BlockScholes

10TH MARCH 2023

UTXO vs Account-Based Models

There are two popular accounting methods that blockchains choose to follow; the UTXO model, adopted by blockchains such as Bitcoin and Cardano, and Account-Based Models employed by chains like Ethereum and Binance Smart Chain. In this report, we discuss the similarities and differences between these two systems, and explore the subtle implications they have when accessing data.

Comments or questions: research@blockscholes.com

SUBSCRIBE TO OUR PLATFORM

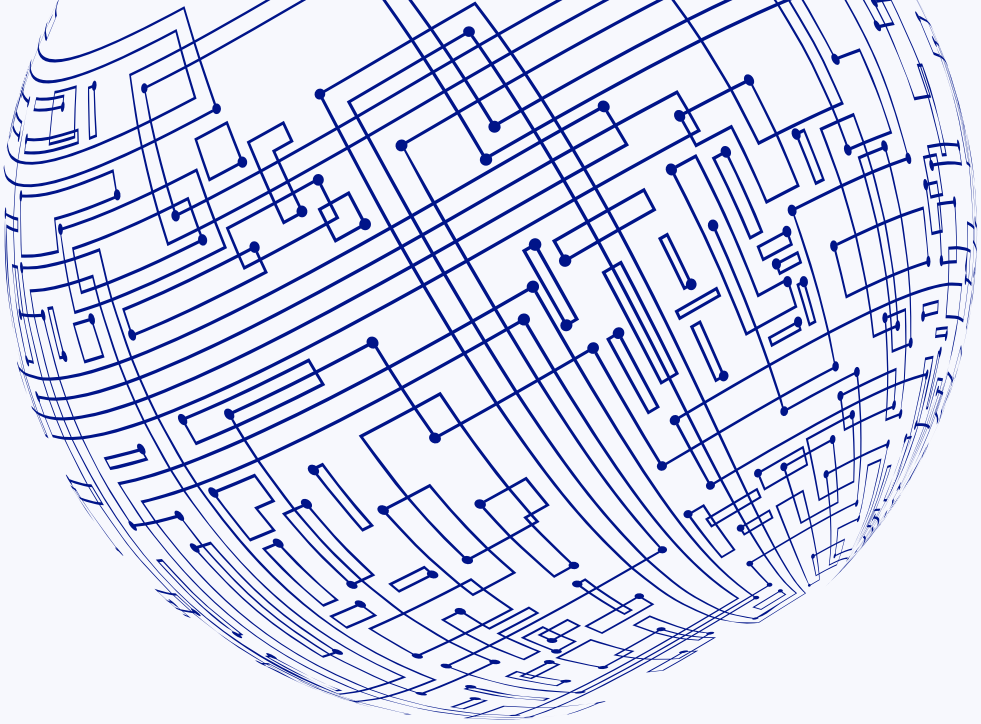


Table of contents

There is a key difference between how popular blockchains like Bitcoin and Ethereum operate their onchain accounting. This has implications for the data that is stored in the messages sent between nodes (miners for Bitcoin and proposers/validators for Ethereum) on the network, as well as how the end result (state of the network) is stored locally by those nodes. By the state of the network, we mean the result of executing every change specified by transactions since the genesis block of its chain. This is a sum of all the information conveyed by the blockchain up to a given snapshot in time.

In both cases, the actual messages sent by nodes contain changes to the state of the network, whereas the actual state is stored locally by anyone listening to the messages and is the result of applying all state changes to the genesis block. What is different between the two models that we will discuss is the structure of the state that the messages change.

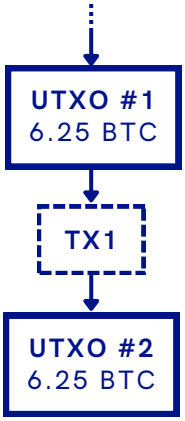
UTXO Model

The Unspent Transaction Output (UTXO) model is most famously employed by the Bitcoin network. The state of this system, stored locally by miners and updated with every block, does not contain information about account addresses or their balances. Instead, the network tracks Transaction Outputs, which are data objects that have a transaction amount (size in BTC of the transaction), a “to” address, and a reference to the address which initiated the transaction.

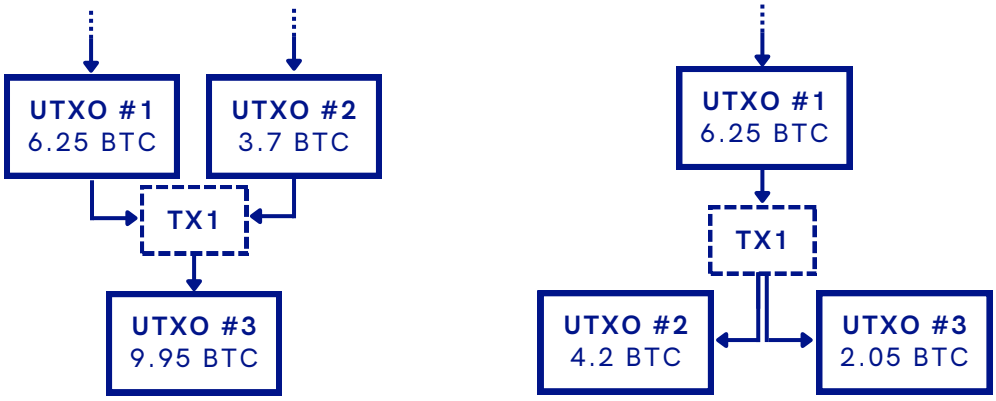
Unspent transaction output objects are used up in transactions (that are contained in blocks by miners), transforming them into spent transaction outputs and creating new UTXO objects. The state that is stored locally by the network’s nodes is the tree of transaction inputs and outputs, linked together by the transactions broadcast in blocks.

TRANSACTIONS

When submitting a transaction to the network, a user must include the receiver’s address, the amount of BTC to send, and a reference to the UTXO (or UTXOs) that their private key controls that they wish to use as an input.

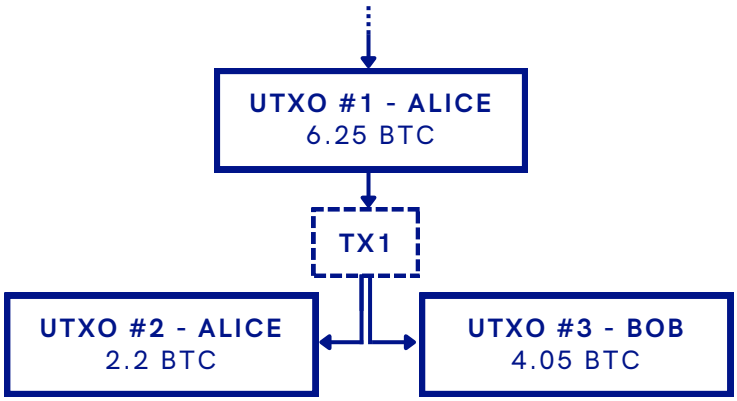


The amount of BTC across all UTXOs that are used as inputs to a transaction must equal the amount of BTC that the transaction outputs in new UTXOs. If the sender wishes to send only part of a UTXO to a recipient address, then their transaction creates 2 new UTXOs; one that goes to the recipient, and another that is returned to the sender's address. Multiple inputs mean that the BTC that the user is spending were received in multiple transactions and recombined into a single transaction. Multiple outputs mean that the BTC that the user is spending is being sent to multiple addresses.



CHANGE TRANSACTIONS

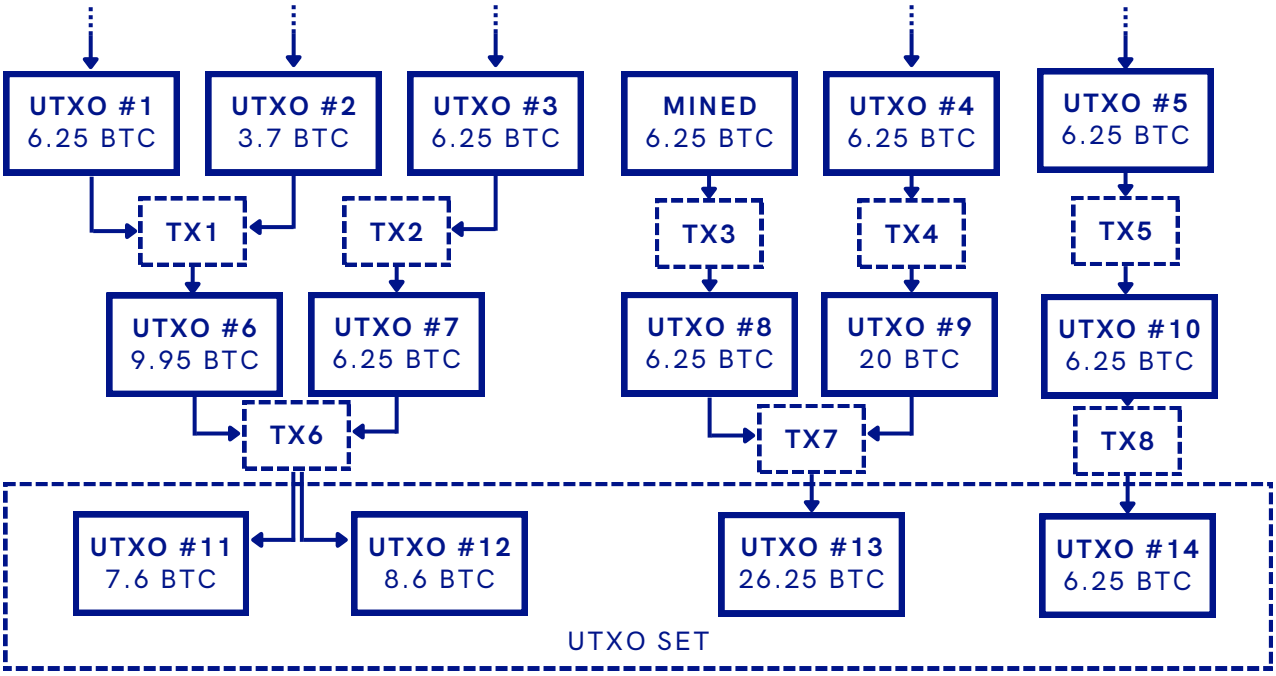
UTXOs must be used as a whole - users cannot use only some of the BTC in a UTXO for a transaction and "hold onto" the rest. If they wish to only spend part of the BTC in a UTXO, they must create two output transactions: one sending part of the UTXO's BTC to the intended recipient, and the other sending the remaining BTC in the UTXO back to themselves. This is often thought of in analogy to paying for an item with a ten-pound note - a customer spends all of the ten-pound note and receives change in return.



This model means that each UTXO in the current set can be traced back through each transaction that it has been used in. The origin of every BTC contained in UTXOs in the UTXO set is a transaction (or transactions, if UTXOs have been recombined) with no sender or parent UTXO, awarded to the successful miner of the block in which it is contained.

THE UTXO SET

Each Bitcoin network block contains a list of transactions that "use up" UTXOs" to create new ones. The set of transactions that have not yet been used as inputs to transactions onchain is called the "UTXO set".



This large list is not stored in the blocks created and published by miners, but is instead, the result of applying all transactions contained in each block in the chain to the genesis block. It is stored locally by all nodes listening to the blockchain, and used to determine the validity of a transaction submitted to the mempool. As multiple UTXOs can be combined as inputs to a single transaction, this UTXO set is not necessarily monotonically increasing - although whilst it can shrink, it has trended larger since Bitcoin's inception.

Ownership of a UTXO in the set is controlled by its recipient user's private key: if a user wants to use that UTXO as an input to a new transaction, they must send a (cryptographically) signed message to the network that can be decrypted using the address' public key. Only the holder of the private key can encrypt the data such that it can be decrypted using the public key, thereby proving their ownership and authorising its use in a new transaction.

Addresses are generated from public keys, which are in turn generated from a user's private key. There are many possible public keys and addresses derivable from a single private key. There is no way to link a set of public keys generated by the same private key without knowledge of the private key itself. This makes it difficult to track the sum of UTXOs owned by a "wallet" of addresses controlled by a single user.

Account Based Model

The account-based model bears great similarities to how traditional banks operate and maintain their ledgers. The Ethereum network is one such blockchain that uses the Account-based Model. A ledger containing a set of accounts and their associated assets is maintained and updated every time a transaction is made.

There are two types of accounts on the Ethereum blockchain; Externally Owned Accounts (EOA) and Contract Accounts (CA). EOA are accounts that users control directly and use to interact with the Ethereum network. A user wishing to make a transaction which sends tokens from their EOA to another account must initiate the transaction by signing with their private keys. Holding the private keys to an EOA automatically grants the user access to the assets associated with the account, much like how in a UTXO model the private key controls ownership of unspent transactions in the UTXO set.

A CA, on the other hand, is an account controlled exclusively by a smart contract. By this, we mean that no other no smart contract has access to another smart contract's Account. CAs are created and deployed either by users or by other smart contracts.

These accounts do not have private keys associated with them, which disqualifies them from initiating transactions of their own. Therefore, transactions made by CAs are always triggered by an EOA, which itself initiates a transaction by calling a function of the smart contract (an action itself contained within a transaction). The initiating EOA is responsible for paying any associated gas fees the CA transaction incurs when making its transaction. Despite this, CAs still have balances associated with them, and can send tokens they hold based on the rules specified by the smart contract itself.

BLOCK CONTENTS

Blocks on the Ethereum network do not contain a record of the states of each account. This is stored on the local machines of nodes and validators of the network. As with Bitcoin miners, validators are required to maintain this network state as it is a necessity to perform their duty to verify transactions. Ethereum validators do this by checking that the transition from a prior state to a new state is a valid state transition.

What blocks do contain is a record of transactions, which are simply instructions that detail state changes to be made to specific accounts that are referenced by the transaction. As such, the most recent block in the chain contains the most up-to-date state update instructions for specific accounts on the network.

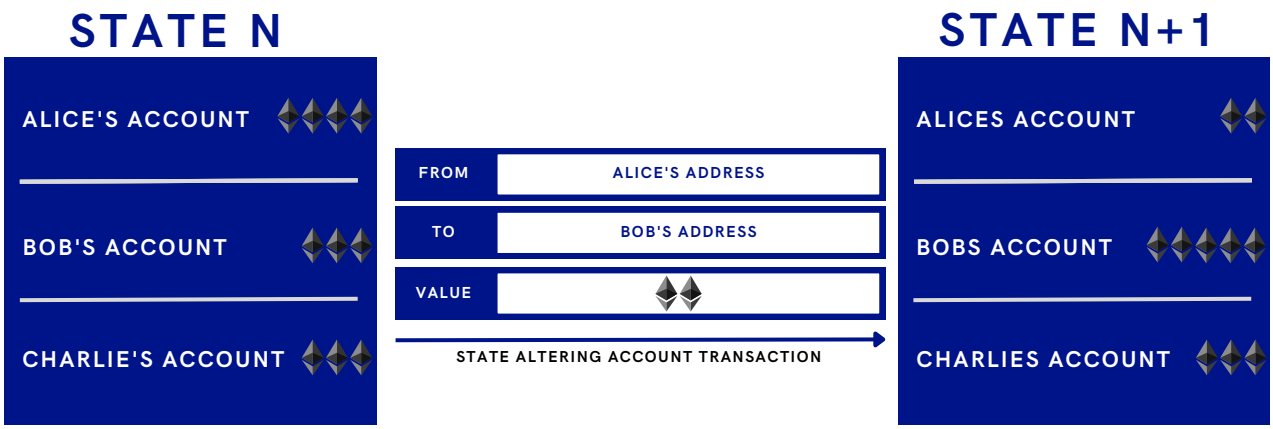
Any transaction which transfers tokens thereby changing address-balance pairs will result in a different network state being reached. In fact, all transactions sent on the network will result in a different network state due to gas costs associated with all transactions.

The sequence of blocks in the chain allows any node to reconstruct the state of any account on the network by sequentially executing transactions in the order they were presented in blocks, starting from the genesis block. As such, one can easily query the balance of a user at any given block by requesting this data from a node that stores this data without having to sum up all the incoming and outgoing transactions associated with the address themselves.

Conversely, in the UTXO model each UTXO in the set is linked to an address that cannot be linked to the other address controlled by its corresponding private key. As a result, there is no user entity recoverable from the blockchain data and a “wallet” of addresses is known only to the private key holder.

SENDING A TRANSACTION

Alice has the private keys to an EOA with 10 ETH and wants to send 1.75 ETH to Bob, another holder of an EOA with ETH. In making the transaction, Alice specifies Bob’s address as the recipient address, as well as the amount of ETH she wants to send to Bob. Then, she signs the transaction and broadcasts it to the network.



Validators on the network will check Alice's initial balance from their record of the most recent state and make sure that she has enough ETH to send to Bob. This check is not necessary in the UTXO model, as the miner must only verify that Alice controls the private keys of the address that holds the UTXO. If this verification is successful and the transaction is validated, 1.75 ETH (plus gas fees) is decremented from Alice's account and credited to Bob's account. The post-transaction Ethereum state will contain the updated balances of both accounts; Alice with 8.25 ETH and Bob with 1.75 ETH.

CONTACT US

AUTHORS

AHMAD MUSTAFA KIDA

Data Analyst

ahmad.kida@blockscholes.com

ANDREW MELVILLE

Research Analyst

andrew.melville@blockscholes.com

Block Scholes Ltd.

27 Old Gloucester Street
London WC1N 3AX

research@blockscholes.com

SUBSCRIBE TO OUR PLATFORM