LUCE: A BLOCKCHAIN-BASED DATA SHARING PLATFORM FOR MONITORING DATA LICENSE ACCOUNTABILITY AND COMPLIANCE

ABSTRACT

Easy access to data is one of the main avenues to accelerate scientific research. As a key element of scientific innovations, data sharing allows the reproduction of results, helps prevent data fabrication, falsification, and misuse. Although the research benefits from data reuse are widely acknowledged, the data collections existing today are still kept in silos. Indeed, monitoring what happens to data once they have been handed to a third party is currently not feasible within the current data-sharing practices. We propose a blockchain-based system to trace data collections, and potentially create a more trustworthy data sharing process. In this paper, we present the LUCE (License acountability and Compliance) architecture as a decentralized blockchain-based platform supporting data sharing and reuse. LUCE is designed to provide full transparency on what happens to the data after they are shared with third parties. The contributions of this work are: the definition of a generic model and an implementation for decentralized data sharing accountability and compliance and to incorporates dynamic consent and legal compliance mechanisms. We test the scalability of the platform in a real-time environment where a growing number of users access and reuse different datasets. Compared to existing data-sharing solutions, LUCE provides transparency over data sharing practices, enables data reuse and supports regulatory requirements. The experimentation shows that the platform can be scaled for a large number of users.

Keywords Blockchain · Data sharing · Distributed ledgers · Data reuse · Dataset tracking · GDPR

1 Introduction

The scientific community recognizes the importance of data sharing [1,2]. As a key element of scientific research, data sharing allows the reproduction of scientific results [1,3], helps prevent data fabrication and falsification [4]. Currently, data sharing is often required by funding bodies, publishers [1,3] and several EU and US funding initiatives [5–8].

Although the benefits of data sharing and reuse are widely acknowledged, evidence of data sharing practices is limited. Currently, a majority of researchers share their data only directly, that is, from person to person (e.g. by email), and mostly with collaborators, which suggests that trust is an important factor in sharing data. One-third of researchers
do not share their data at all and public data sharing occurs through the appendix of research articles, stand-alone publications in data journals, repositories, personal websites, and specific websites [13]. These ways of data sharing make it hard to find data even if it is shared publicly.

Several factors explain the large gap between the agreed importance of sharing data and today’s actual practices [1,3,9]. (i) There is no commonly accepted definition of what data sharing exactly means and which data should be shared. (ii) Researchers lack the expertise, training, infrastructure, and resources, to share their data. (iii) Researchers rarely receive credit for sharing data, partly because how to cite and attribute data is not commonly defined yet. (iv) Privacy concerns, control over what happens to the data, and ethical issues prevent researchers from sharing their data.

One way to deal with these concerns is to use data licensing [10]. Licenses clearly state the data reuse conditions, thereby creating legal clarity for the researchers who reuse the data. Yet, due to a lack of awareness over data licensing, a significant proportion of shared data is not licensed. Even when data are licensed, restrictive licenses are most often chosen due to the difficulty of tracking how data are used once they have been shared [1]. The ability to track data reuse is crucial. When the data contains information of data subjects. In this context, the General Data Protection Regulation (GDPR) [11] is an important regulation describing the right to access for individuals. Data subjects are given access rights to access information about their individual data, such as who is using the data and for which purpose. Two other elements of the GDPR are of high significance in the context of data sharing and reuse: the individual’s rights to erasure (involving the deletion of individual records) and to rectification (involving the update of individual records). In general, the lack of traceability and clarity on the secondary use of data is driving the unwillingness to share datasets. Centralized platforms have already been proposed with the aim of facilitating data sharing (Google data [12], kaagle [13], GitHub [14]), however, these platforms are controlled by one single authority and do not properly address the traceability and clarity of secondary data use. A blockchain-based solution can overcome many of the challenges of data reuse. Blockchain technology provides a methodology for constructing data structures that track data transaction history in an immutable manner. A blockchain-based data structure is designed to be append only, thus, it traces the data in a verifiable, tamper-proof immutable history of transactions. Defining a blockchain based model for data reuse provides a guarantee of a transparent data exchange process. Blockchain enabled data sharing has been proposed for different domains, such as health data exchange (i.e. [15][18]), the internet of things [19], energy [20,21] and many other use cases. In these scenarios, blockchains are prototypes resulting in platform designs that are not generic enough to be used for data-sharing and reuse. Some works focusing on enabling generic data sharing platforms with blockchain technology have been described in [22,25]. These works cover many relevant issues to the data sharing process, such as data subject inclusion [22] and enforcement of regulations via smart contracts [23], however, fall short in designing the data sharing process for a transparent data sharing and reuse: including how records are explored and accessed, how they are reused in relation to user consent and how to revoke or update the records based on the requests from data subjects.

To overcome these limitations, we present LUCE, a blockchain-based platform that addresses key challenges in data sharing and reuse. LUCE is a web-based accessible solution for data subjects, data providers, data requesters as well as data authorities. The prototyped model facilitates compliance with data licensing terms and helps researchers to comply with the rights to access, rectification, and erasure aspects of the GDPR. LUCE facilitates data sharing and the verification of what happens thereafter. The contributions of this work are twofold: the definition of a generic decentralized data sharing model for accountability and compliance with data re-use and the definitions of dynamic consent and legal compliance mechanisms. We implement and test the scalability of the platform in a real-time environment where a growing number of users access and reuse different datasets. We conclude that the solution provides transparency over data sharing practices, enables data reuse and supports regulatory requirements. The experimentation shows that the platform can be scaled for a large number of users. The remainder of this paper is organized as follows. In section 2 we introduce the building blocks of our solution, namely blockchain technology, smart contracts, and the EU General Data Protection Regulation. In section 3 we describes the LUCE architecture and the interaction protocol. Thereafter we present the implementation of LUCE in section 4 and the evaluation of the platform in section 5. Further, in section 6 we follow with the current state of the art for data sharing solutions. We conclude this paper in section 7 with a discussion on current and future work.

2 Background

In this section, we briefly discuss background work on blockchain, smart contracts, and GDPR.

2.1 Blockchain

Blockchain can be thought of as an append-only and distributed transactional database [25,27]. A blockchain network has three main components: (i) the blockchain itself, that is, the file containing the records of all transactions. (ii)
the peer-to-peer (P2P) network where the participants interact via the blockchain protocol to transact and update the blockchain. (iii) the consensus mechanism. The records of all transactions in the blockchain network are stored in the blockchain. Transactions are organized in blocks that are linked to each other in chronological order. Each block contains the records of transactions and the identifier of the block preceding it in the chain [26]. The records of transactions that are stored in the blockchain are auditable and verifiable but cannot be modified once they have been added. This is achieved through the use of cryptographic hashing [28]. A hash function maps the data of variable length to a fixed-length digest. Any change to the input data results in an unpredictable change in the hash. In a blockchain model, every transaction is added to a data structure (namely called the chain) which includes a hash of the previous transactions. If any of the transactions is changed at any point, the subsequent hash would no longer be valid. Given that there is no single central authority managing the blockchain, a consensus mechanism is necessary to formally encode rules regarding how transactions are validated and how they are added to the ledger. These transactions are verified and validated by the specific nodes, called miners who append new blocks in the blockchain. Mining can be an activity open to all or controlled to specific nodes. The latest model is also known as a permissioned blockchain model. Permissioned blockchains have seen some adoption in several business domains however they have shown low adoption from individuals [29]. By design, they limit decentralisation to a consortium of partners and that is why we focus on open blockchain models (also known as permissionless). In open blockchains, miners typically add blocks by solving a computationally expensive cryptographic puzzle. Whoever solves it first, receives a monetary reward. The miner who creates a new candidate block broadcast it to the network. Using the consensus protocol, the nodes verify and validate the candidate block which is afterwards added to the blockchain. Cryptographic puzzles have been proven to be expensive for the scalability and for the energy consumption of blockchain systems [30], therefore alternative protocols to cryptographic puzzles (also known as proof of work models) are also available (such as proof-of-stake models [31]).

**Smart Contract:** A smart contract is a computer code that runs on the top of a blockchain network and that contains rules - rights and obligations - defining the interaction between the parties to the smart contract [32]. When all parties meet pre-defined conditions, the smart contract automatically enforces the agreement between them. Transactions with the smart contracts are recorded within the blockchain on the top of which it is running. Smart contracts allow for several parties who do not especially trust each other to transact with each other without a trusted third party. Smart contracts are used for simple economic transactions but also for other, more complex, purposes, such as registering ownership, intellectual rights, or data-sharing practices.

### 2.2 Data Sharing Policies

The EU General Data Protection Regulation (GDPR) came into effect in May 2018 to extend the requirements of organizations about collecting and processing personal data of EU residents [11]. The GDPR applies to EU and non-EU organizations that collect and process EU residents’ data. The main actors involved in the GDPR are: the data subject - an identified or identifiable natural person whose data are contained in the dataset, Data subjects can authorize a data controller to access his or her personal data, with the possibility to transfer these to a data processor in charge of processing these data. Another key actor is the Supervisory Authority which is a controlling body. Each EU member state has its own National Supervisory Authority. In the context of scientific data sharing and reuse, individuals whose data records are collected by a researcher, are the data subjects and the researcher can be seen as a data controller. In case the researcher shares the dataset constituted by all the subjects’ data records and that another researcher reuses it, the second researcher can be seen as a data processor. The Supervisory Authority remains unchanged. Under GDPR data subjects have the right to obtain from the controller confirmation as to whether or not personal data concerning him or her are being processed [11]. If that is the case, the controller also has to give the data subject access to her or his personal data, as well as some additional information such as the purpose of the processing and to whom their data has been transferred. The right to rectification states that the data subject has the right to have his or her personal data rectified by the controller, in case they are inaccurate. Finally, the data subject can also exercise a right to erasure where the controller is obliged to erase the subject’s personal data. The fourth element of crucial importance, in the case of data sharing and reuse, is that of consent. Indeed, the GDPR requires that the data subjects freely give their informed consent to the collecting and processing of their data. Under this article, if personal data are shared and reused, it is vital that the purpose for which they are reused is compatible with the original purpose for which they were collected.

As the primary data collector, the data provider remains responsible for the use of data and the main contact point for the data subjects.

In the US, there is no specific federal data protection law, instead, it can be found in various legislations like Federal Trade Commission (FTC) Act [33], Health Insurance Portability and Accountability Act (HIPAA) [34] and California Consumer Privacy Act 2018 (CCPA) [35]. The key privacy rights in these laws are the right to access the data or copies of these data, right to deletion of data, right to error rectifications, right to object to processing, right to data portability, right to withdraw consent, right to object to marketing and right to complain to relevant data authorities. Similarly, in UK, Data Protection Act 2018 [36] and in Australia, Privacy Act 1988 [37] promote the protection of data and privacy.
of individuals. In UAE, Dubai International Financial Centre (DIFC) has adopted the DIFC Law no. 1 of 2007 [38] and Abu Dhabi Global Markets (ADGM) [39] adopted the Data Protection regulations 2015 which are in line with EU GDPR regulations.

3 LUCE Data Sharing Architecture

We define a model that i) manages licensing terms attached to a dataset, ii) dynamically manages user consent and reuse purpose of the dataset and, iii) enables compliance with the rights to access, rectification, and erasure of GDPR. Figure 1 shows the main actors involved in data sharing and reuse, as well as how they interact. There are four main actors involved: the data provider, e.g. a researcher willing to share a dataset, the data requester, e.g. a researcher requesting to reuse a dataset, the supervisory authority, e.g. a national public authority in charge of monitoring the adherence to data regulations and the data subjects, e.g. any individual whose data are being collected, held, or processed. Data subjects trigger interactions via data providers or supervisory authorities, either by exercising their rights to access, erasure, modification, or by lodging a complaint via the supervisory authority.

The interactions between data providers and data requesters are managed by smart contract agreements. These store information that is required to support data subjects to manage their data-sharing preferences and to exercise their GDPR rights. All the interactions in relation to data sharing are managed within the smart contract agreements and stored in an open blockchain network. The data themselves are not stored in the blockchain and token-based mechanisms are used to control the access to off-chain data. Each data provider, data requester, and supervisory authority is a node on a peer-to-peer blockchain network. LUCE checks that data requesters comply with the license of a dataset. For this purpose, data requesters are required to accept the licensing terms and to periodically renew their commitments within the smart contract. LUCE focuses on three main data sharing steps: sharing data, reusing a data and complying with the GDPR’s rights to access, rectification and erasure. We make some assumptions regarding these steps:

1. Datasets are shared as a whole.
2. Dataset integration is not taken into account, meaning that we currently do not cover cases in which several datasets are combined when reused.
3. The license under which the dataset is shared is compatible with the consent of the data subjects.
4. Personal data are de-identified and only the data provider can map data subjects to their records. The data subject and data provider can coincide when the individual is sharing their own data.

Assumptions 3 and 4 are only relevant in case the shared dataset contains data records of data subjects. All assumptions will be removed in future developments of LUCE. For example, in [15] we already proposed an individual consent model for data subjects. In this model, data subjects can explicit their own consent over data sharing. In this way
datasets are constructed based on the requirements of data requesters and the consent of data subjects. Fig. 2 shows an overview of the architecture of LUCE. In this diagram, the four primary actors – data provider, data requester, supervisory authority, and data subject interact with the components of the platform. In the following we explain the main components of Fig. 2 and their interactions.

3.1 Actors

In LUCE, the owner of the data never changes. Data requesters subscribe to use a specific dataset and are encouraged to unsubscribe from it once its usage is no longer required. This minimizes the cognitive burden to the data requesters in that they only need to keep track of the licensing terms for the datasets currently in active use. It also gives data providers and the supervisory authority valuable chronological information on the time-frame and purpose for which a given dataset was used. If a dataset is to be re-used again at a later time or for a new purpose, access is requested again stating the new purpose. This keeps the purpose of use specific for any given access period.

A web interface facilitates interactions among users. The web interface offers different functionalities depending on the role of the user. The identity of the users is tracked via a user management system and each user identifies themselves to the system with their unique blockchain address. A Data Provider can 1) Register with the system 2) Publish data 3) Update Data, and 4) Remove data. Similarly, a Data Requester is able to 1) Register 2) Search for data 3) Request access to the data, and 4) Confirm license compliance periodically to maintain their access rights. The supervisory authority can access a mapping between registered users and blockchain addresses and access a list of all published datasets (past and present) and their corresponding smart contracts. Finally, the Data Subjects are able to Register as a data subject and View which datasets have their information and for which purpose it is used.

3.2 Smart Contract Agreements

One smart contract is generated per published dataset. The smart contract: i) Keeps track of the access rights to the corresponding dataset. No data can be accessed without obtaining permission via the interaction with the smart contract. Data requesters accept the licensing terms and enter a binding agreement with data providers. ii) Acts as proof of provenance. The hash of the dataset file is included in the contract so if competing claims of ownership arise, possession of the data at a specific point in time can be proven unambiguously by the data provider who published the contract.

---

1 The following video link shows different LUCE actors interacting with the LUCE platform via the LUCE web interface

https://tinyurl.com/y22s9sq9
to the blockchain. This provides a basic layer of protection against data theft and intellectual property infringements.
iii) Creates an immutable history of who has had access to a given dataset, for how long, and for what purpose. This information is of interest to data subject who might exercise their right to information on how their data is being used. It is also used by the supervisory authority to audit whether actors comply with the licensing terms.

3.3 Maintaining the state in LUCE

Several components were also introduced to maintain a local overview of the state of the platform. The secure Data Storage component (see Fig.2) is responsible for securely accessing the shared datasets. This component ensures that valid tokens, issued within LUCE are used to access data. A user registry contract keeps track of which blockchain addresses have been registered with the system. Only these addresses are allowed to interact with the dataset contracts. This way we can ensure that there exists a mapping between addresses and the real identity of platform users. All sensitive information pertinent to the users (data requester and/or data providers) is stored within LuceDB. The smart contracts finally act as the gatekeepers to a dataset and ensure that access is only granted to data requesters who commit to adhere to the licensing terms. The smart contracts also contain a hash of the corresponding dataset, as well as metadata information, license type, permitted purposes, and a mapping of users with active access rights together with the purpose for accessing the data. The smart contracts hold the ultimate reference of truth and can be publicly inspected if desired. LuceDB stores a synchronized version of the same truth which can be regarded as a cache. It allows for a seamless access to information via the interface without requiring to query all contracts on the blockchain about their current state every time that a user makes a request.

3.4 Sharing a dataset

The first step in sharing data is a data provider sharing a dataset. Data are described and published as meta-data in the Metadata Repository, an online shared directory where data requesters, can search for shared datasets. The shared meta-data includes the description of the dataset and the license attached to it. Uploading this information also creates the associated smart contract and the necessary information associated with the dataset. This sequence of interactions is presented in the diagram displayed in Fig 3.

3.5 Reuse a dataset

The second step in the process is to support data re-use. In this case a data requester wants to search, find and access a shared dataset. This happens in two steps. The first is to access the dataset and the second is the monitoring of compliance with the licensing terms of the dataset by the data requester.

3.5.1 Accessing the dataset

The first step in reuse a dataset is accessing it. Fig.4 illustrates the data access protocol to be used by data requesters. In particular, the steps below describe the process in detail:

1. **Query** Data requesters, who seek to reuse a dataset in the context of their own research, query the Metadata repository. For each dataset matching the query, the Metadata repository provides a list containing: the dataset identifier, its meta-data, the type of license attached to it and the smart-contract address. Based on this list, the
data requester decides which database to reuse. As it has been discussed in [40], more complex negotiation protocols on the data can be modelled at this stage.

2. **Request** A request for accessing the dataset as well as the purpose of data reuse is sent to the smart contract by the data requester. In case the dataset contains data related to data subjects, the purpose indicated by the data requester is checked against the initial purpose these data were collected for. In case both purposes are compatible, the data provider grants access to the data requester. The compatibility of the data subject’s consent and data requester’s purpose is checked through a smart contract [15]. The specification of the purpose of reuse is also required in cases where the dataset does not contain data collected from data subjects.

3. **Accept** The data requester is granted access to a dataset only after accepting the licensing terms. Thereafter, the data requester must provide information to the smart contract regarding the continued use of the dataset.

4. **Download token** After the data requester accepts the licensing terms and the check on the purpose of use, the smart contract provides the data requester with a verification token and a link to the repository where the dataset is stored.

### 3.5.2 Monitoring compliance with the dataset’s license

The second step in reuse a dataset is the monitoring of compliance with its license by the data requester. This is illustrated in Fig. 5. The following interaction is used to monitor compliance with the data:

1. **Access token:** To access a given dataset, the data requester receives an access token. The access token is generated by the smart contract after the data requester accepts the licensing terms and describes the purpose of use for the data. The given token associates the token ID with the data requester. We extended the ERC-721 token [41] to track the ownership of the data. This implies that data requesters can use tokens for three purposes: accessing the data, renewing access time to the data, and deleting their access to the data. This work is out of the scope of this paper and will be thoroughly presented in follow-up work.

2. **Report compliance** Once a data requester obtains the access token, they can access the dataset. Two models of monitoring compliance are possible with LUCE: 1) creating an executable within a cloud-based model that monitors the modifications made on the dataset. This module continuously checks whether the actions performed by the data requester on the data comply with the licensing terms and records these actions in a file that constitutes a log of events, or, 2) Supporting a publish-subscribe model where users show/declare that they comply with the rules by periodically sending transactions to the contract. In this way, we are not fully checking license agreements, but we have continuous checkpoints and the public commitment of data requesters towards licensing terms. This model allows data requesters to download the dataset and requests
frequent renewal of the token for continued use. The first option implies that data reuse and analysis is possible only on provided software. The second model implies a more flexible data reuse as long as the token is renewed. LUCE requests the renewal of the token for continued use, independently from which compliance model is used.

3. **Renew token** The access token is used by the data requester to obtain access to the dataset. The token is only valid during a period of duration T. When that period ends, the data requesters renew their access token by requesting the renewal to the smart contract. The smart contract verifies whether the data requester has complied with the licensing terms during the last period. If so, the contract renews the access token to the data requester. However, in case the licensing terms were not complied with, the access token is not renewed, meaning that the data requester cannot access the dataset anymore.

Independently from the chosen monitoring model, a data provider can check whether data requesters have complied with the license terms attached to a shared dataset by accessing the events in the smart contract associated with the dataset.


3.6 GDPR compliance

The third data-sharing step covers the cases in which a data subject exercises their right to access, erasure, or rectification. Fig. 6 shows how we model the interactions with regards to the rights to rectification and erasure. In particular, the interaction steps can be described as follows:

1. Request rectify/erase: In concordance with the GDPR, the data subject has the right to request information from the data provider as to how their data is used, by whom, and request changes or deletion of their records. Data providers are the only ones that hold a mapping between a data subject and the dataset in which their records are included. At any time a data requester can query the smart contract to identify who is reusing the data and for what purpose and provide this information to data subjects. In case the data subject requests the data provider to either rectify or erase their data from the dataset, it must be done not only in the dataset of the data provider but also in the copies of the dataset that data requesters are using.

2. Data updates: consists of two actions
   (a) Rectify/erase: A data provider must anonymize the data before sharing them for further reuse. The mapping of data subjects to anonymized IDs is carefully stored by data providers (this information is not shared with the LUCE platform). After the request, a data provider uses the anonymized ID of the data subject to modify or erase the data. New data requesters will only access the updated records.
   (b) Update data: The smart contract contains all the identifiers of data requesters who are re-using the dataset. Using the anonymized identifier of the data subject, the erasure or modification of specific records is required for all the data requested who are re-using the data. For this, the smart contract generates an event for all the data requesters who are currently using the data.

3. Confirm update: All the data requesters who are using a non-updated copy of the dataset must erase or modify the data records as requested. The change is confirmed via a confirm update message sent to the smart contract.

4. Renew token: The data providers must renew their token after an update. This operation ensures that the data requesters who do not comply with the request are revoked access to the data.

5. Confirm rectify/erase: The data provider queries the smart contract to collect the information about the data requesters who are reusing the data and the state of the update. All the information is sent to the data subject. The confirmation of a rectify/erase has a delay which is dependent on the frequency of token renewal. From a data subject request of a rectify/erase the data subject waits at maximum the time of a token renewal. Such parameter needs to be carefully decided to ensure a good equilibrium between the expense that occurred by performing frequent transactions in the system and the accuracy of information to be provided to data subjects and supervisory authorities. In our current implementation, the frequency of token renewal is set to 2 weeks.

6. Complaint: Should the request of a data subject not have a satisfactory outcome, a complaint to the supervisory authority can be made.

7. Monitor compliance: LUCE enables the supervisory authority to check for GDPR compliance. The supervisory authority can monitor all the transactions of the blockchain platform. With the information provided by the data subject, it can verify the actions of data requesters in the corresponding smart contract. They also might collect evidence documents as well as check the behavior of a data requester towards multiple other datasets or data subjects.

8. Update: After monitoring compliance, the supervisory authority notifies the data subject about the outcome.

4 Implementation

In this section, we will provide the details regarding the implementation and the experimental setup of LUCE and a proof of concept web portal to access and share data on the LUCE platform. Moreover, we will discuss the LuceVM – a virtual machine implementation of LUCE and LuceDocker – a dockerized version of LUCE to illustrate the usage of the LUCE platform.

4.1 Implementation and Experimental setup

We implement LUCE on top of Ethereum [42] blockchain, an open-source and public blockchain-based distributed computing platform for building decentralized applications. Choosing the Ethereum platform implies that LUCE’s blockchain can be defined as a public platform, and anyone can inspect transactions as well as add or verify new ones.
While it is technically possible for users to interface directly with Ethereum, LUCE provides a user-friendly web-based solution. For the development of the web platform, we use the python Django framework [43], and integrate it to Ethereum via web3 [44] and py-solc. The platform includes a web interface as well as an application programming interface that allows for programmatic access to the system. Thus, LUCE users can interact with the platform via the LUCE Data Exchange web interface or programatically via a set of functionally equivalent APIs. The back-end encapsulates a Web Server (WSGI) and the application code which together provide the business logic. The database (LuceDB) handles user account management and acts as a cache for metadata to feed the web interface.

To run our experiments, we run LuceVM virtual machine on a 64 bit Ubuntu 16.04 LTS (Xenial Xerus) Linux operating system. The virtual machine is equipped with 4096 MB RAM. The virtual machine is managed by Vagrant [45] to further abstract away the layer of VM configuration for the end-users. Moreover, the VM is preconfigured with the blockchain development environment, libraries, and connected servers. Our LUCE platform implementation is available as open-source.

For fast deployment of LUCE, we use Docker [46] to bundle the LUCE software. The dockerized image of LUCE is deployed on a server hosted by the Institute of data science, Maastricht University. With this web-hosted version, a user can register with LUCE and share the dataset on the platform.

4.1.1 LUCE Smart Contract

As previously described, smart contracts [32] capture the interactions between data providers and requesters. Each data provider has a smart contract that is used to publish the dataset on the platform with preset conditions. Fig. 7 shows the code snippet of our smart contract used in the LUCE platform. When publishing a dataset, a data provider sets within the smart contract the data sharing conditions on the LUCE platform (publishData). In doing so, a description and a persistent URL to the data is provided by the data provider. Data requesters can use a shared dataset by calling the addDataRequester function. To call this function, they would have first to read the license terms with the getLicence function. For continued access to data, tokens must be renewed via the renewToken function. If data requesters comply with the license, then the token will always be renewed, otherwise the token is revoked and so is the access to the data. The interactions with the smart contract generate transactions that are added to the blockchain. It takes some time to add a transaction in the Ethereum blockchain that typically ranges between ~10-20 seconds. We test LUCE by simulating

---

[2] https://github.com/vjaiman/LUCE
interactions within the platform. All our simulated experiments were repeated 4 times to draw an overall average. We set up our experiments on private, and a public blockchain testnet.

4.1.2 Private Testnet

For testing purposes, Ethereum enables users to set up private blockchains that are separated by the Ethereum mainnet. This solution is used to test the blockchain environment before exposing it to the Ethereum mainnet where real costs occur (in the form of ether, the Ethereum currency). We make use of geth [47] to connect our node to the testnet. We conduct the experiments on a private blockchain testnet by creating 6000 accounts on it. Each account has to be unlocked before initiating any transaction and having an associated address to it by which it can be identified. Moreover, each account can hold ether to bear the transaction cost. In a private blockchain, ether is received for free.

4.1.3 Public Testnet

We also tested LUCE on a public testnet. Public testnets are very similar to the Ethereum mainnet network in terms of operation (i.e. transaction times are of a dynamic length and very similar to the mainnet. This is not the case with a private testnet). The difference is that while Ether must be acquired to operate on the main Ethereum network, in public testnets Ether remains free. We test the LUCE on the public Rinkeby testnet [48]. We make use of geth [47] to connect our node to the Rinkeby testnet. We conduct our experiments by creating 2000 accounts on the public network. The coinbase account can be funded through https://faucet.rinkeby.io for 3 ether by publishing the coinbase address on Twitter or Facebook. All the pending/completed transactions are available at Rinkeby Etherscan explorer [49]. The coinbase address is “0x9f913ef90c695ae1529e6cf5f1a5d407fe1a4178” and all the completed/pending transactions can be viewed by coinbase address on Rinkeby Etherscan explorer [49].

5 Evaluation

We evaluate the effectiveness of our solution in terms of scalability and cost. We compare the cost with a baseline smart contract performing minimal operations i.e. to set the input at a given address. The code of the baseline smart contract is publicly available [11]. Moreover, we compare it with consent-based LUCE [15], a more complex model where a data requester gets access to the data only if the purpose of use of a data requester and the consent use of a data provider match. The consent-based model adds to the licensing terms with more detailed information on the purposes under which data reuse is allowed (This model was presented in [15]). Our evaluation aims to answer the following questions:

1. How well does the data sharing model of LUCE scale with an increasing number of data requesters requesting the same dataset?
2. How the data sharing model scales with an increasing number of data requesters requesting different datasets?
3. How cost-effective are the operations in the LUCE platform?

The first question analyses how well LUCE scales with an increasing traffic on a single smart contract and the respective dataset. The second question looks at how well LUCE scales with increasing traffic in general and the third questions analyses the transaction costs introduced by user interactions.

Fig. 8 shows the execution time in LUCE with an increasing number of data requesters (5-100) who request access to the same dataset. We calculate the execution time of mined transactions. In general, mining a transaction takes between 10-20 seconds and can be longer depending on the network congestion and the gas price. In this experiment, after submitting a transaction, we wait for the transaction receipt from the miners. Afterwards, we calculate the total execution time taken by LUCE and Ethereum private testnet. We observe that, as the number of data requesters increases, the total execution time is almost the same in LUCE compared to the baseline. We can observe a little difference once data requesters increase to 80-100 which might be explained by the list scrolling operations of the smart contract to identify that transactions initiated by data requesters went through the required interaction protocol. Overall the results show that the smart contract model of LUCE introduces low computational cost, compared to the transaction cost introduced by the architecture of the Ethereum blockchain. Compared to the Consent-based LUCE [15], the second model takes a longer time to execute as the smart contract performs several checks to determine that the purpose of use matches a given consent. This second model suggests that consent-based access to the data may further increase the latency of transactions, especially when multiple concurrent data access queries are made on the same dataset. Fig. 9 shows the execution time with an increasing number of data requesters accessing data in the LUCE platform. In this experiment, we submit transactions from an increasing number of data requesters (100-5000). Contrary
to the previous experiment in Fig. 8 we do not wait for the transactions to be mined by Ethereum and calculate the execution time for submitting the transactions by the LUCE platform. The objective was to see the actual execution cost of functions defined in the smart contract. In this experiment, LUCE smart contract takes 178.09s whereas baseline smart contract takes 134.08s. We can observe that the smart contracts of LUCE impose low operational cost for data sharing compared to the baseline smart contract. In fact, consent-based LUCE takes 507.55s due to the submission of consent before getting access to the data. In Fig. 10 we vary the mining threads to see the impact of miners on the incoming data requests. For 100 data requesters, with 16 threads, it takes an optimum time of 1264s whereas with 1 thread it takes a maximum time of 2066s. We observe that as the number of data requesters increases the mining
Figure 10: Impact of varied mining threads on the private testnet.

Figure 11: Latency of data requesters in LUCE on private testnet and public rinkeby testnet.

time in each thread group also increases. Moreover, mining time drops from 1 to 16, and afterwards due to thrashing and network congestion, mining time increases again for 32 threads. We conclude that under these settings, LUCE works efficiently with more miners when the number of data requesters increases. For public testnet experiments, we submit transactions on Rinkeby testnet by increasing the data requesters from 100 to 2000 for the consent-based LUCE. The objective is to observe the execution time difference for LUCE compared to the private testnet. In Fig. 11 we observe that consent-based LUCE takes 51.67s for 1000 data requesters compared to the 58.15s for private testnet. Similarly, the public testnet takes 130.25s for executing the 2000 data requesters compared to 98.84s for the private testnet. We observe that, when the number of transactions increases, the transactions create a congestion of the public
Table 1: Base cost for the core functions of LUCE.

| Actions       | Transaction cost (in gas) | Execution cost (in gas) | Ether cost (in ETH) | Cost* (in $) |
|---------------|---------------------------|-------------------------|---------------------|--------------|
| Contract Deployment | 1339598             | 964030                  | 0.0428671           | $79.28       |
| publishData  | 79652                 | 56460                   | 0.002548            | $4.71        |
| setLicense   | 24201                 | 2737                    | 0.0007744           | $1.43        |
| addDataRequester | 105842              | 84186                   | 0.003386            | $6.26        |
| updateData   | 47756                 | 24884                   | 0.001528            | $2.82        |
| renewToken   | 16149                 | 9685                    | 0.0005268           | $0.97        |
| getLink      | 24780                 | 3316                    | 0.000793            | $1.46        |
| getLicense   | 22384                 | 1112                    | 0.000716            | $1.32        |

*= Ether conversion with present date price (Average = 32 Gwei & 1 ETH = $1849.44)

5.1 Cost analysis and feasibility

In Ethereum, every transaction in the smart contract consumes gas (a measure of the computational effort required to perform an operation). Gas consumption varies based on the complexity of the functions defined in the smart contract. We perform a cost analysis of the functions defined in the LUCE smart contract. Several parameters are considered:

1. The total amount of gas spent during LUCE contract deployment.
2. The amount of gas consumed when publishing the dataset and setting the license.
3. The amount of gas consumed when updating the data or renewing the token.
4. The amount of gas spent while getting the license and the token.

Table [1] shows the total gas consumption to execute the contract is 1339598. We considered the average gas price of 32 Gwei according to the current date of ETH gas station [40]. Therefore, the relevant cost of contract deployment is 0.0428671 ETH with a corresponding price of $79.28 (1 ETH ≈ $1849.44). It is noted that the transaction cost includes the default cost of 21000 gas which is occurred by Ethereum for performing a transaction on it and therefore the actual function cost is mentioned as Execution cost in the table [1]. We calculated the cost of data sharing by the accumulative cost of contract deployment, the cost for publishing the dataset, and the cost of setting the license. Some additional costs are involved when data is updated or the token is renewed. We observe that publishing a dataset and adding a data requester involves more gas consumption compared to the other operations. This is due to the involved parameters, such as setting the description of the dataset or setting up the link to the dataset location. Similarly, adding a data requester requires mapping a new user against the existing ones, therefore, such operation requires more computational efforts and gas consumption.

6 Related Work

The technological advancement of the last few decades has enabled collaborations and markets for sharing data. In the current practices, personal data from different data sources are collected and combined (often without explicit user consent) with the purpose of aiding data-driven decisions. This way of sharing and re-using personal data has brought important legal and social implications. In response, several decentralized solutions to data sharing have been proposed (for example [51–55]). The overall aim of these solution is similar to LUCE, decentralize the control over data to bring more control to the users and support transparency over data exchange. In particular, in [51] and in [54] the authors define permissioned blockchain solutions that enables respectively sharing of personal health records in emergency

---

*16/07/2021*
situations and cross-domain image sharing. In [51], the authors focus on securing and preserving transactions from being tampered. This is done by means of smart contracts in which patients can enable access for specific doctors. While in [54] the authors have a theoretical model defining how concepts such as the definitions of the study, the source and access mechanisms work towards data sharing. In this work we seek not only to enable data-sharing but also monitor their use. Moreover, while these works, and many others that could have been mentioned here (such as [56–58]), focus on solving some aspects of data sharing and specializing the solution for the given scenarios, they do not fully address data sharing as a more generic process, requiring data to be shared but also updated maintained and eventually deleted. Some works have specifically focused on data provenance [23, 24] which describes the history of data, such as where they originate from, their owner, the changes made to the data, as well as who made them [23, 24]. DataProv [23] and ProvChain [24] are both blockchain-based solutions for data provenance accountability. Once collected, data provenance records are published to the blockchain network, verified, and eventually added to the blockchain. In other words, DataProv and ProvChain provide an immutable and secured ledger for data provenance records. Both solutions deal with data stored in the cloud. Even though these solutions seem to provide a way for researchers to comply with the GDPR’s right to access, how to comply with the rights to rectification and erasure is not addressed. Furthermore, compliance with licensing terms of shared data is not taken into account. Neisse et al. [22] propose a blockchain-based solution for data provenance accountability. Their solution aims at empowering data subjects by enabling them to track who has accessed their data and whether these were used accordingly to their consent. It also aims at helping data controllers prove that they have received that consent, the proof of which is a smart contract to which both the data controller and subject are parties. The smart contract also encodes the policies for data access, usage, and transfer, as well as data provenance information. This solution focuses on the relationship between data subjects and data controllers. In contrast, the goal of LUCE is to define the generic interaction protocols between data providers (including data subjects) and data requesters. The advantage of the LUCE approach is that data can be shared and reused by data requesters, while data providers and data subjects maintain control over the access and the type of reuse of data. The Ocean Protocol [25] is an industry-wide initiative to implement marketplaces for data sharing. This blockchain-based solution aims at facilitating the sharing of datasets (as well as algorithms and services, such as storage and processing) in a transparent, traceable, and trustworthy way, with the data providers keeping control over their datasets. However, the Ocean Protocol mainly targets companies that collect data. Indeed, data providers can choose to be monetarily rewarded for sharing their data and there is no considerations made for the data subjects. Even though the Ocean Protocol allows for traceability regarding what happens with datasets, it does not allow for the recording of the purpose for which the data have been used. Thus with OCEAN is not possible to guarantee compliance with the GDPR’s right to access, rectification and erasure. Finally, in [53], a permissioned blockchain is adopted to support data sharing a GDPR compliant way. Compliance to GDPR is achieved by means of user consent. Consent can be revoked however the deletion of records is not explicitly addressed. As in OCEAN, the authors define a compensation is required as a way to incentivize data-sharing. The solution differs to LUCE in several aspects, firstly the use of permissioned blockchain means that the decentralisation is limited by design in the platform. LUCE is envisioned as an open platform, engaging with end-users as data providers and data requesters. The way LUCE monitors data re-use is to include structured descriptions of purpose of use, as described in detail in [15]. Moreover, purpose of use and consent are dynamically checked and enforced within the smart contract.

7 Conclusions and Future Work

In this paper, we presented LUCE, a blockchain-based solution for automatic data management of licensing terms and accountability in a GDPR compliant manner. LUCE creates new opportunities for data sharing and reuse and makes it easier for researchers to track what happens to the dataset once they have been shared. Moreover, LUCE enables the enforcement of licensing terms and provides a solution for complying with the right to access, rectification, and erasure of GDPR. We show that with an increasing demand for data re-use LUCE scales similarly to any Ethereum based network in private and in public configurations of the network. The cost involved in the platform are greater for data providers. In follow up work we experimented with different incentive schemes [55] and showed that the system can be calibrated to incentivise all the involved actors. Furthermore, this work acts as our base architecture model and we are continuously working on developing LUCE and its components:

**Consent compliance** In the future, we will further investigate how to comply with the consent given by data subjects. The assumption here is that the license matches the consent of data subjects, which is true in most research datasets, however, there is a need for personalization of the individual consent of the data subjects which is also in line with existing legal frameworks. We already proposed an extension of LUCE [15] which considers the consent codes developed by the Global Alliance for Genomic and Health [59, 60] to systematically record data usage conditions based on the data subject’s consent and purpose of use. This approach however is suitable for personal health data. We will
investigate how to generalize the approach by identifying generic consent codes that support data collected in different domains.

Data integration Another point that LUCE will address in the future is merging datasets for further reuse. It occurs in practice and is even one of the goals of data sharing and reuse. To this extent, we will seek the use of Digital Object Identifiers (DOI) in combination with a DOI suffix that tracks the merging of data to the original data sources.

Evaluation In this paper, we evaluate the technical feasibility and scalability of the LUCE platform. With LUCE, users can share data securely and transparently. Overcoming these two important barriers of data-sharing is expected to improve user engagement with data sharing practices and the benefits it brings at an individual and societal level. LUCE will need to be further evaluated on how well it helps users to perform data sharing and reuse. Therefore, behavioral evaluations are required to investigate to which extent the users of LUCE would be willing to make use of the platform and which features should be modified, removed, or added to ensure a broad user engagement.

Acknowledgment

This work was supported in part by the NWO Aspasia (Grant 91716421) and by the Maastricht York-Partnership Grant. The authors would like to thank Andine Haveange for her initial insights on the work which helped to define the LUCE architecture. They also would like to thank Prof. David Townend and Dr. Birgit Wouters for their insights on compliance with GDPR law and regulations.

References

[1] Leiden University’s Centre for Science and Technology Studies (CWTS) and Elsevier. Open Data - The researcher perspective. https://www.elsevier.com/about/open-science/research-data/open-data-report, 2017. accessed May 22, 2021, from Elsevier.

[2] Nelson Bryn. Empty archives. Nature, 461:160–163, 2009.

[3] C. L. Borgman. The conundrum of sharing data. Journal of the American Society for Information Science and Technology, 63:1059–1078, 2012.

[4] C. Tenopir, S. Allard, K. Douglass, A.M. Aydinoglu, L. Wu, E. Read, M. Manoff, and M. Frame. Data sharing by scientists: Practices and perceptions. PLoS ONE, 6, 2011.

[5] Elixir. Elixir. https://www.elixir-europe.org/, 2017. Accessed June 21, 2021.

[6] The European Commission. European Open Science Cloud. https://eosc-portal.eu, 2018. Accessed June 21, 2021.

[7] National Center for Advancing Translational Sciences (NCATS). Biomedical Data Translator. https://ncats.nih.gov/translator, 2018. Accessed June 21, 2021.

[8] National Institutes of Health (NIH). Data Commons Pilots. https://commonfund.nih.gov/commons, 2018. Accessed June 21, 2021.

[9] D. Stuart, G. Baynes, I. Hrynaszkiewics, K. Allin, D. Penny, M. Lucraft, and M. Astell. Research Data - Practical challenges for researchers in data sharing [White paper]. https://www.springernature.com/in/open-research/open-data/practical-challenges-white-paper, 2018. Accessed May 22, 2021, from Springer Nature.

[10] V. Stodden. The legal framework for reproducible scientific research: Licensing and copyright. Computing in Science & Engineering, 11(1):35–40, 2009.

[11] GDPR. Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation). https://tinyurl.com/y8kwtdzqu. Accessed June 22, 2021.

[12] Google dataset search. https://datasetsearch.research.google.com Accessed February 11, 2021.

[13] Kaggle. https://www.kaggle.com Accessed February 11, 2021.

[14] Github. https://github.com Accessed February 11, 2021.

[15] Vikas Jaiman and Vissara Urovi. A consent model for blockchain-based health data sharing platforms. IEEE Access, 8:143734–143745, 2020.
[16] Fadila Zerka, Visara Urovi, Akshayaa Vaidyanathan, Samir Barakat, Ralph TH Leijenaar, Sean Walsh, Hanif Gabrani-Juna, Benjamin Miraglio, Henry C Woodruff, Michel Dumontier, et al. Blockchain for privacy preserving and trustworthy distributed machine learning in multicentric medical imaging (c-distrim). IEEE Access, 8:183939–183951, 2020.

[17] Kai Fan, Shangyang Wang, Yanhui Ren, Hui Li, and Yintang Yang. Medblock: Efficient and secure medical data sharing via blockchain. Journal of medical systems, 42(8):1–11, 2018.

[18] Cornelius C Agbo, Qusay H Mahmoud, and J Mikael Eklund. Blockchain technology in healthcare: a systematic review. In Healthcare, volume 7, page 56. Multidisciplinary Digital Publishing Institute, 2019.

[19] Tiago M Fernández-Caramés and Paula Fraga-Lamas. A review on the use of blockchain for the internet of things. IEEE Access, 6:32979–33001, 2018.

[20] Merlinda Andoni, Valentin Robu, David Flynn, Simone Abram, Dale Geach, David Jenkins, Peter McCallum, and Andrew Peacock. Blockchain technology in the energy sector: A systematic review of challenges and opportunities. Renewable and Sustainable Energy Reviews, 100:143–174, 2019.

[21] Poppit Wongthongtham, Daniel Marrable, Bilal Abu-Salih, Xin Liu, and Greg Morrison. Blockchain-enabled peer-to-peer energy trading. Computers & Electrical Engineering, 94:107299, 2021.

[22] Ricardo Neisse, Gary Steri, and Igor Nai-Fovino. A blockchain-based approach for data accountability and provenance tracking. In Proceedings of the 12th International Conference on Availability, Reliability and Security, pages 1–10, 2017.

[23] Aravind Ramachandran, Dr Kantarcioglu, et al. Using blockchain and smart contracts for secure data provenance management. arXiv preprint arXiv:1709.10000, 2017.

[24] Xueping Liang, Sachin Shetty, Deepak Tosh, Charles Kamhoua, Kevin Kwiat, and Laurent Njilla. 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), pages 468–477. IEEE, 2017.

[25] Ocean. Ocean Protocol: A Decentralized Substrate for AI Data and Services Technical Whitepaper. https://oceanprotocol.com/tech-whitepaper.pdf. Accessed May 18, 2021.

[26] Satoshi Nakamoto. Bitcoin: A peer-to-peer electronic cash system, 2009.

[27] Shashank (edureka!). What is Blockchain Technology? How Blockchain Works. https://www.edureka.co/blog/how-blockchain-works/, 2018. Accessed June 26, 2021.

[28] Shai Halevi. Hash functions and their many uses in cryptography, August 2009.

[29] Christine V Helliar, Louise Crawford, Laura Rocca, Claudio Teodori, and Monica Veneziani. Permissionless and permissioned blockchain diffusion. International Journal of Information Management, 54:102136, 2020.

[30] Anamika Chauhan, Om Prakash Malviya, Madhav Verma, and Tejinder Singh Mor. Blockchain and scalability. In 2018 IEEE International Conference on Software Quality, Reliability and Security Companion (QRS-C), pages 122–128. IEEE, 2018.

[31] Fahad Saleh. Blockchain without waste: Proof-of-stake. Available at SSRN 3183935, 2020.

[32] Nick Szabo. Smart contracts: building blocks for digital markets. EXTROPY: The Journal of Transhumanist Thought, 16, 18, 1996.

[33] FTC. Federal trade commission act. https://www.ftc.gov/enforcement/statutes/federal-trade-commission-act, 2006. Accessed February 10, 2021.

[34] Hipaa. https://www.hhs.gov/hipaa/index.html. Accessed Dec 2, 2021.

[35] State of California Department of Justice. California consumer privacy act (ccpa). https://oag.ca.gov/privacy/ccpa, 2018. Accessed February 10, 2021.

[36] UKGovernment. Data protection act 2018. https://www.legislation.gov.uk/ukpga/2018/12/contents/enacted, 2018. Accessed February 10, 2021.

[37] AustralianGovernment. Privacy act 1988. https://www.legislation.gov.au/Details/C2014C00076, 1988. Accessed February 10, 2021.

[38] DFSA. Dubai international financial centre (difc). https://www.difc.ae/business/laws-regulations/, 2004. Accessed February 10, 2021.

[39] ADGM. Abu dhabi global markets (adgm). https://www.adgm.com/operating-in-adgm/office-of-data-protection/overview, 2015. Accessed February 10, 2021.
[40] Alevtina Dubovitskaya, Visara Urovi, Imanol Barba, Karl Aberer, and Michael Ignaz Schumacher. A multiagent system for dynamic data aggregation in medical research. *BioMed research international*, 2016, 2016.

[41] W. Entriken, D. Shirley, J. Evans, and N. Sachs. Erc-721, 2018.

[42] Gavin Wood. Ethereum: A secure decentralised generalised transaction ledger. http://gavwood.com/Paper.pdf, 2015.

[43] Django. https://docs.djangoproject.com/en/3.0/ Accessed June 27, 2018.

[44] Ethereum javascript api. https://web3js.readthedocs.io/en/v1.2.6/ Accessed June 27, 2021.

[45] Vagrant. https://www.vagrantup.com Accessed May 27, 2021.

[46] Docker. https://www.docker.com Accessed May 20, 2021.

[47] Go ethereum. https://geth.ethereum.org Accessed May 14, 2021.

[48] Rinkeby: Ethereum testnet. https://www.rinkeby.io Accessed May 14, 2021.

[49] Testnet rinkeby (eth) blockchain explorer. https://rinkeby.etherscan.io Accessed May 14, 2021.

[50] Eth gas station. https://ethgasstation.info/.

[51] Ahmed Raza Rajput, Qianmu Li, and Milad Taleby Ahvanooy. A blockchain-based secret-data sharing framework for personal health records in emergency condition. In *Healthcare*, volume 9, page 206. Multidisciplinary Digital Publishing Institute, 2021.

[52] Jiancheng Chi, Yu Li, Jing Huang, Jing Liu, Yingwei Jin, Chen Chen, and Tie Qiu. A secure and efficient data sharing scheme based on blockchain in industrial internet of things. *Journal of Network and Computer Applications*, 167:102710, 2020.

[53] Ajay Kumar Shrestha, Julita Vassileva, and Ralph Deters. A blockchain platform for user data sharing ensuring user control and incentives. *Frontiers in Blockchain*, 3:48, 2020.

[54] Vishal Patel. A framework for secure and decentralized sharing of medical imaging data via blockchain consensus. *Health informatics journal*, 25(4):1398–1411, 2019.

[55] Vikas Jaiman, Leonard Pernice, and Visara Urovi. User incentives for blockchain-based data sharing platforms, 2021.

[56] Asaph Azaria, Ariel Ekblaw, Thiago Vieira, and Andrew Lippman. Medrec: Using blockchain for medical data access and permission management. In *2016 2nd international conference on open and big data (OBD)*, pages 25–30. IEEE, 2016.

[57] Bingqing Shen, Jingzhi Guo, and Yilong Yang. Medchain: Efficient healthcare data sharing via blockchain. *Applied sciences*, 9(6):1207, 2019.

[58] Xueping Liang, Juan Zhao, Sachin Shetty, Jihong Liu, and Danyi Li. Integrating blockchain for data sharing and collaboration in mobile healthcare applications. In *2017 IEEE 28th annual international symposium on personal, indoor, and mobile radio communications (PIMRC)*, pages 1–5. IEEE, 2017.

[59] Stephanie O. M. Dyke, Anthony A. Philippakis, Jordi Rambla De Argila, Dina N. Paltoo, Erin S. Luetkemeier, Bartha M. Knoppers, Anthony J. Brookes, J. Dylan Spalding, Mark Thompson, Marco Roos, Kym M. Boycott, Michael Brudno, Matthew Hurles, Heidi L. Rehm, Andreas Matern, Marc Fiume, and Stephen T. Sherry. Consent codes: Upholding standard data use conditions. *PLOS Genetics*, 12(1):1–9, 01 2016.

[60] GA4GH. Global Alliance for Genomics and Health. https://www.ga4gh.org/, 2019. Accessed June 21, 2021.