J Grid Computing (2022) 20:35
https://doi.org/10.1007/s10723-022-09626-x

A Blockchain-Based Architecture for Trust in Collaborative Scientific Experimentation

Raiane Coelho · Regina Braga · José Maria N. David · Victor Stroele · Fernanda Campos · Mário Dantas

Received: 3 March 2022 / Accepted: 19 September 2022 / Published online: 12 October 2022

Abstract In scientific collaboration, data sharing, the exchange of ideas and results are essential to knowledge construction and the development of science. Hence, we must guarantee interoperability, privacy, traceability (reinforcing transparency), and trust. Provenance has been widely recognized for providing a history of the steps taken in scientific experiments. Consequently, we must support traceability, assisting in scientific results' reproducibility. One of the technologies that can enhance trust in collaborative scientific experimentation is blockchain. This work proposes an architecture, named BlockFlow, based on blockchain, provenance, and cloud infrastructure to bring trust and traceability in the execution of collaborative scientific experiments. The proposed architecture is implemented on Hyperledger, and a scenario about the genomic sequencing of the SARS-CoV-2 coronavirus is used to evaluate the architecture, discussing the benefits of providing traceability and trust in collaborative scientific experimentation. Furthermore, the architecture addresses the heterogeneity of shared data, facilitating interpretation by geographically distributed researchers and analysis of such data. Through a blockchain-based architecture that provides support on provenance and blockchain, we can enhance data sharing, traceability, and trust in collaborative scientific experiments.

Keywords Collaborative scientific experiments · Blockchain · Provenance · Reproducibility · Cloud computing · Genomic sequencing · Coronavirus

R. Coelho · R. Braga (✉) · J. M. N. David · V. Stroele · F. Campos · M. Dantas
Computer Science Graduate Program, Federal University of Juiz de Fora, Juiz de Fora, MG, Brazil
e-mail: regina.braga@ufjf.edu.br

R. Coelho
e-mail: raianeqc@gmail.com

J. M. N. David
e-mail: jose.david@ufjf.edu.br

V. Stroele
e-mail: victor.stroele@ice.ufjf.br

F. Campos
e-mail: fernanda.campos@ufjf.edu.br

M. Dantas
e-mail: mario.dantas@ice.ufjf.br

1 Introduction

The advancement of modern science increasingly depends on interaction between scientists and the effective use of collective intelligence. Scientists are being gradually driven to collaborate and share information in distributed scenarios, as well as to reuse data from their peers [2, 6, 13, 35, 61]. Over the last decade, the data-oriented science paradigm has become a widespread reality [32–34]. Therefore, data traceability (provenance) and trust have become critical aspects of data-driven experiments. Managing data-driven science projects is a complex task.

Provenance is defined as the origin or lineage of data that helps understand the scientific experiment results [17]. The importance of provenance in reproducible computational research is well documented in the literature [25, 46, 58]. In collaborative experiments, it is important to use provenance to help researchers analyze quality, verify authorship, and reproduce the results. Provenance data related to a scientific experiment are considered intellectual property [7, 18]. Only authorized people can share or view results before publication. On the other hand, researchers must have confidence in conducting collaborative experiments. Thus, all data updates must be tracked and verified aimed to know how the data was created over time. Moreover, data availability is essential since scientists conduct scientific experiments in distributed and heterogeneous environments.

There are several definitions for trust [39]. According to Kochovski [39], trust is crucial when cyber-physical systems rely on resources and services owned by various entities, such as in scientific experimentation. One of the definitions relates the trust concept to credibility, i.e., the degree to which the data source or the data is seen to be believable, a concept that can be extended to any data item, such as a video frame, an AI model, or an experimentation process.

In the scientific experimentation scenario, there is a need for an environment that allows scientists to share data and interact in the experimentation process that guarantees confidence in and availability of the data (the data is not in only one place and can be in several formats). One way to share data is using a standard provenance model, which will guarantee the capture of the history of all activities related to the scientific experiment, that is, involving the processes and data related to the experiment. However, using a standard provenance model does not guarantee confidence in the data or its availability. Blockchain-based systems [50] can be an alternative to bring trust to collaborative and distributed activities. Blockchain also has the potential to improve interoperability and transparency.

Several studies in the literature (see Section 3.1) emphasize the importance and the need to support provenance in scientific research, considering trust and availability. These requirements point to the potential applicability of blockchain as a facilitator for the experiments when executed on scientific platforms [14, 19, 37, 63], bringing trust to provenance data.

We argue that using provenance and blockchain can aid in data traceability, availability, and trust in collaborative scientific experiments. Existing works partially provide solutions in this regard, emphasizing trust, provenance, and sharing, but not altogether. However, none of them support the capture of provenance from SWMS directly and store it in a blockchain solution.

This work presents an architecture based on blockchain and provenance standards, called BlockFlow, to support trust, sharing and traceability of data and processes in collaborative research. The aim is to allow scientists to work in a distributed environment, reliably sharing provenance data to support the reproducibility of the results. Through examples and application scenarios, we discuss the proposal's feasibility in supporting collaborative scientific experimentation traceability and trust, integrating data from different Scientific Workflow Management Systems (SWMS) in an infrastructure supported by a blockchain network.

We investigated the following Research Question (RQ): How can the BlockFlow architecture assist scientists in collaborative scientific experiments, offering an environment that supports data sharing, traceability, and trust? From this RQ we derived Secondary Research Questions (SRQ), i.e., (SRQ1) Can BlockFlow provide an overview of provenance data transparently, where geographically distributed researchers can verify how the provenance data was created in the blockchain over time? (SRQ2) Can BlockFlow be used as a collaborative and trustable scientific environment supporting the interoperability of provenance data coming from heterogeneous SWMSs? (SRQ3) Can BlockFlow be used as a trustable provenance data exchange environment in data-intensive workflows? (SRQ4) Can BlockFlow be used as an environment that provides privacy to provenance data, where data is shared only between authorized partners? (SRQ5) Can BlockFlow be used as an environment to support reproducibility?

Considering the research questions, we verified the feasibility of the proposal based on the Design Science Research (DSR) methodology [30, 31]. An evaluation was conducted using the genomic sequencing of the new SARS-CoV-2 coronavirus. The results pointed to the feasibility of the architecture.

The main objective of this work is to create a blockchain-based architecture to support trust, traceability, and sharing in collaborative research, guaranteeing privacy, interoperability, and the reproducibility of the results.

The specific contributions of this work are:

- The specification of collaborative environments and blockchain networks anchored by a cloud infrastructure for executing data-intensive workflows.
- The specification and implementation of a provenance collector that uses web services technology to capture provenance.
- A wrapper that translates and integrates heterogeneous provenance data from different Scientific Workflow Management Systems (SWMSs) into the ProvONE model.
- Immutable storage and management for querying, analyzing, and viewing provenance data from collaborative scientific experiments.
- An API to connect BlockFlow with other applications and platforms to allow the creation of blockchain networks to support scientific experiments.
- A GUI-based application allows researchers to implement blockchain networks easily and then collaborate.
- A systematic mapping of the literature, which identified and categorized related works that use blockchain as a mechanism to bring trust to provenance data.

This paper is organized into five sections, including this one. Section 2 presents the main concepts related to the proposed solution. Section 3 discusses the proposed solution using the DSR methodology, and Section 4 presents the results. Section 5 discusses the final considerations and contributions of the work, its limitations, and future works.

2 Background

Hey et al. [32] defined e-Science as a global collaboration of key areas of science together with the generation of a computational infrastructure capable of supporting it. Scientific Workflows (SWs) have become a standard to perform e-Science experiments. Scientific Workflow Management Systems (SWMs) are used for SW modeling and execution. There are several SWMs with distinct characteristics and behaviors, such as VisTrails [9], Taverna [45], Swift/T [67], Kepler [43], Pegasus [20], Chiron [44] and Galaxy [24], among others. E-Science presupposes the construction of a computational infrastructure for distributed use, allowing collaboration between scientists, and involving the intensive use and sharing of data, which are often heterogeneous [32]. The importance of collaboration and data sharing among researchers is based on reproducibility. It is common to create collaborative networks between groups of geographically distributed researchers [8].

As multiple parties are involved in these collaborative networks the management of trust and transparency in information sharing among researchers is challenging. For exchanging information in collaborative scientific environments, blockchain-based systems [37, 63] can be an alternative. Blockchain can be defined as an immutable, decentralized, and shared book that maintains a sequence of chronological blocks, encrypted, and connected, over a peer-to-peer (P2P) network [22, 68]. The blocks form a chain, a linear sequence that enables the auditing and traceability of information. The blockchain records data inputs in a decentralized way and allows entities to interact with each other without a trusted third party. This communication is done through transactions and reflects the semantics of the application, and can be any information, i.e., currency, scientific data, or others.

Blockchain networks can be classified as permissionless, where anyone can join, make transactions, or leave the network; or permissioned, i.e., a network controlled by a group of known nodes with a central authority that decides and assigns the right to peers to write/read operations. One of the technologies for blockchain implementation is the Hyperledger Fabric[1] [3]. In this technology, network access is restricted to authorized people, characterizing it as permissioned [64, 65].

In Hyperledger Fabric version 1.4.1, RAFT is used as the ordering service. RAFT protocol is crash fault tolerance (CFT), not Byzantine fault tolerance (BFT). The Hyperledger Fabric network consists of a set of geographically distributed peers (nodes) running in

[1] https://www.hyperledger.org/use/fabric.

docker containers[2]. Each node maintains the ledger state and transaction log through Apache CouchDB[3] or LevelDB[4]. The transactions are controlled and generated through smart contracts known as chaincodes. Creating an isolation mechanism known as a channel is necessary to maintain privacy and confidentiality and isolate activities between authorized parties. Participating nodes need to register and have identities to perform transactions. Identity records are provided by a Certificate Authority (CA), which also issues certificates with signing transactions. Along with the CA, another important component for identification is the Membership Service Provider (MSP), responsible for mapping certificates between nodes.

In the scientific experimentation process, it is necessary to have information about the data transformations from their origin to the results generated. This type of information is known as provenance data [26]. Provenance helps scientists understand the experiment, interpret, explain results, and diagnose problems throughout the scientific process. There are different provenance types. Lim et al. [42] and Koop and Freire [40] classify provenance, especially considering scientific workflows, into three types, prospective[5], retrospective[6], and evolutionary[7] provenance. In collaborative experiments, the provenance capture must be independent of a SWMS, allowing interoperability between distributed scientific workflows captured from different SWMS. Several community efforts culminated in the development of generic models to represent provenance and promote interoperability, including the OPM (Open Provenance Model) [49] and the PROV [27, 46]. ProvONE is a provenance model that extends the PROV [10] and was created specifically for scientific workflows. ProvONE enables interoperability by integrating heterogeneous information from multiple workflows produced by different SWMS into a standard format. Furthermore, it represents prospective, retrospective, and evolutionary provenance.

3 Methods and Materials

Blockflow was developed considering the DSR methodology [30], in two cycles. We constructed the BlockFlow architecture in the first cycle and carried out a Proof of Concept (PoC) [15]. In the second cycle, we improved the BlockFlow architecture and performed a Case Study (CS) in genomic sequencing of the new SARS-CoV-2 coronavirus.

We followed some steps in DSR conduction [31], namely problem definition, literature review and discussion on existing solutions, artifact development, evaluation, and discussion of results. Considering these steps, Section 3.1. discusses related works based on a Systematic Mapping result. Section 3.2 details the BlockFlow architecture. Section 3.3 presents the evaluation steps. The Results and Discussions are presented in Section 4.

3.1 Systematic Literature Mapping

We conducted a Systematic Literature Mapping to identify research investigating the topics of blockchain technology related to provenance. It is not the purpose of this paper to detail the systematic mapping but rather to present its main results. The search string was specified as follows: (“blockchain”) AND (“provenance” OR “data provenance”) AND (“approach” OR “architecture” OR “framework” OR “infrastructure” OR “method” OR “model” OR “solution” OR “technique” OR “platform” OR “tool” OR “process” OR “software”).

The selected control papers were: (i) ProvChain: A Blockchain-Based Data Provenance Architecture in a Cloud Environment with Enhanced Privacy and Availability [41]; (II) SmartProvenance: A Distributed, Blockchain-Based Data Provenance System [55]; (III) Blockchain-Based Provenance Sharing of Scientific Workflows [12] ; III) Business process engineering for data storing and processing in a collaborative distributed environment based on provenance metadata, smart contracts and blockchain technology [21]. Considering the search string, the total number

[2] https://www.docker.com/.
[3] https://couchdb.apache.org/.
[4] https://dbdb.io/db/leveldb.
[5] captures the structure and static context of a workflow, i.e., it expresses the steps to be followed to generate a dataset. It is a specification of the computational tasks that will be performed in the experiment.
[6] it is associated with information about the execution of a workflow, i.e., information about the activities performed - steps taken to derive a dataset. More specifically, it is a detailed log of the execution of each task in the workflow.
[7] reflects changes made between two executed versions of the workflow, i.e., the evolution history, keeping all changes applied throughout its lifecycle.

of articles returned, from 2008 to 2022 according to each library, were: (i) ACM Digital Library (dl.acom.org) 194, (ii) EI Compendex (engineeringvillage.com) 167, (iii) IEEExplore (ieeexplore.ieee.org) 94, (iv) Scopus (scopus.com) 189, (v) Springer (springer.com) 140, (vi) Web of Science (app.webfknowledge.com) 90. From the inclusion and exclusion criteria (Appendix 1), 61 articles were analyzed and classified to answer the mapping questions (Appendix Table 7).

Next, we detail the main works resulting from the mapping study. Shantharam et al. [56] describe OSC's command line utility that preserves the integrity of research datasets. OSC does not capture the data in collaborative workflow execution as a Blockflow. Besides, it does not use a provenance model. Pajooh et al. [54] detail a distributed data storage of a blockchain-enabled large-scale IoT system. The focus is on system performance. Although they discuss provenance capture issues, they do not use a provenance model and do not capture provenance during workflow execution. Möller et al. [48] introduce a blockchain-based data provenance information system, enabling decentralized information sharing. Although they discuss provenance data and blockchain issues, they do not use a standard provenance model, which is important in a collaborative environment, nor do they detail their blockchain framework. The Vassago system [28] focuses on query processing over multiple blockchains. It does not discuss provenance capture and blockchain storage issues. SciChain [1] proposes a blockchain system for provenance services on HPC. It captures provenance considering a proprietary model. SciChain does not use a standard provenance model as PROV, which hinders its use in a collaborative environment with different SWMSs. PROV-HL [21] is an architecture similar to ours that uses Hyperledger Fabric and manages provenance securely. However, it does not use a provenance standard model, such as PROV, and does not capture provenance from workflow execution.

Song et al. [60] present an integrated solution using blockchain and PROV model. The work is similar to ours. However, Blockflow uses a permissioned blockchain, which is essential for scientific experimentation and has a provenance capture service directly connected to workflow execution in SWMSs. Trac2Chain [66] is a framework that provides linkage privacy as well as full tracking/tracing functions for the provenance graphs stored on blockchain. Blockflow uses a similar approach but emphasizes provenance capture based on a standard model (PROV) and provides a provenance capture mechanism independent of a specific SWMS. The BSCDP Architecture [36] is proposed to provide secure cloud storage. This work does not deal with collaborative workflow execution and the capture of provenance, considering a standard model that allows the integration of provenance data.

Chen et al. [12] proposed ProChain to enable data sharing from scientific workflows among geographically distributed scientists. They do not consider real-time provenance collection and do not use a distributed and high-performance storage infrastructure such as Blockflow. SciBlock [23] provides trustworthy and tamper-proof storage for data from scientific workflows in a collaborative environment. It does not consider the capture and storage of provenance. Furthermore, they do not consider a provenance capture model, which is important for interoperability in a collaborative workflow. Liang et al. [41] proposed ProvChain architecture to ensure the decentralization, integrity, and trustability of provenance data in cloud storage applications. However, data can be accessed and viewed by unauthorized users belonging to the network. This access can hinder the privacy and intellectual property of scientific workflows. In BlockFlow, data is managed between different research partners securely and privately. Tosh et al. [62] proposed BlockCloud a permissionless blockchain-based framework for provenance data on a cloud platform. In BlockFlow, the permissioned blockchain takes advantage of faster protocols to reach consensus. Therefore, blockchain is a more realistic option for collaborative scientific workflows to share provenance data.

Smartprovenance/DataProv [55] is an architecture based on blockchain for the secure and immutable management of provenance data based on access control. However, it does not have a mechanism for querying provenance data, which is essential for collaborative scientific workflow scenarios. Kim et al. [38] proposed a traceability ontology called TOVE, related to a blockchain, to ensure confidence in supply chain provenance. Unlike the TOVE ontology, in BlockFlow, provenance is represented through the ProvONE standard provenance model, which supports the interoperability of provenance data from heterogeneous scientific data and ensures trust across the blockchain.

Costa et al. [16] proposed ProvSearch, which combines distributed workflow management techniques with provenance data management as an extension of the PROV model for scientific workflows. However, information storage has security issues as an authorized user can corrupt or modify the provenance data. In BlockFlow, provenance data is immutably stored in the blockchain environment. Mendes et al. [44] proposed the Polystore approach to represent heterogeneous provenance data generated by different SWMSs. Oliveira et al. [53] proposed integrating data from distributed and heterogeneous workflows. PBase [10] proposed a provenance repository of scientific workflows that implements the ProvONE ontology, allowing storage, analysis, and replication of scientific experiments. SciCumulus [18] proposed a middleware to orchestrate scientific workflows through SWMS in distributed and parallel environments. However, these solutions have a centralized storage system for provenance data. Data can be compromised and lost if the central server is unavailable. At BlockFlow, there is no single point of failure because the provenance data is decentralized and shared among geographically distributed researchers.

Other related works deserve to be discussed. Deelman et al. [19] discuss some challenges to workflow management, including those related to provenance data and the importance of having an integrated provenance model. However, it does not discuss issues related to blockchain and trustability. Azaria et al. [5] discuss the use of blockchain for the secure storage of medical records using the EMR standard. The article does not use data provenance and uses the Ethereum protocol, which is unsuitable for a private network such as a scientific network. Hang et al. [29] propose a similar approach, using blockchain to distribute and store EMR files securely. The difference is the use of a permissioned blockchain. However, the paper does not discuss the use of provenance data, which limits the approach to medical records.

Some valuable projects discuss semantic data, provenance, and blockchain, such as the OntoChain project (https://ontochain.ngi.eu) and its specific subprojects. BlockFlow is an architecture that aims to support the capture and storage of data from collaborative workflows, using a permissioned blockchain network and a standard provenance model. BlockFlow can be integrated to use and provide specific services to the OntoChain ecosystem. OntoChain is a huge project involving important scientific players, and significant services will be available shortly that can support BlockFlow´s functionalities.

Analyzing the works presented above, we can cite seven aspects discussed in most of these papers. Transparency: it is possible to have an overview of the provenance data transparently, verifying how the provenance data was created over time; Trust: it is possible to guarantee the trust and integrity of the provenance data; that is, it is possible to certify and verify whether a data has been manipulated or not; Decentralized and secure solution: the proposed solution ensures that data is shared in such a way that there is no point of failure; Provenance model: the solution proposes the use of provenance, or has a standard provenance model, such as PROV, OPM or ProvONE; Distributed storage: Data is stored in a distributed and scalable way. These aspects are considered in Table 1. We use these aspects to compare these works with our solution.

According to Table 1, many approaches do not use a standard and interoperable provenance model, and others do not support transparency, trust, or data integrity. Thus, from the mapping results, we identified some aspects not covered in the works reported in the literature regarding the capture of provenance in collaborative, distributed, and heterogeneous workflows. We also consider the support for information sharing, trust, traceability, and reproducibility.

3.2 BlockFlow Architecture

BlockFlow architecture was developed considering the following non-functional requirements: (1) Reproducibility: BlockFlow provenance data is collected and stored securely and reliably. This mechanism prevents arbitrary data manipulation, either intentionally or inadvertently; (2) Privacy: in BlockFlow, data is shared only between authorized parties; (3) Transparency: All nodes in the network (scientists connected to a peer in the network) that make up an experiment can verify how the data in the chain (blockchain) was created over time. As a result, all data updates can be tracked between nodes; (4) Interoperability: SWMS stores provenance data in different formats. In BlockFlow, to support the interoperability of provenance data, the ProvONE model is used [10]. BlockFlow was specified based on the layered architectural model and services.

Table 1 Comparison between BlockFlow and related works

Rel. Work	Trust and integrity	Decentralized and security solution	Information privacy	Standard and interoperable provenance model	Distributed and scalable storage
ProChain [12]			√		
SciBlock [23]	√	√	√		
PROV-HL [21]	√	√	√		√
BlockCloud [62]	√	√	√		
Smartprovenance [55]	√	√		√	√
TOVE [38]	√	√	√	√	
ProvSearch [16]				√	
Polystore [44]				√	
Oliveira et al. [53]				√	
PBase [10]				√	
Möller et al. [48]		√			
Trac2Chain [66]			√	√	
Pajooh et al. [54]				√	
BlockFlow	√	√	√	√	√

BlockFlow is part of a Scientific Software Ecosystem (SSECO) [8], named E-SECO [2, 15] developed in an E-Science joint project led by the Federal University of Juiz de Fora, Brazil. E-SECO provides a platform that allows the accomplishment of experiment steps. BlockFlow connects to E-SECO through an API (Fig. 1). BlockFlow provides mechanisms that bring trust and traceability to data and processes in collaborative scientific experiments. Although BlockFlow can be considered part of E-SECO, it can be used by any application that securely needs provenance data capture, storage, and query of collaborative scientific workflows. BlockFlow provides a specific API for this connection. Figure 1 presents BlockFlow's main modules and the connection with E-SECO.

Figure 1a shows an abstract view of the E-SECO platform, comprising a Development Environment, an Integration Module, an API, User Interface, and External Module. The Development Environment aims to support the execution of experiments and the management of code and its versions with the help of GitHub. E-SECO enables the storage of information about the experiment process in a detailed way, including experiment steps, execution conditions, input and output data, iterations, results analysis, and guaranteeing experiment quality. The Integration Module allows the E-SECO platform connection with additional services, including the connection with BlockFlow, as detailed in Fig. 1c. The integration module is directly connected with the External module, which provides a specific mechanism to connect with Scientific Workflow Management Systems (SWMSs), currently Tavena[8] and Kepler[9]. End users interact with the platform using the User Interface, through which they conduct experiments and develop artifacts. The Application Developer Interface provides specific services to support the experiment´s execution. It includes the ontological service that supports storing and processing domain ontologies related to the experimentation. Therefore, OWL (Ontology Web Language) files can be stored, and inference mechanisms can be processed to provide specific information for the experimentation process. The extendable architecture provides a connection point to E-SECO[10] specific services, such as the PRIME service [2].

However, although the E-SECO data repository is decentralized, it does not have a system that provides trust for provenance data storage and sharing. To cope with this problem, E-SECO connects with the

[8] http://www.taverna.org.uk.
[9] https://kepler-project.org.
[10] It is not our aim to detail E-SECO platform. In [2] and [13] a detailed specification of E-SECO is provided.

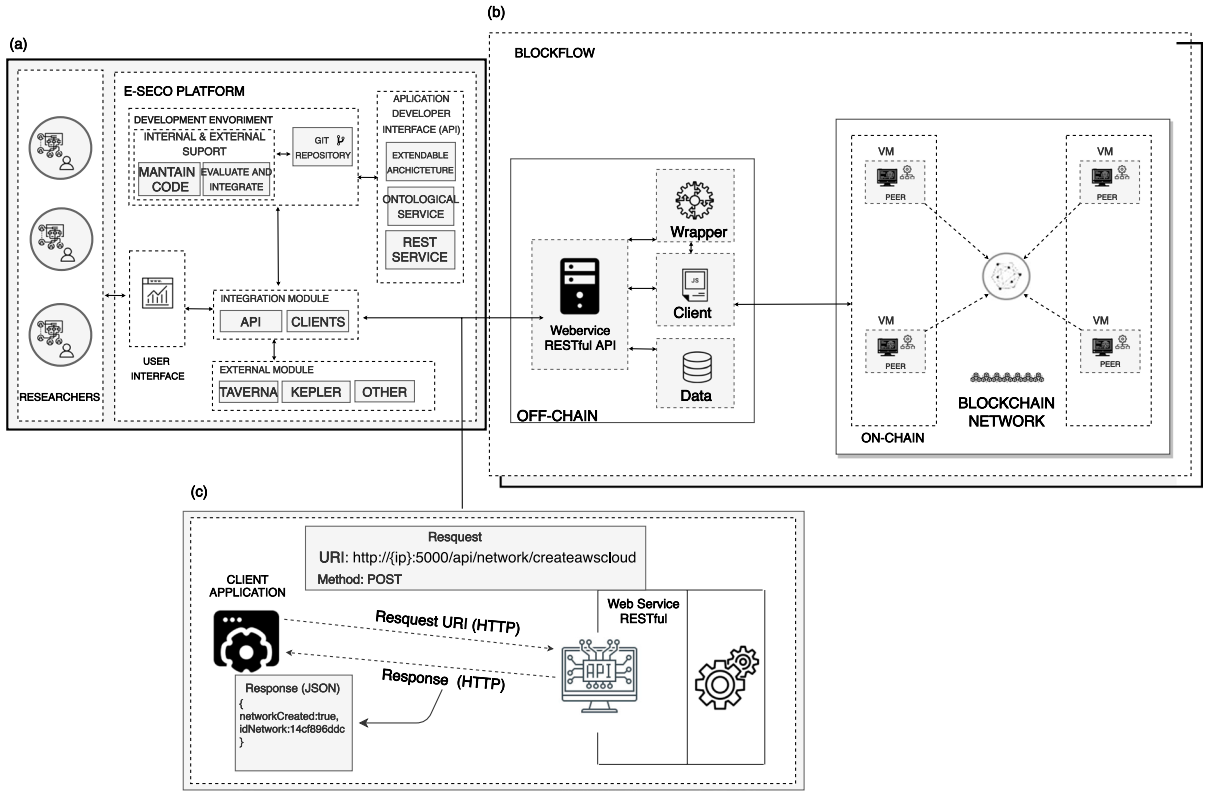


Fig. 1 E-SECO and BlockFlow integration (abstract view) Source: Prepared by the author

BlockFlow architecture. We will discuss these modules in the following sections.

3.2.1 RESTful Web Service API Layer

The RESTful API layer is an API (Application Programming Interface) developed to allow BlockFlow to be integrated with any other platform or application, based on communication via REST (Representational State Transfer) web services, which can be integrated with any other tool that works with the HTTP (RFC 2616) communication protocol. The request operations are through endpoints or URI, and the responses are through JSON. Figure 1c details an example in which the request-response flow between a client (E-SECO platform) and the BlockFlow's API layer is represented.

Based on the request to the web service, via the URI: "http://ip:5000/api/network/createawscloud", a blockchain network for E-SECO´s researchers to collaborate on their experiments in the cloud is created. In response, in JSON format, a Boolean value, "networkCreated: true" is returned if the network is successfully created along with a unique identifier for the created network, "idNetwork:14cf89ddo".

3.2.2 Wrapper Layer

The Wrapper layer translates and integrates the heterogeneous provenance data from different SWMS into the ProvONE model (Fig. 2), which is used as a standard and integrator model in BlockFlow.

To understand the provenance captured by BlockFlow, it is important to detail how BlockFlow organizes the tasks of a SWMS. Therefore, in Fig. 3, a scientific workflow can be seen. It is a directed graph whose nodes are its tasks (t - tn). Each workflow task (t) represents a computational step and has a set of input (ip) and output (op) ports. These tasks consume data (di) as parameters in their input ports and produce data (do) bound

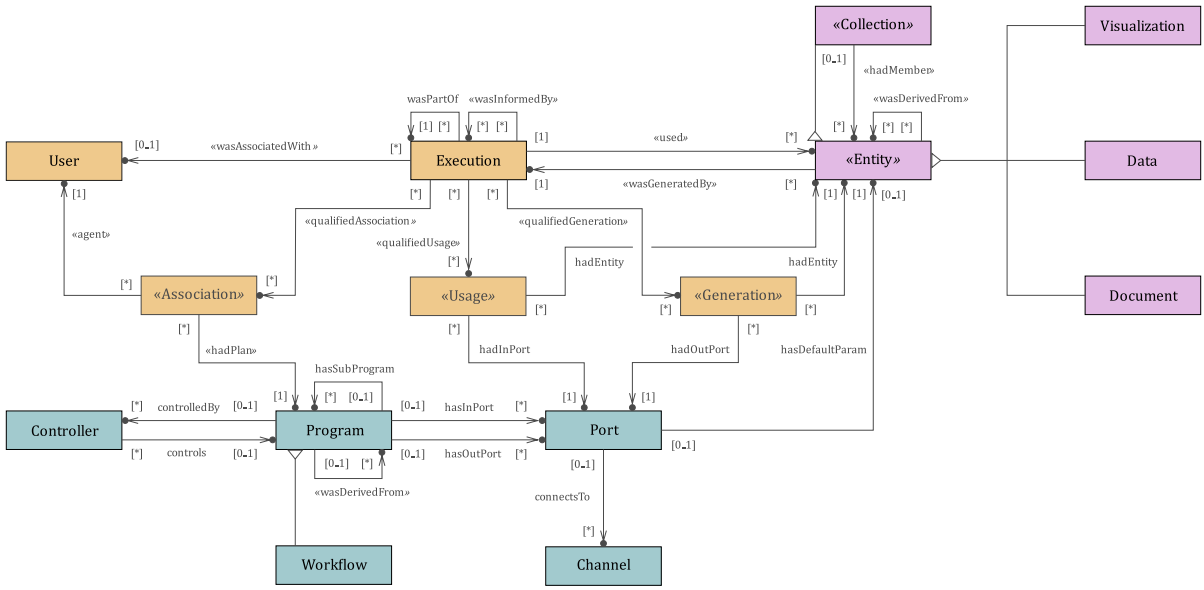


Fig. 2 ProvOne model. Source: [10]

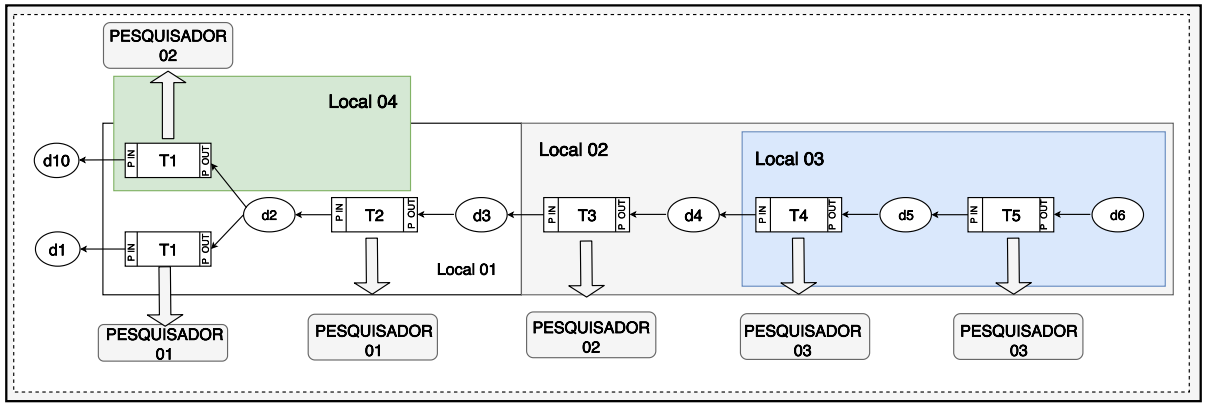


Fig. 3 Task flow in a scientific workflow

to the output ports. Edges denote how these values flow from one task to another and represent data dependencies between those tasks. In Fig. 3, tasks T1 and T2 are performed by researcher O1 at location L1, while task T3 by researcher O2 at location L2, or even task T1 can be performed by researchers 1 and 2 with different data types. In BlockFlow, the mapping of provenance to the ProvONE model (Fig. 2) takes place by observing the invocation of tasks and mainly the life cycle of the datasets consumed or produced during the execution of the workflow.

Then, the capture of provenance data is done through a web service. Each task is instrumented with this service to capture the task's input and output information. Figure 4 represents the provenance mapping of a task belonging to a workflow to specific classes of the ProvONE model (Fig. 2)

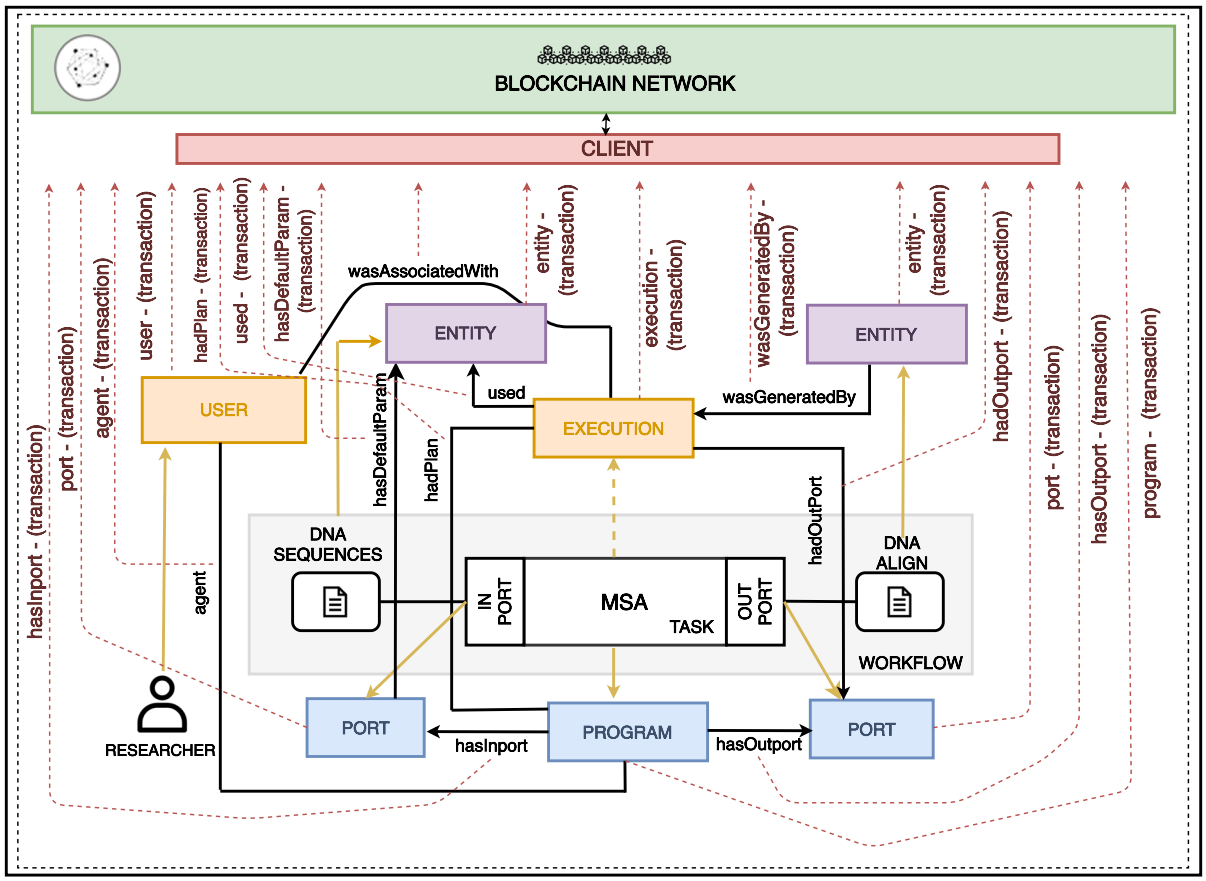


Fig. 4 An example of a task mapping using the ProvONE model

through the Wrapper layer. The workflow task presented in Fig. 4 consists of a Multiple Sequence Alignment (AMS) activity, which receives a multi-stretch file containing DNA sequences as input and generates the alignment of this sequence as output.

Considering the ProvONE model classes and relationships (Fig. 4), the task (AMS) is mapped to the *Program* class of the ProvONE model, and its input and output ports to the *Port* class, where the relationship *hasInport* and *hasOutport* relate them, respectively, to a *Program* (the dashed arrows in yellow express the correspondence between the Workflow elements and the classes of the ProvONE model). When the task is accomplished, it is mapped to the *Execution* class, and the input file "DNA SEQUENCES" is mapped as an *Entity*, which expresses the *hasDefaultParam* relationship for an input *Port* and which is used by an *Execution*. To reduce the volume of provenance data, we store the hash, i.e., the input path and output data, rather than the data, in the blockchain. The *DNA ALIGN* file, mapped as an *Entity*, generated as output, is *wasGenerationBy* by the MSA task *Execution*. The execution of the MSA task, related to the *Execution* class, *wasAssociatedWith* with a researcher, mapped to the *User* class, which in turn *hadPlan* with an instance of the *Program* class.

The Wrapper layer, after mapping, sends each piece of these collected provenance information (classes and relationships) to the Client layer, which then sends them as transactions to the Blockchain Network. The Blockchain Network then, after a series of transformations, records each provenance transaction in the blockchain file system and stores it in CouchDB[11]. Figure 5a represents this mapping in JSON format and examples

[11] https://couchdb.apache.org/.

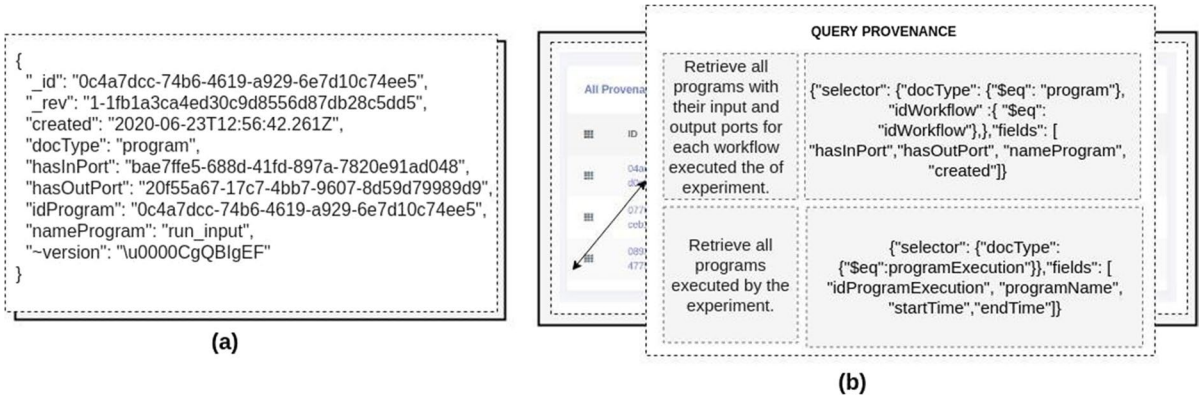


Fig. 5 Example of mapping a task from a workflow to a ProvONE model

Table 2 Task mapping to ProvONE Model

Mapping Workflow	ProvONE:Class	ProvONE:Association
Workflow	Workflow	*wasDerivedFrom*
task	program	*hadPlan*
task.execution	execution	*hadPlan, hadOutPort*
task.port.input	port	*hasInport*
task.port.output	port	*hasOutport*
data.input	entity	*used, hasDefaultParam*
data.output	entity	*wasGeneratedBy, used*

of queries that can be performed, see Fig. 5b. Table 2 presents the correspondence between mapping a set of tasks to the ProvONE Model.

3.2.3 Model Layer

This layer encompasses the model that keeps information related to the configuration flow and BlockFlow API calls. It includes information about creating networks for an experiment, scientists (peers) collaborating on an experiment, and configurations of blockchain networks and channels. Figure 6 shows the data model.

Figure 1c presents a diagram in which the Restful API requests a resource from the data layer. The request to the web service is through the URI: "http://ip:5000/api/network/id", which requests information from a network (experiment) by its unique identifier "id". The data is returned in JSON format, which allows the researcher to visualize the information over a created network. This data encompasses directory, channels, orders, CAs, organizations and peers that are part of an experiment.

3.2.4 Client Layer

This layer allows client applications to connect to the network and invoke calling codes to the ledger, such as query calls, calls to invoke transactions, etc.[12]. Figure 7 details the request flows in the Client layer, to which a researcher can make requests, such as:

- Send provenance data, i.e., after invoking the invokeTransaction() method, the Client layer connects to the peer to update the ledger. Likewise, when a researcher needs to retrieve provenance information, the queryTransaction() method is invoked. After receiving the method request, the Client layer connects to the peer to retrieve the provenance information from the ledger.
- Install or instantiate chaincodes, i.e., for a peer to be able to send or read transactions, a chaincode must be installed on the peer, which is instantiated on the channel to which it belongs. For this, the method installChaincode() is invoked, and the Client layer connects to the peer and installs the chaincode. Likewise, upon receiving the request to instantiate a chaincode on a channel, the

[12] The managing access rights to provenance data and metadata in the system needs to be improved. In this BlockFlow´s version it is not fully implemented.

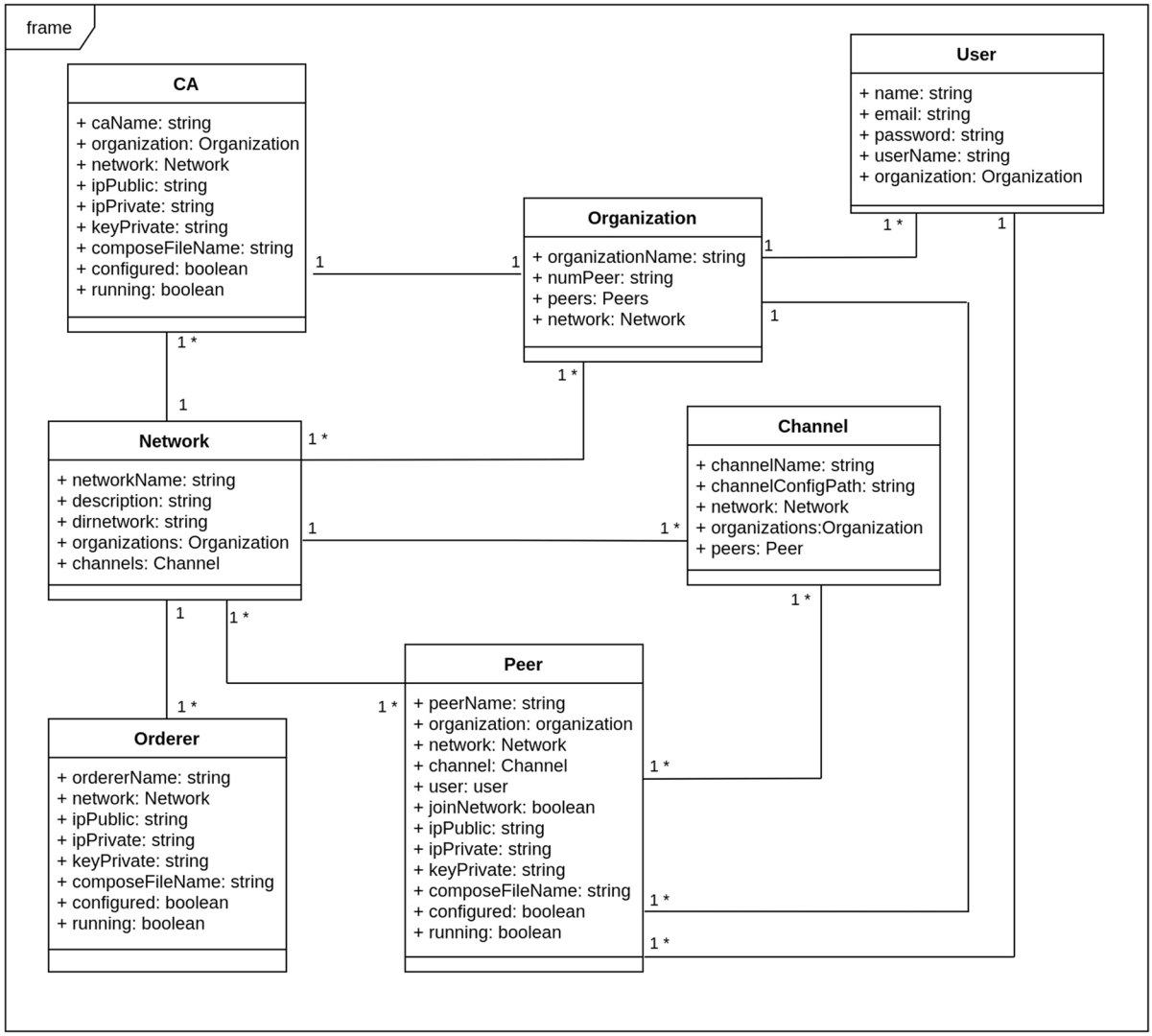


Fig. 6 BlockFlow architecture class model

instantiateChaincode() method is invoked, and the Client layer connects to the network and the channel to instantiate the chaincode.

3.2.5 Blockchain Network Layer

The Blockchain network layer represents a collaborative workflow whose nodes are geographically distributed researchers, connected through instances of local machines or virtual machines in the cloud. Figure 8 represents this collaborative workflow, where each researcher belongs to an experiment and is connected to a node of a blockchain network. In the Blockchain Network, all provenance data collected between researchers will be stored in blocks and distributed among peers. Each node participating in the network has its copy of the ledger, thus allowing data processing, auditing, and transparent querying by different workflows executed in geographically distributed nodes.

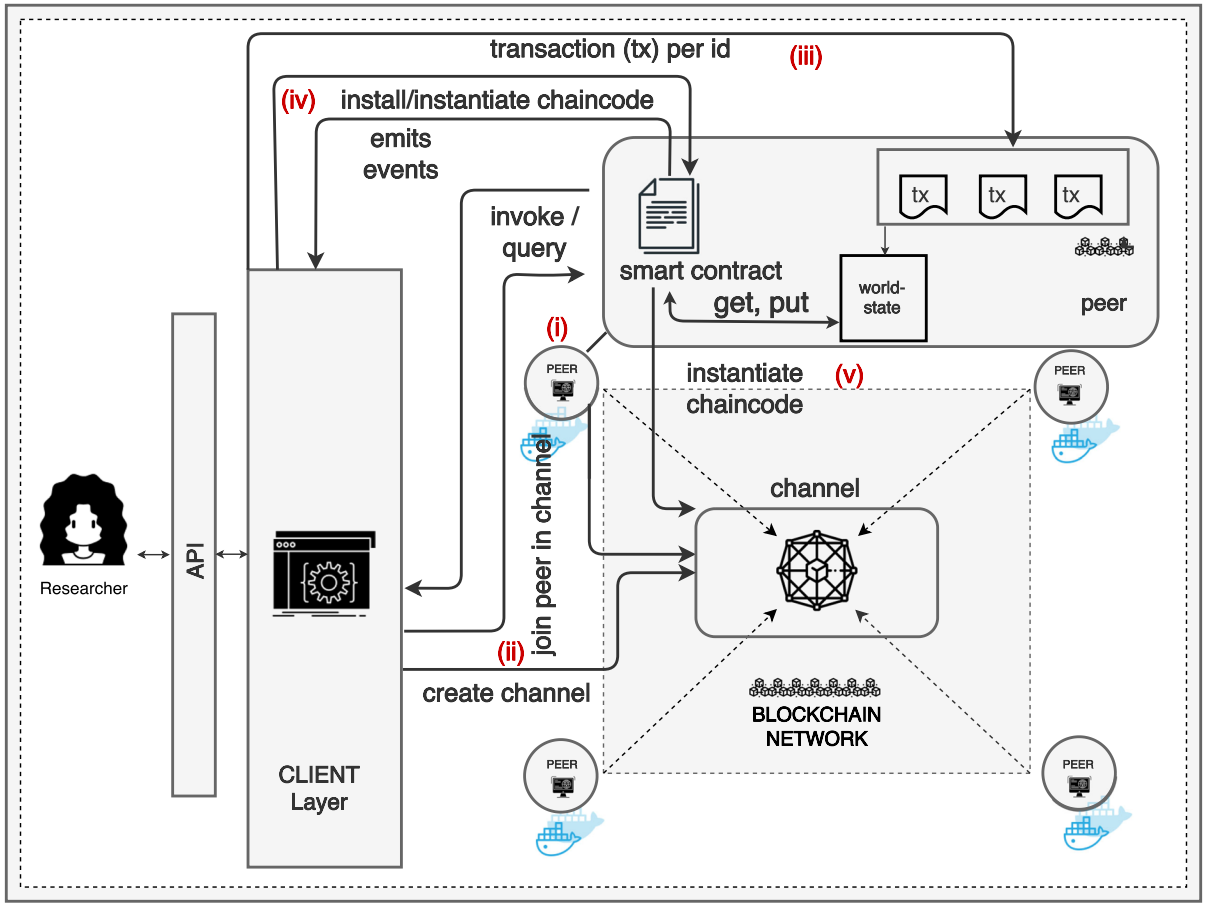


Fig. 7 Request flows to the Client layer (i) start peers, (ii) create channels, (iii) create identities, (iv) install chaincode, (v) instantiate chaincode Source: Prepared by the author

3.2.6 Implementation Technologies

The BlockFlow architecture, available at[13], was divided into two modules: on-chain, and off-chain[14]. The off-chain module comprises the RESTful API web service, Client, Wrapper, and Model layers. The RESTful API web service and the Wrapper Layer were implemented using Node.js technology. The Client layer was implemented using the Hyperledger Fabric SDK for Node.js to interact with the Blockchain Network. The Model layer was implemented using the MongoDB database.

The on-chain module was implemented using the Hyperledger Fabric platform. All modules of the Hyperledger Fabric architecture work based on Dockers container technology and have been specified in yaml files and are initialized using the docker-compose tool. The BlockFlow architecture uses release 1.4 of Hyperledger[15]. Raft is the consensus protocol used.

[13] https://github.com/RaianeQC/blockflow-trust-provenance.

[14] The RESTful API web service and Wrapper Layer were implemented using Node.js technology (https://nodejs.org/en/). The Client layer was implemented using the Hyperledger Fabric SDK for Node.js (https://hyperledger.github.io/fabric-sdk-node/release-1.4/module-fabric-network.html) to interact with the Blockchain Network. The Model layer was implemented using the MongoDB database (https://www.mongodb.com/). The on-chain module was implemented using the Hyperledger Fabric platform (https://www.hyperledger.org/), all the modules of the Hyperledger Fabric architecture work based on the Dockers container technology (https://www.docker.com/) and were specified in yaml files (https://yaml.org/) and are initialized using the docker-compose tool (https://docs.docker.com/compose/).

[15] https://hyperledger.github.io/fabric-sdk-node/release-1.4/module-fabric-network.html.

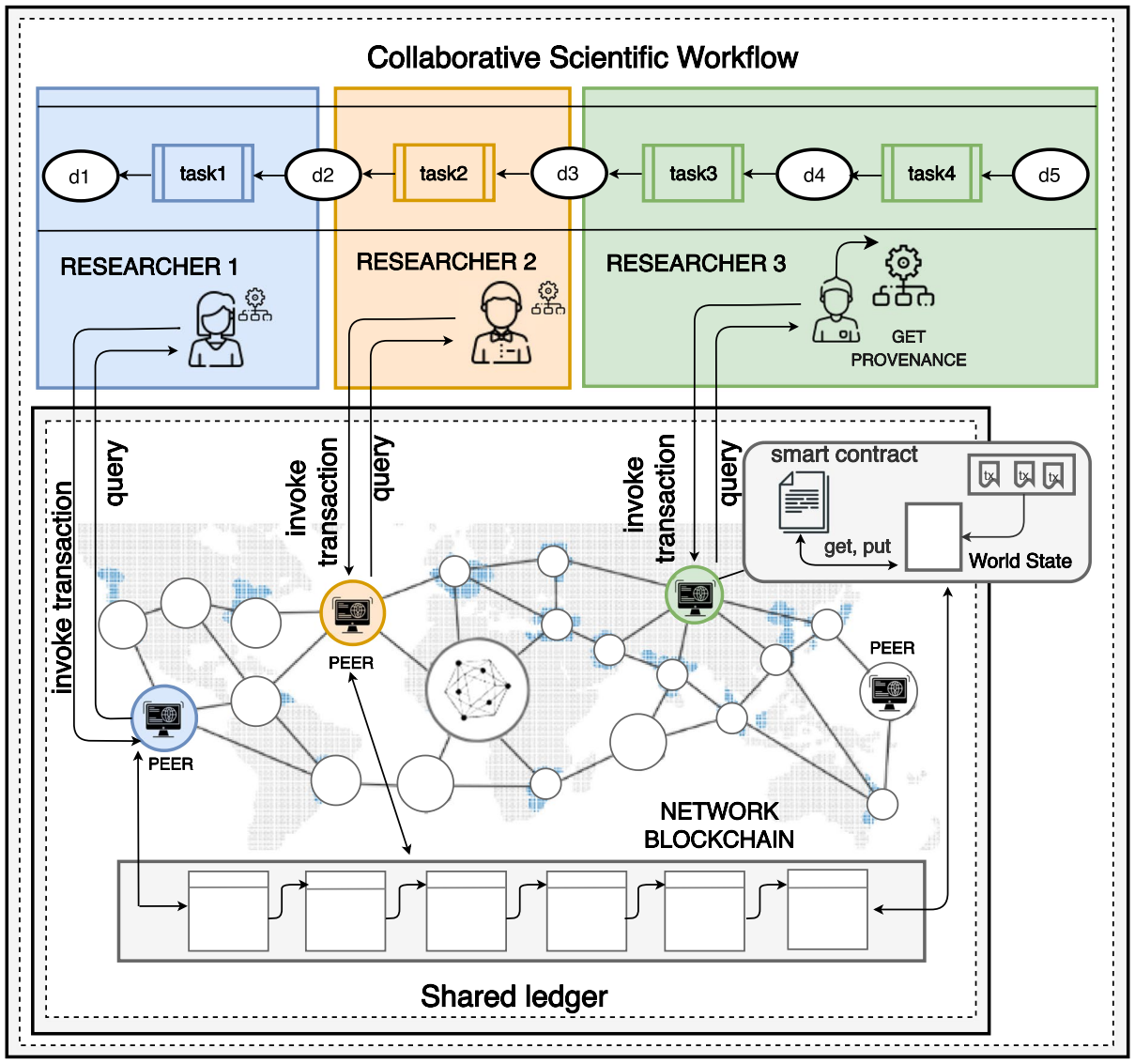


Fig. 8 Blockchain network, collaborative scientific workflow

3.2.7 Performance Analysis

We can have several provenance records during a workflow execution in many data-intensive scenarios. Furthermore, in a collaborative scientific experiment, researchers often simultaneously store or query the provenance repository, to monitor or plan future actions. Thus, they need efficient mechanisms for storing and querying provenance data. To verify if our solution meets this need, we evaluated the performance of the BlockFlow architecture.

The evaluation was conducted on a VM instance on Amazon Elastic Compute Cloud (EC2) with Intel(R) Xeon(R) CPU E5-2690, 2.60 GHz, 24-core CPU, 24GB RAM running Ubuntu 16.04., the Hyperledger Caliper benchmark, and the Hyperledger Fabric Version 1.4.1. The Hyperledger Caliper is provided by the Hyperledger project and is a benchmarking tool used to measure blockchain performance. Some metrics supported by Hyperledger Caliper include (i) Transaction Throughput, which indicates the number of

Table 3 Transaction transfer rate

Transaction type	10	100	1000	10,000
Invoke	4.1(s)	4.7(s)	5.0(s)	5.0(s)
Query	6.2(s)	5.1(s)	5.0(s)	5.0(s)

Table 5 Send rate

Type	10	100	1000	10,000
Invoke	6.3(s)	5.1(s)	5.0(s)	5.0(s)
Query	6.3(s)	5.1(s)	5.0(s)	5.0(s)

transactions performed, valid, and confirmed in the blockchain network per second; (ii) Transaction latency, which indicates how long a transaction takes to be available across the network. This metric is calculated per transaction. The Caliper measures latency through three metrics: (1) minimum transaction latency; (2) maximum latency of a transaction; (3) average latency of all transactions; and (iii) Send Rate, which is an actual Hyperledger Caliper send rate, not based on the destination TPS.

We evaluate BlockFlow according to the specified metrics, varying the transaction workload (10 to 10,000) between requests (write/invoke and query) from provenance data in the ledger performed by a set of pairs simultaneously. The results are shown in Tables 3, 4, and 5.

After the analysis, we had evidence that the architecture can operate at low latency even when dealing with large provenance datasets. This result also provides initial evidence that we can meet scalability and efficiency in distributed environments of scientific experimentation.

3.3 Evaluation

The DSR methodology emphasizes the importance of proper evaluation. As detailed before, we conducted two DSR cycles. Based on the first cycle, we conducted this evaluation, i.e., a Case Study (CS) related to genomic sequencing of the new SARS-CoV-2 coronavirus, to answer our research question correctly.

We aim to analyze BlockFlow architecture with respect to supporting sharing, trust, and traceability in scientific workflows from the point of view of groups of geographically distributed scientists in the context of collaborative experiments.

In this scope, we derive the following research question [57, 69] that will guide us in conducting CS, (RQ) "How can the BlockFlow architecture assist scientists in collaborative scientific experiments, offering an environment that supports data sharing, traceability, and trust?"

From this RQ we derived a Secondary RQ (SRQ), i.e., (SRQ1) Can BlockFlow provide an overview of provenance data transparently, where geographically distributed researchers can verify how the provenance data was created in the blockchain over time? (SRQ2) Can BlockFlow be used as a collaborative and trustable scientific environment supporting the interoperability of provenance data coming from heterogeneous SWMSs? (SRQ3) Can BlockFlow be used as a trustable provenance data exchange environment in data-intensive workflows? (SRQ4) Can BlockFlow be used as an environment that provides privacy to provenance data, where data is shared only between authorized partners? (SRQ5) Can

Table 4 Transaction latency

Type	Workload	Minimum latency	Maximum latency	Average latency
Invoke	10	0.82(s)	2.42(s)	1.62(s)
Invoke	100	0.79(s)	3.45(s)	1.40(s)
Invoke	1000	5.1(s)	5.0(s)	5.0(s)
Invoke	10,000	0.35(s)	3.63(s)	1.30(s)
Query	10	0.01(s)	0.02(s)	0.01(s)
Query	100	0.01(s)	0.02(s)	0.01(s)
Query	1000	0.01(s)	0.01(s)	0.01(s)
Query	10,000	0.01(s)	0.11(s)	0.01(s)

BlockFlow be used as an environment to support reproducibility?

3.3.1 Context

Sequencing is the reading of the genome or transcriptome of an organism composed of DNA or RNA[16]. Genome and transcriptome projects generally have computational support, in which workflows are designed to transform input fragments (read sequences of RNA or DNA fragments) to extract biological information [4]. This is the case of genomic sequencing of the SARS-CoV-2 coronavirus. Several scientific workflows have been designed and are available, such as SciPhy [52] and SciEvol [51], to facilitate the analysis of genomic data and generate a phylogenetic tree from DNA, RNA, and amino acid sequences. It is possible to reconstruct the evolutionary history of a virus from the identification of changes in the genetic sequences from different patients, considering that the virus is transmitted through a population and accumulates mutations in its genetic code. Research teams worldwide have massively sequenced and published viral genomic sequences to study the origin of the new coronavirus. These various virus genomic sequences (SARS-CoV2) have been publicly disclosed in many public databases, including NCBI[17], GISAID[18], and ViPR[19].

The main difficulty in conducting these experiments is data processing. New computing techniques are required such as collaborative, distributed, or high-performance environments (HPC), and grids or clouds for their execution [70]. Furthermore, the expectation that the experiment is reproducible is of fundamental importance.

3.3.2 Planning

From the investigation of the RQ, a CS was conducted (using workflows) involving a phylogenetic analysis based on the complete genome sequences of different coronaviruses, including coronavirus strains (SARS, MERS, and SARS-CoV-2), which were obtained from GISAID and NCBI GenBank. The workflows used in this evaluation were scientific workflows that require processing power and involve a large volume of data, making it an ideal scenario to evaluate the BlockFlow architecture.

Table 6 Virtual machines configuration

VM	Description
Virtual Machine 1	EC2 ID: m4. large – 8 GB RAM, 2 cores.
Virtual Machine2	EC2 ID: m4.4xlarge – 16 GB RAM, 4 cores.
Virtual Machine 3	EC2 ID: m5. large – 8 GB RAM, 2 cores.
Virtual Machine 4	EC2 ID: m5. xlarge – 16 GB RAM, 4 cores

We used the Amazon Elastic Compute Cloud (EC2)4 cloud environment. Four virtual machines were instantiated, with different characteristics, and physically distributed in other places. Table 6 summarizes the different types of virtual machines used in the experiment, with their respective hardware configurations.

To execute the workflows, the following program versions were used: MAFFT version 7.471[20], Readseq version 2.1.19[21], ModelGenerator version v0.85[22], and RAxML and 8.2.12[23]. For ViReport: ViralMSA version 1.0.6[24] using minimap2 version v2 0.17, FastTree version 2.1.11[25], FastRoot[26] and LSD2[27]. The software used in each virtual machine instance in the cloud, considering the need to configure and initialize the necessary services of the collaborative environment based on the blockchain Hyperledger Fabric technology, was Ubuntu Linux 18.04.1 LTS., Docker Engine version (18.06.1-ce), Docker-Compose version (1.13.0), Node (v8.11.4), Hyperledger Fabric (v1.4.1) and Go Lang — 1.12.0.

[16] Both are formed by a set of letters, nitrogenous bases, which work as a code (words), known as nucleotide sequences (Adenine (A), Cytosine (C), Thymine (T) and Guanine (G)). In the case of RNA, we have Uracil (U) in place of Thymine) [50].
[17] ncbi.nlm.nih.gov.
[18] gisaid.org.
[19] viprbrc.org.
[20] https://mafft.cbrc.jp/alignment/software/source.html.
[21] https://readseq-bioinformatics-data-conversion.soft112.com/.
[22] http://mcinerneylab.com/software/modelgenerator/.
[23] https://cme.h-its.org/exelixis/web/software/raxml/.
[24] https://github.com/niemasd/ViralMSA.
[25] http://www.microbesonline.org/fasttree/.
[26] https://github.com/uym2/MinVar-Rooting.
[27] https://github.com/tothuhien/lsd2.

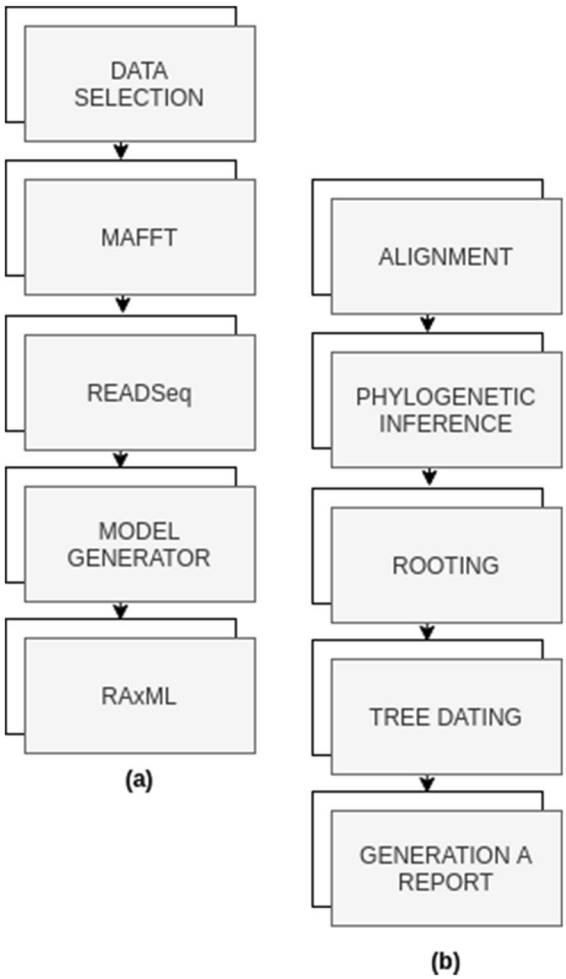


Fig. 9 SciPhy (**a**) and ViReport (**b**) workflows high-level view

Although Phylogenetic Analysis experiments can be implemented using different configurations, we consider two workflow implementations, SciPhy[28] [52] and ViReport [59]. Figure 9a illustrates a high-level view of SciPhy's activities. The second workflow used was ViReport[29] [59]. Figure 9b presents a high-level view of ViReport's activities.

The scenario for experimenting encompasses 4 Research Institutions ("UFJF" "UFMG", "UFF", and "UFRJ") involving four researchers (R), one from UFJF, one from UFMG, and one from UFF. They form a team named Team A, and 1 researcher from UFRJ, named Team B. Considering the geographically distributed teams, in order to experiment, the researchers needed a collaborative environment that offered trustworthiness to the processed data and allowed provenance data querying, considering that reproducibility is essential. Furthermore, these researchers needed an environment that would support data interoperability from different institutions, considering that the researchers run parts of the experiment using different SWMSs. Finally, they needed an environment that offered a scalable, robust, and high-performance infrastructure to meet the needs of running the data-intensive experiment.

The genetic alignment activities (MSA) were performed in a distributed way. Each researcher could align a piece of data that was then concatenated to generate a super alignment. Thus, using Blockflow, each geographically distributed team adopted slightly different approaches (different genomes to generate different trees), managed by different workflows running in different SWMSs.

This scenario led to data interoperability, where researchers had to analyze the data, especially the provenance data, coming from different SWMS, in an integrated way. Thus, Team A, used Taverna SWMS and Team B, used Kepler SWMS. Although the SciPhy and ViReport workflows differ, they have the same goal, and their results can be comparable. They could query integrated provenance data, generated by the SciPhy and ViReport workflows and consolidated by Blockflow using ProvONE.

3.3.3 Execution

We created a channel to share provenance transactions only between interested parties, considering privacy. This channel was called "sarscovChannel", and the institutions "UFJF" "UFMG", "UFF", and "UFRJ" are part of it.

Next, the settings of the virtual machine instances in the cloud, such as public IP, private IP, username, public DNS, and cloud server access keys, were specified. For researchers to collaborate, all the necessary configuration to start the network in the cloud was also performed automatically, such as (i) starting peers, (ii) creating channels, (iii) creating identities, (iv) installing chaincode, (v) instantiating chaincode. Figure 10 illustrates the sequence.

[28] Sciphy [51] is a scientific phylogenetic analysis workflow that was designed to generate phylogenetic trees.

[29] ViReport [59] is a workflow to perform phylogenetic analyzes on viral sequences and generate comprehensive molecular epidemiological reports.

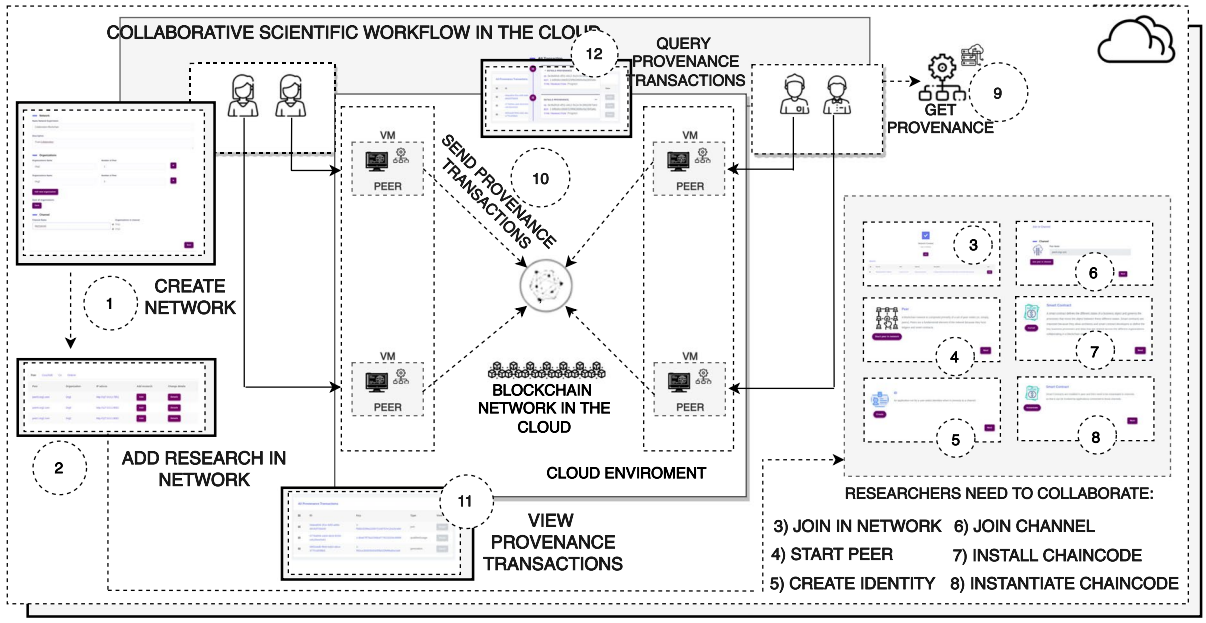


Fig. 10 Action flows to create a collaborative environment using BlockFlow architecture. Source: Prepared by the author

Figure 11 presents the interface that enabled researchers to implement and configure their blockchain networks through BlockFlow. The following were specified: 1 - (Fig. 11)-A - Name of the experiment as "ColaboracaoNaNuvemCovid19"; 2 - (Fig. 11)-B - A description of the experiment as "Laboratório colaborativo para o novo coronavírus (SARS-CoV-2). Árvores filogenéticas."; 3 - (Fig. 11-C) - Which organizations and number of distributed peers (researchers) would collaborate in the experiment ((Fig. 11-C-1) UFJF institution with 1 peer, (Fig. 11-C-2) UFF institution with 1 peer, (Fig. 11-C-3) UFMG institution with 1 peer, ((Fig. 11-C-4) UFRJ institution with 1 peer.

To share provenance transactions only between interested parties, considering privacy, a channel called "sarscovChannel" was created, of which, according to Fig. 11-D, "UFJF" and "UFMG", "UFF", "UFRJ" are part.

Provenance capture is done in real-time and is independent of SWMS. It was necessary to instrument each activity (*task*) of the workflows with a web service to capture the data. Figure 12 shows the SciPhy workflow with different tasks instrumented by BlockFlow´s web service in SWMS Taverna. Furthermore, for a complete provenance translation into the ProvONE model, the provenance at runtime (retrospective provenance) must be linked to the prospective provenance. It is important to emphasize that this task was not trivial, and the researchers needed support for some steps.

Figure 13 presents the registered data of the executed workflows[30]. It is required, but not mandatory, to store the data during the execution. In this CS, we used 25 (Fig. 14) complete genome sequences of coronavirus strains (including SARS, MERS, and SARS-CoV-2) as input files. Also, 61 coronavirus strains are from different countries and regions. All sequences were obtained from GISAID and NCBI GenBank. After uploading, we generated a hash for each file. This option is important considering the reproducibility of the workflow, as it allows researchers to

[30] This data is important because during the collection of the retrospective provenance, the transaction hash data, or the workflow id must be sent to the web service, as well as the user's token received during authentication. This is a way of relating the workflow and the researcher (user) with their respective execution (retrospective provenance). Furthermore, the researcher token (user) is a way to confirm the authenticity and retrieve the user's identity as a node belonging to the blockchain network, in addition to ensuring that the data is later signed on the network. If the user's identity cannot be verified, the information will be rejected and disregarded, and the provenance data will not be recorded.

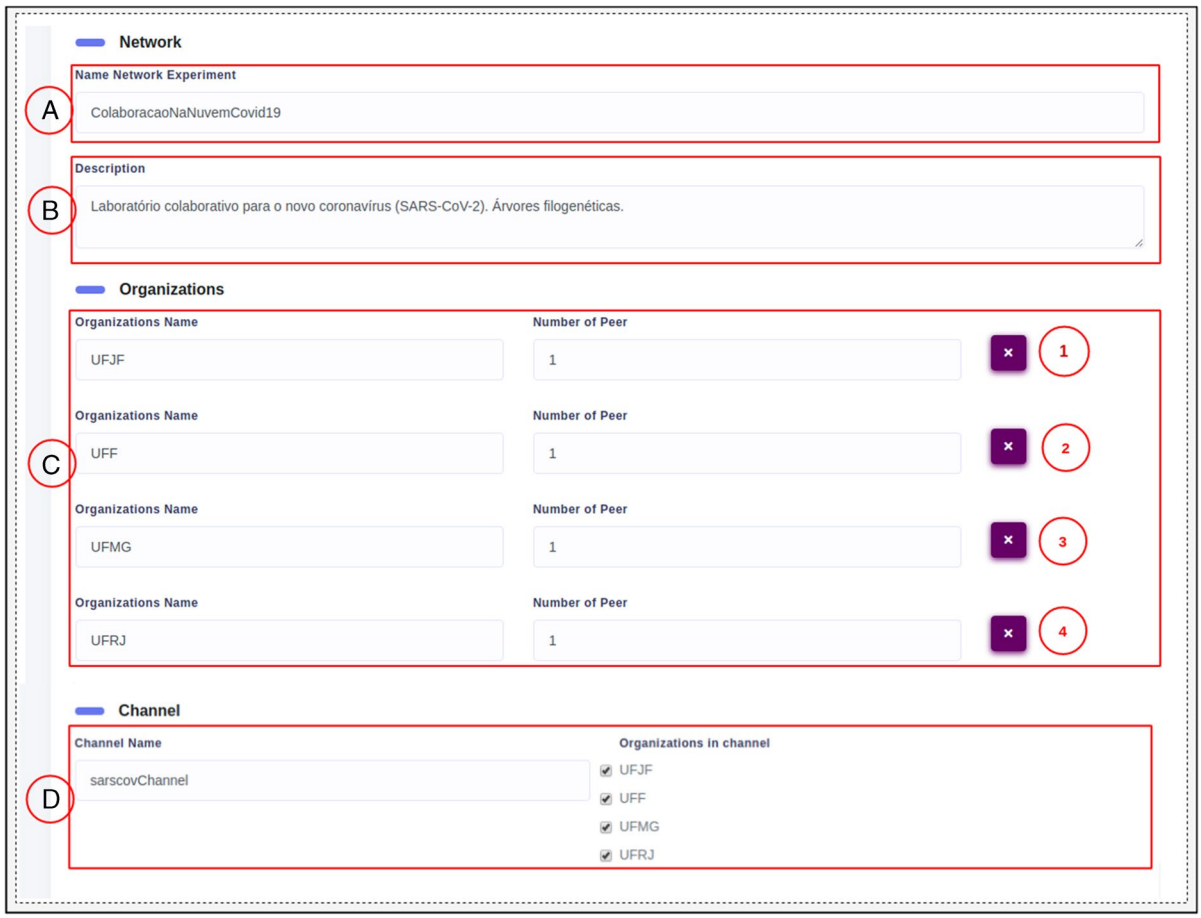


Fig. 11 User interface for researchers to create collaborative networks using the BlockFlow architecture. Source: Prepared by the author

analyze whether the research data used or generated in an experiment has content equivalent to what was published and shared in the experiment. Therefore, this data can be compared with the data saved in the ProvONE classes (*Entity*) stored in the blockchain during the workflow execution, which is immutable[31].

The Wrapper layer transformed each data to the ProvONE format and then sent it as transactions to the blockchain network. This operation guaranteed the immutability (integrity) and transparency of the provenance information. The provenance collected during the execution of the workflows can be seen as shown in Fig. 15. Due to the distributed and immutable nature of the blockchain, this shared provenance is transparent. All nodes (scientists), connected to the network that makes up the experiment can verify and visualize how the provenance was created in the blockchain over time. In addition, the data can be peer-reviewed. Thus, as detailed in the next section, we tracked all data updates between nodes.

3.3.4 Provenance Data Analysis and Queries

The researchers were able to perform queries from the BlockFlow web interface, as shown in Fig. 17, to obtain an overview of the provenance data collected during the execution of the experiment. These queries can be with fixed components, as

[31] In the following section, during the provenance analysis, an analysis will be made considering the collected provenance and comparison with the collected hash.

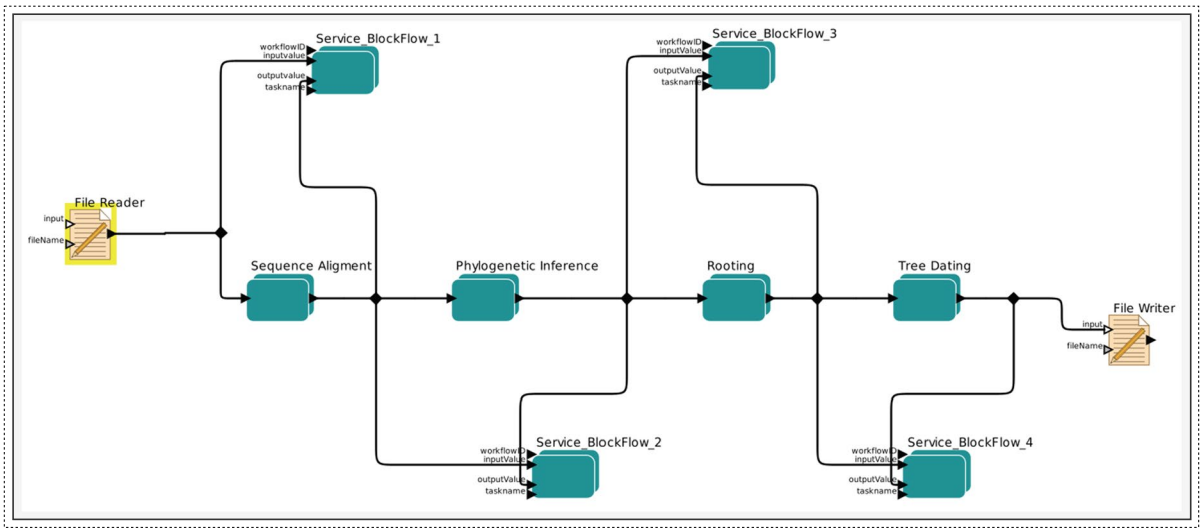
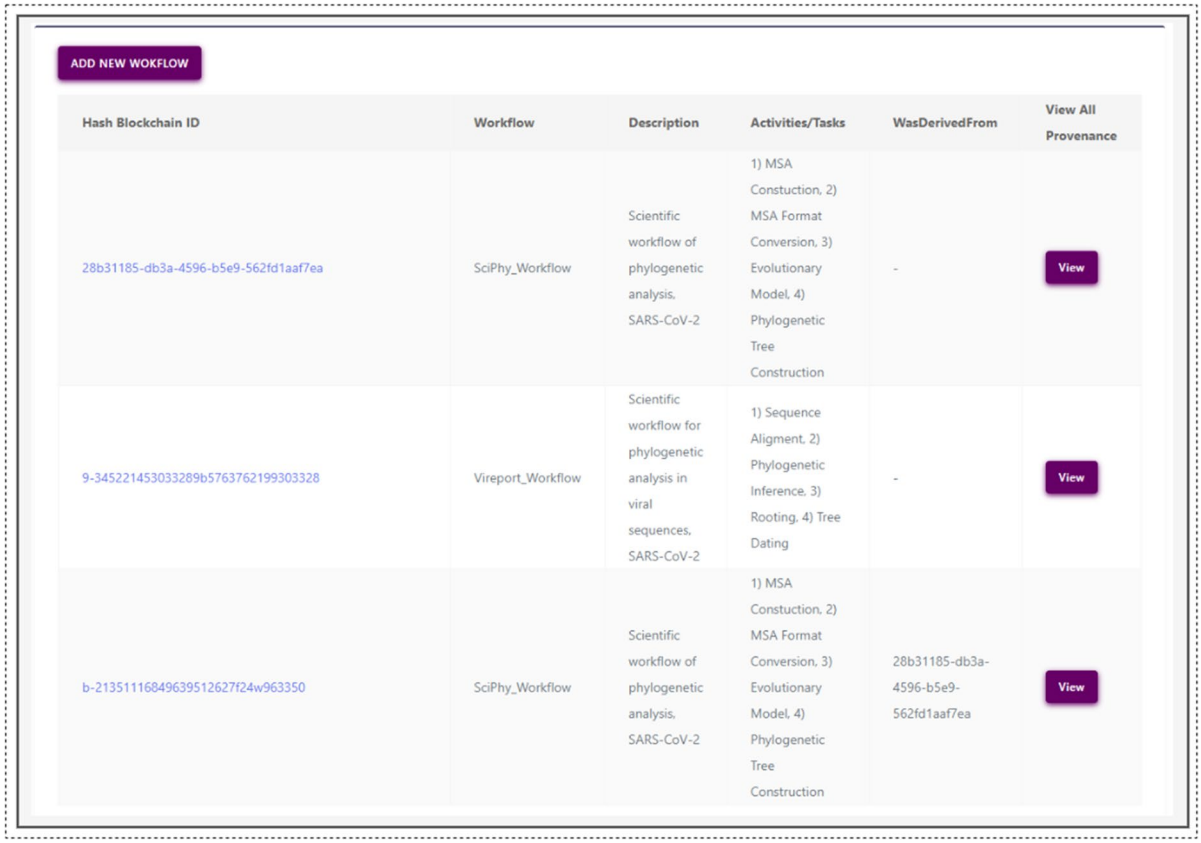


Fig. 12 Sciphy workflow, using Taverna SWfMS, instrumented with BlockFlow's web service. Source: Prepared by the author



ADD NEW WOKFLOW

Hash Blockchain ID	Workflow	Description	Activities/Tasks	WasDerivedFrom	View All Provenance
28b31185-db3a-4596-b5e9-562fd1aaf7ea	SciPhy_Workflow	Scientific workflow of phylogenetic analysis, SARS-CoV-2	1) MSA Constuction, 2) MSA Format Conversion, 3) Evolutionary Model, 4) Phylogenetic Tree Construction	-	View
9-345221453033289b5763762199303328	Vireport_Workflow	Scientific workflow for phylogenetic analysis in viral sequences, SARS-CoV-2	1) Sequence Aligment, 2) Phylogenetic Inference, 3) Rooting, 4) Tree Dating	-	View
b-213511168496395126 27f24w963350	SciPhy_Workflow	Scientific workflow of phylogenetic analysis, SARS-CoV-2	1) MSA Constuction, 2) MSA Format Conversion, 3) Evolutionary Model, 4) Phylogenetic Tree Construction	28b31185-db3a-4596-b5e9-562fd1aaf7ea	View

Fig. 13 Sciphy, ViReport data stored in BlockFlow. Source: Prepared by the author

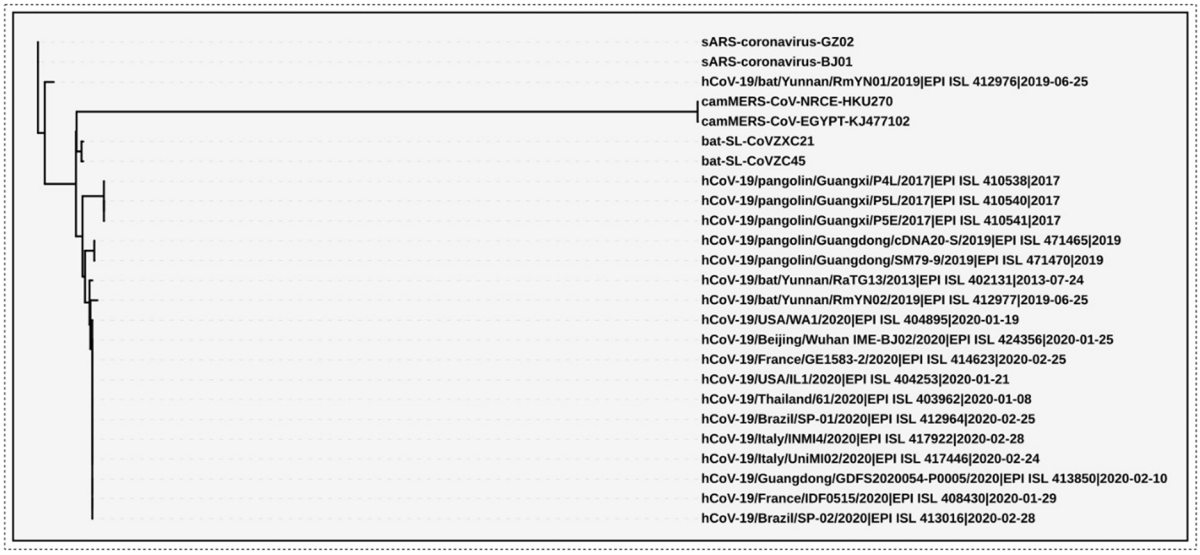


Fig. 14 Phylogenetic tree based on the sequences of 25 genomes, complete of coronaviruses, including SARS-CoV-2, SARS-CoV, HCoV, SARS-like CoV, and MERS-CoV. Source: Prepared by the author

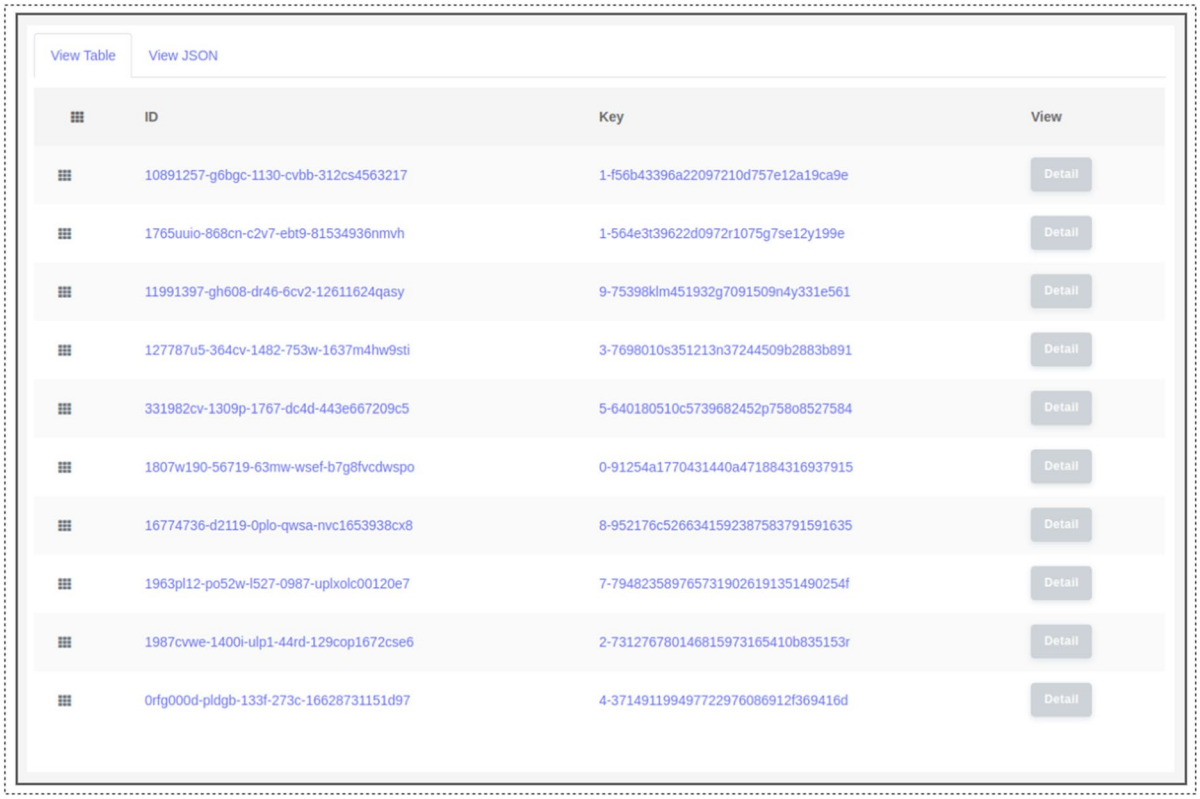


Fig. 15 Provenance collected during the execution of the experiment

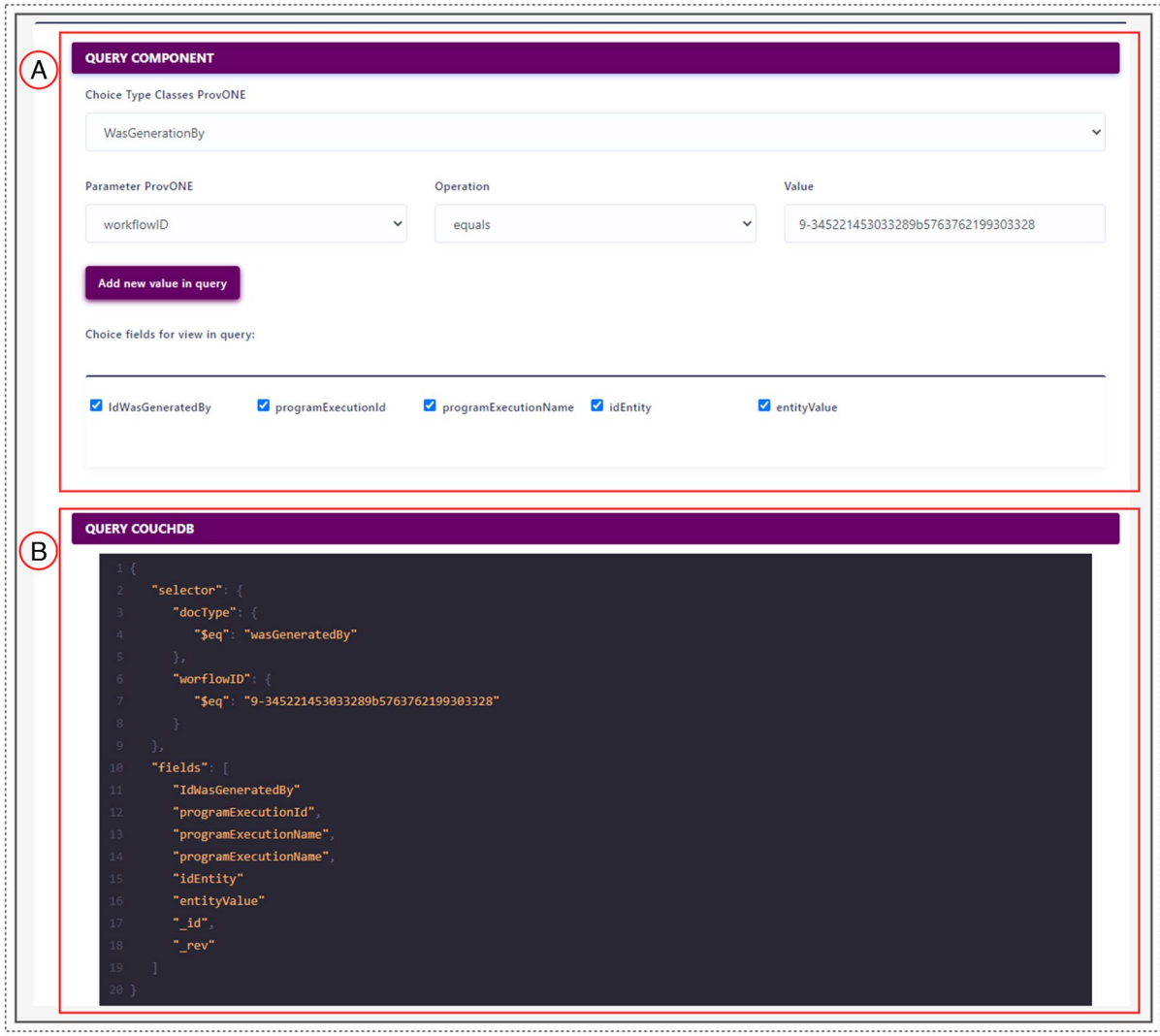


Fig. 16 Query (Q1). Source: Prepared by the author

shown in Fig. 16 – A, or through queries in the CouchDB database format, as shown in Fig. 16 - B. As shown in Figs. 16 and 17, in BlockFlow, the visualization of the returned data through the executed queries, can be done through a table or using the JSON format. The latter allows interoperability with the E-SECO platform or any other application that needs to consume BlockFlow data.

Next, we present a query executed by the researchers on the provenance dataset collected during the execution of this CS, considering the SciPhy and ViReport workflows.

Q1) Retrieve all activity executions with their generated data for the ViReport workflow provenance graph. Figure 17 - A shows the query using the fixed components. In this, the relationship (*wasGenerationBy*) of the ProvONE model was chosen, the parameter (*workFlowID*) to specify the (id) of the specific workflow, to which the researchers wanted to access the data and, below, the fields they would like to be returned (*idWasGeneratedBy, programExecution, programExecutionName, idEntity, entityValue*). Figure 17 - B illustrates the same query but in the CouchDB database format.

After processing the queries, the researchers could download the results in JSON format. Figure 18

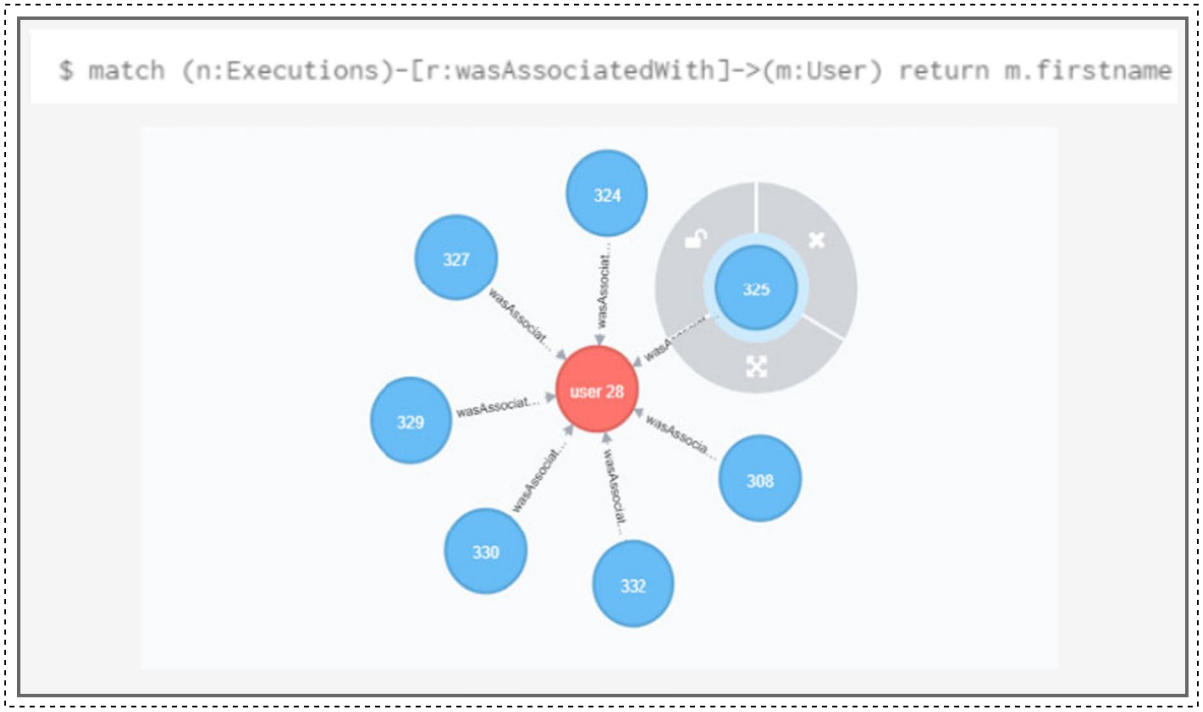


Fig. 17 Cypher search. Source: Prepared by the author

Fig. 18 Comparison of data used or generated in an experiment with its previously captured provenance data. Source: Prepared by the author

shows a search in Cypher, where it was possible to associate and return a researcher with their respective executions. This integration with Neo4j also illustrates another example of the connection capability of BlockFlow with external applications.

4 Results

According to [47], one of the possible causes of the reproducibility crisis is the lack of raw data used in research. Therefore, it is important to make the data processed during the execution of workflows available in a trustable way, as researchers did in the evaluation presented in the previous section, using the Blockflow architecture. Using BlockFlow during the provenance collection, the researchers could register all the experiment´s task invocations. The data processed by these tasks were collected and organized according to the ProvONE model. The hash of this data was stored, and its path, date, and time of execution, i.e., its provenance information. It also was possible to compare the data processed during a workflow executed by a specific researcher to promote trust and reproducibility.

The researchers carried out queries in the collected provenance, as shown in the previous section, and compared it with the data stored and made available by other researchers. We uploaded all the data used and generated during the execution of the Sciphy workflow. Then all data previously recorded from the executed workflow were retrieved (provenance data). We compared the hash data (provenance data) with the data available in the architecture. The results, in JSON format, can be seen in Fig. 18. This comparison related each data with the provenance collected. So, it could be observed whether the data used or generated in the experiment are equivalent to what was published and shared.

As a result, it was possible to verify the feasibility of the BlockFlow architecture in helping geographically distributed researchers to capture, store, and query integrated provenance data in a trustworthy and transparent environment, sharing their data, allowing for the experiment's reproducibility.

Revisiting the secondary research questions (SRQ), we present the following arguments:

(**SRQ1**) Can BlockFlow provide an overview of provenance data in a transparent way, where geographically distributed researchers can verify how the provenance data was created in the blockchain over time?

As we can see during CS execution, due to the distributed and immutable nature of the blockchain, the provenance data shared between researchers in an experiment is shared in real-time. Moreover, all provenance data updates can be tracked and visualized, through queries, by geographically distributed nodes (researchers). BlockFlow can provide an integrated view of provenance data transparently, as provenance data can be queried in an integrated way, as presented in Figs. 17 and 18.

(**SRQ2**) Can BlockFlow be used as a collaborative and trustworthy scientific environment supporting the interoperability of provenance data coming from heterogeneous SWMSs?

During CS, we executed the workflows in different SWMs. As shown in Fig. 10, the SciPhy workflow with its different tasks was instrumented in SWMS Taverna. The same was done in Kepler SWMS. It should be noted that although there are advantages offered using these SWMs in managing an experiment, they capture provenance data in models that are not fully interoperable. In order to capture and interoperate the provenance data of these workflows between the distributed peers, BlockFlow used a web service component. The capture was done in real-time and independent of SWMS. After capturing the data, they were stored according to the ProvONE model in the blockchain. It was therefore possible to perform integrated queries to the provenance data of the different executed workflows (Figs. 17 and 18) and verify that BlockFlow can be used in a trustworthy way, supporting the interoperability of provenance data coming from heterogeneous SWMSs.

(**SRQ3**) Can BlockFlow be used as a trustworthy provenance data exchange environment in data-intensive workflows?

The scientific workflows chosen to run CS are phylogenetic workflows that are data intensive. These often need to run in high-performance, collaborative environments such as cloud computing environments. To do so, we specify a collaborative environment that is (i) high-performance and (ii) trustworthy, immutable, and private. To meet item (i) we used a cloud computing environment, provisioning virtual machine instances (Amazon Elastic Compute Cloud (Amazon EC2)). This environment offered a variety of resources such as hardware and software, under elasticity, without the need for scientists to acquire computational infrastructure. To support item (ii), we developed this environment based on blockchain, where no data can be changed, making it trustworthy and immutable. Data could not be accessed by unauthorized parties, which makes it private. Furthermore, access to the distributed data was made transparent, that is, the scientist did not need to worry about the details to access the data. Thus, there are indications that BlockFlow can be used as a trustworthy provenance exchange environment in data-intensive workflows.

(**SRQ4**) Can BlockFlow be used as an environment that provides privacy to provenance data, where data is shared only between authorized partners?

In a blockchain without permission, any node can check any transaction that has taken place in the chain. Thus, when privacy or confidentiality is needed in transactions, other cryptographic means are needed. BlockFlow was based on the blockchain with permission - Hyperledger Fabric. During the creation of the execution environment, Blockflow requires that geographically distributed organizations and peers that must collaborate in the experiment and that are part of a channel be specified. Channels maintain privacy, confidentiality, and isolate activities between authorized parties. Thus, in the CS execution, the participating nodes had to register and have identities to transact and send provenance data. Furthermore, during provenance collection they had to send the *Token* to confirm the authenticity and recover their identity as a user belonging to the blockchain network to guarantee that the data was signed on the network. Thus, it was demonstrated that BlockFlow can be used as an environment that provides privacy to provenance data. Data was shared only between authorized parties or persons, thus maintaining privacy.

(**SRQ5**) Can BlockFlow be used as an environment for collaborative scientific experimentation, considering reproducibility?

In Blockflow, reproducibility is related to querying, processing, and analyzing provenance, which helps understand the results of a scientific experiment. As we could see in the CS execution, Blockflow allowed the collection of provenances, its immutable storage, and the provenance query in a transparent and trustworthy way. Finally, it allowed the analysis (Fig. 18) of the data used and generated during the experiments supported by the workflows' execution.

From the analysis of secondary research questions (SRQ), we were able to analyze the main research question (RQ) "How can the BlockFlow architecture assist scientists in collaborative scientific experiments, offering an environment that supports data sharing, traceability, and trust? ".

Considering the secondary research questions (SRQ) results, we have evidence that BlockFlow offers a trusted environment for managing distributed provenance data. The architecture offers components that can facilitate collaboration in scientific experimentation, considering reproducibility (QS5), interoperability (QS2), transparency (QS1), privacy (QS4), and trust (QS3) of data-intensive scientific workflows, in addition to the correct interpretation of scientific data among geographically distributed researchers, based on provenance data query. However, it is important to note that new experiments must be carried out to evaluate BlockFlow. Furthermore, the results shown here are only valid for this dataset. However, we can identify similar scenarios where similar results can be achieved.

4.1 Threats to Validity

It is important to analyze the reliability, especially whether the results are biased. We therefore discuss some issues that can affect the validity of the results.

Construct Validity We selected indicators to evaluate traceability and trust attributes. During the experiment, the data processed was available to the researchers in a trackable and trustable way. All provenance data updates can be tracked and visualized. However, these indicators might not be good indicators for the experiment context. We can use additional indicators to mitigate this threat, considering different contexts. Moreover, BlockFlow supports the interoperability of provenance data from heterogeneous SWMSs. For this purpose, data are stored according to the ProvONE model. However, the limited number of experiments used and the diversity of scientists can represent a threat. Additional evaluations need to be carried out to reduce this threat.

Internal Validity During the CS, the scientific workflows were phylogenetic, and we specified a collaborative environment to assess BlockFlow's quality of trust and provenance attribute. The results are still preliminary, and although they indicate a positive outcome, a more detailed study is needed to present additional findings. However, the features offered by the collaborative environment can pose a threat. In a more complex context, other collaborative services need to be integrated, and, as a result, we must reassess trust and provenance. Additionally, if we need a cloud computing environment different from the one considered in the case study, we should use other variables to re-evaluate trust and provenance.

External Validity The Case Study deals with a dataset associated to a specific experiment that deals with gene sequencing. We need to carry out evaluations considering other e-Science contexts before generalizing our results. However, it is possible to identify situations where we can obtain similar CS results, and the knowledge acquired can be transferred to similar real-world experiments.

Reliability We attempted to present details of the execution of the study, but some information was probably not complete. We have made the documentation available[32] to ensure the case study reruns to mitigate this threat.

5 Conclusion

This work presented BlockFlow: a blockchain-based architecture to support trust and traceability in collaborative research, helping in transparency, interoperability, and reproducibility of experiment results. It provides mechanisms that bring trustability to data and processes in collaborative scientific workflows. The proposed solution is integrated with a scientific software ecosystem platform called E-SECO but is independent, i.e.; it can be integrated with other applications or ecosystems. This work also presented a systematic literature mapping to support the proposed approach, which identified and categorized existing works related to blockchain and provenance data.

Through an evaluation in the genetic sequencing domain, we discussed the feasibility of the proposal in supporting scientists to work collaboratively and distributed, addressing the following aspects: (i) sharing provenance data in a more trustworthy way to ensure transparency and reproducibility of the results obtained, and (ii) supporting the execution of data-intensive workflows, anchored by the cloud computing paradigm. Another contribution is to support systems that require interoperability and reproducibility of results from scientific workflows, interoperating from different SWMSs, and increasing trust in collaborative research. From the evaluation, we answered the SRQs, supporting the RQ. We observed that the proposed solution helped in scientific collaboration in the genetic sequencing domain by providing sharing, traceability and trust. Furthermore, it addresses the heterogeneity of data shared in collaborative scientific workflows, facilitating geographically distributed researchers' interpretation and analysis of these data.

In general, we can highlight the following contributions: (i) An API to connect BlockFlow with other applications or platforms that aim to allow its users to create blockchain networks to collaborate and promote trust, traceability and sharing of scientific experiments; (ii) A GUI-based interface for creating collaborative environments and blockchain networks, which allows researchers to easily implement blockchain networks to collaborate; (iii) The specification of collaborative environments and blockchain networks, through cloud infrastructures, for the execution of data-intensive workflows; (iv) The specification and implementation of a provenance collector that uses a RESTful WebService API for provenance capture; (v) A wrapper that translates and interoperates heterogeneous provenance data, coming from different SWMS, to the ProvONE model format, which is used as standard and integrator model in BlockFlow; (vi) an infrastructure for immutable storage and for querying and analyzing information from collaborative scientific experiments; (vii) The exporting of data collected from provenance to the JSON model, allowing the data to be integrated with other platforms; (viii) The possibility of uploading provenance data used and generated during the execution of an experiment so as to analyze whether the data actually used, or generated in an experiment (provenance data), have content equivalent to those published and shared by researchers to carry out an analysis related to reproducibility of the experiment.

This work was developed to increase the trust and traceability of data from scientific experiments, also promoting reproducibility, privacy, transparency, and interoperability. However, the results are limited and cannot be generalized but the knowledge built, and the results can be transferred to other contexts.

As limitations of the architecture, we can cite that in a blockchain-based application, storing whole files is not possible as it is necessary to store hashes of information. Although this limitation can be overcome with the IPFS blockchain, in BlockFlow, all input and output data generated during the collaborative workflow execution is shared outside the chain. Thus, it is necessary to verify the integrity of the data, comparing whether the stored hash matches the data used as input and output during the execution of the workflow. In future BlockFlow releases, we plan to settle this issue using IPFS approaches and investigate the use of ontologies in the processing of provenance data. In addition, we intend to conduct new case studies in other contexts to assess the support offered by Blockflow. Other provenance models derived from PROV may support the capture of provenance in different domains. We are also carrying out studies to support the capture of provenance related to software processes using BlockFlow, evolving the work of [11].

[32] https://github.com/RaianeQC/blockflow-trust-provenance.

Table 7 Mapping research questions results

MQ	Results
How has blockchain technology been used as a mechanism, method, and tool for provenance?	This MQ aimed to identify which application areas blockchain technology has been used as a mechanism, method, or tool for provenance. Nine application areas were identified, according to the selected primary studies. Internet of Things, Cloud Computing, Scientific Workflow, Research Data Sharing, Data Sharing, Health, Fog Computing, Supply Chain, and others. Internet of Things and Cloud Computing represent the majority of works, i.e., 46% of the papers. Internet of Things represented 23% of the total and Cloud Computing, 21%. Supply Chain and Data sharing represent 9% each. Scientific Workflow represented 7% of the total, Health 7%, Research Data Sharing 2%, Fog computing 2%, and finally others, which represented 14%.
In which vehicles were the articles published?	Through the results, it was possible to identify that most of the studies were published in scientific conferences, totaling 74% of the verified articles.
What is the distribution of studies over the years?	We could see that most works were published from 2018 onwards, with an ascending curve. The range from 2019 to 2022 was the most promising, which indicates an increase in published works in the area, showing the importance of this research.
What approaches are most used by the researchers?	It was possible to identify that most studies detail architectures, representing 66% of the analyzed articles. Frameworks represent 25%, Models represent 5% and, finally, Prototypes and Approach represent 2% of each of the analyzed articles.
What are the advantages and benefits obtained from the approaches using blockchain technology for provenance?	The three biggest motivations were: i)"Information Integrity" with a total of 20%, integrity is guaranteed in a blockchain since data cannot be deleted or changed, thus ensuring the credibility of the provenance data, ii) "Trust Assurance" represented a total of 19%, i.e., in a blockchain as members share a single view of the data, it is possible to see all the details of a transaction, which offers trust and iii) "Increased security " represented a total of 17%, i.e., the distributed nature of a blockchain allows each node that participates in the network to have and verify the ledger data, thus increasing information security.
What are the methods, standards, or technologies most used (or proposed) by the authors to support their approaches?	The objective of this MQ was to discover which platforms were used and to identify their main benefits and applicability considering the provenance data. The most used blockchain platform was Ethereum. The choice of Ethereum was justified because it is the most known platform and because it is open source. On the other hand, there has been an increase in the use of the Hyperledger Fabric platform. However, in most studies, 41%, there was no consensus about the mechanism used.
In the approaches found, which models are used to represent provenance data?	Most of the studies did not present a specific provenance model. 94% of the works use some proprietary model. 4% of the papers use OPM and 2% PROV. This clearly shows that PROV, being a new model, has not yet been widely used, despite its benefits. This result confirms the importance of having an approach that allows the interoperability and integration of provenance data, since 87% of the studies use proprietary models that, without any specific mechanism, are not interoperable.

Funding This work was partially funded by UFJF/Brazil, CAPES/Brazil, CNPq/Brazil (grant: 311595/2019-7), and FAPEMIG/Brazil (grant: APQ-02685-17), (grant: APQ-02194-18).

Data Availability The datasets generated during and/or analyzed during the current study are available on Github at https://github.com/RaianeQC/blockflow-trust-provenance.

Declarations

Conflict of Interest The authors declare that they have no conflict of interest.

Appendix 1 Criteria for Inclusion and Exclusion of Articles

Inclusion criteria were: (IC1) - The study proposes a solution that uses the blockchain as a mechanism for the storage and management of provenance data; (IC2) - The study was written in English; (IC3) - The study was published from 2008 to 2022; (IC4): Available as full papers in digital libraries.

Exclusion Criteria were: (EC1) - Matches the keyword in the search string, but the context is different from the search purposes; (EC2) - The abstract did not address any aspect of the research questions; (EC3) - Duplicated, that is, the work has already been retrieved from another digital library; (EC4) - The article does not contain an abstract; (EC5) -It is not a primary study; (EC6) - Not available for the university (UFJF) credentials; (EC7) - The study was published as a short paper; (EC81) – The study is not written in English; (EC9) - The study was not published in a conference or journal related to Computer Science; (EC10) - The study was not published in a peer review vehicle; (EC11) - The study was published before 2008; (EC12) - The study does not propose a solution that uses blockchain as a mechanism for the storage and management of provenance data.

To assist in the mapping, the Parsif.al[33] tool was used. Some exclusion criteria have already been applied when using this tool, due to the availability, by Parsif.al, of filters, such as the publication year filter.

[33] https://parsif.al/.

Appendix 2 Analysis of Mapping Research Questions

Table 7

References

1. Al-Mamun, A., Yan, F., Zhao, D.: SciChain: Blockchain-enabled Lightweight and Efficient Data Provenance for Reproducible Scientific Computing. 2021 IEEE 37th International Conference on Data Engineering (ICDE), pp.1853–1858 (2021). https://doi.org/10.1109/ICDE51399.2021.00166
2. Ambrosio, L., Magaldi, H., David, J., Braga, R., Arbex, W., Campos, M., Capilla, R.: Enhancing the reuse of scientific experiments for agricultural software ecosystems. J. Grid Comput. (2021). https://doi.org/10.1007/s10723-021-09583-x
3. Androulaki, E., Barger, A., Bortnikov, V., Cachin, C., Christidis, K., De Caro, A., Enyeart, D., Ferris, C., Laventman, G., Manevich, Y., Muralidharan, S., Murthy, C., Nguyen, B., Sethi, M., Singh, G., Smith, K., Sorniotti, A., Stathakopoulou, C., Vukolic, M., Cocco, S., Yellick, J.: Hyperledger fabric: a distributed operating system for permissioned blockchains. In: Proceedings of the Thirteenth EuroSys Conference, 1–15 (2018). https://doi.org/10.1145/3190508.3190538
4. Ansorge, W.: Next-generation DNA sequencing techniques. New Biotechnol. **25**(4), 195–203 (2009). https://doi.org/10.1016/j.nbt.2008.12.009
5. Azaria, A., Ekblaw, A., Vieira, T., Lippman, A., Medrec: Using blockchain for medical data access and permission management. In: 2016 2nd iNternational Conference on Open and Big Data (OBD) (pp. 25–30). IEEE (2016). https://doi.org/10.1109/OBD.2016.11
6. Belloum, A., Inda, M., Vasunin, D., Korkhov, V., Zhao, Z., Rauwerda, H., Breit, T., Bubak, M., Hertzberger, L.: Collaborative e-science experiments and scientific workflows. IEEE Internet Comput. **15**(439–47) (2011). https://doi.org/10.1109/MIC.2011.87
7. Bhuyan, F., Lu, S., Reynolds, R., Zhang, J., Ahmed, I.: A security framework for scientific workflow provenance access control policies. IEEE Trans. Serv. Comput. (2019). https://doi.org/10.1109/TSC.2019.2921586
8. Bosch, J.: From software product lines to software ecosystems. SPLC, 2009, Pittsburgh, PA, USA: Proceedings of the 13th International Software Product Line Conference, 111– 119 (2009)
9. Callahan, S., Freire, J., Santos, E., Scheidegger, C., Silva, C., Huy, V.O.: T, VisTrails: visualization meets data management. In: Proceedings of the 2006 ACM SIGMOD international conference on Management of data, 745–747 (2006). https://doi.org/10.1145/1142473.1142574
10. Cao, Y., Jones, C., Cuevas-Vicenttín, V., Jones, M.B., Ludäscher, B., McPhillips, T.M., Missier, P., Schwalm, C.R., Slaughter, P., Vieglais, D., Walker, L., Wei, Y.: ProvONE: extending PROV to support the DataONE scientific community. Available via (2016). http://homepages.cs.ncl.ac.uk/paolo.missier/doc/dataone-prov-3-years-later.pdf cited Jan 2021

11. Castro, G., Werner, C., Braga, R., Teixeira, E., Stroele, V., Araújo, M.: Design, application and evaluation of PROV-SwProcess: A PROV extension data model for software development processes. J. Web Semant. V **71**, 100676 (2021). https://doi.org/10.1016/j.websem.2021.100676
12. Chen, W., Liang, X., Li, J., Qin, H., Mu, Y., Wang, J.: Blockchain based provenance sharing of scientific workflows. In: w2018 IEEE International Conference on Big Data (Big Data). IEEE, 3814–3820 (2018). https://doi.org/10.1109/BigData.2018.8622237
13. Classe, T., Braga, R., David, J.M., Campos, F., Arbex, W.: A distributed infrastructure to support scientific experiments. J. Grid Comput. **1**, 1–26 (2017). https://doi.org/10.1007/s10723-017-9401-7
14. Coelho, R., Braga, R., David, J.M., Dantas, M., Stroele, V., Campos, F.: Blockchain for reliability in collaborative scientific workflows on cloud platforms. In: 2020 IEEE Symposium on Computers and Communications (ISCC). IEEE, 1–7 (2020). https://doi.org/10.1109/ISCC50000.2020.9219729
15. Coelho, R., Braga, R., David, J.M., Dantas, M., Stroele, V., Campos, F.: Integrating blockchain for data sharing and collaboration support in scientific ecosystem platform. In: Proceedings of the 54th Hawaii International Conference on System Sciences, 264 (2021). https://doi.org/10.24251/HICSS.2021.031
16. Costa, F., De Oliveira, D., Mattoso, M.: Towards an adaptive and distributed architecture for managing workflow provenance data. In: 2014 IEEE 10th International Conference on e-Science. IEEE, 79–82 (2014). https://doi.org/10.1109/eScience.2014.59
17. Davidson, S., Freire, J.: Provenance and scientific workflows: challenges and opportunities. In: Proceedings of the 2008 ACM SIGMOD International Conference On Management of Data, p. 1345–1350 (2018). https://doi.org/10.1145/1376616.1376772
18. De Oliveira, D., Baião, F., Mattoso, M.: Towards a taxonomy for cloud computing from an e-science perspective. In: Cloud Computing, pp. 47–62. Springer, London (2010). https://doi.org/10.1007/978-1-84996-241-4_3
19. Deelman, E., Chervenak, A.: Data management challenges of data-intensive scientific workflows. In: 2008 Eighth IEEE International Symposium on Cluster Computing and the Grid (CCGRID) (pp. 687–692). IEEE (2008). https://doi.org/10.1109/CCGRID.2008.24
20. Deelman, E., Mehta, G., Singh, G., Su, M., Vahi, K.: Pegasus: mapping large-scale workflows to distributed resources. In: Workflows for e-Science, pp. 376–394. Springer, London (2007). https://doi.org/10.1007/978-1-84628-757-2_23
21. Demichev, A., Kryukov, A., Prikhod'ko, N.: Business process engineering for data storing and processing in a collaborative distributed environment based on provenance metadata, smart contracts and blockchain technology. J. Grid Comput. **19**, 3 (2021). https://doi.org/10.1007/s10723-021-09544-4
22. Fanning, K., Centers, D.: Blockchain and its coming impact on financial services. J. Corp. Acc. Finan. **27**(5), 53–57 (2016). https://doi.org/10.1002/jcaf.22179
23. Fernando, D., Kulshrestha, S., Herath, J., Mahadik, N., Ma, Y., Bai, C., Yang, P., Yan, G., Lu, S.: SciBlock: A blockchain-based tamper-proof non-repudiable storage for scientific workflow provenance. In: 2019 IEEE 5th International Conference on Collaboration and Internet Computing (CIC). IEEE, 81–90 (2019). https://doi.org/10.1109/CIC48465.2019.00019
24. Fraser, H., Parker, T., Nakagawa, S., Barnett, A., Fidler, F.: Questionable research practices in ecology and evolution. PLoS One. **13**(7), e0200303 (2018). https://doi.org/10.1371/journal.pone.0200303
25. Freire, J., Chirigati, F.: Provenance and the different flavors of computational reproducibility. IEEE Data Engineering Bulletin, v. **41**(1), 15 (2018)
26. Freire, J., Koop, D., Santos, E., Silva, C.: Provenance for computational tasks: a survey. Comput. Sci. Eng. **10**(3), 11–21 (2008). https://doi.org/10.1109/MCSE.2008.79
27. Groth, P., Moreau, L.: PROV-overview. An overview of the PROV family of documents. Available via (2013). http://eprints.soton.ac.uk/id/eprint/356854 cited Jun 2021
28. Han, R., et al.: Vassago: Efficient and Authenticated Provenance Query on Multiple Blockchains. 2021 40th International Symposium on Reliable Distributed Systems (SRDS), pp. 132–142 (2021). https://doi.org/10.1109/SRDS53918.2021.00022
29. Hang, L., Choi, E., Kim, D.-H.: A novel EMR integrity management based on a medical blockchain platform in hospital. Electronics **8**, 467 (2019). https://doi.org/10.3390/electronics8040467
30. Hevner, A., March, S., Park, J., Ram, S.: Design science in information systems research. MIS Q. 75–105 (2004). https://doi.org/10.2307/25148625
31. Hevner, A., March, S., Park, J., Ram, S.: Design science in information systems research. Manage. Inform. Syst. Q. **28**(1), 6 (2008)
32. Hey, T., Tansley, S., Tolle, K., et al.: The fourth paradigm: data-intensive scientific discovery. Microsoft research [S.l.], Redmond (2009)
33. Hey, T., Trefethen, A.: The fourth paradigm 10 years on. Informatik Spektrum. **42**(6), 441–447 (2020). https://doi.org/10.1007/s00287-019-01215-9
34. Himanen, L., Geurts, A., Foster, A., Rinke, P.: Data-driven materials science: status, challenges, and perspectives. Adv. Sci. **6**, 1900808 (2019). https://doi.org/10.1002/advs.201900808
35. Jandre, E., Dirr, B., Braganholo, V.: Provenance in collaborative in oisilico scientific research: a survey. ACM SIGMOD Rec. **49**(2), 36–51 (2020). https://doi.org/10.1145/3442322.3442329
36. Jyoti, A., Chauhan, R.K.: A blockchain and smart contract-based data provenance collection and storing in cloud environment. Wirel. Netw **28**, 1541–1562 (2022). https://doi.org/10.1007/s11276-022-02924-y
37. Karastoyanova, D., Stage, L.: Towards collaborative and reproducible scientific experiments on blockchain. In: International Conference on Advanced Information Systems Engineering. Springer, Cham, p. 144–149 (2018). https://doi.org/10.1007/978-3-319-92898-2_12
38. Kim, H., Laskowski, M.: Toward an ontology-driven blockchain design for supply-chain provenance. Intell. Syst. Account. Finan. Manag. **25**(1), 18–27 (2018). https://doi.org/10.1002/isaf.1424
39. Kochovski, P., Gec, S., Stankovski, V., Bajec, M., Drobintsev, P.D.: Trust management in a blockchain based

fog computing platform with trustless smart oracles. Futur. Gener. Comput. Syst. **101**, 747–759 (2019). https://doi.org/10.1016/j.future.2019.07.030
40. Koop, D., Freire, J.: Reorganizing workflow evolution provenance. In: 6th USENIX Workshop on the Theory and Practice of Provenance (TaPP 2014) (2014)
41. Liang, X., Shetty, S., Tosh, D., Kamhoua, C., Kwiat, K., Njilla, L.: Provchain: A blockchain-based data provenance architecture in cloud environment with enhanced privacy and availability. In: 2017 17th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGRID). IEEE, p. 468–477 (2017). https://doi.org/10.1109/CCGRID.2017.8
42. Lim, C., Lu, S., Chebotko, A., Fotouhi, F.: Prospective and retrospective provenance collection in scientific workflow environments. In: 2010 IEEE International Conference on Services Computing. IEEE, p. 449–456 (2010). https://doi.org/10.1109/SCC.2010.18
43. Ludascher, B., Altintas, I., Berkley, C., Higgins, D., Jaeger, E., Jones, M., Lee, E., Tao, J., Zhao, Y.: Scientific workflow management and the Kepler system. Concurr. Comput. Pract. Experience. **18**(10), 1039–1065 (2006). https://doi.org/10.1002/cpe.994
44. Mendes, Y., Braga, R., Stroele, V., De Oliveira, D.: Polyflow: A soa for analyzing workflow heterogeneous provenance data in distributed environments. In: Proceedings of the XV Brazilian Symposium on Information Systems, p. 1–8 (2019). https://doi.org/10.1145/3330204.3330259
45. Missier, P., Soiland-Reyes, S., Owen, S., Tan, W., Nenadic, A., Dunlop, I., Willians, A., Oinn, T., Goble, C.: Taverna, reloaded. In: International Conference On Scientific and Statistical Database Management. Springer, Berlin, p 471–481 (2010). https://doi.org/10.1007/978-3-642-13818-8_33
46. Missier, P., Woodman, S., Hiden, H., Watson, P.: Provenance and data differencing for workflow reproducibility analysis. Concurr. Comput. Pract. Experience. **28**(4), 995–1015 (2016). https://doi.org/10.1002/cpe.3035
47. Miyakawa, T.: No raw data, no science: another possible source of the reproducibility crisis. Mol. Brain **13**, 24 (2020). https://doi.org/10.1186/s13041-020-0552-2
48. Möller, J., Fröschle, S., Hahn, A.: Permissioned blockchain for data provenance in scientific data management. In: Ahlemann, F., Schütte, R., Stieglitz, S. (eds.) Innovation Through Information Systems. WI 2021. Lecture Notes in Information Systems and Organisation, vol. 48. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-86800-0_2
49. Moreau, L., Freire, J., Futrelle, J., McGrath, R.E., Myers, J., Paulson, P.: The open provenance model: An overview. In: International Provenance and Annotation Workshop, pp. 323–326. Springer, Berlin (2008). https://doi.org/10.1007/978-3-540-89965-5_31
50. Nakamoto S.: Bitcoin: a peer-to-peer electronic cash system. Decentralized Business Review, 21260 (2008)
51. Ocana, K., De Oliveira, D., Horta, F., Dias, J., Ogasawara, E., Mattoso, M.: Exploring molecular evolution reconstruction using a parallel cloud based scientific workflow, In: Brazilian Symposium on Bioinformatics. Springer, Berlin, p 179–191 (2012). https://doi.org/10.1007/978-3-642-31927-3_16
52. Ocana, K., De Oliveira, D., Ogasawara, E., Dávila, A., Lima, A., Mattoso, M.: SciPhy: a cloud-based workflow for phylogenetic analysis of drug targets in protozoan genomes. In: Brazilian Symposium on Bioinformatics. Springer, Berlin, p 66–70 (2011). https://doi.org/10.1007/978-3-642-22825-4_9
53. Oliveira, W., Missier, P., Ocana, K., De Oliveira, D., Braganholo, V.: Analyzing provenance across heterogeneous provenance graphs. In: International Provenance and Annotation Workshop, pp. 57–70. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-40593-3_5
54. Pajooh, H., Rashid, M.A., Alam, F., et al.: IoT Big Data provenance scheme using blockchain on Hadoop ecosystem. J. Big Data **8**, 114 (2021). https://doi.org/10.1186/s40537-021-00505-y
55. Ramachandran, A., Kantarcioglu, M.: Smartprovenance: a distributed, blockchain based dataprovenance system. In: Proceedings of the Eighth ACM Conference on Data and Application Security and Privacy., p. 35–42 (2018). https://doi.org/10.1145/3176258.3176333
56. Shantharam, M., Lin, K., Sakai, S., Sivagnanam, S.: Integrity protection for research artifacts using open science chain's command line utility. In Practice and Experience in Advanced Research Computing (PEARC '21). Association for Computing Machinery, New York, Article: 31, 1–4 (2021). https://doi.org/10.1145/3437359.3465587
57. Shull, F., Mendonça, M., Basili, V., Carver, J., Maldonado, J., Fabbri, S., Travassos, G., Ferreira, M.: Knowledge-sharing issues in experimental software engineering. Empir. Softw. Eng. **9**(1), 111–137 (2004). https://doi.org/10.1023/B:EMSE.0000013516.80487.33
58. Silva, C., Freire, J., Callahan, S.: Provenance for visualizations: Reproducibility and beyond. Comput. Sci. Eng. **9**(5), 82–89 (2007). https://doi.org/10.1109/MCSE.2007.106
59. Song, M., Moshiri, N.: An analysis of SARS-CoV-2 using ViReport. Available via (2020). https://doi.org/10.1101/2020.06.20.163162 cited Jun 2021
60. Song, Z., et al.: An improved data provenance framework integrating blockchain and PROV Model, 2020. International Conference on Computer Science and Management Technology (ICCSMT), pp. 323–327 (2020). https://doi.org/10.1109/ICCSMT51754.2020.00073
61. Tenopir, C., Dalton, E., Allard, S., Frame, M., Pjesivac, I., Birch, B., Pollock, D., Dorsett, K.: Changes in data sharing and data reuse practices and perceptions among scientists worldwide. PLoS One. **10**(8), e0134826 (2015). https://doi.org/10.1371/journal.pone.0134826
62. Tosh, D., Shetty, S., Liang, X., Kamhoua, C., Njilla, L.: Consensus protocols for blockchain-based data provenance: Challenges and opportunities. In: 2017 IEEE 8th Annual Ubiquitous Computing, Electronics and Mobile Communication Conference (UEMCON). IEEE, p. 469–474 (2017). https://doi.org/10.1109/UEMCON.2017.8249088
63. Van Rossun, J.: Blockchain for research: Perspectives on a new paradigm for scholarly communication. Digital Science, November (2017). https://doi.org/10.6084/m9.figshare.5607778.v1
64. Wan, S., Li, M., Liu, G., Wang, C.: Recent advances in consensus protocols for blockchain: a survey. Wirel. Netw. **26**(8), 5579–5593 (2020). https://doi.org/10.1007/s11276-019-02195-0

65. Wang, W., Hoang, D., Hu, P., Xiong, Z., Niyato, D., Wang, P., Wen, Y., Kim, D.: A survey on consensus mechanisms and mining strategy management in blockchain networks. IEEE Access. **7**, 22328–22370 (2019). https://doi.org/10.1109/ACCESS.2019.2896108
66. Wenyi, T., Changhao, C., Chanyang, J., Taeho, J.: Trac-2Chain: trackability and traceability of graph data in blockchain with linkage privacy. In: Proceedings of the 37th ACM/SIGAPP Symposium on Applied Computing (SAC '22). Association for Computing Machinery, New York, NY, USA, 272–281 (2022). https://doi.org/10.1145/3477314.3506993
67. Wozniak, J., Armstrong, T., Wilde, M., Katz, D., Lusk, E., Foster, I.: Swift/t: Large-scale application composition via distributed-memory dataflow processing. In: 2013 13th IEEE/ACM International Symposium on Cluster, Cloud, and Grid Computing. IEEE, p. 95–102 (2013). https://doi.org/10.1109/CCGrid.2013.99
68. Xu, X., Weber, I., Staples, M.: Architecture for Blockchain Applications. Springer, Cham (2019)
69. Yin, R., Robert, K.: Case Study Research Design and Methods. Sage, Los Angeles (2014)
70. Zhao, Y., Fei, X., Raicu, I., Lu, S.: Opportunities and challenges in running scientific workflows on the cloud. In: 2011 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery. IEEE, p. 455–462 (2011). https://doi.org/10.1109/CyberC.2011.80