Lakat
An open pluralistic base layer for academic publishing

Leonhard Horstmeyer
(Do not distribute)

June 14, 2023

Abstract

In this paper, we present three contributions to the field of academic publishing. Firstly, we introduce Lakat, a novel base layer for a publishing system that fosters collaboration, pluralism and permissionless participation. Drawing inspiration from the philosophy of Imre Lakatos, Lakat is designed as a peer-to-peer process- and conflict-oriented system that supports continuous integration across multiple branches. This architecture provides a robust foundation for the integration of existing reputation systems and incentive structures or the development of new ones. Secondly, we propose a new consensus mechanism, called Proof of Review, which ensures the integrity and quality of the content while promoting active participation from the community. Lastly, we present Lignification, a new finality gadget specifically designed for branched, permissionless systems. Lignification provides a deterministic way to find the consensual state in these systems, ensuring the system's robustness and reliability in handling complex scenarios where multiple contributors may be proposing changes simultaneously. Together, these contributions aim to provide a convenient starting point to tackle some of the issues in traditional paper-formatted publishing of research output. By prioritizing collaboration, process-orientation, and pluralism, Lakat aims to improve the way research is conducted and disseminated and ultimately hopes to contribute to a healthier and more productive academic culture.

Contents

1 Introduction

With the vast amount of data structures, of query and storage systems, of versioning and networking tools and of large language models, one may engineer publishing systems by posing certain requirements that give rise to a different and arguably more collaborative, efficient and healthy academic culture. This approach can be contrasted with an incremental adjustment of the existing system, which in many quantitative sciences is called the greedy approach. We propose an architecture leveraging the available technology that we call *Lakat*. Lakat is a distributed database with a local peer-review concensus layer. The system serves as a permissionless continuous integration solution for collaborative research. One may conceptually think of Lakat as a peer-to-peer version of Wikipedia with a branch structure similar to git and a peer review system. Our starting point is a set of eight core requirements, that we posit for a publishing system:

1. **Open** – Content and code base[1] should be accessible freely[2].
2. **Permissionless** – No one should be barred from contributing.
3. **Pluralistic** – No monopoly on research opinion.[3]
4. **Process-oriented** – Emphasizing the process rather than an outcome.
5. **Conflict-oriented** – Making conflicts a feature rather than a bug.
6. **Curatable** – Making the organization of the content part of the output.
7. **Sustainable** – Data and compute resources should be low and reuse of fragments encouraged.
8. **AI friendly** – Allowing all kind of entities to contribute, indivduals, groups or AI agents.

[1] Here we refer to the code base of any client implementation.
[2] Internet service providers are not free. So we refer here to additional charges.
[3] This is not not necessarily the same as "No single source of truth".

The research paper, as the gold standard of publicizing research output, poses several threats to the overall scientific endeavor. It is a relict from the times where the printing press had been the latest innovation and where the channels for communicating had a large latency. We mention six issues associated with the paper-formatted research output that are addressed by Lakat:

- It incentivizes the creators of scientific output to withhold preliminary results or results that are either not significant or at odds with a hypothesis. Even if there are significant results[4], they may not meet the eye or mind of other creators or consumers of scientific output until the entire paper has been published. It may then even take on the order of tens of months for the paper-formatted research output to be accessible, which is particularly problematic for impactful research. Thus the process of building on top of previous work and of critical engagement is hindered and in the best case deferred.

- It incentivizes creators to wrap minor changes into the guise of an entire research paper, reusing a possibly templated introduction over and over again.

- The output is but a polished snapshot of a process, an inorganic blob "data structure". The process of reaching a result or of not reaching it as well as the review process are generally not part of the output and not naturally representable in the rigid paper-format. The process often doesn't stop with the paper-publication, but continues thereafter and it requires awkward hacks in the form of addenda, corrections or new paper-formatted versions to account for changes.

- It creates rigid and isolated islands of content, disregarding potentially conflicting or agreeing intersections. Papers address these intersections with citations that are often placed in an unspecific context, and tend to reference an entire paper or body of work rather than a particular part. These intersections between different scientific outputs are not only constrained to citations, but entire paragraphs such as introduction or method sections are often simply replicated from previous papers. Thus, making conflicting or agreeing intersections a manifest part of the data structure can overcome the hacky fixes and shortcommings of the paper-format.

- The question of who contributed how much to a research output often causes conflict among researchers. A process-oriented publication system facilitates the tracking of contributions and may reduce the cases of unjust allocation of contributorship. In paper-formatted publications the contributorship is proxied by a negotiated ordered list of co-authors, which cannot capture contributions and inevitably leads to unjust allocations.

- The effective barring of potential contributors in paper-formatted research does not increase the level of scrutiny, creativity, or quality of the output. On the contrary, maybe another set of eyes can add insights or expand on the results. Why should the self-declared co-authors be in the best position to conduct the research? The fear for the theft of ideas is mostly inherent to bulk-publications and less to process-based research output.

Apart from the abovementioned problems with paper-formatted research, Lakat may also be instrumental for solving other problems with scientific publishing such as the exploitation of scientists regarding their review services and production of output. Even though Lakat does not directly address this, it does provide a base layer upon which a system of incentives can be built.

1.1 Related Work

Various solutions have been proposed to improve the process of science publishing with respect to transparency, review, ownership, decentralization, collaboration, openness, and fairness. We exhibit proposed solutions and their benefits or shortcomings. Since Lakat sits at the intersection of branchable version control (c.f. git[**?**, **?**]), large collaborative enyclopedias (c.f. wikipedia[**?**]) and peer-to-peer (c.f. human society [**?**]) protocols (c.f. urbit [**?**] or file sharing protocols[**?**]), we will focus on solutions in that general triangle.

The platform Scholarpedia was launched in 2006 [**?**]. It is a wiki-based format with a peer review layer, where institutional affiliation is required for contributon. It is thus integrating a scholarly component into wikipedia. The requirement of affiliation is also one of the drawbacks of this solution, as it bars some potential contributors. Furthermore, the authors of an article are either chosen or elected. This to our mind has two further problems.

[4]These may be perceived as significant or later recognized as significant by the community

First, it raises the question who elects those that elect. Second, the collaborative dimension of wikipedia is lost. In contrast Lakat – like wikipedia – retains the permissionless so that no one is barred from editing or from proposing pull requests to change content (see Section **??** for details). In 2007 the Citizendium fork of the English wikipedia launched [**?**] with the objective to add a quality assurance layer on top of wikipedia. The concept of approved aricles played an important role. However, who approves the articles. What happens to subsequent changes? Would they have to be approved again or does the approval yield a sort of finality for the manuscript? Another wiki-formatted solution is the Manubot platform [**?**], which allows for the collaborative preparation of research articles that can then be sent to peer-reviewed journals. However, Manubot is not a publication platform itself but aids the collaborative process of reaching a traditional publication.

There are also many attempts to put part of the existing publishing logic onto a cryptographically secured distributed ledger. Everipedia[**?**] was a fork of wikipedia. They have also tried to build a quality assurance system on top of it using reputation tokens that can be staked and potentially lost in the process of edits, thus leveraging distributed ledger technology. So instead of tokenizing ownership of edits, they tokenized reputation. Those tokens were deployed on a blockchain (EOS and later Polygon). The project has been archived. Orvium [**?**] on the other hand aims to put submission of manuscripts, revisions and publications onto a blockchain or at least have them stored using some decentralized storage provider. Unfortunately it is not evident who stores what, how and where. There is for instance not much information about whether they are creating a dedicated blockchain or use an existing one. The Scienceroot project [**?**] was launched in 2018 with the intention to create an on-chain economy around the publishing system using a reward token called Science Token (ST), which is deployed on the Waves blockchain. They also created or attempted to create an academic journal that ties into their economy. Pluto[**?**] is a blockchain-based platform for academic publishing that supports peer review, open access and micropayments. ARTiFACTS[**?**] is a project that aims to create a blockchain-based platform for scholarly research that enables researchers to create a permanent, time-stamped record of their various items that support their research such as data sets, images, figures etc. PubChain[**?**] is a project that aims to create a decentralized open-access publication platform that combines a funding platform with decentralized publishing. Like Scienceroot, it has its own token coincidentally also called Science Token (ST), which is used to exchange funds, store articles on ipfs and store their content identifiers on the blockchain. They also plan to integrate crowdfunding through their marketplace. TimedChain [**?**] is a project that aims to create a blockchain-based editorial management system that organizes manuscripts by publishers, authors, readers and other third parties. EUREKA [**?**] is a project that aimed to create a blockchain-based peer-to-peer scientific data publishing platform with peer review, open access and micropayments. It was developed by the team behind ScienceMatters, an existing open access publisher that conducts triple-blind peer review. EUREKA also aimed to provide a blockchain-based rating and review system that allows readers to evaluate the impact of published articles. It is, however, not any more maintained. The Open Science company Desci Labs is developing a project called Desci Nodes [**?**]. Similar to Scienceroot, DeSci Nodes is a tool for creating research objects, which are a type of verifiable scientific publication that combines manuscripts, code, data, and more into a coherent unit of knowledge. The 2018 "nature index" article [**?**] entitled "Could Blockchain Unblock Science?" focusses on the question of how blockchain could be used to improve the process of current science publishing. Brock also mentions that data edits could be made permanently visible, which aludes to the idea of securing continuous editing in an immutable and consensual manner. He also developed and deployed the Frankl, which is an open source blockchain-based publishing platform [**?**]. Further insights into the landscape of blockchain-based solutions for the scientific publishing are provided in [**?**]. Apart from providing an overview of the landscapt until 2019, they proposes a governance framework for scientific publishing based on a consortium blockchain model. Some of those solutions aim to make the reward structure more open and introduce on-chain reward systems [**?**].

When developing solutions for academic publishing, blockchain technology seems appealing because it yields effectively immutable, globally agreed data in an open and transparent way without the need for a single source of trust. However, one must not fall into the fallacy of searching for nails for a hammer. At the heart of the blockchain paradigm lies the idea of a consensus about a global unique truth. This is a very useful technology for fiat (e.g. printed money or cryptocurrency), which exists through a global consensus. However, research output is not a fiat currency. It is subject to conflicting theories, opposing views and possibly irreconcilable results. All of those drive the continuous process that is science. One may build solutions on top of a blockchain to allow for potentially conflictual editing, but this is not what it was designed for. Instead we suggest to make Lakat a base layer that satisfies the requirements for a publication system by design. A comprehensive study on the development of decentralized consensus mechanisms in blockchain networks, such as the work by Wang et al. [**?**], which provides an in-depth review of the state-of-the-art consensus protocols from both the perspective of distributed consensus system design and incentive mechanism design.

There are also some solutions that attempt to decentralize version control systems or anchor them in a blockchain. One of the most prominent examples of a decentralized version of a version control system with branches is git-ssb, a decentralized version control system based on the secure scuttlebutt protocol that allows for distributed version control without a central authority [**?**]. The Radicle protocol is another example, which is a peer-to-peer network for code collaboration that extends git with a networking protocol called Radicle Link [**?**]. The project is governed through the RAD token, which is deployed on the Ethereum blockchain [**?**]. Another project that explores ways to decentralize the storage of versioned data is Ceramic, a decentralized network for managing mutable information based on the idea of streams, which are append-only logs of JSON objects. The streams are anchored in a blockchain, which is used as a global ordering mechanism, and stored in a decentralized storage network [**?**]."

With the onset of large language models (LLMs) and AI agents that are capable of statistically extrapolating from a vast set of existing resources we are entering an era where some portions of the scientific research process can and should be outsourced to those models. AI agents should be able to take part in the process of scientific discovery. The impact and power of AI-aided or AI-generated research can be seen in multiple ways. For instance in the field of health and drug research, AI has helped to improve the accuracy and efficientcy of imaging [**?**], the interpretation of large datasets [**?**, **?**, **?**] or the discovery of drugs[**?**]. Moreover, the emergence of large language models has led to the development of autonomous scientific research capabilities, where these models can generate new hypotheses, design experiments to test these hypotheses, and interpret the results to draw conclusions, thereby playing a significant role in the scientific discovery process [**?**].

1.2 Imre Lakatos

The entire architecture of Lakat is heavily inspired by concepts developed by the Hungarian philosopher, Imre Lakatos. In an attempt to contribute to the demarcation problem [**?**, **?**, **?**, **?**] that was prominent in the field of philosophy of science during Lakatos' times, he developed the concept of a *research programme* [**?**, **?**, **?**], also called *Lakatosian research programme*, to avoid confusion with the colloquial use of the former term. The demarcation problem asks about the criteria that distinguish science from 'pseudo-science'. Lakatos develops his theory on the grounds of a process-oriented account of science. So rather than saying that this or that monolitic bulk of work or set of statements is or is not scientific, he posits that this distinction can only be made on the grounds of processes of theoretical amendments to an existing corpus of statements. He distinguishes between progressive and degenerative amendments depending on whether they strengthen the programme's predictive power. For Lakatos a research programme consists of a *hard core*, which is a set of constituting assumptions, axioms as it were, that capture the essence of a research endeavour and a *protective belt* of auxiliary hypotheses. The key ideas that the Lakat-architecture takes from the concept of the Lakatosian research programmes are threefold: 1) The pluralism of various research undertakings. 2) The process-orientation 3) The distinction between a core and a protective belt. At the heart of these foundational concepts lies the idea that science lives through arguments, differences and discourse. The input of Lakatosian concepts into Lakat can then be described as follows: A research programme corresponds to a branch or a set of branches to which researchers contribute changes or ammendments. There is no single master branch, but rather every research programme has its own branch or set of branches. Conflicts with other branches or even within the same branch are an important aspect of Lakat and can be the source of progress (c.f. progressive ammendments in Lakatosian research programmes). A programme can maintain a set of feature branches that support the core branch. These side branches behave like a protective belt.

1.3 Overview

With Lakat, we propose a manifestly pluralistic, process-oriented, and conflict-oriented architecture for the continuous integration of publications, with a primary use case of research publications. In this way Lakat becomes a living document. At its core, the architecture consists of a linked data structure that resembles a DAG, where the key objects are branches. This data structure facilitates collaborative work in much the same way as git does. Branches may be thought of as the analogue of a journal in traditional publishing. The role of journal editors is covered largely by branch contributors. Branches are chains of blocks that contain submissions. The addition of another block happens via a proof of peer review, where the peers are the contributors to that branch. In that sense branches resemble blockchains with blocks consisting of submitted changes instead of transactions. As a consensus mechanism we discuss a solution that combines a proof-of-review at branch-level, a local (i.e. involving just branch-contributors) consensus rather than a global one, with a new finality gadget called Lignification. The review process is open. In a first version of Lakat the identities of the reviewers and the creators of the reviewed

content are disclosed, however we wish to migrate to a weak[5] form of a double-blind protocol leveraging zero-knowlege proofs, where each party may reveal their identity. Data are content-addressable and conform to the ipld multihash format. Storage is handled by a networking component in Lakat, which delegates the bulk of data storage to a selection of other storage providers, including decentralized storage networks such as ipfs, storj, and others. This improves resiliance and longevity.

In the following, we discuss the individual elements of the proposed system and highlight their interaction. We start with the data structure in Section 2, which is the core of the system. There we introduce the objects of a data bucket, a branch, a submit as well as storage related aspects such as the data-trie and the database. We also discuss non-persisted parts of the data structure, namely the branch requests. In Section **??** we discuss the participants of the system and in particular the concept of a branch contributor. Then in Section **??** we discuss the protocol that handles the broadcasting, the consensus mechanism via a proof of review, a new finality gadget called Lignification and also how branches can be created, modified or operated on in this protocol.

2 Data Structure

2.1 Bucket

The most elementary data object is the *bucket*. Each and every submitted item is submitted in a bucket: datasets, paragraphs, images and formulae are contained in buckets. These are examples of *atomic buckets*, expressing the fact that they are the building blocks of the system. Instead of a folder structure, we solve the containment relation through designated buckets that we call *molecular buckets* (like *tree* nodes in git). The data part of those buckets contains merely an arrangement of atomic buckets. One may think of them as the analogue of an article, a book or some other curated content.

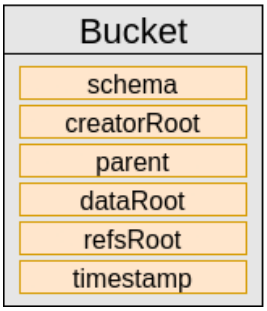

Figure 1: The most elementary type of data container is the bucket. It contains only immutable entries (orange), such as the *schema*, the *creatorRoot*, the *parent*, the *dataRoot*, the *refsRoot* and the *timestamp*.

Every bucket contains six entries: A *schema*, a *creatorRoot*, a *parent*, a *dataRoot*, a *refsRoot* and a *timestamp*. See Figure **??** for an illustration. Here and henceforth the word root refers to the root of a Merkle tree. We go through the entries in turn. The *schema* contains details about the format of the data. For instance we have already mentioned that the data in the molecular buckets is formatted as an arrangement[6]. The *creatorRoot* points to information about the creator of this bucket. Identity on *Lakat* is solved through proofs (see Section **??**). The *parent* is the *content identifier* of the parent bucket. For genesis buckets that would be 0. The *dataRoot* is a content identifier of the data contained in the bucket. In future versions the schema could be absorbed into the dataRoot using the ipld multihash format. This would require a Lakat-specific codec. The *refsRoot* points to all references made to other buckets within the data. This is necessary, since references to other buckets might be obscured inside the data-encoding. This is an analogue of a list of citations. The *timestamp* records the time of inclusion of the bucket into the branch. It is important to note that we use Ethereum [**?**] and some Layer2 block hashes as time stamps in our first version, since the local consensus is too weak to ensure that all participants are truthful to the time otherwise. Anticipating block hash is close to impossible. One cannot change the data inside the bucket. One would have to create a new bucket that points to the original bucket via its parent entry.

[5] Weak is to be understood in the sense that both parties may choose to reveal their identity.

[6] The purposefully vague formulation of an 'arrangement' is due to the intention to keep that format flexible. One may think of this as an ordered list, but one might also consider further directives or clustering of content in a directed hypergraph.