Other details

Accounts

Accounts in Monad are identical to Ethereum accounts . Accounts use the same address space (20-byte addresses using ECDSA). As in Ethereum, there are Externally-Owned Accounts (EOAs) and Contract Accounts.

Transactions

The transaction format in Monad matches Ethereum , i.e. it complies with EIP-2718 , and transactions are encoded with RLP .

Access lists (EIP-2930) are supported but not required.

Linearity of Blocks and Transactions

Blocks are still linear, as are transactions within a block. Parallelism is utilized strictly for efficiency; it never affects the true outcome or end state of a series of transactions.

Gas

Gas (perhaps more clearly named "compute units") functions as it does in Ethereum, i.e. each opcode costs a certain amount of gas. Gas costs per opcode are identical to Ethereum in Monad, although this is subject to change in the future.

When a user submits a transaction, they include a gas limit (a max number of units of gas that this function call can consume before erroring) as well as a gas price (a bid, in units of native token, per unit of gas).

Leaders in the default Monad client use a priority gas auction (PGA) to order transactions, i.e. they order transactions in descending gas price order. In future times there may be alternative mechanisms for transaction ordering. The choice of order is orthogonal to everything that happens downstream; valid choice of order is not enshrined into the Monad protocol.

Previous Transaction lifecycle in Monad Next Using Monad Last updated5 months ago