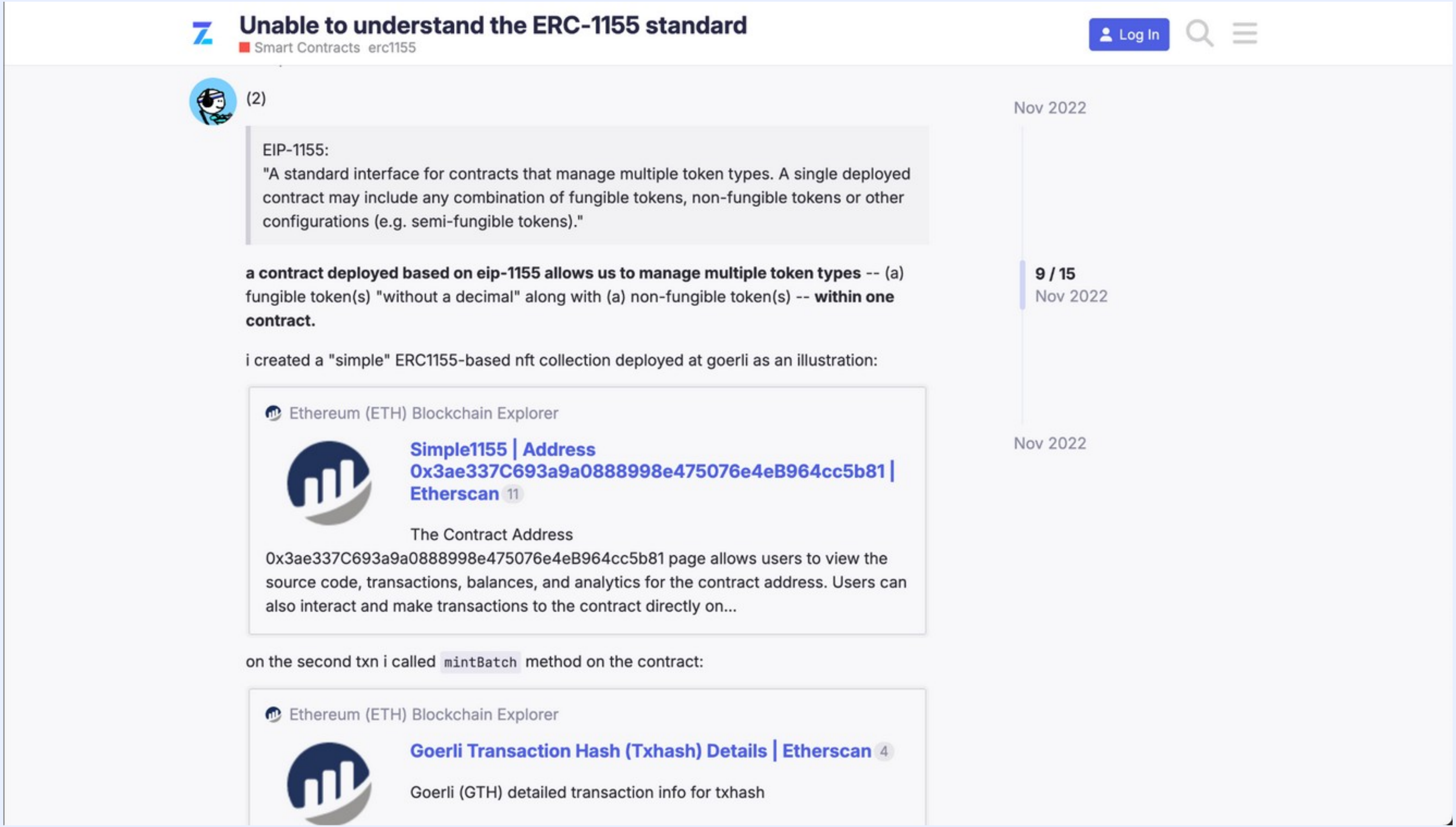
# The State of Source-Code Verification: Closed, Painful, and Ready for Change

## Kaan Uzdoğan

sourcify.dev

ARGOT



# Unable to understand the ERC-1155 standard

Smart Contracts erc1155

(2)

> EIP-1155:
> "A standard interface for contracts that manage multiple token types. A single deployed contract may include any combination of fungible tokens, non-fungible tokens or other configurations (e.g. semi-fungible tokens)."

**a contract deployed based on eip-1155 allows us to manage multiple token types** -- (a) fungible token(s) "without a decimal" along with (a) non-fungible token(s) -- **within one contract.**

i created a "simple" ERC1155-based nft collection deployed at goerli as an illustration:

Ethereum (ETH) Blockchain Explorer

**Simple1155 | Address 0x3ae337C693a9a0888998e475076e4eB964cc5b81 | Etherscan** 11

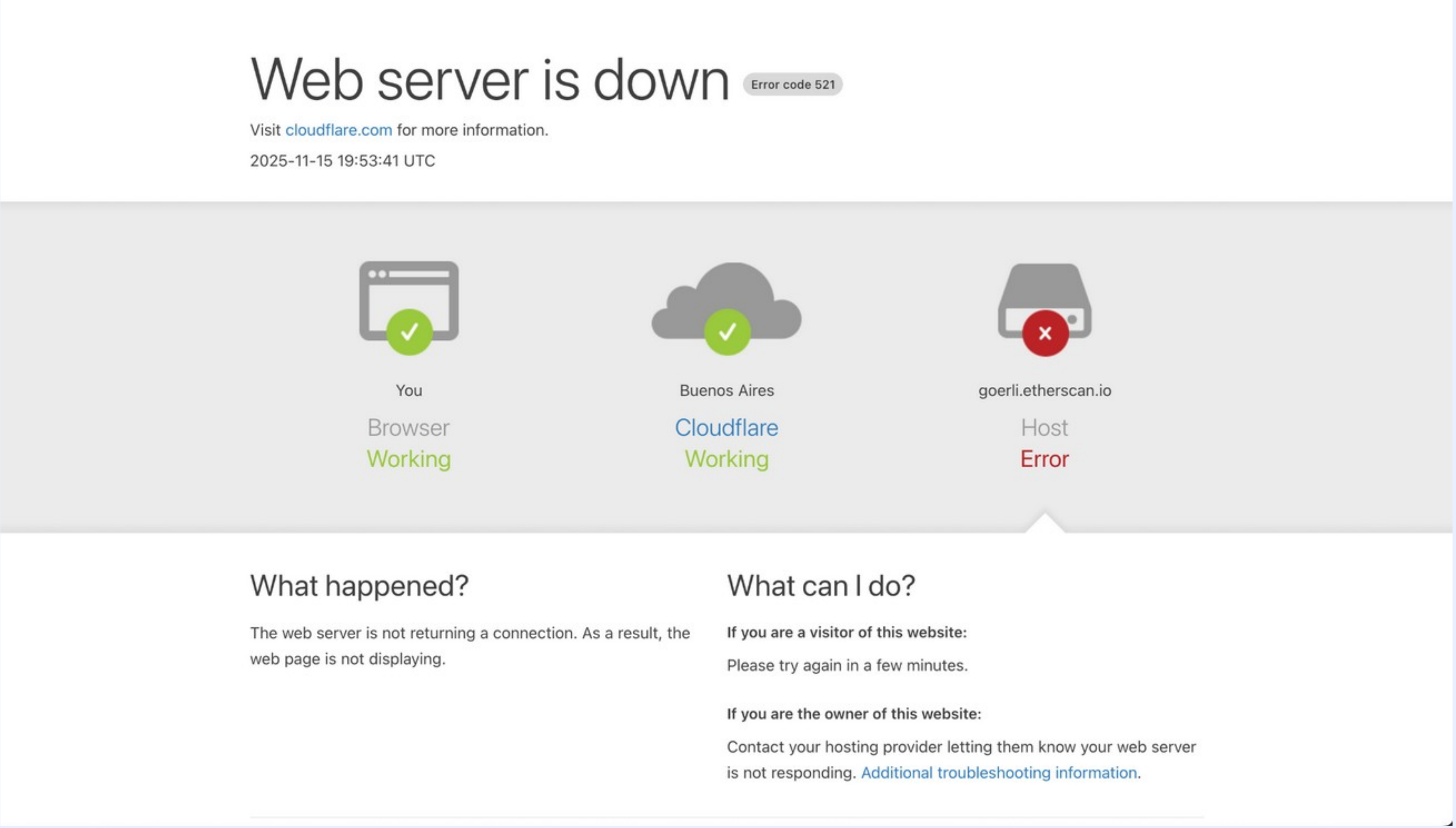
The Contract Address 0x3ae337C693a9a0888998e475076e4eB964cc5b81 page allows users to view the source code, transactions, balances, and analytics for the contract address. Users can also interact and make transactions to the contract directly on...

on the second txn i called `mintBatch` method on the contract:

Ethereum (ETH) Blockchain Explorer

**Goerli Transaction Hash (Txhash) Details | Etherscan** 4

Goerli (GTH) detailed transaction info for txhash



# Web server is down

Error code 521

Visit cloudflare.com for more information.

2025-11-15 19:53:41 UTC

You
Browser
Working

Buenos Aires
Cloudflare
Working

goerli.etherscan.io
Host
Error

## What happened?

The web server is not returning a connection. As a result, the web page is not displaying.

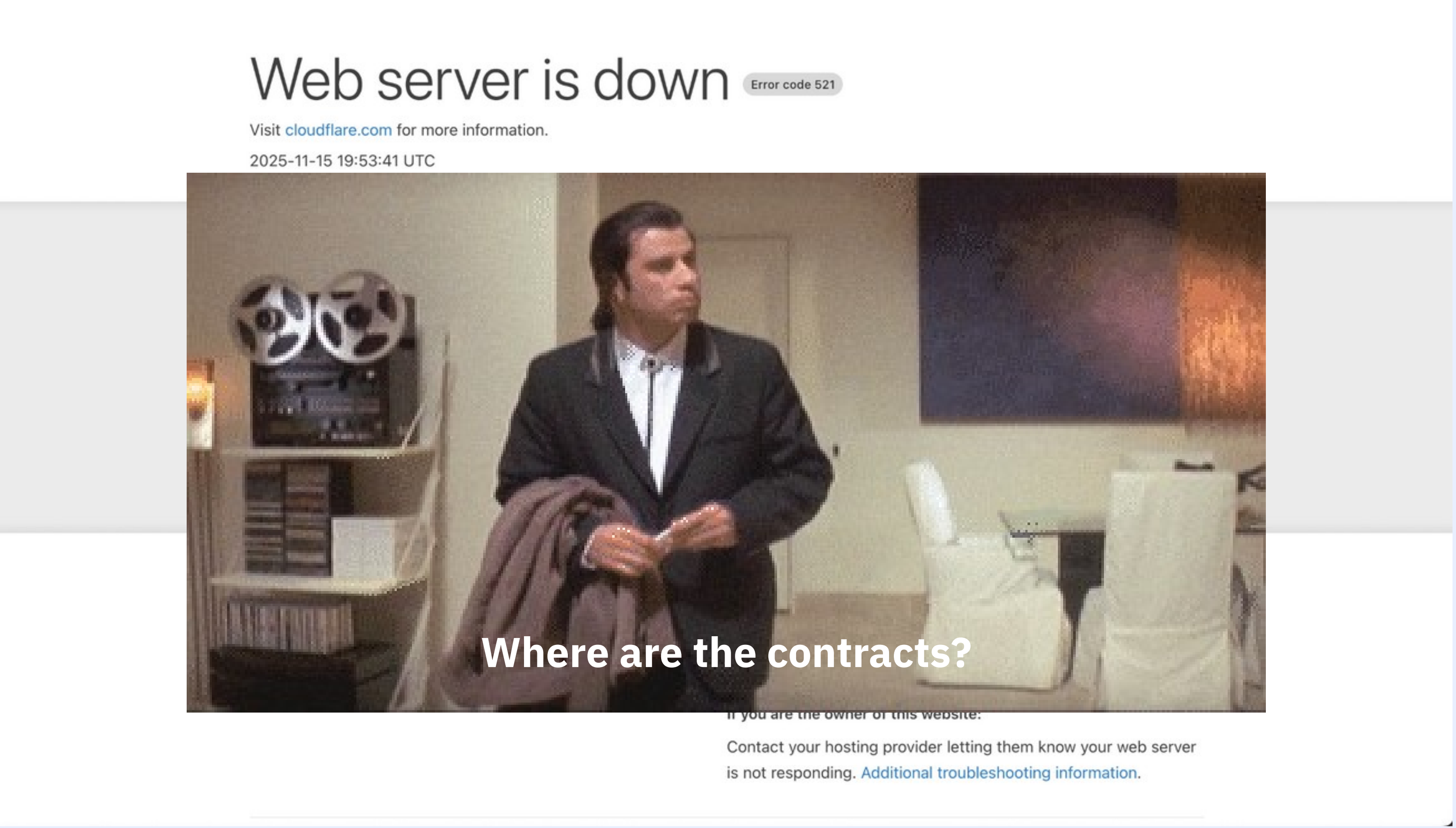
## What can I do?

**If you are a visitor of this website:**

Please try again in a few minutes.

**If you are the owner of this website:**

Contact your hosting provider letting them know your web server is not responding. Additional troubleshooting information.



# Web server is down

Error code 521

Visit cloudflare.com for more information.

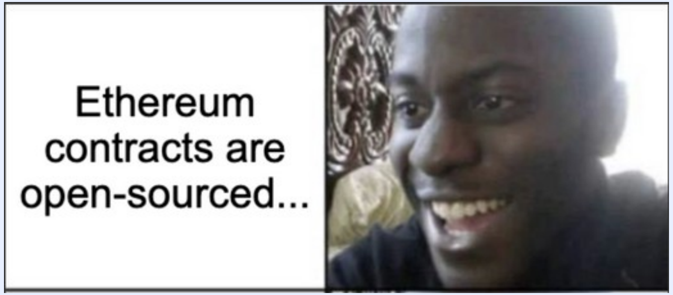
2025-11-15 19:53:41 UTC

Where are the contracts?

If you are the owner of this website:

Contact your hosting provider letting them know your web server is not responding. Additional troubleshooting information.

# Ethereum has good verification culture


Ethereum contracts are open-sourced...

Ethereum contracts are open-sourced...
...through close source verifiers

Ethereum contracts are open-sourced...
...through close source verifiers
...and closed datasets

# Problems

- **Open-source**
  - Largest verifier is close-source. Verification process is a **black box**.
- **Closed data**
  - EVM data is onchain and public, anyone can build a block explorer
  - Verified contracts are **offchain and siloed**, you depend on whose database it's on.
- **Centralized**
  - If it's gone, it's gone
  - ▯ [http://goerli.etherscan.io/address/0x7a250d5630B4cF539739dF2C5dAcb4c659F2488D](http://goerli.etherscan.io/address/0x7a250d5630B4cF539739dF2C5dAcb4c659F2488D)
  - ▯ [https://rinkeby.etherscan.io/address/0x74FbCcEB9a11821cF7098494d887881C2f0e479a](https://rinkeby.etherscan.io/address/0x74FbCcEB9a11821cF7098494d887881C2f0e479a)
- **Stagnated tooling and suboptimal devX**
  - Why did the verification fail?
  - API keys
  - Lacking rich compilation data
    - sourceMaps, bytecodes, storageLayout, immutables, full std-JSON input/output

# Problems

- Current players not incentivized to open-up themselves.
- Requires active pushing and raising awareness
- Not a “sexy” thing to build

sourcify.dev

Verifier Alliance

sourcify.dev

- Solidity side project, within EF
- Spun out w/Solidity to argot.org
- Non-profit, **no revenue**
- 3 person team

ethereum foundation

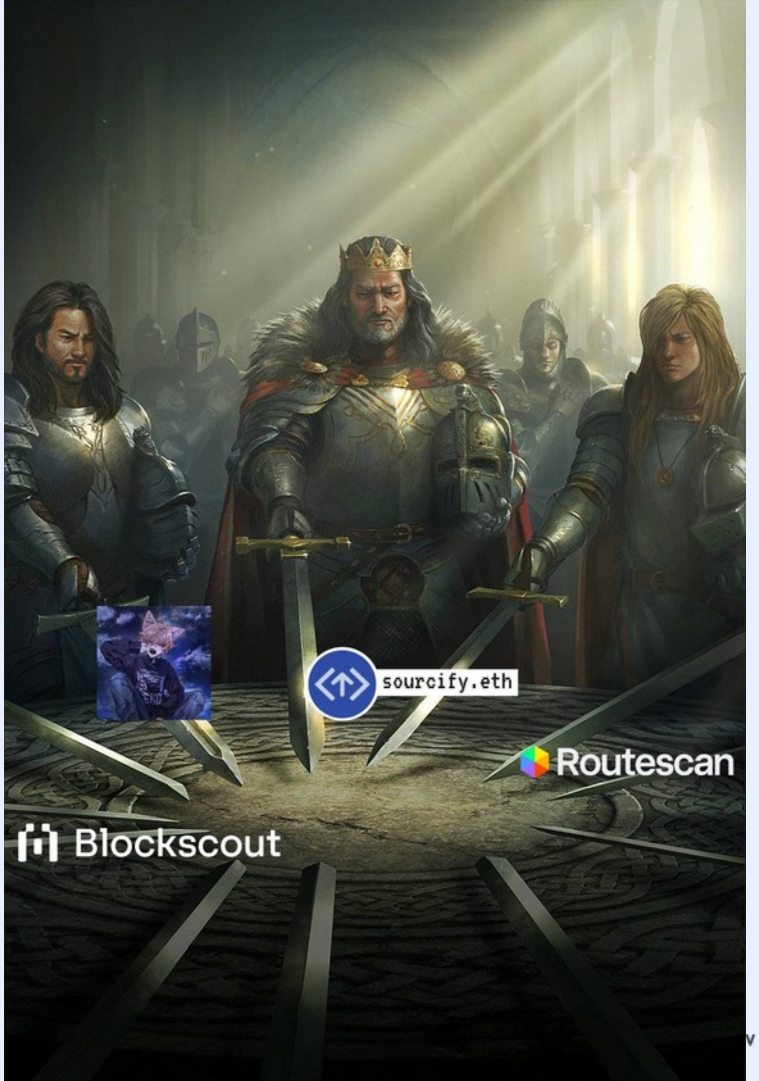
ARGOT

@manuelwedler

@kuzdogan

@marcocastignoli

# Verifier Alliance



# Verifier Alliance

- Easy, unified, and open access to the source-code of EVM smart contracts.
- Public shared Postgres database of verified contracts.
- Exported daily in Parquet format

Blockscout

Routescan

sourcify.dev

verifieralliance.org

# Solutions

# Solutions

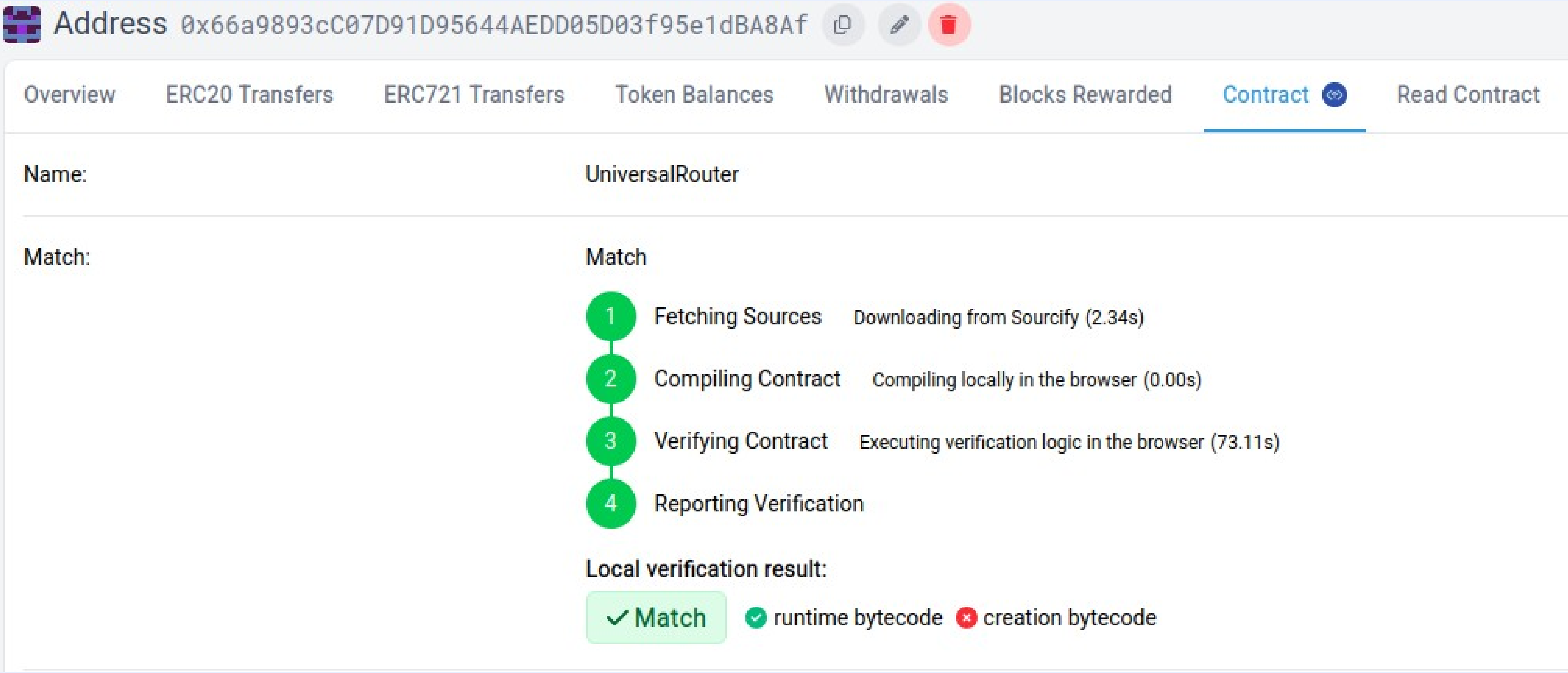
- Open Source EVERYTHING

# Solutions

- **Open Source EVERYTHING**
  - Verification logic `@ethereum-sourcify/lib-sourcify`

# Solutions

- **Open Source EVERYTHING**
  - Verification logic `@ethereum-sourcify/lib-sourcify`
    - *Browser friendly*
    - Example: Local verification on **Otterscan**



Address 0x66a9893cC07D91D95644AEDD05D03f95e1dBA8Af

Name: UniversalRouter

Match: Match

1 Fetching Sources Downloading from Sourcify (2.34s)

2 Compiling Contract Compiling locally in the browser (0.00s)

3 Verifying Contract Executing verification logic in the browser (73.11s)

4 Reporting Verification

Local verification result:

✓ Match ✓ runtime bytecode ✗ creation bytecode

# Solutions

- **Open Source EVERYTHING**
  - Verification logic `@ethereum-sourcify/lib-sourcify`

# Solutions

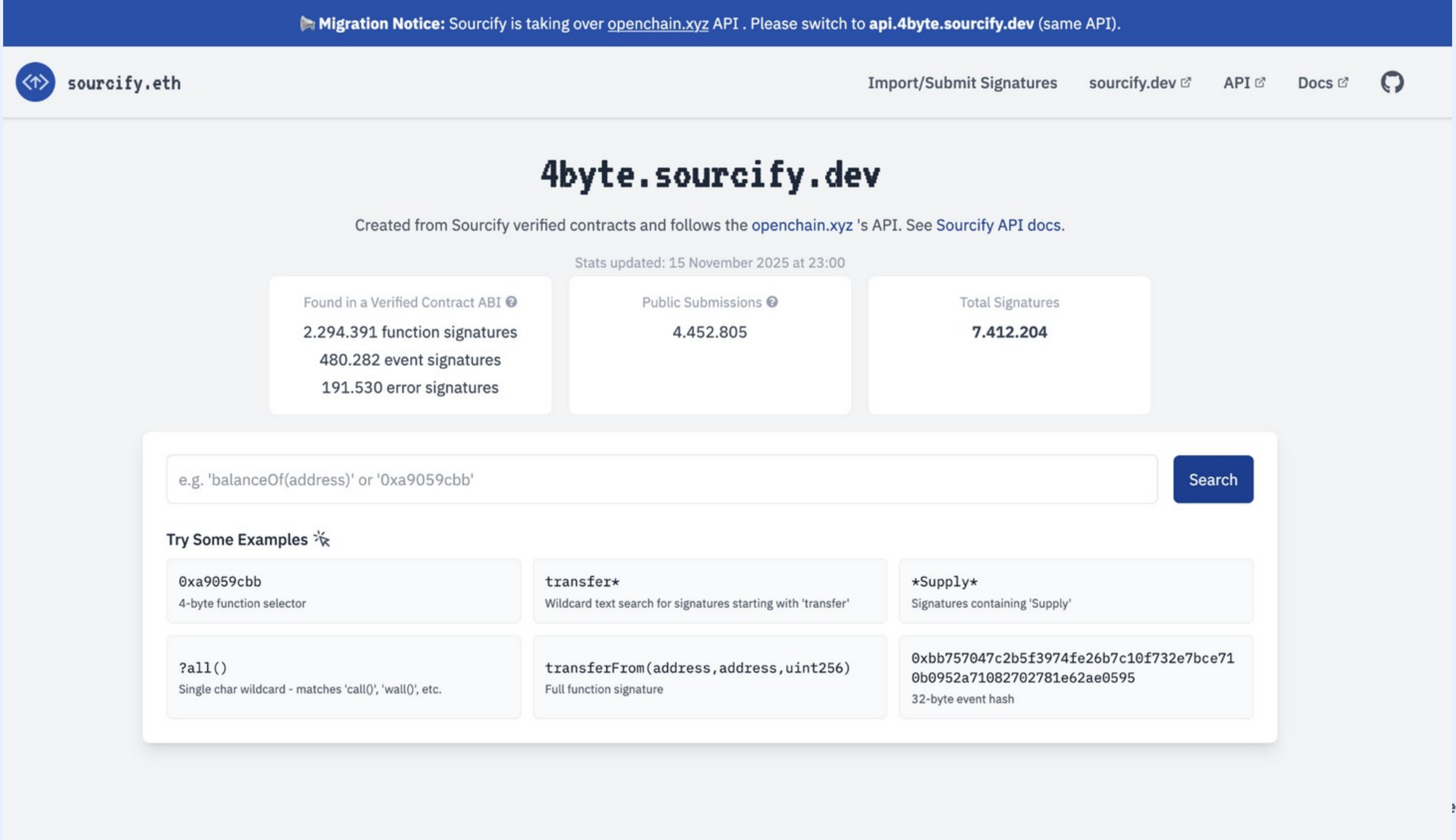
- **Open Source EVERYTHING**
  - Verification logic `@ethereum-sourcify/lib-sourcify`
  - Sourcify server: https://sourcify.dev/server

# Solutions

- **Open Source EVERYTHING**
    - Verification logic `@ethereum-sourcify/lib-sourcify`
    - Sourcify server: https://sourcify.dev/server
    - Repository UI: https://repo.sourcify.dev

# Solutions

- **Open Source EVERYTHING**
  - Verification logic `@ethereum-sourcify/lib-sourcify`
  - Sourcify server: https://sourcify.dev/server
  - Repository UI: https://repo.sourcify.dev
  - 4byte signature database: https://4byte.sourcify.dev
    - api.openchain.xyz → api.4byte.sourcify.dev



**Migration Notice:** Sourcify is taking over openchain.xyz API . Please switch to **api.4byte.sourcify.dev** (same API).

# 4byte.sourcify.dev

Created from Sourcify verified contracts and follows the openchain.xyz 's API. See Sourcify API docs.

Stats updated: 15 November 2025 at 23:00

Found in a Verified Contract ABI

2.294.391 function signatures
480.282 event signatures
191.530 error signatures

Public Submissions

4.452.805

Total Signatures

**7.412.204**

e.g. 'balanceOf(address)' or '0xa9059cbb'

Try Some Examples

`0xa9059cbb`
4-byte function selector

`transfer*`
Wildcard text search for signatures starting with 'transfer'

`*Supply*`
Signatures containing 'Supply'

`?all()`
Single char wildcard - matches 'call()', 'wall()', etc.

`transferFrom(address,address,uint256)`
Full function signature

`0xbb757047c2b5f3974fe26b7c10f732e7bce710b0952a71082702781e62ae0595`
32-byte event hash

ev

# Solutions

- **Open Source EVERYTHING**
  - Verification logic `@ethereum-sourcify/lib-sourcify`
  - Sourcify server: https://sourcify.dev/server
  - Repository UI: https://repo.sourcify.dev
  - 4byte signature database: https://4byte.sourcify.dev
    - api.openchain.xyz → api.4byte.sourcify.dev

# Solutions

- **Open Source EVERYTHING**
  - Verification logic `@ethereum-sourcify/lib-sourcify`
  - Sourcify server: https://sourcify.dev/server
  - Repository UI: https://repo.sourcify.dev
  - 4byte signature database: https://4byte.sourcify.dev
    - api.openchain.xyz → api.4byte.sourcify.dev
  - Self hosted Sourcify instances:
    - https://server-verify.hashscan.io
    - https://sourcify-api-monad.blockvision.org
    - https://sourcify.roninchain.com/server
    - https://sourcify.parsec.finance
    - https://verify-api.vechainstats.com
    - https://sourcify.walnut.dev

# Solutions

- Open Source EVERYTHING

# Solutions

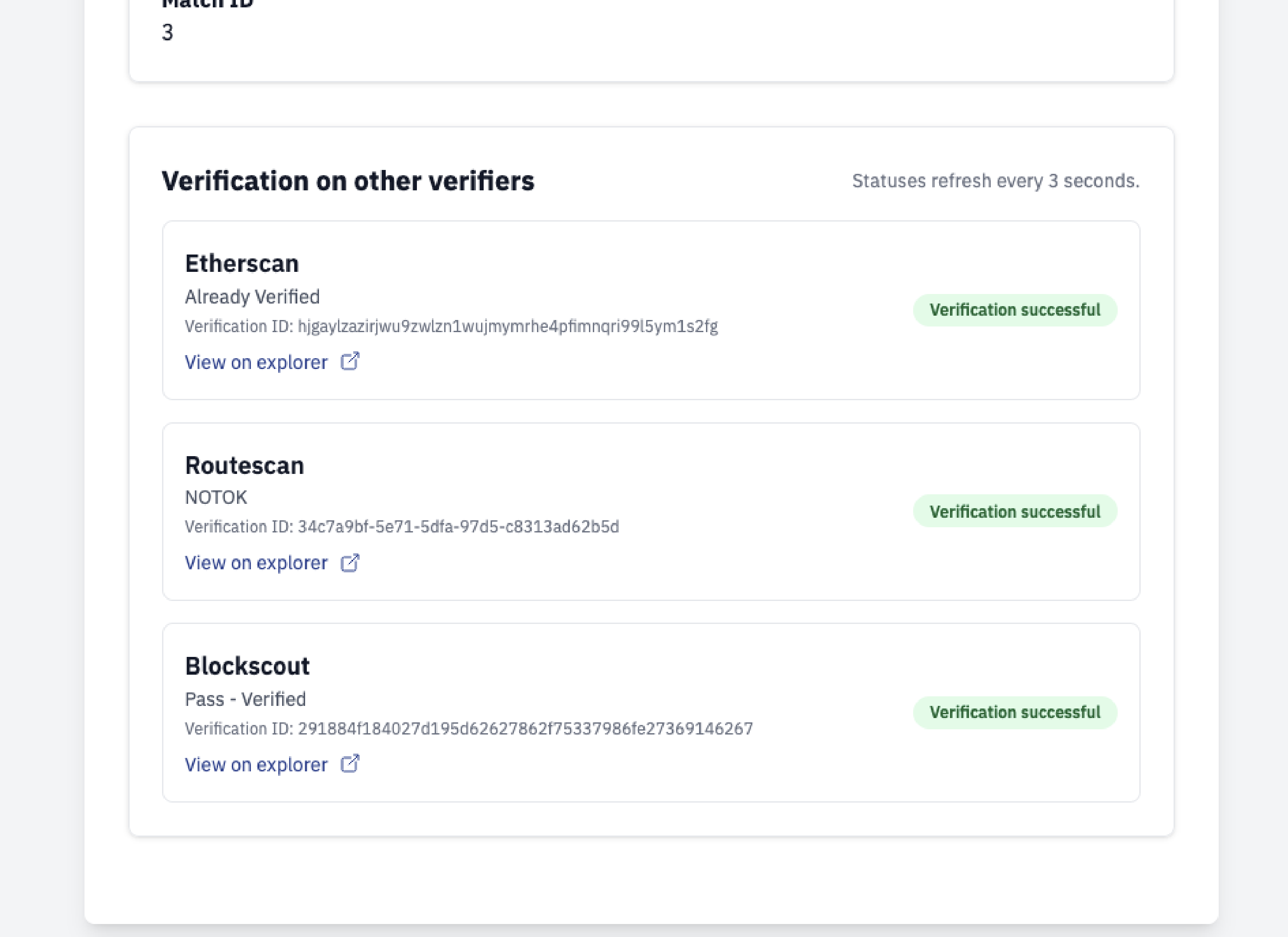
- Open Source EVERYTHING
- Verify EVERYWHERE

# Solutions

- **Open Source EVERYTHING**
- **Verify EVERYWHERE**
    - Sourcify backend: Verified on Sourcify → Call other verifiers
        - Etherscan limitation: Max. 250 verif./day regardless of tier

# Solu

- O
- Ve



Match ID

3

## Verification on other verifiers

Statuses refresh every 3 seconds.

**Etherscan**

Already Verified

Verification ID: hjgaylzazirjwu9zwlzn1wujmymrhe4pfimnqri99l5ym1s2fg

View on explorer

Verification successful

**Routescan**

NOTOK

Verification ID: 34c7a9bf-5e71-5dfa-97d5-c8313ad62b5d

View on explorer

Verification successful

**Blockscout**

Pass - Verified

Verification ID: 291884f184027d195d62627862f75337986fe27369146267

View on explorer

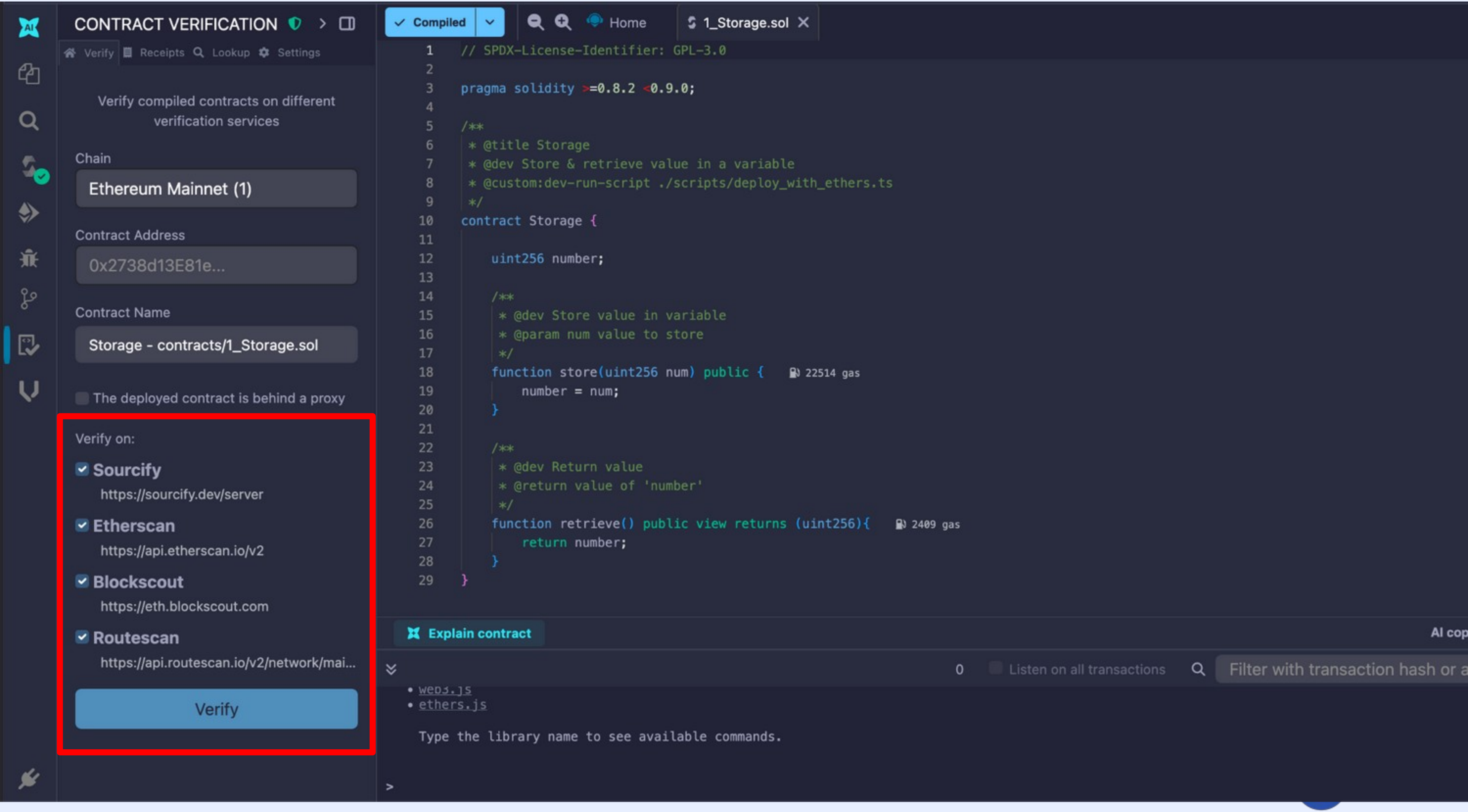
Verification successful

# Solutions

- **Open Source EVERYTHING**
- **Verify EVERYWHERE**
  - Sourcify backend: Verified on Sourcify → Call other verifiers
    - Etherscan limitation: Max. 250 verif./day regardless of tier
  - Verifier Alliance

# Solutions

- **Open Source EVERYTHING**
- **Verify EVERYWHERE**
    - Sourcify backend: Verified on Sourcify → Call other verifiers
        - Etherscan limitation: Max. 250 verif./day regardless of tier
    - Verifier Alliance
    - Remix Plugin



Verify compiled contracts on different verification services

Chain: Ethereum Mainnet (1)

Contract Address: 0x2738d13E81e...

Contract Name: Storage - contracts/1_Storage.sol

□ The deployed contract is behind a proxy

Verify on:

☑ **Sourcify**
https://sourcify.dev/server

☑ **Etherscan**
https://api.etherscan.io/v2

☑ **Blockscout**
https://eth.blockscout.com

☑ **Routescan**
https://api.routescan.io/v2/network/mai...

Verify

```
// SPDX-License-Identifier: GPL-3.0

pragma solidity >=0.8.2 <0.9.0;

/**
 * @title Storage
 * @dev Store & retrieve value in a variable
 * @custom:dev-run-script ./scripts/deploy_with_ethers.ts
 */
contract Storage {

    uint256 number;

    /**
     * @dev Store value in variable
     * @param num value to store
     */
    function store(uint256 num) public {
        number = num;
    }

    /**
     * @dev Return value
     * @return value of 'number'
     */
    function retrieve() public view returns (uint256){
        return number;
    }
}
```

```
• web3.js
• ethers.js

Type the library name to see available commands.

>
```

# Solutions

- **Open Source EVERYTHING**
- **Verify EVERYWHERE**
  - Sourcify backend: Verified on Sourcify → Call other verifiers
    - Etherscan limitation: Max. 250 verif./day regardless of tier
  - Verifier Alliance
  - Remix Plugin

# Solutions

- **Open Source EVERYTHING**
- **Verify EVERYWHERE**
    - Sourcify backend: Verified on Sourcify → Call other verifiers
        - Etherscan limitation: Max. 250 verif./day regardless of tier
    - Verifier Alliance
    - Remix Plugin
    - Hardhat v3
    - Foundry (TBD)

# Solutions

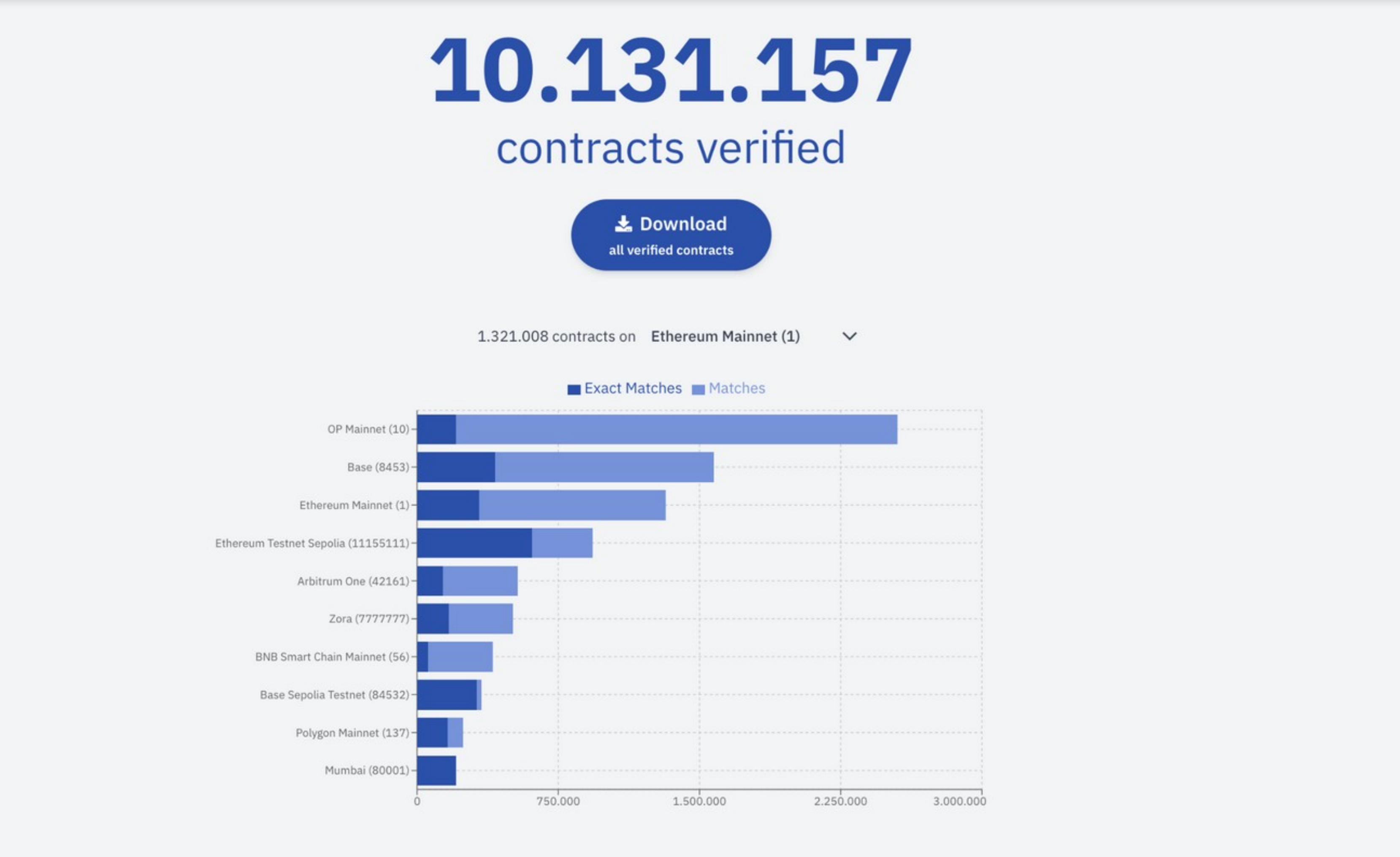
- Open Source EVERYTHING
- Verify EVERYWHERE

# Solutions

- Open Source EVERYTHING
- Verify EVERYWHERE
- Open data

# Solutions

- **Open Source EVERYTHING**
- **Verify EVERYWHERE**
- **Open data**
  - Share whole dataset daily in Parquet format: export.sourcify.dev & export.verifieralliance.org



# 10.131.157

contracts verified

Download
all verified contracts

1.321.008 contracts on Ethereum Mainnet (1)

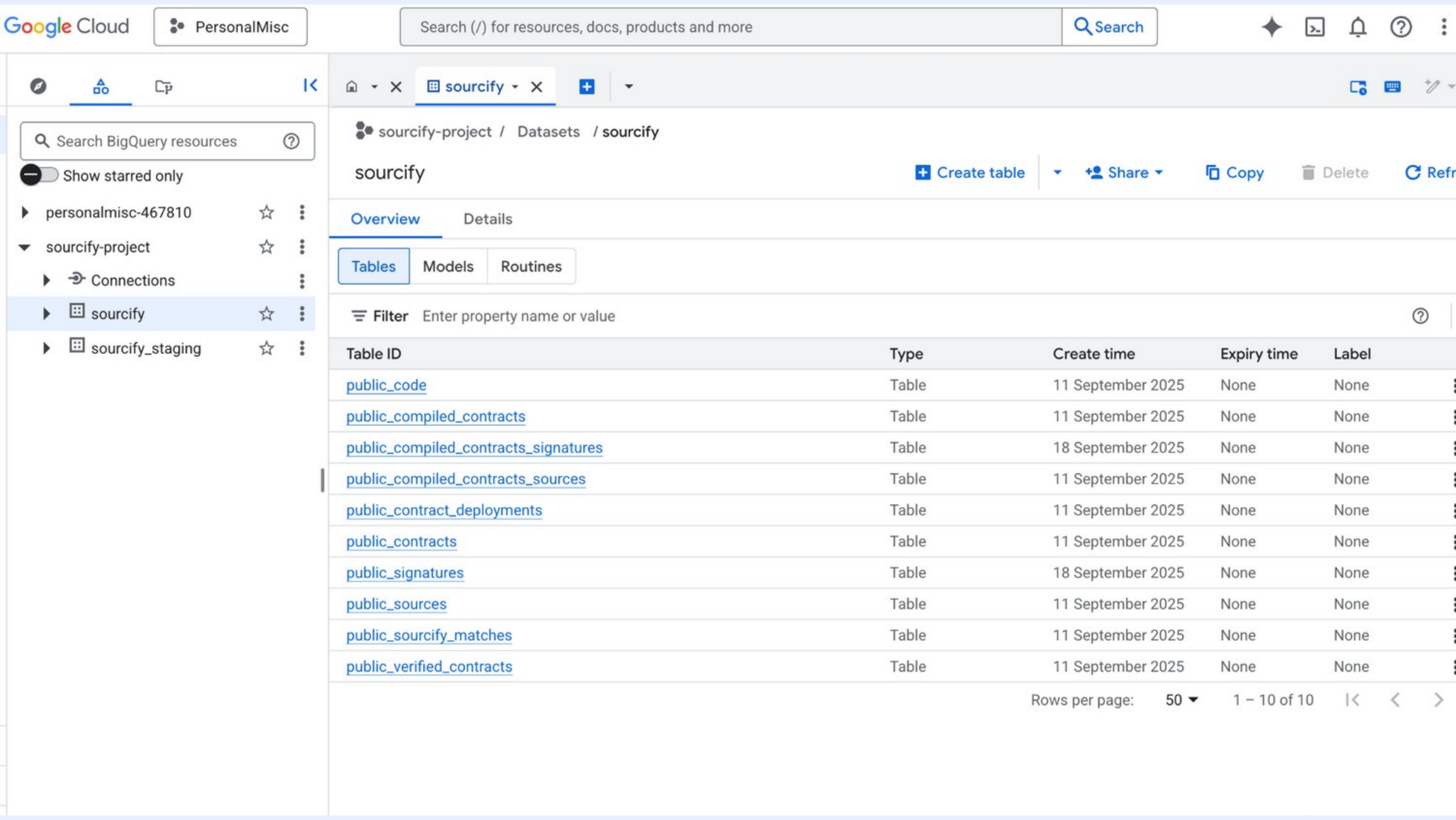
ourcify.dev

# Solutions

- **Open Source EVERYTHING**
- **Verify EVERYWHERE**
- **Open data**
  - Share whole dataset daily in Parquet format: export.sourcify.dev & export.verifieralliance.org

# Solutions

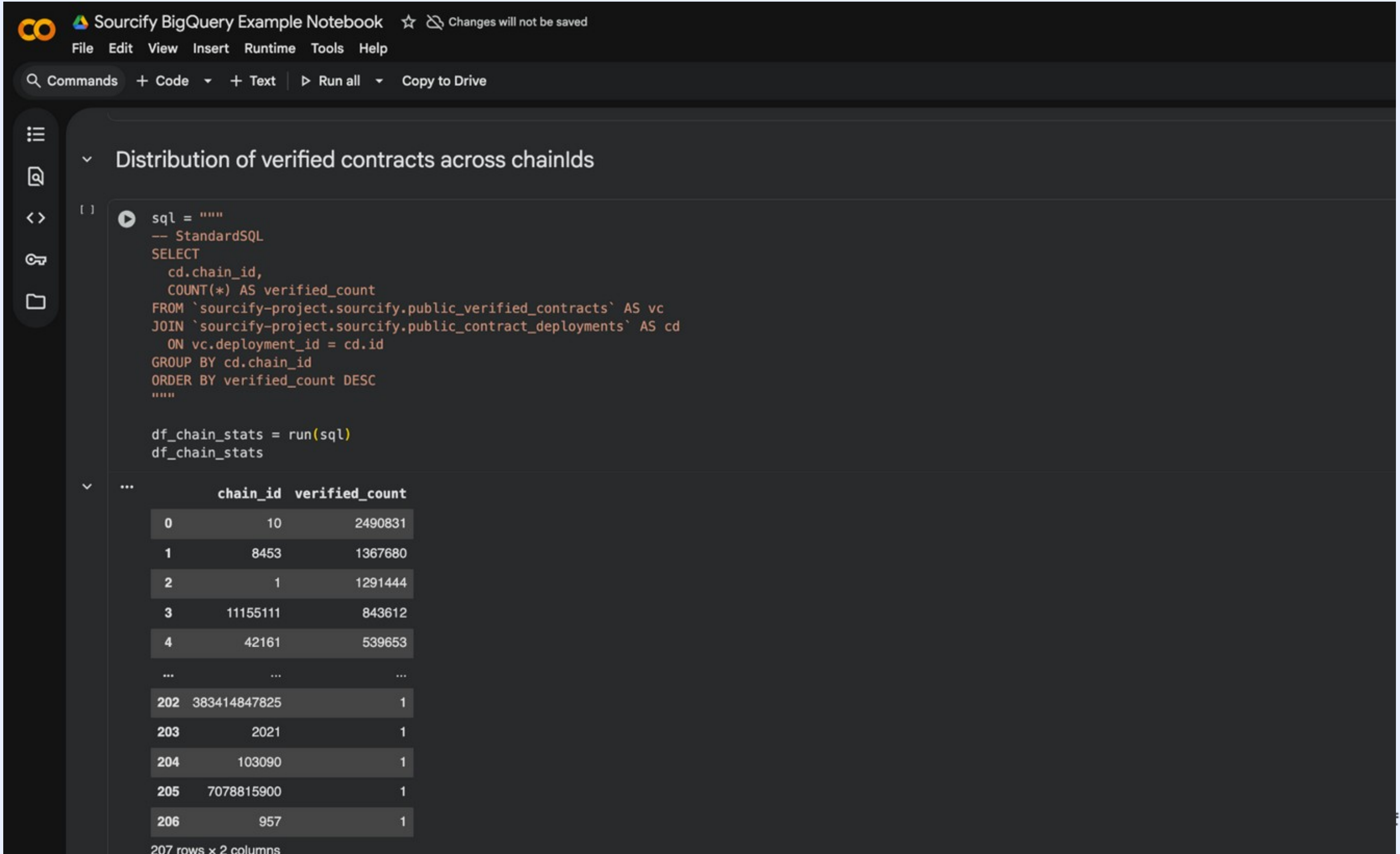
- **Open Source EVERYTHING**
- **Verify EVERYWHERE**
- **Open data**
  - Share whole dataset daily in Parquet format: export.sourcify.dev & export.verifieralliance.org
  - BigQuery and Google Colab (Jupyter) Notebook



sourcify

| Table ID | Type | Create time | Expiry time | Label |
|---|---|---|---|---|
| public_code | Table | 11 September 2025 | None | None |
| public_compiled_contracts | Table | 11 September 2025 | None | None |
| public_compiled_contracts_signatures | Table | 18 September 2025 | None | None |
| public_compiled_contracts_sources | Table | 11 September 2025 | None | None |
| public_contract_deployments | Table | 11 September 2025 | None | None |
| public_contracts | Table | 11 September 2025 | None | None |
| public_signatures | Table | 18 September 2025 | None | None |
| public_sources | Table | 11 September 2025 | None | None |
| public_sourcify_matches | Table | 11 September 2025 | None | None |
| public_verified_contracts | Table | 11 September 2025 | None | None |

Rows per page: 50 1 – 10 of 10



## Distribution of verified contracts across chainIds

```
sql = """
-- StandardSQL
SELECT
  cd.chain_id,
  COUNT(*) AS verified_count
FROM `sourcify-project.sourcify.public_verified_contracts` AS vc
JOIN `sourcify-project.sourcify.public_contract_deployments` AS cd
  ON vc.deployment_id = cd.id
GROUP BY cd.chain_id
ORDER BY verified_count DESC
"""

df_chain_stats = run(sql)
df_chain_stats
```

| | chain_id | verified_count |
|---|---|---|
| 0 | 10 | 2490831 |
| 1 | 8453 | 1367680 |
| 2 | 1 | 1291444 |
| 3 | 11155111 | 843612 |
| 4 | 42161 | 539653 |
| ... | ... | ... |
| 202 | 383414847825 | 1 |
| 203 | 2021 | 1 |
| 204 | 103090 | 1 |
| 205 | 7078815900 | 1 |
| 206 | 957 | 1 |

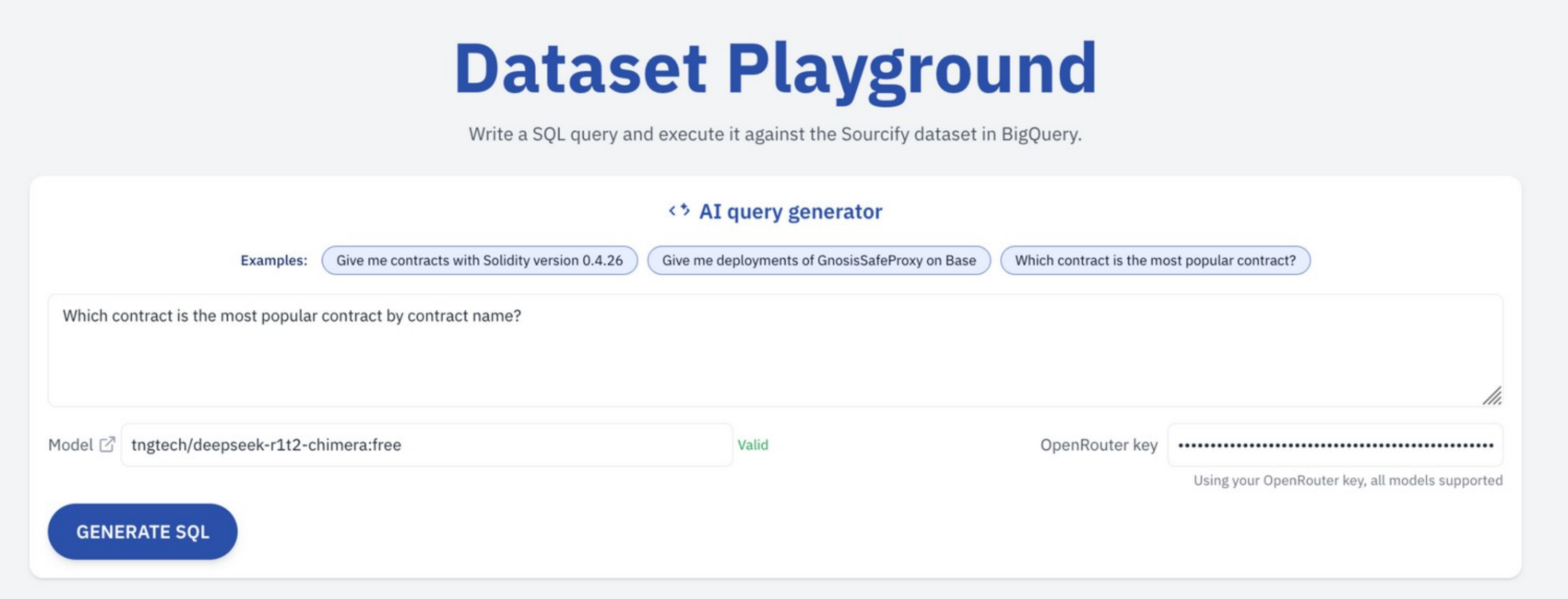
207 rows × 2 columns

# Solutions

- **Open Source EVERYTHING**
- **Verify EVERYWHERE**
- **Open data**
    - Share whole dataset daily in Parquet format: export.sourcify.dev & export.verifieralliance.org
    - BigQuery and Google Colab (Jupyter) Notebook

# Solutions

- **Open Source EVERYTHING**
- **Verify EVERYWHERE**
- **Open data**
  - Share whole dataset daily in Parquet format: export.sourcify.dev & export.verifieralliance.org
  - BigQuery and Google Colab (Jupyter) Notebook
  - AI Playground



# Dataset Playground

Write a SQL query and execute it against the Sourcify dataset in BigQuery.

## AI query generator

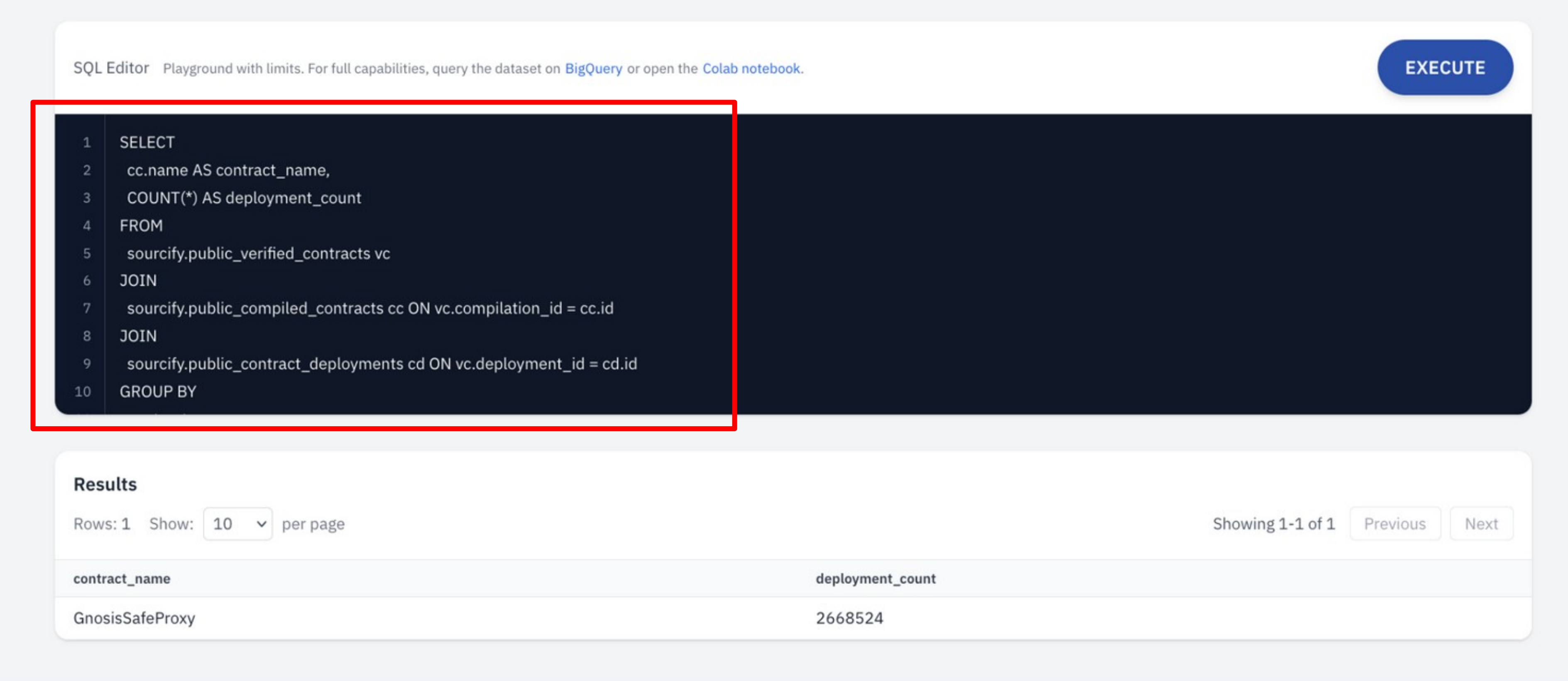
Examples: Give me contracts with Solidity version 0.4.26 | Give me deployments of GnosisSafeProxy on Base | Which contract is the most popular contract?

Which contract is the most popular contract by contract name?

Model: tngtech/deepseek-r1t2-chimera:free Valid

OpenRouter key: ••••••••••••••••••••••••••••••••••••••••••••••••

Using your OpenRouter key, all models supported



SQL Editor Playground with limits. For full capabilities, query the dataset on BigQuery or open the Colab notebook.

```
SELECT
 cc.name AS contract_name,
 COUNT(*) AS deployment_count
FROM
 sourcify.public_verified_contracts vc
JOIN
 sourcify.public_compiled_contracts cc ON vc.compilation_id = cc.id
JOIN
 sourcify.public_contract_deployments cd ON vc.deployment_id = cd.id
GROUP BY
```

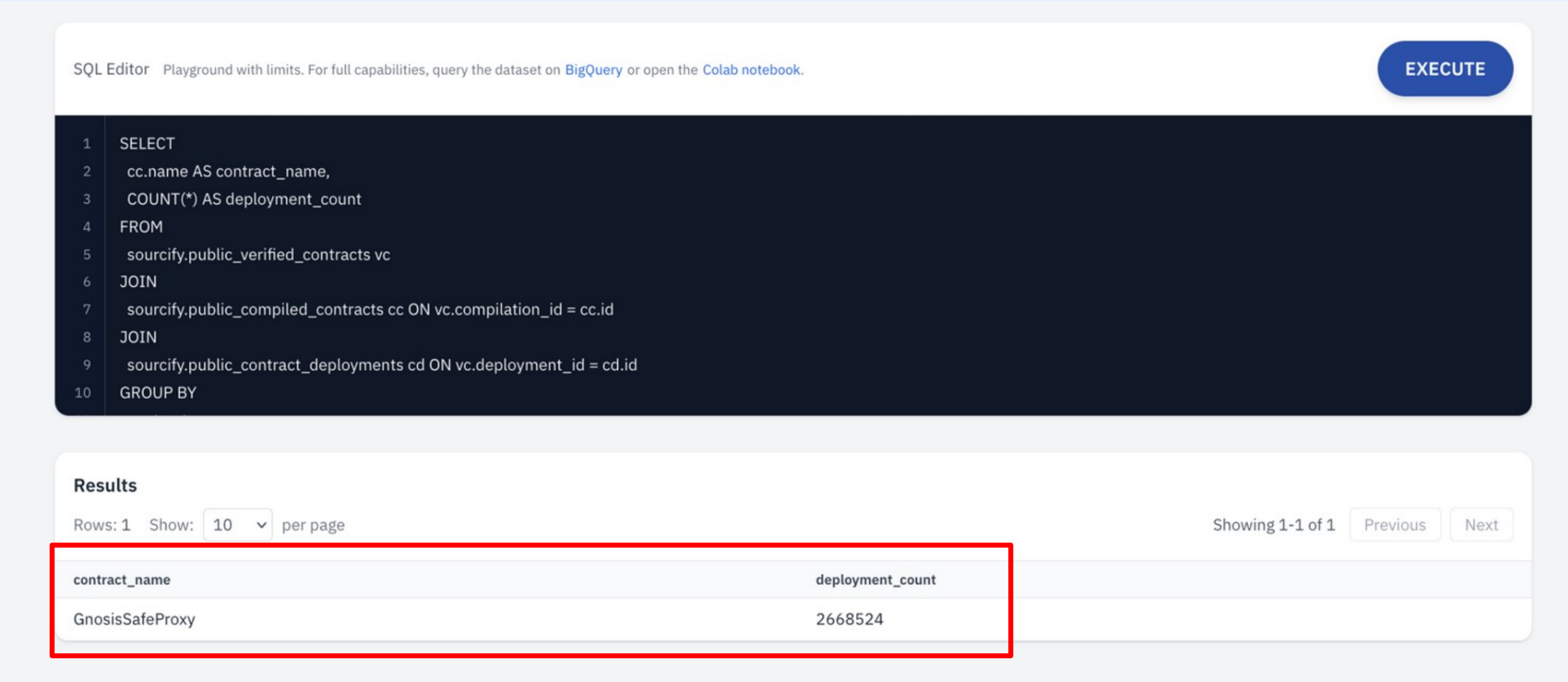
**Results**

Rows: 1 Show: 10 per page

Showing 1-1 of 1

| contract_name | deployment_count |
|---|---|
| GnosisSafeProxy | 2668524 |

sourcify.dev



SQL Editor Playground with limits. For full capabilities, query the dataset on BigQuery or open the Colab notebook.

```
SELECT
 cc.name AS contract_name,
 COUNT(*) AS deployment_count
FROM
 sourcify.public_verified_contracts vc
JOIN
 sourcify.public_compiled_contracts cc ON vc.compilation_id = cc.id
JOIN
 sourcify.public_contract_deployments cd ON vc.deployment_id = cd.id
GROUP BY
```

**Results**

Rows: 1 Show: 10 per page Showing 1-1 of 1

| contract_name | deployment_count |
|---|---|
| GnosisSafeProxy | 2668524 |

sourcify.dev

# Solutions

- **Open Source EVERYTHING**
- **Verify EVERYWHERE**
- **Open data**
    - Share whole dataset daily in Parquet format: export.sourcify.dev & export.verifieralliance.org
    - BigQuery and Google Colab (Jupyter) Notebook
    - AI Playground

# Solutions

- **Open Source EVERYTHING**
- **Verify EVERYWHERE**
- **Open data**
  - EVM Tooling testing in bulk
  - Data analysis
  - Feed an AI model
  - Build better decompiler
  - Find vulnerability patterns
  - **Use this valuable dataset to move EVM tooling forward!**

Where are the contracts?

# Thank you

https://x.com/sourcifyeth

https://x.com/verifalliance

Matrix: @kuzdogan:matrix.org
Telegram: @kuzdogan

# What is source code verification?

# What is source code verification?

- Ethereum contracts are stored on the blockchain in **bytecode**

# What is source code verification?

- Ethereum contracts are stored on the blockchain in **bytecode**

```
0x60806040526004361061003f5760003560e01c806301ffc9a71461004457806322895118146100a4578063621fd130146101ba578063c5f2892f1
4610244575b600080fd5b34801561005057600080fd5b506100906004803603602081101561006757600080fd5b50357fffffffff00000000000000
0000000000000000000000000000000000000000001661026b565b604080519115158252519081900360200190f35b6101b86004803603608081101
56100ba57600080fd5b8101906020810181356401000000008111156100d557600080fd5b8201836020820111156100e757600080fd5b8035906020
019184600183028401116401000000008311171561010957600080fd5b91939092909160208101903564010000000081111561012757600080fd5b8
2018360208201111561013957600080fd5b80359060200191846001830284011164010000000083111715610 15b57600080fd5b919390929091602 0
81019035640100000000811115610179576000 80fd5b8201836020820111156101 8b57600080fd5b80359060200191846001830284011164010000 0
000831117156101ad57600080fd5b919350915035610304565b005b3480156101c657600080fd5b506101cf6110b5565b6040805160208082528351
818301528351919283929083019185019080838360005b838110156102095781810151838201526020016101f1565b50505050905090810190601f1
680156102365780820380516001836020036101000a031916815260200191505b509250505060405180910390f35b34801561025057600080fd5b50
6102596110c7565b604080519182525190819003602001 90f35b60007fffffffff00000000000000000000000000000000000000000000000000000
00082167f01ffc9a70000000000000000000000000000000000000000000000000000000014806102fe57507fffffffff0000000000000000000000
000000000000000000000000000000000082167f8564090700000000000000000000000000000000000000000000000000000000145b92915050565
b6030861461035d576040517f08c379a000000000000000000000000000000000000000000000000000000000815260040180806020018281038252
602681526020018061180560269139604001915050604051809103 90fd5b602084146103b6576040517f08c379a0000000000000000000000000000
00000000000000000000000000000815260040180806020018281038252603681526020018061179c6036913960400191505060405180910390fd5b
6060821461040f576040517f08c379a0000000000000000000000000000000000000000000000000000000008152600401808060200182810382526
029815260200180611878602991396040019150506040518091039 0fd5b670de0b6b3a76400003410156104705760 40517f08c379a0000000000000
00000000000000000000000000000000000000000000815260040180806020018281038252602681526020018061185260269139604001915050604
05180910390fd5b633b9aca0034061561 04cd576040517f08c379a00000000000000000000000000000000000000000000000000000000081526004
01808060200182810382526033815260200180611 7d60339139604001915050604051809103 90fd5b633b9aca00340467ffffffffffffffff81111
5610535576040517f08c379a00000000000000000000000000000000000000000000000000000000081526004018080602001828103825260278152
6020018061182b60279139604001915050604051809103 90fd5b6060610540826114ba565b90507f649bbc62d0e31342afea4e5cd82d4049e7e1ee9
```

# What is source code verification?

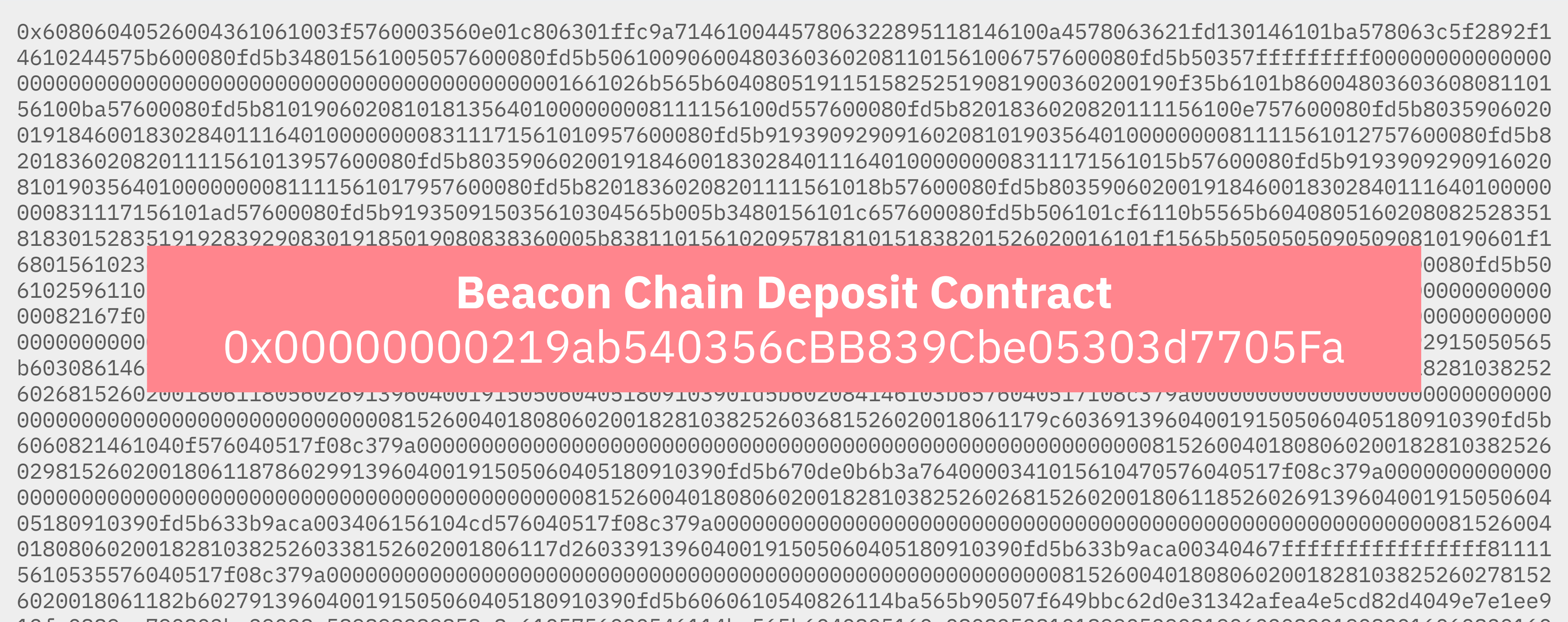
- Ethereum contracts are stored on the blockchain in **bytecode**

```
0x6080604052600436106100 3f5760003560e01c806301ffc9a714610044578063228951181461 00a4578063621fd130146101ba578063c5f2892f1
4610244575b600080fd5b348015610050576000 80fd5b506100906004803603602081101561006757600080fd5b50357ffffffff000000000000000
000000000000000000000000000000000000000001661026b565b604080519115158252519081900360200190f35b6101b860048036036080811 01
56100ba57600080fd5b810190602081018135640100000000811115610 0d557600080fd5b8201836020820111156100e757600080fd5b8035906020
0191846001830284011164010000000083111715610109576000 80fd5b919390929091602081019035640100000000811115610127576000 80fd5b8
2018360208201111561013957600080fd5b80359060200191846001830284011164010000000083111715610 15b57600080fd5b9193909290916020
81019035640100000000811115610179576000 80fd5b8201836020820111156101 8b57600080fd5b803590602001918460018302840111640100000
000831117156101ad57600080fd5b919350915035610304565b005b3480156101c657600080fd5b506101cf6110b5565b604080516020808252 8351
818301528351919283929083019185019080838360005b83811015610209578181015183820152602001610 1f1565b50505050905090810190601f1
680156102365780820380516001836020036101000a031916815260200191505b509250505060405180910390f35b34801561025057600080fd5b50
6102596110c7565b604080519182525190819003                                        0000000000000000000000000000000000000000
00082167f01ffc9a700000000000000000000000                                        02fe57507ffffffff00000000000000000000000
0000000000000000000000000000000000082167f                                       0000000000000000000000000145b92915050565
b6030861461035d576040517f08c379a000000000000000000000000000000000000000000000000000000000815260040180806020018281038252
602681526020018061180560269139604001915050604051809103 90fd5b602084146103b6576040517f08c379a0000000000000000000000000000
0000000000000000000000000000081526004018080602001828103825260368152602001806117 9c6036913960400191505060405180910 390fd5b
6060821461040f576040517f08c379a00000000000000000000000000000000000000000000000000000000081526004018080602001828103825 26
029815260200180611878602991396040019150506040518091 0390fd5b670de0b6b3a76400003410156104705760 40517f08c379a0000000000000
0000000000000000000000000000000000000000000081526004018080602001828103825260268152602001806118526026913960400191505060 4
05180910390fd5b633b9aca003406156104cd576040517f08c379a00000000000000000000000000000000000000000000000000000000081526004
0180806020018281038252603381526020018061 17d6033913960400191505060405180910390fd5b633b9aca00340467ffffffffffffffff81111
5610535576040517f08c379a00000000000000000000000000000000000000000000000000000000081526004018080602001828103825260278152
6020018061182b6027913960400191505060405180910 390fd5b60606105408261 14ba565b90507f649bbc62d0e31342afea4e5cd82d4049e7e1ee9
```

What does it do?

# What is source code verification?

- Ethereum contracts are stored on the blockchain in **bytecode**



```
0x60806040526004361061003f5760003560e01c806301ffc9a71461004457806322895118146100a4578063621fd130146101ba578063c5f2892f1
4610244575b600080fd5b34801561005057600080fd5b506100906004803603602081101561006757600080fd5b50357fffffffff00000000000000
000000000000000000000000000000000000000000001661026b565b604080519115158252519081900360200190f35b6101b86004803603608081101
56100ba57600080fd5b8101906020810181356401000000008111156100d557600080fd5b8201836020820111156100e757600080fd5b8035906020
0191846001830284011164010000000083111715610109576000080fd5b91939092909160208101903564010000000081111561012757600080fd5b8
2018360208201111561013957600080fd5b80359060200191846001830284011164010000000083111715610015b57600080fd5b9193909290916020
8101903564010000000081111561017957600080fd5b82018360208201111561018b57600080fd5b803590602001918460018302840111640100000
000831117156101ad57600080fd5b919350915035610304565b005b3480156101c657600080fd5b506101cf6110b5565b6040805160208082528351
818301528351919283929083019185019080838360005b83811015610209578181015183820152602001610f1565b5050505090509081019060f1
6801561023                                                                                                   0080fd5b50
6102596110                                                                                                   0000000000
00082167f0                                                                                                   0000000000
0000000000                                                                                                   2915050565
b603086146                                                                                                   8281038252
60268152602001806118056026913960400191505060405180910390fd5b602084146103b6576040517f08c379a0000000000000000000000000000
0000000000000000000000000000081526004018080602001828103825260368152602001806117 9c6036913960400191505060405180910390fd5b
6060821461040f576040517f08c379a00000000000000000000000000000000000000000000000000000000081526004018080602001828103825 26
029815260200180611878602991396040019150506040518091 0390fd5b670de0b6b3a76400003410156104705760 40517f08c379a00000000000000
000000000000000000000000000000000000000000081526004018080602001828103825260268152602001806118526026913960400191505060 4
05180910390fd5b633b9aca003406156104cd576040517f08c379a00000000000000000000000000000000000000000000000000000000081526004
018080602001828103825260338152602001806117d26033913960400191505060405180910390fd5b633b9aca00340467ffffffffffffffff81111
5610535576040517f08c379a00000000000000000000000000000000000000000000000000000000081526004018080602001828103825260278152
602001806118 2b6027913960400191505060405180910390fd5b6060610540826114ba565b90507f649bbc62d0e31342afea4e5cd82d4049e7e1ee9
```

**Beacon Chain Deposit Contract**

0x00000000219ab540356cBB839Cbe05303d7705Fa

# What is source code verification?

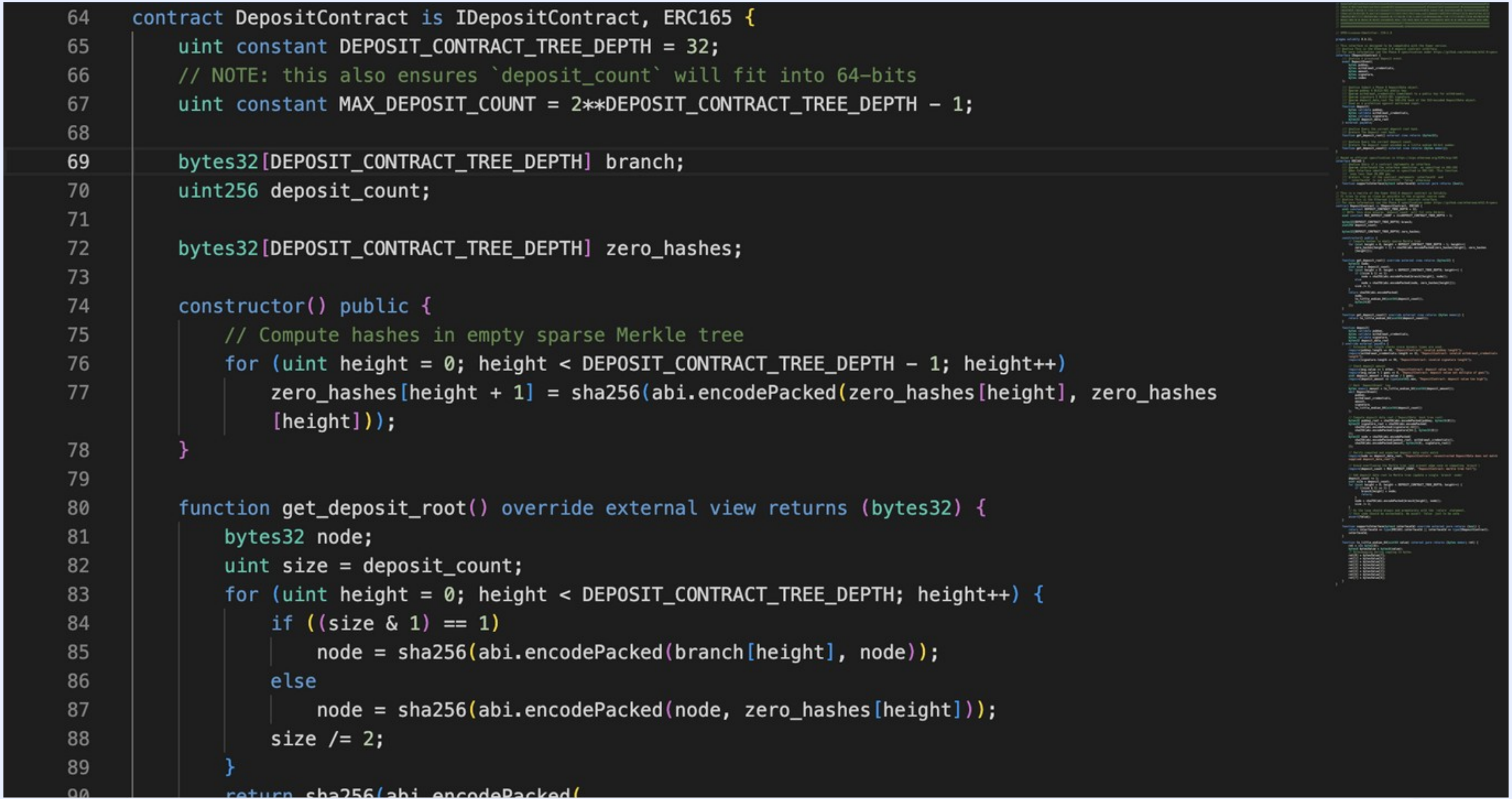
- Ethereum contracts are stored on the blockchain in **bytecode**

# What is source code verification?

- Ethereum contracts are stored on the blockchain in **bytecode**
- Humans can’t read/write bytecode

# What is source code verification?

- Ethereum contracts are stored on the blockchain in **bytecode**
- Humans can’t read/write bytecode
- Humans read/write in prog. languages (Solidity, Vyper...)



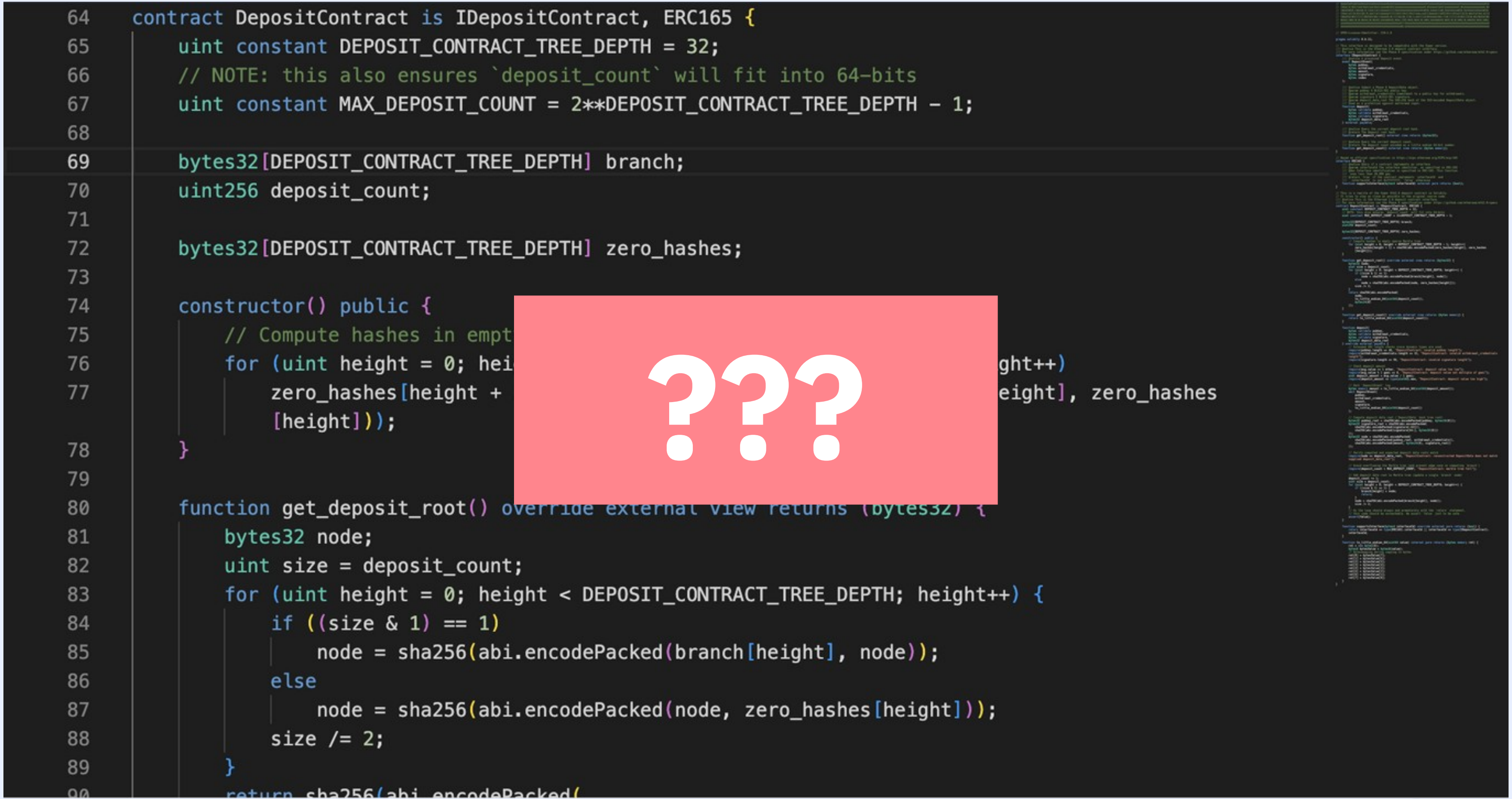
```solidity
contract DepositContract is IDepositContract, ERC165 {
    uint constant DEPOSIT_CONTRACT_TREE_DEPTH = 32;
    // NOTE: this also ensures `deposit_count` will fit into 64-bits
    uint constant MAX_DEPOSIT_COUNT = 2**DEPOSIT_CONTRACT_TREE_DEPTH - 1;

    bytes32[DEPOSIT_CONTRACT_TREE_DEPTH] branch;
    uint256 deposit_count;

    bytes32[DEPOSIT_CONTRACT_TREE_DEPTH] zero_hashes;

    constructor() public {
        // Compute hashes in empty sparse Merkle tree
        for (uint height = 0; height < DEPOSIT_CONTRACT_TREE_DEPTH - 1; height++)
            zero_hashes[height + 1] = sha256(abi.encodePacked(zero_hashes[height], zero_hashes[height]));
    }

    function get_deposit_root() override external view returns (bytes32) {
        bytes32 node;
        uint size = deposit_count;
        for (uint height = 0; height < DEPOSIT_CONTRACT_TREE_DEPTH; height++) {
            if ((size & 1) == 1)
                node = sha256(abi.encodePacked(branch[height], node));
            else
                node = sha256(abi.encodePacked(node, zero_hashes[height]));
            size /= 2;
        }
        return sha256(abi.encodePacked(
```



```
contract DepositContract is IDepositContract, ERC165 {
    uint constant DEPOSIT_CONTRACT_TREE_DEPTH = 32;
    // NOTE: this also ensures `deposit_count` will fit into 64-bits
    uint constant MAX_DEPOSIT_COUNT = 2**DEPOSIT_CONTRACT_TREE_DEPTH - 1;

    bytes32[DEPOSIT_CONTRACT_TREE_DEPTH] branch;
    uint256 deposit_count;

    bytes32[DEPOSIT_CONTRACT_TREE_DEPTH] zero_hashes;

    constructor() public {
        // Compute hashes in empt
        for (uint height = 0; hei                     ght++)
            zero_hashes[height +                      eight], zero_hashes
            [height]));
    }

    function get_deposit_root() override external view returns (bytes32) {
        bytes32 node;
        uint size = deposit_count;
        for (uint height = 0; height < DEPOSIT_CONTRACT_TREE_DEPTH; height++) {
            if ((size & 1) == 1)
                node = sha256(abi.encodePacked(branch[height], node));
            else
                node = sha256(abi.encodePacked(node, zero_hashes[height]));
            size /= 2;
        }
        return sha256(abi.encodePacked(
```

???

sourcify.dev

# What is source code verification?

- Ethereum contracts are stored on the blockchain in **bytecode**
- Humans can’t read/write bytecode
- Humans read/write in prog. languages (Solidity, Vyper…)

# What is source code verification?

- Ethereum contracts are stored on the blockchain in **bytecode**
- Humans can’t read/write bytecode
- Humans read/write in prog. languages (Solidity, Vyper...)
- Source code → Bytecode

## Source Code Verification

▯ **MyContract.sol**

▯ Ownable.sol

▯ ERC20.sol

...

## Source Code Verification

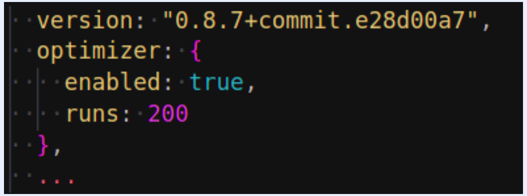
▯ MyContract.sol

▯ Ownable.sol

▯ ERC20.sol

...

Compilation Settings



```
  version: "0.8.7+commit.e28d00a7",
  optimizer: {
    enabled: true,
    runs: 200
  },
  ...
```

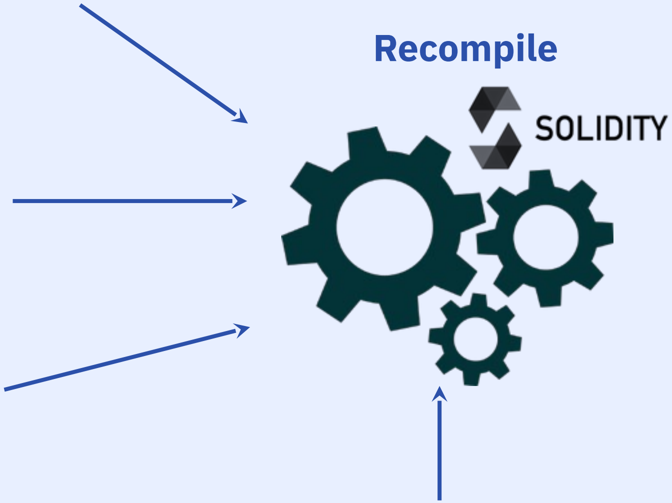
## Source Code Verification

▯ MyContract.sol

▯ Ownable.sol

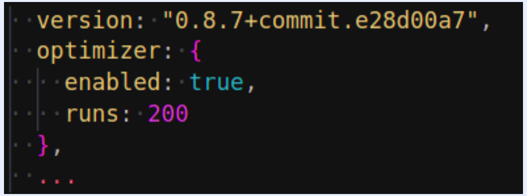
▯ ERC20.sol

...



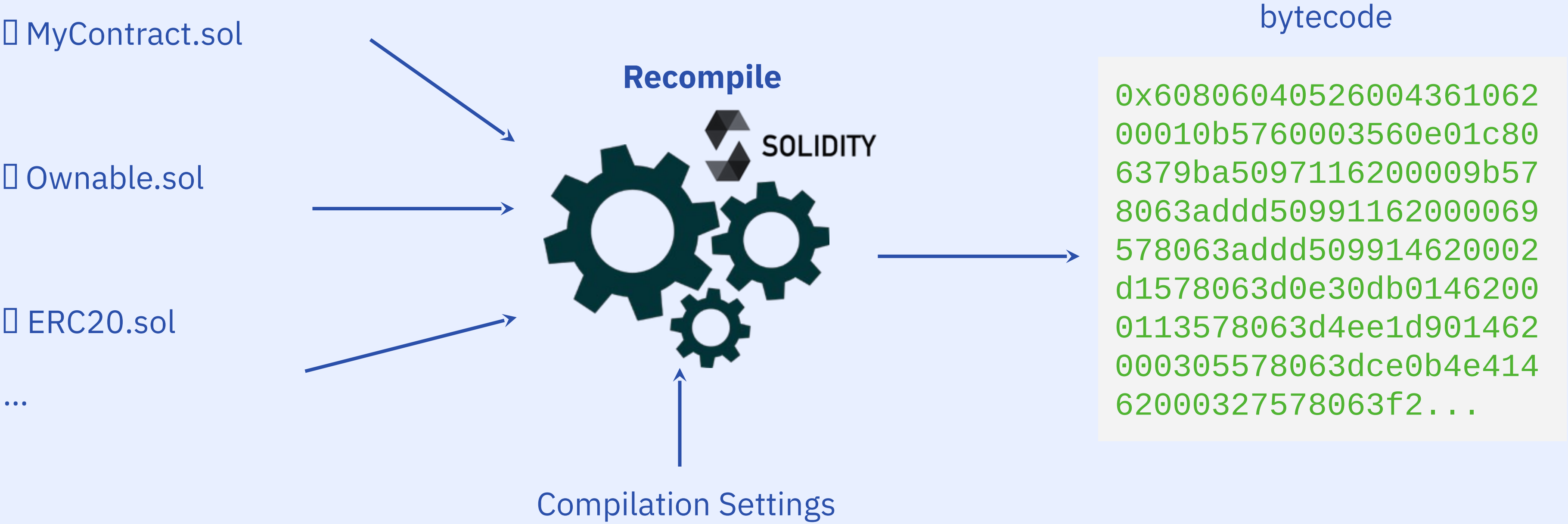
Recompile

Compilation Settings



```
  version: "0.8.7+commit.e28d00a7",
  optimizer: {
    enabled: true,
    runs: 200
  },
  ...
```

## Source Code Verification



▯ MyContract.sol

▯ Ownable.sol

▯ ERC20.sol

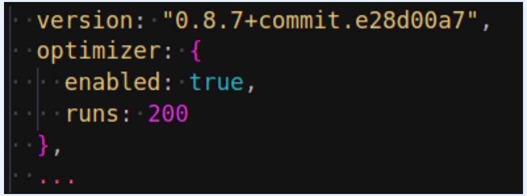
...

**Recompile**

bytecode

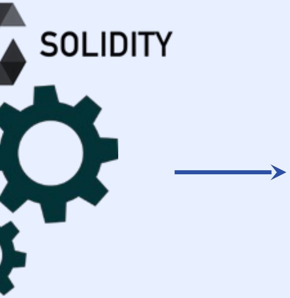
```
0x60806040526004361062
00010b5760003560e01c80
6379ba5097116200009b57
8063addd50991162000069
578063addd509914620002
d1578063d0e30db0146200
0113578063d4ee1d901462
000305578063dce0b4e414
62000327578063f2...
```

Compilation Settings



```
  version: "0.8.7+commit.e28d00a7",
  optimizer: {
    enabled: true,
    runs: 200
  },
  ...
```

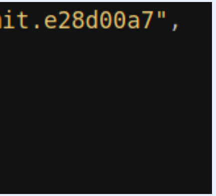
# Source Code Verification



bytecode

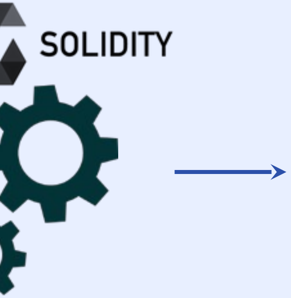
```
0x6080604052600436106200
00010b5760003560e01c80
6379ba5097116200009b57
8063addd509911620000069
578063addd50991462000002
d1578063d0e30db0146200
0113578063d4ee1d901462
000305578063dce0b4e414
620003275780063f2...
```

Settings



```
it.e28d00a7",
```

## Source Code Verification

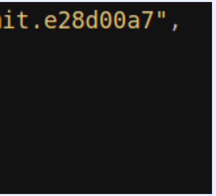


bytecode

```
0x60806040526004361062
00010b5760003560e01c80
6379ba509711620009b57
8063addd50991162000069
578063addd509914620002
d1578063d0e30db0146200
0113578063d4ee1d901462
000305578063dce0b4e414
62000327578063f2...
```

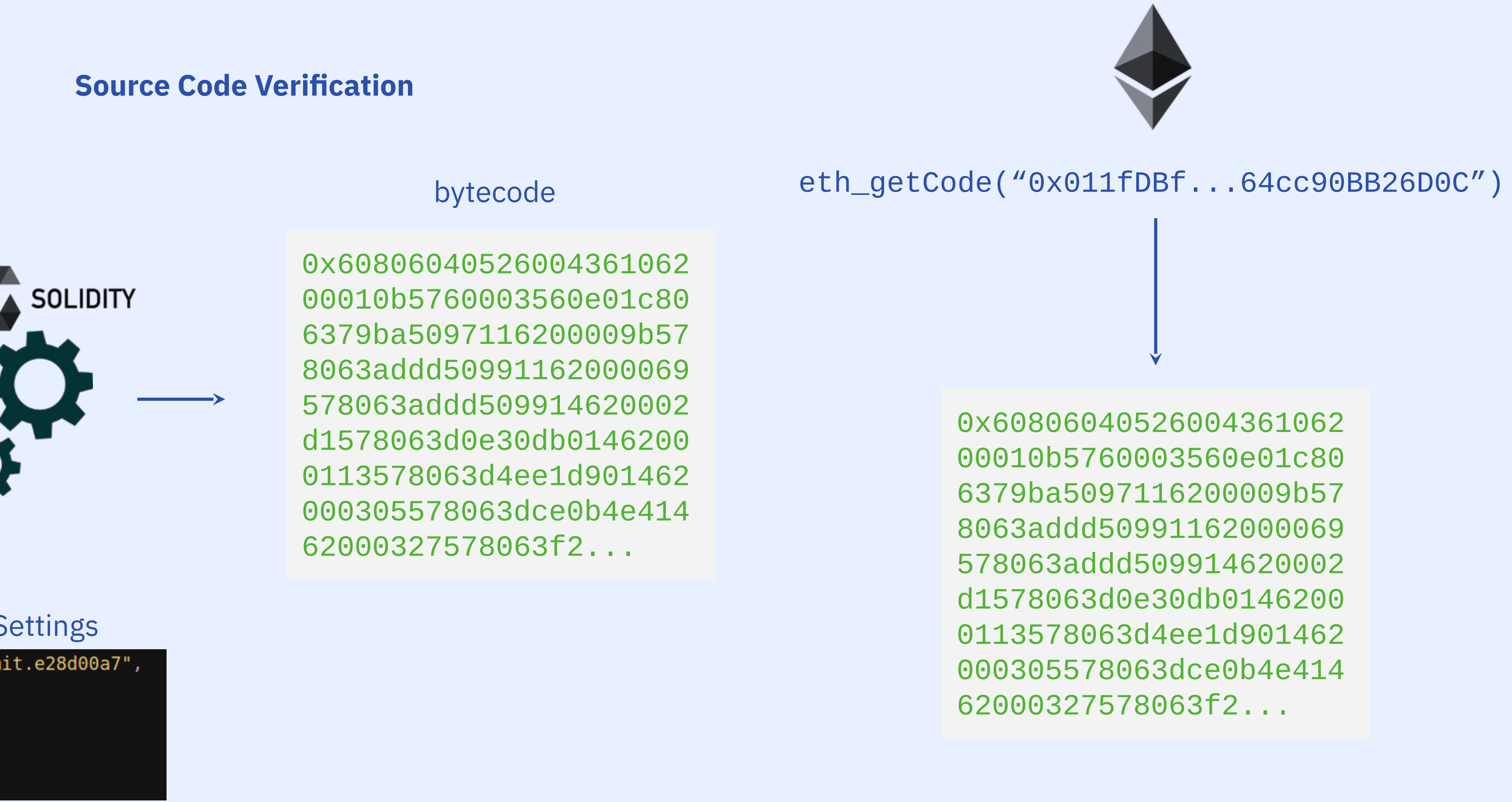
`eth_getCode(“0x011fDBf...64cc90BB26D0C”)`

Settings



```
it.e28d00a7",
```

## Source Code Verification



bytecode

```
0x608060405260043610620
00010b5760003560e01c80
6379ba50971162000 9b57
```

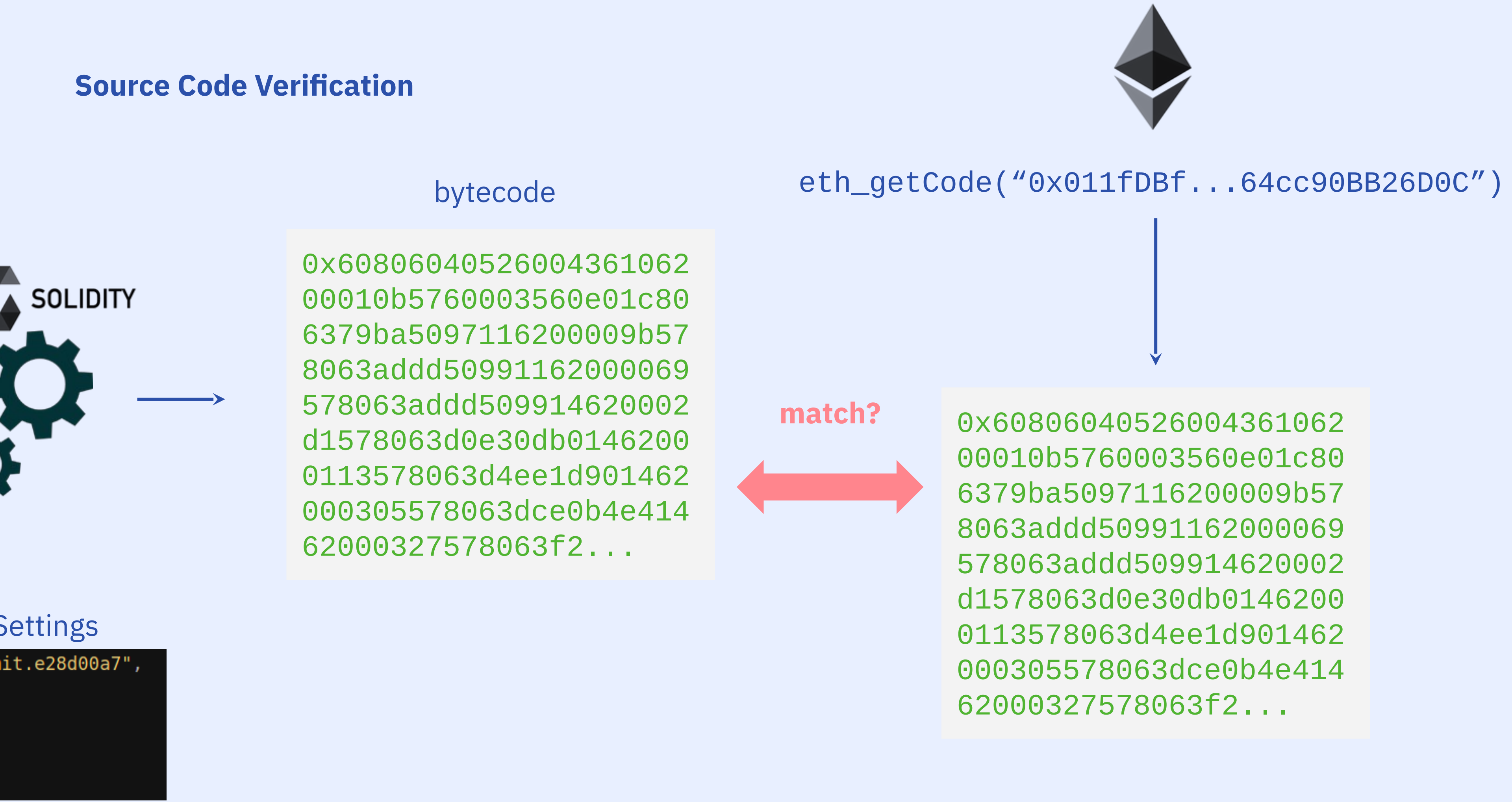
eth_getCode(“0x011fDBf...64cc90BB26D0C”)

```
0x6080604052600436106200
00010b5760003560e01c80
6379ba5097116200009b57
8063addd50991162000069
578063addd509914620002
d1578063d0e30db0146200
0113578063d4ee1d901462
000305578063dce0b4e414
62000327578063f2...
```

Settings

```
it.e28d00a7",
```

## Source Code Verification



bytecode

```
0x6080604052600436106200
010b5760003560e01c80
6379ba509711620009b57
8063addd5099116200006957
8063addd509914620002
d1578063d0e30db0146200
0113578063d4ee1d901462
000305578063dce0b4e414
6200032757806 3f2...
```

match?

eth_getCode("0x011fDBf...64cc90BB26D0C")

```
0x6080604052600436106200
010b5760003560e01c80
6379ba509711620009b57
8063addd5099116200006957
8063addd509914620002
d1578063d0e30db0146200
0113578063d4ee1d901462
000305578063dce0b4e414
6200032757806 3f2...
```

ettings

```
it.e28d00a7",
```

# Ethereum has good verification culture

- Community demands contracts to be verified

# Ethereum has good verification culture

- Community demands contracts to be verified
- **DO NOT INTERACT WITH UNVERIFIED CONTRACTS**
    - It’s basically arbitrary black box code