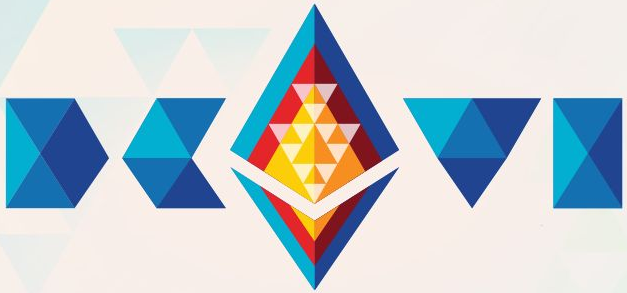


zkEVM Vs EVM

Full Equivalence?

Ignasi Ramos

Polygon zkEVM Team

What is this talk about?

- What is a ZK-EVM?
- Polygon's zkEVM
- Differences
  - Storage - SMT
  - Memory
  - Zk-counters
  - Selfdestruct
  - Precompileds
- Other differences

Whats is a zk-EVM?

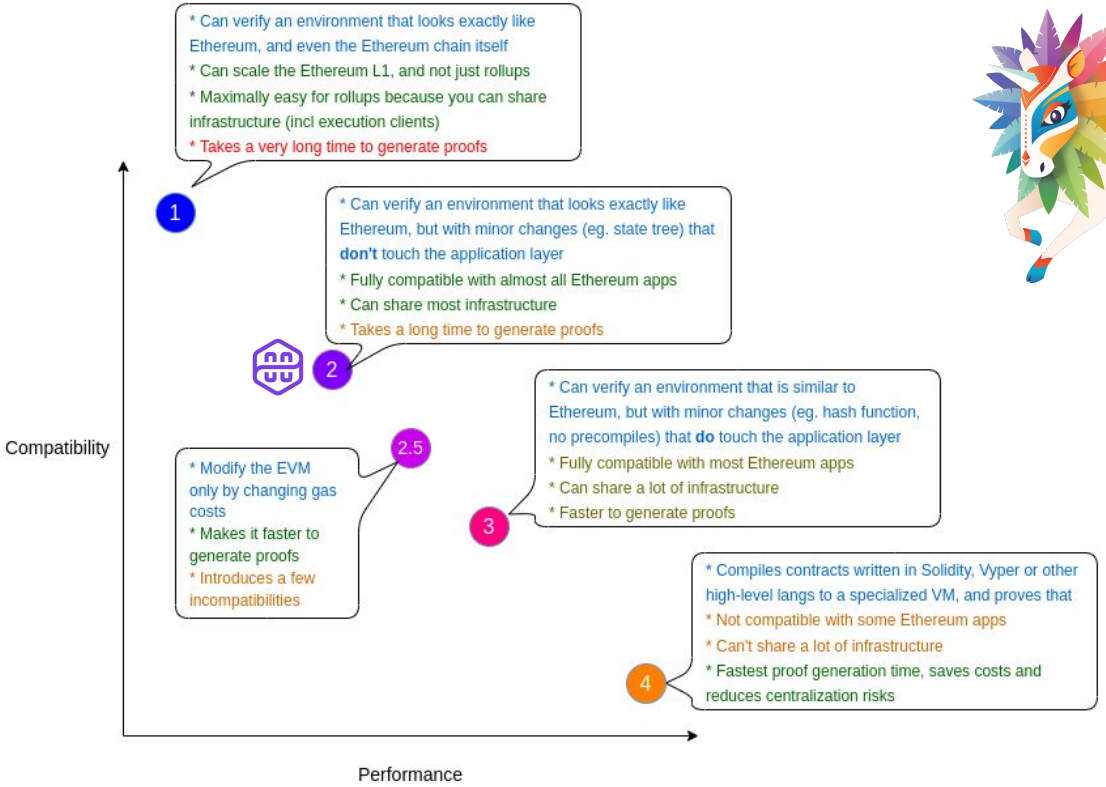
A ZK-EVM

- Definition: a virtual machine that executes smart contracts in a way that is compatible with **zero-knowledge-proof** computation

- Purpose: Scale Ethereum

- How? use **ZK-SNARK** technology to make cryptographic proofs of execution of Ethereum-like transactions.

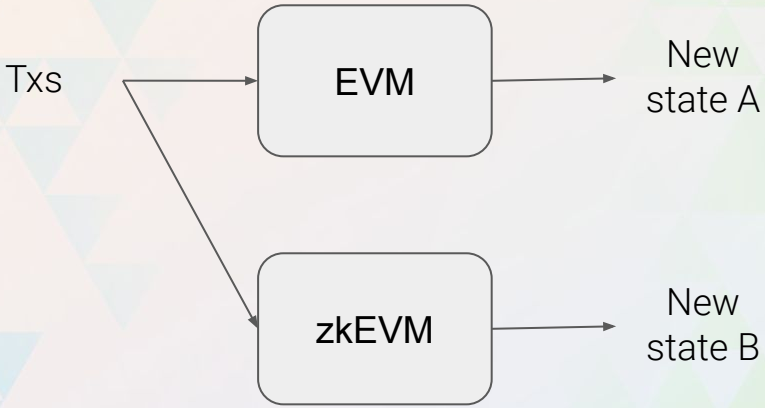
The ZK-EVM “dilemma”



Source: https://vitalik.eth.limo/general/2022/08/04/zkevm.html

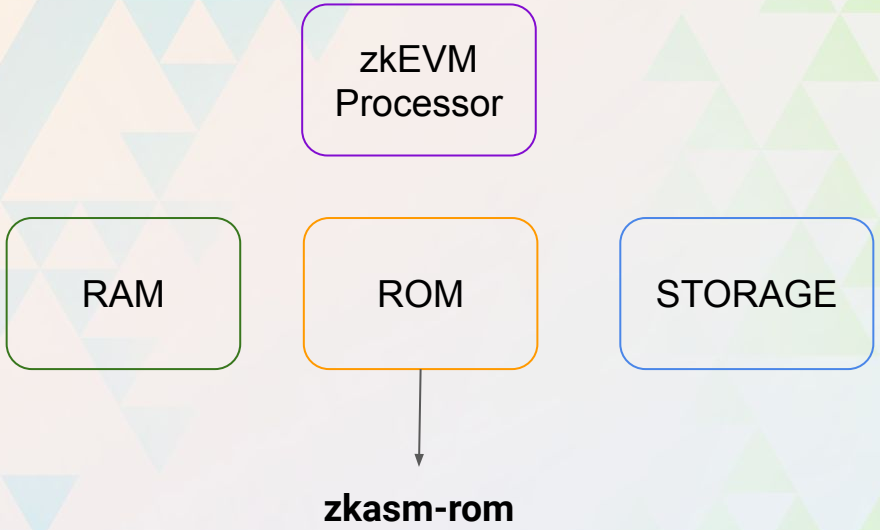

polygon zkEVM

Fully EVM-equivalent



New state A = New state B

Polygon zkEVM

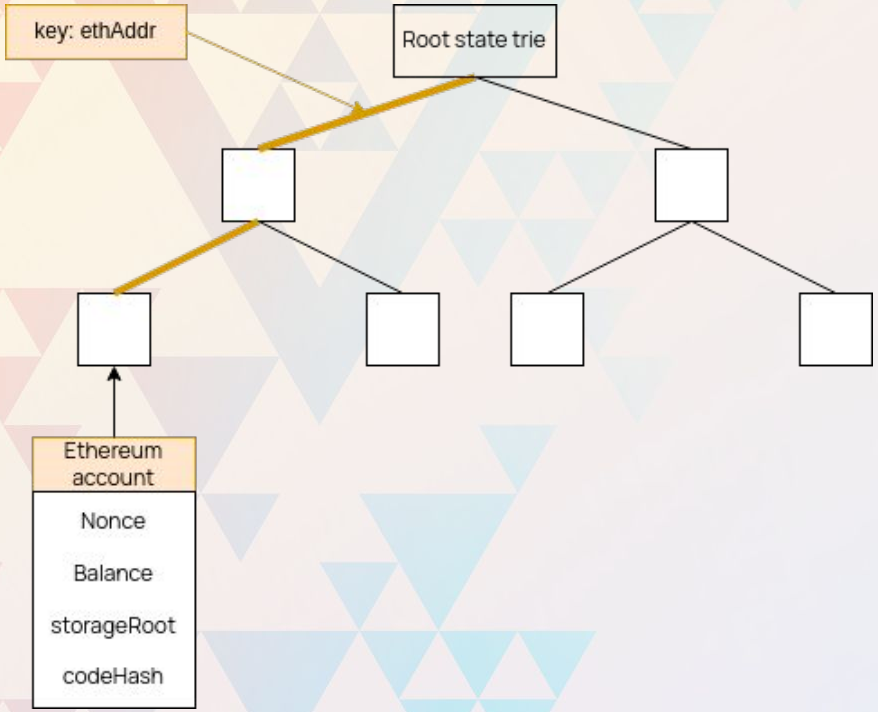


- **zkasm**: defines the steps
- **PIL**: verifies the correctness of the steps

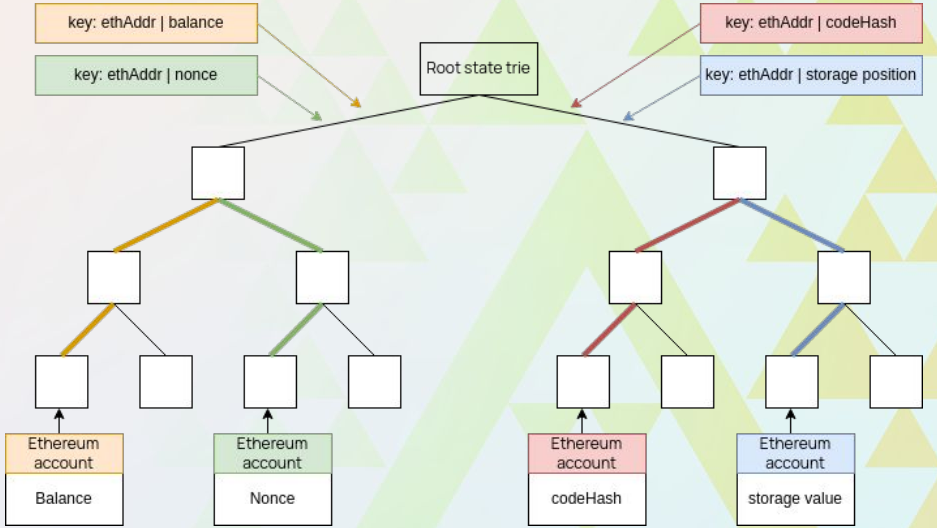

zkEVM vs EVM

Storage

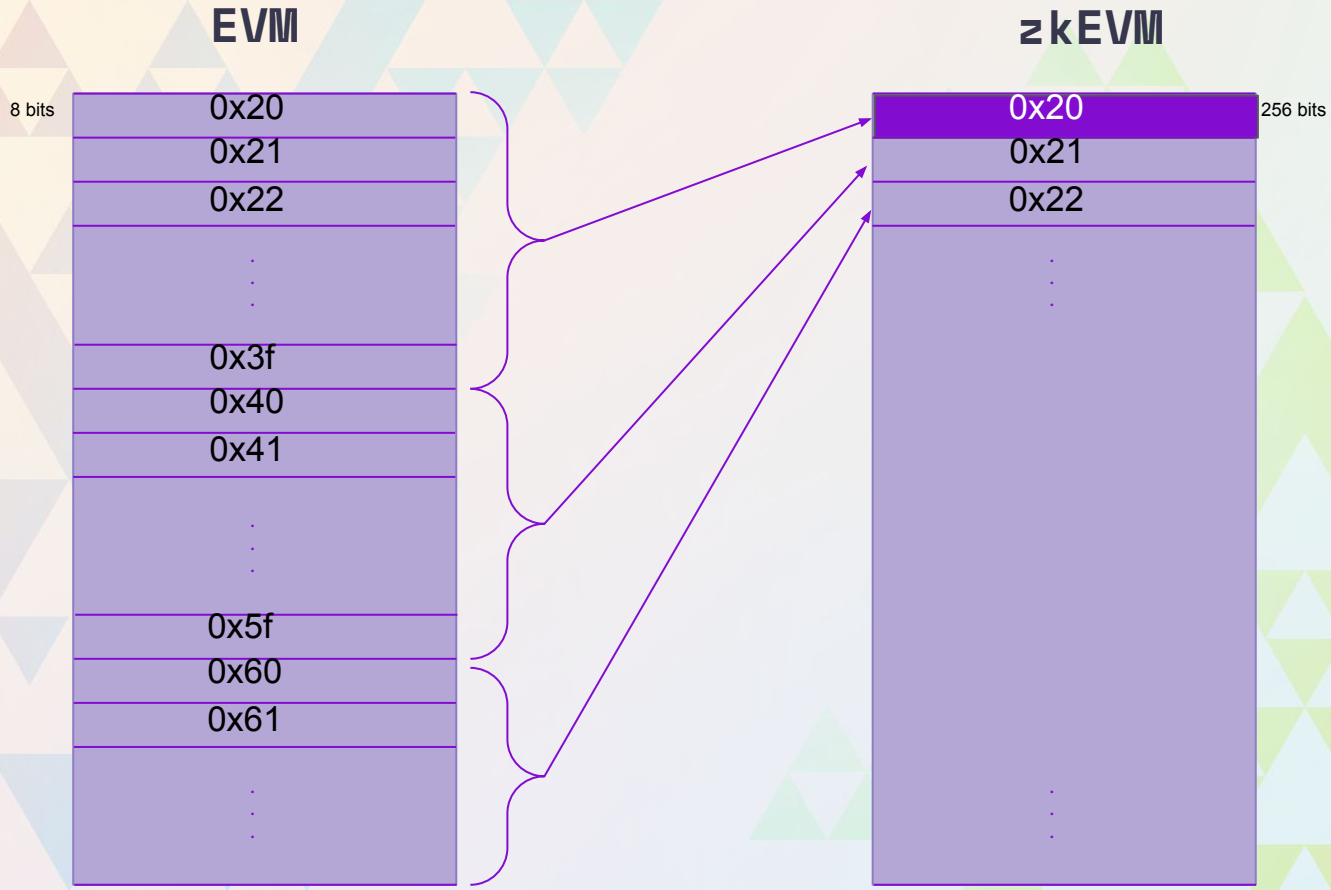
EVM



zkEVM



Memory


EVM
zkEVM
8 bits
256 bits
0x20
0x21
0x22
0x3f
0x40
0x41
0x5f
0x60
0x61
0x20
0x21
0x22

Example MLOAD

- The easiest case: MLOAD 32 bytes && offset%32 == 0

  offset = 0x20

  memory[0x20] = [0x20,0x21,0x22,0x23, … ,0x3e,0x3f]

- MLOAD 32 bytes && offset%32 != 0

  offset = 0x23 → memory[0x23, 0x43]

  memory[0x20] = [0x20,0x21,0x22,0x23, … ,0x3e,0x3f]

  memory[0x21] = [0x40,0x41,0x42,0x43, … ,0x5e,0x5f]

Example MLOAD

- MLOAD X bytes (X < 32 bytes) && offset%32 == 0

```
offset = 0x20, length = 4
memory[0x20] = [0x20,0x21,0x22,0x23, … ,0x3e,0x3f]
```

- MLOAD X bytes && offset%32 != 0 && offset%32+length < 32

```
offset = 0x23, length = 4 → [0x23,0x26]
memory[0x20] = [0x20, … ,0x23,0x24,0x25,0x26, … ,0x3f]
```

- MLOAD X bytes && offset%32 != 0 && offset%32+length > 32

```
offset = 0x3e, length = 4 → [0x3e,0x41]
memory[0x20] = [0x20,0x21,0x22,0x23, … ,0x3e,0x3f]
memory[0x21] = [0x40,0x41,0x42,0x43, … ,0x4e,0x4f]
```

zk-counters

- Counters are a way to control that the total number of steps do not exceed the maximum polynomial size.
- If we go out of counters, then, there is an error in the processing of the batch (it is not an error of the user).

Binary	Arithmetic	Memory Align
Keccak	Padding	Poseidon
Steps		

zk-counters example

```
opEQ:

    %MAX_CNT_BINARY - CNT_BINARY - 1 :JMPN(outOfCountersBinary)
    %MAX_CNT_STEPS - STEP - 120 :JMPN(outOfCountersStep)

    SP - 2           :JMPN(stackUnderflow)
    SP - 1 => SP
    $ => A           :MLOAD(SP--)
    $ => B           :MLOAD(SP)
    GAS-3 => GAS     :JMPN(outOfGas)
    $                :EQ,MSTORE(SP++)
    1024 - SP        :JMPN(stackOverflow)
                     :JMP(readCode)
      laisolizq, 8 months ago • First opcodes
```

SELFDESTRUCT

SENDALL

- **EIP-4758**: Deactivate SELFDESTRUCT

- This EIP renames the SELFDESTRUCT opcode to SENDALL, and replaces its functionality. The new functionality is there to only send all Ether in the account to the caller.

Precompileds

Address	Name	Supported
0x01	ecRecover	Yes
0x02	SHA-256	Ongoing
0x03	RIPEMD-160	Ongoing
0x04	identity	Yes
0x05	modexp	Yes
0x06	ecAdd	Ongoing
0x07	ecMul	Ongoing
0x08	ecPairing	Ongoing
0x08	blake2f	Ongoing

Section 4

Other differences

Other differences

- **EXTCODEHASH**: returns hashContract from zkEVM tree, hashed with Poseidon
- **BLOCKHASH**: returns all previous block hashes (not just the last 256 blocks)
- **Memory limits**: 0x20000, which require 8.5 million GAS of memory expansion.

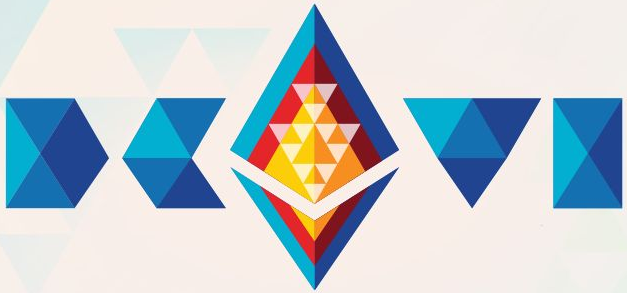
- **Pre EIP-155 and EIP-2718 TXS:** not supported yet, is our current priority

We are passing

97% of the Ethereum Test Suit

Final Remarks

- We are full-EVM equivalent
- Differences are addressed aiming at performance and equivalence
- **We are on public Testnet!**
  public.zkevm-test.net



Thank you!

Ignasi Ramos
Protocol Team, Polygon zkEVM
ignasi@polygon.technology

@0xIgnasi

polygon zkEVM

Appendix

Some assets.