

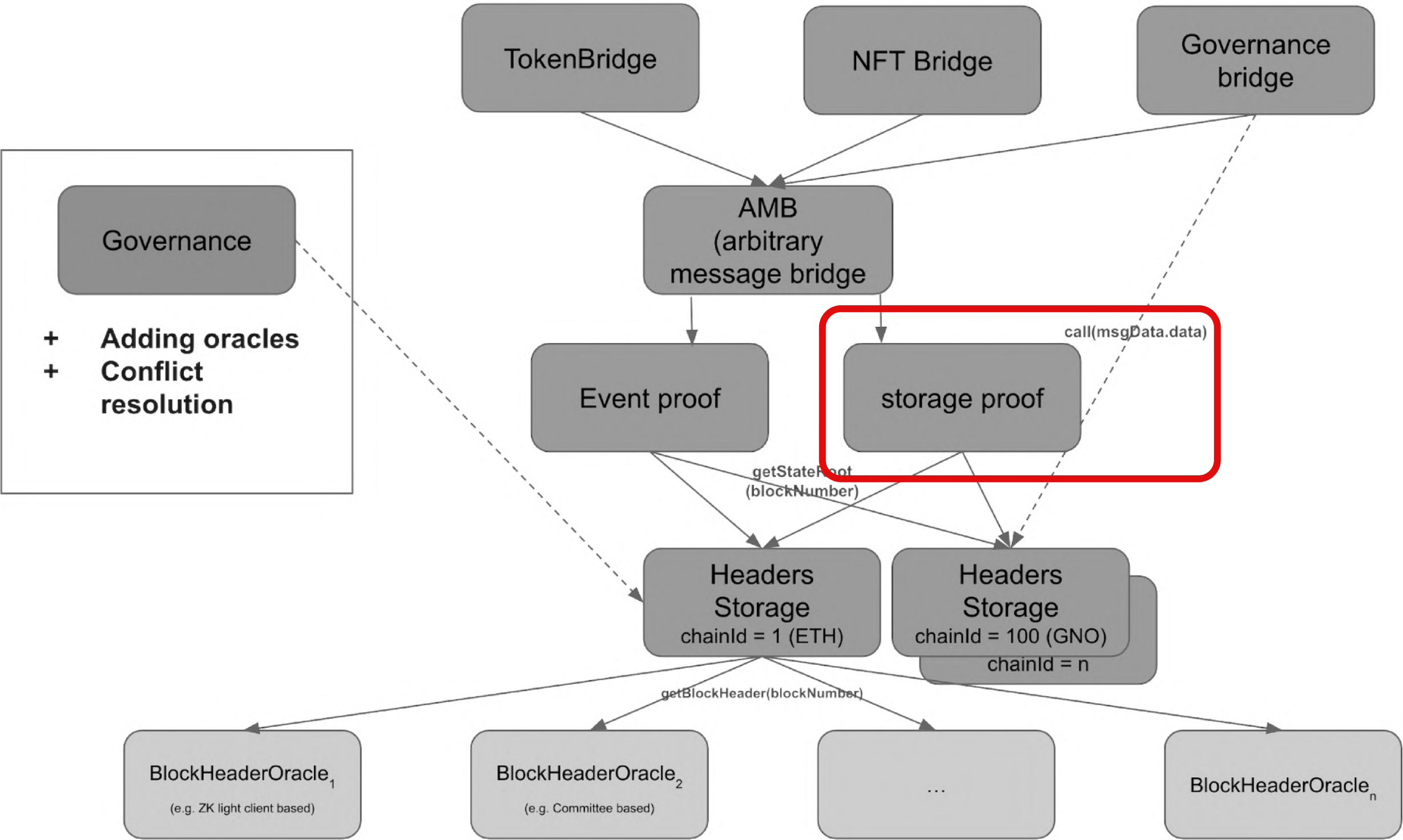
L1StateOracle

Trustlessly Time Travel 🚀 L1 state on L2

Problem

- ZK bridges enable trustless messaging between L1 and L2 chains
- But they don't give L2s access to L1 on-chain data
- e.g. I want to check the ownership of CryptoPunk#420 at block 100000000

Inspiration



<https://ethresear.ch/t/hasi-a-principled-approach-to-bridges/14725>

Technical Introduction

Hashi - An EVM hash oracle aggregator by Gnosis

- Provides additive security to bridges
 - By comparing L1 block header hashes from multiple bridges

Axiom - The ZK Coprocessor for Ethereum

- Allows trustless access to historic on-chain data from your smart contract
 - By creating ZK proofs of any L1 historical state,
 - And verifying proofs on-chain via block header hashes

Solution

Hashi 橋 => Reliable L1 block header on L2

+

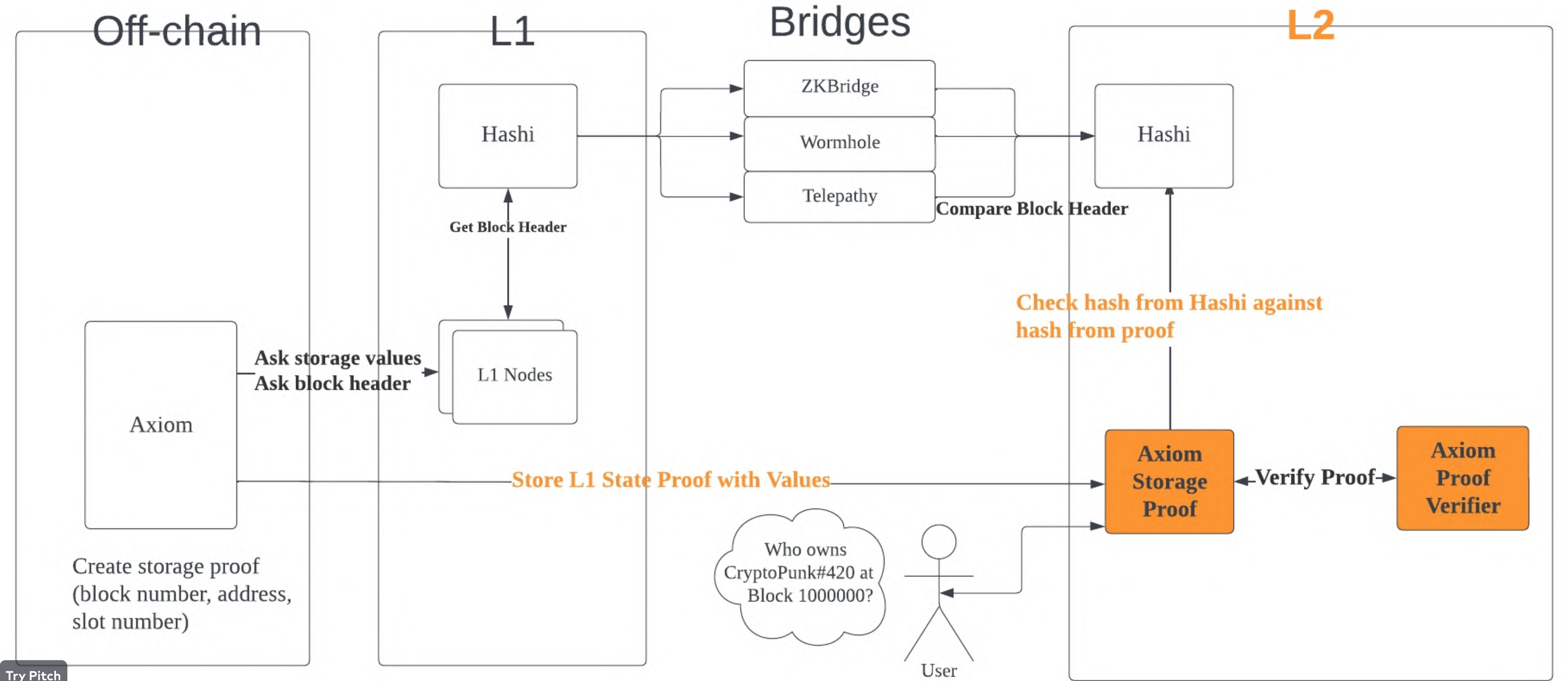
Axiom  => Provable L1 state associated with the block header

=

L1StateOracle => Trustlessly Time Travel  L1 state on L2

Axiom 🛠️ + Hashi 橋 => L1StateOracle (Time Travel 🚀 L1 state on L2)

Existing
New




VERIFY

FOR ANYONE
Account age

FOR PROTOCOLS
Token price (V2)

FOR PROTOCOLS
Token price (V3)

FOR ORACLES
Randomizer


 Custom

Custom

Block number	Address
10000000	0xb47e3cd837dDF8e4c57F05d70Ab865de6e193

Storage slot	How do I find this?
0f92f3ad435570e9f610d535ca71c2a4c5ef34aa438e925fe55dbb614b291b4b	

Value at slot: 0x00000000000000000000000000000000c352b534e8b987e036a93539fd6897f53488e56a

 Add slot

Regenerate proof

```
{
  "block": "10000000",
  "address": "0xb47e3cd837dDF8e4c57F05d70Ab865de6e193BBB",
  "slots": [
    {
      "value":
"0x00000000000000000000000000000000c352b534e8b987e036a93539fd6897f5348
8e56a",
      "slotNumber":
"0f92f3ad435570e9f610d535ca71c2a4c5ef34aa438e925fe55dbb614b291b4b
"
    }
  ],
```


Deeper Dive

Axiom Storage Proof contract

- verifies the proof with an on-chain verifier

```
(bool success,) = verifierAddress.call(proof);
if (!success) {
    revert("Proof verification failed");
}
```

- parses the data from Axiom to get up to 10 proofs of storage slots
- saves attestations based on hash of (blockNumber, account, slot, slotValue)

```
uint16 i = 0; // slot number to find
uint256 slot = (uint256(bytes32(proof[384 + 128 + 128 * i:384 + 160 + 128 * i])) << 128)
| uint128(bytes16(proof[384 + 176 + 128 * i:384 + 192 + 128 * i]));
uint256 slotValue = (uint256(bytes32(proof[384 + 192 + 128 * i:384 + 224 + 128 * i])) << 128)
| uint128(bytes16(proof[384 + 240 + 128 * i:384 + 256 + 128 * i]));
(bytes32 hashedVal) = keccak256(abi.encodePacked(blockData.blockNumber, account, slot, slotValue));
slotAttestations[hashedVal] = true;
// CryptoPunk#420 address at the slot for the given block
return address(uint160(slotValue));
```

Demo

<https://github.com/gnosis/hashi/pull/11>

```
const expectedAddress = await storageProof.callStatic.attestCryptoPunk420AddressWithHashi(
  blockHashWitness,
  proof,
  CHAIN_ID,
  [ambAdapter.address, ambAdapter.address]
)
expect(
  expectedAddress.toLowerCase()
).toEqual(CryptoPunk420ownerAtBlock10Mil.toLowerCase())
```

End-to-end tests

Execution layer

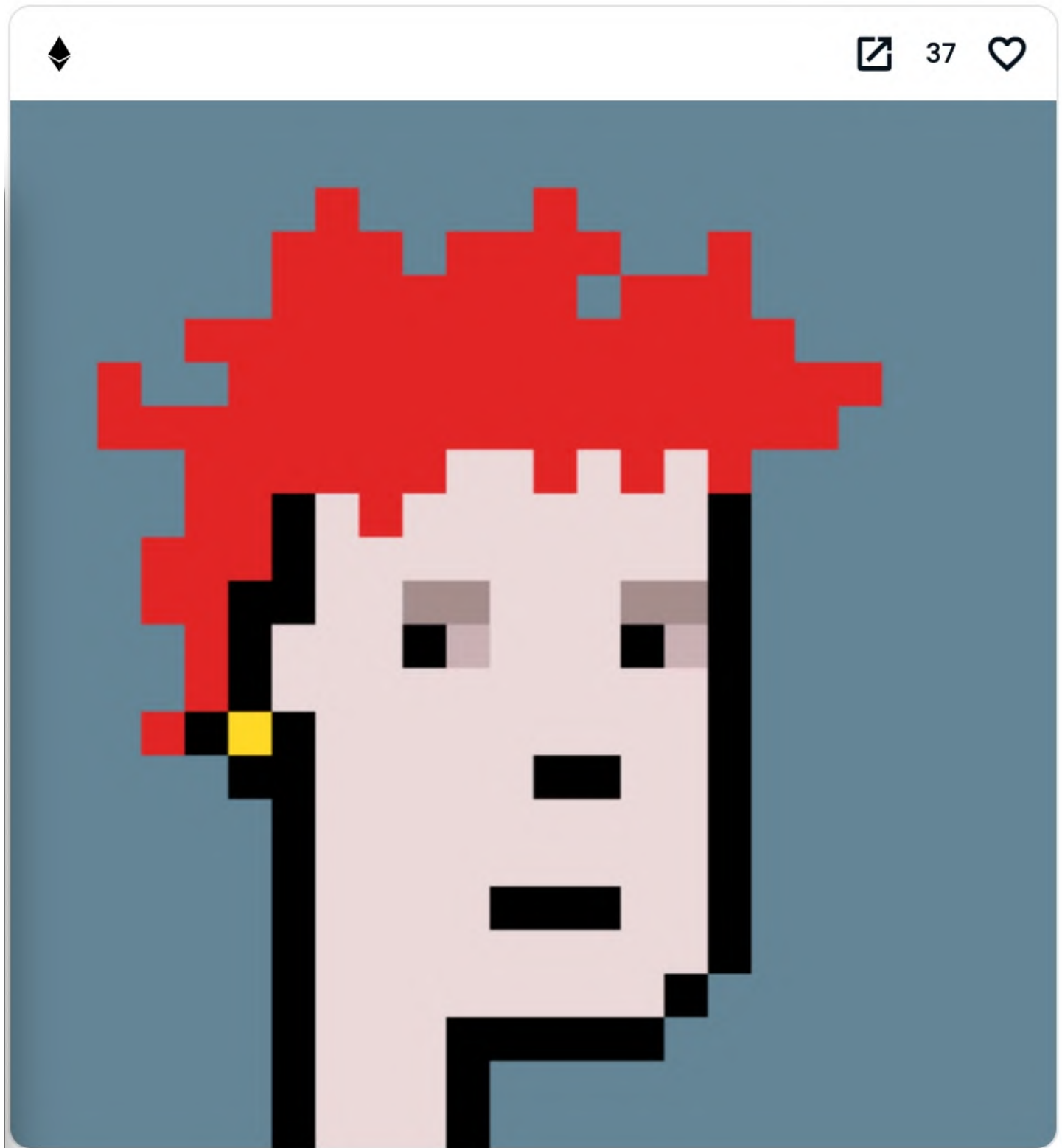
- ✓ Attest slots for the claimed block head with the block hash agreed on by N adapters (2459ms)
- ✓ Reverts if the claimed block header is different from the block hash agreed on by N adapters (143ms)

Execution layer

Expected: 0xc352b534e8b987e036a93539fd6897f53488e56a Got: 0xc352b534e8b987e036a93539fd6897f53488e56a

- ✓ Get the correct cryptopunk#420 owner address with the proof (150ms)

3 passing (3s)



CryptoPunks

CryptoPunk #420

Owned by YugaLabs

1.7K views 37 favorites PFPs

Price History



No events have occurred
Check back later

Transfer

C352B5

LarvaLabs

2y ago

Yes, the owner was 0xc352b534e8b987e036a93539fd6897f53488e56a at block 10,000,000

 **Thank You** 

Hire me!

resume.sjlee.me