelligent, and Software Intensive Systems. IEEE, 2013: 475-480.
- [15] DONA. Digital object interface protocol version 2.0 [Z]. 2018.
- [16] HELLIWELL J R, MINOR W, WEISS M S, et al. Findable accessible interoperable re-usable (FAIR) diffraction data are coming to protein crystallography [J]. Applied Crystallography, 2019, 52 (3): 495-497.
- [17] EHSAN A, ABUHALIQA M A M E, CATAL C, et al. RESTful API testing methodologies: rationale, challenges, and solution directions[J]. Applied Sciences, 2022, 12 (9): 4369.
- (收稿日期: 2025-12-16)

作者简介:

林绍福(1967-), 男, 博士, 教授, 主要研究方向: 数据要素治理、区块链。

李云江(2001-), 男, 硕士研究生, 主要研究方向: 数据要素治理、区块链。

贾晓丰(1985-), 通信作者, 男, 博士, 主要研究方向: 大数据、数据要素治理。E-mail: jiaxf@jxj.beijing.gov.cn。

网络安全与数据治理

版权声明

凡《网络安全与数据治理》录用的文章，如作者没有关于汇编权、翻译权、印刷权及电子版的复制权、信息网络传播权与发行权等版权的特殊声明，即视作该文章署名作者同意将该文章的汇编权、翻译权、印刷权及电子版的复制权、信息网络传播权与发行权授予本刊，本刊有权授权本刊合作数据库、合作媒体等合作伙伴使用。同时，本刊支付的稿酬已包含上述使用的费用，特此声明。

《网络安全与数据治理》编辑部

www.pcachina.com