# Access Rights Management in Decentralized Distributed Computing Systems*

Andrey Demichev[1], Alexander Kryukov[1], and Nikolay Prikhod'ko[2]

[1] Skobeltsyn Institute of Nuclear Physics
Lomonosov Moscow State University, Moscow, Russia
demichev@theory.sinp.msu.ru, kryukov@theory.sinp.msu.ru
[2] Independent Researcher, Velikiy Novgorod, Russia
niko2004x@mail.ru

**Abstract.** The paper presents a solution for decentralized management of data access rights in geographically distributed systems with users from different institutions. This implies possible lack of trust between the user groups. The solution is based on the distributed ledger technology (DLT) together with provenance metadata driven data management.

**Keywords:** distributed storage · access rights · provenance metadata · blockchain · Hyperledger Fabric.

## 1 Introduction

The main task that middleware of geographically distributed computer systems (DCS) solves is the integration of resources remote from each other and from users into a single pool. Shared DCS resources may include computing nodes, data storages, data themselves and application software. The middleware enable managing data files at remote storage resources, distributing computational tasks over data processing services, returning results to users, controlling access rights to resources, monitoring resources, accounting their use and performing a number of other actions.

In its most general form, the architecture of most DCSs has three basic layers (see, e.g. [1–4]): a user interface layer that provides access to the system of users and administrators; a layer of systemic centralized services that manage the DCS as a whole; a layer of sites with gateway servers providing access to local resources. Thus, although the storage and data processing resources of such a DSC are geographically distributed, they are combined into a single pool by using the infrastructure based on the centralized services. Examples of such centralized services are data management services, monitoring services, services

* This work was funded by the Russian Science Foundation (grant No. 18-11-00075).

for managing data access rights and automatic renewal of proxy certificates required for delegation of such rights, etc.

Relying on such central services significantly degrades the functionality of a DCS since centralized services can be points of failure, malicious intrusion and taking control of DCS, and can also be the bottleneck for the system. In addition, users are forced to trust central services administrators in matters of system operation, including access control, data management and use. Therefore, development of fully decentralized methods for managing data and access rights to them in distributed computer environments is an important problem.

The solution to this problem considered in this paper is applicable to collaborative distributed computing system (CDCS). This term is used to refer to distributed systems formed by combining into a single pool of computer resources of various institutions to work together within a project and, possibly, in conditions of complete or partial lack of trust between the user groups. In this case, it is fruitful to use the distributed ledger technology (DLT) [5], including blockchains [6, 7], to manage data in CDCS based on consensus between the parties involved. In the works [8, 9] a general approach to solving the above problem was suggested which is based on permissioned blockchain, smart contracts and metadata driven data management. This paper discusses access control methods in such systems.

## 2 Decentralized management of access rights in collaborative distributed systems

The basic scenario of using CDCSs assumes that a virtual organization (VO) is formed for the joint implementation of a certain project. VO includes several real institutions, in turn including data providers and handlers affiliated with them. The CDCSs can be formed by integrating computing resources of the institutions entering the VO and/or by renting cloud resources. For certainty, it is further assumed that the data are stored as files, i.e. a file is a unit of data. Each local storage that enters such a CDCS can have its own data management system (DMS). The problem is to combine all these data storages into a single system in a dynamically changing environment, and also ensure the implementation of reciprocal access policies to the data of the parties involved. This implies the development of decentralized management methods both for data access rights in such a dynamically changing environment and for ensuring a reliable, immutable recording the history of committed transactions, that is, recording provenance metadata (PMD). The PMD for a file consist of its global identifier (ID) and its attributes, including storage ID, creator ID, date/time of creation, number of file downloads, etc. The set of values for all attributes of a file determines its current state. The state of the entire distributed storage system is determined by the set of files stored in it with their states at the moment. In addition to the task of recording the immutable history of working with data in a distributed storage environment, the task of providing distributed management of access rights to data is set. For example, the owner of a data file (the user who created the data, the organization to which it belongs through its authorized representative) must be able to manage access rights to the file for other users. A natural solution [9] for the implementation of a distributed immutable ledger for the PMD records is the use of the Hyperledger Fabric blockchain platform (HLF; www.hyperledger.org) [10].

To describe the business process of data sharing within the framework of HLF platform, a number of concepts are used, the main ones are assets, participants and transactions. Assets are tangible or intellectual resources, services or property, records of which are kept in the ledger. Assets must have a unique identifier, but they can also contain any properties defined for them. In our case, the assets are data files; their properties (attributes) are the provenance metadata. Participants are members of the business network. They can own assets and make transaction requests. Like assets, the participants must have an ID and can have any other properties if necessary. The transaction is the mechanism of interaction of participants with assets.

On the basis of the HLF blockchain platform the ProvHL system for managing data in CDCSs were developed [8, 9]. It uses metadata driven data management. That means that the metadata are written to the blockchain beforehand, and DMSs of local storages entering a CDCS refer to the blockchain and performs the transactions recorded there. Along with the main assets, namely data files, data operations consisting of a set of consecutive transactions are considered as separate assets. The definition of the operations as assets leads to a number of advantages, in particular: keeping track of the operation own history recorded in the blockchain; improving the level of correspondence between the history recorded in the blockchain and the real history of the data in a CDCS [9]; making the mechanism of the delegation of rights in a distributed environment natural and flexible (see below).

In addition to files and operations, user groups are defined as additional assets. The introduction of user groups provides a fine grained mechanism for controlling access to data: along with granting access rights to individual users, one can grant the rights to entire groups. The most important attributes of a group are its ID (used as the primary key) and a list of its administrators' IDs. A new group of users within a virtual organization can be created by the administrator of the latter (the channel administrator in the HLF terminology). Modifications of this asset is allowed to the VO and group administrators while reading information about groups is allowed to all users. The definition of groups as assets allows tracking the history of their creation and modifications.

Changes in the group membership are reflected in the one more blockchain asset, namely *GroupMembership*, with the following main attributes: *assetID, groupID, userID*, as well as two Boolean attributes *admin_approval* and *member_approval* with *true/false* values. Group membership of a user is active if both member_approval and admin_approval have the value true. Modification of these values is carried out by means of the transactions *SetGroupMembershipAsMember* (application/consent of a user to become a member of the group) and

*SetGroupMembershipAsAdmin* (approval of the membership by the administrator).

File permissions in the ProvHL system are managed using the attributes of this asset containing access control lists. There are three such lists: *readACL* list is for access to read the file; *writeACL* is for access to modify the file; and *execACL* is for access to the file which is used either as a program for processing other files or as an input file for a data processing service. Each of these lists contains links to either users or user groups. As it was mentioned already, the ability to grant rights not only to individual users, but also to groups provides ability for a well-structured data access policy. Modifications to access lists are done using transactions with the self-explaining names *FileAccessGrant* and *FileAccessRevoke*, which are allowed only to the file owner.

In the case of directories, there is an additional transaction *SetStickyRights*, which changes the value of the Boolean attribute *StickyRights*. If the operations "upload" or "transform" create a file in a directory for which *StickyRights = true*, access rights to this file are carried over from this directory to the file. Creating this transaction is allowed only if the user owns the directory.

Rights delegation is the process of a user or a Web service handing over their authentication credentials to another executing Web service. We will present the mechanism of rights delegation between services on the example of the data copy operation from one local storage (Storage1) to another (Storage2). The definition of operations with files as the assets makes the mechanism of the delegation very natural and flexible. The operation definitions contain the obligatory attributes "requester" and "executor" as well as inherit "file owner" attribute from the file asset definition. This is important for the delegation mechanism. Upon receiving a request from a user for a file copying, the *DMS_Storage1* (the data management system of the *Storage1* which contains the file to be copied) detects the type of the copy operation, namely decides if this is local copying (within the *Storage1*) or copying to another storage. In the latter case it initiates, on behalf of the user, the operation of uploading the required file to destination *Storage2*. For this aim it interacts with the smart contract which, among other actions, defines that: (a) while for the initial copy operation the value of the requester attribute is equal to the user and the executor is *DMS_Storage1*, for the induced upload operation the requester is *DMS_Storage1* and the executor is *DMS_Storage2*; (b) the owner of the file copy on the *Storage2* is the same as the owner of source file on the *Storage1*. Thus, the second request is executed at the request of the user by the *DMS_Storage1* (source storage), however the file ownership does not change. This means that all goals of a delegation are completed. It is worth stressing that in contrast to the scheme based on proxy certificates [12], in the blockchain-based approach the delegation is restricted solely to the specified operation. This makes the delegation procedure much more secure.

To demonstrate the capabilities of the technologies used, as well as the benefits and potential of the suggested approach, we have set up a testbed of the ProvHL system. The testbed simulates a CDCS, in the environment of which there is a virtual organization including two real organizations. Each of the real

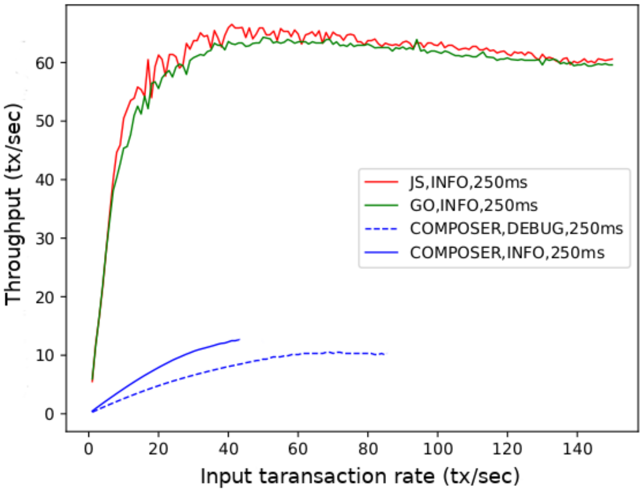

**Fig. 1.** Pure HLF vs HLF+Composer performance. Red and green plots show throughput for pure HLF (with smart contracts written in JavaScript and Go, respectively); blue and dashed blue plots are for HLF+Composer performance; indicated time (250ms) refers to max time for creating a block in the blockchain.

organizations is represented by two sites with their peers, instances of a distributed registry and ordering services. In addition, the testbed includes the security infrastructure based on the Membership Service Provider of the HLF blockchain platform. Peers and ordering services on different sites of the same organization mimic representatives of various user groups that may participate in approving or rejecting transactions according to the chosen policy.

The performance indicators of the developed system are under study and will be presented elsewhere. The preliminary measurements on the ProvHL testbed show that the overheads related to the operation processing by the ProvHL system is of the order of $4 \div 7$ *sec* depending on setup variables such as maximal time of block forming, etc. This is fully consistent with the extensive results of the recent work [11] on the performance of the Hyperledger platform itself, with the measurements in this work were carried out on a testbed similar to ours. In particular, it was shown that for the input transaction rate up to $800\, tx/sec$, the transaction latency is $\lesssim 1\, sec$, and the transaction throughput is $\sim 800\, tx/sec$. If we take into account that each file operation consists of $3 \div 7$ transactions (depending on the type of the operation), we get matching results for the latency, while for for the throughput we get $\sim 100\, ops/sec$. These values, obtained on the testbed with very modest computer facilities, are quite acceptable for operations with files of sufficiently large volumes, the handling time of which (copying, downloading, uploading, etc.) is tens or more seconds. Such volumes of data files are typical for distributed storages intended for large scientific experiments. It is worth mentioning that there exists a convenient tool for business process modeling in the HLF framework, namely the Hyperledger Composer (hyperledger.github.io/composer). However, as studies at our testbed showed, the performance of the HLF platform together with Hyperledger Composer is significantly worse than the performance of HLF platform alone, see Figure 1. Therefore, Hyperledger Composer can be used at the stage of developing a business process model within the blockchain network, but for real work a pure HLF platform should be used.

## 3 Conclusion

In this paper we have presented a solution for decentralized management of data access rights in geographically distributed collaborative computing systems (CDCS) with users from different institutions and with total or partial lack of trust between the user groups. The solution is based on the blockchain technology together with provenance metadata driven data management. It is free from the significant drawbacks inherent to other existing solutions, in particular, from the vulnerabilities associated with the presence of a central services which can be bottlenecks and points of failure. Within the framework of the proposed solution, it is possible to define user groups (as an independent asset), manage user membership in them, track and record the history of their appearance and evolution. The presence of groups enable a well-structured management of access rights to resources within the framework of CDCS. In addition, the suggested blockchain-based delegation proves to be reliable and flexible.

## References

1. WLCG Homepage, https://wlcg.web.cern.ch. Last accessed 24 May 2020
2. Open Grid Forum, https://www.ogf.org/ogf/doku.php/documents/documents. Last accessed 24 May 2020
3. Foster, I., Kesselman C. (eds): The Grid, Blueprint for a New computing Infrastructure. Morgan Kaufmann Publishers, New York (1998). Foster, I., Kesselman C. (eds): The Grid 2: Blueprint for a New Computing Infrastructure. Morgan Kaufmann Publishers, New York (2004)
4. Kryukov, A., Demichev, A.: Architecture of Distributed Data Storage for Astroparticle Physics. Lobachevskii Journal of Mathematics **39**(9), 1199–1206 (2018)
5. Maull, R., et al.: Distributed ledger technology: Applications and implications. Strategic Change, **26**(5), 481–489 (2017)
6. Norman, A. T.: Blockchain Technology Explained: The Ultimate Beginners Guide About Blockchain Wallet, Mining, Bitcoin, Ethereum, Litecoin, Zcash, Monero, Ripple, Dash, IOTA And Smart Contracts. CreateSpace Independent Publishing Platform (2017)
7. Baliga A.: Understanding Blockchain Consensus Models. Tech. rep., Persistent Systems Ltd (2017)

8. Demichev A. et al.: Provenance metadata management in distributed storages using the Hyperledger blockchain platform. In: Proceedings of The III International Workshop "Data life cycle in physics experiments 2019" (DLC'2019), pp. 35–42 (2019)
9. Demichev A., Kryukov A., Prikhod'ko N.: Metadata driven data management in distributed computing environments with partial or complete lack of trust between user groups. In: Proceedings of the 2019 Ivannikov ISPRAS Open Conference (ISPRAS'2019), IEEE Xplore Digital Library, IEEE Computer Society, pp. 35–42 (2020)
10. Androulaki E., et al.: Hyperledger Fabric: A Distributed Operating System for Permissioned Blockchains. In: Proceedings of the Thirteenth EuroSys Conference, article No. 30. ACM, Porto, Portugal (2018)
11. Baliga A., et al.: Performance Characterization of Hyperledger Fabric. In: Crypto Valley Conference on Blockchain Technology (CVCBT'2018), pp. 65–74 (2018)
12. Tuecke S, et al.: Internet X.509 Public Key Infrastructure Proxy Certificate Profile. Tech. Rep. RFC 3820 (2004).