A Blockchain-based Mapping System

IETF 98 – Chicago
March 2017

Jordi Paillissé, **Albert Cabellos**, Vina Ermagan, Fabio Maino
acabello@ac.upc.edu

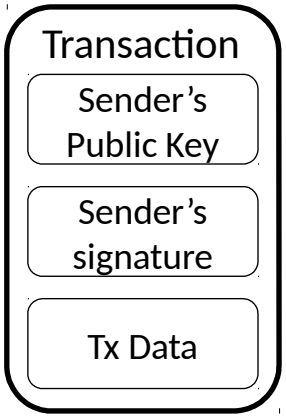
http://openoverlayrouter.org

A short Blockchain tutorial

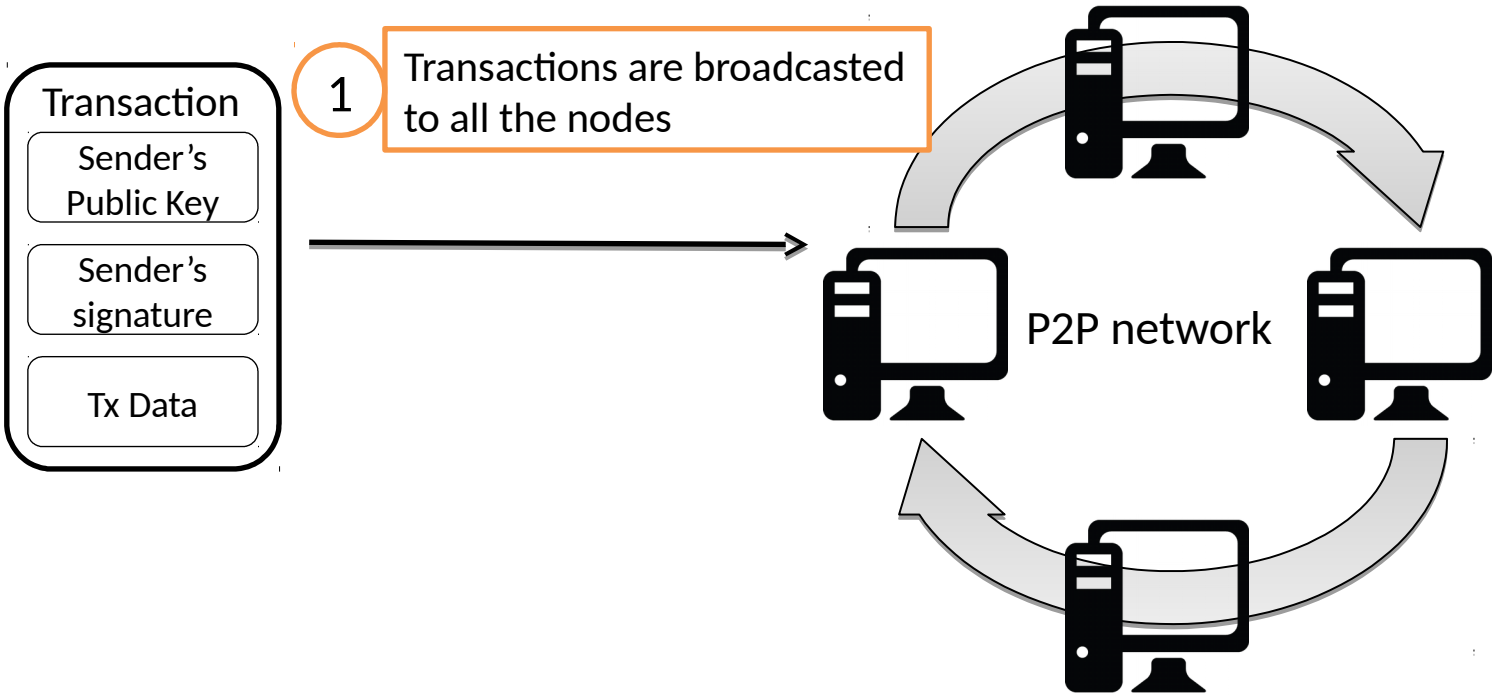
Blockchain - Introduction

- Blockchain = decentralized, secure and trustless database
- Add blocks of data one after another
- Protected by two mechanisms:
  - Chain of signatures
  - Consensus algorithm
- First appeared: Bitcoin, to exchange money
- Many more applications are possible

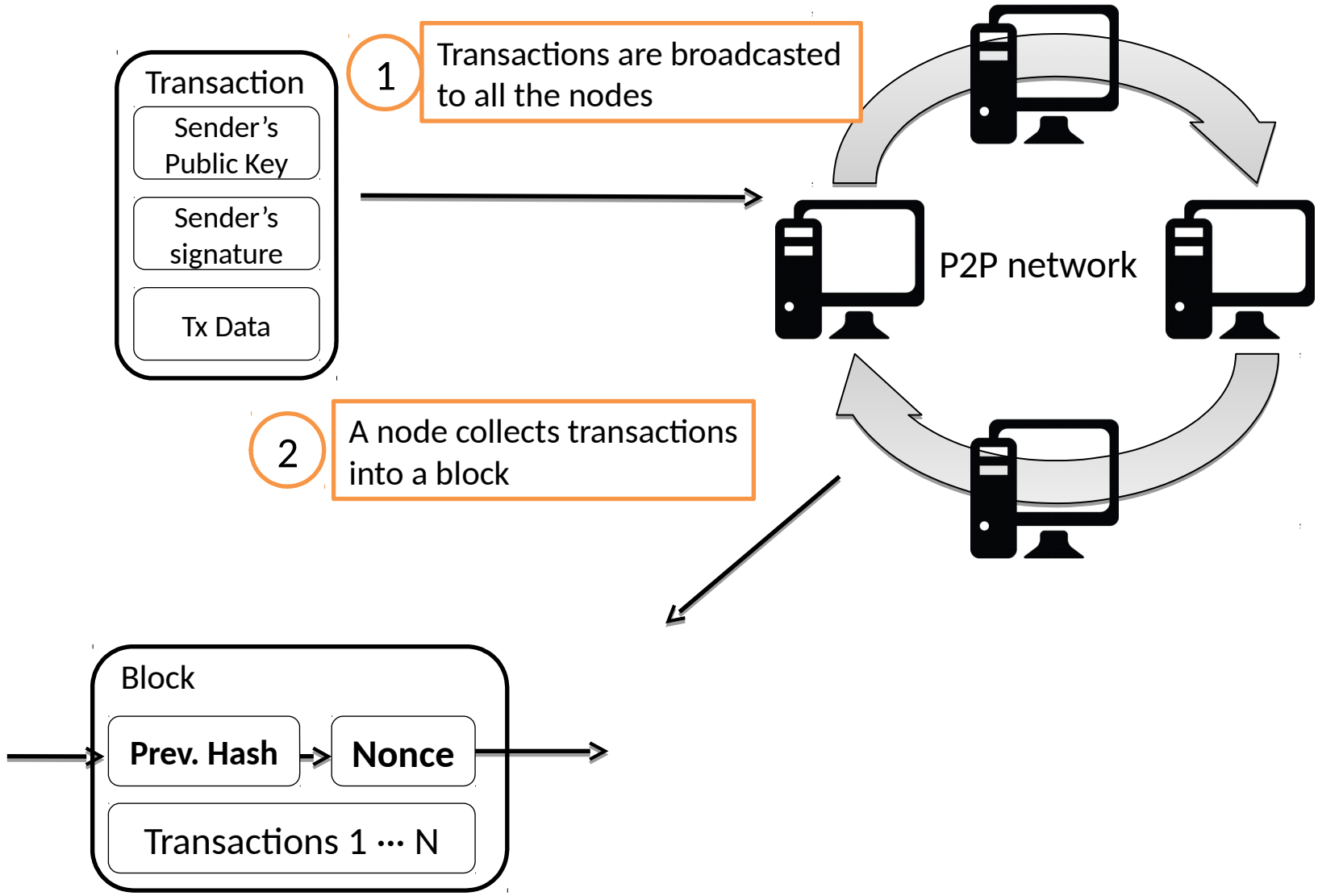
Blockchain - Transactions



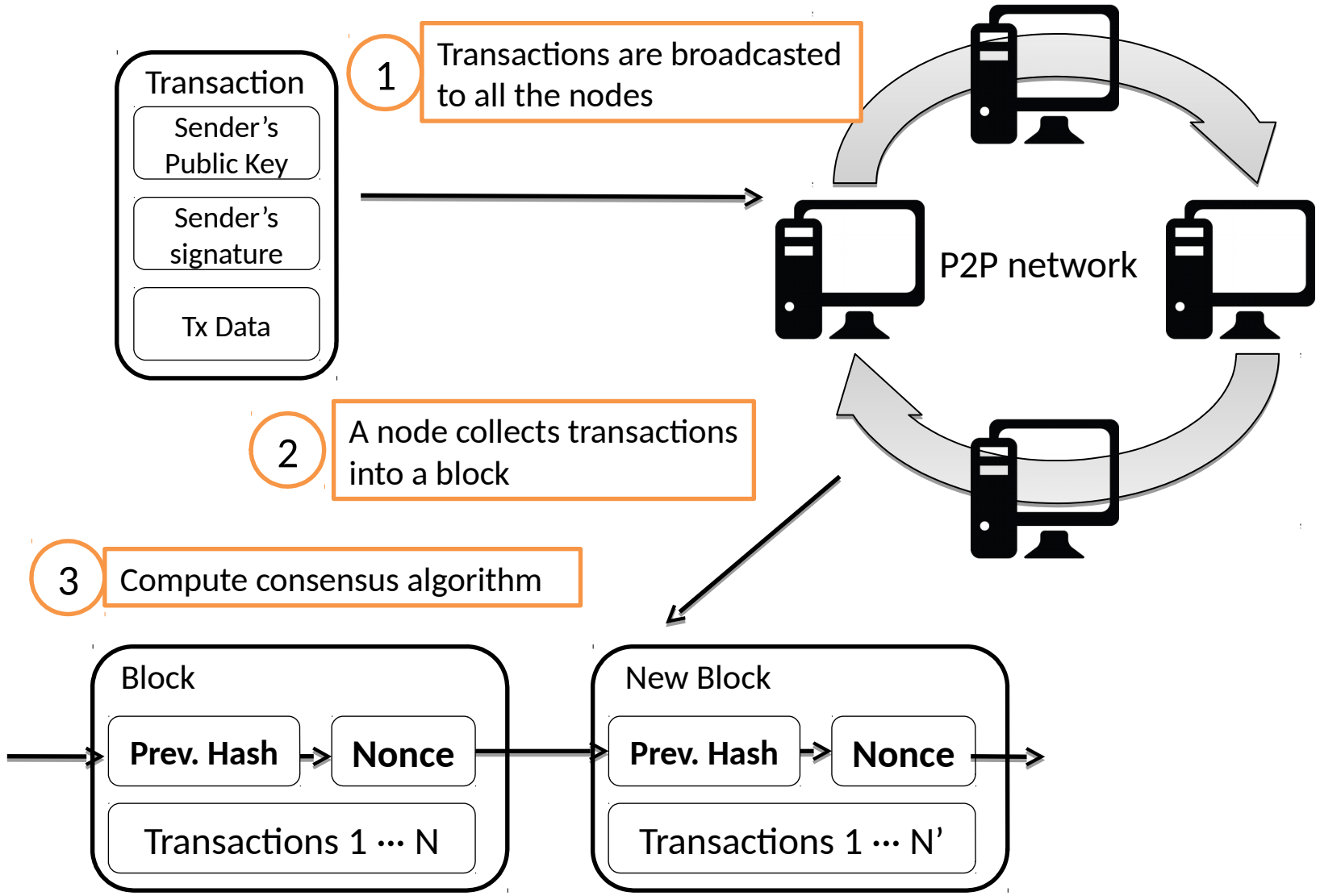
Blockchain - Transactions



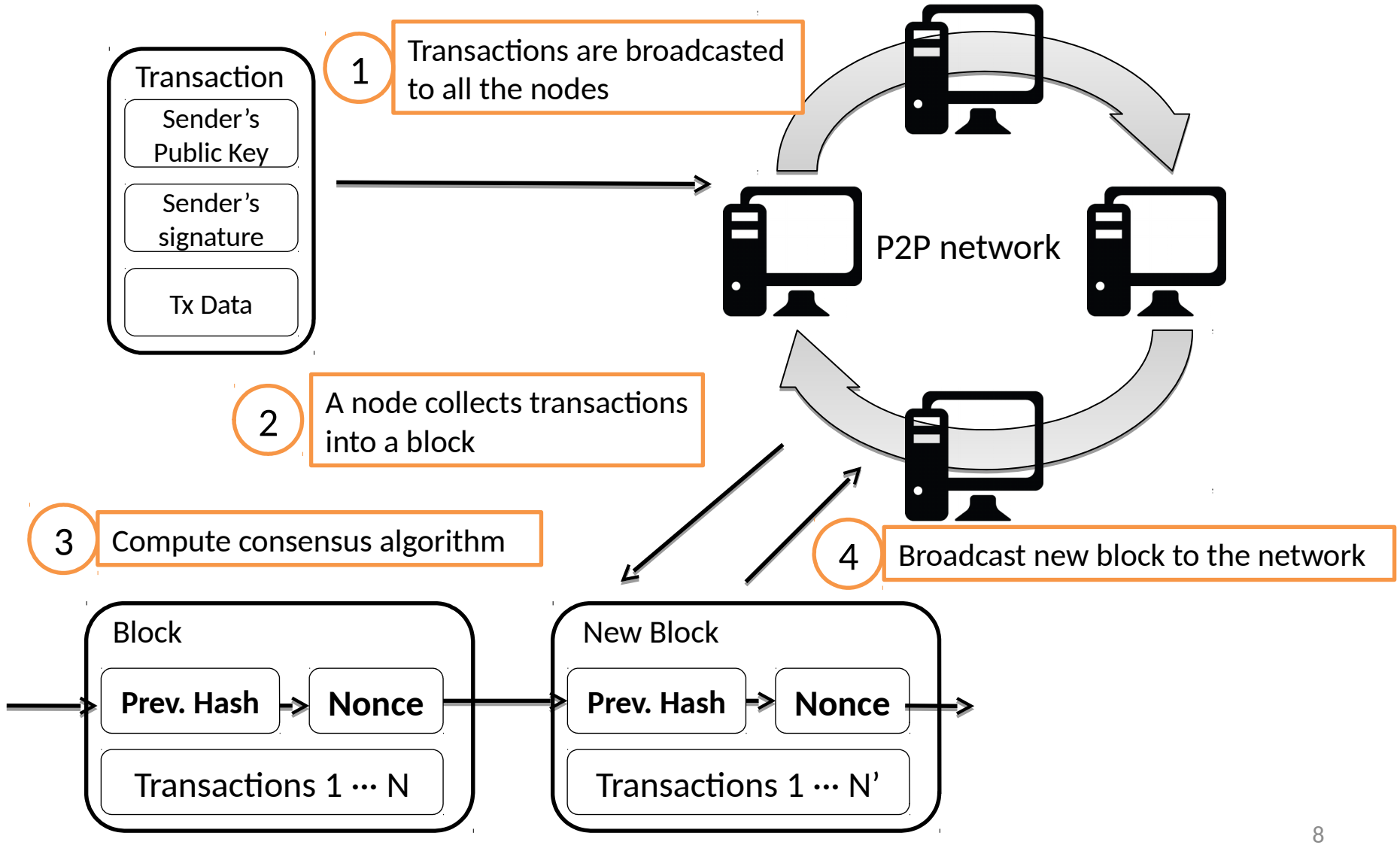
Blockchain - Transactions



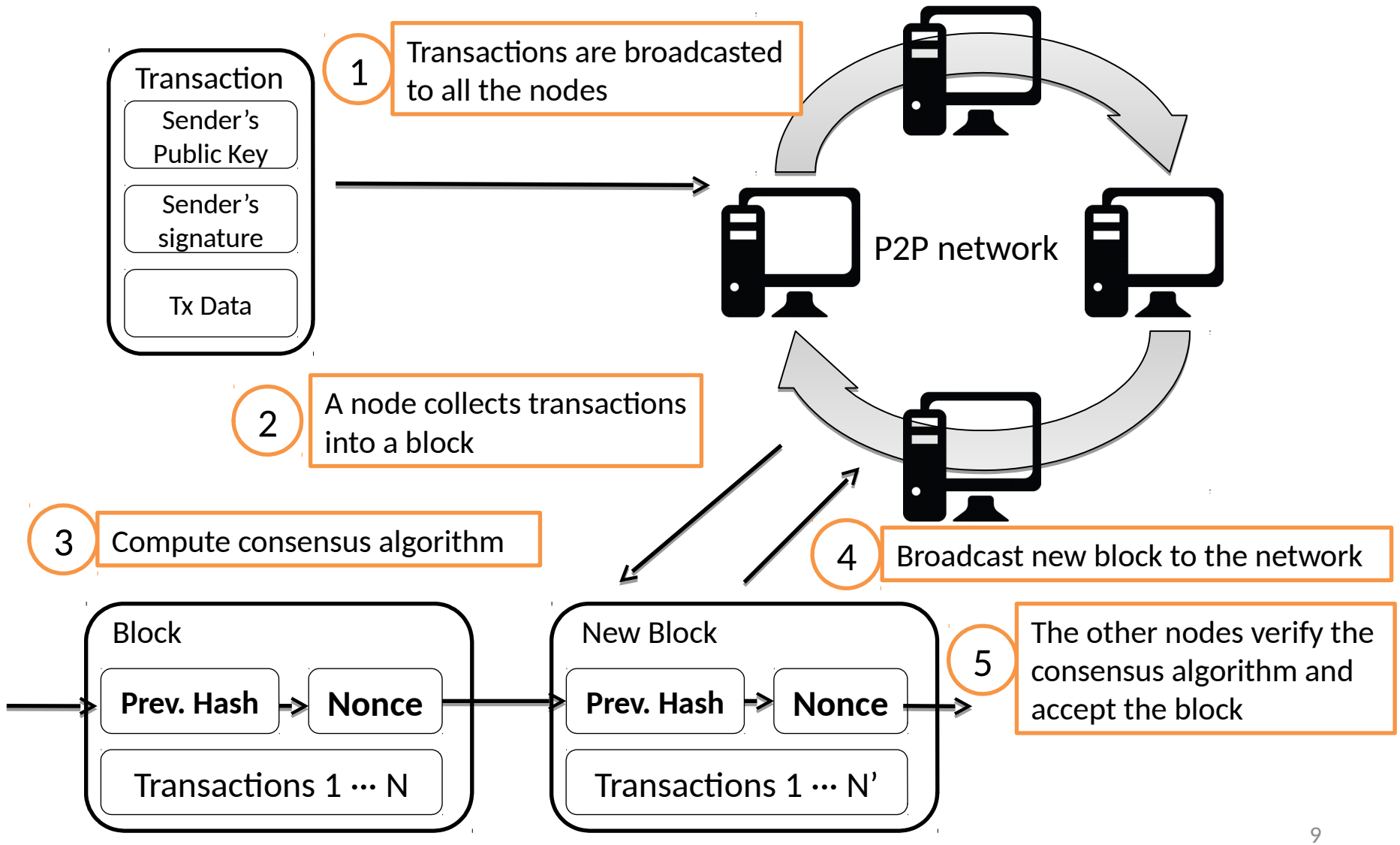
Blockchain - Transactions



Blockchain - Transactions



Blockchain - Transactions



Blockchain - Properties

- Decentralized: all nodes have the entire blockchain
- No prior trust required
- Decouples ownership from identity
- Append-only and immutable: added transactions cannot be modified
- Verifiable

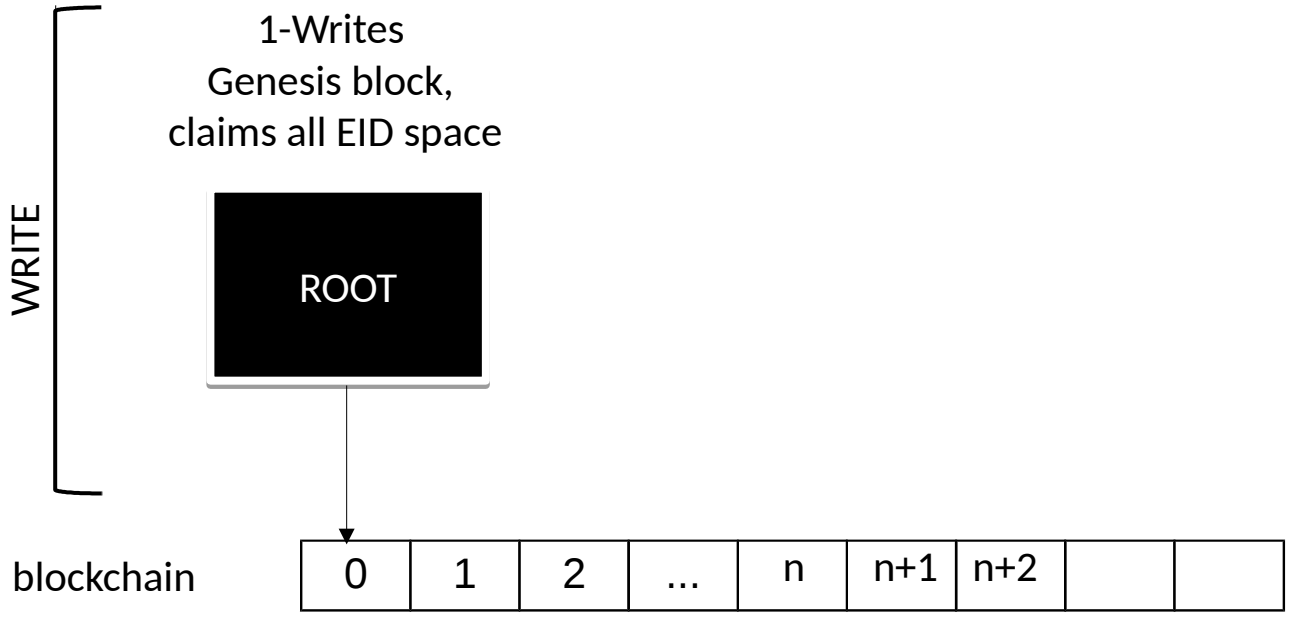
"Blockchain Technology", Sutardja Center (UC Berkeley)
http://scet.berkeley.edu/wp-content/uploads/BlockchainPaper.pdf

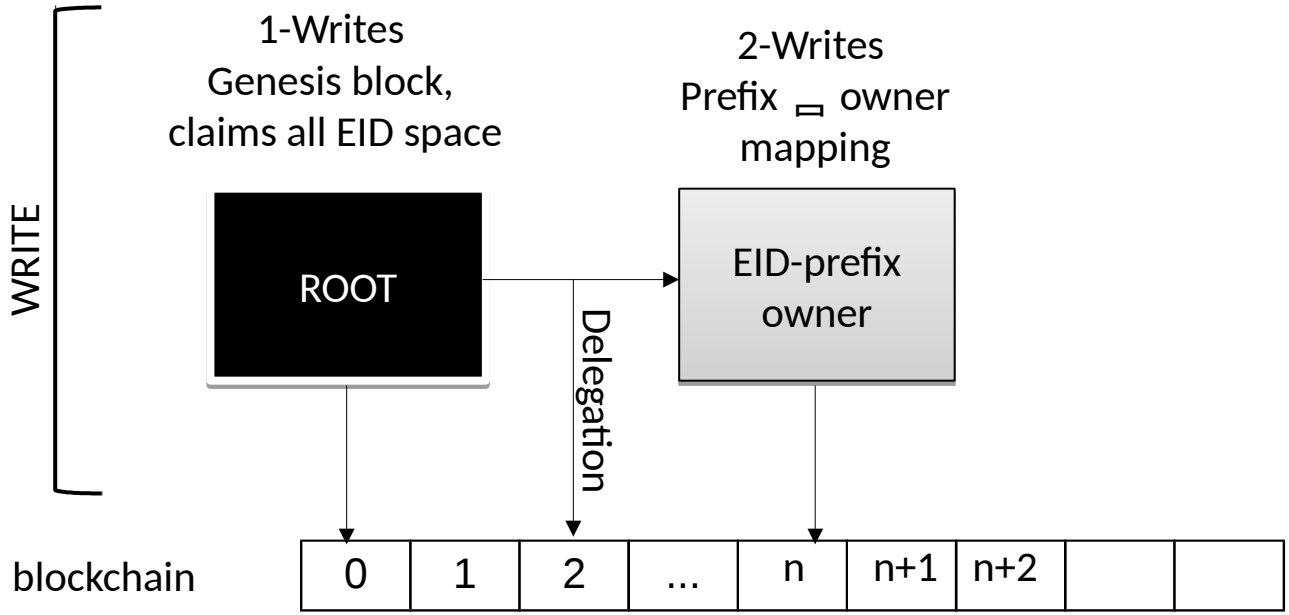
A Blockchain-based Mapping System **Overview**

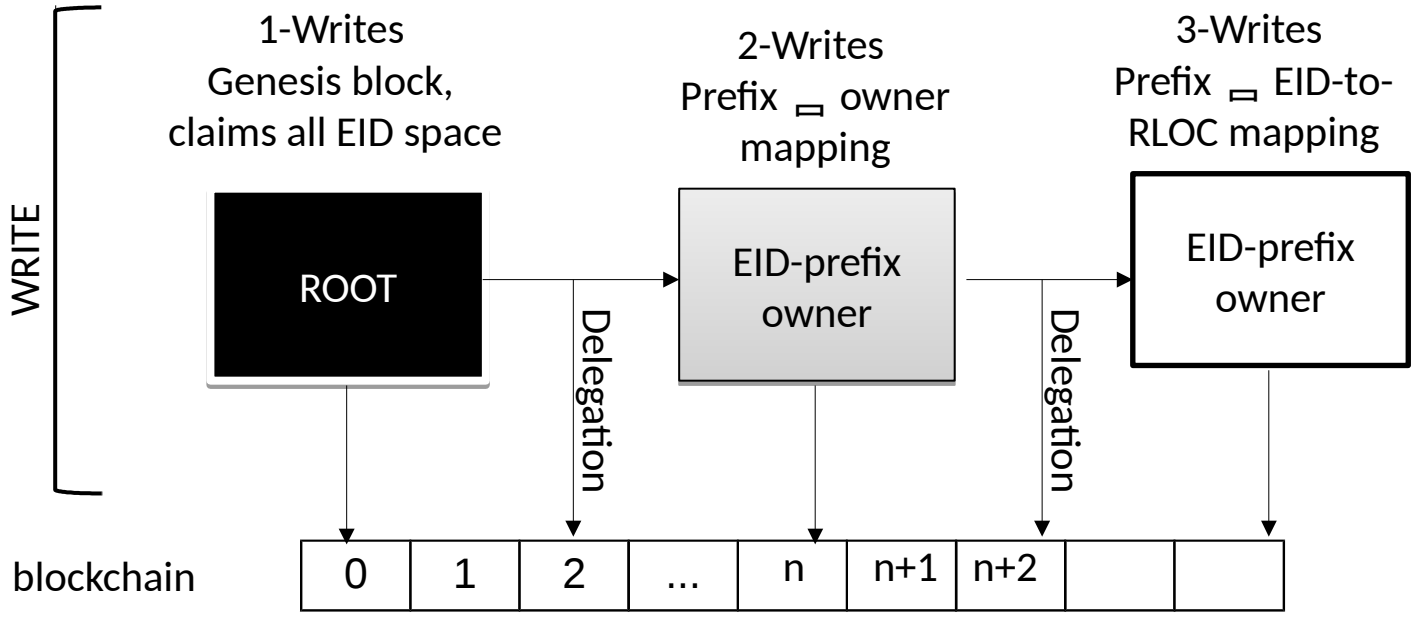
Basic Idea

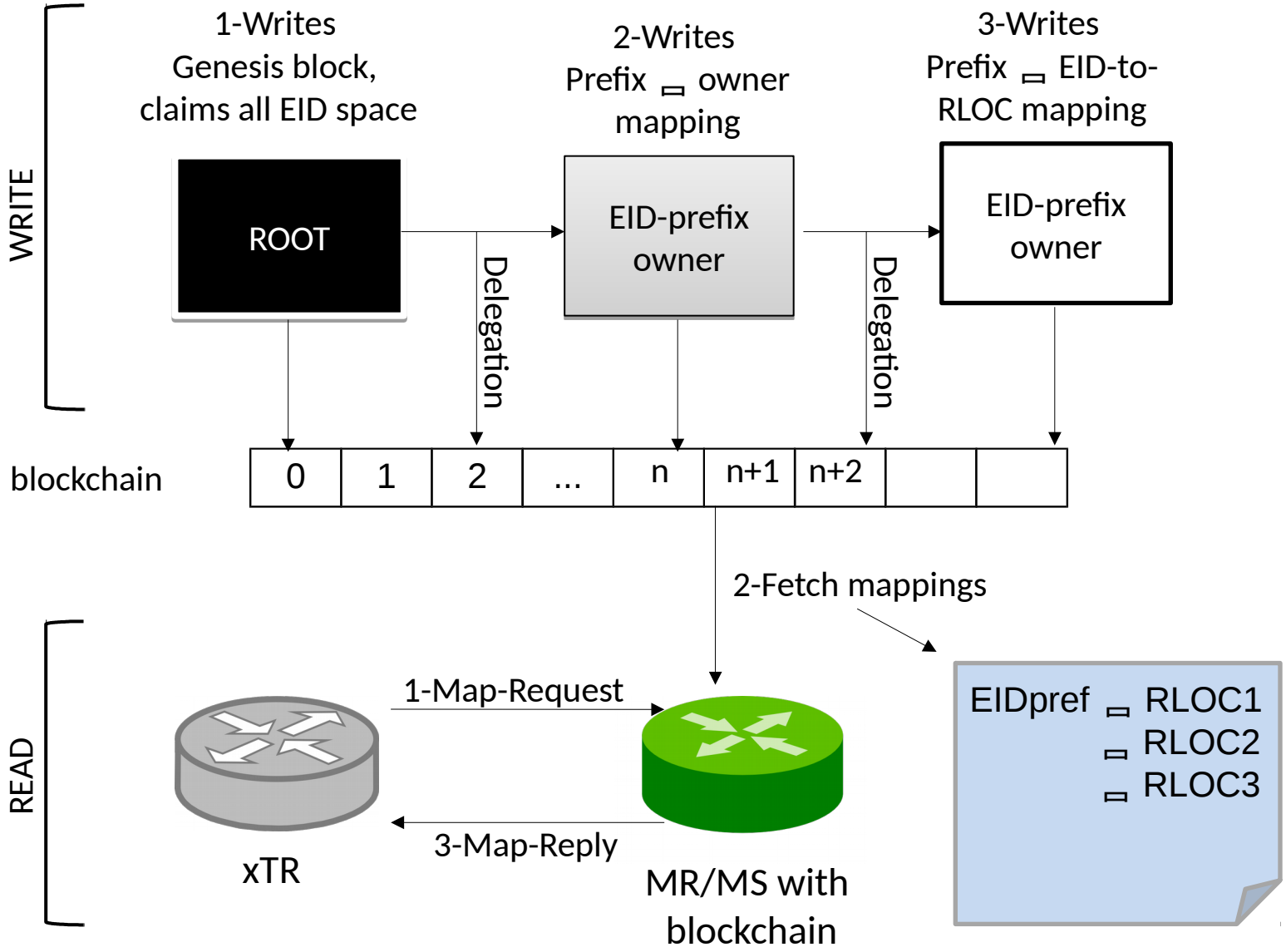
- **Objective**: Securely store:
  - EID prefix delegations (as in RPKI or DDT-ROOT)
  - EID-to-MS information (as in DDT)
  - EID-to-RLOC mappings (as in MS)
- Map Resolvers read the blockchain to find the mappings
- **Idea**: An EID is equivalent to a coin
  - Wallet: A set of EIDs
  - Transaction: Delegating EIDs or binding them to a MS or a set of RLOCs
  - Blockchain: A public ledger of the transactions

A Blockchain-based
Mapping System
Storing EID delegations and
EID-to-RLOC mappings

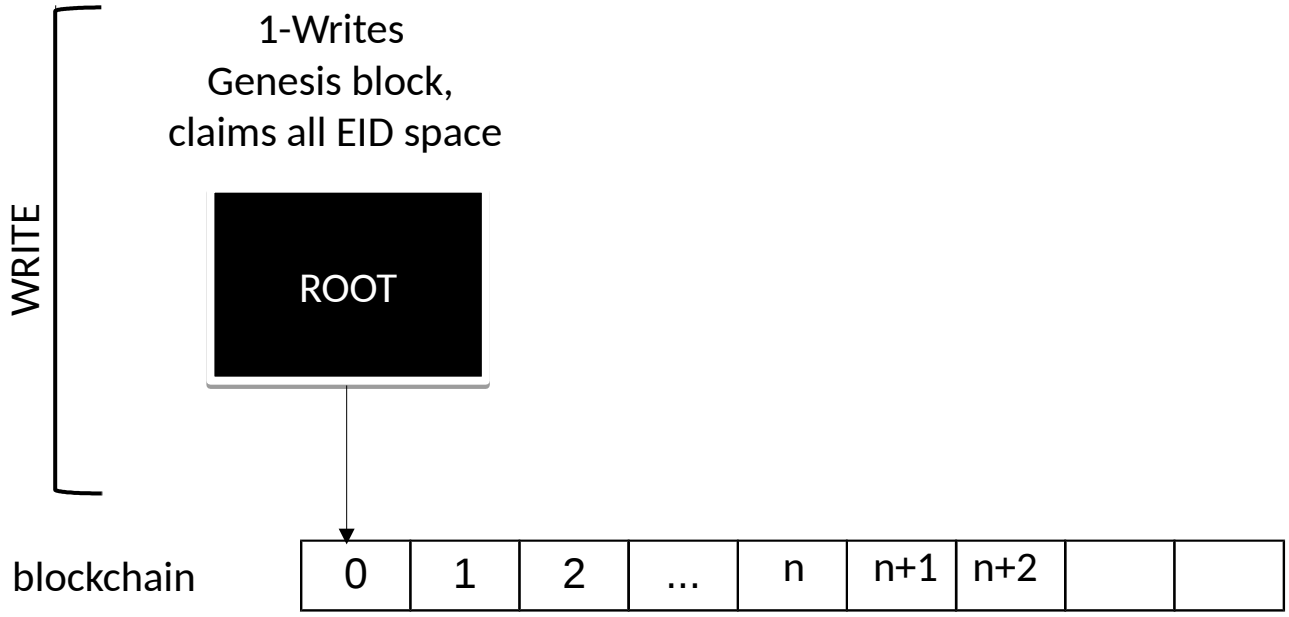

1-Writes
Genesis block,
claims all EID space
WRITE
ROOT
blockchain
0
1
2
...
n
n+1
n+2

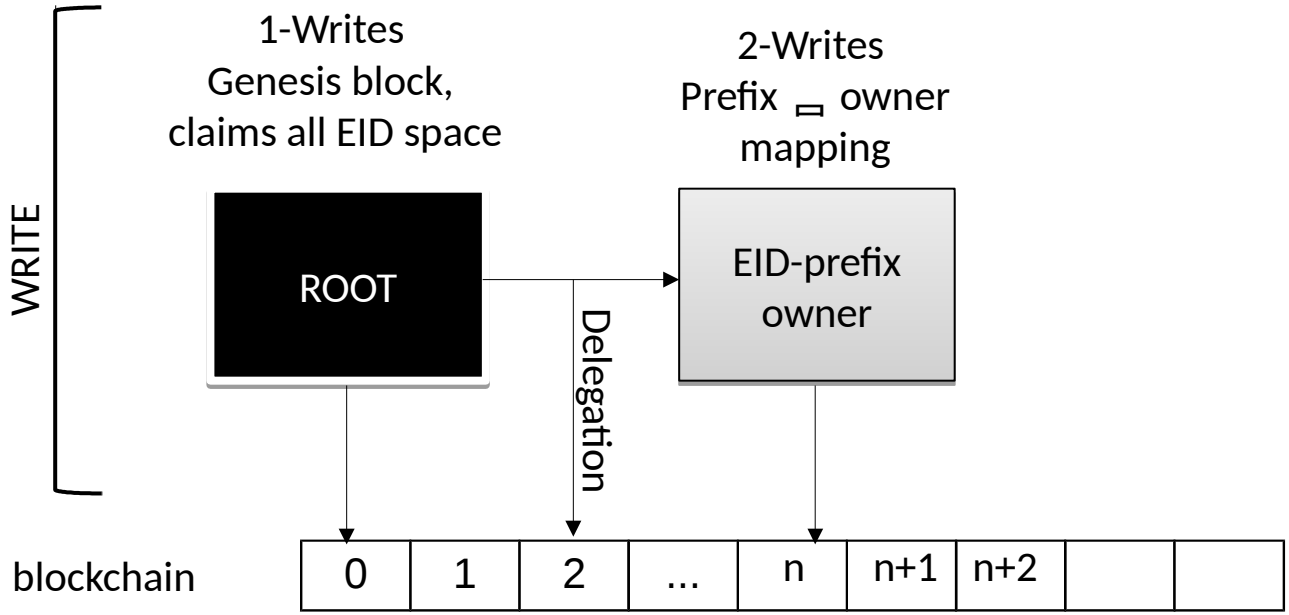

1-Writes
Genesis block,
claims all EID space
2-Writes
Prefix ␣ owner
mapping
WRITE
ROOT
EID-prefix
owner
Delegation
blockchain
0
1
2
...
n
n+1
n+2

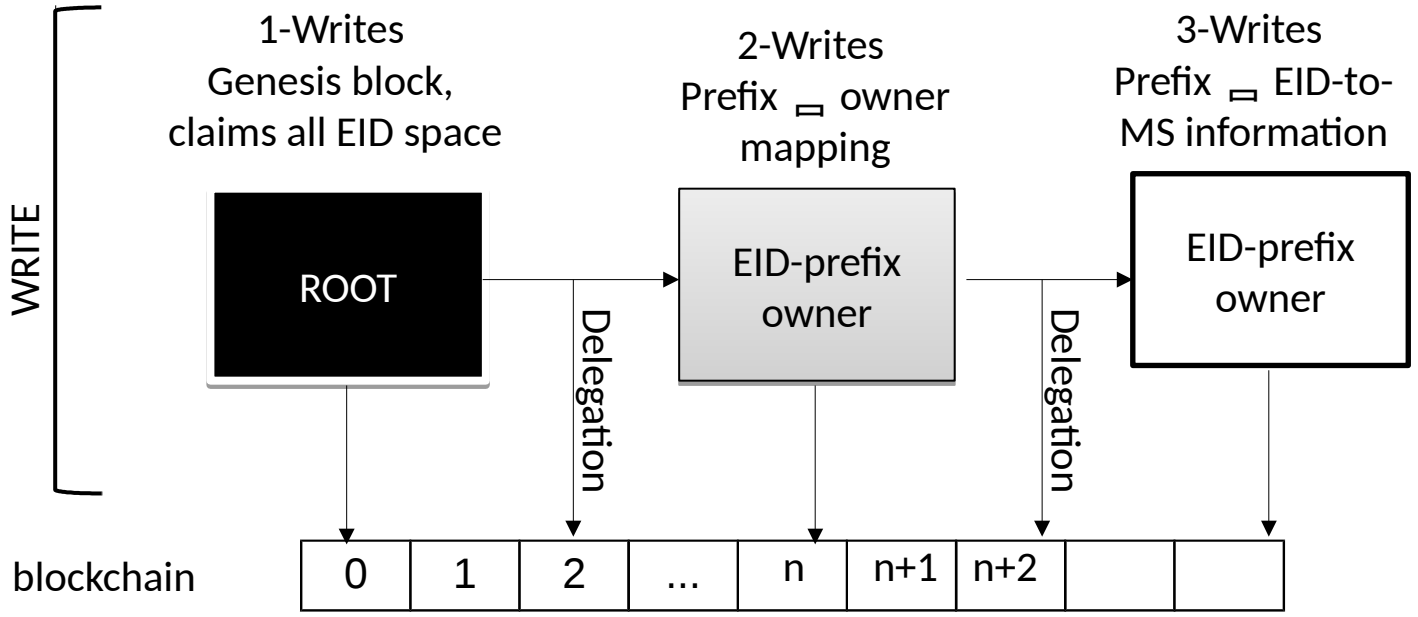

WRITE
1-Writes
Genesis block,
claims all EID space
2-Writes
Prefix ⊏ owner
mapping
3-Writes
Prefix ⊏ EID-to-
RLOC mapping
ROOT
EID-prefix
owner
EID-prefix
owner
Delegation
Delegation
blockchain
0
1
2
...
n
n+1
n+2

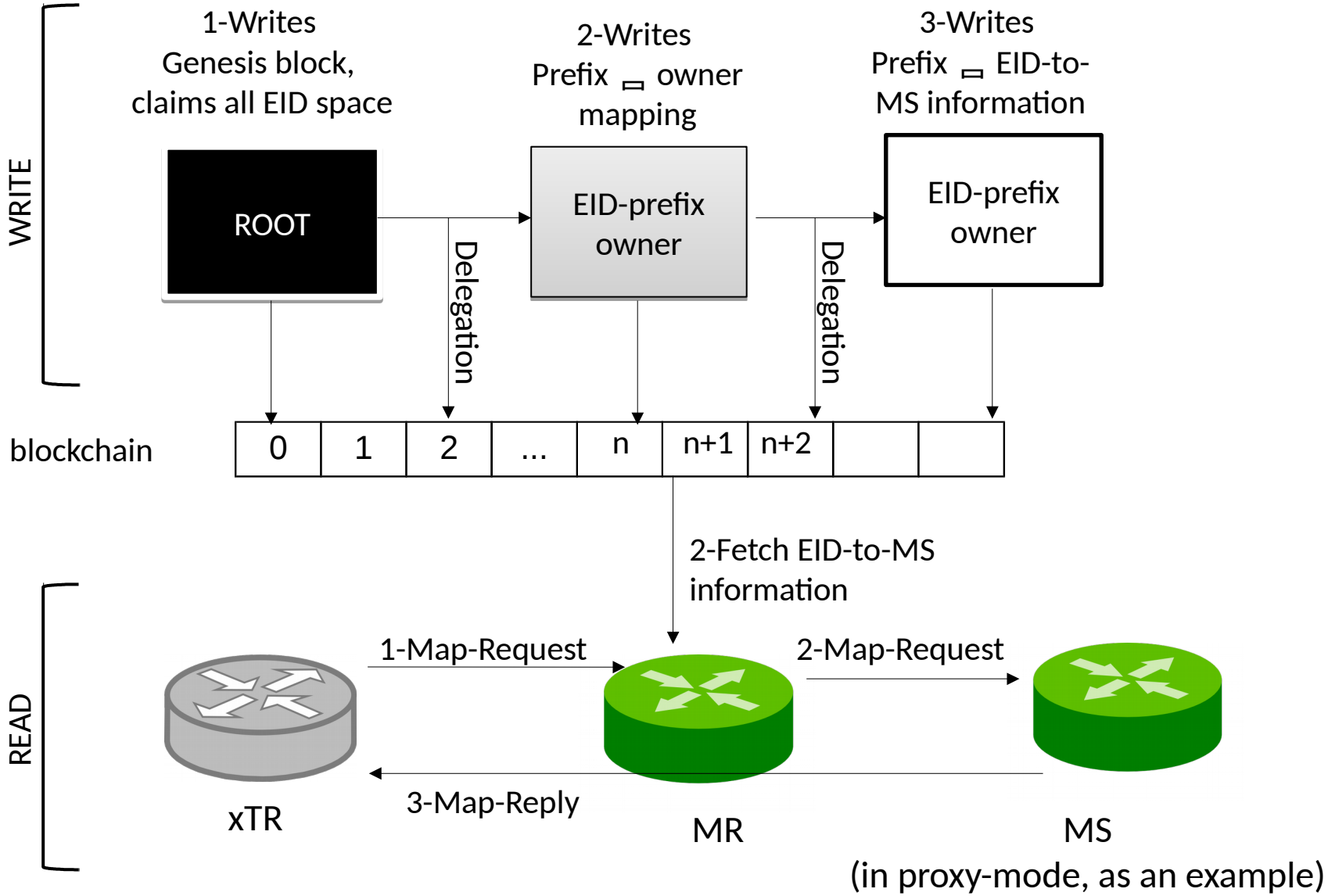

WRITE
1-Writes
Genesis block,
claims all EID space
2-Writes
Prefix ▭ owner
mapping
3-Writes
Prefix ▭ EID-to-
RLOC mapping
ROOT
Delegation
EID-prefix
owner
Delegation
EID-prefix
owner
blockchain
0
1
2
...
n
n+1
n+2
2-Fetch mappings
READ
1-Map-Request
3-Map-Reply
xTR
MR/MS with
blockchain
EIDpref ▭ RLOC1
▭ RLOC2
▭ RLOC3

A Blockchain-based
Mapping System
Storing EID delegations
and EID-to-MS information


1-Writes
Genesis block,
claims all EID space
WRITE
ROOT
blockchain
0
1
2
...
n
n+1
n+2


1-Writes
Genesis block,
claims all EID space
2-Writes
Prefix ␣ owner
mapping
WRITE
ROOT
EID-prefix
owner
Delegation
blockchain
0
1
2
...
n
n+1
n+2


WRITE
1-Writes
Genesis block,
claims all EID space
2-Writes
Prefix ␣ owner
mapping
3-Writes
Prefix ␣ EID-to-
MS information
ROOT
EID-prefix
owner
EID-prefix
owner
Delegation
Delegation
blockchain
0
1
2
...
n
n+1
n+2


WRITE
1-Writes
Genesis block,
claims all EID space
2-Writes
Prefix ␣ owner
mapping
3-Writes
Prefix ␣ EID-to-
MS information
ROOT
Delegation
EID-prefix
owner
Delegation
EID-prefix
owner
blockchain
0
1
2
...
n
n+1
n+2
2-Fetch EID-to-MS
information
READ
1-Map-Request
2-Map-Request
3-Map-Reply
xTR
MR
MS
(in proxy-mode, as an example)

Pros and Cons

Pros

- Infrastructure-less and decentralized
- Fast lookup
- Secure, without certs
  - Non-repudiation
  - Resilience
  - Integrity
  - Authentication
- No prior trust required
- Simple rekeying

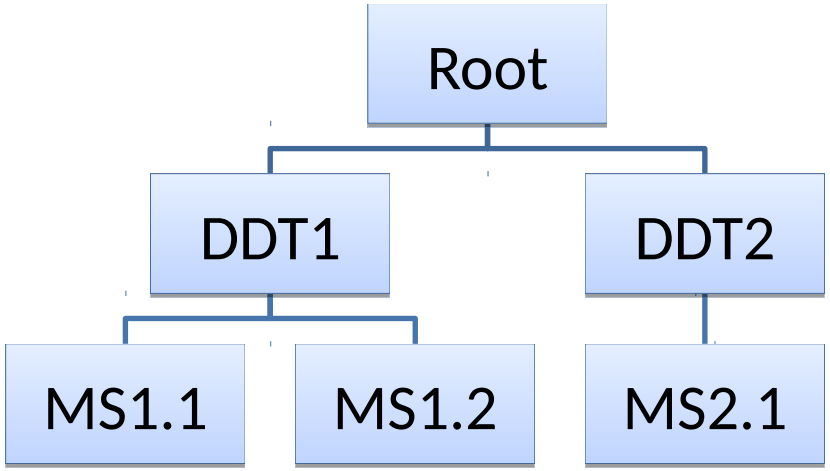
Cons

- Challenges with incentives
- Slow updates
  - **Mappings can be stored in a MS, then performance is as fast as DDT**
- Costly bootstrapping
- Large storage required

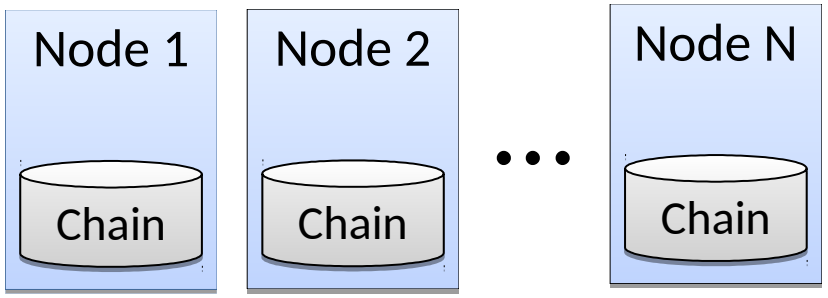
Can be mitigated using a dedicated chain

Comparison with LISP-DDT

LISP-DDT



Blockchain



+ Fast update ⇨ Dynamic mappings
- Manual configuration

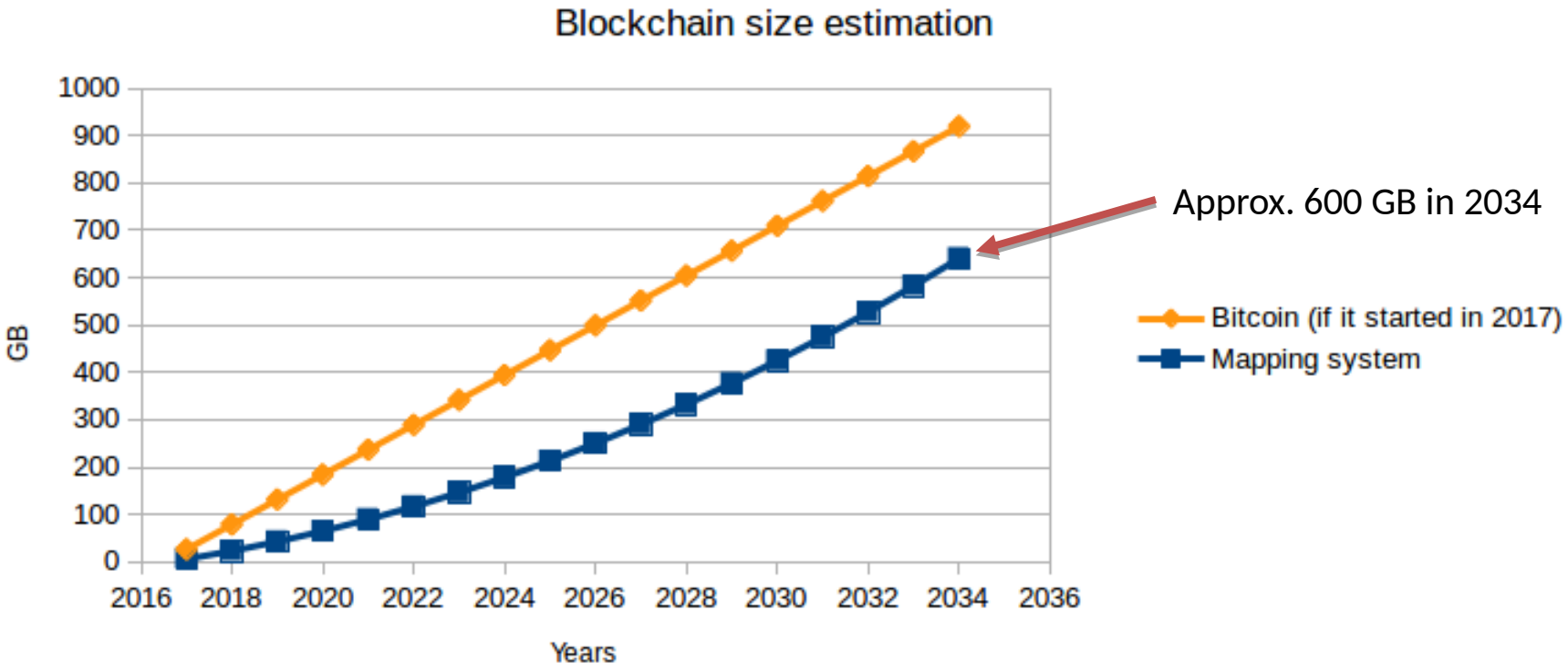
+ Less infrastructure
+ No certificates
+ Fast queries
- Large storage required
- Update mappings slow ⇨ Store Mappings in MS (same performance as MS)

Issues with RPKI

	RPKI	Blockchain
Anonymity [1]	Prefixes linked to owner name	Prefixes linked to a public key
Revocation	Performed by CAs	Performed automatically (validity time) or impossible
Certificate management [2]	Complex	No certificates

[1] Wählisch, Matthias, et al. "RiPKI: The tragic story of RPKI deployment in the Web ecosystem." *Proceedings of the 14th ACM Workshop on Hot Topics in Networks*. ACM, 2015.
[2] George, Wes. "Adventures in RPKI (non) Deployment." NANOG, 2014.

Scalability



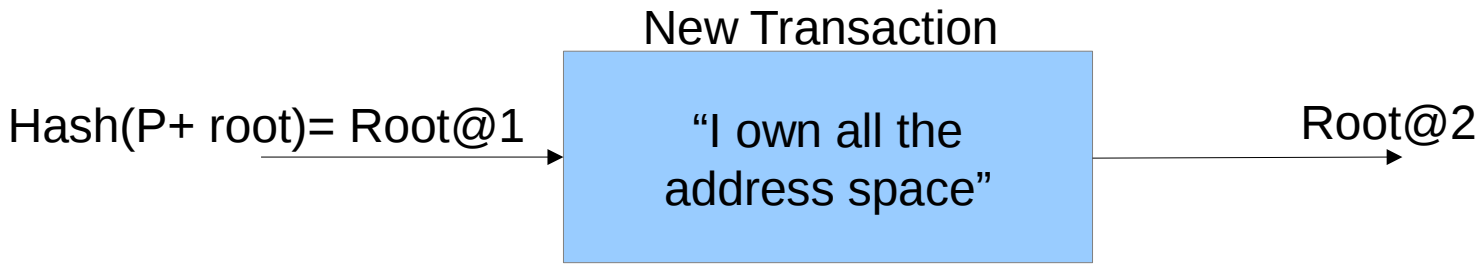
- One mapping for each block of /24 IPv4 address space
- Growth similar to BGP churn*
- Prefix delegation + mappings
- Each transaction approx. 400 bytes
- Only prefixes: approx. 40 GB in 20 years (worst case + BGP table growth*)

*Source: http://www.potaroo.net/ispcol/2017-01/bgp2016.html

A Blockchain-based
Mapping System
Transactions

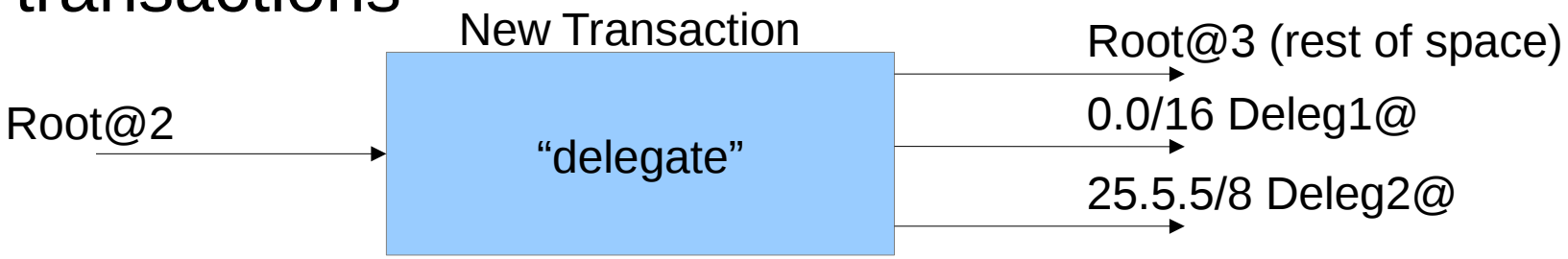
First transaction

- Map-Resolver trust the Public Key of the Root, that initially claims all EID space by writing the genesis block
- Root can delegate all EID space to itself and use a different keypair

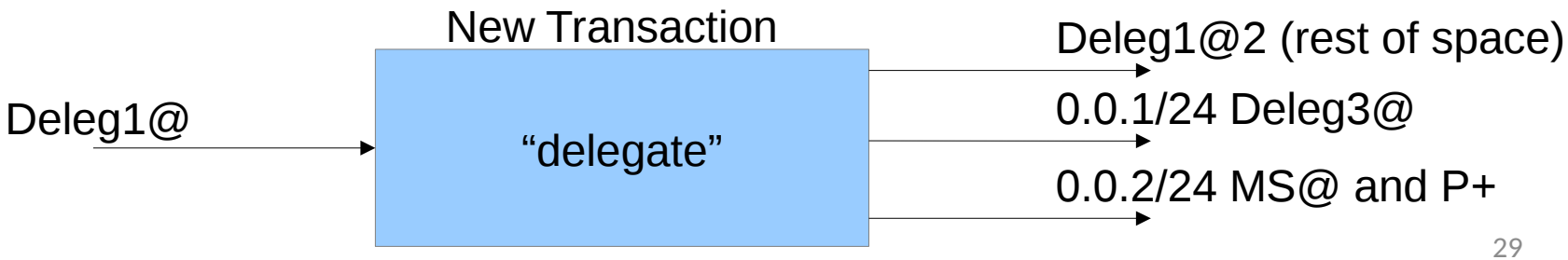


Prefix delegation

- Root delegates EID-prefixes to other entities (identified by Hash(Public Key)) by adding transactions

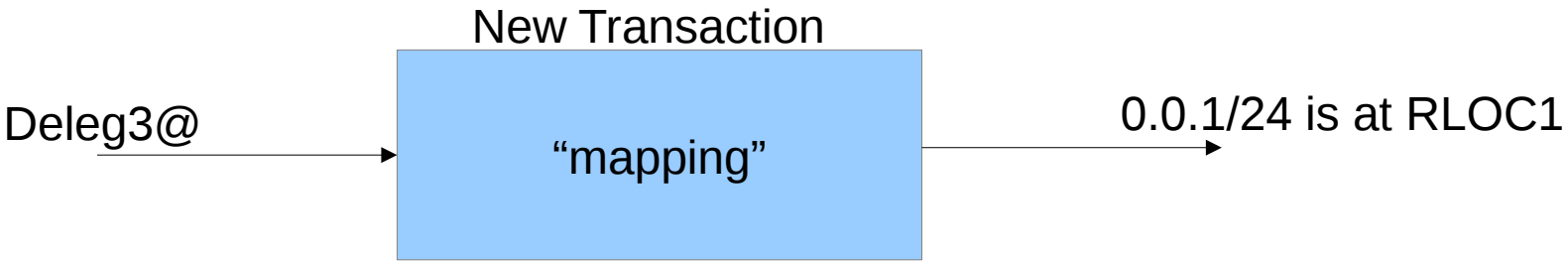


- Owners can further delegate address blocks to other entities or write MS addresses (and MS’s Public Key)



Writing mappings

- Just like delegating a prefix, but instead of the Map Server address, we write the mapping

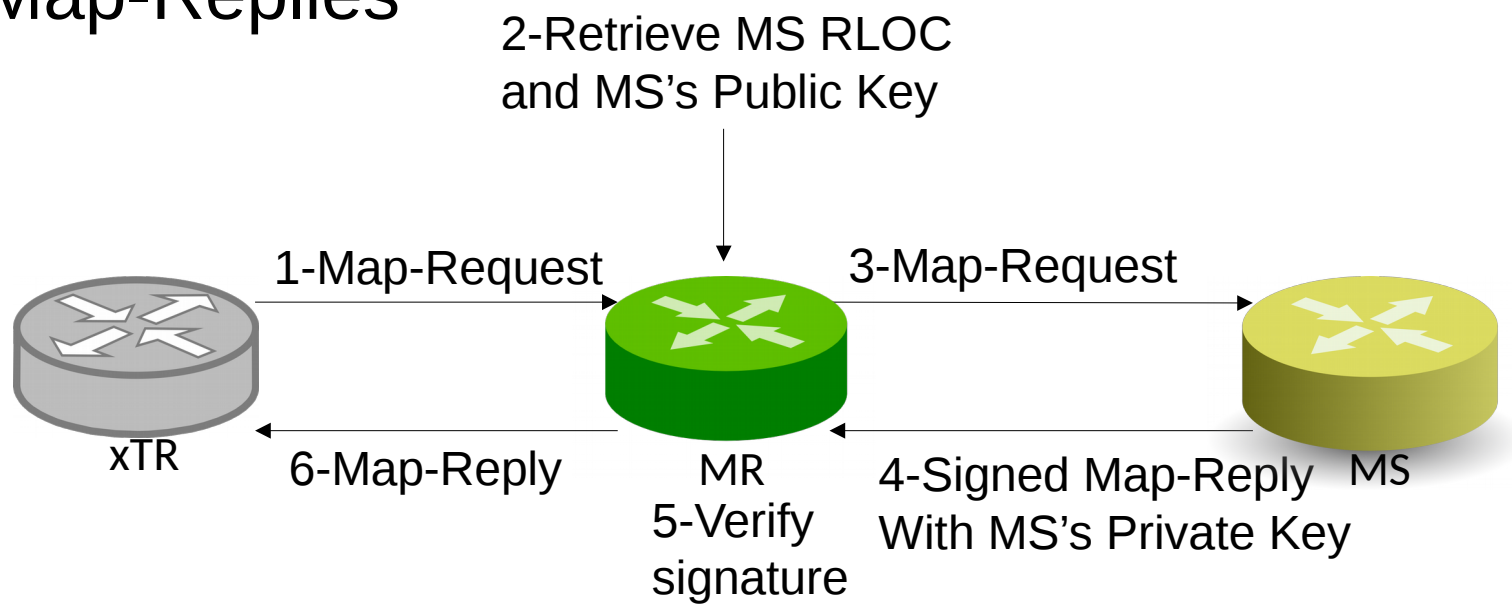


Rekeying

- Delegating the owned EID-prefixes to itself using a new key set.
- Simpler than traditional rekeying schemes
- Can be performed independently, i.e. each owner can do it without affecting other owners
- Same procedure for mappings

Map-Reply Authentication

- MS public key can also be included in the delegations
- Since blockchain provides authentication and integrity for this key, MRs can use it to verify Map-Replies



A Blockchain-based Mapping System **Prototyping**

Design considerations

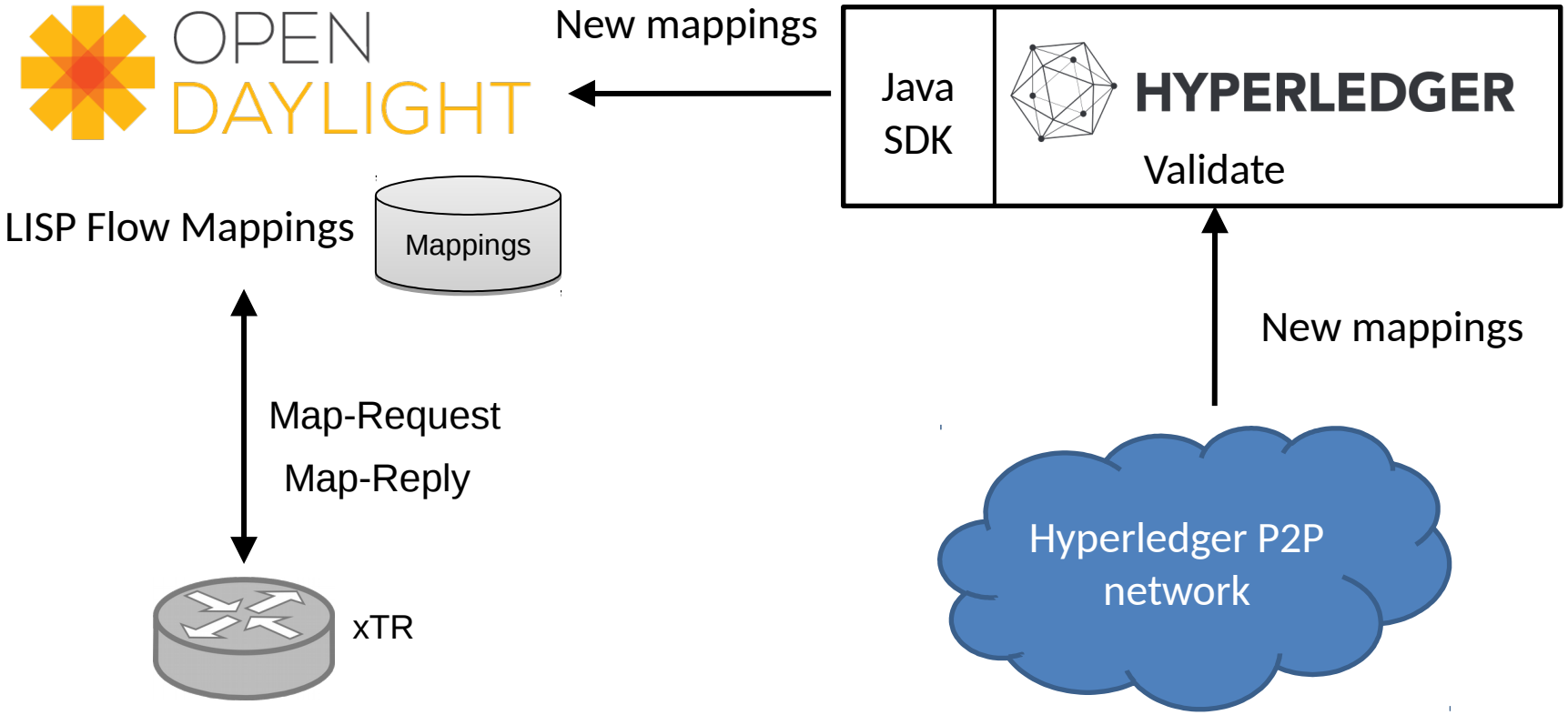
- Bitcoin is too restrictive:
  - Only for money transfer
  - Huge blockchain file size (approx. 100 GB)
  - High bootstrap time (several days*)
  - Low throughput (7 transactions/sec.)
- New blockchain technologies:
  - More scalable
  - Smart contracts

*depends on connection speed

Dedicated chain

- Public (anyone can use it) but dedicated (only for mappings)
- Stores:
  - Prefix delegations – **Replaces DDT ROOT**
  - EID-to-MS information – **Replaces DDT-Nodes**
  - EID-to-RLOC mappings (if you don´t expect many updates) – **xTR does NOT need a Map-Server**
- **We plan to deploy it in LISP-Beta**

Prototype


OPEN DAYLIGHT
New mappings
Java SDK
HYPERLEDGER
Validate
LISP Flow Mappings
Mappings
New mappings
Map-Request
Map-Reply
xTR
Hyperledger P2P network

A Blockchain-based Mapping System

IETF 98 – Chicago
March 2017

Jordi Paillissé, Albert Cabellos, Vina Ermagan, Fabio Maino
acabello@ac.upc.edu

http://openoverlayrouter.org

More about the Consensus Algorithm

- Rules used by nodes to agree on which data to accept
- Eg. Bitcoin uses Proof of Work
- Miners compute Proof of Work
  - Finding a nonce that when added to the data makes its hash start with N zeros.
  - Hard
- Other algorithms are being explored:
  - Proof of Stake: nodes with more assets are more likely to add blocks
  - Practical Byzantine Fault Tolerant: reach a minimum number of endorsements from nodes in order to add data
  - Deposit-based: assets are lost if a node performs an illegal operation (security deposit)

https://www.linkedin.com/pulse/consensus-mechanisms-used-blockchain-ronald-chan