ICS 35.040
CCS L70

团　体　标　准

T/IQA 31—2024

去中心化软件标识技术要求

Technical Requirements of Software Decentralized Identity

2024-06-28 发布　　2024-06-29 实施

中关村智联软件服务业质量创新联盟　发布

目　　次

前　言

本文件按照 GB/T 1.1—2020《标准化工作导则　第 1 部分：标准化文件的结构和起草规则》的规定起草。

本文件由中关村智联软件服务业质量创新联盟标准化管理委员会提出并归口。

本文件起草单位：京东科技信息技术有限公司、工业和信息化部电子第五研究所、中国信息安全测评中心、中关村智联软件服务业质量创新联盟、北京奕斯伟计算技术股份有限公司、吉利汽车集团、麒麟软件有限公司、统信软件技术有限公司、苏州棱镜七彩信息科技有限公司、深圳市金蝶天燕云计算股份有限公司、深圳开源互联网安全技术有限公司、国民认证科技（重庆）有限公司、北京辰光融信技术有限公司、北京北大软件工程股份有限公司、北京人大金仓信息技术股份有限公司、北京万里红科技有限公司、武汉达梦数据库股份有限公司、北京神州新桥科技有限公司、全聚合数字技术有限公司、杭州云象网络技术有限公司、北京简单一点科技有限公司、天津南大通用数据技术股份有限公司、中信信托有限责任公司。

本文件主要起草人：柴思跃、郑伟娜、高松、邵帅、王晓萌、文波、孙丽丽、周子隆、郭斐、黄步添、肖然、宋晓旭、李颖祎、付剑、詹文君、梁大功、王媛媛、王博、王颉，胡要中、张超、李俊、沈玮、刘庆、王红蕾、李伟彬、陈曦、岳贯集、朱何龙、陈静、黄葳唯，胡一鸣、孙祝广、王滢、杨雷、杨熊，王荣颜、康伟、冯文忠、庄伟波、黄佳、沈美。

去中心化软件标识技术要求

1 范围

1.1 概述

各种规模的组织都需要追踪其拥有和操作的软件，以执行用户支持、库存管理和有效安全防护有效。实现软件可跟踪性的核心是允许软件在其生产、分发、安装、运维生命周期各环节产生的信息相关联。目前，各软件生产方已在生产过程中应用唯一标识符UUID作为组织本身的认证标识，如URI(统一资源标识符)或其他UUID资源标记。但这类标识符往往由中心化权威机构发布，外部用户只允许在规定情况定向解析。这种不具备对公开、透明特性的标识，用户只能被动选择中心化认证机构，用户信认的标识数据会随着组织的失败而消失或失效，或者随着中心化网络攻击而造成数据泄露或数据恶意盗窃，即"身份盗窃"或中间人攻击事件。

2022年，万维网联盟（W3C)发布 《Decentralized Identifiers (DIDs) v1.0 (草案)》规范，公开了一种分布式标识算法。但这一规范在记录目标对象信息内容时，存在标识语义内容泛化问题，需要集中定义通信协议对象，并需要上链查找对应协议并获取原数据信息。本文件面向软件生存周期领域，明确定义软件产品标识语义，并创新可配置自动化模板，可规范化实现单空间多语义表达，解决DID标识语义泛化问题。

1.2 目的

本文件提供了去中心化软件标识技术的原则、内容、流程操作要点，旨在代码文件、软件包跨组织、跨生命周期的互认中规范使用分布式、去中心化软件标识技术。

本文件适用于采用区块链和去中心化软件标识技术，向社会公众提供软件生产可审计信息的生产者、审计机构与软件应用用户，为相关方技术和业务人员在生产、识别、存储、分发、验证软件标识中提供可操作指南。

2 规范性引用文件

下列文件中的内容通过文中的规范性引用而构成本文件必不可少的条款。其中，注日期的引用文件，仅该日期对应的版本适用于本文件；不注日期的引用文件，其最新版本（包括所有的修改单）适用于本文件。

ISO/IEC/IEEE 32675-2022 信息技术—DevOps—构建可靠和安全的系统，包括应用程序构建、打包和部署

GB/T 8566-2022 系统与软件工程 软件生存周期过程

GB/T 36328-2018 信息技术 软件资产管理 标识规范

GB/T 42752-2023 区块链和分布式记账技术 参考架构

GM/T 0067-2019 基于数字证书的身份鉴别接口规范

YD/T 4566-2023 基于区块链的物联网设备标识与认证系统的总体技术要求

YD/T 3453-2019 基于eID的多级数字身份管理技术参考框架

YD/T 3204-2016　网络电子身份标识eID体系架构

3　术语和定义

下列术语和定义适用于本文件。

3.1　软件生存周期 systems development life cycle

软件系统、产品、服务、项目从概念到退役的演变。

[来源:GB/T 42752-2023,3.10]

3.2　软件标识 software identification

在软件产品可被唯一识别软件的标识符。

3.3　软件标识生命周期 software identification lifecycle

伴生于软件生存周期的标识本体，在软件产品开发、打包、分发、部署、运行和卸载生命周期过程中创建直到被废弃或淘汰的过程。

3.4　分布式账本 distributed ledger

在分布式节点间共享并使用共识机制实现具备一致性的账本。

[来源:GB/T 42752-2023,3.10]

3.5　去中心化标识 decentralized identity

是一种基于分布式账本技术的、，独立于中央注册表、身份提供者和证书权威机构的新型身份标识符。

3.6　去中心化标识文档 decentralized identity document

描述去中心化标识主体的一组公开数据。

注1：去中心化标识文档包括密码公钥等机制，声明方验证或关联认证机制。

注2：去中心化标识文档记录在分布式账本上进行完全信息披露。

3.7　去中心化标识解析 decentralized identity Resolve

是指给定DID标识获取DID文档的过程。

注：任何满足DID规范的标识必须具备反向映射解析DID原始数据的能力。

3.8　声明 claims

对于特定对象的确定信息。

[来源:ISO 20078-3:2021, 3.3]

3.9 可验证凭证 verifiable credentials

是一种已发布、可被证明的数字化声明。

注1：凭证可验证性并不意味着其中声明信息或声明摘要的真实性。

注2：本规范通过规范定义软件产品在生存周期模型中的状态声明，基于去中心化身份提供一个标准化的方法来表示和交换可验证信息，以增强软件身份验证和数据完整性评价功能。

3.10 声明协议模板 claim protocol template

一种用于标准化和结构化声明数据的模板，它定义了声明格式和内容，以便于生成、共享和验证。

注：结构化格式包括声明的类型、描述、数据格式、验证规则等。可根据具体应用场景和需求进行扩展，使得声明数据在跨系统和跨平台间具有一致性和互操作性。

4 缩略语

下列缩略语适用于本文件

DID:去中心化标识 decentralized Identity

VC :可验证凭证 verifiable credentials

CPT:声明协议模板 claim protocol template

5 整体架构

5.1 主体简介

去中心化软件标识系统是基于分布式账本和 DID 协议标准之上的技术体系。系统内涉及主体包括:

a) 申请人：一般为软件企业生产者，为软件进行标识申请的主体。

b) 用户代理：一般为代理程序，为软件生产者提供软件标识、密钥及可验证凭证的托管、代理软件标识注册等功能。

c) 验证人：一般为软件用户，使用软件并对该软件标识及可验证凭证进行验证的主体。

d) 智能合约：区块链存储网络或其他分布式账本技术。

5.2 标识运行原则

去中心化标识体系允许软件生产主体自主管理与身份相关的信息。软件生产创建标识符，并在不依赖中央机构（如服务提供商或政府）的情况下申请和持有软件可验证凭证。标识及其凭证注册在区块链上，以加密安全、尊重隐私和可机器验证的方式表达。

a) 软件标识技术体系运行原则：

b) 去中心化：不依赖于任何中心化的注册机构，可以在多个不同的去中心化系统中创建和管理。

c) 自主性：用户可以完全控制他们的标识身份，无需依赖于第三方服务。

d) 可验证性：标识身份可以与可验证凭证（Verifiable Credentials）一起使用，提供强大的身份验证能力。

e) 隐私保护：标识标准设计支持用户对主体信息的选择性披露。

5.3 运行架构

运行架构流程见图 1：

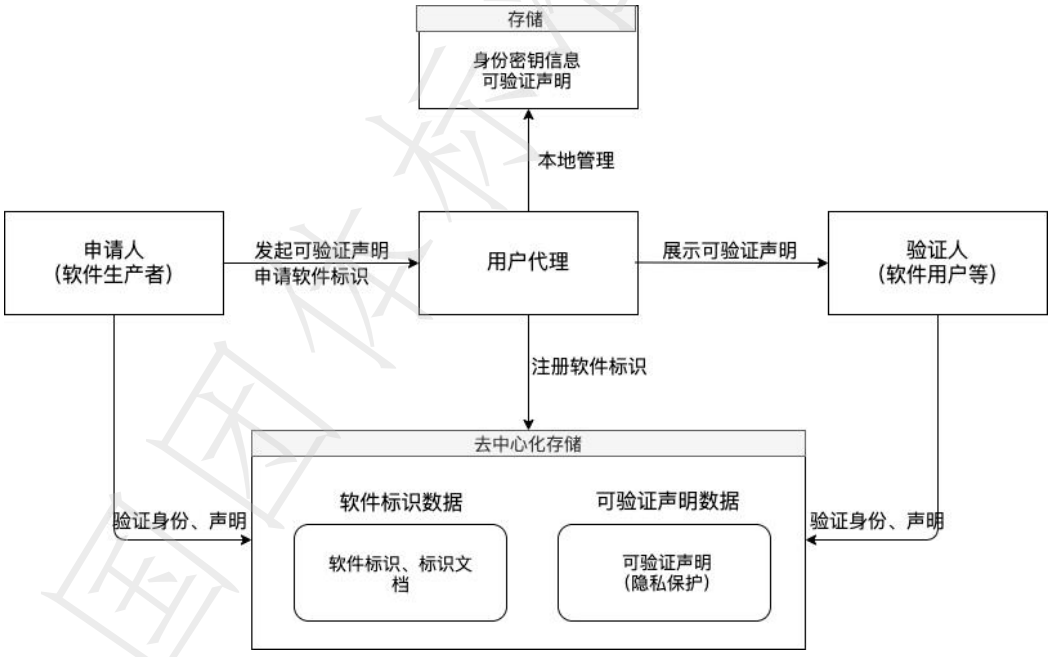

图1. 去中心化软件标识运行架构

a) 软件标识申请：软件生产主体申请软件标识，并写入分布式智能合约。

b) 可验证凭证验证：软件生产主体将软件的可验证凭证通过用户代理程序向软件用户等验证人出具或展示，软件用户可对软件标识和可验证凭进行验证和认证声明。

6 标识数据规范

6.1 标识规范

软件标识应用ID规范延用W3C规范体系，实现为软件数字身份的分散化和自主控制。软件标识由DID标识和DID文档组合而成。标识分为三段，由DID协议机制、方法和自定义标识组成。见图1：

图 1 引用 W3C 规范化 DID 标识表达

a) DID 协议机制字符：常使用“did”表示。

b) DID 方法字符：规范采用 UIMP，长度一般不超过 8 个字符。

c) 自定义标识：保证软件包唯一性字符串，通常可采用杂凑摘要、UUID 等算法生成，或其他自定义格式。

6.2 标识文档

DID文档是主体所附带相关属性的描述文档，生成要求如下：

a) 采用JSON-LD格式，UTF-8字符编码格式。

b) 必须包含软件DID标识符；

c) 至少包含一个加密材料，如公钥。

d) 至少包含一个服务端点。

e) 规范化时间戳。使用Unix时间戳规范，表示自1970年1月1日（UTC/GMT 午夜）以来的精确毫时间长度，单位为毫秒。

f) DID文档合法性签名，可选用JSON-LD格式签名。

g) 声明软件生产主体与软件产品的从属关系。

结构说明如下：

表 1 去中心化标识文档结构

字段	类型	必填	说明
@context	String	是	用于描述 did 文档结构等信息说明
id	String	是	当前文档描述主体 DID 标识
controller	String[]	是	软件生产主体 DID，示例：did:UIMP:mychannel:123
alsoKnownAs	String	是	文件索引字符，提升人工可读性，如："微软 Windows 11"
created	Long	是	创建时间戳，毫秒级，示例：1645201775000
updated	Long	是	更新时间，毫秒级，示例：1645201776000
deactivated	Boolean	是	是否失效。 true 代表已经失效，false 代表未吊销。
frozen	Boolean	是	是否冻结。如果 deactivated 为 false，则 true 代表 did 已经冻结，false 代表 did 依然正常。
verificationMethod	PublicKey[]	是	存储 DID 身份的公钥信息数组列表，但其中仅一个有效公钥。PublicKey 结构说明见附录 A.1
service	Service[]	否	service 描述数用于描述跟当前 DID 相关的服务。 Service 结构说明见附录 A.2
metadata	String	否	DID 主体公开声明的 JSON-LD 序列化信息 Service 结构说明见附录 A.3

7 软件可验证凭证规范

凭证由同一实体提出的一个或多个声明的集合，包括标识符和元数据，用于描述凭证的属性，如颁发者、有效日期和时间段、代表性图像、验证材料、状态信息等。可验证凭证是可由身份主体设定的不可篡改凭证，是主题明文信息的映射，具有可信分发、防篡改、易验证等特征。可验证凭据结构示例如图2:

图 2 可验证凭证

7.1 声明协议模板数据结构

本文创新定义声明协议模板（CPT），用于抽象声明元数据信息，明确定义和表达软件生存周期内软件产品的主题信息，实现软件产品在生产、打包、分发、部署、运维过程中的供需双方透明传递软件产品基本信息。见表2:

表 2 可验证凭证模板数据结构

字段名称	类型	说明
title	String	CPT 名称，示例，"信创认证证书模板"
description	String	CPT 描述信息，示例，"用于申请软件信创认证"
created	Long	创建时间，毫秒级，示例：1645201775000
updated	Long	更新时间，毫秒级，示例：1645201775000
properties	String	主体可自主定义声明，包括字段类型、结构、长度等等。主体应按照 JSON Schema 进行格式化规范。但为保证数据传输效率，建议声明信息大小不超过 1MB。
publisher	String	CPT 的发布者
proof	Proof	CPT 的发布者对 claim 结构的摘要签名。见附件 A.4
deactivated	Boolean	代表 CPT 状态，true 代表已经废止，false 代表生效中

7.2 可验证凭证数据结构

名称	类型	说明
issuer	String	声明方的 DID，一般是软件生产主体
issuanceDate	Long	声明时间，毫秒级，示例：1645201775000
expirationDate	Long	有效期，为空就是无限期，毫秒级，示例：1645201775000
holder	String	VC 持有者的软件 did
cptId	String	对应 CPT 的编号
claim	String	指定 CPT 模板的声明对象。
salt	String	盐值，选择性披露时用于保证不被类似彩虹表的方式反向破解 hash 算法。默认 salt 为空，表示数据全披露，这时 claim 字段不做 hash；vc 需要隐私性验证时，此字段才有值。
proof	Proof	声明方对 Claim 的摘要签名。见附件 A.4
isRevoke	Boolean	是否吊销。默认为 false

8 标识解析接口

本节定义了标识resolve解析方法接口。W3C规范要求每类DID必须实现至少一种标识解析功能，并且必须能够返回至少一种符合规范的标识文档表示形式。

8.1 resolve 方法

根据指定DID标识查询标识对应原始信息，包括didResolutionMetadata 、didDocument和didDocumentMetadata。见表3：

表 3 标识解析方法 didResolutionMetadata 结构

字段名称	类型	说明
didResolutionMetadata	String	主要包括解析结果码与错误码定义
didDocument	String	did文档结构
didDocumentMetadata	String	did文档元数据信息。见附件A.5

附 录 A
（规范性附录）
去中心化标识文档示例

A.1 DID文档内身份公钥信息结构与示例

表 4 PublicKey 结构说明

名称	类型	必填	说明
id	String	是	公钥 ID
type	String	是	链的密码学算法,示例：“SM2”
controller	String	是	指定控制对应私钥的 Entity 如果为空，则表明持有者是 Document 的 id 字段。
publicKeyPem	String	是	SM2，pem 编码的公钥内容
expired	Long	是	更新公钥会废止历史的公钥。默认为 0L，表示状态正常；否则为公钥的废止时间

示例：

```
{
    "id":"did:UIMP:mychannel:123#key-skiLatest",
    "type": "SM2",
    "controller": "did:UIMP:mychannel:123",
    "publicKeyPem": "-----BEGIN PUBLIC
KEY-----MII8YbF3s8q3c...j8Fk88FsRa3K\n-----END PUBLIC KEY-----",
    "expired": 0L
}
```

注：仅当 expired为0L代表公钥是当前有效状态。

A.2 DID文档内Service结构说明

表 5 DID 文档内 Service 结构

名称	类型	必填	说明
id	String	是	服务 ID，全局唯一。
serviceEndpoint	String	是	是有效 url 地址。

示例：

```
{
    "id":"did:example:123#bar", "type": "LinkedDomains",
"serviceEndpoint": "https://bar.example.com"
```

}

A.3 DID文档内metadata结构说明

表 6 DID 文档内 metadata 结构

名称	类型	说明
version	String	DID 文档版本 ID。
ownerdid	String	软件 DID 生产主体 ID。
type	String	标识 did 的类型
remark	String	可嵌套字段
url	String	资源的存储地址

示例如下：

```
{
    "version": "xxxxxxx",
    "ownerdid":"did:UIMP:mychannel:didhuawei",
    "type":"software",
    "remark": {...},
    "url":"xxx"//url 是。
}
```

A.4 摘要签名proof结构说明

表 7 摘要签名的 proof 结构体说明

字段名称	类型	说明
type	String	使用的签名算法，例如，"SM2"
created	Long	签名时间，1645201775000
verificationMethod	String	对 properties 签名使用的 verificationMethod，did:methodid:mychannel:123#key-ski,在 CPT 中必然是 publisher 的 verificationMethod
signature	String	对 properties 的签名内容

示例：

```
{
    "type": "SM2",
     "created": 1645201775000,
      "verificationMethod": "did:methodID:mychannel:123#key-ski",
    "signature":
    "eyJhbGciOiJFZERTQSIsImI2NCI6ZmFsc2UsImNyaXQiOlsiYjY0Il19..2xjpkHW6EY-cKD8DrMIkki
```

B2Q_k6kHynTbR7XGgtYR92blQWpL6Q-2nTdQi1rNhJtmHw1wWWssKMO0EdIEnsCw"
}

A.5 didDocumentMetadata结构

表 9 A.5 didDocumentMetadata 结构说明

名称	类型	说明
created	String	DID文档创建时间戳。示例: 2020-12-20 T19:17:47Z
updated	String	DID文档最后一次更新操作时间戳。示例: 2020-12-20T19:17:47Z
deactivated	Bool	表示 DID 是否已停用，默认为false
nextUpdate	String	表示下一次更新操作的时间戳。示例: 2020-12-20 T19:17:47Z
versionId	String	表示文档版本的版本号
nextVersionId	String	表示下一次更新操作的版本号