The Approach to Managing Provenance Metadata and Data Access Rights in Distributed Storage Using the Hyperledger Blockchain Platform

A. Demichev, A. Kryukov and **N. Prikhod'ko**

SINP MSU

Novgorod State University

Supported by RSF grant No. 18-11-00075

Provenance Metadata (PMD)

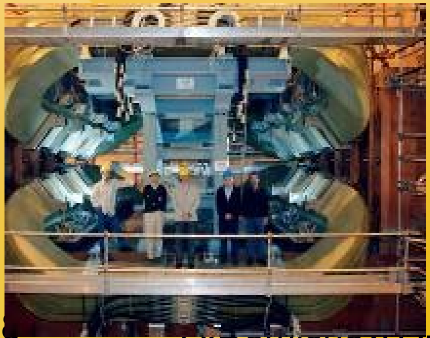
- Metadata describing data, provide context and are vital for accurate interpretation and use of data
- One of the most important types of metadata is provenance metadata (PMD)
  - tracking the stages at which data were obtained
  - ensuring their correct storage, reproduction and interpreting
  - ⇒ ensures the correctness of scientific results obtained on the basis of data
- The need for PMD is especially essential when large volume (big) data are jointly processed by several research teams

Examples of Large Experiments and Distributed Storages: **WLCG** (1/2)

- The Worldwide LHC Computing Grid (WLCG)
  - It was designed by CERN to handle the prodigious volume of data produced by Large Hadron Collider (LHC) experiments in high-energy (elementary particle) physics
    - approximately 25 petabytes per year
  - an international collaborative project
  - **grid**-based computer network infrastructure incorporating over 170 computing/storage centers in 36 countries

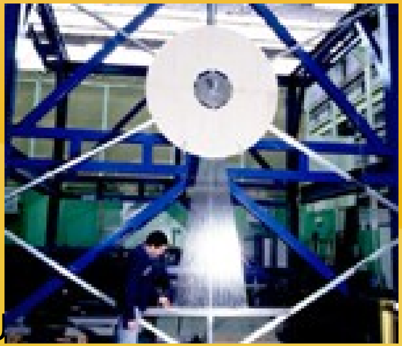
CMS

LHCb



ATLAS

ALICE



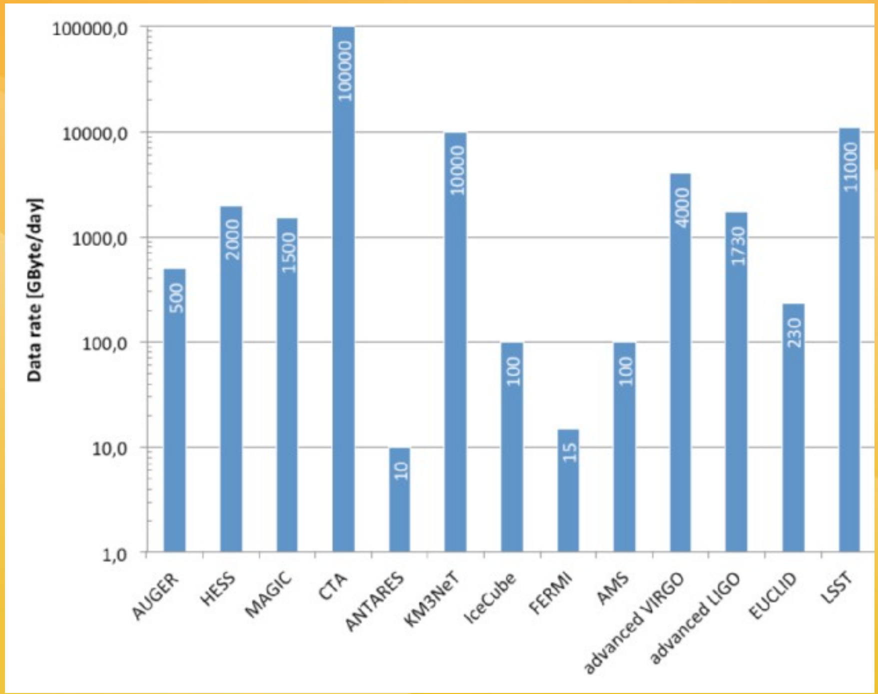
Examples of Large Experiments and Distributed Storages: WLCG (2/2)

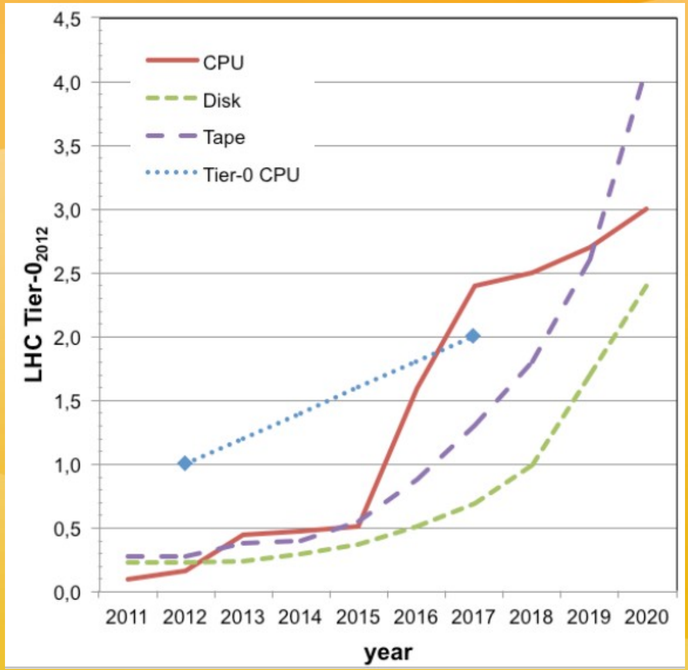
- time of active work of LHC ⇒ generation of big scientific data, is several tens of years, and the processing time of the data will be at least twice as much
  - without detailed and correct PMD comparing the results obtained with an interval, for example, in a few years, will be simply impossible

Examples of Large Experiments and Distributed Storages: Astrophysics

- While 10--15 years ago there were 1--10 Tb of data per year in astrophysics, new experimental facilities generate data sets ranging in size from 100's to 1000's of terabytes per year.

Berghöfer, T., et al. "Towards a model for computing in european astroparticle physics." ArXiv: 1512.00988 (2015)





Types of storages: extremal cases

- Centralized
  - problems:
    - very expensive ⇒ funding ?
    - planning in advance the necessary storage capacity
- P2P-storage with special mechanisms of coding, fragmentation and distribution
  - problems:
    - to ensure a stable pool of resource providers,
    - before such a P2P-based storage can work stably, it requires significant technical, organizational and time costs in the absence of a result guarantee

Types of storages: intermediate solution

- organizations participating in a large project
  - integrate their local storage resources into a unified distributed pool
  - if necessary, rent in addition cloud storage resources, perhaps from multiple providers.
- may be particularly advantageous
  - if there is a need to store large amounts of data for a limited duration of a project
  - in a situation where the project brings together many organizationally unrelated participants
- ⇒ dynamically changing distributed environment

PMD MS Construction: Distributed Solution

- distributed environment ⇒ distributed registry for PMD
- we suggested to use the blockchain technology which provides
  - that no records were inserted into the registry in hindsight
  - no entries were changed in the registry
  - the registry has never been damaged or branched
  - monitoring and restoring the complete history of data processing and analysis

PMD MS Construction: Which Blockchain (1/2)

- type of the blockchains
  - permissionless blockchains, in which there are no restrictions on the transaction handlers
  - permissioned blockchains, in which transaction processing is performed by specified entities
- permissionless:
  - algorithms are based on
    - Proof-of-Work – highly resource-consuming, probability of reaching a consensus, which grows with time elapsing, ...
    - Proof-of-Stake – Nothing-at-Stake problem,…
  - suitable for open (public) networks of participants (Bitcoin, etc.)

PMD MS Construction: Which Blockchain (2/2)

- Permissioned:
  - there is a fixed number of trusted transaction/block handlers
    - from different administrative domains
  - the handlers must come to a **consensus** about the content and the order of the recorded transactions
    - distributed consensus algorithm should be involved
  - form a more controlled and predictable environment than permissionless blockchains
  - suitable for networks with naturally existing trusted parties
    - our case: DMS, data owners,...

System state

- The state of the entire distributed storage = aggregated state of the set of files stored in it with their states at the moment
- The state of a data file is determined by PMD:
  - global ID + attributes, including:
    - local file name in a storage: fileName;
    - storage identifier: storageID;
    - creator identifier: creatorID;
    - owner identifier: ownerID
    - type: type=primary/secondary/replica
    - …

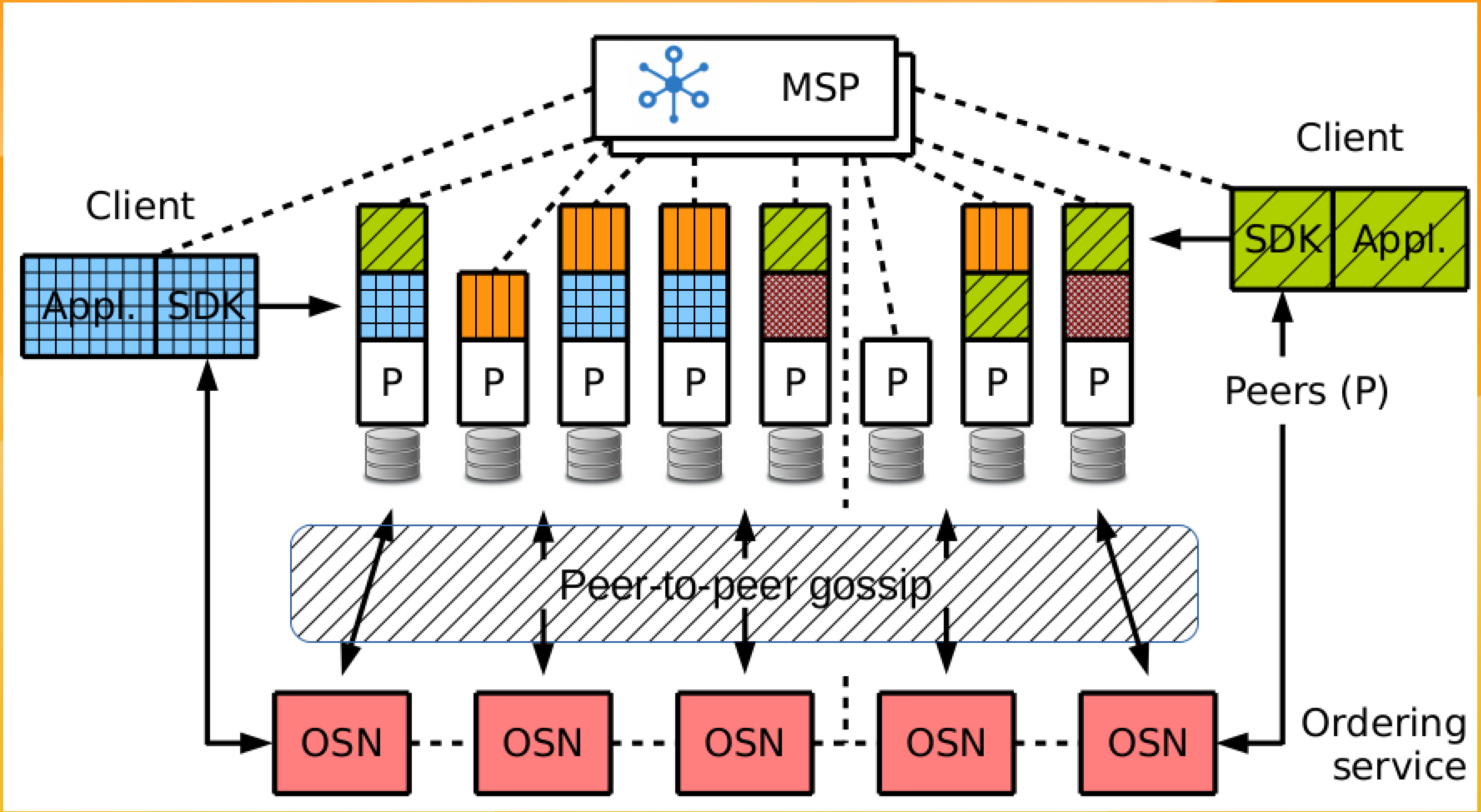
Basic operations ⇒ transactions

- new file upload
- file download
- file deletion
- file copy
- copying a file to another repository
- transferring a file to another repository
  - each active transaction ⇒ update of some state attributes
    - for example, after the transaction "file download" the values of the keys change: "number of file downloads" and "users who downloaded the file".

HyperLedger Fabric (1/2)

- Analysis of existing platforms shows that the formulated problems most naturally can be solved on the basis of the
  - Hyperledger Fabric blockchain platform (HLF; www.hyperledger.org)
  - together with Hyperledger Composer (HLC; hyperledger.github.io/composer) = set of tools for simplified use of blockchains
- permissioned blockchains
  - transactions are processed by a certain list of trusted network members

HyperLedger Fabric (2/2)



From: *E. Androulaki et al. "Hyperledger Fabric: A Distributed Operating System for Permissioned Blockchains," in Proc 13th EuroSys Conf. 2018*

Business process within (HLF&C)-platform

- **Assets** are tangible or intellectual resources, records of which are kept in registers
  - in our case, the assets are data files; their properties (attributes) are provenance metadata
- **Participants** are members of the business network.
  - they can own assets and make transaction requests
  - can have any properties if necessary
- **Transaction** is the mechanism of interaction of participants with assets
- **Events:** messages can be sent by transaction processors to inform external components of changes in the blockchain

HyperLedger Fabric → ProvHL (1/3)

- ProvHL = Provenance HyperLedger
  - status: Proof of concept
- operation of smart contracts (chaincodes)
  - sophisticated adaptation of HLF for the business process of sharing storage resources
- provides a record of transactions & advanced query tools
- advanced means for managing access rights
  - access rights can be managed by network members within their competence

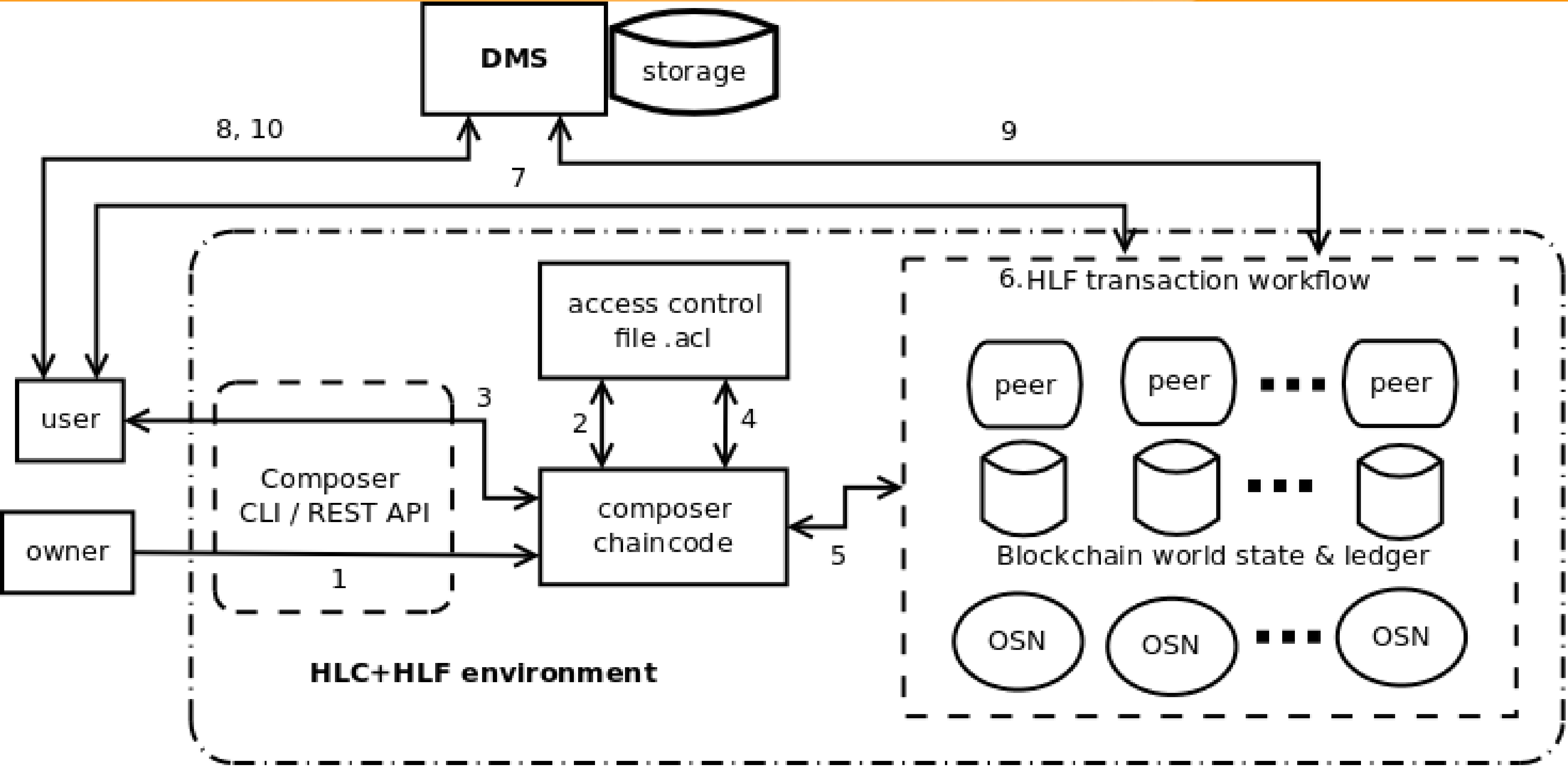
HyperLedger Fabric → ProvHL (2/3)

- Participants
  - Person
  - StorageProvider
- Assets
  - File
  - Storage
  - Operation
  - Group
- Transactions
  - FileAccessGrant
  - FileAccessRevoke
  - OperationUploadCreate
  - OperationUploadSetState
  - ...

HyperLedger Fabric → ProvHL (3/3)

- thanks to its modular structure, it allows using different algorithms to reach consensus between business process participants
- has a developed built-in security system based on PKI

ProvHL operation (1/3)



Simplified scheme for recording transactions with provenance metadata and managing data access rights based on HLF&C

ProvHL operation (2/3)

- Operations with files comprise of two types of transactions recorded in the blockchain:
  - first corresponds to client requests,
  - second corresponds to server responses
- Operation states
  - STARTED
  - PENDING
  - COMPLETED
  - ERROR

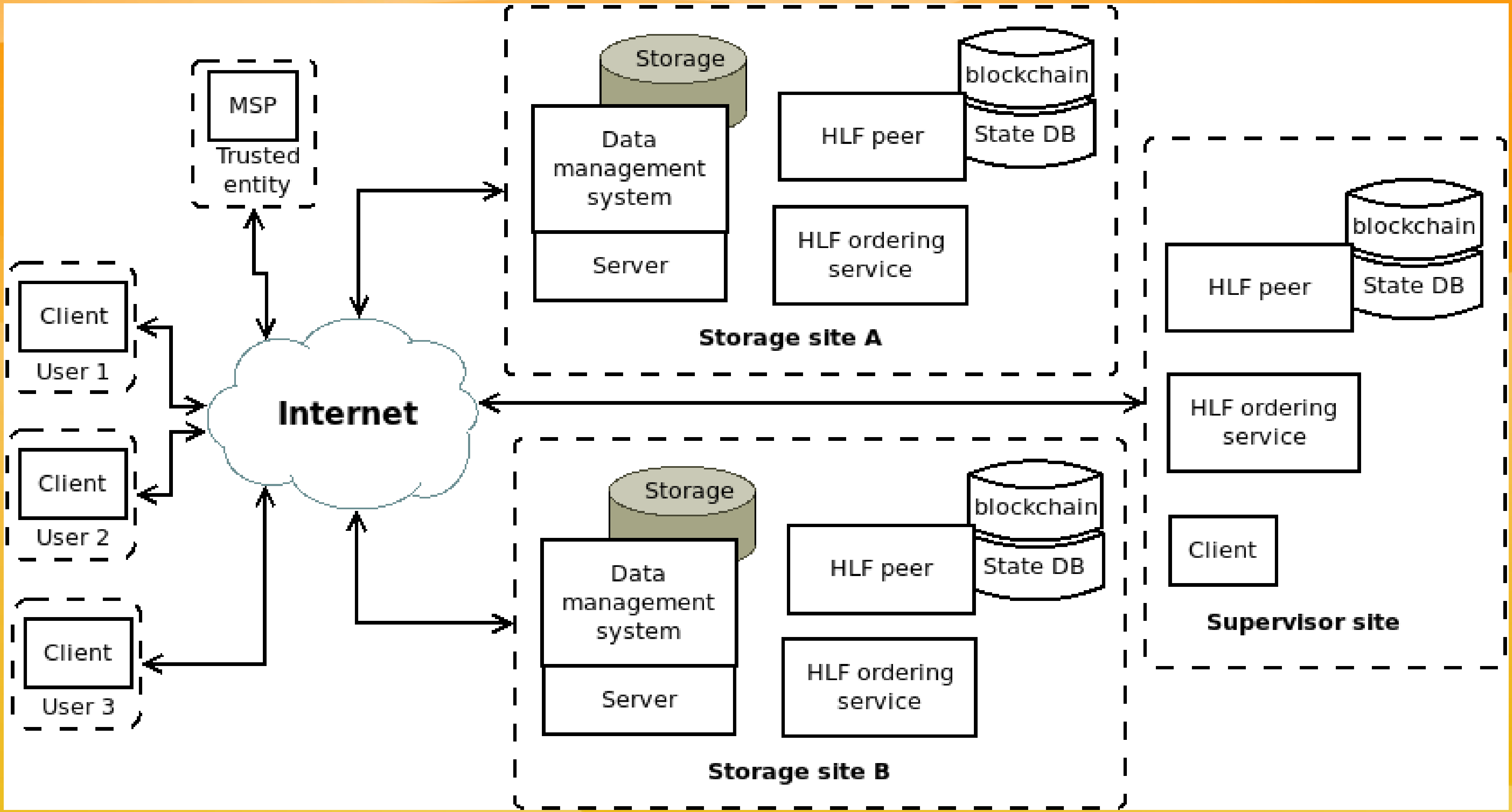
ProvHL operation (3/3)

- Example - "new file upload" transaction:
  - a new asset — a data file — with the "temporary" label is first recorded in the blockchain
  - only after the actual upload of the file in the storage, DMS initiates a transaction removing the label "temporary" and turns the uploaded file into a fully valid asset.
- Together with the splitting of transactions into the client and server parts ⇒ level of correspondence (history recorded in blockchain) ⇔ (real history of the data in the distributed storage) practically acceptable.

ProvHL Testbed (1/2)

- At present, a testbed has been created on the basis of the SINP MSU
  - a preliminary version of the ProvHL prototype was deployed to implement the developed principles and refine the algorithms of the system
  - a trivial consensus algorithm is currently used (centralized orderer Solo in the terminology of HLF).
  - full-fledged Byzantine fault tolerant consensus algorithms is under implementation
    - PBFT

ProvHL Testbed (2/2)



Conclusion

- The new approach to managing PMD and data access rights in distributed storage has been developed