

# Human-friendly contract interactions with



sourcify verification

---

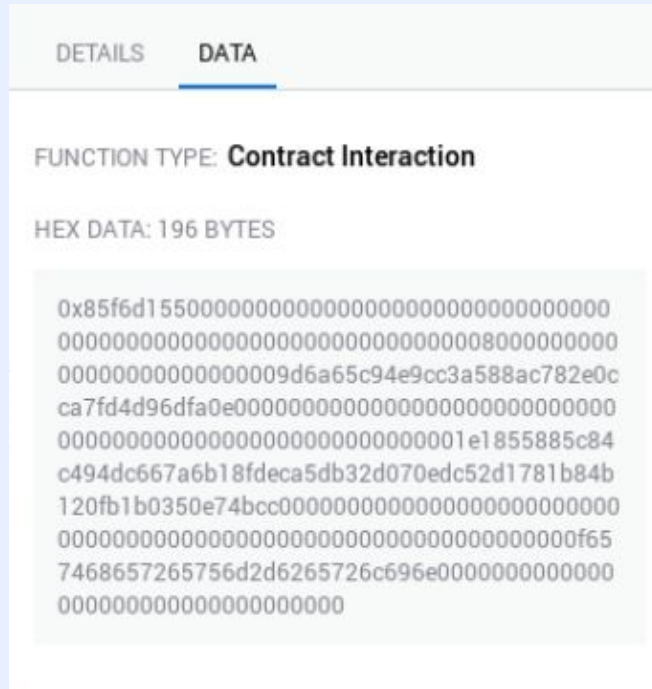
**Kaan Uzdoğan**  
Ethereum Foundation

Solidity Summit  
20.04.2022 - Devconnect Amsterdam

# **Source code verification matters**

**Source code verification matters  
for end users too**

## Typical Interaction



...aka YOLO signing 🏴‍☠️



**How to achieve this?**

# As a contract developer:

- 1) **Use NatSpec documentation**
- 2) Publish metadata and source code on IPFS
- 3) Source code verification on Sourcify

# Natspec

```
contract Wallet {  
...  
    /// @dev Allows to swap/replace an owner from the Safe with another address.  
    ///      This can only be done via a Safe transaction.  
    /// @notice Replaces the owner `oldOwner` in the Safe with `newOwner`.  
    /// @param prevOwner Owner that pointed to the owner to be replaced in the linked list  
    /// @param oldOwner Owner address to be replaced.  
    /// @param newOwner New owner address.  
    function swapOwner(  
        address prevOwner,  
        address oldOwner,  
        address newOwner  
    ) public authorized {...
```



# Natspec

```
contract Wallet {
```

```
... *
```

## Developer Documentation

```
/// @dev Allows to swap/replace an owner from the Safe with another address.
```

```
///      This can only be done via a Safe transaction.
```

```
/// @notice Replaces the owner `oldOwner` in the Safe with `newOwner`.
```

```
/// @param prevOwner Owner that pointed to the owner to be replaced in the linked list
```

```
/// @param oldOwner Owner address to be replaced.
```

```
/// @param newOwner New owner address.
```

```
function swapOwner(
```

```
    address prevOwner,
```

```
    address oldOwner,
```

```
    address newOwner
```

```
) public authorized {...
```

# Natspec

```
contract Wallet {  
    ...  
  
    /// @dev Allows to swap/replace an owner from the Safe with another address.  
    ///      This can only be done via a Safe transaction.  
    /// @notice Replaces the owner `oldOwner` in the Safe with `newOwner`.  
    /// @param prevOwner Owner that pointed to the owner to be replaced in the linked list  
    /// @param oldOwner Owner address to be replaced.  
    /// @param newOwner New owner address.  
  
    function swapOwner(  
        address prevOwner,  
        address oldOwner,  
        address newOwner  
    ) public authorized {...
```

[User Documentation](#)

# Natspec

```
contract Wallet {  
...  
  
    /// @dev Allows to swap/replace an owner from the Safe with another address.  
    ///      This can only be done via a Safe transaction.  
    /// @notice Replaces the owner `oldOwner` in the Safe with `newOwner`.
```

## Parameters

```
    /// @param prevOwner Owner that pointed to the owner to be replaced in the linked list  
    /// @param oldOwner Owner address to be replaced.  
    /// @param newOwner New owner address.
```

```
function swapOwner(  
    address prevOwner,  
    address oldOwner,  
    address newOwner  
) public authorized {...
```

# Natspec: Dynamic Expressions

`@notice Replaces the owner `oldOwner` in the Safe with `newOwner``

# Natspec: Dynamic Expressions

@notice Replaces the owner ``oldOwner`` in the Safe with ``newOwner``

# Natspec: Dynamic Expressions

@notice Replaces the owner ``oldOwner`` in the Safe with ``newOwner``

## Becomes:

Replaces the owner `0xC60F45e0507032036033b361d3a6457b9F0283D`

in the Safe with `0x83D0360050703233b361d3a6457b9F2cC60F45e0`

# Natspec: Dynamic Expressions

`@notice Replaces the owner `oldOwner` in the Safe with `newOwner``

## Even better:

Replaces the owner **wallet1.eth** in the Safe with **wallet2.eth**

**Where to find the documentation?**



## In Solidity **Contract Metadata:**

<https://docs.soliditylang.org/en/latest/metadata.html>

# Solidity Contract Metadata

JSON file generated by the Solidity compiler which contains... metadata:

- ABI
- Userdoc + devdoc
- Compilation info
- Source file info

```
▼ {
  ▼ "compiler": {
    "version": "0.7.6+commit.7338295f"
  },
  "language": "Solidity",
  ▼ "output": {
    ► "abi": [...], // 49 items
    ► "devdoc": {...}, // 5 items
    ► "userdoc": {...} // 3 items
  },
  ▼ "settings": {
    ► "compilationTarget": {...}, // 1 item
    "evmVersion": "istanbul",
    "libraries": {},
    ► "metadata": {...}, // 2 items
    ▼ "optimizer": {
      "enabled": false,
      "runs": 200
    },
    "remappings": []
  },
  ▼ "sources": {
    ► "contracts/GnosisSafe.sol": {...}, // 3 items
    ► "contracts/base/Executor.sol": {...}, // 3 items
    ► "contracts/base/FallbackManager.sol": {...}, // 3 items
    ► "contracts/base/GuardManager.sol": {...}, // 3 items
    ► "contracts/base/ModuleManager.sol": {...}, // 3 items
```

```
▼ "userdoc": {  
  "kind": "user",  
  ▼ "methods": {  
    ▼ "addOwnerWithThreshold(address,uint256)": {  
      "notice": "Adds the owner `owner` to the Safe and updates the threshold to `_threshold`."  
    },  
    ▼ "changeThreshold(uint256)": {  
      "notice": "Changes the threshold of the Safe to `_threshold`."  
    },  
    ▼ "disableModule(address,address)": {  
      "notice": "Disables the module `module` for the Safe."  
    },  
    ▼ "enableModule(address)": {  
      "notice": "Enables the module `module` for the Safe."  
    },  
    ▼ "removeOwner(address,address,uint256)": {  
      "notice": "Removes the owner `owner` from the Safe and updates the threshold to `_threshold`."  
    },  
    ▼ "requiredTxGas(address,uint256,bytes,uint8)": {  
      "notice": "Deprecated in favor of common/StorageAccessible.sol and will be removed in next version."  
    },  
    ▼ "swapOwner(address,address,address)": {  
      "notice": "Replaces the owner `oldOwner` in the Safe with `newOwner`."  
    }  
  },  
},
```

## As a contract developer:

- 1) Use NatSpec documentation
- 2) Publish metadata and source code on IPFS**
- 3) Source code verification on Sourcify

e.g.  
when compiling  
with Remix

The image shows the Remix IDE interface. On the left is a sidebar with icons for Explorer, Search, Run and Debug, and Deploy. The main area is divided into three panels. The top-left panel is the 'SOLIDITY COMPILER' settings, which includes a 'COMPILER' dropdown set to '0.8.13+commit.abaa5c0e', an unchecked 'Include nightly builds' checkbox, a 'LANGUAGE' dropdown set to 'Solidity', an 'EVM VERSION' dropdown set to 'default', and a 'COMPILER CONFIGURATION' section with unchecked checkboxes for 'Auto compile', 'Enable optimization' (set to 200), and 'Hide warnings'. A large blue button 'Compile 1\_Storage.sol' is present, along with a grey button 'Compile and Run script'. The bottom-left panel shows the 'CONTRACT' dropdown set to 'Storage (1\_Storage.sol)' and a 'Publish on Ipfs' button. The right panel is a code editor showing Solidity code for a storage contract. The code includes a SPDX license header, a pragma statement for Solidity 0.8.13, a contract named 'Storage' with a 'uint256' state variable 'num1', and functions 'setNum1' and 'getNum1'. The bottom-right panel shows a terminal with the message 'Welcome to Remix 0.8.13'.

**SOLIDITY COMPILER**

COMPILER

0.8.13+commit.abaa5c0e

☐ Include nightly builds

LANGUAGE

Solidity

EVM VERSION

default

COMPILER CONFIGURATION

☐ Auto compile

☐ Enable optimization 200

☐ Hide warnings

Compile 1\_Storage.sol

Compile and Run script

CONTRACT

Storage (1\_Storage.sol)

Publish on Ipfs


```
1 // SPDX-License-Identifier: MIT
2
3 pragma solidity ^0.8.13;
4
5 /**
6  * @title Storage
7  * @dev Stores a value in state and returns it
8  * @custom:oz-upgrades-unsafe-allow constructor
9  */
10 contract Storage {
11
12     uint256 num1;
13
14     /**
15      * @dev Sets the value of num1
16      * @param num1 the value to set
17      */
18     function setNum1(uint256 num1) public {
19         num1 = num1;
20     }
21
22     /**
23      * @dev Returns the value of num1
24      * @return the value of num1
25      */
26     function getNum1() public view returns (uint256) {
27         return num1;
28     }
29 }
```

0 ☐ listen on all





Welcome to Remix 0.8.13


Your files are stored in the browser's local storage.

You can use this to store your data.



# SOLIDITY COMPILER



COMPILER 

0.8.13+commit.abaa5c0e

☐ Include nightly builds

LANGUAGE

Solidity

EVM VERSION

default



COMPILER CONFIGURATION

☐ Auto compile

☐ Enable optimization 200



☐ Hide warnings

Compile 1\_Storage.sol

Compile and Run script  



CONTRACT

Storage (1\_Storage.sol)

 Publish on Ipfs 






Home

```
1 // SPDX-License-Identifier: MIT
2
3 pragma solidity ^0.8.13;
4
5 /**
6  * @title Storage
7  * @dev Stores and returns a value
8  * @custom:example
9  */
10 contract Storage {
11
12     uint256 value;
13
14     /**
15      * @dev Sets the value
16      * @param value the value to set
17      */
18     function set(uint256 value) public {
19         numl
20     }
21
22     /**
23      * @dev Returns the value
24      * @return the value
25      */
26     function get() public view returns (uint256) {
27         retu
28     }
29 }
```

  0 ☐ listen on all

Welcome to Remix 0.8.13  
Your files are stored on IPFS  
You can use this tool to interact with your contracts

or  
when deploying



## DEPLOY & RUN TRANSACTIONS

ENVIRONMENT

JavaScript VM (London)

ACCOUNT

0x5B3...eddC4 (100 ether)

GAS LIMIT

3000000

VALUE

0

Wei

CONTRACT

Storage - contracts/1\_Storage.sol

Deploy

☒ Publish to IPFS

OR

At Address

Load contract from Address



Transactions recorded 0

Deployed Contracts

Currently you have no contract

Home 1\_5

```
1 // SPDX-Licens
2
3 pragma solidity
4
5 /**
6  * @title Stor
7  * @dev Store
8  * @custom:dev
9  */
10 contract Stora
11
12     uint256 nu
13
14     /**
15      * @dev St
16      * @param
17      */
18     function s
19         number
20     }
21
22     /**
23      * @dev Re
24      * @return
25      */
26     function r
27         return
28     }
29 }
```

  0 ☐ listen on all tra






Welcome to Remix 0.23

Your files are stored :

You can use this termi

>





DEPLOY & RUN TRANSACTIONS

ENVIRONMENT  
JavaScript VM (London)

ACCOUNT  
0x5B3...eddC4 (100 ether)

GAS LIMIT  
3000000

VALUE  
0 Wei

CONTRACT  
Storage - contracts/1\_Storage.sol

Deploy

☒ Publish to IPFS

OR

At Address Load contract from Address

Transactions recorded 0

Deployed Contracts

Currently you have no contract

Home 1\_5

```
1 // SPDX-Licens
2
3 pragma solidity
4
5 /**
6  * @title Stor
7  * @dev Store
8  * @custom:dev
9  */
10 contract Stora
11
12     uint256 nu
13
14     /**
15      * @dev St
16      * @param
17      */
18     function s
19         number
20     }
21
22     /**
23      * @dev Re
24      * @return
25      */
26     function r
27         return
28     }
29 }
```

0 ☐ listen on all tra

Welcome to Remix 0.23

Your files are stored :

You can use this termi

>

## As a contract developer:

- 1) Use NatSpec documentation
- 2) Publish metadata and source code on IPFS
- 3) **Source code verification on Sourcify**

# Source Code Verification



**MyContract.sol**



Ownable.sol



ERC20.sol

...

# Source Code Verification

 MyContract.sol

 Ownable.sol

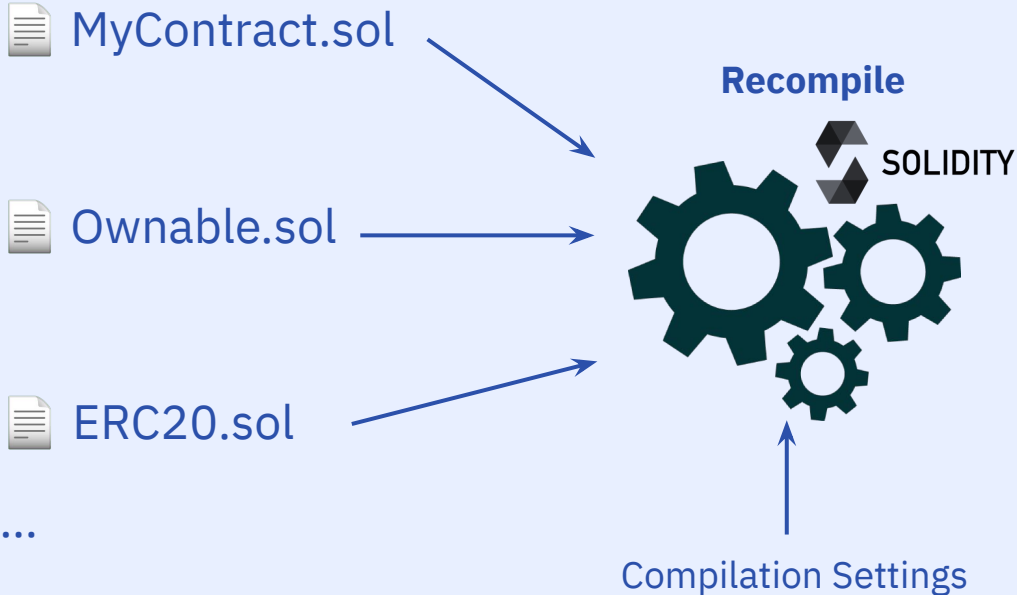
 ERC20.sol

...

## Compilation Settings

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

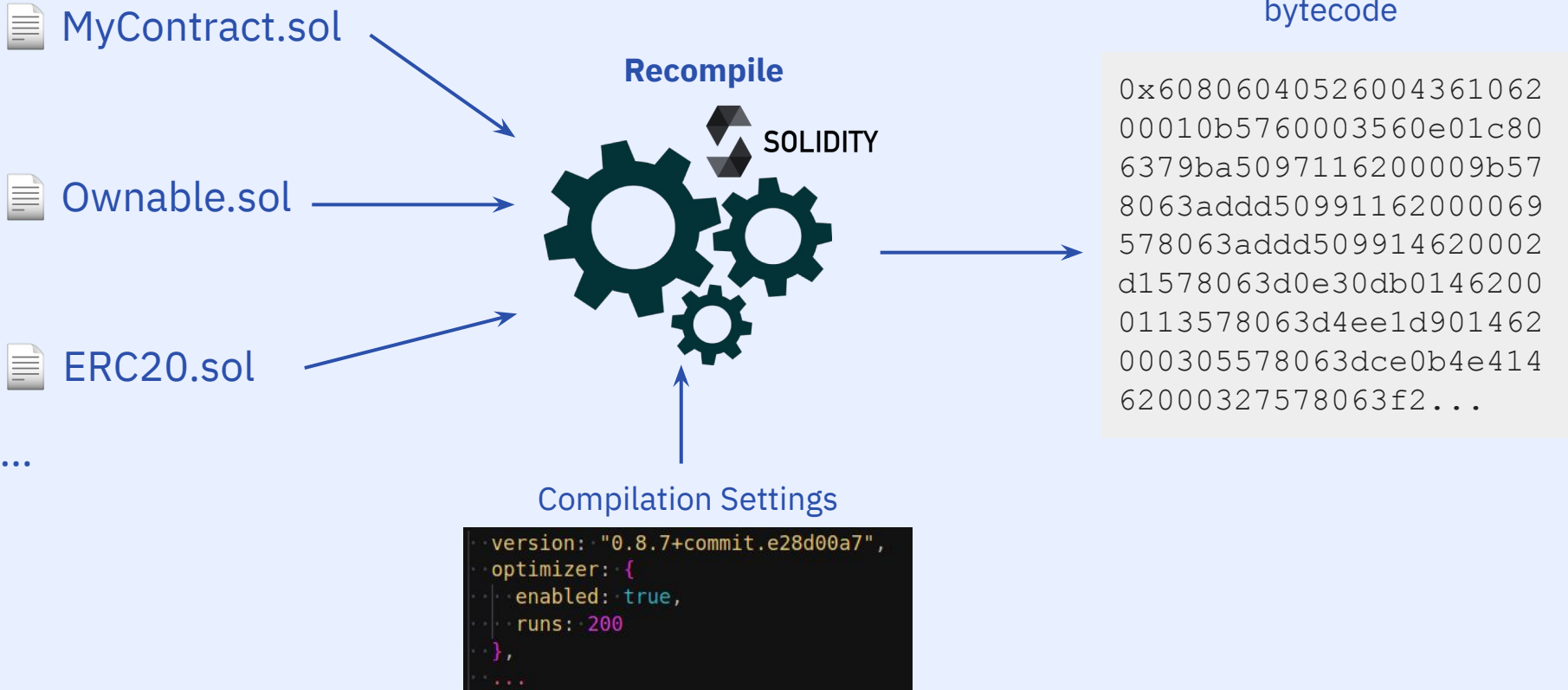
# Source Code Verification



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

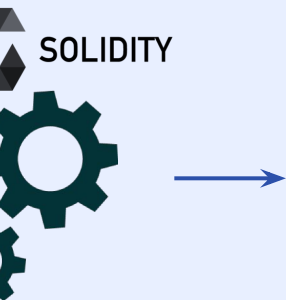
```

# Source Code Verification



# Source Code Verification

bytecode



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

Settings

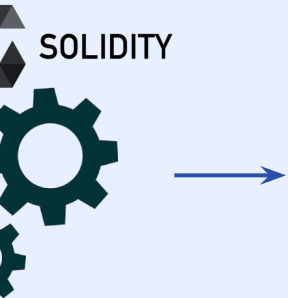
```
...it.e28d00a7",
```

# Source Code Verification



bytecode

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



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

Settings


```
...it.e28d00a7",
```



# Source Code Verification



bytecode



0x60806040526004361062  
00010b5760003560e01c80  
6379ba5097116200009b57  
8063add50991162000069  
578063add509914620002  
d1578063d0e30db0146200  
0113578063d4ee1d901462  
000305578063dce0b4e414  
62000327578063f2...

eth\_getCode("0x011fDBf...64cc90BB26D0C")



0x60806040526004361062  
00010b5760003560e01c80  
6379ba5097116200009b57  
8063add50991162000069  
578063add509914620002  
d1578063d0e30db0146200  
0113578063d4ee1d901462  
000305578063dce0b4e414  
62000327578063f2...

Settings

...it.e28d00a7",

# Source Code Verification



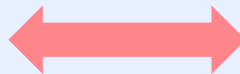
bytecode

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



```
0x60806040526004361062
00010b5760003560e01c80
6379ba5097116200009b57
8063add50991162000069
578063add509914620002
d1578063d0e30db0146200
0113578063d4ee1d901462
000305578063dce0b4e414
62000327578063f2...
```

match?





```
0x60806040526004361062
00010b5760003560e01c80
6379ba5097116200009b57
8063add50991162000069
578063add509914620002
d1578063d0e30db0146200
0113578063d4ee1d901462
000305578063dce0b4e414
62000327578063f2...
```

Settings

...it.e28d00a7",

# Source Code Verification

 **Partial match** = bytecodes match

 **Full / Perfect Match** = bytecode + metadata match

# Source Code Verification

😞 **Partial match** = bytecodes match

😊 **Full / Perfect Match** = bytecode + metadata match

Full matches cryptographically guarantee the source code and metadata is exactly the same as when deployed - even comments, variable names etc.

**How?**

# Full/Perfect Verification: How?

## Contract Bytecode

```
919050565b600082610b9a57634e487b7160e01b81526012600452602481fd5b500490565b80825b6001808611610bb157
50610bdc565b818704821115610bc357610bc3610cd5565b80861615610bd057918102915b9490941c938002610ba2565b
94509492505050565b600061046860001960ff851684600082610c0157506001610468565b81610c0e5750600061046856
5b8160018114610c245760028114610c2e57610c5b565b6001915050610468565b60ff841115610c3f57610c3f610cd556
5b6001841b915084821115610c5557610c55610cd5565b50610468565b5060208310610133831016604e8410600b841016
1715610c8e575081810a83811115610c8957610c89610cd5565b610468565b610c9b8484846001610b9f565b8086048211
15610cad57610cad610cd5565b02949350505050565b6000816000190483118215151615610cd057610cd0610cd5565b50
0290565b634e487b7160e01b600052601160045260246000fdfea264697066735822122078b530288f1cffe879bb7d9062
e904deb3aa9b9c8d27ea4ecafa987583c18a6e64736f6c63430008010033
```

# Full/Perfect Verification: How?

Solidity compiler appends extra data at the end of the bytecode in CBOR encoding, including the **hash of the metadata**, which enables full matches.

## Contract Bytecode

```
919050565b600082610b9a57634e487b7160e01b81526012600452602481fd5b500490565b80825b6001808611610bb157
50610bdc565b818704821115610bc357610bc3610cd5565b80861615610bd057918102915b9490941c938002610ba2565b
94509492505050565b600061046860001960ff851684600082610c0157506001610468565b81610c0e5750600061046856
5b8160018114610c245760028114610c2e57610c5b565b6001915050610468565b60ff841115610c3f57610c3f610cd556
5b6001841b915084821115610c5557610c55610cd5565b50610468565b5060208310610133831016604e8410600b841016
1715610c8e575081810a83811115610c8957610c89610cd5565b610468565b610c9b8484846001610b9f565b8086048211
15610cad57610cad610cd5565b02949350505050565b6000816000190483118215151615610cd057610cd0610cd5565b50
0290565b634e487b7160e01b600052601160045260246000fdfea264697066735822122078b530288f1cffe879bb7d9062
e904deb3aa9b9c8d27ea4ecafa987583c18a6e64736f6c63430008010033
```

# Full/Perfect Verification: How?

 MyContract.sol

 Ownable.sol

 ERC20.sol

...

# Full/Perfect Verification: How?

 MyContract.sol  $\xrightarrow{\text{Hashed}}$  0xb6ee9d...

 Ownable.sol  $\xrightarrow{\text{Hashed}}$  0x41e281...

 ERC20.sol  $\xrightarrow{\text{Hashed}}$  0x9fd73f...

...



# Full/Perfect Verification: How?

 MyContract.sol → Hashed 0xb6ee9d...

 Ownable.sol → Hashed 0x41e281...

 ERC20.sol → Hashed 0x9fd73f...

...

## Metadata

```
{
  ▶ "compiler": { ... }, // 1 item
  "language": "Solidity",
  ▶ "output": { ... }, // 3 items
  ▶ "settings": { ... }, // 6 items
  ▼ "sources": {
    ▼ "contracts/MyContract.sol": {
      "keccak256": "0xb6ee9d528b336942dd70d3b41e2811...",
      "license": "GPL-3.0",
      ▼ "urls": [
        "bzz-raw://fe52c6e3c04ba5d83ede6cc1a43c45fa...",
        "dweb:/ipfs/QmawU3NM1WNWkBauRudYCiFvuFE1tTL..."
      ]
    },
    ▶ "contracts/Ownable.sol": { ... }, // 3 items
    ▶ "contracts/ERC20.sol": { ... } // 3 items
  }
}
```

# Full/Perfect Verification: How?

MyContract.sol Hashed → 0xb6ee9d...

Ownable.sol Hashed → 0x41e281...

ERC20.sol Hashed → 0x9fd73f...

...

## Metadata

```
{
  ▶ "compiler": { ... }, // 1 item
  "language": "Solidity",
  ▶ "output": { ... }, // 3 items
  ▶ "settings": { ... }, // 6 items
  ▼ "sources": {
    ▼ "contracts/MyContract.sol": {
      "keccak256": "0xb6ee9d528b336942dd70d3b41e2811...",
      "license": "GPL-3.0",
      ▼ "urls": [
        "bzz-raw://fe52c6e3c04ba5d83ede6cc1a43c45fa...",
        "dweb:/ipfs/QmawU3NM1WNWkBauRudYCiFvuFE1tTL...",
      ]
    },
    ▶ "contracts/Ownable.sol": { ... }, // 3 items
    ▶ "contracts/ERC20.sol": { ... } // 3 items
  }
}
```

# Full/Perfect Verification: How?

MyContract.sol  $\xrightarrow{\text{Hashed}}$  0xb6ee9d...

Ownable.sol  $\xrightarrow{\text{Hashed}}$  0x41e281...

ERC20.sol  $\xrightarrow{\text{Hashed}}$  0x9fd73f...

...

## Metadata

```
{
  "compiler": { ... }, // 1 item
  "language": "Solidity",
  "output": { ... }, // 3 items
  "settings": { ... }, // 6 items
  "sources": {
    "contracts/MyContract.sol": {
      "keccak256": "0xb6ee9d528b336942dd70d3b41e2811...",
      "license": "GPL-3.0",
      "urls": [
        "bzz-raw://fe52c6e3c04ba5d83ede6cc1a43c45fa...",
        "dweb:/ipfs/QmawU3NM1WNWkBauRudYCiFvuFE1tTL..."
      ]
    },
    "contracts/Ownable.sol": { ... }, // 3 items
    "contracts/ERC20.sol": { ... } // 3 items
  }
}
```

```
6942dd70d3b41e2811be10a473776352009fd73f85604f5ed206" ,
```

```
5d83ede6cc1a43c45fa43caa435b207f64707afb17d3af1bcf1" ,  
kBauRudYCiFvuFE1tTLHB98akyBvb9UwWA"
```

```
// 3 items
```

```
3 items
```

## Metadata

```
6942dd70d3b41e2811be10a473776352009fd73f85604f5ed206",
```

```
5d83ede6cc1a43c45fa43caa435b207f64707afb17d3af1bcf1",  
kBauRudYCiFvuFE1tTLHB98akyBvb9UWwA"
```

```
// 3 items
```

```
3 items
```

**Metadata**

**IPFS hash**

QmWTqspM5B1quNvdhXbS6TbXzyLZ5cUGHnTV8ZWJPqrQqj



QmWTqspM5B1quNvdhXbS6TbXzyLZ5cUGHnTV8ZWJPqrQqj



QmWTqspM5B1quNvdhXbS6TbXzyLZ5cUGHnTV8ZWJPqrQqj

encoded



### Contract Bytecode

```
919050565b600082610b9a57634e487b7160e01b81526012600452602481fd5b500490565b80825b6001808611610bb157
50610bdc565b818704821115610bc357610bc3610cd5565b80861615610bd057918102915b9490941c938002610ba2565b
94509492505050565b600061046860001960ff851684600082610c0157506001610468565b81610c0e5750600061046856
5b8160018114610c245760028114610c2e57610c5b565b6001915050610468565b60ff841115610c3f57610c3f610cd556
5b6001841b915084821115610c5557610c55610cd5565b50610468565b5060208310610133831016604e8410600b841016
1715610c8e575081810a83811115610c8957610c89610cd5565b610468565b610c9b8484846001610b9f565b8086048211
15610cad57610cad610cd5565b029493505050565b6000816000710483118215151615610cd057610cd0610cd5565b50
0290565b634e487b7160e01b600052601160045260246000fdfea264697066735822122078b530288f1cffe879bb7d9062
e904deb3aa9b9c8d27ea4ecafa987583c18a6e64736f6c63430008010033
```

# Full/Perfect Verification: How?

MyContract.sol → Hashed → 0xb6ee9d...

Ownable.sol → Hashed → 0x41e281...

ERC20.sol → Hashed → 0x9fd73f...

...

## Metadata

```
{
  "compiler": { ... }, // 1 item
  "language": "Solidity",
  "output": { ... }, // 3 items
  "settings": { ... }, // 6 items
  "sources": {
    "contracts/MyContract.sol": {
      "keccak256": "0xb6ee9d528b336942dd70d3b41...",
      "license": "GPL-3.0",
      "urls": [
        "bzz-raw://fe52c6e3c04ba5d83ede6cc1a43...",
        "dweb:/ipfs/QmawU3NM1WNWkBauRudYCiFvuF..."
      ]
    },
    "contracts/Ownable.sol": { ... }, // 3 items
    "contracts/ERC20.sol": { ... } // 3 items
  }
}
```



# Full/Perfect Verification: How?

MyContract-**diff**.sol Hashed → 0xb6ee9d...

Ownable.sol Hashed → 0x41e281...

ERC20.sol Hashed → 0x9fd73f...

...

## Metadata

```
{
  "compiler": { ... }, // 1 item
  "language": "Solidity",
  "output": { ... }, // 3 items
  "settings": { ... }, // 6 items
  "sources": {
    "contracts/MyContract.sol": {
      "keccak256": "0xb6ee9d528b336942dd70d3b41...",
      "license": "GPL-3.0",
      "urls": [
        "bzz-raw://fe52c6e3c04ba5d83ede6cc1a43...",
        "dweb:/ipfs/QmawU3NM1WNWkBauRudYCiFvuF..."
      ]
    },
    "contracts/Ownable.sol": { ... }, // 3 items
    "contracts/ERC20.sol": { ... } // 3 items
  ]
}
```

# Full/Perfect Verification: How?

MyContract-**diff**.sol Hashed → 0xa2fc16...

Ownable.sol Hashed → 0x41e281...

ERC20.sol Hashed → 0x9fd73f...

...

## Metadata

```
{
  "compiler": { ... }, // 1 item
  "language": "Solidity",
  "output": { ... }, // 3 items
  "settings": { ... }, // 6 items
  "sources": {
    "contracts/MyContract.sol": {
      "keccak256": "0xb6ee9d528b336942dd70d3b41...",
      "license": "GPL-3.0",
      "urls": [
        "bzz-raw://fe52c6e3c04ba5d83ede6cc1a43...",
        "dweb:/ipfs/QmawU3NM1WNWkBauRudYCiFvuF..."
      ]
    },
    "contracts/Ownable.sol": { ... }, // 3 items
    "contracts/ERC20.sol": { ... } // 3 items
  }
}
```

# Full/Perfect Verification: How?

MyContract-**diff**.sol Hashed → 0xa2fc16...

Ownable.sol Hashed → 0x41e281...

ERC20.sol Hashed → 0x9fd73f...

...

## Metadata

```
{
  "compiler": { ... }, // 1 item
  "language": "Solidity",
  "output": { ... }, // 3 items
  "settings": { ... }, // 6 items
  "sources": {
    "contracts/MyContract.sol": {
      "keccak256": "0xa2fc161e281128b336942dd7...",
      "license": "GPL-3.0",
      "urls": [
        "bzz-raw://d83e43c45fe52c6e3c04ba5e6...",
        "dweb:/ipfs/QmawU3NM1WNWkCiFvu8akyFE1..."
      ]
    },
    "contracts/Ownable.sol": { ... }, // 3 items
    "contracts/ERC20.sol": { ... } // 3 items
  }
}
```

```
b336942dd70e10a47304f5ed2776352009d3b4bfd73f85606",
```

```
e3c04ba5e6cc1ab207f64707afba43caa435d17d3af1bcf1",  
iFvu8akyFE1tTLBauRudYHB9Bvb9UWwA"
```

```
3 items  
items
```

**Metadata**

**IPFS hash**

QmWTqspM5B1quNvdhXbS6TbXzyLZ5cUGHnTV8ZWJPqrQqj

```
b336942dd70e10a47304f5ed2776352009d3b4bfd73f85606",
```


```
e3c04ba5e6cc1ab207f64707afba43caa435d17d3af1bcf1",  
iFvu8akyFE1tTLBauRudYHB9Bvb9UWwA"
```

```
3 items  
items
```

**Metadata**

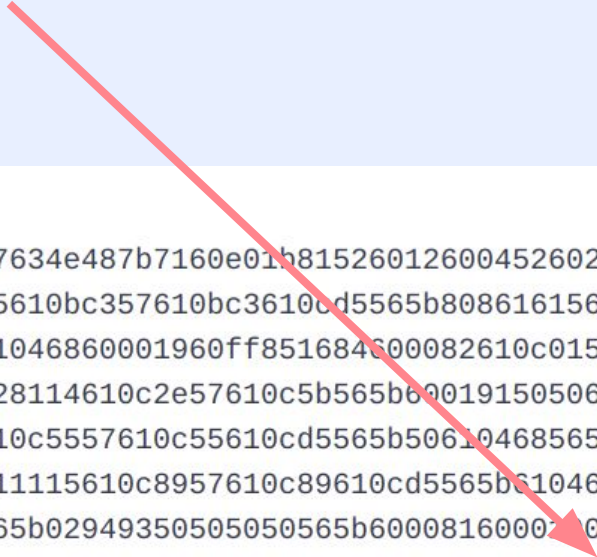
**IPFS hash**

**QmawU3NM1WNWkCiFvu8akyFE1tTLBauRudYHB9Bvb9UWwA**



QmawU3NM1WNWkCiFvu8akyFE1tTLBauRudYHB9Bvb9UWwA

### Contract Bytecode



```
919050565b600082610b9a57634e487b7160e01b81526012600452602481fd5b500490565b80825b6001808611610bb157
50610bdc565b818704821115610bc357610bc3610cd5565b80861615610bd057918102915b9490941c938002610ba2565b
94509492505050565b600061046860001960ff851684600082610c0157506001610468565b81610c0e5750600061046856
5b8160018114610c245760028114610c2e57610c5b565b6001915050610468565b60ff841115610c3f57610c3f610cd556
5b6001841b915084821115610c5557610c55610cd5565b50610468565b5060208310610133831016604e8410600b841016
1715610c8e575081810a83811115610c8957610c89610cd5565b610468565b610c9b8484846001610b9f565b8086048211
15610cad57610cad610cd5565b029493505050565b6000816000710483118215151615610cd057610cd0610cd5565b50
0290565b634e487b7160e01b600052601160045260246000fdfea264697066735822122078b530288f1cffe879bb7d9062
e904deb3aa9b9c8d27ea4ecafa987583c18a6e64736f6c63430008010033
```

QmawU3NM1WNWkCiFvu8akyFE1tTLBauRudYHB9Bvb9UWwA

### Contract Bytecode

919050565b600082610b9a57634e487b7160e01...81fd5b500490565b80825b6001808611610bb157  
50610bdc565b818704821115610bc357610bc3610c45565b60001015610bd057918102915b9490941c938002610ba2565b  
94509492505050565b600061046860001960ff851684600082610c0157506001610468565b81610c0e5750600061046856  
5b8160018114610c245760028114610c2e57610c5b565b6001915050610468565b60ff841115610c3f57610c3f610cd556  
5b6001841b915084821115610c5557610c55610cd5565b50610468565b5060208310610133831016604e8410600b841016  
1715610c8e575081810a83811115610c8957610c89610cd5565b50610468565b610c9b8484846001610b9f565b8086048211  
15610cad57610cad610cd5565b029493505050565b600081600070483118215151615610cd057610cd0610cd5565b50  
0290565b634e487b7160e01b600052601160045260246000fdfea264697066735822122078b530288f1cffe879bb7d9062  
e904deb3aa9b9c8d27ea4ecafa987583c18a6e64736f6c63430008010033

no full match



## Contract 0x5ed4a410A612F2fe625a8F3cB4d70f197fF8C8be on Ethereum Mainnet

[View on Sourcify](#)[View on Etherscan](#)

### Contract Bytecode

```
919050565b600082610b9a57634e487b7160e01b81526012600452602481fd5b500490565b80825b6001808611610bb157
50610bdc565b818704821115610bc357610bc3610cd5565b80861615610bd057918102915b9490941c938002610ba2565b
94509492505050565b600061046860001960ff851684600082610c0157506001610468565b81610c0e5750600061046856
5b81600181114610c2457600281114610c2e57610c5b565b6001915050610468565b60ff841115610c3f57610c3f610cd556
5b6001841b915084821115610c5557610c55610cd5565b50610468565b5060208310610133831016604e8410600b841016
1715610c8e575081810a83811115610c8957610c89610cd5565b610468565b610c9b8484846001610b9f565b8086048211
15610cad57610cad610cd5565b029493505050565b6000816000190483118215151615610cd057610cd0610cd5565b50
0290565b634e487b7160e01b600052601160045260246000fdfea264697066735822122078b530288f1cffe879bb7d9062
e904deb3aa9b9c8d27ea4ecafa987583c18a6e64736f6c63430008010033
```

### CBOR decoding

CBOR length: **51 Bytes** [See on CBOR Playground](#)

```
{
  "ipfs": "0x122078b530288f1cffe879bb7d9062e904deb3aa9b9c8d27ea4ecafa987583c18a6e",
  "solc": "0x000801"
}
```

### Solidity compiler version (decoded)

0.8.1

### Metadata Hash (decoded)

ipfs://QmWTqspM5B1quNvdhXbS6TbXzyLZ5cUGHnTV8ZWJPqrQqj

Try it out in **Playground**: [playground.sourcify.dev](https://playground.sourcify.dev)



**How to verify?**

# sourcify.dev UI

## Verifier

Verify smart contracts by recompiling with the Solidity source code and metadata.

Old verifier UI available at [legacy.sourcify.dev](https://legacy.sourcify.dev)

### File Add Zone

Add the Solidity source files and metadata of all contracts you want to verify.

Import from remote file or zip

<https://github.com/Uniswap/v3-core/archive/refs/heads/main.zip>

CLEAR FILES

#### Added Files

metadata.json  
1\_Storage.sol

### Contracts

COLLAPSE ALL

#### Storage

Chain & Address Missing



Please provide contract address and network

Address

0x00878Ac0D6B8d981ae72BA7cDC967eA0Fae69df4

Network

Ethereum Testnet Görli (5)



Verify

[Detailed View](#)

Once a contract is verified it can't be removed from the Sourcify repository.

# API

## Attempt contract verification

Sends provided files for verification.

**URL :** `/verify` or `/`

**Method :** `POST`

```
{
  "address": ...,
  "chain": ...,
  "files": {
    "file-name1.sol": ...,
    "file-name2.sol": ...
  }
}
```

# API

- Check by addresses (full match) : `GET /check-by-addresses?addresses={addresses}&chainIds={chainIds}`
- Check by addresses (full or partial match) : `GET /check-by-all-addresses?addresses={addresses}&chainIds={chainIds}`
- Get file tree (full match) : `GET /files/tree/:chain/:address`
- Get file tree (full or partial match) : `GET /files/tree/any/:chain/:address`
- Get source files (full match) : `GET /files/:chain/:address`
- Get source files (full or partial match) : `GET /files/any/:chain/:address`
- Get contract addresses (full or partial match) : `GET /files/contracts/:chain`

[docs.sourcify.dev](https://docs.sourcify.dev)

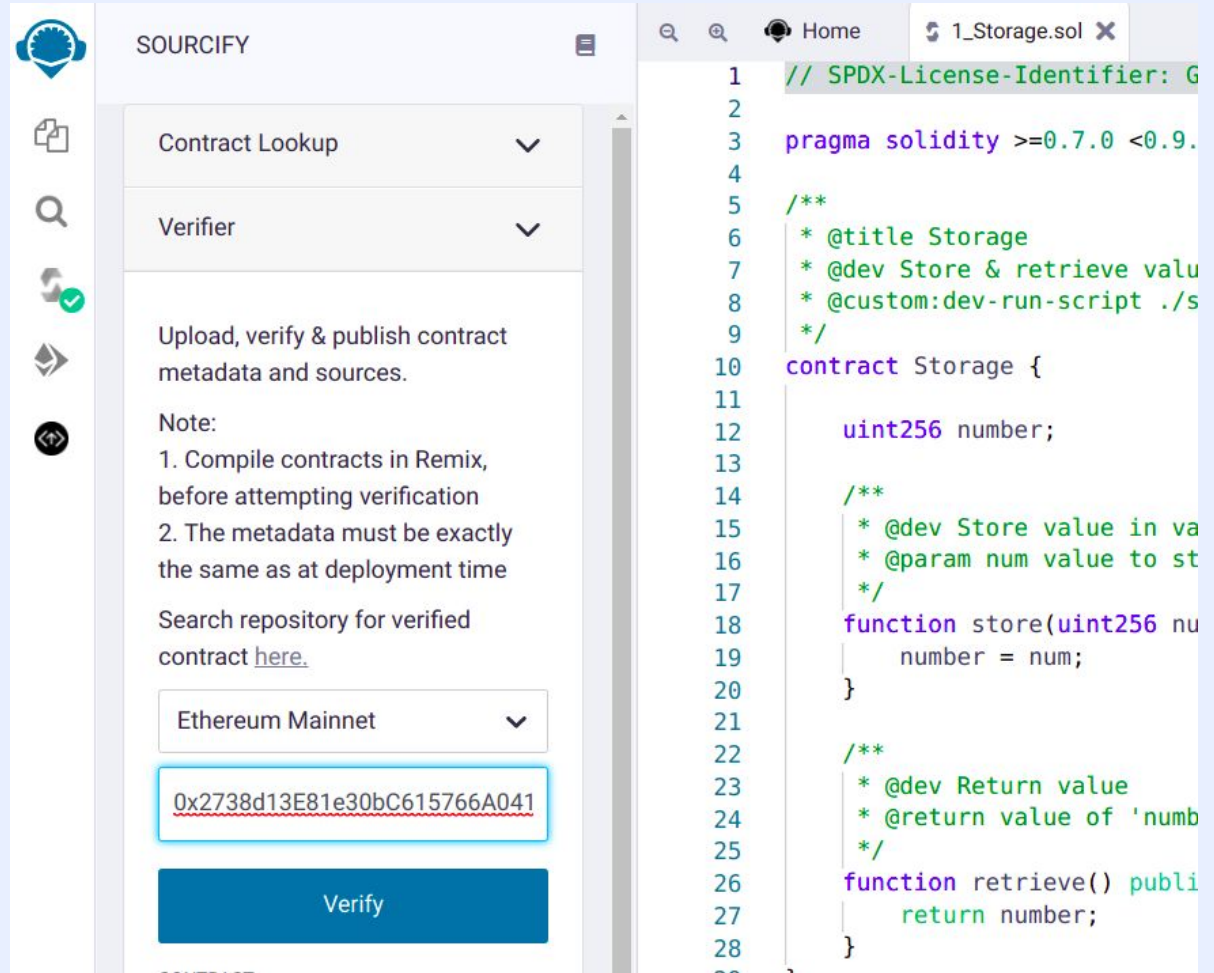
# Tooling

[@wighawag/hardhat-deploy](#)

```
$ hardhat --network mainnet sourcify
```

# Tooling

## Remix Plugin



**SOURCIFY**

Contract Lookup

Verifier

Upload, verify & publish contract metadata and sources.

Note:

1. Compile contracts in Remix, before attempting verification
2. The metadata must be exactly the same as at deployment time

Search repository for verified contract [here](#).

Ethereum Mainnet

0x2738d13E81e30bC615766A041

Verify

```
// SPDX-License-Identifier: G
pragma solidity >=0.7.0 <0.9.
/**
 * @title Storage
 * @dev Store & retrieve valu
 * @custom:dev-run-script ./s
 */
contract Storage {
    uint256 number;

    /**
     * @dev Store value in va
     * @param num value to st
     */
    function store(uint256 nu
        number = num;
    }

    /**
     * @dev Return value
     * @return value of 'numb
     */
    function retrieve() publi
        return number;
    }
```

# Automatic Verification

- i.e. **Monitor**
- Catches contract creations on **Ethereum Mainnet** and **Testnets**
- Tries to fetch metadata and source files from IPFS
- Automatically verifies

## As a contract developer:

- 1) Use NatSpec documentation
- 2) Publish metadata and source code on IPFS
- ~~3) Source code verification on Sourcify~~



## As a contract developer:

- 1) Use NatSpec documentation
- 2) Publish metadata and source code on IPFS
- ~~3) Source code verification on Sourcify~~

**If you publish on IPFS we verify for you! 🎉**

# Contract repository

- Served over HTTP ([repo.sourcify.dev](https://repo.sourcify.dev)) and IPFS
- Over 1M small files ~11GB
- IPNS updated regularly
  - currently every 6 hours
- [/ipns/k51qzi5uqu5dll10ocge71eudqnrnogmbr37gsgl12uubsinphjoknl6bbi41p/contracts/{\(full\\_match | partial\\_match\)}/{chainId}/{address}](https://ipns/k51qzi5uqu5dll10ocge71eudqnrnogmbr37gsgl12uubsinphjoknl6bbi41p/contracts/{(full_match | partial_match)}/{chainId}/{address})

# Contract repository

- Served over HTTP ([repo.sourcify.dev](https://repo.sourcify.dev)) and IPFS
- Over 1M small files ~11GB
- IPNS updated regularly
  - currently every 6 hours
- [/ipns/k51qzi5uqu5dll10ocge71eudqnrnogmbr37gsql12uubsinphjoknl6bbi41p/contracts/{\(full\\_match | partial\\_match\)}/{chainId}/{address}](https://ipns/k51qzi5uqu5dll10ocge71eudqnrnogmbr37gsql12uubsinphjoknl6bbi41p/contracts/{(full_match | partial_match)}/{chainId}/{address})

## EXTERNAL LINKS

Documentation

Contract Repository (IPFS)

Brand Resources

# From wallet/block explorer

1) Fetch metadata

[https://repo.sourcify.dev/contracts/full\\_match/{chainId}/{address}/metadata.json](https://repo.sourcify.dev/contracts/full_match/{chainId}/{address}/metadata.json)

# From wallet/block explorer

1) Fetch metadata

[https://repo.sourcify.dev/contracts/full\\_match/{chainId}/{address}/metadata.json](https://repo.sourcify.dev/contracts/full_match/{chainId}/{address}/metadata.json)

# From wallet/block explorer

1) Get contract bytecode `eth_getCode("0x1fa47...b2c53f")`

```
15610cad57610cad610cd5565b02949350505050565b6000816000190483118215151615610cd057610cd0610cd5565b50
0290565b634e487b7160e01b600052601160045260246000fdfea264697066735822122078b530288f1cffe879bb7d9062
e904deb3aa9b9c8d27ea4ecafa987583c18a6e64736f6c63430008010033
```

# From wallet/block explorer

1) Get contract bytecode `eth_getCode("0x1fa47...b2c53f")`

```
15610cad57610cad610cd5565b02949350505050565b6000816000190483118215151615610cd057610cd0610cd5565b50
0290565b634e487b7160e01b600052601160045260246000fdfea264697066735822122078b530288f1cffe879bb7d9062
e904deb3aa9b9c8d27ea4ecafa987583c18a6e64736f6c63430008010033
```

2) Fetch metadata via the IPFS hash

## Metadata Hash (decoded)

`ipfs://QmWTqspM5B1quNvdhXbS6TbXzyLZ5cUGHnTV8ZWJPqrQqj`

# From wallet/block explorer

1) Get contract bytecode `eth_getCode("0x1fa47...b2c53f")`

```
15610cad57610cad610cd5565b02949350505050565b6000816000190483118215151615610cd057610cd0610cd5565b50  
0290565b634e487b7160e01b600052601160045260246000fdfea264697066735822122078b530288f1cffe879bb7d9062  
e904deb3aa9b9c8d27ea4ecafa987583c18a6e64736f6c63430008010033
```

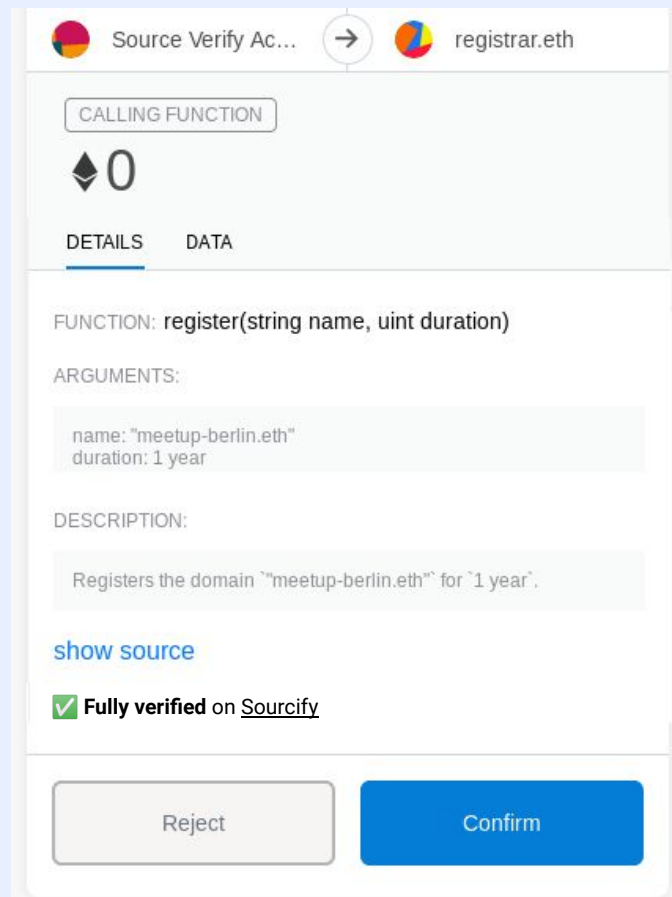
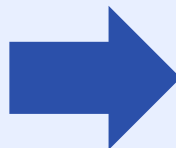
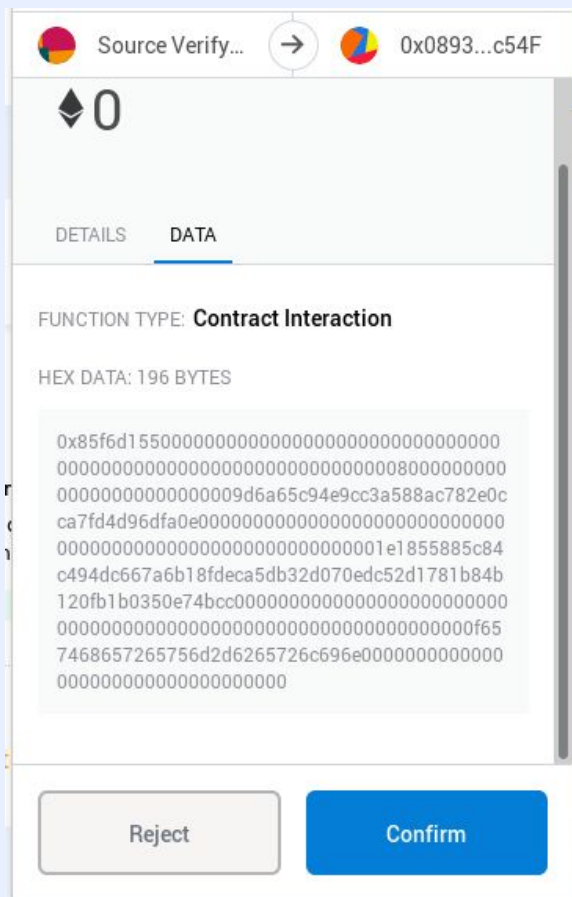
2) Fetch metadata via the IPFS hash

## Metadata Hash (decoded)

`ipfs://QmWTqspM5B1quNvdhXbS6TbXzyLZ5cUGHnTV8ZWJPqrQqj`

3) Decode **ABI** + populate **NatSpec** comments or **show sources**





# Thank you



[sourcify.dev](https://sourcify.dev)



[ethereum/sourcify](https://github.com/ethereum/sourcify)



[@sourcifyeth](https://twitter.com/sourcifyeth)



Gitter: [ethereum/source-verify](https://gitter.im/ethereum/source-verify)



Matrix chat: [#ethereum\\_source-verify:gitter.im](https://matrix.to/#/#ethereum_source-verify:gitter.im)




sourcify.dev

# **Additional Slides**

Rinkeby Test Network

 Account 1



 0x283...B7F5

New address detected! Click here to add to your address book.

https://app.ens.domains

COMMIT

DETAILS

DATA

HEX

FUNCTION TYPE: Commit (Bytes32)

commitment:

0x2be0d8d8c3dd293b3e7f88ecff8ce15749992e  
e66acfb0c4dcc4e36e4ca001e7



 Copy raw transaction data




Verified contract on [Etherscan](#)  
Decoded by Truffle

# Other chains?

Avalanche

 Hot Wallet 

→

 0xE54...9106

New address detected! Click [here](#) to add to your address book.

https://app.pangolin.exchange

SWAP EXACT A V A X FOR TOKENS


14.3852547

DETAILS

DATA

HEX

FUNCTION TYPE: Swap Exact A V A X For Tokens  
(Uint256, Address[], Address, Uint256)

 Transaction decoding is not available for chainId 43114

Rinkeby Test Network



Account 1



0x283...B7F5

New address detected! Click here to add to your address book.

`https://app.ens.domains`

COMMIT

DETAILS

DATA

HEX

FUNCTION TYPE: Commit (Bytes32)

commitment:

0x2be0d8d8c3dd293b3e7f88ecff8ce15749992e  
e66acfb0c4dcc4e36e4ca001e7



Copy raw transaction data



Verified contract on [Etherscan](#)

Decoded by Truffle

Rinkeby Test Network

Account 1 → ENS Contract  
verified by TrustUs

New address detected! Click here to add to your address book.

<https://app.ens.domains>  
COMMIT

DETAILS DATA HEX

FUNCTION TYPE: Commit (Bytes32)

commitment:  
0x2be0d8d8c3dd293b3e7f88ecff8ce15749992e  
e66acfb0c4dcc4e36e4ca001e7

Copy raw transaction data

Verified contract on [Etherscan](#)  
Decoded by Truffle

e.g.  
Trusted contract registries

## **Two main questions:**

- **Am I talking to the right contract?**



## Two main questions:

- Am I talking to the right contract?
- **Is this contract call doing what I want to do?**