ICS 35.040

CCS L 71

团体标准

T/AI XXX.XX—XXXX

|  |
| --- |
|  |

信息技术 媒体版权及溯源管理 第3部分：数字权利元数据 WD-v0.8

Information technology - Copyright and traceability management of Media Part 3: Digital Rights Metadata

点击此处添加与国际标准一致性程度的标识

|  |
| --- |
|  |
| （在提交反馈意见时，请将您知道的相关专利连同支持性文件一并附上） |

XXXX - XX - XX实施

中关村视听产业技术创新联盟 发布

XXXX - XX - XX发布

目  次

[前  言 Ⅱ](#_Toc1487125466)

[引  言 Ⅲ](#_Toc1703023276)

[数字权利元数据 1](#_Toc1050152516)

[1 范围 1](#_Toc1892725366)

[2 规范性引用文件 1](#_Toc359963351)

[3 术语和定义 1](#_Toc442606658)

[3.1 定义 1](#_Toc6747798)

[3.2 缩略语 3](#_Toc1943873471)

[4 数字权利元数据 3](#_Toc1012705286)

[4.1 概述 3](#_Toc1729839327)

[4.2 数字权利元数据信息模型 5](#_Toc775955803)

[4.3 版权权利模型和数据格式规范 6](#_Toc1968476437)

[4.3.1 定义和结构 7](#_Toc50410977)

[4.3.2 数据格式规范 7](#_Toc1148733521)

[4.4 版权授权权利模型和数据格式规范 8](#_Toc1865393023)

[4.4.1 定义和结构 8](#_Toc546775008)

[4.4.2 数据格式规范 9](#_Toc565033943)

[4.5 操作许可权利模型和数据格式规范 10](#_Toc1092710255)

[4.5.1 定义和结构 10](#_Toc2048590288)

[4.5.2 数据格式规范 10](#_Toc51658065)

[附　录　A 关于本部分规范与DRM终端的关系 11](#_Toc1299506196)

[附　录　b （资料性） JSON验证 12](#_Toc1299506196)

[附　录　C （资料性） 版权通证示例 20](#_Toc891946182)

[C.1 版权通证示例 20](#_Toc1503624814)

[C.2 版权授权通证示例 21](#_Toc1982174649)

[C.3 操作许可通证示例 23](#_Toc495509832)

[参 考 文 献 25](#_Toc92163358)

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。

本文件由中关村视听产业技术创新联盟提出并归口。

本文件起草单位：北京邮电大学、北京三快网络科技有限公司

本文件主要起草人：郭莉、康天宇、刘书昌、邓磊、刘嘉夕、黄文伟

引  言

媒体版权及溯源管理是贯穿数字内容整个生命周期的技术基础设施，具备良好兼容性的数字权利元数据是在行业内形成共识共信的关键。在《信息技术 媒体版权及溯源管理》标准中，拟由7个部分构成：

——第1部分：系统

——第2部分：溯源元数据

——第3部分：数字权利元数据

——第4部分：数字水印

——第5部分：真实性溯源

——第6部分：区块链

——第7部分：测试

该部分为《信息技术 媒体版权及溯源管理》中的第3部分：数字权利元数据，目的在于规范数字版权确权及其相关权利，制定专用于数字权利元数据的信息模型、数据结构和格式规范。

本文件的发布机构提请注意如下事实，声明符合本文件时，可能涉及到XXX技术相关的专利的使用。

本标准的发布机构提请注意如下事实，声明符合本标准时，可以使用涉及x.y，…，x.y.z条和附录x中中如下34项与音频编解码技术相关的专利的使用。专利名称如下：

本文件的发布机构对于该专利的真实性、有效性和范围无任何立场。

专利持有人已向本文件的发布机构保证，他愿意同任何申请人在合理且无歧视的条款和条件下，就专利授权许可进行谈判。该专利持有人的声明已在本文件的发布机构备案。相关信息可以通过以下联系方式获得：

联 系 人：

通讯地址：

邮政编码：

电子邮件：

电 话：

传 真：

网 址：

请注意除上述已经识别出的专利外，本标准的某些内容有可能涉及专利。本标准的发布机构不承担识别这些专利的责任。

数字权利元数据

1. 范围

数字权利元数据适用于描述数字内容作品在价值传递链中的版权元信息，主要含其著作权人享有的版权权利和约束、被授权人享有的版权权利和约束、终端消费者享有的使用数字产品的权利和约束。

1. 规范性引用文件

下列文件对于本文件的应用是必不可少的。凡是注日期的引用文件，仅所注日期的版本适用于本文件。凡是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。

ITU-T X.1400 Terms and definitions for distributed ledger technology

GB/T 20090.6 信息技术 先进音视频编码 第6部分：数字媒体版权管理

1. 术语和定义
   1. 定义
      1. 非同质化通证

资产的唯一数字化表示，等同于NFT

[来源：ITU-T X.1400]。

* + 1. 主体

一种强制性对象元素，也是抽象类型元素，针对其在价值传递链中所处的不同位置，分为权利所有者、权利发布者和权利接受者。

注：当存在多个主体共同行使某一权利时，可通过主体组对该集合进行定义。

* + 1. 权利

权利所有者可以行使的权利（版权权利或操作许可）集合。

* + 1. 权利所有者

对数字作品享有著作权的主体。

* + 1. 权利发布者

依据著作权规定的权利类型，发布某项或多项数字作品权利的主体。

* + 1. 权利接受者

依据著作权规定的权利类型，被授予某项或多项数字作品权利的主体。

* + 1. 作品（资源）

包含视频、音频、图像、动画、文本、点云等媒体内容的数字资产。

* + 1. 约束

权利所有者对资源使用相应权利时应满足的条件。

* + 1. 义务

权利所有者在行使一定权利的同时承担的各种责任。

* + 1. 版权通证

经某一个或多个确权主体背书，在版权链上以NFT形式存储的某作品的版权权利信息。

* + 1. 版权授权通证

由某一个或多个权利所有者向另一个版权授权接受主体授予，在版权链上以NFT形式存储的对某项作品的授权权利的信息。

* + 1. 操作许可通证

由某一个或多个权利发布者为另一权利接受者颁发，在版权链上以NFT形式存储的对某项作品的使用权利的授权信息。

* + 1. 引用通证标识

被引用的版权通证标识符。

* + 1. 管理主体

在版权链上拥有版权通证管理和控制权的主体。

* + 1. 权利主体组

享有版权通证规定的版权权利的所有权利主体，及其权利的约束与分账比例。

* + 1. 确权主体

版权确权机构。

* + 1. 版权授权接受主体

被著作权人授权在一定约束和限制下行使版权权利的自然人或法人。

注：版权授权接受主体仅具有使用作品财产权的权利。

* + 1. 版权权利类型

《中华人民共和国著作权法》描述的17项著作权人权利的每项分项权利为一种版权权利类型。

* 1. 缩略语

下列缩略语适用于本文件。

AVS 数字音视频编解码技术标准工作组(Audio and Video Coding Standard Workgroup).

NFT 非同质化通证(Non-Fungible Token).

JSON JS对象简谱(JavaScript Object Notation) .

DREL 数字权利描述语言(Digital Rights Expression Language) .

1. 数字权利元数据
   1. 概述

数字权利元数据定义了开放和可信环境中进行数字版权管理的语法和语义规则，旨在为数字作品的出版、交易、分配和消费等操作提供一种具有灵活性、互操作性和兼容性的权利规范机制，用于描述受保护的数字资源、权利和主体之间的关系。

数字权利元数据用于规范数字资源的版权及相关权利的呈现形式，其输入信息包括：数字作品及版权人相关信息、数字作品源文件以及版权局等权威机构出具的确权文件，其输出信息为版权链上NFT所承载的数字版权通证、数字版权授权通证以及数字权利操作许可通证。

数字权利元数据基于GB/T 20090.6核心档及其中定义的AVS-DREL，是在兼容AVS-DREL信息模型的基础上，将内容访问或使用权利向上扩展，使其能够描述版权权利、版权授权权利、操作许可权利三层权利及其相互间关系。操作许可权利与AVS-DREL所描述的权利保持可兼容性。

本标准包括数字权利元数据的信息模型、数据结构和通证绑定。信息模型规范数字权利元数据的语法结构和框架模型。数据结构给定数字权利通证的格式规范。通证绑定作为附录，描述如何以区块链通证的形式对数字权利进行语法描述和验证。

* 1. 数字权利元数据信息模型

数字权利元数据的信息模型概述了数字权利元数据所要描述的权利信息。如图1所示，信息模型主要由版权权利、版权授权权利、操作许可权利三种权利信息构成。

* 1）版权权利是版权所有者的权利，为第一层权利，用于描述数字资源的著作权人依据《著作权法》所持有的各项版权权利。
* 2）版权授权权利是版权授权接受者的权利，为第二层权利，用于描述数字资源的著作权人通过授权的形式将部分或全部版权权利授予接受者，使其可依法在一定范围内受一定限制地代表著作权人行使版权权利。
* 3）操作许可权利是使用数字作品的终端消费者拥有的权利，为第三层权利。著作权人、版权授权接受者通过发放操作许可，允许其用户在一定约束和义务限制下对数字媒体资源进行播放、下载、复制等操作。

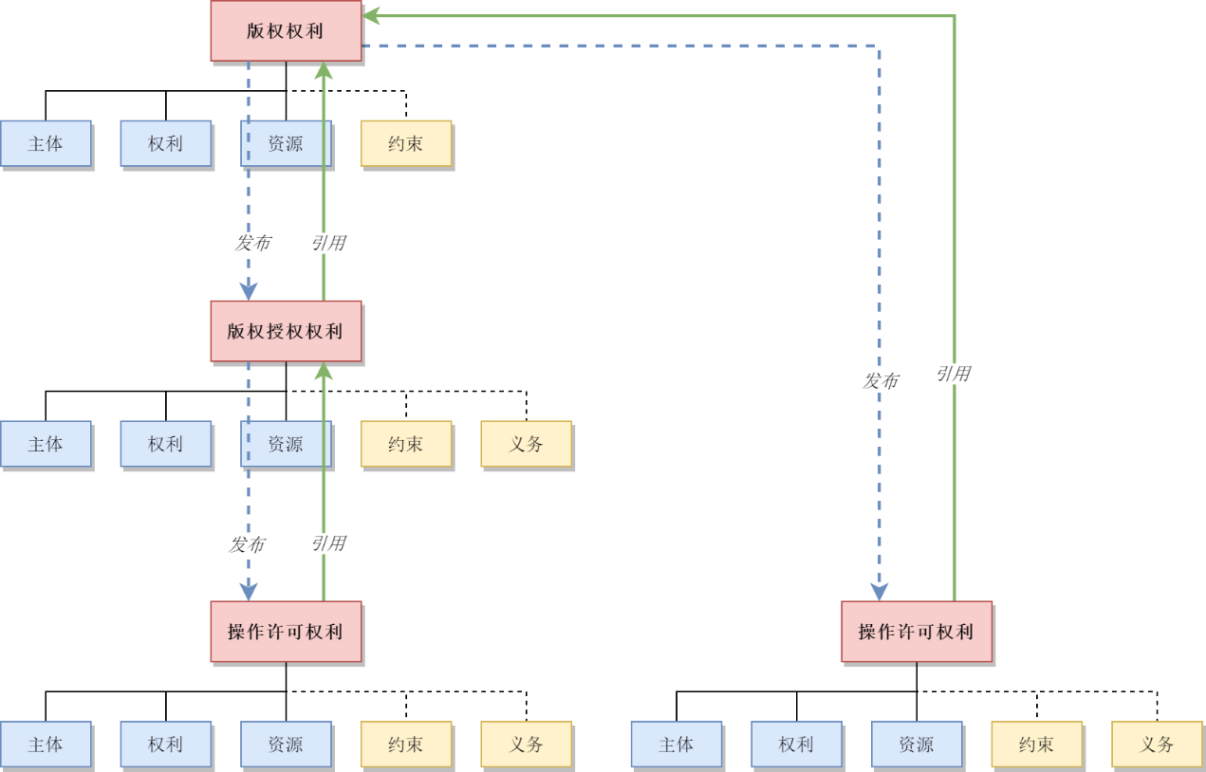


图1 数字权利元数据信息模型结构图

三种权利信息之间的层次关系如图2所示。顶层权利能发布底层权利，同时底层权利通过引用标识其权利来源，下层权利的存在依赖于上层权利。

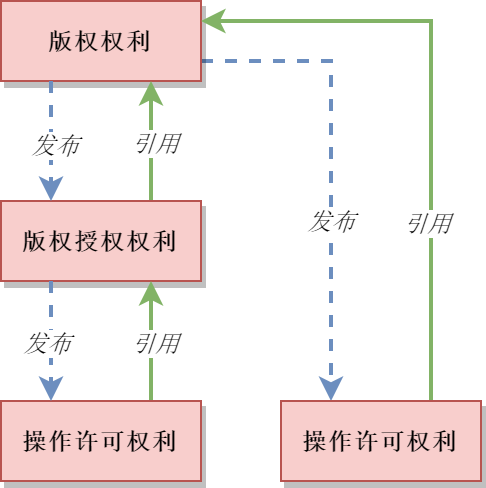


图2 数字权利信息关系图

版权/授权/操作许可权利由主体、权利、资源、约束、义务等要素组成。图3所示的数字权利构成概要图，用于描述权利信息的基本要素，并不表示各要素之间的层次关系。要素的具体含义请参见本文第3章。

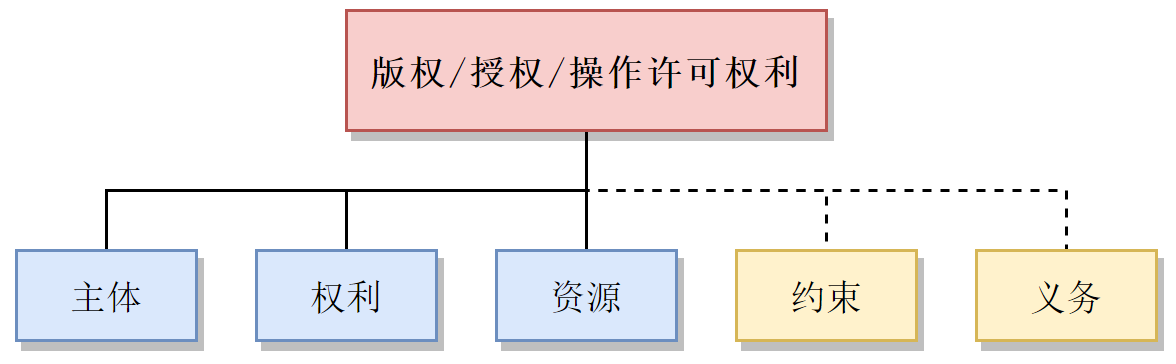


图3 数字权利构成概要图

* 1. 版权权利模型和数据格式规范
     1. 定义和结构

版权权利的基本单位为版权通证，即权利主体（著作权人）在版权链上发布的，经过某一个或多个确权主体（通常是版权行政管理机构）确权的，存储某项数字作品的版权权利信息。通过版权通证中版权权利和约束的定义，明确权利主体享有版权权利的类型、范围及限制。

版权通证由管理主体、确权主体、版权权利、权利主体组、资源等要素组成（如图4所示），其具体含义请参见本文第3章。

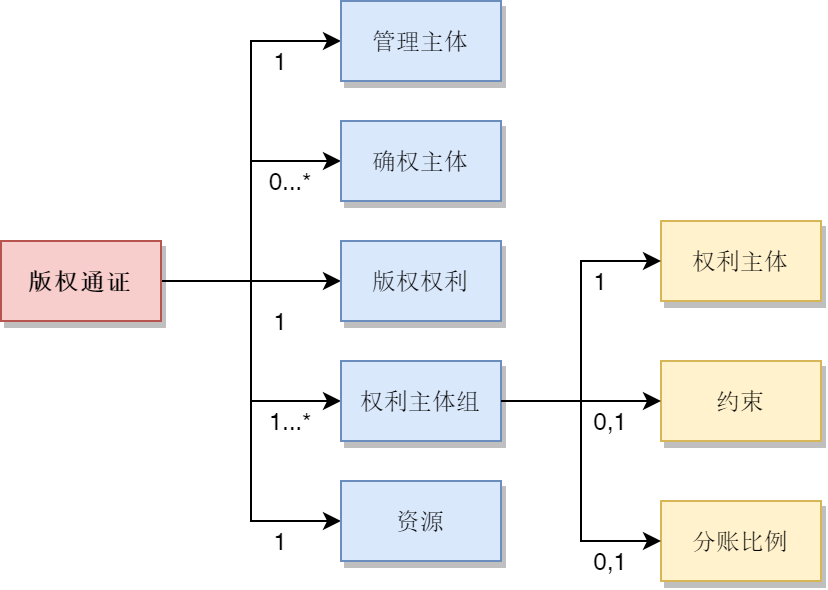


图4 版权通证模型结构图

* + 1. 数据格式规范

数字版权通证的数据格式要求见下表。

表1 数字版权通证数据格式要求

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 编号 | 元素名称 | 元素标识符 | 是否必须 | 数据  类型 | 限制 | 备注 |
| 1 | 版权通证 | copyrightId | Y | String | 固定256bit |  |
| 2 | 管理主体 | adminAddress | Y | String | 合法的区块链地址 |  |
| 3 | 确权主体 | authenticationInstitudeName | N | Array | 合法的区块链地址 |  |
| 4 | 版权权利 |  |  |  |  | 每个版权通证只对应一项版权分项权利 |
| 4.1 | 版权权利类型 | copyrightType | Y | Number | 0-16 | 0-发表权1-署名权2-修改权3-保护作品完整权4-复制权5-发行权6-出租权7-展览权8-表演权9-放映权10-广播权11-信息网络传播权12-摄制权13-改编权14-翻译权15-汇编权16-其他 |
| 4.2 | 权利取得方式 | copyrightGetType | Y | Number | 0-5 | 0-原创1-转让2-继承3-承受4-赠予5-其他 |
| 5 | 权利主体组 | copyrightUnit | Y |  |  |  |
| 5.1 | 权利主体 | address | Y | String | 合法的区块链地址 |  |
| 5.2 | 约束 | constraint | N |  |  |  |
| 5.2.1 | 版权约束 | copyright Constraint | N | Array |  |  |
| 5.2.2 | 授权约束 | appr Constraint | N | Array |  |  |
| 5.2.3 | 许可约束 | licenseConstraint | N | Array |  |  |
| 5.3 | 分账比例 | proportion | N | String | 不超过16bit | 值为百分数。 |
| 6 | 资源 |  |  |  |  |  |
| 6.1 | 作品标识 | workID | Y | String | 固定256bit |  |

* 1. 版权授权权利模型和数据格式规范
     1. 定义和结构

版权授权权利的基本单位为版权授权通证，即版权主体（著作权人）在版权链上发布的，存储版权授权接受主体（被授权人）在一定约束和限制下行使的版权权利信息。通过版权授权通证中版权授权权利和约束的定义，明确版权授权接受主体享有版权权利的类型、范围及限制。

版权授权通证由版权授权接受主体、引用通证标识、授权类型、约束、义务等要素组成（如图5所示），其具体含义请参见本文第3章。

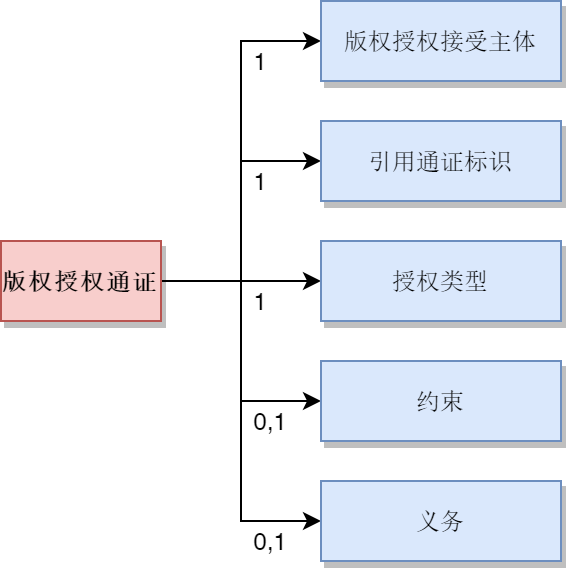


图 5 版权授权通证模型结构图

* + 1. 数据格式规范

数字版权授权通证的数据格式要求见下表2

表2 数字版权授权通证数据格式要求

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 编号 | 元素名称 | 元素标识符 | 是否必须 | 数据  类型 | 限制 | 备注 |
| 1 | 版权授权通证 | approveID | Y | String | 固定256bit |  |
| 2 | 版权授权接受主体 | address | Y | String | 合法的区块链地址 | 版权授权通证的唯一所有者，不能是一个群体 |
| 3 | 引用通证标识 | refID | Y | String | 固定256bit |  |
| 4 | 授权类型 | approveType | Y | Number | 0-2 | 0-普通1-独家2-独占 |
| 5 | 约束 | constraint | N |  |  |  |
| 5.1 | 类型的约束 | approveStatus | N | Number | 0-1 | 0-不可转让1-可转让 |
| 5.2 | 渠道的约束 | approveChannel | N | Number | 0-4 | 0-网络1-全渠道2-产品发布会3-电视4-网络电影 |
| 5.3 | 空间的约束 | approveArea | N | Number | 0-2 | 0-中国1-亚洲2-世界 |

表2（续）

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 编号 | 元素名称 | 元素标识符 | 是否必须 | 数据  类型 | 限制 | 备注 |
| .4 | 时间的约束 | approveTime | N | Number | 0-3 | 0-半年1-一年2-三年3-永久 |
| 6 | 义务 | duty | N |  |  |  |
| 6.1 | 计酬方式 | distributionMethod | N | Number | 0-3 | 0-直接付款1-收入分成2-计次付款3-计时付款 |
| 6.2 | 计酬描述 | distributionDesc | N | String | 不超过4096bit |  |
| 6.3 | 应收酬金 | deliverablePayment | N | String | 不超过64bit | 若计酬方式为“收入分成”，该字段填写为分成比例 |
| 6.4 | 已收酬金 | deliveredPayment | N | String | 不超过64bit |  |
| 6.5 | 未收酬金 | toDeliverPayment | N | String | 不超过64bit |  |
| 6.6 | 结算日期 | balanceDate | N | String | 不超过16bit |  |

* 1. 操作许可权利模型和数据格式规范
     1. 定义和结构

操作许可权利的基本单位为操作许可通证，即著作权人或其被授权人在版权链上发布的，存储许可权利接受者（一般为其用户）在一定约束和义务限制下对数字媒体资源进行播放、下载、复制等操作的许可信息。操作许可通证由引用通证标识、操作许可单元等要素组成，其具体含义请参见本文第3章。

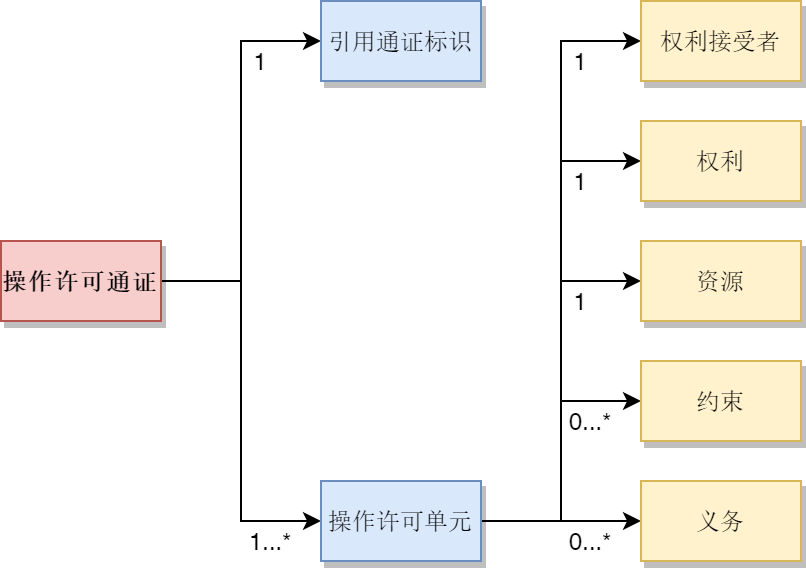


图6 操作许可通证模型结构图

* + 1. 数据格式规范

操作许可通证基本数据格式要求见表3，主体元素的数据格式要求见表4。权利、资源、约束和义务等元素的数据格式要求参照GB/T 20090.6的AVS-DREL数据字典列表中相应数据元素。

表3 操作许可通证基本数据格式要求

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 编号 | 名称 | 标识符 | 是否必须 | 数据  类型 | 限制 | 备注 |
| 1 | 操作许可通证 | licenseID | Y | String | 固定256bit |  |
| 1.1 | 操作许可单元 | licenseUnit | Y | String | 不超过4096bit | 许可证基本组成单元 |

表4 主体数据格式要求

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 编号 | 名称 | 标识符 | 是否必须 | 数据  类型 | 限制 | 备注 |
| 2 | 主体 | subjectType |  |  |  |  |
| 2.1 | 引用通证标识 | refID | Y | String | 固定256bit |  |
| 2.2 | 权利接受者 | address | Y | String | 合法的区块链地址 |  |

（资料性）

关于本部分规范与DRM终端的关系

如下图所示，本部分用于规范《信息技术 媒体版权及溯源管理 第6部分：区块链》标准中媒体版权及溯源管理区块链上实现的三类数字权利通证的元数据域数据格式。媒体版权及溯源管理区块链通过操作许可通证实现与AVS-DRM终端兼容。

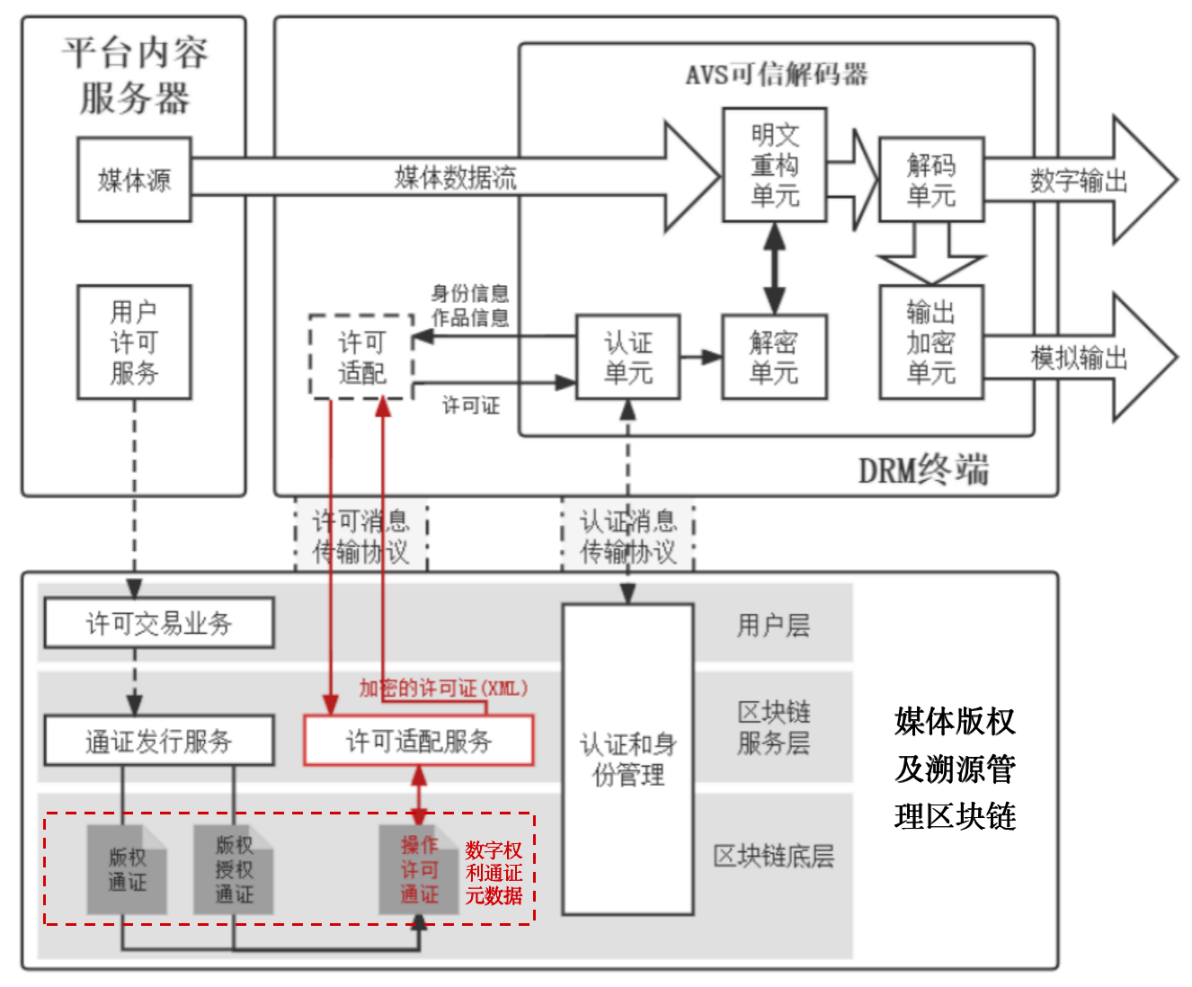


图 数字权利元数据在媒体版权及溯源管理区块链中的定位及与AVS-DRM终端的兼容关系

1. （资料性）  
   JSON验证

数字权利元数据可以由多种描述性语言表述，在此以JSON作为示例。

版权通证通过以下过程生成。

|  |
| --- |
| function genCopyrightToken(copyrightInfo) {  const copyrightSchema = Joi.object().keys({  address:  blockchainCustom.blockchain().address().required(),  copyrightUnit:  Joi.array().items(Joi.object().keys({  address:  blockchainCustom.blockchain().address().required(),  proportion:  Joi.number().min(0).max(100).required(),  copyrightExplain:  Joi.string().required(),  })).required(),  workID:  Joi.string().hex().length(64).required(),  copyrightType:  Joi.number().integer().min(0).max(16).required(),  copyrightGetType:  Joi.number().integer().min(0).max(5).required(),  });  // 校验数据  try {  copyrightSchema.valIDate(copyrightInfo);  } catch(err) {  err.details.map((detail, index) => {  console.log('error message ' + index + ':', detail.message);  });  }  // 组合通证请求数据  let address = copyrightInfo.address;  delete copyrightInfo.address;  let defaultInfo = {  authenticationInfo: [],  copyrightStatus: 0,  };  copyrightInfo = Object.assign(copyrightInfo, defaultInfo);  let tokenID = sha256(copyrightInfo.workID + copyrightInfo.copyrightType).toString();  remote.connect(function(err, res) {  if (err) {  return console.log('err:',err);  }  let seq = (await requestAccountInfo(publisher.address, remote)).account\_data.Sequence; // 获取发行者账号信息的序列号  await buildPubTokenTx(remote, publisher.address, publisher.secret, seq, address, 'copyrightToken', tokenID, copyrightInfo, true); // 调用接口，生成版权通证  remote.disconnect();  });  } |

授权通证通过以下过程实现。

|  |
| --- |
| function genApproveToken(approveInfo) {  const approveSchema = Joi.object().keys({  address:  blockchainCustom.blockchain().address().required(),  copyrightID:  Joi.string().hex().length(64).required(),  approveType:  Joi.number().integer().min(0).max(2).required(),  constraint:  Joi.object().keys({  approveChannel:  Joi.number().integer().min(0).max(4).required(),  approveArea:  Joi.number().integer().min(0).max(2).required(),  approveTime:  Joi.number().integer().min(0).max(3).required(),  approveStatus:  Joi.number().integer().min(0).max(1).required(),  reapproveType:  Joi.number().integer().min(0).max(2).required(),  }),  duty:  Joi.array().items(Joi.object().keys({  stockInfo:  Joi.array().items(Joi.object().keys({  address:  blockchainCustom.blockchain().address().required(),  proportion:  Joi.number().min(0).max(100).required(),  })).required(),  incomeDistribution:  Joi.array().items(Joi.object().keys({  distributionMethod:  Joi.number().integer().min(0).max(3).required(),  distributionDesc:  Joi.string().required(),  receivablePayment:  Joi.number().required(),  receivedPayment:  Joi.number().required(),  toReceivePayment:  Joi.number().required(),  balanceDate:  Joi.date().format('yyyy-MM-dd').required(),  })).required(),  })),  refAddress:  blockchainCustom.blockchain().address().required(),  refSecret:  blockchainCustom.blockchain().secret().required(),  });  // 校验数据  try {  approveSchema.valIDate(approveInfo);  } catch(err) {  err.details.map((detail, index) => {  console.log('error message ' + index + ':', detail.message);  });  }  // 组合通证请求数据  let address = approveInfo.address;  let copyrightID = approveInfo.copyrightID;  let refAddress = approveInfo.refAddress;  let refSecret = approveInfo.refSecret;  delete approveInfo.address;  delete approveInfo.copyrightID;  delete approveInfo.refAddress;  delete approveInfo.refSecret;  let tokenID = sha256(approveInfo).toString();  remote.connect(function(err, res) {  if (err) {  return console.log('err:',err);  }  let seq = (await requestAccountInfo(publisher.address, remote)).account\_data.Sequence; // 获取发行者账号信息的序列号  await buildPubRefTokenTx(remote, publisher.address, publisher.secret, seq, address, 'approveToken', tokenID, approveInfo, copyrightID, refAddress, refSecret, true); // 调用接口，生成授权通证  remote.disconnect();  });  } |

 操作许可通证通过以下过程实现。

|  |
| --- |
| function genlicenseToken(licenseInfo) {  const licenseSchema = Joi.object().keys({  address:  blockchainCustom.blockchain().address().required(),  refID:  Joi.string().hex().required(),  licenseUnit:  Joi.array().items(Joi.object().keys({  right:  Joi.array().items(Joi.number().integer().min(0).max(10)).unique().required(),  constraint:  Joi.object().keys({  spatial:  Joi.string().required(),  bounds:  Joi.object().keys({  count:  Joi.number().integer().positive(),  range:  Joi.object().keys({  min:  Joi.number(),  max:  Joi.number(),  }),  }).required(),  temporal:  Joi.object().keys({  dateTime:  Joi.object().keys({  start:  Joi.date().format('yyyy-MM-dd HH:mm:ss'),  end:  Joi.date().format('yyyy-MM-dd HH:mm:ss'),  fixed:  Joi.date().format('yyyy-MM-dd HH:mm:ss'),  }),  accumulated:  Joi.number().integer().positive(),  interval:  Joi.number().integer().positive(),  period:  Joi.object().keys({  periodType:  Joi.number().integer().min(0).max(2).required(),  start:  Joi.date().format('yyyy-MM-dd HH:mm:ss'),  end:  Joi.date().format('yyyy-MM-dd HH:mm:ss'),  fixed:  Joi.date().format('yyyy-MM-dd HH:mm:ss'),  }),  }).required(),  system:  Joi.object().keys({  cpu:  Joi.array().items(Joi.string()),  screen:  Joi.array().items(Joi.string()),  storeDevice:  Joi.array().items(Joi.string()),  memory:  Joi.array().items(Joi.string()),  printer:  Joi.array().items(Joi.string()),  drmVersion:  Joi.array().items(Joi.string()),  decVersion:  Joi.array().items(Joi.string()),  }).required(),  }),  duty:  Joi.object().keys({  paymentMethod:  Joi.object().keys({  paymentMethodType:  Joi.number().integer().min(0).max(2).required(),  fee:  Joi.object().keys({  amount:  Joi.number().positive().required(),  currency:  Joi.number().integer().required(),  taxPercent:  Joi.number().min(0).max(100),  currency:  Joi.number().integer(),  }).required(),  }),  accept:  Joi.string(),  register:  Joi.string(),  tracked:  Joi.string(),  }),  })).required(),  refAddress:  blockchainCustom.blockchain().address().required(),  refSecret:  blockchainCustom.blockchain().secret().required(),  });  // 校验数据  try {  licenseSchema.valIDate(licenseInfo);  } catch(err) {  err.details.map((detail, index) => {  console.log('error message ' + index + ':', detail.message);  });  }  // 组合通证请求数据  let address = licenseInfo.address;  let refID = licenseInfo.refID;  let refAddress = licenseInfo.refAddress;  let refSecret = licenseInfo.refSecret;  delete licenseInfo.address;  delete licenseInfo.refID;  delete licenseInfo.refAddress;  delete licenseInfo.refSecret;  let tokenID = sha256(licenseInfo).toString();  remote.connect(function(err, res) {  if (err) {  return console.log('err:',err);  }  let seq = (await requestAccountInfo(publisher.address, remote)).account\_data.Sequence; // 获取发行者账号信息的序列号  await buildPubRefTokenTx(remote, publisher.address, publisher.secret, seq, address, 'licenseToken', tokenID, licenseInfo, refID, refAddress, refSecret, true); // 调用接口，生成操作许可通证  remote.disconnect();  });  }  // 获取区块链账号的序列号  export function requestAccountInfo(a, r, showRes) {  let req = r.requestAccountInfo({  account: a  });  return new Promise((resolve, reject) => {  req.submit(function(err, result) {  if(err) {  console.log('err:', err);  reject(err);  }  else if(result) {  if(showRes) {  console.log('requestAccountInfo', result);  }  resolve(result);  }  });  });  } |

以下过程用于发行上级通证。

|  |
| --- |
| function buildPubTokenTx(remote, publisher, secret, seq, dest, name, ID, tokenInfos, showRes) {  let tx = remote.buildPubTokenTx({  publisher: publisher,  receiver: dest,  token: name,  tokenID: ID,  tokenInfos: obj2tokenInfos(tokenInfos),  });  tx.setSecret(secret);  if(seq) {  tx.setSequence(seq);  }  return new Promise((resolve, reject) => {  tx.submit(function(err, result) {  if(err) {  console.log('err:',err);  reject(err);  }  else if(result){  if(showRes) {  console.log('buildPubTokenTx:', result);  }  else {  console.log('buildPubTokenTx:', result.engine\_result + "\_" + result.tx\_json.Sequence);  }  resolve(result);  }  });  });  } |

以下过程用于发行下级通证。

|  |
| --- |
| function buildPubRefTokenTx(remote, publisher, secret, seq, dest, name, ID, tokenInfos, refID, refAddr, refSecr, showRes) {  let tx = remote.buildPubTokenTx({  publisher: publisher,  receiver: dest,  token: name,  tokenID: ID,  tokenInfos: obj2tokenInfos(tokenInfos),  referenceID: refID, // 发行该通证关联的上级版权通证ID  }); // 调用ERC721原生接口，发行通证  tx.setSecret(secret);  if(seq) {  tx.setSequence(seq);  }  tx.ownerSign({  account: refAddr,  secret: refSecr,  }); // 通过ownerSign({})方法完成通证关联的版权通证owner签名  return new Promise((resolve, reject) => {  tx.submit(function(err, result) {  if(err) {  console.log('err:',err);  reject(err);  }  else if(result){  if(showRes) {  console.log('buildPubRefTokenTx:', result);  }  else {  console.log('buildPubRefTokenTx:', result.engine\_result + "\_" + result.tx\_json.Sequence);  }  resolve(result);  }  });  });  }  // 格式化为tokenInfos，规范输入  function obj2tokenInfos(obj) {  let tokenInfos = [];  for(let k in obj) {  let tokenInfosObj = {  type: k,  data: obj[k].toString()  };  tokenInfos.push(tokenInfosObj);  }  return tokenInfos;  }  const blockchainCustom = Joi.extend((joi) => {  return {  type: 'blockchain',  base: joi.string(),  messages: {  'blockchain.hash': '{{#label}} is not a valID hash',  'blockchain.address': '{{#label}} is not a valID address',  'blockchain.secret': '{{#label}} is not a valID secret',  },  rules: {  hash: {  valIDate(value, helpers) {  if(!u.isValIDHash(value)) {  return helpers.error('blockchain.hash');  }  }  },  address: {  valIDate(value, helpers) {  if(!u.isValIDAddress(value)) {  return helpers.error('blockchain.address');  }  }  },  secret: {  valIDate(value, helpers) {  if(!u.isValIDSecret(value)) {  return helpers.error('blockchain.secret');  }  }  },  },  };  }); |

1. （资料性）  
   版权通证示例

C.1 版权通证示例

|  |
| --- |
| {  "TokenInfo": {  "Flags": 0, // 冻结标志位；0-可以流通，1-不可以流通  "FundCode": "copyrightToken",  "Issuer": "jGt4jgACfyFaW7AkPb48PRsD64491M85GF",  "LedgerEntryType": "ERCToken",  "LowNode": "0000000000000269",  "PreviousTxnID": "4FA751D356BAB7D3C5560D02FC94ECA577B676DA1F29E99BD181948E458167F6",  "PreviousTxnLgrSeq": 152046,  "TokenID": "000151817102F8E8A609E69A6BED545EBEC2EB44ED5C8E85514055497C2B0C7F", // 通证ID  "TokenInfos": [  {  "TokenInfo": {  "type": "authenticationInfo",  "data": [  {  "authenticationInstitudeName": role1.address,  "authenticationId": 'rightId001',  "authenticatedDate":'2021-12-31 07:10:00'  },  {  "authenticationInstitudeName": role2.address,  "authenticationId": 'rightId002',  "authenticatedDate":'2021-12-31 07:10:00'  }  ],  "account": "jGt4jgACfyFaW7AkPb48PRsD64491M85GF"    }, // 确权信息  {  "TokenInfo": {  "type": "copyrightUnit",  "data": [  {  "address": "jnuf1Hrd5cMZaxDHBPYPzmLvgt4svvSpjG",  "proportion": "40",  "copyrightExplain": ""  },  {  "address": "jML2zGdi1udiNojfdq7pYA9P78MUzFA3fS",  "proportion": "60",  "copyrightExplain": ""  }  ],  "account": "jGt4jgACfyFaW7AkPb48PRsD64491M85GF"  }  }, // 权利主体组，包含分账比例信息  {  "TokenInfo": {  "type": "workID",  "data": "A76B2AF54FE131E4A32CA225946F38F110D205E712AE071B49ABED2855003CE4",  "account": "jGt4jgACfyFaW7AkPb48PRsD64491M85GF"  }  }, // 作品标识  {  "TokenInfo": {  "type": "copyrightType",  "data": "4",  "account": "jGt4jgACfyFaW7AkPb48PRsD64491M85GF"  }  }, // 版权权利类型  {  "TokenInfo": {  "type": "copyrightStatus",  "data": "1",  "account": "jGt4jgACfyFaW7AkPb48PRsD64491M85GF"  }  }, // 版权状态  {  "TokenInfo": {  "type": "copyrightGetType",  "data": "0",  "account": "jGt4jgACfyFaW7AkPb48PRsD64491M85GF"  }  } // 权利取得方式  ],  "TokenOwner": "jnuf1Hrd5cMZaxDHBPYPzmLvgt4svvSpjG", //管理主体  "index": "DBD19F1ABA6F2DC763AD953DE33A6FD9F0B3E010B8FC4DEF7537C8D5AA10300B"  },  "index": "DBD19F1ABA6F2DC763AD953DE33A6FD9F0B3E010B8FC4DEF7537C8D5AA10300B",  "ledger\_hash": "EFD55CC624F0CDFEC8BA993424541D019DC4D1074B8B829ED5FBC8CFD60766DA",  "ledger\_index": 822532,  "valIDated": true  } |

C.2 版权授权通证示例

|  |
| --- |
| {  "TokenInfo": {  "Flags": 0,  "FundCode": "approveToken",  "Issuer": "jJB4YxZz3haRJ7Njh86SHtGd7nk7xtyPR6",  "LedgerEntryType": "ERCToken",  "LowNode": "0000000000000000",  "PreviousTxnID": "E33B003B58D3543542C6E5242EBF4B224928D0E4218EB4D5C7B58F239A9F47A9",  "PreviousTxnLgrSeq": 50858,  "ReferenceID": "9BA218C98660BD3219DFDEACA8269F7748A33C09011AB70A5B31BCBF42CFF9B1", // 引用通证标识  "TokenID": "010C55E226F9FBED6BF509B4135BC5FCD4FAB940C3FFDB5FD15BE5CD7F0B3A8A", // 授权通证ID  "TokenInfos": [  {  "TokenInfo": {  "type": "approveType",//授权类型  "data": "0",  "account": "jJB4YxZz3haRJ7Njh86SHtGd7nk7xtyPR6"  }  },  {  "TokenInfo": {  "type": "constraint",  "data": {  "approveStatus": "0",  "approveChannel": "0",  "approveArea": "2",  "approveTime": "1",  "reapproveType": "1"，  },  "account": "jJB4YxZz3haRJ7Njh86SHtGd7nk7xtyPR6"  }  }, // 约束  {  "TokenInfo": {  "type": "duty",  "data": {  "stockInfo": [  {  "address": "jML2zGdi1udiNojfdq7pYA9P78MUzFA3fS",  "proportion": "60",  },  {  "address": "jnuf1Hrd5cMZaxDHBPYPzmLvgt4svvSpjG",  "proportion": "40",  }  ], // 分账信息  "incomeDistribution": {  "distributionMethod": "0",  "distributionDesc": "",  "receivablePayment": "1000",  "receivedPayment": "1000",  "toReceivePayment": "0",  "balanceDate": "2021-08-30"  } // 计酬信息，授权通证的收益分配方式  },  "account": "jJB4YxZz3haRJ7Njh86SHtGd7nk7xtyPR6"  }  } // 义务  ],  "TokenOwner": "j4M6Hb65iKUVyzRNVYCCvqnpB6q4j7TT9Q",//管理主体  "index": "76EE00A098E0AAE5DF42926633A6B2DDFACF64686ED7B25D30B59894FB99BA3C"  },  "index": "76EE00A098E0AAE5DF42926633A6B2DDFACF64686ED7B25D30B59894FB99BA3C",  "ledger\_hash": "61F3D694EA9606DBE9CB6C987EF2C518EB01BBD508B0D7EFB49E6DA081A321CA",  "ledger\_index": 822591,  "valIDated": true  } |

C.3 操作许可通证示例

|  |
| --- |
| {  "TokenInfo": {  "Flags": 0,  "FundCode": "licenseToken",  "Issuer": "jJB4YxZz3haRJ7Njh86SHtGd7nk7xtyPR6",  "LedgerEntryType": "ERCToken",  "LowNode": "0000000000000000",  "PreviousTxnID": "E33B003B58D3543542C6E5242EBF4B224928D0E4218EB4D5C7B58F239A9F47A9",  "PreviousTxnLgrSeq": 50858,  "ReferenceID": "9BA218C98660BD3219DFDEACA8269F7748A33C09011AB70A5B31BCBF42CFF9B1",  "TokenID": "010C55E226F9FBED6BF509B4135BC5FCD4FAB940C3FFDB5FD15BE5CD7F0B3A8A",  "TokenInfos": [  {  "TokenInfo": {  "type": "licenseUnit",  "data": [  {  "right": ["0", "1", "2"], // 操作许可类型；0-播放，1-输出，2-打印，3-修改，4-分割，5-打包，6-移动，7-复制，8-备份，9-保存，10-撤销  "constraint": {  "spatial": "囯内", // 操作许可地域  "bounds": {  "count": "1000",  "range": {  "min": "0",  "max": "10"  }  },  "temporal": {  "dateTime": {  "start": "2021-08-30 24:00:00",  "end": "2022-08-30 24:00:00"  }  }, // 操作许可时间  "system": {  "cpu": ["Intel"],  "screen": ["AOC"]  }  },//约束  "duty": {  "paymentMethod": {  "paymentMethodType": "0",  "fee": {  "amount": "1000",  "currency": "CNY",  "taxPercent": "10",  "code": "CNY"  },  "accept": "接受版权声明",  "tracked": "可追踪"  }  } // 义务  }  ],  "account": "jJB4YxZz3haRJ7Njh86SHtGd7nk7xtyPR6"  }  } // 操作许可权利单元  ],  "TokenOwner": "j4M6Hb65iKUVyzRNVYCCvqnpB6q4j7TT9Q", //管理主体  "index": "76EE00A098E0AAE5DF42926633A6B2DDFACF64686ED7B25D30B59894FB99BA3C"  },  "index": "76EE00A098E0AAE5DF42926633A6B2DDFACF64686ED7B25D30B59894FB99BA3C",  "ledger\_hash": "61F3D694EA9606DBE9CB6C987EF2C518EB01BBD508B0D7EFB49E6DA081A321CA",  "ledger\_index": 822591,  "valIDated": true  } |

参 考 文 献

[1] GB/T 20090.6 信息技术 先进音视频编码 第6部分：数字媒体版权管理

[2] ITU-T X.1400 Terms and definitions for distributed ledger technology

[3] 中国人大网.中华人民共合作著作权法[EB/OL]. <http://www.npc.gov.cn,> 2020-11-09