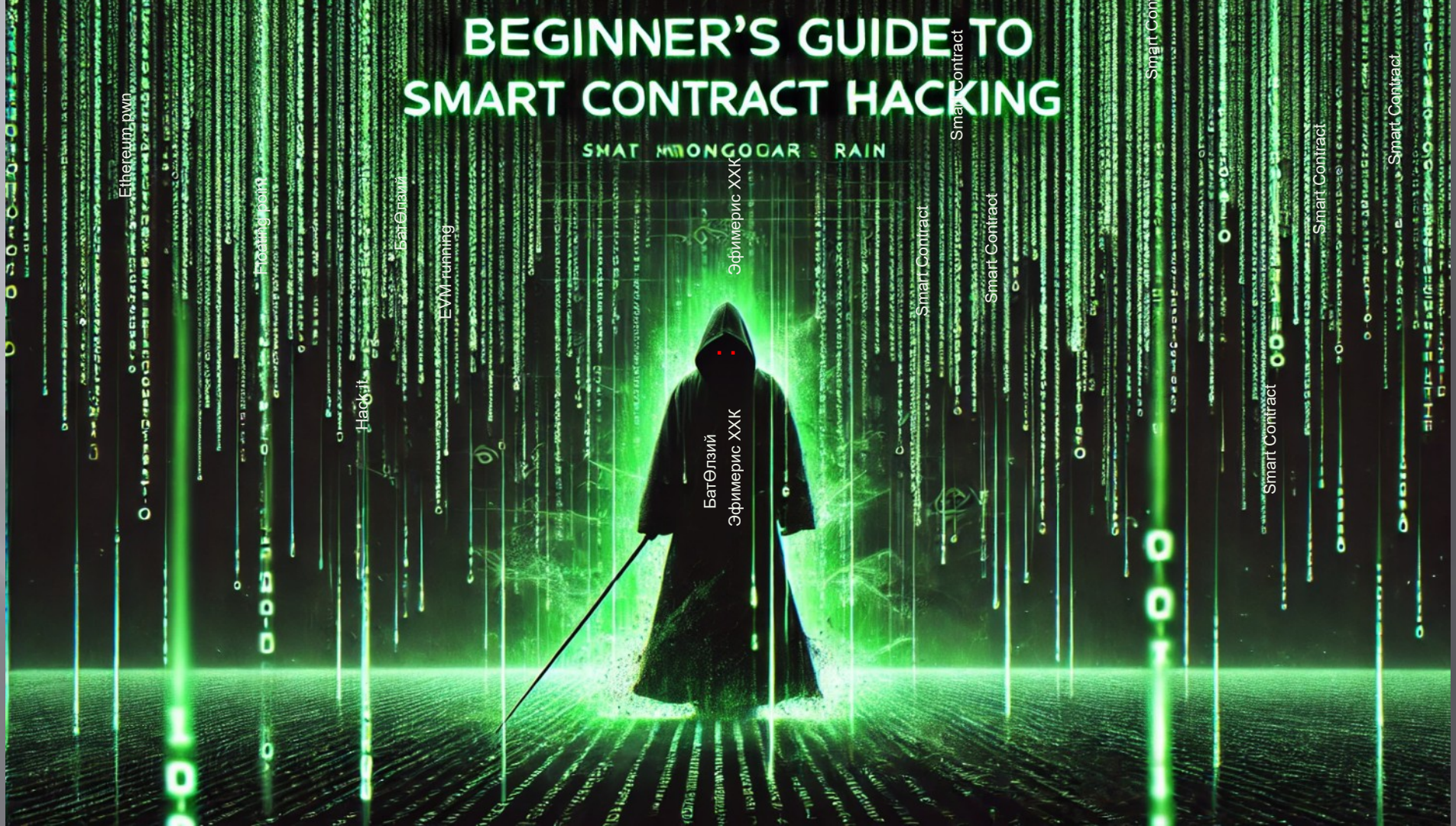

BEGINNER'S GUIDE TO
SMART CONTRACT HACKING
БатЭлзий
Эфимерис ХХК

root@kali:~ #

File Actions Edit View Help

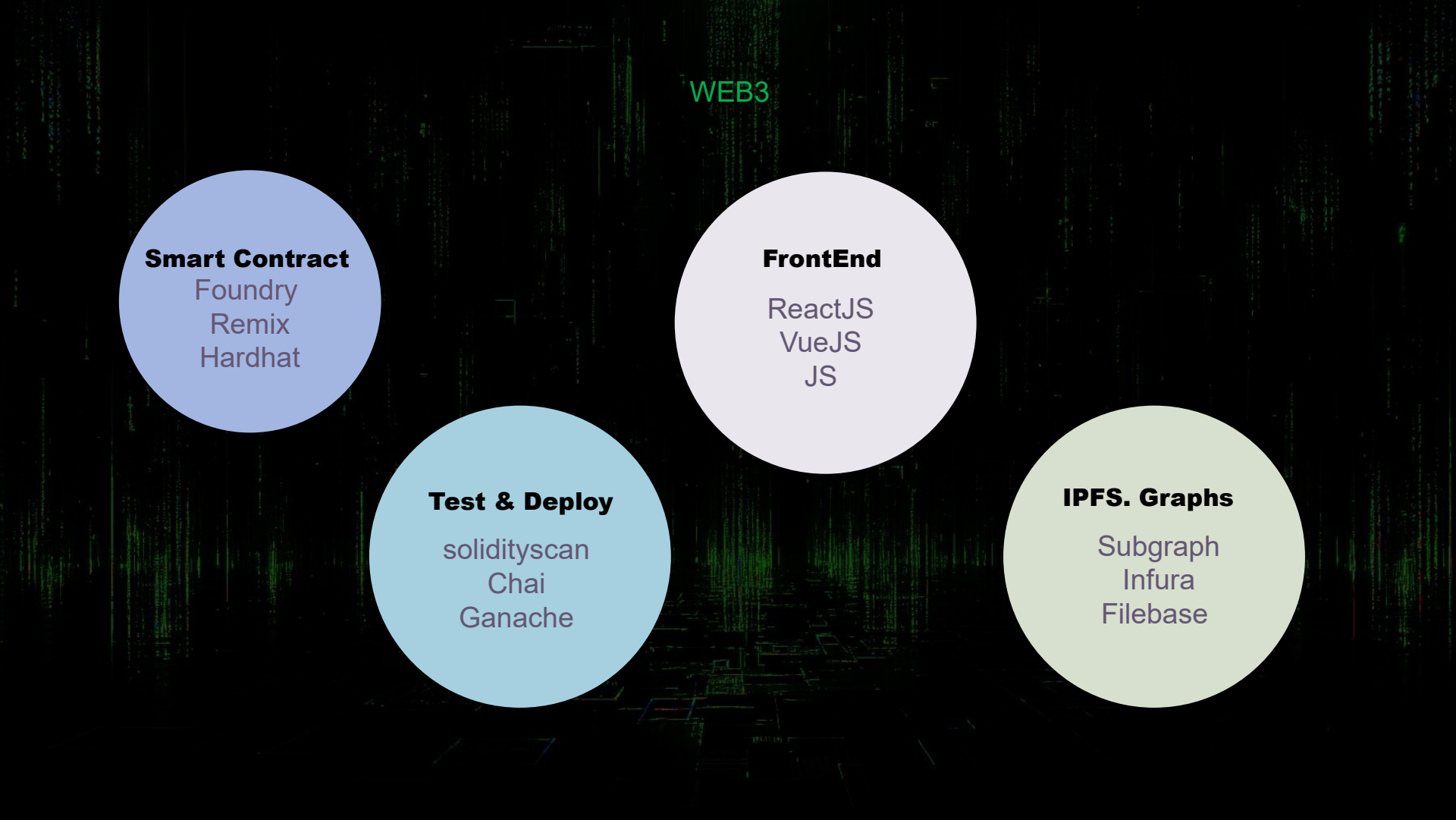
root@kali:~ # Introduction to Smart Contracts

root@kali:~ # Understanding Ethereum and EVM

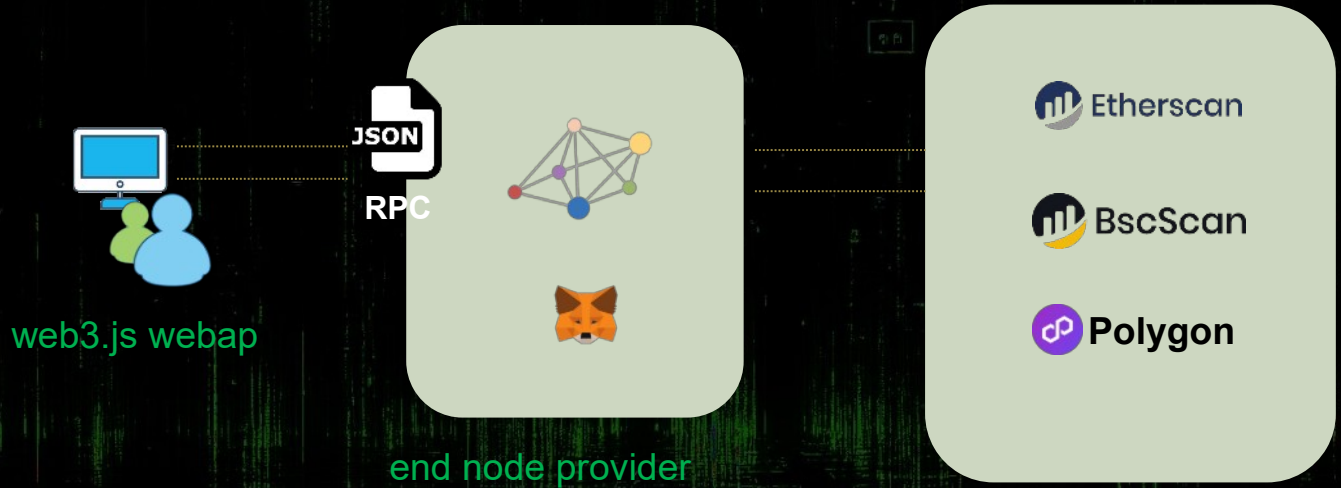
root@kali:~ # Smart contract development

root@kali:~ # Common Vulnerabilities in Smart Contracts

root@kali:~ # Real-Life Smart Contract Hacking Cases


WEB3
Smart Contract
Foundry
Remix
Hardhat
FrontEnd
ReactJS
VueJS
JS
Test & Deploy
solidityscan
Chai
Ganache
IPFS. Graphs
Subgraph
Infura
Filebase

WEB3



web3.js webap

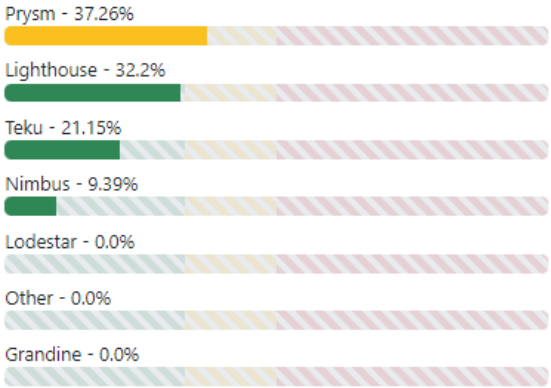
end node provider

Views and interacts

	FULL NODE	ARCHIVE NODE	LIGHT NODE
Data	Stores all blockchain Data.	Stores all blockchain data	Stores block headers
State	For the last 128 blocks	For all the Blocks in the Blockchain	Does not store state
Best for	Querying all Etheruem Blockchain data Serving as a validator	Querying historical Blockchain states	Basic blockchain data from headers Participating in Ethereum network
Hardware	Fast CPU with 4+ cores 16GB RAM, SSD 2TB Fast (NVMe)	SDD Fast (NVMe) Geth 14TB, Erigon: 3TB 25Mb/s bandwidth	CPU with 2+ cores 4GB RAM, SSD 50GB

Consensus Clients

Client diversity has improved!

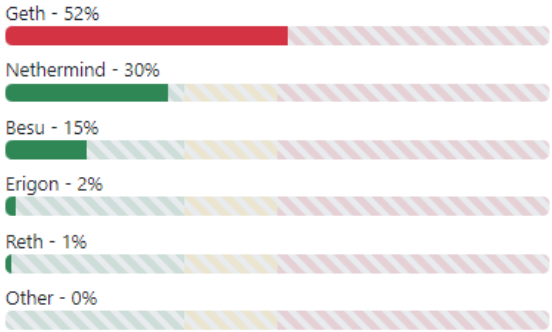


Data provided by Sigma Prime's Blockprint — updated daily.
Data may not be 100% accurate. (Read more)

Data source (read more):
Sigma Prime's Blockprint | Miga Labs | Rated.Network

Execution Clients

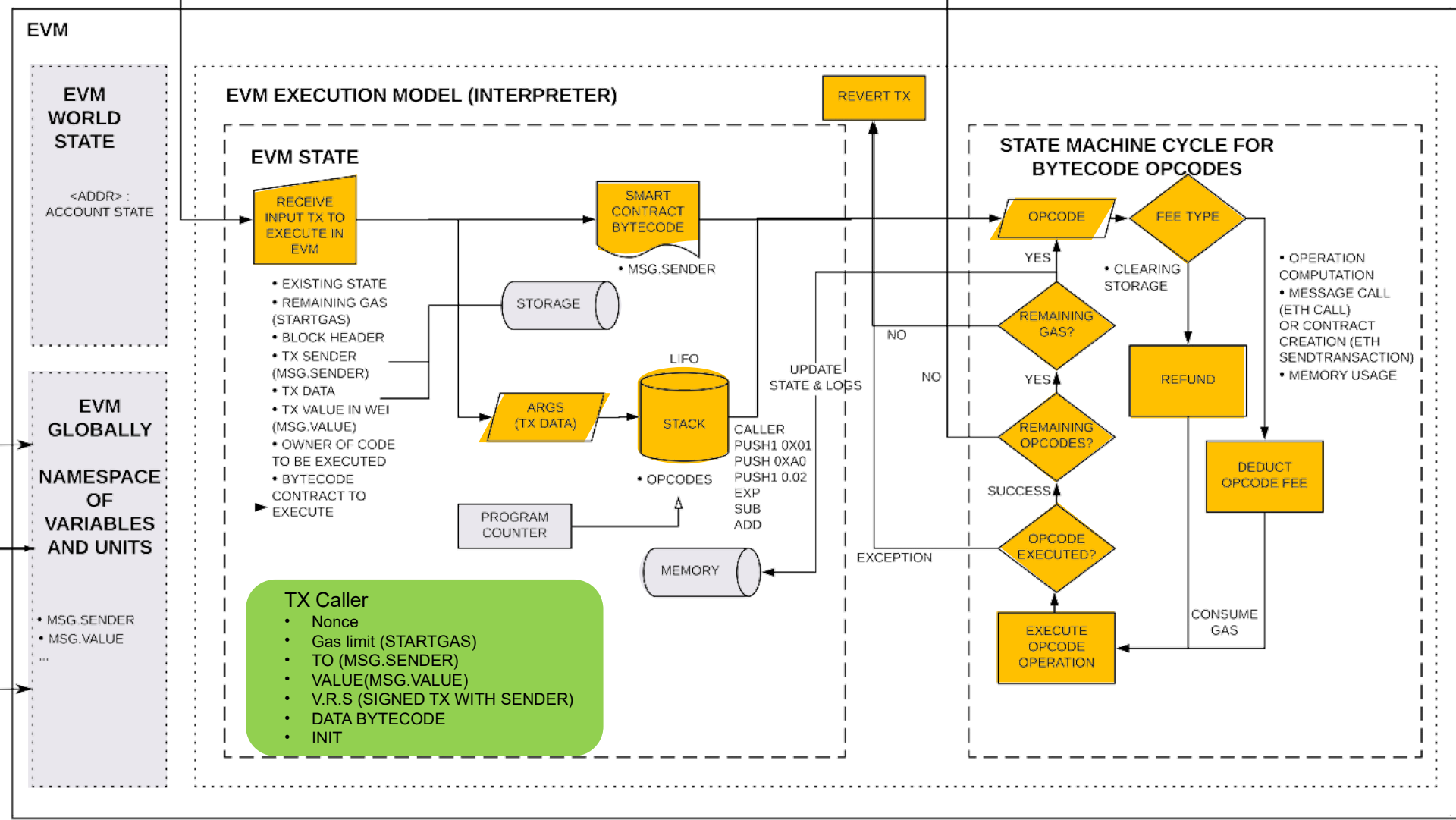
Switch from Geth to a minority client!

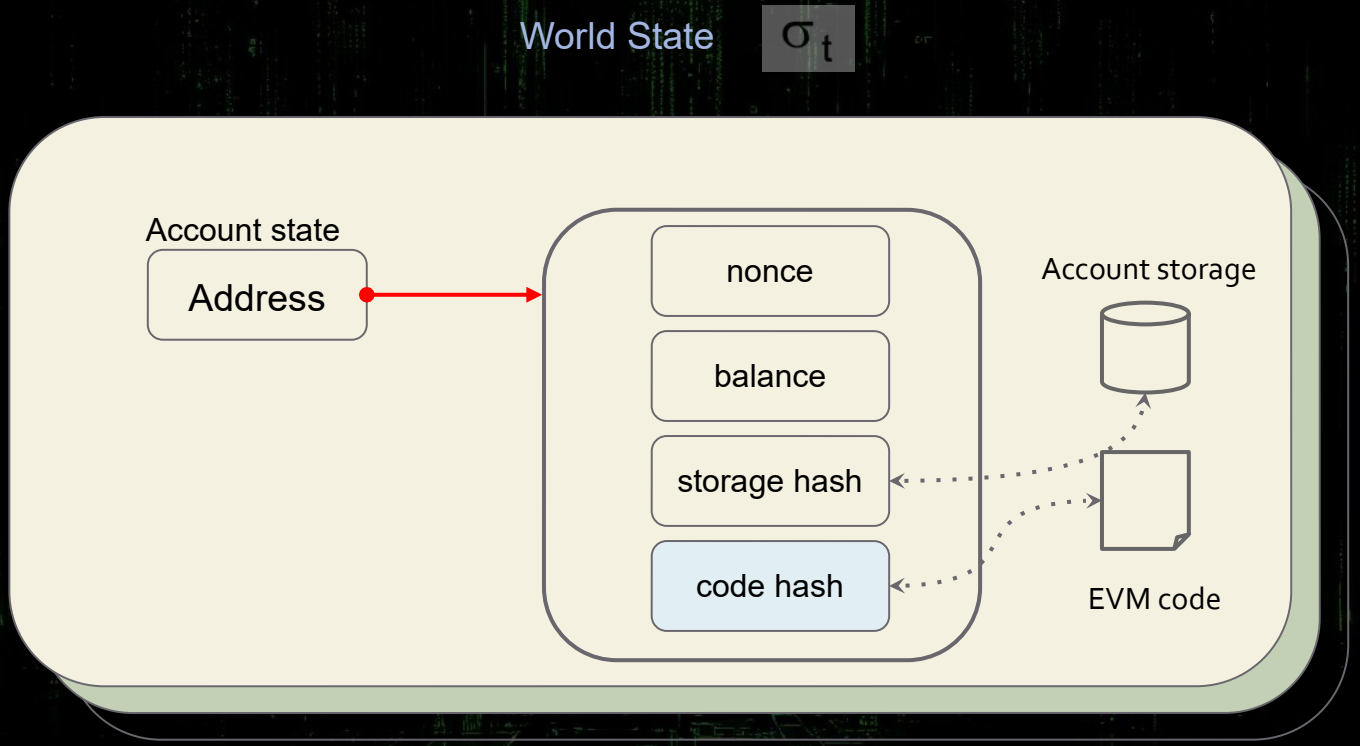


Data provided by supermajority.info — updated manually.
Data may not be 100% accurate. (Read more)

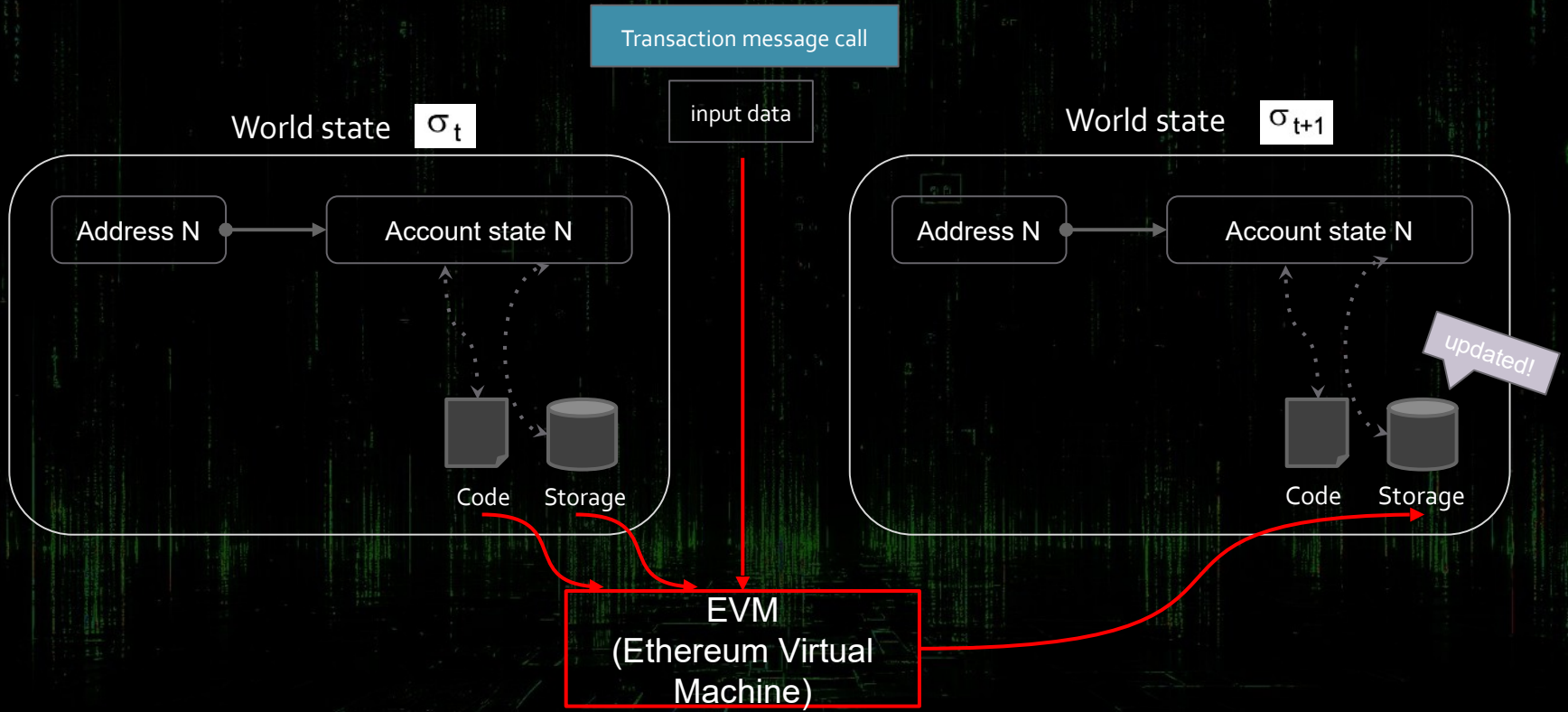
Based on 68.5% self-reported network coverage with the remaining assumed to be mostly Geth.

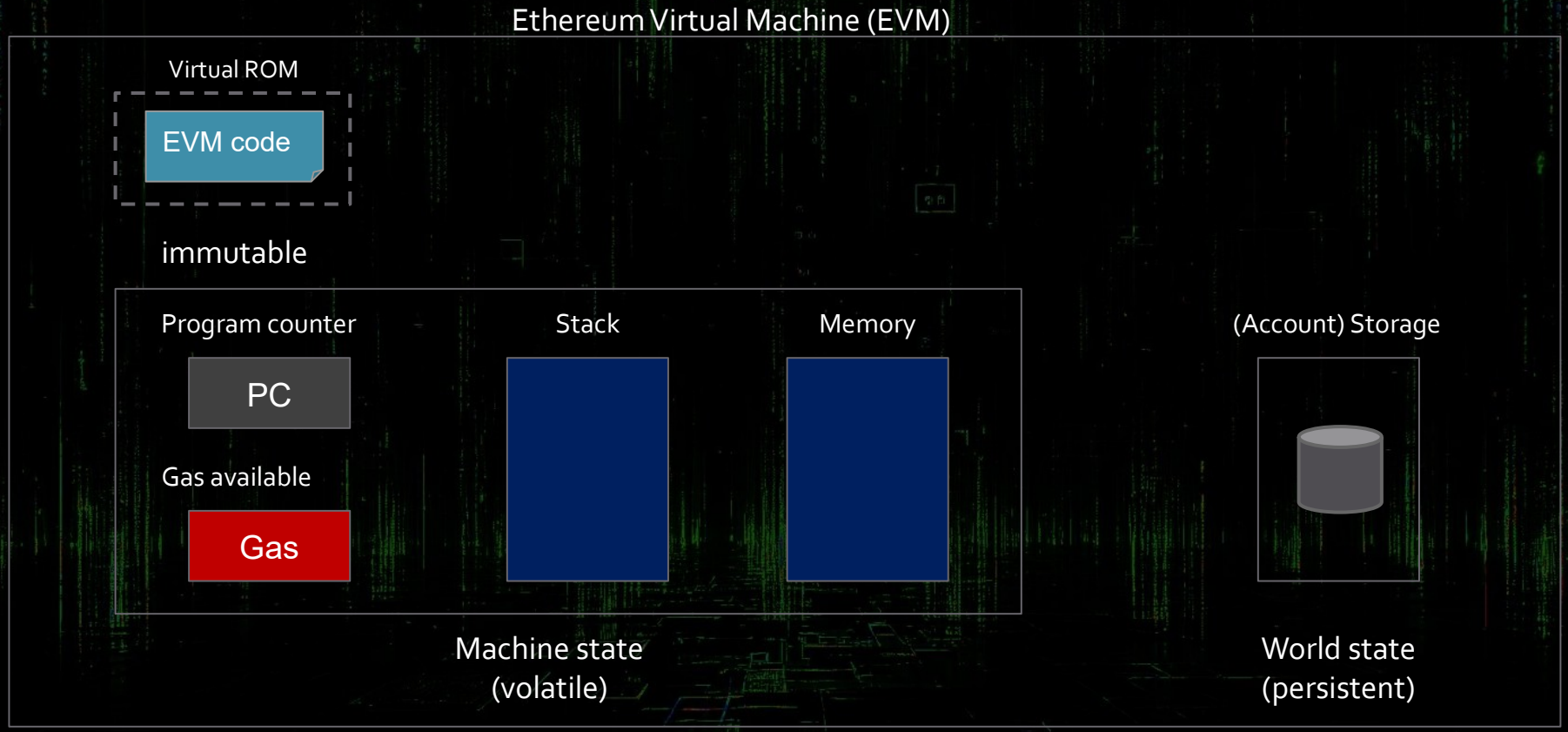
Data source (read more):
Supermajority.info | Ethernodes

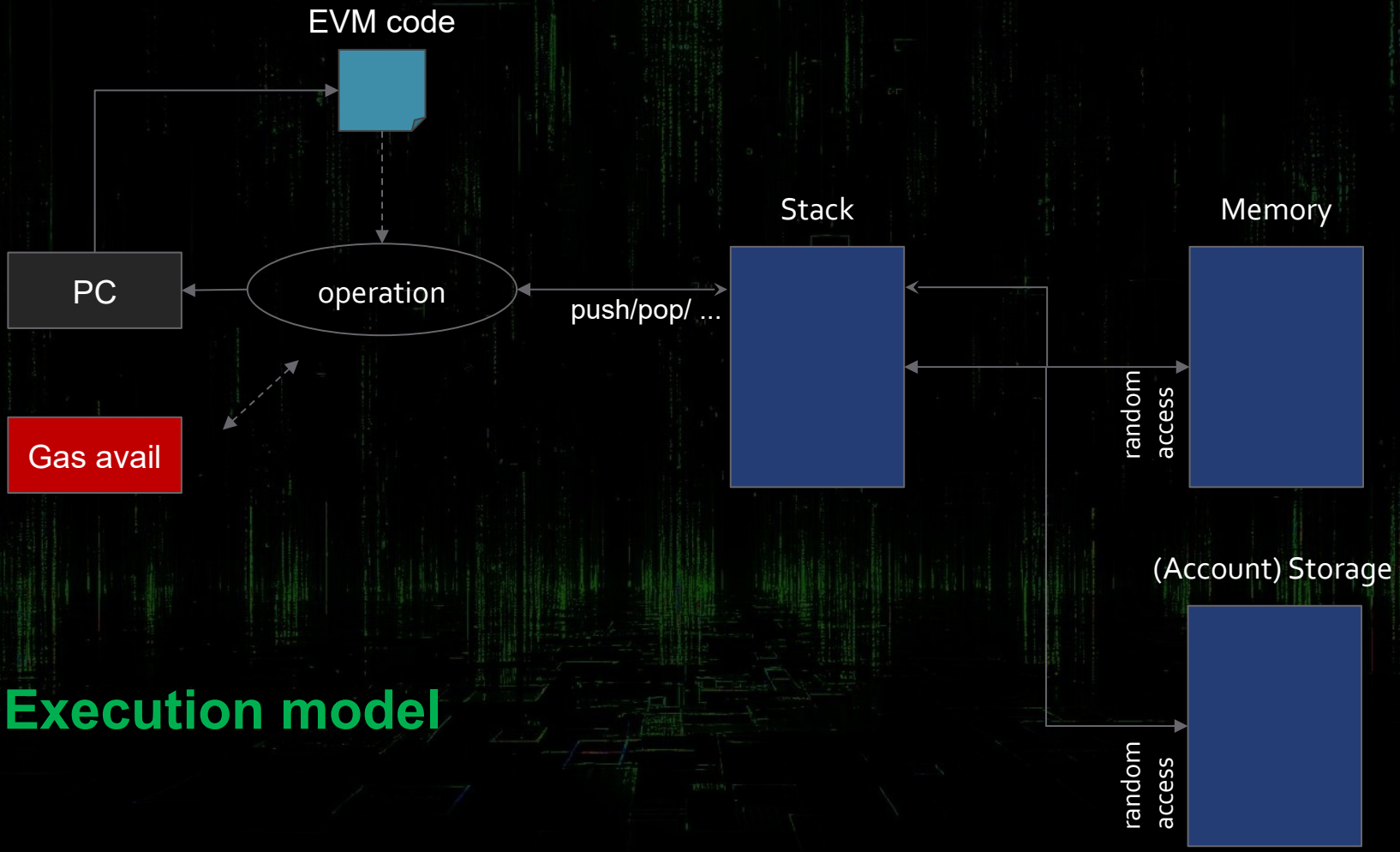

EVM
EVM WORLD STATE
<ADDR> : ACCOUNT STATE
EVM GLOBALLY
NAMESPACE OF VARIABLES AND UNITS
• MSG.SENDER
• MSG.VALUE
...
EVM EXECUTION MODEL (INTERPRETER)
EVM STATE
RECEIVE INPUT TX TO EXECUTE IN EVM
• EXISTING STATE
• REMAINING GAS (STARTGAS)
• BLOCK HEADER
• TX SENDER (MSG.SENDER)
• TX DATA
• TX VALUE IN WEI (MSG.VALUE)
• OWNER OF CODE TO BE EXECUTED
• BYTECODE CONTRACT TO EXECUTE
SMART CONTRACT BYTECODE
• MSG.SENDER
STORAGE
ARGS (TX DATA)
LIFO
STACK
• OPCODES
PROGRAM COUNTER
MEMORY
CALLER
PUSH1 0X01
PUSH 0XA0
PUSH1 0.02
EXP
SUB
ADD
UPDATE STATE & LOGS
REVERT TX
TX Caller
• Nonce
• Gas limit (STARTGAS)
• TO (MSG.SENDER)
• VALUE(MSG.VALUE)
• V.R.S (SIGNED TX WITH SENDER)
• DATA BYTECODE
• INIT
STATE MACHINE CYCLE FOR BYTECODE OPCODES
OPCODE
FEE TYPE
• OPERATION COMPUTATION
• MESSAGE CALL (ETH CALL) OR CONTRACT CREATION (ETH SENDTRANSACTION)
• MEMORY USAGE
YES
• CLEARING STORAGE
REMAINING GAS?
NO
NO
YES
REFUND
REMAINING OPCODES?
DEDUCT OPCODE FEE
SUCCESS
OPCODE EXECUTED?
EXCEPTION
CONSUME GAS
EXECUTE OPCODE OPERATION



An account state could contain EVM code and storage


Transaction 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)


Ethereum Virtual Machine (EVM)
Virtual ROM
EVM code
immutable
Program counter
PC
Gas available
Gas
Stack
Memory
(Account) Storage
Machine state
(volatile)
World state
(persistent)



Execution model

High Level Language

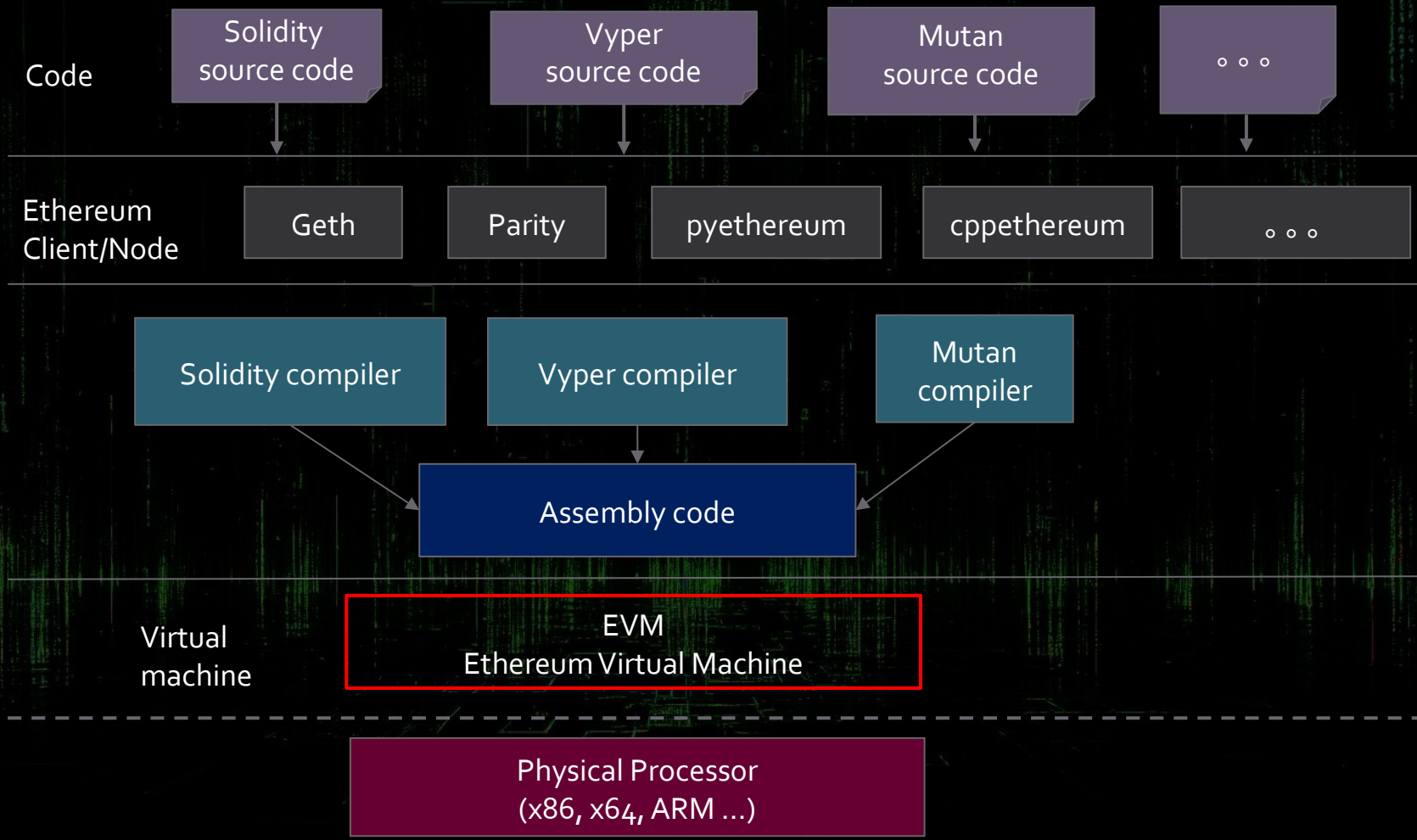
```
pragma solidity ^0.5;

contract Faucet {

    constructor () public payable {}

    // Give out ether to anyone who asks
    function withdraw(uint withdraw_amount) public {

        // Limit withdrawal amount
        require(withdraw_amount <= 10000000000000000);

        // Send the amount to the address that requested it
        msg.sender.transfer(withdraw_amount);
    }

    // Accept any incoming amount
    function () external payable {}

}
```

ByteCode

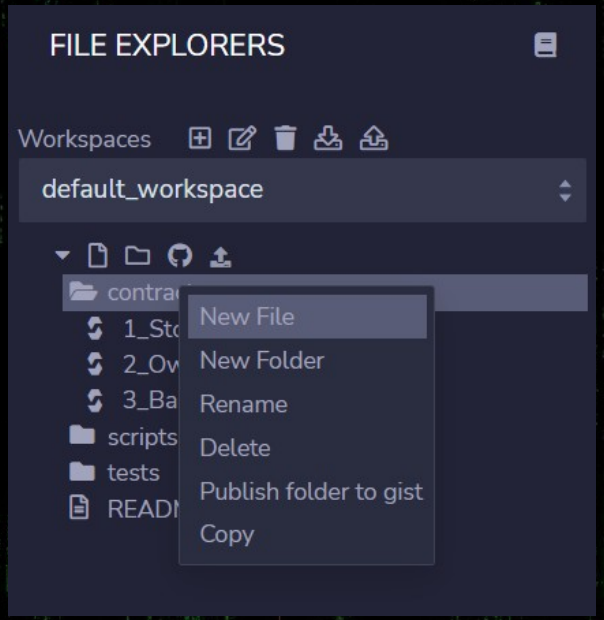
```
PUSH1 0x80              GAS: 3
PUSH1 0x40              GAS: 3
MSTORE                  GAS: 3
CALLVALUE               GAS: 2
...
JUMPI                   GAS: 10
POP                     GAS: 2
...
SLOAD                   GAS: 200
...
```

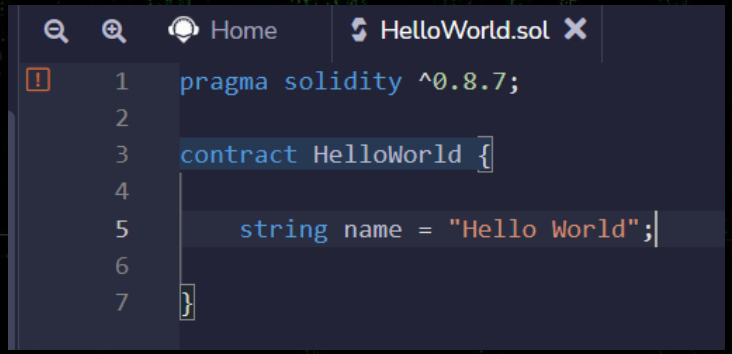
Opcode

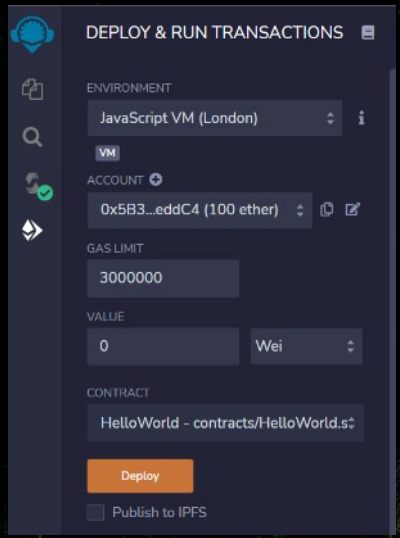
```
608060405260e080610012600039
6000f3fe6080604052600436106 0
1c5760003560e01c80632e1a7d4d
14601e575b005b34801560295760
0080fd5b506053600480360360 20
811015603e57600080fd5b810190
8080359060200190929190505050
6055565b005b67016345785d8a00
...
```

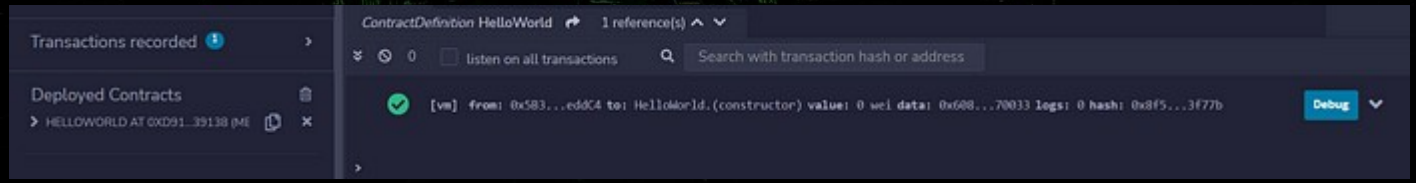

Code
Solidity source code
Vyper source code
Mutan source code
. . .
Ethereum Client/Node
Geth
Parity
pyethereum
cppethereum
. . .
Solidity compiler
Vyper compiler
Mutan compiler
Assembly code
Virtual machine
EVM
Ethereum Virtual Machine
Physical Processor
(x86, x64, ARM ...)

Getting Started

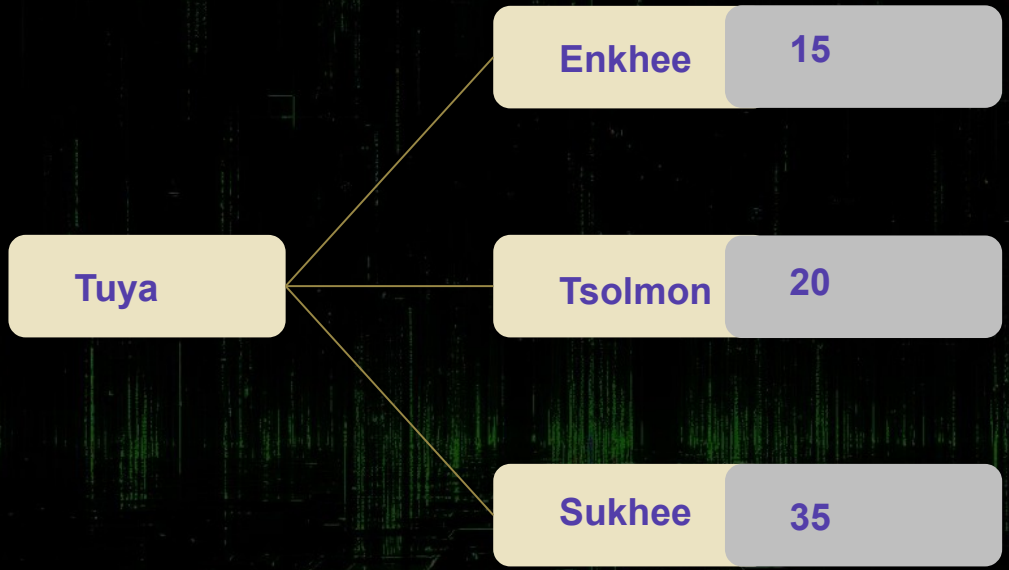








mapping(address => mapping(address => uint))



```
mapping(address => mapping(address => uint)) public allowance
```

```
function approve(address _spender, uint256 _value) public returns (bool success)
{ allowance[msg.sender][_spender] = _value;
emit Approval(msg.sender, _spender, _value);
return true;
}
```

```
function transferFrom(address _from, address _to, uint256 _value) public returns (bool success) {

require(allowance[_from][msg.sender] >= _value);
balanceOf[_from] -= _value;
balanceOf[_to] += _value;
emit Transfer(_from, _to, _value);
allowance[_from][msg.sender] -= value;
return true;
}
```

pwn@kali:~ #

File Actions Edit View Help

pwn@kali:~ # Turing Completeness

pwn@kali:~ # EVM system vulnerabilities

pwn@kali:~ # Transaction protocol

pwn@kali:~ # Solidity and programming language

pwn@kali:~ # Client implementation

pwn@kali:~ # Consensus

Unexpected Ether

Selfdestruct()

```
uint public payoutMileStone = 3 ether;
mapping(address => uint) redeemableEther;
function play() public payable {
    require(msg.value == 0.5 ether); // each play is 0.5 ether
    uint currentBalance = this.balance + msg.value;
    require(currentBalance <= finalMileStone);
    if (currentBalance == payoutMileStone) {
        redeemableEther[msg.sender] += mileStone1Reward;
    }
return;
}
```

```
function claimReward() public {
    // ensure the game is complete
    require(this.balance == finalMileStone);
    // ensure there is a reward to give
    require(redeemableEther[msg.sender] > 0);
    redeemableEther[msg.sender] = 0;
    msg.sender.transfer(transferValue);
}
```

```
function withdrawFunds(uint256 _weiToWithdraw) public {
require(balances[msg.sender] >= _weiToWithdraw);
        require(_weiToWithdraw <= withdrawalLimit);
        require(now >= lastWithdrawTime[msg.sender] + 1 weeks);
        require(msg.sender.call.value(_weiToWithdraw)());
        balances[msg.sender] -= _weiToWithdraw;
        lastWithdrawTime[msg.sender] = now;
    }
```

```
function attackEtherStore() public payable {
        require(msg.value >= 1 ether);
        etherStore.depositFunds.value(1 ether)();
        etherStore.withdrawFunds(1 ether);
    }

function () payable {
        if (etherStore.balance > 1 ether) {
            etherStore.withdrawFunds(1 ether);
        }
    }
```

Reentrancy

function withdrawFunds(uint256 _weiToWithdraw) public nonReentrant

```
Modifire nonReentrant () {
        require(!locked)
        locked = true;
        _;
        locked = false;
}
```

Not easy to solve

Nested Calls * External Contract *Dynamic Calls*Time-Dependent Conditions

```
contract TimeLock {

    mapping(address => uint) public balances;
    mapping(address => uint) public lockTime;

    function deposit() public payable {
        balances[msg.sender] += msg.value;
        lockTime[msg.sender] = now + 1 weeks;
    }

    function increaseLockTime(uint _secondsToIncrease) public {
        lockTime[msg.sender] += _secondsToIncrease;
    }

    function withdraw() public {
        require(balances[msg.sender] > 0);
        require(now > lockTime[msg.sender]);
        balances[msg.sender] = 0;
        msg.sender.transfer(balance);
    }
}
```

Arithmetic Over Underflows

Transaction Ordering Dependence

```
contract Auction {
    address public highestBidder;
    uint256 public highestBid;

    function bid() public payable {
        require(msg.value > highestBid, "Bid not high enough");
        if (highestBid != 0) {
            payable(highestBidder).transfer(highestBid);
        }
        highestBidder = msg.sender;
        highestBid = msg.value;
    }
}
```

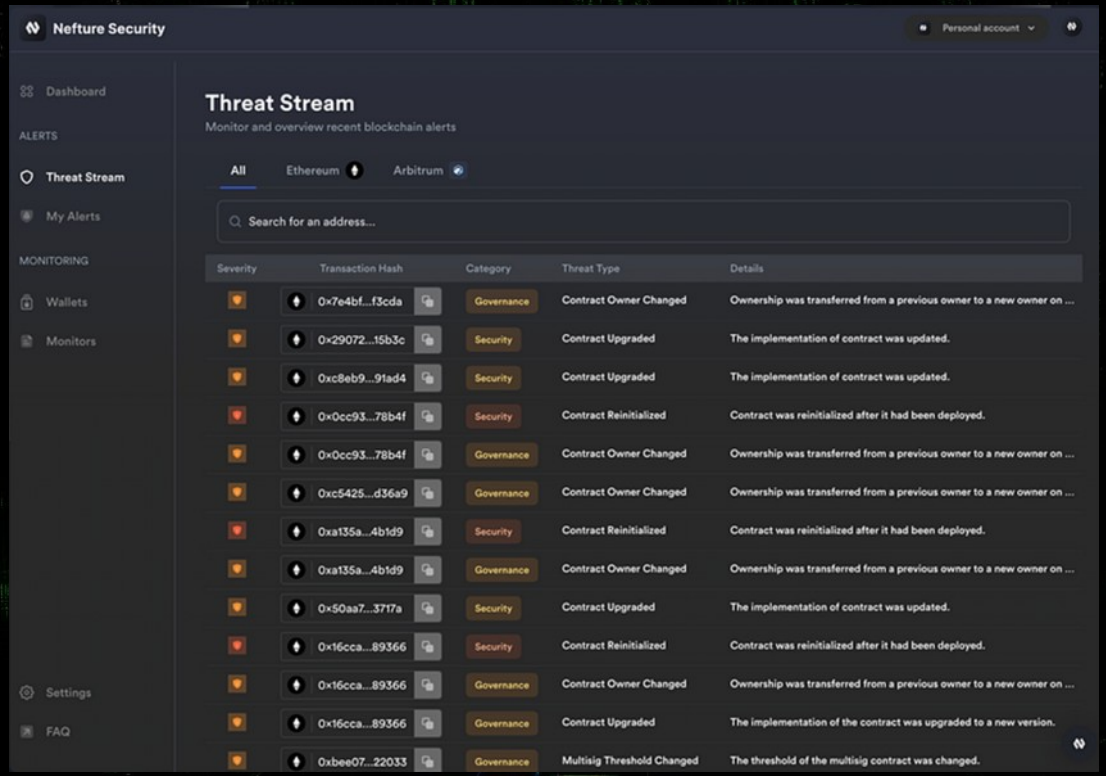
```
web3.eth.subscribe('pendingTransactions', (error, result) =>
 { if (!error) { console.log(result); } });
```

https://etherscan.io/txsPending

Real-World Example: Parity Multisig Wallet $31M

```
function initMultiowned(address[] _owners, uint _required) {
    m_numOwners = _owners.length + 1;
    m_owners[1] = uint(msg.sender);
    m_ownerIndex[uint(msg.sender)] = 1;
    for (uint i = 0; i < _owners.length; ++i)
    {
      m_owners[2 + i] = uint(_owners[i]);
      m_ownerIndex[uint(_owners[i])] = 2 + i;
    }
    m_required = _required;
  }
```

This contract provides a foundation for a secure and distributed way to manage funds or execute protected transactions where multiple parties must agree. (GPT4o)


Nefture Security
Personal account
Dashboard
ALERTS
Threat Stream
My Alerts
MONITORING
Wallets
Monitors
Settings
FAQ
Threat Stream
Monitor and overview recent blockchain alerts
All
Ethereum
Arbitrum
Search for an address...
Severity
Transaction Hash
Category
Threat Type
Details
0x7e4bf...f3cda
Governance
Contract Owner Changed
Ownership was transferred from a previous owner to a new owner on ...
0x29072...15b3c
Security
Contract Upgraded
The implementation of contract was updated.
0xc8eb9...91ad4
Security
Contract Upgraded
The implementation of contract was updated.
0x0cc93...78b4f
Security
Contract Reinitialized
Contract was reinitialized after it had been deployed.
0x0cc93...78b4f
Governance
Contract Owner Changed
Ownership was transferred from a previous owner to a new owner on ...
0xc5425...d36a9
Governance
Contract Owner Changed
Ownership was transferred from a previous owner to a new owner on ...
0xa135a...4b1d9
Security
Contract Reinitialized
Contract was reinitialized after it had been deployed.
0xa135a...4b1d9
Governance
Contract Owner Changed
Ownership was transferred from a previous owner to a new owner on ...
0x50aa7...3717a
Security
Contract Upgraded
The implementation of contract was updated.
0x16cca...89366
Security
Contract Reinitialized
Contract was reinitialized after it had been deployed.
0x16cca...89366
Governance
Contract Owner Changed
Ownership was transferred from a previous owner to a new owner on ...
0x16cca...89366
Governance
Contract Upgraded
The implementation of the contract was upgraded to a new version.
0xbee07...22033
Governance
Multisig Threshold Changed
The threshold of the multisig contract was changed.

SolidityScan | Powered by CRED SHIELDS

Pricing QuickScan Audit Reports Detectors Web3 HackHub Resources Contact Us Go to Dashboard

Smart Contracts, Smarter Security with AI.

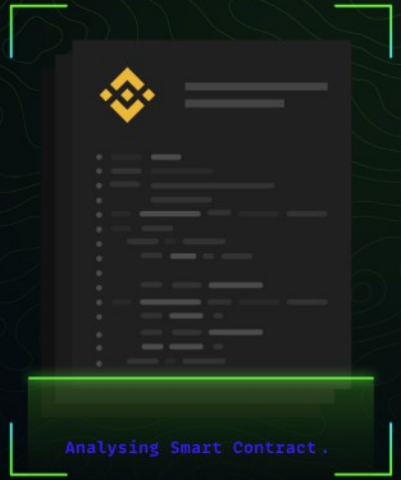
Fix|

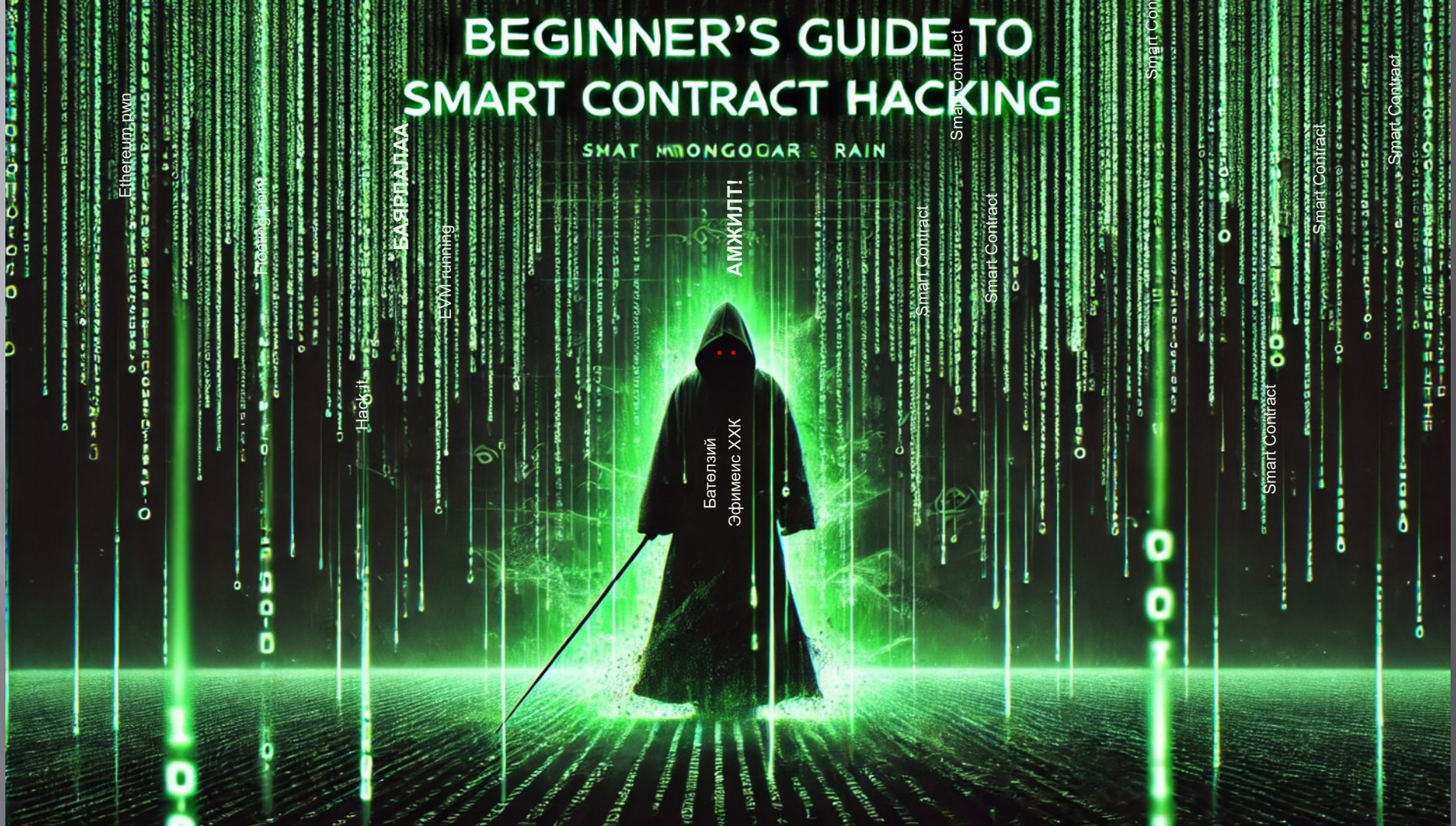
An advanced smart contract scanning tool designed to uncover vulnerabilities and proactively address risks within your code.

Signup For Free Trial

Run A QuickScan

#1 PRODUCT OF THE MONTH
Developer Tools




BEGINNER'S GUIDE TO
SMART CONTRACT HACKING
Бателзий
Эфимеис XXK
АМЖИЛТ!
БАЯРЛАЛАА
Hacker
EVM running
Floating point
Ethereum own
Smart Contract