Using Blockchain Technology To Develop A decentralized & Digital Scientific Community Driven By Crypto Incentives To Streamline Processes’ In The Current Academic Business Model

Table of Contents

Abstract

The scientific enterprise relies on a peer-review process to maintain the quality of academic discourse and to ensure researchers develop a valid and consistent cumulative body of knowledge. In recent years, it appears that the review capacity in academia has decreased, which indicates that the community's hunger for publication accompanies only a modest appetite for providing the necessary support to sustain the consequent increase in peer-review load. The advent of blockchain technologies and the proliferation of cryptocurrencies presents an opportunity to develop a token-based peer-review payment system that can clear the congested review pipelines while also controlling for quality and spreading the equity that peer review generates in a fair fashion through market-regulation mechanisms. The transparent, immutable and distributed nature of blockchain technology also gives birth to the notion of an "internet of value" which for the first time in human history allows individuals to truly own their own data rendering the possibility to monetise ones digital presence, or rather to generate real tangible value to our digital persona and intellectual property and hence opening the gate to new and improved business models that operate in the best interests of the community rather than a few elite at the top. Despite the digital transformation of the publishing industry, little has been done thus far to address the inefficiency of the review process. The typical review cycles, which are measured in years, suggest that something needs to change. Developing a token-based peer-review payment system may be an opportunity not only to address the apparent challenge in the peer-review process but also to assert our proclaimed role as stewards of the digital revolution. We build off of our previous literature review which analysed the need and possibility to develop a new digital science ecosystem which leverages blockchain technology to streamline various process' such as peer review. In this paper extend those findings and present our theoretical model which sees the development of a robust system architecture which implements many blockchain concepts such as community consensus and distributed governance through a system of smart contracts in order to fully realise a decentralised scientific peer-review ecosystem.

1.0 Introduction

Projects in fundamental science, as well as more applied Research and Technology Development (RTD) aim to produce outcomes that are beneficial for the public commons, in other words, that they serve the good of the people and societies through the advancement of knowledge, culture and lives. In general, only results that are credible, in a sense that they withstand independent validation in compliance with widely accepted standards and methods, represent genuine value slated to make significant impact over time.

Over several decades, a scientific culture has been established that is based on the evaluation of scientific initiatives and their outcomes through peer-review of proposals for funding and publications. While this system has worked reasonably well and is now deeply engraved in many aspects of the scientific community, including ranking systems **[1]** of researchers and their organisations, It does expose some intrinsic shortcomings of the current academic business model. Amongst them include poor anchoring in prior art and proper acknowledgement of manifold types of individual contributions and the skint integration of the wisdom of the crowd. This is not to mention that soilism and elitism can sometimes play a role where author affiliations might be a decisive factor in accessing high ranking journals and conferences, **[2]).** In some cases, ubiquitous predatory publishers focus on collecting publication fees rather than assuring the quality of available scientific information as discussed in this paper **[3].** Although, many aspects of this legacy system work incredibly well, it must be acknowledged that like many industries and corporations, there is a tend to nurture an unproductive, author centric culture, leading to the pervasive publishing of fractional and intermediate results without rewarding solid validation. That being said, the traditional business model of academia does work well and has prospered for many years; but the constant evolution of the world wide web has been changing everything. It opens up the possibility of authors communicating directly with readers without any intermediary **[4].** The development of the Internet enabled an expansion of the proposals for alternatives for both science dissemination **[5]** and evaluation **[6].** The reduction of distribution costs enabled wider access to scientific knowledge and has questioned the role of traditional publishers **[7].** It is acknowledged that the Open Access and Open Science movements have successfully reduced the economic cost of readers to access knowledge **[8].** However, it has not successfully challenged traditional publishers' business models **[9]** that are often charging both readers and authors **[10].**

Traditional peer review has suffered a variety of criticisms, and yet only few alternatives have gathered success **[11].** Existing literature, as discussed in **[12]** (actual paper) provides multiple proposals around open peer review [13], and proposals of reputation networks for reviewers **[14].** In fact, a start-up, Publons1, provides a platform to acknowledge reviews and open them up to the wider public. In addition, other alternatives to the traditional science publication process have arisen in the last 20 years. Preprints emerged in the 1990's during the first era of the internet and they are scientific papers that have not been peer-reviewed and therefore have not been published in a journal or conference. Platforms such as arXiv2 and Preprints.org3 have been successful within the scientific community, allowing these pre-published papers to gain more visibility by the wider community **[15].** Social networks have also carved a niche in the scientific community. Platforms such as Academia4 or Research Gate5 are being used by more people every day, allowing researchers to upload their published papers, further connecting the scientific community. Nevertheless, the mentioned platforms are centralized, with an infrastructure typically controlled by a sole private entity. This centralization has multiple implications **[16, 17],** for example, less control and self-management for the scientific community; a requirement of blind trust in a third-party that can change its terms or company policies at anytime (e.g. in case of a buy-in); or problems related to for-profit business models which may affect users, or their data. As we discovered in our review of current literature, their doesn't yet exist any viable alternative to the current academic infrastructure that truly succeeds in redistributing the wealth and ownership of data in the scientific community to the true pioneers of the space such as the researchers, scholars, and the academics who are at the forefront of all scientific innovation. In this paper we explore the idea of blockchain and how the overarching scientific community could perhaps profit from its unique bundle of characteristics. For the first time in human history the decentralised, transparent and immutable nature of blockchain applications give birth to a concept known as the "internet of value" **[18]** (web3.0) where one can truly own their own data and through an free-market like economy built on the premise of incentives and community governance, we now have the possibility monetise our digital presence, or rather to generate real tangible value to our digital persona and intellectual property, opening the gate to new and improved business models that operate in the best interests of the community rather than a few elite at the top.

1.1 Blockchain & The Internet Of Value

Some decentralized alternatives to the peer-review process have been scoped, but despite their promises **[19],** are still in their infancy. A few proposals (none of them functional to date) have appeared recently: peer review proposal platforms using cryptocurrencies **[20],** blockchain-enabled applications with voting and storage of publications **[21],** or a peer review quality control through blockchain-based cohort trainings **[22].** Additionally, the new Ledger6 journal records the publication timestamps in the Bitcoin blockchain. By its very construct, the distributed ledger technology (DLT) blockchain provides solid trust, even between mutually unknown parties in a digital environment, through its immutable, merely algorithmically controlled consensus mechanism. The integrity of the blockchain is secured by demanding the staking of assets from participants, and rewarding good behaviour, typically via its native cryptocurrencies; with a slight ironic touch, this self-sustaining mechanism may arguably be viewed as the largest-scale behavioural incentivization program in human history. The thought of all of this is very promising when viewed or considered in terms of the application in the academic space and in this paper, this is largely what we will be focusing on. The transparency, trust and decentralization have the possibility to enable researchers to build their own digital open ecosystem or economy for research, communication and much more that are all in line with the philosophy of open science. Besides the reproducibility of experiments **[23][24]**, one of the main promised of the application of (BT) in open science is towards trust-orientated problems such as the sometimes profit driven behaviour that is going on in the publishing and peer-review sector **[25]**, not to mention the restrictions of the free and open access to scientific publications, patents, and research **[26].**

Although the outcomes can be considered promising in relation to the decentralization of the trust issue problems such as the peer review process and free open access science, these alone are not enough in terms of substantial revelations or innovations to generate any meaningful impact regarding valuable "change" in the space in relation to how things operate which we concluded in our literature review. However, DLT offers the opportunity to shift the status quo, which boasts incredibly unique and creative transformations in the space, and this is all because BT stands out from other systems in its exceptional technical architecture, which allows the technology to get adapted for a variety of use cases. The idea of merging two industries namely, crypto & science together, yields the possibility and opportunity to create unique business models and incentives for users or entire communities. This gives way to the creation of an entirely digital ecosystem where concepts like crowdfunding, crowdsourcing **[2],** incentivized consensus, liquid governance, tokenomics, prediction markets and arbitration **[27]** (see appendix for said terms) can all be merged to define a virtual scientific community where the distribution of wealth and community drive is shared amongst all network participants while maintaining and incredible standard of open access and valid "information" in all of its defined forms.

This paper scopes a conceptual framework enabled by the joint action of the distributed ledger technology (DLT), commonly termed, "blockchain", to capitalize on crypto economical mechanisms, such as tokenization, consensus, crowdsourcing, smart contracts (see appendix), staking, and reward systems. Project contributions, such as methods, experimental data, modelling, simulation, assessment, predictions, and directions can all be crowdsourced using crypto economical reward and reputation schemes. The ability to open current research to the community at large and could be a substantial step in the right direction in regard to overcoming the current reproducibility crisis in the academic space. The so enabled, highly integrative approach, termed decentralize science ("DeSci"), is slated to move research out of its present silos, and to markedly enhance quality, credibility efficiency, transparency, inclusiveness, impact, and sustainability of research initiatives. The idea is that the integrity of Intellectual property (IP) is verified by trusted parties and stored on a public ledger provided by blockchain, possibly implemented with the Interplanetary File System (IPFS) and operating on the database of knowledge formed by a huge collection of academic research papers, patents etc, can be efficiently managed by smart contracts and decentralized applications ("DApps") executed on the blockchain. The key lubricant to stimulate comprehensive community engagement in such crowd-run projects constitutes attractive incentivization and taking of all contributions by blockchain-innate crypto assets and reputation ranking systems. In the coming sections, we, for the first time, layout an intricate architecture using the web3 toolbox to develop an ecosystem where all of this can take place. This paper severs as the explanation and presentation of our theoretical model and technical architecture to both create a decentralised scientific ecosystem and also to address some of the challenges that come as a result

2.0 Shortcomings Of The Current Scientific Business Model

It remains clear that Blockchain Technology can be painted in a promising light in relation to its application and integration various processes in academia. We now want to address some fundamental problems that currently exists in the current scientific infrastructure, that are currently thus far unaddressed or not being actively solved. We do this to provide context for the ability of BT to render a potentially viable and effective solution to the problems addressed in this section. An extensive analysis was conducted in our literature review which specifically explored tackled this problem, but nonetheless we will re-iterate some of our core findings here again before starting the main sections of this paper

2.1 Irreproducibility of scientific results

Arguably one of the largest problems that exists in science today is the concept of the “reproducibility crisis”. From our conclusion of the current literature in the blockchain for science space, it is exactly this problem that garners a lot of hype where it is considered that BT could likely present a viable solution to due to the transparency of data and integrity of data stored in its underlying ledger. As it stands the entire scientific community is concerned with the inability to consistently reproduce scientific results. Although this is no secret, these papers **[28][29][30]** give evidence that publishing reproducible analyses is a long-standing widespread challenge for the scientific community, funding bodies and also publishers. This problem affects nearly all disciplines of science **[31] (see section 5.1)** and there is surprisingly little work that is making ground in terms of a definitive solution. It is unfortunate because reproducibility is a key property of any scientific endeavor, enabling the progressive structuring of knowledge into innovation through a process of knowledge integration and reuse in future studies to come. Whether how much science is unreproducible because of chance or the because of the fact that the initial results have been overclaimed or even falsified is another question. An essential topic of an open science system is the possibility to provide a collaborative environment. BT and its decentralization can support that goal by enabling, among other things, all users to share the same data version. In detail, data consists of, for example, experiment results, communication content, drafts, open peer-reviews, and raw data. Also, as mentioned, specific groups or the whole network can make decisions collaboratively through ordinary votes that can follow, for example, a democratic approach **[32].** Subjects of these polls could be topics like the future development of the network, to add/remove specific features, or to accept/rate proposed projects and contributions. On a technical perspective, the validation and management of a blockchain infrastructure work as well collaboratively through the consensus mechanism in which all users take part. It also ensures data integrity and consistency in a blockchain. The immutable (tamper-proof) nature of the BT is an ideal feature to fulfil the requirement to prevent censorship of any kind. As we described in section 4.1, cryptographic hashing, a consensus mechanism, and decentralization in combination guarantee the immutability of a blockchain. Participants of a network can only append data but not modify stored data. This property suits to science that should not underlie any censorship. Everyone should be able to freely express his or her opinion without getting restricted in any way. In the use case of research, it also includes the publishing of scientific work that has critical statements or topics. Overall, an open science infrastructure based on BT can provide such a censorship-free environment.

2.2 Scientific Publication Paywalls

In 2016, the EU Ministers of science and innovation, assembled in the Competitiveness Council, resolved that all European scientific publications should be immediately accessible by 2020 which as we can clearly see today is not the case. Scientific journals such as Springer still charge a “pay per view” fee in order to access scientific publications and they are not alone. The subscription-based model of scientific publishing emerged at a certain point in the history of science, when research papers needed extensive typesetting, layout design, printing, and when hardcopies of journals needed to be distributed throughout the world. While moving from print to digital, the publishing process still needs services, but the distribution channels have been completely transformed. There is no valid reason to maintain any kind of subscription-based business model for scientific publishing in the digital world, where Open Access dissemination is maximizing the impact, visibility, and efficiency of the whole research process.

Publishers should provide services that help scientists to review, edit, disseminate, and interlink their work and they may charge fair value for these services in a transparent way. The minimal standards for services expected from publishers are laid down on page 6 of the 2015 ‘Science Europe Principles on Open Access Publisher Services’ **[32]**. These are only a handful of the common occurrences that demonstrate some of the corruption and lean towards centralization in the traditional scientific community. One fact that we can draw from some of these concerns is that new research builds on established results from prior work. The chain, whereby new scientific discoveries are built on previously established results can only work optimally if all research results are made openly available to the scientific community. Although having said all of this, it must also be noted hat in recent years, large efforts have been made to tear down some of the publication paywalls that are withholding research from large fractions of the scientific community and society as a whole **[10]**. However, this trend needs to continue and at an even greater pace if open science is to thrive and flourish in the years to come as monetizing the access to new and existing research results is profoundly at odds with the ethos of science **[33].**

2.3 Proposing a Blockchain based solution

Researchers and research funders have a collective duty to of care for the science system. The 2003 Berlin Declaration **[34]** provides a strong manifestation of the science community to regain ownership of the rules governing the dissemination of scientific information. Also, Science Europe established principles for the transition to Open Access in 2013 **[33]** but wider overall progress has been slow. As mentioned in **section 2.2**, the EU Ministers of science and innovation, declared that all European scientific publications should be immediately accessible by 2020 which as we can see today is not the case. However, in my opinion in light of these ongoing efforts to improve the space, more radical solutions need to be explored. It is crucial to understand the “matrix” as a whole because from the issues discussed in section2.2 and 2.3 we can surmise that there certainly does exists several demands and characteristics that already complement each other between the traditional open science framework and that of a BT proposed framework. For example, it is useful or needed for many functions like providing a “trail” of research so that there is “no censorship” possible in a blockchain network to provide a trustworthy environment. Such an immutable and transparent trail of information could also contribute to solving the irreproducibility problem not to mention the ability to define more robust communication channels between researchers, publishers and funders. And the BC could be the root whereby all collective scientific information would be linked and openly available on a distributed ledger **[35].** Other shortcomings of the current infrastructure such as the sometimes unskilled peer-reviewers, the unrewarding peer-review process itself and the occasional occurrences of malicious peer-review feedback could both be solved by a BT network through the implementation of a incentivized governance model similar to that of a proof-of-stake consensus mechanism as proposed by the Ethereum 2.0 network **[8]** whereby an issued utility token could serve as the basis of an economic model such that good behavior and hard work is strongly incentivized through “token rewards” whilst the opposite is strongly discouraged through the implementation slashing protocols **[36].** This is explained in more detail in **section 6.1** and **6.2**. In this section we highlight all of these inefficiencies and shortcoming of traditional science and probed some of the possibilities that blockchain could bring to the table in order to make the academic space more fluid.

Now that the various problems associated with the academic business model have been brought to light and how blockchain offers worthy merit as a means to solve some of these issues layed forth, we can now set the grounds for describing our own theoretical model which uses common tropes and technologies associated with developing decentralized applications as well as the development and integration of fundamental tried and tested blockchain protocols such as distributed consensus and on-chain governance (see appendix) in order to initiate a digital ecosystem in which the entire scientific community, can openly share knowledge, communicate and validate each other for market based incentives and asset monetization opportunities. Web3 truly allows one to create a notion of value in the so called “metaverse” through many techniques that we will explore in the coming sections

3.0 High Level Overview Of System Architecture

In this section we will define a high-level overview of the entire architecture that underpins our decentralised scientific ecosystem. This section is aimed to serve as a gentle introduction to some of the fundamental blockchain concepts and protocols that work together in synergy to govern the inner workings of our infrastructure. We begin by abstracting the core components of the platform into easy-to-understand concepts before dissecting each in greater detail in the sections to come. Until now we concerned ourselves with the fact that the rise of blockchain technologies and the proliferation of cryptocurrencies **[37]** presents an opportunity to use a token-based peer-review payment system to resolve the apparent shortage or sometimes inefficient use of reviewers in the current publication ecosystem which we outlined in section 2. A few years ago, applying market regulation would have required a centrally managed platform that would impose standards and assert control over the process. Centralized governance architecture of that sort would not cohere with the culture of scholarship and, therefore, would likely be no more efficient that the current system which is in place today. In contrast, blockchain-based decentralized market regulation mechanisms would offer all key stakeholders a platform to manage the supply and demand for competent reviewers in a favourable fashion that is compatible with academia's prevailing socio-cultural practices. Cryptocurrencies such as Bitcoin use blockchain-based tokens that enable users to represent and exchange value and tangible externalities without the need for centralized governance architecture to facilitate clearing and to maintain the market's integrity **[38]**. The diffusion of self-governed decentralized value-exchange solutions marks the advent of various open-Web services that allow users to exchange goods and services without the need for intermediation or central management. In other words, blockchain-based distributed governance models have emerged to emancipate users from the grip of often inefficient, entrusted third parties and replaced them with cryptographic verification. Building on the advent of blockchain technology, a token-based peer-review payment system could provide market regulation of reviewers' availability while maintaining the full independence of the journals and catering for authors' and reviewers' interests.

3.1 Core Concepts

In our envisioned token-based peer-review scheme the entire concept reduces to the notion of "an internet of value". Which we commented on in the introduction. The crucial aspect of market-based peer-review is the implementation of a digital currency in which stakeholders like researchers, academics use to partake in the system to earn rewards through incentive based schemes. Authors will pay submission fees with an ERC20 token which we coin "Decentralised-Science-Token" or DST and reviewers receive DST for their services. The difference here is that the submission fees do not get funnelled to any centralised journal entity but rather they get redistributed to peer-reviewers themselves. Thus, DST, in one sense, is a cryptocurrency that fuels the peer review market in which DST is exchanged, paid, and earned for review services amongst other things. Token-based market regulation uses external market mechanisms such as price and competition to establish peer-review standards that can measure and reward individual contributions. Token-based market regulation can mitigate the drawback of normative regulation and avoid the drawbacks of operational regulation. The market mechanism can clear the congested review pipelines and eliminate free riders while controlling for quality and spreading the equity generated through the peer review in a fair fashion. The diagram below describes the platform architecture in its most abstracted and fundamental level.

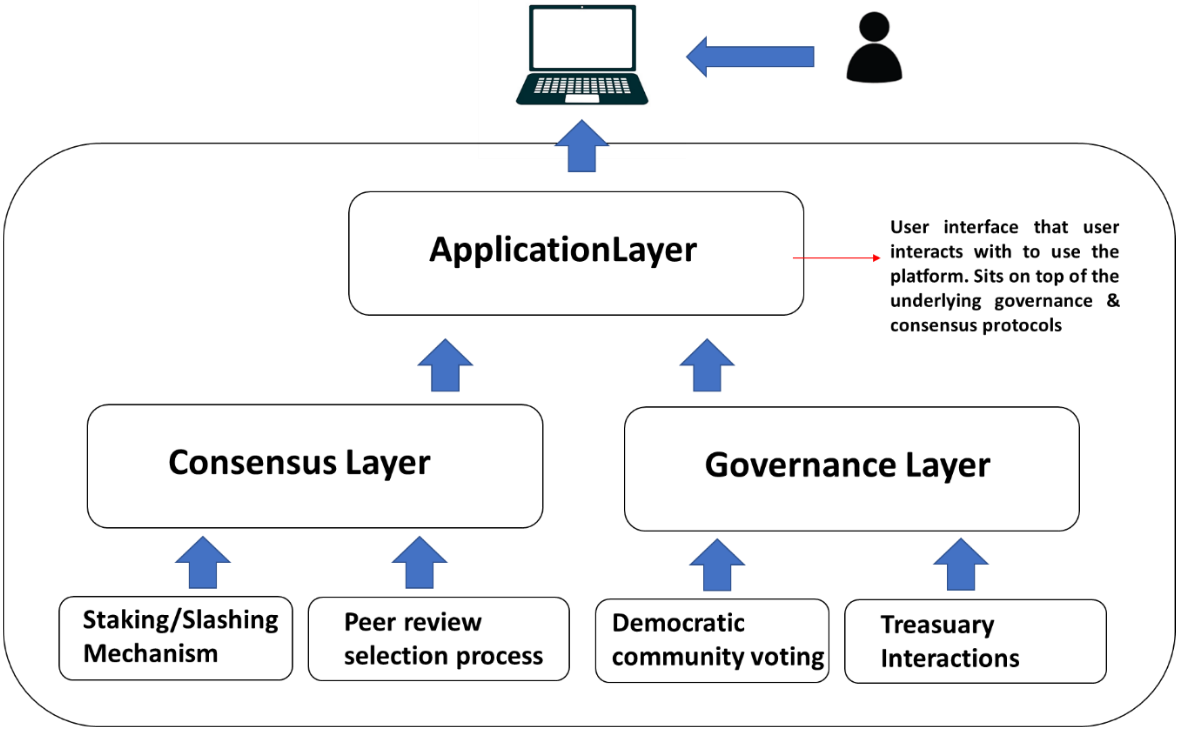

ApplicationLayer
User interface that user interacts with to use the platform. Sits on top of the underlying governance & consensus protocols
Consensus Layer
Governance Layer
Staking/Slashing Mechanism
Peer review selection process
Democratic community voting
Treasuary Interactions

Figure 1: Overview of Technical Architecture of Proposed Application Framework

From **Figure1** above we can see that our infrastructure can be De-structured into three main layers. Those are, the consensus layer, the governance layer and the application layer. The consensus and governance layer work together in synergy on a fundamental protocol level and enable the complex logic that governs the various processes' that run in parallel within the entire system. On top these layers sits the application layer and it is here that the user interacts with our platform through some user interface accessed via a web browser. Both governance and consensus are extremely important concepts in decentralized applications and for our application they have two very contrasting roles within the ecosystem and as we see in **Figure1** they serve as the backbone for all of the other main protocols and mechanisms that amalgamate to create our platform logic

3.2 Consensus Layer

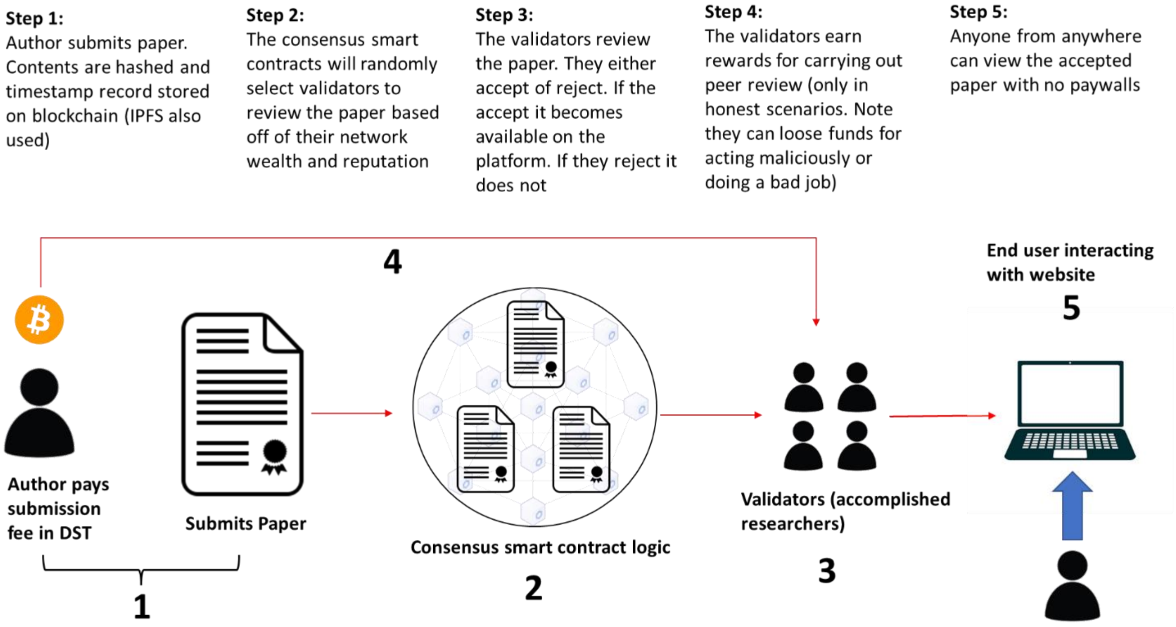
In blockchain theory consensus, commonly referred to as "consensus mechanisms" is a fault-tolerant (see appendix) mechanism that is used to achieve the necessary agreement on a single data value or a single state of the network among distributed processes or multi-agent systems, In other words consensus defines how the agreement amongst mutually unconnected parties can come to agreement on some piece or data in order to determine the absolute truth with the network. A blockchain's consensus mechanisms is what upholds the integrity of the data stored in its ledger and thus it is the most crucial factor for record-keeping, among other things. The two most common types of consensus mechanisms used today are Proof-Of-Work and Proof-Of-Stake. Both are similar in the fact that they are used to achieve unanimity in distributed systems but differ in the methods and practices they employ in order to achieve this unanimity

- **PoW** is the consensus algorithm used in bitcoin. Its core idea is to allocate the accounting rights and rewards through the hashing power competition among the nodes (miners). Based on the information of the previous block, the different miners calculate the specific solution of a mathematical problem. The math's problem is extremely complex and the technical details are not within the scope of what we want to capture in this paper but in short it requires running millions of computations until a miner's software generates and hash output with a certain amount of leading zeros which is dictated by the current difficulty of the problem at the time. The first node that solves this math problem can create the next block and get a certain amount of bitcoin reward.

The expense of computation power is a miners “proof” for doing “work” in order to earn a bitcoin reward.

- **Proof of Stake (PoS)** is an alternative to PoW. The Proof of Stake consensus algorithm uses a selection process that is pseudorandom in nature to pick the validator of the subsequent block from the existing nodes. Unlike POW, POS requires validators (nodes) to “stake” or lock up a large sum of tokens. Based off of the amount the validator has locked, increases its chance of being selected to mint a new block in the future. This process is based on a mix of several factors which include randomization and staking age (see appendix) along with the validators wealth (total of staked tokens)**[41]**. While in PoW, the miner which first solves the complex problem mines the next block and receives rewards, in PoS, the individual validator which creates the next block is selected based on how much they have “staked” in comparison to other competitor nodes through a random lottery **[40]**. stake is usually based on the number of coins the network node has for the particular blockchain it is attempting to mine. In these systems, the transaction fee is generally the reward, and users who want to be among the participants in the forging process need to lock their stake (a certain amount of coins) in a network.

In our model we adopt elements of POS but our approach to consensus fundamentally unique. Our application is not an actual blockchain itself but rather a platform built on top of the application layer of the Ethereum blockchain (see appendix). This means that whenever someone within our app executes a transaction, it will be handled by the Ethereum miners via the process described above in POS. However, our consensus model is more concerned with the question of how a subset of peer-reviewers in the network can come to agreement on whether or not a research paper that gets submitted for publication is of good quality or not. For this we use smart contracts (see appendix) to mimic the characteristics of a typical POS consensus mechanism. The way it works is that we have a set of “validators” (researchers) who are tasked with carrying out the peer-review process on academic papers. Whenever a set of validators are called upon to conduct a peer review through a random selection process (which makes the selection based off of each validators staked token wealth and reputation score). If consensus is achieved amongst the validators, indicating with high confidence that the material is of academic excellence then the work can be ultimately appended our internal ledger (smart contract storage) so that it can be read and cited by the wider community. In short, each validator in this case will be incentivized to carry out the peer review with the upmost integrity in return for a token-based reward, which originates from the fees the author paid on submission and through other various mechanisms which we will discuss later sections. On the other hand if they fail to carry out their duty as a peer reviewer, they can get punished and lose some of their tokens. This dual mechanism is known as staking & slashing.



Step 1: Author submits paper. Contents are hashed and timestamp record stored on blockchain (IPFS also used)

Step 2: The consensus smart contracts will randomly select validators to review the paper based off of their network wealth and reputation

Step 3: The validators review the paper. They either accept of reject. If the accept it becomes available on the platform. If they reject it does not

Step 4: The validators earn rewards for carrying out peer review (only in honest scenarios. Note they can loose funds for acting maliciously or doing a bad job)

Step 5: Anyone from anywhere can view the accepted paper with no paywalls

4
Author pays submission fee in DST
Submits Paper
1
Consensus smart contract logic
2
Validators (accomplished researchers)
3
End user interacting with website
5

Figure 2: Simplified flow diagram of market based peer review process

Figure2 above shows in its most simple from, the flow of the submission/review process. It is important to note that the diagram above leaves out a numerous features such as the case where a submission is rejected, or the case where a validator acts in bad faith and is punished etc. This is intended as this diagram is meant to serve as an easy to follow visual to understand the basic process. We will be exploring the more complicated concepts in the sections to come but until then this flow will suffice.

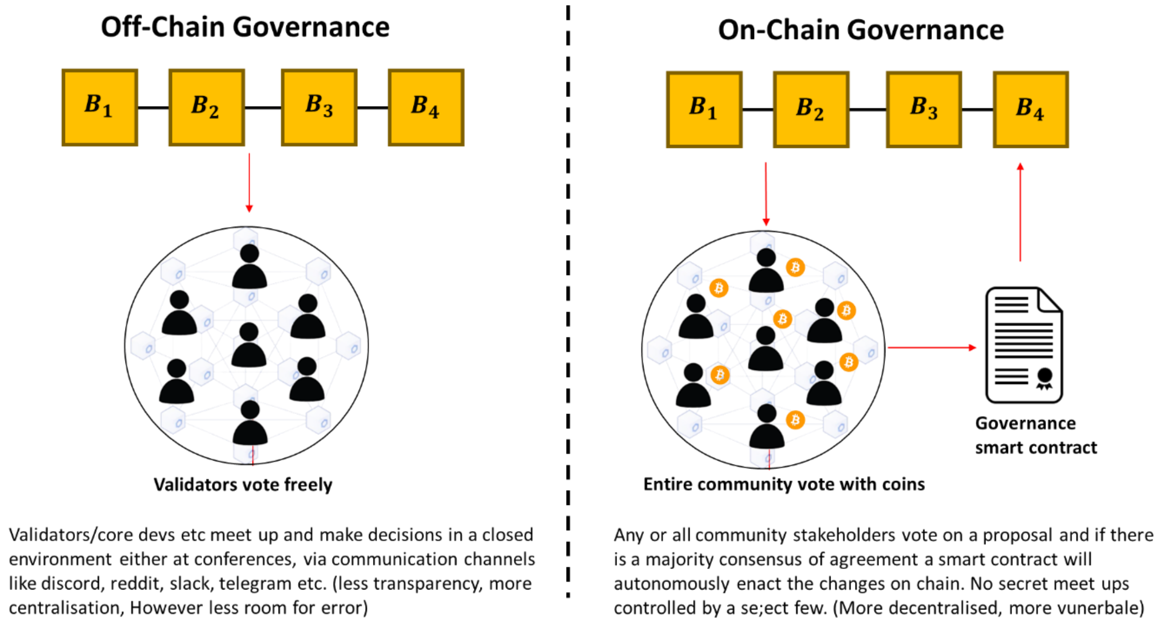
Staking & Slashing

From **Figure1** we can see the reduction of the consensus layer into two sub facets, one of those facets is the staking and slashing mechanism. Staking and slashing play an integral role in any POS like consensus algorithm, and they are the at at the crux of any successful implementation of such a protocol. As we already know, we use the requirement of locking up or the "staking" of coins for validators so that our underlying consensus algorithm can determine who will inevitably get selected to review a paper. The concept of staking, or coin age and coin wealth are very deliberate. In traditional POS consensus the ideology is that a validator who has their tokens locked up for longer periods of time without ever being slashed (coin age) **[42]** are generally considered more reputable in the network as they have been around longer have no or little records of acting nauseously, therefore they should be ranked higher in the network and thus have better chances of being selected to mint a block. The same thing applies in our scenario only that we not only require validators to stake tokens, but also to stake their reputation which is gained over a long period of time by participating in the ecosystem and carrying out good deeds such as quality peer-review. So, staking in other words, is just a mechanism that we can use in a distributed network in order to rank stakeholders based on their prior actions within the network. It generally goes without saying that someone with a high reputation, and coin age is more than likely going to be a better candidate to secure the network than someone who has credentials at the opposite end of the spectrum. Staking is also used as an incentivization tactic by the consensus protocol. Users who have more tokens locked are going to earn a higher rate or rewards of ROI (return on investment). The same is doubly true in our scenario where a user's eligible reward is based both on their reputation within the network as well as the total sum of their locked tokens. This tactic incentives stakeholders to refrain from acting with mal-intent to ensure that both their reputation and stake are kept as high as possible yielding higher returns

Slashing on the other hand is an in built mechanism to handle the inevitability of malicious actors in the system. The penalties or punishments associated with slashing are deliberately kept extremely harsh in order to discourage the entire Populus from trying to cheat the system at all costs. In traditional POS mechanisms as laid out in the likes of the Ethereum or Polkadot protocols use slashing in cases where validators are off-line or try to double spend transactions for example. In our system the same idea applies, only our implementation will punish stakeholders for things like slander, poor quality peer-reviews, late peer-reviews etc. In our model any stakeholder, (any community member who owns DST) can issue a slashing proposal on someone who they believe to have misbehaved as long as they have supporting evidence. Each slashing proposal must be voted to achieve super-majority consensus of agreement before the penalties are enacted or rejected. Thus, the slashing mechanism ties in heavily with our governance model (see section 6). These concepts own their own section later in this report (section8) where we will explore the indicate dynamics of these two protocols and how they are used in our ecosystem.

3.3 Governance Layer

The other core component from Figure1 that defines our architecture is the governance layer. In blockchain and distributed systems governance is a system for managing and implementing changes to the underlying protocol. Just like consensus between nodes is required within a decentralized blockchain network to continue validating and securing data, governance is required among networks of stakeholders to in order to change its laws and processes **[43]**. Because there is no centralized authority, decentralized networks and platforms rely on increasingly innovative governance mechanisms to make decisions on updates and roadmaps, and to resolve disputes in an equitable and inclusive manner. A blockchain’s governance model defines the project’s level of decentralization, accessibility, equitability, and also how it balances the interests of various stakeholders of the network. There are two subsets of “governance” models when it comes to blockchains. On-chain governance and off-chain governance. Off-chain governance functions when all stakeholders agree and make all relevant updates and implementations in unison. In contrast, on-chain governance is a mechanism that enables a decentralized community to update a protocol by voting directly on-chain. On-chain governance for a blockchain application typically takes place in the form of a community wide vote, and we usually must hold the platforms native coin to participate in its governance. Typically, in simple implementations, the weight of anyone’s vote is determined by the number of coins they hold. On-chain governance also gives birth to the concept of DAO’s or decentralized autonomous organizations which are completely democratic entity’s which are completely governed by autonomous code laid out by the protocol developers **[44]**. In a lot of cases many on-chain governance models can abruptly fail because of an uncaught loophole or bug in the underlying code. To this extreme we can summarize that the huge gain in decentralization that emerges from on-chain governance models comes at the sometimes-dire cost of manipulation and the exploitation of loopholes by attackers.



Validators/core devs etc meet up and make decisions in a closed environment either at conferences, via communication channels like discord, reddit, slack, telegram etc. (less transparency, more centralisation, However less room for error)

Any or all community stakeholders vote on a proposal and if there is a majority consensus of agreement a smart contract will autonomously enact the changes on chain. No secret meet ups controlled by a se;ect few. (More decentralised, more vunerbale)

Figure 3: Demonstration of the difference between on-chain & off-chain governance

Our platform uses on-chain governance despite the aforementioned risks. The governance model that we employ is heavily inspired from the model developed by the Polkadot blockchain (see appendix) which is extremely robust and implements many safeguards for various exploits and attack vectors. The governance protocol that we implement is responsible for putting many things within the network to vote amongst the community in the form of proposals. There exists different types of proposals for many different things such as issuing treasury proposals (see below & section (8)), proposing the creation of new scientific communities for stakeholders to join and subscribe to, as well as other things like slashing and the selection of new validators or reelection of new council/board and validator members (see below). Just like in Polkadot's model, we implement the loose notion of a hierarchy in our community. Different roles exists within the ecosystem that in theory anyone is eligible to fill. Just like we have authors and validators (reviewers) associated with the consensus protocol, our governance system employs its own statused roles. These come in the form of regular stakeholders, council members and board members. The responsibility of the council is to represent passive stakeholders who may not be able to vote on every referendum, The council will be elected with the ambition of proposing sensible referenda, as council members will stake their expertise and experience in developing, maintaining, and using decentralized networks. However, above we mentioned that our ecosystem will support micro-communities which each represent a specific niche or facet within science. For example, a community for quantum gravity research, a community for solid molecular biology etc. Each sub-community will have its own validator set and this is done to ensure that the peer-reviews are being carried out by reviewers who experts in their particular area in science. That being said this is where the council comes in. The council have limited scope in that they serve the particular community they reside in. This means that each sub-community in our ecosystem will have its own council to represent passive stakeholders and to prose sensible referenda. The Board is a similar entity to the various Councils, only they operate one level up and when it comes to extremely important governance proposals. Usually, the last say in any important vote funnels though them. In essence many of the large network changing events will go through two votes, the first round being put to the community and the second round being cast to the council or board depending on the type of proposal.

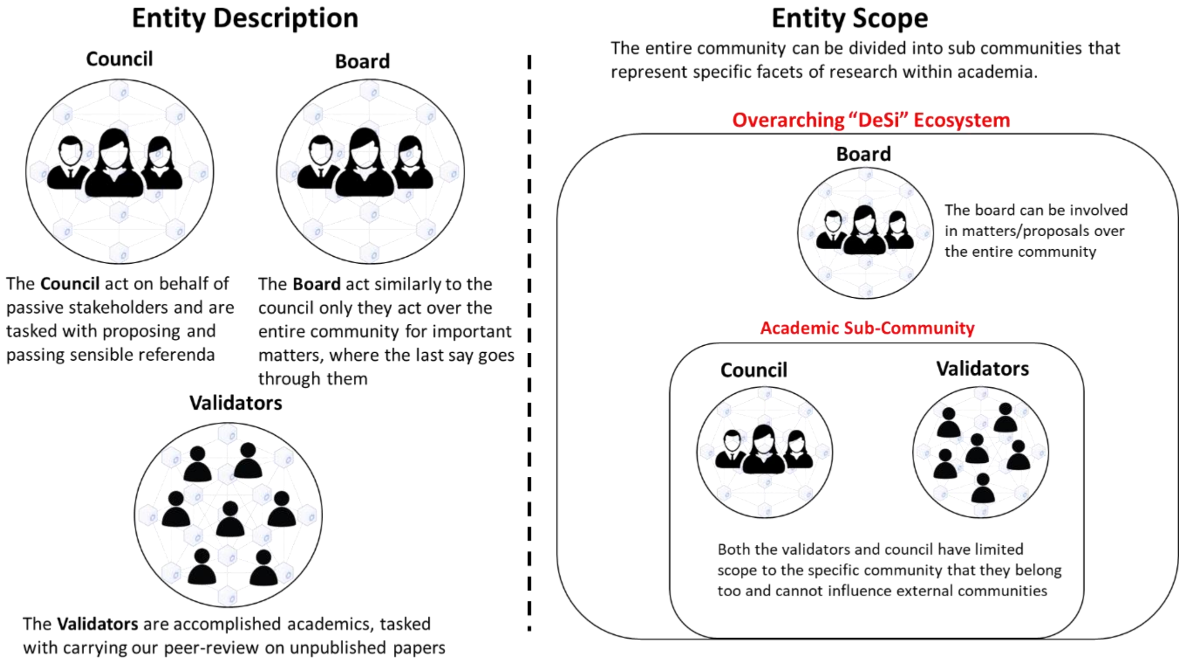

Entity Description
Council
Board
The **Council** act on behalf of passive stakeholders and are tasked with proposing and passing sensible referenda
The **Board** act similarly to the council only they act over the entire community for important matters, where the last say goes through them
Validators
The **Validators** are accomplished academics, tasked with carrying our peer-review on unpublished papers
Entity Scope
The entire community can be divided into sub communities that represent specific facets of research within academia.
Overarching "DeSi" Ecosystem
Board
The board can be involved in matters/proposals over the entire community
Academic Sub-Community
Council
Validators
Both the validators and council have limited scope to the specific community that they belong too and cannot influence external communities

Figure 4: Overview of different roles within our decentralized open-science platform

Treasury

The last important concept to discuss is the idea of the Treasury. From **Figure4** we can see that the treasury is a symptom of the governance protocol. Nearly All large POS blockchains will have some form of a treasury. The key idea behind the necessity for a treasury is the concept of sustainability. The treasury is a body of funds in which transaction (author) fees, slashing penalties, donations and funds from external funding bodies are funneled. The funds within the treasury are used then used to support the wider ecosystem so that it can be self-sustaining. The treasury is tightly linked to the governance model is because many proposals that get created can come in the form requests to spend the treasury's funds, be it to fund a particular scientific project, or to outsource work to external developers to implement new features on a technical level or conversely bug bounties to name but a few. There is a lot of economics and other complex decentralized finance concepts such as inflation and decentralized lending that are involved with both treasury spending and also in regard to its sustainability that are beyond the scope of this section. However, we will be exploring the inner workings of the treasury Later in this report in section 8.

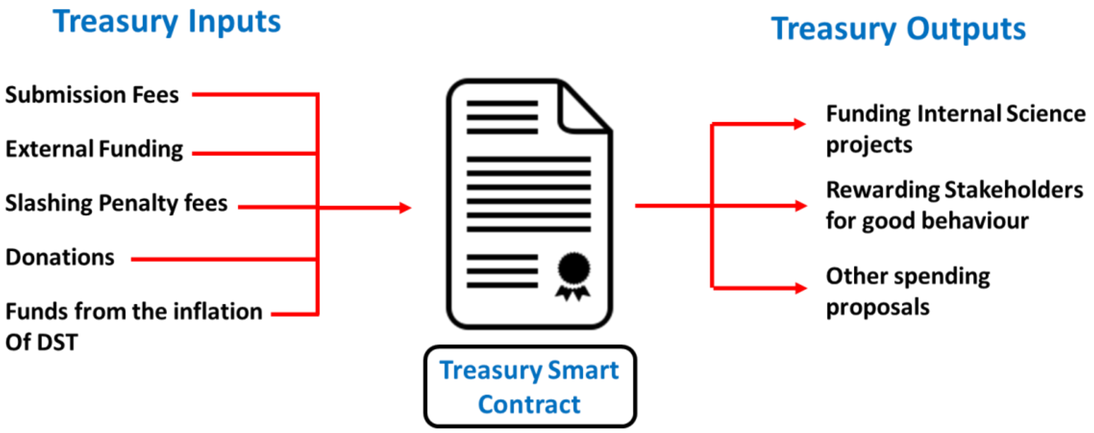


Figure 4.5: Demonstration Of the flow of money to and from the treasury smart contract

Section 3.4 Overview of Platform Design

We will conclude this section by examining in a little more detail by exploring how the technology architecture of our platform effectively operates in relation to the different technologies and frameworks that will be used to develop it out. This will be followed by a brief explanation of our smart contracts infrastructure which will be used to encapsulate the logic we want to capture through our consensus and governance protocols. To begin, we must understand that any web application has two parts. A backend and a frontend. The frontend is simple the user interface that the end users interact with to use the platform. The backend on the other hand consists of the application logic that governs how the app behaves whenever a user clicks on a button or does state changing action on the user interface. Traditional web applications are simpler to execute in nature in that their backend is relatively simple and usually would consist of a centralized database (MySQL) and a scripting language like Node.js & express.js to allow communication between the user interface and the database itself. Web3 applications built on the application layer of a blockchain are slightly more complex in their architecture. Both the frontend and backend components become increasingly more complex in comparison to their web2.0 counterparts. This is because on the frontend side of things the user now has to use a cryptocurrency wallet to use the app. The reason for this is because all web3.0 applications interact with the underlying Blockchain that their built on top of in the form of sending transactions. In order to send a transaction a user must pay transaction fees to facilitate fro the miners that are validating the transaction and appending its Proof-Of-Existence to the Blockchain. And thus every user must have a wallet in order to access the app so that that they can any sign transactions that they send and also to pay the underlying fees for doing so. However, thing set even more complex when we consider th backend of a web3.0 application. In the last scenario we only had to deal with a scripting language and a database. However, there are many components that make up the backend of a web3.0 app. They include:

- **Smart Contracts** logic for changing the state of the blockchain
- **Blockchain Explorer** for querying data in the blockchain ledger
- **Centralized Database** for storing insensitive data for better performance
- **Scripting Languages** to allow communication between backend & frontend

The smart contracts (see appendix) are the immutable pieces of code that are responsible for directly changing the state of the blockchain ledger. Smart contracts are by definition, what make web3.0 applications decentralized, because once a smart contract get deployed, its contents cannot be change, altered or modified by anyone and the code will continue to exist on the Blochian forever. In our platform all of the sensitive logic such as the governance and consensus protocols will be coded through a system of interdependent smart contracts. The other thing that we use in web3.0 applications is what's known as a block explorer. A block explorer is a tool that is used to query data from a blockchain. In other words, execute to "read only" functions. They are primarily used to query and call the various parameters in smart contract that may change over time such as the number of users stored in it or to get the token balances of each user in a smart contract etc. It might seem strange to use a centralized database as a viable method to store data using a decentralized app but it is a commonly employed method in industry and this is because of the fact that BT is still very much in its infancy, meaning that processing larger and frequent query's to the blockchain can become quite slow and induce large waiting times on the fronted side of things which ultimately leads to a bad user experience. For this reason, less sensitive data, such as a log of users and their transaction history for example, are usually stored in traditional centralized databases for quick access to display on the platforms user interface. The argument in favor of this technique is the fact if a scenario arose where a user's data became compromised for whatever reason, then a record of everything is still available on the blockchains ledger. So, in summary, as a decentralized application scales and becomes increasingly more complex, it becomes much less feasible to sustain the applications performance metrics without the implementation of a centralized database reduce the load on direct request from the blockchain. Now that we understand the typical layout of a web3.0 application we can look to the diagram below which shows the technology stack that will most likely be used to build our decentralized peer review platform and more impertinently how each process in the architecture fits together

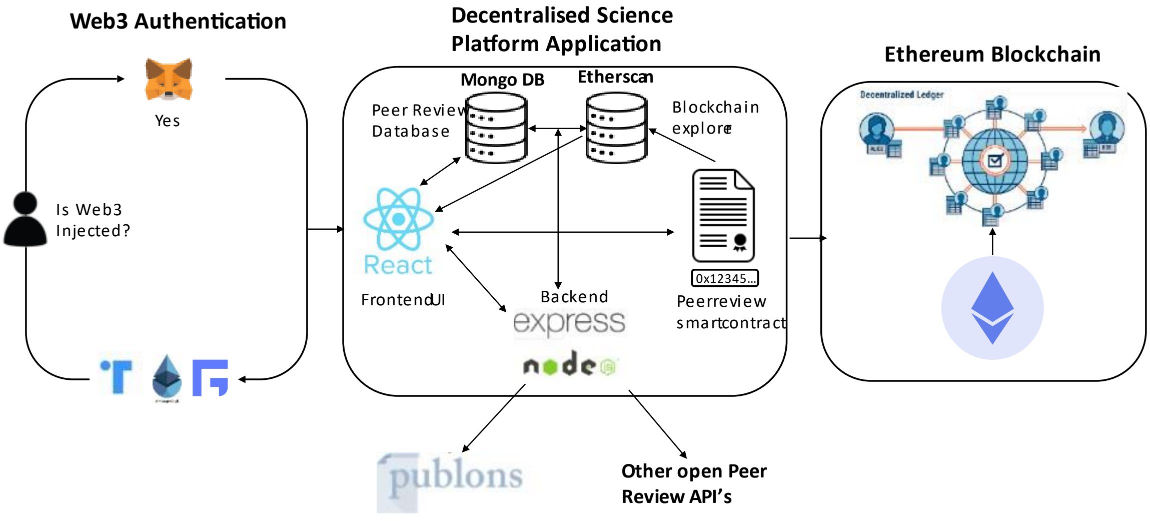


Figure 5: Platform Technology & architecture design

Figure 5 Shows the flow and interactions between all of the components that we discussed above. In order to use our application a user must own a web3 wallet (see appendix). Metamask is the common choice here and is the most popular web3 wallet to date. To handle authentication, we will employ the concept of digital signatures. The basic premise is that whenever a user tries to login they will be forced to do so with their wallet to confirm their identity.

Basically, this involves our backend code sending a unique message to the users wallet. If the user manages to sign this message with their private key (see appendix) then they have successfully confirmed their ownership of that wallet and will be granted access to the application. This type of authentication doesn't require the slow burden or username password login flows and is also cryptographically secure, by nature, we call it web3 authentication (see appendix for more). Once the user is successfully authenticated, they are free to use the application to look up papers or engage in community activities etc or vote on community initiatives. This will all be hosted by a fronted built in React.js, a popular JavaScript framework that is commonly used to build websites in industry. The react Frontend will use Node.js and express.js which are backend scripting languages that allow communication between our off-chain database (Mogo DB) and also with our smart contract protocols. We will also utilise node and express in order to fetch important researcher metrics such as an induvial **h_index**, for determining their on-chain reputation. Our entire application will be able to interact with the Ethereum Blockchain through a JavaScript Library called Web3.js which renders it possible to facilitate and execute Blockchain query's from the likes of etherscan (Ethereum's block explorer Figure1) which are needed to read data from our smart contract code to be used in a variety of different manners.

3.5 Overview Of Smart Contract Design

To conclude this section, we will outline a basic overview of our platforms smart contract design. Shown below in Figure is an abstracted version showing how the core smart contracts will most likely link together. We have three main contracts. The Journal, Paper and Reputation Contracts. Recall from section that we declared that our platform will be able to cater for the creation of sub-communities each unique to various facets within science. This is achievable through our Journal Contract. The idea here is that unique instances of this contract will get deployed whenever the creation of a new sub community gets created. Each journal contract has its own set of validators that can be reflected though governance and elected through consensus. The paper contract represents the submission of papers to our platform. Papers themselves are smart contract and by doing this we can store the authors, peppers reviews amongst other parameters all within the same body of code. Lastly the Reputation contract will also be unique to each journal instance, and this will store the on-chain reputation and token stake of each stakeholder subscribed to that community. It is important to note that the consensus and governate contracts are missing here for the of keeping thing simple, but the governance and consensus protocols would be parent contracts o each journal instance meaning that each deployed journal contract will inherit from the overarching consensus and governance contracts. This will be explained in more detail in the coming sections.

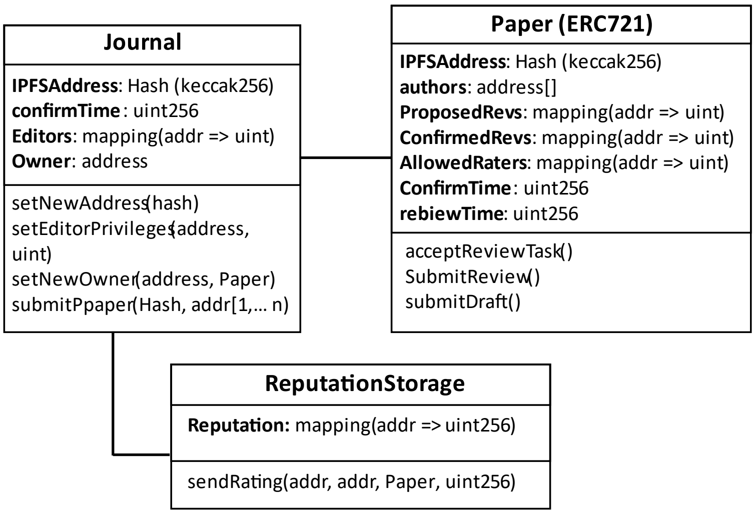

Journal
IPFSAddress: Hash (keccak256)
confirmTime: uint256
Editors: mapping(addr => uint)
Owner: address
setNewAddress(hash)
setEditorPrivileges(address, uint)
setNewOwner(address, Paper)
submitPpaper(Hash, addr[1,… n)
Paper (ERC721)
IPFSAddress: Hash (keccak256)
authors: address[]
ProposedRevs: mapping(addr => uint)
ConfirmedRevs: mapping(addr => uint)
AllowedRaters: mapping(addr => uint)
ConfirmTime: uint256
rebiewTime: uint256
acceptReviewTask()
SubmitReview()
submitDraft()
ReputationStorage
Reputation: mapping(addr => uint256)
sendRating(addr, addr, Paper, uint256)

Figure 6: Basic smart contract layout

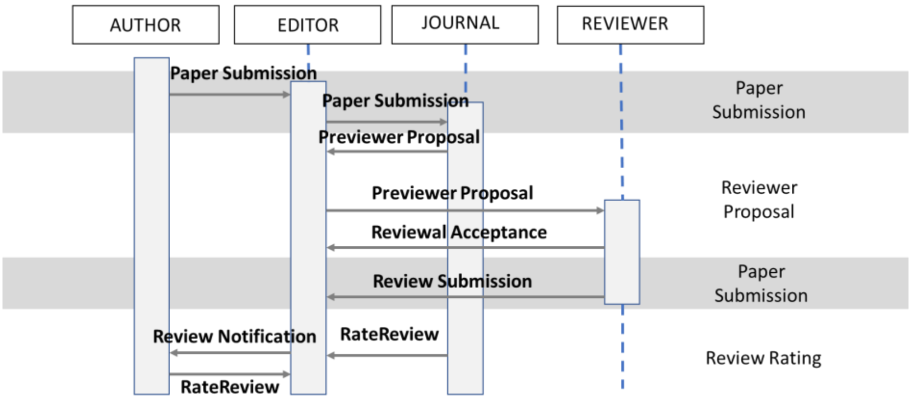


Figure 7: Flow diagram of the different events and states for a paper submission

Figure6 and Figure7 demonstrate the connection and dependencies of our core contract design as well as the different events and phases of a typical paper submission. Although there are more aspects of our smart contract design that are not captured in the above diagrams, such as flows to do with governance and consensus, the above do give an insight into the main stages of a typical submission/publication event which is largely the most crucial aspect within our platform.

Paper submission:

The submission process has three steps within the system. First, the paper is uploaded to the interplanetary File Storage System (IPFS) network, then the platform will recover the unique identifier of that paper through the hash of its content. This will be the papers IPFSAddress as seen in Figure. Finally, the platform (Journal Contract) will create an unique Ethereum smart contract containing the file address and the addresses of the authors, randomly selected, to record the submission on the blockchain. This creates a transaction in Ethereum that can be used to verify that the authors submitted the paper. Furthermore, this smart contract generates an Ethereum address that acts as a paper's unique identifier inside and outside the platform. This contract will be a special type of contract known as an ERC721 smart contract. These types of smart contracts are unique in that they represent non-fungible (non-divisible) assets. A non-fungible asset is just like any other type of asset in that it can be transferred and gain value over time. However, It differs from its fungible counterparts in that it is completely unique. A common analogy used to explain is that land is considered an asset, but each section of land is unique in that it has its own properties, this is in contrast to for example a currency which also has monetary value but currency can be split into many identical counterparts. ERC721 contracts are the basis behind NFT's. The implementation of our paper submissions as NFT's grants the ability for the author to earn rewards through citations over time. If the is well received and its citations gown, then the papers value grows too as a result. This mechanism allows us to create a mechanism that is able to read authors for producing good work. This is explained in much more detail in the next section.

Reviewer proposal:

Once an author or researcher successfully submits a paper for review our consensus algorithm will execute a WRS (Weighted random selection) algorithm which will randomly selected (n) validators to review the paper, creating a review task in the paper's smart contract. The transaction will record the Ethereum address of the reviewers and, optionally, a deadline to submit the review. The invited reviewees may accept or reject the review task (which will also be recorded into the blockchain). If the task is rejected, the editor can assign another reviewer. But conversely, they will loose out on the opportunity to earn a token reward for conducting the review. However, we do realize that in some scenarios a validator may not have the time to conduct a review (due to personal reasons) and the fore we think they should have the option to reject and the whim of suffering a penalty in the scenario where no rejection option is made available

Submit review

Once each of the validators carry out their review, they will each rank the paper by giving it a score of satisfaction. This ratification score will be used in other mechanisms that reside in the consensus protocol, such as the recalculation of reputation and rewards. To submit a review, the reviewers or validators must all research a consensus agreement that the paper is of good quality and should carry out a transaction that will record the acceptance/rejection and the IPFS address (i.e. the location) of the detailed review. In the event of a reviewer sending a review when the time has expired, a penalty is applied to the reviewer's reputation in the reputation system, and they will also lose their staked tokens based on the criteria discussed in section 8.

This section served as a gentle introduction into the main concepts and design of the various protocols that work in unity to define the various processes that are at work in our platform. Now that we have a basic understanding of things like consensus, staking/slashing and governance, as well as an insight into the proposed smart contract design that we will adopt to govern all of this logic, we can now begin to explore each topic in greater detail in the coming sections. The remainder of this report aims to dissect and expand upon all of the main ideas laid forth in this Section. The next section is concerned with the breakdown of our consensus protocol which is concerned with randomly selecting validators to carry our the peer review and also how rewards and or penalties are enacted for good/bad behavior within the system

4.0 Consensus Engine & Dynamic On-Chain Reputation

Arguably the largest and most important element of the system architecture for our platform is the concept of consensus. To achieve unanimity amongst all of the network validators (peer-reviewers) we will adopt proof of stake consensus (POS) as highlighted in **section 3.1**. POS based algorithms all break down to two main categories. Block production and finality. The consensus engine used to govern the automated selection of "peer reviewers" on our platform will heavily mirror some of the tried and tested features from that of a traditional proof of stake based system. In our model we will develop the platform on the Ethereum blockchain and create the application logic through a dense system of smart contracts whose functionality will mirror all of the features required to develop a decentralized and trust less application such as consensus and governance. Utilizing blockchain and smart contract technology, the proposed P2P publication model aims to bring two fundamental changes to the current publication model. The first is the introduction of a transparent and equitable recognition, rewarding, and responsibility-sharing mechanism between researchers and reviewers. For the P2P publication model, we aim to develop a vast ecosystem of individual journal entities as discussed in the last section. The main reason for this is to help access to a cross-discipline pool of reviewers from multiple journals. Since blockchain provides an indisputable log of events, the journals or publishing companies, forming a network at a global scale, would be able to detect multiple simultaneous submissions or at least have an indisputable log of previous submission. Furthermore, a consortium of journals from different disciplines can grant access to a cross-discipline pool of reviewers. A reputation profile will be issued for researchers and reviewers at the time of joining the network, both as a researcher and reviewer, namely, $Reseacher_{rep}$and $Reviewer_{rep}$. The profile score will fluctuate dynamically according to the earning/deduction of reputation during their lifetime of operating on that journal-specific network. Reputation is a critical component of a digital economy and projects like the "educational reputation currency, Kudos" introduced by Open University have already proposed the idea of generating educational currency using blockchain linked with academic achievements and credits. We build upon this notion in **section 7** (staking) on the implementation of open sourcing and crowdsourcing the scientific research process through the use of incentivised tokenomics (more on this later).

4.1 Understanding Block Production

Block production is the way in which new blocks (containing important data) get proposed and added to the immutable blockchain ledger. For our approach, We, are not constructing our own blockchain, but rather developing on the application layer of Ethereum or alternative EVM layer2 (see appendix) blockchain and developing a system of smart contracts to mimic the lower level features of blockchains such as things like consensus, governance block production and finality in our platform architecture. This is because, the original idea was to have researchers and scientists in the academic space make proposals of intellectual property (IP) such as research papers, scientific patents etc, that will ultimately form the contents of the blockchain ledger. However, in this scenario, to append large files such as academic papers to on-chain storage is far too costly as the size difference between a research paper and a typical transaction on the likes of bitcoin and Ethereum differs on the order of one magnitude

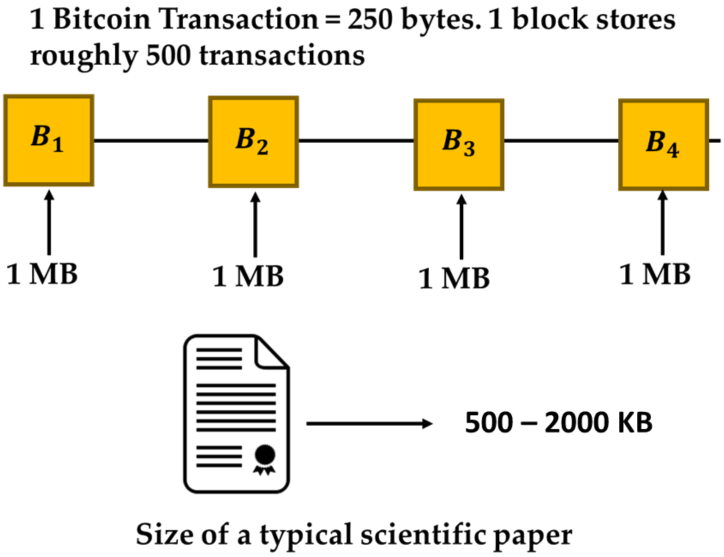

1 Bitcoin Transaction = 250 bytes. 1 block stores roughly 500 transactions
B_1
B_2
B_3
B_4
1 MB
1 MB
1 MB
1 MB
500 – 2000 KB
Size of a typical scientific paper

Figure8: Size Comparison Between A bitcoin TX & A scientific paper

Thus, appending actual scientific research it its raw form would be far too costly and not maintainable as the blockchain grows over time. This is because in order to secure any blockchain network, each node needs to have a full copy of the entire ledger. Thus, dramatically increasing the storage requirements that a node would have to meet in order to run the blockchain on its software and this would have the consequence of dramatically increasing the centralisation which in this case defeats the purpose of our goal. This is further accompanied by the fact that most academics are not going to want to run blockchain mining software on their personal computers. Thus, for our approach we devise a model that simply uses smart contracts to mimic the block production process of a typical POS based blockchain.

One other issue with developing our own peer review academic reduces to the idea of empty blocks. On blockchain like bitcoin or Ethereum blocks are mined every could of seconds/minutes. On Ethereum its roundly 6 seconds and on bitcoin the block time is roughly 10 minutes. Both Bitcoin and Ethereum have one thing in common in that they have millions of active users constantly submitting transactions in the form on transfers and payments etc. However, if we consider our scenario where our ledger would consist of transactions defining the publication of intellectual property (IP) such as a scientific paper or article. In many cases our platform would not be getting 3000 papers published every 10 minutes (TX count in a typical bitcoin block) even with mass adoption amongst the broader scientific community. Thus, the idea of an underlying blockchain that requires validators to gather publications to append to newly minted blocks would be inefficient in our scenario. To explain this, we can take the idea of how a typical POS consensus algorithm works. Every time it comes to mine a new block a random selection algorithm randomly selects a validator from the validator set with probability (p) (based off of their total locked stake) to gather the transactions and sign off on the new block. In such a scenario as ours where the likelihood of having empty blocks would be very high because the volume of papers getting published in the network would be quite low. Therefore, we would have a situation that look something similar to the diagram below

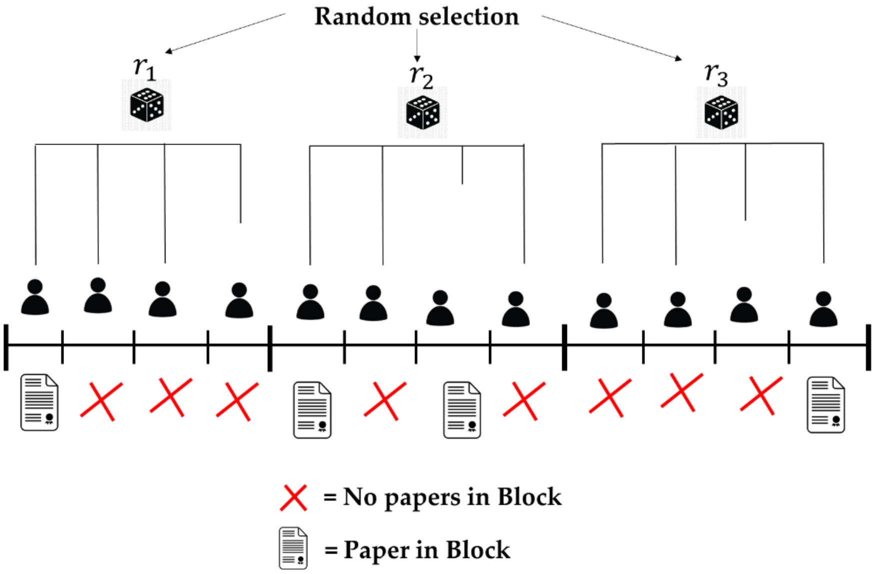


Figure9: Demonstration of Empty blocks due to lack of network adoption (as a result of the slow turnout of papers)

In **Figure9** we can see that validators are randomly getting selected to mint new blocks. However, in many cases due to the inherent low volume of the platform many blocks don't have any contents for the validators to mint. This would waste computation whilst simultaneously forging a blockchain with a high percentage of empty blocks which is undesirable. Blockchains with such traits are commonly referred to as GHOST chains. The lack of data in the blocks would eventually over time discourage validators from locking up their stake as the opportunity for rewards on GHOST chains is much inferior to a constantly active blockchain Therefore, the solution that we propose is not to have a constant fixed time interval in which new blocks are minted but rather to have the time in which blocks get minted dynamically fluctuate depending on how many publications there are in the network which are pending reviewal. When there is at least $(n_{pending} = 2)$ papers, then the Random selection process will begin, inviting a sample of validators to carry out the reviews. We do note that as our platform becomes more active over time this number will need to be increased.

4.2 Network Participation Requirements

In order for the general public and scientific to participate in contributing to the network they must have skin in the game. The entire application is governed through on-chain reputation and network stake (see section 11). This means that whenever someone wishes to submit scientific work to the network for publication, they must pay a small fee to the form of an ERC20 token DST (Decentralised Science Token). On such an event we will first adopt a Proof Of Existence approach whereby the contents of the current work get hashed and submitted to IPFS (Interplanetary File System). This hash will then be able to map back to the accountID of the author and also to the timestamp that the moment of creation that the work was submitted at. Also, at this time an smart contract event will get emitted which will be indefinitely stored on the underlying blockchain which will include other useful information about the submission transaction. The file that gets submitted will be stored off-chain on IPFS to solve the storage issue that we mentioned above in **section 4.1**. Initially the file will be in a "pending" state. In other words, it will not be readably available for public viewing on the platform by the general public because it has not yet been reviewed and accepted by the network validators. However, it is important to note that although the scientific content is not directly available on the blockchain at this point, all information associated with the actual event IS on the blockchain in the form of the fired event. This way we always have proof to query whenever someone makes a proposal. So, this is in essence how out platform will get populated with papers. Notably form repeated occurrences of this proposal event. The next important thing to look at is the question of how are validators selected and how do they arrive at consensus in terms of agreeing on what proposed work should become final and get published to our platform for public consumption? This is not a difficult issue to address (by traditional POS conventions).

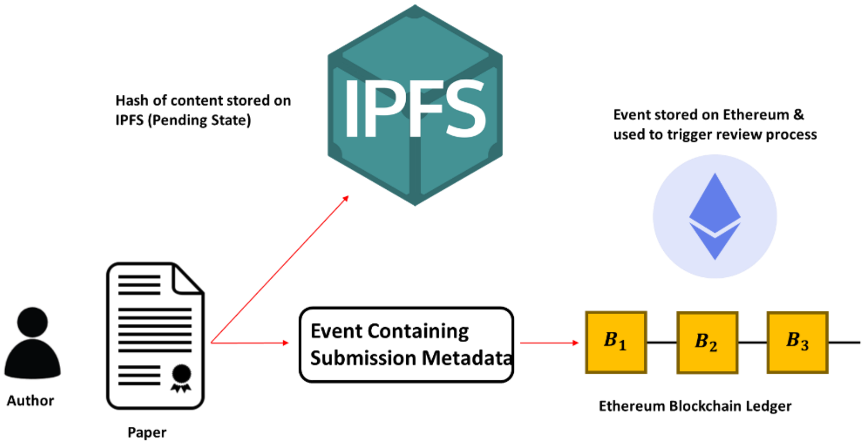

Hash of content stored on IPFS (Pending State)
IPFS
Event stored on Ethereum & used to trigger review process
Author
Paper
Event Containing Submission Metadata
B_1
B_2
B_3
Ethereum Blockchain Ledger

Figure10: Initiation of a paper submission

Our underlying consensus protocol is underpinned by the validator set. The validator set is a dynamically and constantly evolving array of individuals who are tasked with the on-chain peer review process. The validators are network participants who are considered to be "accomplished" academics and are voted into the position by the our protocols governance model (see section 6). Each validator has two types of network stake that is used to calculate their overall worth in the network. Validators a required to stake digital assets in the form of (DST) and reputation in order to be incentivised to contribute fairly to their tasks in the network in fear of losing some of their collateral. These assets are in the form on monetary assets and reputation (see section 5 on staking). We also define an on chain weighted random selection process (WRS) to fairly select a sample of validators based off of their work or contribution in the network to carry out the market-based peer review process. The WRS algorithm (explained in detail below) uses a random generator function to select a subset of validators from the validator set to carry out the peer review

4.3 Random Selection Process

The most important feature of our consensus mechanism is the way in which validators are randomly selected to carry out the peer review process. As mentioned above the mechanism that we develop is heavily based off of already existing proof of stake algorithms seen in blockchains such as Ethereum and Polkadot. This is, that each validator is ranked based off of their stake in the overarching network and through their ranking they are periodically selected in the securing of the network. The biggest point of failure in many proof of stake algorithms is the random selection process. How can we make sure that the way in which we select validators to review a paper is fair and well incentivized? The way that our consensus protocol works is that we have an internal application timeclock. This timeclock is split up into periods of time known as epochs. Epochs themselves do not have a fixed length of time but rather an they are subdivided into (n) slots where $(n = 10)$. Each one of these slots represents a period of time when our WRS algorithm is called, randomly sampling a group of validators to review pending submissions. So, after 10 intervals of the execution of the WRS algorithm, a new epoch will begin. At the beginning of each new epoch a random seed denoted by (τ) will be calculated as the range spanning from $0, \tau$, in which the validator will be assigned weights over (more on this in **section 4.3**)

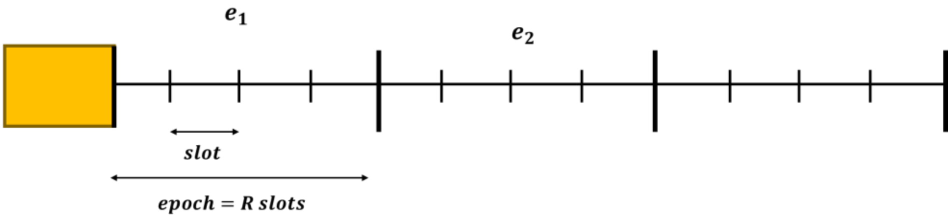


Figure11: Demonstration of epochs as an internal time clock (used to determine range for WRS algorithm)

To decide when the WRS selection algorithm is fired we have an internal off-chain function that runs indefinitely on a fixed interval. The purpose of this function is to check for any "pending" papers that are awaiting review. Whenever the number of pending submissions is equal or greater than two such that:

$$\sum P_{pending} \geq 2$$

$$Where\ P_{pending} = Pending\ Submissions$$

Then the WRS algorithm will be executed, randomly selecting (n) validators to carry out the review. The definition or magnitude of (n) is not constant and rather is determined based off of the size of the validator set for that specific scientific community. This means that the more the network grows, the more validators there are tasked in reviewing the submission, increasing the validity of the review. The way in which the WRS algorithm fairly selects the reviewing panel is based off of assigning a weight to each validator in the overarching set which is based off of their stake in the network. Each validator stake is divided into two sub parameters. That is their reputation score and their stake of DST in the network.

$$Reputation_{total} = 30\%_{token-stake} + 70\%_{reputation}$$

However, the reputation score of each validator is twofold. We identified a problem with defining only and on-chain reputation score. If we consider the scenario where two researchers join the network at the same time. The first researcher, $Researcher_A$ is an accomplished academic and well renowned in the traditional system. On the other hand, $Researcher_B$ is a fresh postgraduate who has not yet gained a decent reputation for himself in the traditional system. Thus, in the case where we propose ONLY an on-chain reputation, meaning that both $Researcher_A$ and $Researcher_B$ would be initially assigned an on-chain reputation score of zero. This obviously shows a huge flaw in the system as $Researcher_A$ should be ranked higher. But due to the nature of smart contracts there is no way for us to be able to account for the reputation of a researcher off chain in the traditional system. Thus, to overcome this we have implemented he idea of an oracle which is able to use the **h_index** ranking system to fetch data on each existing reputation of a given researcher/scientist. Thus, we define the on-chain reputation for a given validator as a weighted sum of both there on-chain and off-chain **(h_index)** score. Consider now

$$Reputation_{total} = 30\%_{token-stake} + (100\%_{h_{index}} + 0\%_{on-chain-rep})$$

$$Reputation_{total} = 30\%_{token-stake} + 70\%_{reputation}$$

$$Where\ 70\%reputation\ =\ (100\%_{h_{index}} + 0\%_{on-chain-rep})$$

A flow diagram of the total calculation of a validator's reputation score is show below in Figure8. The on-chain reputation is determined by a smart contract and the off-chain **h_index** weight is determined by an oracle that we define which takes the average **h_index** score for a particular validator from querying multiple scientific API's such as Dimensions Analytics and a variety of others. This is done in order to obtain the best truth in relation to a given validators off-chain reputation ranking

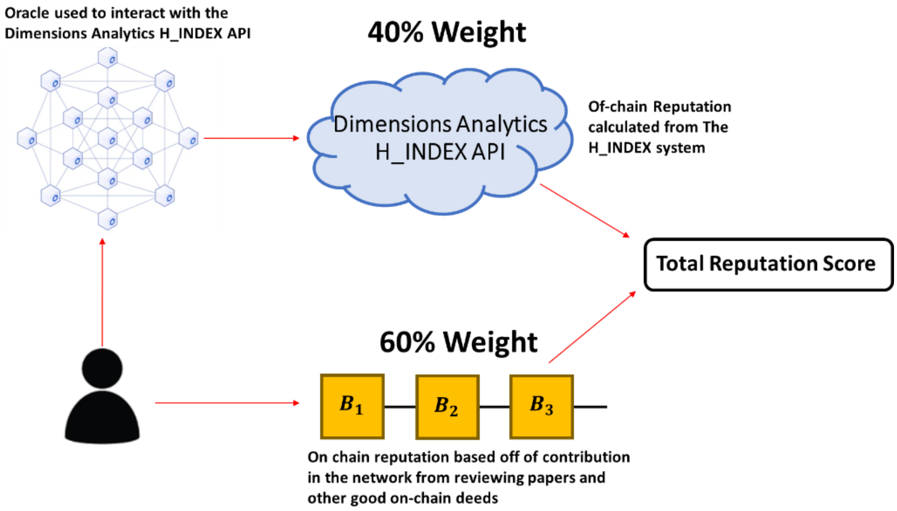

Oracle used to interact with the Dimensions Analytics H_INDEX API
40% Weight
Dimensions Analytics H_INDEX API
Of-chain Reputation calculated from The H_INDEX system
Total Reputation Score
60% Weight
B_1
B_2
B_3
On chain reputation based off of contribution in the network from reviewing papers and other good on-chain deeds

Figure12: Demonstration of the calculation of a researchers Total Reputation score

One final important thing to note is the evaluation of the reputation score. The idea of initially including a weigh for a researchers **h_index** evaluation in the calculation of the on-chain reputation is to make the random selection process more fair to newcomers in the network. However, with the growth of the platform we want to eventually phase out the need for the reliance of a researchers **h_index** score. Therefore, we also introduce the idea of a time bonding curve. The idea of this is to eventually phase out the need for a user's off chain reputation in the determination of their worth in the network as time progresses. Whenever a validator joins the network, as seen above their off-chain reputation will account for 100% weight of their total reputation. However, through the implementation of the time bonding curve the percentage value of their -off-chain reputation weight will decrease slowly until eventually only their both their off-chain **h_index** rank and on-chain reputation rank will have equal weights of 50% each.

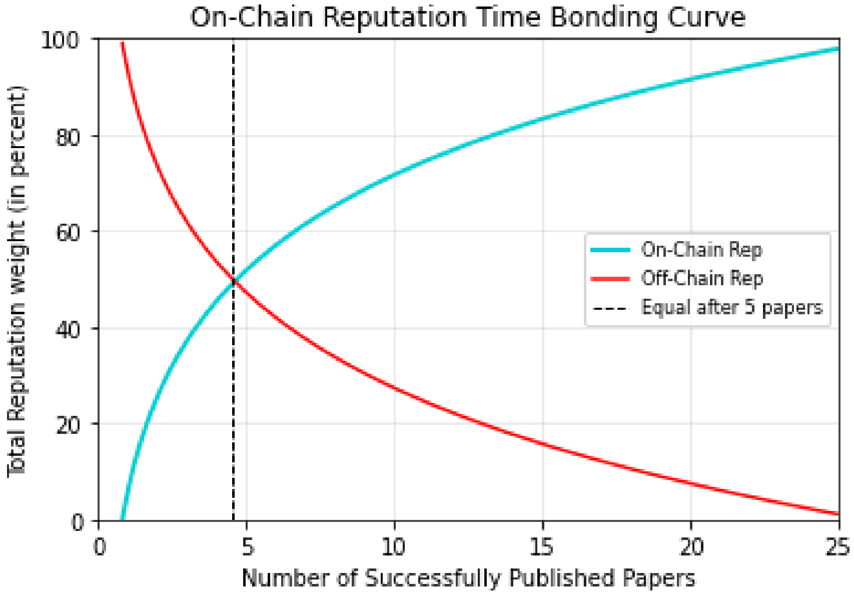


Figure13: Time bonding curve for the readjustment of a researchers total reputation score

Once we have the total stake a validator has locked into the network along with their total reputation score. Then we use substitute these values into the formula for $Reputation_{total}$ above to calculate their weight for the WRS selection process.

4.4 Weighted Random Selection Algorithm

Now that we understand how Each validators' worth in the network is calculated, we can now define how the provably verifiable weighted random selection (PWRS) process works. Essentially each validators weights are assigned to an array. Their weights are distributed from $0, \tau$. Here (τ) is the random seed. The value of (τ) is not constant and is re-calculated at the beginning of each epoch. The definition of (τ) is such that it is the SHA_256 hash of the total locked stake in the network (S_{e_j}), the total number of members in the validator set (n_{e_j}) and the value of (τ) from the previous epoch.

$$\tau_{e_j} = SHA\_256\,(S_{e_j}||\,n_{e_j}\,||\tau_{e_{j-1}})$$

The reason that we define the random seed as such, is that it boasts more security as the seed is constantly changing as time progresses in the network as it is constantly dependant on the number of active validators and the total stake locked into the system at any given time. The seed is how we can ultimately prove that the random number chosen is authentic

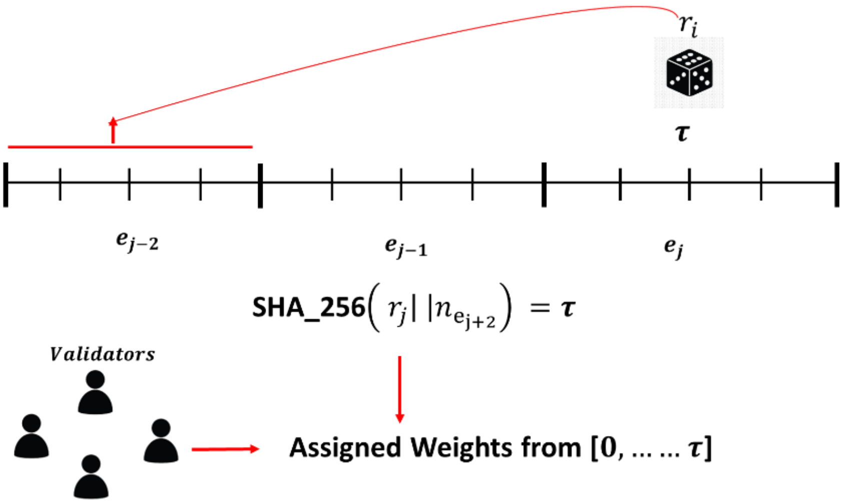

r_i
τ
e_{j-2}
e_{j-1}
e_j
SHA_256$\left(r_j||n_{e_{j+2}}\right) = \tau$
Validators
Assigned Weights from [0, … … τ]

Figure14: Demonstration of the calculation of the value of (τ) which is used as the range for the weighting of the validators probabilities for the WRS algorithm

Once the validators weights have been calculated and distributed from $0, \tau$ athe WRS algorithm is called to random select a subset of (n) validators to carry out the peer-review on pending papers. If we let (ε) be a random variable distributed over the set $\{a_0, a_1, \ldots\ldots, a_{n-1}\}$ whist corresponding probabilities $\{p_0, p_1, \ldots\ldots, p_{n-1}\}$. A fast and simple method for generating sample values for (ε) has been described by many people. The method that we adopt is adapted from this paper **[45]** and is constant in time. It is extremely important that we employ a quick algorithm such as this as the size of our validator set grows over time. We set out to produce a set of sample values in time proportional to the sample size. This means that (ε) randomly selected validators will be invited to review pending submissions and the larger that the validator set (n) becomes, the larger our sample size (ε) and thus the more confident we can be in the quality of the review. The algorithm that we define is equivalent to producing two sub algorithms that run in parallel, namely the $init$ and $rand$ methods which both share and satisfy.

- The input to $init$ is an array denoted (p) representing a probability distribution such that:

$$p_j \geq 0 \qquad and \qquad \sum_{j=0}^{n-1} p_j = 1$$

- The effect of the $init$ is the initialization of the $rand$ method to a function of no arguments (the behaviour of $rand$ depends only the internal state) which returns an integer (j) from the set $\{0, \ldots\ldots, n-1\}$ with probability (p_j).

If the array (a) contains the range of (ε) such that the probability of $(\varepsilon = a_j)$ is (p_j), then a sample value for (ε) is obtained by (a_{rand}). We also assume the existence of a function $uniform(n)$ which returns a sample value for a random variable uniformly distributed over the real interval $[0, n)$ which returns the floor of its argument in constant time. Our description of $rand$ follows that given by Knuth **[46]**. We let $prob$ and $alias$ be arrays initialised by the $init$ method described above. The body of $rand$ is defined in pseudocode below:

```
U =uniform(n)
If (U - j) 5 $prob_j$ then return
j else return $alias_j$.
```

Figure15: pseudocode for the $rand$ method (adapted from **[45][46]**)

The above algorithm executes in constant time. Our version of $init$ proceeds in two stages. The first stage divides the indices of the input into two arrays, small and large via the rule

$$p_j > \frac{1}{n} \quad => \quad j \in large$$
$$p_j \leq \frac{1}{n} \quad => \quad j \in small$$

The second stage uses the probability distribution of (p) together with $small$ and $large$ to initialize the arrays $prob$ and $alias$. The idea behind this stage is motivated by an analysis of $rand$. There are two situations in which $rand$ returns (j).

- $j = [u]$ and $(u - j) \leq prob_j$ then (j) is returned. This situation occurs with probanility:

$$\frac{1}{n} prob_j$$

- If $i = u[j]$, then $(u - i) > prob_i$, and $alias_i = j$ then (j) is returned. This situation on the other hand occurs with probability:

$$\frac{1}{n} \sum_{\substack{i=0 \\ j=alias_i}}^{n-1} 1 - prob_i$$

If we first suppose that $j \in small$ and $prob_j$ were np_j. If every entry of $alias$ is a member of larger, then only the first situation described above can occur. Hence the $rand$ method returns (j) with probability $\left(\frac{1}{n}\right)prob_j = p_j$, as required. If we then suppose that (k) $\in large$, and that when the assignment of $prob_j = n_p$, was made for the previously considered $j \in small$, the entry $alias_j$was also defined to be (k). Then in this case the $rand$ method could return (k) with probability $\frac{1}{n}(1 - prob_j) = p_j$, which is a term of the second situation. If $p_k = p_k - \frac{1}{n}(1 - prob_j)$, we could integrate these two procedures after reclassifying (k) to being small of large. This again, is described in the pseudocode below.

```
1 = 0;s = 0
For j=O to n-1
if $p_j$ > i
then largel = j ; 1 = 1 + 1
else small, = j ; s = s + 1
While s # 0 and 1 # 0
s = s - 1 ; j = small,
l = I - 1 ; k = largel
probj = n *pj
aliasj = IC
if pk > 1
then large1 = k ; l = l + 1
else small, = k ; s = s + 1
While s > 0 do s = s - 1 ; probsmallS = 1
While 1 > 0 do 1 = 1 - 1 ; problaTgel = 1
```

Figure16: pseudocode for the WRS algorithm (adapted from **[45][46]**)

The $init$ method runs in $O(n)$ time. The first loop cycles (n) times and the second loop decreases $(l + s)$ on ech teration, and initially $(l + s = n)$. The last two loops complete this decrement of (l) asnd (s) to 0. This is on a high level the theoretical implementation of the entire WRS algorithm for the sampling of validators for invitation to review pending works. Each validator has a has dedicated amount of time in which they must complete their individual review, grading the work based off a set criterion determined by the community via the platforms governance protocol (see section on governance). If the validator for whatever reason fails to complete their review within the allocated time they will lose a small percentage of both their token stake and reputation. More on this will be discussed in section on staking. Once each validator reviews the work, they give the current submission a score and based on the total average of all the individual scores the work will either be accepted into the network via an emitted transaction or rejected and an event will again be emitted to the blockchain containing metadata about the review details and also showing the new updated status which is binary in nature (accepted or rejected). Once the entire procedure is complete, each validators reputation will be recalculated. The dynamics of this process are described below. A low level GO implementation of this algorithm is given in the appendix

4.5 The Review Process & The Dynamic Adjustment of Reputation

So, until now we have identified that for researchers, reviewers, and readers, the P2P publication model introduces a scoring-based schema which is heavily inspired from this paper **[47]** where we build upon the initial design and tweak the initial formula for our specific application. Upon successful completion of a unit of work for a preregistered experiment, the researcher will upload their work to a decentralized storage system such as IPFS and broadcast a transaction on the journal specific network, which includes the metadata of the research publication. Subsequently, the journal will be notified and after review, the smart contract will be invoked. And based off of our WRS algorithm N reviewers will be invited to review the work. On important concept we will conclude this section with, is how the outcome of the review process updates and recalculates each reviewing validators reputation. Based off of each reviewers weight the smart contract will allocate a total of λ percent of authorship to the reviewers, giving each reviewer a λ/N percentage of authorship for that submission, regardless of the outcome of the submitted work (see Figure 1). This will encourage the reviewers to act in good faith as rejecting or accepting the submitted work bears no difference in benefits. Like the traditional approach, each reviewer will review the work and provide feedback to the researcher in a transparent manner, along with their affirm decision. In addition to the feedback and decision, the reviewers will also provide a satisfaction score S for the reviewed work, which will be used to apportion the authorship share between researcher and reviewer(s). This step can be iterative if the feedback is for modification and improvement purposes. For each iteration, an average submission score denoted $Submission_{score}$ will be calculated as we described above where it is the average score submitted by all reviewers.

$$Submission_{score} = \frac{\sum_{n=1}^{N} S_n}{N}$$

This feedback will facilitate researchers to improve the quality of their work and averaging the $Submission_{score}$ will safeguard researchers from any form of reviewers' bias or abuse. Once the reviewers reach a consensus on the acceptance/rejection decision, the smart contract will publish the submission along with a feedback thread. Each reviewers on-chain reputation score will be re-calculated accordingly

$$Reputation_{on-chain} = \frac{\lambda(submission_{score})}{N}$$

For example, in the case of where each validator receives a 33% reviewer's share of the review and from three reviewers the paper received an average of 8.5 $submission_{score}$, then each reviewer will get $Reputation_{on-chain}$ score of 0.935 [(0.33 *8.5)/3]. Since reviews could vary significantly in length and depth, it is important that the reviewers are rewarded fairly for their contribution. Another notion that **[47]** introduces is the fact that the value of reviewer's feedback is gauged by allowing readers to rate the reviewer's comments, recorded as a documentary of the work. This rating range (R) (with positive/ negative value) will help to gauge reader's perception toward the rigor of individual reviewer's feedback and will be used as a weight to calculate individual reviewer's reputation, $Reputation_{on-chain}$. Even though initially each reviewer was allocated a λ/N share in ownership based of the ranked weight of each reviewer in the reviewer array, the readers eventually decide the reviewer's stake in the published submission, based on their contribution to the review process. To ensure fair usage of the system and avoid spams, we propose that only the readers registered with the network should be allowed to rate the reviewer's work. Similar to the reviewers and researchers, readers (who are classified as the general public that at least own some DST) will be assigned a reputation score which will change over time based on the reader's contribution to the network. A fraction of reader's repute (r) can be used to determine the weightage of the rating (R). Readers with higher community contribution will have a higher weight on the reviewer's score as compared to a reader with little or no contribution to the publication network. This will create a community-governed ecosystem such as the likes of Reddit and Stack overflow.

$$Reputation_{on-chain} = submission_{score}\left(\frac{\sum_{i=1}^{N}\left(R * r^{\delta}\right)_i}{n}\right)$$

The $Reputation_{on-chain}$ of the $(N = 3)$ reviewers, having Revunit score of 0.935, receive an average (R) rating of -1, 0.5, and 1 respectively. Thus, their rev can be calculated as

$$Reviewer1\colon 0.57 \times -1 = -0.57$$

$$Reviewer2\colon 0.570.51 = 0.285$$

$$Reviewer3: 0.57 \times 1 = 0.57$$

respectively. It can be clearly seen from the example above that the neutral readers can severely impact the overall share or contribution score of the reviewers, encouraging them to provide constructive and helpful feedback while simultaneously discouraging them to sign up for the review process with a sole objective of gaining $Reputation_{on-chain}$. This social assessment approach can minimise or potentially eliminate the current asymmetrical decision-making environment where critical and trivial reviews often bear the same recognition **[48]**. This will also help readers to get a critical understanding of the submission and may form the basis of a crowdsourced double reviewing approach (reviewing the reviewers). This $Reputation_{on-chain}$ score will evolve as more readers read and rate the submission of publication, causing the reputation of the individual reviewer to increase or decrease over time thus having the after effect or either minimizing or maximizing their chances in being selected by the WRS algorithm as a future reviewer. This dynamic feature of the algorithm will make the reviewer accountable for the lifetime of their review transaction history. The smart contract will also assign a $Researcher_{score}$ to the researcher at hand:

$$Researcher_{score} = Submission_{score} - \lambda(Submission_{score})$$

This total $Researcher_{score}$ (in this example will be $8.5 - 0.33(8.5) = 5.70$.) will further be divided within the contributing researchers defined by the smart contract for that paper.

For researchers, a separate score, $Cit_{unit-score}$ score that captures direct and indirect citations of the publication, will be created. This also incentivizes researchers to collect rewards by publishing high-impact work on the proposed network and built reputation in an indirect way instead of the current direct citation metric. The equation below represents the $Cit_{unit-score}$ score in the form of direct (β) and indirect citations (Φ). For indirect citations, a cap (e.g., three levels deep) can be introduced in the smart contract.

$$Cit_{unit-score} = \sum_{i=0}^{n} \beta_i + \sum_{i=0}^{n} \Phi_j$$

A publication with 10 direct and 50 indirect citations can have $Cit_{unit-score}$ score of 12.5 (indirect citations having a smaller contribution to the $Cit_{unit-score}$score). Like reviewer's reputation, researcher's reputation will also be weighted by the number of direct and indirect citations and dynamically updated over the lifetime of the journal. Finally, following the same example, $Reputation_{utationon-chain}$ can be calculated as $6.7 + 12.5 = 18.1$. (for more details of this process see, the original algorithm which is modified and adapted to our needs from **[47]**). This is a very trivial example showing how the review process is handled and how all parties in the network, namely reviewers, researchers and passive readers are infertilely incentivised to produce the most quality work and content possible in return for a reward in the Form of DSP or conversely for the loss of their staked DSP. For more details on how a users on-chain reputation determines the amount of token reward that they are entitled too see section5 on staking and slashing. With this we conclude this section on consensus. It must be pointed out that many of the algorithms discussed here are purely theoretical at the moment and certain parameters and formulae may change when taking this concept to production. In the next section we will introduce the staking and slashing mechanism which is the protocol which determines how tokens reared are distributed to network participants who are actively contribute to the network and conversely how penalties are issued to bad actors

The concepts and dynamics of our consensus mechanism discussed in this section are fairly complex and there are many different moving parts that combine together to define our model. However now that we understand the basic flow from the selection of validators through the WRS algorithm outlined in **section 4.4** to the readjustment and re-calculation of the validators' and researchers' reputation score, we can now move onto the next section where we will explore the concept of staking and slashing in further detail to understand on a fundamental level how stakeholders within the entire community are incentivised to contribute to the nest of their ability in order to receive increasingly higher rewards for exceptional community involvement.

5.0 Staking & Slashing Mechanics For Incentivised Good Behaviour

Now that we've discussed the main features of our consensus protocol, we can explore the concept of staking and slashing in greater detail. Both concepts staking and slashing underpin the entire consensus system and are used as tools to incentivise users in the network to act appropriately in return for being rewarded whilst conversely being punished for malicious behaviour. The idea of staking and slashing is a fundamental concept in blockchain architecture and it's a system designed to make the risk outweigh the rewards when it comes to trying to cheat the system. As we discussed above in section 4 above, in order to participate in the network, a user must stake both a monetary asset in the form of DST and also their reputation. It is by having so called, skin in the game that users can either be rewarded for good deeds or punished for mal intent or negligence. Any stakeholder is subject to slashing for a variety of reasons that we will discuss below but earning rewards for good deeds is mainly limited to researchers (i.e creators of knowledge) and validators (peer-reviewers). Although researchers, don't have as much of an opportunity to earn token rewards as validators, they can still be regularity reward on successful publication of a work through both their on-chain reputation and citation score which we defined above. For all community members their exists different bands that people can fall under based on their total "net worth" that earn them more rewards for contributions. It goes without saying that the higher your value in the community the more rewards you can potentially gain for completing good deeds, the same goes for punishment. The higher your value the higher the severity of your punishment if you get slashed and conversely the higher rewards you can potentially earn. In this section we will go through each scenario explaining how one can calculate how much they will earn or loose based of their on-chain reputation.

5.1 Staking Mechanism

Any members of the community who are active academics, in that they are constantly researching in their facet of science and publishing papers can mainly earn rewards every time they successfully publish a paper. We know from the last section, that (n) validators from the validator set will be randomly selected based off of their reputation weight via our WRS algorithm to review the work. On consensus the average score that the validators give the paper is used to recalculate the citation reputation for the author. Based on this factor, as well as their on-chain reputation and total amount of staked DST, the author will be rewarded for their contribution to the network. The formulas for the reward that both researchers and validators author earn is based on the below formulae. First, we ned to decide what band the author falls into in comparison with all of his peers in the network. There is 5 bands. Each zone is split into 20% increments. At the beginning of each band the rewards reach a triple multiplier. This notion is captured in **Figure17.** However, researchers & validators who fall within the same band do not earn the same rewards. The rewards increase on a log scale from the minimum multiplier possible (0.003) to the largest multiplier (7.29%). Each researcher/validator will be subject to a different reward based on their percentile ranking. This is illustrated better in **Figure18.** It is at the boundary between bands that the rewards multiplier from the beginning of the band prior will reach triple from its last value. This has the effects of dramatically increasing the rewards multiplier for ranking higher in the whole system, encouraging researchers to constantly try to perform at the top oof their game.

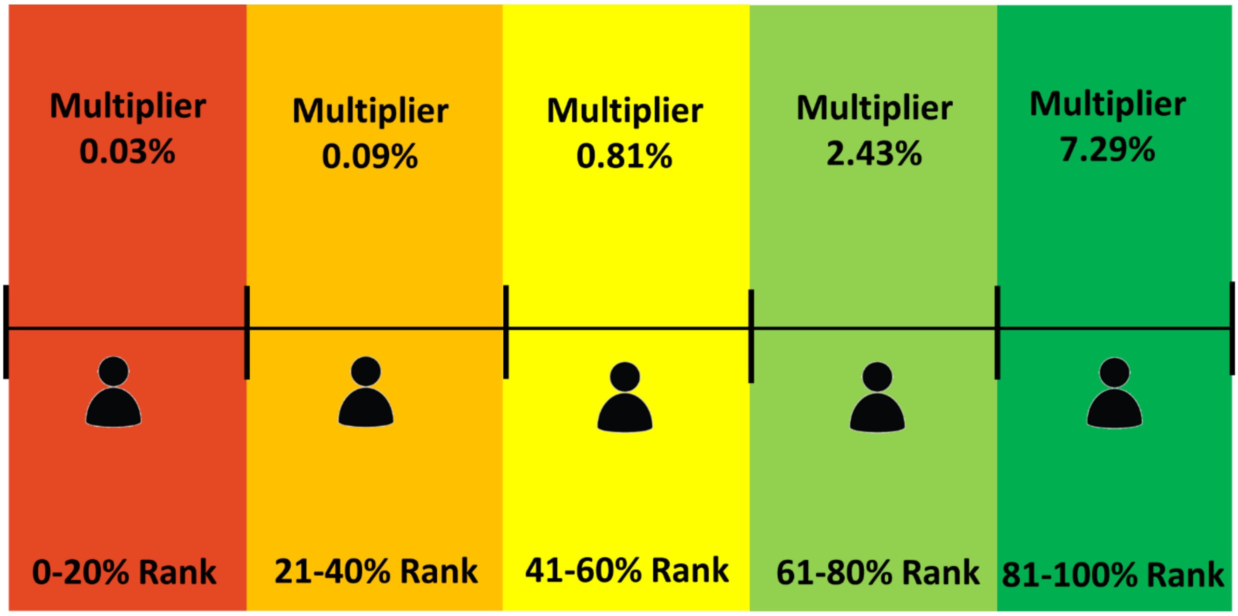


Figure 17: Staking rewards multiplier categories for researchers based on their rank in the network

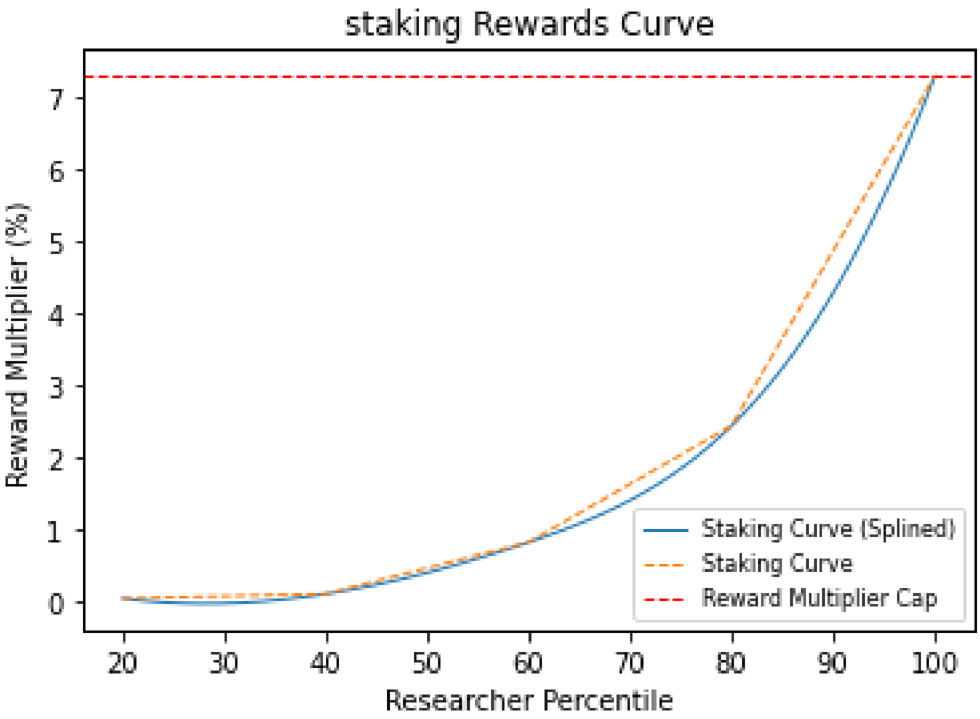


Figure 18: graph demonstrating the dramatic increase of reward opportunities for better performance

Researcher Token Rewards

Researchers are incentivised to constantly try to increase their performance. The multiplier is tripled in each band. So, the large increase in multiple is intended to motivate researchers to try rank as high as possible to maximize the gains that they can potentially receive. The ranking multiplier is not the only variable that determines how much of a token reward they will revive. The other variant comes from the overall ranking that their work receives from the validators. In the last section we took the simple example whereby three reviewers were randomly silenced and from their combined review the average score for the work was calculated to be:

$$Submission_{score} = \frac{\sum_{n=1}^{N} S_n}{N}$$

$$Submission_{score} = \frac{8_{reviewer1} + 8_{reviewer2} + 8_{reviewer3}}{3} = 8 = 80\% = 0.8$$

The submission score is used as a multiple in combination with the authors multiplier from above to determine how much they should be rewarded. In the case above this particular author scored 80% which would correspond to a multiplier of 0.8. The product of the band multiplier and the submission score defines that authors final multiplier, and the reward they revive is given by their total stake divided by their multiplier.

$$Multiplier = Multiplier_{band} \times Submision_{score}$$

$$TotalReward = Author_{stake} \times Multiplier$$

If we take the example where we have two authors. $researcher_A$ in the $21 - 40\%$ range of all stakeholders and $researcher_B$ in the $61 - 80\%$ range. If the who is ranked. Both researchers have the same amount of DST staked, so we will let this be $Author_{stake} = 1000\ DST$ for this example. If $researcher_A$ received a submission score of 75% by the validators and $researcher_B$ scored, 90% then we can caluctae the number of tokens each will get as a reward in either case

$$Multiplier = 0.8 \times 0.75 = 0.6$$
$$TotalReward_{researcher_A} = 1000\ DST\ \times 0.06 =\ 60\ DST$$

$$Multiplier = 1.2 \times 0.9 = 01.08$$
$$TotalReward_{researcher_B} = 1000\ DST\ \times 0.108 =\ 108\ DST$$

We hope that this will incentives researchers to constantly better themselves to produce the best material possible for publishing to the network. In terms of the reputation recalculation this is calculated as per the methods discussed in **section 4.4**

Validator Token Rewards

A similar method is used to determine the token reward for the validator class. However, although validators will also have different reputation rankings, we realise that in the case that a member who is new to the network, in that they do not yet have a large token stake or on chain reputation then the above method for calculating their reward for carrying out peer-review is not justified. For example, if a validator spent a valuable amount of time for reviewing someone's paper in return for just a 0.03% reward multiplier, then this would be very discouraging and would not act as a sufficient incentive for that validator to want to participate in the peer-review process. Therefore, for validators the regards schema is modified slightly, and we propose that there will be two sets of token rewards. In the first round each validator will receive the same reward and there are no bands or categories that allow validators to be eligible for a higher reward in comparison to others. The multiplier for validators will be set at roughly $4 - 7\%$ for there token reward multiplier. We enact a high return rate for a variety of reasons. The first is due to the random nature of the WRS algorithm. The likelihood of the same validator being selected to carry out reviews more than twice in a row is very unlikely. This means that the opportunity for validators to review is completely random and prolonged periods of time may elapse before the same validator is selected again to carry out their peer review duties. Since this is the case, we implement a high ROI or rewards multiplier that gets enacted only on the total DST stake that a validator has locked up. This will encourage validators to stake higher amounts in order to net them higher rewards. The reason that we suggest calculating a validators reward only from their locked stake is for simplicity. Validators will still be highly incentivized and encouraged to constantly gain a higher total network reputation score so that their actual chances of being selected by the WRS algorithm are greatly increased. The second-round proceeds in the same fashion as described above for a researcher where based on their on-chain reputation, each validator will have a chance to earn an extra bonus where the multiplier is determined again from the curve shown in **Figure18.** This extra layer serves as another incentive for validators to constantly try to preform and rank higher in the community. We can again carry out an example calculation for two validators, $Validator_A$ and $Validator_B$ who have staked 500 and 1000 DST respectively. Although the actual rate is not set in stone, for the purposes of this example we will take the higher end of the range that we mentioned above, so the multiplier for the peer-review reward will be taken as 7%.

$Round\ 1\ Rewards$:

$$Multiplier = 0.07$$

$$TotalReward_{Validator_A} = 500\,DST\ \times 0.07 =\ 35\,DST$$
$$TotalReward_{Validator_B} = 1000\,DST\ \times 0.07 =\ 140\,DST$$

$Round\ 2\ Rewards$:

$$Multiplier_{Validator_A} = 3\%\quad ,\quad Multiplier_{Validator_B} = 5\%$$

$$TotalReward_{Validator_A} = 500\,DST\ \times 0.03 =\ 35\,DST$$
$$TotalReward_{Validator_B} = 1000\,DST\ \times 0.05 =\ 140\,DST$$

$Totals$:

$$TotalReward_{Validator_A} = 500\,DST + 35\,DST =\ 70\,DST$$
$$TotalReward_{Validator_B} = 140\,DST + 50DST =\ 190\,DST$$

We hope that this will incentives validators to constantly better themselves to produce the best material possible for publishing to the network. In terms of the reputation recalculation this is calculated as per the methods discussed in **section 4.4**

5.3 Slashing Mechanism

The next mechanism we will discuss is how authors can get punished or slashed. The criterion for such situations again spans across multiple categories just like the criteria for staking rewards. The range of categories that authors can get slashed for is much less than that of a validator, but the severity of the slash remains the same. Unlike above there is only one metric that is used to determine punishment induced for a slash.

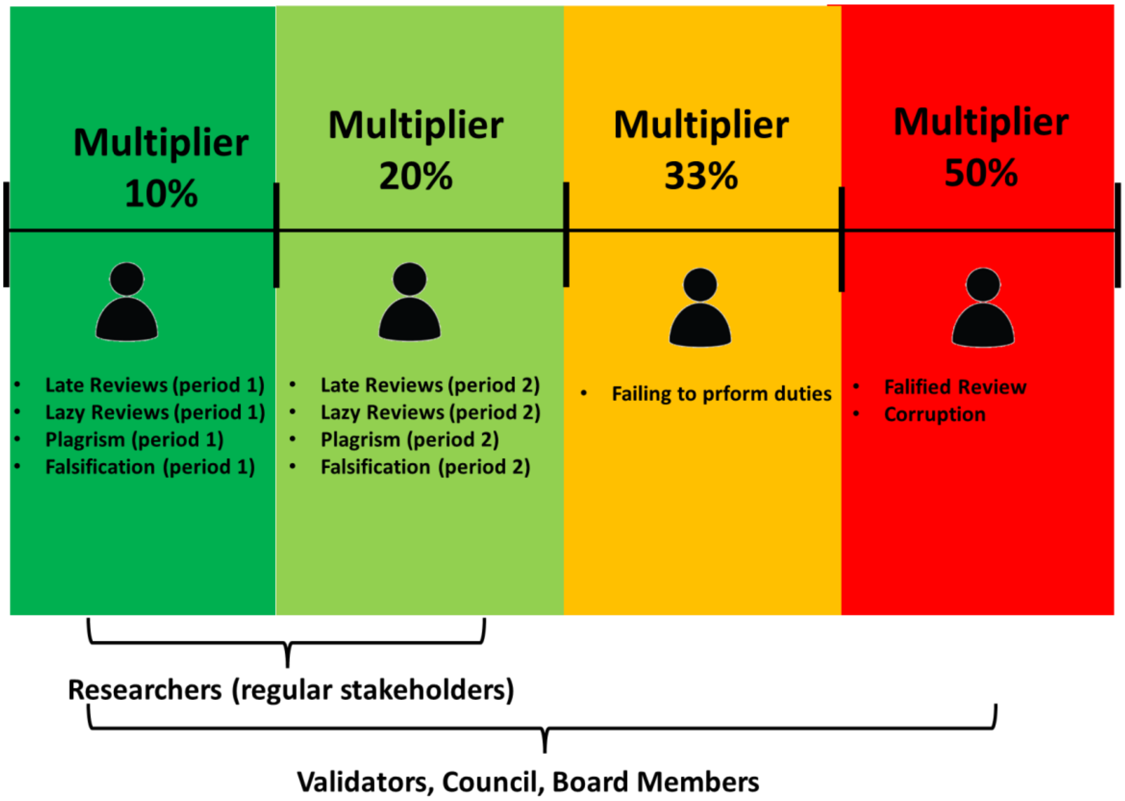


Figure 19: slashing penalty multiplier categories

In the **Figure19** above we can see the various multipliers associated with different types of slashing criteria. The way in which this multiplier affects a stakeholder's reputation and token stake is that the percentage of the multiplier is reduced from both that persons stake and on-chain reputation. The slashing protocol is strict on purpose to really discourage anyone from acting in bad faith against the interests of the community. We can again take the example where we have $validator_A$ and $researcher_A$. IF we first suppose that $researcher_A$ is found to have polarized some part of their paper, then then the validators are tasked to decide the severity of the punishment. In this case we will assume that the researcher will get slashed in the first band so a 10% reduction will be applied. In this case the researchers token stake and reputation will both get reduced by 10%. If we take scenario where a researcher has 1000 DST staked and an on-chain reputation score of 30, then the slash will proceed as follows:

$Researcher_A\ Slash$:

$$Reducer = 10\%$$

$$TotalStake_{Researcher_A} = 1000\ DST -= 1000\ DST\ \times 10\% =\ 900\ DST$$
$$TotalReputation_{Researcher_A} = 30_{on-chain_{rep}} -= 30_{on-chain_{rep}} \times 10\% =\ 27_{on-chain_{rep}}$$

The next slashing scenario is concerned with $Validator_B$. In this case the validator has submitted a late review. The way late reviews will be calculated will be through the staking/slashing smart contract code. In this case the validator has gone over the second limit deadline and thus will receive a 20% reduction to both their token stake and reputation. If we now take the scenario where the validator has again a 1000 DST stake and an on-chain reputation core of 50, then the reduction applies as follows:

$Validator_B\ Slash$:

$$Reducer = 20\%$$

$$TotalStake_{Validator_B} = 1000\ DST -= 1000\ DST\ \times 20\% =\ 800\ DST$$
$$TotalReputation_{Validator_B} = 50_{on-chain_{rep}} -= 50_{on-chain_{rep}} \times 20\% =\ 40_{on-chain_{rep}}$$

The above example demonstrates trivial scenarios of how community stakeholders such as validators and researchers can get rewarded and punished depending on their actions. One thing that is important to note is the fact that in the above examples, the slashing events were determined in the publication procedure. Namely for $Researcher_A$, the validators came to consensus on the fact that plagrism was detected. Meanwhile for $Validator_B$, the smart contract detected the submission of his/her late review and enacted the penalty. However, for other forms of slashing criteria, many times the outcome will be put to vote through the governance model and only then will the punishment be enacted. In out governance model the opportunity exists for any stakeholder to submit a slashing proposal on any other stakeholder if they have sufficient evidence to back their claim. All of such proposals will be put to vote and if there is a clear agreement in any direction of the vote then a slashing punishment can get executed on the accused. This will be discussed in more detail below in section 6, so now that we have explored our staking/slashing mechanism we can move onto the next large protocol that is responsible for decision making in our ecosystem. That is the concept of community governance.

6.0 Governance Model For Democratic Decision Making

The underlying protocols that govern our platform will all need to change and adapt gracefully over time in order to stay relevant with the needs and wants of the community. Thus, we set out to design our infrastructure to have a transparent and sophisticated process to not only approve or reject changes proposed by the stakeholders, but also to enact them automatically with on-chain fully automated smart contract code. Blockchain governance frameworks have, so far, faced several problems. Forks split communities as well as software, and the dependence on security and adoption creates a zero-sum game where only one chain emerges. Some claim to have no governance at all, and groups can fork (see appendix) the network over parameters like block size and must defend their forks with religious fervour. Others use off-chain collectives that organize over phone calls or at conferences, which either leads to shadow hierarchies where only a few, unwritten people make decisions or, lacking a framework to make a decision, the collective never advances. These problems have led some to implement coin-voting protocols to make decisions. Coin voting is a good first step toward transparent, open, on-chain governance, but low turnouts make it susceptible to large voters controlling a vote. In all blockchains to date, governance stops at decision making. Even if a collective or a coin vote leads to an agreement, they lack the means to enact their decision; the true power still lies outside the protocol, for example with miners or validators. Just because a country holds elections, for example, doesn't mean people consider it a democracy; the system must include the means to enact the outcome. The same applies to blockchains and web3 applications. Coin voting is not sufficient if it is not binding. In this section we define has several ways for users to express their wishes for change. Besides making it easy for users to propose changes, we define a governance structure for users to form collective groups that carry unique privileges. The motivation behind collectives comes from seeing votes in other decentralized protocols or applications controlled by a single voter. These decisions have included sensitive topics like killing the application.

5.1 Eligibility for Voting In The Network

From section 4.0 above, we defined that there are largely three main types of user accounts for our platform. The first type of account is the general public who may not be actively publishing papers or articles but are none the less owners of our governance DST token. The second account type refers to actively publishing academics, and finally the last account type is that of the network validators. Namely, the individuals who are tasked with carrying out the peer review process. One common trait amongst all of these account types is that they all must own an allocation of DST. Thus, we make the requirement that in order to have the power to vote on proposed network changes or in order to have the power propose network changes, the community members must own a predefined amount of DST. The minimum amount is not yet defined but it should be a sufficient amount to discourage bad actors from just acquiring an easily obtainable amount of DST to spam votes or propose useless network changes. If we now look back to Figure2: which shows the link between some of the core smart contract of our protocol, we can see that there are three main smart contracts.

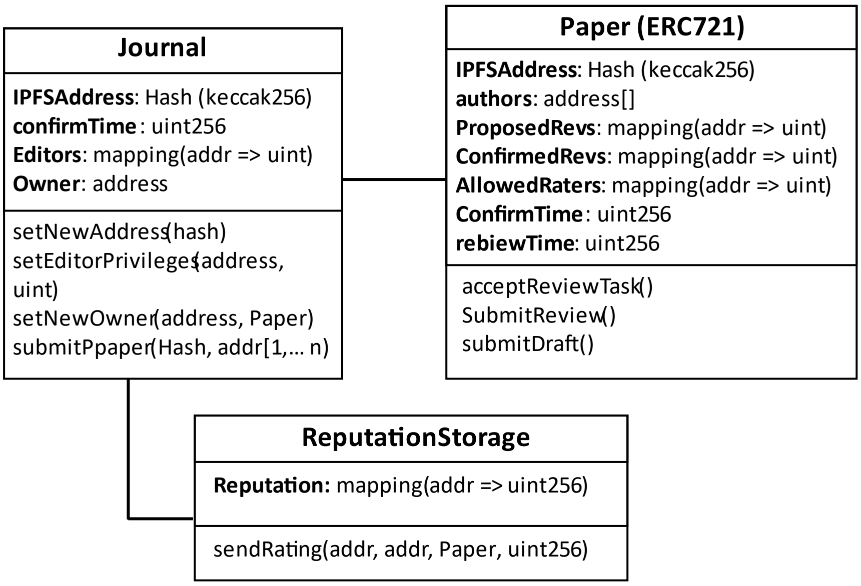

Journal
IPFSAddress: Hash (keccak256)
confirmTime: uint256
Editors: mapping(addr => uint)
Owner: address
setNewAddress(hash)
setEditorPrivileges(address, uint)
setNewOwner(address, Paper)
submitPpaper(Hash, addr[1,… n)
Paper (ERC721)
IPFSAddress: Hash (keccak256)
authors: address[]
ProposedRevs: mapping(addr => uint)
ConfirmedRevs: mapping(addr => uint)
AllowedRaters: mapping(addr => uint)
ConfirmTime: uint256
rebiewTime: uint256
acceptReviewTask()
SubmitReview()
submitDraft()
ReputationStorage
Reputation: mapping(addr => uint256)
sendRating(addr, addr, Paper, uint256)

Figure 20: Basic smart contract layout

The Journal contract controls the paper submission, the selection of editors, the assignment of reviewers, and the acceptance of reviewers. The Paper contract identifies a paper within the system, controls the review submissions, and shares who may rate a review. Finally, the Reputation Storage contract stores the ratings of the peer reviews, receive new rates, updating the reputation of reviewers if allowed by their Paper contract, and shares the reviewers' reputation. It is important to understand that these three smart contracts just serve as the core components of the system and all of the lower level protocols such as the governance, consensus, crowdsourcing and staking mechanisms (see later sections for last two) are all inherited from these main contracts. One Major flaw that we can easily point out with this system is that if we have one huge smart contract that contains everyone's data that represents the "decentralized Journal" so to speak, then it goes without saying that all scientists from every academic discipline are all under the same category. In this scenario this means that potentially the validator set could be composed of scientists from different disciplines who would be at some point required to review work in an area of science that they have no expertise in. This boasts a major flaw in our system.

However, it was for this reason that the design is reduced to these three simple contracts. The smart contract programming language solidity has a way of duplicating smart contract logic to unique locations or in other words we are effectively able to deploy the same smart contract logic to unique addresses. This feature is known as the Factory contract and it is a widely used technique in some of the most popular decentralized finance (De-Fi) protocols such as uniswap **[49]**, 1inch **[50]** and more. The way they work is that we have one parent contract which defines all of the logic that we want to capture in our smart contract. A child (factory) contract is then defined and inherited from the parent contract that we want to deploy instances of. In order to deploy instances of our factory contract we execute a function from the child contract which deploys the parent to a unique address each time the function is called. The address of each deployed contract is stored in an on-chain array data structure that lies within the factory contract so information about each deployed factory instance is recorded an available for fetching via the blockchain We can consider the diagram below

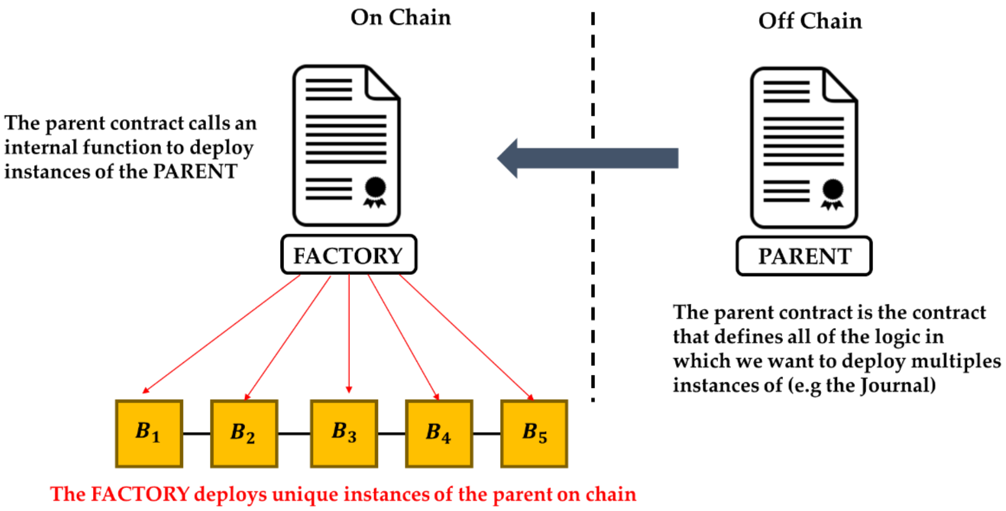

On Chain
Off Chain
The parent contract calls an internal function to deploy instances of the PARENT
FACTORY
PARENT
The parent contract is the contract that defines all of the logic in which we want to deploy multiples instances of (e.g the Journal)
B_1
B_2
B_3
B_4
B_5
The FACTORY deploys unique instances of the parent on chain

Figure 21: Demonstration of Factory contracts solidity

The whole idea is to spread out the computation for a particular instance of governing logic. We have already pointed out that have one governing Journal contract is flawed because of the unavoidable possibility of reviewers being selected to review works in areas outside of their usual domain. But the old setup is also flawed in that it is incredibly computationally inefficient. If we only had one governing journal contract, then over time as the size of the network grows potentially 1000s of users would be stored in one on-chain data structure which would become increasingly slow and computationally inefficient long term. This optimized solution helps distribute the amount of data that each smart contract needs to handle and process making it more computationally effective storage wise. When realized in out case the Journal and Reputation Storage contracts would act as the factory contracts which we would then use to deploy multiple versions. The reason that this is useful is because we can effectively, through on chain democratic proposals, choose to create new Journal and reputation contracts for emerging niches in science. We can think of our overarching platform as a Decentralized Autonomous Organization (DAO). A DAO is just a democratic organization that is completely governed immutable smart contract logic and decision made in the network are delegated to the community who use there voting power to decide the future of the network. In order for a community stakeholder to propose the creation of a new sub-DAO or Journal for a specific niche in science then they would have to first lock up a given amount of DST in order to enact the proposal. If then, enough members in the overarching community agreed on the proposal and a super-majority consensus was received on the proposal then the Journal Factory would automatically deploy a new instance tailored specifically for peer reviews in the given niche defined in the proposal. This way we can elegantly create sub community of scientific in different factions but still have the entire community connected on a greater level. This deviation was intended to first give some context on why community governance and voting mechanism are needed in a decentralized organization such as this. There are many problems associated with this and in order for thing to work we need to define some robust governance protocols that will make sure the system cant be manipulated or hijacked. This is only one aspect where governance is needed in our application and other areas will be pointed out later in this section

5.2 Collectives

All 3rd generation blockchains such as Ethereum, Cardano and Polkadot employ robust governance models in order to perform as efficiently as possible. The model that we propose in this paper is heavily inspired from that of the governance model developed by the Polkadot Network **[51]**. One idea that is common amongst all governance protocols is that stakeholders should ultimately have the control, which is why all changes in should go through public referenda, but stakeholders should also have the capability to elect representation for such decisions. Collectives protect masses of more passive users from the whims of a body, large token holders. In our model there exists two special collectives related to governance. They are the Technical Committee and the Council/Board. The council is an on-chain body or collective that exists to represent passive stakeholders. It does this by proposing important changes and also cancelling any uncontroversially dangerous proposals. Any DST token holder can run for the council, but their reputation is at stake to act in good faith for the network. A unique collective will exist in each Journal or micro-community sub DAO and will have a board of 10 members but this number is subject to change depending on the size of the DAO. Council elections will run every quarter (meaning 4 times a year) and any community member who holds DST can both run as a council member or vote for other nominees. The election itself will assign the top 10 voted nominees as council members for that particular instance of the Journal factory.

The Technical Committee

The technical committee is composed of a group of the platforms core developers and is designed to act as the last line of defense against software errors or detrimental smart contract bugs. Since smart contracts are immutable, once they are deployed, they contract code cannot be changed. This means that in the case that a bug was missed during an audit and was discovered after deployment then the role of the technical committee is to fast track all other network proposals and enactment delays to fix said bugs and software issues to prevent against external attacks. The idea of having the core developers being the only members of the technical committee seems rather flawed and defeats the purpose of keeping things decentralized, therefore the technical committee board is subject to change and new members with technical prowess can run for the position externally although this is something that should be explored in the future, and we have currently not explored a viable solution as of yet. However, one thing we can do to combat the scenario of malicious intent by the core developers (for whatever reason) is to not allow them to make proposals, but rather to only allow them to be able to fast track existing proposals to happen in a shorter period of time than normal to fix only technical issues. If unanimous, then the technical committee can skip the enactment delay (see below) an enact a software fix as soon as it happens. Although the Technical Committee is not elected, they have a limited scope, and the proposals that they fast track still need to go through a public referendum. They can only make governance for critical bug fixes happen faster than normal but cannot control the network.

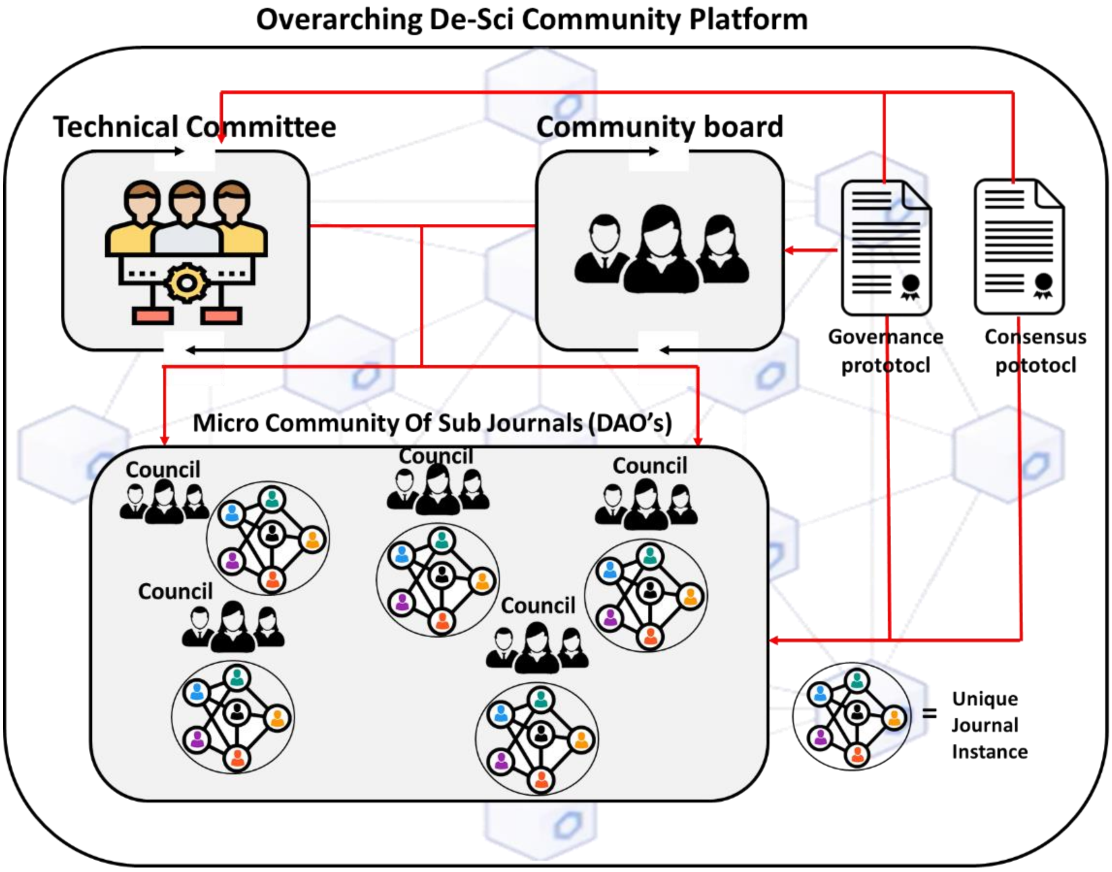
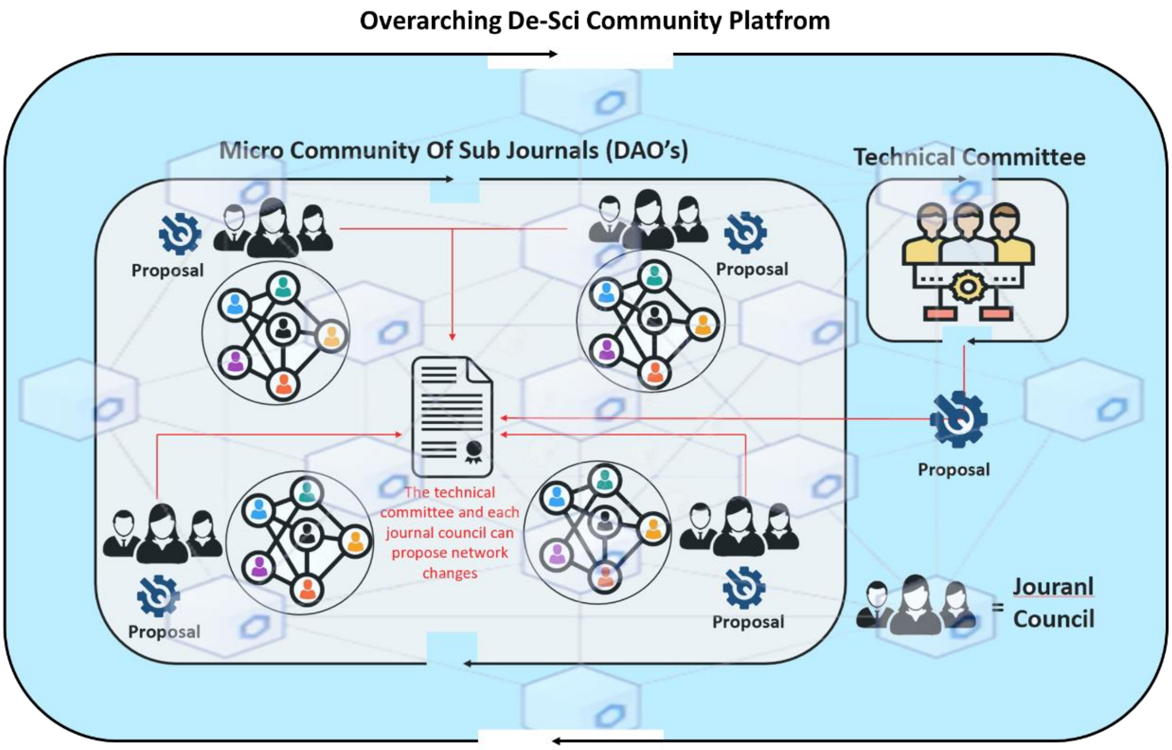


Figure 22: Overview of Sub-communities and the scope of the board/council & technical committee in the ecosystem


Overarching De-Sci Community Platfrom
Micro Community Of Sub Journals (DAO's)
Technical Committee
Proposal
Proposal
Proposal
Proposal
Proposal
The technical committee and each journal council can propose network changes
Jouranl Council

Each Journal instance has its own council who propose network changes that act in the best interest of that particular sub community as well as the public can make proposals. The technical committee is a body that lies beyond the scope of each sub DAO Journal and can enact software and bug fixes

Figure 23: Overview of Sub-communities and the scope of the board/council & technical committee in the ecosystem (bonus: how proposals get created by each of the said bodies)

5.3 The Creation & Enactment Of Proposals

All governance decisions begin as proposals and pass through what's called a public referendum. A proposal can be any one of a set of privileged functions that are not available to most users. For example, actions such as changing protocol parameters such as the size of the validator set, or as we mentioned earlier proposing the creation of a new journal contract to govern some emerging niche in science or even things like slashing bad actors in the network. All action such of these are not things any regular user can execute on their own, but rather they can propose these decisions to get enacted through proposals. One major role of governance other than the creation of new journals is to allow users to issue a slashing proposal on another member provided they have sufficient evidence of foul play (this is discussed in greater detail below). In our model, proposals can start in three ways:

- From the public, as in any DST holder
- From the Council, which consists of publicly elected DST Token holder
- As the result of the enactment of another proposal

Regardless of the origin, a proposal starts merely as the hash of a privileged function call. Since all proposal are stored on-chain in a queue, then it is far too costly to store the an object or Struct containing all of the data associated with the proposal. On the User interface side of things, The user submitting the proposal is required to meet some defined criteria. They must send a required amount of DST with the transaction which will get immediately sent to the platforms Treasury. They then must fill out a forum which described the details of their proposal. When the transaction is executed the has of the transaction is stored on-chain in the proposal queue and an event is fired which is stored on the blockchain and this event contains the details of the user who submitted the proposal as well as a time stamp of creation. The purpose of the vent is for simple look up via something such as a block explorer such as ethers can. Any number of proposals can exists simultaneously, but only one can make it to a public referendum during each voting period (1 month). This encourages users to choose carefully on what the ultimately decide to vote on, ensuring that the final enacted proposal is truly the change that is in the best interest of the network as a whole.

Public Referenda

A core tenet of our governance mechanism is that a majority of stake, defined as the total number of tokens in issuance, can always command the network. Blockchains are economic vehicles and do not understand democratic one-person-one-vote systems. [4] Those who want influence in the direction of the system must take an active stake in it. Proposals must pass through a public referendum where all stakeholders can express their opinion. Every thirty days, we suggest that our governance system autonomously selects the next proposal to go to referendum by alternating between awaiting Council and public proposals to ensure that public proposals have an equal chance of reaching referendum. Once a referendum begins, users can begin voting. But unlike other blockchains, votes are not strictly the number of tokens in an account. Every vote comes with some conviction, some skin in the game. By default, users who voted for a passed proposal must lock tokens up until the proposal's enactment. This lock makes them stay in the network and endure the ramifications of their vote, while those on the losing side of the referendum are free to exit. But users can increase their voting power by committing to the decision for a longer period of time and thereby increasing their exposure to the outcome. Each doubling of the lock time increases the power of a user's vote, all the way up to six times the account's balance (which would be a lock of 32 enactment periods). [5] This mechanism exists to ensure that users with little stake but strong opinions can express their conviction in referenda. At the end of the voting period, the votes will be tallied and the results calculated. If the proposal passes, then our governance smart contract logic will automatically schedules it for enactment, normally 30 days later to give time for external services to make any necessary adjustments and for those who oppose the decision to exit. Fast-tracked referenda, presumably for an emergency technical fix, can take effect immediately.

5.4 The Dynamics Of On-Chain Voting

Every28 days, a new referendum will come up for a vote, assuming there is at least one proposal in on of the queues. Namely the public voting queue and the Council voting queue. The referendum to be voted upon alternates between the top proposals in either queue, i.e the proposal in each queue which ahs the most staked backing by the community at large. The "top" proposal is determined by the amount of stake boned behind it. If the given queue whose turn it is to create a referendum is empty, and the alternative queue is not, then the top proposal in the other queue will become a referendum. It is important to note that in our design multiple referenda cannot be voted upon in the same period. The only exception to this is an emergency referendum issued by the Technical Committee such as a bug fix or detrimental software issue. Thus, an emergency referendum occurring at the same time as a regular referendum (either public- or council-proposed) is the only time that multiple referenda will be able to be voted on at once. To vote, a voter must lock up their tokens for at least the for the proposal they choose to vote on. A certain proportion of each voters stake will be sent to the treasury smart contract (see section) and upon the enactment of a referendum the rest will be returned to the voter. It is possible to vote without staking at all, but in this case the voters vote will only be worth a small fraction of a normal vote, given his total locked stake at large. At the same time, only holding a small amount of tokens does not mean that the holder cannot influence the referendum result thanks to a concept that is explored by the Polkadot governance protocol, known as time locking. For our model we will adopt this idea, modified slightly for the particular needs of our platform. For an example of how time locking increases the weight of someone's vote consider the below.

$$Evan: Votes\ \text{NO}\ with\ 100\ DST\ for\ a\ 128\ week\ lock\ period => 10 \times 6 = 60\ Votes$$

$$Leo: Votes\ \text{YES}\ with\ 100\ DST\ for\ a\ 4\ week\ lock\ period => 20 \times 1 = 20\ Votes$$

$$\begin{aligned} Andrew: Votes\ \text{YES}\ with\ 100\ DST\ for\ a\ 8\ week\ lock\ period => 15 \times 2 \\ = 30\ Votes \end{aligned}$$

$$Where\ 1\ lock\ period = 4\ weeks$$

Although the above example is heavily exaggerated on purpose, it effectively conveys the dynamics of time locked votes. Even though combined, both Leo and Andrew vote with more DST than Evan, the lock period for both of their

votes is far less than that of Evan, inevitable leading to their voting power counting as less. The purpose of this is to allow extremely passionate stakeholders to ensure that their voice can be heard in such a scenario where that person may not be as financially equipped as other member sin the community. This prevents only the rich or more wealthy stakeholders form having more control over the network via their voting presence. The table below shows the hwo the number of time in which a voter locks their tokens is affected by the vote multiplier.
This multiplier essentially doubles each lock period. (1 lock period is4 weeks) So every time a voter locks their voting stake up for one more locking period, the power of their vote is doubled.

Lock Periods	Vote Multiplier
0	0
1	1
2	2
4	3
8	4
16	5
32	6

Table1: Example of Tallying Scenarios

The maximum number of "doublings" of the lock period is set to 6. There is no need to argue the inclusion of a locking limit. This prevents users for infinitely locking up virtually no tokens for an unlimited amount of ime in order to get more voting power. It is also important to note that when a user's tokens are locked from a vote, they can still use those tokens within the network on thing such as tother votes and other staking purposes, but they are prohibited from transferring the tokens to other accounts. Depending on which entity proposed an given proposal and whether all council members voted YES, there are three different scenarios that can play out. The are listed in the table below

Entity	Metric
Public	Positive Turnout Bias (Super-majority Approve)
Council (complete Agreement)	Negative Turnout Bias (Super-majority Against)
Council (Majority Agreement)	Simple Majority

Table2: scenarios for using positive & negative turnout bias's as well as simple majority

Also, we need the following information and apply one of the formulas listed below to calculate the voting result. For example, lets use the public proposal as an example, so the **Super-Majority Approve** formula will be applied. There is no strict quorum, but the super majority required increases with lower turnout

$approve\ -\ The\ total\ number\ of\ YES\ votes$

$against\ -\ The\ total\ number\ of\ NO\ votes$

$turnout \;\; - \;\; the\ total\ number\ of\ voting\ tokens$

$electorate \;\; - \;\; the\ toal\ number\ of\ tokens\ circulating\ in\ the\ network$

Table1: Example of Tallying Scenarios

Super-Majority Approve

A positive turnout bias, whereby a heavy super-majority of aye votes is required to carry at low turnouts, but as turnouts increase towards 100%, it becomes a simple-majority case as demonstrated below also

$$\frac{against}{\sqrt{turnout}} < \frac{approve}{\sqrt{electorate}}$$

Super-majority Against

A negative turnout bias, whereby a heavy super-majority of NO votes is required to reject at low turnouts, but as the turnout increases towards 100%, it also just like the Super-Majority Approve scenario, becomes reduced to a simple-majority carries

$$\frac{against}{\sqrt{turnout}} < \frac{approve}{\sqrt{electorate}}$$

Simple-Majority

In this case just the majority carries. Meaning that which ever side has more votes (YES or NO), then then that side wins and the proposal is straight up accepted or rejected with no weighting as demonstrated in the two formulae above.

$$approve > against.$$

If we now take the fictional scenario where we suppose that the total supply of DST in the entire network 1500 DST. This is not a realistic scenario and in practice the total supply for a given networks token would be in the 10s of millions. However, we use this small number to keep the concept simple to understand. The total number of stakeholders in the network is five as seen below. Therefore, we set Up the scenario where a proportion of the stakeholders in the network vote in the given proposal. Using the formula, we described above we can calculate the result of this vote based which again, is dependent on the total circulating supply of tokens, the time locked weight of each voters vote and the number of tokens active in the voting process.

$$\boldsymbol{Evan}: 500\ DST$$

$$\boldsymbol{Leo}: 100\ DST$$

$$\boldsymbol{Andrew}: 150\ DST$$

$$\boldsymbol{Robert}: 150\ DST$$

$$\boldsymbol{Laura}: 600\ DST$$

$$\boldsymbol{Evan}: Votes\ \text{YES}\ for\ a\ 4\ week\ lock\ period => 500 \times 1 = 500\ Votes$$

$$\boldsymbol{Leo}: Votes\ \text{YES}\ for\ a\ 4\ week\ lock\ period => 100 \times 1 = 100\ Votes$$

$$\boldsymbol{Robert}: Votes\ \text{NO}\ for\ a\ 8\ week\ lock\ period => 150 \times 3 = 450\ Votes$$

We can now demonstrate the outcome by calculating our 4 voting parameters

$$approve = 500_{Evan} + 100_{Leo} = 600$$

$$against = 450_{Robert}$$

$$Turonout = 1500_{Total-Tokens} - (150_{Laura} + 600_{Andrew}) = 750$$

$$electorate = 1500\ (total\ Tokens)$$

Applying the super-majority formula, we can calculate the result of the vote

$$\frac{450}{\sqrt{750}} < \frac{600}{\sqrt{1500}}$$

$$16.432 < 15.492$$

Again, since the above example is a public referendum, the Super-Majority Approve formula would be used to calculate the result. The Super-Majority Approve formula requires more YES votes to pass the referendum when the turnout (proportion of total community members active in the vote) is low. A proportion of each voters tokens as mentioned earlier will directly be sent to the treasury and after the locking period is finished, the remainder of the tokens will be returned to their respective owners. The reason that not all of the tokens are returned is because the benefits resulting for the proposals of network changes should come at a small cost which will help fund the ecosystem on a whole to make it more sustainable in the long term. If on the contrary to the above result, the proposal was accepted, then it would be autonomously enacted y the governance protocols smart contracts after a predefined and constant enactment period which is 1 month. This gives the protocol time to prepare the changes that need to be enacted to ensure a smooth and seamless rollout is achieved.

Adaptive Quorum Biasing: A way to solve Low Proposal Turnouts

Another very important concept that our model implements which is inspired from Polkadot is the idea of adaptive quorum biasing. Adaptive quorum biasing functions as a lever that the council can use to alter the effective super-majority required to make it easier or more difficult to pass proposals in the case that there is no clear, black and white majority of voting power in one direction (YES or NO). To explain the concept we can loo to the image below

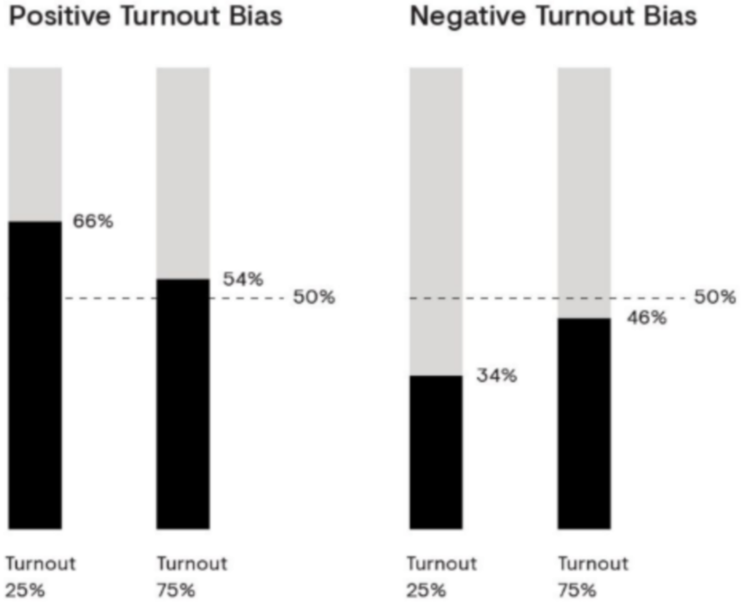


Figure 24: Demonstration Of Adaptive Quorum Biasing (taken from [**51]**)

If a public ally submitted referendum only has a 25% turnout, the tally of YES votes need to reach 66% (Super-Majority) for it to be accepted since we applied Positive Turnout bias. In contrast when if the same proposal has a 75% turnout, the total number of YES votes only needs to reach 54% in order for it to be accepted. This means that the Super-Majority required for any given proposal to pass decreases as the vote turnout increases. When the council proposes a new proposal through unanimous consent, the referendum would be put to a vote using Negative Turnout Bias. In this case, it is easier to pass this proposal with low turnout and requires a super majority to reject. As more token holders participate in the voting mechanism, the bias approaches a plain majority carries. Referring to the above image, when a referendum only has a 25% turnout, the tally of YES votes has to reach 34% for it to pass. In short, when the turnout rate for a proposal is low, a super-majority is required to reject the proposal, which means a lower threshold of YES votes have to be reached, but in contrast, as the turnout increases towards 100% it becomes a simple majority. All three tallying mechanisms - majority carries, super-majority approve, and super-majority against - equate to a simple majority-carries system at 100% turnout.

5.5 Important Voting Applications

Now that we have explored some of the mechanisms and safeguards that define the functionality of our governance protocol, in this section we aim to cover some important scenarios in which the mechanism defined above determine the outcome of community proposals. The first thing that we should cover is the process in which new council member set elected. All community stakeholders are free to signal their approval of any of the registered candidates. Only members in the validator set(s) of micro communities are eligible to run as a council member for that particular Journal instance. The council elections are run every quarter year or 4 months. It is possible for council members to run twice or keep their position but in order to keep thing from centralising it is defined that a council member may only keep their seat for two successive elections. To represent passive stakeholders, we introduces the idea of a "council". The council is an on-chain entity comprising several actors, each represented as an on-chain account. The council should be a small body consisting of only a few members, In Polkadot’s model the council currently consists of 13 members. Along with controlling the treasury, the council is called upon primarily for three tasks of governance: proposing sensible referenda, cancelling uncontroversially dangerous or malicious referenda, and electing the technical committee.

Proposals of new Journals

Arguably the most important crux of our platform reduces to the proposal by the community of new sub journal entities that represent a specific niche academia. As we discussed earlier, the reason for this is to uphold the integrity the peer-review process for different areas in science by ensuring that all of the validators are experts in that field. The way in which the proposal of a new journal entity is enacted is twofold. Any stakeholder in the community and council members propose to create a new Journal entity. In our platform the entire community is still heavily connected at large. It is important to note that the creation of sub journals is just safeguard put into place to cater for scientists needs, interests and wants as well as ensuring the integrity of the peer review prices at heart. But this architecture does not isolate the overarching community in any way when it comes to governance and voting mechanisms. In order to. Stake holders can both vote on proposals that concern the entire community or proposals that are niche to their specific journal. However, when it comes to the case where there is a journal specific vote, only members who are subscribed to that journal can vote. This is to prevent outsiders from having a biased say in the outcome. When it comes to proposing new journals, this is a wider community proposal type. Therefore, anyone in the community can voice their say. For the proposal to make it to a referendum the supermajority formula that we defined above is applied for a positive turnout bias. If there is enough support for the proposal, either from the case that is passes from a high or low turnout, then the proposal will be submitted to the board, where then again it will be voted upon by the board members using a negative turnout bias. If after these two voting phases the proposal passes, then, the Journal contract will be consulted and after the enactment delay the new journal will get created, and its validator selected using the selection algorithm that we defined in section 3.

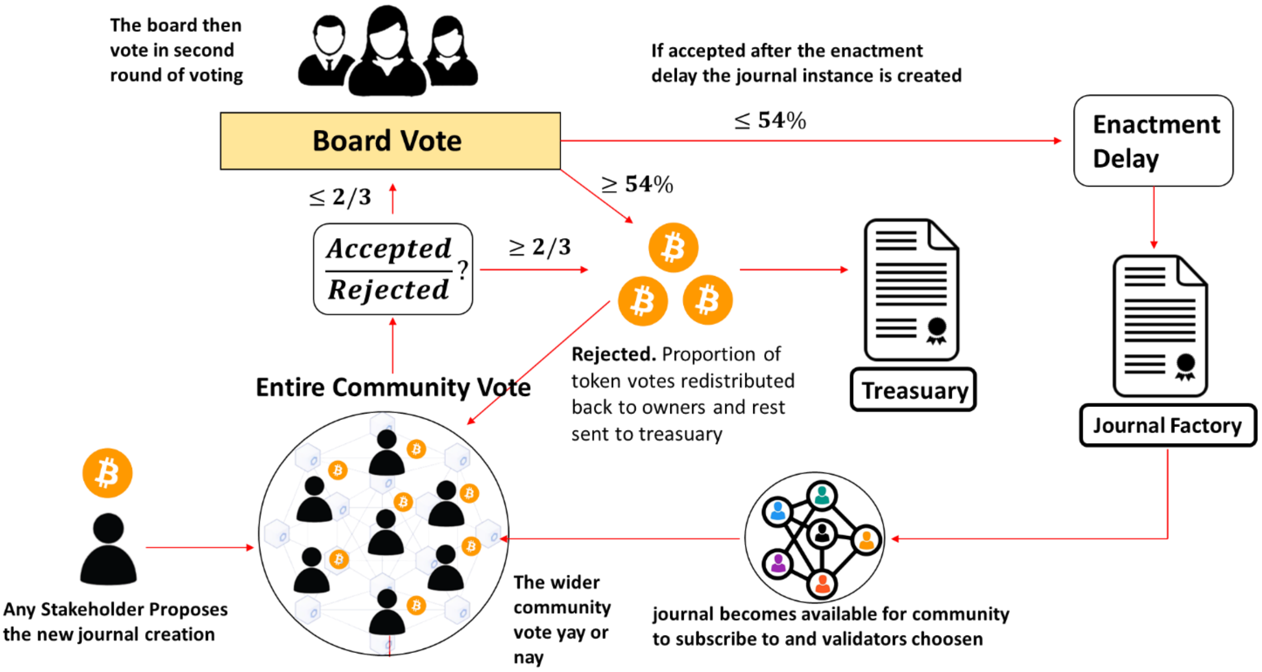

The board then vote in second round of voting
Board Vote
If accepted after the enactment delay the journal instance is created
≤ 54%
Enactment Delay
≤ 2/3
≥ 54%
Accepted
Rejected
?
≥ 2/3
Treasuary
Journal Factory
Rejected. Proportion of token votes redistributed back to owners and rest sent to treasuary
Entire Community Vote
Any Stakeholder Proposes the new journal creation
The wider community vote yay or nay
journal becomes available for community to subscribe to and validators choosen

Figure 25: Typical flow for a proposal created by a stakeholder within the system

In figure4 above, the process begins with any stakeholder who initialise the proposal by locking up tokens and broadcasts it to the wider community. Here, the community then have the option to second (vote on) the proposal either with a YES or NO vote. Since this is a public vote a positive turnout bias is used, meaning that a super Majority (66%) consensus needs to be achieved for the proposal to move onto the second round of voting. If this is not achieved and the proposal is rejected, a proportion of the user’s tokens which were used to vote are returned and the rest are sent to the treasury smart contract. One thing that’s important here is that many protocol would burn the tokens at this point of rejection. This introduces risk and dismays stakeholders from broadcasting useless proposals in fear that they will lose their tokens. However, burning the tokens does not help the network in any way in my opinion. In this scenario sending tokens to the treasury promotes sustainability and these rejected vote tokens will be able to be used for treasury spending in the future which will ultimately benefit the community. On the other hand if the proposal gets accepted by the public then it is cast onto the community board of electives who will vote using a negative turnout bias (meaning a supermajority needs to reject or in other words only 54% of the board needs to vote YES). If the proposal gets rejected here the same distribution process is executed as before is the first round vote. Finally, If both phases of voting are successful then the proposal will be prepared to be executed during the enactment delay and the Journal factory smart contract will autonomously create a new journal instance which will become available for the community to subscribe to. And the Consensus protocol will assign the initial validators.

There is one other scenario in how this vote could take place. It is possible for a board member to propose the creation of a new journal. In this scenario only the second voting round carries, meaning that only the board members will vote on the proposal. The reason that this can happen is that recall we ultimately have two voting ques in which proposals are stored. The public queue and the council queue. Since the board is a high-level version of the council then they too can initialise proposals. Recall that the council is the body of electives that are specific to each journal instance. They handle important governance issues and act on behalf of passive stakeholders specific to the journal in which they belong to. Whereas the Board have the same responsibility but over the global community so their scope is larger.

Treasury Proposals.

Another important mechanism that is completely controlled by the governance protocol is access to the platform Treasury. Although we have not discussed the functionality and tole of the treasury in detail, yet we will do so in the next section. However, in short, the treasury is an accumulation of funds in the form of external funding, donations, rejected proposal token redistribution and tokens from slashed validators. The purpose f the treasury is to help fund emerging scientific endeavours and also to keep the platform sustainable as time ages, so that there is always available capital for spending on a variety of things that help the platform thrive. However, it goes without sating that no one person can spend the treasury's funds on the whim. All treasury spending proposals go through the same voting processes as we have discussed thus far. However, unlike the case of proposing new journal entities, the public can not issue proposals for treasury spending. These privileges lie with the councils and board members. A common use case of treasury spending is to fund projects that researchers in the overarching community are working on. For example, the board could listen to the public and choose to create a proposal to fund n already existing project or create a new project from scratch. In the case of journal council boards, they can also create proposals to access the treasury funds from listening internally to the micro-community of their specific journal entity. The only difference is, like before if one of the many councils approve a treasury spending proposal, just like in the example above, the proposal will enter a second round of voting and must be approved by the board of electives before being enacted. This safeguard is just to add extra security and to ensure that the proposal is aligned with the views of the entire community. For example, in the case that a council approves a proposal on a given project specific to their scientific niche, there could be a scenario where the vote is biased, therefore we can filter out any of this bias by using the method of second round voting. However, if the Board makes a proposal for treasury's spending a one round vote carries. Each journal entity also contains there very own unique treasury contract in which funds will be sent from the main platform treasury on the acceptance of a treasury spending proposal. From here other internal smart contracts will decide how the funds will be allocated and siphoned into circulation over time. This is one section of our architecture that is still in its infancy in regard to exactly how the internal of the spending of funded money will work. For acknowledgement od this fact please see the last section in this report of Future work.

Slashing Proposals.

The last important implementation of our governance protocol that we will highlight here (albeit that there are many more) is slashing of community stakeholders. Anyone in the community is eligible for slashing. Users are generally slashed when the misbehave, act maliciously or in a manner that is not in line with the views of the network and for being inactive (specific for validators). In our model slashing is open to anyone. Everyone can create a slashing proposal provided they have good evidence to back their claim. Unlike many other slashing mechanisms in other blockchains like Ethereum or Polkadot where slashing is limited to validators, in our architecture we encourage anyone and everyone to report suspicious behaviour in return for an incredibly high token reward that is much higher than other methods of earning passive income or rewards. By this incentivisation we hope that stakeholders will be more active in trying to root our bad actors in the network or be more likely to report malicious intent instead of staying quiet because of affiliation for example. In our governance protocol the approval of a slashing proposal is handled again by the council and board only. This is for a variety of reasons. One being that we cannot expected everyone in the community to have enough knowledge about individual acts of mal intent, especially if a large art of the community is far removed from the instance. Therefore, allowing the community to vote would mostly likely result in very low turnouts and/or invite some stakeholders to vote just for the sake voting. Therefore, slashing proposals are strictly handled by the councils and board members. When a regular community member, validator or council/board member thinks a user should be slashed they can issue a proposal. This type of proposal is different tin that the issuer must also include metadata containing strong evidence to back their claim. This evidence will be used to inform the council/board of the situation so that they can carry out and investigation in which the ultimate decision will be based off. From here the regular two round vote will carry, if the issuer of the proposal is a regular community stakeholder, whereby the origin of the proposal is from within one the journals. Even if it is a council member who issues the proposal the same two round vote will carry. However, if the origin of the slashing proposal comes from one the board members then a one round vote will be used. It should also be noted that council members can slash other council members and this apples to the board also. So in other words anyone can slash anyone regardless of their position in the network or reputation. In the scenario where a slashing proposal is accepted the issuer or issues will be rewarded from the funds in the treasury and their reputation will be increased. On the other hand the reputation of the attacker will be severely reduced however, and they will loose a large percentage of their staked tokens also. The severity of the reduction is based on the type of slashing event. However, since different convictions have varying ranges of severity it is possible for the accused to fight his case if he can provide sufficient evidence. In many cases although this will probably not be enough to completely reverse the slash, if enough evidence is provided the severity could be reduced. This is for cases where the accused made a genuine or unavoidable mistake. This is explained in more detail in the section on slashing, so we will refrain for reiteration. A diagram of the flow of a slashing proposal is given below

7.0 The Treasury

In the last section we discussed the democratic governance model that underpins the evolution of our ecosystem over time. We discussed how the flow of money is distributed throughout the system through various mechanisms such as rewarding or panelising validators, issue spending proposals for funding research endeavours etc. A lot of this monetary spending originates from the platform treasury. The treasury is a stand-alone smart contract which links to many other areas of our code base. As stated, the main purpose of the treasury is to fund the whole ecosystem so that it stands as a sustainable entity. The main methods of funding the treasury are through transaction fees, slashing, staking and from outside funding. These funds will be held in the treasury smart contract, and they can only be spent by making a proposal that if approved by the network (see section on governance) will enter a waiting period and after this, the funds will then be distributed out according to the details set out by the spending proposal. The treasury smart contract will link tightly with the governance smart contact. This way, the funds will always be locked away and can only be spent when the community as a whole agrees on some proposition.

7.1 Sustainability Of Staking

One major problem that arises when looking at any incentivised token reward schemes is the issue of sustainability. If a system has a fixed sum of money, then how can it be sustainable to reward members of the community for good participation when there is no inflow of money on the other end. Eventually the money will run out and there will be no way to keep rewarding stakeholders, this is also true if external investors decide to fund the network, eventually they too will also need to be paid back. Therefore, the Treasury also serves another fundamental purpose to keep the ecosystem running smoothly. That is whenever someone decides to join the network, be it to become a validator or to join the network so they can participate in voting, recall that they must own and stake DST. For validators there will be a pre-defined minimum in regards to how much tokens they must lock up. The locked tokens must remain locked until the validator decides to leave or resign from their role There is also a delayed withdrawal period that will be set in place that a stakeholders (specially validators) must weight through once they initiate a withdrawal request. This is to prevent anyone from potentially breaking the rules and unlocking their stake before the network has time to react. Each validators staked tokens are held in reserves in the treasury. Here the funds will be put to work by the treasury smart contract. In other words there tokens will be lent out to others with interest. The details about how this will be done have not been fully scoped yet and this concept is outside of the scope of this paper. However, the idea of decentralised lending is a common theme is web3 applications when It comes to sustainability. The most viable scenario is to lend out the locked tokens on a trusted platform such as Aave **[51]** which we will discuss below. The other method that rewards will draw from, is through the inflation of the currency supply of DST. Each time a paper gets accepted or rejected, a sum of DST will be minted to make up part of the each validators reward. This is similar to how bitcoin mines new bitcoins each time a miner carries out “work” by solving computationally expensive mathematical puzzles.

Aave is a decentralized lending system that allows users to lend, borrow and earn interest on crypto assets, all without middlemen. Running on the Ethereum blockchain, Aave instead is a system of smart contracts that enables these assets to be managed by a distributed network of computers running its software. This means Aave users do not need to trust a particular institution or person to manage their funds. They need only trust that its code will execute as written. At its core, the Aave software enables the creation of lending pools that enable users to lend or borrow multiple different crypto currencies. Like other decentralized lending systems on Ethereum, Aave borrowers must post collateral before they can borrow. Furthermore, they can only borrow up to the value of the collateral they post. Borrowers receive funds in the form of a special token known as an “aToken”, which is pegged to the value of another asset. This token is then encoded so lenders receive interest on deposits. A borrower may post collateral in DAI, for example, and borrow in ETH. This allows a borrower to gain exposure to different cryptocurrencies without owning them outright. Aave can also introduce additional features, such as instant loans, and other forms of issuing debt and credit that take advantage of the unique design properties of blockchains.

The fact that Aave requires borrowers on the other side to be over-collateralized means that the lender has a 100% guarantee that they can not be conned or cheated. This is because when a borrower wants to take out a loan, they must have already locked up the value of their loan before being eligible to borrow tokens. If the scenarios arose, where the borrower decided to not pay back their loan, or were late in doing so, then their locked tokens would be liquidated and send to the lender. This way no matter the scenario the lender can be always be certain that they will make back there money including interest. The idea of using the locked tokens of stakeholders to passively earn interest is not new for Defi applications and in fact every yield farming application employs this technique in order to reward their users with the Return on investment (ROI’s) that they promise. So this is in essence one aspect of how the treasury will be able to sustain the token rewards it pays out to validators and good community participants. In order to guarantee a robust fool proof system that will not be hacked a lot of research needs to be done to develop out a model for this, which unfortunately could form the contents of its own paper which will come in the future, but for now its important to be aware of the concept. However, in such an implementation, the diagram below shows how the setup would look

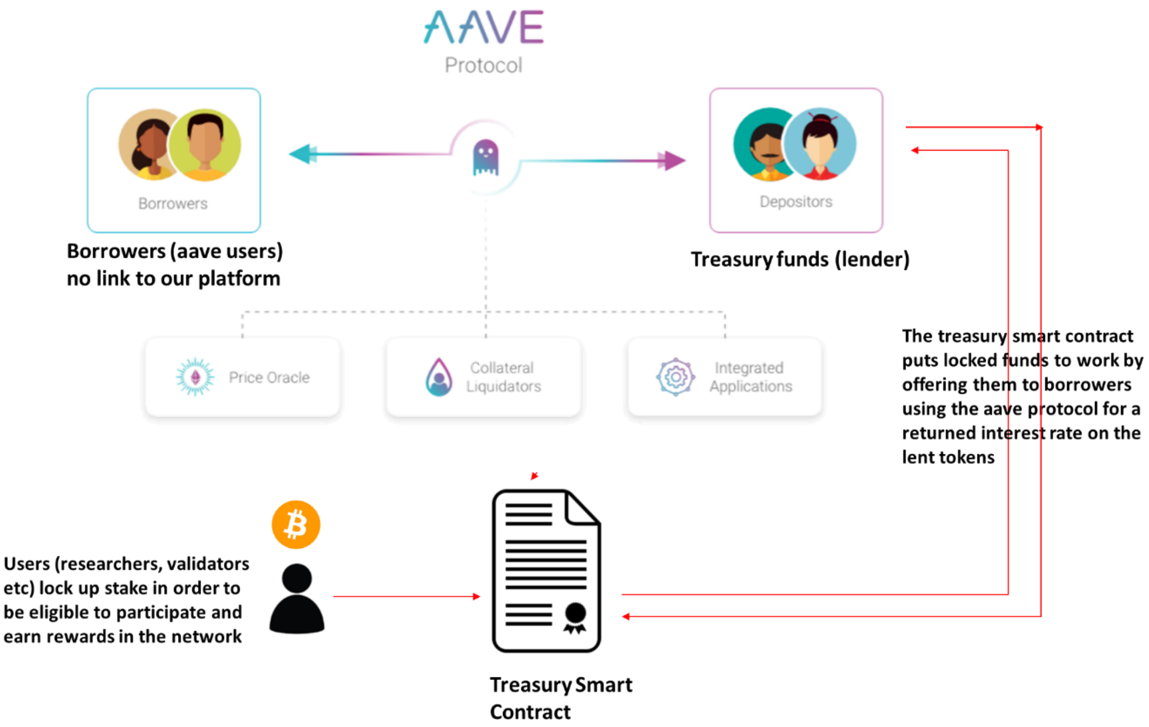

AAVE
Protocol
Borrowers
Depositors
Borrowers (aave users)
no link to our platform
Treasury funds (lender)
The treasury smart contract puts locked funds to work by offering them to borrowers using the aave protocol for a returned interest rate on the lent tokens
Price Oracle
Collateral Liquidators
Integrated Applications
Users (researchers, validators etc) lock up stake in order to be eligible to participate and earn rewards in the network
Treasury Smart Contract

Figure 26: Theoretical example of the possibility of using aave to earn interest (passive income) on the treasury funds to use for rewarding network stakeholders for good deeds

As we mentioned above the second form of sustainability comes in the form of inflation. In our design we propose that validators will get rewarded not from the slow accumulation of interest gained from the lending of tokens of the treasury side, but rather from the inflation of the supply of DST. This is the same way in which Bitcoin and Ethereum currently reward their validators. On Bitcoin for example, whenever a miner mines a new, 6.5 (as the time of writing this paper) new bitcoins get pumped into circulation and distributed amongst the miners who minded that block. In our application we propose a similar idea, such that whenever an new epoch begins, then a certain sum of DST will be minted and sent to the treasury smart contract to be used as a means to reward acts of good faith in the network for the current epoch, amongst the validators and regular community stakeholders. The details about how this will be done have not been fully explored but unlike bitcoin, out system will most likely not have a fixed amount of DST minted at each epochs end, but rather the defined amount will stem from the total sum of the rewards that each participating validator is due. Again the concept of sustainability through decentralised lending and by inflationary means should be the subject of their own economic paper that stems from this design. The main problem is not the execution of such systems in our code but rather the development of proofs to make sure that said systems are well audited and resistant to hacks. More on this in our section below on limitations and future work

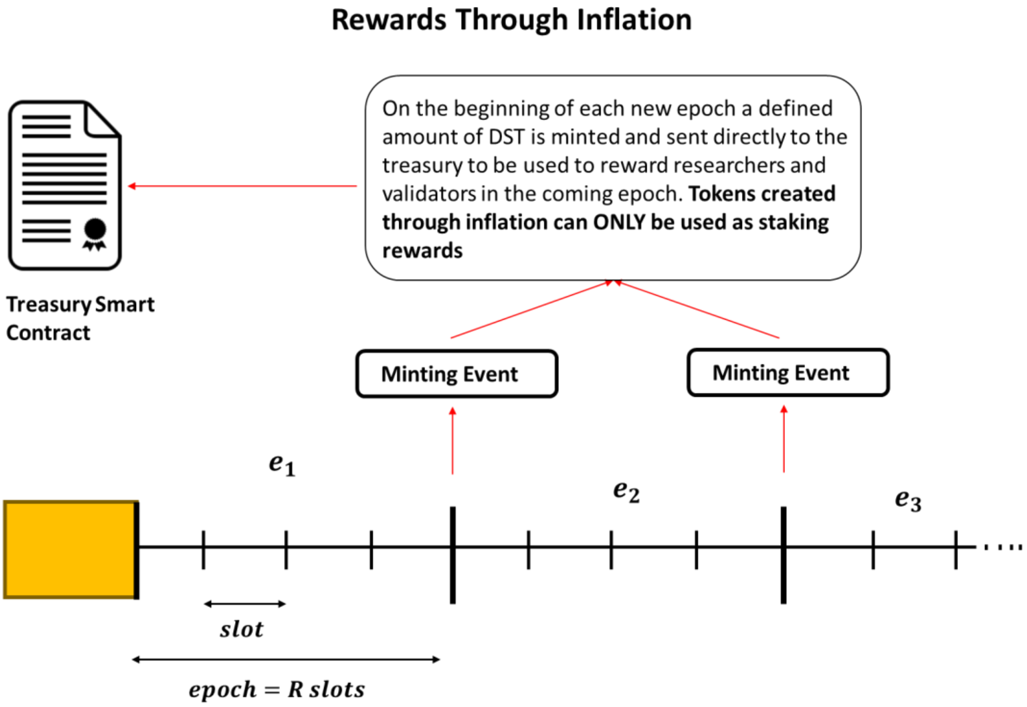

Rewards Through Inflation
On the beginning of each new epoch a defined amount of DST is minted and sent directly to the treasury to be used to reward researchers and validators in the coming epoch. Tokens created through inflation can ONLY be used as staking rewards
Treasury Smart Contract
Minting Event
Minting Event
e1
e2
e3
slot
epoch = R slots

Figure 27: An example of how the inflation of DST can be used to generate rewards used to reward network stakeholders for good deeds

9.0 Limitations & Future Work

In this section, we describe in the context of our third research question challenges and research potentials that we identified during our analysis. Future works should address them in order to eliminate technological and legal insecurities and to enhance the usability of the BT for open science and beyond. We focused on some of the most relevant and promising topics in our view, which have yet to be fully or insufficiently investigated yet. They shall provide an impulse in the form of starting points for further research in the development of our model. As a positive side effect, addressing these issues can partially also foster other non-scientific areas. We want to point out that the challenges presented in this section are very complex and profound, so we do not expect them to get resolved in the near future. For example, the correctness problem of software which is fundamental to smart contracts is around since the early days of programming, and till today a solution is not yet in sight. Therefore, the following topics are an outlook into vital pillars that need to be considered in the course of a broad integration of BT

9.1 Risks & Smart Contract Validation

Trustworthiness is a key element of BT and one of its main drivers, so as developers we should design all aspects in their applications in a way to support and provide that property. In this regard, we see SCs that get used in many projects as critical because they can offer various possibilities for malicious behaviour and are prone to crucial coding errors in their development. The ability to use Turing-complete programming. Blockchain for Open Science opens up not only numerous use cases and functionalities but also increases the complexity and thus the potential for human mistakes and the number of backdoors/exploits. These can cause, for example, crashes of the processes or vulnerabilities of the program itself that may allow hackers to steal the resources that a digital contract manages **[49] [50]**. The novelty of SCs justifies the circumstance that the common knowledge about their design, implementation, programming, and validation is not well developed yet. One approach to counteract vulnerabilities of SCs is to limit the expressiveness of the underlying programming language **[51]**. Another possibility is the several commercial providers of audit services that have got founded in the last years. They are checking SCs to make sure they fulfill their purpose without eventual weak points. Examples are Runtime Verification29 and Securify30. In that sense, we see research potential in investigating ways to automate the formal verification of SCs through software to quickly eliminate the possibility of specific attacks **[49][52].** A further approach can be a modular construction kit to be able to build digital contracts piece by piece for reliable, simple applications. Hence no great coding skills are required, and the creation process gets eased, similar to OpenZeppelin (see appendix). Also, standards can generally improve the design procedure and security. There is still much to do on this topic to enable an efficient and secure large-scale use of SCs for all application areas

9.2 Missing Standardisation & Frameworks

Established standards and frameworks for technologies can be vital and bring several advantages with them like time-saving, error prevention, and increased security. Through our analysis, we have concluded that these are largely absent in BT. So far, blockchain developers have taken a pioneering role and mostly programmed their applications in different languages without technical specifications. Thus, many unique application structures emerged that have their advantages and disadvantages as well as security risks and vulnerabilities. Standards for BT can help to foster its adoption, interoperability, make systems more secure, in particular, build trust **[53].** Also, they enhance the accessibility into the general development of blockchain applications. In terms of software communication, standardized APIs can make the design of new interfaces redundant in most cases. There is still a lot of potential in researching suitable standards and frameworks for the BT, for example, to ease the design and development of blockchain-based software, or to integrate a blockchain into research workflows. Also interesting are unified methods of how academic publishers can use this technology to improve certain of their processes and benefit from it. In our opinion, infrastructural frameworks like Hyperledger will play an even more prominent role in the future in creating a variety of new applications. One general goal of standards and frameworks must be to facilitate the entry into blockchains in order to address non-experts and break down access barriers. Altogether, both topics offer a lot of promising research possibilities, and we think they will be a cornerstone of the BT in the future.

9.3 Legal Uncertainties

Some research has already been done on blockchain-based cryptocurrencies **[54]**, SCs, and DAOs **[55]** in connection with legal issues and topics, but there is still a lot of demand for further work and clarification (Werbach, 2018). Several blockchain projects we analysed are relying, for instance, on timestamps to prove different aspects like the existence of specific information at a certain time or want to issue certificates to verify the ownership of digital assets. A concrete example is the timestamping of a dashcam recorded video that shows a car accident to confirm the moment of the crash and the authenticity of the video along with other details that can be important for the decision of a legal process. The question is, what is the legal status and acceptance when such blockchain-based evidence gets used in a lawsuit? In the case of that uncertainty, we see it as problematic that a few analysed projects work with promises which are not juridically secured. Further, SCs are also legally unspecified. For example, what happens if resources managed by them are no longer tangible or lost due to incorrect programming; which party is to blame and how does compensation work? SCs or DAOs can barely cover all possible real-world case constellations within their program code. In this respect, is there a technical or non-technical way to deal with unforeseen events? More questions are how juristic systems should treat SCs compared to traditional ones, and what possibilities exist to secure the contracting parties **[55]**? A general challenge is the different laws and courts in every country or state **[56]**, which mean that a solution that functions in a particular location is unlikely to work in all other places. So, most likely, there will not be a global consensus, but countrywide specifications would eliminate many legal uncertainties. With the increasing importance of BT and its growing adoption, we believe that juridical topics are playing a major role in the future and should be addressed to support further developments.

9.4 Blockchain Economics for Self-Sustainability

One of the more complex features that we discussed in the Treasury section (section 7) was the fact that in order for our platform to remain sustainable, we would need to implement complex blockchain economical tactics and techniques in the form of sustainability through inflation and decentralised lending. Although in the realm of typical DeFi applications, the enabling of self-sustainable protocols that are able to both keep a constant supply of money whilst simultaneously rewarding stakeholders is common place, however, the actually development of such protocols in the sense that they are well audited and secure is an extremely endearing task and in many cases requires financial experts and extremely talented protocol developers in order to root out and potential loopholes in the creation of such schemes. Over the past few years there has been countless “hacks” on some of the top regarded decentralised autonomous organisations (DAO’s) that have seen the perpetrators run off with millions and sometimes billions of dollars’ worth of users’ crypto assets. Thus, this poses a severe limitation on one of the most crucial aspects of our platform, as being self-sustained is arguably the most important feat to accomplish if we wish our platform to last indefinitely with the times. Therefore, much research needs to be explored into this topic because although it is programmatically easily to inflate our DST to create rewards for users and although it is easy to programmatically execute lending on platforms such as aave to earn interest, we need to be absolutely certain that our code can not be ever exploited. Thus, these crypto-economic concepts should be explored in their own paper which would serve as a follow up of future work to the general model that w have developed in this paper

Conclusion

This paper proposes explores the development of an digital peer-review platform that implements blockchain technology to streamline various processes in the scientific business model such as peer-review amongst other things. Our architecture aims to accomplish a variety of goals namely, the opening and decentralization of different peer review and publication functions such as the selection and recognition of peer reviewers, the distribution of scientific knowledge, and the peer review process communication. Arguably, this decentralization of the infrastructure could help to challenge the central role of middlemen such as traditional publishers. Distributed technologies such as blockchain and IPFS may finally realize the promise of Open Access, while enabling new models of science dissemination. Opening and decentralizing the infrastructure enhances the transparency and accountability of the system, and may provide a new arena to foster innovation. Note that the proposed system adopts a crypto-economical market based approach where the inclusion of an ERC20 utility token serves as the basis of economic transactions, where the community strives to perform good deeds through incentivisation tactics and techniques.

The transparency provided by opening the peer review process allows the construction of a reputation system of reviewers, but also raises concerns about privacy and fairness. Furthermore, the introduction of a new public metric (reviewers’ reputation) may also affect researcher careers, adding pressure to the already straining processes for academic survival. However, it is noted here that much of the work we laid forth is purely theoretical and the next step in the development of our ecosystem needs to be the actual creation of the platform itself. Some challenges of the system remain open as future work, such as the detection and prevention of fake reviews, or revenge ratings to game the reputation system. Blockchain technologies can be used to replicate the privacy settings currently used in peer review processes. However, Blockchain can also be used to introduce a new review model that supports the accountability of peer reviewing while maintaining the anonymity of blind and double blind reviews to improve fairness. The implications of such accountable, open and anonymous review models are still to be revealed, since an incentive based reputation system it could also support negative dynamic changes such as increasing competitive dynamics, or gender bias. Additionally, the proposed system's infrastructure relies on new technologies with their own challenges.

The development of the model in this paper was not a straightforward process and many challenges have been raised some of which include the fact that blockchain technologies face scalability issues, transaction costs, inclusiveness and usability problems that remain open and under discussion. On the other hand, distributed file systems such as IPFS may be more resilient, but they still need somebody in charge of preserving and providing data, since without that responsible actor, it may result in an unpredictable loss of content. Considering these archiving issues, whether this new technologies will allow the creation of durable science repositories able to interoperate with legacy, current and future systems remain open. Other open issues that require further research and may be explored in future work are the implementation of the proposed privacy settings, the exploration of different copyright regimes, the challenging of traditional journal-cantered metrics to rate publication quality, different reputation algorithms, different levels of openness, and the exploration of decentralized autonomous journals. This is not to mention the further work that needs to be conducted in relation to the self-sustainability tactics needed to make it possible to reward good behaviour in the first place, through the inflation of our DST utility currency and from decentralised lending methods used to earn interest on locked tokens in the treasury through lending protocols such as Aave.

Despite the existing challenges, we are confident that decentralizing the processes that Science relies on, would open up a whole new playing field, with implications we cannot possibly foresee. Nevertheless, we conclude that the technology may already make substantial contributions in a variety of areas, such as enhancing researchers' present workflows, developing confidence in technological systems, allowing new partnerships, and minimizing existing challenges. However, a considerable of more work still needs to be done in terms of standards, governance models, user friendliness, interfaces, security and legal challenges, and instructional efforts to fully realize the technology's promise. We expect the BT to become more mature over time as its adoption expands. In this sense, overcoming the mentioned difficulties will be critical in the future. Altogether, after our review, we summarize that the capabilities of the BT for open science are by far not exhausted yet. We conclude that the technology can have a positive impact on scientific work and its open ecosystems but that primarily depends on the scientific community's and all other connected stakeholders' approval of the technology, which is currently uncertain. However, on that note we should conclude with the final realisation that the a simple shift in direction in regard to the way processes are enacted or implemented in academia such as through the implementation of an emerging technology such as BT or any other, could not possibly boast the claim to solve everything. So much more needs to be done but we do want to highlight that blockchain technology "could" possibly act as a cog in the entire system to help realise the vision and ethos of science

Appendix

1. **Colored Coins:** The term "Colored Coins" loosely describes a class of methods for representing and managing real world assets on top of the Bitcoin Blockchain through the attachment of transactional metadata which can be tracked and referenced.

2. **Tokenomics**: Tokenomics is the topic of understanding the supply and demand characteristics of cryptocurrency

3. **Prediction Markets:** The prediction market is a market where people can trade contracts that pay based on the outcomes of unknown future events. The market prices generated from these contracts can be understood as a kind of collective prediction among market participants. These prices are based on the individual expectations and willingness of investors to put their money on the line for those expectations.

4. **Arbitrage**: With foreign exchange investments, the strategy known as arbitrage lets traders lock in gains by simultaneously purchasing and selling an identical security, commodity, or currency, across two different markets. This move lets traders capitalize on the differing prices for the same said asset across the two disparate regions represented on either side of the trade.

5. A **smart contract** is a self-executing contract with the terms of the agreement between buyer and seller being directly written into lines of code. The code and the agreements contained therein exist across a distributed, decentralized blockchain network. The code controls the execution, and transactions are trackable and irreversible.

6. **On-chain governance** is a system for managing and implementing changes to cryptocurrency blockchains. In this type of governance, rules for instituting changes are encoded into the blockchain protocol. Developers propose changes through code updates and each node votes on whether to accept or reject the proposed change.

7. in an **off-chain governance** network, **stakeholders compete for control** by collaborating in a variety of ways. Discussions on social media, online forums, conferences, and other events are some popular examples of off-chain governance procedures on public blockchains.

8. **Fault tolerance** refers to the ability of a system (computer, network, cloud cluster, etc.) to continue operating without interruption when one or more of its components fail.

9. A **consensus mechanism** is a fault-tolerant mechanism that is used in computer and blockchain systems to achieve the necessary agreement on a single data value or a single state of the network among distributed processes or multi-agent systems, such as with cryptocurrencies. It is useful in record-keeping, among other things.

10. **Coin age** is determined by how many idle days old the sake is, multiplied by the amount of coin at stake. If the validators stake is idle for 30 days or more, they may be selected to forge the next block.

11. **Polkadot** is a software that seeks to incentivize a global network of computers to operate a blockchain on top of which users can launch and operate their own blockchains.

12. **Web3 wallets** are essentially digital wallets. As such, they have the ability to store digital assets. This includes everything from fungible to non-fungible (NFTs) tokens. Second, a Web3 wallet also opens the door to the crypto realm, allowing you to interact with dApps on various blockchains.

13. **MetaMask** is a software cryptocurrency wallet used to interact with the Ethereum blockchain. It allows users to access their Ethereum wallet through a browser extension or mobile app, which can then be used to interact with decentralized applications. MetaMask is developed by ConsenSys Software Inc., a blockchain software company focusing on Ethereum-based tools and infrastructure.

14. The basic idea behind **Web3 Authentication** is that it's cryptographically easy to prove the ownership of an account by signing a piece of data using a private key. If you manage to sign a precise piece of data generated by our back end, then the back end will consider you the owner of that public address. Therefore, we can build a message-signing-based authentication mechanism with a user's public address as their identifier.

15. **A private key**, also known as a secret key, is a variable in cryptography that is used with an algorithm to encrypt and decrypt data. Secret keys should only be shared with the key's generator or parties authorized to decrypt the data.

16. **OpenZeppelin** provides security products to build, automate, and operate decentralized applications. We also protect leading organizations by performing security audits on their systems and products.

17. **EVM Layer-2 chain** refers to a network or technology that operates on top of an underlying blockchain protocol to improve its scalability and efficiency.

18. **3rd Gen Blockchain:** Third-gen blockchains aim to resolve fundamental flaws including scalability and interoperability which means blockchain can sustain mass adoption and not suffer problems like slow transaction time and closed systems

19. **The double spending problem:** The double spending problem is a phenomenon in which a single unit of currency is spent simultaneously more than once. This creates a disparity between the spending record and the amount of that currency available.

20. **51% attack:** A 51% attack refers to an attack on a blockchain—most commonly Bitcoin, for which such an attack is still hypothetical—by a group of miners controlling more than 50% of the network's mining hash rate or computing power.

21. **Inter Planetary File System (IPFS):** IPFS is a peer-to-peer (p2p) storage network. Content is accessible through peers located anywhere in the world, that might relay information, store it, or do both. IPFS knows how to find what you ask for using its content address rather than its location.

22. **Permissioned blockchain:** Permissionless blockchains are blockchains that require no permission to join and interact with. They are also known as public blockchains. Most of the time, permissionless blockchain is ideal for running and managing digital currencies.

23. **Permissionless blockchain:** Permissionless blockchains are blockchains that require no permission to join and interact with. They are also known as public blockchains. Most of the time, permissionless blockchain is ideal for running and managing digital currencies.

24. **The Byzantine Fault Tolerance Problem:** the "Byzantine Generals Problem", developed to describe a situation in which, in order to avoid catastrophic failure of the system, the system's actors must agree on a concerted strategy, but some of these actors are unreliable.

25. **Side Chains:** A sidechain is a side blockchain that is linked to another blockchain, referred to as the main chain, via a two-way peg. They are usually used to store data off-chain that would otherwise be too costly to store on the main blockchain.

26. **Web3.**js: web3.js is a collection of libraries that allow you to interact with a local or remote Ethereum node using HTTP, IPC or WebSocket. ... js as well as providing an API reference documentation with examples

27. **Genesis Block:** A Genesis Block is the name given to the first block a cryptocurrency, such as Bitcoin, ever mined

28. **Coin Voting:** Coin voting in blockchain is a way for community members to have a say in the decisions that are made in a blockchain protocol. They must have skin-in-the-game to have a say in votes. The more coins or stake someone has the greater their voting power

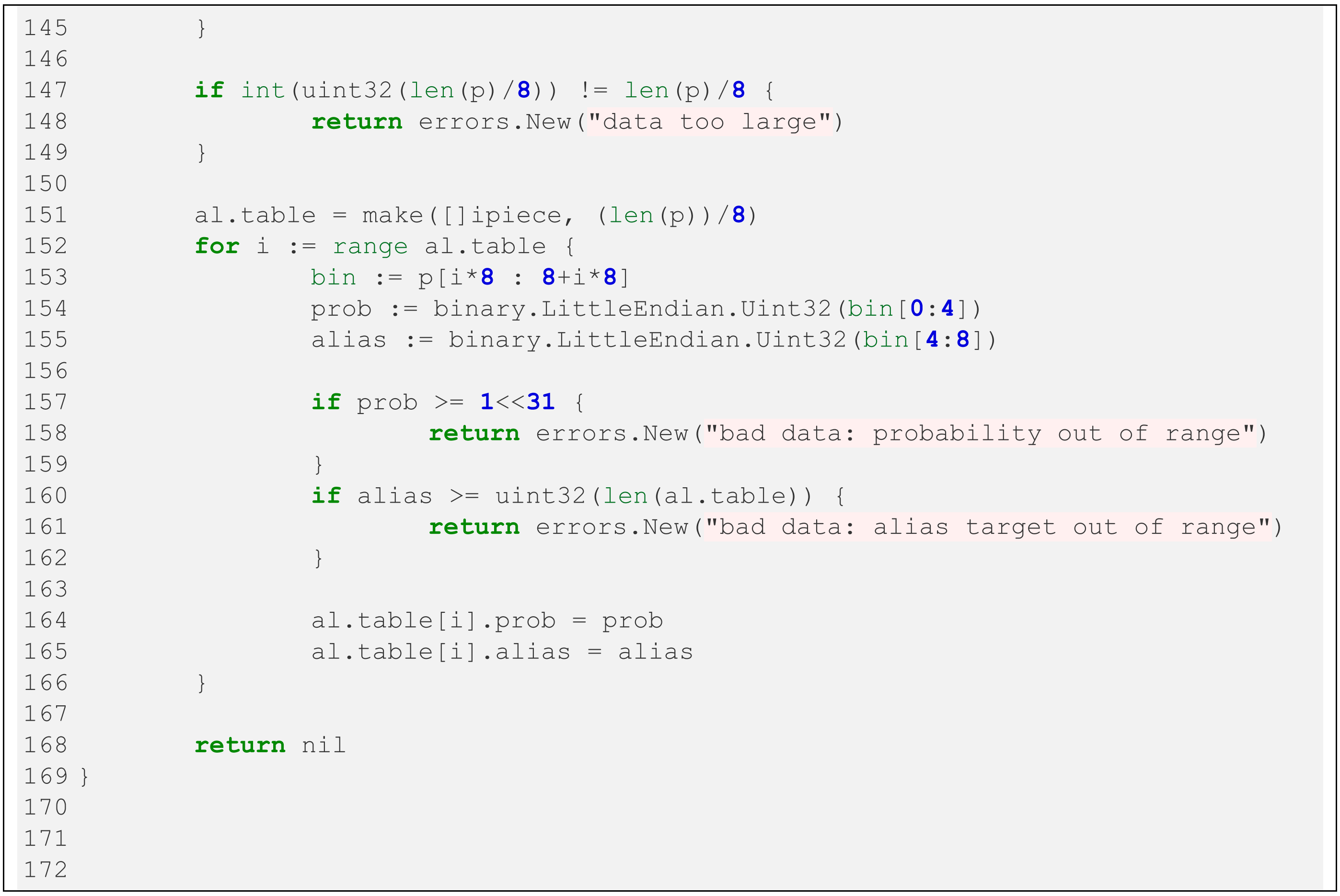
29. **Forking:** Blockchain forks are essentially a split in the blockchain network. The network is built on an open source software, and the code is freely available. Forks occur when the software of different miners disagree over the best way forward for blockchain. It's up to miners to decide which blockchain to continue using. This disagreement causers miners to work the now two different forks the original blockchain before and the new one with changed protocols after it

Implementation Of WRS algorithm discussed in section4 implemented in the GO programming language as by the WRS algorithm inspired by **[45]**

```
// The alias package picks items from a discrete distribution
// efficiently using the alias method.


import (
	"encoding/binary"
	"errors"
	"math/rand"
)

type Alias struct {
	table []ipiece
}

type fpiece struct {
	prob  float64
	alias uint32
}

type ipiece struct {
	prob  uint32 // [0,2^31)
	alias uint32
}

// Create a new alias object.
// For example,
//   var v = alias.New([]float64{8,10,2})
// creates an alias that returns 0 40% of the time, 1 50% of the time, and
// 2 10% of the time.
func New(prob []float64) (*Alias, error) {

	// This implementation is based on
```

```go
        // http://www.keithschwarz.com/darts-dice-coins/

        n := len(prob)

        if n < 1 {
                return nil, errors.New("too few probabilities")
        }

        if int(uint32(n)) != n {
                return nil, errors.New("too many probabilities")
        }

        total := float64(0)
        for _, v := range prob {
                if v <= 0 {
                        return nil, errors.New("a probability is non-positive")
                }
                total += v
        }

        var al Alias
        al.table = make([]ipiece, n)

        // Michael Vose's algorithm

        // "small" stack grows from the bottom of this array
        // "large" stack from the top
        twins := make([]fpiece, n)

        smTop := -1
        lgBot := n

        // invariant: smTop < lgBot, that is, the twin stacks don't collide

        mult := float64(n) / total
        for i, p := range prob {
                p = p * mult

                // push large items (>=1 probability) into the large stack
                // others in the small stack
                if p >= 1 {
                        lgBot--
                        twins[lgBot] = fpiece{p, uint32(i)}
                } else {
                        smTop++
                        twins[smTop] = fpiece{p, uint32(i)}
                }
        }

        for smTop >= 0 && lgBot < n {
                // pair off a small and large block, taking the chunk from the large block that's wanted
                l := twins[smTop]
                smTop--

                g := twins[lgBot]
```

```go
		lgBot++

		al.table[l.alias].prob = uint32(l.prob * (1<<31 - 1))
		al.table[l.alias].alias = g.alias

		g.prob = (g.prob + l.prob) - 1

		// put the rest of the large block back in a list
		if g.prob < 1 {
			smTop++
			twins[smTop] = g
		} else {
			lgBot--
			twins[lgBot] = g
		}
	}

	// clear out any remaining blocks
	for i := n - 1; i >= lgBot; i-- {
		al.table[twins[i].alias].prob = 1<<31 - 1
	}

	// there shouldn't be anything here, but sometimes floating point
	// errors send a probability just under 1.
	for i := 0; i <= smTop; i++ {
		al.table[twins[i].alias].prob = 1<<31 - 1
	}

	return &al, nil
}

// Generates a random number according to the distribution using the rng passed.
func (al *Alias) Gen(rng *rand.Rand) uint32 {
	ri := uint32(rng.Int31())
	w := ri % uint32(len(al.table))
	if ri > al.table[w].prob {
		return al.table[w].alias
	}
	return w
}

// MarshalBinary implements encoding.BinaryMarshaller.
func (al *Alias) MarshalBinary() ([]byte, error) {
	out := make([]byte, len(al.table)*8)
	for i, piece := range al.table {
		bin := out[i*8 : 8+i*8]
		binary.LittleEndian.PutUint32(bin[0:4], piece.prob)
		binary.LittleEndian.PutUint32(bin[4:8], piece.alias)
	}
	return out, nil
}

// UnmarshalBinary implements encoding.BinaryUnmarshaller.
func (al *Alias) UnmarshalBinary(p []byte) error {
	if len(p)%8 != 0 {
		return errors.New("bad data length")
```



```
        }

        if int(uint32(len(p)/8)) != len(p)/8 {
                return errors.New("data too large")
        }

        al.table = make([]ipiece, (len(p))/8)
        for i := range al.table {
                bin := p[i*8 : 8+i*8]
                prob := binary.LittleEndian.Uint32(bin[0:4])
                alias := binary.LittleEndian.Uint32(bin[4:8])

                if prob >= 1<<31 {
                        return errors.New("bad data: probability out of range")
                }
                if alias >= uint32(len(al.table)) {
                        return errors.New("bad data: alias target out of range")
                }

                al.table[i].prob = prob
                al.table[i].alias = alias
        }

        return nil
}
```

References

1. Beckerguides.wustl.edu. 2022. BeckerGuides: Tools for Authors: What is the h index?. [online] Available at: <https://beckerguides.wustl.edu/authors/hindex#:~:text=The%20h%20index%20is%20a%20metric%20for%20evaluating%20the%20cumulative,have%20not%20yet%20been%20cited.> [Accessed 21 February 2022].
2. Jenns paper
3. My lit revew

4. Smith, R., 2022. [online] Available at: <https://www.jstor.org/stable/25466452> [Accessed 21 February 2022].

5. G. Eysenbach, “Citation advantage of open access articles,” PLoS biology, vol. 4, no. 5, p. e157, 2006.

6. R. Walker and P. Rocha da Silva, “Emerging trends in peer review—a survey,” Frontiers in Neuroscience, vol. 9, May 2015.

7. B. Whitworth and R. Friedman, “Reinventing academic publishing online. part i: Rigor, relevance and practice,” First Monday, vol. 14, no. 8, 2009.

8. J. A. Evans and J. Reimer, “Open access and global participation in science,” Science, vol. 323, no. 5917, pp. 1025–1025, 2009.

9. V. Lariviere, S. Haustein, and P. Mongeon, “The ` oligopoly of academic publishers in the digital era,” PloS one, vol. 10, no. 6, p. e0127502, 2015.

10. R. Van Noorden et al., “The true cost of science publishing,” Nature, vol. 495, no. 7442, pp. 426–429, 2013.

11. M. Ware, “Peer review: benefits, perceptions and alternatives,” Publishing Research Consortium, vol. 4, pp. 1–20, 2008

12. B

13. E. Ford, “Defining and characterizing open peer review: A review of the literature,” Journal of Scholarly Publishing, vol. 44, no. 4, pp. 311–326, 2013.

14. P. Frishauf, “Reputation systems: a new vision for publishing and peer review,” Journal of Participatory Medicine, 2009.

15. X. Shuai, A. Pepe, and J. Bollen, “How the scientific community reacts to newly submitted preprints: Article downloads, twitter mentions, and citations,” PloS one, vol. 7, no. 11, p. e47523, 2012.

16. Y. Benkler, “Degrees of freedom, dimensions of power,” Daedalus, vol. 145, no. 1, pp. 18–32, 2016.

17. A. Chaudhry, J. Crowcroft, H. Howard, A. Madhavapeddy, R. Mortier, H. Haddadi, and D. McAuley, “Personal data: thinking inside the box,” in Proceedings of The Fifth Decennial Aarhus Conference on Critical Alternatives, pp. 29–32, Aarhus University Press, 2015.

18. Pandey, I., 2022. *"Web3 will revolutionize how the world interacts with the Internet of Value"HackerNoon*. [online] Hackernoon.com. Available at: <https://hackernoon.com/web3-will-revolutionize-how-the-world-interacts-with-the-internet-of-value> [Accessed 16 March 2022].

19. S. Bartling and B. Fecher, “Blockchain for science and knowledge creation. zenodo,” Publisher Full Text, 2016. Page 4643

20. J. P. Tennant, J. M. Dugan, D. Graziotin, D. C. Jacques, F. Waldner, D. Mietchen, Y. Elkhatib, L. B. Collister, C. K. Pikas, T. Crick, et al., “A multi-disciplinary perspective on emergent and future innovations in peer review,” F1000Research, vol. 6, 2017.

21. A. C. Kade Morton, "Aletheia: blockchain for scientific knowledge with a community management framework," 2017.

22. V. Dhillon, "From bench to bedside: Enabling reproducible commercial science via blockchain," Bitcoin Magazine, 2016.

23. Furlanello, C., De Domenico, M., Jurman, G., and Bussola, N. (2017). Towards a scientific blockchain framework for reproducible data analysis. Comput. Res. Reposit. arxiv: 1707.06552. Available online at: https://arxiv.org/abs/1707.06552

24. Collins, F. S., and Tabak, L. A. (2014). Policy: NIH plans to enhance reproducibility. Nature 505, 612–613. doi: 10.1038/505612a

25. Schiltz, M. (2018). Science without publication paywalls: cOAlition S for the realisation of full and immediate open access. PLoS Med. 15:e1002663. doi: 10.1371/journal.pmed.1002663

26. Stahel, P. F., and Moore, E. E. (2014). Peer review for biomedical publications: we can improve the system. BMC Med. 12:179. doi: 10.1186/s12916-014- 0179-1

27. Ducrée, J., Etzrodt, M., Bartling, S., Walshe, R., Harrington, T., Wittek, N., Posth, S., Wittek, K., Ionita, A., Prinz, W., Kogias, D., Paixão, T., Peterfi, I. and Lawton, J., 2021. Unchaining Collective Intelligence for Science, Research, and Technology Development by Blockchain-Boosted Community Participation. Frontiers in Blockchain, [online] 4, p.1. Available at: https://www.frontiersin.org/articles/10.3389/fbloc.2021.631648/full

28. I Lawrence and K. Lin. A concordance correlation coefficient to evaluate reproducibility. Biometrics, pages 255–268, 1989

29. F. S. Collins and L. A. Tabak. NIH plans to enhance reproducibility. Nature, 505(7485):612, 2014

30. Challenges in irreproducible research. Nature, Special in Vol. 541, 2017. http://www.nature.com/ news/reproducibility-1.17552

31. ,,

32. Osgood, R. (2016). The Future of Democracy: Blockchain Voting. Technical report. Available online at: http://www.cs.tufts.edu/comp/116/archive/fall2016/ rosgood.pdf

33. Science Europe. Science Europe Principles on Open Access to Research Publications. Apr 2013 (updated May 2015). Available from: http://scieur.org/opennew. Cited 29 Aug 2018.

34. Merton RK. The Normative Structure of Science. In: Merton RK. The Sociology of Science: Theoretical and Empirical Investigations. Chicago: University of Chicago Press; 1973.

35. Max Planck Society. Berlin Declaration on Open Access to Knowledge in the Sciences and Humanities. 22 Oct 2003. Available from: https://openaccess.mpg.de/Berlin-Declaration. Cited 29 Aug 2018.

36. I Lawrence and K. Lin. A concordance correlation coefficient to evaluate reproducibility. Biometrics, pages 255–268, 1989

37. Group, O., 2021. The Open Definition - Open Definition - Defining Open in Open Data, Open Content and Open Knowledge. [online] Opendefinition.org. Available at: <http://opendefinition.org/> [Accessed 24 November 2021].

38. Tapscott, D., & Tapscott, A. (2016). Blockchain revolution: How the technology behind Bitcoin is changing money, business, and the world. New York: Penguin.

39. Avital, Michel (2018) "Peer Review: Toward a Blockchain-enabled Market-based Ecosystem," Communications of the Association for Information Systems: Vol. 42 , Article 28. DOI: 10.17705/1CAIS.04228 Available at: http://aisel.aisnet.org/cais/vol42/iss1/28

40. E. Ferreira Jesus, R. L. Vanessa, C. VN de Albuquerque, and A. A. D. A. Rocha, “A survey of how to use blockchain to secure internet of things and the stalker attack,” Security and Communication Networks, vol. 34, 2018

41. F. Irresberger, Coin Concentration of Proof-of-Stake Blockchains, Leeds University Business School Working Paper, London, UK, 2018

42. Tokens-economy.gitbook.io. 2022. *Proof of Stake (PoS) - consensus*. [online] Available at: <https://tokens-economy.gitbook.io/consensus/chain-based-proof-of-stake/proof-of-stake-pos#:~:text=Coin%20age,set%20by%20the%20coin's%20APR.> [Accessed 21 March 2022].

43. 2022. [online] Available at: <https://www.gemini.com/cryptopedia/blockchain-governance-mechanisms> [Accessed 21 March 2022].

44. Reiff, N. and Rasure, E., 2022. *Decentralized Autonomous Organization (DAO)*. [online] Investopedia. Available at: <https://www.investopedia.com/tech/what-dao/> [Accessed 21 March 2022].

45. Walker, A. J. (September 1977). "An Efficient Method for Generating Discrete Random Variables with General Distributions". *ACM Transactions on Mathematical Software*. **3** (3): 253–256. doi:10.1145/355744.355749

46. Walker, A. J. (September 1977). "An Efficient Method for Generating Discrete Random Variables with General Distributions". *ACM Transactions on Mathematical Software*. **3** (3): 253–256. doi:10.1145/355744.355749

47. Khan, I. and Shahaab, A., 2021. A Peer-To-Peer Publication Model on Blockchain. *Frontiers in Blockchain,* 4.

48. Zeeshan, J., Third, A., Bachler, M., and Domingue, J. (2018). “Peer-reviews on the blockchain,” in 1st Workshop on Reframing Research (RefResh), Cologne, Germany, December 5–7, 2018

49. Bigi, G., Bracciali, A., Meacci, G., and Tuosto, E. (2015). “Validation of decentralised smart contracts through game theory and formal methods,” in Programming Languages with Applications to Biology and Security, Lecture Notes in Computer Science, Vol. 9465,

chapter 11, eds C. Bodei, G. Ferrari, and C. Priami (Cham: Springer), 142–161. doi: 10.1007/978-3-319-25527-9_11

50. Atzei, N., Bartoletti, M., and Cimoli, T. (2017). "A survey of attacks on ethereum smart contracts (SoK)," in Proceedings of the 6th International Conference on Principles of Security and Trust, Lecture Notes in Computer Science, Vol. 10204, eds M. Maffei and M. Ryan (Uppsala: Springer New York), 164–186. doi: 10.1007/978-3-662-54455-6_8

51. Dannen, C. (2017). Introducing Ethereum and Solidity: Foundations of Cryptocurrency and Blockchain Programming for Beginners, 1st Edn. Berkeley, CA: Apress. doi: 10.1007/978-1-4842-2535-6

52. Luu, L., Chu, D.-H., Olickel, H., Saxena, P., and Hobor, A. (2016). "Making smart contracts smarter," in Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security (CCS '16) (Vienna: ACM), 254–269. doi: 10.1145/2976749.2978309

53. Deshpande, A., Stewart, K., Lepetit, L., and Gunashekar, S. (2017). Distributed Ledger Technologies/Blockchain: Challenges, Opportunities and the Prospects for Standards. Technical report, The British Standards Institution (BSI). Available online at: https://bit.ly/32QxvKp

54. Ponsford, M. (2015). A comparative analysis of bitcoin and other decentralised virtual currencies: legal regulation in the People's Republic of China, Canada, and the United States. Hong Kong J. Legal Stud. 9, 29–50. doi: 10.2139/ssrn.2554186

55. Savelyev, A. (2017). Contract law 2.0: 'Smart' contracts as the beginning of the end of classic contract law. Informat. Commun. Technol. Law 26, 116–134. doi: 10.1080/13600834.2017.1301036