去中心化标识符 (DID) v1.1

核心架构、数据模型和表示法

W3C Editor's Draft 07 July 2024

▼ **More details about this document**

This version:
https://w3c.github.io/did-core/

Latest published version:
https://www.w3.org/TR/did-core-1.1/

Latest editor's draft:
https://w3c.github.io/did-core/

History:
Commit history

Editors:
Manu Sporny (Digital Bazaar) (v1.0, v1.1)
Dmitri Zagidulin (Invited Expert) (v1.0)
Steve McCown (v1.0)

Former editors:
Amy Guy (Digital Bazaar) (v1.0)
Markus Sabadello (Danube Tech) (v1.0)
Drummond Reed (Evernym/Avast) (v1.0)

Authors:
Manu Sporny (Digital Bazaar)
Dave Longley (Digital Bazaar)
Markus Sabadello (Danube Tech)
Drummond Reed (Evernym/Avast)
Orie Steele (Transmute)
Christopher Allen (Blockchain Commons)

Feedback:
GitHub w3c/did-core (pull requests, new issue, open issues)
public-did-wg@w3.org with subject line `[did-core-1.1] … message topic …` (archives)

Related Documents
DID Use Cases and Requirements
DID Specification Registries
DID Core Implementation Report

摘要

去中心化标识符（DID）是一种新型标识符，支持可验证的、去中心化的数字身份。一个 DID 可以标识由 DID 控制者决定的任何主题（例如个人、组织、事物、数据模型、抽象实体等）。与典型的联合标识符相比，DIDs的设计使其分离于集中式注册表、身份提供者以及证书颁发机。具体来说，虽然可借助第三方协助发现与DID相关的信息，但该设计允许DID控制者无需任何第三方授权的情况下即可证明对其的控制。DIDs 是将 DID 主题与DID 文档相关联的URIs，允许与该主题相关联的可信交互每个 DID 文档都可以表达加密材料、验证方法或服务，提供了一组让DID控制者能够证明对 DID 控制的机制。服务可实现与 DID 主题相关的可信交互。如果 DID 主题是数据模型等信息资源，则 DID 可提供返回 DID 主题本身的方法。本文档规定了 DID 语法、通用数据模型、核心属性、序列化表示、DID 操作，并解释了将 DID 解析为其所代表的资源的过程。

本文现状

本节介绍本文档出版时的状态。当前 W3C 出版物和本技术报告最新修订版的列表可在 https://www.w3.org/TR/ 的 W3C 技术报告索引中找到。

此版本的 DID 核心标准（1.1 版）是试验性的。请勿实施。如果要实施 DID，请使用当前的 1.0 版标准： Decentralized Identifiers (DIDs) v1.0.

本文档由 Decentralized Identifier Working Group 作为编辑草案发布。

作为编辑草案发布并不意味着得到 W3C 及其成员的认可。

本文件为草案，可能随时被其他文件更新、取代或废止。除正在进行的工作外，不宜引用本文件。

本文档由一个根据 W3C 专利政策运作的小组编写。W3C 保留一份公开清单，记录与该小组交付成果相关的任何专利披露；该页面

还包括披露专利的说明。如果个人实际知晓一项专利，并认为该专利包含必要权利要求，则必须根据 W3C 专利政策第 6 节披露该信息。

本文件受 2023 年 11 月 3 日 W3C 流程文件管辖。

1. 导言

本节为非规范性内容。

作为个人和组织，我们中的许多人都在各种场合使用全球唯一标识符。它们可用作通信地址（电话号码、电子邮件地址、社交媒体上的用户名）、身份证号码（护照、驾照、纳税 ID、医疗保险）和产品标识符（序列号、条形码、RFIDs）。URIs（统一资源标识符）用于网络资源，您在浏览器中浏览的每个网页都有一个全球唯一的 URL（统一资源定位符 `Uniform Resource Locator`）。

这些全球唯一标识符绝大多数不受我们控制。它们是由外部机构发布的，这些机构决定它们所指的是谁或什么，以及何时可以撤销。它们只在某些情况下有用，只被某些机构认可，而不是由我们选择。它们可能会随着组织的倒闭而消失或失效。它们可能泄露个人信息。在许多情况下，它们可能会被恶意的第三方以欺诈方式复制和利用，这就是通常所说的“身份盗窃”。

本规范定义的去中心化标识符（DID）是一种新型的全球唯一标识符。其目的是让个人和组织能够使用他们信任的系统生成自己的标识符。这些新标识符使实体能够通过使用数字签名等加密证明进行验证，从而证明对标识符的控制权。

由于去中心化标识符的生成和声明是由实体控制的，每个实体都可以根据需要拥有尽可能多的去中心化标识符，以保持其所需的身份、角色和互动的分离。这些标识符的使用范围可以根据不同的情况进行适当调整。它们支持与其他人、机构或系统的交互，这些交互要求实体识别自己或自己控制的事物，同时提供对个人隐私数据披露程度的控制，而所有这些都不依赖于中央机构来保证标识符的持续存在。DID 用例文档 [DID-USE-CASES] 对这些想法进行了探讨。

本规范并不预设任何特定的技术或加密技术来支持 DID 的生成、持久性、解析或解释。例如，实施者可以根据在联合或中心化身份管理系统中注册的标识符创建去中心化标识符。事实上，几乎所有类型的标识符系统都可以增加对 DID 的支持。这就在中心化、联合式和去中心化标识符之间架起了一座互操作性桥梁。这也使实施者能够设计特定类型的 DID，以便与他们信任的计算基础设施（如分布式账本、去中心化文件系统、分布式数据库和点对点网络）协同工作。

本规范适用于：

- 希望了解作为去中心化标识符基础的核心架构原则的任何人；
- 希望生产和使用去中心化标识符及其相关数据格式的软件开发人员；
- 希望了解如何在其软件和硬件系统中使用去中心化标识符的系统集成商；
- 希望创建符合本文档所述生态系统的新 DID 基础设施（称为 DID 方法）的标准作者。

除本标准外，读者可能会发现去中心化标识符的用例和要求 [DID-USE-CASES] 文档也很有用。

1.1 一个简单的例子

本节为非规范性内容。

DID 是一个简单的文本字符串，由三部分组成：

1）DID URI 方案标识符；（`did URI scheme identifier`）

2）DID 方法标识符；（`DID method`）

3）DID 方法特定标识符。（`DID method-specific identifier`）

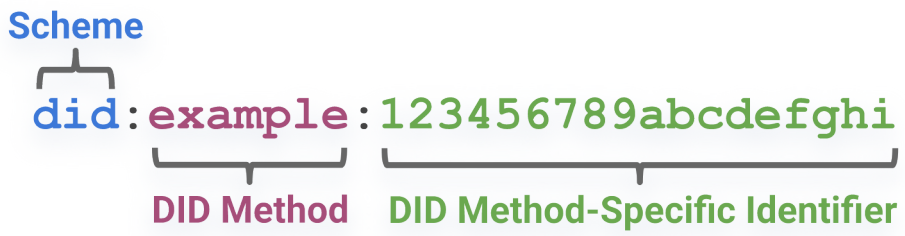

图 1 分散式标识符 (DID) 的简单示例

上面的 DID 示例解析为一个 DID 文档。DID 文档包含与 DID 相关的信息，如对 DID 控制器进行加密验证的方法。

> 示例 1: 一个简单的DID文档

```
{
  "@context": [
    "https://www.w3.org/ns/did/v1",
    "https://w3id.org/security/suites/ed25519-2020/v1"
  ],
  "id": "did:example:123456789abcdefghi",
  "authentication": [{
    // used to authenticate as did:...fghi
    "id": "did:example:123456789abcdefghi#keys-1",
    "type": "Ed25519VerificationKey2020",
    "controller": "did:example:123456789abcdefghi",
    "publicKeyMultibase": "z6MkmM42vxfqZQsv4ehtTjFFxQ4sQKS2w6WR7emozFAn5cxu"
  }]
}
```

1.2 设计目标

本节为非规范性内容。

去中心化标识符是更大系统的一个组成部分，如可验证凭证生态系统 [VC-DATA-MODEL]，这影响了本规范的设计目标。这里总结了去中心化标识符的设计目标。

目标	描述
去中心化	在标识符管理（包括全球唯一标识符、公共验证密钥、服务和其他信息的注册）中消除对中心化机构或单点故障的要求。
控制权	赋予实体（包括人类和非人类实体）直接控制其数字标识符的权力，而无需依赖外部机构。
隐私	使实体能够控制其信息的隐私，包括最小化、选择性和渐进式地披露属性或其他数据。
安全性	为请求方提供足够的安全性，使其能够依赖 DID 文档来获得所需的保证级别。
基于证明	让 DID 控制者在与其他实体交互时提供加密证明。
可发现性	使实体可以发现其他实体的 DID，以了解更多信息或与这些实体进行交互。
互操作性	使用互操作性标准，使 DID 基础设施可以利用为实现互操作性而设计的现有工具和软件库。
可移植性	与系统和网络无关，使实体能够在任何支持 DID 和 DID 方法的系统中使用其数字标识符。
简便性	简化功能，使技术更易于理解、实施和部署。
可扩展性	只要不严重妨碍互操作性、可移植性或简易性，应尽可能实现可扩展性。

1.3 架构概述

本节为非规范性内容。

本节提供去中心化标识符架构主要组件的基本概述。

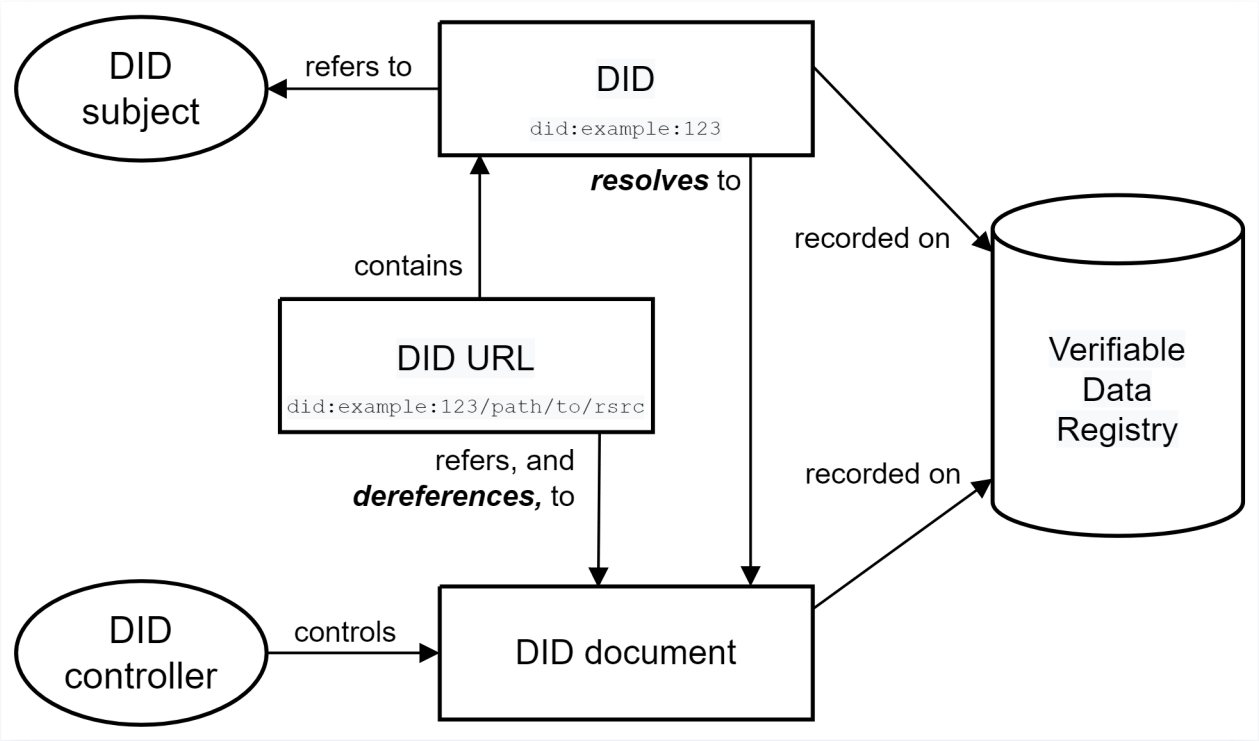

图 2 DID 架构概览和基本组件之间的关系。另见：叙述性说明。

DIDs 和 DID URLs

去中心化标识符或 DID 是由三部分组成的 URI：方案 did:(`scheme did:`)、方法标识符(`method identifier`)和 DID 方法(`DID method`)指定的唯一特定方法标识符。DID 可解析为 DID 文档(`DID documents`)。DID URL 扩展了基本 DID 的语法，纳入了路径、查询和片段等其他标准 URI 组件，用于定位特定资源--例如，DID 文档中的加密公钥或 DID 文档外部的资源。这些概念将在3.1 DID 语法和3.2 DID URL 语法中详细阐述。

DID 主题

DID 主题顾名思义就是 DID 所标识的实体。DID 主题也可以是 DID 控制器。任何东西都可以是 DID 的主题：人、团体、组织、事物或概念。5.1.1 DID 主题对此有进一步定义。

DID 控制者

DID 控制者是指有能力（如 DID 方法所定义）对 DID 文档进行更改的实体（个人、组织或自主软件）。这种能力通常由代表控制者的软件对一组加密密钥的控制来实现，但也可能通过其他机制来实现。请注意，一个 DID 可能有不止一个控制器，而 DID 主题可以是 DID 控制器，也可以是其中之一。这一概念在 5.1.2 DID 控制器中有所阐述。

可验证的数据注册

为了能解析 DID 文档，DID 通常会记录在某种底层系统或网络中。无论使用何种特定技术，任何支持记录 DID 并返回生成 DID 文档所需数据的系统都称为可验证数据注册中心。这方面的例子包括分布式分类账、去中心化文件系统、各种数据库、点对点网络以及其他形式的可信数据存储。这一概念将在8. 方法中进一步阐述。

DID 文档

DID 文档包含与 DID 相关的信息。它们通常表达与 DID 主题交互相关的验证方法（如加密公钥）和服务。DID 文档支持的通用属性在5.核心属性中。DID 文档可序列化为字节流（见 6.表示法）。DID 文档中的属性可根据 8.方法。

DID 方法

DID 方法是创建、解析、更新和停用特定类型 DID 及其关联 DID 文档的机制。DID 方法使用单独的 DID 方法规范来定义，如8. 方法中的定义。

DID 解析器和 DID 解析

DID 解析器是一个系统组件，它将 DID 作为输入，并将符合要求的 DID 文档作为输出。这一过程称为 DID 解析。解析特定类型 DID 的步骤由相关 DID 方法规范定义。DID 解析过程详见 7. 解析。

DID URL 解除引用器和 DID URL 解除引用

DID URL 解除引用器是一个将 DID URL 作为输入并将资源作为输出的系统组件。这一过程称为 DID URL 解除引用。7.2 DID URL 解除引用中详细介绍了 DID URL 解除引用的过程。

1.4 一致性

除标注为非规范性的部分外，本规范中的所有编写指南、图表、示例和注释都是非规范性的。本规范中的其他内容均为规范性内容。

本文档中的关键词 MAY、MUST、MUST NOT、OPTIONAL、RECOMMENDED、REQUIRED、SHOULD 和 SHOULD NOT，当且仅当它们以大写字母出现时，应按照 BCP 14 [RFC2119] [RFC8174] 中的描述进行解释，如此处所示。

本文档包含包含 JSON 和 JSON-LD 内容的示例。其中一些示例包含无效字符，例如内联注释 (//) 和使用省略号（...）来表示对示例没有什么价值的信息。如果实施者希望将这些信息用作有效的 JSON 或 JSON-LD，请务必删除这些内容。

某些示例包含本规范未定义的术语（包括属性名称和属性值）。这些术语用注释（//外部（属性名|值））表示。这些术语在 DID 文档中使用时，应在 DID 规范注册表 [DID-SPEC-REGISTRIES] 中注册，并链接到正式定义和 JSON-LD 上下文。

测试DID 和 DID 文档实现的互操作性是通过评估创建和解析符合本规范的 DID 和 DID 文档的能力。DID 和 DID 文档生产者和消费者的互操作性是通过确保 DID 和 DID 文档符合规范来实现的。DID 方法规范的互操作性参考每个 DID 方法规范中的具体细节。例如，像网络浏览器不需要实现所有已知的 URI 方案一样，与 DID 兼容的软件也不需要实现所有已知的 DID 方法。但是，特定 DID 方法的所有实现都应具有互操作性。

符合要求的 DID 是对 3.标识符 中规定的规则在符合该节中的相关规范声明下的具体实现。

符合要求的 DID 文档是本规范中描述的数据模型在符合 4. 数据模型 和 5. 核心属性 下的实现。符合要求的文档的序列化格式是确定的、双向的和无损的，如 6. 表述 。

符合规范的生产者是以软件和/或硬件形式实现的任何算法，用于在符合 6.表述 中的相关规范声明下生成符合规范的 DID 或符合标准的 DID 文档。

符合标准的消费者是指作为软件和/或硬件实现的任何算法，它消费符合标准的 DID 或符合标准的 DID 文档，并遵守 6.表述 中的相关规范声明。

符合标准的 DID 解析器是指在符合 7.1 DID 解析中的相关规范声明下任何以软件和/或硬件形式实现的算法。

符合要求的 DID URL 解除器是指以软件和/或硬件形式实现的、符合 7.2 DID URL 解除中相关规范声明的任何算法。

符合标准的 DID 方法是指符合 8.方法中相关规范声明的任何规范。

2. 术语

本节为非规范性内容。

本节定义了本标准和整个去中心化标识符基础中使用的术语。只要这些术语在本标准中出现，就会提供相关链接。

放大攻击（`amplification attack`）
一类攻击，攻击者通过向系统提供少量有效输入，试图耗尽目标系统的 CPU、存储、网络或其他资源，从而造成破坏性影响，其处理成本可能以指数形式超过输入本身。

验证（`authenticate`）
身份验证是一个实体使用一种或多种验证方法证明其拥有特定属性或控制特定秘密的过程。就 DID 而言，一个常见的例子是证明与 DID 文档中公布的公钥相关的加密私钥的控制权。

加密套件（`cryptographic suite`）
为实现特定安全目标而定义特定加密原语用法的标准。这些文件通常用于指定验证方法、数字签名类型、其标识符和其他相关属性。

去中心化识符（`decentralized identifier (DID)`）
全球唯一永久的标识符，不需要中央注册机构，通常以加密方式生成和/或注册。3.1 DID 语法中定义了 DID 的通用格式。具体的 DID 方案在 DID 方法标准中定义。许多——但并非所有——DID 方法都使用分布式账本技术（DLT）或其他形式的去中心化网络。

去中心化身份管理（`decentralized identity management`）
基于使用去中心化标识符为的身份管理。将生成、注册和分配标识符的权力扩展到传统的信任根基之外，如 X.500 目录服务、域名系统和大多数国家的身份证系统。

DID 控制者（`DID controller`）
具有能力更改 DID 文档的实体。一个 DID 可能有不止一个 DID 控制器。DID 控制器可通过 DID 文档顶层的可选控制器属性表示。请注意，DID 控制器可能就是 DID 主题。

DID 委托人（`DID delegate`）

DID 控制员通过 DID 文档授权使用与 DID 相关的验证方法的实体。例如，控制孩子 DID 文档的家长可能允许孩子使用个人设备进行身份验证。在这种情况下，孩子就是 DID 委托人。孩子的个人设备包含私人加密材料，使孩子能够使用 DID 进行身份验证。不过，未经父母允许，孩子可能不得添加其他个人设备。

DID 文档（`DID document`）

描述 DID 主题的一组数据。其包括 DID 主题或 DID 委托人可用于验证自身身份并证明与 DID 关联的机制（如加密公钥）。一个 DID 文档可能有一个或多个不同的表示法，如 6.表示法或 W3C 文件中的定义。表示法或 W3C DID 标准注册表 [DID-SPEC-REGISTRIES] 中定义的一种或多种不同表示法。

DID 片段（`DID fragment`）

DID URL 中第一个散列符号字符 (#) 之后的部分。DID 片段语法与 URI 片段语法相同。

DID 方法（`DID method`）

特定 DID 方法方案实现方式的定义。DID 方法由 DID 方法标准规定，该标准规定了创建、解析、更新和停用 DID 和 DID 文档的详细操作。请参阅 8.方法 。

DID 路径（`DID path`）

DID URL 中以第一个正斜线 (/) 字符开头并包含该字符的部分，以问号 (?)、片段哈希符号 (#) 字符或 DID URL 结尾结束。DID 路径语法与 URI 路径语法相同。请参阅路径。

DID 查询（`DID query`）

DID URL 中第一个问号字符 (?) 之后的部分。DID 查询语法与 URI 查询语法相同。请参阅查询。

DID 解析（`DID resolution`）

将 DID 和一组解析选项作为输入，并以符合要求的表示形式返回 DID 文档和附加元数据的过程。该流程依赖于适用 DID 方法的 “读取 ”操作。该流程的输入和输出在 7.1 DID 解析中定义。

DID 解析器（`DID resolver`）

DID 解析器是执行 DID 解析功能的软件和/或硬件组件，它将 DID 作为输入，并将符合要求的 DID 文档作为输出。

DID 方案（`DID scheme`）

去中心化标识符的正式语法。通用 DID 方案以 3.1 DID 语法中定义的前缀 did: 开始。每个 DID 方法标准都定义了与特定 DID 方法配合使用的特定 DID 方法方案。在特定 DID 方法方案中，DID 方法名称以第一个冒号开头，以第二个冒号结尾，如 did:example:

DID 主题（`DID subject`）

由 DID 认证并由 DID 文档描述的实体。任何东西都可以成为 DID 主题：人、群体、组织、客观事物、数字事物、主观事物。

DID URL

DID 加上符合 3.2 DID URL 语法定义的任何附加语法组件。这包括可选的 DID 路径（带前导 / 字符）、可选的 DID 查询（带前导 ? 字符）和可选的 DID 片段（带前导 # 字符）。

DID URL 解引用（`DID URL dereferencing`）
将一个 DID URL 和一组输入元数据作为输入，并返回一个资源的过程。该资源可能是 DID 文档加上附加元数据，也可能是 DID 文档中包含的二级资源，还可能是与 DID 文档无关的资源。该流程使用 DID 解析来获取 DID URL 中包含的 DID 所指示的 DID 文档。解引用流程可对 DID 文档处理，以返回 DID URL 所指的解引用资源。该流程的输入和输出在 7.2 DID URL 解引用中定义。

DID URL 解引用器（`DID URL dereferencer`）
对给定 DID URL 或 DID 文档执行 DID URL 解引用功能的软件和/或硬件系统。

分布式账本（`distributed ledger (DLT)`）
用于记录事件的非集中式系统。这些系统为参与者建立了足够的信任，使其能够依赖他人记录的数据做出操作决策。它们通常使用分布式数据库，不同节点使用共识协议确认加密签名交易的排序。经过数字签名的交易随着时间的推移被连接起来，这通常会使账本的历史不可更改。

公钥描述（`public key description`）
包含在 DID 文档中的数据对象，其中包含使用公钥或验证密钥所需的所有元数据。

资源（`resource`）
由 [RFC3986] 定义："……术语'资源'用于一般意义上的任何可由 URI 标识的内容"。同样，任何资源都可以作为由 DID 标识的 DID 主题。

表示法（`representation`）
正如 [RFC7231] 为 HTTP 所定义："旨在反映指定资源的过去、当前或期望状态的信息，其格式可通过协议轻松通信，由一组表示元数据和可能无限制的表示数据流组成"。DID 文档是描述 DID 主题的信息表征。见 6. 表示。

特殊表征条目（`representation-specific entries`）
DID 文档中的条目，其含义是特定表示法所特有的。在 4. 数据模型》和《6. 表示法中定义。例如，JSON-LD 表示法中的 @context 就是特定于表示法的条目。

服务（`services`）
通过一个或多个服务端点与 DID 主题或相关实体进行通信或交互的方式。例如，发现服务、代理服务、社交网络服务、文件存储服务和可验证凭证存储库服务。

服务端点（`service endpoint`）
一个服务代表 DID 主题运行的网络地址，如 HTTP URL。

统一资源标识符（`Uniform Resource Identifier (URI)`）
由 [RFC3986] 定义的万维网上所有资源的标准标识符格式。DID 是 URI 方案的一种。

可验证凭证（`verifiable credential`）
W3C 可验证凭证规范 [VC-DATA-MODEL] 所定义的加密可验证数字凭证的标准数据模型和表示格式。

可验证数据注册中心（`verifiable data registry`）
一种便于创建、验证、更新和/或停用去中心化标识符和 DID 文档的系统。可验证数据注册中心也可用于其他加密可验证数据结构，如可验证凭证。更多信息，请参阅 W3C 可验证凭证规范 [VC-DATA-MODEL]。

可验证时间戳（`verifiable timestamp`）
可验证时间戳可让第三方验证数据对象在某一特定时刻是否存，以及自该时刻起是否被修改或损坏。如果数据完整性从该时刻起已被修改或损坏，则时间戳无法验证。

验证方法（`verification method`）
一组参数，可与流程一起用于独立验证证明。例如，加密公共密钥可用作数字签名的验证方法；在这种用法中，它可以验证签名者是否拥有相关的加密密钥。

本定义中的 "验证 "和 "证明 "意在广泛适用。例如，在 Diffie-Hellman 密钥交换过程中，可以使用加密公开密钥来协商加密共享对称密钥。这保证了密钥协议过程的完整性。因此，这也是另一种验证方法，尽管对这一过程的描述可能不会使用 "验证 "或 "证明 "等词。

验证关系（`verification relationship`）
DID 主题与验证方法之间关系的表述。5.3.1 验证就是验证关系的一个例子。

全球唯一标识符（`Universally Unique Identifier (UUID)`）
由 [RFC4122] 定义的一种全球唯一标识符。UUID 与 DID 类似，都不需要集中注册机构。UUID 与 DID 的不同之处在于它们不可解析或不可加密验证。
除上述术语外，本标准还使用 [INFRA] 规范中的术语来正式定义数据模型。当使用 [INFRA] 术语（如字符串、集合和映射）时，会直接链接到该标准。