DailySocial.id

Web3 Developer Bootcamp

Building the Builders of the Future

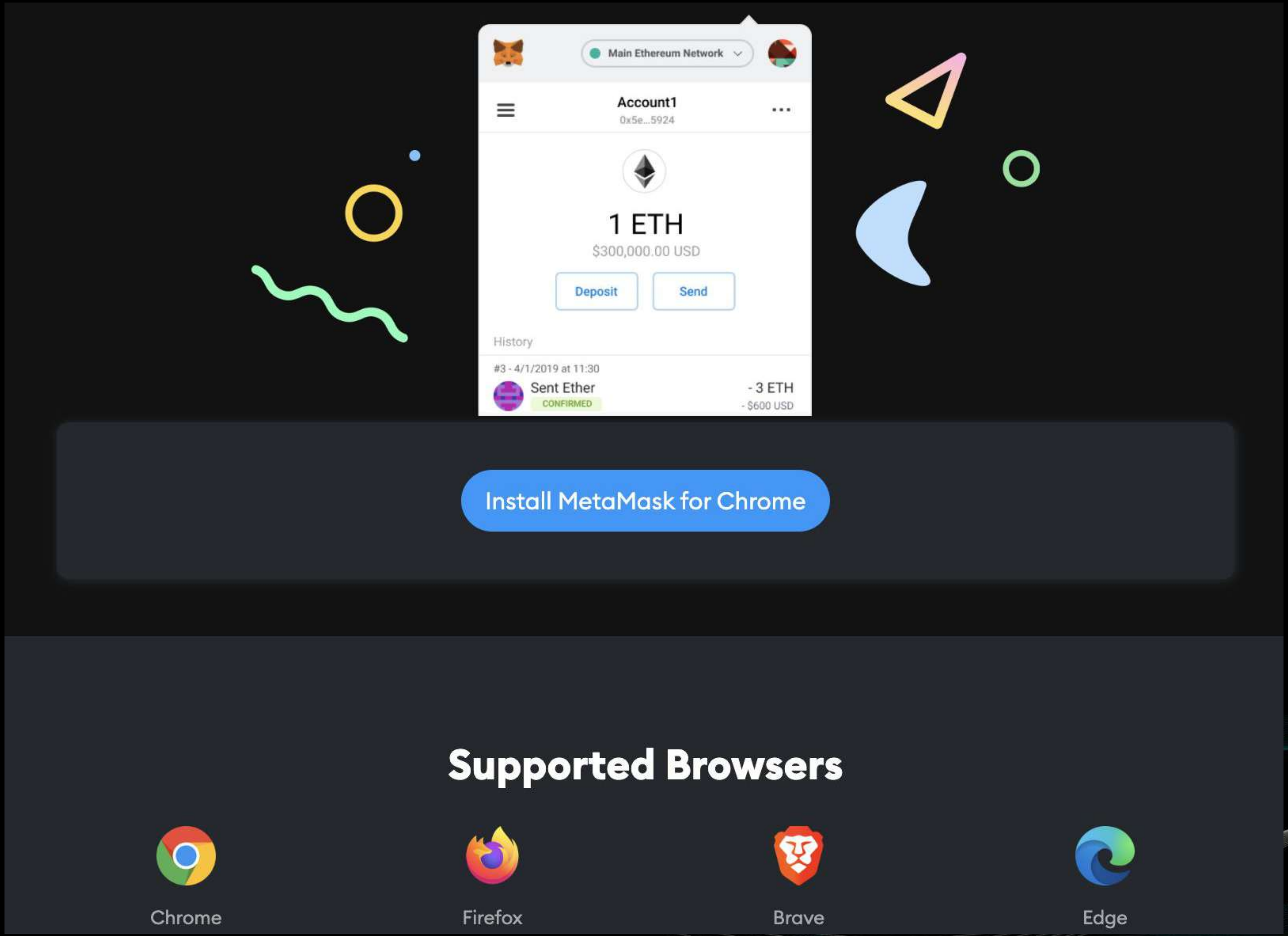
Frontend Web Application Using Web3.js

@sofianhw

Contents

- Install Metamask
- What Metamask Do
- Connect Metamask
- Interact with SmartContract

Install Metamask



Account1
0x5e...5924

1 ETH
$300,000.00 USD

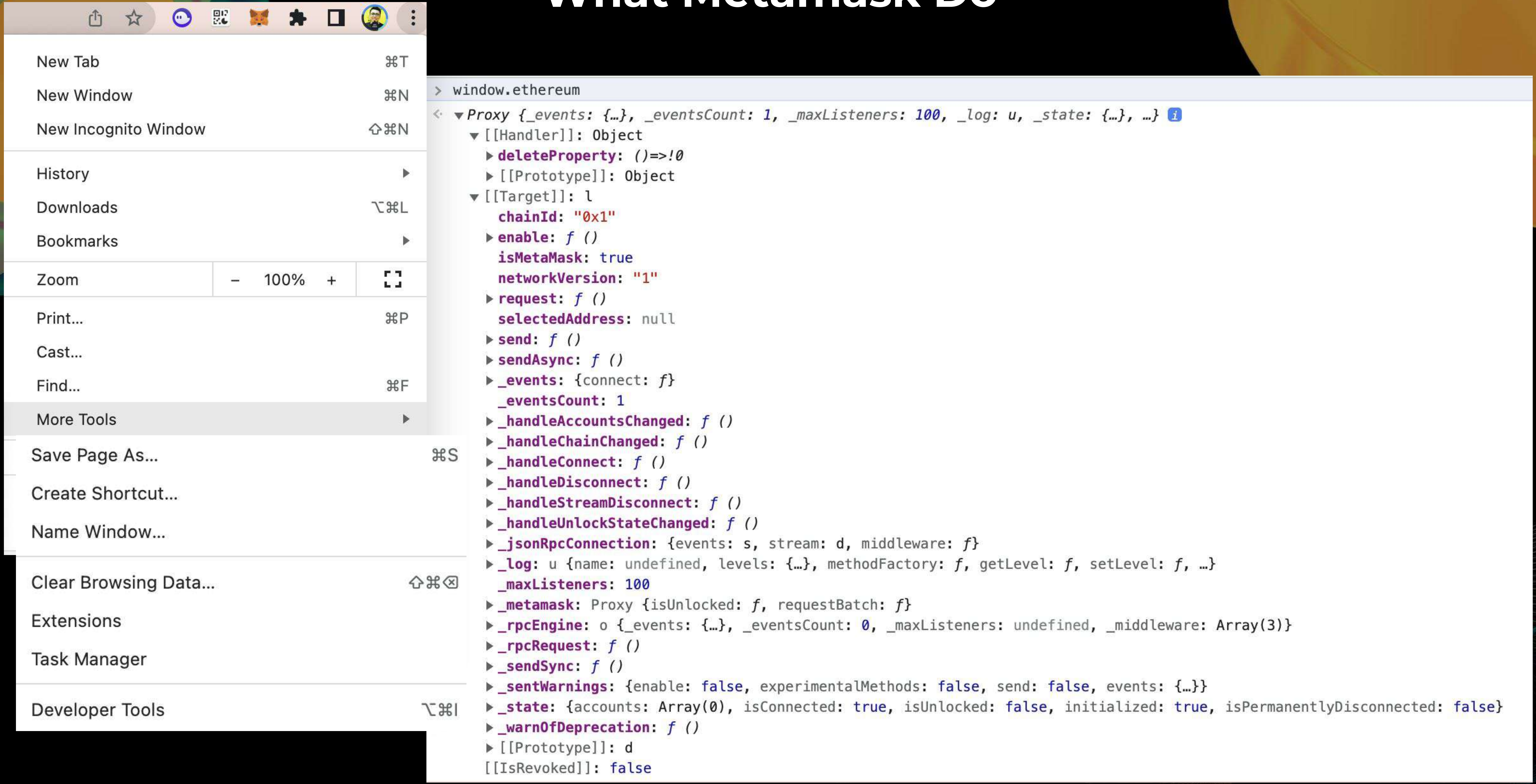
History

#3 - 4/1/2019 at 11:30
Sent Ether CONFIRMED -3 ETH -$600 USD

Supported Browsers

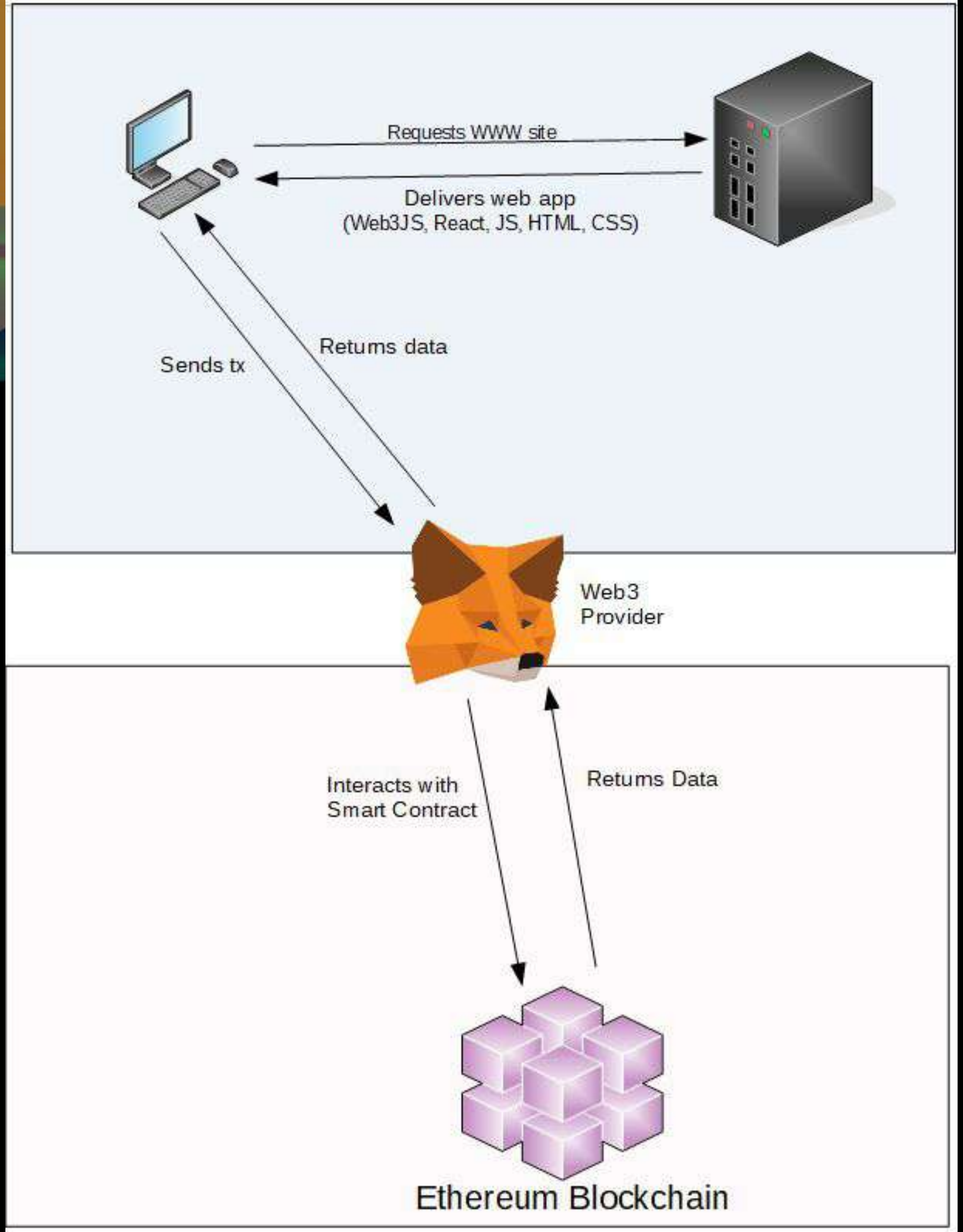
Chrome Firefox Brave Edge

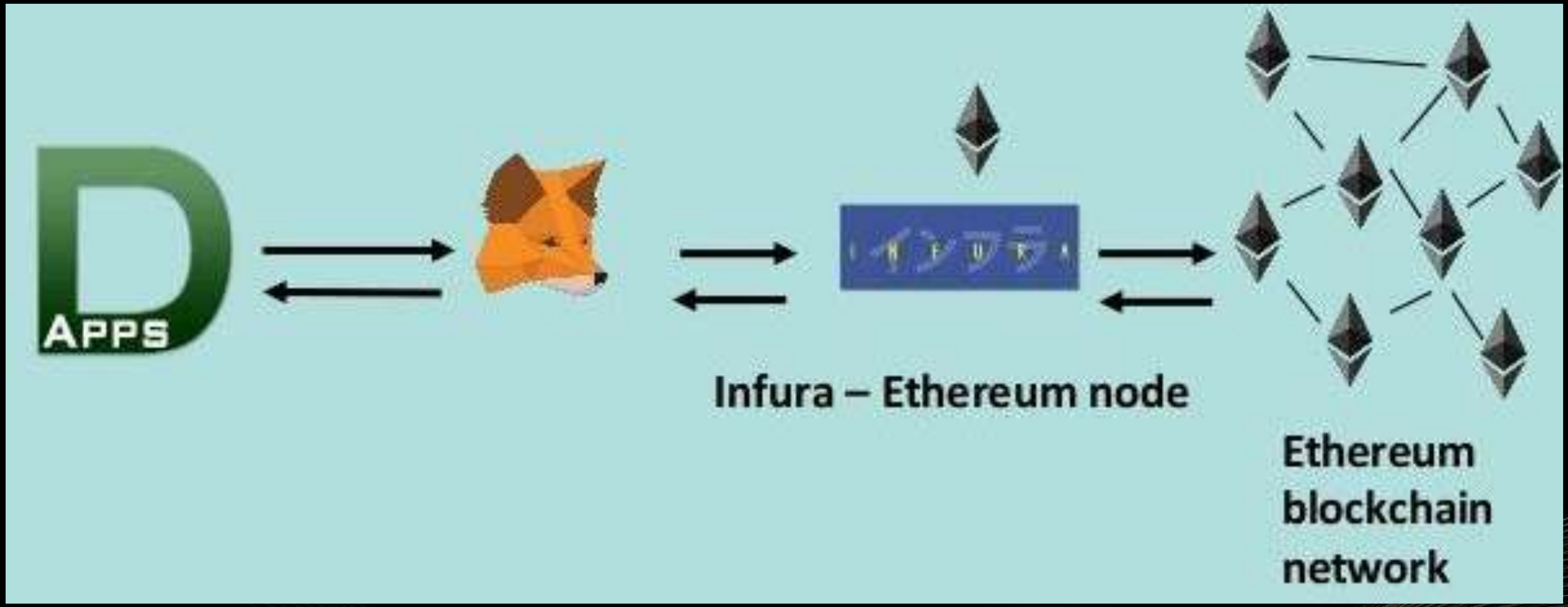
What Metamask Do



```
> window.ethereum
<· ▼Proxy {_events: {…}, _eventsCount: 1, _maxListeners: 100, _log: u, _state: {…}, …} i
    ▼[[Handler]]: Object
      ▶deleteProperty: ()=>!0
      ▶[[Prototype]]: Object
    ▼[[Target]]: l
       chainId: "0x1"
      ▶enable: ƒ ()
       isMetaMask: true
       networkVersion: "1"
      ▶request: ƒ ()
       selectedAddress: null
      ▶send: ƒ ()
      ▶sendAsync: ƒ ()
      ▶_events: {connect: ƒ}
       _eventsCount: 1
      ▶_handleAccountsChanged: ƒ ()
      ▶_handleChainChanged: ƒ ()
      ▶_handleConnect: ƒ ()
      ▶_handleDisconnect: ƒ ()
      ▶_handleStreamDisconnect: ƒ ()
      ▶_handleUnlockStateChanged: ƒ ()
      ▶_jsonRpcConnection: {events: s, stream: d, middleware: ƒ}
      ▶_log: u {name: undefined, levels: {…}, methodFactory: ƒ, getLevel: ƒ, setLevel: ƒ, …}
       _maxListeners: 100
      ▶_metamask: Proxy {isUnlocked: ƒ, requestBatch: ƒ}
      ▶_rpcEngine: o {_events: {…}, _eventsCount: 0, _maxListeners: undefined, _middleware: Array(3)}
      ▶_rpcRequest: ƒ ()
      ▶_sendSync: ƒ ()
      ▶_sentWarnings: {enable: false, experimentalMethods: false, send: false, events: {…}}
      ▶_state: {accounts: Array(0), isConnected: true, isUnlocked: false, initialized: true, isPermanentlyDisconnected: false}
      ▶_warnOfDeprecation: ƒ ()
      ▶[[Prototype]]: d
     [[IsRevoked]]: false
```

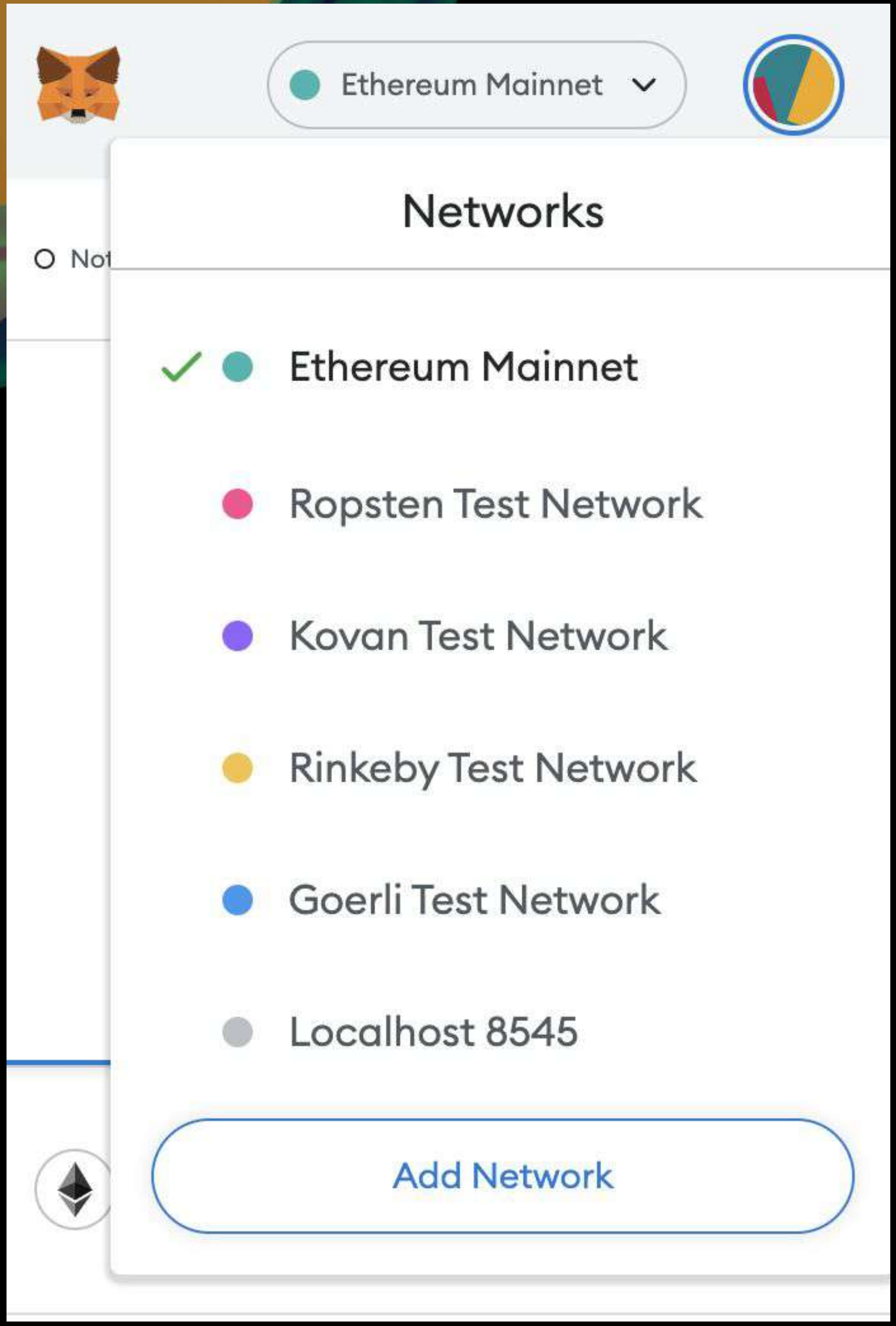
What Metamask Do





window.ethereum

What Metamask Do



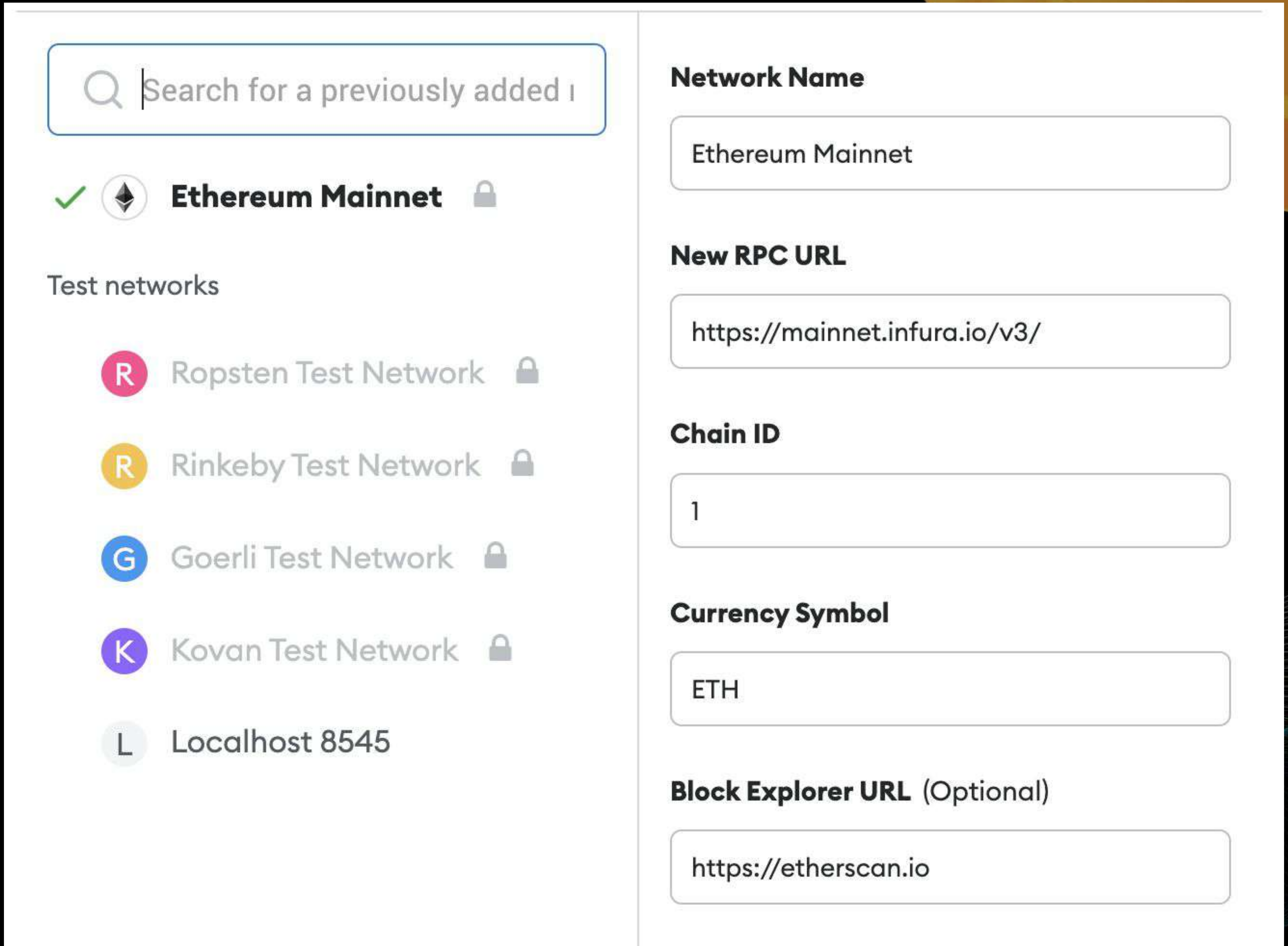
Ethereum Mainnet

Networks

Not

- ✓ ● Ethereum Mainnet
- ● Ropsten Test Network
- ● Kovan Test Network
- ● Rinkeby Test Network
- ● Goerli Test Network
- ● Localhost 8545

Add Network



Search for a previously added ı

✓ Ethereum Mainnet

Test networks

- R Ropsten Test Network
- R Rinkeby Test Network
- G Goerli Test Network
- K Kovan Test Network
- L Localhost 8545

Network Name

Ethereum Mainnet

New RPC URL

https://mainnet.infura.io/v3/

Chain ID

1

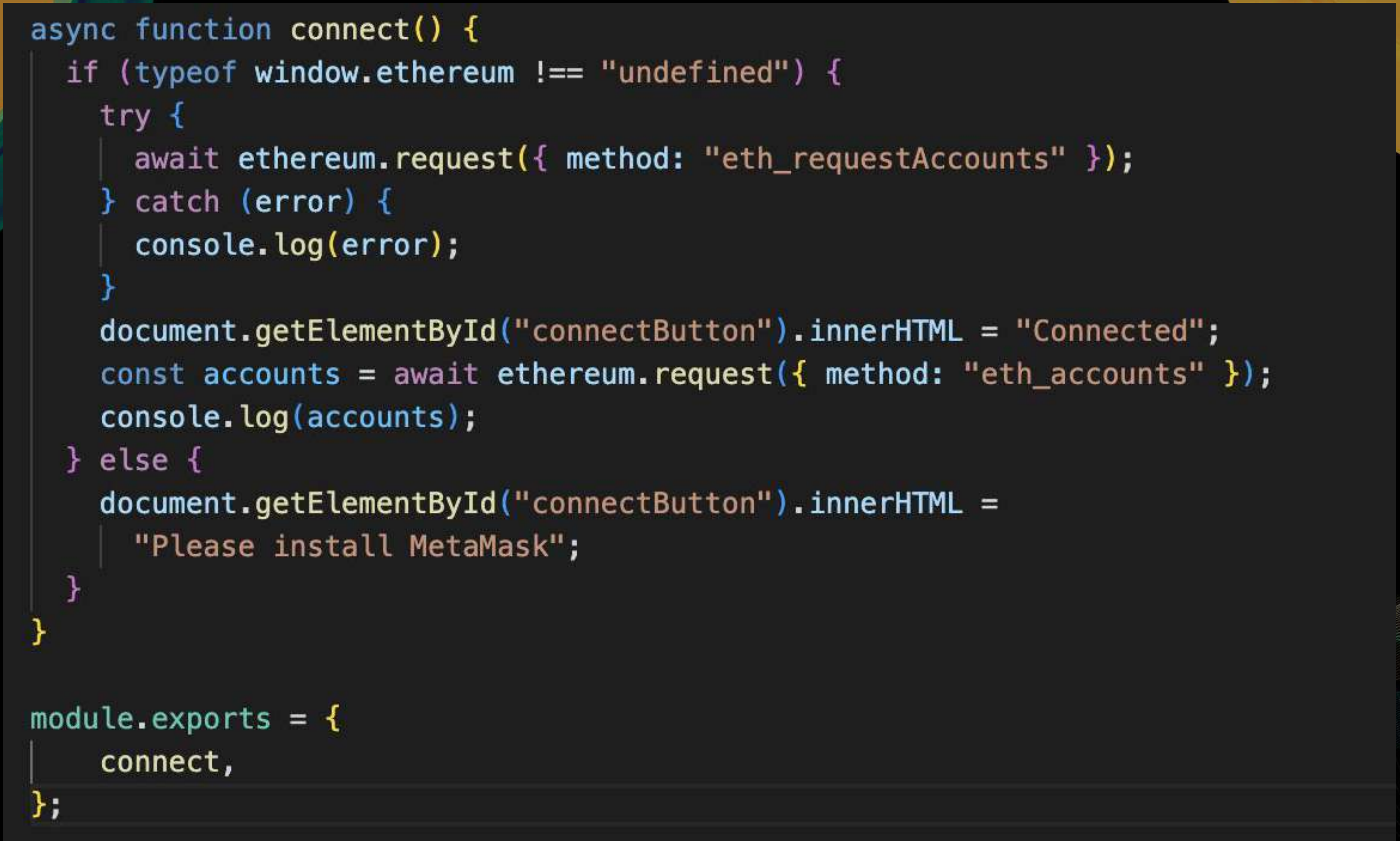
Currency Symbol

ETH

Block Explorer URL (Optional)

https://etherscan.io

Connect Metamask

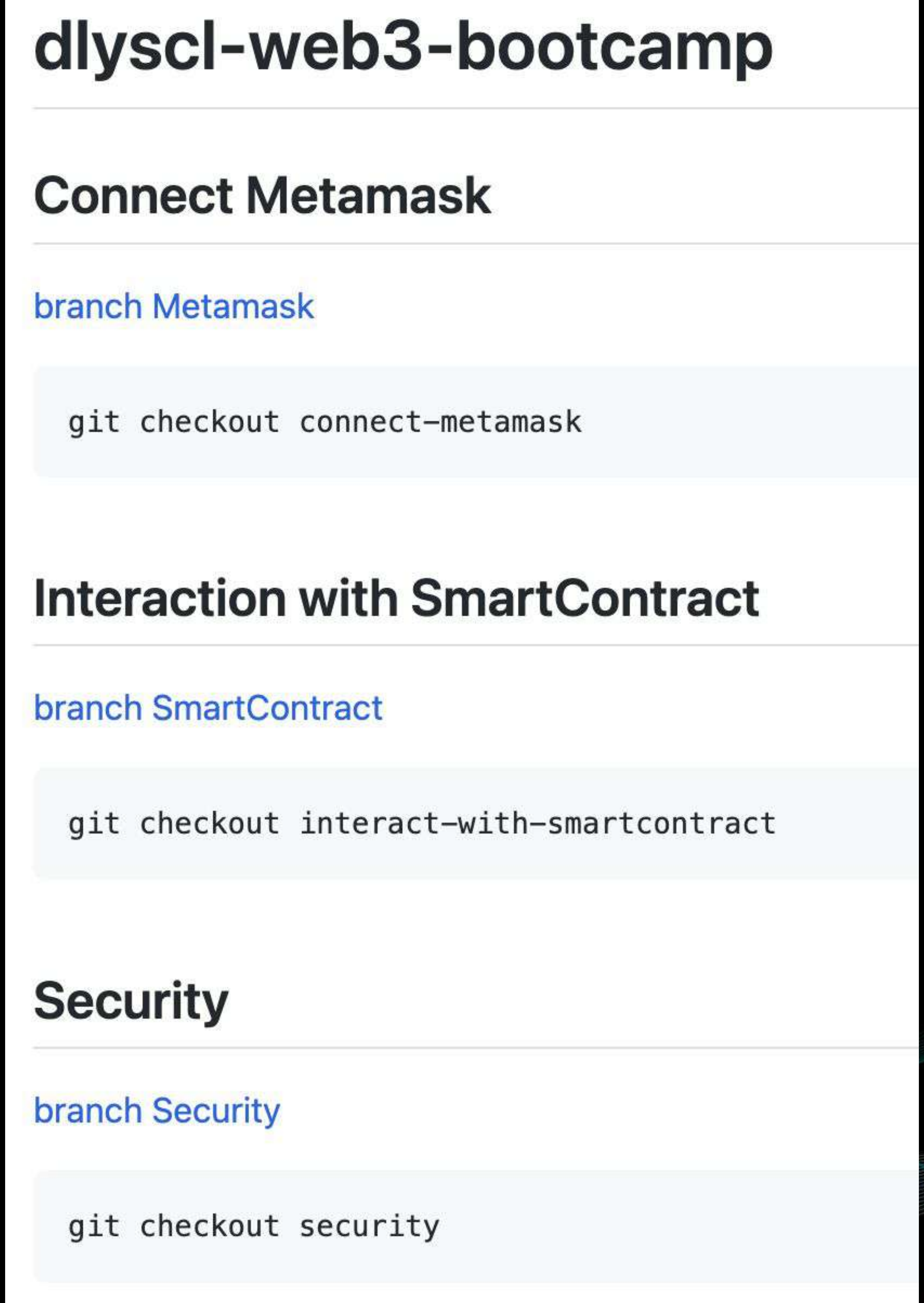


```
async function connect() {
  if (typeof window.ethereum !== "undefined") {
    try {
      await ethereum.request({ method: "eth_requestAccounts" });
    } catch (error) {
      console.log(error);
    }
    document.getElementById("connectButton").innerHTML = "Connected";
    const accounts = await ethereum.request({ method: "eth_accounts" });
    console.log(accounts);
  } else {
    document.getElementById("connectButton").innerHTML =
      "Please install MetaMask";
  }
}

module.exports = {
    connect,
};
```

Let’s Do it

https://github.com/sofianhw/dlyscl-web3-bootcamp



dlyscl-web3-bootcamp

Connect Metamask

branch Metamask

```
git checkout connect-metamask
```

Interaction with SmartContract

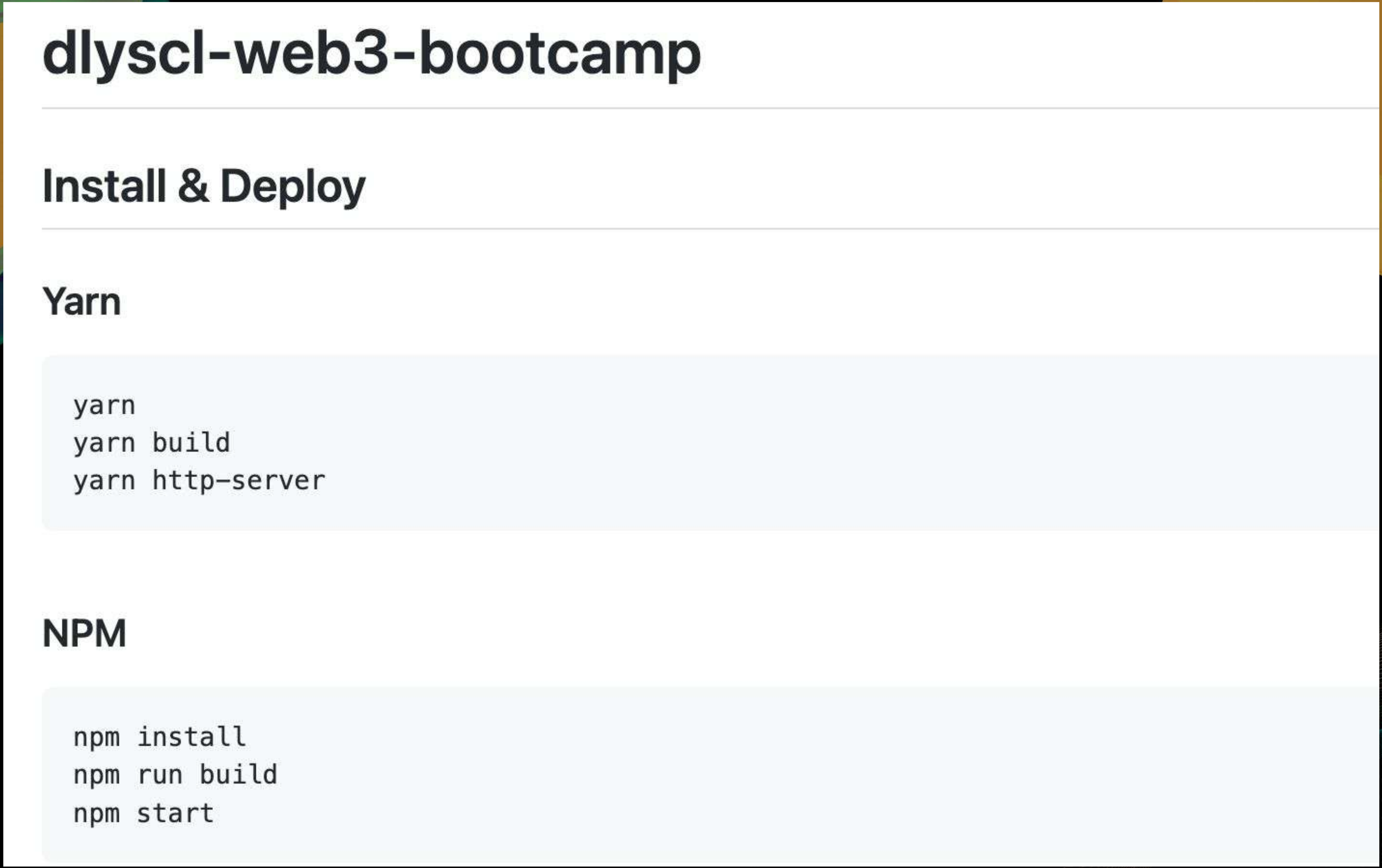
branch SmartContract

```
git checkout interact-with-smartcontract
```

Security

branch Security

```
git checkout security
```



dlyscl-web3-bootcamp

Install & Deploy

Yarn

```
yarn
yarn build
yarn http-server
```

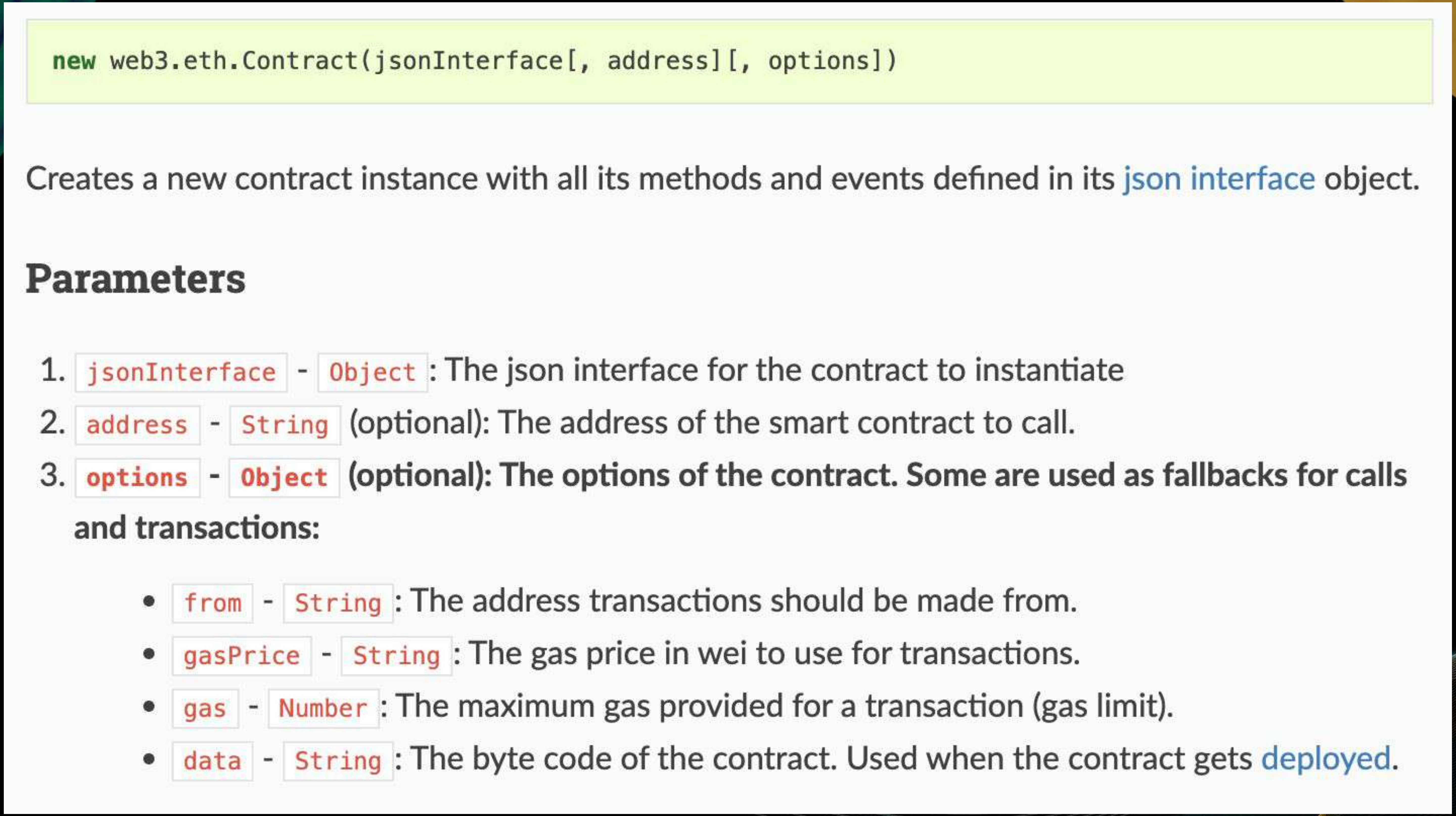
NPM

```
npm install
npm run build
npm start
```

Interact with Smart Contract

- Initiate
- Read
- Write
- Listening Event

Initiate



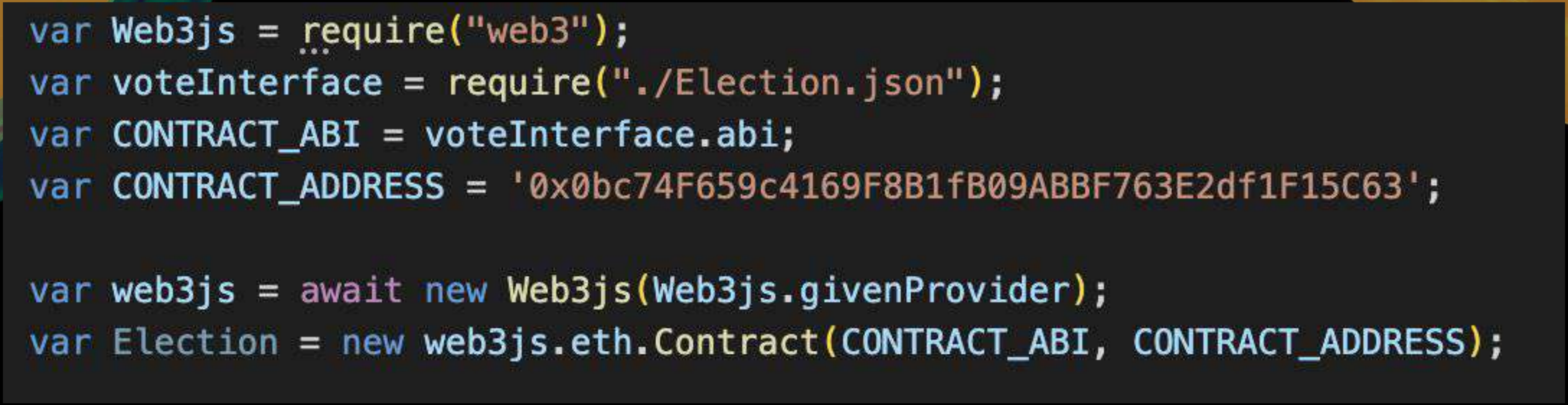
```
new web3.eth.Contract(jsonInterface[, address][, options])
```

Creates a new contract instance with all its methods and events defined in its json interface object.

Parameters

1. `jsonInterface` - `Object`: The json interface for the contract to instantiate
2. `address` - `String` (optional): The address of the smart contract to call.
3. **`options` - `Object` (optional): The options of the contract. Some are used as fallbacks for calls and transactions:**
   - `from` - `String`: The address transactions should be made from.
   - `gasPrice` - `String`: The gas price in wei to use for transactions.
   - `gas` - `Number`: The maximum gas provided for a transaction (gas limit).
   - `data` - `String`: The byte code of the contract. Used when the contract gets deployed.

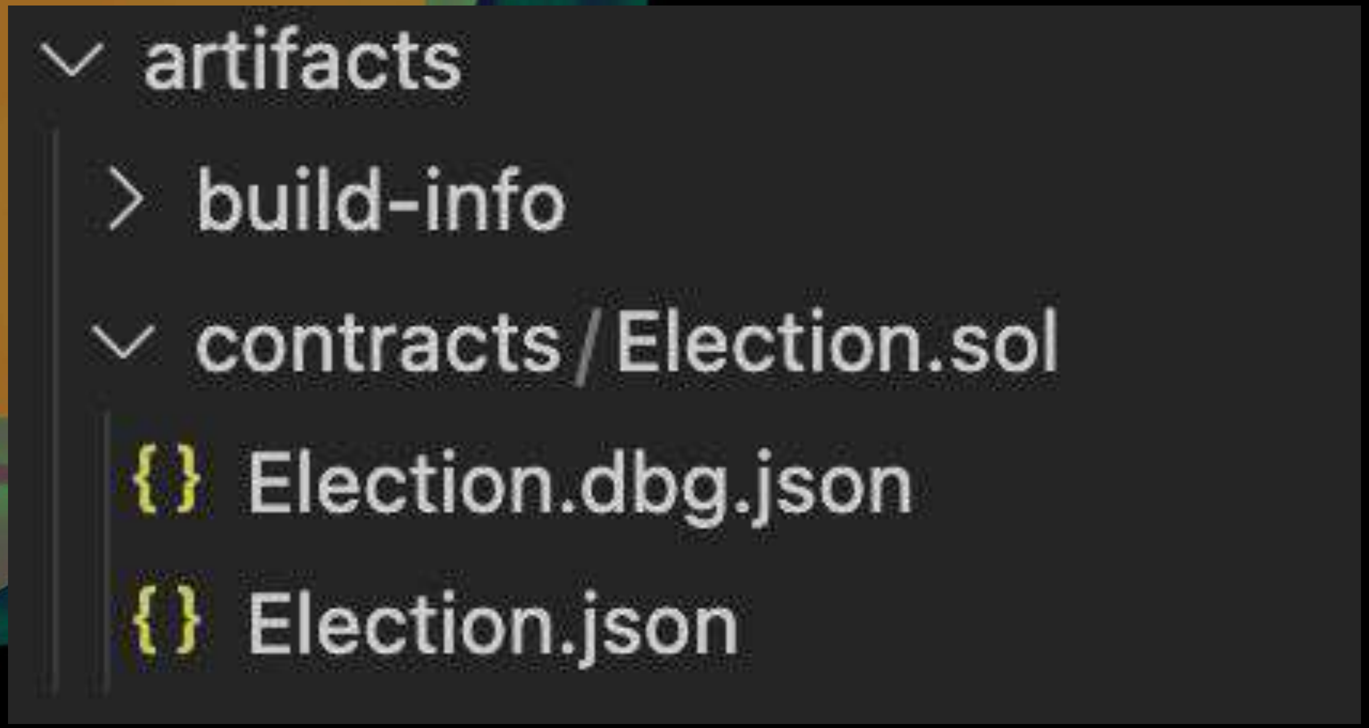
Initiate



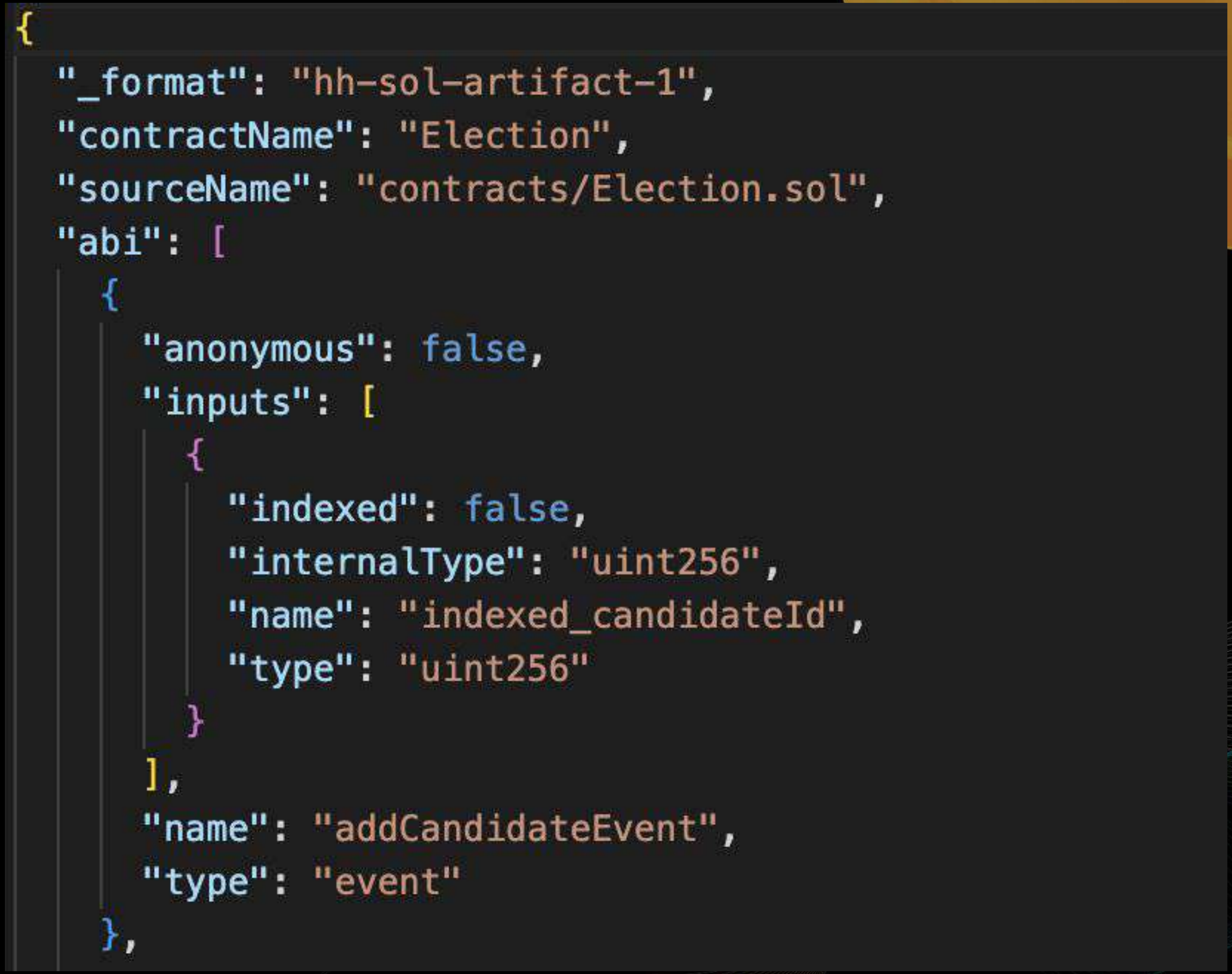
```
var Web3js = require("web3");
var voteInterface = require("./Election.json");
var CONTRACT_ABI = voteInterface.abi;
var CONTRACT_ADDRESS = '0x0bc74F659c4169F8B1fB09ABBF763E2df1F15C63';

var web3js = await new Web3js(Web3js.givenProvider);
var Election = new web3js.eth.Contract(CONTRACT_ABI, CONTRACT_ADDRESS);
```

Initiate

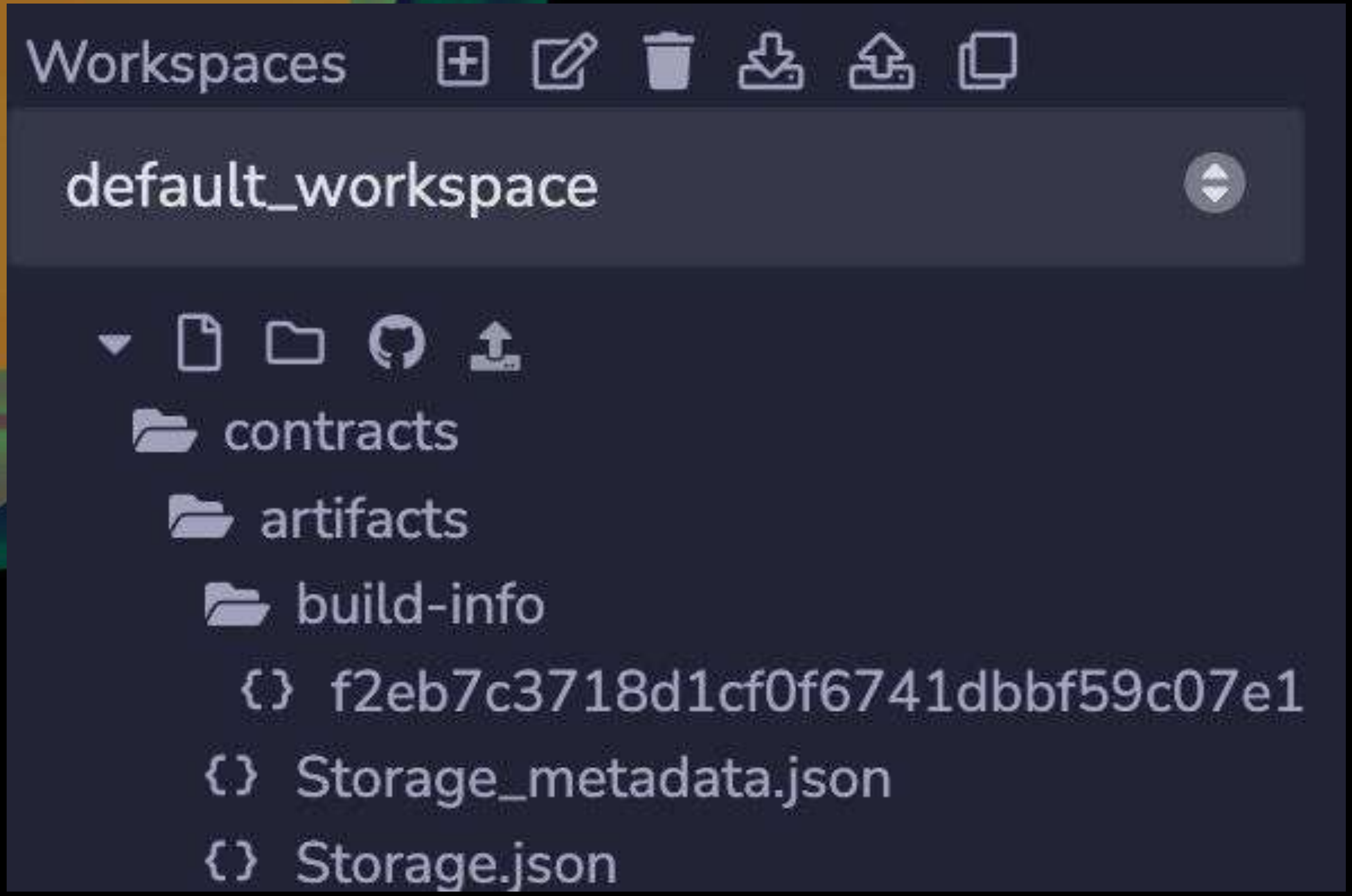


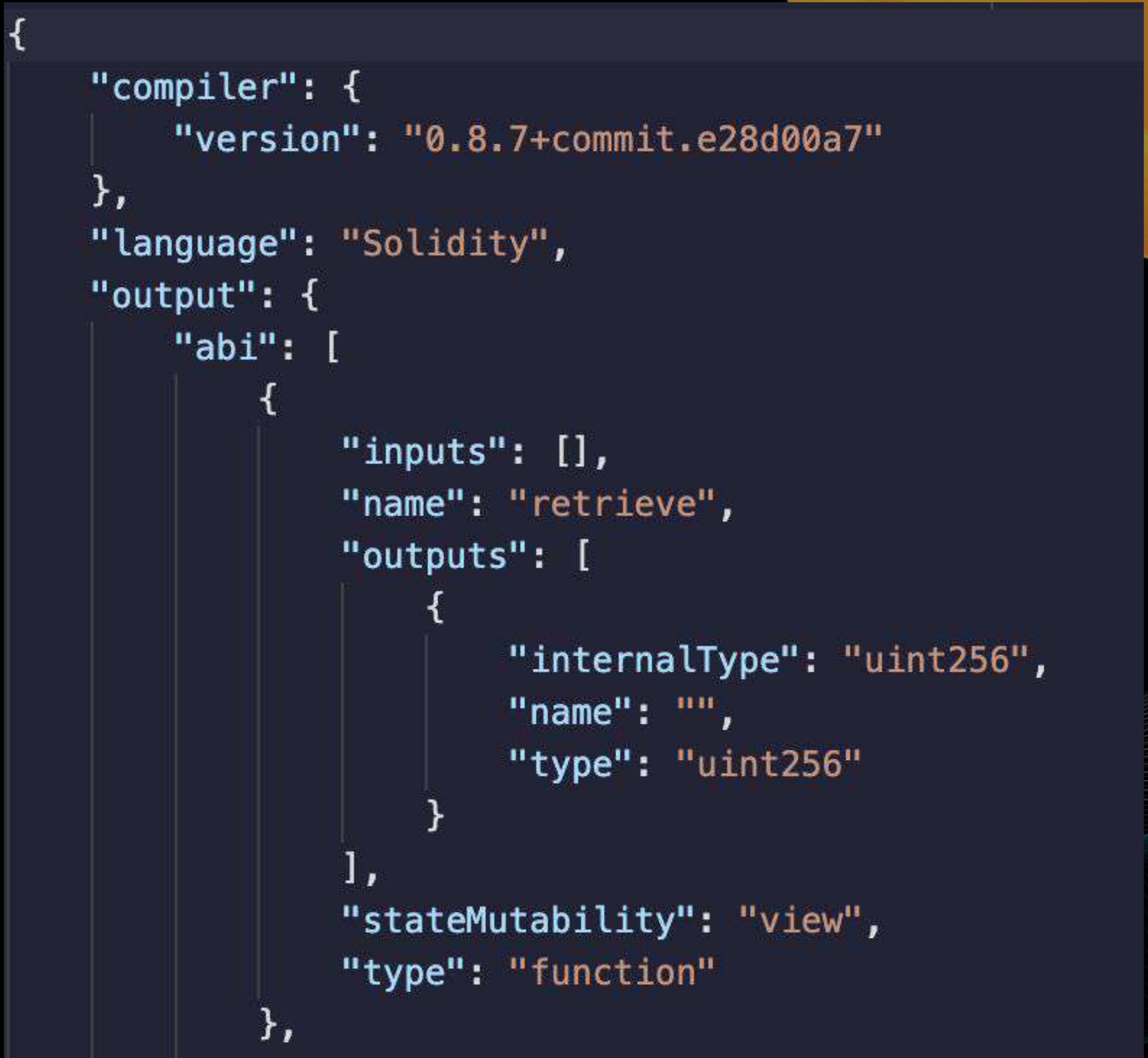
```
artifacts
  build-info
  contracts/Election.sol
    Election.dbg.json
    Election.json
```



```
{
  "_format": "hh-sol-artifact-1",
  "contractName": "Election",
  "sourceName": "contracts/Election.sol",
  "abi": [
    {
      "anonymous": false,
      "inputs": [
        {
          "indexed": false,
          "internalType": "uint256",
          "name": "indexed_candidateId",
          "type": "uint256"
        }
      ],
      "name": "addCandidateEvent",
      "type": "event"
    },
```

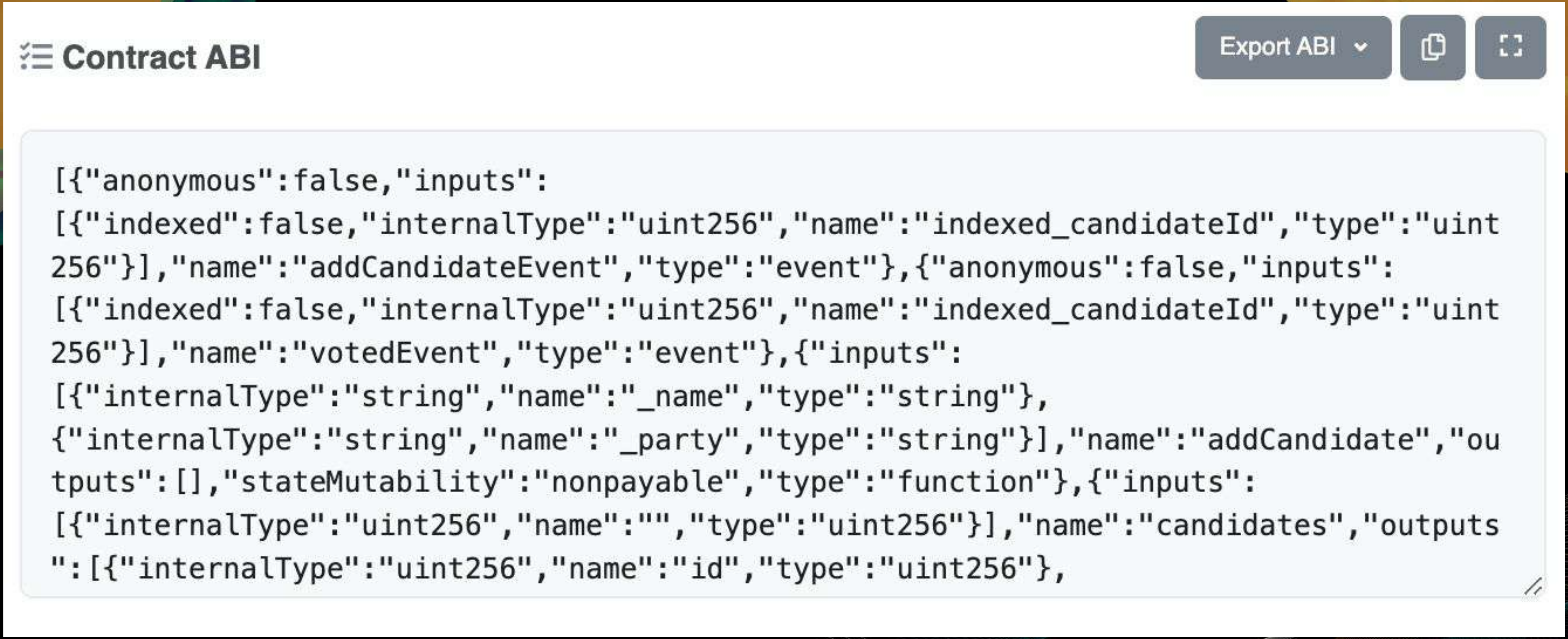
Initiate





```
{
    "compiler": {
        "version": "0.8.7+commit.e28d00a7"
    },
    "language": "Solidity",
    "output": {
        "abi": [
            {
                "inputs": [],
                "name": "retrieve",
                "outputs": [
                    {
                        "internalType": "uint256",
                        "name": "",
                        "type": "uint256"
                    }
                ],
                "stateMutability": "view",
                "type": "function"
            },
```

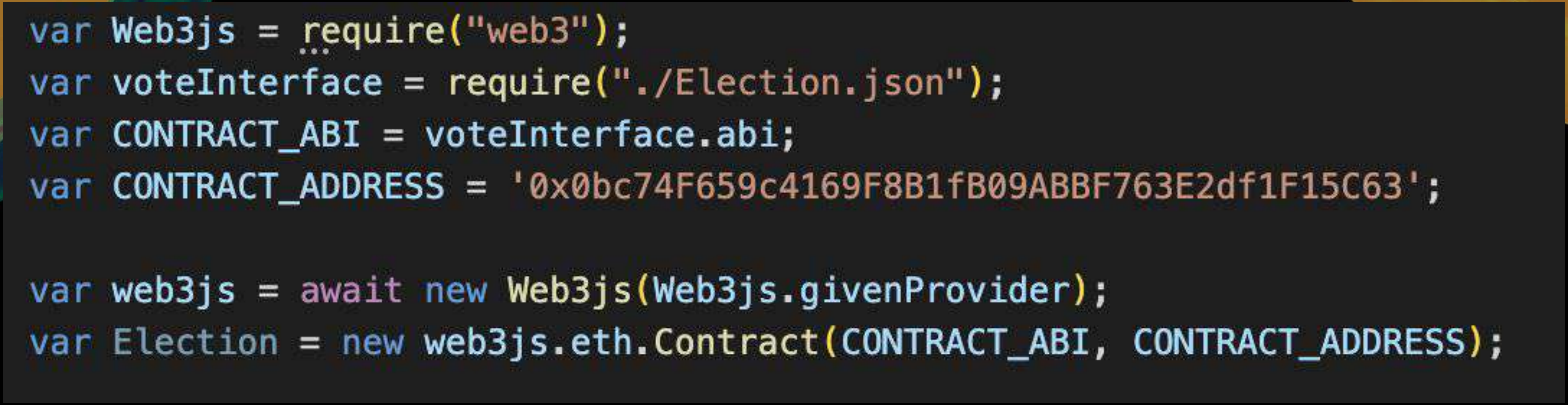
Initiate



Contract ABI

```
[{"anonymous":false,"inputs":
[{"indexed":false,"internalType":"uint256","name":"indexed_candidateId","type":"uint
256"}],"name":"addCandidateEvent","type":"event"},{"anonymous":false,"inputs":
[{"indexed":false,"internalType":"uint256","name":"indexed_candidateId","type":"uint
256"}],"name":"votedEvent","type":"event"},{"inputs":
[{"internalType":"string","name":"_name","type":"string"},
{"internalType":"string","name":"_party","type":"string"}],"name":"addCandidate","ou
tputs":[],"stateMutability":"nonpayable","type":"function"},{"inputs":
[{"internalType":"uint256","name":"","type":"uint256"}],"name":"candidates","outputs
":[{"internalType":"uint256","name":"id","type":"uint256"},
```

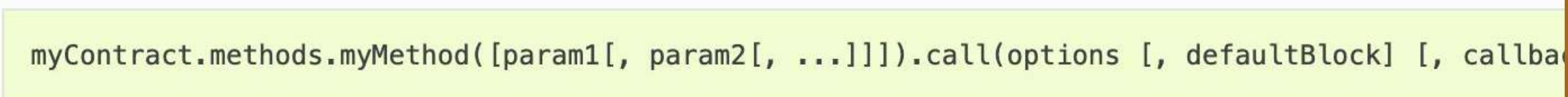
Initiate



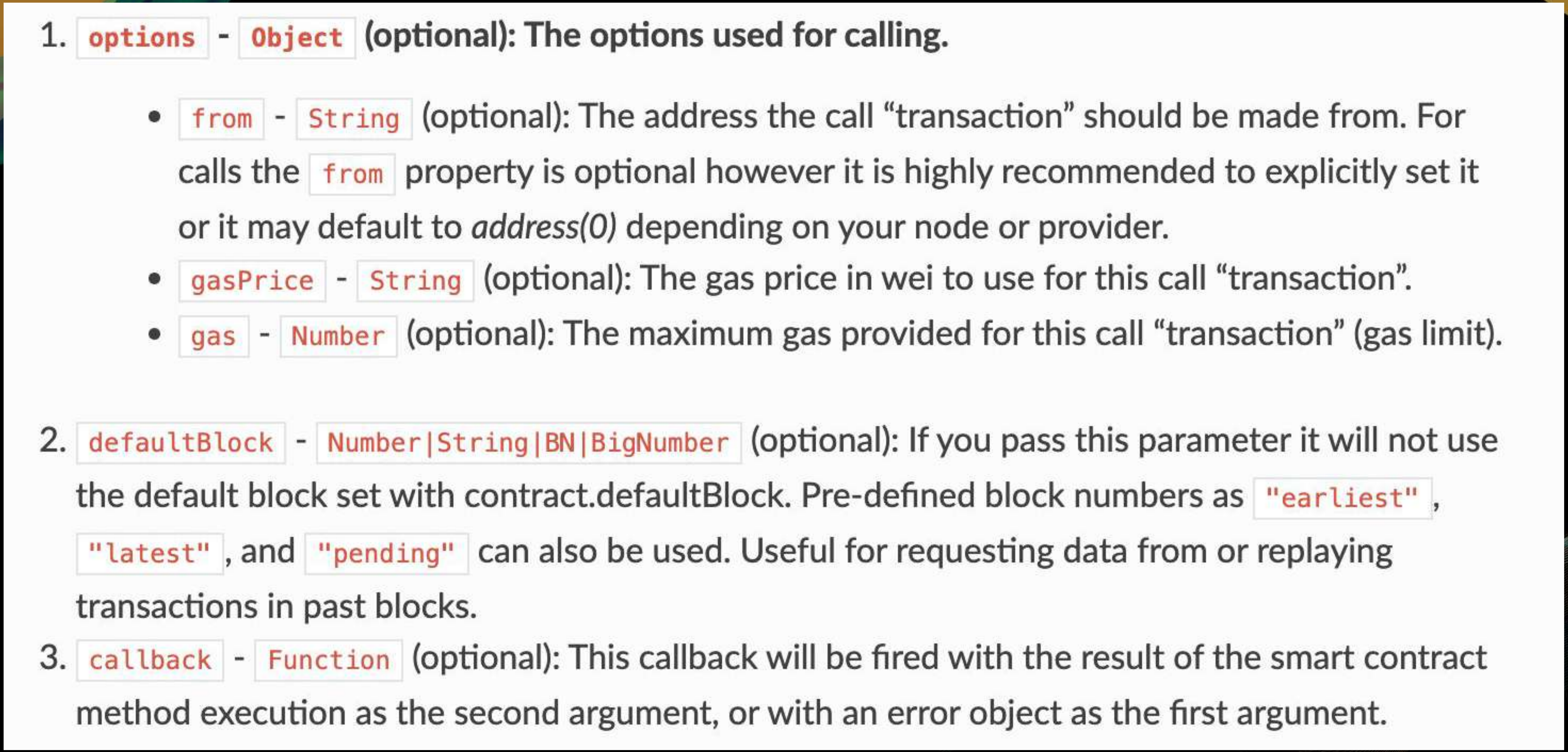
```
var Web3js = require("web3");
var voteInterface = require("./Election.json");
var CONTRACT_ABI = voteInterface.abi;
var CONTRACT_ADDRESS = '0x0bc74F659c4169F8B1fB09ABBF763E2df1F15C63';

var web3js = await new Web3js(Web3js.givenProvider);
var Election = new web3js.eth.Contract(CONTRACT_ABI, CONTRACT_ADDRESS);
```

Read

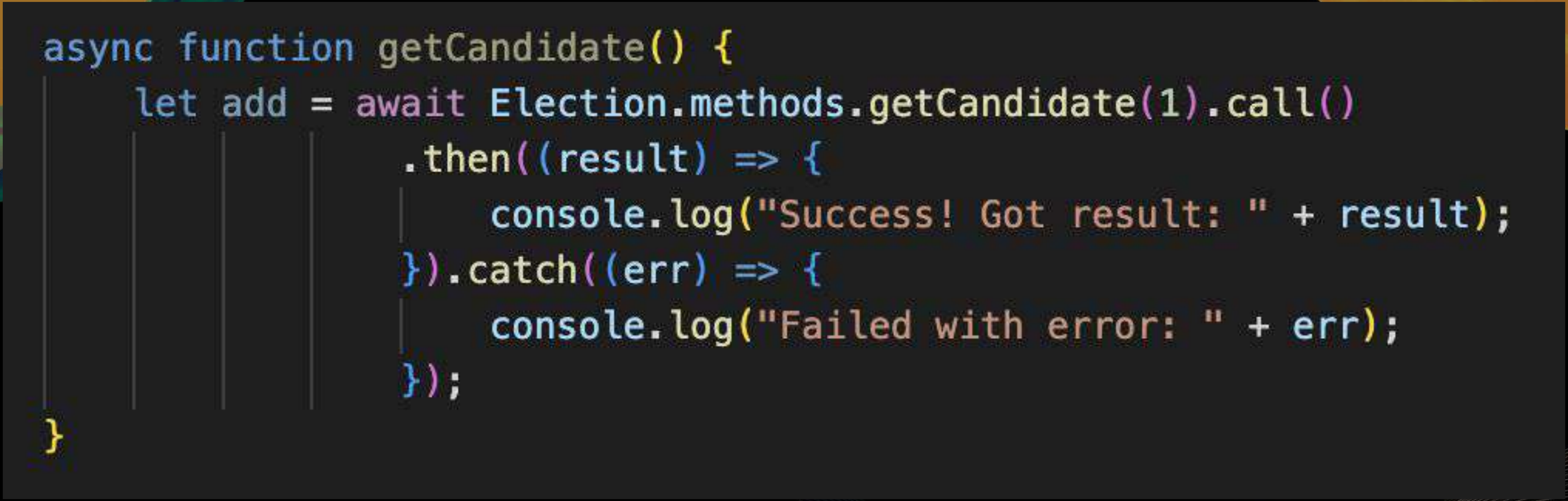


```
myContract.methods.myMethod([param1[, param2[, ...]]]).call(options [, defaultBlock] [, callba
```



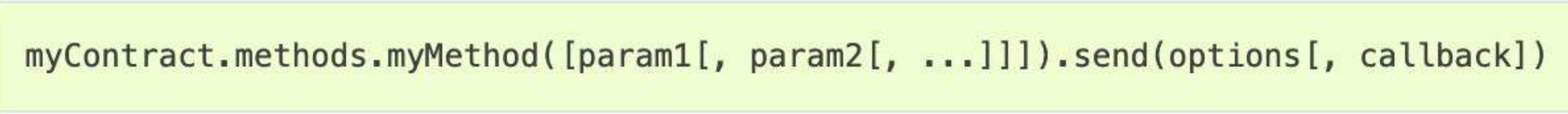
1. `options` - `Object` **(optional): The options used for calling.**
   - `from` - `String` (optional): The address the call "transaction" should be made from. For calls the `from` property is optional however it is highly recommended to explicitly set it or it may default to *address(0)* depending on your node or provider.
   - `gasPrice` - `String` (optional): The gas price in wei to use for this call "transaction".
   - `gas` - `Number` (optional): The maximum gas provided for this call "transaction" (gas limit).
2. `defaultBlock` - `Number|String|BN|BigNumber` (optional): If you pass this parameter it will not use the default block set with contract.defaultBlock. Pre-defined block numbers as `"earliest"`, `"latest"`, and `"pending"` can also be used. Useful for requesting data from or replaying transactions in past blocks.
3. `callback` - `Function` (optional): This callback will be fired with the result of the smart contract method execution as the second argument, or with an error object as the first argument.

Read

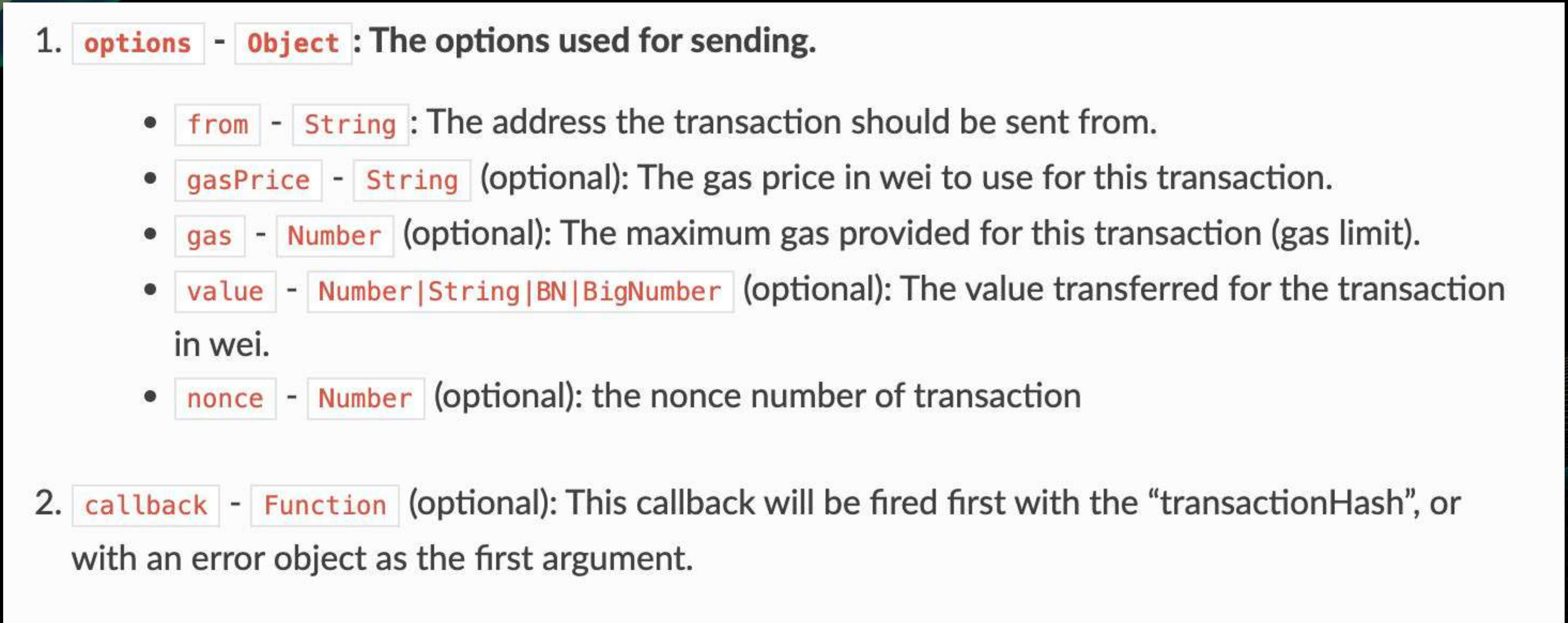


```
async function getCandidate() {
    let add = await Election.methods.getCandidate(1).call()
                .then((result) => {
                    console.log("Success! Got result: " + result);
                }).catch((err) => {
                    console.log("Failed with error: " + err);
                });
}
```

Write



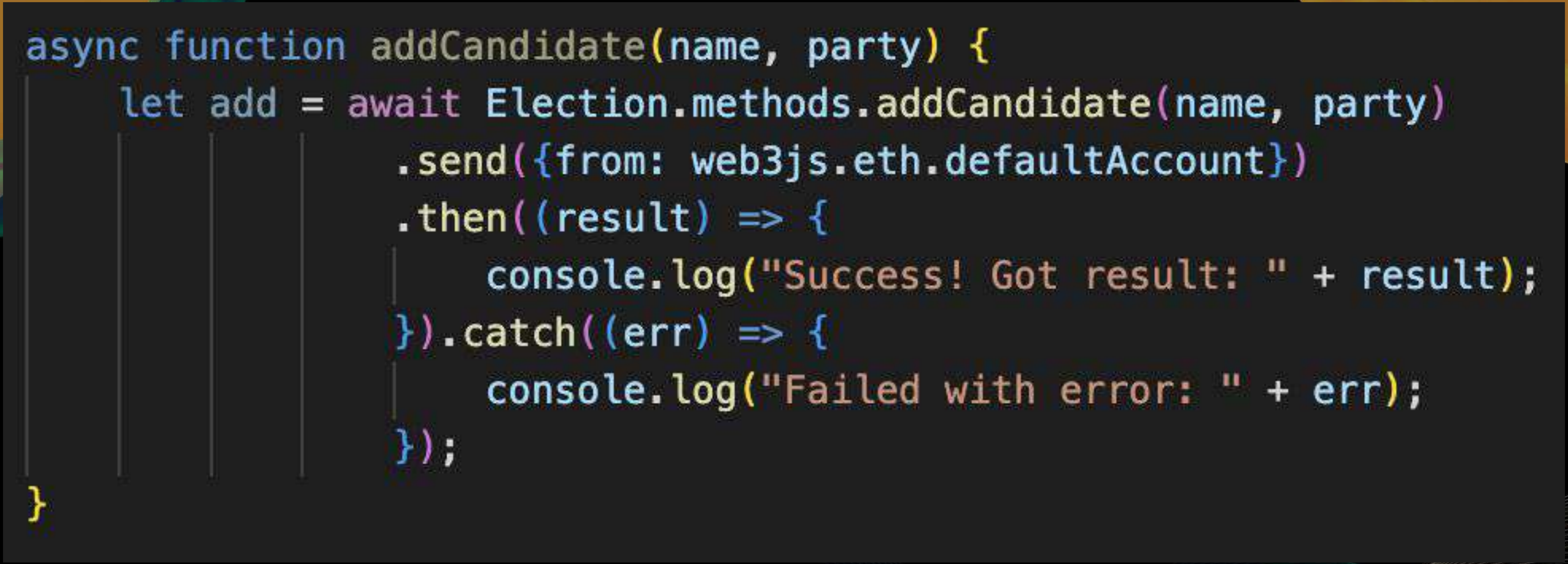
```
myContract.methods.myMethod([param1[, param2[, ...]]]).send(options[, callback])
```



1. `options` - `Object` **: The options used for sending.**
   - `from` - `String`: The address the transaction should be sent from.
   - `gasPrice` - `String` (optional): The gas price in wei to use for this transaction.
   - `gas` - `Number` (optional): The maximum gas provided for this transaction (gas limit).
   - `value` - `Number|String|BN|BigNumber` (optional): The value transferred for the transaction in wei.
   - `nonce` - `Number` (optional): the nonce number of transaction

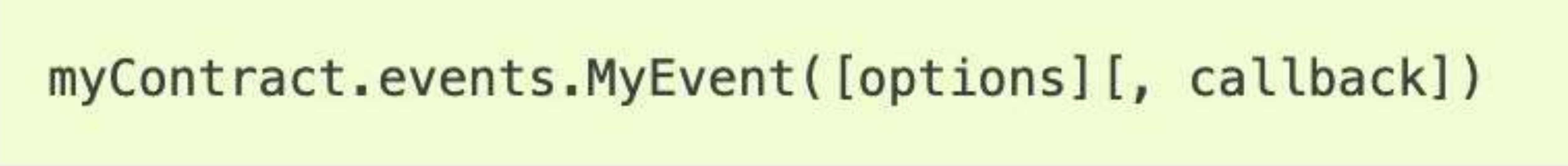
2. `callback` - `Function` (optional): This callback will be fired first with the "transactionHash", or with an error object as the first argument.

Write

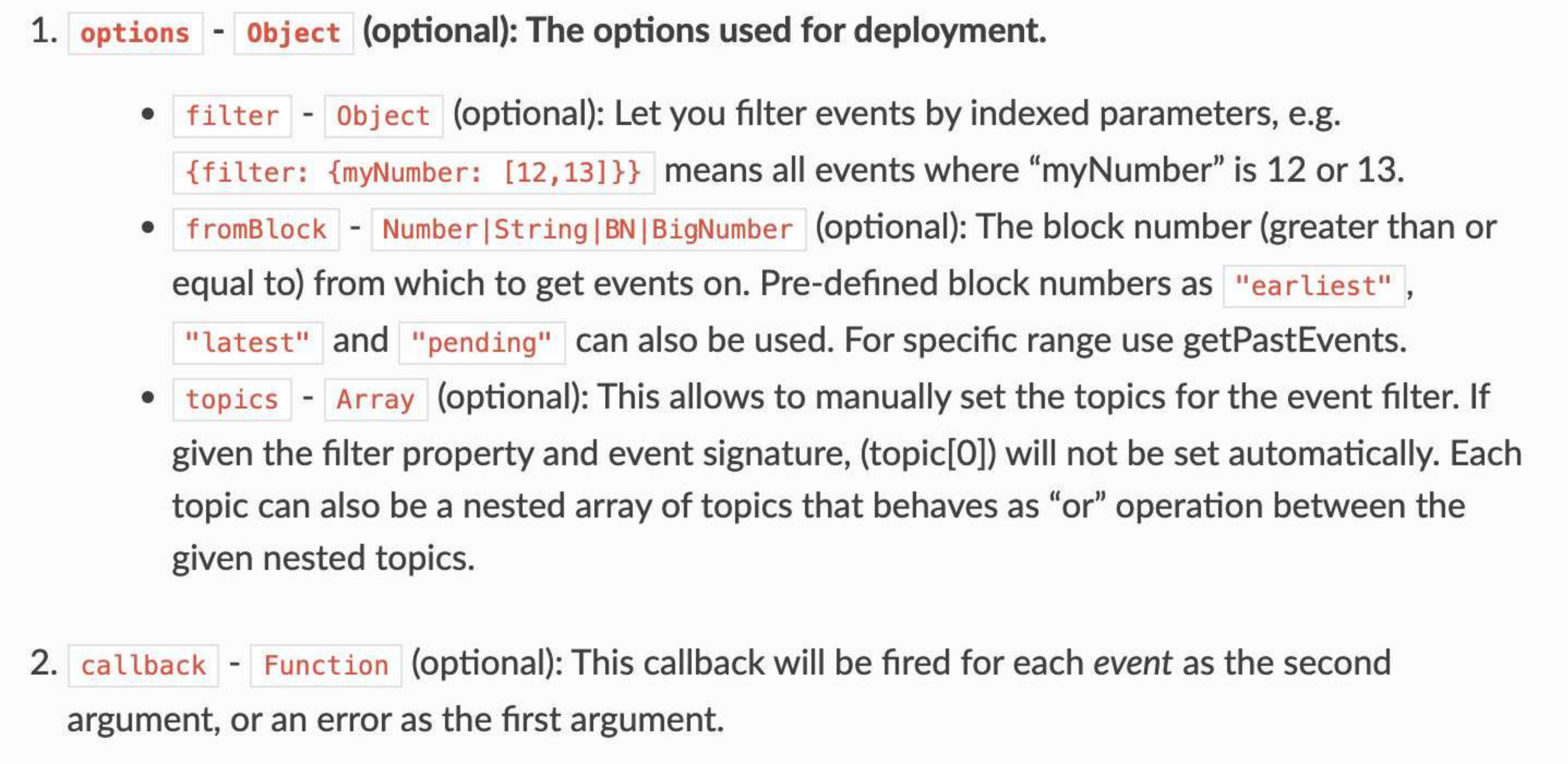


```
async function addCandidate(name, party) {
    let add = await Election.methods.addCandidate(name, party)
                .send({from: web3js.eth.defaultAccount})
                .then((result) => {
                    console.log("Success! Got result: " + result);
                }).catch((err) => {
                    console.log("Failed with error: " + err);
                });
}
```

Listening Event



```
myContract.events.MyEvent([options][, callback])
```

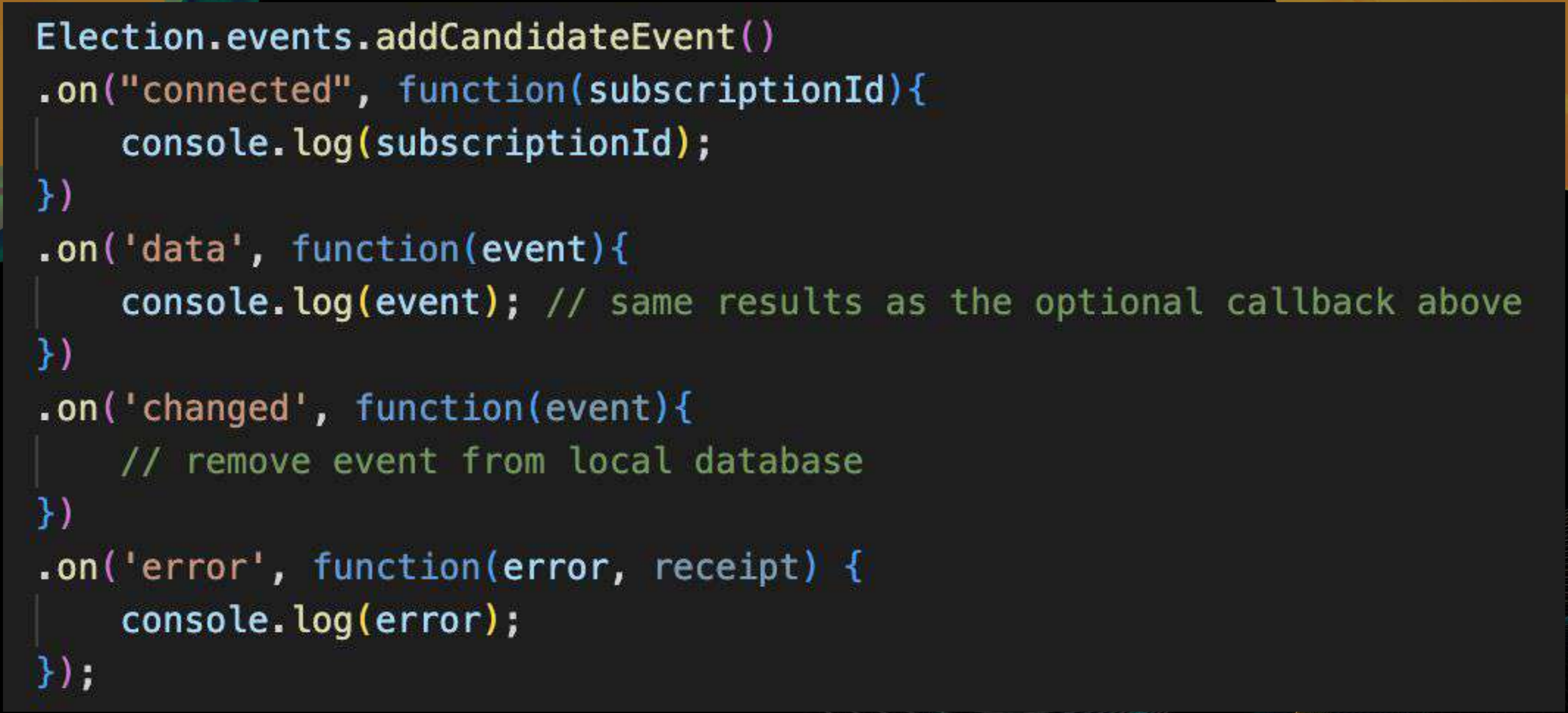


1. `options` - `Object` **(optional): The options used for deployment.**

   - `filter` - `Object` (optional): Let you filter events by indexed parameters, e.g. `{filter: {myNumber: [12,13]}}` means all events where “myNumber” is 12 or 13.
   - `fromBlock` - `Number|String|BN|BigNumber` (optional): The block number (greater than or equal to) from which to get events on. Pre-defined block numbers as `"earliest"`, `"latest"` and `"pending"` can also be used. For specific range use getPastEvents.
   - `topics` - `Array` (optional): This allows to manually set the topics for the event filter. If given the filter property and event signature, (topic[0]) will not be set automatically. Each topic can also be a nested array of topics that behaves as “or” operation between the given nested topics.

2. `callback` - `Function` (optional): This callback will be fired for each *event* as the second argument, or an error as the first argument.

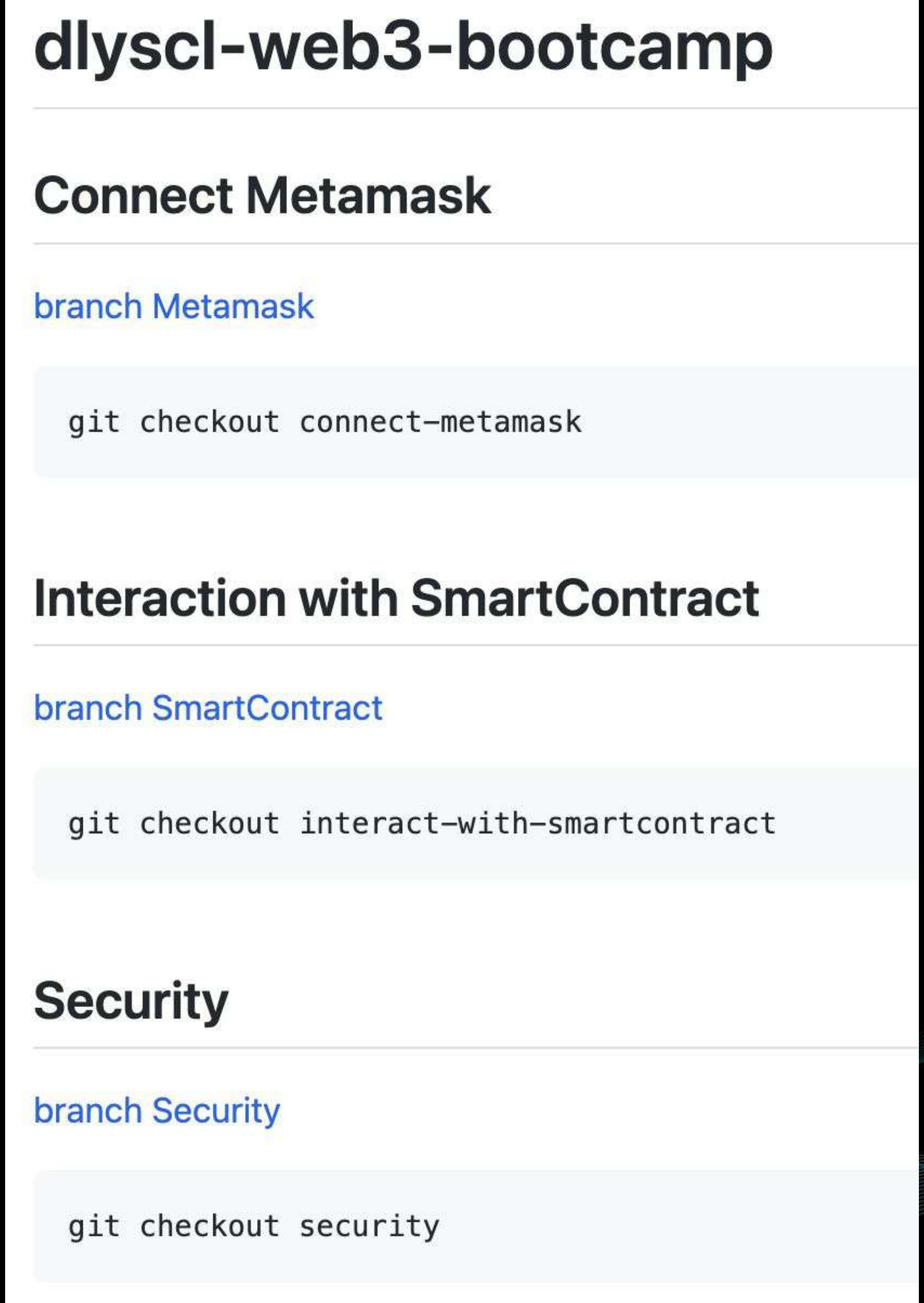
Listening Event



```
Election.events.addCandidateEvent()
.on("connected", function(subscriptionId){
    console.log(subscriptionId);
})
.on('data', function(event){
    console.log(event); // same results as the optional callback above
})
.on('changed', function(event){
    // remove event from local database
})
.on('error', function(error, receipt) {
    console.log(error);
});
```

Let’s Do it

https://github.com/sofianhw/dlyscl-web3-bootcamp



dlyscl-web3-bootcamp

Connect Metamask

branch Metamask

```
git checkout connect-metamask
```

Interaction with SmartContract

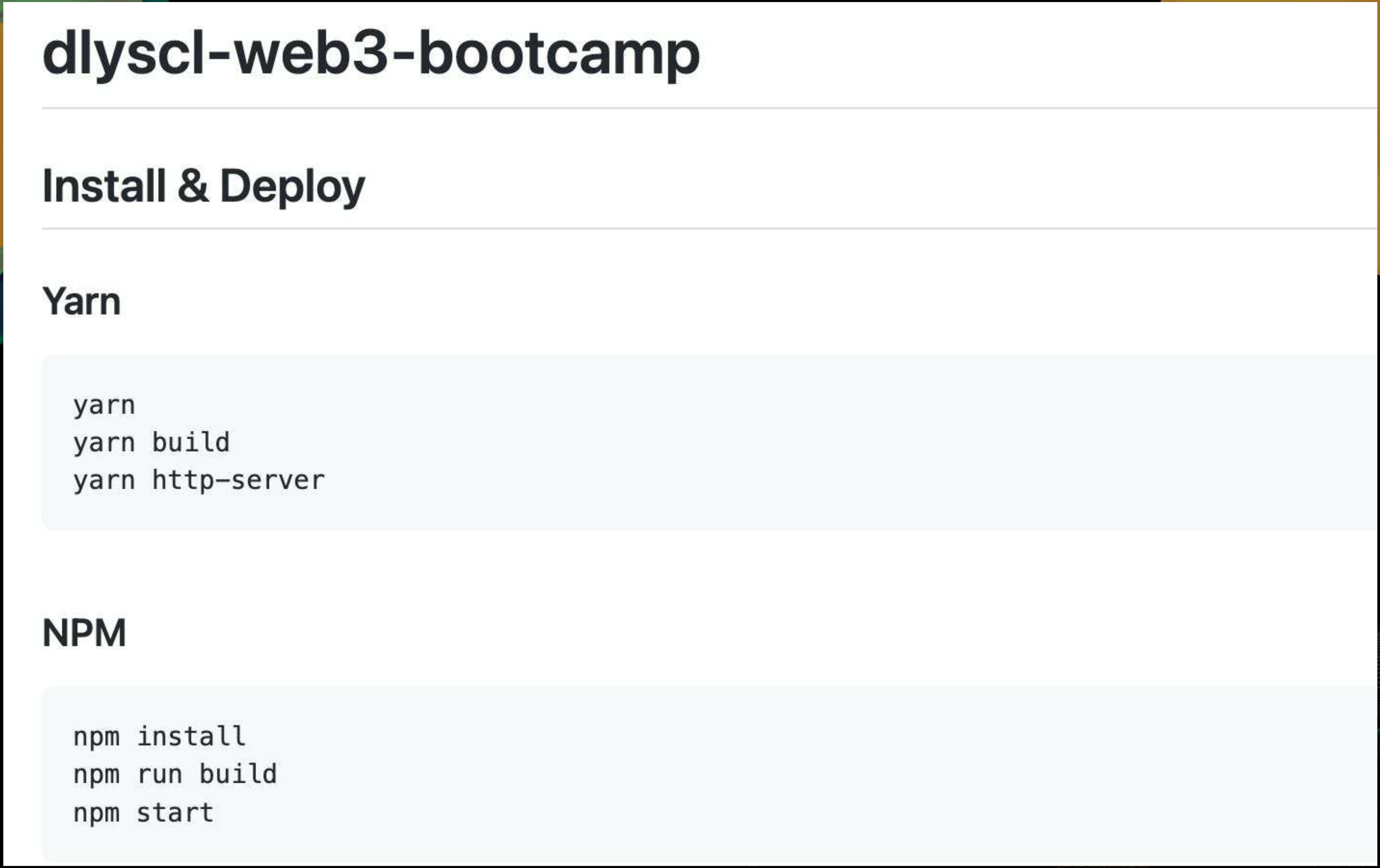
branch SmartContract

```
git checkout interact-with-smartcontract
```

Security

branch Security

```
git checkout security
```



dlyscl-web3-bootcamp

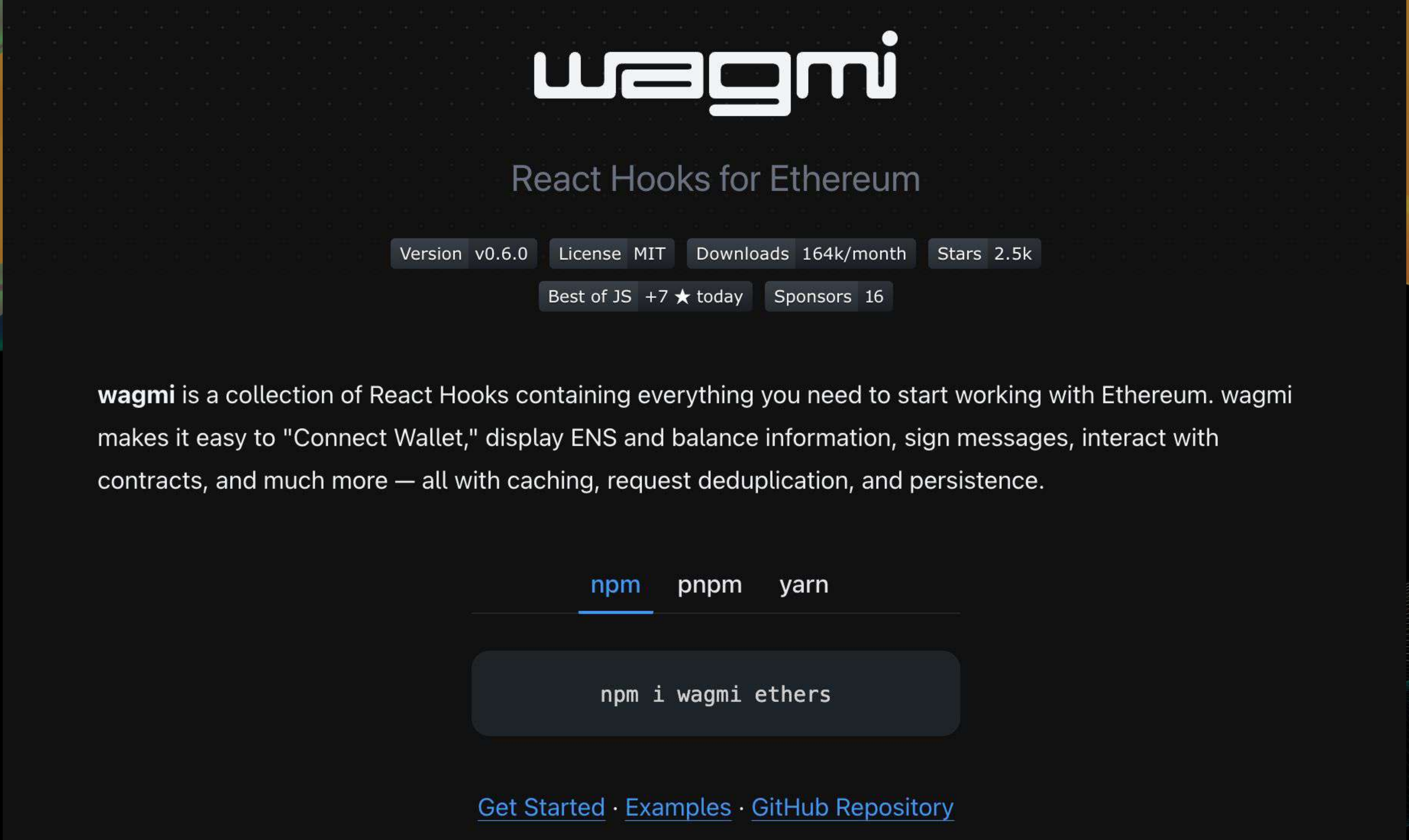
Install & Deploy

Yarn

```
yarn
yarn build
yarn http-server
```

NPM

```
npm install
npm run build
npm start
```



wagmi

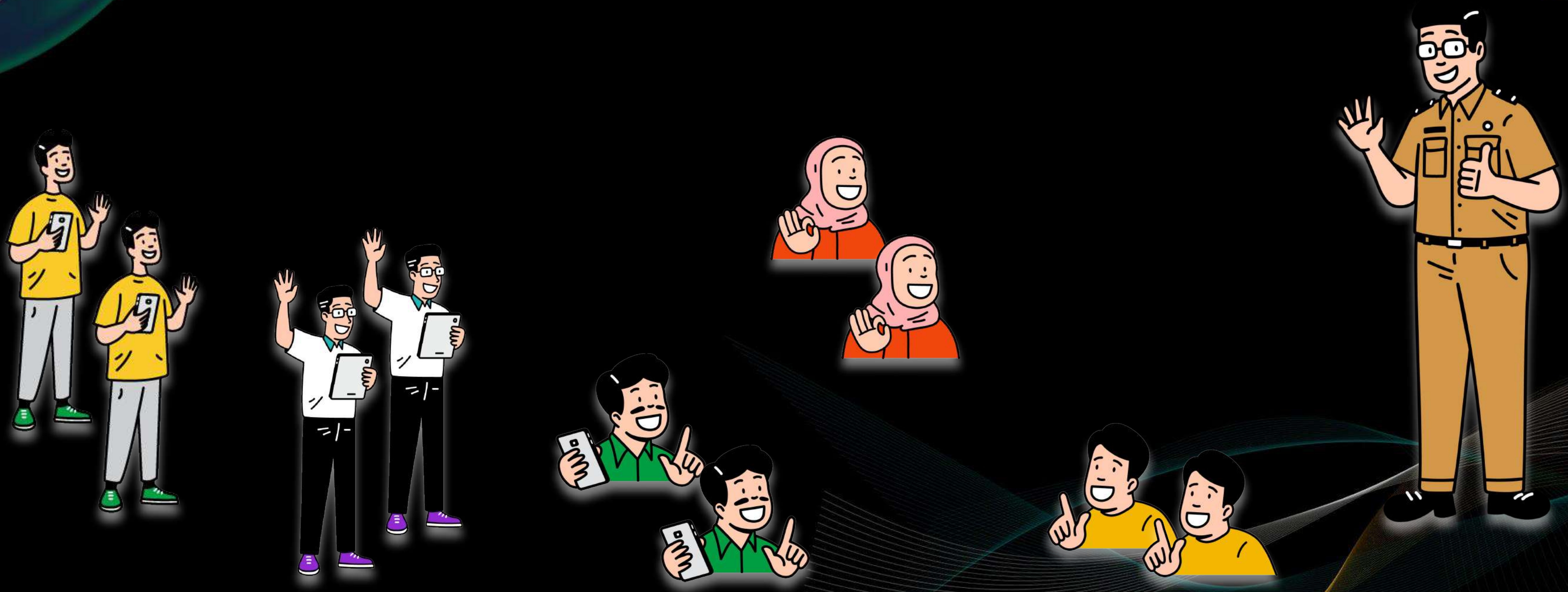
React Hooks for Ethereum

wagmi is a collection of React Hooks containing everything you need to start working with Ethereum. wagmi makes it easy to "Connect Wallet," display ENS and balance information, sign messages, interact with contracts, and much more — all with caching, request deduplication, and persistence.

```
npm i wagmi ethers
```

Get Started · Examples · GitHub Repository

Questions?



DailySocial id

Web3 Developer Bootcamp

Building the Builders of the Future

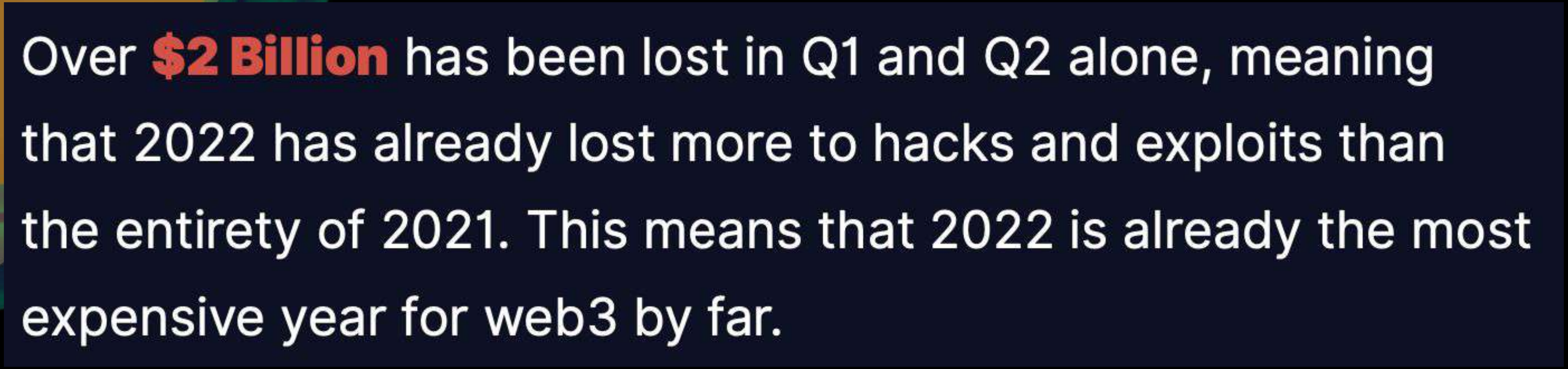
Introduction on Web3 CyberSecurity

@sofianhw

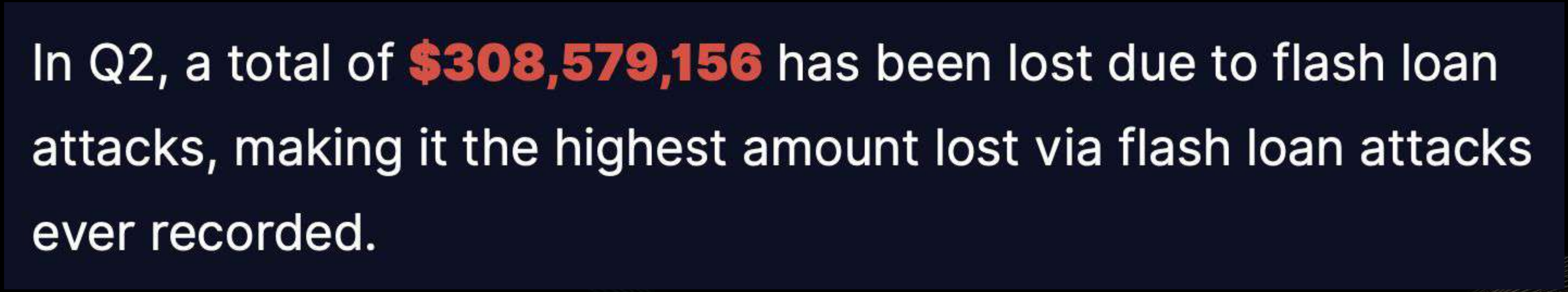
Contents

- Web3 Cybersecurity Overview
- Solidity Security
- Smart Contract Security
- Tools

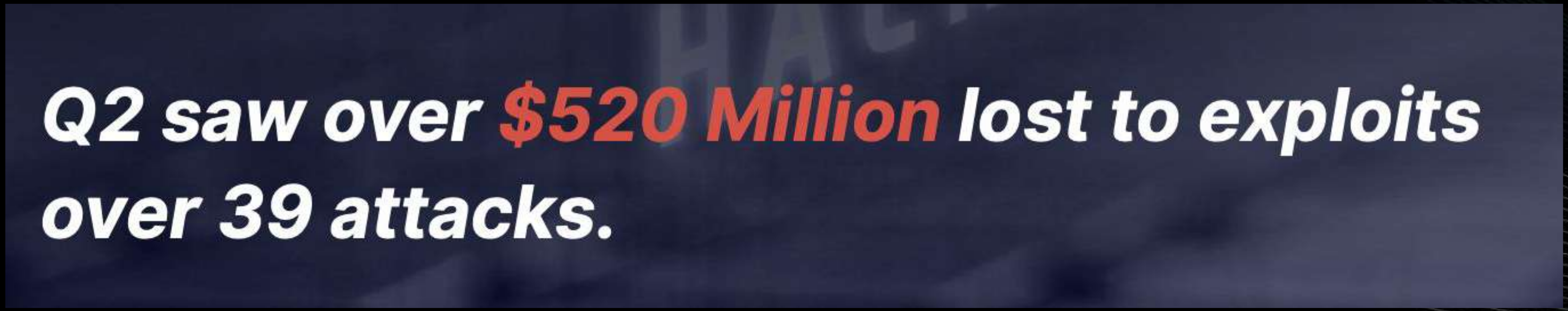
Web3 Security Overview



Over **$2 Billion** has been lost in Q1 and Q2 alone, meaning that 2022 has already lost more to hacks and exploits than the entirety of 2021. This means that 2022 is already the most expensive year for web3 by far.



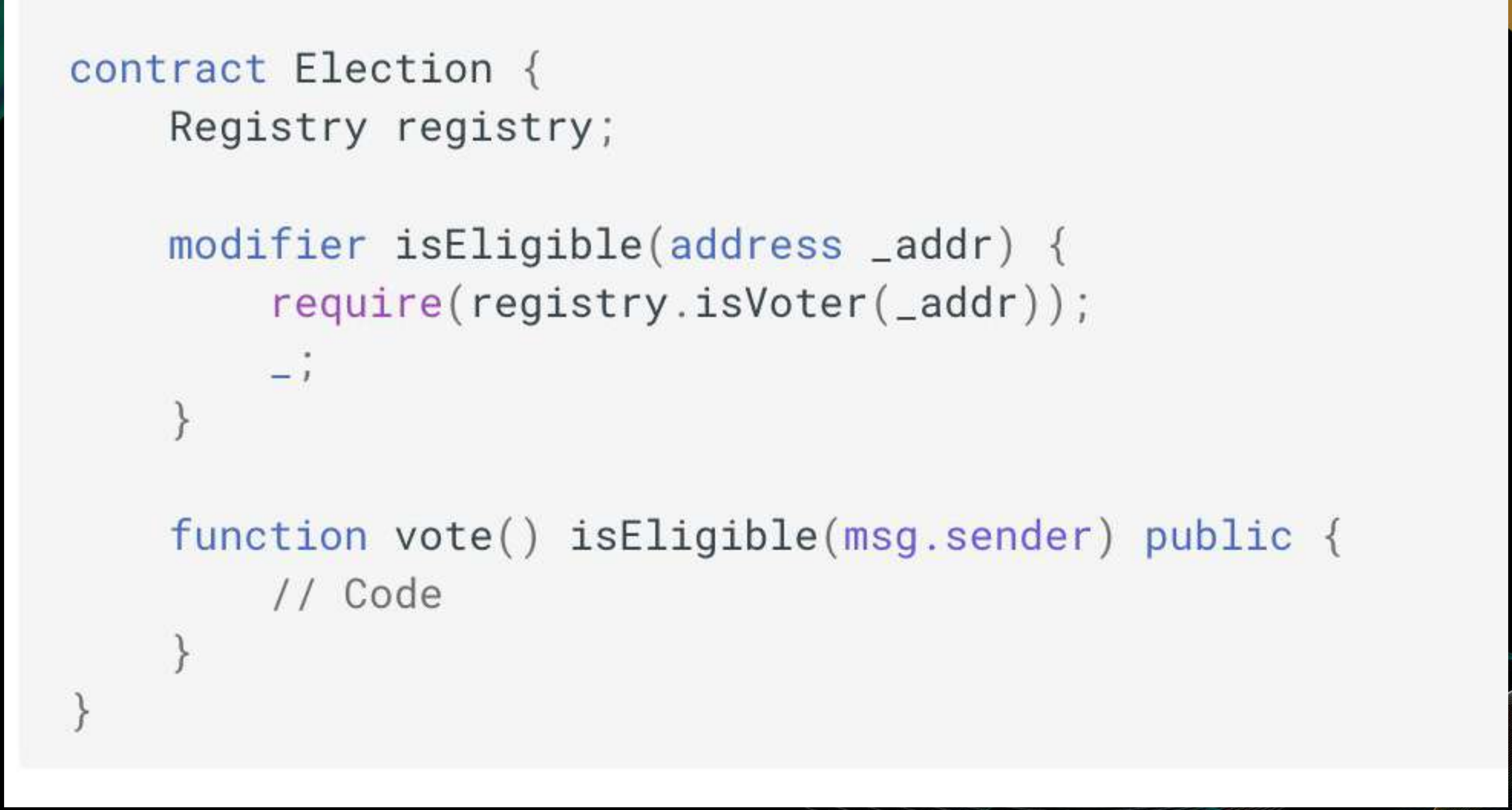
In Q2, a total of **$308,579,156** has been lost due to flash loan attacks, making it the highest amount lost via flash loan attacks ever recorded.



Q2 saw over $520 Million lost to exploits over 39 attacks.

Solidity Security

Modifier as Guard



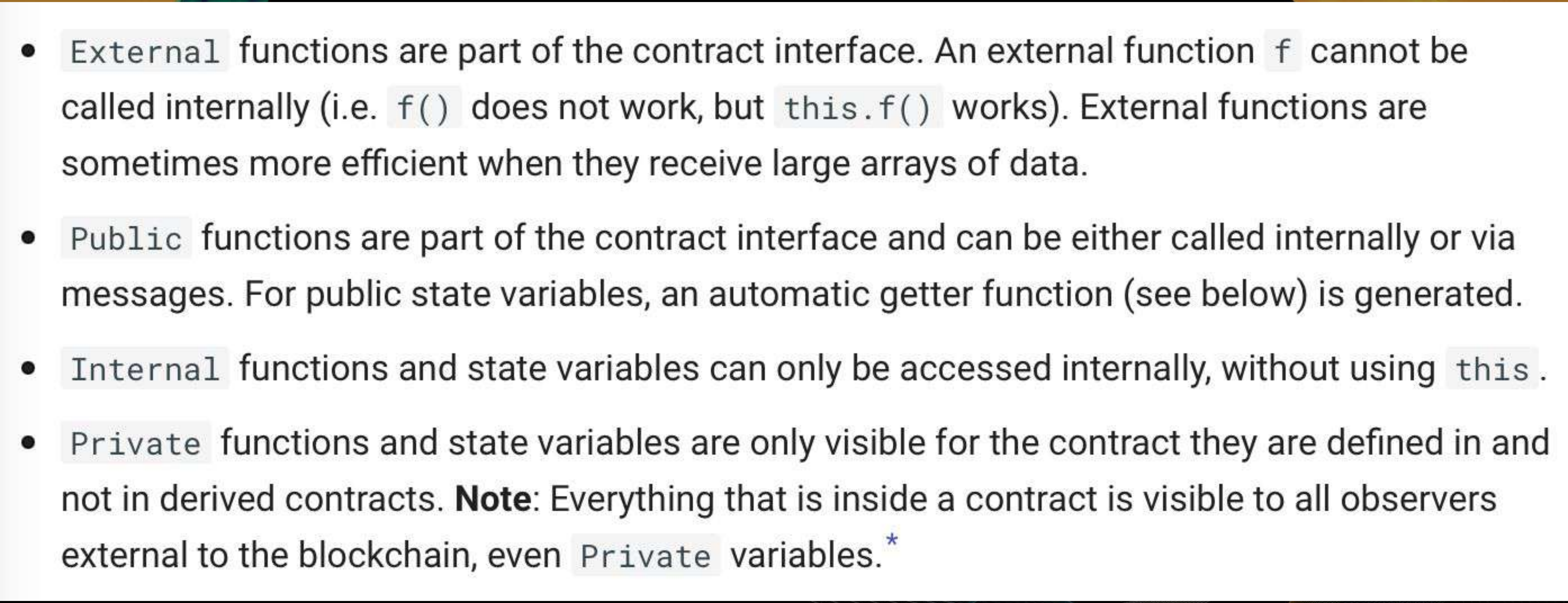
```
contract Election {
    Registry registry;

    modifier isEligible(address _addr) {
        require(registry.isVoter(_addr));
        _;
    }

    function vote() isEligible(msg.sender) public {
        // Code
    }
}
```

Solidity Security

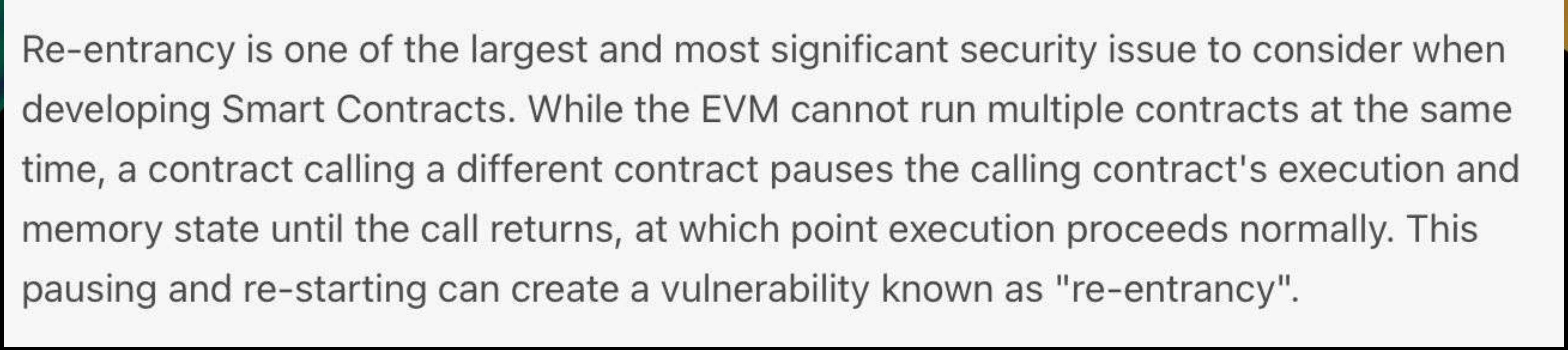
Visibility



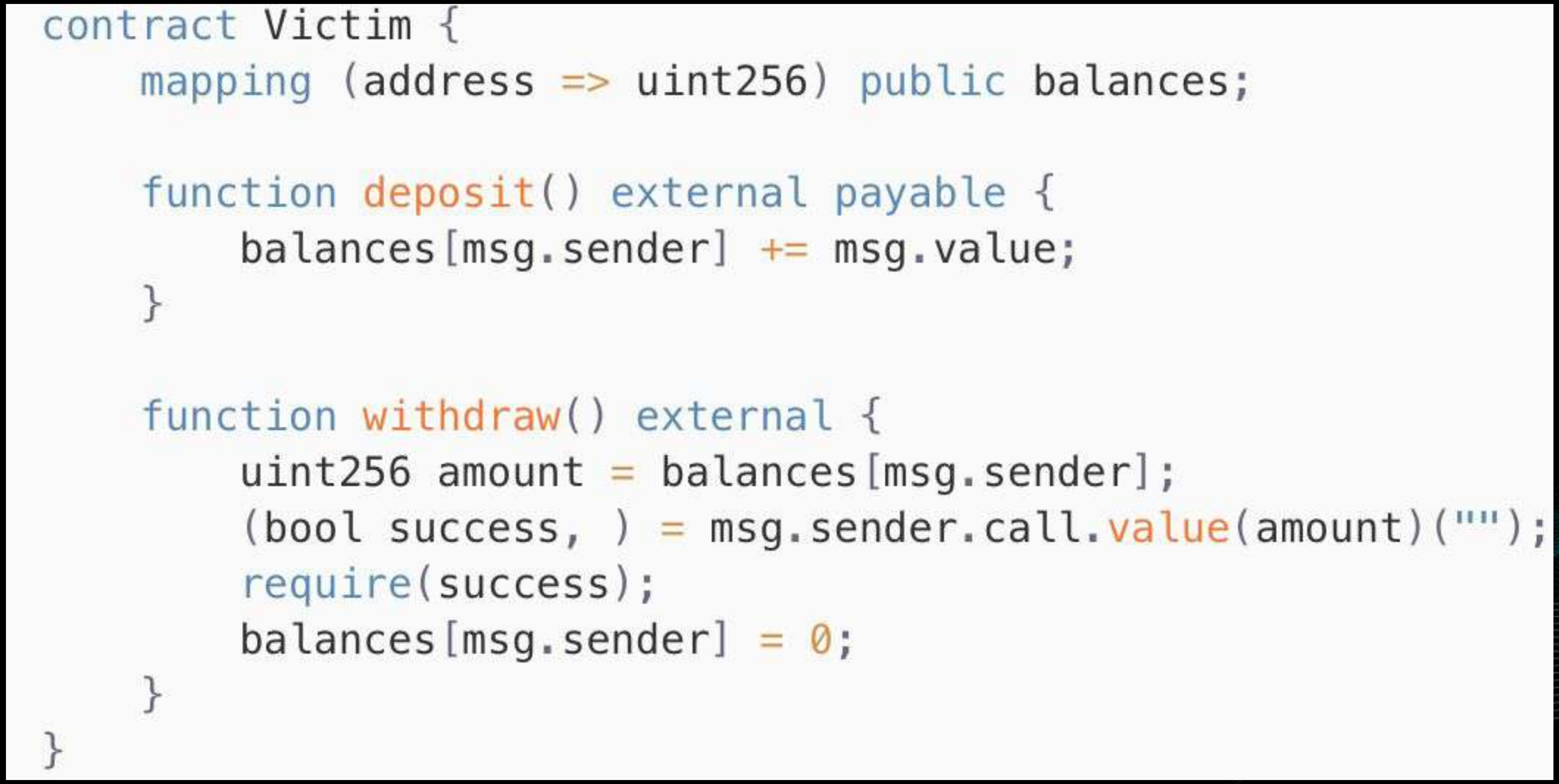
- `External` functions are part of the contract interface. An external function `f` cannot be called internally (i.e. `f()` does not work, but `this.f()` works). External functions are sometimes more efficient when they receive large arrays of data.
- `Public` functions are part of the contract interface and can be either called internally or via messages. For public state variables, an automatic getter function (see below) is generated.
- `Internal` functions and state variables can only be accessed internally, without using `this`.
- `Private` functions and state variables are only visible for the contract they are defined in and not in derived contracts. **Note**: Everything that is inside a contract is visible to all observers external to the blockchain, even `Private` variables.*

Smart Contract Security

Re-entrancy



Re-entrancy is one of the largest and most significant security issue to consider when developing Smart Contracts. While the EVM cannot run multiple contracts at the same time, a contract calling a different contract pauses the calling contract's execution and memory state until the call returns, at which point execution proceeds normally. This pausing and re-starting can create a vulnerability known as "re-entrancy".



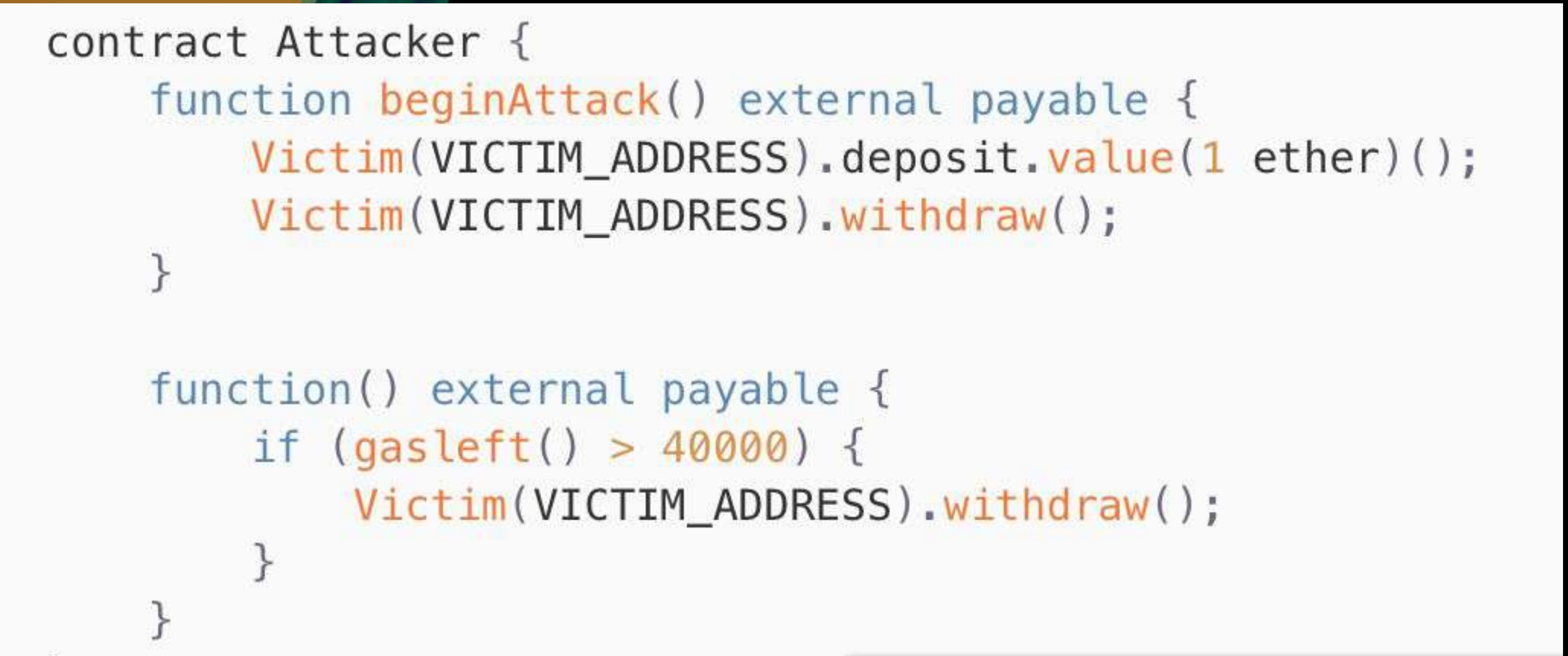
```
contract Victim {
    mapping (address => uint256) public balances;

    function deposit() external payable {
        balances[msg.sender] += msg.value;
    }

    function withdraw() external {
        uint256 amount = balances[msg.sender];
        (bool success, ) = msg.sender.call.value(amount)("");
        require(success);
        balances[msg.sender] = 0;
    }
}
```

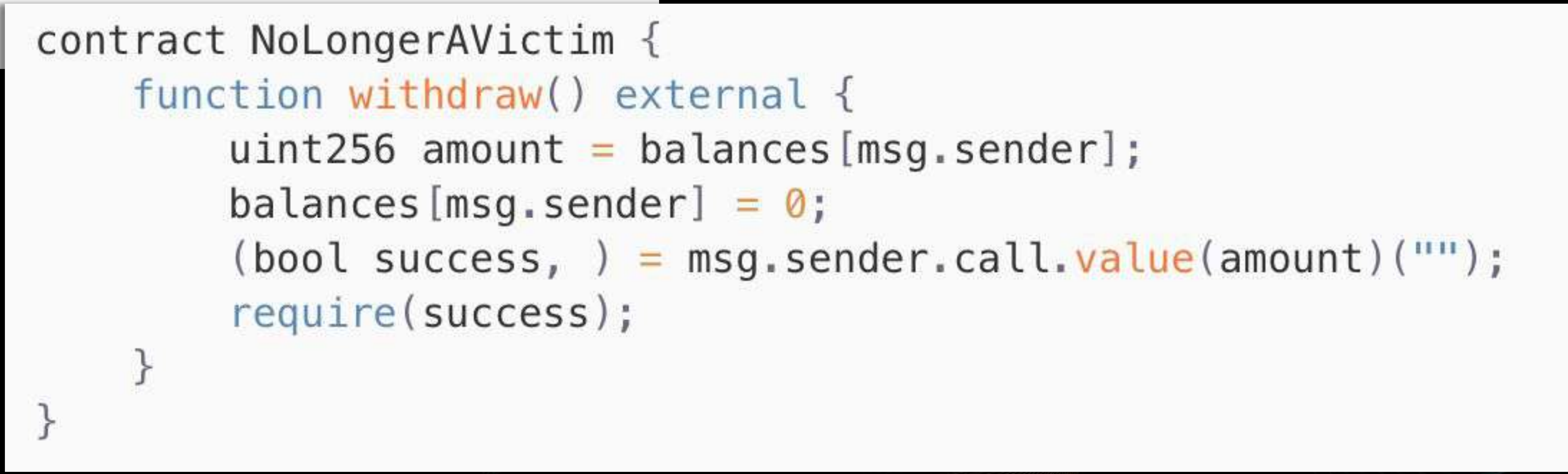
Smart Contract Security

Re-entrancy



```
contract Attacker {
    function beginAttack() external payable {
        Victim(VICTIM_ADDRESS).deposit.value(1 ether)();
        Victim(VICTIM_ADDRESS).withdraw();
    }

    function() external payable {
        if (gasleft() > 40000) {
            Victim(VICTIM_ADDRESS).withdraw();
        }
    }
}
```



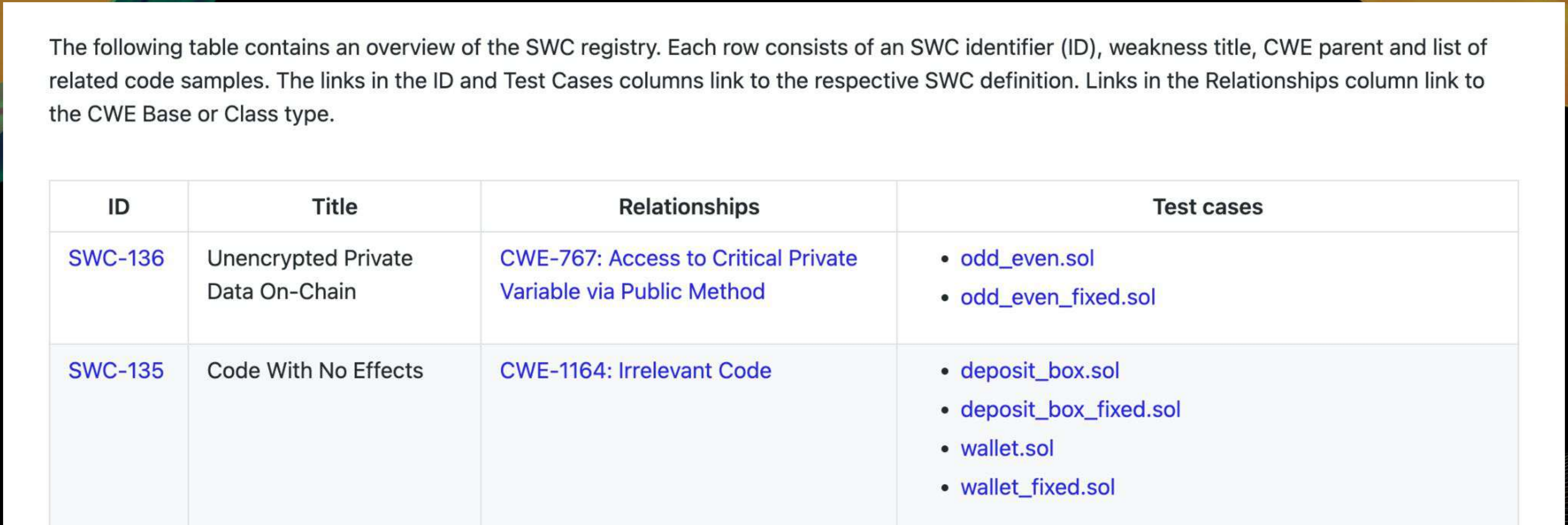
```
contract NoLongerAVictim {
    function withdraw() external {
        uint256 amount = balances[msg.sender];
        balances[msg.sender] = 0;
        (bool success, ) = msg.sender.call.value(amount)("");
        require(success);
    }
}
```

Tools

- **SmartContract Weakness Registry**
- **OpenSource Tools**
- **Paid Tools**

Smart Contract Security

SmartContract Weakness Registry

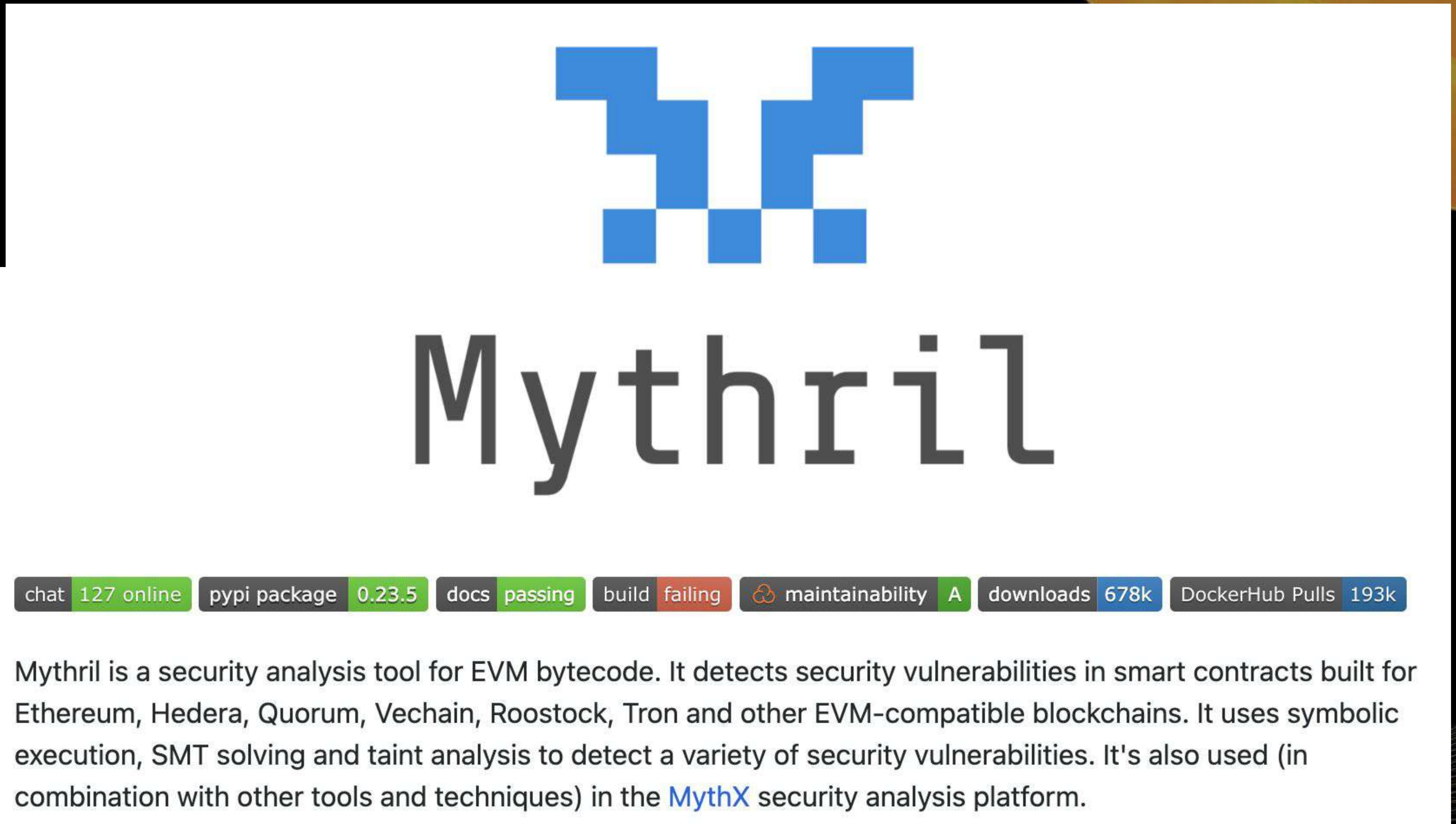


The following table contains an overview of the SWC registry. Each row consists of an SWC identifier (ID), weakness title, CWE parent and list of related code samples. The links in the ID and Test Cases columns link to the respective SWC definition. Links in the Relationships column link to the CWE Base or Class type.

ID	Title	Relationships	Test cases
SWC-136	Unencrypted Private Data On-Chain	CWE-767: Access to Critical Private Variable via Public Method	• odd_even.sol • odd_even_fixed.sol
SWC-135	Code With No Effects	CWE-1164: Irrelevant Code	• deposit_box.sol • deposit_box_fixed.sol • wallet.sol • wallet_fixed.sol

Smart Contract Security

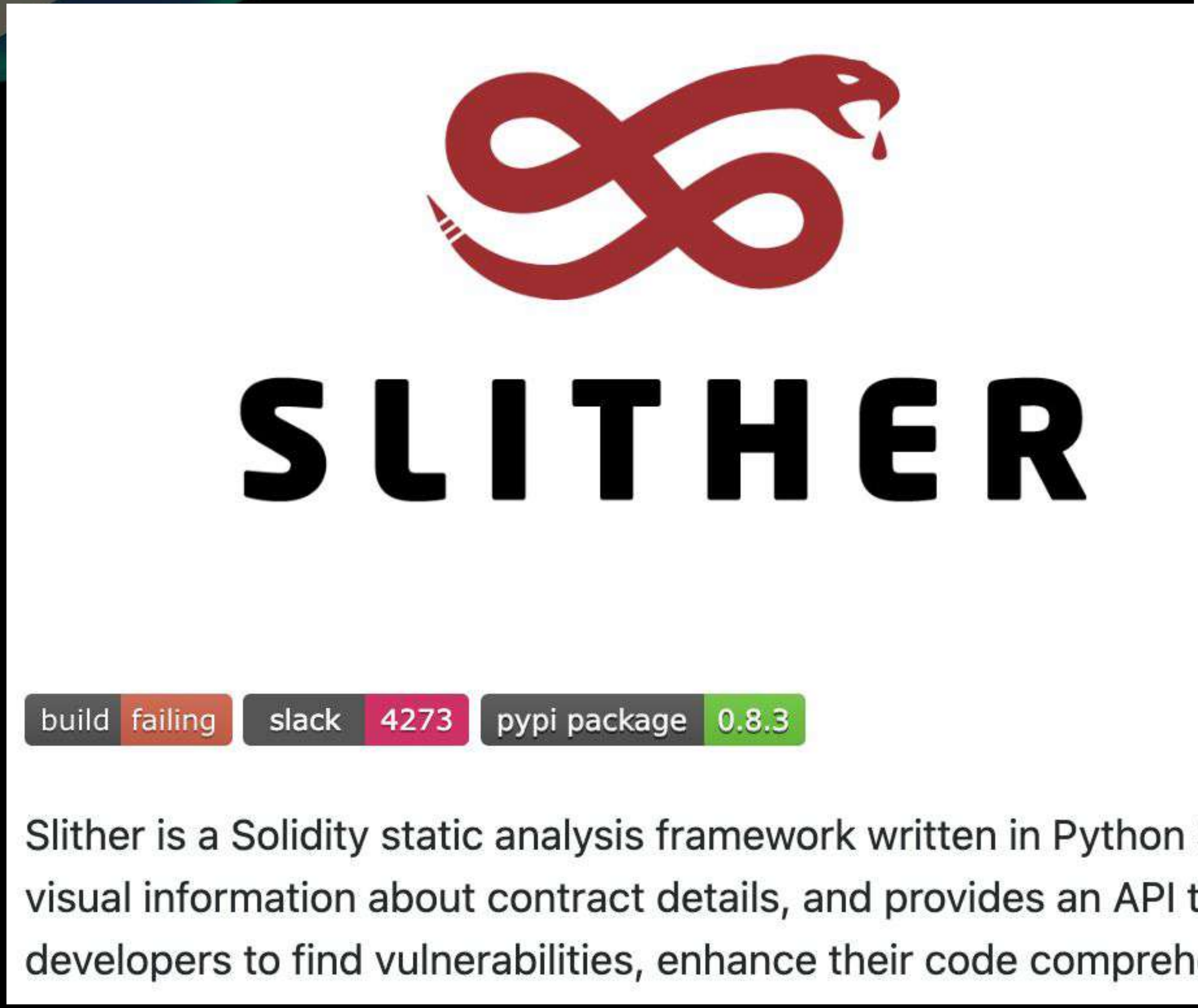
Open Source Tools



Mythril

chat 127 online | pypi package 0.23.5 | docs passing | build failing | maintainability A | downloads 678k | DockerHub Pulls 193k

Mythril is a security analysis tool for EVM bytecode. It detects security vulnerabilities in smart contracts built for Ethereum, Hedera, Quorum, Vechain, Roostock, Tron and other EVM-compatible blockchains. It uses symbolic execution, SMT solving and taint analysis to detect a variety of security vulnerabilities. It's also used (in combination with other tools and techniques) in the MythX security analysis platform.



SLITHER

build failing | slack 4273 | pypi package 0.8.3

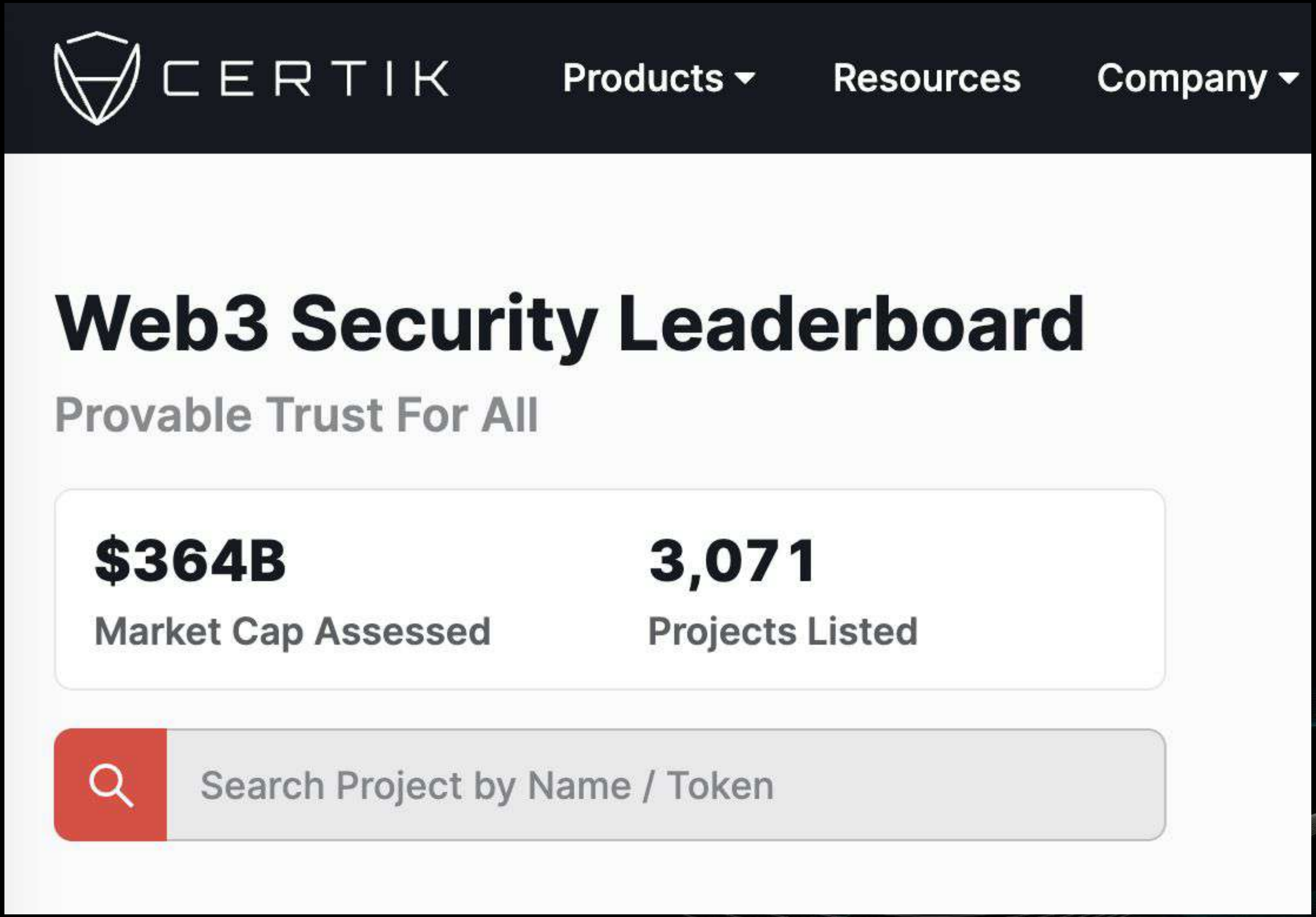
Slither is a Solidity static analysis framework written in Python 3. It runs a suite of vulnerability detectors, prints visual information about contract details, and provides an API to easily write custom analyses. Slither enables developers to find vulnerabilities, enhance their code comprehension, and quickly prototype custom analyses.

Let’s Do it

https://github.com/sofianhw/dlyscl-web3-bootcamp

Smart Contract Security

Paid Tools



Web3 Security Leaderboard

Provable Trust For All

$364B

Market Cap Assessed

3,071

Projects Listed

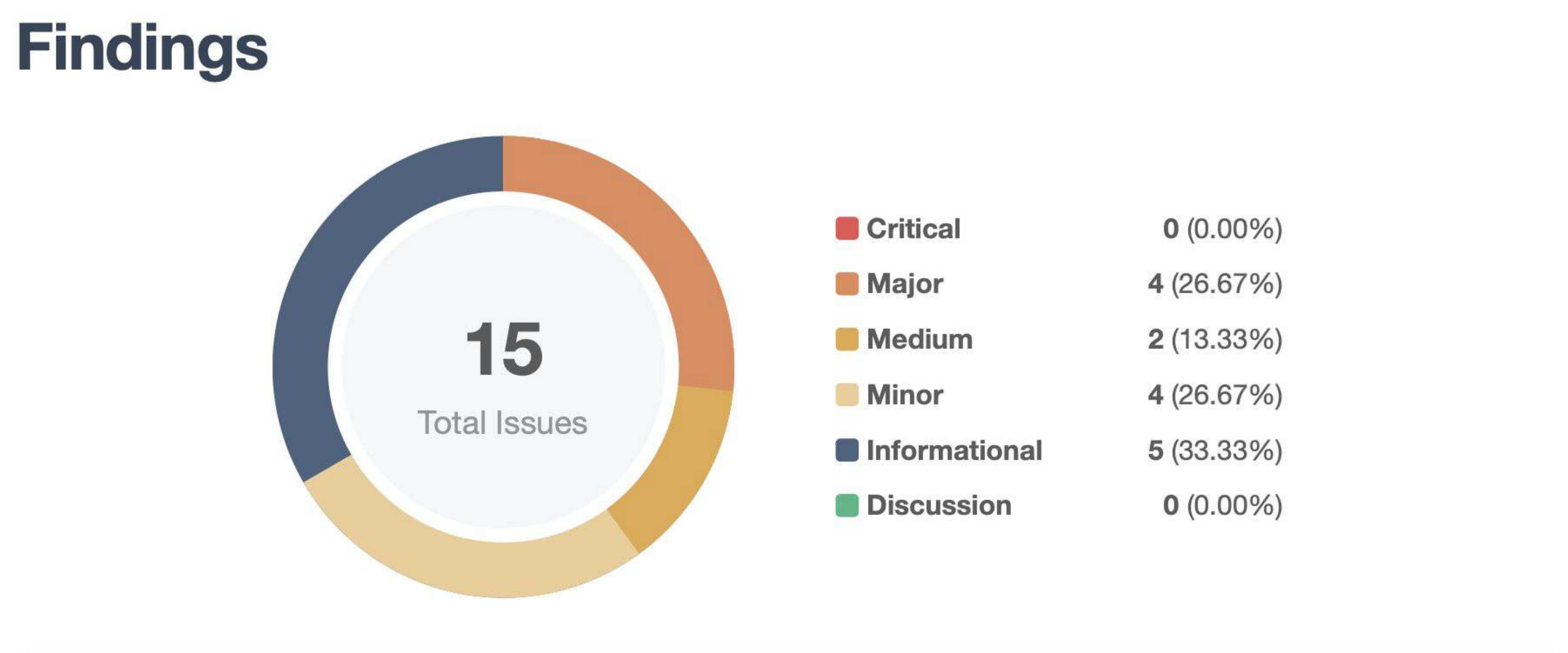
Smart Contract Security

Sample Certik Report

https://drive.google.com/file/d/1qzOVQG4H_g3Q_EStVt7RSMrquCih7QqI/view?usp=sharing

Smart Contract Security

Paid Tools

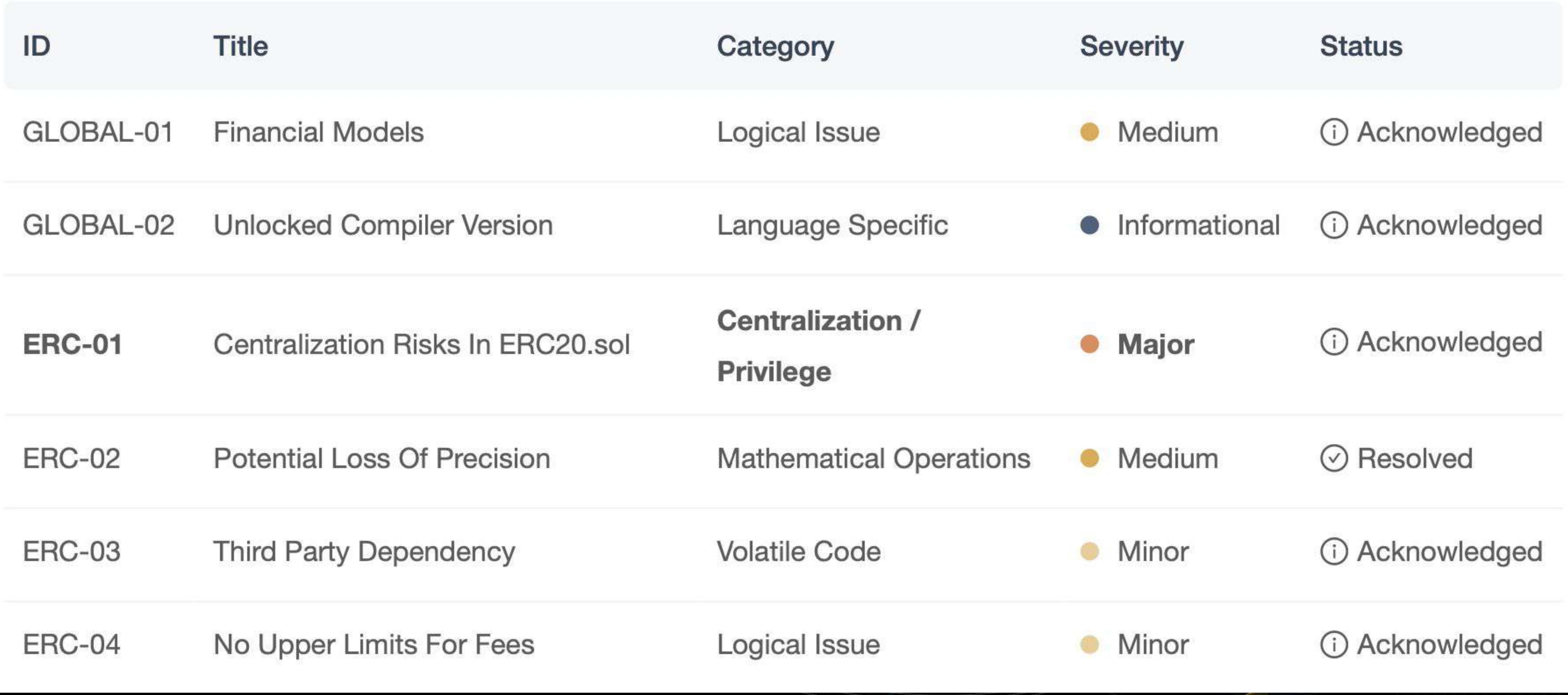


Findings

🟥	Critical	0 (0.00%)
🟧	Major	4 (26.67%)
🟨	Medium	2 (13.33%)
▨	Minor	4 (26.67%)
🟦	Informational	5 (33.33%)
🟩	Discussion	0 (0.00%)

Smart Contract Security

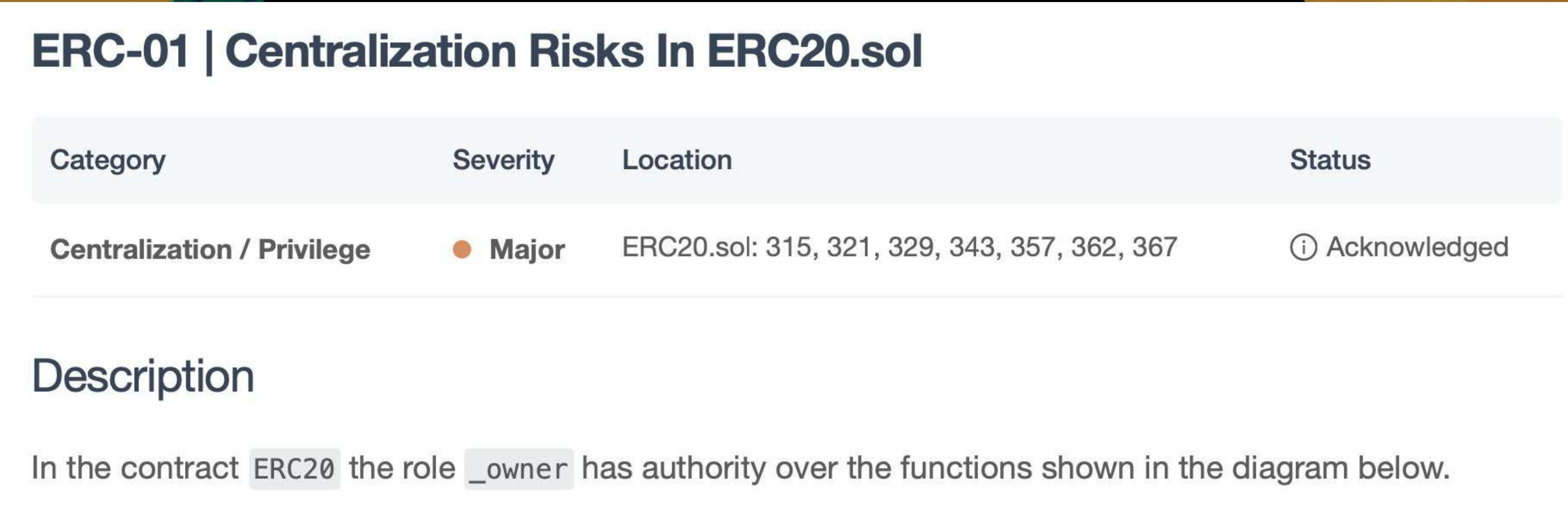
Paid Tools



ID	Title	Category	Severity	Status
GLOBAL-01	Financial Models	Logical Issue	● Medium	ⓘ Acknowledged
GLOBAL-02	Unlocked Compiler Version	Language Specific	● Informational	ⓘ Acknowledged
ERC-01	Centralization Risks In ERC20.sol	**Centralization / Privilege**	● **Major**	ⓘ Acknowledged
ERC-02	Potential Loss Of Precision	Mathematical Operations	● Medium	✓ Resolved
ERC-03	Third Party Dependency	Volatile Code	● Minor	ⓘ Acknowledged
ERC-04	No Upper Limits For Fees	Logical Issue	● Minor	ⓘ Acknowledged

Smart Contract Security

Paid Tools



ERC-01 | Centralization Risks In ERC20.sol

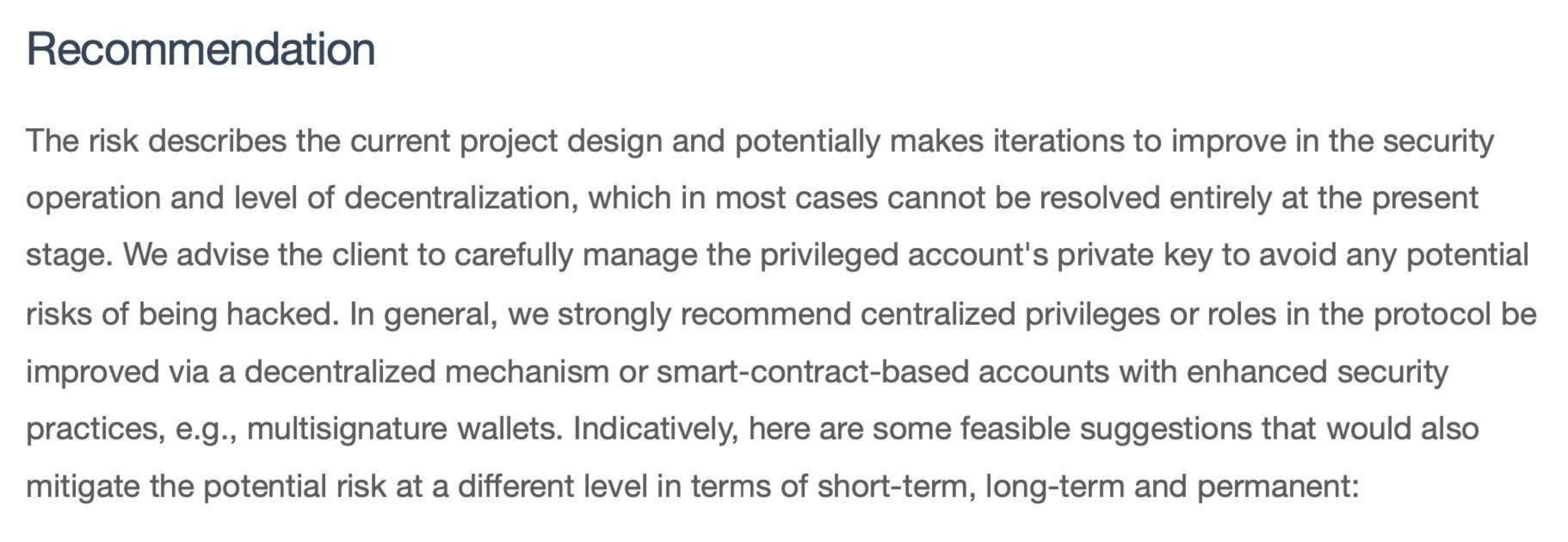
Category	Severity	Location	Status
Centralization / Privilege	● **Major**	ERC20.sol: 315, 321, 329, 343, 357, 362, 367	ⓘ Acknowledged

Description

In the contract `ERC20` the role `_owner` has authority over the functions shown in the diagram below.

Smart Contract Security

Paid Tools

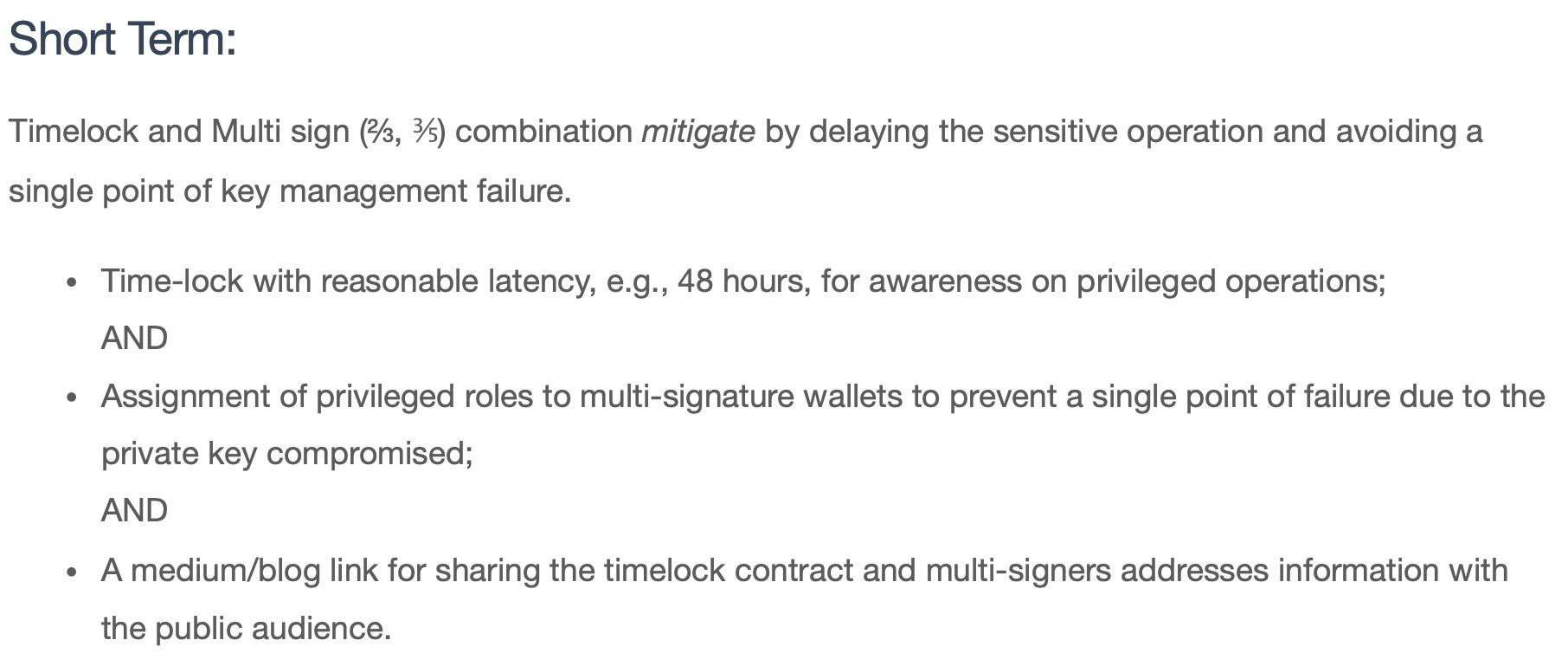


Recommendation

The risk describes the current project design and potentially makes iterations to improve in the security operation and level of decentralization, which in most cases cannot be resolved entirely at the present stage. We advise the client to carefully manage the privileged account's private key to avoid any potential risks of being hacked. In general, we strongly recommend centralized privileges or roles in the protocol be improved via a decentralized mechanism or smart-contract-based accounts with enhanced security practices, e.g., multisignature wallets. Indicatively, here are some feasible suggestions that would also mitigate the potential risk at a different level in terms of short-term, long-term and permanent:

Smart Contract Security

Paid Tools



Short Term:

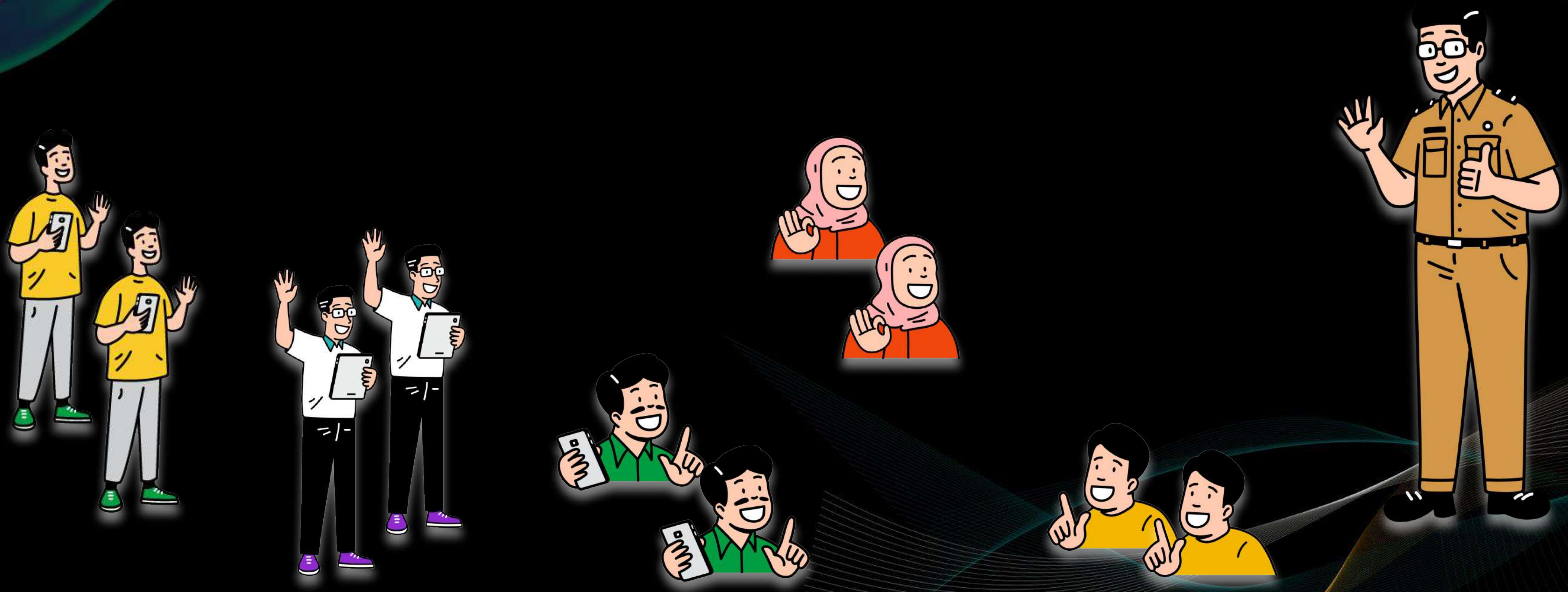
Timelock and Multi sign (⅔, ⅗) combination *mitigate* by delaying the sensitive operation and avoiding a single point of key management failure.

- Time-lock with reasonable latency, e.g., 48 hours, for awareness on privileged operations;

  AND
- Assignment of privileged roles to multi-signature wallets to prevent a single point of failure due to the private key compromised;

  AND
- A medium/blog link for sharing the timelock contract and multi-signers addresses information with the public audience.

Questions?



DailySocial id

Web3 Developer Bootcamp

Building the Builders of the Future

Thanks!

@sofianhw