RESEARCH OR PRACTICAL ARTICLE

# Smart Contracts for Research Data Rights Management over the Ethereum Blockchain Network

Adrian-Tudor Pănescu[a,b] and Vasile Manta[a]

[a]Faculty of Automatic Control and Computer Engineering, “Gheorghe Asachi” Technical University of Iaşi, Dimitrie Mangeron 67, 700050 Iaşi, Romania; [b]figshare, 1 Mark Square, EC2A 4EG, Lodon, UK

**ARTICLE HISTORY**
Compiled September 7, 2018

**ABSTRACT**
This paper presents a new method for managing digital reuse rights of research data which leverages technologies such as the blockchain and smart contracts. This allows, on one hand, the creation of a permanent record on the agreements between the authors of the data and the reusers, with the possibility of verifying compliance at any time, and, on the other hand, a higher level of granularity on defining the conditions of reuse. A practical implementation of such an workflow using the Solidity smart contract language is included, along with a brief analysis over the Ethereum blockchain network.

**KEYWORDS**
smart contracts; digital rights management; research data sharing; repository technologies.

## 1. Introduction

Data sharing has become essential to modern research, mainly due to the requirements of various funding agencies to make publicly-funded research open (European Commission 2017; National Institutes of Health 2018), as a direct response to the so-called *reproducibility crisis*, triggered by a number of studies that failed to reproduce previous results (Phillips 2017; Eklund 2016). Nevertheless, data sharing has not become an integral part of the research cycle, with less than half of the respondents in a survey carried out in 2017 among Springer Nature and Wiley authors reporting to share data *frequently* (Digital Science 2017). Apart from more technical reasons (e.g., difficulty in preparing or storing data), various studies (Federer et al. 2015; Youngseek and Zhang 2015) have identified *perceived risks* in data sharing activities, largely due to the conditioning of academic success on publication volume and impact. A survey among Wellcome Trust awardees showed that *“the main barriers to data sharing are the fear for misuse and misinterpretation of data, the fear to lose publication opportunities [. . . ]”* (Van den Eynden 2016). Another study on articles published in Science between 2011 and 2012 found out that 11% of the authors refuse to share data if the requester does not provide information on how the material will be further used (Stodden, Seiler, and Ma 2018).

CONTACT Adrian-Tudor Pănescu. Email: tudor@figshare.com

With the rise of blockchain technologies, such as Bitcoin (Nakamoto 2008) and Ethereum (Ethereum Foundation 2018a), the concept of smart contracts, a protocol which allows to digitally facilitate, verify, or enforce the negotiation and performance of a contract, has been brought back to light (the concept has been initially defined in 1996-1997 (Szabo 1997)). Apart from the widespread use cases in the financial world, the blockchain and smart contracts have been proposed as potential solutions to various issues in scholarly communications, pertaining to publication sharing; traceability, copyright, and digital rights management could be solved by leveraging certain properties of these technologies, such as persistence and verifiability (van Rossum 2017).

This paper proposes a new method for sharing research data in a way that allows the authors to oversee how it is being accessed and reused. This protocol employs smart contracts in order to record and enforce the terms under which published research data can be utilized by subsequent studies.

The remaining of the paper is organised as follows: in Section 2 we provide some background on both blockchain technologies and research communication, in Section 3 we describe the smart contract protocol for rights management of research data, with a practical implementation and brief analysis in Subsection 3.1. The paper ends with some concluding remarks on future directions.

## 2. Background

### 2.1. *Blockchain*

A blockchain is a continuously growing list of records linked using cryptographically calculated values stored in each block. In its most popular implementation, Bitcoin (Nakamoto 2008), each block records the *hash* value of the previous block, a timestamp, and data about multiple *transactions* (algorithmic operations with one or more inputs and one or more outputs); such a list of transactions is called a *ledger*. Most such systems are implemented as a peer-to-peer network, in which each node stores a full copy of the blockchain, thus removing the requirement for a central authority that needs to verify all transaction data; practically, such systems implement a distributed database in which consensus is achieved via the proof for validating the sequence of blocks.

### 2.2. *Ethereum*

Ethereum (Ethereum Foundation 2018a) is an open platform for building decentralized applications on top of blockchains; it defines a number of protocols for running arbitrarily complex algorithms on the network. Such code is ran on Ethereum virtual machines, which are Turing complete; these virtual machines are stored on each node participating in the network, and each issued instruction is ran on every node. A valid state transition on these virtual machines is one which comes through a transaction (Wood 2018). Ethereum also implements its own value token, called *ether*, but the way in which this is used depends on the application being implemented. One purpose of the value token is to provide a representation of the physical resources required for participating in the network (e.g., electrical power), and, for example, most commonly each operation executed on the virtual machines carries an inherent cost, denoted as *gas* (Wood 2018).

### 2.3. *Smart Contracts and Solidity*

At the base of the Ethereum platform stand two types of entities, called accounts; accounts can be externally owned, which are controlled usually by human actors via private cryptographic keys, or can be contract accounts. Contract accounts are controlled by the code to be ran on the virtual machines, which can be activated only by an externally owned account (for a more detailed explanation see (The Ethereum Community 2018). Contract accounts implement *smart contracts*, systems that usually contain value tokens that will be unlocked only if certain conditions are met.

Solidity (Ethereum Foundation 2018b) is a contract oriented, statically typed, high-level language used for implementing smart contracts on the Ethereum virtual machine (but not limited to it). A contract in Solidity is defined, similar to a class in traditional object-oriented programming, as a collection of functions and data. The invocations of the functions, as well as a history of the values of the stored data, are stored in the underlying blockchain, making the execution of the smart contract fully traceable.

### 2.4. *Research Communication and Sharing*

While the scholarly communication domain is more complex and includes a number of workflows and activities, for the purposes of this paper we will focus on research data sharing. Currently, such output is either self-archived (e.g., authors will post it online on their own website), on platforms hosted by their current research institution, or on systems provided by the various research publishers (see, for example, the guidelines of Springer Nature (Springer Nature 2018b) and Elsevier (Elsevier 2018b)). In between stand services such as ZENODO (ZENODO 2018b) or Dryad (Dryad 2018), which provide free upload services for research content.

From a reuse point of view, the choice above makes little difference, as in any situation the publisher of the research data should specify under which terms and conditions the content can be reused. In certain cases, especially when the supporting article is to be published at a later time, the research data might remain under *embargo* (see for example ZENODO's *"Access and reuse"* policies under (ZENODO 2018a)); in other cases it might remain confidential indefinitely, thus making its reuse practically impossible without intervention from its author (e.g., release under specific clauses to a limited audience).

If the author decides to make the data publicly accessible, this is usually done by stipulating the terms under which it can be reused; the Creative Commons ((Creative Commons 2018)) suite of licenses is highly popular, with options raging from CC BY, which is highly permissive as long as attribution is explicitly specified, to CC BY-NC-ND which does not allow creating derivatives or using the content for commercial purposes. An issue with the current model of sharing and licencing stems from the difficulty of tracking each and every access and reuse of a shared data set. Implementing this process using smart contracts can help alleviate this issue, as every action (e.g., download data set, publish article reusing data set) is permanently recorded and can be inspected by the participating entities. Another issue is posed by the lack of flexibility when it comes to defining the terms of reuse; the Publication Manual of the American Psychological Association states that beside the limits on the distribution of the research data, there should also be an agreement on how the artefacts will be reused (e.g., only for the verification of the results) (American Psychological Association 2015). Current licencing choices might lack such granularity due to their broad range of applications, and publishing platforms are in general only restricted to such standardized options, mostly for practical reasons.

While using blockchain technologies for digital rights management has been theorized ((van Rossum 2017), (Xu et al. 2017)), we are not aware of any practical application on the specific workflow of research data reuse. In (Ramachandran 2017), the authors propose a system implemented over the Ethereum network for ensuring data provenance; while not tackling the same issue, this work provides a number of interesting pointers on issues common to our implementation, such as required number of participating entities and contract operation cost requirements.

## 3. A smart contract for managing reuse rights of research data

In this section we present the mechanism under which a smart contract for tracking digital rights enforcement over research data can function. While we will make references to the protocols imposed by the Ethereum network, here we present only a high level overview; an actual implementation, using the Solidity language, is included in Subsection 3.1.

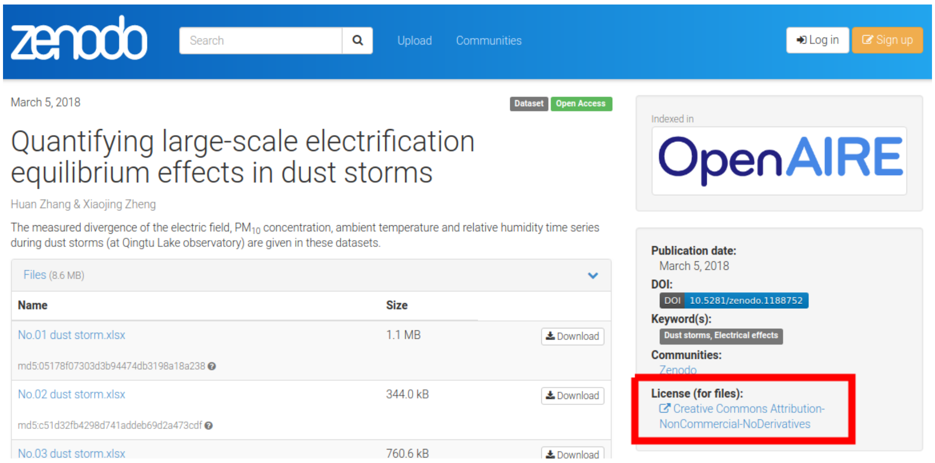
First, we define the two types of accounts required by the Ethereum network. In our case, there will be two externally owned accounts (EOAs), one for the author that initially publishes the data, and one for the entity the wishes to reuse it. The contract account is associated to the smart contract code that will regulate the reuse.

In terms of stored data, the following entities are considered:

(1) Author account address: this variable simply records the address of the EOA of the author (or producer) of the dataset.
(2) Dataset hash: this is a cryptographic hash of the research dataset that is to be published. Note that the smart contract (and underlying infrastructure) does not store the actual data files, but only their hash; the complete content can continue to reside on one of the existing platforms presented in Subsection 2.4.
(3) Dataset terms hash: this is a cryptographic hash of the terms and conditions, specifying the way in which the published dataset might be reused. We take here the same approach as with the dataset, of not storing the actual contents of the terms, as the current platforms presented in Subsection 2.4 have specific features for presenting such information, as the one in Fig. 1. Nevertheless, in some cases the author might opt for using a custom definition of the terms (e.g., embargoes such as *"reusers should not publish journal articles on findings based on the original dataset for one year since the initial release of the data"*) which cannot be hosted by such platforms, or there simply might be a requirement to ensure the preservation of the terms and, in such cases, the full text could be stored on the blockchain similar to (Brown 2015).
(4) Data reuser account address: similar to the first item in this list, this variable records the EOA address of the entity planning on reusing the data set.
(5) Reuse work hash: this variable records a hash (or any other type of appropriate information, such as a Digital Object Identifier (DOI)) for the work that reuses the original data set. This will ensure that the complete cycle of reuse is recorded on the blockchain, making it easy to inspect at any future moment.

Finally the smart contract will consist of three functions, that need to be called in the following order:

(1) Publish dataset: this function will record the author account address, dataset



March 5, 2018

Quantifying large-scale electrification equilibrium effects in dust storms

Huan Zhang & Xiaojing Zheng

The measured divergence of the electric field, $PM_{10}$ concentration, ambient temperature and relative humidity time series during dust storms (at Qingtu Lake observatory) are given in these datasets.

Files (8.6 MB)

| Name | Size |
|---|---|
| No.01 dust storm.xlsx<br>md5:05178f07303d3b94474db3198a18a238 | 1.1 MB |
| No.02 dust storm.xlsx<br>md5:c51d32fb4298d741addeb69d2a473cdf | 344.0 kB |
| No.03 dust storm.xlsx | 760.6 kB |

**Publication date:**
March 5, 2018
**DOI:**
DOI 10.5281/zenodo.1188752
**Keyword(s):**
Dust storms, Electrical effects
**Communities:**
Zenodo
**License (for files):**
Creative Commons Attribution-NonCommercial-NoDerivatives

**Figure 1.** A ZENODO (ZENODO 2018b) record (http://doi.org/10.5281/zenodo.1188752) with the CC BY-NC-ND licence applied on its files (highlight on right-hand side).

hash and dataset terms hash variables; optionally, it can emit an event (see the *Events* section of (Ethereum Foundation 2018b)) in order to advertise the release of the dataset and enable certain automatic workflows (e.g., a system similar to a RSS feed could be put in place to monitor the release of new data).

(2) Release dataset: this function records the data reuser account address, conceptually stating that the author of the dataset released it under specific terms to the requesting entity.

(3) Publish *rework*: this function records the reuse work hash, storing information about how the dataset has been reused, and closing the execution of the contract.

These three functions describe how the smart contract should be developed and executed from a practical point of view, integrated with the current research workflows. As soon as a dataset is published, a version of the smart contract should be deployed and the first function should be called in order to record the terms under which the work can be reused. When there is a wish to access the dataset, the two parties (author and reuser) should come to an agreement on following these terms and record this on the blockchain using the second function. The proof that the terms were actually followed will be recorded by calling the last function. A discussion on this last step relates to the entity that should call this operation; we have considered that the reuser can do this, testifying that the established terms have been followed, and recording this in the permanent record of the blockchain. If this is considered insufficient, an additional step can be considered, in which the original author of the dataset, or any other suitable entity, verifies that terms were indeed obeyed, closing the execution of the contract.

### 3.1. *Implementation of the smart contract using Solidity*

An implementation of the smart contract presented previously, using the Solidity language, is included in Fig. 2.

Two minor differences from the overview in the previous section can be noticed with the implementation:

- The `requestDataset` function allows expressing the intent to access the published dataset. This is implemented in order to further automate the workflow,

```
// version 1.0.2
pragma solidity ^0.4.0;


contract DatasetPublishReuse {

    address datasetAuthor;
    uint128 datasetHash;
    uint128 datasetTermsHash;
    address reuser;
    uint128 workHash;


    // event to be emitted  when a new dataset is published
    event evDatasetPublished(
        address _from,
        uint128 _datasetHash,
        uint128 _datasetTermsHash
    );

    // event to be emitted when an entity wishes to acquire a dataset for reuse
    event evRequestDataset(
        address _from,
        uint128 _datesetHash
    );

    // event to be emitted when a dataset is released for reuse
    event evDatasetReleased(
        address _from,
        address _to,
        uint128 _datasetHash,
        uint128 _datasetTermsHash
    );

    // event to be emitted when a work based on the original dataset is
    // published
    event evWorkPublished(
        address _from,
        uint128 _workHash,
        uint128 _datasetHash,
        uint128 _datasetTermsHash
    );


    function publishDataset(uint128 _datasetHash, uint128 _datasetTermsHash)
    public returns (bool) {
        datasetHash = _datasetHash;
        datasetTermsHash = _datasetTermsHash;
        datasetAuthor = msg.sender;
        emit evDatasetPublished(datasetAuthor, datasetHash, datasetTermsHash);
        return true;
    }

    function requestDataset(uint128 _datasetHash)
    public returns (bool) {
        assert(_datasetHash==datasetHash);
        emit evRequestDataset(msg.sender, _datasetHash);
        return true;
    }

    function releaseDataset()
    public returns(bool) {
        reuser = msg.sender;
        emit evDatasetReleased(reuser, datasetAuthor,
                               datasetHash, datasetTermsHash);
        return true;
    }

    function publishRework(uint128 _workHash)
    public returns (bool) {
        assert(msg.sender==reuser);
        workHash = _workHash;
        emit evWorkPublished(reuser, workHash, datasetHash, datasetTermsHash);
        return true;
    }

} // end contract
```

**Figure 2.** Solidity smart contract implementing the research data reuse terms management workflow.

| Operation name | Transaction cost | Execution cost | Total gas | USD price |
|---|---|---|---|---|
| Creating contract | 613654 | 421050 | 1034704 | $1.47134 |
| publishDataset | 55140 | 33612 | 88752 | $0.1262 |
| requestDataset | 23313 | 1913 | 25226 | $0.03591 |
| releaseDataset | 44728 | 23456 | 68184 | $0.09698 |
| publishRework | 30134 | 8734 | 38868 | $0.05524 |

**Table 1.** Gas costs as defined by the Ethereum Virtual Machine and network of the implemented smart contract for rights management of research data. Equivalents in US dollars, calculated using the rates available in March 2018, are included. While the *transaction cost* is an estimation of work required to define the contract on the network, the *execution cost* considers the actual execution cost of the transaction, based on the number of computing operations.

as without this functionality the requester would have to contact the author of the dataset using some other channel, such as email or the features provided by the platform storing the actual data set.

- Events are implemented and emitted at each invocation of the functions of the smart contract; as mentioned, this could prove useful for automating certain workflows.

We have also carried out a brief analysis of the costs of running such a routine on the Ethereum Virtual Machine and network, by considering the gas requirement for a full invocation of the smart contract. In order to calculate this we have measured the transaction cost of the functions in the smart contract (the gas cost for setting up the functions on the blockchain), and the actual execution cost (this depends on how much processing steps are required by an Ethereum Virtual Machine for executing the functions); these concepts are explained in depth in Section *Account Types, Gas, and Transactions* of (The Ethereum Community 2018). The costs are expressed in Ethereum network gas; we have set a rate of $2 * 10^{-9}$ ether per gas used, with the price of $710.95238 per ether (these values were current at the time of writing of the paper, in March 2018). The results are presented in Table 1.

As it can be observed, the total cost for setting up and executing the contract is below two US dollars. If researchers would be to support these costs, as opposed to having the network fully sustained by research publishers or any other scholarly communication entity, they are still negligible when compared to other usual expenses, such as, for example, Open Access publishing (e.g., the costs required by Springer Nature and Elsevier can easily go over $1000 (Elsevier 2018a; Springer Nature 2018a)).

An interesting aspect that stems from this analysis and takes inspiration from the widespread use of blockchain technologies in financial workflows relates to the implementation of incentives for both publishing research data and considering replication studies. Given that the framework for implementing such mechanisms is inherent in the platform which we targeted, it is fairly easy to extend the smart contract to consider, for example, awarding a number of value tokens (ethers) when a new data set is published; similarly, a number of value tokens could be subtracted from the EOA of the reuser, thus creating further incentives to stay in line with the established terms.

## 4. Concluding remarks

This work presents a practical implementation of a research data rights management solution using the blockchain as a mean of recording and verifying reuse. We have started from the observation that a considerable number of researchers prefer not to share the data behind a study in order to protect their work, and devised a new mechanism which allows greater flexibility in defining reuse terms and provides means to record and verify the execution of the terms.

As of future directions, thought needs to be given to the integration of the presented smart contract in the current workflow and tools. There are two sides of this; first, the blockchain network needs to be deployed and maintained, and here, the decentralized nature of the blockchain helps, as there is no need for an entity to act as the hosting party of the system (but nevertheless such an entity could take up the burden of maintaining the system from the individual researchers). Second, the system needs to record enough participation from all sides of the research workflow, including here the researches, publishers and repositories; for example, repositories need to include ways of executing with ease the functions of the smart contract (e.g., provide a facile web inteface for describing the custom terms of reuse), and publishers need to ensure that the last step of the contract (where the original author can verify if the terms were respected) is actually executed before a new work is published in an established medium.

Thus, we can see that there are both technical and social challenges in deploying such a solution, in an environment which can be rather slow to adopt new workflows. Nevertheless, we believe that our practical implementation of a smart contract for rights management can represent a solid base upon which infrastructure can be built in order to improve the behaviours around data sharing, which will naturally lead to improving replicability and reproducibility. Moreover, the approach can be extended to other research outputs which require means of establishing and enforcing reuse terms.

## References

American Psychological Association. 2015. *Publication Manual of the American Psychological Association* (6th ed.). Washington DC: American Psychological Association.

Brown, J. 2015. *Storing compressed text in Ethereum transaction logs.* Accessed October 7, 2017. https://web.archive.org/web/20171007025247/http://jonathanpatrick.me:80/blog/ethereum-compressed-text.

Creative Commons. 2018. *Creative Commons.* Accessed April 22. https://web.archive.org/web/20180422040418/https://creativecommons.org/.

Digital Science. 2017. *The State of Open Data 2017 – A selection of analyses and articles about open data, curated by Figshare.* doi:10.6084/m9.figshare.5481187.

Dryad. 2018. *Dryad Digital Repository.* Accessed April 8. https://web.archive.org/web/20180408203612/https://datadryad.org/.

Eklund, A., Thomas E. N., and H. Knutsson. 2016. *Cluster Failure: Why fMRI Inferences for Spatial Extent Have Inflated False-Positive Rates.* Proceedings of the National Academy of Sciences 113 (28). Proceedings of the National Academy of Sciences: 7900–7905. doi:10.1073/pnas.1602413113.

Elsevier. 2018a. *Open Access.* Accessed April 13. https://web.archive.org/web/20180413022235/https://www.elsevier.com/about/open-science/open-access.

Elsevier. 2018b. *Sharing Research Data.* Accessed February 3. https://web.archive.org/web/20180203084939/https://www.elsevier.com/authors/author-services/research-data.

The Ethereum Community. 2018. *Ethereum Homestead Documentation.* Accessed March 4. https://web.archive.org/web/20180304043559/http://ethdocs.org/en/latest/index.html.

Ethereum Foundation. 2018a. *Ethereum Blockchain App Project.* Accessed April 22. https://web.archive.org/web/20180422085428/https://www.ethereum.org/.

Ethereum Foundation. 2018b. *Solidity*, revision 9e109560. Accessed March 15. https://web.archive.org/web/20180315175715/https://solidity.readthedocs.io/en/develop/.

European Commission, Directorate-General for Research & Innovation. 2017. *H2020 Programme – Guidelines to the Rules on Open Access to Scientific Publications and Open Access to Research Data in Horizon 2020*, version 3.2. https://web.archive.org/web/20180414170704/http://ec.europa.eu/research/participants/data/ref/h2020/grants_manual/hi/oa_pilot/h2020-hi-oa-pilot-guide_en.pdf.

Federer, L. M., Lu Y.-L., Joubert, D. J., Welsh, J., and B. Brandys. 2015. *Biomedical Data Sharing and Reuse: Attitudes and Practices of Clinical and Scientific Research Staff.* Edited by Jyotshna Kanungo. PLOS ONE 10 (6). Public Library of Science (PLoS): e0129506. doi:10.1371/journal.pone.0129506.

Nakamoto, S. 2008. *Bitcoin: A Peer-to-Peer Electronic Cash System.* https://web.archive.org/web/20180417221636/https://bitcoin.org/bitcoin.pdf.

National Institutes of Health (NIH). 2018. *NIH Public Access Policy Details.* Accessed April 21. https://web.archive.org/web/20180421191423/https://publicaccess.nih.gov/policy.htm.

Phillips, N. 2017. *Online Software Spots Genetic Errors in Cancer Papers.* Nature 551 (7681). Springer Nature: 422–23. doi:10.1038/nature.2017.23003.

Ramachandran, A., and M. Kantarcioglu. 2017. *Using Blockchain and smart contracts for secure data provenance management*, https://arxiv.org/abs/1709.10000.

Springer Nature. 2018. *Open Choice.* Accessed April 3. https://web.archive.org/web/20180403161002/http://www.springer.com/gp/open-access/springer-open-choice.

Springer Nature. 2018. *Submitting to Research Data Support.* Accessed May, 6. https://web.archive.org/web/20180506125647/https://www.springernature.com/gp/authors/research-data-policy/submitting/15403694.

Stodden, V., Seiler, J., and Z. Ma. 2018. *An Empirical Analysis of Journal Policy Effectiveness for Computational Reproducibility.* Proceedings of the National Academy of Sciences 115 (11). Proceedings of the National Academy of Sciences: 2584–89. doi:10.1073/pnas.1708290115.

Szabo, N. 1997. *Formalizing and Securing Relationships on Public Networks.* First Monday 2 (9). University of Illinois Libraries. doi:10.5210/fm.v2i9.548.

Eynden, V. v. d., Knight G., Vlad, A., Radler B., Tenopir, C., Leon, D., Manista, F., Whitworth, J., and L. Corti. 2016. *Survey of Wellcome Researchers and Their Attitudes to Open Research.* Figshare. doi:10.6084/m9.figshare.4055448.v1.

Digital Science, and J. v. Rossum. 2017. *Blockchain for Research.* Figshare. doi:10.6084/m9.figshare.5607778.v1.

Wood, G. 2018. *Ethereum: A secure decentralised generalised transaction ledger*, EIP-150 Revision. Accessed April 14. https://web.archive.org/web/20180414173431/http://gavwood.com/Paper.pdf.

Xu, R., Zhang, L., Zhao, H., and Y. Peng. 2017. *Design of Network Media's Digital Rights Management Scheme Based on Blockchain Technology.* In 2017 IEEE 13th International Symposium on Autonomous Decentralized System (ISADS). IEEE. doi:10.1109/isads.2017.21.

Youngseek, K., and P. Zhang. 2015. *Understanding Data Sharing Behaviors of STEM Researchers: The Roles of Attitudes, Norms, and Data Repositories.* Library & Information Science Research 37 (3). Elsevier BV: 189–200. doi:10.1016/j.lisr.2015.04.006.

ZENODO. 2018a. *General Policies.* Accessed September 27, 2017. https://web.archive.org/web/20170927005046/http://about.zenodo.org:80/policies/.

ZENODO. 2018b. *ZENODO.* Accessed March 15. https://web.archive.org/web/20180315101712/https://zenodo.org/.