One Solution For Multimedia Subscription Using Blockchain

Igor Srdić, Đorđe Glišić, Marija Jovanović

***Abstract*—With the expansion of blockchain technologies in different areas from health care to voting systems and emerging demand on video content delivery platforms, it is interesting to investigate possibilities to combine those two into a new system that will help digital content availability. We present and discuss relevant work in the industry. Paper proposes one solution for subscription rights management using blockchain technologies. Proof of concept is done using the Ethereum blockchain ecosystem for a video-on-demand service. A similar approach could be used in other fields of DTV services, like cable TV subscription services.**

***Index Terms*—subscription management, blockchain, smart contracts, Ethereum, DTV content subscription, Android TV, solidity, web3j.**

I. Introduction

Blockchain is a growing list of records linked to the use of the cryptocurrency method. After the sudden expansion of Bitcoin [1], it became clear that the potential of blockchain is a lot greater than its use for money. This technology's decentralized system is fertile ground for many ideas and provides a new approach and opportunities [2]. There are ideas like a voting system for elections, where vote theft and rigging of results could not occur, and many other ideas, each of which aims to improve the current system, which has its virtues and many flaws. The system's main problem with the central authority, with a central database, is that such systems have one critical point - the system's bottleneck, which, if endangered, puts the whole system in danger.

Ethereum is a platform that works on top of blockchain technology which has revolutionized this field. It is a platform that allows its users to develop their decentralized applications [3]. It also resolves one additional issue in such networks: they cannot function without many users - nodes - who work on them.

Buying and renting multimedia content is the default functionality on all TVs, most often as Video On Demand (VOD), where users temporarily purchase rights to view a movie or series. With increased online shopping abilities, new options come to mind. What if a user can sell its rights to another user? Does it always have to be between the content provider and the end-user? In the real world, there are second-hand shops for used goods. Why don't such digital shops exist in the entertainment industry? Blockchain technology may be the right step towards that. It is reasonable to think that users would rather sell digital valuables than offer them free to the public. Some movements are done in that direction with the appearance of NFT (non-fungible token).

Some work has been done toward decentralized video streaming platforms using blockchain technologies in work by Tan et al. [4]. Focus that work was around content creators, advertisers, and consumers and the availability of the content in networks based on blockchain. More discussion on the usability of blockchain smart contracts in the entertainment industry could be found in the work of Pons [5].

Working consortium grouped around Sony, Samsung, and Google work on creating a next-generation video entertainment blockchain. A published white paper [6] presents the new architecture of the distributed content delivery network called Theta Mainnet 4.0. They have anticipated Theta Metachain and Theta Edge Network, as illustrated in Fig. 1. The first network is distributed blockchain of blockchains, where the "metachain" name comes from, resulting in support for an unlimited number of blockchains. The second network is responsible for interaction with users. Using publicly available Theta Video API, users can upload content and request access to content. The network has endpoints for protected content storage, encoding new content uploaded, and delivery to the user with an appropriate NFT-based digital rights management (DRM) system. It is expected to be available by the end of 2022.

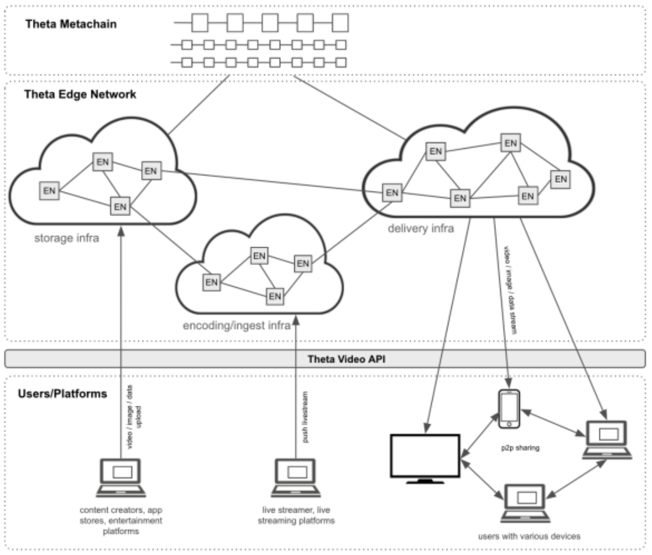

Theta Metachain
Theta Edge Network
EN
storage infra
encoding/ingest infra
delivery infra
video / image / data stream
Theta Video API
Users/Platforms
video / image / data upload
push livestream
p2p sharing
content creators, app stores, entertainment platforms
live streamer, live streaming platforms
users with various devices

Fig. 1. The architecture of Theta Mainnet 4.0 with Theta Metachain and Theta Edge Network

Igor Srdić – RT-RK Institute for Computer Based Systems, Novi Sad, Srbija, (e-mail: Igor.Srdic@rt-tk.com)

Đorđe Glišić – RT-RK Institute for Computer Based Systems, Novi Sad, Srbija, (e-mail: Djordje.Glisic@rt-tk.com)

Marija Jovanović – RT-RK Institute for Computer Based Systems, Novi Sad, Srbija, (e-mail: Marija Jovanovic@rt-tk.com

In this paper, we are investigating the possibilities of using blockchain technology in the world of digital television. In particular, we tried to use the smart contracts concept from the Ethereum ecosystem to allow users to buy video content from their homes [7]. Once a transaction is initiated, it takes more than 10 seconds for the change to be permanently stored in the blockchain, which is the waiting time for the return values of these functions. Detailed workflow can be seen in Fig. 2. This implies that Ethereum applications are not suitable for all systems that need faster data exchange and cannot correctly use current blockchain technology.

A smart contract as a programming code cannot be changed once it enters the blockchain. The mistakes and oversights are only solvable by creating new smart contracts and redirecting the applications to new contracts. The main problem is that errors are discovered only when they happen, and in such a system, the results of errors are harmful to the user.

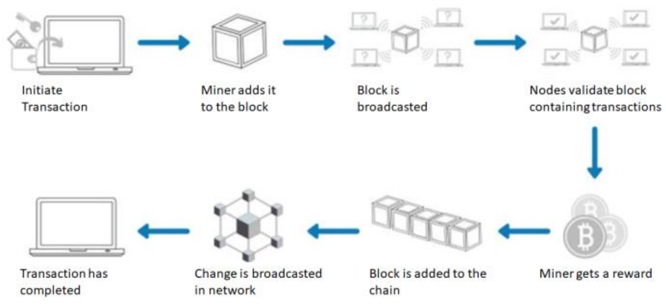

Initiate Transaction
Miner adds it to the block
Block is broadcasted
Nodes validate block containing transactions
Miner gets a reward
Block is added to the chain
Change is broadcasted in network
Transaction has completed

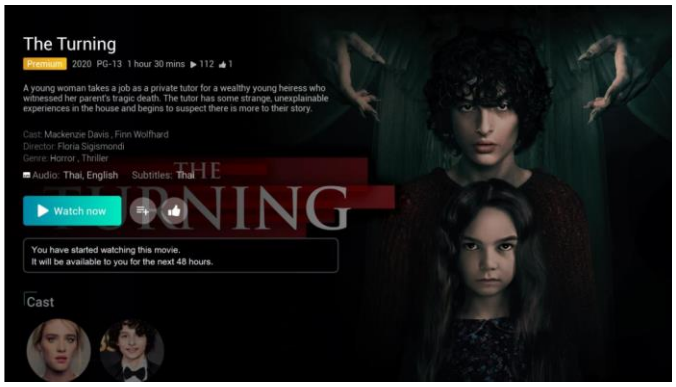
Fig. 2. Workflow for a transaction in the Ethereum network

In this work, smart contracts will be used for exchanging digital currencies for video-on-demand (VOD) services. We will introduce a digital wallet, an application used for storing digital funds (for example, cryptocurrency). On the side of the DTV device, a DTV application will be altered to support additional payment methods for its video-on-demand service. Those are the building blocks necessary to complete the task of creating a subscription management system with blockchain technology.

The paper consists of three parts. The second section will present functional requirements. The third section will explain all technologies and implementation of the system and the TV application adopted to be the client for the blockchain payment method. The fourth and last part contains an overview of the solution itself, shortcomings, and problems that have arisen during implementation and testing, and finally, we will discuss possible improvements.

II. Requirements

Purchase of content should be made on the chain itself. It should be implemented as an Ethereum smart contract that will provide functionalities such as purchase and storage purchase history. Purchased history has two parts, valid and expired. A valid subscription is the one that is still active and allows the owner to consume it (Fig. 3), in our case, to watch a movie or TV show. An expired subscription is essential for validating existing payments and resolving any problems. Additional requirements involve checking content access rights for specific users, content validation by the provider, etc. There is a set of functions related to administration, which can be performed exclusively by the provider. It must be ensured that the user cannot call them.


The Turning

Premium 2020 PG-13 1 hour 30 mins ▶ 112 👍 1

A young woman takes a job as a private tutor for a wealthy young heiress who witnessed her parent's tragic death. The tutor has some strange, unexplainable experiences in the house and begins to suspect there is more to their story.

Cast: Mackenzie Davis , Finn Wolfhard
Director: Floria Sigismondi
Genre: Horror , Thriller
Audio: Thai, English Subtitles: Thai

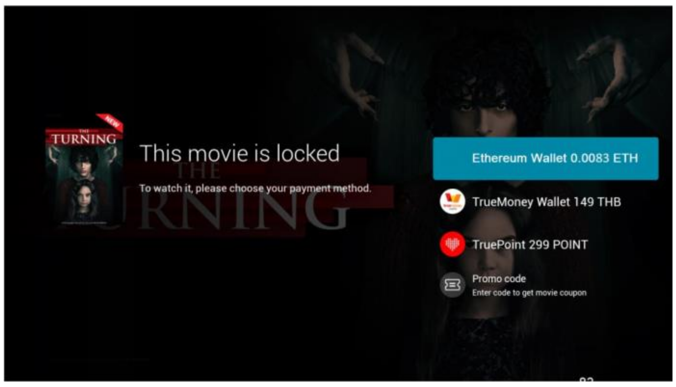
You have started watching this movie.
It will be available to you for the next 48 hours.

Fig. 3. The user interface for the Video on Demand service

In addition to the account user created in the TV application, the user also received his Ethereum account and can deposit money at physical ATMs or online. That order is permanently linked to a TV application account, and every content purchase is made through that account.

In this research, we did not want to use any currency assets. Instead, we selected the Ethereum test network Rinkeby. In this network, the currency has no real value, so we could deposit any amount of money into the account for testing purposes. For that reason, work does not cover depositing currencies. That does not reduce the applicability of the proposed solution, as only the target network has to be changed to make it usable in the real world.

Executing the method from smart contracts may take some time, but it is necessary to ensure that the application runs smoothly while waiting for the answers to those calls. An example screen for purchasing content is shown in Fig. 4. In particular, Android has a built-in capability to interrupt application execution that does not work for a while by sending ANR - Application Not Responding error.


This movie is locked

To watch it, please choose your payment method.

Ethereum Wallet 0.0083 ETH
TrueMoney Wallet 149 THB
TruePoint 299 POINT
Promo code
Enter code to get movie coupon

Fig. 4. The user interface for Video on Demand service when content is locked (needs payment)

After the purchase, the user should be able to view the content for 30 days before the first review or two days from when the content was first viewed. After the period expires, the content should be locked again until the next purchase. Given that money can be sent by calling smart contract functions, it is necessary to ensure that, in case of any error when calling these functions, the payments are aborted, and user funds remain untouched in the account.

III. Implementation

To implement our system, we had to implement two main components. The first one is smart to contact that will be positioned in the Ethereum test network. We used Solidity high-level object-oriented programming language especially designed for Ethereum networks. We selected Remix development environment (IDE), written in JavaScript and running under any web browser as a development environment.

On the client-side, we had an Android TV application [7] running on a set-top box (STB) device [8][9]. The application was written in Java. We have selected the web3j library for integration into the existing application to establish a connection with Ethereum smart contract API [10].

For storing the user's digital currency, we selected a digital wallet Metamask, which was used to create the Ethereum account. Its purpose is to save access to the digital currency details in the Ethereum network.

To simulate the Ethereum network, we created a network of miners. For that purpose, we used an Infura cluster made of full nodes. The cluster can execute contracts and alter the chain.

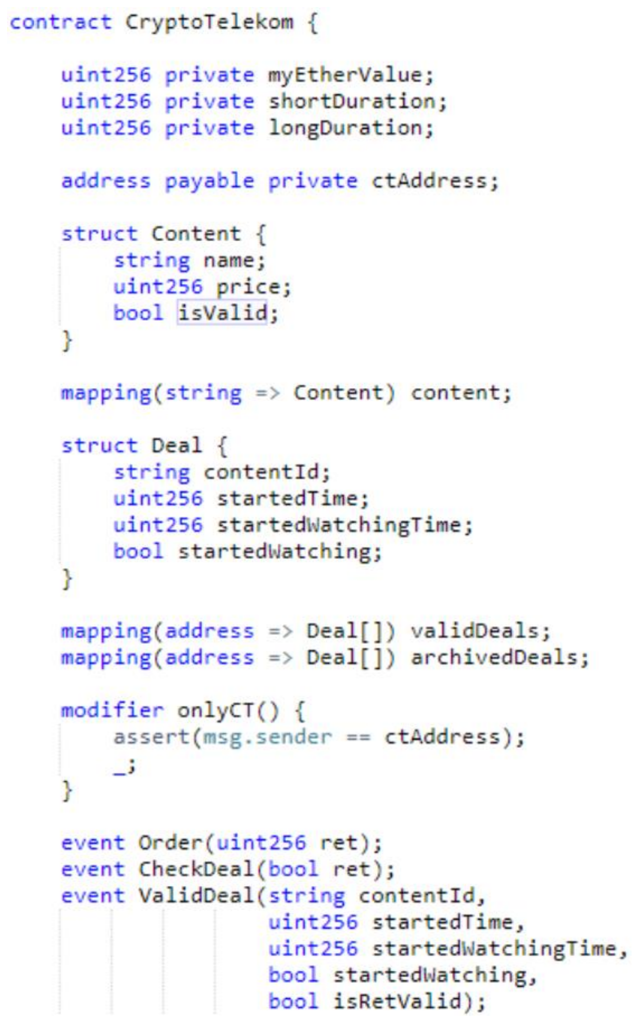
Part of the smart contract is present in Fig. 5. There are three fields of integer type. The *myEtherValue* field is an auxiliary used to convert some numerical data into currency Ether. At the same time, *shortDuration* and *longDuration* represent the lengths of the period in which the buyer is allowed to watch content. The *ctAddress* field is of the address type, the type of data it represents in Solidity public addresses of users in the Ethereum network. This is the address of the crypto-telecom (provider) which will have administrator privileges in the system.

In addition, there are two structures, *Content* and *Deal*. Structure *Content* represents the content and contains fields for name, price, and one auxiliary field. Structure *Deal* is proof of purchase, the contract concluded between provider and customer. It has the identifier of purchased content, the time of the purchase, the time when the user first looked at the content, and the field that indicates whether the content is still viewed. The content folder, which maps the content identifiers to data of the *Content* type, contains all available content of the provider. Maps *validDeals* and *archivedDeals* map users (addresses) into concluded contracts and are in the valid and expired contracts, respectively.

The Web3j library receives the output of the Solidity compiler [11][12] consisting of binary and ABI file and generates a Java surrounding class smart contract, in this case, class *CryptoTelecom*. Some of the methods of this class:

- *load* - loading the contract instance with the given address, web3j object, user credentials, and gas variables.

- *order* – Wrapper method of the smart contract order function. The return value is *RemoteFunctionCall <TransactionReceipt>*, which represents identification of transaction. Once the function is called, it will return its value after the function is executed, the state of the contract is changed, and that state is saved in the new transaction as part of a new block in the blockchain.



```
contract CryptoTelekom {

    uint256 private myEtherValue;
    uint256 private shortDuration;
    uint256 private longDuration;

    address payable private ctAddress;

    struct Content {
        string name;
        uint256 price;
        bool isValid;
    }

    mapping(string => Content) content;

    struct Deal {
        string contentId;
        uint256 startedTime;
        uint256 startedWatchingTime;
        bool startedWatching;
    }

    mapping(address => Deal[]) validDeals;
    mapping(address => Deal[]) archivedDeals;

    modifier onlyCT() {
        assert(msg.sender == ctAddress);
        _;
    }

    event Order(uint256 ret);
    event CheckDeal(bool ret);
    event ValidDeal(string contentId,
                    uint256 startedTime,
                    uint256 startedWatchingTime,
                    bool startedWatching,
                    bool isRetValid);
```

Fig. 5. The interface of the contract class in Solidity.

- *getOrderEvents* - Read events from the order method in a given transaction.

- *getNumValidDeals* - Wrapper method for function *getNumValidDeals()* from smart contract. Example of a return method that does not change the state of the contract. The return value is immediately available because only data is read.

Details about how we used Metamask, Rinkeby network, and Infura cluster can be found in [13]. Also, the authors omitted implementation details about the Android Java application as reference code can be found in [9]. Work does not lose clarity or applicability without those details, as that information can be found both in given references and online resources.

IV. Conclusion

The novelty that this approach brings, and at the same time one of its advantages, is that proof of the contract is no longer stored in the provider's database - all are stored on the Ethereum chain. This is interesting for the provider for two reasons. The first is that maintenance of the database by the provider is completely avoided, and the second is that no one can write data to a chain without the consensus of the entire network. Therefore, no one can falsely present themselves as having the right to specific content. Furthermore, users who purchase content are confident that their content rights will remain untouched because data ends on a chain that cannot be changed.

The virtues of this approach are numerous. However, there are also drawbacks. Developers must pay close attention when developing smart contracts because consequences in a system that operates with money can be substantial [14]. Also, it is important to optimize the program code as much as possible. The gas is charged to the initiator of the transaction. Calls to the smart contract function are calculated for every assembler instruction executed. Therefore, additional effort should be made to make the code as optimal as possible to lower gas prices. Also, as transaction initiators are charged for gas, and transactions are initiated by the users who buy the content, the provider has no operating cost for the network. Another approach is to subtract the price gas from the total price of the content that the user buys, which would result in the situation where the provider pays for network service and the user for the content.

References

[1] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system." Decentralized Business Review, 2008.

[2] M. Å. Hugoson, “Centralized versus decentralized information systems.” IFIP Conference on History of Nordic Computing. Springer, Berlin, Heidelberg, 2007. Vitalik Buterin, “Ethereum Whitepaper”, https://ethereum.org/whitepaper/, 2013.

[3] Y. Tan, S. Kadhe, K. Ramchandran, “Proof-of-Stream: A Robust Incentivization Protocol for Blockchain-based Hybrid Video on Demand Systems”, UCB/EECS-2021-42, 2021

[4] J. Pons, “Blockchains and smart contracts in the culture and entertainment business”, Réalités industrielles, 2017

[5] N. Szabo, “The Idea of Smart Contracts”, https://nakamotoinstitute.org/the-idea-ofsmart-contracts/, 1997.

[6] Theta Labs, “Theta Mainnet 4.0 - Introducing Theta Metachain to Power Web3 Businesses”, accessed May 2022, https://assets.thetatoken.org/theta-mainnet-4-whitepaper.pdf

[7] I. Pan, N. Lukić, “Design and architecture of software systems: Android-based systems”, FTN Publishing, Novi Sad, 2015.

[8] K. Yaghmour, “Embedded Android”, O ‘Reilly Media, 2013.

[9] N. Schapeler, “A example to Ethereum Development On Android using Web3j and Infura”, accessed May 2022, https://medium.datadriveninvestor.com/an-introduction-to-ethereum-development-on-android-using-web3j-and-infura-763940719997

[10] "Ethereum documentation", accessed May 2022, https://ethdocs.org/en/latest/

[11] "GitHub", Solidity, accessed May 2022, https://github.com/ethereum/solidity

[12] “Solidity documentation”, accessed May 2022, https://solidity.readthedocs.io/en/v0.6.9/.

[13] I. Srdic, BSc thesis “Realizacija multimedijalne pretplate korišćenjem blokčejna”, ETF, 2020

[14] K. O'Hara, "Smart Contracts - Dumb Idea," in IEEE Internet Computing, vol. 21, no. 2, pp. 97-101, Mar.-Apr. 2017, doi: 10.1109/MIC.2017.48.