dΞth

Ethereum dev tools

AGENDA

01

Who? & Why?

02

Tools available

03

Tools in progress

04

Q&A

OUR TEAM

Krzysztof kaczor

@krzKaczor

Leonid Logvinov

@Logvinov_Leon

Maciej Kukiełka

@KukielkaMaciej

AVAILABLE TOOLS

TypeChain

TypeScript bindings for Ethereum smart contracts

Deth Code

View source of deployed Ethereum smart contracts in VS Code

Eth SDK

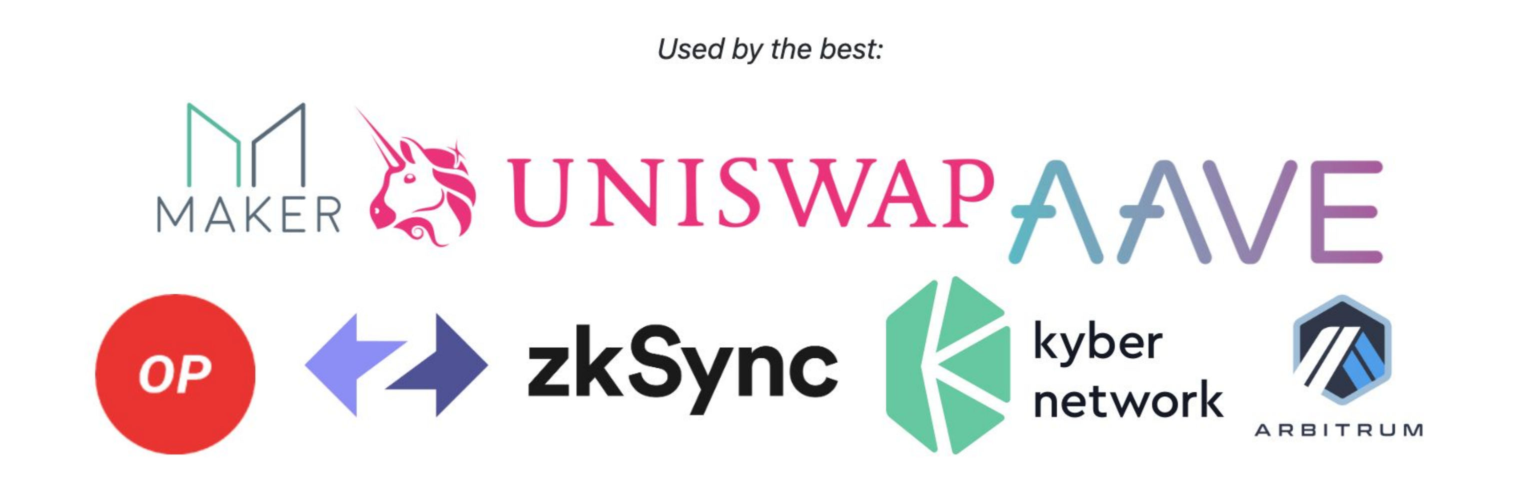
Type-safe, lightweight SDKs for Ethereum smart contracts

earl.js

Ergonomic, modern and type-safe assertion library for TypeScript

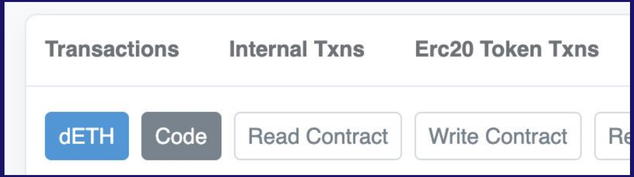
TypeChain

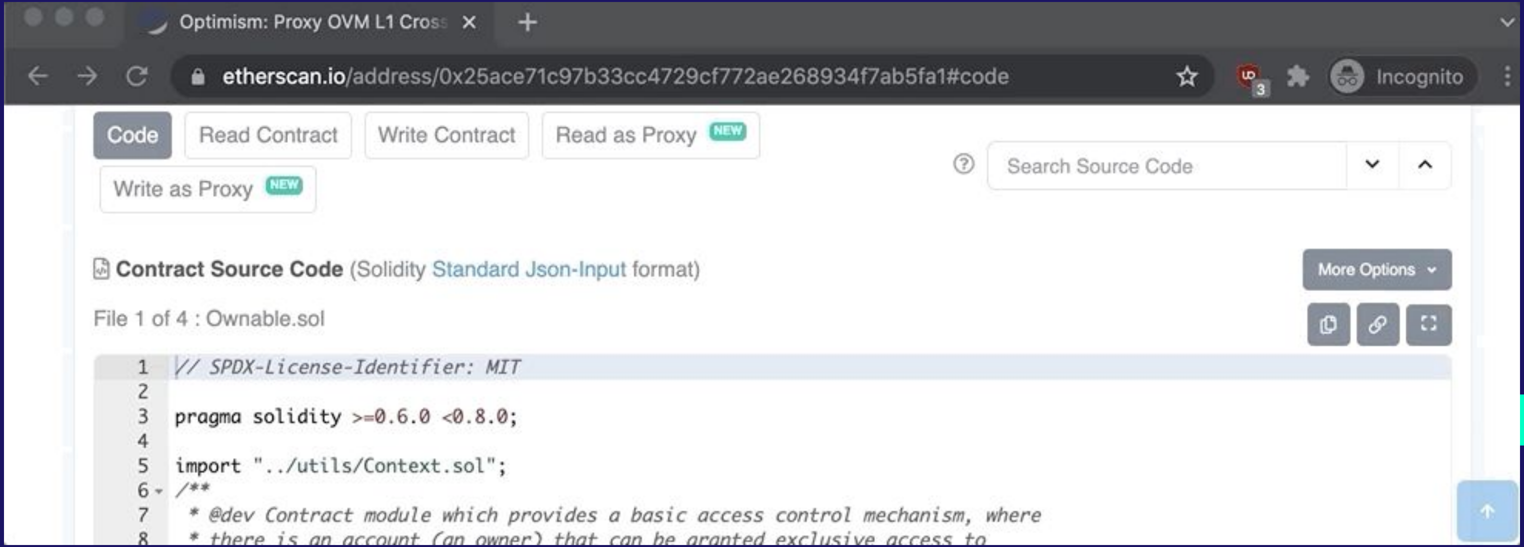
Starred 1.8k



Used by the best:

Deth Code



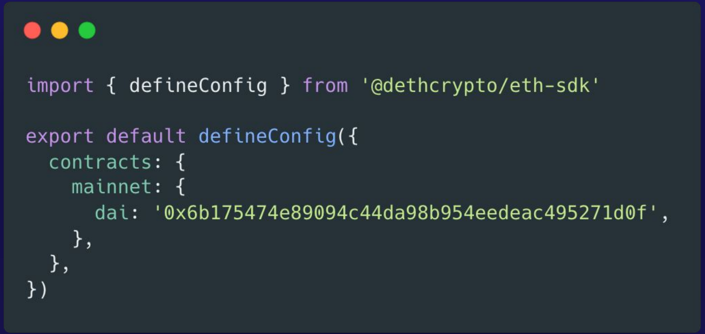


Contract Source Code (Solidity Standard Json-Input format)

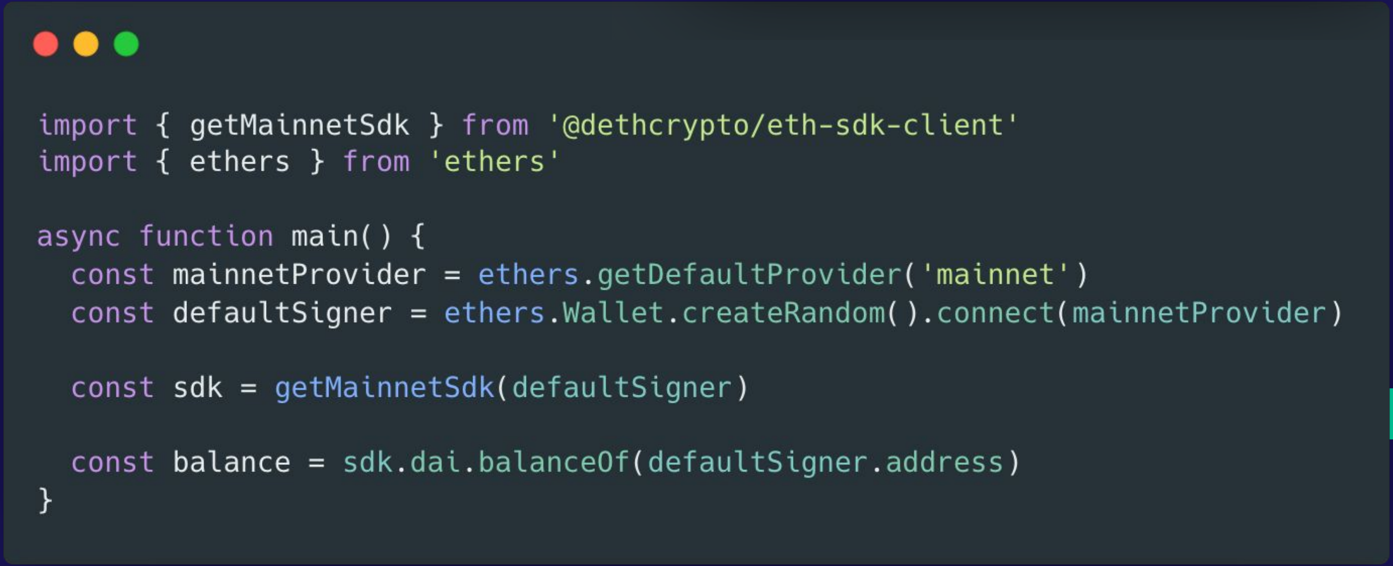
File 1 of 4 : Ownable.sol

```
// SPDX-License-Identifier: MIT

pragma solidity >=0.6.0 <0.8.0;

import "../utils/Context.sol";
/**
 * @dev Contract module which provides a basic access control mechanism, where
 * there is an account (an owner) that can be granted exclusive access to
```

Eth SDK

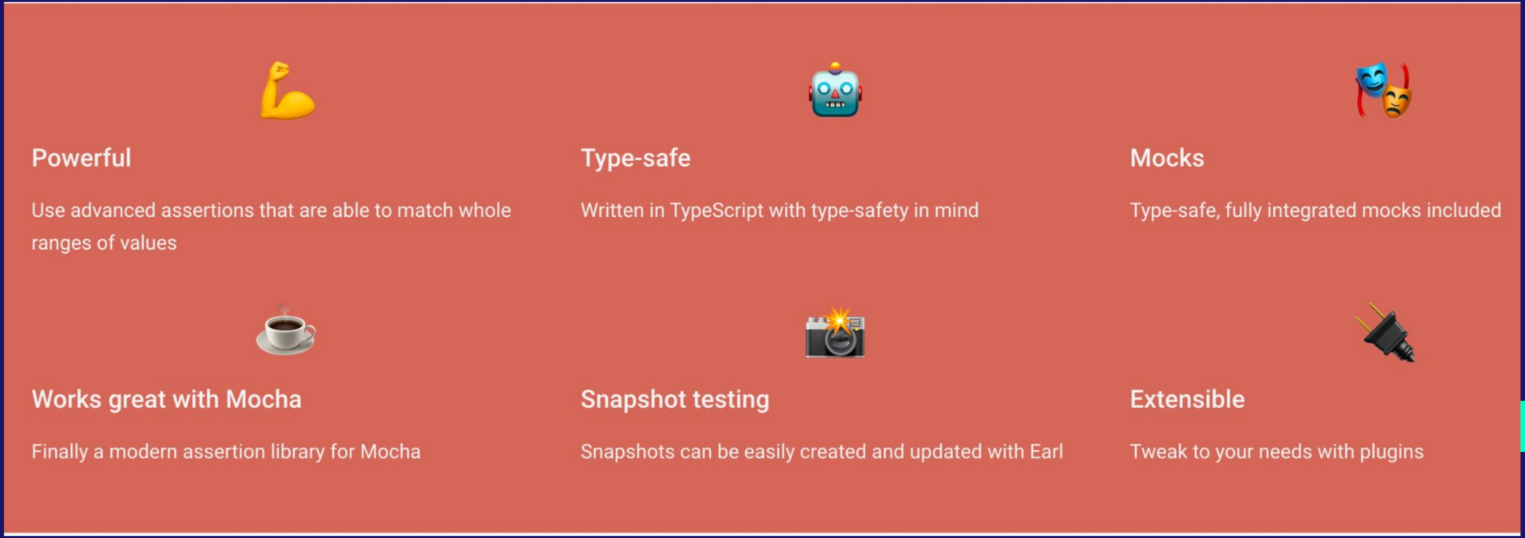


```
import { defineConfig } from '@dethcrypto/eth-sdk'

export default defineConfig({
  contracts: {
    mainnet: {
      dai: '0x6b175474e89094c44da98b954eedeac495271d0f',
    },
  },
})
```



```
import { getMainnetSdk } from '@dethcrypto/eth-sdk-client'
import { ethers } from 'ethers'

async function main() {
  const mainnetProvider = ethers.getDefaultProvider('mainnet')
  const defaultSigner = ethers.Wallet.createRandom().connect(mainnetProvider)

  const sdk = getMainnetSdk(defaultSigner)

  const balance = sdk.dai.balanceOf(defaultSigner.address)
}
```

earl.js



Powerful

Use advanced assertions that are able to match whole ranges of values

Type-safe

Written in TypeScript with type-safety in mind

Mocks

Type-safe, fully integrated mocks included

Works great with Mocha

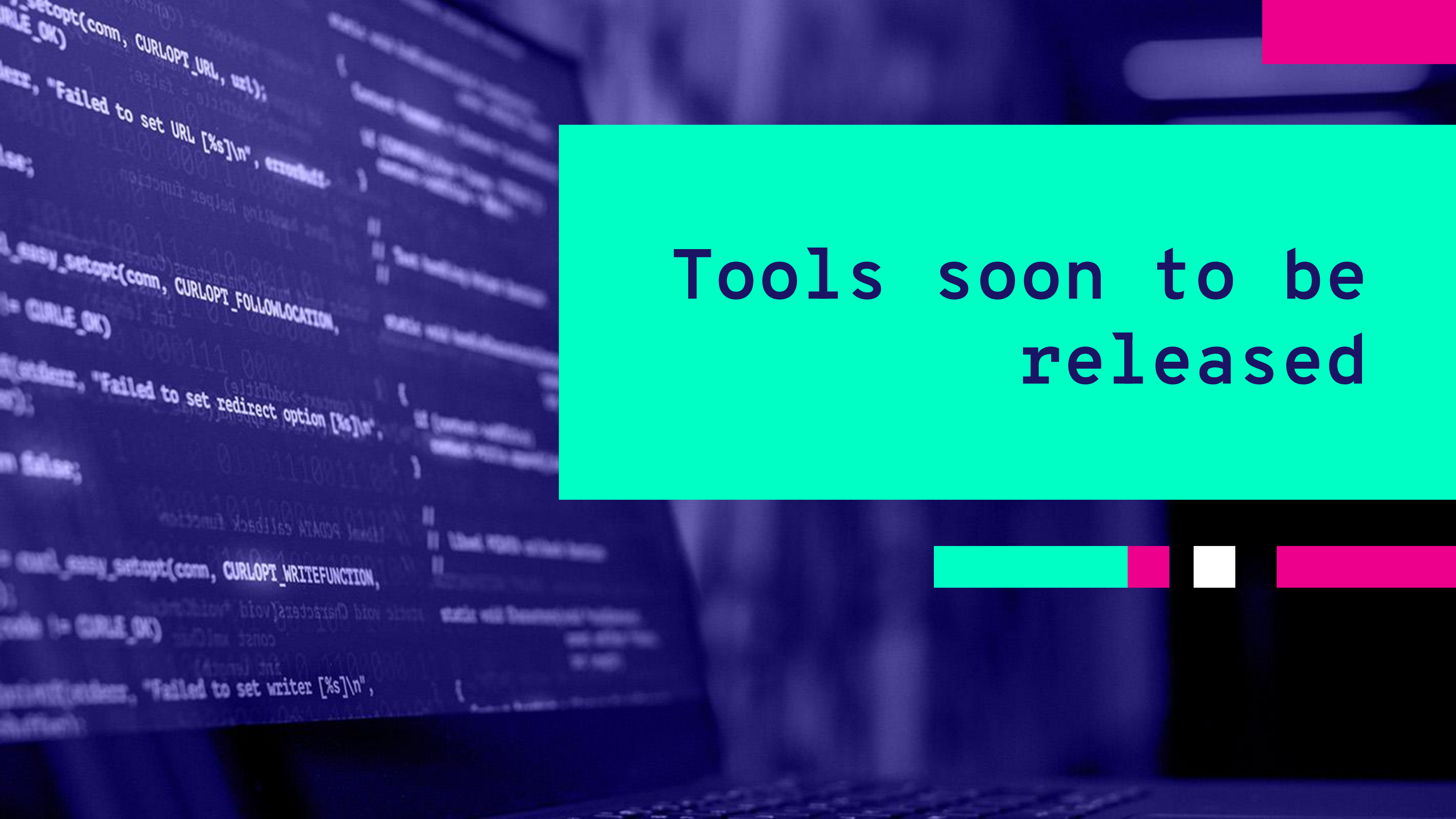
Finally a modern assertion library for Mocha

Snapshot testing

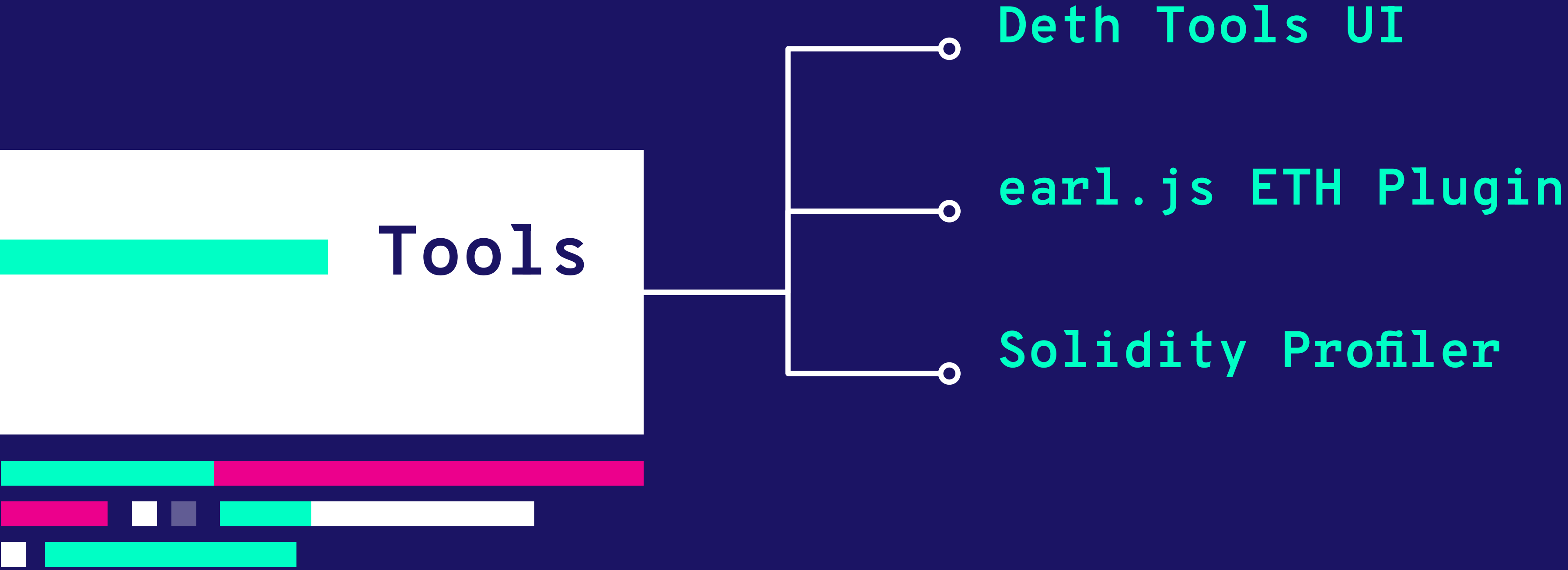
Snapshots can be easily created and updated with Earl

Extensible

Tweak to your needs with plugins



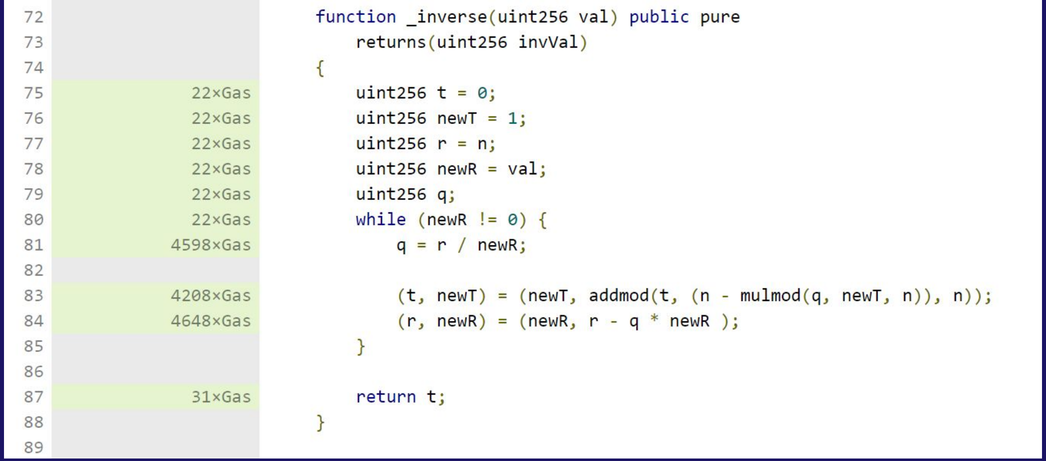
Tools soon to be released



Tools

- Deth Tools UI
- earl.js ETH Plugin
- Solidity Profiler

Bit of history



```
                 function _inverse(uint256 val) public pure
                     returns(uint256 invVal)
                 {
   22×Gas            uint256 t = 0;
   22×Gas            uint256 newT = 1;
   22×Gas            uint256 r = n;
   22×Gas            uint256 newR = val;
   22×Gas            uint256 q;
   22×Gas            while (newR != 0) {
 4598×Gas                q = r / newR;

 4208×Gas                (t, newT) = (newT, addmod(t, (n - mulmod(q, newT, n)), n));
 4648×Gas                (r, newR) = (newR, r - q * newR );
                     }

   31×Gas            return t;
                 }
```

Solution

Change based analysis &
control flow analysis

Problem

Information overload & noise

THANKS!

@dethcrypto

CREDITS: This presentation template was created by Slidesgo, incluiding icons by Flaticon, and infographics & images by Freepik.