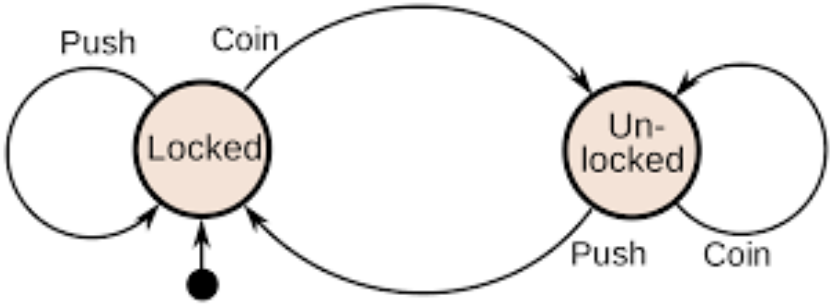
EVM and Smart Contracts

Ethereum and Programmability

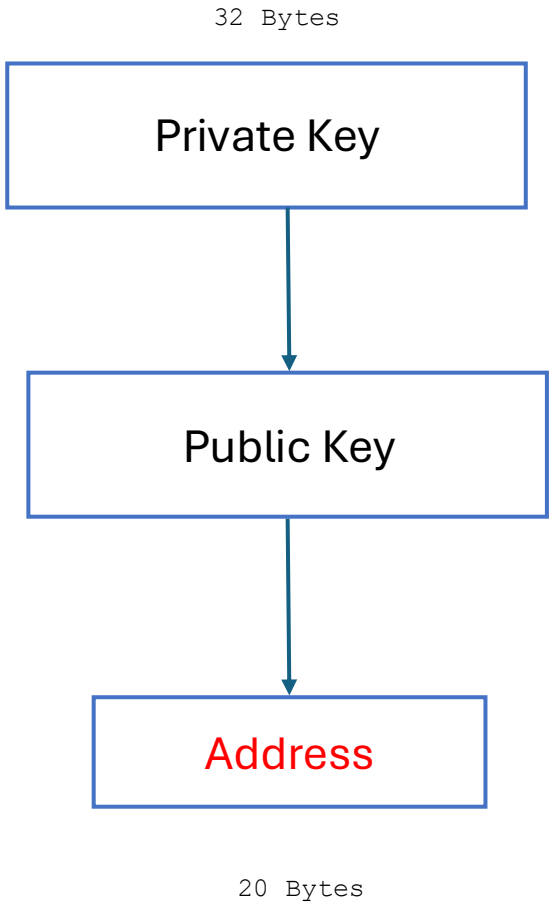
- Conceived in 2013 by Vitalik Buterin
- Proof of Stake
- Unlike Bitcoin which was primarily designed to be digital money, Ethereum is a global virtual computer running programs without the need for a central server.
- The programs stored in blockchain are called smart contracts
- Limitless possibilities

State Machine

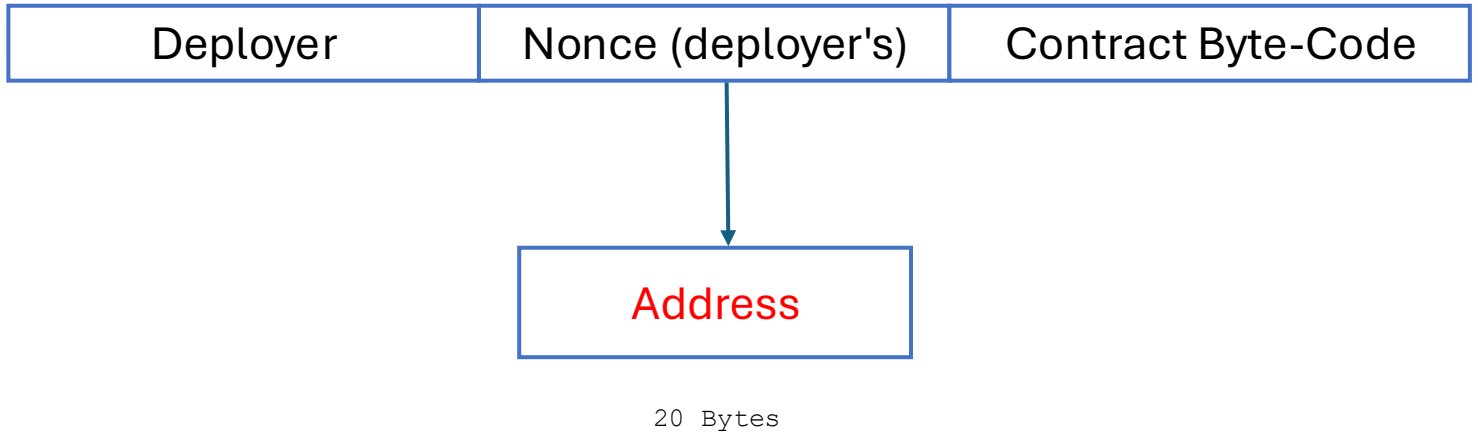


- Initially at state s_0
- Changes state based on input.
- Output is the function of the input and current state

Address

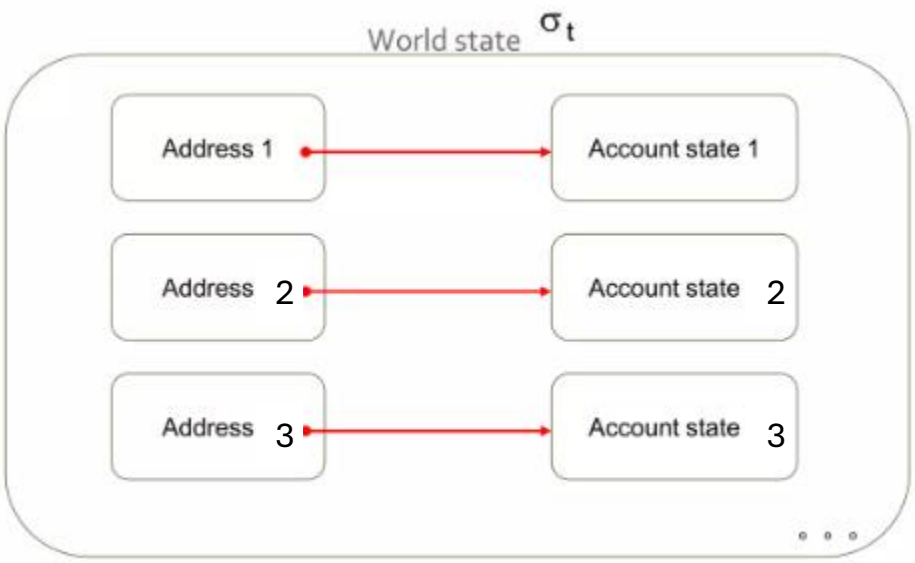


External Owned Address

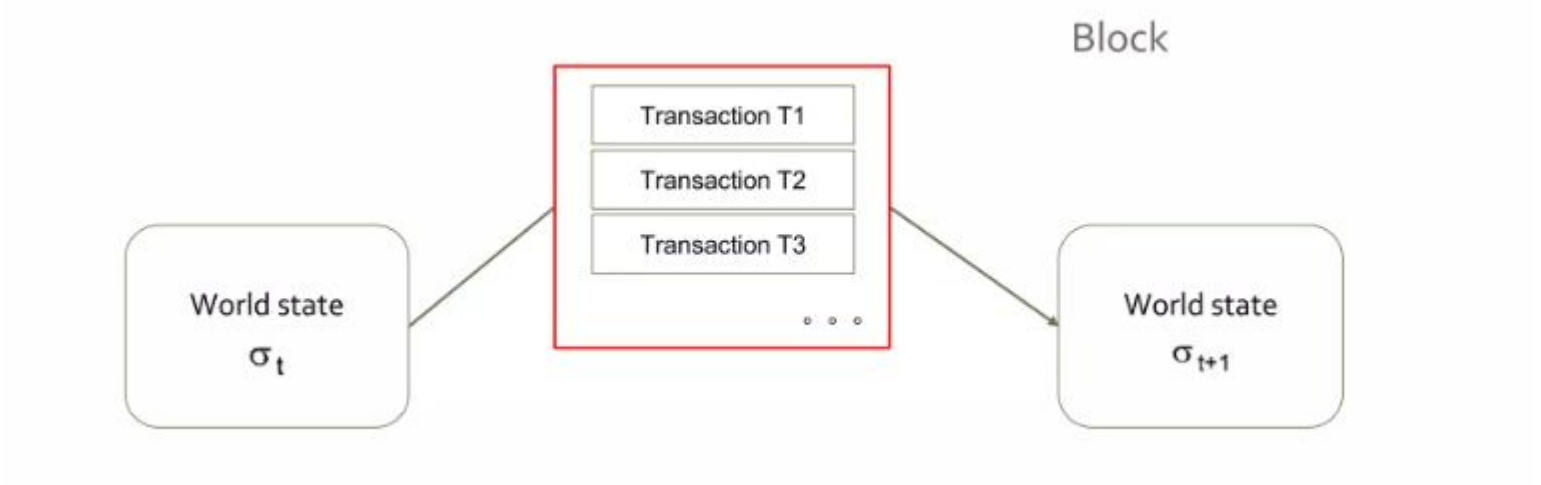


Contract address

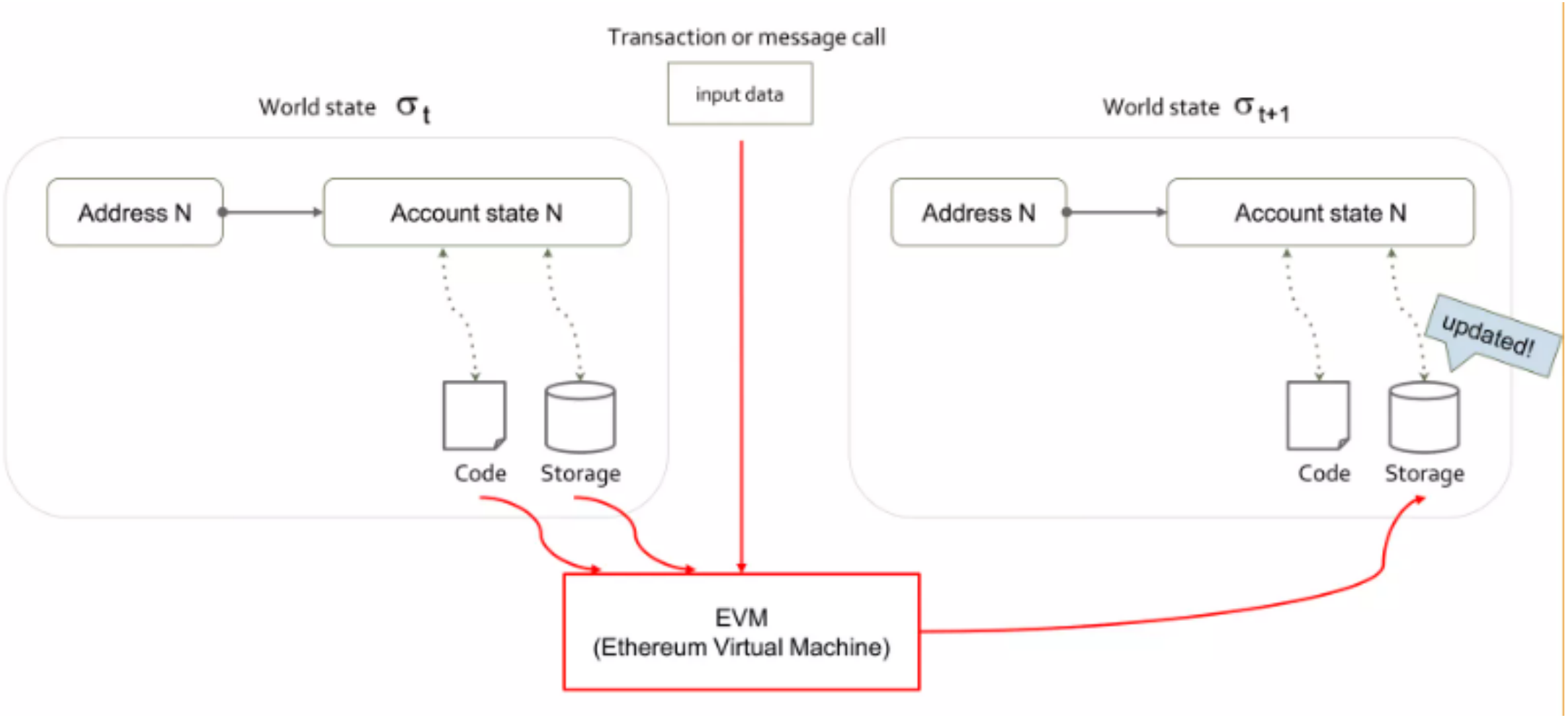
Ethereum's state



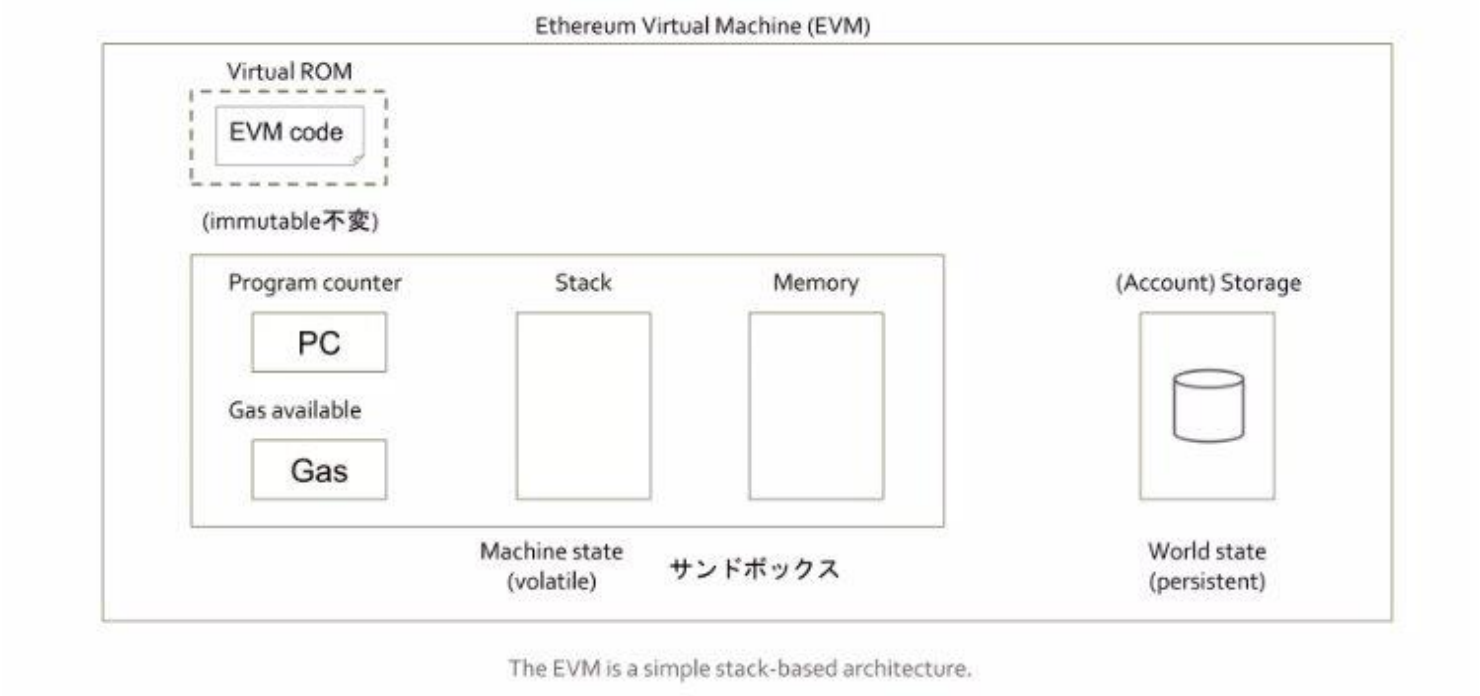
Transaction changes the state



EVM

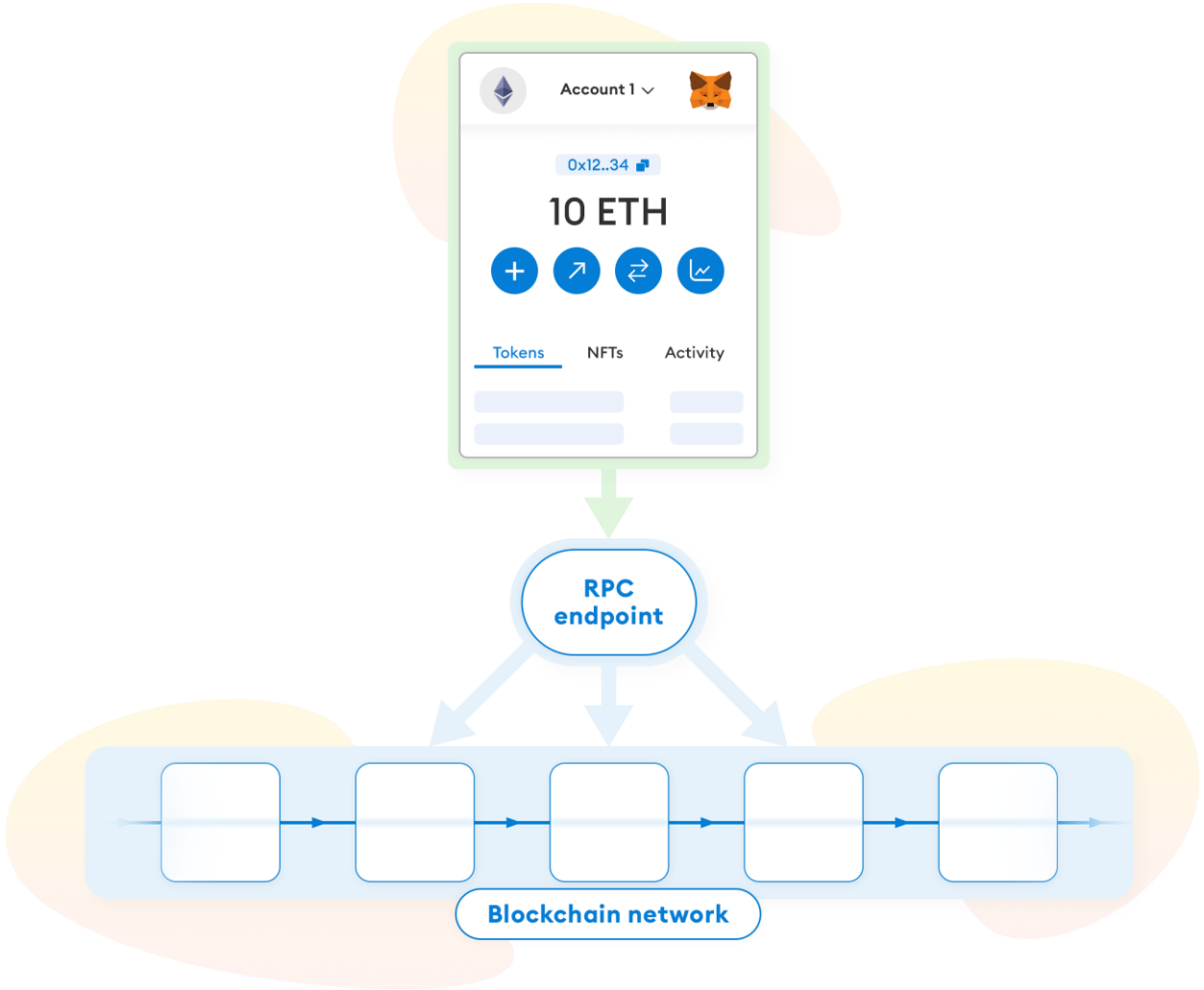

Transaction or message call
input data
World state σ_t
Address N
Account state N
Code
Storage
World state σ_{t+1}
Address N
Account state N
updated!
Code
Storage
EVM
(Ethereum Virtual Machine)

EVM Architecture



The EVM is a simple stack-based architecture.

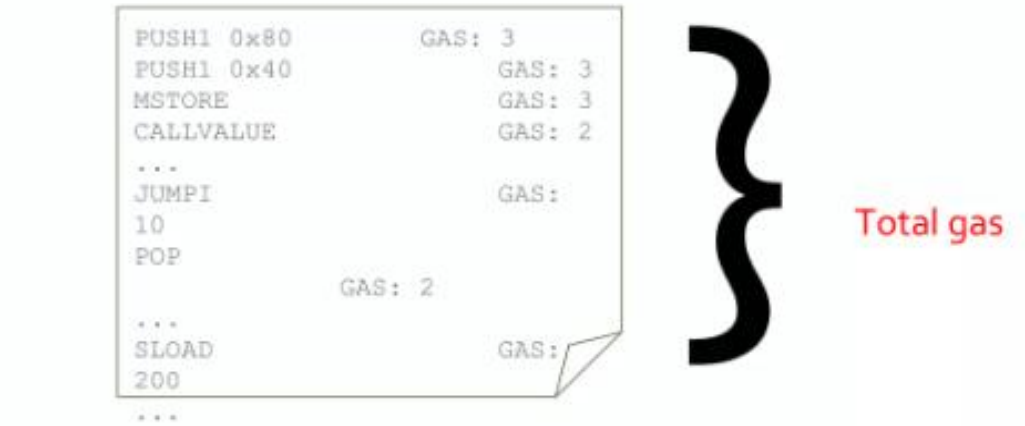
Off chain and on chain

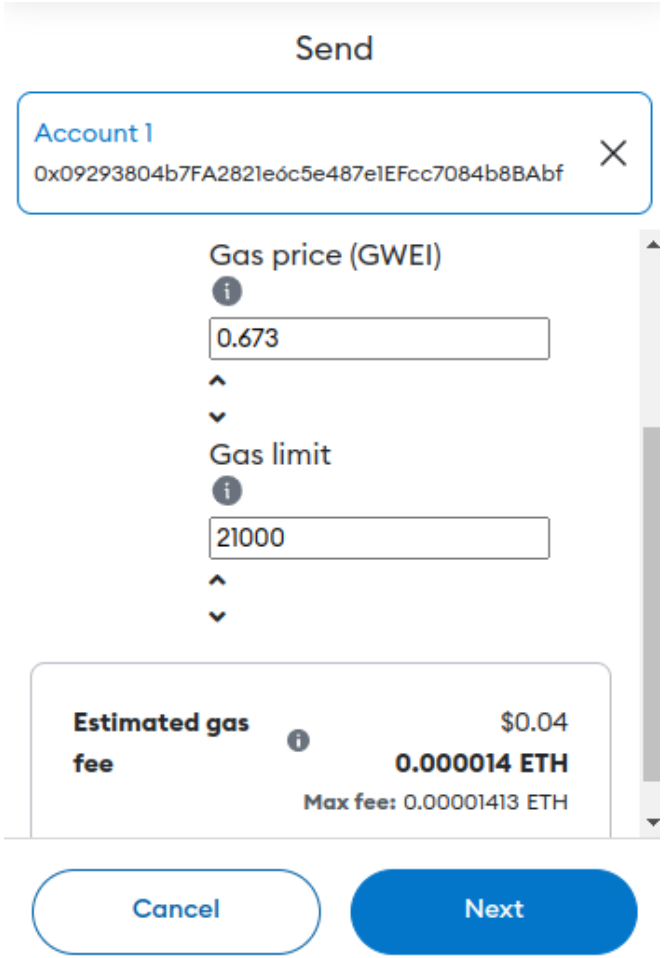


Wallet

- Program that manages your keys
- Signs the transaction with your private key
- Portal between the off-chain and on-chain component.
- Manages your assets

Gas and fee


PUSH1 0x80 GAS: 3
PUSH1 0x40 GAS: 3
MSTORE GAS: 3
CALLVALUE GAS: 2
...
JUMPI GAS:
10
POP
GAS: 2
...
SLOAD GAS:
200
...
Total gas
Real gas consumed
Total gas
Gas price
Fee
Gas limit
The maximum amount of gas sender willing to pay


Send
Account 1
0x09293804b7FA2821e6c5e487e1EFcc7084b8BAbf
Gas price (GWEI)
0.673
Gas limit
21000
Estimated gas fee
$0.04
0.000014 ETH
Max fee: 0.00001413 ETH
Cancel
Next

Smart contracts

- Collection of code and data that resides at a particular address in Ethereum.
- Immutable program that runs in EVM
- Identified by an address on chain.

Solidity

- A statically-typed curly-braces programming language designed for developing smart contracts that run on Ethereum
- Object oriented and high level language
- Gets compiled to opcodes that EVM can understand.

Example code

```
1  // SPDX-License-Identifier: GPL-3.0
2
3  pragma solidity >=0.8.2 <0.9.0;
4
5  /**
6   * @title Storage
7   * @dev Store & retrieve value in a variable
8   * @custom:dev-run-script ./scripts/deploy_with_ethers.ts
9   */
10 contract Storage {
11
12     uint256 number;
13
14     /**
15      * @dev Store value in variable
16      * @param num value to store
17      */
18     function store(uint256 num) public {    22520 gas
19         number = num;
20     }
21
22     /**
23      * @dev Return value
24      * @return value of 'number'
25      */
26     function retrieve() public view returns (uint256){    2415 gas
27         return number;
28     }
29 }
```