Scholarly publishing on the blockchain – from smart papers to smart informetrics

Michał Robert Hoffman [a,*], Luis-Daniel Ibáñez [b] and Elena Simperl [c]

[a] *Web and Internet Science Lab, School of Electronics and Computer Science, University of Southampton, Southampton, S017 1BJ, United Kingdom*
E-mail: M.R.Hoffman@soton.ac.uk; ORCID: https://orcid.org/0000-0001-9361-2468
[b] *Web and Internet Science Lab, School of Electronics and Computer Science, University of Southampton, Southampton, S017 1BJ, United Kingdom*
ORCID: https://orcid.org/0000-0001-6993-0001
[c] *Web and Internet Science Lab, School of Electronics and Computer Science, University of Southampton, Southampton, S017 1BJ, United Kingdom*
ORCID: https://orcid.org/0000-0003-1722-947X

Abstract. Blockchains provide decentralised, tamper-free registries of transactions among partners that may not trust each other. For the scientific community, blockchain smart contracts have been proposed to decentralise and make more transparent multiple aspects of scholarly communications. We show how an Ethereum-based suite of smart contracts running on top of a Web-enabled governance framework can facilitate decentralised computation of citations that is trustworthy. We implement and evaluate Smart Papers, and extend it with a model for decentralised citation counts. We show how our approach complements current models for decentralised publishing and informetrics calculation, and analyse cost and performance implications.

Keywords: Scholarly communications, smart contract governance, citation analysis, decentralised computation

1. Introduction

This is an extended version of our original "Smart Papers" article [22] in which we argued that more focus needs to be placed on avoiding the excessive concentration of power in the hands of big academic publishers. With the advent of digitisation and Web technologies, dissemination of scientific research objects has become faster and less expensive. However, several authors (e.g. [11,21]) pointed out that Web-based scholarly authoring tools appear to be "stuck in the past" to the extent that they are currently

---

*Corresponding author. E-mail: M.R.Hoffman@soton.ac.uk.

mimicking the print-based format dating back to the 15th century. In this work, we first remind the reader of the original motivation for Smart Papers – free, governable, transparent and tamper-proof records of research – only to extend our approach with a new governance framework and a new calculation of citations on a blockchain.

With a view to transforming research objects from static to *evolutionary* entities, conceptual models like Liquid Publications [11] and Dynamic Publication Formats [21] have been proposed. In such models, authors collaborate on a *living* version of the research object that, upon the authors' agreement, has periodical snapshots or *releases* published on the Web. Releases can be open for comments and reviews from the members of the public, or submitted to Calls for Contributions of conferences or journals. Authoring tools like Dokie.li [10] go one step further and provide *decentralised* implementations of living research objects that allow authors to retain the ownership of, and sovereignty over, their data. This supplies an alternative to the current state of play, where scholarly publication processes are centralised in publishing houses and large technology providers. However, an under-explored aspect in these models is how to manage the interactions between authors and contributors to a research object in a trusted way, which is of utmost importance for computing bibliometrics transparently. Examples of these interactions are (i) Agreement between authors on which snapshot of a working version should be released, (ii) Agreement between authors on the attribution due to each of them for each release of a living research object, (iii) Public comments and reviews of public releases, both as a mean to complement bibliometrics, an often-overlooked yet crucial labour in academia.

From the point of view of a single scholar that co-authors several papers with different teams, receives reviews and comments from peers, and reviews and comments on research made by others, data produced by these interactions, used to measure their performance, is not only controlled by her, or a single third party, but also by many other scholars (or their trustees). Any accidental or malicious change in a data store that is out of her control might have catastrophic impact on her performance measures.

Our work advances several cross-disciplinary research areas, including trust management for the Semantic Web, transparent informetrics, governance approaches for the blockchain and decentralised scholarly communications. By proposing a system that uses distributed ledgers and smart contracts to manage trust in a scenario which has been long understood as a critical showcase of semantic technologies, we provide a timely contribution to an ongoing discourse on the role and future of the Web as a re-decentralised platform for progress and social good.

We aim to answer two research questions in the context of open decentralised publishing systems: RQ1. *How can releases and their attribution agreements be managed in a trusted way?*; and RQ2. *How can we avoid malicious/accidental modifications in remote data stores affecting the computation of informetrics?*

Recently, distributed ledger technologies, commonly known as *blockchains* [34], have emerged as a novel tool that provides a decentralised solution to the problem of managing transactions of digital assets among parties that do not necessarily trust each other, while guaranteeing the immutability and verifiability of records. The record-keeping capabilities of blockchains have been augmented with user-defined distributed software that can be used to specify rules governing transactions, a concept known as *smart contracts*. Smart contracts offer guarantees of *security*, *tamper-resistance* and *absence of central control*.

In our original paper [22], we introduced a system called *Smart Papers* to manage the attributions and annotations of scholar publications, filling the gap of existing open decentralised publishing models. Some very interesting things happened in the landscape of scholarly communications since we published our original manuscript. At the time of writing the present work, there is a shake-up happening at

the Elsevier-owned Journal of Informetrics [12]. The original editorial board has left the big publisher, justifying their move by the need to make bibliographic references freely available for analysis, and by an aspiration towards being truly Open Access and community-owned. The journal's board members wanted Elsevier to lower its article-publishing charges and to take part in the Initiative for Open Citations, so as to free up citation data for research. When Elsevier declined to meet the requests, they launched a free-to-read journal called Quantitative Science Studies (QSS) published by MIT Press.[1]

1.1. Contributions of the extended version

With such seismic changes taking place in the sphere of informetrics, we decided to give this area the attention that it is due. First of all, the struggle for open metrics fundamentally requires a transparent approach to how references are stored and citations calculated – something that did not exist before. We have, therefore, extended our implementation and analysis with a tamper-proof approach to referencing and counting citations by linking Smart Papers together and updating the relevant counts using smart contract events (that we explain later). However, as the scope of the original project grew, so did the need for flexible yet reliable governance. Towards this end, we re-work our original implementation atop the Aragon governance framework that runs on the Ethereum blockchain, which we explain in Section 4.

In our extended approach, an Aragon App, comprising of several smart contracts closely tied to secure Web (JavaScript) code, is deployed on top of the Ethereum platform, and reusability is achieved by an unbounded number of research objects calling those contracts, and storing publication metadata in a distributed ledger. The smart contracts take the place of a trusted third party in keeping records, with the critical difference being that of data and execution not being controlled by a single entity, but rather inheriting all the guarantees of the host blockchain platform. The Aragon platform allows the collaborating parties to maintain a high level of governability over their projects.

In our Motivating Examples (Section 2), we highlight some of the most critical problems with current models – issues that directly affect the quality and trustworthiness of the existing approaches. In Section 3, we survey the existing models and implementations concerned with scientific authoring. We subsequently propose the Smart Papers model and its implementation in Ethereum on top of Aragon, paying particular attention to the issues of trust, identity, and platform technological considerations. Our discussion then progresses to cost analysis, after which our conclusions and future work recommendations are presented, in the final section of this paper.

2. Motivating examples

We use examples of Alice and Bob – we imagine that they both represent "typical" scholars from two separate institutions, who agree to work together on a new publication. The purpose of our motivating examples is to give the reader an idea of what sort of problems those scholars might currently encounter within the constraints of traditional ways of organising research and disseminating findings.

Bob and Alice begin working by employing their collaborative authoring tool of choice to start a draft version of their paper. After a few weeks of writing, they decide to release a public version in order to start receiving open comments and reviews. Charlie is a scholar from a third institution that finds Bob and Alice's release through an aggregator website or a search engine. He reads the article and leaves

---

[1]https://www.mitpressjournals.org/loi/qss

some comments on it that are stored in his personal data store and linked to the release, for instance, using the Web Annotation ontology.[2]

Alice and Bob then integrate Charlie's comments in their working version. They continue their work and eventually publish a second release. This time, they submit it to the Call for Contributions of a conference that uses open reviewing. Diane is one of the assigned reviewers. Her review is linked to the release which she read, as stored in the conference's data store.

When it is finally time for Bob, Diane and Charlie's appraisal meeting, their employers ask them for the dynamic publications that they have been involved in. Bob shows the full sequence of releases of the publication, while Charlie shows the comment he made on Bob and Alice's paper, and Diane shows the review she made for the conference. Employers apply their preferred credit models to assign weights to each type of attribution described in the attribution metadata, and quantify their values.

However, when reputation, credit, and ultimately, jobs are involved, social interactions can lead to problematic outcomes, with people trying to game the system in their favour, or to disfavour others. Below, we outline some examples of when things can go awry:

2.1. Example 1

Alice trusts Bob for creating the releases and their attribution metadata, as Bob controls the data store. However, Bob can publish a release with the metadata giving more attribution to himself. In a decentralised authoring tool like Dokie.li, each author would hold a copy of the working version, and they could independently generate the release, but if the attribution metadata differs between them, who solves this disagreement? How does an external agent know which copy to trust?

2.2. Example 2

Bob and Alice could collude to show different versions of the attribution metadata. For example, consider that employers use two different services to query dynamic publications linked to their faculty members. It is not hard to imagine a semantic store that returns a different version of the attribution metadata, depending on which agent is asking.

2.3. Example 3

Bob and Alice could collude to ignore Charlie's comment, in an attempt to not share part of the credit with him. In a decentralised model, a link to the comment and Charlie's identity should be stored in Bob and Alice's data store; however, if Bob and Alice control the data store, nothing prevents them from deleting the link. Charlie would have the copy of the comment and the link to the release, but he might have a hard time convincing a third party (his employer for example), that the comment was not forged.

2.4. Example 4

If Diane's review is considered unfair, the persons in control of the data store might be tempted to make it disappear. A third party agent querying the conference's data store would see nothing. An agent following links from Bob and Alice's data store would get a de-referencing failure (404). Even if Bob and Alice decide not to delete the unfavourable review from their data store, how can they prove that they are not forging fake reviews to damage Diane's reputation?

---

[2]https://www.w3.org/TR/annotation-vocab/

2.5. Example 5

Bob and Alice's paper becomes so successful that hundreds of scholars start citing it in their work. However, different informetrics providers like Scopus and Google Scholar seem not to be able to agree on the number of citations for Bob and Alice's paper. One provider is missing a few high-impact conference proceedings papers quoting Bob and Alice's paper, whereas the other provider seems to have included lots of grey literature, resulting in two very different citation counts. How can third parties (most importantly, Bob and Alice's employers and funders) be sure of the true citation count of Bob and Alice's original work?

2.6. Example 6

Alice and Bob subsequently work on a paper with 4 more co-authors from other institutions (Edna, Fred, Gina and Hal) who also happen to be based in different time zones. The six scholars now need to add a complex figure to their paper. There are a few alternatives for plotting the figure, and each approach has its own unique pros and cons. How can we carry out a vote to enable the whole team to reach a single decision?

2.7. Example 7

Alice, Bob, Edna, Fred, Gina and Hal now have to decide whether to publish the current version of the paper. Edna believes it is not camera-ready and submits corrections, whereas Hal pushes for immediate submission without corrections. How can the resolution of this conflict be facilitated in a transparent manner?

2.8. Towards a proposed solution

The common problem of these scenarios is that for all actors (Alice, Bob, Charlie, Diane and their employers, funders and contributors), the data that is crucial to show or measure their performance is not entirely under their control, making it vulnerable to manipulation.

Our approach addresses this problem by empowering all collaborators with the following:

- The notarisation of releases providing evidence that all the authors agreed to the release of a particular version of their paper.
- The notarisation of the attribution metadata linked to a release, ensuring that all authors have agreed on it, and guaranteeing to third parties that none of them can tamper with it.
- A mechanism that ensures that annotations made on releases by agents other than authors cannot be repudiated by annotators or their recipients, guaranteeing to both authors and third parties querying this data, that it was not tampered with.
- An index of links and data concerning a particular dynamic publication. This potentially facilitates the task of Web agents that compute bibliometrics, as there is no need to either trust the data store of the authors, or to crawl the Web in search of the comments and reviews to the publication.
- A mechanism for linking papers that reference each other.
- Transparent and trustworthy citation counts produced in a decentralised environment.
- A voting mechanism for facilitating the resolution of governance conflicts using pre-configurable quora.

• The notarisation of voting outcomes on the blockchain.

It is worth noting that through implementing the above requirements, our platform partially automates the fulfilment the Vancouver (ICMJE) guidelines on Authorship.[3] We will review the related work before explaining how we modelled and implemented an approach that fulfils the above requirements.

3. Related work

3.1. Dynamic publication models

Several models have been proposed to take advantage of digital and Web tools to improve the ways in which academic publications are produced and managed. Liquid Publications [11] proposes evolutionary, collaborative, and composable scientific contributions, based on a parallel between scientific knowledge artefacts and software artefacts, and leveraging lessons learned in collaborative software development. Their model is based on the interaction between Social Knowledge Objects, *i.e.*, digital counterparts of the traditional paper unit; people and roles, *i.e.*, agents involved in the scientific knowledge processes, playing various cooperating and competing roles (from traditional ones, like author, reviewer or publisher, to new ones derived from the model itself, like classifiers, quality certifiers, credit certifiers); and processes to manage its lifecycle, namely: authoring collaboration, access control, IPR and legal aspects, quality control and credit attribution and computation.

The Living Document model [18] aims at creating documents that "live" on the Web by allowing them to interact with other papers and resources. It lets authors build social networks, with their interactions defined through the papers they write.

Heller et al. [21] propose Dynamic Format Publications, where working versions are collaboratively edited by a small group of authors, that decide when a version or revision become widely available, following a formalised gate-keeping mechanism (e.g., consent among authors and/or peer-review). Only the Living Document approach provided a prototype (inactive at this time), and none of them discusses the security and trust implications of their models. Our work provides a foundation that can be used to track and manage credit attribution (and by extension, IPR and legal aspects) that can be easily plugged into a broader authoring model.

3.2. Decentralisation

Concerning the decentralisation of scholarly communications, Dokie.li [10] is a fully decentralised, browser-based authoring and annotation platform with built-in support for social interactions, through which people retain the ownership of and sovereignty over their data. Dokie.li implements most of the functionalities described in the previously described conceptual models in a decoupled way. In a nutshell, a Dokie.li document is an HTML5 document enriched with RDFa, which is stored in the author's personal data store. The Linked Data Platform (LDP) protocol implementation enables the creation, update and deletion of documents. Interactions with documents are registered using the Web Annotation Vocabulary.[4]

Documents are connected statically through links and dynamically through Linked Data Notifications [9], proving the viability of a decentralised authoring and annotation environment built according to Web

---

[3] http://www.icmje.org/icmje-recommendations.pdf
[4] https://www.w3.org/TR/annotation-vocab/

standards. Authors consider that in a fully decentralised setting, each source is filterless and responsible for its own quality and reputation, whilst everyone is free to selectively distrust certain sources using any mechanism they desire. We argue that, although this assumption holds for trust in the *content* of the research object, stronger measures are needed for social interaction data on research objects that could be used to compute bibliometrics. Our approach also aims at solving some security issues that arise in decentralised environments, notably, the possibility of malicious deleting or updating of records to impact bibliometrics [27].

With respect to the application of blockchains for scholar processes, the Blockchain for Science association maintains a living document [6] that collects and proposes applications, use cases, visions and ventures that use blockchains for science and knowledge creation, providing an index of the potential impact of Distributed Ledger Technologies in all stages of the research workflow. For the particular case of publishing and archiving, timestamping and credit attribution of Dynamic Publications is mentioned as a promising use case. To the best of our knowledge, the open-source system that comes closest to ours is Manubot,[5] a tool for writing scholarly manuscripts via GitHub. Manubot automates citations and references, versions manuscripts using git, and enables collaborative writing via GitHub. Data from Git related to commitment and authorship can be used to establish attribution. An innovation introduced by Manubot's authors [6] is the timestamping of manuscript versions on the Bitcoin blockchain, to prove the existence of the manuscript at a given point of time in a decentralised way. Our approach generalises Manubot's idea to further social interactions around publications.

Concerning the permanence and immutability of Web artefacts, Trusty URIs [25] propose to append to URIs the cryptographic hash of the Web artefact they represent, enabling the verification to contain the content the URI is supposed to represent. Trusty URIs are immutable in the sense that any change in an artefact would change its URI as well, and permanent, under the assumption that Web archives and search engines that crawl them are permanent.

Our approach implements functionality analogous to Trusty URIs, but also solves the further problem of conflicting metadata: if each author could publish metadata on the attribution about the research object, each one with its own Trusty URI, and both can be verified to not have been tampered with, then which one should an external agent use? To answer this question, we describe how our Smart Papers model provides a single version of the truth.

3.3. Informetrics and citation analysis

Citation analysis belongs to the field of scientometrics (the study of the quantitative aspects of the process of science as a communication system [29]), which itself is part of informetrics (a wider study of how information moves through social systems, not just science-oriented ones [5]). Davenport and Cronin [13] view citations as "links" between papers that form tokens representing *trust* in that when Alice cites Bob, Alice assumes that Bob's claims are *trustworthy* enough to support her claims. In this general case, citation statistics may be used in recommender systems to prioritise highly cited items. However, many authors take issue with this simplistic "positive" view of citations, instead pointing out the relevance of various "meanings", "contexts" and "sentiments" of citations [2]. Hence, under a more nuanced model, we would have to pay attention to the polarity of the citation, either positive, representing trust, or negative, representing mistrust in someone else's findings. Even more complex NLP-driven scientometric citation analysis methods have been proposed recently [24] but they remain out of scope

---

[5] https://github.com/greenelab/manubot-rootstock
[6] https://github.com/greenelab/deep-review/pull/274

of this paper. Also out of scope is the discussion of the modern standards by which academics are judged and evaluated in their careers, and the widespread criticism of basing complex grant and hiring decisions on rather simplistic metrics like citation counts [26]. For the curious reader, a good overview of the limitations of scientometrics is provided in [1].

Coming from a different angle, citations can be explained as symbolic payments of intellectual debt [31], making citation analysis a useful tool for sociological analysis into how credit (or discredit) flows first to the publication, and then to the associated authors, institutions, countries, and journals that are its attributes. Despite the perceived great significance of citation counts as a scientometric resource, alarmingly, currently available citation counts are not free from miscalculation [3]. For example, one of the biggest platforms nowadays to provide citation counts for scientific articles, Google Scholar, contains errors that are reported in literature [20]. The authors of this paper are even able to confirm that the Google Scholar count for their own original paper is incorrectly reported (one source has been counted twice). Similar significant informetrics errors like phantom citation counts, missing citations and duplicate references have also been reported for other scholarly data providers such as Scopus [16].

Our approach enables the author who is referencing someone else's work to register this reference on the blockchain, as a link between two Smart Papers. Citation counts are updated automatically due to the properties of the underlying blockchain and the smart contract code that encapsulates the counters. A Smart Paper can be easily extended to store and register other informetrics that can be represented as monotonic counters, e.g., a citation sentiment, or polarity (negative or positive). We describe the Smart Papers model in the next section.

4. The smart papers model

The Motivating Example (Section 2) illustrated the importance of trust management throughout the collaborative process. When reviewing related work (Section 3), we highlighted a strong need for making agreements and setting their outcomes in stone so that they cannot be later repudiated. Furthermore, all the essential artefacts associated with those agreements must be timestamped and securely stored in a truly permanent way. Currently available collaborative tools solve some trust issues, for example Dokie.li removes centralisation so that the authoring parties do not have to rely on an intermediary to publish and annotate their documents. This is a very welcome step towards removing the overhead associated with middleman activities (publishing house), albeit it merely shifts the trust towards the authoring parties (author, reviewer). It is easy to imagine a situation in which the authors destroy their data, the reviewers could do the same, and any track of their writing will be lost forever. We also made a case for decentralised informetrics computation to achieve a single version of the truth in citation counts.

The purpose of our model is to provide trust where it has not existed before. Smart Papers provide a collaborative platform that preserves a single version of the truth throughout the collaborative process. This is somehow similar to employing a trusted third party (e.g. a notary public) to keep track of contracts signed by multiple parties, alongside with all the certified photocopies of all the evidence attached to the contracts as relevant appendices. An example of such notarised contract would be Alice and Bob signing an agreement specifying the ordering of their names on a paper (e.g. "Bob, Alice") and then attaching a certified photocopy of their paper in its current version as an appendix. We use smart contracts for maintaining all such signed agreements in order to implement Smart Papers. Table 1 summarises how smart contracts can provide the functionality analogous to that of a traditional trusted third party.

Table 1

Blockchain smart contracts as compared to a traditional trusted third party

Notary public function	Blockchain function
Authenticate parties using their legal identification	Identify parties cryptographically
Take statutory declarations, store them and certify photocopies	Store data permanently and securely and provide real time access
Prepare and certify contractual instruments	Store and execute smart contracts
Provide a trusted record for the above	Provide a trusted record for the above
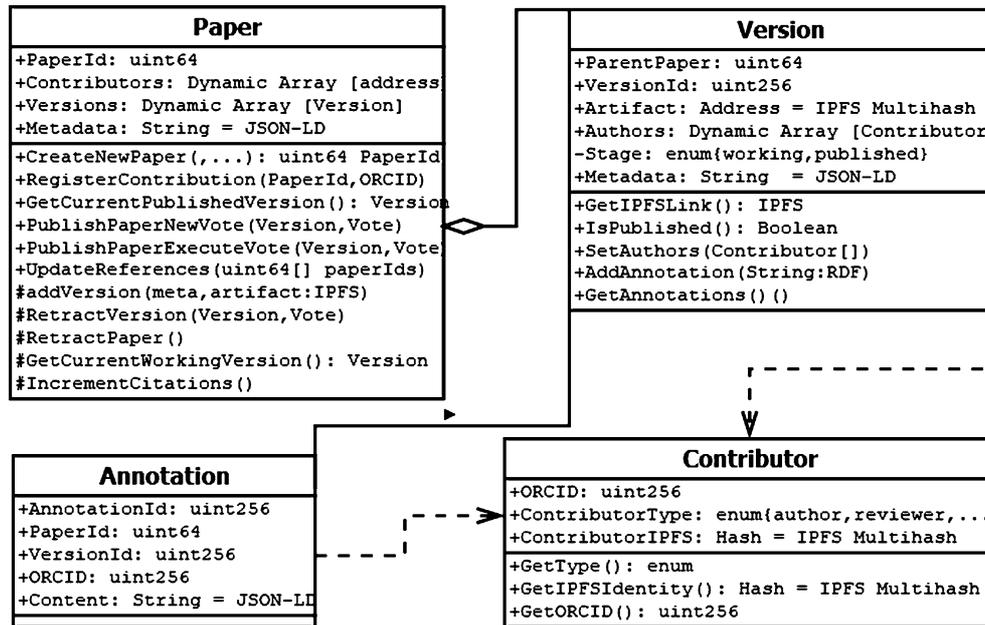
Fig. 1. The smart papers distributed application design.

4.1. Design

To design the Smart Papers model, we shall assume that all authors successfully identify through their ORCID (Open Researcher and Contributor ID [19]) which is a non-proprietary alphanumeric code to uniquely identify scientific and other academic authors and contributors. ORCIDs are mapped to authors' signing and encryption keys using a smart contract. The main functionality for our model is then designed using the separation of concerns (SoC) design principle [23], such that each contract file addresses a different concern, i.e. a different set of information that jointly affects the global state for the Smart Papers use case. We use UML to model the main classes corresponding to our smart contracts. It is important to note that smart contracts and OOP classes (as modelled by the UML) are not quite the same. The semantics are very similar in many cases, but some fundamental differences arise from the fact that smart contracts can store and send value and have a public address once deployed [35]. The UML diagram in Fig. 1 shows how we group these concerns into the following four categories: Paper, Version, Annotation and Contributor.
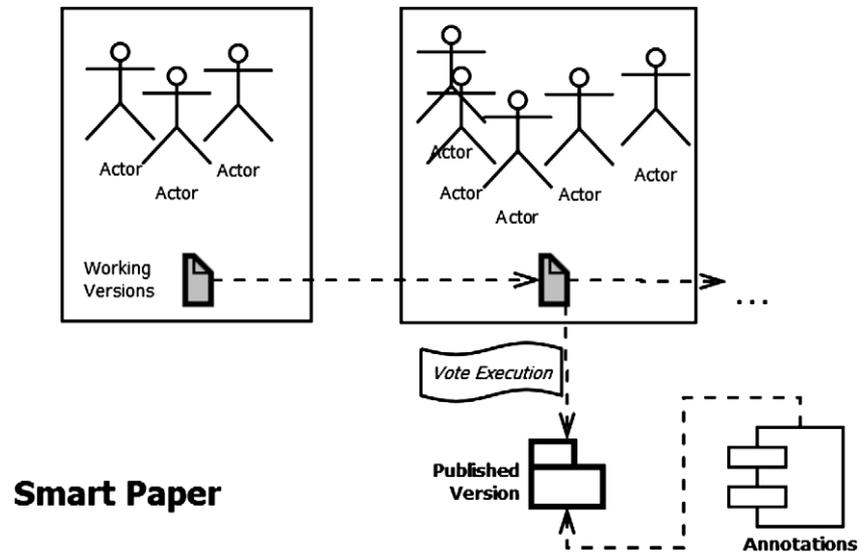
Fig. 2. The workflow of a smart paper involves multiple working versions with dynamic collaborators. Versions can become published and made available for annotating.

To begin with, an article and its metadata (e.g., attribution encoded in the ScoRO ontology[7]) is submitted by a writer (we shall refer to her as Alice, from our motivating example earlier), and stored in a distributed file store, all of which is recorded on the blockchain. Alice will have been set up in the system through the Aragon framework that we will explain in the next subsection. In our implementation, the *Contributor* contract requires Alice to have a valid ORCID. The default type for Alice is "author". Bob is also set up as an "author", but Diane uses a different argument for the *Contributor* contract, and so she becomes registered as a "reviewer".

Smart contracts often act as state machines, meaning that they have certain stages making them act differently, and in which different functions can be invoked. A function invocation often transitions the contract into the next stage which can be used to model workflows. We use this feature of smart contracts to model the Smart Paper workflow, as seen in Fig. 2, which allows the participants to release new versions of their paper and to publish versions when enough authors agree to do so.

Papers can also be retracted. As illustrated in Fig. 2, once instantiated, a Smart Paper becomes a dynamic list of versions, each of which can exist in a working state or become published. The number of contributors and their formal ordering is allowed to change on a per-version basis. Annotations can be left by reviewers on published versions.

To create a new Smart Paper, either Alice or Bob call *createNewPaper* in the *Paper* Contract which will return a valid *PaperId* that uniquely identifies their new publication. This also instantiates the workflow with an initial, blank, working version of this paper manufactured by the *Version* contract. Bob and Alice work on their preferred authoring tool to produce a first draft (e.g., to show to a trusted colleague), to register it in the Smart Paper, Bob calls *addNewVersion* in the Version contract, including the artefact, its metadata and his signature. Before committing the transaction, the Smart Paper will wait for Alice (marked as a contributor to the paper) to also perform a call to *addNewVersion* using the same artefact, metadata and her signature.

---

[7]http://www.sparontologies.net/ontologies/scoro

The procedure is repeated each time Bob and Alice want to register a new version. For marking a version as public, Bob calls *publishVersion* in the *Paper* contract, providing the *versionID* and his signature. Similar to *addNewVersion*, Alice needs to issue her signature through a function call to *publishVersion* before the Smart Paper commits the transaction. The *getCurrentPublishedVersion* and *getCurrentWorkingVersion* return a versionID that can then become the input to the *getIPFSLink*. Up to this point, we have provided a solution for the issues between authors described in Example 1, the Smart Paper only commits a version (including metadata) if all authors sign their agreement to it. An external agent that gets a version from a Smart Paper instance has the assurance that it was approved by all authors, and that the Smart Contract consistently returns the correct version and metadata, solving the issue described in Example 2.

Interactions with external actors like reviewers or annotators, are abstracted as *Annotations*. When Charlie or Diane want to leave their comment or review, they call *addAnnotation* using the *versionID* of the version they want to comment on, and their signatures. Contrary to the *Version* functions, no approval from authors is needed. The annotation is registered in the main Ethereum blockchain and can be retrieved by calling *getAnnotation*. Looking back at Example 3, Charlie can now point to the Smart Paper to show that he made that comment. For the case of Example 4, the Smart Paper holds a register of the reviews. Alice and Bob can now prove that the annotation held by the Smart Paper was signed by Diane.

4.2. Bibliometrics

To solve our Motivating Example 5 (in which citation inconsistencies were present across different centralised providers), we can entice scholars to store references as links between Smart Papers using our Smart Papers web interface. To store a reference, a living Smart Paper (A) can, at any time, register a citation of another Smart Paper (B) through calling the registerCitation(B-paperId) function on A's Paper contract. Whilst the Smart Paper A is in its working version, and unpublished, this only updates the references internally. Nevertheless, publishing the Paper A will trigger an event PaperPublished which, in turn, triggers the CitationUpdate functionality. PaperB's incrementCitationCount will be called at that point. This chain of events will result in Paper B's citation count being updated with the new reference. However, if at any point A is retracted, the PaperRetracted event deals with calling previously cited papers to decrement their counts.

4.3. Votes

For those Smart Papers use cases that require a quorum to reach a decision, we replaced the low-level multi-signature approach of our original implementation, with a new Vote-based approach in the extended version. Most importantly, once more than 1 author has been added to a Smart Paper instance, a Vote now needs to be carried out before the paper can be published.

Votes are also carried out for other common authoring scenarios that require a decision-making protocol, such as:

- adding a new contributing author to an existing Smart Paper instance,
- approving a new significant revision to the existing Smart Paper,
- choosing the name of a paper – an example of how Votes can be used for brainstorming as well as signing approval,
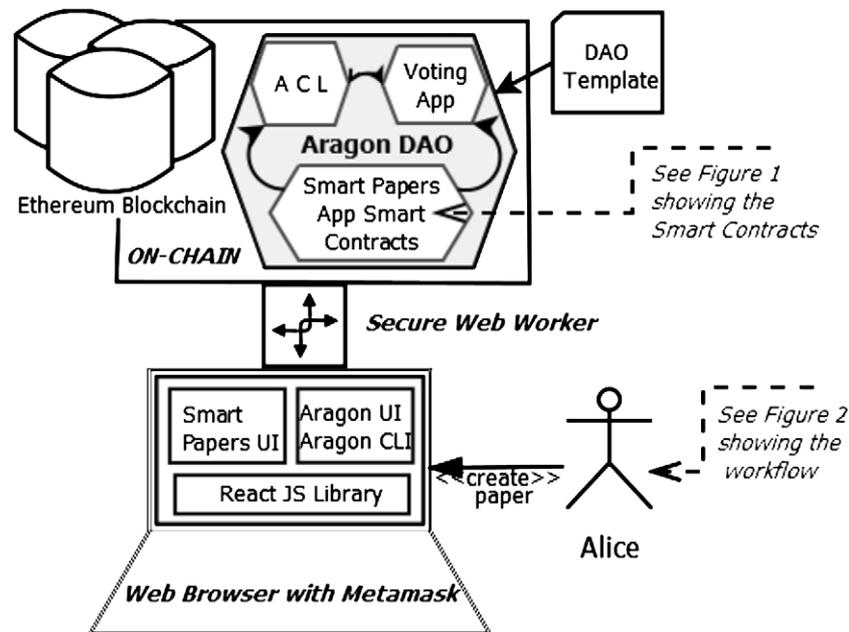- voting on the listing order of authors' names.

Fig. 3. The architecture of the smart papers implementation involves frameworks and features that work on-chain, as well as those that run locally in the browser. The secure web worker mediates communications.

The quorum for most of our use cases is simple democracy (more than 50% voters agree with a particular outcome). However, the quorum requirement should be set to 100% for publishing a final version of the paper. Submitting papers without the knowledge and permission of co-authors appears to be a worrying trend that is contrary to professional standards.[8] Taking governance to a meta-level, the beauty of using a framework like Aragon for managing governance is that we can even schedule a Vote to decide about what the thresholds (quora) should be for all future Votes.

4.4. Implementation

Although in theory, the Smart Papers model could be implemented on any smart contract-enabled platform, the choice of the implementation framework dramatically impacts development time and costs. Whilst there are multiple distributed ledger technologies, such as Corda[9] or HyperLedger,[10] that could be utilised to develop trusted smart contract code that runs on top of the blockchain, for this paper, we elect to develop on top of the Ethereum platform [35] which is the most commonly used technology of its kind [4]. We defer the feasibility and cost of development in other platforms for future analysis.

4.4.1. Background: Ethereum and IPFS
Ethereum is an open-source, public, blockchain-based distributed computing platform featuring smart contract functionality [35]. It plays the role of the trusted third party for all Smart Papers agreements in our model. Ethereum blockchain was designed to be deterministic. This means, that everyone should always end up with the same, correct state, if they try to replay the history of Ethereum transactions.

---

[8]https://www.elsevier.com/connect/co-authors-gone-bad-how-to-avoid-publishing-conflicts
[9]https://github.com/corda/corda
[10]https://www.hyperledger.org/

In Ethereum, the code execution layer is provided by the Ethereum Virtual Machine (EVM), a Turing complete 256bit VM that allows anyone to execute code that references and stores blockchain data in a trust-less environment. Ethereum smart contracts are written using a language called Solidity. Every contract on the Ethereum blockchain has its own storage which only it can write to; this is known as the contracts state and it can be seen as a flexible database albeit at a high cost. When deployed, Ethereum contracts get an *address*, that can be considered similar to a URI in Ethereum's namespace. Using this address, a client can call functions defined in a smart contract, in a similar fashion to a web service.

When implementing our model, we chose to store all the artefacts using IPFS [7]. The InterPlanetary File System is used for efficiently distributing and referencing hash-linked data in a way that is not centralised and does not necessarily involve blockchain transactions, thus avoiding the economic penalties associated with on-chain storage [14]. In many ways, IPFS is similar to the World Wide Web, but it could be also seen as a single BitTorrent swarm for exchanging objects. Furthermore, the IPFS specification contains a special *commit* object which represents a particular snapshot in the version history of a file. This allows us to reference resources in an immutable way, akin to Trusty URI functionality. Using IPFS we can, therefore, limit the role of Ethereum, so that it only deals with the application logic; the data layer is provided by the InterPlanetary (IPFS) stack, and the two layers are integrated via hash references.

We also use SafeMath, which is a Solidity math library specially designed to support safe mathematical operations, to decrement and increment citation counts on existing papers being referenced – changes that are triggered by publishing new papers referencing them.

4.4.2. Smart papers as a web app

One of the core requirements of the SmartPaper model is the ability to provide a tool for all collaborators to agree with the result of a certain interaction. Decision making can be implemented in different ways. In our implementation, the number of collaborators can be unbounded, but to make a decision, an agreement needs to be reached by a quorum of authors that is configurable on a per-paper basis. This means that you could have a Smart Paper, in which only one person's vote is enough to make a decision, but you could also have a Smart Paper where everyone is required to agree on a particular course of action. The default quorum for Smart Papers is 50%, which we call 'simple democracy'. As a Smart Paper evolves, its collaborators can change the quorum at a later time through a vote. Interacting with Smart Papers is done through a Web browser, which is why we call it a Web app.

4.4.3. Aragon

Aragon[11] is a Web project that helps Internet users collaborate remotely and freely organise without intermediaries, to transfer data and value, and to make joint decisions. When starting a new project on Aragon, a DAO (decentralised autonomous organisation, [28]) is created. Technically speaking, a DAO is a set of smart contracts that define a governance structure and rules via a web interface. Yet in an abstract sense, a DAO represents a group of people working together towards a shared goal. An Aragon DAO has an Ethereum account and a Web address associated with it (in our case, soton.aragonid.eth[12]).

4.4.4. Governance

Each DAO also has a creator account who initially has all the voting power. This voting power is expressed through 100% DAO token ownership. This is because DAOs have their own tokens, in our case it is the University of Southampton (SOT) token. New members joining the DAO are provided

---

[11]http://aragon.org
[12]https://rinkeby.aragon.org/#/soton.aragonid.eth/

tokens enabling their participation in the DAO through a process that requires existing members to vote on approving the newcomer.

Aragon DAOs support apps[13] that are custom bundles of smart contracts closely integrated to the Aragon platform via their bespoke secure Web interfaces. The Voting app is an example of an app that acts as an entity that will execute a set of actions on other entities (e.g. other apps, smart contracts, user's balances) if the token holders of a particular governance token (i.e., members of the DAO, like SOT holders for the Soton DAO) decide to do so. The Smart Papers app[14] is our suite of smart contracts that has been integrated into Aragon to support its front- and backend. Essentially, any Smart Papers transaction can now be signed by the user in their Web browser with a Metamask[15] extension, and all the interactions requiring multiple signatures can be signed individually by different users as part of a single Aragon vote. We show the overall architecture of this set-up in Fig. 3, where we also summarise how a new DAO can be deployed from a DAO template, which will install our custom Smart Papers app interacting with both the ACL and Voting components of Aragon.

Aragon DAOs are always deployed with ACLs (Access Control Lists). This is a computer security mechanism [30] whereby permissions are attached to users or processes, or in the context of blockchain, to user accounts. Within the realm of Aragon, a permission is defined as the ability for users to perform actions (that are grouped by roles) in a certain app, for example, the Voting app[16] or our custom Smart Papers app[14]. ACLs are grouped for a particular DAO. New members can be granted permissions in an ACL from the creator and then grant permissions to others (if permitted). The Access Control List remembers who can perform what actions where. Sometimes users are not allowed to perform actions directly, but only through a decision-making process, such as a vote. The ACL handles this use case through a concept called forwarders. A forwarder is an app that can execute actions on someone's behalf if the ACL permits it. An example of a forwarder is the Voting app, and we can specify an ACL entry allowing a user to carry out a certain operation if a Vote is carried out. The Voting app will only execute the user's desired function call if the vote passes.

One final advantage of using Aragon to manage our smart contracts is that it manages the upgrading of major, minor and patch versions of contracts in a way that is seamless (using a proxy pattern) and was unachievable in our original implementation – upgrading old-style Smart Papers would have required a completely new contract address – which now can be avoided with Aragon.

5. Evaluation

Two important factors affecting the performance of a distributed application like Smart Papers are firstly, how much it costs to run this app's operations like creating new papers and registering references, and secondly, how long these operations take to complete in the decentralised environment. We set out to evaluate both the costs and the speeds for four main use cases for Smart Papers:

(1) creating a new Smart Paper on the Ethereum blockchain from the Smart Papers app,
(2) adding a citation (a link between two Smart Papers) through the Smart Papers app,
(3) creating a new open vote using the Voting app, and
(4) casting a ballot / participating in an open vote in the Voting app.

---

[13]http://hack.aragon.org
[14]https://github.com/mikehoff/SmartPapersApp
[15]http://metamask.io
[16]https://wiki.aragon.org/dev/apps/voting/

5.1. Test set-up

To create a test environment, the reader will need a POSIX-compliant platform with NodeJS (at least version 8.0.0) and NPM packager installed. The NPM package "@aragon/cli" installs Aragon CLI, a tool for creating, testing and publishing Aragon applications. A test DAO identity called "soton" is already registered on the Rinkeby Ethereum testnet (https://rinkeby.aragon.org/#/soton.aragonid.eth/). A "soton" project can be initialised locally by typing in "aragon init soton" which will prepare the ground for local testing. The code for the Smart Papers app can be checked out from GitHub .[17] The app can then be compiled and launched with "npm run start:app", whereas a local Aragon environment can be launched through "npm run start:aragon:http" and subsequently navigated through the web browser on localhost port 3000. This will start a development chain you can interact with (ganache-core[18] is used, so it is a full testrpc[19] instance – testrpc was used to evaluate the original Smart Papers implementation).

5.2. Notes on cost

Execution of a smart contract begins with a transaction that is sent to the blockchain. This transaction specifies the address for the contract, the arguments, and an amount of Ethereum's currency to pay for the execution. It is commonly observed in small-to-medium size contracts that most of the cost is taken up by a fixed "base fee". This base fee of 21,000 is expressed in "gas" which is an abstract unit. While gas is fixed per each transaction, it is additionally fixed for every operation called from within the smart contract. Each low-level operation available in the EVM is called an OPCODE. These include operations such as *ADD* – adding two integers together, *BALANCE* – getting the balance of an account, and *CREATE* – creating a new contract with supplied code. Each of these OPCODEs has a fixed amount of gas that it costs to execute. The fixed amount of gas has been chosen by the designers of Ethereum for each OPCODE in a way that reflects the relative complexity of that OPCODE.

Whereas gas is fixed and predictable, the amount a user pays per gas, the *gas price*, is dynamic and dictated by market conditions. The price is usually given in subunits of *ether*, such as *gwei* (1 Gwei = 0.000000001 Ether). Miners receive ether fees based on the amount of gas multiplied by the gas price, which incentivises them to prioritise those transactions that attract higher fees. It also follows that the higher gas price you are willing to pay, the faster your transaction will be processed, and the sooner your contract will be allowed to execute. While offering a high gas price can speed things up, there is a limit to the acceleration.

5.3. Data collection and replication

We estimate the gas prices with ETH Gas Station [15], the de-facto reference for understanding the current gas market conditions and miners' current policies. The *Recommended User Gas Prices* section of ETH Gas Station shows the range of gas prices one might pay for an expected transaction commitment time. We compile and graph the gas price predictions from the time of writing this article in Fig. 4. This data has been collected using the ETH Gas Station Aplication Programming Interface.[20] Data collec-
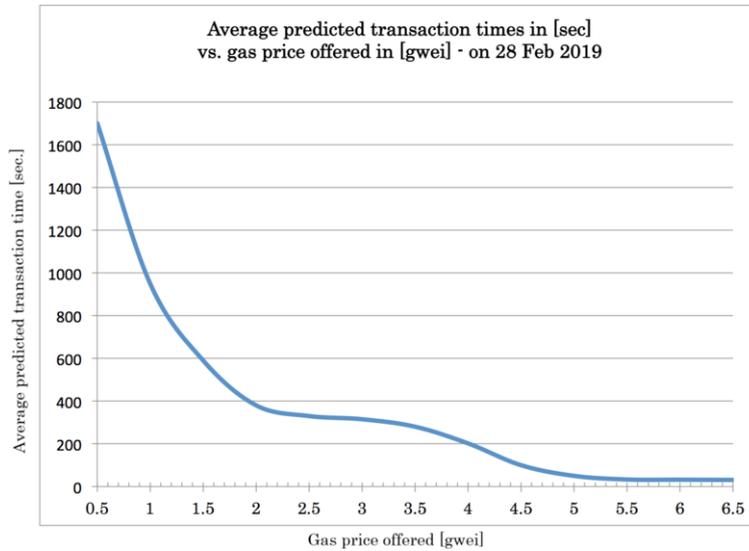
---

[17]https://github.com/mikehoff/SmartPapersApp
[18]https://github.com/trufflesuite/ganache-core
[19]https://github.com/trufflesuite/ganache-cli
[20]https://docs.ethgasstation.info/

Fig. 4. The influence of gas price on execution times.

Use Case / Gas Cost	Fee Grade	Transaction cost	Speed	Transaction time
Adding a Citation / Cost: 104,438 [gas]	Low (3 gwei)	$0.04	Slow	5 min 16 sec
	Average (4 gwei)	$0.06	Average	3 min 13 sec
	High (5 gwei)	$0.07	Fast	31 sec
Participating in an Open Vote / Cost: 146,151 [gas]	Low (3 gwei)	$0.06	Slow	5 min 15 sec
	Average (4 gwei)	$0.08	Average	3 min 13 sec
	High (5 gwei)	$0.10	Fast	31 sec
Creating a New Paper / Cost: 204,966 [gas]	Low (3 gwei)	$0.08	Slow	5 min 15 sec
	Average (4 gwei)	$0.11	Average	3 min 13 sec
	High (5 gwei)	$0.14	Fast	31 sec
Creating a New Vote / Cost: 315,818 [gas]	Low (3 gwei)	$0.13	Slow	5 min 30 sec
	Average (4 gwei)	$0.17	Average	3 min 22 sec
	High (5 gwei)	$0.22	Fast	32 sec
Adding a New Version to Paper / Cost: 475,281 [gas]	Low (3 gwei)	$0.19	Slow	5 min 30 sec
	Average (4 gwei)	$0.24	Average	3 min 22 sec
	High (5 gwei)	$0.32	Fast	32 sec

Fig. 5. The results for our main use cases.

tion can also be performed manually using Metamask extension and the relevant steps are documented online.[21]

5.4. Results and discussion

Our results for the evaluation of Smart Papers are presented in Fig. 5 and sorted by average transaction cost, with adding citations being the cheapest (costing as low as $0.04) and creating a new open vote being the most expensive (costing up to $0.32). The more expensive operations are those that use

---

[21]https://github.com/MetaMask/metamask-extension/issues/3037

Aragon's Voting app, due to the complexity of setting up and executing Votes. It is worth noting that the absolute costs are as much as 57-fold smaller than in our original implementation. This can be attributed to Ether (the Ethereum currency) now being much cheaper than at the time of printing the previous version. There have also been bytecode optimisations to the Solidity compiler yielding faster and cheaper code.

Based on Web of Science data, Marx and Bornmann estimated that, in 2010, the average number of references in a scholarly paper was 32.21, with a low average of 27 for Engineering papers, and a high average of 42 for Social Sciences papers [8]. Assuming Small's view that references represent intellectual debt [31], this debt can be symbolically paid off by the co-authors of a new Smart Paper through covering the fees associated with registering the citations resulting from their newly published work. Therefore, for the most costly case of a new Smart Paper created in the Social Sciences domain, paying for all the citation counts that were changed by references from that Smart Paper, would cost 42 (average reference count) times $0.04 (transaction cost for registering a citation), which would cost an average of $1.68 per paper.

In terms of our Motivating Examples, we have seen that the notarisation of scholarly papers and their associated metadata on a blockchain, using Smart Papers, can solve the problems presented in Examples 1–4. We have also shown how the governance benefits of Aragon open votes help us solve the problems described in Examples 6–7. Finally the citation-counting functionality of Smart Papers resolves the dilemmas presented in Example 5.

Regarding cost, our result suggest that a new Smart Paper can be created for as little as $0.08. A vote with three participants can be carried out for as little as $0.31 ($0.13 to create a vote and $3 * 0.06 for each ballot cast). We already estimated the cost of storing references as $1.68 (the costly case). Adding all these costs together yields a paper that, throughout its total life cycle, costs around two dollars in total to govern, update and disseminate. These costs are in stark contrast to publishing houses' costs that have been previously estimated to be in the range of $3,500–$4,000 per article [32]. When comparing these dollar figures, the reader has to be mindful of what exactly is being compared. For example, services like typesetting and printing would not be provided by Smart Papers, even though they may be still included by traditional publishing house in their cost estimates. The discussion of whether traditional print services are needed to successfully disseminate a paper in today's digital era is beyond the scope of this article, and we assume a paper to be successfully published after it's made its way to the blockchain. However, the additional costs of IPFS file storage are a completely valid concern. IPFS has no concept of transaction or storage costs, but also lacks an incentive layer. So currently, a researcher may elect to run a local IPFS node, the cost of which could vary wildly depending on the machine specifications and the local networking fees. The real costs may be inferred from those quoted by cloud IPFS providers (around $0.15/GB/mo.[22]). All in all, in the era of decentralised Web, Smart Papers offers a cost-effective alternative to the services being offered by big publishers.

It is worth noting that for most purposes, the Low Cost options should be used for all use cases, as the slowest observed transaction time in that instance is 5 minutes 30 seconds which we believe is still acceptable for most scholarly communications use cases, especially considering that it would normally take weeks, if not months, to get a scholarly paper successfully disseminated through a publishing house.

In the Smart Papers approach, a valid question arises about who should pay for the transaction fees that we estimated. Should it be the scholars, their universities, funders, etc? Should the payment mechanism be different for the different use cases – for example, should adding references be funded in a

---

[22]pinata.cloud, eternum.io

different way than creating votes? Currently, we allow for as much flexibility as possible. Aragon offers a multi-token Finance[23] App that can be used to manage budgets, to track income and expenses in an organisation, as well as to perform payments transparently according to customisable rules that can be different for each DAO.

6. Conclusions and future work

There is an incentive to use blockchain technology for collaborative processes because it is inherently trustworthy. In Smart Papers, we used Ethereum to provide the framework for collaborative authoring, IPFS for the storage of the papers, and Aragon to manage governance through a Web interface.

We analysed four use cases demonstrating how the nature of scientific publishing would benefit from storing agreements and artefacts in a verifiable distributed database that does not reside within the confines of a single point of failure, and also does not rely on a centralised party to provide proofs. We found that blockchains, by their design, are appropriate for these use cases.

We have conducted testing to run simulations using a suite of Ethereum smart contracts that we have developed based on our Smart Paper model and workflow. Future development should be focused on implementing a robust workflow allowing scholars to integrate their Smart Papers into a wider conference/journal ecosystem. One strongly suggested avenue is to create a peer-review platform on the blockchain, in which the reviewed papers can be represented by Smart Papers, and the reviews themselves are notarised on the blockchain. Aragon can be easily reused for managing the governance aspect of such a transparent, decentralised and tamper-proof set-up for peer review.

Within the context of informetrics, crucial work remains to take our requirements further so that they encompass calculations of more substantial metrics such as citation counts per author, h-Index values, and to provide trustworthy decentralised computations of many useful author-, journal- and article- based bibliometrics. With the growth of distributed data analytics on the horizon, solutions are slowly appearing that enable big data analytics on blockchains [33], which shows real promise for developing our informetrics use case.

Further research work is needed to explore how the market conditions for transaction execution may impact our design, and how market volatility could impact user behaviour through the variable nature of gas pricing and transaction completion times. The stability and security of the Ethereum network is currently seeing novel research which needs to be constantly monitored. We would like to further explore the storage options for artefacts, metadata and reviews, to optimise for cost and flexibility (in particular, the exploration of STORJ,[24] FileCoin[25] and the Swarm proof-of-concept[26]). Additionally, novel work is required to establish approaches for the management of transparent, fair and efficient budgets for funding Smart Papers transactions.

We believe that blockchains succeed at providing an essential level of decentralised trust in collaborative processes such as scholarly authoring. In our case, these guarantees can be provided at a level as low as *$0.04* per Smart Paper transaction. Future work needs to address the cost of more frequent yet even smaller operations like comments on articles. We would also like to explore the mapping of the Smart Papers workflow to a relevant ontology (for example PWO [17]) to allow each paper to be traced

---

[23]https://wiki.aragon.org/dev/apps/finance/

[24]https://storj.io/

[25]https://filecoin.io/

[26]https://swarm.ethereum.org

in a semantically standardised way. Current work of the authors is focused around implementing a Peer Review layer as a blockchain DAO template that allows relevant reviews to be easily linked to newly published Smart Papers.

Acknowledgements

References

[1] G. Abramo, Revisiting the scientometric conceptualization of impact and its measurement, *Journal of Informetrics* **12**(3) (2018), 590–597. doi:10.1016/j.joi.2018.05.001.

[2] A. Abu-Jbara, J. Ezra and D. Radev, Purpose and polarity of citation: Towards nlp-based bibliometrics, in: *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2013, pp. 596–606.

[3] D. Adam, Citation analysis: The counting house, Nature Publishing Group, 2002.

[4] M. Alharby and A. van Moorsel, Blockchain-based smart contracts: A systematic mapping study, *CoRR* abs/1710.06372 (2017). http://arxiv.org/abs/1710.06372.

[5] J. Bar-Ilan, Informetrics at the beginning of the 21st century – a review, *Journal of informetrics* **2**(1) (2008), 1–52. doi:10.1016/j.joi.2007.11.001.

[6] S. Bartling, Blockchain for open science and knowledge creation, Technical Report, Blockchain for Science. http://www.blockchainforscience.com/2017/02/23/blockchain-for-open-science-the-living-document/.

[7] J. Benet, Ipfs-content addressed, versioned, p2p file system, arXiv preprint arXiv:1407.3561 (2014).

[8] L. Bornmann and W. Marx, Methods for the generation of normalized citation impact scores in bibliometrics: Which method best reflects the judgements of experts?, *Journal of Informetrics* **9**(2) (2015), 408–418. doi:10.1016/j.joi.2015.01.006.

[9] S. Capadisli, A. Guy, C. Lange, S. Auer, A. Sambra and T. Berners-Lee, Linked data notifications: A resource-centric communication protocol, in: *The Semantic Web*, Vol. 10249, E. Blomqvist, D. Maynard, A. Gangemi, R. Hoekstra, P. Hitzler and O. Hartig, eds, Springer International Publishing, Cham, 2017, pp. 537–553. ISBN 978-3-319-58067-8 978-3-319-58068-5. doi:10.1007/978-3-319-58068-5_33.

[10] S. Capadisli, A. Guy, R. Verborgh, C. Lange, S. Auer and T. Berners-Lee, Decentralised authoring, annotations and notifications for a read-write web with dokieli, in: *Web Engineering*, Lecture Notes in Computer Science, Springer, Cham, 2017, pp. 469–481. ISBN 978-3-319-60130-4 978-3-319-60131-1. doi:10.1007/978-3-319-60131-1_33.

[11] F. Casati, F. Giunchiglia and M. Marchese, Liquid publications: Scientific publications meet the web, Technical Report, 1313, University of Trento, 2007.

[12] D.S. Chawla, Open-access row prompts editorial board of Elsevier journal to resign, 2019. doi:10.1038/d41586-019-00135-8. https://www.nature.com/articles/d41586-019-00135-8.

[13] E. Davenport and B. Cronin, Knowledge management: Semantic drift or conceptual shift?, *Journal of Education for library and information Science* (2000), 294–306.

[14] J. Eberhardt and S. Tai, On or off the blockchain? Insights on off-chaining computation and data, in: *European Conference on Service-Oriented and Cloud Computing*, Springer, 2017, pp. 3–15. doi:10.1007/978-3-319-67262-5_1.

[15] *ETH Gas Station*, accessed January 2018. https://ethgasstation.info/FAQpage.php.

[16] F. Franceschini, D. Maisano and L. Mastrogiacomo, The museum of errors/horrors in scopus, *Journal of Informetrics* **10**(1) (2016), 174–182. doi:10.1016/j.joi.2015.11.006.

[17] A. Gangemi, S. Peroni, D. Shotton and F. Vitali, The publishing workflow ontology (PWO), *Semantic Web* **8**(5) (2017), 703–718. doi:10.3233/SW-160230.

[18] A. Garcia-Castro, A. Labarga, L. Garcia, O. Giraldo, C. Montana and J.A. Bateman, Semantic web and social web heading towards living documents in the life sciences, *Web Semantics: Science, Services and Agents on the World Wide Web* **8**(2–3) (2010), 155–162. doi:10.1016/j.websem.2010.03.006.

[19] L.L. Haak, M. Fenner, L. Paglione, E. Pentz and H. Ratner, ORCID: A system to uniquely identify researchers, *Learned Publishing* **25**(4) (2012), 259–264. doi:10.1087/20120404.

[20] A.-W.K. Harzing and R. Van der Wal, Google scholar as a new source for citation analysis, *Ethics in science and environmental politics* **8**(1) (2008), 61–73. doi:10.3354/esep00076.

[21] L. Heller, R. The and S. Bartling, Dynamic publication formats and collaborative authoring, in: *Opening Science*, S. Bartling and S. Friesike, eds, Springer International Publishing, Cham, 2014, pp. 191–211. doi:10.1007/978-3-319-00026-8_13.

[22] M.R. Hoffman, L.-D. Ibáñez, H. Fryer and E. Simperl, Smart papers: Dynamic publications on the blockchain, in: *The Semantic Web*, A. Gangemi, R. Navigli, M.-E. Vidal, P. Hitzler, R. Troncy, L. Hollink, A. Tordai and M. Alam, eds, Springer International Publishing, Cham, 2018, pp. 304–318. ISBN 978-3-319-93417-4. doi:10.1007/978-3-319-93417-4_20.

[23] W.L. Hursch and C.V. Lopes, Separation of concerns, Technical Report, NorthEastern University, 1995.

[24] R. Jha, A.-A. Jbara, V. Qazvinian and D.R. Radev, NLP-driven citation analysis for scientometrics, *Natural Language Engineering* **23**(1) (2017), 93–130. doi:10.1017/S1351324915000443.

[25] T. Kuhn and M. Dumontier, Making digital artifacts on the web verifiable and reliable, *IEEE Transactions on Knowledge and Data Engineering* **27**(9) (2015), 2390–2400. doi:10.1109/TKDE.2015.2419657.

[26] P.A. Lawrence, The mismeasurement of science, *Current Biology* **17**(15) (2007), R583–R585. doi:10.1016/j.cub.2007.06.014.

[27] E.D. López-Cózar, N. Robinson-Garcia and D. Torres-Salinas, Manipulating Google scholar citations and Google scholar metrics: Simple, easy and tempting, arXiv preprint arXiv:1212.0638 (2012).

[28] I. McGregor-Lowndes et al., The rise of the DAO disrupting 400 years of corporate structure, *Proctor, The* **39**(1) (2019), 32.

[29] J. Mingers and L. Leydesdorff, A review of theory and practice in scientometrics, *European journal of operational research* **246**(1) (2015), 1–19. doi:10.1016/j.ejor.2015.04.002.

[30] R.S. Sandhu and P. Samarati, Access control: Principle and practice, *IEEE communications magazine* **32**(9) (1994), 40–48. doi:10.1109/35.312842.

[31] H. Small, On the shoulders of Robert Merton: Towards a normative theory of citation, *Scientometrics* **60**(1) (2004), 71–79. doi:10.1023/B:SCIE.0000027310.68393.bc.

[32] R. Van Noorden, Open access: The true cost of science publishing, *Nature News* **495**(7442) (2013), 426. doi:10.1038/495426a.

[33] H.T. Vo, M. Mohania, D. Verma and L. Mehedy, Blockchain-powered big data analytics platform, in: *Big Data Analytics*, A. Mondal, H. Gupta, J. Srivastava, P.K. Reddy and D.V.L.N. Somayajulu, eds, Springer International Publishing, Cham, 2018, pp. 15–32. ISBN 978-3-030-04780-1. doi:10.1007/978-3-030-04780-1_2.

[34] R. Wattenhofer, *The Science of the Blockchain*, 1st edn, Inverted Forest Publishing, Erscheinungsort nicht ermittelbar, 2016, OCLC: 952079386. ISBN 978-1-5227-5183-0.

[35] G. Wood, Ethereum: A secure decentralised generalised transaction ledger, *Ethereum Project Yellow Paper* **151** (2014). https://github.com/ethereum/yellowpaper.