

Inclusion Gas:

Inclusion gas refers to the gas charged regardless of whether the transaction is executed. It ensures that, under all circumstances, the block always covers the inclusion cost.

state_transition ()

Determine
Inclusion cost
(check_block_static)

Validate the sponsor
commitment of the coinbase

This involves reviewing all transactions to calculate their inclusion costs. "Inclusion Cost" refers to the fee required by the protocol to post certain data (potentially representing non-executable transactions) on-chain..

Check signature:
`Address(Signature(header)) == Coinbase`

To prevent replay attacks, the coinbase commits to the block's header, signaling its willingness to sponsor gas fees.

Determine transaction inclusion cost
(*check_transaction_static*)

Inclusion Gas Handling:

Inclusion gas, which covers the gas for calldata and blobs, is paid upfront by the block's coinbase. This creates a clear disincentive for the coinbase to include transactions with issues that might require skipping execution at a later stage.

apply_body()

Coinbase pays inclusion cost

Determine Gas Available

check_transaction()

Base Fee Checks:

Can the sender afford the base fee? We verify two conditions:

- The maximum transaction fee per gas is sufficient.
- The sender's balance is adequate to cover the cost.

Check tx can pay the base fee

YES

NO

$(\text{Calldata} + \text{Blobs}) * \text{base_fee}$

Coinbase account -
total_inclusion_cost

gas_limit
- total_inclusion_gas

We subtract the total inclusion gas of all transactions from the total gas available. This ensures that no transaction can later use the gas required for inclusion during execution.

$\text{tx.max_fee_per_gas}$
 $< \text{base_fee_per_gas}$

Here we check if the transactions specified gas price is high enough for the block's base fee. If that's not the case, the transaction can still be sponsored by the block's coinbase.

We now check if the sender can







