Metadata Driven Data Management in Distributed Computing Environments with Partial or Complete Lack of Trust Between User Groups

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

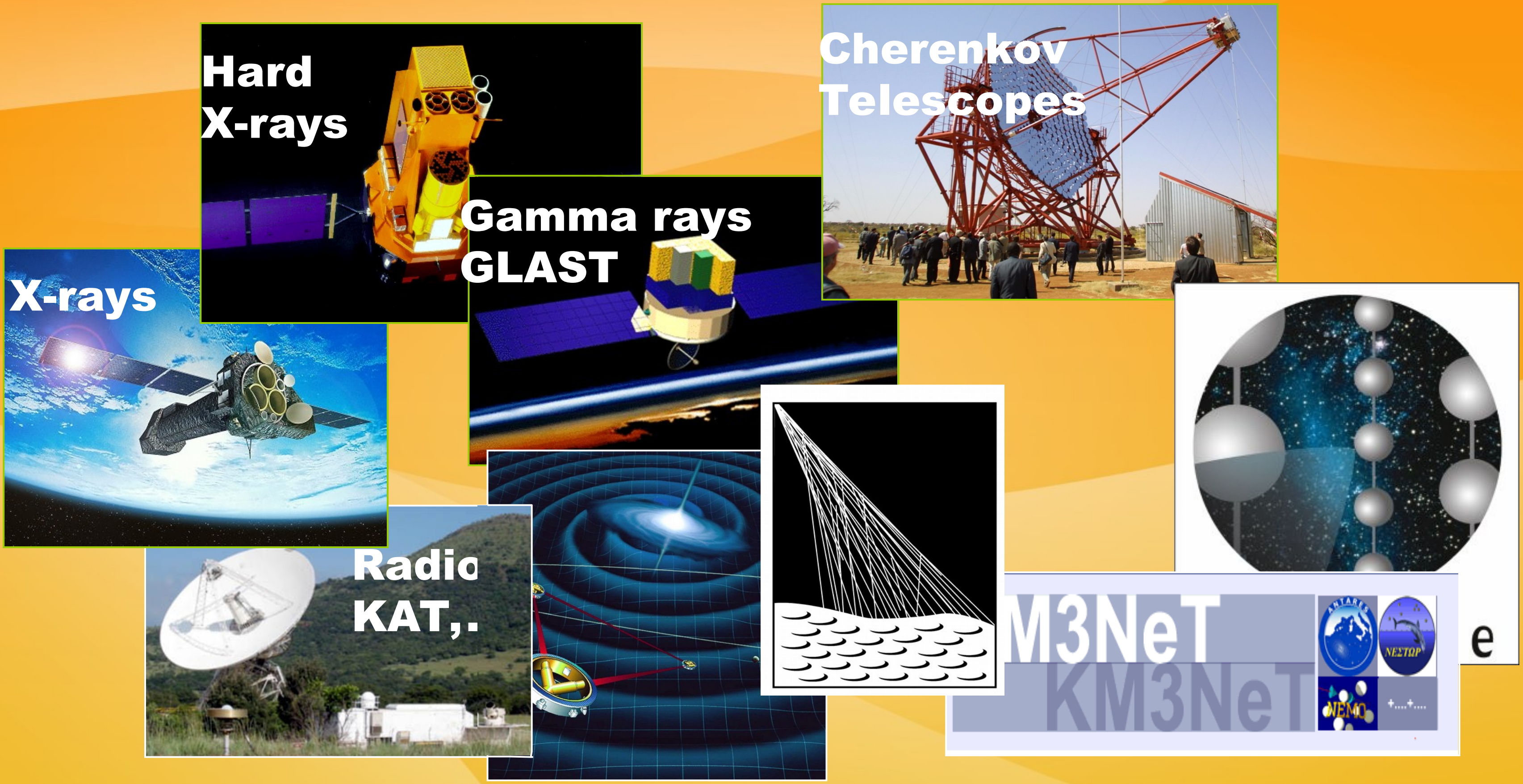
SINP MSU

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

Multimessenger astronomy



Organizations participating in a large project integrate their local computing/storage resources into a unified distributed pool

PMD MS Construction for Collaborative DCS

- distributed environment ⇒ distributed registry for PMD
- conditions of incomplete trust or lack of trust between groups of users of the system
- ⇒ **blockchain** = distributed registry + provides:
  - no records can 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

PERMISSIONLESS BLOCKCHAIN

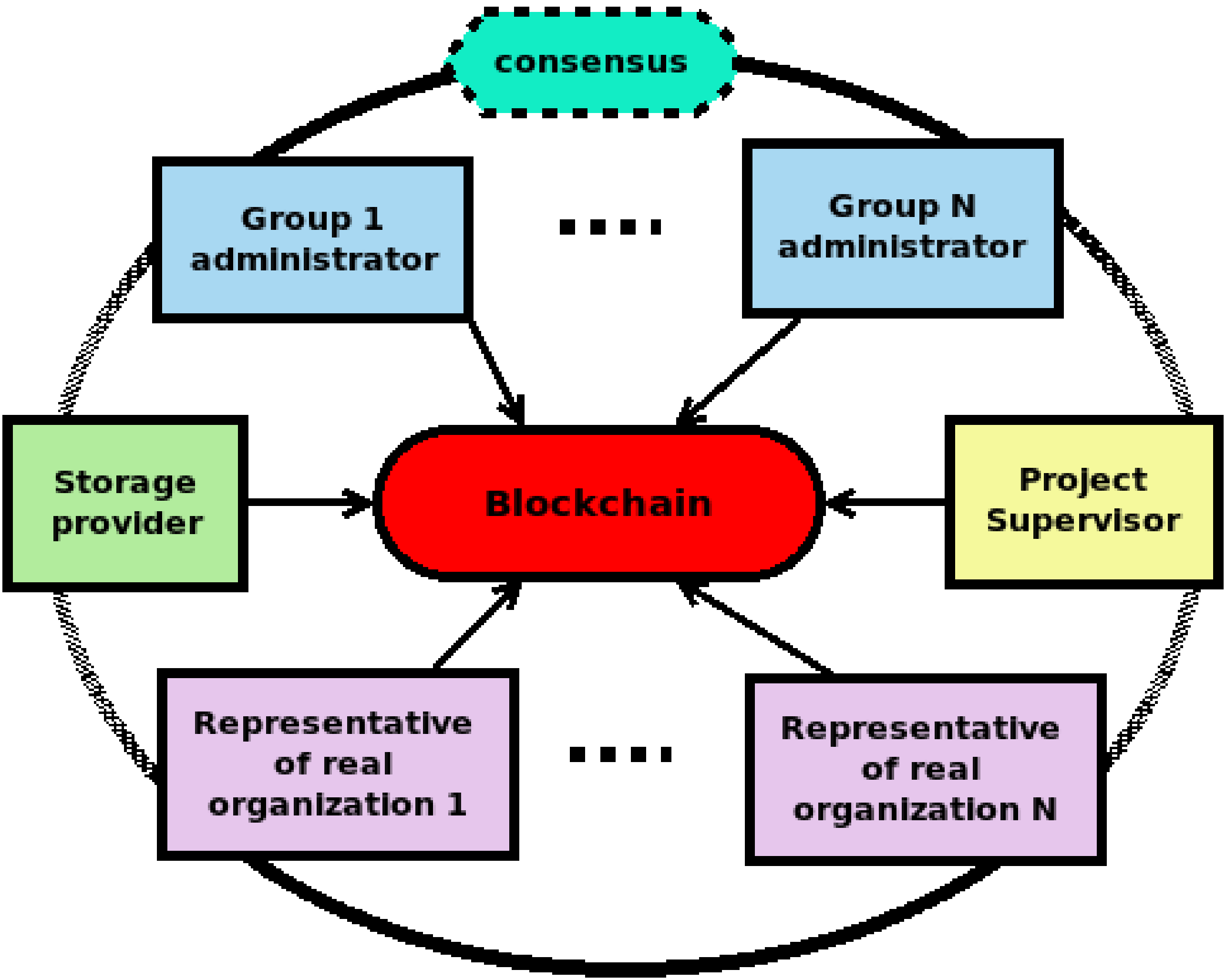
Trust is enforced by cryptographic proof from the software protocol

PERMISSIONED BLOCKCHAIN

Trust is enforced through restricted access to transaction validation

- permissionless: no restrictions on the transaction handlers
  - open (public) networks of participants (Bitcoin, etc.)
- **permissioned**: transaction processing by specified entities
  - the handlers must come to a consensus about the content and the order of the recorded transactions
  - form a more controlled and predictable environment
  - suitable for networks with naturally existing trusted parties
    - our case: storage providers, representatives of real organizations participating in the project,...

Transaction Handlers in Collaborative DCS



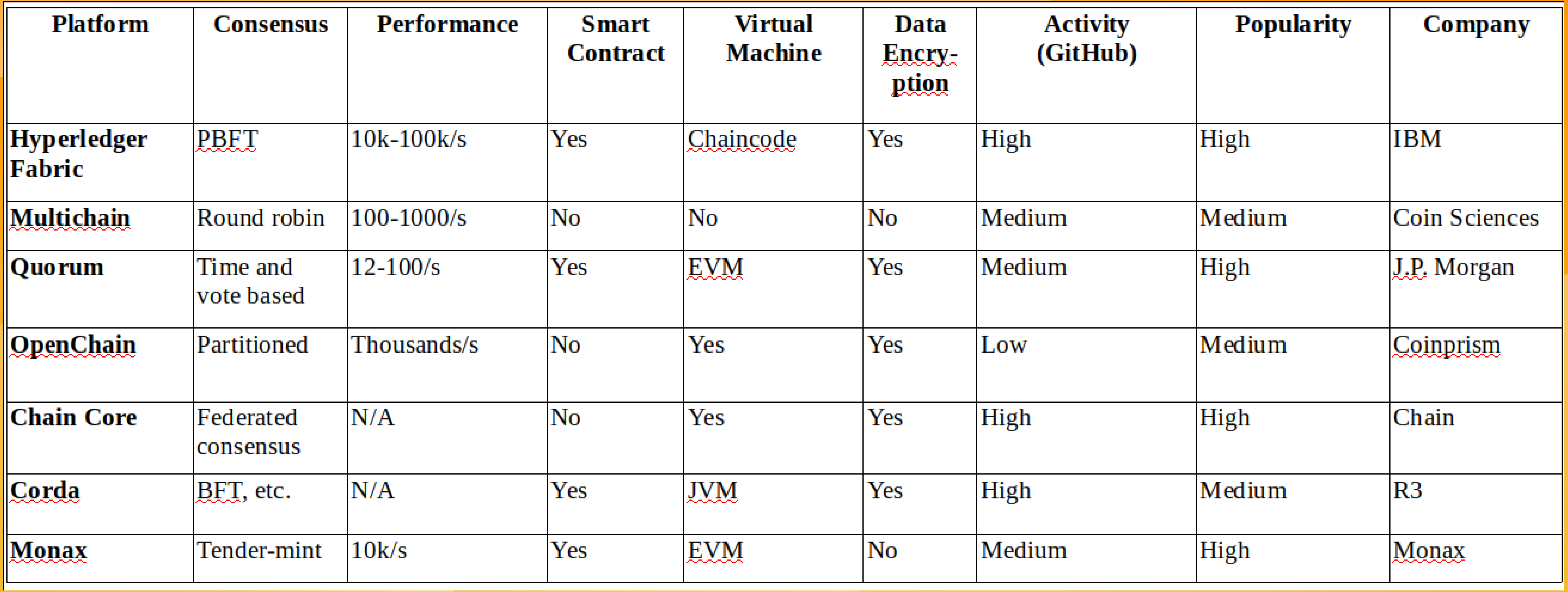
System State of DCS Recorded in the Blockchain: preview

- 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 metadata
  - 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
    - …

Smart contracts

- Smart contracts along with the registry form the basis of a blockchain system
  - determines the executable logic that generates new states to be added to the registry
- Parties of a business process must define a common set of contracts covering common terms, data, rules, concept definitions and processes.
  - Taken together, these contracts define a business model that governs all interactions between transactional parties.
- A smart contract defines these rules between the parties in the form of executable code.

Permissioned Blockchain Platforms



Platform	Consensus	Performance	Smart Contract	Virtual Machine	Data Encryption	Activity (GitHub)	Popularity	Company
Hyperledger Fabric	PBFT	10k-100k/s	Yes	Chaincode	Yes	High	High	IBM
Multichain	Round robin	100-1000/s	No	No	No	Medium	Medium	Coin Sciences
Quorum	Time and vote based	12-100/s	Yes	EVM	Yes	Medium	High	J.P. Morgan
OpenChain	Partitioned	Thousands/s	No	Yes	Yes	Low	Medium	Coinprism
Chain Core	Federated consensus	N/A	No	Yes	Yes	High	High	Chain
Corda	BFT, etc.	N/A	Yes	JVM	Yes	High	Medium	R3
Monax	Tender-mint	10k/s	Yes	EVM	No	Medium	High	Monax

Analysis of existing platforms: the **Hyperledger Fabric** (hyperledger.org) is most suitable for the use case under consideration

Hyperledger Fabric (HLF) → ProvHL

- ProvHL = Provenance HyperLedger
- operation of smart contracts (chaincodes)
  - 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

PMD driven data management

- two approaches are possible
  - data management systems (DMS) manage data and use a blockchain simply as a distributed log
    - data driven data management
  - metadata is written to the blockchain beforehand, and DMSs refer to the blockchain and performs the transactions recorded there
    - metadata driven data management
- ProvHL implements the second approach

Business process within HLF-platform

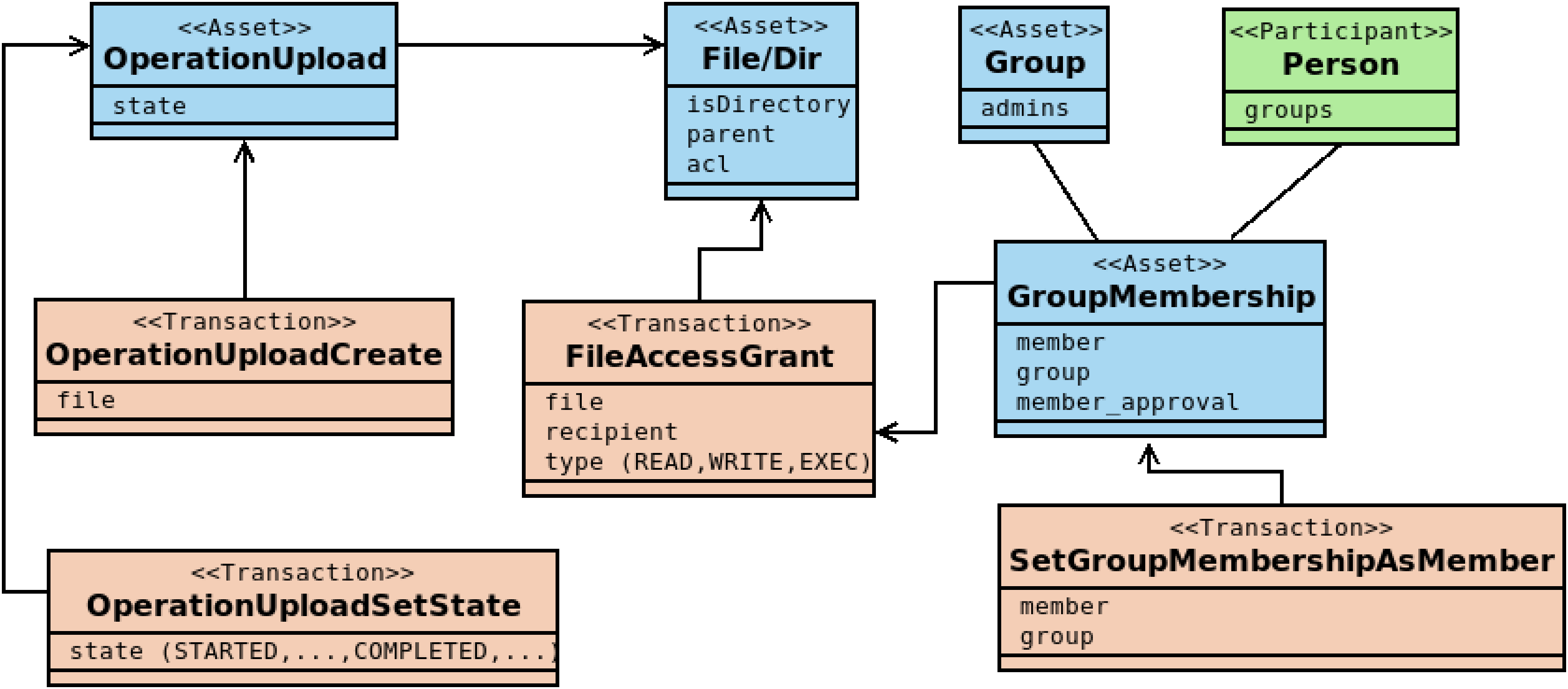
- **Assets** are tangible or intellectual resources, records of which are kept in registers
  - in our case, the assets are data files; their properties (attributes) are provenance metadata
- **Participants** are members of the business network.
  - they can own assets and make transaction requests
  - 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

- Participants
  - Person
  - StorageProvider
- Assets
  - File/Directory
  - Storage
  - Operation
  - Group
  - GroupMembership

- Transactions
  - FileAccessGrant
  - FileAccessRevoke
  - OperationUploadCreate
  - OperationUploadSetState
  - …

DCS Management Model: Core Structure



There exist MANY other classes and relationships, in particular:
`StorageProvider, OperationCopy,…, SetGroupMembershipAsAdmin, FileAccessRevoke,… "sticky rights", ...`

ProvHL: Basic operations ⇒ transactions

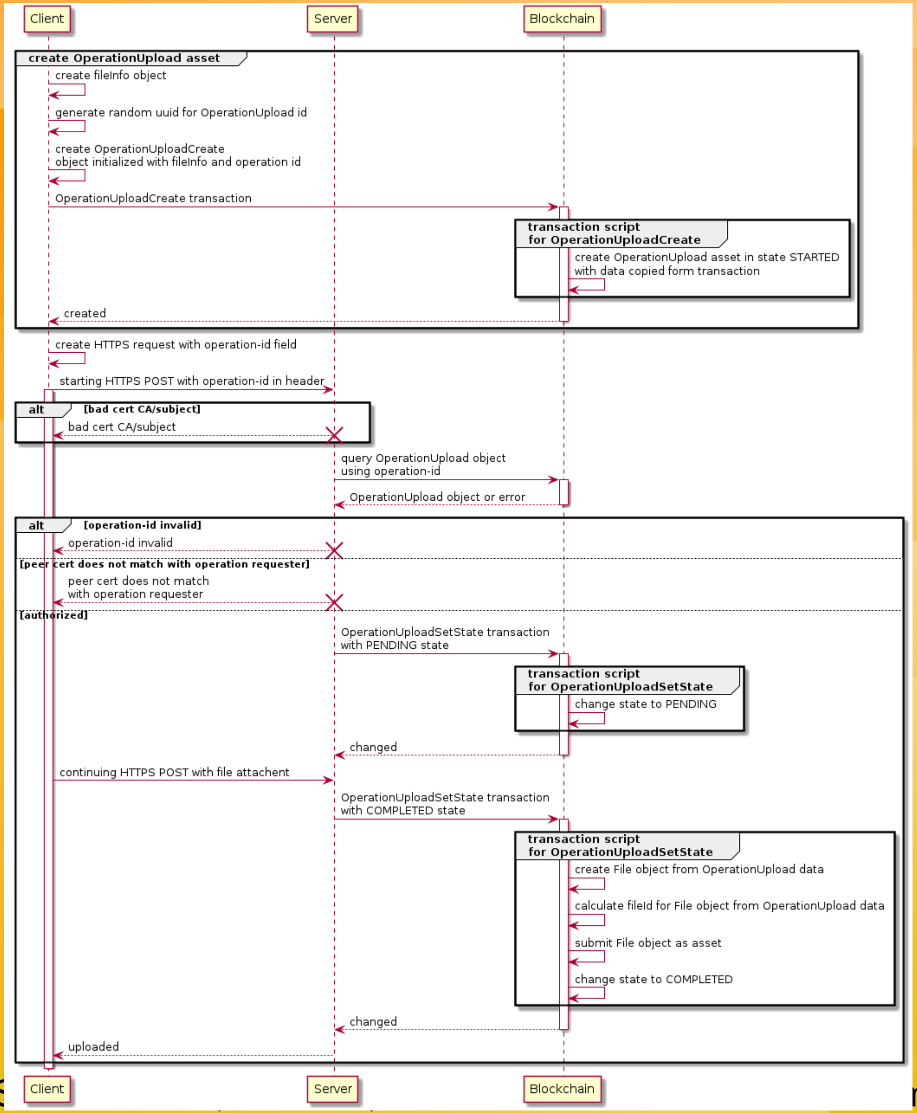
- new file upload
- file download
- file deletion
- file copy within local storage
- file copy/transfer to another local storage
- file transformation by a special service ⇒ grid-like DCS
    - each operation comprises of a number of transactions
    - each valid transaction ⇒ update of some state attributes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

ProvHL operation

- Operations with files comprise of at least two types of transactions recorded in the blockchain:
  - client requests,
  - server responses
- Operation states: ***`STARTED, COMPLETED, ERROR,`*** …
- Operation = asset ⇒
  - level of correspondence (history recorded in blockchain) ⇔ (real history of the data in the distributed storage) practically acceptable
  - delegation of rights: user/service → service
  - ...

Sequence Diagrams

Upload



Client
Server
Blockchain

create OperationUpload asset

create fileInfo object

generate random uuid for OperationUpload id

create OperationUploadCreate
object initialized with fileInfo and operation id

OperationUploadCreate transaction

transaction script
for OperationUploadCreate

create OperationUpload asset in state STARTED
with data copied form transaction

created

create HTTPS request with operation-id field

starting HTTPS POST with operation-id in header

alt **[bad cert CA/subject]**

bad cert CA/subject

query OperationUpload object
using operation-id

OperationUpload object or error

alt **[operation-id invalid]**

operation-id invalid

[peer cert does not match with operation requester]

peer cert does not match
with operation requester

[authorized]

OperationUploadSetState transaction
with PENDING state

transaction script
for OperationUploadSetState

change state to PENDING

changed

continuing HTTPS POST with file attachent

OperationUploadSetState transaction
with COMPLETED state

transaction script
for OperationUploadSetState

create File object from OperationUpload data

calculate fileId for File object from OperationUpload data

submit File object as asset

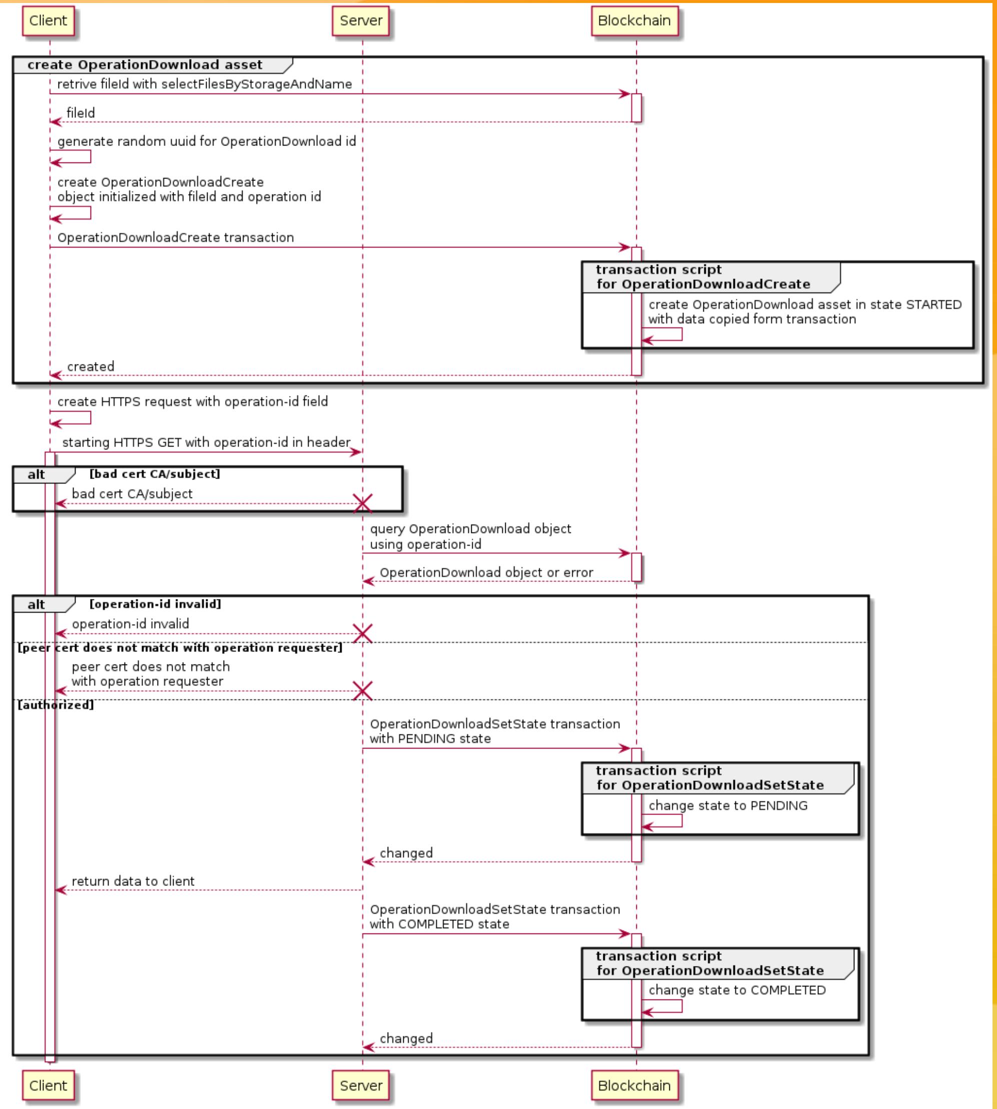
change state to COMPLETED

changed

uploaded

Client
Server
Blockchain

Download



Client
Server
Blockchain

create OperationDownload asset

retrive fileId with selectFilesByStorageAndName

fileId

generate random uuid for OperationDownload id

create OperationDownloadCreate
object initialized with fileId and operation id

OperationDownloadCreate transaction

transaction script
for OperationDownloadCreate

create OperationDownload asset in state STARTED
with data copied form transaction

created

create HTTPS request with operation-id field

starting HTTPS GET with operation-id in header

alt **[bad cert CA/subject]**

bad cert CA/subject

query OperationDownload object
using operation-id

OperationDownload object or error

alt **[operation-id invalid]**

operation-id invalid

[peer cert does not match with operation requester]

peer cert does not match
with operation requester

[authorized]

OperationDownloadSetState transaction
with PENDING state

transaction script
for OperationDownloadSetState

change state to PENDING

changed

return data to client

OperationDownloadSetState transaction
with COMPLETED state

transaction script
for OperationDownloadSetState

change state to COMPLETED

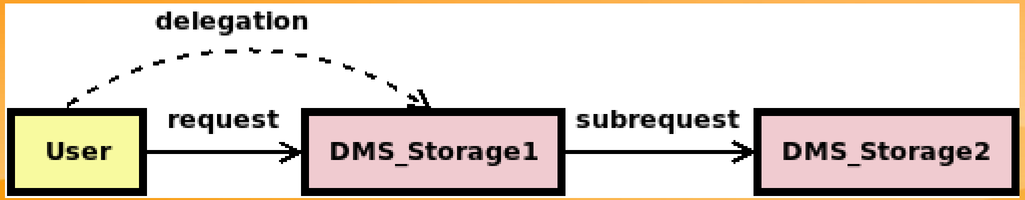
changed

Client
Server
Blockchain

Consensus in Hyperledger Fabric/ProvHL

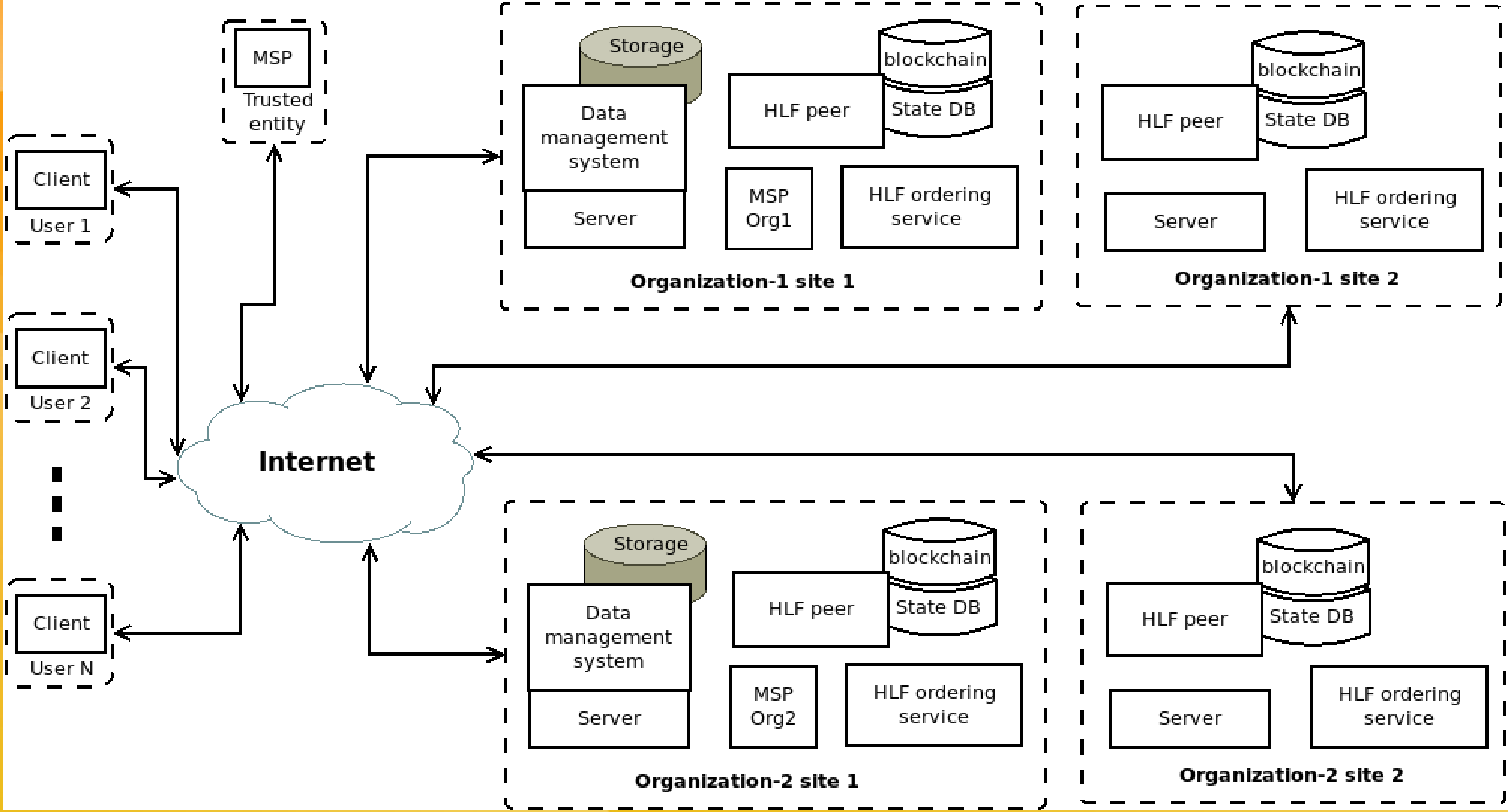
- Transaction **endorsement**: endorses the transactions by simulating the transaction execution process
- **Ordering**: set of ordering services take endorsed transactions and decide on a sequence in which the transactions will be written to the ledger
  - Ordering Consensus Algorithms
    - SOLO, Raft, Kafka, BFT,...
- **Validation and commitment**: committing peers first validate the transactions received from the orderers and then commit that transaction to the ledger

Rights Delegation in ProvHL



- Usual proxy-based delegation in DCS: low level of security + central service = point of failure, intrusion and bottleneck
- Due to its distributed nature, the blockchain-based delegation proves to be fully adequate to distributed computing systems.
- The use of smart contracts, in turn, provides flexibility because they allow one to define various conditions for the delegation of rights in DCSs.

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 ⇒ fault-tolerant
  - safe and secure PMD and data management system
  - well granular access control management
    - including delegation of rights
  - testbed performance characteristics are promising