Access Rights Management in Decentralized Distributed Computing Systems

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

SINP MSU

Supported by RSF grant No. 18-11-00075

Collaborative DCS

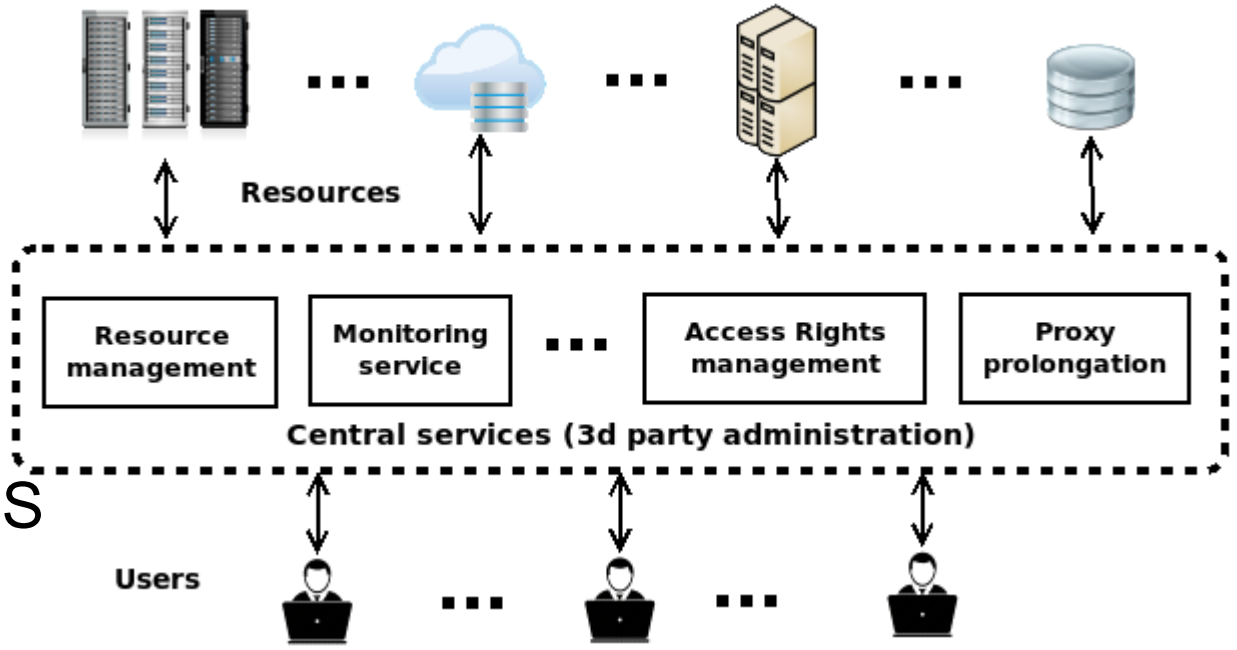
- organizations participating in a large project integrate their local computing resources into a unified distributed pool
  - administratively unrelated user groups share computing resources based on an agreed policy
  - in the conditions of a partial or complete lack of trust between them
- such DCSs correspond to the baseline use case being under discussion at this Workshop
- ⇒ dynamically changing distributed environment
  - ensuring reliable and safe operation of systems in such conditions is a complex problem

Typical structure of a CDCS

- **distributed** resources are combined into a single pool using the infrastructure based on **centralized** services

- layer of user interfaces
- layer of resource sites
- layer of system-wide

centralized services that manage the work of CDCS as a whole



→potential points of failure, malicious intrusion and/or bottlenecks

→users have to trust 3d party administrators

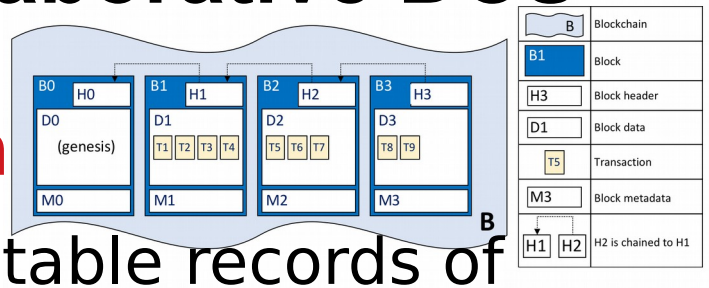
Approach to Decentralization

- built upon the integration of the following solutions (DLC-2019):
  - use of a distributed ledger based on permissioned blockchain;
  - use of smart contracts
    - defines rules of data sharing in CDCS in the form of executable code
  - metadata driven data management;
    - provenance metadata (system history) are written to the blockchain in advance, and data management system accesses the blockchain and performs the transactions recorded there;
  - consensus between representatives of the parties in the data sharing process.

The Approach Implementation: ProvHL

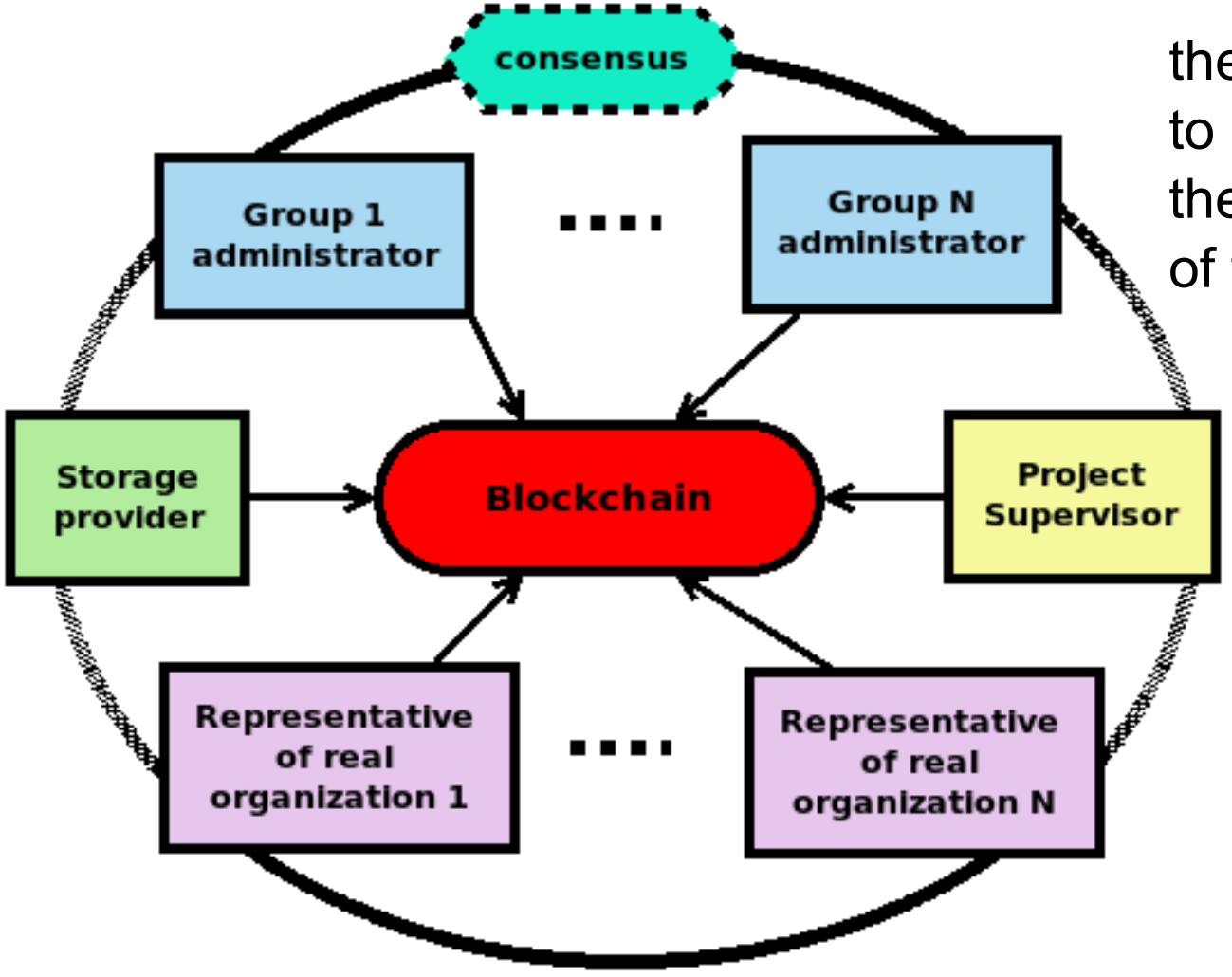
- ProvHL = Provenance HyperLedger
  - based on the Hyperledger Fabric (HLF) blockchain platform
  - provenance metadata are at the core of the system
    - tracking the stages at which data were obtained
- provides decentralized data management, including
  - advanced means for managing access rights
  - access rights can be managed by CDCS users within their competence

Decentralized Ledger for Collaborative DCS



- distributed ledger = **blockchain**
  - data structure that provides immutable records of provenance metadata = history of all activities in the framework of a DCS
- permissioned blockchain:
  transactions are processed by specified entities
- form a more controlled and predictable environment than public blockchains (Bitcoin, etc.) + much better performance
- suitable for networks with naturally existing trusted entities for the transaction processing

Examples of Transaction Handlers in Collaborative DCSs



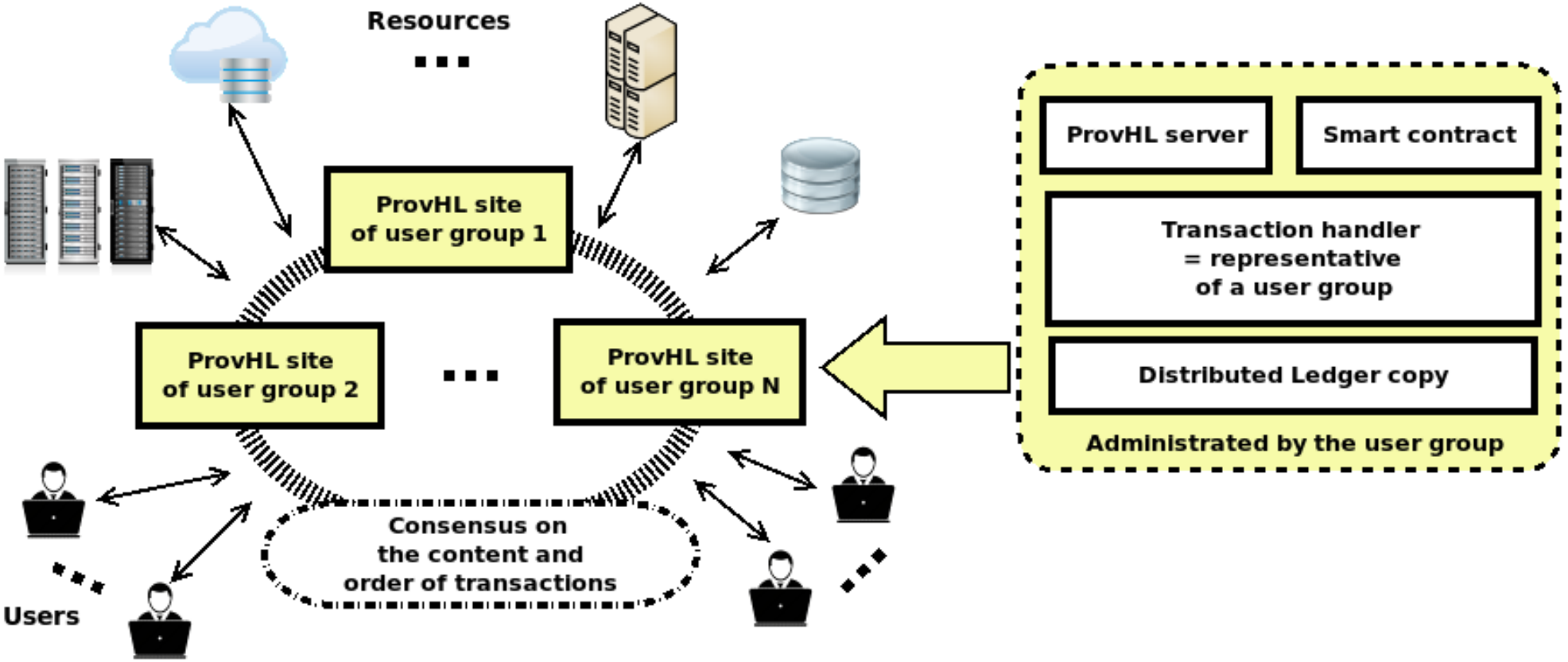
the handlers must come to a consensus about the content and the order of the recorded transactions

this is achieved via distributed consensus algorithms

State of the DCS Recorded in the Blockchain

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

General Structure of a Collaborative DCS under ProvHL Management

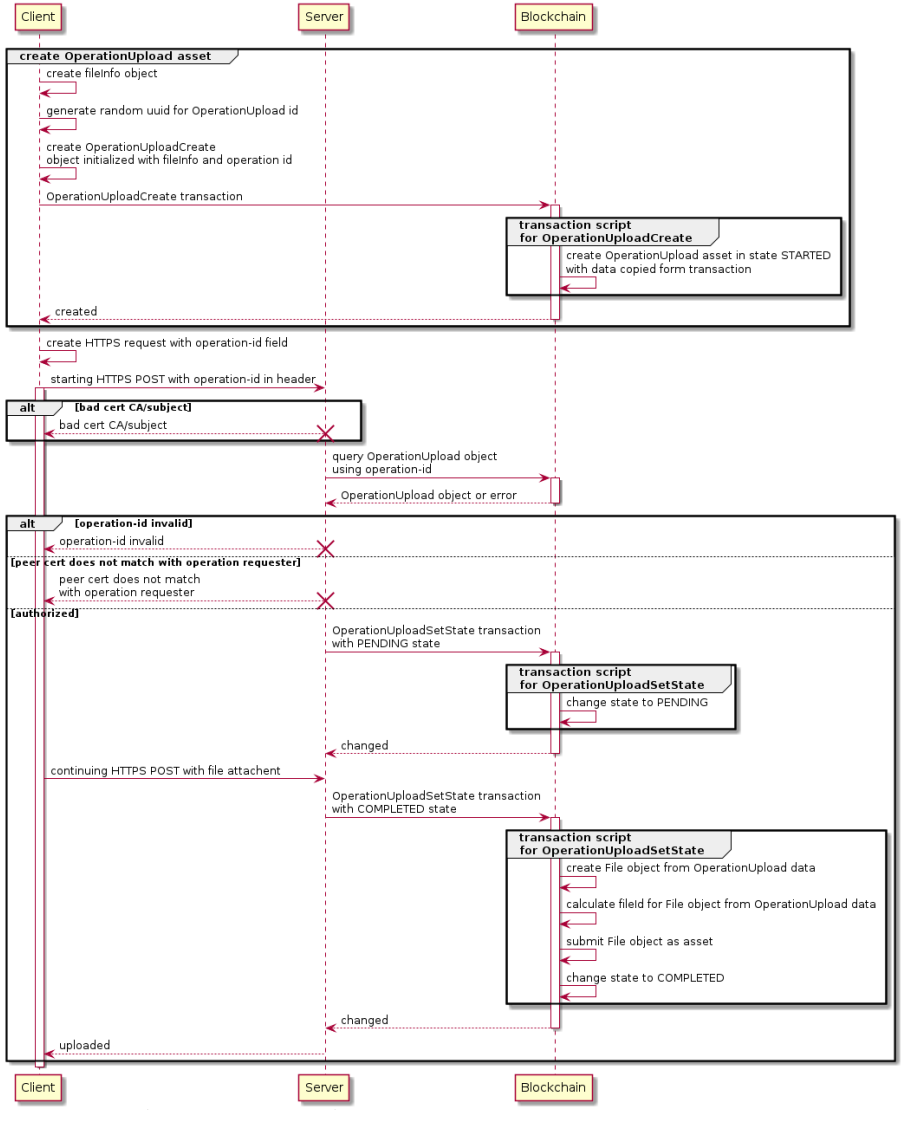


ProvHL: Basic operations ⇒ transactions

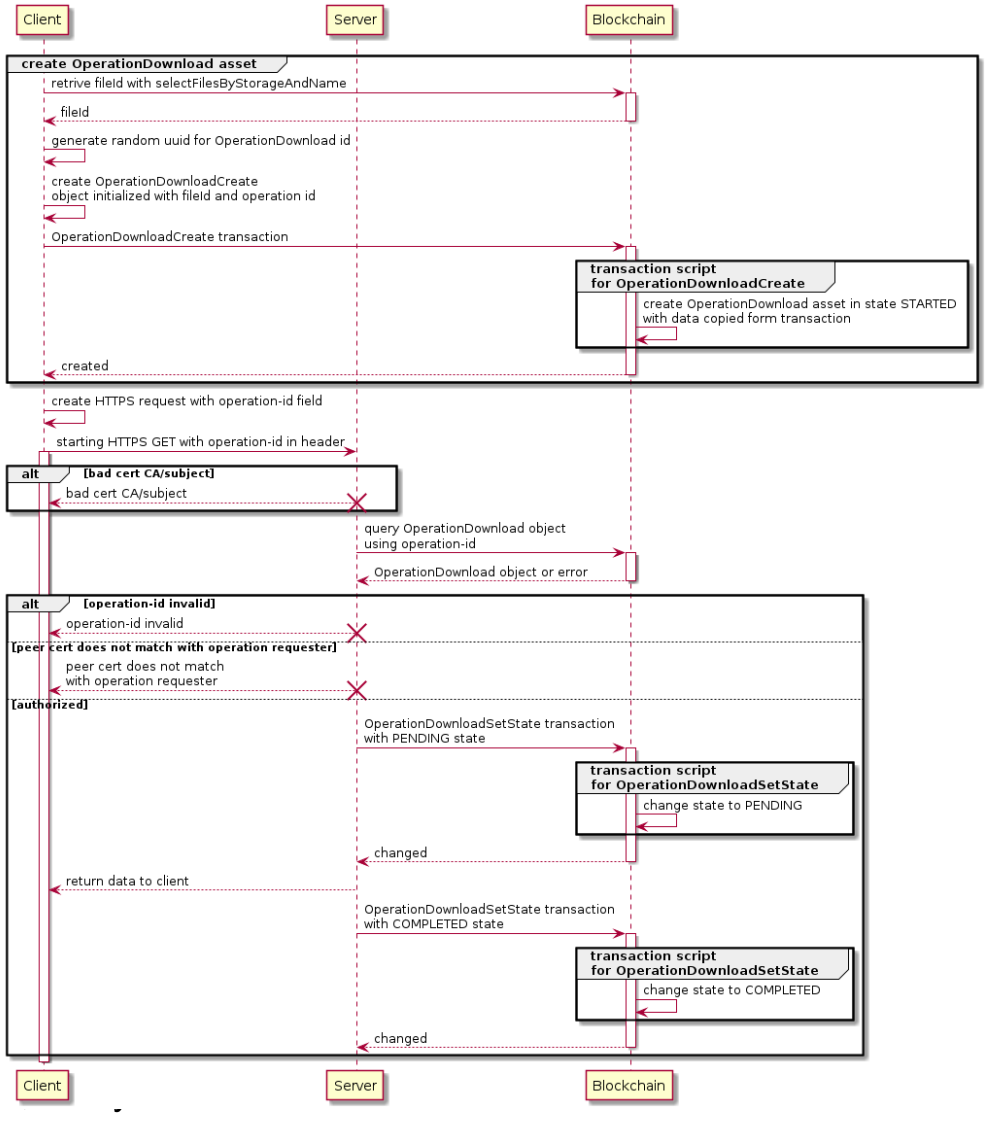
- new file upload
- file download
- file deletion
- file copy within local storage
- file copy/transfer to another local storage
- file transformation by a special service ⇒ grid-like DCS
  - each operation comprises of a number of transactions
  - each valid transaction ⇒ update of some state attributes
    - for example, after the transaction "file download" the values of the keys change: "number of file downloads" and "users who downloaded the file".

Sequence Diagrams

Upload



Download



Decentralized Access Rights Management: File Permissions

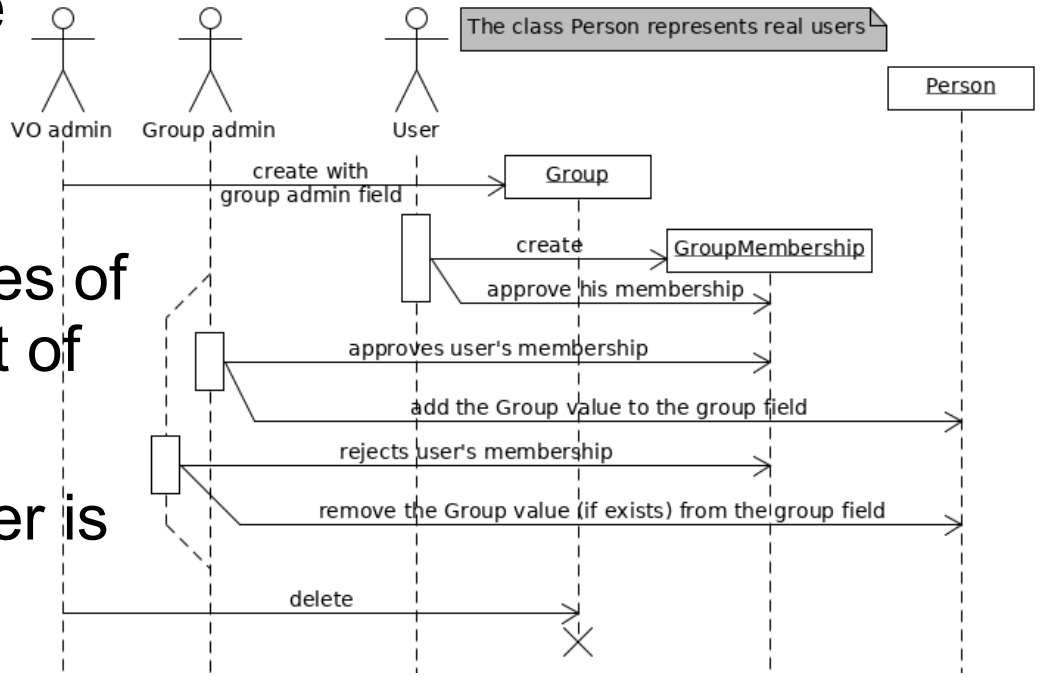
- managed by using the **ACLs** attributes of a file containing access control lists
  - readACL list is for access to read the file;
  - writeACL is for access to modify the file;
  - execACL is for access to the file
    - used either as a program for processing other files
    - or as an input file for a data processing service
  - each of these lists contains links to either users or user groups ⇒ well-structured management of access rights

Decentralized Access Rights Management (2)

- when a user requests an operation with a file, the smart contract generating the required transactions (atomic parts of the operation) checks that the user/groups's ID is in the appropriate ACL of the file
  - thereby providing access rights control
- modifications to ACLs ⇒ special transactions
  - allowed only to the file/directory owner

Group Management

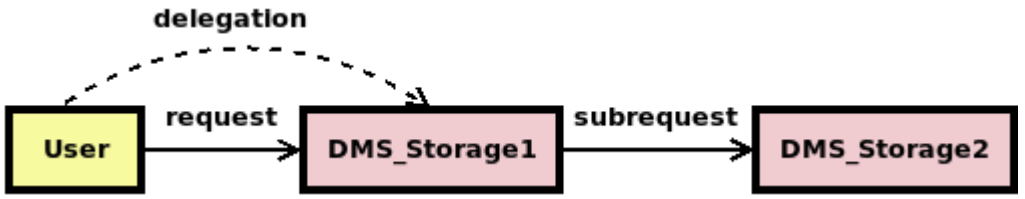
- a new group of users – as a blockchain asset - within a virtual organization can be created by the project administrator
  - the most important attributes of a group are its ID and a list of its administrators' IDs
  - group membership of a user is active if both the group administrator and the user himself approved it.



Directories

- virtual (overlay) directories throughout the CDCS
  - information about which is contained in the blockchain
  - each of the local storages that are included in the CDCS and actually store the files has its own local directory structure
- for the directories, there is an additional transaction **SetStickyRights**
  - 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 the transaction is allowed only if the user owns the directory

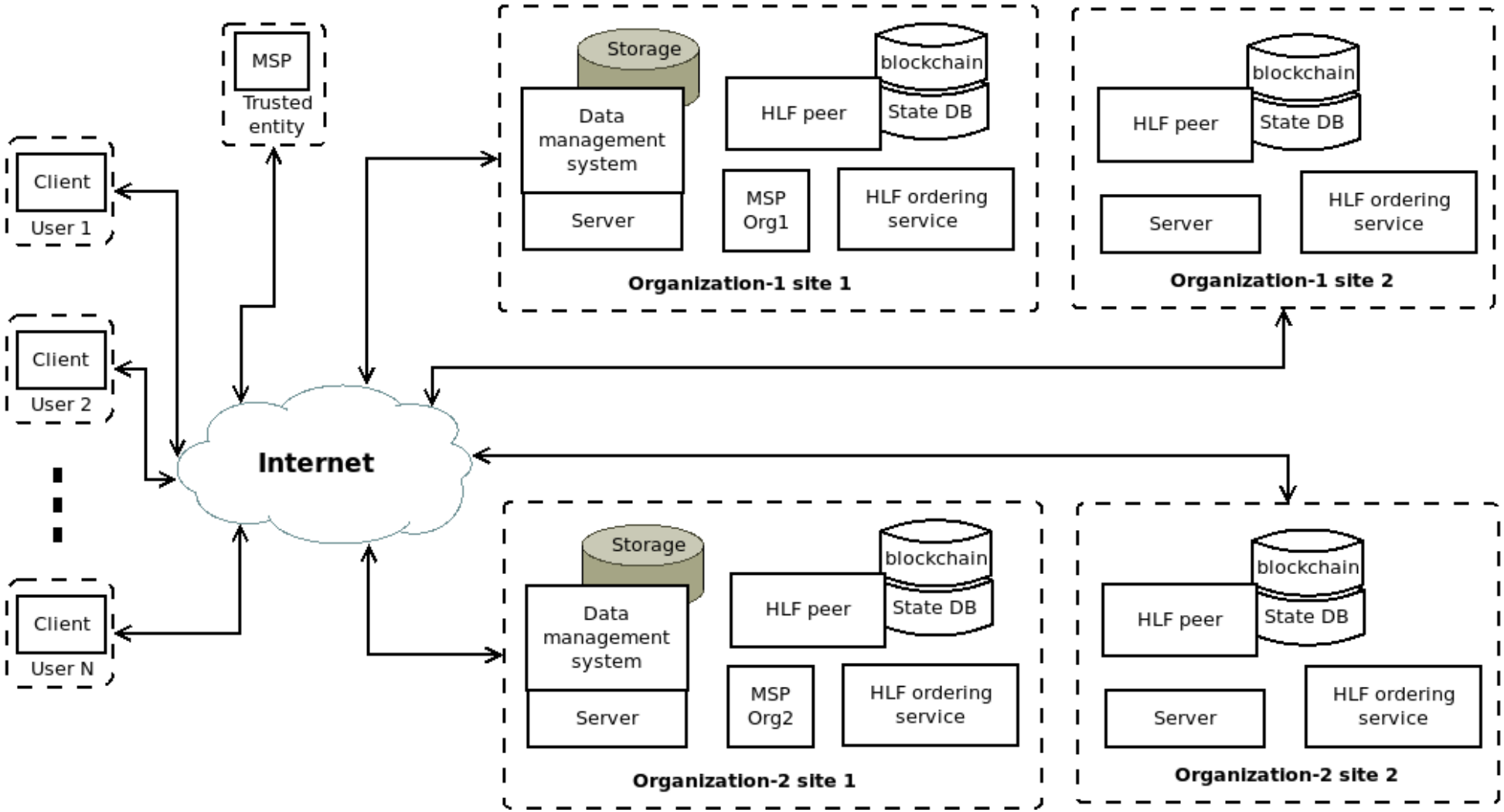
Rights Delegation in ProvHL



delegating rights from a user or service to another service in the ProvHL environment

- Usual proxy-based delegation in DCS:
  - too many rights are transferred to the proxy recipient ⇒ low level of security
  - central service for proxy certificate prolongation
- ProvHL:
  - all the info about delegation process is recorded into the blockchain
  - services check the validity of the delegated transaction under control of smart contract

ProvHL Testbed



Performance Characterization of HLF & ProvHL

- HLF
  - for the input transaction rate up to 800 tx/sec, the transaction latency is ≤1 sec
  - transaction throughput is ~ 800 tx/sec
- ProvHL (each file operation consists of 3 ÷ 7 transactions)
  - ⇒ matching results for the latency ~ 4 ÷ 7 sec
  - throughput ~ 100 ops/sec.
- quite acceptable for operations with files of sufficiently large volumes
  - typical for DCS for large scientific experiments

Conclusion (1/2)

- we have suggested the new approach to the PMD driven data management in DCSs based on the integration of
  - blockchain technology
  - smart contracts
  - metadata driven data management
  - consensus algorithms
- intended for operation in a distributed environment with administratively unrelated organizations participating in joint projects
  - conditions of incomplete trust or lack of trust between groups of users of the system

Conclusion (2/2)

- ProvHL system on the top of Hyperledger Fabric blockchain platform
  - completely distributed/decentralized ⇒ fault-tolerant
    - free from the vulnerabilities associated with the presence of central services which can be bottlenecks and points of failure
  - safe and secure PMD and data management system
  - well granular access control management
    - including delegation of rights
  - testbed performance characteristics are promising
  - further details:

https://theory.sinp.msu.ru/dokuwiki/doku.php/rsf00075/publications