**by Josef Sevcik[1], Pavel Kravchenko[2], Bohdan Skriabin[3] and Matej Michalko[4]**

## Abstract

DECENT is an independent decentralized peer-to-peer network owned by its users allowing digital content sharing in order to provide borderless information and data flow while avoiding any third party influence. As Bitcoin, DECENT uses blockchain in order to ensure transparency, security, trust, efficiency and cost effectiveness. A modified distributed hash table is used for data storage management. Asymmetric cryptography and advanced secret sharing algorithms are used to secure proper encryption and decryption of data. DECENT enables cost effective distribution of any text, music, picture or video, including media streaming. As an open source platform DECENT permits everyone to build their own application on top of it.

## 1. Introduction

[the start of the sentence is missing, please complete!] domains of application. By providing seamless automation, clearing and settlement can run without any human interaction and under control of an indestructible set of business rules. Similarly, Blockchain technologies enable banks' assets to be exchanged without third party validation. True innovation is rare, especially in the media world. Even more difficult it is to underpin which aspect of media thrives most for development. The majority of mainstream media does not in fact truly innovate as it is satisfied with its current business models. Whereas most of its incumbents do not provide as many benefits to the *content consumers* as they could because of lack of incentives. The market of digital content distribution is dominated by oligopolies and their centralized 'for-profit' platforms. Moreover, having the obligation to use a third party to access digital content is unnecessary. The market of digital content content distribution can redefined as a competitive one for the benefit of its consumers. Blockchain, as a publicly distributed ledger of transactions, has a great potential to be the next big thing since the advent of World Wide Web. As we write, many a financial institutions investigates this field and some of them even have their own blockchain research departments. Blockchain technologies can help banks to reduce their operative and infrastructural costs in various domains of application.

### 1.1 Conventions

In the scope of this document, we use a number of nomenclature conventions for the formal notation.

[1] Chief Architect, DECENT. Avenue Industrielle 4, GE-1227 Carouge, Switzerland. http://decent.ch. josef@decent.ch.

[2] FUNCTION?, DECENT. pavel@decent.ch

[3] FUNCTION?, DECENT bohdan@decent.ch

[4] Founder and CEO, DECENT. m@decent.ch

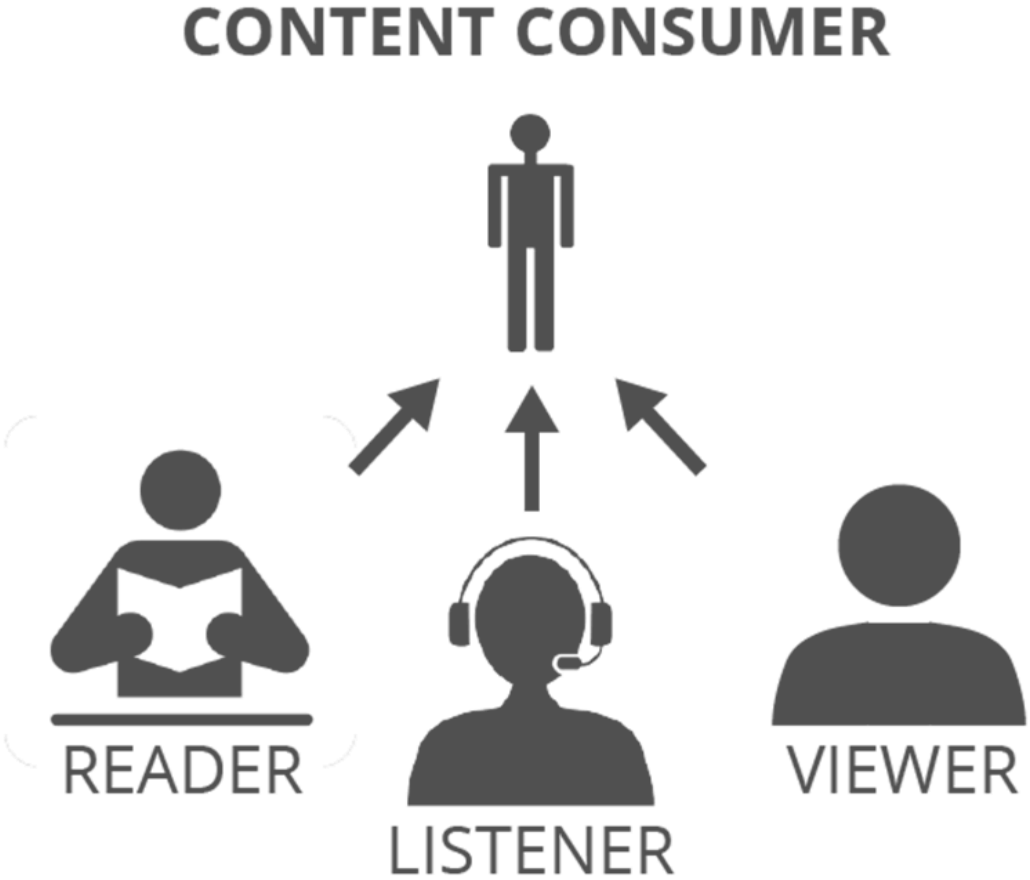


**Figure 1.3.1: Roles in DECENT**

One can see illustrated in Figure 1.3.1 the different types of content consumer. Content consumers are a generalization of readers, listeners and viewers. Authors produce content and upload it to the network. Publishers are the key element in keeping DECENT network up and running. The incentive for their time and resources are DCT digital tokens, similar to BTC sent to Bitcoin miners.

## 2. General description

DECENT was designed with principles of functional decomposition. We have developed three independent layers, each serving specific functions:

1. The **data distribution layer**, which is used for storing and sharing huge amounts of user-generated content.
2. The **blockchain**, where all the transactions and contracts happen in a publicly verifiable manner. Equally, blockchain is used for storing permanent data. These [data?] are buying transactions, publishing transactions, ratings, etc.

3. The **recommendation layer**, consisting of a set of independent recommendation engines, helping the consumer to select most relevant content.

The functionality of each layer is further described in later chapters. From a user perspective, there are three **roles** the user can have:

1. The **author**. The author creates a new content and publishes it on the network.
2. The **content consumer**. The **content consumer** reads, watches, listens or otherwise consumes the content created by the authors.
3. The **publisher** and **miner**. The publisher plays a vital role in the network: the publisher stores the content created by the author and distributes it to the consumers. The miner verifies the transactions and realizes payment distribution and is therefore responsible for the consistency of the network.

The interaction and flow principles are as follows:

- The author can set a price for the content. Only users who paid the price set by the author can access the content. On the other hand, the payment will be processed only if the user verifiably received the content.

- The author pays the publisher for acting as storage and content distribution network. The payment will be released over time to motivate the publisher to keep the content as long as required. The payment for network storage is also effective anti-spam mechanism.

- The consumer can rate the content. These ratings can be used for various purposes, e.g. assuming content quality, author's reputation or by recommendation engines.

- The network/protocol shall be as secure as possible to ensure mutual trust. No unauthorized user can get the content. The payment can be processed only if the content was delivered.

- 100% of the payment for the content goes to the author. There are no network or processing fees.

- Author and consumer can remain fully anonymous if they wish it.

### 2.1 Proof of Retrievability/Proof of Custody

Proof of retrievability or proof of custody is a schema, where the prover, also known as a data storage server, proves that he is in possession of a given data. Due to the nature of the DECENT blockchain technology, the proof must be verifiable by the miners who have no knowledge about the actual data, thanks to its total encryption. Therefore, additional features of the scheme are zero knowledge, publicly verifiability and non-interactivity. We studied and tested various publicly verifiable zero-knowledge proof schemas, and selected the one described in Shacham and Waters (2008)[5],as a most suitable candidate.

[5] **Shacham, H. and Waters, B.** "Compact proofs of retrievability". Cryptology ePrint Archive, Report 2008/073; 2008. http://eprint.iacr.org/. chapter 3.3.

The implementation of the pairings over elliptic curves is based on the PBC library, described in Lynn (2007: X)][6].

### 2.2 Content distribution and secret sharing

The digital content to be shared is represented by one or more data files, of which some can be marked as “secret” and some as “public”. The secret part of the content is to be encrypted using AES cipher with 256 bit key length. The key has to be new and unique. Once the relevant files are encrypted, the info dictionary/torrent file according to Loewenstern and Norberg (2007 : X)[7] and Hazel, G. and Norberg (2008)[8] is created. In the nodes’ key, own public address and the addresses of the selected publishers have to be filled.

The author has to announce itself in the DHT using mechanism described in Loewenstern and Norberg (2007 : Idem).

Magnet link is created according to Cohen (2008)[9]. The content is then shared with the publishers, who have to download first the info dictionary, announce themselves in DHT and then download the content.

The encryption/decryption key is split using Shamir secret sharing (Shamir, 1979)[10] into *n* particles, while *m* is the required number of particles to restore the key. The values *m* and *n* are user-selected and shall fulfill that $n>m>2$. The *n* represents the number of selected publishers.

The key particles are further processed as described in chapter 3.7.

Since the torrent info-hash is published in the blockchain, the consumer can easily construct the magnet link and download the info dictionary from the publishers (Hazel, G. and Norberg,2008).

Any party can use extension to this mechanism that ensures anonymity and/or privacy, e.g. the one from the Tribler project (Trible, DATE?)[11].

### 2.3 Blockchain scalability

According to our experience, Blockchains of common Blockchain based applications, such as Bitcoin, can grow to a size that is not convenient for daily use for most people, i.e. 50 GB in the case of Bitcoin. In order to prevent the uncontrolled growth, we evaluated and adopted the mechanism as described in section 2 for reclaiming disk space. Since all transactions are written in blocks and hashed in Merkle trees, some transactions can be deleted from the given block assuming they are replaced by their hash, or a hash of a respective subtree, if all transactions inside the subtree are to be deleted. This way the

[6] **Lynn, B.** “On the implementation of pairing-based cryptosystems”; 2007. https://crypto.stanford.edu/pbc/thesis.pdf

[7] **Loewenstern, A.  and Norberg, A.** “BEP0005: DHT Protocol”; 2008. http://www.bittorrent.org/beps/bep_0005.html.

[8] **Hazel, G. and Norberg, A.** “BEP0009: Extension for Peers to Send Metadata Files”; 2008. http://bittorrent.org/beps/bep_0009.html.

[9] **Cohen, B.** “BEP0003: The BitTorrent Protocol Specification”; 2008. http://www.bittorrent.org/beps/bep_0003.html.

[10] **Shamir, A**. "How to share a secret". Communications of the ACM 22 (11): 612–613; 1979

[11] Tribler **REFERENCE MISSING IN BIBLIOGRAPHY!**

hashes used for constructing blockchain remain untouched, and significant amount of space can be reclaimed.

Transactions candidates for deletion are the following:

1. transfer and coinage transactions that are already spent and the spend transactions are confirmed;
2. *deliver_keys* transactions of a content which validity has expired;
3. *proof_of_custody* transactions of a content which validity has expired;

Would more significant saving be required, additionally also the *content_submit*, *request_to_buy* and *leave_rating* transactions related to the expired content can be possible candidate for deletion. However, in such case, it must be taken into consideration that other tools, e.g. recommendation engines, are using these transactions. Hence, a sufficient amount of time should be kept to ensure they included it into the database or there is enough backup copies available for their use. We will evaluate the exact saving potential based on real data, when it will be clear how much metadata in average the authors publish in their transactions and the quantity [number?] of the transactions.

### 2.4 Recommendations

The author can use existing ID while publishing a piece of content. This way they can build up reputation in the network together with their content, and the newly published content inherits part of the reputation. The consumers provide two types of rating events: the implicit recommendation by buying the content, and the explicit recommendation by rating the content. Both are represented by transactions in the blockchain. A recommendation engine can import and utilise these events to provide recommendation to the consumers. DECENT evaluated various algorithms and built its recommendation engine on Alternate Least Square algorithm [8].

## 3. Transaction engine description

DECENT keeps records of shared data. In order to have the operations over these data verifiable and safely stored, the transactions are stored in a Blockchain, a distributed database with a continuously growing number of transaction records.

### 3.1 DECENT Transaction Types

DECENT transaction engine is principally based on the Bitcoin's transaction idea. New types of transactions are added to the DECENT protocol, mainly following the same principles. For example, Bitcoin's CoinBase transaction has a specific format and handling rules. There is always only one per block, it is always the first one in the block, it has single empty input and its outputs must wait 100 confirmations until they become spendable. Peercoin's CoinStake transaction[12] has another type. It is second in the block, its inputs have always required coinAge, and its out value is input value summed with the reward value.

DECENT stores transaction data in a Blockchain and uses separate layer (off chain) for data transfer.

[12] Anonymous. Peercoin https://peercoin.net/whitepaper.S.D.

There are 6 new transaction types (compared to Peercoin's codebase):

- ready_to_publish
- content_submit
- proof_of_custody
- request_to_buy
- deliver_keys
- leave_rating

### 3.2 DECENT script flags

Bitcoin uses the script language to reach flexibility of parameters which determines what is needed to spend coins and then corrects validation of transactions. DECENT extends Bitcoin's script with adding some functionality to validate the content sharing transactions.

New DECENT OP_CODES are (compared to PeerCoin's codebase):

- OP_CHECKPROMISETOPAY
- OP_CHECKDELIVERKEYS
- OP_CHECKPROOF
- OP_CHECKRATING
- OP_READYTOPUBLISH

### 3.3 Fee

Each transaction includes a small fee. In DECENT, this fee is destroyed, that means, it goes to no one. The main idea of the fee is to prevent spam transactions, but the publishers do not receive fees as a reward for block generating. Nevertheless, publishers who participate in Proof-of-Stake block creation are motivated to include transactions into their block, because they have additional rewards for publishing process.

### 3.4 Transaction creation

**Basic coin transfer TX**

| Field | Script |
|---|---|
| Input-1 | custom bitcoin input |
| Input-... | custom bitcoin input |
| Output-0 | custom bitcoin output |
| Output-... | custom bitcoin output |

Everyone can send DCT using a basic transaction type, you can add as many inputs and outputs as you want.

**Ready_to_publish TX**

| Field | Script |
|---|---|
| Input-1 | <Sig> <PubKey> |
| Input-... | custom bitcoin input |
| Output-0 | custom bitcoin output (actually it is change) |
| Output-1 | OP_RETURN “ready to publish” |

Using this recording new publishers report Authors that they are ready to get and share content.
The last output says that publisher with PubKey specified in the first input is ready to share the space on his disk.

**Content_submit TX**

| Field | Script |
|---|---|
| Input-0 | custom bitcoin input script |
| Input- ... | custom bitcoin input script |
| Output-0 | custom bitcoin change script |
| Output - 1 | OP_RETURN <publisher’s PubKey> <encrypted Shamir share> |
| Output - ... (n items) | OP_RETURN <publisher’s PubKey> <encrypted Shamir share> |
| Output n + 1 | OP_RETURN <magnetHash> <min number of shares> <lifetime> <size> |
| Output n + 2 | <price> OP_CHECKPROMISETOPAY |
| Output n + 3 | <CustodyData> OP_CHECKPROOF |
| Output n + 4 | OP_RETURN <synopsis> |

This is the structure of a transaction used by authors to submit new content to the network. To build this transaction an Author needs to choose number of seeders (publishers who have ability to share content), min number of seeders (min number of publishers that allows the buying process to occur) and the content price. Furthermore, the author must reserve coins to pay publishers, this value is actually destroyed as a fee and then publishers receive rewards from the newly generated coins.

**Proof_of_custody TX**

| Field | script |
|---|---|
| Input-0 | <proof-of-custody> <sign> <pubKey> |
| Input-1 | custom bitcoin input script |
| Input- ... | custom bitcoin input script |
| Output-0 | custom bitcoin change script |

With this type Publisher can push to Blockchain a proof that he successfully downloaded and is ready to share content. Input-0 contains the proof, signature and public key specified as publisher.

**Request_to_buy TX**

| Field | Script |
|---|---|
| Input-0 | *empty* |
| Input-1 | custom bitcoin input script |
| Input- ... | custom bitcoin input script |
| Output-0 | custom bitcoin change script |
| Output-1 | OP_CHECKDELIVERKEYS |
| Output-2 | <passWordHash> OP_CHECKRATING |

With this transaction type, content consumer can push requests to buy certain content. The content is referenced by the Input-0. The script of the Input-0 is empty. Furthermore, this transaction must have a promise-to-pay (see “promise to pay” section). In the Output-2 content consumer must specify hash of the secret password, it will be needed when leaving a rating.

**Deliver_keys TX**

| Field | Description |
|---|---|
| Input-0 | <encrypted share> <sign> <pubKey> |
| Input-1 | custom bitcoin input script |
| Input- ... | custom bitcoin input script |
| Output-0 | custom bitcoin change script |

With this transaction type publisher can deliver a share to customer.

**Leave_rating TX**

| Field | Description |
|---|---|
| Input-0 | <passWord> |
| Input-1 | custom bitcoin input script |
| Input- ... | custom bitcoin input script |
| Output-0 | custom bitcoin change script |

To leave the rating, content consumer must provide a pre-image of hash.

### 3.5 Finding a publisher

Each publisher is identified by his public key. When someone wants to become a publisher he can do it just by sending ready_to_publish transaction, where he must specify his public key. Author must choose a set of publishers for each content before uploading it. The DECENT protocol has a default search mechanism for that purpose. It is actually based on the publisher's responsibility. It automatically gets a complete list of the publishers and chooses the most active subset of them.

### 3.6 Promise to pay

Promise to pay is a proof that a content consumer has enough money to pay for content and that he cannot spend it later. It is implemented in the request_to_buy transaction: customer builds a transaction in which the difference between the input and the output values equals to the fee summed with the content price. As a result content consumer has burnt the price value and can not spend it. If the request occurs before a period of time has expired, the author will receive newly generated coins. Otherwise the customer will receive the payment.

### 3.7 Buying process

The buying process consists of the following steps:

1. Sending request for buying content
2. Downloading encrypted content
3. Decryption and compile an encryption key

During step 1, the content consumer specifies an actual content and freezes coins to pay for it. In such way the consumer shows to publishers that he wants to buy a certain content. Now each publisher sends a share of encryption key which is encrypted with the public key of consumer, which will be stored in blockchain. During step 2, the content consumer downloads encrypted content from the network. If a minimum number of shares is delivered during one day since the request, freezed coins will go to the author's address. Otherwise freezed coins are going to be returned to the content consumer. As final step, the content consumer gets keys delivered from blockchain, decrypts each of them and compiles the encryption key. He decrypts the content with symmetric encryption key.

### 3.8 Repeatedly spendable outputs

Several requests to buy can be referenced to the same content, but basic protocol doesn't support double referencing (double spending). In such cases the DECENT transaction system has special outputs which have the ability to be spent repeatedly. In a content_submit transaction the outputs n + 2 and n + 3 are repeatedly spendable. Anyone can create a request_to_buy transaction. If you are a publisher for that content you can create a proof_of_custody transaction referencing the repeatedly spendable output. In a request_to_buy transaction the output for delivery of keys is also repeatedly spendable. If you are a publisher for that content you can create a deliver_keys transaction with a reference to a repeatedly spendable output. In total, there are three types of many time spendable outputs. They can have only a zero value because it is needed to count buying process and have no connection with money.

### 3.9 Transaction validation

Although DECENT transactions use the same script language as Bitcoin, the validation process looks easy. However, sometimes the script's execution becomes slightly more complicated because of the content's duration and the specification of publishers.

### 3.10 Block validation

We have implemented additional verification step for each block to check whether the publishing performed by the block's transactions was correct. Everything else concerning the block's validation is still borrowed from Peercoin.

### 3.11 DECENT Content Sharing Mechanism

DECENT uses its own transaction-based engine to handle content sharing securely. In DECENT protocol, sharing is based on the chain of events. Each content's distribution has its own chain of events created as transaction records and stored on the Blockchain. We can simply specify five groups for those events:

```
[UPLOADS] --> [PUBLISHING] --> [REQUESTS] --> [BUYING] --> [RATING]
----------------------------------
BLOCKCHAIN
===================
[BLOCK]<->[BLOCK]<->[BLOCK]<->[BLOCK]<->[BLOCK]<->[BLOCK]<->[BLOCK]<->[BLOCK]
```

### 3.12 Transaction Model

The upload event is triggered when the content_submit TX is sent. The chain always starts from the upload event. The publishing event is triggered when the proof_of_custody TX is sent. Content could have as many publishers as the author has specified when uploading it.

```
[content_submit TX] --- [proof_of_custody]
                         \
                           [proof_of_custody]
                         \
                           [proof_of_custody]
```

Content could have unlimited number of request_to_buy offers from customers while the publishers are still sharing.

```
[content_submit TX] --- [request_to_buy]
                         \
                           [request_to_buy]
                         \
                           [request_to_buy]
```

The buying event is triggered when deliver_keys transactions are created by publishers. Each specified publisher can take action to deliver key for every request.

```
[request_to_buy TX] --- [deliver_keys]
                         \
                           [deliver_keys]
                         \
                           [deliver_keys]
```

When buying process is finished, the customer can issue the leave_rating TX, it the last event in the chain.

```
[request_to_buy TX] --- [leave_rating]
```

### 3.13 PublisherMiner

In the Bitcoin protocol there is a special entity named “Bitcoin Miner” which works to generate new coins according to the Proof-of-work concept. DECENT has a PeerMiner module to do roughly the same. DECENT actually has one more worker who tries to generate new coins by taking care of sharing the content. DECENT uses a PublisherMiner module that is looking for the newly uploaded content in which the user is specified as a publisher and starts downloading each content that matches. He will be the seeder of this content for customers, and will receive rewards while the proof_of_custody transactions, submitted by him, are not outdated in the Blockchain. Furthermore, PublisherMiner looks for recent request_to_buy transactions and takes action to deliver the key if he has it. He will receive rewards for each successful delivery too.

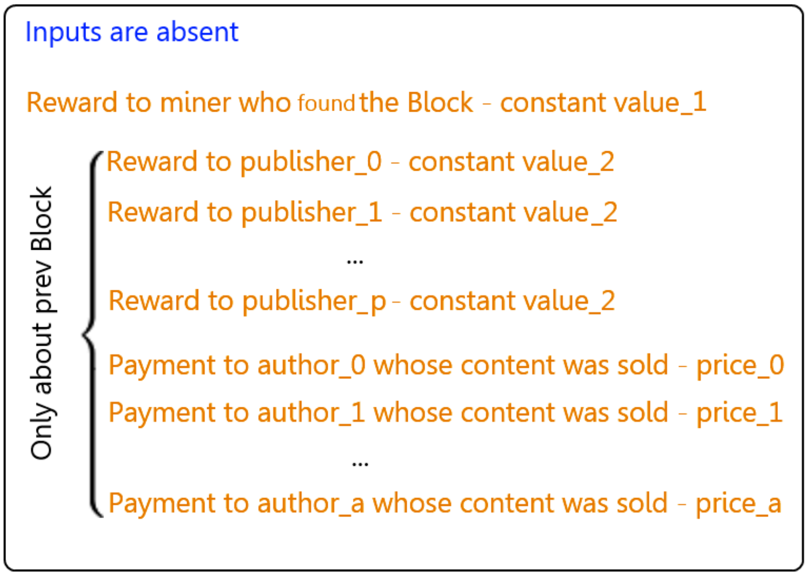
### 3.14 DECENT Reward Distribution

In Bitcoin, the CoinBase transaction is used for distribution of newly generated coins with proof-of-work and it is always the first transaction in the block. Hence, CoinBase has only one empty input and can contain many outputs and in the sum out value are newly generated coins plus fees. DECENT uses CoinBase transaction for distribution of publishing rewards. Here are only the Output-0 is designed for proof-of-work rewards, and others for publishing. Each node which performs block generating build block with mandatory transactions:

- CoinBase (always)
- CoinStake (if PoS has started)

The CoinBase tx shall include PoW reward in the Output-0 (once the PoW has finished, it will remain empty) and publish rewards optionally. If PoS has started, CoinStake tx shall be the second transaction in the block, and it is the same like in PeerCoin. It is important to note that the CoinBase transaction includes PoW reward for current block and the rewards for publishing events up to previous block. It means that distribution of DECENT rewards in current block doesn’t cover events occurred in the same (current) block.

*Coinbase TX - first TX in every Block*


Inputs are absent
Reward to miner who found the Block - constant value_1
Only about prev Block
Reward to publisher_0 - constant value_2
Reward to publisher_1 - constant value_2
...
Reward to publisher_p - constant value_2
Payment to author_0 whose content was sold - price_0
Payment to author_1 whose content was sold - price_1
...
Payment to author_a whose content was sold - price_a

### 3.15 Computing value to pay for custody

Each author shall give a promise-to-pay for chosen publishers which will store and share his content. The value in the promise is derived from several parameters: content size, number of publishers and content lifetime. The default value for that is 0.00001 DCT per MB for one publisher per day. So when the author is uploading new content, he reserves some coins for storing it in the DECENT network with the promise-to-pay concept.

### 3.16 Rewards for miners

Reserved amount of DCT coins is distributed to the miners holding the content. Once per 24 hours the miner is expected to publish proof_of_custody for each content they hold, and then they receive the proportional part of the reserved amount, calculated by the following formula:

$$R = \frac{T}{86400} * \frac{3+\frac{T}{86400}}{4}$$

The goal is to maximize reward for miners that are staying online and thus making the content available, and discourage them to publish the content too often. Ideally, they are expected to publish proof_of_custody once per day.

In addition, miners are rewarded for delivering the keys with deliver_keys transactions, in the amount of 0.000001 DCT per transaction.

### 3.17 Vulnerability

Author can tamper with the rating of his own content. According to the DECENT rules the content consumer can rate a content only after purchasing it, further once per each request to buy transaction. Since the author will receive the price payment to himself he can issue so many requests and as a result he can leave rating many times actually spending many only for fees.

## 4. Use Cases

It seems appropriate at this stage to examine the way media works nowadays. Authors have to go through publishers, recording studios or governments who decide whether the content can be, should be or is good enough to be released. Any app builder will easily evaluate how many restrictions there are. Hence DECENT offers to app developers a safe and secure way for their apps' users to share the content freely. Not only will it provide direct to readers/authors a payment system through blockchain and data distribution. It also includes the recommendations that secure the quality and popularity of the content.
The typical use case of DECENT can be the publishing of articles and stories, similar to Medium[13]. The author uses the application to write and organize articles or add media files. When the author is happy with the result, he or she presses the "Publish" button. Later the author can specify the price for the content, select a part of the article that will be free to read and add metadata. The application will encrypt the content, then find the publishers. These are independent computers connected to the DECENT network running publishing software to keep the network running and receive a reward for doing so. It will thereafter calculate the publishing fee and, after confirmation, it will instruct the publishers' computers to download the content and broadcast any relevant metadata over blockchain. Once content consumers find the content of their interest, they may be notified that their favourite author has published a new article. They will then get recommendations based on their preferences or they will simply browse newly published content. They can choose to download and read the "free to read" part. And then they can decide to buy the rest of the article by paying the small fee specified by the author. Finally, the DECENT protocol will process the payment that will be attributed to the author and the content consumer's application will get the decryption keys for the rest of the article. Over time, *publishers* will be rewarded for storing content and will get their fair share of the publishing fee the *author* has paid. Everyone is welcome to build applications or clients on the top of the DECENT protocol with their independent business models. This will enable *authors* to share their content. It can be any kind of digital content: video or audio files, texts (books, articles, news) or pictures. And it actually offers all sorts of possibilities and opportunities, for example:

- Medium like blogging and publishing
- Soundcloud like music publishing
- Amazon like e-book publishing
- Software sale
- Shutterstock like photo sharing
- Electronic newspaper publishing
- Cost-effective academic paper publishing
- Video streaming
- Audio streaming

[13] https://medium.com/

### 5. Bibliography

**Anonymous**. Peercoin https://peercoin.net/whitepaper.S.D.

**Cohen, B.** “BEP0003: The BitTorrent Protocol Specification”; 2008. http://www.bittorrent.org/beps/bep_0003.html.

**Hazel, G. and Norberg, A.** “BEP0009: Extension for Peers to Send Metadata Files”; 2008. http://bittorrent.org/beps/bep_0009.html.

**Loewenstern, A. and Norberg, A.** “BEP0005: DHT Protocol”; 2008. http://www.bittorrent.org/beps/bep_0005.html.

**Lynn, B.** “On the implementation of pairing-based cryptosystems”; 2007. https://crypto.stanford.edu/pbc/thesis.pdf

**Nakamoto, S.** “Bitcoin: A Peer-to-Peer Electronic Cash System”; 2009. https://bitcoin.org/bitcoin.pdf.

**Shacham, H. and Waters, B.** “Compact proofs of retrievability”. Cryptology ePrint Archive, Report 2008/073; 2008. http://eprint.iacr.org/.

**Shamir, A**. "How to share a secret". Communications of the ACM 22 (11): 612–613; 1979.

**Takács, G. and Tikk, D.** “Alternating Least Squares for Personalized Ranking”. RecSys’12; 2012.