

Special Paper

방송공학회논문지 제29권 제7호, 2024년 12월 (JBE Vol. 29, No. 7, December 2024)

<https://doi.org/10.5909/JBE.2024.29.7.1115>

ISSN 2287-9137 (Online) ISSN 1226-7953 (Print)

NFT Description Method for Managing Ownership of Media Thing Content

Jonghoon Chun^{a)} and Sang-Kyun Kim^{a)‡}

Abstract

The Internet of Media Things aims to provide intelligent media services to users by connecting objects and exchanging media data. Media content generated from media things such as cameras or object recognition devices is continuously used or analyzed by other media things to provide media services to users. At this time, the ownership of media content can be transferred to other media thing owners by converting it into NFT. ISO/IEC 23093-3 and 23093-6 are standardizing the API and metadata description structure required for NFT conversion of media content generated from media things. This paper describes a use case for NFTization of media data in terms of media sensors and analysers. It shows how external blockchain systems and media things can be connected to create and transmit NFT media data. It introduces the contents of the media NFT standard (ISO/IEC 23093-3) required for implementing the use case.

Keyword : Internet of Media Things, media sensors, media analysers, NFT, NFT description and API

1. Introduction

Internet of Media Things (IoMT) aims to provide intelligent media services to users by connecting (media) things to each other and exchanging media data. Media things (MThings) are composed of a media sensor (MSensor) that can acquire media and a media actuator (MActuator) that reproduces media. MThing also includes a media analyser (MAnalyser) that analyses the obtained

media and extracts meaningful information and a media storage (MStorage) that stores acquired media content or analysed information^[1].

The ISO/IEC 23093 series provides an architecture and specifies APIs and compressed representation of data flowing between media things (MThings). The APIs for the MThings facilitate discovering other MThings in the network, connecting and efficiently exchanging data between MThings. The APIs also support transaction tokens to access valuable functionalities, resources, and data from MThings.

MThing-related information comprises characteristics and discovery data, mission descriptions from system designers and end-users, raw and processed sensed data and actuation information. The ISO/IEC 23093 series specifies

a) Department of Convergence Software, Myongji University

‡ Corresponding Author : Sang-Kyun Kim

E-mail: goldmunt@gmail.com

Tel: +82-2-300-0637

ORCID: <https://orcid.org/0000-0002-2359-8709>

· Manuscript October 14, 2024; Revised November 21, 2024; Accepted November 25, 2024.

input and output data formats for media sensors, actuators, storages, and analysers. In addition, media analysers can process sensed data from media sensors to produce analysed data, which can be cascaded to other media analysers to extract semantic information. Multiple MThings can be gathered and operated autonomously using mission descriptions given by system designers and end-users^[2-5].

The ISO/IEC 23093-3 and 6 contain the tools to describe data exchanged between media things (e.g., media sensors, media actuators, media analysers, and media storages) and their APIs. It addresses the normative aspects of the data and APIs for media things and also illustrates non-normative examples.

Media content generated from media sensors and media analysers has its value. Video or audio acquired from media sensors may have special value depending on the situation, and the results of analysing such content may also have special value. This value can be defined as NFT and transferred between media things. Media things can convert media generated from other media things into NFTs and purchase them. Media things that are requested to generate NFT media content have the function of setting the price of the NFT and recording the generated NFT on the blockchain. In order to generate and transfer NFTs, standardised NFT data and APIs must be available. This paper introduces the contents of ISO/IEC 23093-3 to request the

NFT price and generate the NFT media content.

The composition of the paper is as follows. Section 2 describes the use case of NFT generation, including the IoMT architecture. Section 3 introduces the standard NFT data format and APIs defined in the standard, followed by the conclusion in Section 4.

II. NFT use case in IoMT

1. IoMT architecture and scope

The IoMT media data formats and API standard (i.e., ISO/IEC 23093-3) specifies the syntax and semantics of description schemes to represent data exchanged by media things (e.g., media sensors, media actuators, and media storages). Moreover, it specifies the APIs to exchange these data between media things.

The IoMT media data formats and API for distributed AI processing standard specifies the syntax and semantics of description schemes to represent data exchanged by media analysers for distributed AI processing. Moreover, it specifies the APIs to exchange these data between media things. The following interfaces are under the scope of these documents:

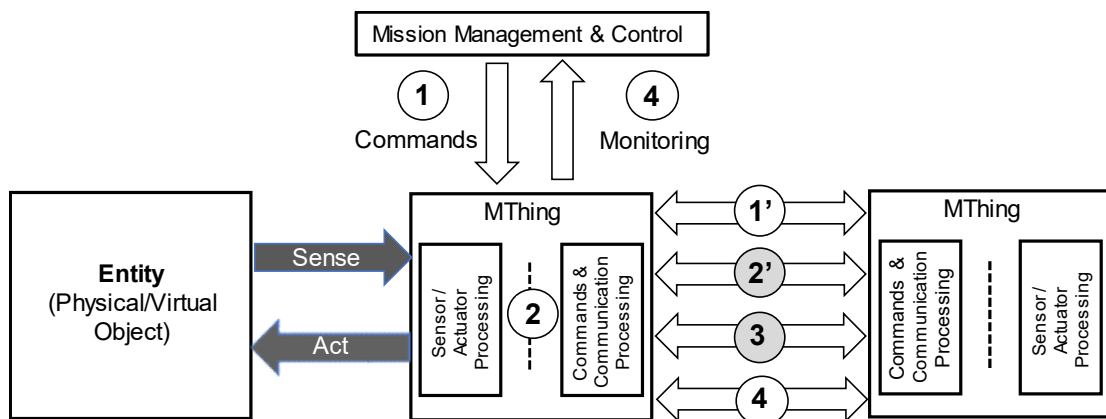


Fig. 1. Architectural view of the Internet of Media Things^[2]

- APIs for accessing media sensors, analysers, actuators, storage, managers, controllers, and aggregators;
- structured data formats (XML) representing media thing's base data types, including sensors, analysers, actuators, storage, managers, aggregators, controllers, and their elements (described in section 3.1.1);
- structured data formats (XML) representing the media sensor and analyser output data referred to in Figure 1-2'; and,
- structured data formats (XML) representing media actuator's commands referred to in Figure 1-3.

2. Use case of NFT from media sensors and analysers

The content generated from MThings can be a unique resource defined as a Non-Fungible Token (NFT). The video or audio clip generated from MSensors can be NFTs,

while the analysed data from MAnalysers can also be NFTs. Figure 2 shows how the MObjectDetector requests an NFT from the MCamera.

The process description is as follows:

- ① A MObjectDetector asks an MCamera the cost of creating NFT content (getNFTVideoURL_CostPerMinute()), checks where to pay the cost (getWalletAddress() defined in ISO/IEC 23093-2, SubClause 4.2), pays the amount (sendTokens() defined in ISO/IEC 23093-2, SubClause 4.2), checks transaction completion (checkTransactionCompletion() defined in ISO/IEC 23093-2, SubClause 4.2), and request the NFT address by sending tid through getNFTVideoURL(tid, address). Here, the address parameter indicates the recipient wallet address of the MObjectDetector (Figure 2-1).
- ② The MCamera mints the NFT with NFT metadata, including the video URL. The NFT is transferred to the

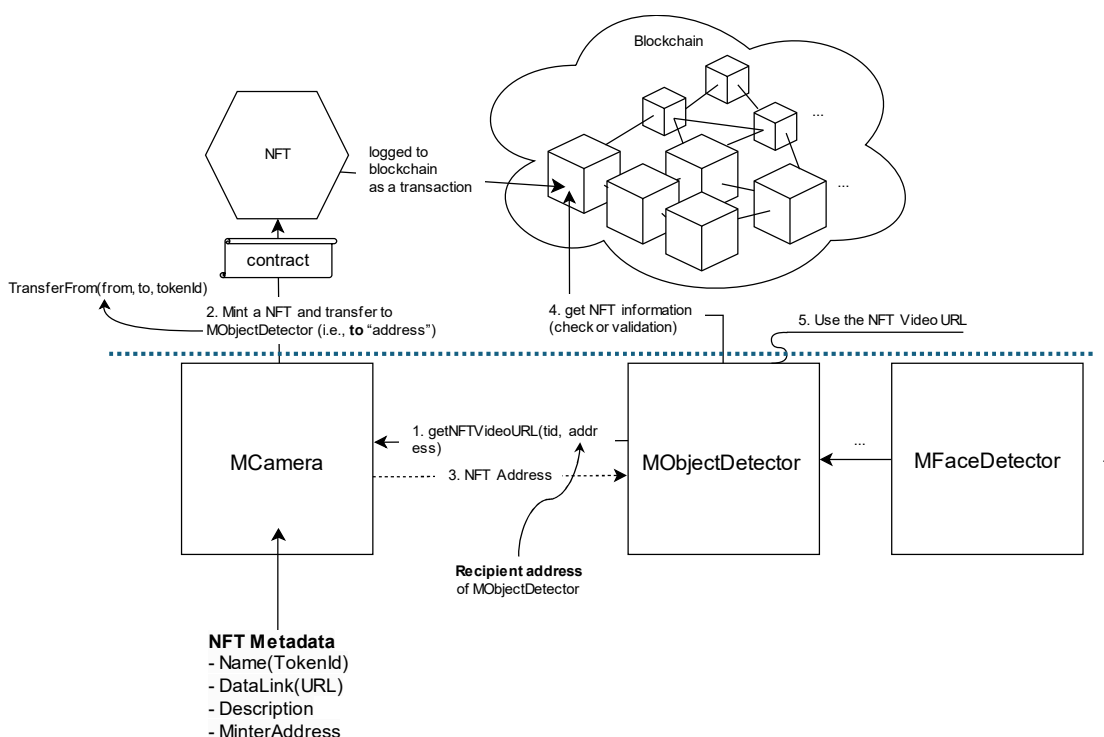


Fig. 2. An example of the NFT generation process of MSensor

MObjectDetector’s address through the smart contract’s automated purchase function (assuming it is implemented) (Figure 2-2). (Outside of IoMT)

- ③ The MCamera returns the NFT address to the MObjectDetector (Figure 2-3).
- ④ The MObjectDetector checks whether the NFT has been transferred properly through the blockchain using the returned NFT address (Figure 2-4). (Outside of IoMT)
- ⑤ After verification, the MObjectDetector can legally access the video URL in the NFT (Figure 2-5). (Outside of IoMT)

The following example demonstrates a similar case for generating an NFT of an MAnalyser. Figure 3 shows how the MFaceDetector requests an NFT from the MObjectDetector.

The process description is as follows:

- ① A MFaceDetector asks a MObjectDetector the cost of creating NFT content (getNFTObjectDetectionResult_Cost()), checks where to pay the cost (getWalletAddress()) defined in ISO/IEC 23093-2, SubClause 4.2), pays the amount (sendTokens()) defined in ISO/IEC 23093-2, SubClause 4.2), checks transaction completion (checkTransactionCompletion() defined in ISO/IEC 23093-2, SubClause 4.2), and request the NFT address by sending the transaction ID (i.e., tid) through getNFTGetObjectDetectionResult(tid, address). Here, the address parameter indicates the recipient wallet address of the MFaceDetector (Figure 3-1).
- ② The MObjectDetector mints the NFT with NFT metadata, including the object detection result URL. The NFT is transferred to the MFaceDetector’s address

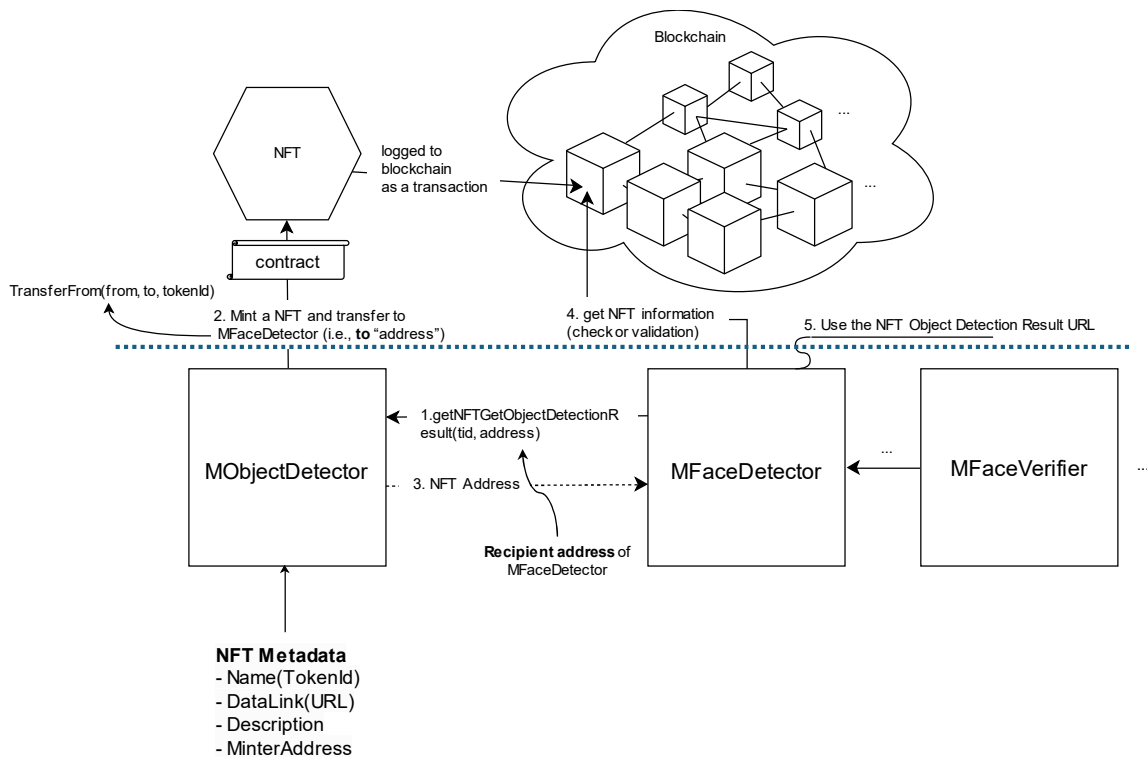


Fig. 3. An example of the NFT generation process of MAnalyser

through the smart contract's automated purchase function (assuming it is implemented) (Figure 3-2). (Outside of IoMT)

- ③ The MObjectDetector returns the NFT address to the MFaceDetector (Figure 3-3).
- ④ The MFaceDetector checks whether the NFT has been transferred properly through the blockchain using the returned NFT address (Figure 3-4). (Outside of IoMT)
- ⑤ After verification, the MFaceDetector can legally access the object detection result URL in the NFT (Figure 3-5). (Outside of IoMT)

III. NFT Metadata and APIs

1. MThing base data type and elements

This subsection specifies base data types and elements of MThing data formats. An MThing can have a data ID, a device ID, an IP address, and a URL as base

attributes. Besides, data can include the supported cryptocurrencies and legal tenders to use (or borrow) functionalities, resources, NFT data, and information on MThings. The NFT data of the MThing base data type includes the NFTName, MinterAddress, Datalink, and Description^[4].

1.1 Syntax of NFT data type in MThing base data type

The NFT data of the MThing base data type includes the NFTName, MinterAddress, Datalink, and Description (Figure 4). These data are used for NFT generation in the blockchain.

1.2 Syntax of the root element

The “NFTData” element can be generated optionally In the root element “MThingInfo” (Figure 5). In other words, the “NFTData” can be described as one of the MThing information generated by a media thing.

```

<attributeGroup name="MThingBaseAttributes">
  <attribute name="id" type="string" use="optional"/>
  <attribute name="idRef" type="string" use="required"/>
  <attribute name="ipAddress" type="string" use="optional"/>
  <attribute name="url" type="anyURI" use="optional"/>
</attributeGroup>

<complexType name="NFTDataType">
  <sequence>
    <element name="NFTName" type="string"/>
    <element name="MinterAddress" type="string"/>
    <element name="DataLink" type="string"/>
    <element name="Description" type="string" minOccurs=0/>
  </sequence>
</complexType>

```

Fig. 4. Syntax of NFT data type required for generating NFT media data

```

<element name="MThingInfo">
  <complexType>
    <sequence>
      <element name="TimeStamp" type="dateType" minOccurs="0"/>
      <element name="supportedTokenList" type="mtdl:SupportedTokenListType"
minOccurs="0" maxOccurs="1"/>
      <element name="NFTData" type="mtdl:NFTDataType" minOccurs="0" maxOccurs="1"/>
      <choice>
        <element ref="mtdl:MSensor"/>
        <element ref="mtdl:MActuator"/>
        <element ref="mtdl:MAnalyser"/>
        <element ref="mtdl:MStorage"/>
        <element ref="mtdl:MManager"/>
        <element ref="mtdl:MAggregator"/>
        <element ref="mtdl:MController"/>
      </choice>
    </sequence>
    <attributeGroup ref="mtdl:MThingBaseAttributes"/>
  </complexType>
</element>

```

Fig. 5. Syntax of the loMT root element containing the NFT data

1.3 Binary syntax of NFT data type

The NFT data type can be described in binary format as defined in the following binary syntax (Table 1). This binary syntax makes the NFT description data more concise and compact.

Table 1. The binary syntax of NFT data

NFTDataType {	Number of Bits	Mnemonic
descriptionFlag	1	bslbf
NFTName	See ISO 10646	UTF-8
MinterAddress	See ISO 10646	UTF-8
DataLink	See ISO 10646	UTF-8
if(descriptionFlag) {		
Description	32	fsbf
}		
}		

1.4 Semantics of NFT data type

The detailed definition of each element and attribute of the NFT data type is shown in the following table 2.

Table 2. The semantics of NFT data type and its element and attribute

Name	Definition
NFTDataType	Tool for describing an NFT Metadata.
descriptionFlag	This field, only present in the binary representation, indicates which attribute shall be used. If it is set to "0," the Description element follows. If it is set to "1," the Description element follows.
NFTName	It describes the name of NFT.
MinterAddress	It describes the crypto-currency wallet address of the NFT issuer.
DataLink	It describes the URL where the original content is stored.
Description	It describes the human-readable Description of the original content.

According to this semantics, each element and attribute can be transformed into the attributes forming the NFT.

1.5 Example

Figure 6 shows an XML instance following the syntax described in Figures 5 and 6. The NFT media data named “mcam1” was minted from the address “0x123456789abcdef123456789abcdef123456789” (i.e., camera’s account). It contains a simple data link, “http://iomtvideo.com/1.mp4”, enabling access to the video content.

As shown in Figure 2, any MAnalyser can ask an MCamera the cost of creating NFT content using the standard API `getNFTVideoURL_CostPerMinute()`. It will return the amount of payment per minute to use `getNFTVideoURL()`. The standard API `getNFTVideoURL()` returns an NFT address, including a video URL, to the MAnalyser (i.e., NFT requester).

IV. Conclusion

This paper briefly introduced the use cases and process of using NFT data for media content generated by media sensors and media analysers. The NFT data type and its example APIs in ISO/IEC 23093-3 (i.e., Internet of media things - Media data formats and API) standard currently being standardised are presented to implement the use cases as well. In the future, we plan to develop an API and data format that can exchange the NFT information of each media thing (e.g., MAnalyser).

References

- [1] Jonghoon Chun, Sang-Kyun Kim, “Mission Description Structure for Autonomous Collaboration of Media Things,” JBE Vol. 27, No. 7, December 2022.
- [2] “Text of ISO/IEC DIS 23093-1 IoMT architecture,” MDS24017_

```

<MThingInfo id="s1" idRef="s1-101-22">
  <NFTData>
    <NFTName>mcam1</NFTName>
    <MinterAddress>0x123456789abcdef123456789abcdef123456789</MinterAddress>
    <DataLink>http://iomtvideo.com/1.mp4</DataLink>
    <Description>"A three-minute video of a square"</Description>
  </NFTData>
</MThingInfo>
    
```

Fig. 6. An XML instance describing an NFT video from a camera. 3.2 Example of APIs related to NFT

Table 3. APIs for requesting NFT video data from the camera

string	<code>getNFTVideoURL(string tid, string walletAddress)</code>
	This function returns a NFT address including a video URL. The tid is the transaction ID of payment for using this function. The walletAddress is the wallet address of the NFT requester.
float	<code>getNFTVideoURL_CostPerMinute(string tokenName)</code>
	This function returns the amount of payment per minute to use <code>getNFTVideoURL()</code> . The tokenName is described in a string (e.g., term ID or binary representation) from TokenTypeCS specified in A.4. If the requested token is not supported, it returns -1. The tokenName denotes only the “cryptocurrency”. Ex) <code>getNFTVideoURL_CostPerMinute("ETH")</code> or <code>getNFTVideoURL_CostPerMinute("00000002")</code>

WG07_N00921, 146th MPEG, 2024.

[3] "Text of ISO/IEC DIS 23093-2 IoMT discovery and communication API," MDS24019_WG07_N00923, 146th MPEG, 2024.

[4] "Text of ISO/IEC DIS 23093-3 Ed 3 IoMT Media data formats and

APIs," MDS24024_WG07_N00926, 146th MPEG, 2024.

[5] "Text of ISO/IEC DIS 23093-6 IoMT Media data formats and API for distributed AI processing," MDS24026_WG07_N00928, 146th MPEG, 2024.

Introduction Authors



Jonghoon Chun

- 1992. : Computer Science, Univ. of Denver, B.S.(1986), Northwestern Univ., M.S.(1988), Ph.D.(1992)
- 1992. 09. ~ 1995. 06. : Assistant Professor of Computing Science, Univ. of Central Oklahoma
- 1995. 09. ~ 2016. 02. : Professor of Computer Engineering, Myongji University
- 2016. 03. ~ Current : Professor of Data Technology, Myongji University
- 2011. 04. ~ Current : President and CEO, Prompt Technology Co., Ltd.
- ORCID : <https://orcid.org/0000-0003-3396-4239>
- Research interests : Database, Dataset Retrieval, Intelligence Software



Sang-Kyun Kim

- 1997. : Computer Science, Univ. of Iowa, B.S.(1991), M.S.(1995), PhD(1997)
- 1997. 03. ~ 2007. 02. : Professional Researcher, Multimedia Lab. of Samsung Advanced Institute of Technology
- 2007. 03. ~ 2016. 02. : Professor of Computer Engineering, Myongji University
- 2016. 03. ~ Current : Professor of Data Technology, Myongji University
- ORCID : <https://orcid.org/0000-0002-2359-8709>
- Research interests : Digital Content(image, video and audio) analysis and management, 4D media, Blockchain, VR, Internet of Things and multimedia standardization