Experiences on the Use of Decentralized Systems for Data Management and Retrieval

Stefano Ferretti

Dept. of Pure and Applied Sciences, University of Urbino

stefano.ferretti@uniurb.it

Outline

- Preliminary overview of some activities
- Smart transportation use case
  (example of data management)
- Can we perform complex queries for data retrieval over DLTs?

Analysis and Simulation of the Blockchain

LUNES-blockchain

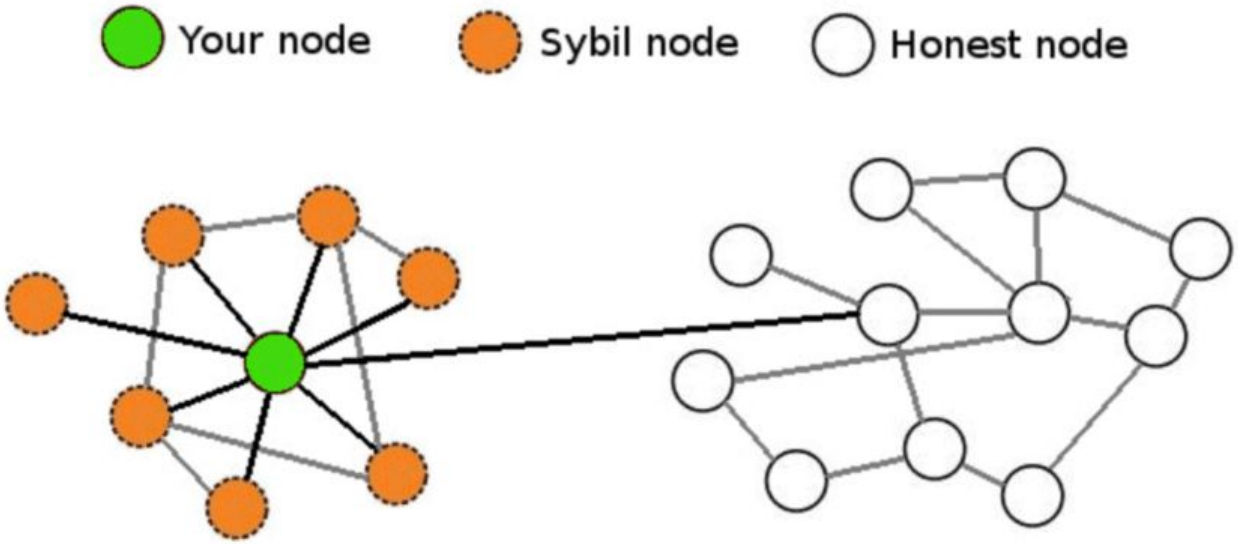
Lunes-blockchain is a discrete events simulator that is able to reproduce the behaviour of a Bitcoin-based blockchain and to simulate certain attacks on the system. It consists of three phases that are executed separately:

- Network Creation
- Simulation Execution
- Attacks Evaluation

Sybil Attack

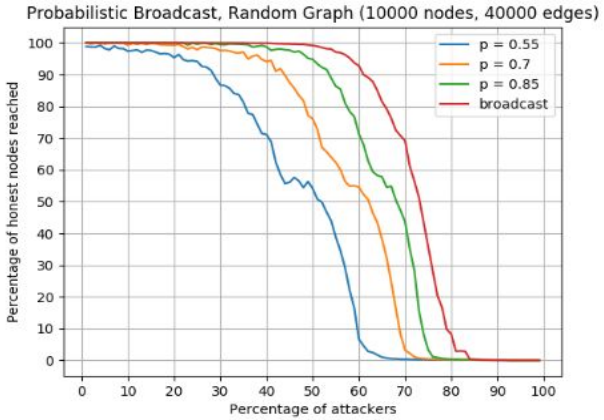
The Sybil Attack is a type of Denial of Service attack where an attacker creates a large number of pseudonymous identities and uses them to gain a disproportionately large influence.

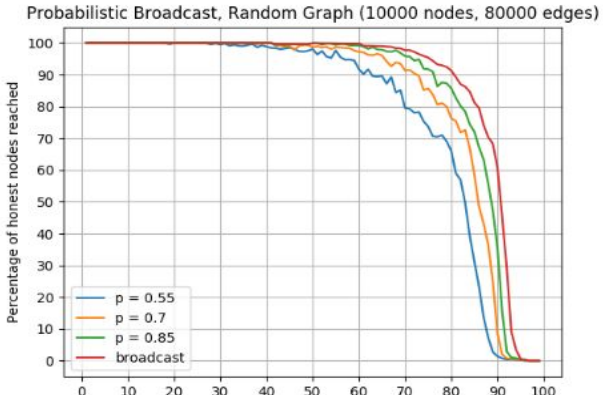
In our case the attacker will not relay the transactions of a certain node.

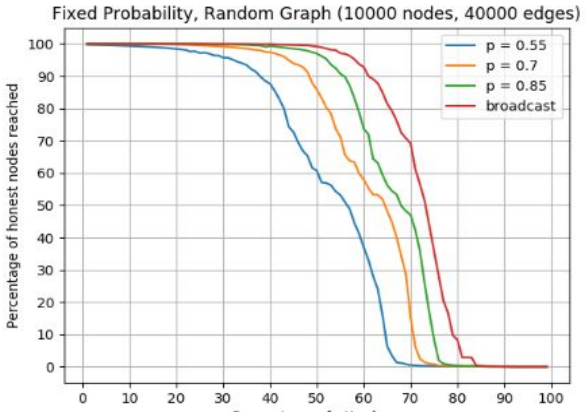


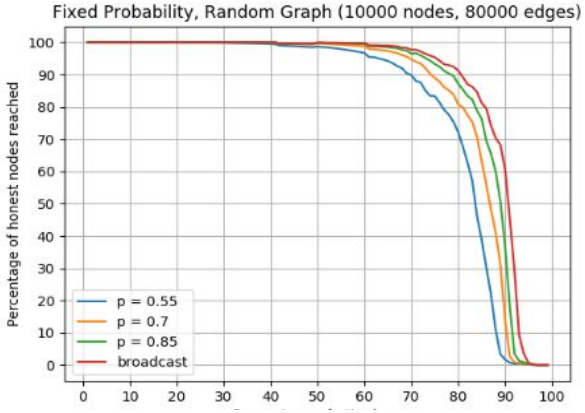
Influence of the Gossip Protocols on the Attack

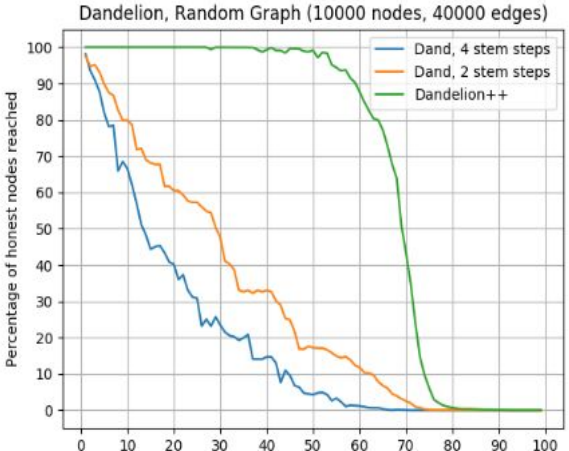
- Fixed Probability
- Probabilistic Broadcast
- Dandelion
- Dandelion ++

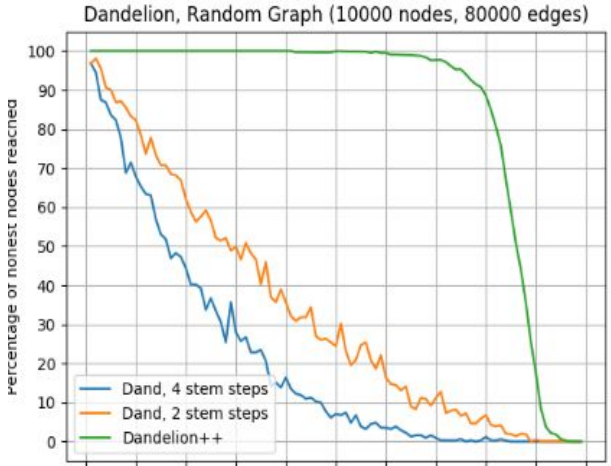

Probabilistic Broadcast, Random Graph (10000 nodes, 40000 edges)
Percentage of honest nodes reached
Percentage of attackers
p = 0.55
p = 0.7
p = 0.85
broadcast


Probabilistic Broadcast, Random Graph (10000 nodes, 80000 edges)
Percentage of honest nodes reached
p = 0.55
p = 0.7
p = 0.85
broadcast


Fixed Probability, Random Graph (10000 nodes, 40000 edges)
Percentage of honest nodes reached
p = 0.55
p = 0.7
p = 0.85
broadcast


Fixed Probability, Random Graph (10000 nodes, 80000 edges)
Percentage of honest nodes reached
p = 0.55
p = 0.7
p = 0.85
broadcast


Dandelion, Random Graph (10000 nodes, 40000 edges)
Percentage of honest nodes reached
Dand, 4 stem steps
Dand, 2 stem steps
Dandelion++


Dandelion, Random Graph (10000 nodes, 80000 edges)
Percentage of honest nodes reached
Dand, 4 stem steps
Dand, 2 stem steps
Dandelion++

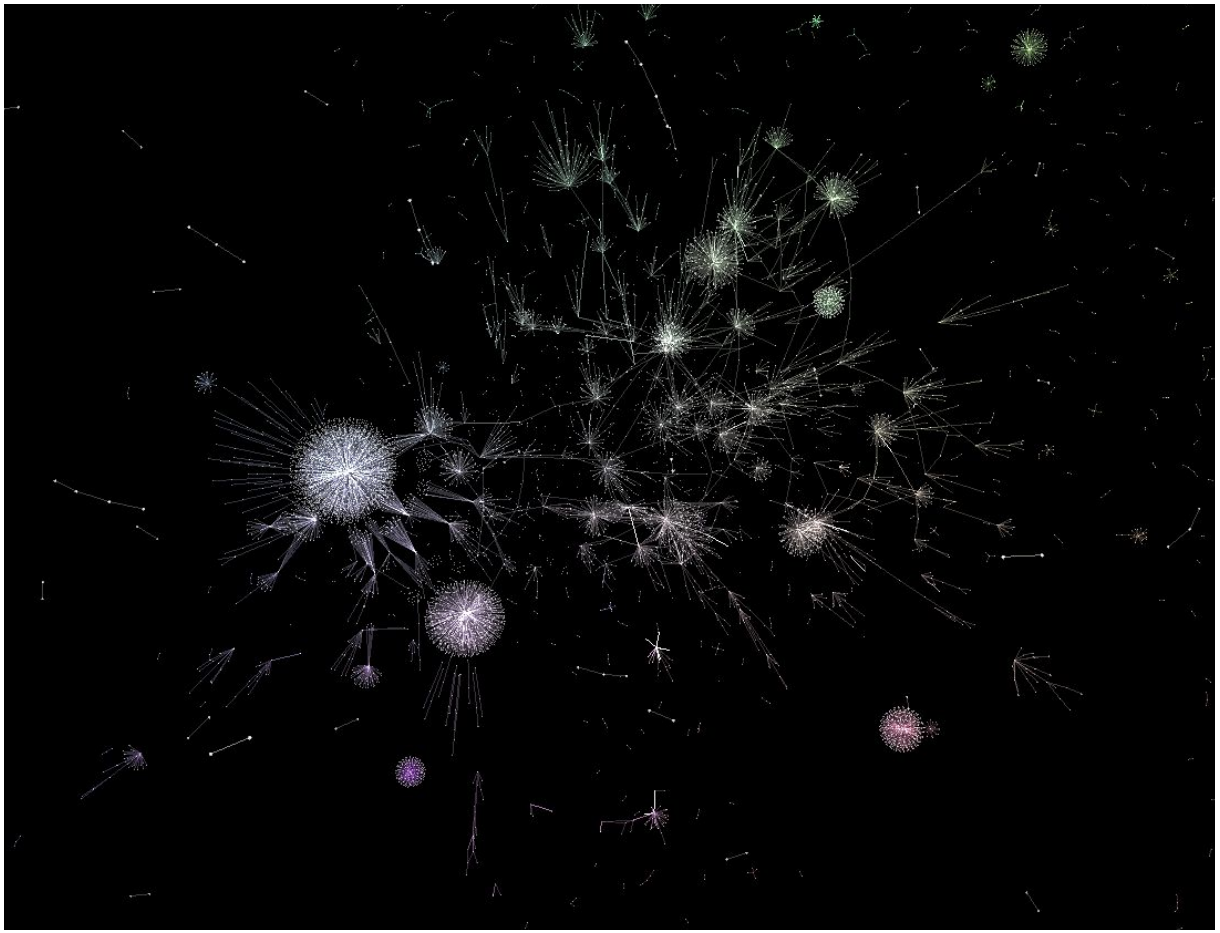
DILENA

DILENA is a software tool for the analysis of the graphs based on networks' transactions. It is structured in two parts:

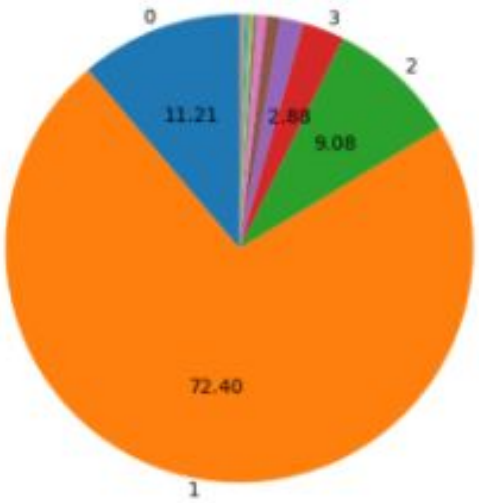
- Graph Generator: the transactions of a certain blockchain referring to a specified period of time are downloaded and the corresponding directed graph is created.

- Graph Analyzer: some metrics are calculated on the graph, in order to determine whether it has small world properties or it doesn't.

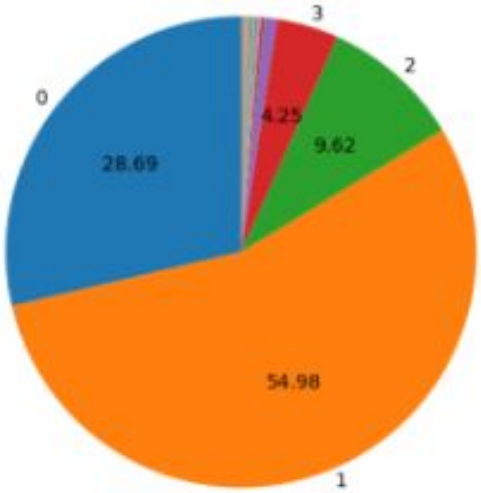
The blockchain as a network



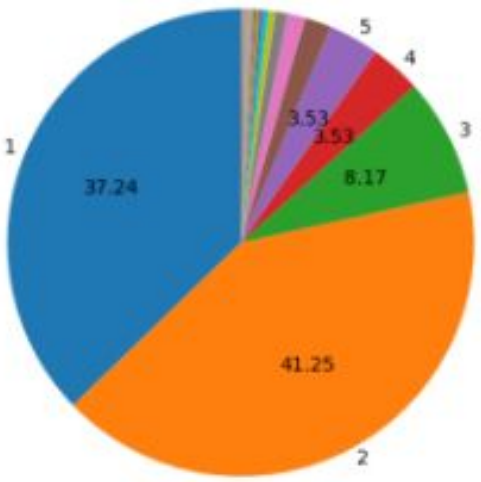
Ethereum Degree Distribution



(a) Ethereum in degree distribution



(b) Ethereum out degree distribution



(c) Ethereum total degree distribution

The node with the highest degree showed an amount of connections with almost the 10% of the node set.

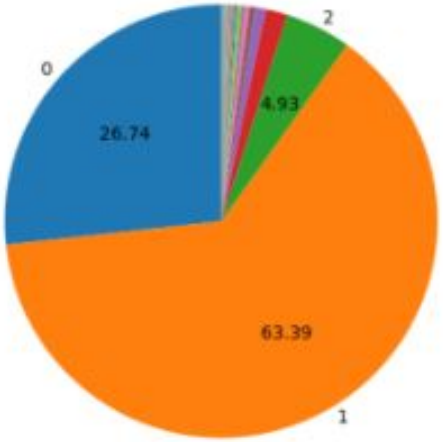
Around 10 nodes with a degree higher than 2000

Metrics on Ethereum

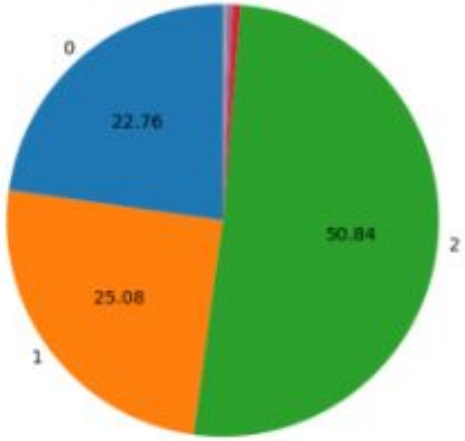
Graph	Graph ACC	Main Component ASPL	Main Component ACC
Ethereum	0.02099	1.4256	0.02134
Random	0.000014	10.3584	0.000015

- The ratio of the average clustering coefficient between the Ethereum and the random generated graph is 1469
- The ratio of the average shortest path length between the Ethereum and the random generated graph is 0.14

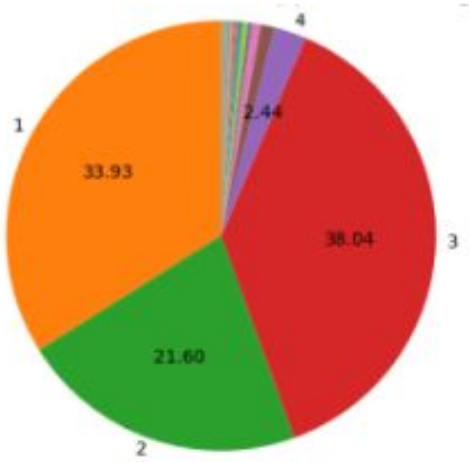
Bitcoin Degree Distribution



(a) Bitcoin in degree distribution



(b) Bitcoin out degree distribution



(c) Bitcoin total degree distribution

Almost 1/2 of the nodes has either 0 in-degree or 0 out-degree

Few nodes with a very high degree, acting as hubs of the network

Metrics on Bitcoin

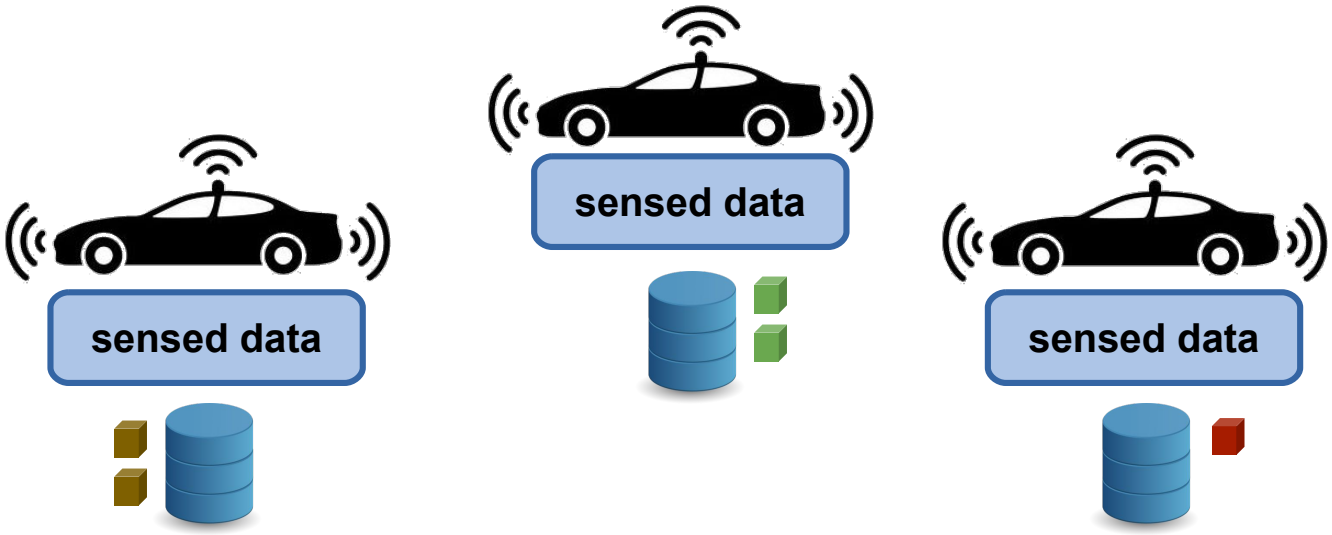
Graph	Graph ACC	Main Component ASPL	Main Component ACC
Bitcoin	0.0235	190.4879	0.024
Random	0.000026	6.461	0.000029

- The ratio of the average clustering coefficient between Bitcoin and the random generated graph is 828

- The ratio of the average shortest path length between Bitcoin and the random generated graph is 29.5

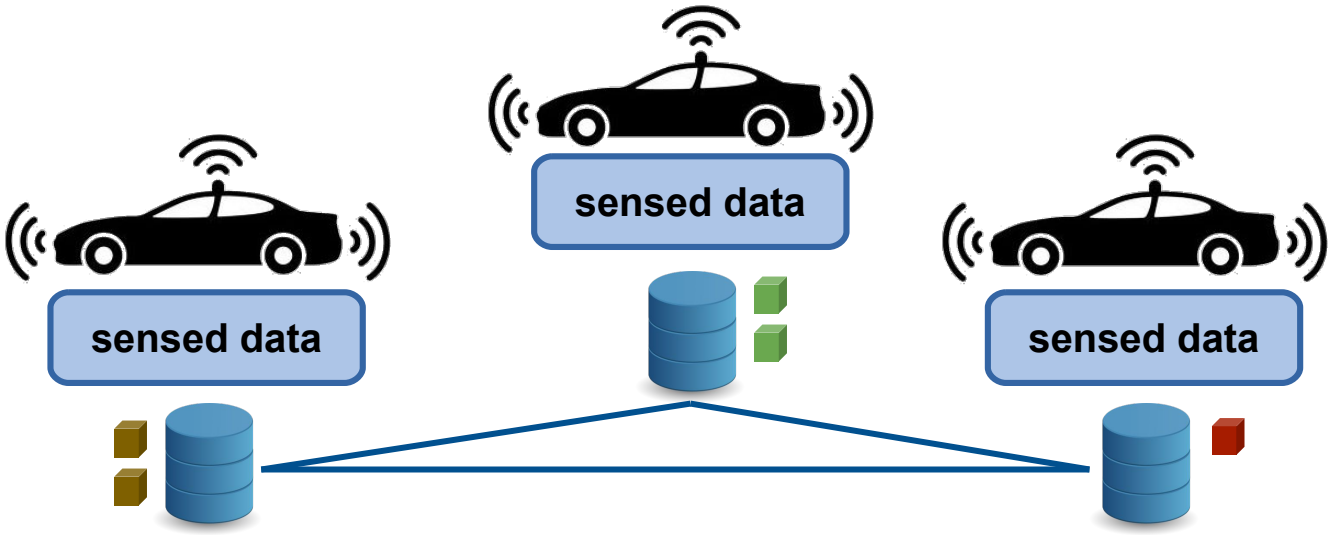
Let me switch to a use case on data management …
Smart transportation

Smart Transportation Systems



Personal data

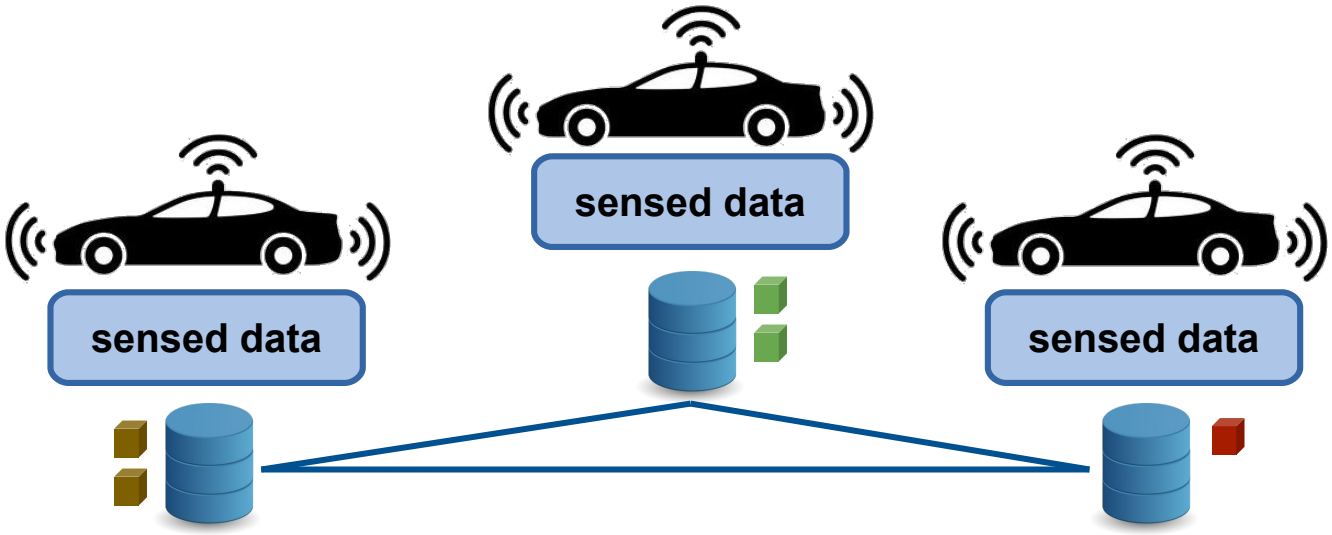
Smart Transportation Systems



Data storage

Personal data

Opt1: central entity maintains crowdsourced data

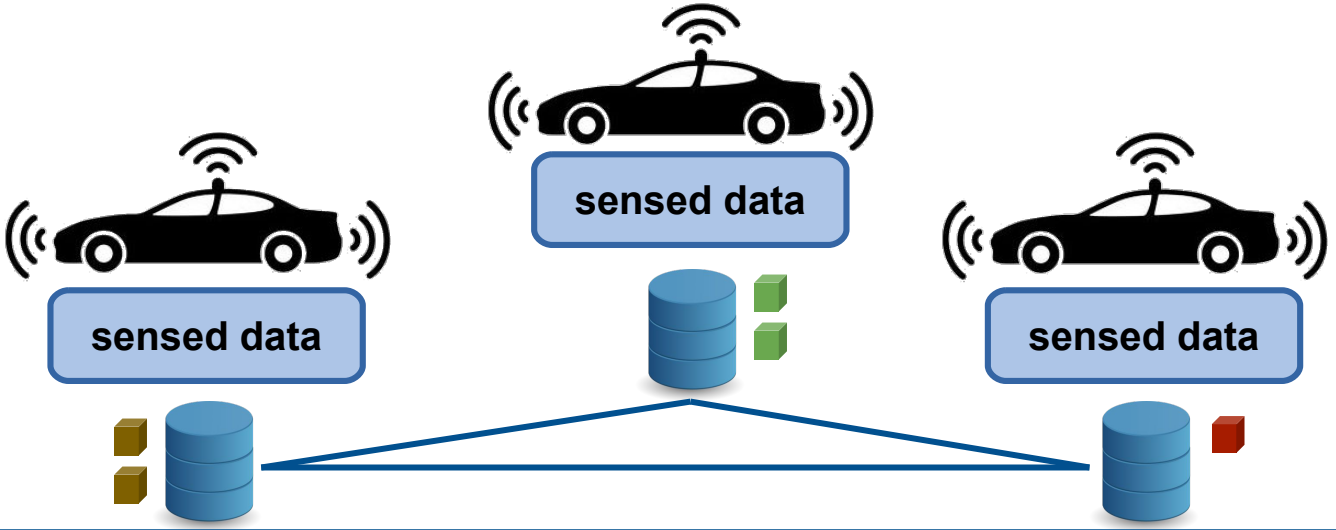


Users lose the sovereignty over their data
- data controller gains all the data
- data controller can alter the data
- users must rely on the controller

Data storage

Personal data

Opt2: keep the data locally and distribute upon request



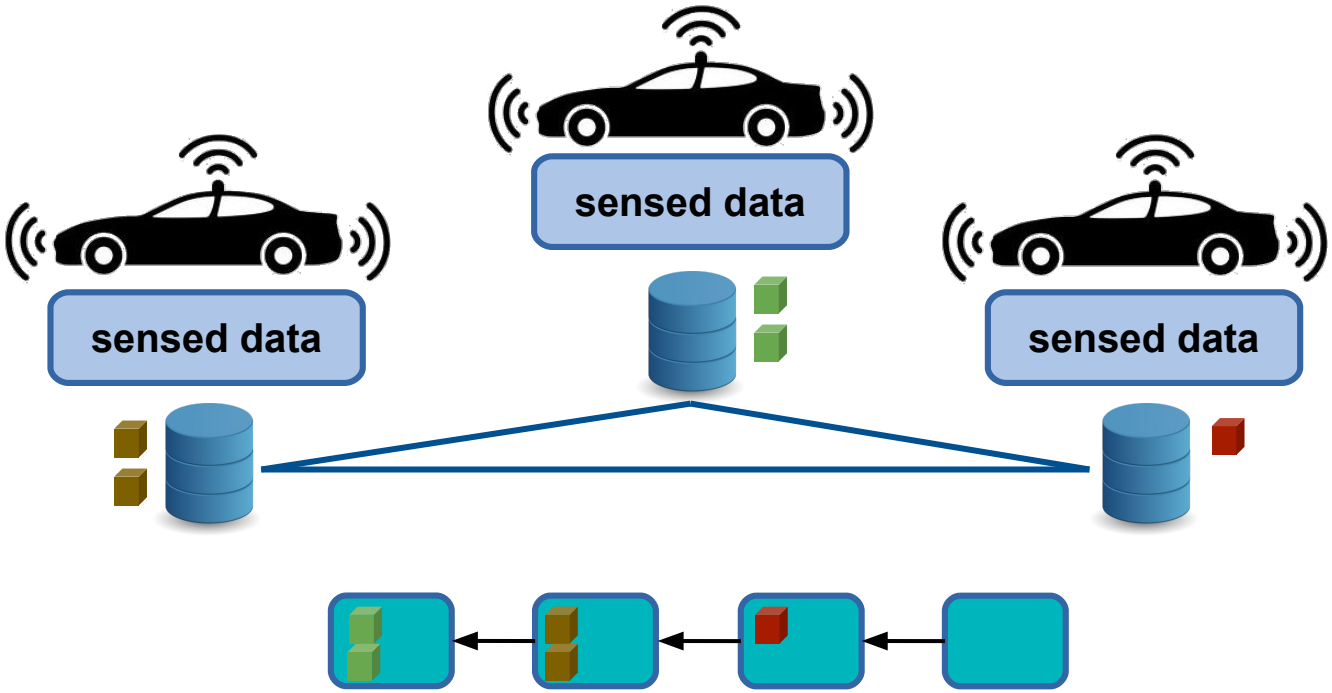
Pros
- you maintain your own data

Cons
- not practical, you need to be always reachable
- storage, computation, communication capabilities needed

Data storage

Personal data

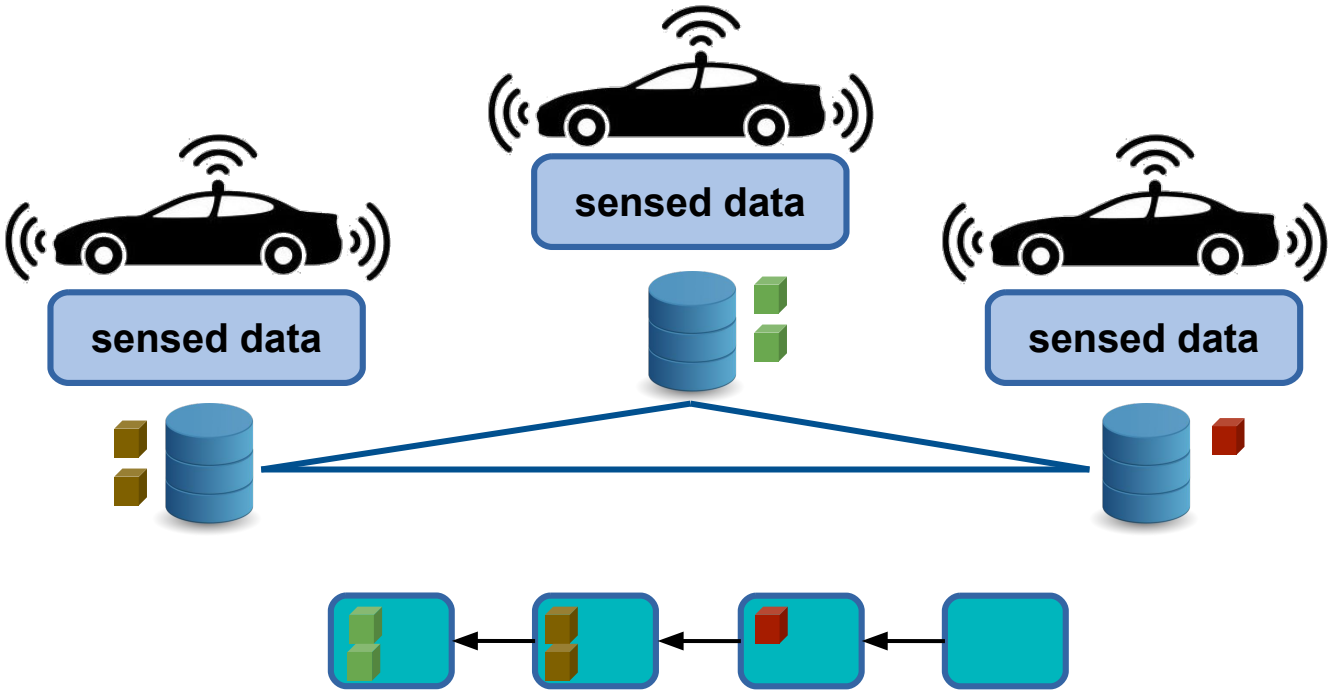
Opt3: use a ledger to register data



Data storage

Personal data

Opt3: use a ledger to register data

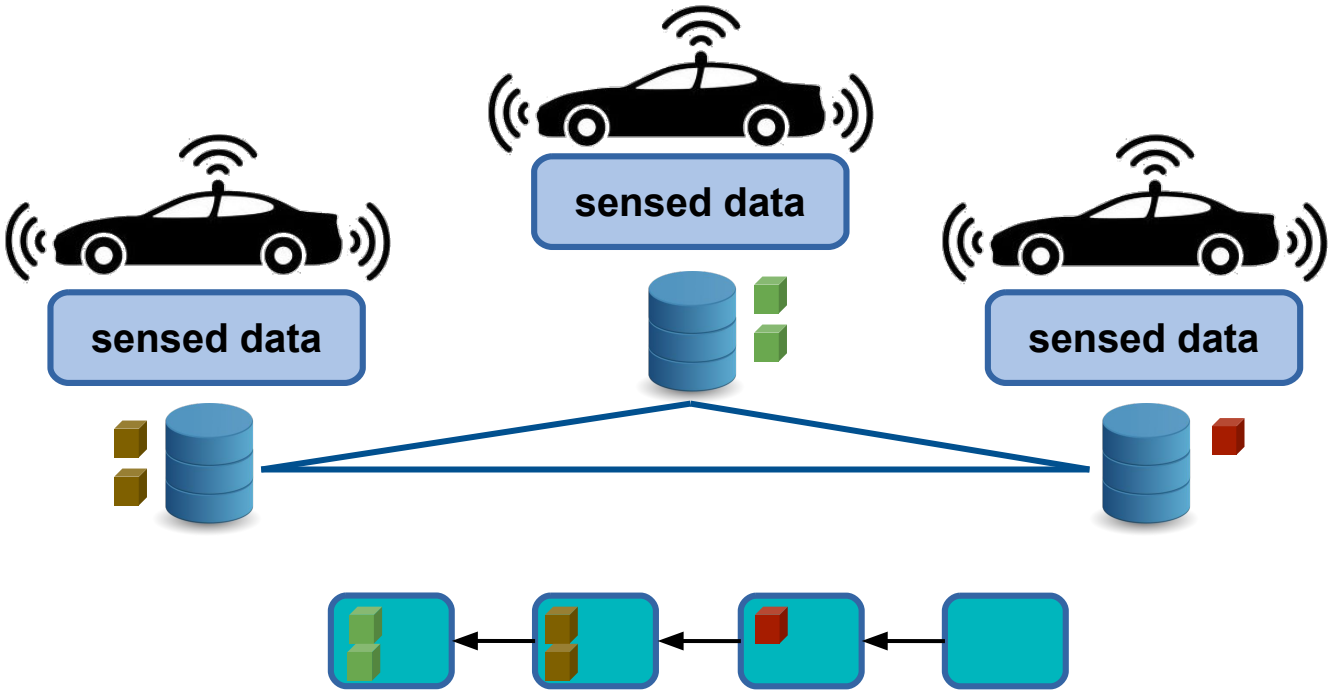


Pros:
- rather simple
- data integrity
- traceability

Data storage

Personal data

Opt3: use a ledger to register data



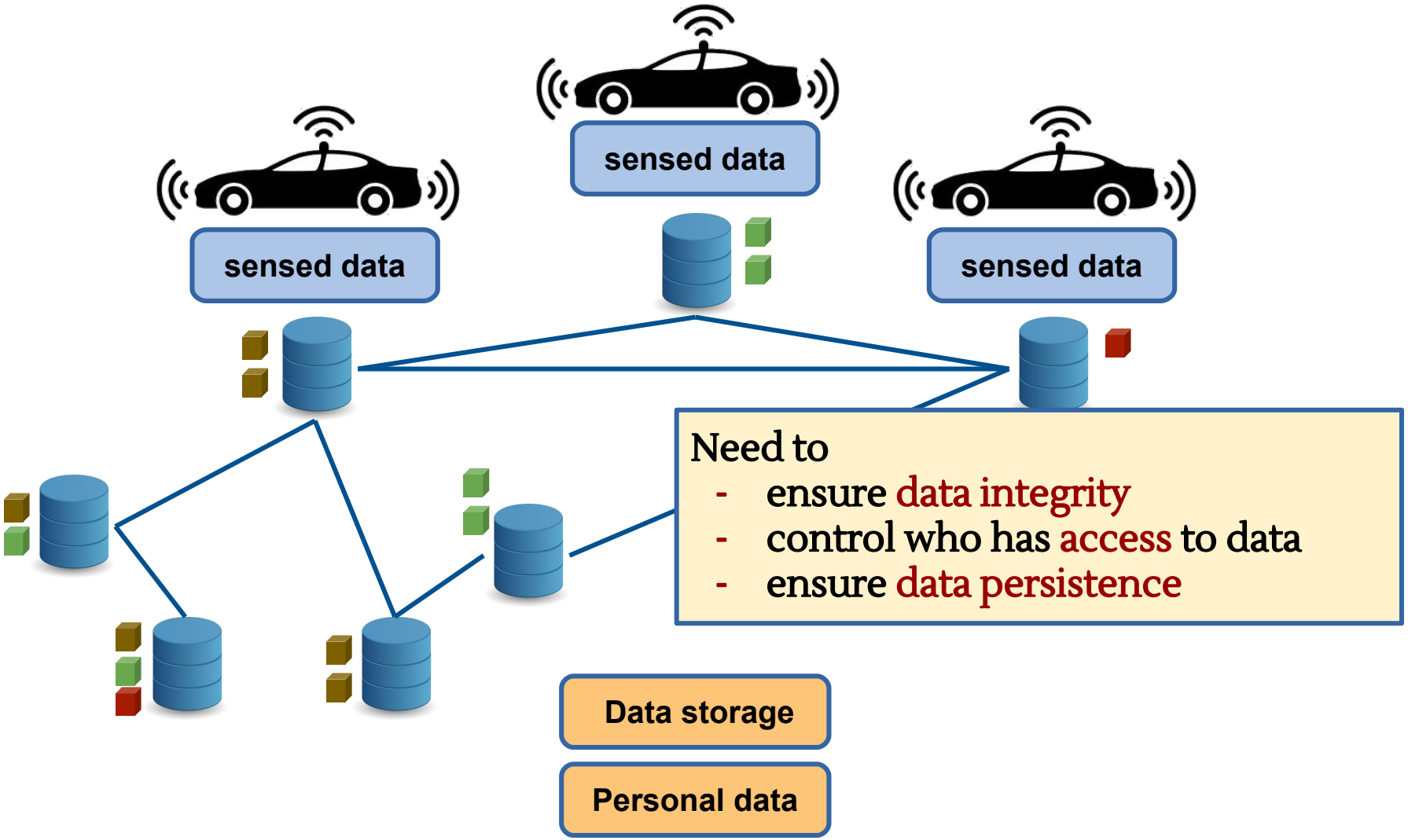
Pros:
- rather simple
- data integrity
- traceability

Data storage

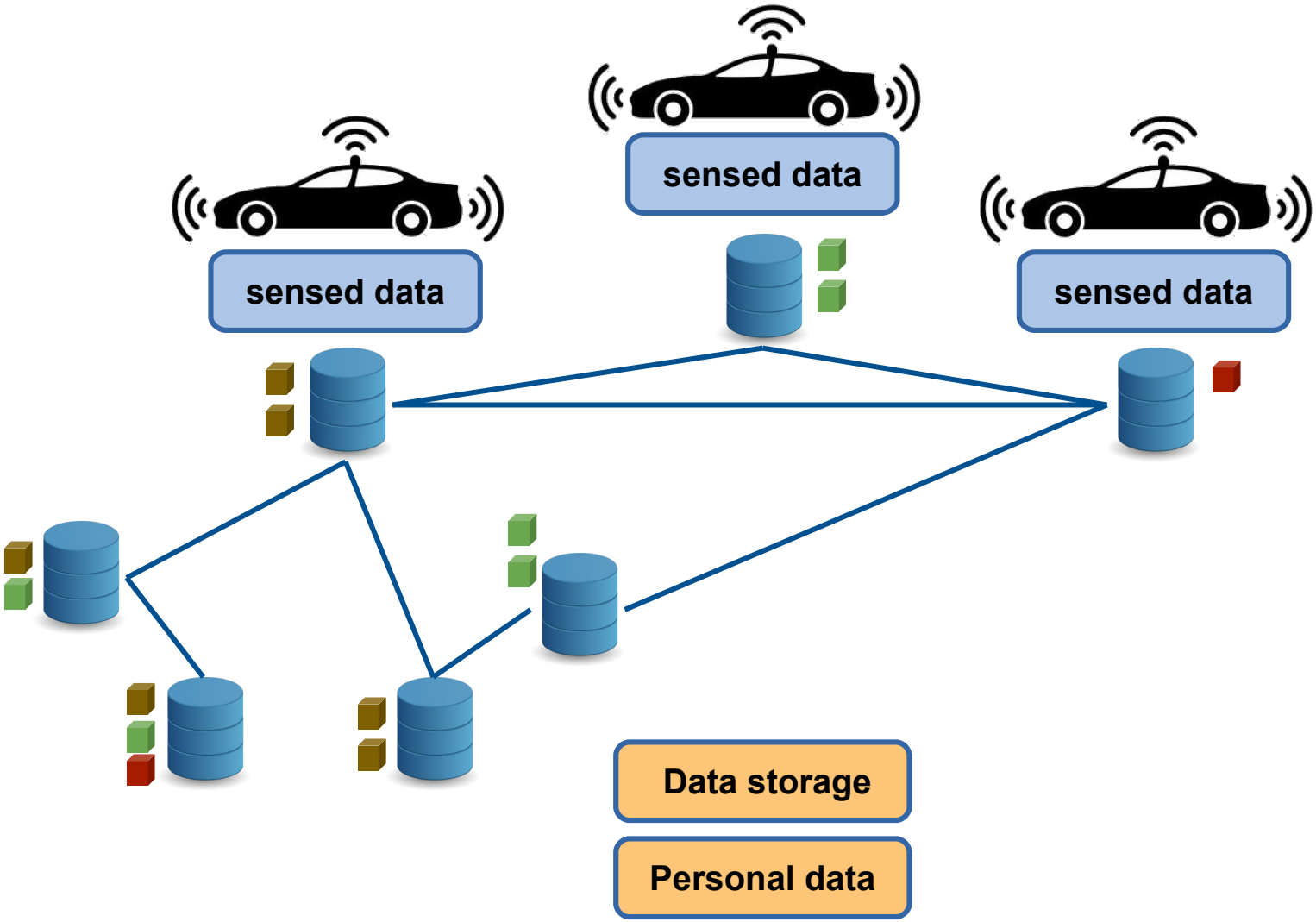
Personal data

Cons:
- ok with small sized data, only
- no right to be forgotten/rectified
- latencies

Opt4: use decentralized file systems for crowdsourced data



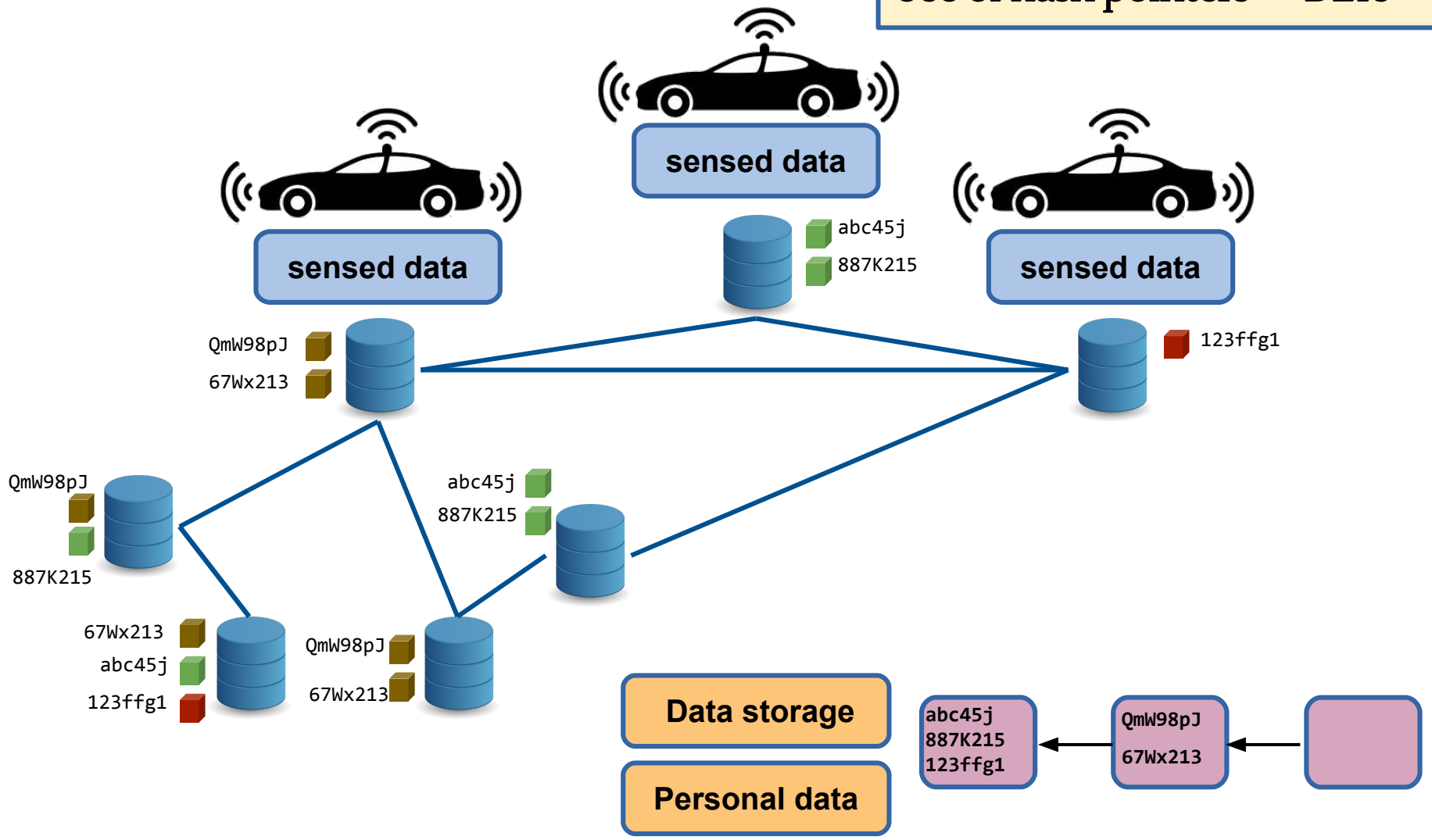
Data integrity



Data integrity

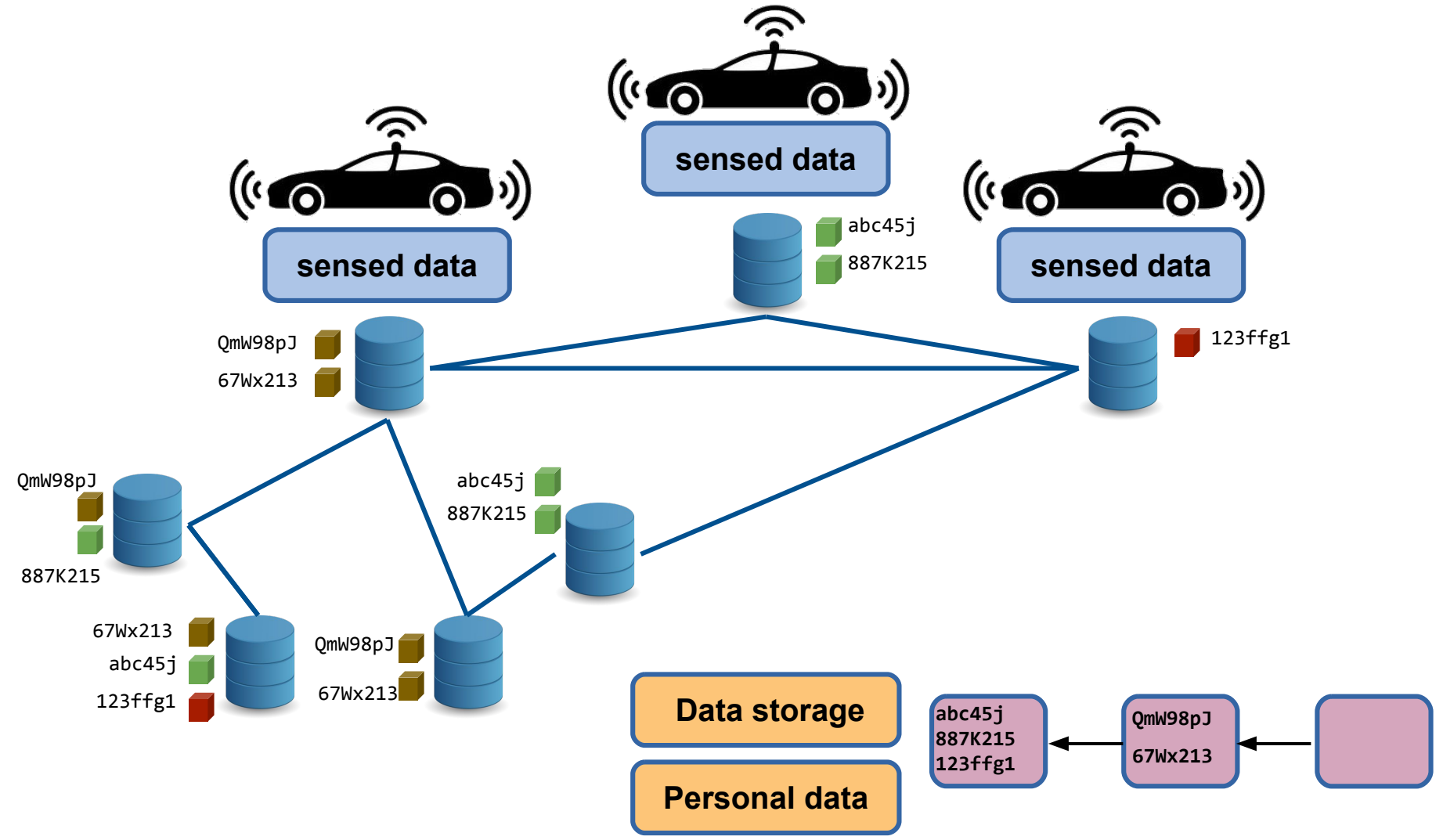
Content based addressing
(instead of location based)

Use of hash pointers → DLTs

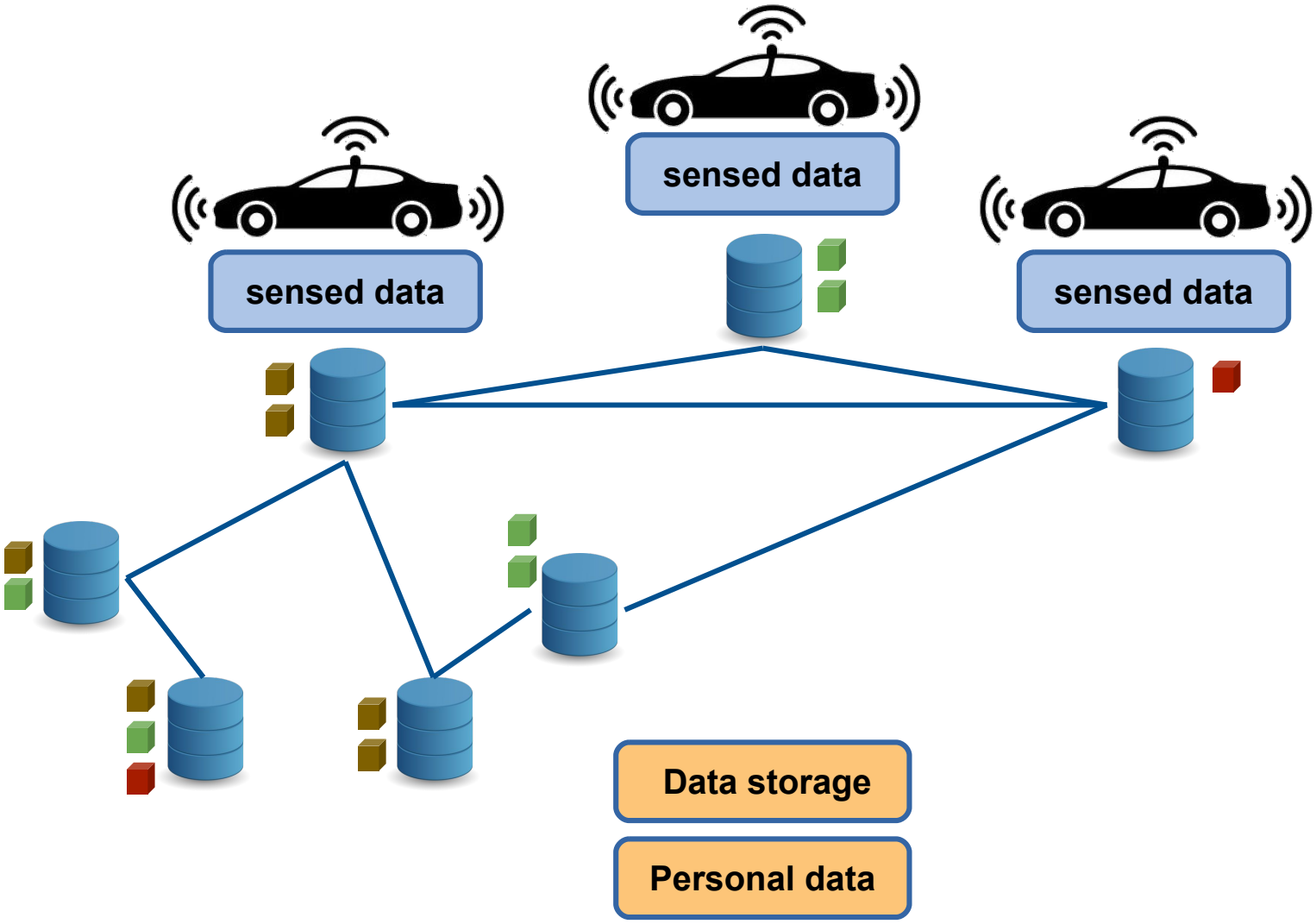


Data integrity

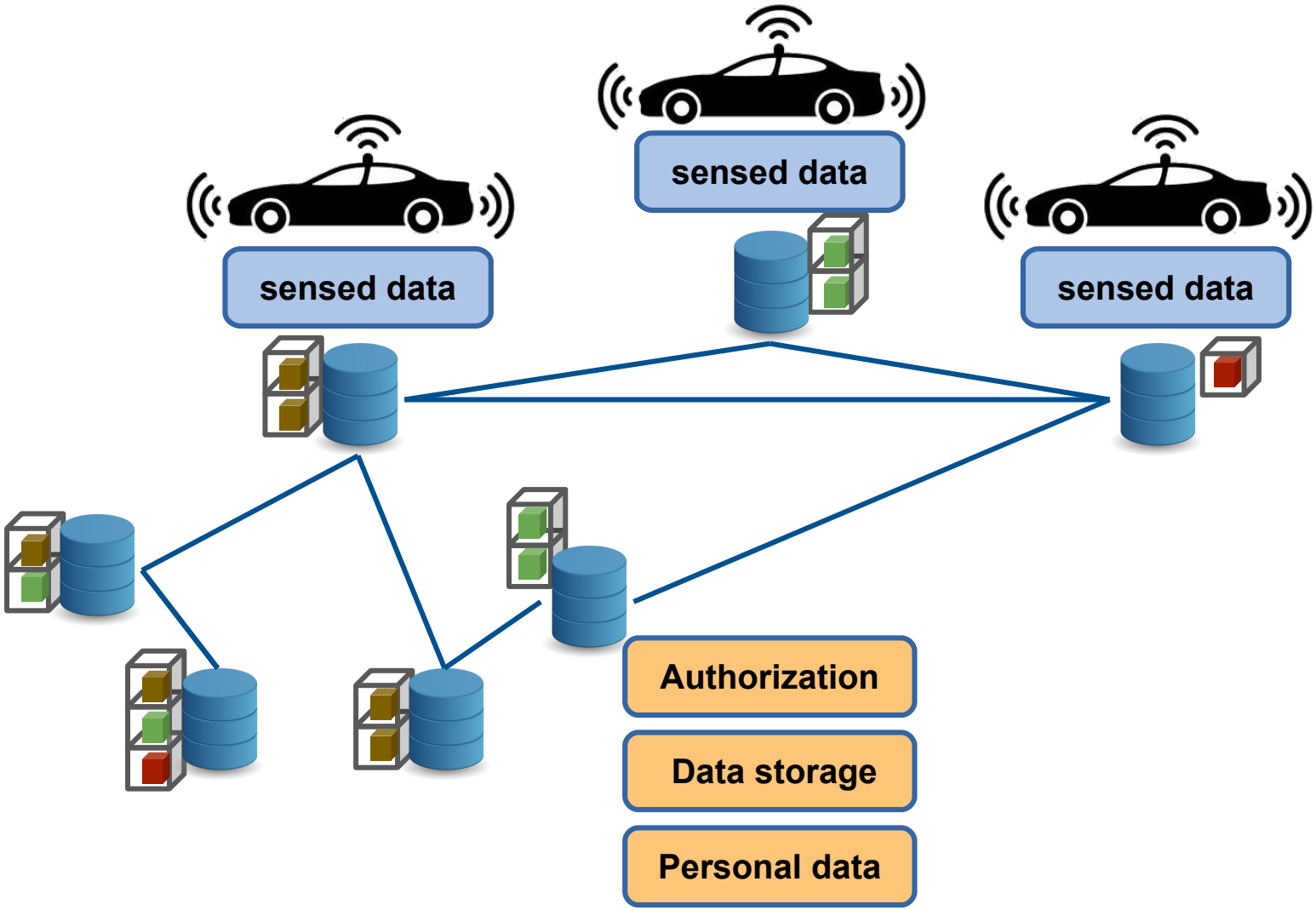
- Possible to remove/modify data
- Fast data upload to DFS
- Latencies to upload hashes less problematic



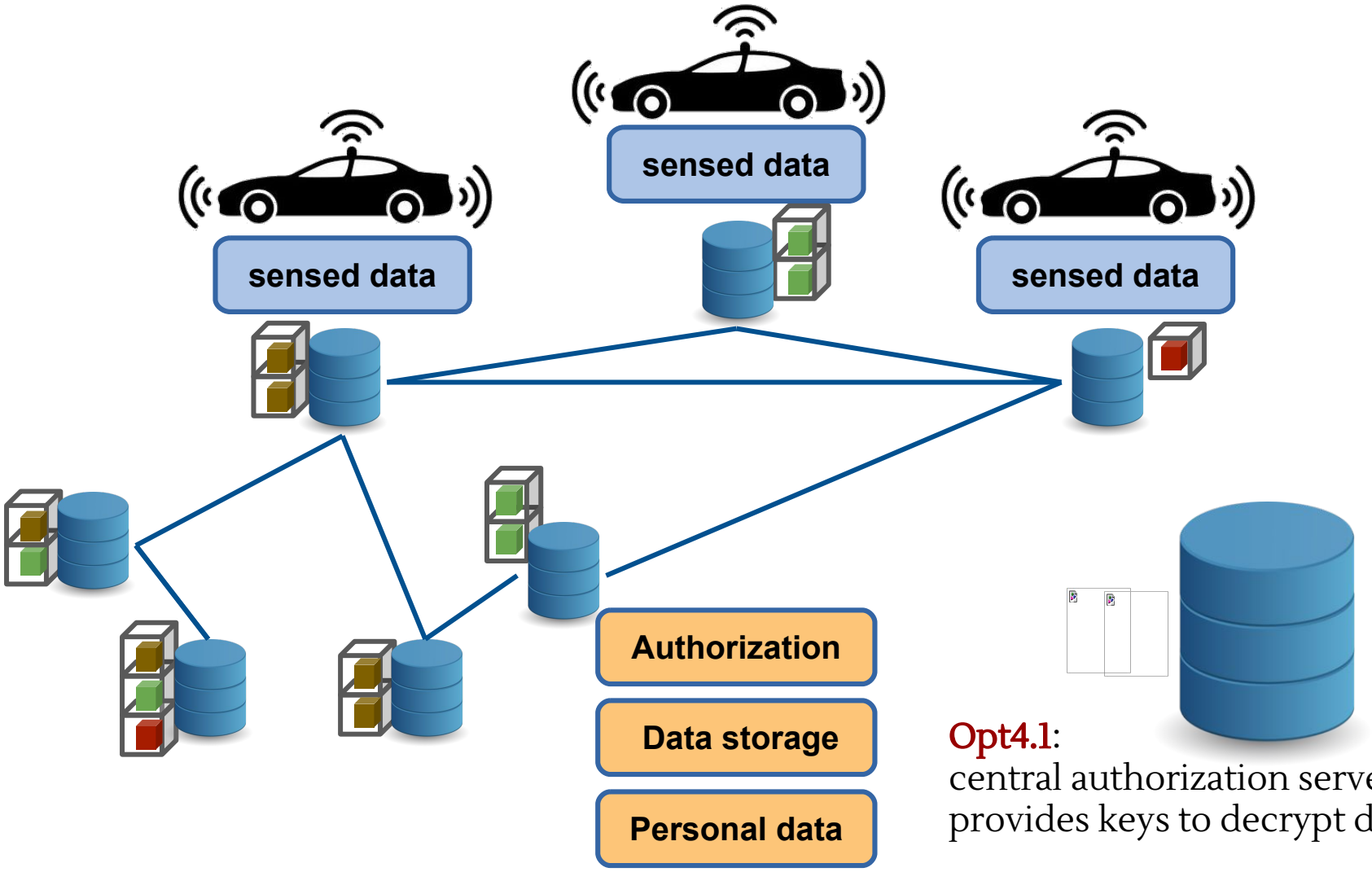
Access Control?



Access Control: DFS + Encryption

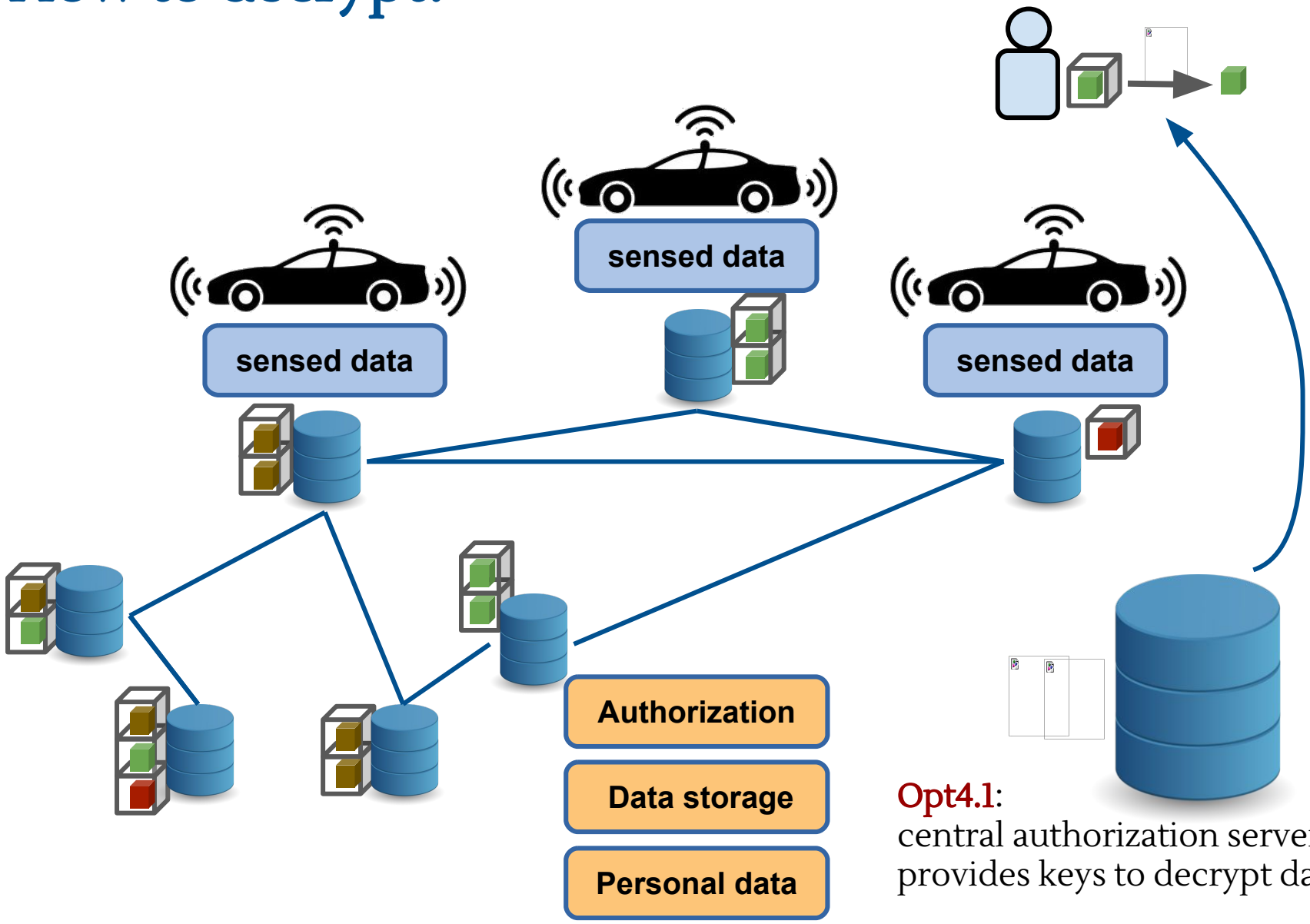


How to decrypt?



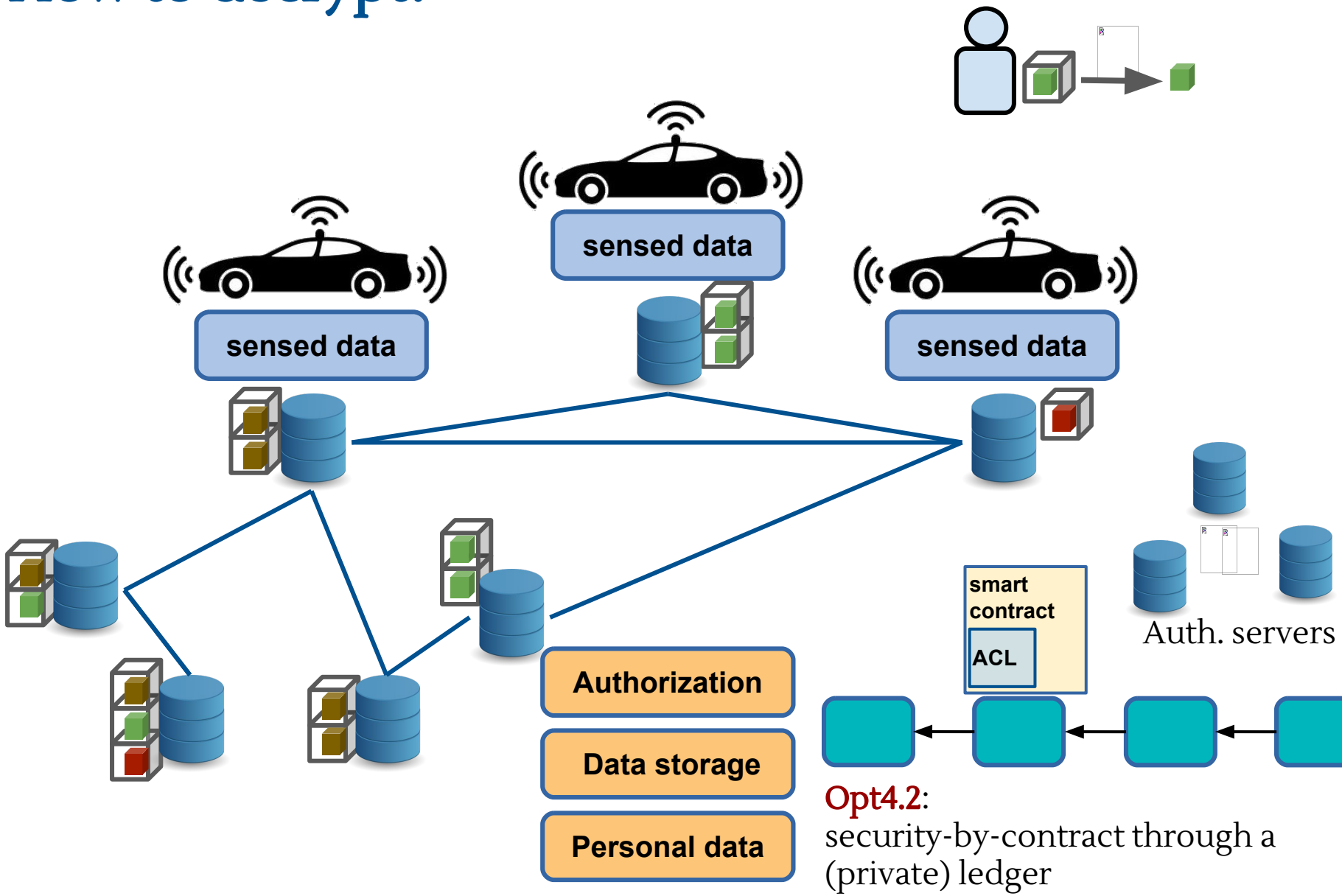
Opt4.1:
central authorization server that provides keys to decrypt data

How to decrypt?



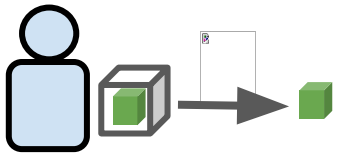
Opt4.1:
central authorization server that provides keys to decrypt data

How to decrypt?

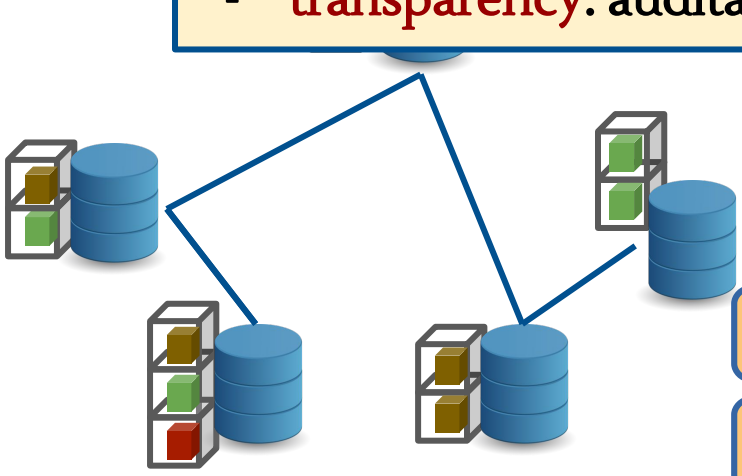


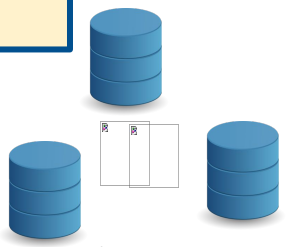
Opt4.2:
security-by-contract through a (private) ledger

How to decrypt?

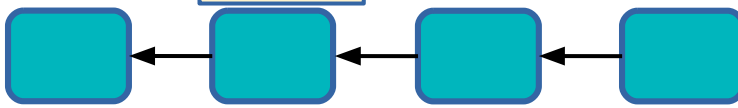


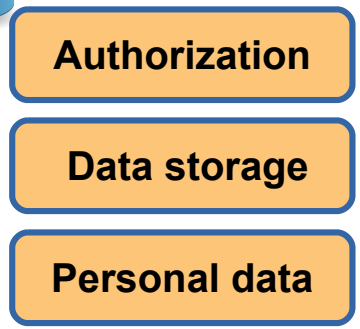
Benefits:
- decentralization of keys custody
  - no single point of failure
  - mitigates privacy leakage
- transparency: auditability of access permissions to data





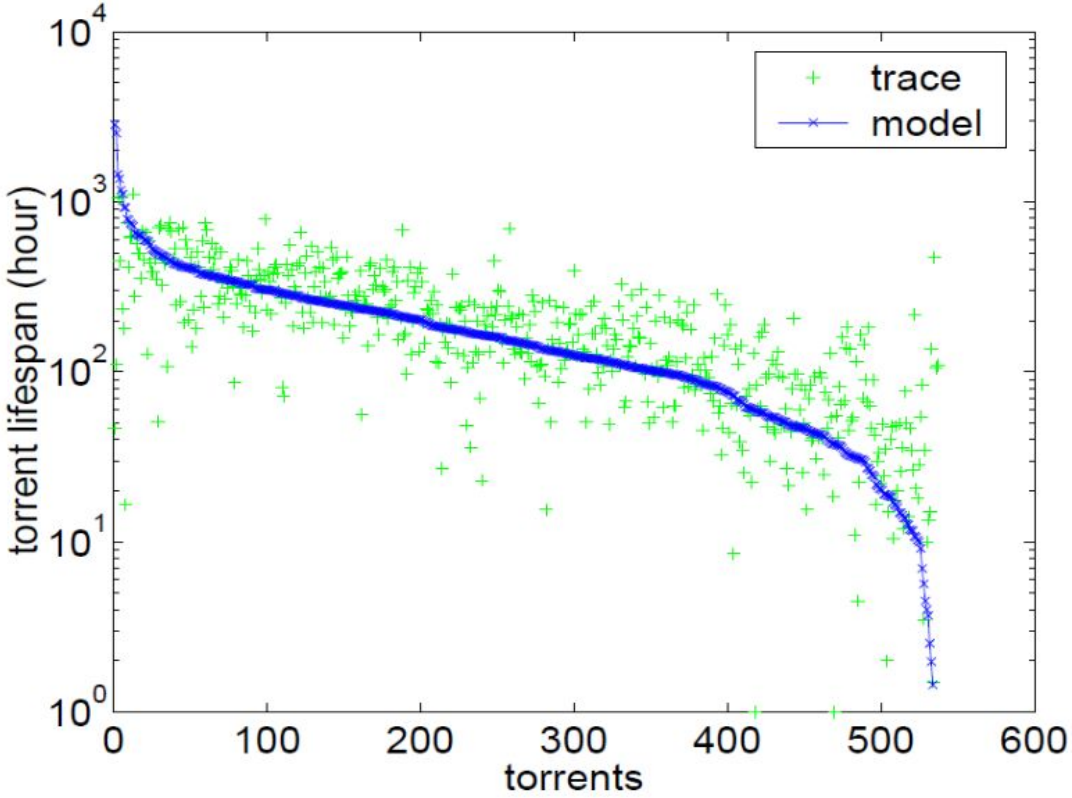
Auth. servers





Opt4.2:
security-by-contract through a (private) ledger

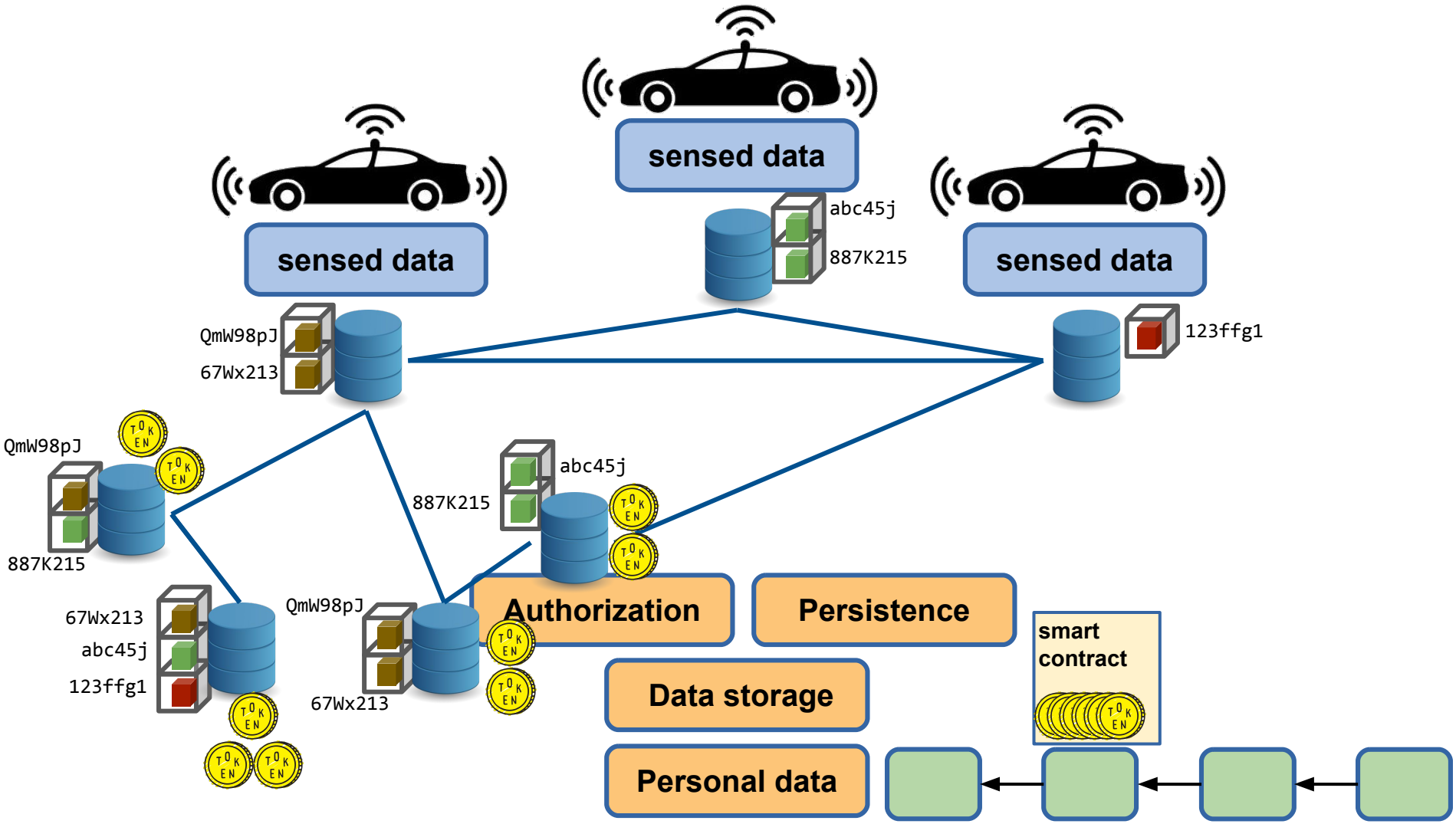
Data persistence: we'd like to avoid this



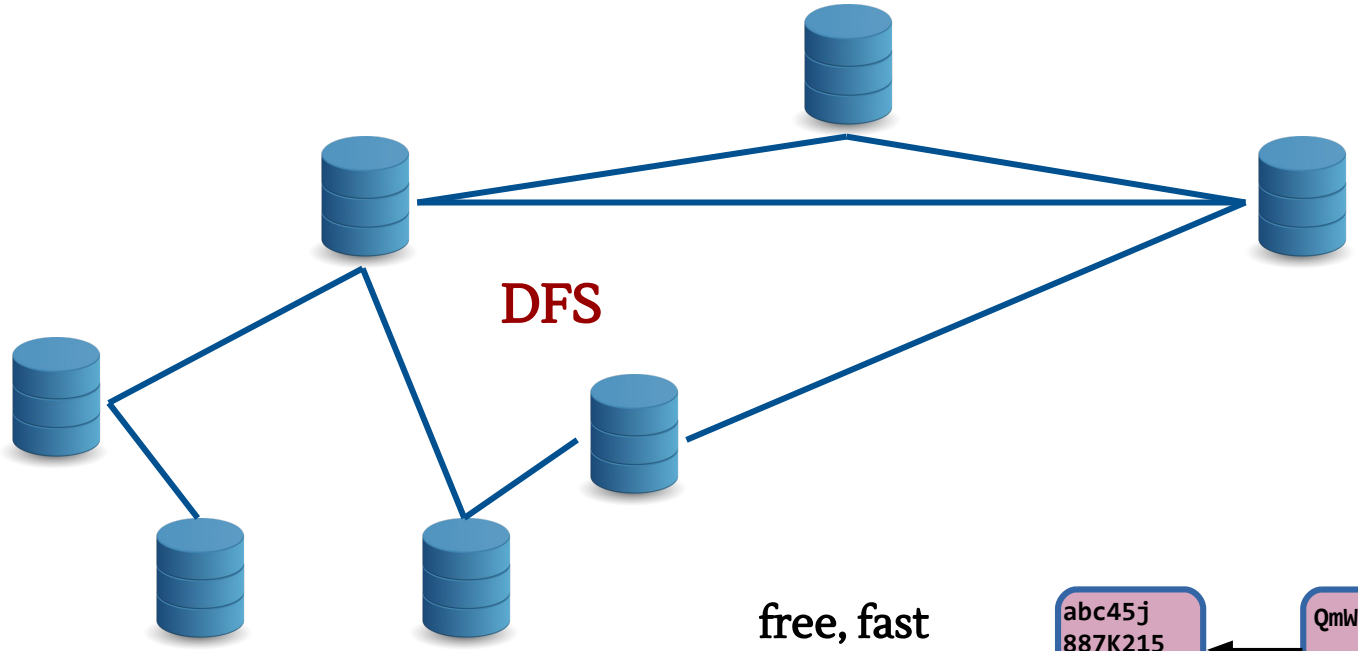
Guo, Lei & Chen, Songqing & Xiao, Zhen & Tan, Enhua & Ding, Xiaoning & Zhang, Xiaodong. (2007). A performance study of BitTorrent-like peer-to-peer systems. Selected Areas in Communications, IEEE Journal on. 25. 155 - 169. 10.1109/JSAC.2007.070116.

Data persistence

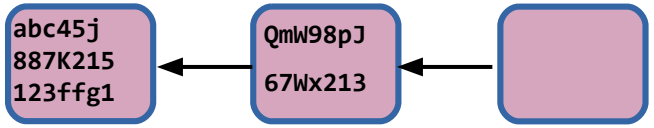
Incentives to cooperate:
blockchain based tokens
E.g. Filecoin, Sia, Storj, Swarm (??)



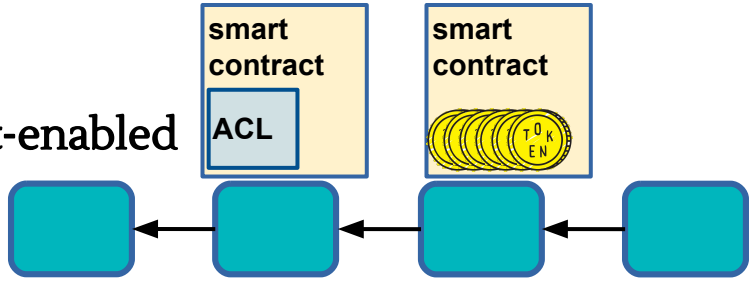
The overall system



free, fast
DLT



smart contract-enabled
DLT



"Does it work?"
"Yes!"

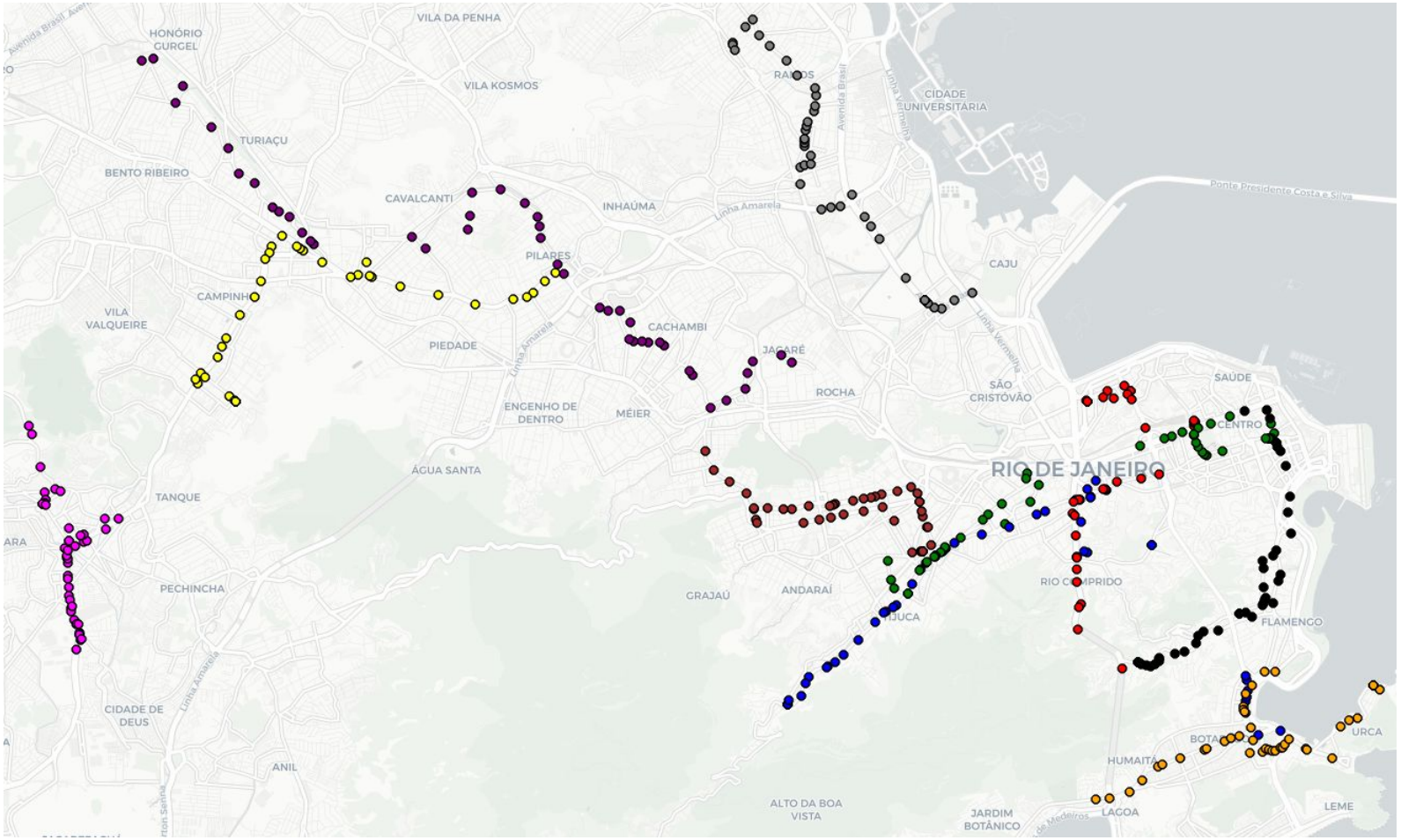
"Does it scale?"
"..."

Some tests on **data upload** to the decentralized system

- the most tricky aspect in this sense, probably

ITS Real Mobility Traces

Dataset of real mobility traces of buses in Rio de Janeiro (Brasil)

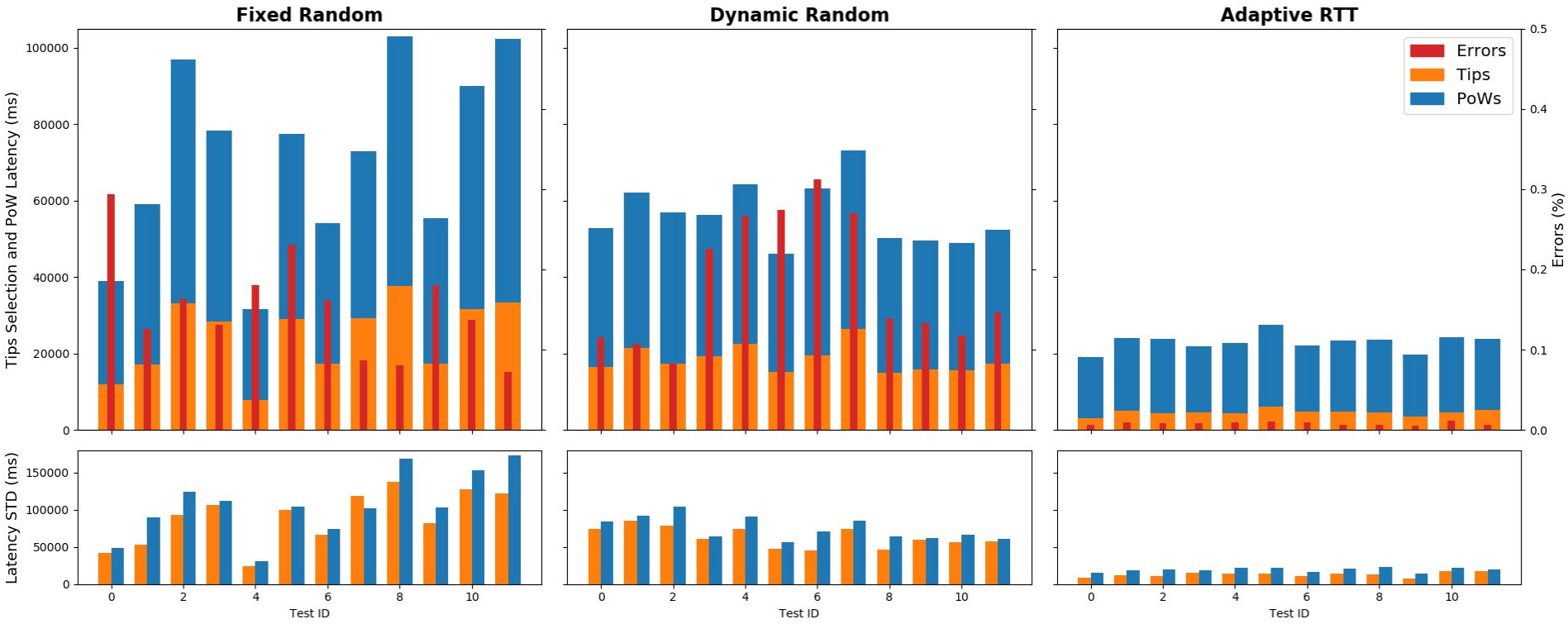


M .Zichichi, S. Ferretti, G. D'Angelo, "A Framework based on Distributed Ledger Technologies for Data Management and Services in Intelligent Transportation Systems", *IEEE Access*, IEEE, May 2020

What we tested

- "Opt3" → DLT:
  - **IOTA**
    - in this context, should perform better than traditional blockchains
- "Opt4" → DFS:
  - **IPFS**: https://ipfs.io/
  - **Sia**: https://sia.tech/

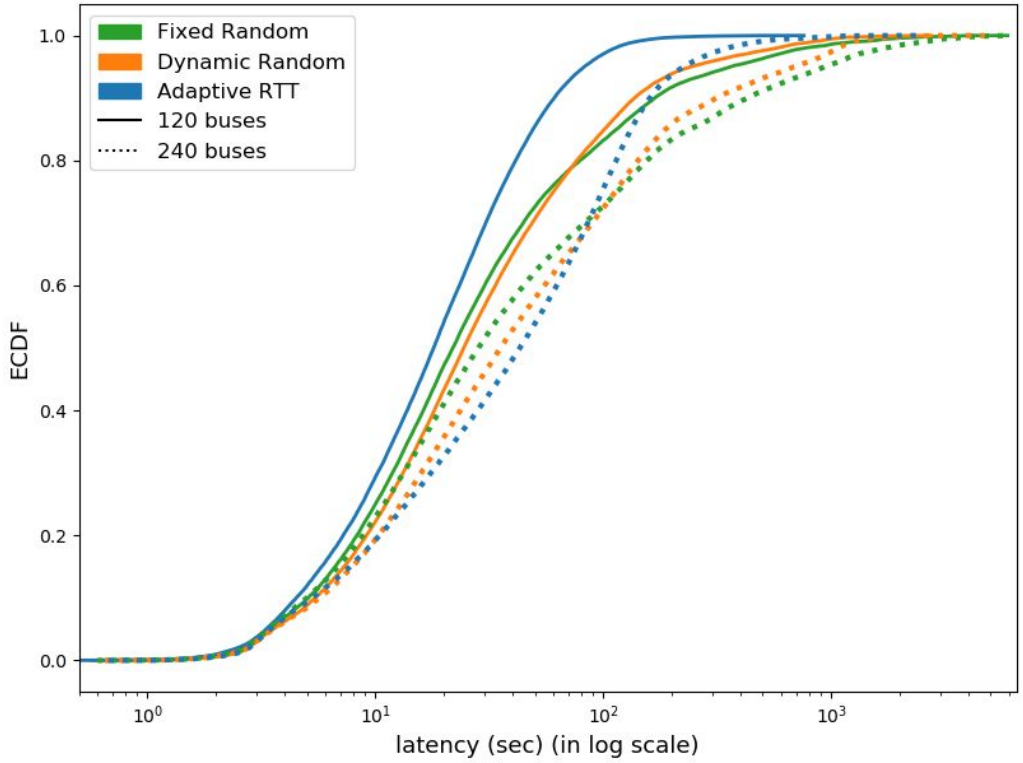
IOTA Results (data upload to DLT)



- Through a proper selection of full nodes it is possible to achieve reliable ledger updates (low errors)
- However, the measured latencies are relevant

M .Zichichi, S. Ferretti, G. D'Angelo, "A Framework based on Distributed Ledger Technologies for Data Management and Services in Intelligent Transportation Systems", *IEEE Access*, IEEE, May 2020

IOTA Results (data upload to DLT)

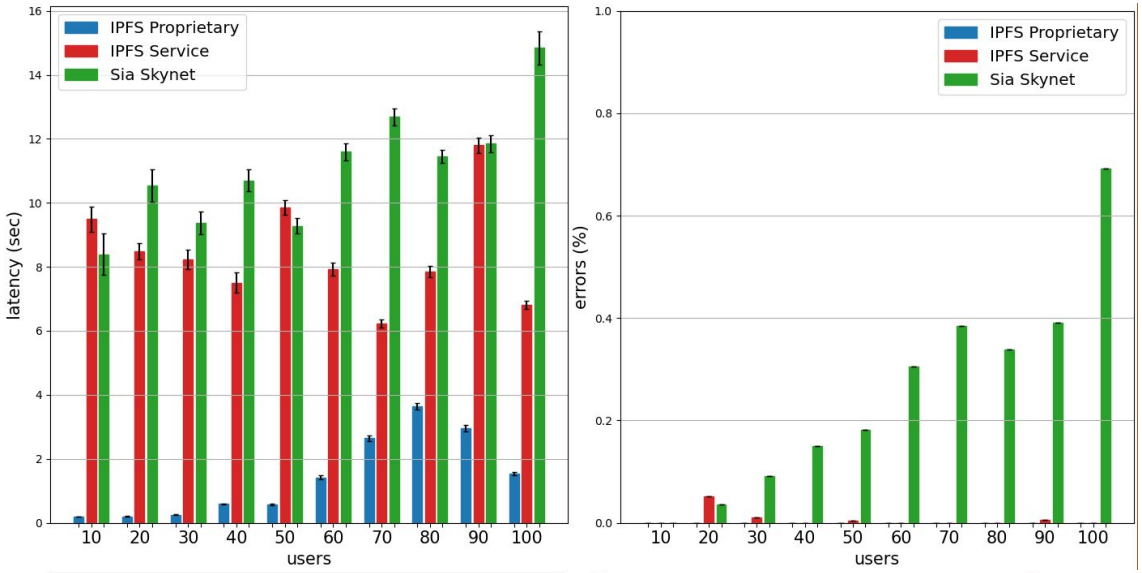


- Through a proper selection of full nodes it is possible to achieve reliable ledger updates (low errors)
- However, the measured latencies are relevant

M .Zichichi, S. Ferretti, G. D'Angelo, "A Framework based on Distributed Ledger Technologies for Data Management and Services in Intelligent Transportation Systems", *IEEE Access*, IEEE, May 2020

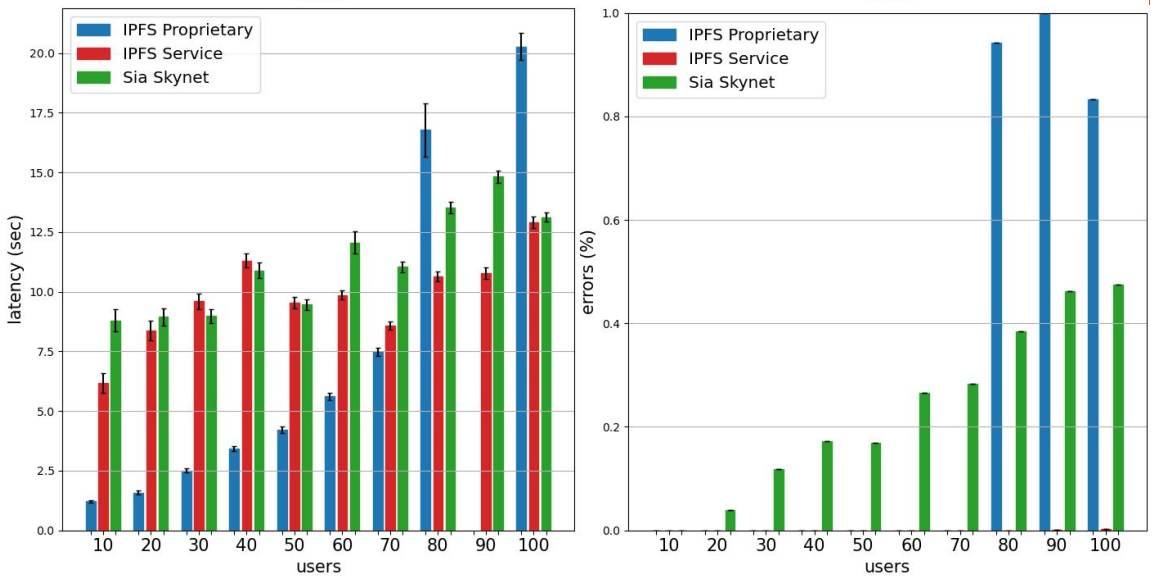
DFS Results

M. Zichichi, S. Ferretti, G. D'Angelo, "On the Efficiency of Decentralized File Storage for Personal Information Management Systems", in *Proc. of 25th IEEE Symposium on Computers and Communications (ISCC)*, July 2020.



small sized JSON data (100B)

```
{
  payload:
    { latitude:'-22.976509',
      longitude: '-43.19902' },
  timestampISO:
      '2020-04-05T14:54:11.288Z'
}
```



larger files (1MB)

Conclusions

- Current decentralized architectures
  - distributed ledger techs (DLT)
  - decentralized file storage (DFS)
  - smart contracts
  - authorization schemes

  allow building reliable and modern services for smart transportation
- Use a layered architecture composed of DFS and DLTs
- Users maintain sovereignty over their data
- **Some concerns**
  - level of scalability and responsiveness
    - need to increase the ability to handle churns
  - be careful while handling sensitive data

Towards Decentralized Complex Queries over Distributed Ledgers

Motivation

- Distributed Ledger Technologies and Decentralized File Storages increasingly used to create common, decentralized and trustless infrastructures
  - high data availability, but also integrity, auditability, confidentiality
  - ability to automate and enforce processes (through smart contracts)

Motivation

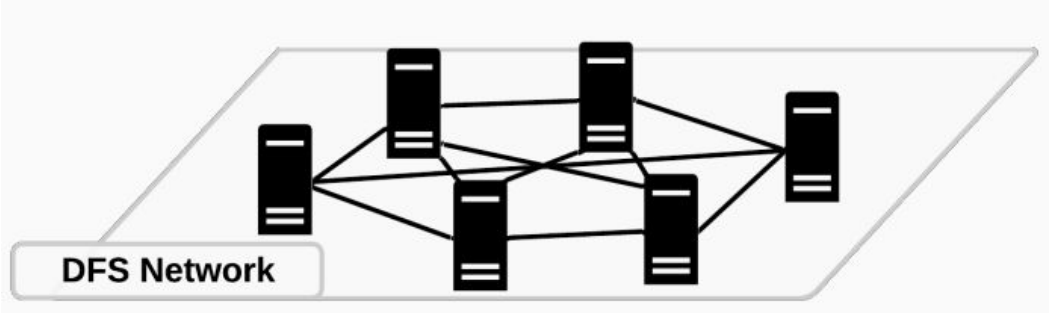
- Distributed Ledger Technologies and Decentralized File Storages increasingly used to create common, decentralized and trustless infrastructures
  - high data availability, but also integrity, auditability, confidentiality
  - ability to automate and enforce processes (through smart contracts)
- Problems
  - data stored in DLTs and DFS are usually unstructured and need to be filtered and indexed before any complex query
  - there are no diffused efficient mechanisms to query a certain type of data, that do not involve centralization (e.g. index data in a central database)

Our work

- Distributed Hash Table (DHT) → distributed data structure that maps "keys" into "values"
- A Decentralized Autonomous Organization (DAO) → smart contracts to manage rewards and organizational decisions

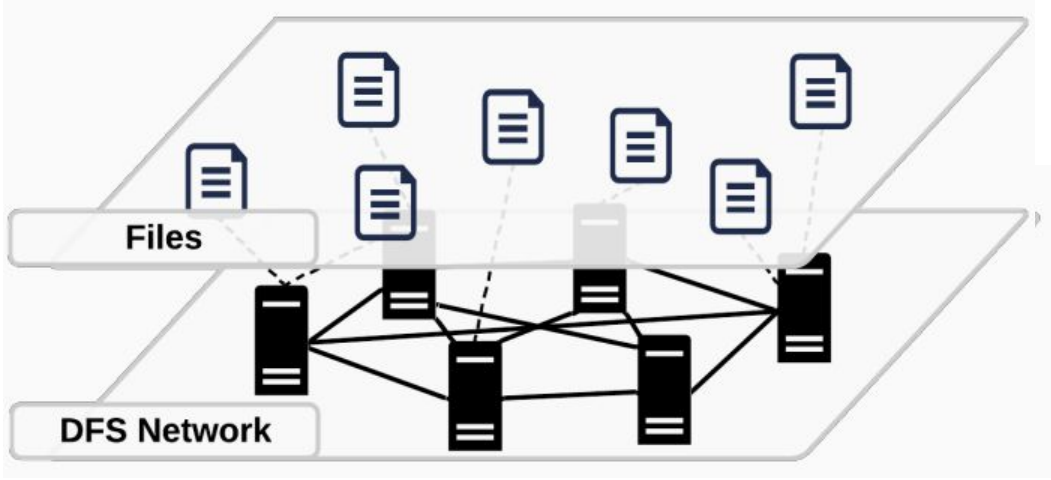
System

DFS P2P network → IPFS is using Content Based Addressing, i.e. items are directly queried through the network rather than establishing a connection with a server



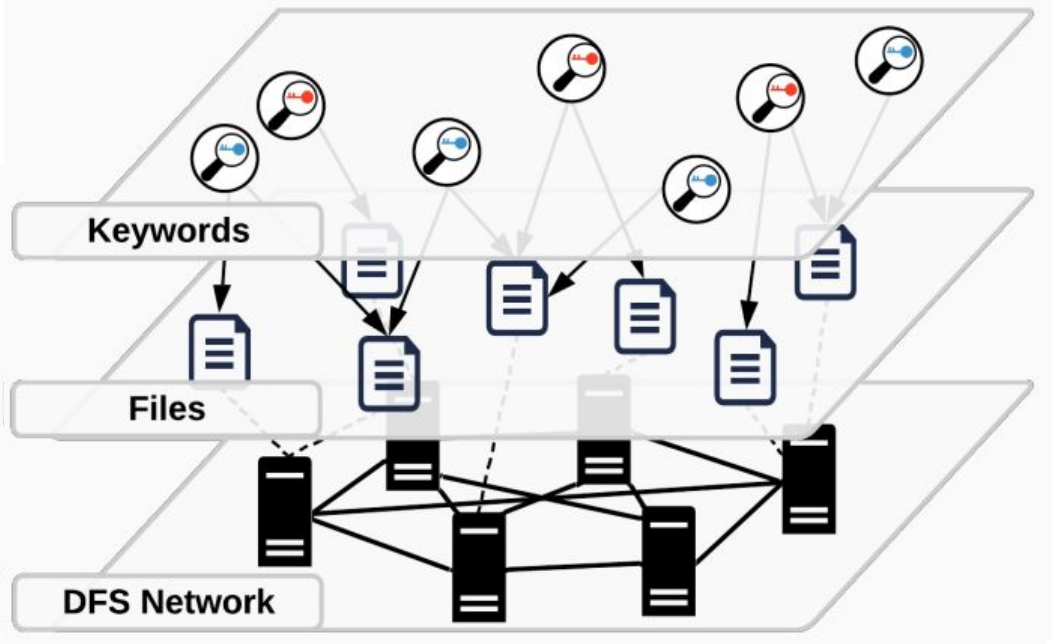
System

The P2P network that runs the IPFS protocol, stores and shares files in the form of IPFS objects that are identified by a Content IDentifier (CID), obtained through an hash function

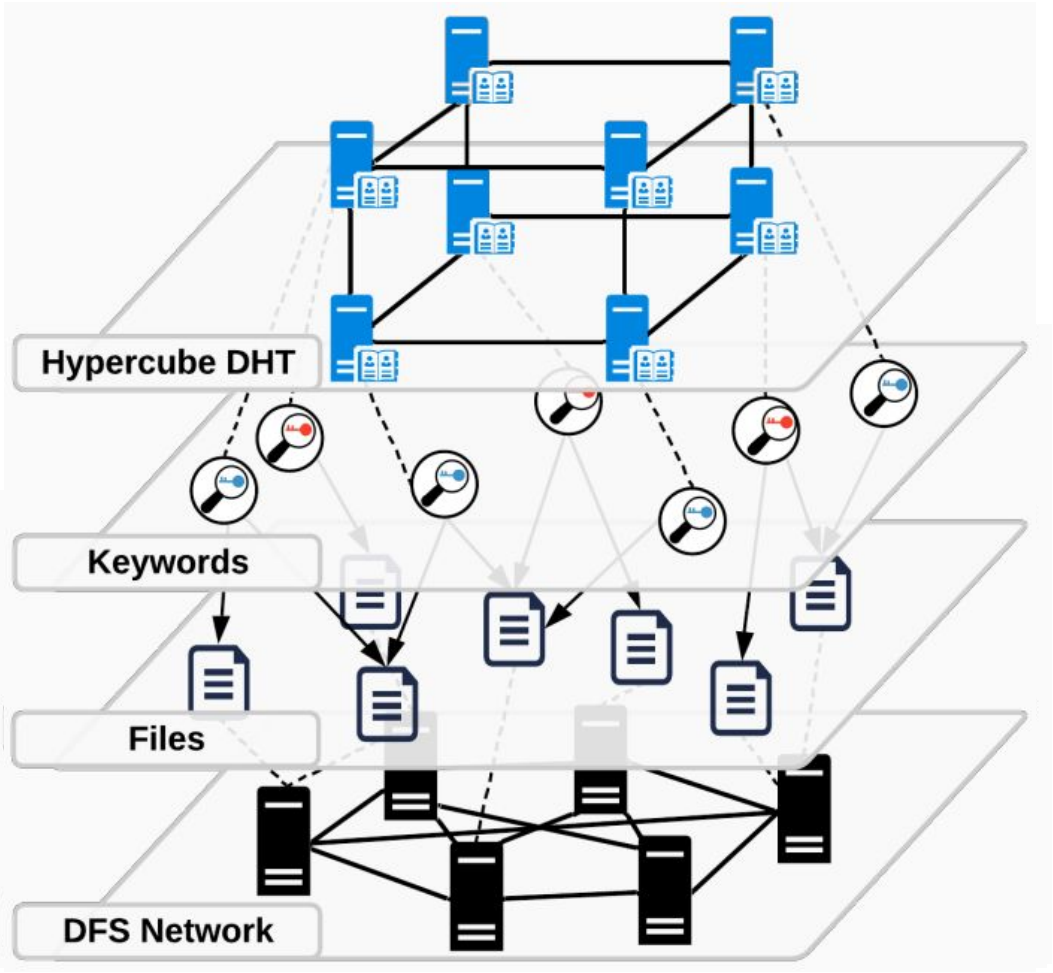


System

We can map keywords to any IPFS Object $o \in O$ using a keywords set $Ko \subseteq W$ (keyword space W)

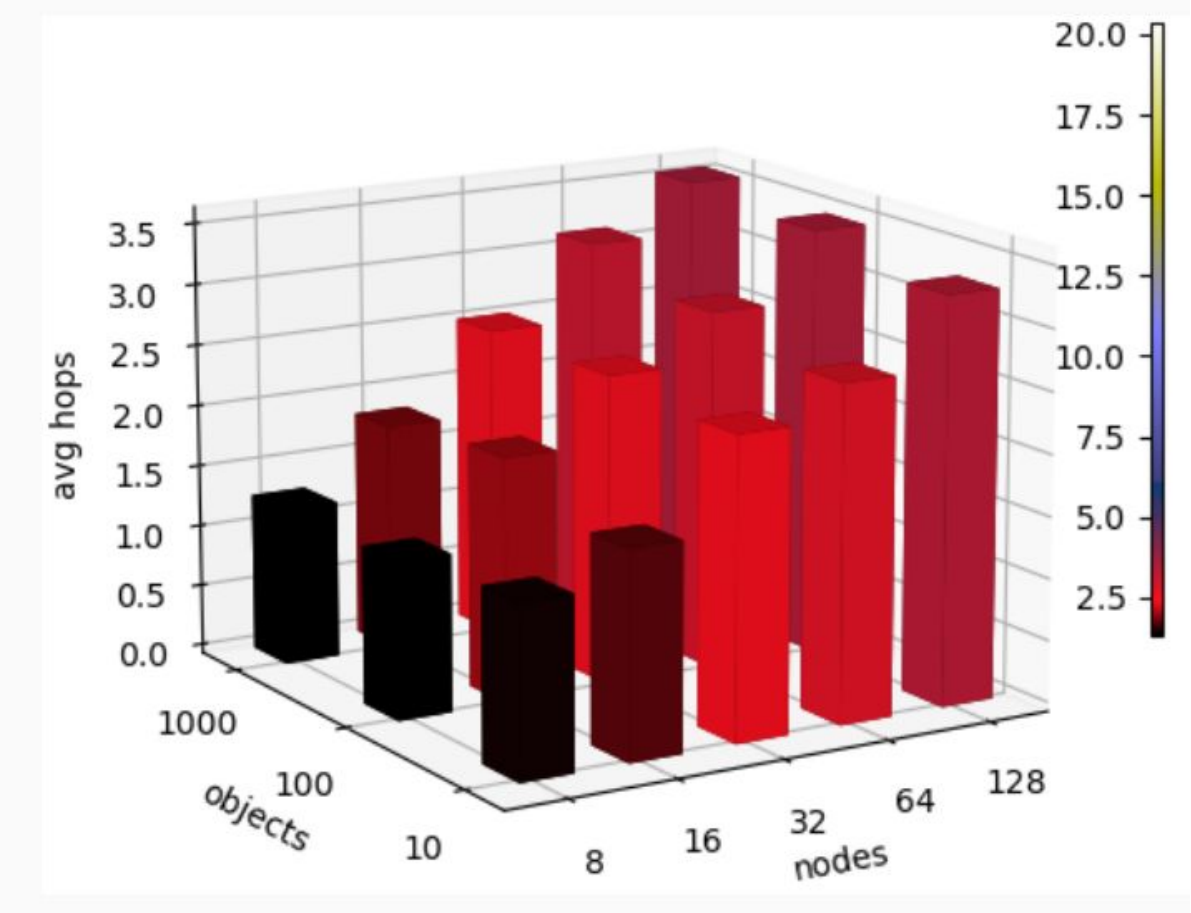


System



A DHT can be exploited to perform multiple keyword based queries. In particular one that takes the form of a r-dimensional hypercube $H_r(V, E)$

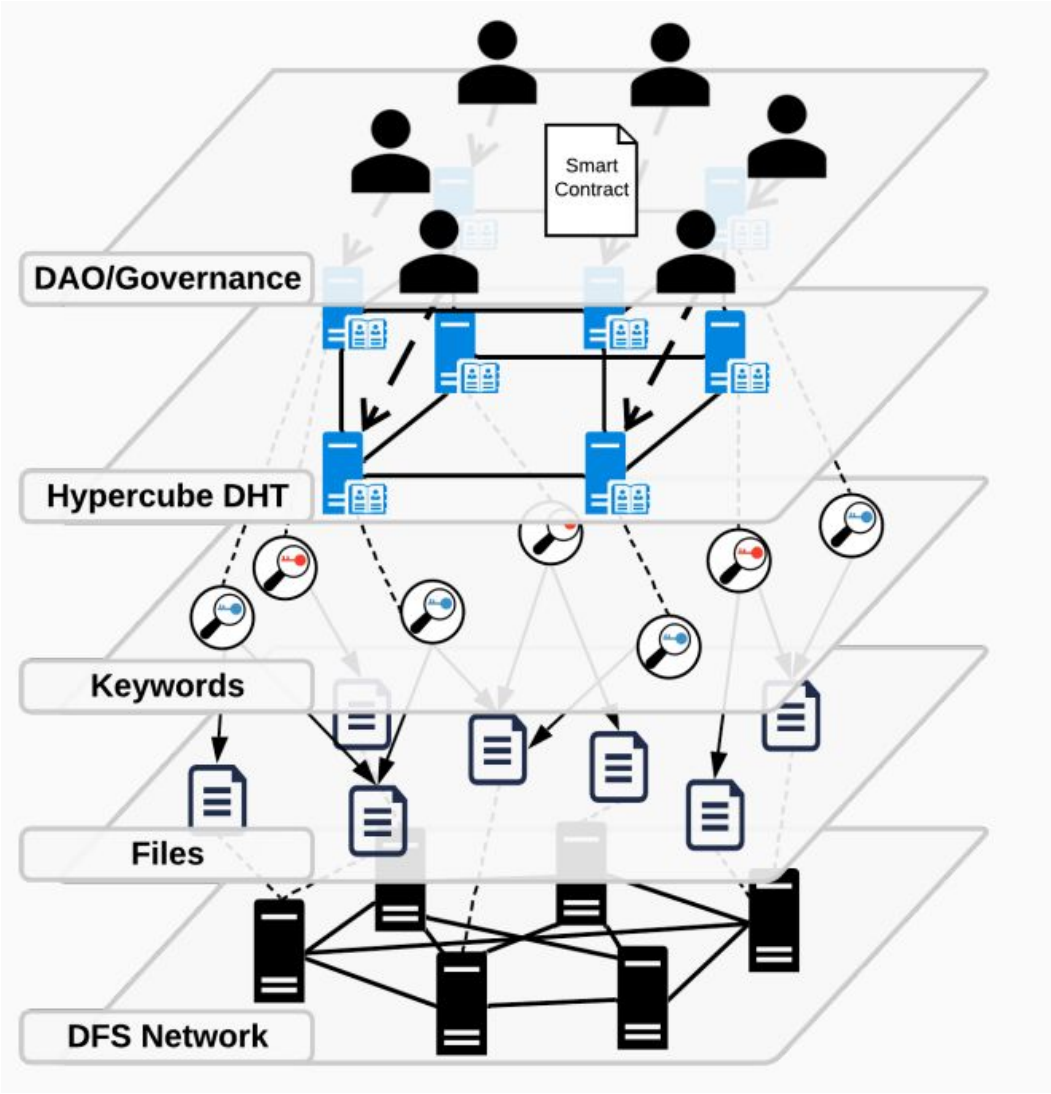
Some Results



Smart Contracts and Decentralized Autonomous Organizations

- Smart contracts → programs whose execution is performed in a distributed way
- Smart contracts can be used to automatize and supervise the exchange of digital or physical assets, e.g. tokens, and to allow the management of a DAO
- Decentralized Autonomous Organizations (DAO) → members can make proposals and also vote those through transparent mechanisms

DAO


Smart
Contract
DAO/Governance
Hypercube DHT
Keywords
Files
DFS Network

Use Case: Data Marketplace

- **Consumer** looks for some data through the **HyperCube**
- Interacts with a **smart contract** to get access authorization
- Asks the **decentralized authorization service** for the decryption key
- Gets the data from the **DFS**

Conclusion

- Hypercube DHT → decentralized system that manages keyword-based queries for contents stored in IPFS (and not only)
- Efficient trade-off between memory space and response time → maximum number of hops of log(number of nodes) = r, i.e. the hypercube dimension
- DAO → related to the economic sustainability and development of the above system
- DAO ERC20 tokens allows to reward nodes that have actively contributed

Stefano Ferretti

stefano.ferretti@uniurb.it

Department of Pure and Applied Sciences
University of Urbino "Carlo Bo"
P.zza della Repubblica 13
61029, Urbino
Italy