Trail: A Blockchain Architecture for Light Nodes

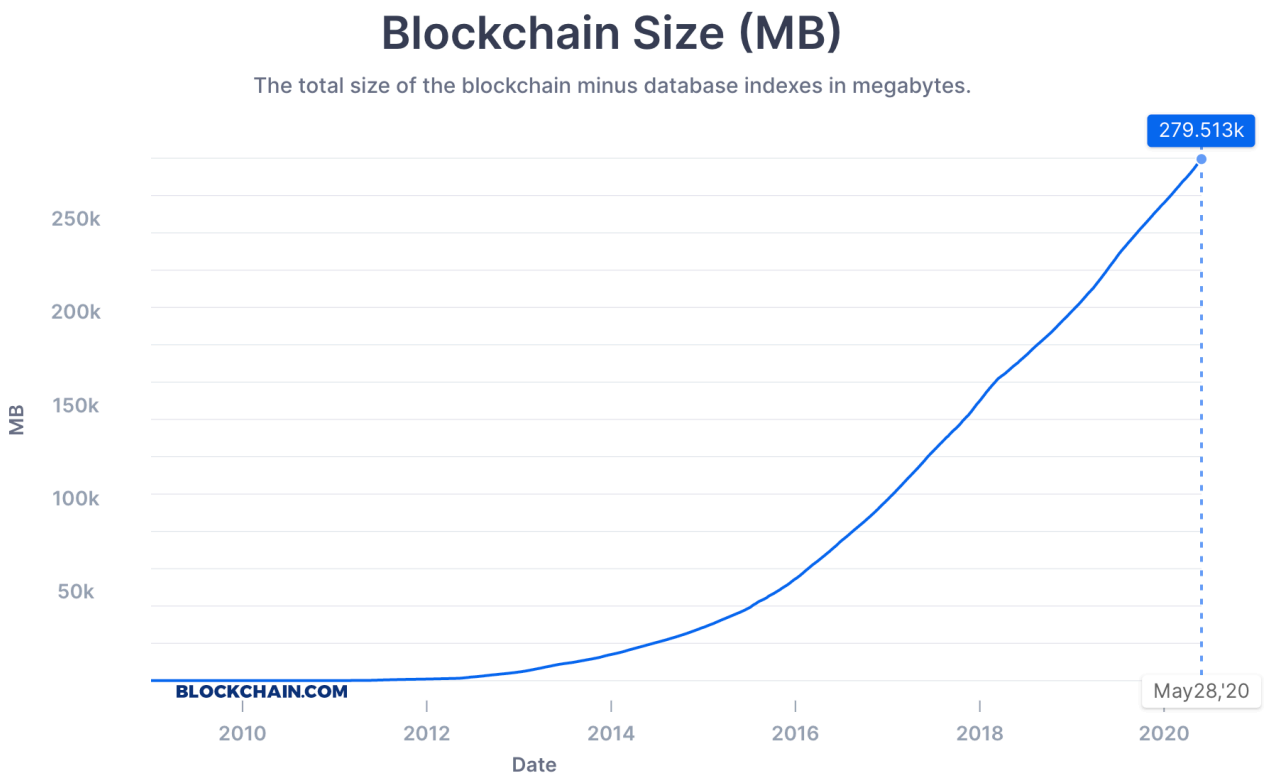
Ryunosuke Nagayama, Ryohei Banno, Kazuyuki Shudo

Tokyo Institute of Technology

Tokyo Tech

Blockchain size continues to increase.

- Bitcoin : 280 GB (June 2020)[1], Ethereum: 140 GB (June 2020)[2]

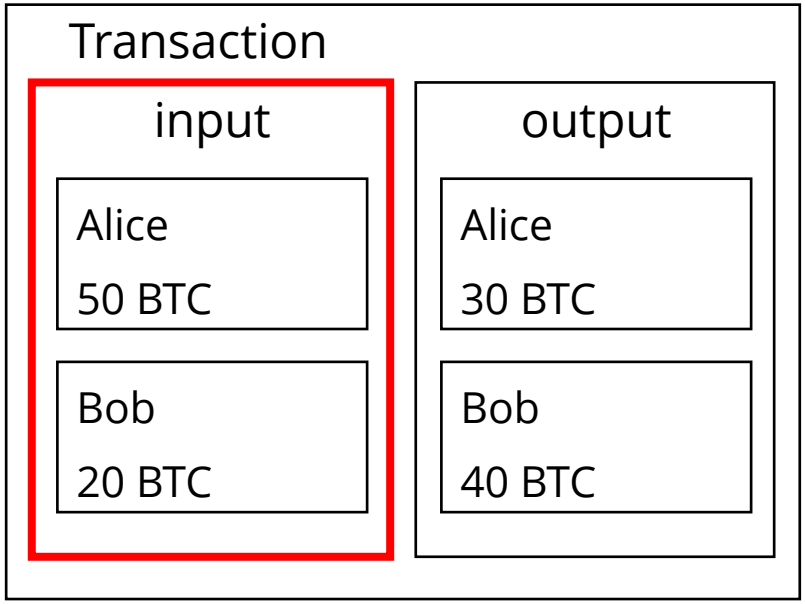


[1, figure] https://www.blockchain.com/charts/blocks-size
[2] https://blockchair.com/ethereum/charts/blockchain-size

Why is the blockchain so large?

Validators need to keep **all of proofs** for validation such as account states and transactions.

A Transaction in Bitcoin
(UTXO-based)



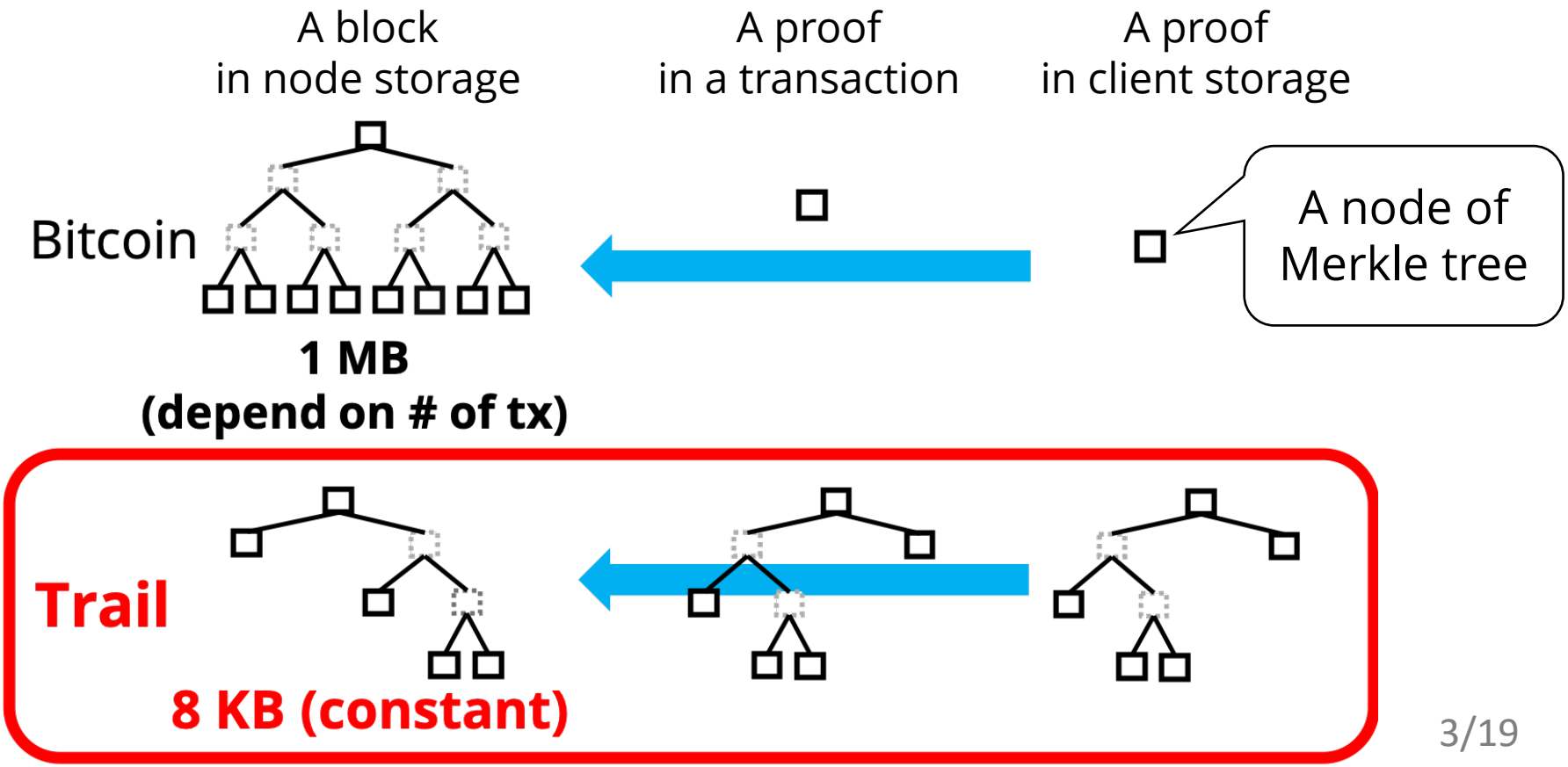
Validators refer to past transactions to validate whether UTXOs in input are already used as input.

Contribution

Clients keep proofs of own assets.

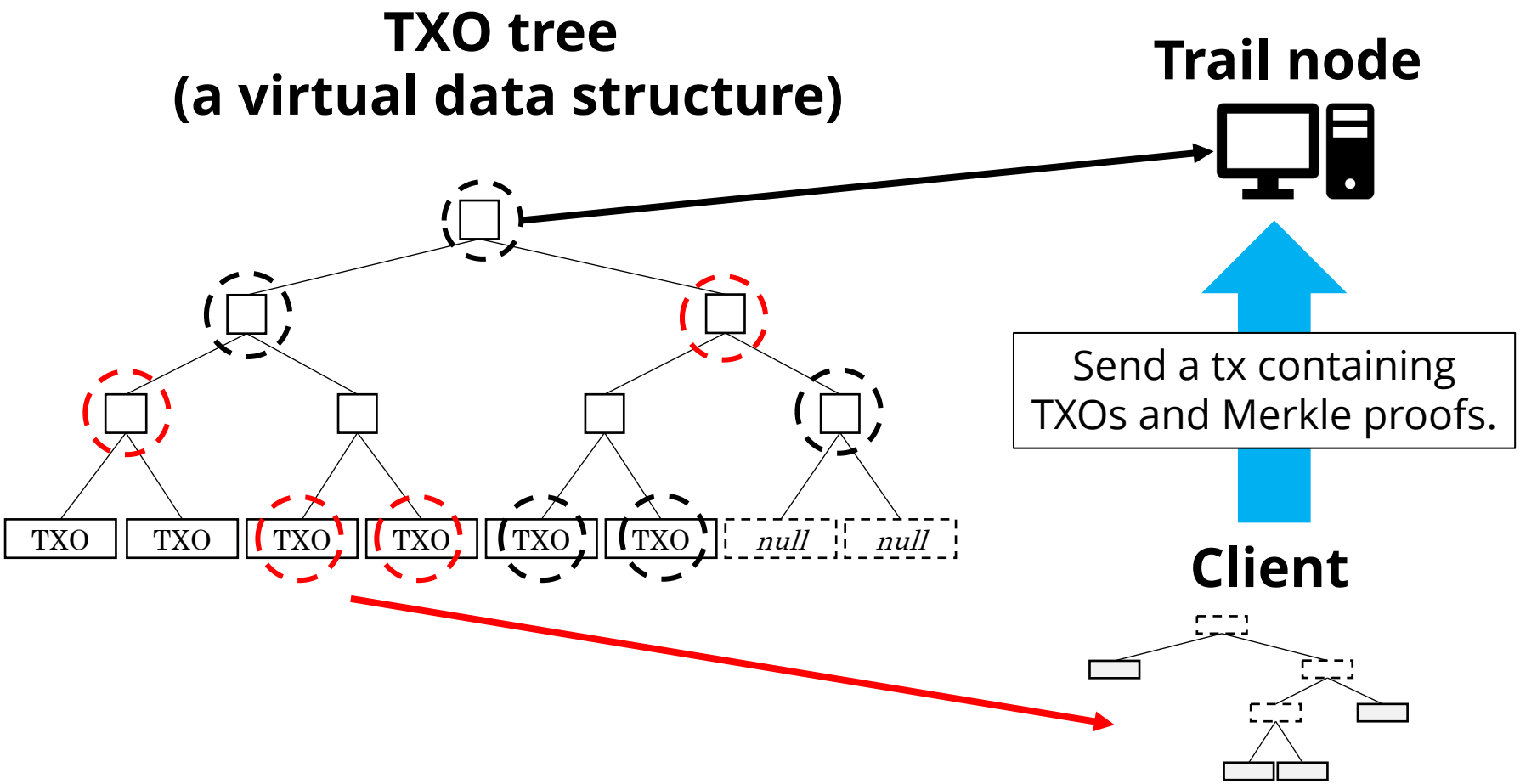
Block size is 8 KB, and it's constant.

Trail improves decentralization of a blockchain.



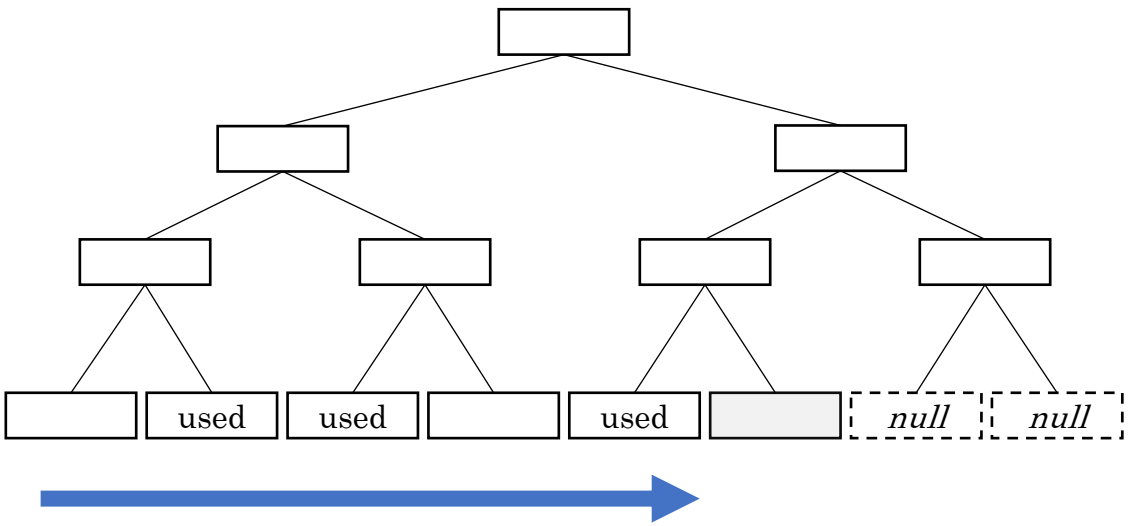
Trail

Trail reduces data on a node by including verification proofs in transactions instead of including the proofs in the block.



TXO tree

- Trail nodes and clients maintain a single TXO tree virtually. Blocks record the state of TXO tree as root hash.
- TXO tree is a perfect binary tree.



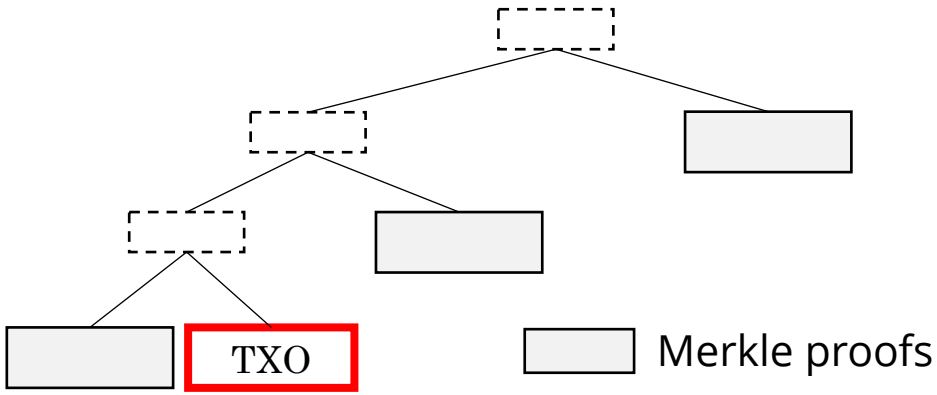
Insert a hash value of TXOs from the left.

All previously approved TXOs are assigned to leaf nodes.

If a leaf node has not been assigned TXO, the node stores $hash(null)$.

Generating a transaction

Clients keep own TXOs and update history of their Merkle proofs, and generate transactions from them.

Transaction	
BlockHash	Hash value of the block that Merkle proofs of this transaction are based on.
Inputs	
Outputs	New TXOs
Sigs	Signatures of senders

Validating transactions by Trail nodes

Trail nodes validate transactions using only the latest block.

Validation

1. Whether the block hash of the transaction is equal to the hash value of the latest block.

2. Whether each TXO in the Inputs of the transaction is not in another transaction to include in new block.

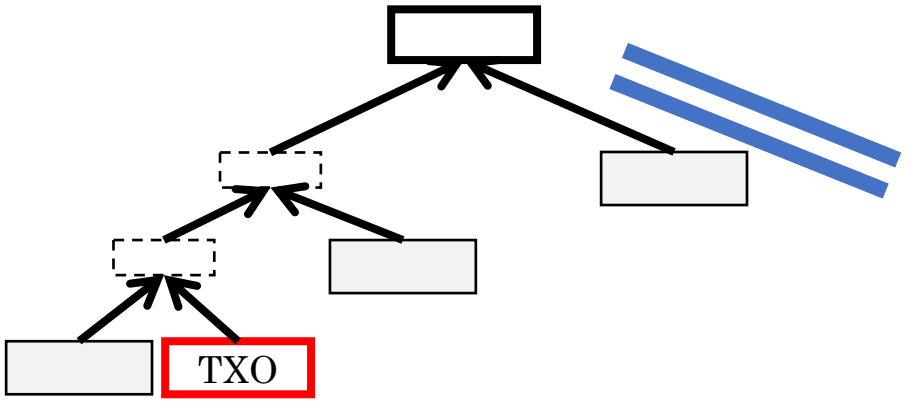
3. Whether the total value of Outputs is less than or equal to the total value of Inputs minus fees.

$$Inputs - fees > Outputs$$

Validating transactions by Trail nodes

Validation

4. Whether the root of the TXO tree calculated from the Merkle proof in the Inputs and the hash value of TXO $hash(TXO)$ is equal to the root of the latest block.



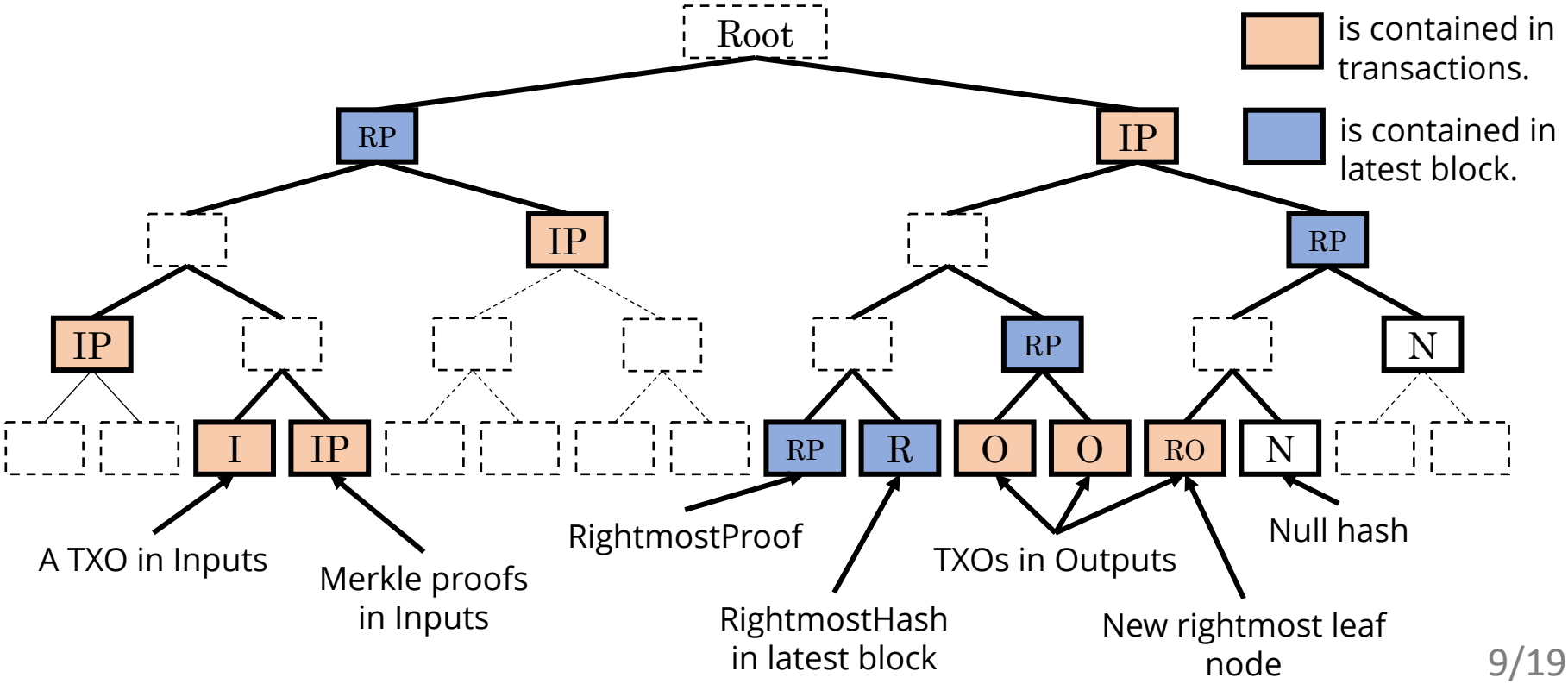
Latest block

Parent block hash
Root hash (of TXO tree)
RightmostIndex
…

5. Whether the Index of the TXO in the Inputs is less than or equal to the **RightmostIndex** in the latest block.

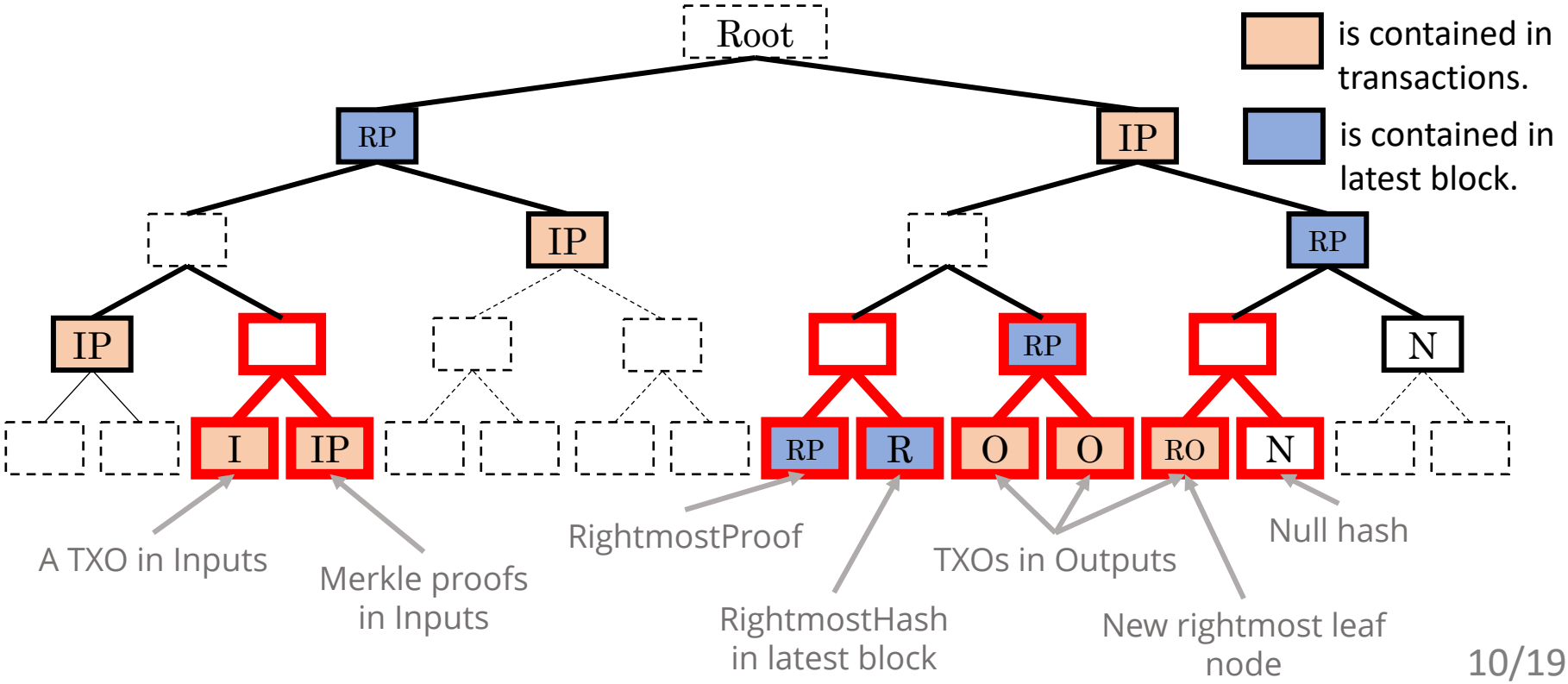
Generating a block by Trail node

- Assign TXO of Outputs from the left to the leaf node to which TXO is not assigned yet.
- After that, hash value of nodes in transactions and parent block are assigned to the corresponding nodes, and the Trail node calculates a new root.



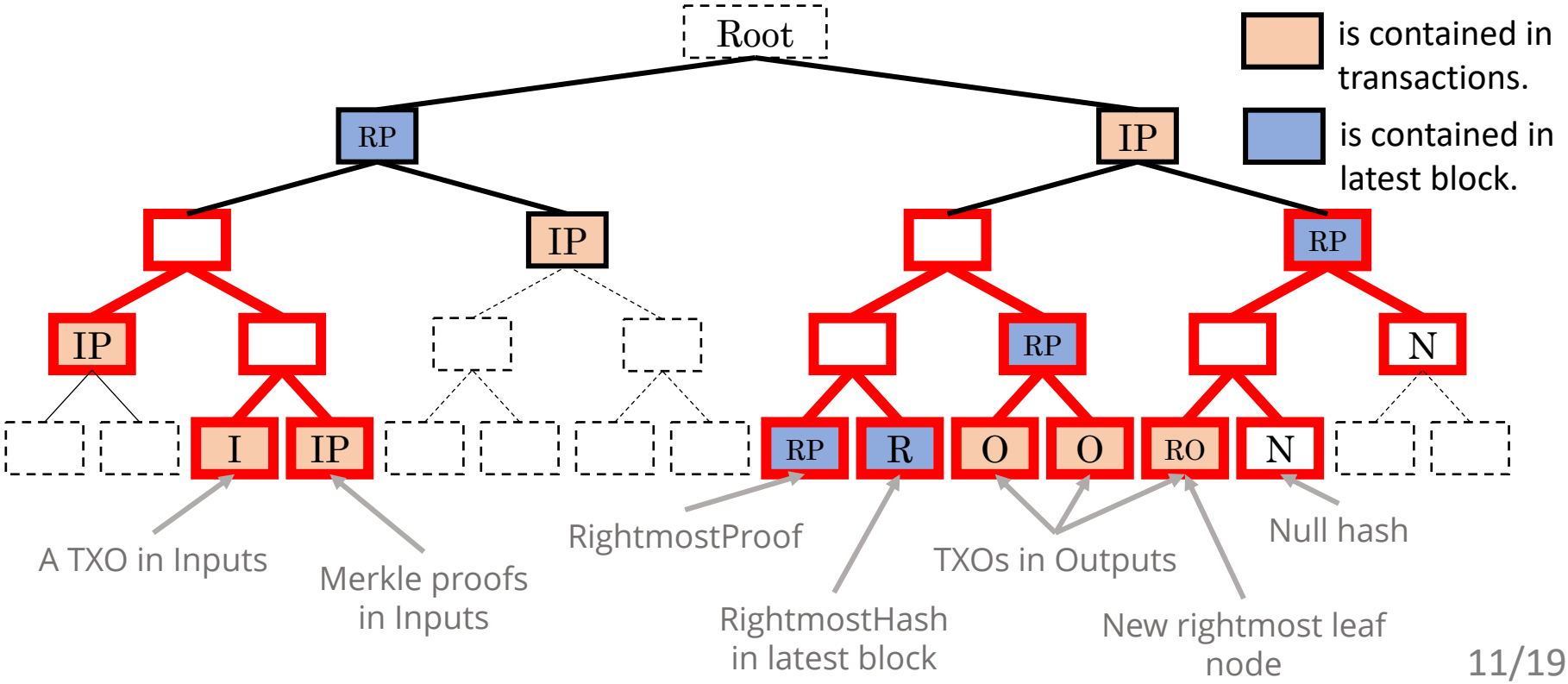
Generating a block by Trail node

- Assign TXO of Outputs from the left to the leaf node to which TXO is not assigned yet.
- After that, hash value of nodes in transactions and parent block are assigned to the corresponding nodes, and the Trail node calculates a new root.



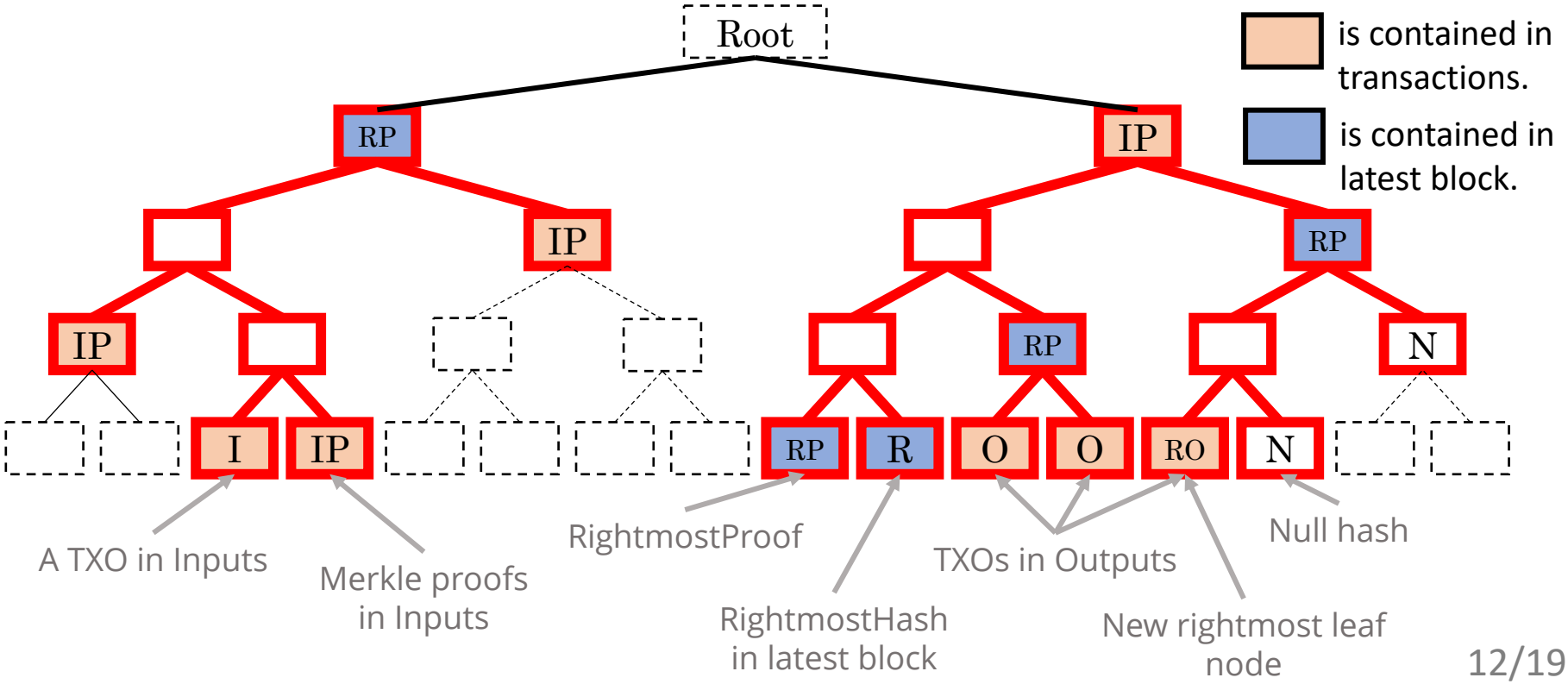
Generating a block by Trail node

- Assign TXO of Outputs from the left to the leaf node to which TXO is not assigned yet.
- After that, hash value of nodes in transactions and parent block are assigned to the corresponding nodes, and the Trail node calculates a new root.



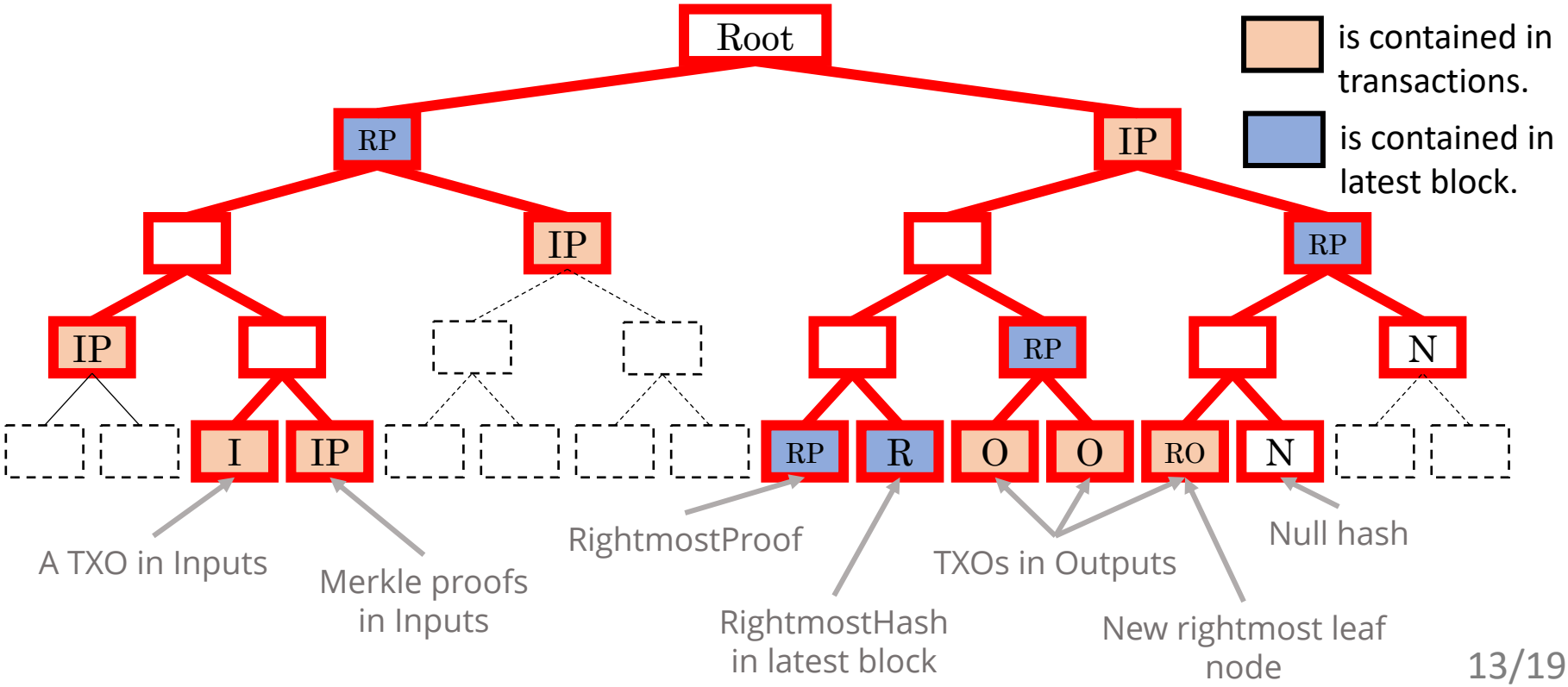
Generating a block by Trail node

- Assign TXO of Outputs from the left to the leaf node to which TXO is not assigned yet.
- After that, hash value of nodes in transactions and parent block are assigned to the corresponding nodes, and the Trail node calculates a new root.



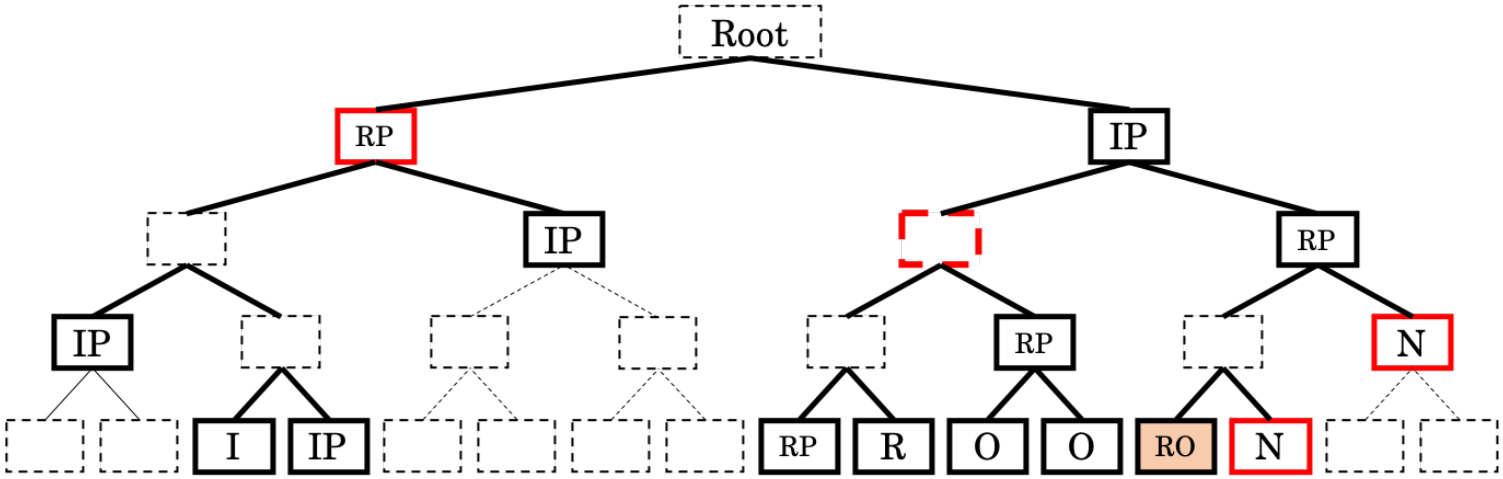
Generating a block by Trail node

- Assign TXO of Outputs from the left to the leaf node to which TXO is not assigned yet.
- After that, hash value of nodes in transactions and parent block are assigned to the corresponding nodes, and the Trail node calculates a new root.



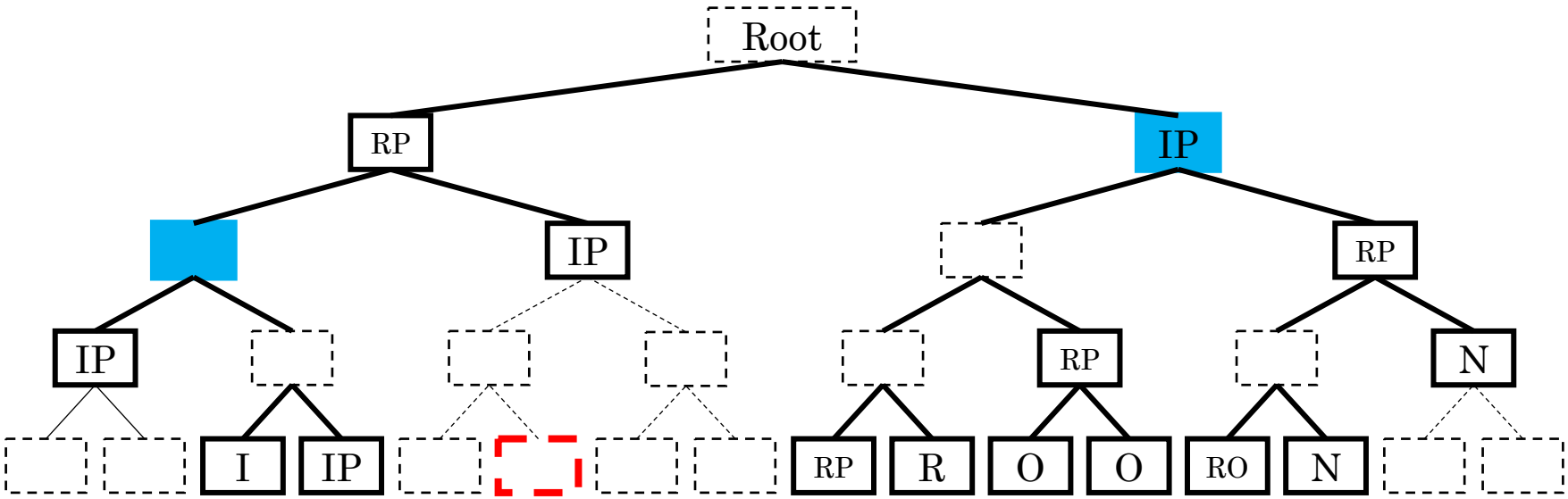
Generating a block by Trail node

Block	
Parent	Hash value of parent block
Root	New root of TXO tree Root
RightmostHash	Hash value of rightmost leaf node RO
RightmostIndex	Index of RO
RightmostProof	Merkle proof of RO =



Updating the client's data

- Clients need to update own TXOs and Merkle proofs to generate new transactions.



If a client owns the TXO assigned to , the client receive new hash values of and IP which are Merkle proof of .

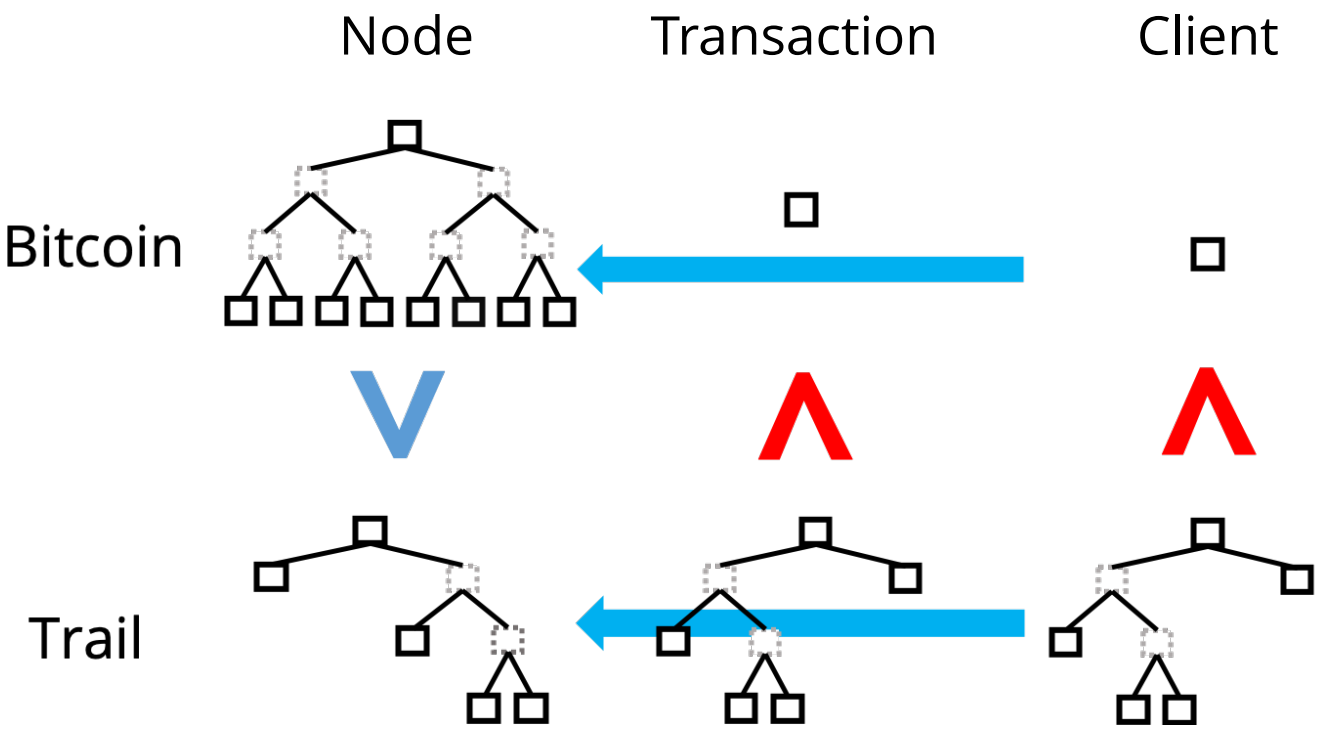
If a client owns I, the client marks the TXO used.

If a client owns O, the client keeps the TXO and its Merkle proof.

Data size optimization

Tx size and client storage size are large.

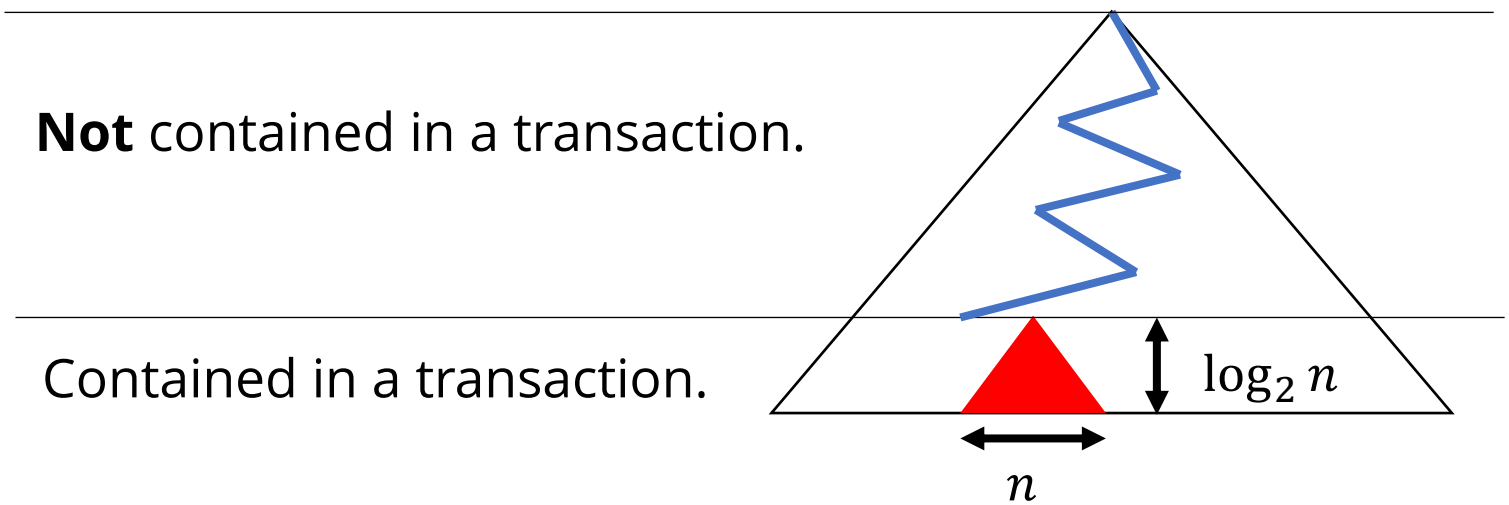
Client should optimize these data.



Transaction size optimization

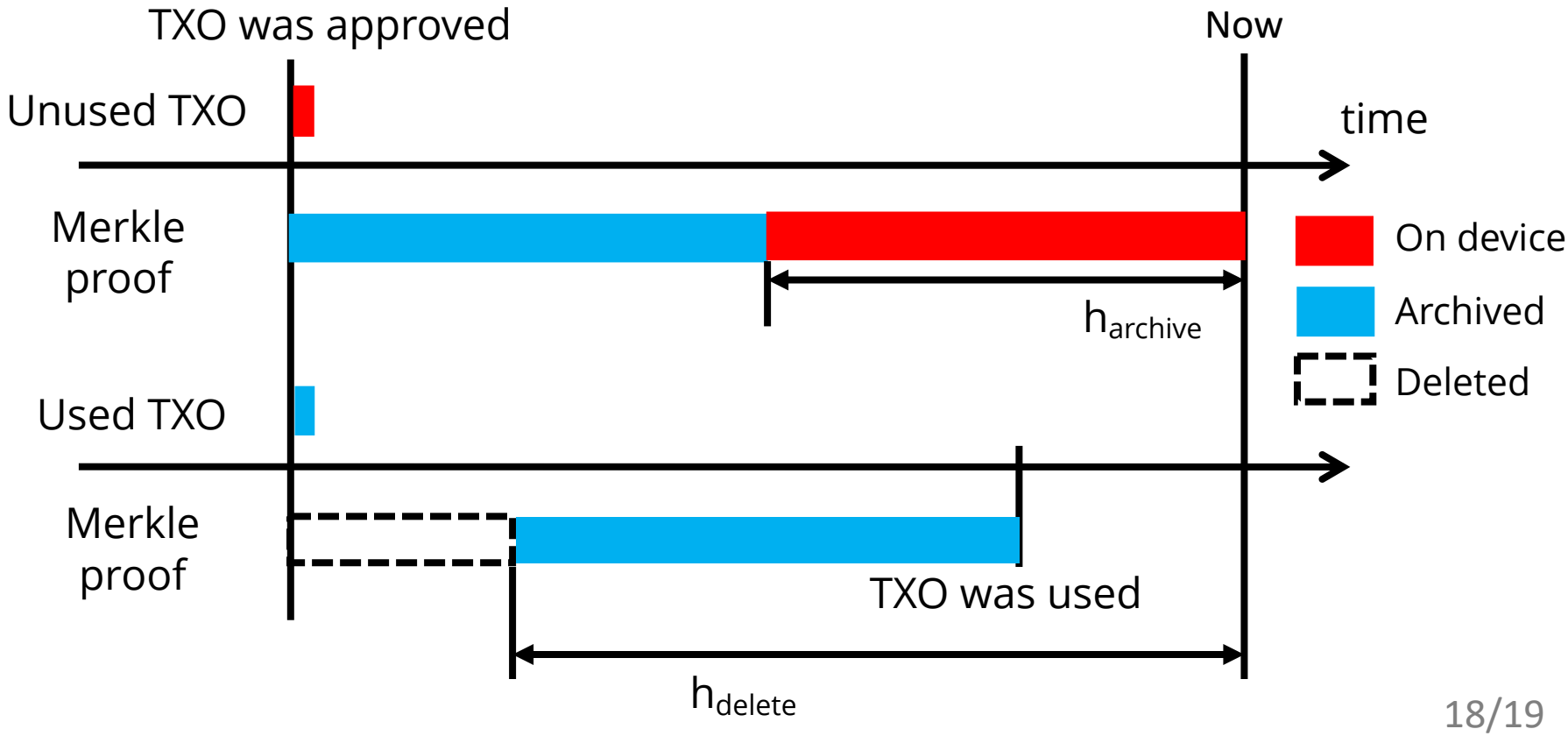
- The data size of a transaction is large, if a transaction contains all Merkle proofs of input TXOs.
  Ex. 8288 bytes/input TXO

- n TXOs are approved over a period of time.
- If index of a TXO is larger than RightmostIndex $- n$,
  nodes in Merkle proof of that TXO are same as RightmostProof at the height $\geq \log_2 n$.



Storage optimization

- Since the client needs to keep the TXO Merkle Proof update history, the client's storage size may be large.
- Client optimizes storage as follows;



Conclusion

Trail facilitates decentralization of a blockchain.

- Small blockchain size
  - Block size is only 8 KB.
    It's is constant regardless of number of transactions and accounts.
- Algorithm neutral
  - Trail can be applied to any consensus algorithm and fork choice rule.
- Works on mobile devices.