2017 IEEE/ACM 12th International Workshop on Software Engineering for Science (SE4Science)

Advancing Open Science with Version Control and Blockchains

Jonathan Bell, Thomas D. LaToza, Foteini Baldmitsi and Angelos Stavrou
Department of Computer Science
George Mason University
Fairfax, VA 22030
Email: {bellj, tlatoza, foteini, astavrou}@gmu.edu

***Abstract*—The scientific community is facing a crisis of reproducibility: confidence in scientific results is damaged by concerns regarding the integrity of experimental data and the analyses applied to that data. Experimental integrity can be compromised inadvertently when researchers overlook some important component of their experimental procedure, or intentionally by researchers or malicious third-parties who are biased towards ensuring a specific outcome of an experiment. The scientific community has pushed for "open science" to add transparency to the experimental process, asking researchers to publicly register their data sets and experimental procedures. We argue that the software engineering community can leverage its expertise in tracking traceability and provenance of source code and its related artifacts to simplify data management for scientists. Moreover, by leveraging smart contract and blockchain technologies, we believe that it is possible for such a system to guarantee end-to-end integrity of scientific data and results while supporting collaborative research.**

I. Introduction

As reported in a recent *Nature* article, the scientific research community faces a "reproducibility crisis" [7]. 70% of the 1,576 scientists surveyed (from various fields, including chemistry, physics, earth and environmental science, biology and medicine) reported that they had tried and failed to reproduce another scientist's experiments. A 2012 review of 2,047 biomedical and life-science research articles that had been retracted found that 43% of those retractions were due to fraud (or suspected fraud). To maintain confidence in scientific results, we must ensure the integrity of the scientific workflow, from data collection to article publication.

Towards addressing this crisis, there has been a rise of online repositories to share scientific data, protocols or findings. By publishing the data that led to a scientific result, researchers invite the reproduction of their experiments, even in cases where future researchers might have difficulties re-acquiring that same data (i.e., if specialized and expensive machinery were required). However, researchers have been slow to adopt such repositories: they are often seen as a burden, requiring time and effort to decide what data should be released and what form it should be. Moreover, some researchers may be wary to make large data sets available before they believe that they have achieved the maximum benefit (in terms of additional publications) from that data.

Even if researchers *do* choose to make their data sets public, this still does not solve the problem of data *integrity*. Who is to say that the data reported is unmodified? Or, in the event that it is advertised as modified, that it is in fact modified in the exact way described? If a researcher claims that they are releasing an entire data set, are they? These challenges can arise even if we assume that no researchers would purposely make a false claim, as managing large ecosystems of data repositories is complex, and data sets are vulnerable to accidental corruption.

Historically, lab science data has been maintained in lab notebooks, where researchers might sign and date each page to attest to the authenticity of each step and collected data. But as data collection and analysis has become increasingly electronic, this record has begun to vanish. At the same time, provenance has itself grown complex, as research collaborations may be geographically distributed involving data from multiple researchers and analyses may make use of historical datasets. Moreover, the increasingly sophisticated scripts and tools for data analysis, which may themselves be shared or not shared, makes it important to also maintain traceability links to the exact code used to analyze primary data, particularly if scripts are later improved or found to have defects.

Ideally, a data management system should:

- Support private collaboration — enabling researchers at the same lab or across institutions to collaborate in generating and analyzing unpublished private data before publication
- Support data collection and analysis workflows — ensuring that the system is well integrated into all steps of the data generation and analysis process and that its complexities do not become a barrier to adoption
- Support integration with the diverse data repositories already in use, as well as with the diverse authentication and authorization systems used by academic and research institutions
- Provide guarantees of the integrity of the data and results — maintaining immutable traceability links between the the primary data collected, scripts used to analyze data, and the results of analyses
- Allow, but not require, full disclosure of data sets
- Allow the general public to easily discover data sets that have been publicly released
- Be open and decentralized, having no single point of failure or trust

978-1-5386-2791-4/17 $31.00 © 2017 IEEE
DOI 10.1109/SE4Science.2017.11

Current scientific data management tends to focus on dissemination of final results, with many repositories and approaches focusing on where those archive should be stored and indexed [1], [2]. What is lacking is a system that researchers can use in their lab, as they perform their research, to maintain immutable traceability links between the original data as captured, the transformations and analyses applied to it, and the results presented for publication.

We argue that by leveraging insights from software engineering and cryptography, we can provide the scientific research community with a data management infrastructure that meets all of these goals. [6]

II. Discussion

Software engineers have long wrestled with problems of traceability and version control of *code*, which are directly applicable to scientist's data management concerns. In principle, off-the-shelf version control could solve many traceability problems: if researchers used, for instance, git to store their data sets and analysis tools, then they could easily maintain a version history. Upon release, an outside party could see the transformations made to a data set as long as it is tracked by the version control system. Git is a decentralized version control system: there need not be a single "main" server that stores repository information, and hence, also can satisfy our requirement of having no single point of failure.

However, such a system might not necessarily be immediately adapted to environments where data is stored in binary forms (and hence, opaque to traditional change tracking tools). Moreover, software version control systems such as git are not designed to preserve an immutable audit trail: a malicious actor could easily tamper with the contents of a repository, changing the historical record. For example, git explicitly supports rewriting history through actions such as reordering commits, changing messages, or removing commits entirely, as commits are designed to provide a simplified and idealized record of code history rather than an immutable audit trail.

One solution could be to trust a recognized third party (for instance, the editorial board of a journal or an archival company such as the Open Science Framework [5]) to preserve a copy of the repository state at each stage of research. If we trust the integrity of that third party, then we can have some reasonable assurance in the integrity of the scientific experiments being performed. However, this third party would immediately become a point of failure — should it go offline or be compromised, then the integrity of the entire data ecosystem is compromised. Moreover, this third party could be suspect of tampering with the data that they are storing or vulnerable to security breaches by malicious actors.

We propose that by combining version control systems with blockchains technologies, we can create a fully distributed immutable ledger of scientific experiments. In simple words, one can describe the blockchain as a distributed database (or an append only ledger) that utilizes cryptographic techniques (such as hashing and digital signatures) to achieve the addition of new entries in a secure, linear and chronologically ordered way. The nodes that maintain the blockchain work together so that at all times they reach a single consensus of the most up-to-date version of the blockchain, even when the nodes are run anonymously, have poor connectivity with one another, or have potentially malicious operators. Blockchains are designed in such a way that the cost to rewrite or alter any part recorded on them is prohibitively expensive.

We envision a system where researchers will use a common (potentially public) blockchain to post their data sets and results. Storing actual data on the blockchain is not practical for scalability reasons. Therefore, every post on the blockchain will only contain a pointer to the actual data (which would be stored in a separate version control system), a cryptographic hash of the result/data, a proof of ownership (via a digital signature) and access permissions. Posting a hash of a scientific result on the blockchain will serve as a "cryptographic proof" that the owner of the post possesses the result at the exact time of posting without necessarily revealing the actual result yet. Note also, that the privacy of the data is not in danger given that actual data is never published on the blockchain. Scientists could choose to release their underlying data immediately, or to keep it private indefinitely, in which case it could still be audited at some point in the future.

Depending on the application we can use *permissionless* or *permissioned* blockchains. In permissionless blockchain systems, such as Ethereum [3], everyone can participate and post, while in permissioned systems, such as the Hyperledger Fabric [4], approved parties are given a participation credential that allows them to post on the blockchain The participation credential could be issued by a third party, or collectively, though a voting process, from the already participating parties.

The use of blockchain technologies can open up even more possibilities by the use of *smart contracts*. A smart contract is a software piece, posted in the blockchain, that allows the automated execution of an action if, say a specific record appears on the blockchain or another publicly verifiable event happens. For instance a smart contract could allow the automated "opening" of multiple results, as long as a specific number of parties commit on the blockchain that they have completed their experiments.

There are still many questions unanswered as to how exactly to use blockchains and version control to manage scientific data, and we believe that this is just the beginning of a longer conversation.

Acknowledgements

We thank Huimin Chen for helpful discussions.

References

[1] Data dryad. http://datadryad.org.
[2] Dataverse project. http://dataverse.org.
[3] Ethereum. https://www.ethereum.org/.
[4] Hyperledger. https://www.hyperledger.org/.
[5] Open science framework. https://osf.io.
[6] A Goodman et al. 10 simple rules for the care and feeding of scientific data. *CoRR*, 2014.
[7] M. Baker. 1,500 scientists lift the lid on reproducibility. *Nature*, 533:452–454, May 2016.