Differences between Ethereum and OP Stack Chains

OP Stack chains are designed to be[EVM equivalent(opens in a new tab)] and introduces as few changes as possible to the Ethereum protocol. However, there are some minor differences between the behavior of Ethereum and OP Stack chains that developers should be aware of.

Opcodes

Opcode Solidity Equivalent Behavior COINBASE block.coinbase Returns the address of the current Sequencer's fee wallet. Effectively the same as Ethereum with the caveat that the value typically does not change from block to block. PREVRANDAO block.prevrandao Setpseudorandomly for each block by the Sequencer as opposed to the stronger guarantees provided by[RANDAO(opens in a new tab)] on Ethereum. ORIGIN tx.origin If the transaction is an L1 ⇒ L2 transaction triggered by a smart contract on L1, thentx.origin is set to the[aliased address] of the address that triggered the L1 ⇒ L2 transaction. Otherwise, this opcode behaves normally. CALLER msg.sender If the transaction is an L1 ⇒ L2 transaction triggered by a smart contract on L1, and this is the first call frame (rather than an internal transaction from one contract to another), the same[address aliasing] behavior applies.

Address Aliasing

Address aliasing is an important security feature that impacts the behavior of transactions sent from L1 to L2 by smart contracts. Make sure to read this section carefully if you are working with cross-chain transactions. Note that theCrossChainMessenger contracts will handle address aliasing internally on your behalf. When transactions are sent from L1 to L2 by an Externally Owned Account, the address of the sender of the transaction on L2 will be set to the address of the sender of the transaction on L1.However, the address of the sender of a transaction on L2 will be different if the transaction was triggered by a smart contract on L1 .

Because of the behavior of theCREATE opcode, it is possible to create a contract on both L1 and on L2 that share the same address but have different bytecode. Even though these contracts share the same address, they are fundamentally two different smart contracts and cannot be treated as the same contract. As a result, the sender of a transaction sent from L1 to L2 by a smart contract cannot be the address of the smart contract on L1 or the smart contract on L1 could act as if it were the smart contract on L2 (because the two contracts share the same address).

To prevent this sort of impersonation, the sender of a transaction is slightly modified when a transaction is sent from L1 to L2 by a smart contract. Instead of appearing to be sent from the actual L1 contract address, the L2 transaction appears to be sent from an "aliased" version of the L1 contract address. This aliased address is a constant offset from the actual L1 contract address such that the aliased address will never conflict with any other address on L2 and the original L1 address can easily be recovered from the aliased address.

This change in sender address is only applied to L2 transactions sent by L1 smart contracts. In all other cases, the transaction sender address is set according to the same rules used by Ethereum.

Transaction Source Sender Address L2 user (Externally Owned Account) The user's address (same as in Ethereum) L1 user (Externally Owned Account) The user's address (same as in Ethereum) L1 contract (usingOptimismPortal.depositTransaction) L1_contract_address + 0x1111000000000000000000000000000000001111

Transactions

Transaction Fees

Transactions OP Stack chains must pay for an[L1 data fee] on top of the standard[execution gas fee] you would expect on Ethereum. Refer to the guide on[OP Stack Transaction Fees] for more information.

EIP-1559 Parameters

The base fee on OP Stack is, like Ethereum, computed via the[EIP-1559(opens in a new tab)] mechanism. The EIP-1559 parameters used by OP Stack differ per chain.

Mempool Rules

By default, OP Stack chains do not have a large public mempool like Ethereum. OP Stack mempools are typically only visible to the Sequencer of the given chain and transactions are generally executed in priority fee order (highest fee first). This is not a required behavior and certain chains may choose to have a public mempool.

Getting Started: OP Stack Superchain Explainer