Encoding of Media Value Chain Processes Through Blockchains and MPEG-21 Smart Contracts for Media

Journal: IEEE Multimedia, August 2023

Author: Mirko Zichichi; Victor Rodriguez-Doncel

Publisher: IEEE

The final published version is available online at:
https://doi.org/10.1109/MMUL.2023.330339

Encoding of Media Value Chain Processes Through Blockchains and MPEG-21 Smart Contracts for Media

Mirko Zichichi, Víctor Rodríguez-Doncel
Ontology Engineering Group
Universidad Politécnica de Madrid
ETSIINF, Boadilla del Monte (MD), Spain
mirko.zichichi@upm.es, vrodriguez@fi.upm.es

***Abstract*—Distributed Ledger Technologies can be used for rights management in the audiovisual production sector, where dominant business models still need to provide a sustainable way to support the claims of content creators and rights holders fully. This paper describes the combination of the current ISO/IEC 21000 standards, which are the MPEG specifications for the multimedia framework, with Distributed Ledger Technologies and smart contracts. Their gathering shapes the Smart Contracts for Media, a specification that can be used to encode the terms and conditions of a contract for media-related delivery and consumption. The MPEG-21 framework includes means to convey the encoding of digital media intellectual property rights information and the media value chain. The involvement of smart contracts in this framework enables the twofold process of reducing the complexity of contract terms compliance validation and making stakeholders more aware of the media value chain. As an example, technical details are provided for a Video-On-Demand Services setting.**

***Index Terms*—MPEG, Distributed Ledger Technology, Smart Contract, Non Fungible Token**

I. Introduction

MPEG (Moving Picture Experts Group), a working group of ISO/IEC, has developed several well-known media encoding standards for audio, video, and genomic information. One of its endeavors is the definition of the Multimedia Framework, known as ISO/IEC 21000 or MPEG-21, which is an open framework for delivering and consuming multimedia [1]. MPEG-21 describes an abstract content capsule, the Digital Item, and the means for its identification, description, adaptation, verification, and quality assessment. Other parts of the standard are concerned with the intellectual property of the works conveyed in the Digital Item. A Rights Expression Language and a Rights Data Dictionary are defined to clearly represent under which conditions the intellectual property works can be consumed and transmitted in the context of a Digital Rights Management platform [2]. The standard also supports the precise description of the Media Value Chain (Media Value Chain Ontology, MVCO [3]) and the representation of contracts transacting the rights of multimedia content, either as XML (Contract Expression Language, CEL [4]) or as RDF (Media Contract Ontology, MCO [5]).

The earliest parts of MPEG-21 allowed the representation of rights in a machine-readable form, a significant advance that enabled access control to audiovisual works in various information systems. The following parts of MPEG-21 leveraged the benefits of the Semantic Web to improve the interoperability of rights expressions, precisely define data models using computer ontologies, and enable description-logic-based authorization algorithms. The newest part of MPEG-21 is titled Smart Contracts for Media (MPEG-21 SCM), International Standard since 2022 [6]. This new part benefits from a new technological paradigm, i.e., Distributed Ledger Technologies (DLTs) and smart contracts, to address a set of very well-defined challenges [7].

Distributed Ledger Technologies can easily fit into the MPEG-21 framework to exploit their advantages. These technologies can act as resonance boxes for instances created using the MPEG-21 framework and directly enforce what has been determined in terms of use of the media [8]. Generally speaking, the introduction of DLTs has led to a renewed consideration of the concept of trust in computer systems [9]. DLTs were initially thought of as systems based on the distribution of a ledger between many nodes in a Peer-to-Peer (P2P) network for transacting digital currencies without the need for a central authority. The first DLT implemented was the Bitcoin blockchain for exchanging value, i.e., cryptocurrency [10]. In this case, the DLT ledger takes the name of a blockchain because the ledger's form is a chained list of blocks. Generally speaking, DLT applications and use cases go far beyond the financial context [10]–[12]. Indeed, the features these technologies provide can reduce the opacity of complex systems processes [13]: (i) transparency, for the append-only ledger, is auditable by the whole network; (ii) immutability, as data cannot be easily tampered with; (iii) traceability and non-repudiation, because each network participant cryptographically signs each transaction issued in the immutable ledger and (iv) decentralized execution of immutable instructions, i.e., smart contracts.

Smart contracts are part of the second revolution brought by DLTs [10]. They are software procedures that can be run to ensure the proper execution of new types of applications directly on a DLT. In some implementations, a smart contract can be considered a specific interpretation and translation of some contractual terms. However, the mere fact that a smart contract is stored on a DLT does not give rise to a legal agreement [14]. More generally, smart contracts can be the means for the automatic performance of all or some parts of the contract, derived from corresponding legal prose, i.e., the written expression of a mutual agreement on contractual terms [15]. Legal contracts need to comply with a complex hierarchy of laws and regulations at the local, national and international levels, which may limit the ability of a smart contract to give rise to a legal agreement and the scope of their enactment [14].

In combination with the MPEG-21 framework, smart contracts can be used to encode the terms and conditions of a contract for media-related asset trading. Smart contracts can be used to establish and enforce IP agreements such as licenses and enable the transmission of real-time payments to IP owners; IP rights information in protected media content, then, can be encoded using the MPEG-21 framework and directly and uniquely linked to a smart contract, i.e., a Smart Contract for Media. In other words, smart contracts can be used to allow music and video media royalties to be administered almost instantaneously and manage usage allowances and restrictions. Indeed, rather than passing through centralized intermediaries, revenue from a stream or download of media content could be distributed automatically to rights holders through a DLT and a smart contract. The SCM instructions are encoded according to agreed terms and conditions (e.g., revenue splits) and executed as soon as an asset is downloaded or streamed. We argue that bringing together the MPEG-21 multimedia framework and smart contracts provides the following advantages: (i) to foster transparency in the media value chain and reduce disputes over royalty amounts; (ii) to provide automatic mechanisms for the execution of the agreements; (iii) to reduce the complexity of validating compliance with contract terms and conditions, including limiting revenue losses due to contract violation and illegal distribution of content; (iv) to enhance disintermediation and render those directly affected, e.g., content creators, more conscious about the media value chain.

In this paper, our main contribution is demonstrating how to exploit SCM to manage IP rights and administer media royalties in the specific context of the MPEG-21 framework. In the following sections, we provide a detailed description of the SCM, its relation with the MPEG-21 media value chain, and a possible implementation using a DLT, i.e., Ethereum [10].

The remainder of this paper is organized as follows. Section II provides a background on the main concepts and technologies used and on related work. Section III presents a description of the Smart Contract for Media, while Section IV provides a possible implementation. In Section V we describe a use case related to video-on-demand services' media value chain and the performance evaluation of its smart contract implementation. Finally, Section VI provides the concluding remarks.

II. Background and Related Work

In this section, we first provide the background related to the technologies involved in the SCM. Then we discuss some related work.

A. Distributed Ledger Technologies (DLTs)

A DLT consists of a set of protocols and components that guarantee untampered data availability thanks to a P2P network, where nodes mutually agree on a shared state while tolerating failures and malicious behavior to some extent. The append-only ledger is extended through transactions that are disseminated throughout the network and that are independently verified by each node in order to ensure their consistency. DLTs, born with the advent of Bitcoin in the form of blockchain, have subsequently been interested in integrating smart contracts in some implementations. A smart contract is a code deployed in a DLT or the source code from which such code was compiled, whose execution of its immutable instructions is distributed among the nodes of the DLT it is deployed to [16]. This execution is triggered via a DLT transaction and will produce a change in the DLT state. Each node executing the instructions receives the same inputs and produces the same outputs, thanks to a shared protocol.

These properties allow the issuer of a smart contract not to require the presence of a trusted human third-party validator to check the terms of an agreement, i.e., the trust is moved from a third party to a protocol where all those implementing it are peers [9]. However, since the smart contract consists of executable code, its issuer must also be sure that the behavior implemented is correct (e.g., through code verification) [14].

The decentralized applications, i.e., dApps [10], that are possible to build on top of DLTs thanks to smart contracts, exploit the verifiability of information stored on the distributed ledger and authentication based only on cryptographic primitives. This new kind of application created the need for standardized ways of representing information on DLTs. The token representation is one of the most used. It is information recorded on a DLT representing some form of right: ownership of an asset, access to a service, receipt of payment, etc. For instance, the fungible token [17] is one of the most used specifications for creating second-layer cryptocurrencies. The Non Fungible Token (NFT) [18] is a utility token usually implemented to represent and transact with (tangible or intangible) assets on DLTs, where every single token is different from the rest of the tokens, i.e., non fungible. More specifically, NFTs combine both concepts of (i) access rights to an underlying economic value (property) [19], and (ii) permission to access someone else's property or services or collective good. The asset considered here can be of many forms: (i) physical property, e.g., houses or unique artwork, (ii) virtual collectibles, e.g., unique pictures or collectible cards, (iii) negative value assets, e.g., loans, burdens, and other responsibilities. In general, NFTs are distinguishable, and the ownership of each one is tracked separately.

B. Semantic Web technologies

Semantic Web technologies [20] bring structure to the meaningful contents of the Web by promoting common data formats and exchange protocols. Linked Data is the form of its most successful incarnation: data are published in a structured manner so that information can be found, gathered, classified, and enriched using annotation and query languages. The World Wide Web Consortium (W3C) has published over the last twenty years a set of specifications to describe resources that simultaneously address these two design goals: those of the Semantic Web. Whereas these specifications were born to represent data on the Web, their use has gone beyond, and today many applications run offline but using the semantic web specifications. The most spread paradigm to represent information is RDF (Resource Description Framework). In this framework, resources are identified with URIs and described with collections of triples. The precise meaning of each resource can be formally established with OWL ontologies. An ontology is a formal representation of knowledge through a set of concepts and relations between these concepts within a specific domain. Through these ontologies, it is possible to convey the meaning of data, facilitating cross-domain applications and services. Ontologies in these scenarios effectively act as data models. Reasoning with the information represented using these data models is feasible because they are mapped in a formal language.

C. Rights Expression Languages

Rights Expression Languages (RELs) are a central component of contemporary digital rights management systems. They are applied to express permissions, obligations, and prohibitions in a machine-readable form. The authors in [21] propose a classification to understand their functionalities and applications, giving an outlook on how RELs are used to explicate machine-readable rights for access control, trust management, and contracting. Among the most prominent RELs, we find the MPEG-21 framework, the Open Digital Rights Language (ODRL), and the eXtensible Access Control Markup Language (XACML). ODRL [22], on the other hand, is a W3C standard that provides an information model, a vocabulary, and encoding mechanisms for representing statements about the usage of content and services. It is based on the use of Semantic Web technologies to simplify the distribution, sharing, and exploitation of statement information across the Web. Indeed, it can be argued that semantic web technologies can contribute to more intelligent and flexible handling of privacy, security, and policy issues, through supporting information integration and sense-making [23]. This is also why semantic web technologies are included in the MPEG-21 framework [1]. MPEG-21 describes an abstract content capsule, the Digital Item, and the means for its identification, description, adaptation, verification, and quality assessment. Other parts of the standard are concerned with the intellectual property of the works conveyed in the Digital Item. A REL and a Rights Data Dictionary represent under which conditions the intellectual property works can be consumed and transmitted in the context of a Digital Rights Management platform.

D. MPEG-21 Contracts Representation

The earliest parts of the MPEG-21 framework allowed the representation of rights in a machine-readable form, i.e., using the eXtensible Markup Language (XML). This significant advance enabled access control to audiovisual works in various information systems. The other parts of MPEG-21 leveraged the benefits of the Semantic Web to improve the interoperability of rights expressions, precisely define data models using computer ontologies, and enable description-logic-based authorization algorithms. The MPEG-21 framework includes languages and ontologies that facilitate the conversion of media narrative contracts to digital ones and enable the creation of new contracts in machine-readable electronic formats. The technologies supporting machine-readable contracts are the ISO/IEC 21000-20, i.e., MPEG-21 Contract Expression Language (CEL), for XML-based environment, and ISO/IEC 21000-21, i.e., MPEG-21 Media Contract Ontology (MCO), also based on ISO/IEC 21000-19, i.e., MPEG-21 Media Value Chain Ontology (MVCO).

The first part we will describe is the part that supports the detailed description of the media value chain, i.e., Media Value Chain Ontology (MVCO) [3]. MVCO is an ontology used to describe the main entities in the media value chain formally: (i) IP entities, which are the objects subject to copyright law protection such as works (e.g., an original song), manifestations (e.g., its music score), instances (e.g., the performance of the song), or products (e.g., a sellable item); (ii) relevant actions that can be performed on those entities (e.g., adapt an original work, perform a specific work), and (iii) types of users whose actions are rights, obligations, or something else provided by IP law (e.g., creator, producer). The Audio Value Chain Ontology (AVCO) extends MVCO functionalities related to the description of composite IP entities in the audio domain [7].

MVCO is supplemented by the representation of contracts transacting content rights, i.e., Media Contract Ontology (MCO) [5]. MCO builds on MVCO's generic deontic statements (incorporating the concepts of permission, prohibition, and obligation) by providing the elements to shape the structure of media contracts (*mco-core*), to express rights to exploit media content (*mco-ipre*) and to define specific obligations for payments and notifications (*mco-pane*) [5].

In the MVCO, AVCO, and MCO cases, the use of RDF is involved. However, the MPEG-21 also includes a part involving the XML, i.e., the Contract Expression Language (CEL) [4]. This part can be considered equivalent to the combination of MVCO and MCO for expressing rights. CEL provides an extensible model for representing generic agreements between parties (*cel-core*) and defines the most common acts and constraints in the media field and is used in digital media contracts (*cel-ipre*) [24].

Music and media value chain actors can use MPEG-21 CEL/MCO standards to share and exchange, in an interoperable manner, all metadata and contractual information related to creative works, leading to transparent payment of royalties. Furthermore, ontologies enable functional inference and reasoning capabilities to derive knowledge and data through facts and logic based on rich semantic copyright models.

Finally, the newest part of MPEG-21, titled Smart Contracts for Media (SCM), builds upon the MVCO, MCO, and CEL, and it is in the final step to becoming International Standard (IS) at the time of editing this work [6]. It is the specification we will refer to for the rest of this work.

E. Related Work

Even if several related works successfully use RELs such as ODRL and XACML concerning decentralized systems, we use the MPEG-21 framework because it includes one of the earliest (if not the only one) standard specifications that links RELs directly to DLT objects, i.e., the ISO/IEC 21000-23 Smart Contract for Media [6], [7].

1) Decentralized (music) rights management: We first focus on the music industry for the media-related use of DLT in related work. DLTs have been considered key technologies by entrepreneurs since their inception. Their promised change is to be able to disintermediate the music industry value chain in global markets. At the heart of this radical change is the idea that “artistic effort” is usually not rewarded fairly under the current system (e.g., streaming platforms such as Spotify and Apple Music) [25]. Revolving around this idea, many firms started using DLTs to develop new venture ideas in the music industry [25]. Ujo Music [26] is one of the first platforms using smart contracts for music, where artists interact directly with end users. It uses the Ethereum blockchain to digitize the rights and metadata of music. A usage example is that users can purchase music, and the money goes directly to the artists. Vezt [27] is a music rights marketplace that offers artists, songwriters, and producers funding directly from their fans. Using DLTs, Vezt allows fans to earn the right to receive royalties earned from the songs they initially backed up. It limits the need for funding from a music label, i.e., fans become a decentralized music label.

Other recent implementations provide solutions not based on the Ethereum blockchain, such as BitSong, based on a proprietary blockchain built on Cosmos [28]. The first successful P2P music-sharing venture at the beginning of the new millennium, i.e., Napster, has been reborn from its ashes to offer an ecosystem based on Algorand [29], to “to empower and align fans, music makers and rights holders” [30]. However, even if these solutions use DLTs and smart contracts, their focus on media rights management is low and only provides some basic mechanisms for royalties management, mostly based on tokens. In this paper, rather than building an ecosystem, we specifically focus on the expression richness the SCM provides thanks to the MPEG-21 framework.

RChain [31] aims to build a decentralized, censorship-resistant, public compute infrastructure and blockchain. It includes an “asset tracker” that allows packaging large data blobs, such as audio or video files, with metadata about the creators and rights holders of data. Some other works focus on DLT-based Digital Rights Management systems [32], [33]. Meng et al. [34] present a blockchain-based digital watermarking framework to provide a copyright management system. They store the watermark information in the blockchain, and the watermarked media is stored in the InterPlanetary File System (IPFS) [35]. García et al. [36] combine copyright ontologies with an Ethereum blockchain to manage the arrangements of social media users and journalists with an interest in their contents.

Other DLT-based digital right management proposals support users’ privacy and data protection. The PrivDRM [37] (Privacy-Preserving DRM) system proposed by Gaber et al. enables consumers to obtain DRM-protected content and its license without disclosing complete personal data. Vishva and Hussein [38] propose a platform that enables media owners to be aware of data collected from them and how these are used, thanks to the combination of a blockchain, encryption mechanisms, and off-chain storage.

2) Policy-based access control and DLTs: All the above-cited works, however, do not deal with the interoperability of rights, declared policies and metadata related to content. Usually, the use of Semantic Web technologies represents to us the core element that eases metadata interoperability. Semantic Web technologies are used to decouple user data from the applications that use this data. Solid [39] is a project where data is stored in an online storage space called Pod, a Web-accessible storage service that can be deployed on personal or public servers. The authors in [39] focus on the notion of GDPR consent and provide a solution based on exploiting ODRL policies. In their solution, the need to ensure informed and explicit consent led to the inclusion of specific information items in the Pod so that the users can access their consent authorizations. Furthermore, their Pod implementation has methods enabling users to update or revoke the consent previously given. The combination of non-DLT-based solutions with DLTs can provide a way through which data access rights can be shared between users and service providers in a transparent and verifiable manner. It is the case especially when service providers need to enforce legitimate data access rights that may take precedence over users’ ones. The authors in [40] and [41] propose a framework to store data generated by Internet of Things devices in Solid with a DLT for validation purposes. Through an authentication mechanism, any third-party application can gain access to the data in the Solid Pod and verify the authenticity of the data by cross-checking the hash of the data on the DLT.

Other scholars use Semantic Web technologies to create new ontologies for declaring policies and then integrate a DLT. It is the case of [42] that provides an architecture in which the DLT is used to provide a purpose-centric access-control model. Other studies in the literature primarily focus on programming smart contracts for automatically managing access control policies [43]–[45]. For instance, authors in [43] designed a model providing access control in which only authorized parties with user consent can access user data and where all activities are recorded in a DLT. Their access control system uses policies declared based on an ontology that extends the Provenance Ontology (PROV-O) [46] and the XACML model [47].

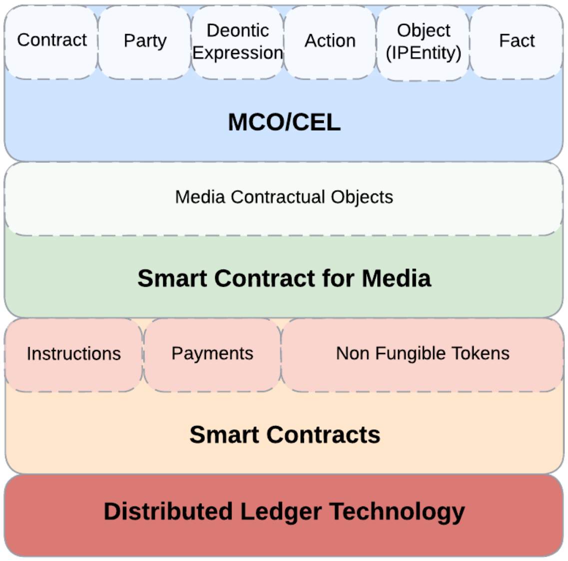

Fig. 1. Smart Contract for Media Environment. A layered view of the technologies and framework that the Smart Contract for Media exploits to execute its functioning.

III. The Smart Contract for Media

In the audiovisual world, contracts are signed at any stage of the value chain: the right to make an adaptation from an original work, the right to broadcast a performance of that work, or the right to distribute that interpretation. Representing this chain of contracts in a blockchain seems a natural way to register these signed commitments.

The Smart Contract for Media specification is a passthrough component designed to be the interlingua that connects the MPEG-21 framework with different DLTs. Thus, it was designed to contain a set of tools interoperable with different types of DLTs and for use in different contexts inherent in the media value chain. In combination with the MPEG-21 framework, smart contracts can be used to encode the terms and conditions of a contract for media-related asset trading. Smart contracts can be used to establish and enforce agreements such as licenses and enable the transmission of real-time access to content recipients. Rights information is protected content, then can be encoded using the MPEG-21 framework and directly and uniquely linked to a smart contract, i.e., an SCM. In other words, smart contracts could allow content policies to be administered almost instantaneously and manage usage allowances and restrictions. The SCM instructions are encoded according to agreed policies and conditions and executed as soon as an asset has to be accessed.

Figure 1 shows a layered view of the environment in which the SCM is executed. Its position is central with regard to the layers related to MPEG-21 MCO/CEL media contracts and the DLT. In particular, the SCM exploits several elements of the media contracts, such as Contract, Party, and Deontic Expression, and encodes new information in a DLT through smart contracts' instructions and NFTs. In this Section, we will go through each layer shown in Figure 1 (top to bottom), all of which constitute the SCM.

A. MPEG-21 MCO/CEL Media Contracts

Narrative contracts share a common structure that consists of a preamble and a body. The MPEG-21 machine-readable contracts expressed in CEL and MCO/MVCO (that, from now on, we will refer to them as MPEG-21 CEL/MCO) are based on this shared structure [7]. The MPEG-21 MCO/CEL contract consists of a series of objects found within the contract structure. We firstly have a main *Contract* object that includes a preamble with:

- contract metadata (e.g., date, version, title);
- contract unique identifier;
- possible relationships with other *Contract* objects (e.g. amendments, prevalence or substitution);
- *Party* objects, representing signatory parties for which the contract is binding.

The *Contract* object also includes the body with:

- the *IP Entity* objects, such as an original work or a music performance, and whose rights are traded in the contract;
- the operative part containing the contact information in the form of *Deontic Expression* objects such as permissions, obligations, and prohibitions. The *Deontic Expression* (or clause) includes:
  - an *Action* object, i.e., the right;
  - a set of *Fact* objects, logically combined (i.e., using union and/or intersection operators) representing the conditions that must be satisfied;
  - an *IP Entity* object, i.e., the media (digital or not) which is the object of the right;
  - a *Party* object, representing the party the *Action* is related to.

  Deontic Expressions can be related among them. For instance, a party has an obligation of payment after broadcasting some media as specified in one of the contract's permission.

B. Structure of a Smart Contract for Media

The SCM is a smart contract that includes or refers to metadata and contractual information connected to creative works, i.e., media, and encodes a contract's terms and conditions. These metadata are called Media Contractual Objects and consist of elements already encoded using the MPEG-21 MCO/CEL representations seen in the previous Sub-Section. The semantic and operational scopes of the original narrative contractual information are bounded to the ones provided using the MPEG-21 CEL/MCO media contracts.

The data handled by SCMs consists of a set of Media Contractual Objects obtained from instances of the MPEG-21 objects described in the previous section. In particular, a conversion process described in the ISO/IEC 21000 Part 23 [48] standard takes as input a MPEG-21 CEL contract or MPEG-21 MCO contract and outputs this standardized set of Media Contractual Objects that is unique for both implementations. Such Media Contractual Objects standard maintains the same high-level objects definition as the one shown in Section III-A, i.e., *Contract, Party, Deontic Expression, Action, IP Entity, Fact*, and their specializations too, e.g., a *Payment* object is a specialization of an *Action* object.

The following describes how the Media Contractual Objects are used in the SCM.

1) Contract: The *Contract* object is the one that includes or refers to the digitalized contractual information extracted from a narrative contract, i.e., the structure, including the preamble and body. A manifestation of a *Contract* object is a unique Smart Contract for Media deployed in a specific DLT. Interoperability of data stored on the DLT can be achieved using simple references, i.e., a smart contract implementing an SCM can reference another SCM through a DLT's smart contract address [10].

2) Parties: The *Party* object represents a human or juridical person bound to the narrative contract. Since identities in DLTs are generally represented through addresses, a *Party* is represented and authenticated in the SCM through a DLT address that, thus, represents this *Party*.

3) IP Entity: The *IP Entity* object encapsulate one or more digital items of intellectual property in the MPEG-21 multimedia framework. Within the scope of a specific SCM contractual information, *IP Entity* objects are uniquely identified on-chain through the use of NFTs. Then, the entire set of information related to a specific *IP Entity* object is linkable to such NFT. Two reasons support this approach: (i) the linkage between IP entities and related SCM is maintained at a high level, particularly when DLTs offer append-only data storage and not a more complex one; (ii) it makes feasible the process of auditing, exploiting at best the immutability feature of DLTs; for the history of all operations executed over an *IP Entity* object, indeed, can be found in one place. For instance, an *IP Entity* object can be represented for the first time by a single NFT with an id equal to *N* in an SCM with an address equal to *X*. Then, another SCM with address *Y* that references that specific *IP Entity* object can reference the id *N* without creating a new NFT (Figure 3).

4) Deontic Expression: The *Deontic Expression* object is included in the body of a *Contract* object and encompasses the properties of an agreed machine-readable contract clause regulating parties' actions and rights. The uniqueness of such an agreement leads to following the same approach used for *IP Entity* objects, i.e., clauses are serialized according to the concept of NFT. The reasons for supporting this approach are: (i) it enables a unique way for storing clauses in DLTs, that is also beneficial in terms of interoperability, in terms of sharing these clauses with other DLT-based applications; (ii) it allows the transfer of value in the form of obligations, permissions and prohibitions, similarly to how cryptocurrency transfers are done.

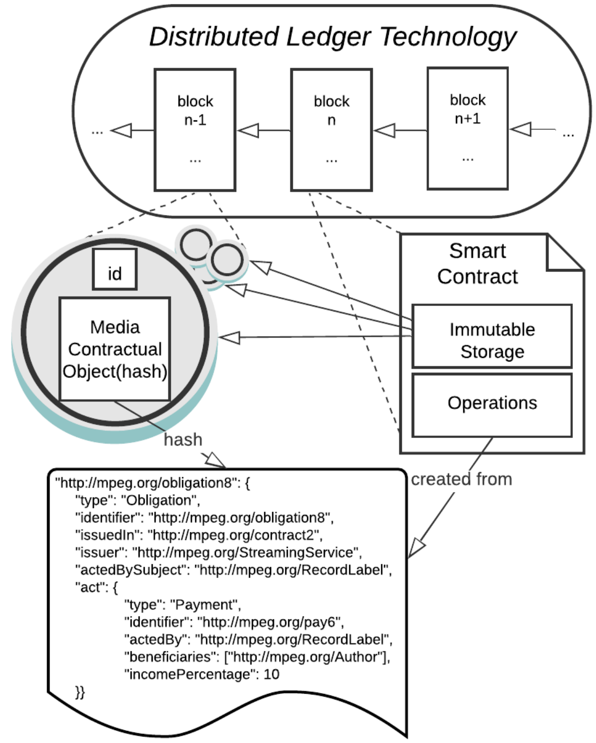

Fig. 2. Smart Contract for Media Structure

C. Smart Contract and Distributed Ledger Technology

From a practical point of view, the SCM can be considered an interface that makes the MPEG-21 framework interoperable with several DLT implementations. Figure 2 shows graphically how the SCM can be subdivided. We can first consider the SCM as a tool to implement and passively enact the operational part of the original narrative media contract. Secondly, we can consider the SCM as immutable storage for the above-referenced MPEG-21 CEL/MCO machine-readable contract information.

1) Instructions: One of the roles adopted by the SCM is to directly and passively enact what is "enactable" (i.e., enforceable) in a DLT, with reference to the clauses indicated in the media contract. We need to elaborate and clarify three points of the previous sentence to capture this other role of the SCM fully:

1) The clauses indicated in the media contract are represented as *Deontic Expression* objects, and the implementation of instructions in the form of smart contract methods are derived from these. For instance, a specific *Payment* object, i.e., a specialization of a *Deontic Expression* object, leads to the creation of a specific payment smart contract method.
2) However, the set of actions that can be included in a media contract is greater than the set of actions that are "enactable" in a DLT'. It means, for instance, that

an *Obligation* object in a contract might limit the exploitation of media in a specific country, but an SCM cannot enact the operation of verification of the exploiter location because the DLT protocol does not allow it [14]. This point heavily depends on the implementation of the DLT and related services, e.g., the use of oracles and proof-of-location might enable a location verification operation [49].
3) Finally, the ‘directly and passively enact’ refers to the abilities that smart contracts generally offer. A smart contract passively enacts an operation because it does not “run in the background” and automatically activates itself when needed. However, an actor (that can generally be whoever) has to “wake up” the smart contract. A smart contract directly enacts an operation because everything needed for its execution is stored on the ledger and can be validated; if a clause’s condition is met on the ledger, a consequent action can be directly triggered.

2) NFTs and Immutable Storage: The second role of the SCM is to crystallize the data encoded using the Media Contractual Object. This is due to the native immutability feature that the DLTs’ ledger generally provides. Thus, once the SCM enters into action, i.e., it is deployed to the DLT, each piece of information related to the original contract can be validated against the stored SCM data, e.g., the address of a party or the fingerprint of a digital media.

Each Media Contractual Object is then stored in the SCM according to what was discussed in the previous subsection. In particular, each *IP Entity* and *Deontic Expression* is stored in a unique NFT, while the rest of the objects are stored in the SCM using an ad-hoc data structure, e.g., a hash map. To be noted is also the fact that the MPEG-21 *Contract* object preamble might include the narrative contract text version, too, in the form of an object or at clause level, making thus explicit the legal isomorphism.

We stress that NFTs are already used for encoding unique works resulting from human creativity and innovation, i.e., what intellectual property rights generally protect, that is the case of an *IP Entity* [50]. However, what is not generally trivial is the use of NFTs to encode information related to the ownership of certain rights, such as permissions, obligations, and prohibitions. Thanks to the *Deontic Expression* object representation, we can create referable rights and duties and save the association between this reference and the relevant party directly in the ledger in an immutable way through NFTs.

IV. Implementation

In the following, we present a possible set of technologies that enable the implementation of an SCM. In particular, our implementation combines the use of (i) a system for obtaining Media Contractual Objects from MPEG-21 MCO/CEL contracts; (ii) the Ethereum blockchain and smart contracts; (iii) the implementation of an ERC721 smart contract for working with Non Fungible Tokens; (iv) asymmetric cryptography key pairs, i.e., a public key and a private key, for working with Ethereum addresses; (v) the InterPlanetary File System (IPFS) for storing data.

A. MPEG-21 CEL/MCO Contracts Parser

Generally speaking, the input for generating an SCM would consist of the text of a narrative contract. However, this is not in the scope of our work, and thus, we only consider a fully validated MPEG-21 CEL/MCO contract as input. For CEL, it means an XML document, while for MCO can be an RDF/TURTLE encoded file. In both cases, we have implemented a system with two different parser components that generate a set of Media Contractual Objects as described in the ISO/IEC 21000 Part 23 standard [6]. These Media Contractual Objects are used to generate an SCM.

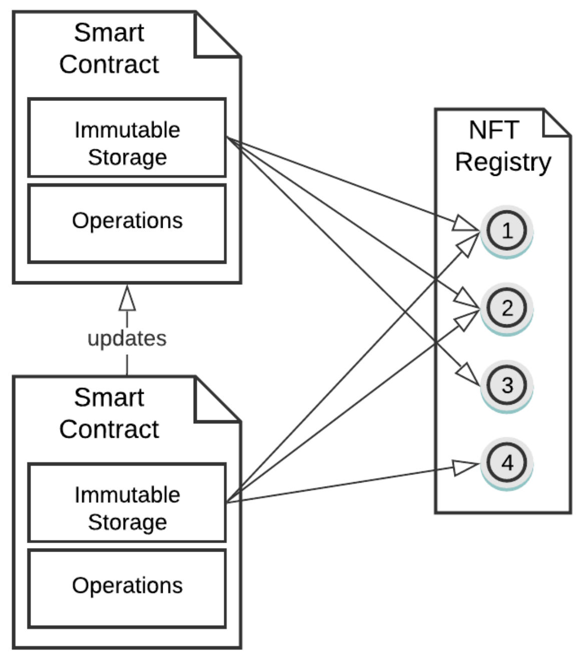

Fig. 3. Non Fungible Tokens schema

B. Ethereum and ERC721

Our implementation is based on the Ethereum blockchain [10]. The Ethereum protocol allows smart contracts to “talk” between each other directly on-chain and to operate cryptocurrencies flow, i.e., coins and tokens.

Moreover, our SCM implementation involves using an NFT smart contract, i.e., the ERC721 Non Fungible Token [18], that can be considered a registry for enumerating tokens. Such a registry contains a list of NFTs, that uniquely identifies an element described by Media Contractual Object, e.g., *IP Entity* or *Deontic Expression*. In particular, the registry maps the Ethereum address of the token owner to a specific token alphanumeric id. The token owner is a contracting party and depends on the particular token type. For instance, if the token represents an *IP Entity*, then the token owner is the *Organization* that holds the rights for that entity. When a new token is created, the ERC721 smart contracts bind the owner to the token id in the registry and then associate it with some metadata. In our implementation, the metadata cannot be further modified after the creation. Indeed, the metadata contains an immutable URI [51] to an off-chain document (this will be better defined later).

Using a single structure representation on-chain, i.e., the registry, enables interoperability at the smart contract level. For instance, each organization can have its own registry smart contract, or there can be single registries open for more kinds of media. At the same time, Ethereum smart contracts deployed for other purposes can reference NFTs in those registries to have a direct link on-chain (Figure 3).

On the other hand, using the ERC721 NFT enables interoperability between web platforms that already implement software interfaces with the Ethereum blockchain. Indeed, Ethereum is currently the most widely used technology to build decentralized applications, and many websites already support the ERC721 interface. This means that SCM can be used and referenced in several already functioning services with low effort. In web applications, users can operate through their software wallets, e.g., Metamask [52], and web applications will access NFT and check the user's claims.

C. Key Pair

The key pair is at the core of the identity authentication for contract parties. Using a digital signature as a binding cryptographic method enables any party to be represented by its public key since, by signing a digital document using the associated private key, anyone can verify that the signature is associated with the key pair's public key. In Ethereum, the public key is then transformed into an address. Thus *Party* objects can be mapped uniquely to Ethereum address, and the represented entities, e.g., persons or organizations, can publicly authenticate themself through the public key. Conversely, the private key is stored (and protected) in the local device used by the party since it is the only information needed to digitally sign on behalf of that entity.

D. Immutable URIs and InterPlanetary File System

In order to be immutable, data do not necessarily have to be stored directly on-chain. Indeed, this practice is costly in terms of time, space, and economics in most DLTs. Off-chain data storage, accompanied by on-chain storage of the result of the cryptographic hash function, is a trade-off that allows for information immutability and more cost-effective processing. It is also in line with data protection requirements and non-disclosure agreements. Thus, in our implementation, the data is kept private by default and not public. However, we maintain the verifiability property, even if the access to the contact information is not public. Indeed each Media Contractual Object is stored in documents that can be maintained private, while their hash, and thus their immutability verification, can be made public. The SCM (and NFTs metadata) only store the document's hash digest in our implementation. Thus, anyone with access to the document can verify if it has been altered, and, at the same time, the content is not shared publicly with anyone without access. This works both for permissioned and permissionless DLTs. For instance, a *Deontic Expression*, such as an *Obligation* for payment, can be stored off-chain, and then its hash is stored within an NFT metadata. This hash can be referred to as a hash pointer and generally as an immutable URI.

Several off-chain storage solutions can be used to store the documents, from traditional centrally-managed relational databases to distributed ones and Decentralized File Systems (DFS). In our implementation, we store contents on the InterPlanetary File System (IPFS). IPFS [35] is a DFS and a protocol thought for distributed environments focusing on data resilience. The IPFS P2P network stores and shares files and directories in the form of IPFS objects identified by a CID (Content IDentifier). This CID is the result of applying a hash function to a file, and it is used to retrieve the referenced IPFS object in the network. It consists of an immutable URI. Whenever an IPFS object is shared in the network, it will be identified by the CID retrieved from the object hash, for instance, a document with CID equal to *QmUA3Nn...* (truncated). If any other node in the network tries to share the same exact document, the CID will always be the same.

IPFS can be used together with the InterPlanetary Linked Data (IPLD) [53] to ensure that a logical object always maps to the same instance of a digital object. IPLD consists of standards and technologies leveraged to create universally addressable data structures, where the CID contains the hash and data decoding information. IPLD enables the linking of resources identified by hashes that can refer to diverse resources.

Thus, any document associated with a Media Contractual Object can be stored in the IPFS distributed environment and referenced through an immutable URI, also thanks to the IPLD specifications.

V. USE CASE AND PERFORMANCE EVALUATION

This section describes a use case where independent producers benefit from the proposed solution in a Video-On-Demand Services scenario.

A. Use Case

1) Preamble: It is not news that the spread of video-on-demand (VOD) services is bringing significant competition for legacy broadcasters. As a result and in compliance with the EU's Audiovisual Media Services Directive, some member states have adopted laws to accommodate the new audiovisual environment by forcing foreign VOD services to contribute to the production of domestic audiovisual content. Measures may take the form of financial contributions through film levy or obligations to invest directly in content [54]. For instance, in Italy, where the obligations for VOD services are more focused on the support of independent producers, the objective is to regulate foreign VOD services to invest in European and Italian works between 15.5% and 20% of their annual net sales revenue in Italy. The tax is higher if independent producers

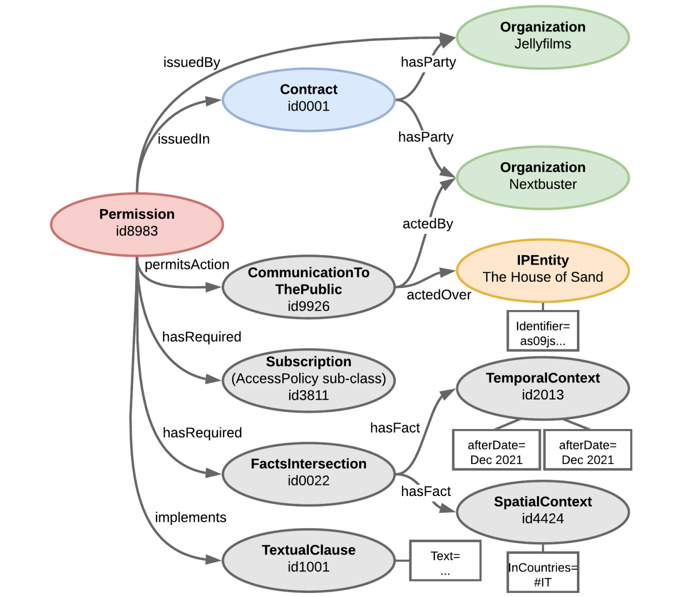

Fig. 4. Permission to communicate to the public using the MPEG-21 MCO. Ovals represent objects; arrows represent objects' properties and can link to other objects or data values in boxes. Objects in red represent *Deontic Expression*, in blue *Contract*, in green *Party*, in yellow *IP Entity*; the others, in gray, represent *Facts* or *Actions*.

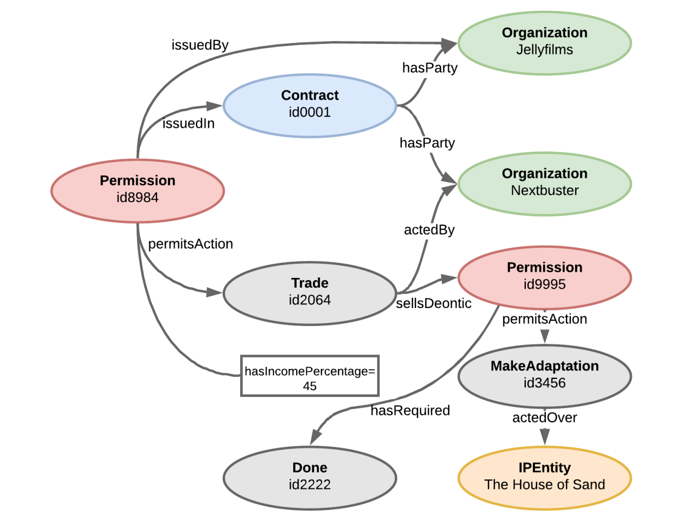

Fig. 5. Permission to trade the permission to make an adaptation of the work with an income percentage represented using the MPEG-21 MCO.

are denied a quota of secondary rights proportional to their financial contribution to a co-production.

2) Scenario: A VOD service called Nextbuster is willing to reach an agreement with an independent producer, Jellyfilms, for the co-production and distribution of a TV series entitled "The House of Sand". The contract stipulated includes an agreement for the acquisition, subject to the co-production agreement, by Nextbuster of 100% of the exploitation rights of the TV series by Jellyfilms by subscription as a form of payment in Italy for 5 years since December 2021. Furthermore, the agreement also includes a 45% share of the revenues from

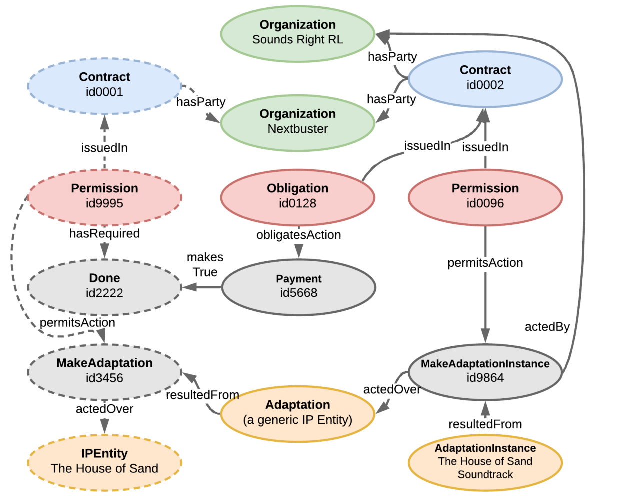

Fig. 6. Second contract including a reference to the first one (dotted lines) for the making of an adaptation instance represented using the MPEG-21 MCO.

the exploitation of secondary rights in favor of Jellyfilms, e.g., from the sale of the soundtrack and merchandise. The latter is achieved in a second contract that Nextbuster stipulates with the Sounds Right Record Label for the production of the "The House of Sand Soundtrack".

3) Analysis: The joint use of Semantic Web and DLTs technologies, in this case of rights management in the Video-on-Demand sector, has yielded the following advantages.

- Interoperability. The use of Semantic Web technologies automatically connects information represented using MPEG-21 MCO with other existing ontologies and knowledge bases, such as EBUCore [55], LinkedMDB [56], or the early BBC semantic web systems [57].
- Reasoning capabilities. Representing rights in RDF using the MPEG-21 MCO ontologies enables some logic inferences. An OWL reasoner can derive new facts from existing ones, producing contracts that are smart in a different way. For example, in the agreement described in the use case, the agreement between Nextbuster and Jellyfilms was limited to customers in Italy. A subsequent agreement limited to the region of Lazio could be automatically evaluated, for existing knowledge bases already know that Lazio is part of Italy.
- Transparency. The rights management can be visible to all the parties, including the weaker ones, such as the independent producer Jellyfilm in the presented use case.
- Technology neutrality. Using ISO/IEC standards and W3C recommendations and the ability to enforce the contracts without additional proprietary-based software such as CRM systems reduces complexity and makes the solution technologically neutral.
- Better compliance. The automated execution of contracts reduces the chances of paying penalties for contract violations, reducing risks to companies before legal suits.
- Data governance. These technologies support the implementation of data governance within Common European

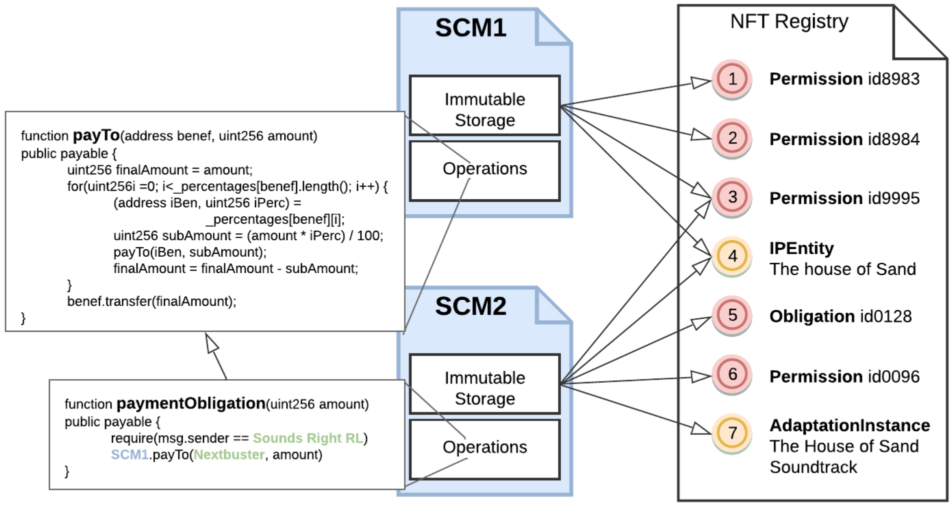

Fig. 7. Use Case Smart Contracts for Media

Data Spaces, in particular in the Media Data Space [58], which aims at enabling transparency and control in the exchange of media assets. Our solution directly addresses one of the technical challenges identified in the construction of these Data Spaces [59]: *Digital Sovereignty: Enforcing data usage rights*, but also indirectly in others such as *sharing by design*, *decentralisation*, *veracity* and *security*.

B. Implementation and Performance Evaluation

The first step to take in order to implement this scenario is the conversion from narrative contracts to MPEG-21 CEL/MCO contracts. Then, the latter is used to create a contract representation on-chain. In order to keep a direct link with our implementation shown in Section IV, we imagine that this private permissioned DLT is implemented following the Ethereum protocol.

1) MPEG-21 CEL/MCO Media Contract: This step is not automatic and requires a supervising human expert. In our case, we created media contracts using the MPEG-21 MCO based on the scenario's actors and agreements. An excerpt of the code, formatted in RDF/TURTLE, is given in the following:

```
<http://mpeg.org/id8983>
        a                         mvco:Permission ;
        rdf:about                 "id8983" ;
        rdfs:label                "Nextbuster can communicate to the
                                  public the 'The House of Sand'" ;
        mco-core:issuedIn         <http://mpeg.org/contract1> ;
        mvco:issuedBy             <http://mpeg.org/Jellyfilms> ;
        mco-core:permitsAction    <http://mpeg.org/id9926> ;
        mco-core:hasRequired      <http://mpeg.org/id3811> ;
        mco-core:hasRequired      <http://mpeg.org/id0022> ;
        mco-core:implements       <http://mpeg.org/id1001> .
<http://mpeg.org/id9926>
        a               mco-ipre:CommunicationToThePublic ;
        rdf:about       "id9926" ;
        rdfs:label      "Communication to the public of the 'The
                        House of Sand'" ;
        mvco:actedBy    <http://mpeg.org/Nextbuster> ;
        mvco:actedOver  <http://mpeg.org/TheHouseOfSand> .
<https://mpeg.org/id3811>
        a               mco-ipre:Subscription ;
        rdf:about       "id3811" ;
        rdfs:label      "The method of payment required by the
                        permission" .
<http://mpeg.org/id0022>
        a               mco-core:FactIntersection ;
        rdf:about       "id0022" ;
        rdfs:label      "The other constraints required by the
                        permission" ;
        mco-core:hasFact <http://mpeg.org/id2013> ;
        mco-core:hasFact <http://mpeg.org/id4424> .
<http://mpeg.org/id2013>
        a                   mco-ipre:TemporalContext ;
        rdf:about           "id2013" ;
        rdfs:label          "Permission allowed from December
                            2021 to December 2026" ;
        mco-ipre:afterDate  "20211201" ;
        mco-ipre:beforeDate "20261201" .
<http://mpeg.org/id4424>
        a                    mco-ipre:SpatialContext ;
        rdf:about            "id4424" ;
        rdfs:label           "The method of payment required
                             by the permission" ;
        mco-ipre:inCountries "#IT" .
<http://mpeg.org/id1001>
        a               mco-core:TextualClause ;
        rdf:about       "id1001" ;
        rdfs:label      "Text of the permission deontic expression" ;
        mco-core:Text   "Narrative contract clause..." .
```

Listing 1. References in the certificate document

In particular, this code refers to one of the permissions stated in the first contract between Nextbuster and Jellyfilms. We now describe the contracts with the aid of Figures 4, 5, and 6.

The *Contract* object with id=*id0001* represents the information extracted from the narrative contract between Nextbuster and Jellyfilms, i.e., the first contract. Such *Contract* object is linked with two *Permission* objects in our example but can be linked with many more objects. The first permission is a *Deontic Expression* object that permits the action *CommunicationToThePublic* by Nextbuster over the *IP Entity* object "The House of Sand", meaning that the VOD service can provide the streaming of such TV series, but under some restrictions. Such requirements are that (i) the access policy to the TV series must be by a subscription payment, (ii) that is valid from Dec 2021 to Dec 2026, and (iii) that is valid only in Italy. The *Permission* object also includes the narrative text clause. Furthermore, the second *Permission* object allows Nextbuster to trade the permission to make an adaptation of the TV series and sets an income percentage of 45% for Jellyfilms.

The second contract, i.e., the *Contract* object with id=*id0002*, represents the information extracted from the narrative contract between Nextbuster and Sounds Right RL. It includes a new *Permission* object, a new *Obligation*, and a reference to one *Permission* of the first contract. The new *Permission* object issued in the second contract, indeed, permits to make of an adaptation instance to Sounds Rights RL with the obligation to make a payment in favor of Nextbuster. This *Payment* action makes true a requirement of the *Permission* for making an adaptation of the first contract, i.e., it enacts the trade defined in the first contract and thus enacts also the income percentage clause. The result of this, i.e., the adaptation instance, is the production of “The House of Sand Soundtrack”.

2) Smart contract for Media Performance Evaluation: For this part of the evaluation, we implemented the Smart Contracts Templates and software components to convert the Media Contract into a Smart Contract for Media. The implemented smart contracts have been developed in Solidity and then stored as Open Source code on Zenodo [60]. The experimentation has been carried out in a local deployed blockchain following the Ethereum protocol.

First of all, it must be considered that this use case is not suited for a public DLT such as the Ethereum public one. In this case, the Smart Contracts for Media mainly help maintain adequate audit traceability due to compliance with the above-mentioned regulations. Moreover, we are not interested in interaction with the general public here. In this case, a private permissioned DLT managed directly by the stakeholders of the media industry, such as the Collection Societies, and possibly also by large companies, such as the main VOD services, is the most suitable choice. This scenario can be easily implemented in Hyperledger, Multichain, and other technologies generally used for permissioned networks [61].

We developed two Smart Contracts Templates that implement the possibility to pay Media Contract’s parties on the basis of the royalties agreements found in the original contract. Figure 7 shows a graphical representation of the two smart contracts. The final deployed smart contracts depend on the conversion process that the Smart Contract for Media specification describes. It starts from the MPEG-21 CEL/MCO Media Contract listed above. We also developed the smart contract for managing the NFT registry.

In Table I, we provide the execution cost in terms of gas usage [10] for the main operations. Gas is a unit that measures the amount of computational effort that takes to execute operations in Ethereum smart contracts. The operation for minting new NFTs for representing *Deontic Expression* or *Object* objects uses $82k$ gas units on average. For the setup of the data structures, gas usage is higher. In the first minting in the NFT registry the operation uses $112k$ gas units while the first minting for a new address (or party) is $97k$. This gas usage is in line with other state-of-the-art NFT registry implementations in Ethereum, e.g., the ones that are normally used for NFT art or profile pictures [62]. In our use case, the minted NFTs are 7 in total and thus the total amount of gas used would be around 574k. Two NFTs represent the *IPEntity* objects considered in the two contracts, i.e. “The House of Sand” and “The House of Sand Soundtrack”. The other 5 NFTs represent the *Permission* objects included in the two contracts and the *Obligation* of Sounds Right RL to pay the Nextbuster party. All these NFTs are referenced in the smart contracts implementing the two contracts through their id (as in Figure 7).

TABLE I
SMART CONTRACTS OPERATIONS COST IN TERMS OF GAS USAGE.

Smart Contract	Operation	Gas usage
NFT Registry	mintNFT()	82 765
NFT Registry	mintFirstTime()	112 765
NFT Registry	mintFIrstTimeForAParty()	97 681
SCM	newContract()	2 815 369
SCM	payTo1()	44 384
SCM	payTo2()	57 869
SCM	payTo3()	71 354
SCM	payTo5cascade()	91 688

The issuing of a new smart contract is an operation that uses a large amount of gas, but that is executed only once per contract. For instance, deploying the smart contract that represents the *Contract* object with id=id0001 uses around $2,800k$ gas units. This gas usage can be lowered by using smart contract templates that deploy optimization techniques, such as Factory or Proxy patterns. By using the EIP-1167 Minimal Proxy pattern [63], that, instead of deploying a new contract each time, creates proxies that invoke methods of an already deployed contract, we managed to halve the gas usage. This smart contract implements the method *payTo()* (see Figure 7), which is used to subdivide the payments to an actor on the basis of the royalties agreement. In our use case, the payment of Sounds Right RL in favor of Nextbuster is split with the Jellyfilms party with an income percentage of 45%. The execution of this method uses 44k gas units on average. When the beneficiaries of a split are more than one the gas usage increases, e.g., $57k$ for 2 beneficiaries and $71k$ for 3. In general, the increase in gas usage follows linearly the number of parties involved in the payment in a formula such as $f = 30k + (14k * beneficiaries\_n)$.

VI. CONCLUSIONS

This paper has demonstrated how to use the Smarts Contracts for Media specification, a proposed bridge to transform rights information and contracts from its MPEG-21 CEL/MCO form to smart contracts executable in blockchains. Details on the implementation have been given, and a specific use case for Video-on-Demand services has been examined in more detail.

The MPEG-21 SCM technical description is in the ISO/IEC standardization path and will become an official specification. However, several resources must be available for the widespread adoption of the standard. First of all, examples of use as the one presented in this paper. Nevertheless, technological developments, new standard-based applications, public contract templates that can be quickly replicated and personalized, and cookbook recipes to leverage the OWL reasoning capabilities or mappings to other ontologies and knowledge graphs. The development of this ecosystem of resources around MPEG-21 and SCM remains an open challenge.

Acknowledgments

This work has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie International Training Network European Joint Doctorate grant agreement No 814177 Law, Science and Technology Joint Doctorate - Rights of Internet of Everything. The authors of this work participated to the drafting of the ISO/IEC IS 21000-23 Part 23 Smart Contracts for Media standard specification.

References

[1] I. Burnett, R. Van de Walle, K. Hill, J. Bormans, and F. Pereira, "Mpeg-21: goals and achievements," *IEEE MultiMedia*, vol. 10, no. 4, pp. 60–70, 2003.

[2] X. Wang, T. De Martini, B. Wragg, M. Paramasivam, and C. Barlas, "The mpeg-21 rights expression language and rights data dictionary," *IEEE Transactions on Multimedia*, vol. 7, no. 3, pp. 408–417, 2005.

[3] V. Rodriguez-Doncel and J. Delgado, "A media value chain ontology for mpeg-21," *IEEE MultiMedia*, vol. 16, no. 4, pp. 44–51, 2009.

[4] E. Rodríguez, J. Delgado, L. Boch, and V. Rodríguez-Doncel, "Media contract formalization using a standardized contract expression language," *IEEE MultiMedia*, vol. 22, no. 2, pp. 64–74, 2015.

[5] V. Rodriguez-Doncel, J. Delgado, S. Llorente, E. Rodriguez, and L. Boch, "Overview of the mpeg-21 media contract ontology," *Semantic Web*, vol. 7, no. 3, pp. 311–332, 2016.

[6] "ISO/IEC IS 21000-23 Information technology - Multimedia framework (MPEG-21) - Part 23: Smart Contracts for Media," International Organization for Standardization, Geneva, CH, Standard, 2022.

[7] P. Kudumakis, T. Wilmering, M. Sandler, V. Rodríguez-Doncel, L. Boch, and J. Delgado, "The challenge: From mpeg intellectual property rights ontologies to smart contracts and blockchains [standards in a nutshell]," *IEEE Signal Processing Magazine*, vol. 37, no. 2, pp. 89–95, 2020.

[8] A. Savelyev, "Copyright in the blockchain era: Promises and challenges," *Computer law & security review*, vol. 34, no. 3, pp. 550–561, 2018.

[9] M. Becker and B. Bodó, "Trust in blockchain-based systems," *Internet Policy Review*, vol. 10, no. 2, 2021.

[10] V. Buterin *et al.*, "Ethereum white paper," 2013. [Online]. Available: https://github.com/ethereum/wiki/wiki/White-Paper

[11] M. Bartoletti and L. Pompianu, "An empirical analysis of smart contracts: platforms, applications, and design patterns," in *International conference on financial cryptography and data security*. Springer, 2017, pp. 494–509.

[12] M. Zichichi, S. Ferretti, and G. D'Angelo, "A framework based on distributed ledger technologies for data management and services in intelligent transportation systems," *IEEE Access*, pp. 100 384–100 402, 2020.

[13] M. Nassar, K. Salah, M. H. ur Rehman, and D. Svetinovic, "Blockchain for explainable and trustworthy artificial intelligence," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 10, no. 1, p. e1340, 2020.

[14] G. Finocchiaro and C. Bomprezzi, "Legal analysis of the use of blockchain technology for the formation of smart legal contracts," *Media Laws Rivista di Diritto dei Media*, 2020.

[15] L. Cervone, M. Palmirani, and F. Vitali, "The intelligible contract." in *HICSS*, 2020, pp. 1–10.

[16] P. De Filippi, C. Wray, and G. Sileno, "Smart contracts," *Internet Policy Review*, vol. 10, no. 2, 2021.

[17] F. Vogelsteller and V. Buterin, "Erc-20 token standard," *Ethereum Foundation (Stiftung Ethereum), Zug, Switzerland*, 2015.

[18] W. Entriken, D. Shirley, J. Evans, and N. Sachs, "Erc-721 token standard," *https://eips.ethereum.org/EIPS/eip-721*, 2018.

[19] P. Caglayan and Z. Ozkan, "NFTs and copyright: challenges and opportunities," *Journal of Intellectual Property Law& Practice*, vol. 16, no. 10, 06 2021.

[20] T. Berners-Lee, J. Hendler, O. Lassila *et al.*, "The semantic web," *Scientific american*, vol. 284, no. 5, pp. 28–37, 2001.

[21] T. Pellegrini, A. Schönhofer, S. Kirrane, S. Steyskal, A. Fensel, O. Panasiuk, V. Mireles-Chavez, T. Thurner, M. Dörfler, and A. Polleres, "A genealogy and classification of rights expression languages-preliminary results," in *Data Protection/LegalTech-Proceedings of the 21st International Legal Informatics Symposium IRIS*, 2018, pp. 243–250.

[22] R. Ianella, "Open digital rights language (odrl)," *Open Content Licensing: Cultivating the Creative Commons*, 2007.

[23] S. Kirrane, S. Villata, and M. d'Aquin, "Privacy, security and policies: A review of problems and solutions with semantic web technologies," *Semantic Web*, vol. 9, no. 2, pp. 153–161, 2018.

[24] E. Rodríguez, J. Delgado, L. Boch, and V. Rodríguez-Doncel, "Media contract formalization using a standardized contract expression language," *IEEE multimedia*, vol. 22, no. 2, pp. 64–74, 2014.

[25] D. Chalmers, R. Matthews, and A. Hyslop, "Blockchain as an external enabler of new venture ideas: Digital entrepreneurs and the disintermediation of the global music industry," *Journal of Business Research*, vol. 125, pp. 577–591, 2021.

[26] ConSensys, "Ujo music," *https://blog.ujomusic.com/*, 2018, online. Accessed 2022.

[27] Vext Inc, "Vezt," *https://medium.com/@vezt*, 2019, online. Accessed 2022.

[28] Angelo Recca and Iulian Anghelin, "Bitsong whitepaper," *https://bitsong.io/*, 2021, online. Accessed 2022.

[29] J. Chen and S. Micali, "Algorand: A secure and efficient distributed ledger," *Theoretical Computer Science*, vol. 777, pp. 155–183, 2019.

[30] Napster, "Napster 3.0 litepaper," *https://www.napster.com/us/wp-content/uploads/sites/5/2022/06/Napster-Litepaper-V1.pdf*, 2022, online. Accessed 2022.

[31] RChain Cooperative, "Rchain whitepaper," *https://rchain.coop/whitepaper.html*, 2021, online. Accessed 2022.

[32] Z. Ma, M. Jiang, H. Gao, and Z. Wang, "Blockchain for digital rights management," *Future Generation Computer Systems*, vol. 89, pp. 746–764, 2018.

[33] L. Liu, W. Zhang, and C. Han, "A survey for the application of blockchain technology in the media," *Peer-to-Peer Networking and Applications*, vol. 14, no. 5, pp. 3143–3165, 2021.

[34] Z. Meng, T. Morizumi, S. Miyata, and H. Kinoshita, "Design scheme of copyright management system based on digital watermarking and blockchain," in *2018 IEEE 42nd Annual Computer Software and Applications Conference (COMPSAC)*, vol. 2. IEEE, 2018, pp. 359–364.

[35] J. Benet, "Ipfs-content addressed, versioned, p2p file system," *arXiv preprint arXiv:1407.3561*, 2014.

[36] R. García, A. Cediel, M. Teixidó, and R. Gil, "Copyrightly: Blockchain and semantic web for decentralised copyright management," in *Economics of Grids, Clouds, Systems, and Services*, K. Tserpes, J. Altmann, J. Á. Bañares, O. Agmon Ben-Yehuda, K. Djemame, V. Stankovski, and B. Tuffin, Eds. Cham: Springer International Publishing, 2021, pp. 199–206.

[37] T. Gaber, A. Ahmed, and A. Mostafa, "Privdrm: A privacy-preserving secure digital right management system," in *Proceedings of the Evaluation and Assessment in Software Engineering*, 2020, pp. 481–486.

[38] A. Vishwa and F. K. Hussain, "A blockchain based approach for multimedia privacy protection and provenance," in *2018 IEEE Symposium Series on Computational Intelligence (SSCI)*. IEEE, 2018, pp. 1941–1945.

[39] B. Esteves, H. J. Pandit, and V. Rodríguez-Doncel, "Odrl profile for expressing consent through granular access control policies in solid," in *2021 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW)*. IEEE, 2021, pp. 298–306.

[40] M. Ramachandran, N. Chowdhury, A. Third, Z. Jan, C. Valentine, and J. Domingue, "A framework for handling internet of things data with confidentiality and blockchain support," 2020.

[41] T. Cai, Z. Yang, W. Chen, Z. Zheng, and Y. Yu, "A blockchain-assisted trust access authentication system for solid," *IEEE Access*, vol. 8, pp. 71 605–71 616, 2020.

[42] A. Banerjee and K. P. Joshi, "Link before you share: Managing privacy policies through blockchain," in *2017 IEEE International Conference on Big Data (Big Data)*. IEEE, 2017, pp. 4438–4447.

[43] M. Davari and E. Bertino, "Access control model extensions to support data privacy protection based on gdpr," in *2019 IEEE International Conference on Big Data (Big Data)*. IEEE, 2019, pp. 4017–4024.

[44] D. Hawig, C. Zhou, S. Fuhrhop, A. S. Fialho, and N. Ramachandran, "Designing a distributed ledger technology system for interoperable and general data protection regulation–compliant health data exchange: a use case in blood glucose data," *Journal of medical Internet research*, vol. 21, no. 6, p. e13665, 2019.

[45] M. Koscina, D. Manset, C. Negri, and O. Perez, "Enabling trust in healthcare data exchange with a federated blockchain-based architecture," in *IEEE/WIC/ACM International Conference on Web Intelligence-Companion Volume*, 2019, pp. 231–237.

[46] T. Lebo, S. Sahoo, D. McGuinness, K. Belhajjame, J. Cheney, D. Corsar, D. Garijo, S. Soiland-Reyes, S. Zednik, and J. Zhao, "Prov-o: The prov ontology," *W3C recommendation*, vol. 30, 2013.

[47] K. Gaaloul, A. Schaad, U. Flegel, and F. Charoy, "A secure task delegation model for workflows," in *2008 Second International Conference on Emerging Security Information, Systems and Technologies*. IEEE, 2008, pp. 10–15.

[48] ISO/IEC JTC 1/SC 29, "Iso/iec is 21000-23 information technology — multimedia framework (mpeg-21) — part 23: Smart contracts for media," International Organization for Standardization, Tech. Rep., 2022.

[49] M. Amoretti, G. Brambilla, F. Medioli, and F. Zanichelli, "Blockchain-based proof of location," in *2018 IEEE International Conference on Software Quality, Reliability and Security Companion (QRS-C)*. IEEE, 2018, pp. 146–153.

[50] S. M. H. Bamakan, N. Nezhadsistani, O. Bodaghi, and Q. Qu, "A decentralized framework for patents and intellectual property as NFT in blockchain networks," oct 2021.

[51] T. Berners-Lee, R. Fielding, and L. Masinter, "Uniform resource identifier (uri): Generic syntax," Tech. Rep., 2004.

[52] Metamask, "Metamask website," *https://metamask.io/*, 2021, online. Accessed 2021.

[53] IPLD Team, "Interplanetary linked data (ipld)," *https://specs.ipld.io/*, 2016.

[54] I. Kostovska, T. Raats, and K. Donders, "The rise of the 'netflix tax'and what it means for sustaining european audiovisual markets," *Innovation: The European Journal of Social Science Research*, vol. 33, no. 4, pp. 423–441, 2020.

[55] "EBU CORE Metadata Set (EBUCore) Specification v. 1.10," European Broadcasters Union, Geneva, CH, Standard, 2020.

[56] O. Hassanzadeh and M. P. Consens, "Linked Movie Data Base," in *Proceedings of the WWW2009 workshop on Linked Data on the Web (LDOW2009)*, April 2009.

[57] Y. Raimond, T. Scott, S. Oliver, P. Sinclair, and M. Smethurst, "Use of semantic web technologies on the bbc web sites," in *Linking Enterprise Data*. Springer, 2010, pp. 263–283.

[58] "Digital european platform of quality content providers," CNECT/2021/OP/0002, Report, 2022.

[59] S. Scerri, T. Tuikka, I. L. de Vallejo, and E. Curry, "Common european data spaces: Challenges and opportunities," *Data Spaces: Design, Deployment and Future Directions*, pp. 337–357, 2022.

[60] M. Zichichi, "miker83z/MPEG-21-Part-23-DLT-Manager: mpeg-21-part-23-dlt-manager-v1.0.0," Nov. 2022. [Online]. Available: https://doi.org/10.5281/zenodo.7327351

[61] M. Dabbagh, K.-K. R. Choo, A. Beheshti, M. Tahir, and N. S. Safa, "A survey of empirical performance evaluation of permissioned blockchain platforms: Challenges and opportunities," *computers & security*, vol. 100, p. 102078, 2021.

[62] S. Casale-Brunet, M. Zichichi, L. Hutchinson, M. Mattavelli, and S. Ferretti, "The impact of nft profile pictures within social network communities," in *Proc. of the Conference on Information Technology for Social Good (GoodIT)*. ACM, 2022, pp. 1–11.

[63] J. M. Peter Murray, Nate Welch, "Eip-1167: Minimal proxy contract," https://eips.ethereum.org/EIPS/eip-1167, 2018.