PD-9.0 Best Practices

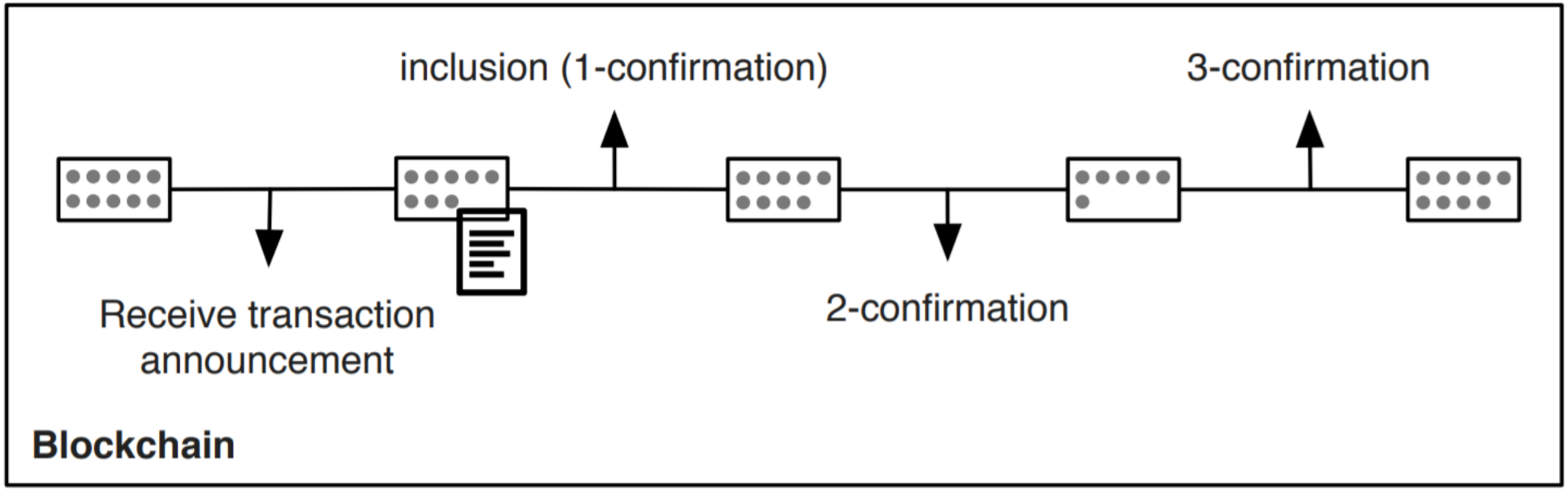


Figure 11: X-Confirmation Pattern

PD-9.1 OpenZeppelin - upgrade

OpenZeppelin

```
import "@openzeppelin/upgrades-core/contracts/Initializable.sol";
// Alternatively, if you are using @openzeppelin/contracts-ethereum-package:
// import "@openzeppelin/contracts-ethereum-package/contracts/Initializable.sol";

contract MyContract is Initializable {
  uint256 value;
  function initialize(uint256 initialValue) public initializer {
    value = initialValue;
  }
}
```

- Initialize function instead of a constructor
  - Don't use variable initialization
  - *Constant* is ok to use
  - Also call initialize on parent contracts
  - Use libraries that also support initialize
- Layout variables must stay the same

Npm install @openzeppelin/truffle-upgrades

https://github.com/OpenZeppelin/openzeppelin-upgrades/tree/master/packages/plugin-truffle

https://docs.openzeppelin.com/upgrades-plugins/1.x/api-truffle-upgrades

https://docs.openzeppelin.com/upgrades-plugins/1.x/faq#what-does-it-mean-for-a-contract-to-be-upgrade-safe

https://docs.openzeppelin.com/upgrades-plugins/1.x/writing-upgradeable

PD-9.1 OpenZeppelin Proxy deploy

Debug1.sol

```
// SPDX-License-Identifier: MIT
// npm install @openzeppelin/truffle-upgrades

pragma solidity ^0.6.0;
import "@openzeppelin/upgrades-core/contracts/Initializable.sol";

contract Debug1 {
    uint public result;

    function initialize(uint q) public {
        result = q;
    }
    function set(uint x) public {
        x +=1;
        x +=2;
        x +=4;
        x +=6;
        x +=8;
        result = x*2;
    }
}
```

https://github.com/web3examples/ethereum/tree/master/pattern_examples/Upgrade/contracts/Debug1.sol

https://github.com/web3examples/ethereum/tree/master/pattern_examples/Upgrade/migrations/2_deploy_contracts.js

2_deploy_contracts.js

```
const { deployProxy } = require('@openzeppelin/truffle-upgrades');

var Debug1 = artifacts.require("Debug1");

module.exports = async function(deployer) {
    const Debug1Contract = await deployProxy(Debug1, [42], { deployer });
    console.log(`Address of Debug1Contract: ${Debug1Contract.address}`)
    console.log("Doing some tests with the just deployed contract");
    var bnx=await Debug1Contract.result() // note result is Big Number
    console.log(`Initialized with 42, X is now ${bnx.toString()}`)
    await Debug1Contract.set(3)
    var bnx=await Debug1Contract.result() // note result is Big Number
    console.log(`Called function set(3), X is now ${bnx.toString()}`)
};
```

Address of Debug1Contract: 0x97..
Doing some tests with the just deployed contract
Initialized with 42, X is now 42
Called function set(3), X is now 48

PD-9.1 OpenZeppelin Proxy upgrade

Debug2.sol

```
// SPDX-License-Identifier: MIT
pragma solidity ^0.6.0;
import "@openzeppelin/upgrades-core/contracts/Initializable.sol";

contract Debug2 {
    uint public result;

    function initialize(uint q) public {
        result = q;
    }
    function set(uint x) public {
      result = x;
    }
   function set2(uint x) public {
      result = x*2;
    }
}
```

https://github.com/web3examples/ethereum/tree/master/pattern_examples/Upgrade/contracts/Debug2.sol

https://github.com/web3examples/ethereum/tree/master/pattern_examples/Upgrade/migrations/3_upgrade_contracts.js

3_upgrade_contracts.js

```
const { deployProxy, upgradeProxy } = require('@openzeppelin/truffle-upgrades');
var Debug1 = artifacts.require("Debug1");
var Debug2 = artifacts.require("Debug2");

module.exports = async function(deployer) {
    const Debug1Contract=await Debug1.deployed()
    const Debug2Contract=await upgradeProxy(Debug1Contract.address, Debug2,{ deployer });
    console.log(`Address of Debug1Contract: ${Debug1Contract.address}`)
    console.log(`Address of Debug2Contract: ${Debug2Contract.address}`)
    console.log("Doing some tests with the just upgraded contract");
    await Debug2Contract.set(5)
    var bnx=await Debug2Contract.result() // note result is Big Number
    console.log(`Called function set(5), X is now ${bnx.toString()}`)
    await Debug2Contract.set2(5)
    var bnx=await Debug2Contract.result() // note result is Big Number
    console.log(`Called function set2(5), X is now ${bnx.toString()}`)
}
```

Address of Debug1Contract: 0x97..
Address of Debug2Contract: 0x97..
Doing some tests with the just upgraded contract
Called function set(5), X is now 5
Called function set2(5), X is now 10

PD-9.2 Proxy contract

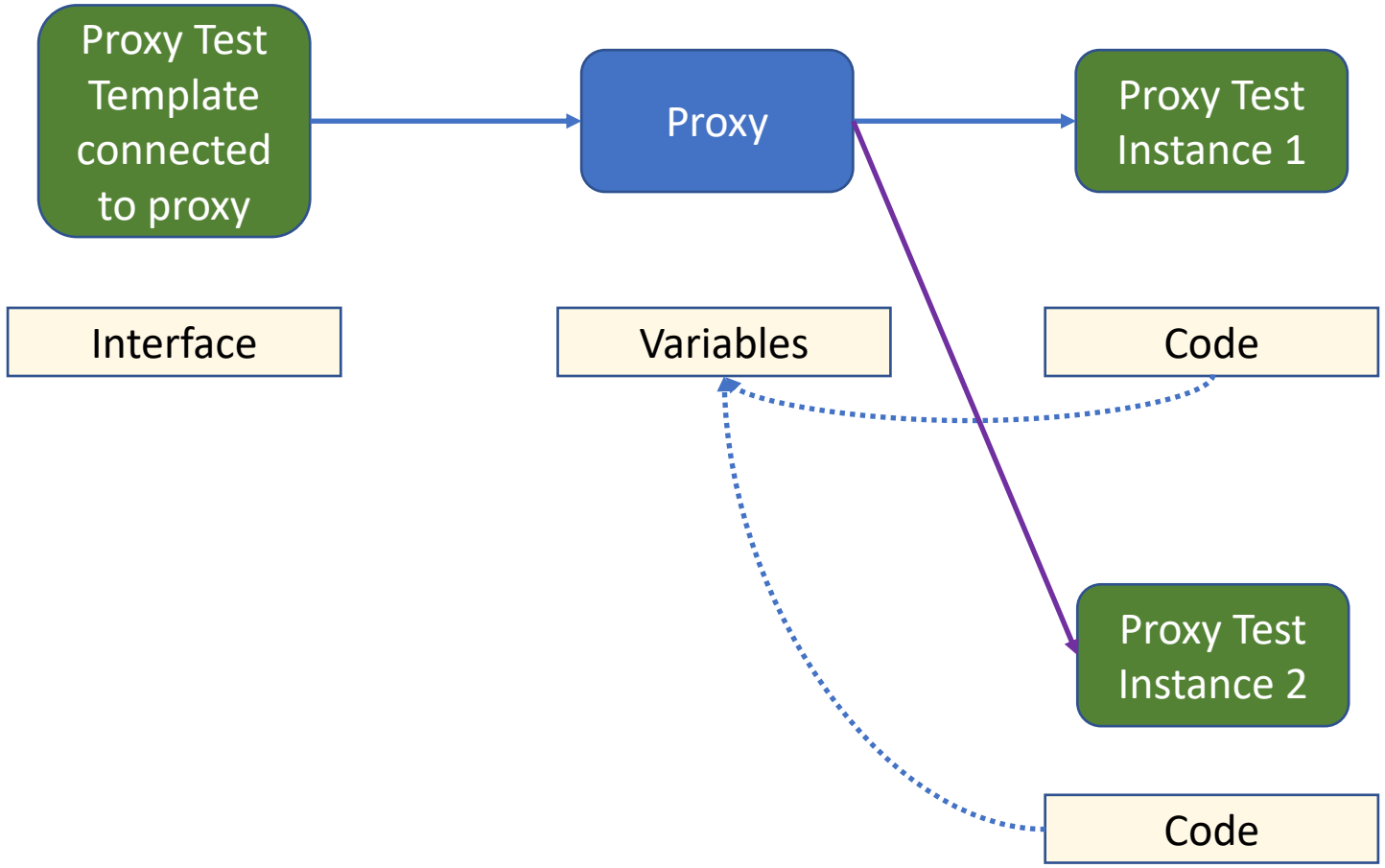
proxy_storage.sol

```
// SPDX-License-Identifier: MIT
pragma solidity ^0.7.0;

contract Version1 { event LogStr(string);string public V;function version() external { V="Version1";emit LogStr(V); }}
contract Version2 { event LogStr(string);string public V;function version() external { V="Version2";emit LogStr(V); }}

contract Proxy_Storage {
    bytes32 private constant implementationPosition = keccak256("web3examples");
    string public V="proxy";
    event LogAdr(address);

    function setV1() public { SetRelay(address(new Version1())); }
    function setV2() public { SetRelay(address(new Version2())); }

    function SetRelay(address newVersion) public {
        bytes32 slot = implementationPosition;
            assembly { sstore(slot, newVersion) }
    }
    function GetRelay() public view returns(address implementation) {
            bytes32 slot = implementationPosition;
            assembly { implementation := sload(slot) }
    }

    fallback() external payable {
            address implementation = GetRelay();
            emit LogAdr(implementation);
            (bool success, /*bytes memory data*/)=implementation.delegatecall(msg.data);
            require(success,"error");
    }
    receive() external payable {}
}
```

https://blog.openzeppelin.com/upgradeability-using-unstructured-storage

https://github.com/web3examples/ethereum/blob/master/pattern_examples/proxy_storage.sol

PD-9.2 Proxy contract



PD-9.2 Attention points with data contracts (delegatecall)

- Constructors don’t work
- Variable initializations don’t work (constants do)
- Don’t change layout of variables
  - Don’t change order
  - Don’t remove
  - Don’t change type
  - Only add varables at the end

https://docs.openzeppelin.com/sdk/2.6/writing-contracts

PD-9.3 Modifiers

modifiers.sol

```
// SPDX-License-Identifier: MIT
pragma solidity ^0.7.0;
contract Owned {
    address public owner;
    uint public creationTime = block.timestamp;
    modifier onlyOwner()              { require(msg.sender == owner,      "Must be owner");      _;}
    modifier onlyBefore(uint _time)   { require(block.timestamp < _time,  "Too late");           _;}
    modifier onlyAfter(uint _time)    { require(block.timestamp > _time,  "Too soon");           _;}
    modifier onlyBy(address account)  { require(msg.sender == account,    "Wrong address" );     _;}
    modifier condition(bool _condition) { require(_condition,             "Condition failed");   _;}
    modifier minAmount(uint _amount)  { require(msg.value >= _amount,     "Not enough ETH send"); _;}

    constructor() { owner = msg.sender;}

    function f() payable
        onlyBy(owner)
        minAmount(2 ether)
        onlyAfter(creationTime + 1 minutes)
        condition(msg.sender.balance >= 50 ether)
        public returns(string memory) { // code
            return "Done";
    }
}
```

https://github.com/web3examples/ethereum/tree/master/pattern_examples/modifiers.sol

PD-9.4 Factory Contract

factory.sol

```
// SPDX-License-Identifier: MIT
pragma solidity ^0.7.0;

contract ChildContract {
    uint public MyId;
    constructor (uint Instance) {
        MyId = Instance;
    }
}

contract ContractFactory {
    ChildContract[] contracts;
    uint ChildNr;
    function CreateChild() public returns(ChildContract) {
        ChildContract Child = new ChildContract(ChildNr++);
        contracts.push(Child);
        return Child;
    }

    function Contracts() public view returns (ChildContract[] memory) {
        return contracts;
    }
}
```

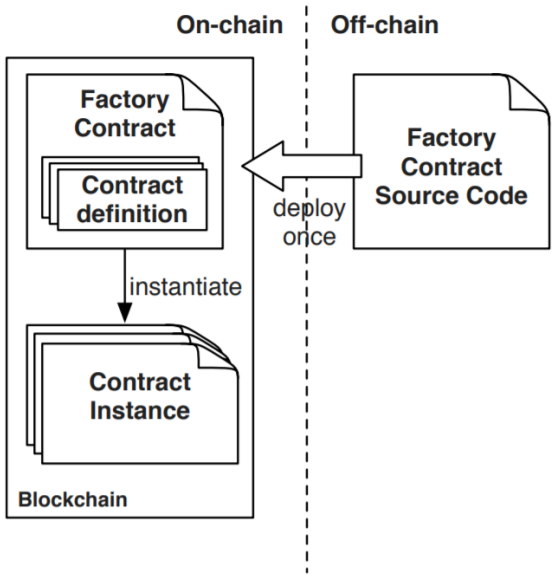


Figure 15: Factory Contract Pattern

https://github.com/web3examples/ethereum/blob/master/pattern_examples/factory.sol

PD-9.5 Selfdestruct & Create2

selfdestruct_create2.sol

```
// SPDX-License-Identifier: MIT
pragma solidity ^0.7.0;

contract Child {
    string public name="Child";
    function destroy() public  { // add security
        selfdestruct(msg.sender);
    }
}

contract Factory {
    Child public deployed;

    function ChildName() public view returns (string memory) {
        return deployed.name();
    }
    function DestroyChild() public { // add security
        deployed.destroy();
        deployed=Child(address(0));
    }

    function Deploy() public returns (Child){
        deployed=new Child{salt: 0x00}(); // create2
        return deployed;
    }
}
```

https://github.com/web3examples/ethereum/tree/master/pattern_examples/selfdestruct_create2.sol

PD-9.6 Commit Reveal

commitreveal.sol

```
// SPDX-License-Identifier: MIT
pragma solidity ^0.7.0;
contract CommitReveal {

    bytes32 commit;
    function CommitValue(bytes32 _commit) internal {
        commit = _commit;
    }
    function RevealValue(string memory _value) internal view returns (string memory) {
        require (commit == keccak256(bytes(_value)),"Revealed value != committed");
        return (_value);
    }
    function TestCommitOk(string memory _value) public returns(bytes32) {
        bytes32 c=keccak256(bytes(_value));
        CommitValue(c);
        RevealValue(_value);
        return c;
    }
    function TestCommitBad(string memory _value) public returns(bytes32) {
        bytes32 c="0x00";
        CommitValue(c);
        RevealValue(_value);
        return c;
    }
}
```

https://github.com/web3examples/ethereum/tree/master/pattern_examples/commitreveal.sol

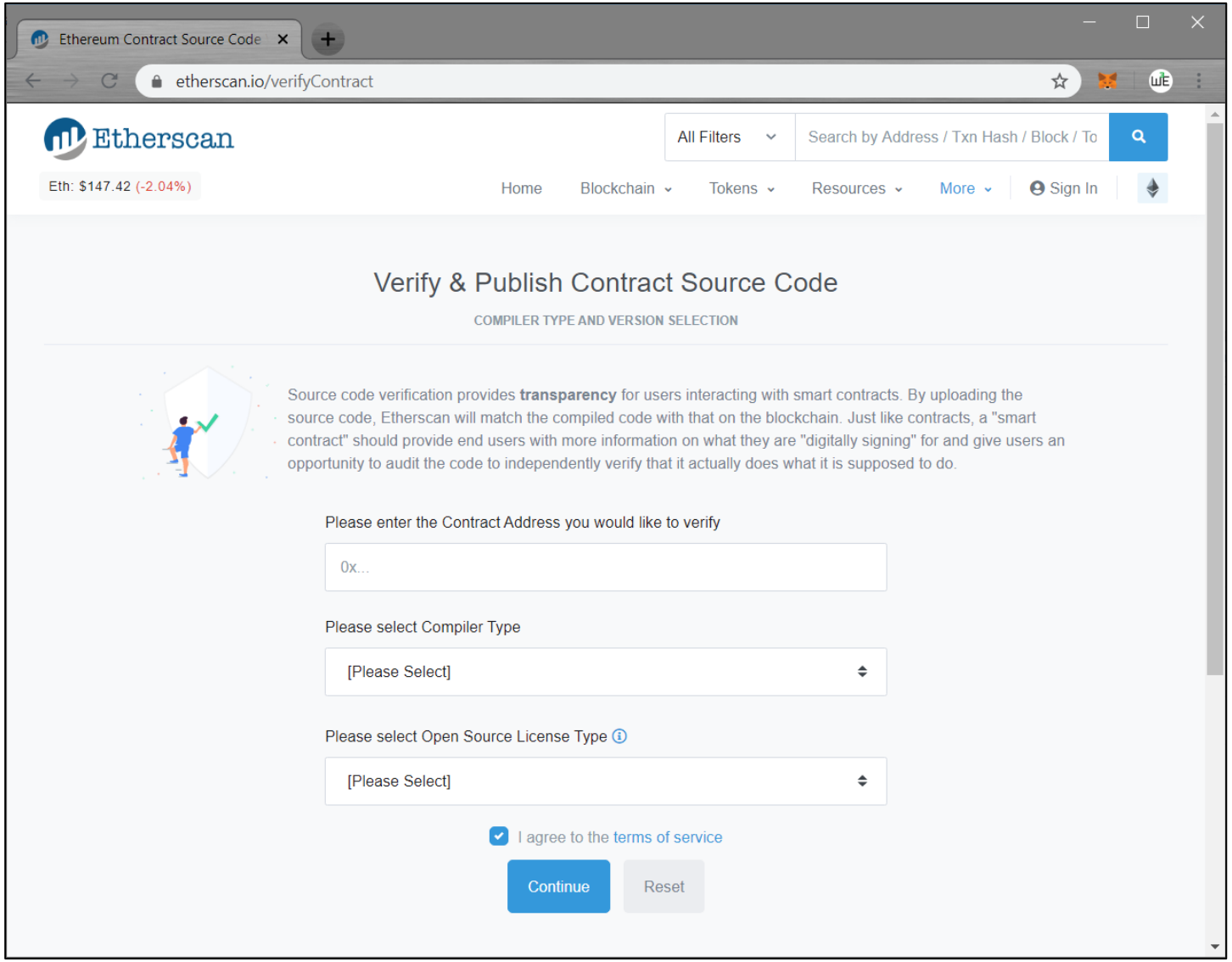
PD-9.7 Send, Transfer, Call

```
function ViaSend(address payable addr) internal   {
    bool success = addr.send(msg.value); // 2300 gas
    require(success, "Pay was not successful.");
}
function ViaTrans(address payable addr) internal  {
    addr.transfer(msg.value); // 2300 gas
}
function ViaCall(address payable addr) internal  {
    (bool success, /* bytes memory response*/) = addr.call{value: msg.value}('');
    require(success, "Pay was not successful.");
}
```

https://github.com/web3examples/ethereum/blob/master/pattern_examples/sendtransfercall.sol

https://diligence.consensys.net/blog/2019/09/stop-using-soliditys-transfer-now/

PD-9.8 Publish source code



https://etherscan.io/verifyContract

https://tokenmint.io/blog/how-to-verify-ethereum-smart-contracts-source-code.html

http://web3examples.com/ethereum/demo/Publish_on_etherscan_and_interact.html

PD-9.8 Publish source code

Ropsten Accounts, Addresses an

ropsten.etherscan.io/address/0x55bdf79860ca3a68d53171d3a3a2fa4696a1...

Transactions | Contract | Events

Code | Read Contract | Write Contract

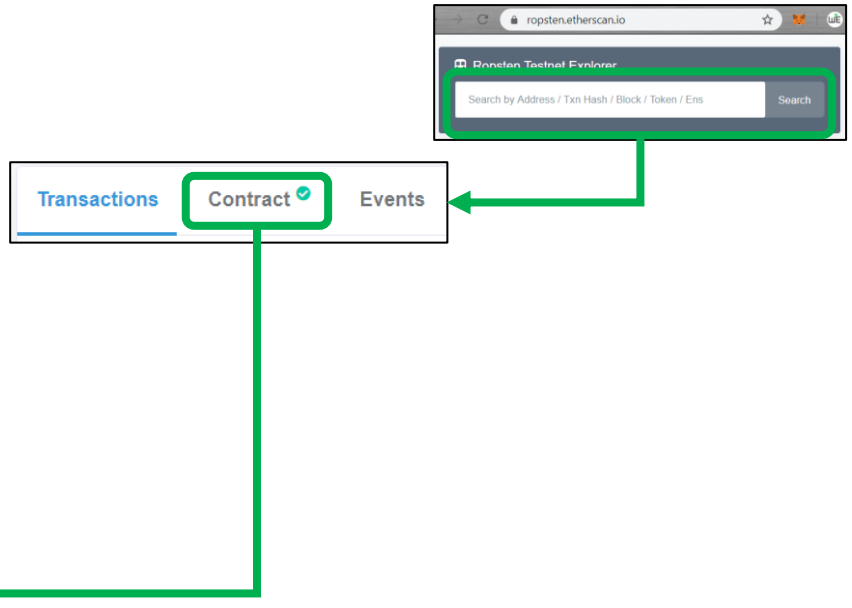
Contract Source Code Verified (Exact Match)

Contract Name:	**TestPublish**	Optimization Enabled:	**Yes** with **200** runs
Compiler Version	**v0.5.10+commit.5a6ea5b1**	Other Settings:	**default** evmVersion, **MIT** license

Contract Source Code (Solidity)

Similar | Outline | Sol2Uml NEW

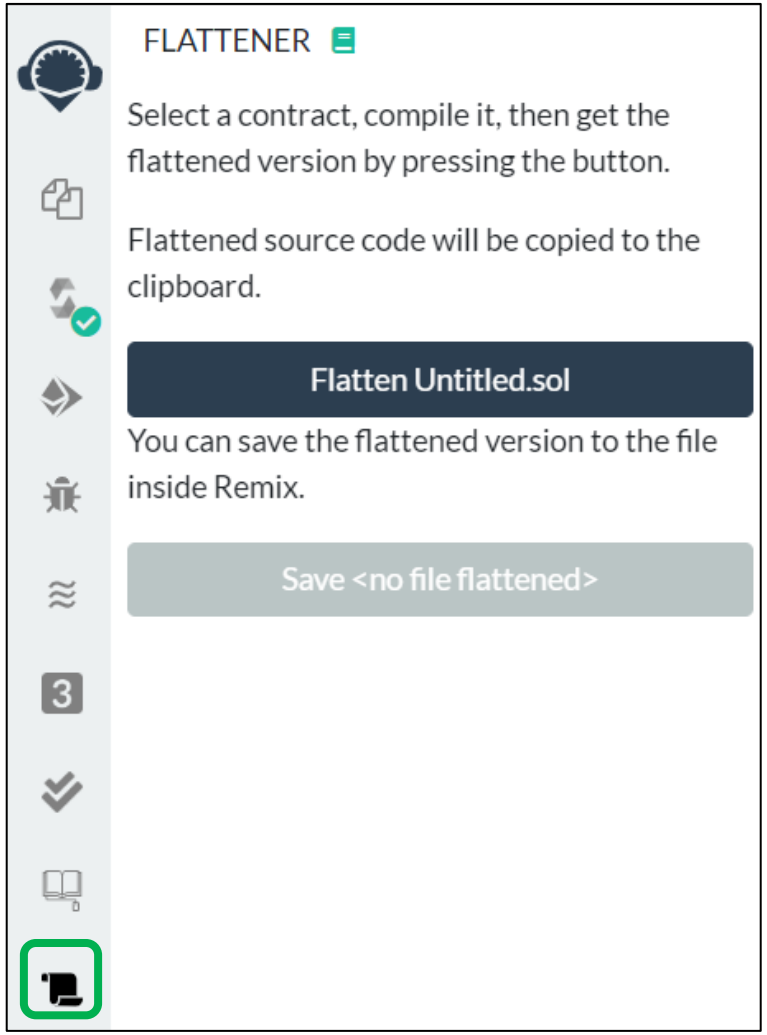
```
/**
 *Submitted for verification at Etherscan.io on 2019-07-26
*/

// Verify code from https://ethereum-play.github.io/editor-solidity/ on etherscan:
// Check tab output.json, notice the compiler version and optimizer:true
pragma solidity >=0.4.0 <0.7.0;
contract TestPublish {
  string public MyName="Test publication to Etherscan";
  function SetMyName(string calldata _MyName) external {
      MyName = _MyName;
  }
}
```



https://github.com/web3examples/ethereum/tree/master/pattern_examples/verifysource.sol

https://ropsten.etherscan.io/address/0x55bdf79860ca3a68d53171d3a3a2fa4696a16f61#code

PD-9.8 Flatten source (for imports)

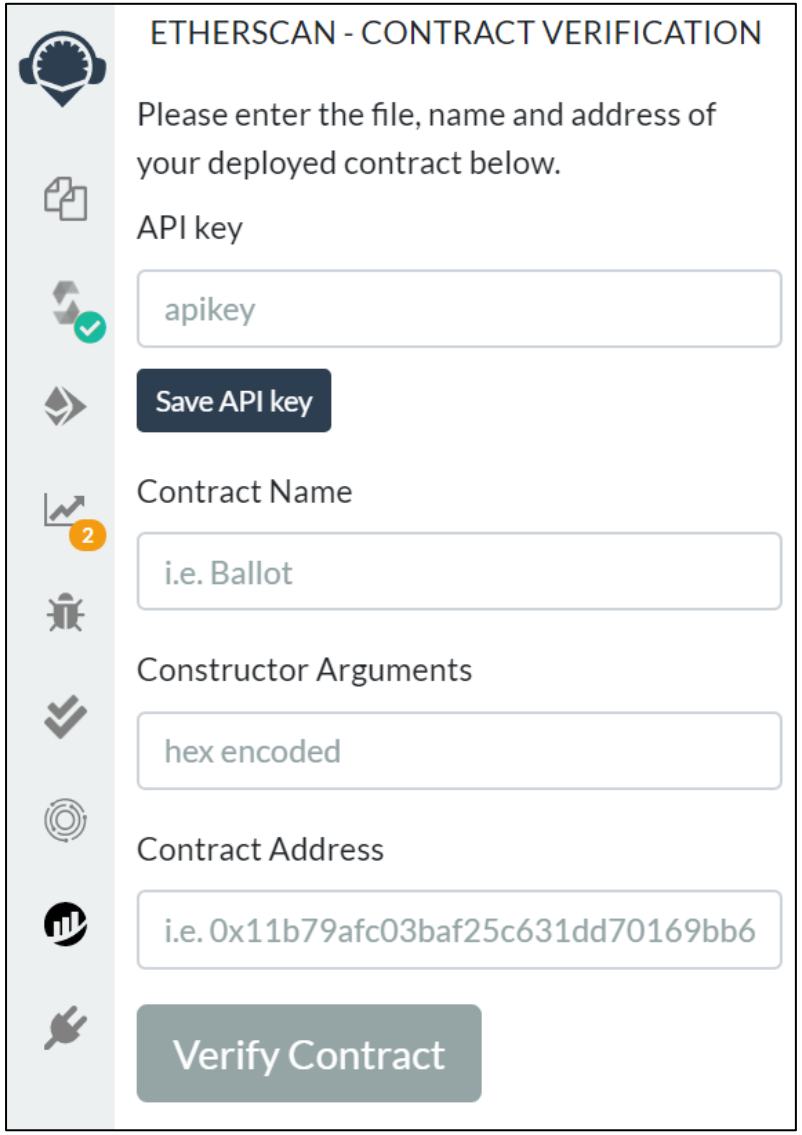


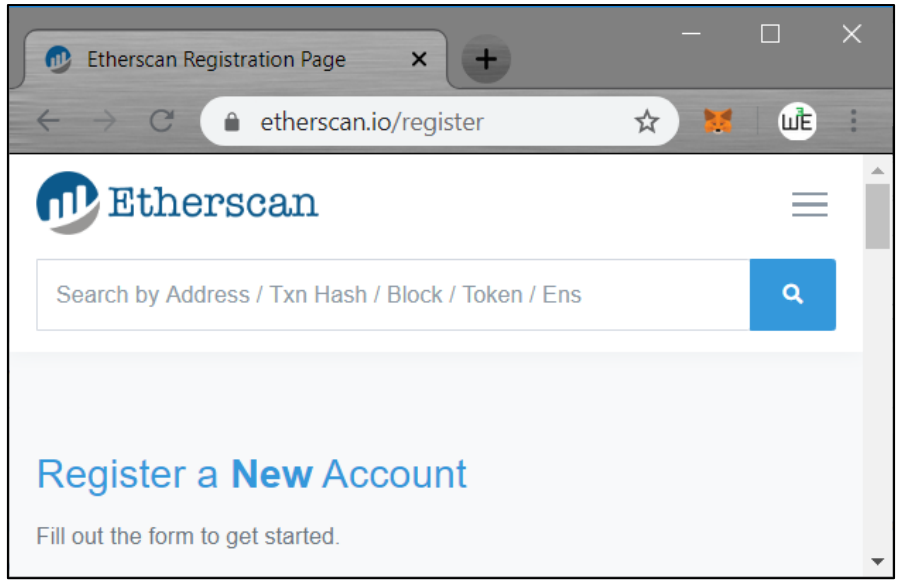
https://github.com/poanetwork/solidity-flattener

https://www.npmjs.com/package/truffle-flattener

https://marketplace.visualstudio.com/items?itemName=tintinweb.vscode-solidity-flattener

PD-9.8 Remix – Etherscan contract verification





https://etherscan.io/register

PD-9.8 Truffle – contract verification

truffle-config.js

```
const HDWalletProvider = require('@truffle/hdwallet-provider');
const fs = require('fs');
const mnemonic = fs.readFileSync(".secret").toString().trim(); // contains mnemonic
const infuraKey = fs.readFileSync(".infura").toString().trim(); // infura key
const etherscanKey = fs.readFileSync(".etherscan").toString().trim(); // etherscan key

module.exports = {
    networks: {
    development: {
      host: "127.0.0.1",     // Localhost (default: none)
      port: 7545,            // Standard Ethereum port (default: none)
      network_id: "*",       // Any network (default: none)
    },
    rinkeby: {
        provider: () => new HDWalletProvider(mnemonic, `https://rinkeby.infura.io/v3/${infuraKey}`),
        network_id: 4,       // rinkeby id
        skipDryRun: true
    }
  },
    mocha: { },
    compilers: { solc: { version: "^0.6.0" } },
    plugins: [
        'truffle-plugin-verify'
    ],
    api_keys: {
        etherscan: etherscanKey
    }
}
```

Note: doesn't work well on windows

> npm install -g truffle-plugin-verify

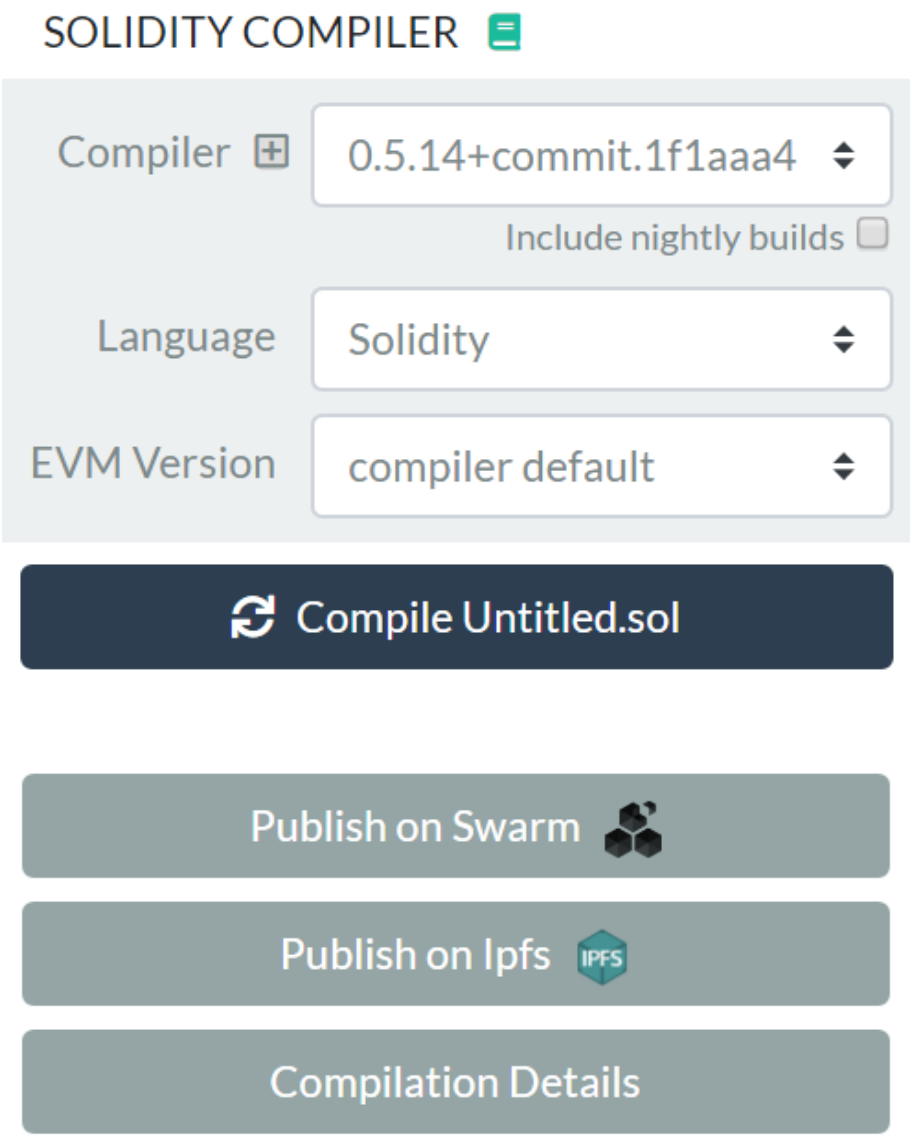
> truffle run verify TestPublish --network rinkeby

Verifying TestPublish
Pass - Verified: https://rinkeby.etherscan.io/address/0x..#contracts
Successfully verified 1 contract(s).

https://www.npmjs.com/package/truffle-plugin-verify

https://kalis.me/verify-truffle-smart-contracts-etherscan/

PD-9.8 Publish Metadata Remix



https://solidity.readthedocs.io/en/latest/metadata.html

PD-9.9 Publish source code play editor

Please select Compiler Version

[Please Select]

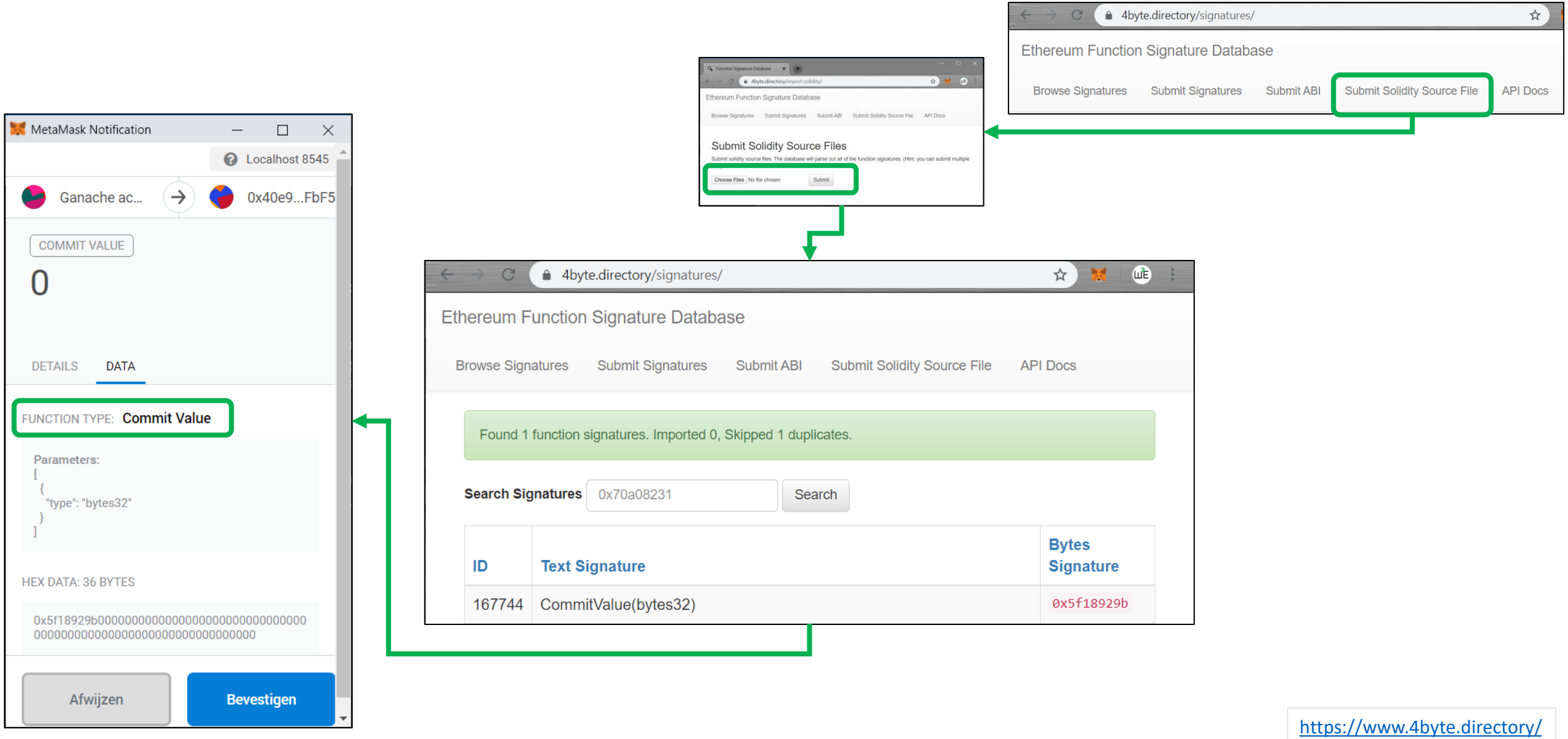
Optimization

No

contract.sol output.json


```
  },
  "compiler": {
    "language": "solidity",
    "version": "v0.5.12-stable-2019.09.01",
    "url": "https://solc-bin.ethereum.org/bin/soljson-v0.5.12+commit.7709ece9.js",
    "evmVersion": "petersburg",
    "optimizer": true,
    "runs": 200
  },
```

PD-9.10 Register function names



https://www.4byte.directory/

PD-9.11 Source layout

In each source file:

1. Pragma statements
2. Import statements
3. Interfaces
4. Libraries
5. Contracts

For each interface/ library / contract

1. Type declarations
2. State variables
3. Events
4. Functions

In each list of functions

1. constructor
2. receive function (if exists)
3. fallback function (if exists)
4. external
5. public
6. internal
7. private

```
pragma solidity ^0.6.0;

contract A {
    constructor() public {
        // ...
    }

    receive() external payable {
        // ...
    }

    fallback() external {
        // ...
    }

    // External functions
    // ...

    // External functions that are view
    // ...

    // External functions that are pure
    // ...

    // Public functions
    // ...

    // Internal functions
    // ...

    // Private functions
    // ...
}
```

https://docs.soliditylang.org/en/latest/style-guide.html#code-layout

https://solidity.readthedocs.io/en/latest/style-guide.html#order-of-layout

PD-9.11 Naming conventions

CapitalizedWords

- Contracts
- Libraries
- Structs
- Enums
- Events

mixedCase

- Functions
- Function arguments
- Variables
- Modifiers

UPPERCASE

- Constants

```
pragma solidity >=0.4.0 <0.7.0;


// Owned.sol
contract Owned {
    address public owner;

    constructor() public {
        owner = msg.sender;
    }

    modifier onlyOwner {
        require(msg.sender == owner);
        _;
    }

    function transferOwnership(address newOwner) public onlyOwner {
        owner = newOwner;
    }
}
```

https://solidity.readthedocs.io/en/latest/style-guide.html#naming-conventions

PD-9.11 More layout

```
pragma solidity >=0.4.0 <0.7.0;

contract A {
    // ...
}


contract B {
    // ...
}


contract C {
    // ...
}
```

```
function thisFunctionNameIsReallyLong(
    address a,
    address b,
    address c
)
    public
    returns (
        address someAddressName,
        uint256 LongArgument,
        uint256 Argument
    )
{
    doSomething()

    return (
        veryLongReturnArg1,
        veryLongReturnArg2,
        veryLongReturnArg3
    );
}
```

https://solidity.readthedocs.io/en/latest/style-guide.html#

PD-9.12 Natspec

sol6_natspec.sol

```
/// Based on https://solidity.readthedocs.io/en/develop/natspec-format.html

pragma solidity ^0.6.1;

/// @title A simulator for trees
/// @author Larry A. Gardner
/// @notice You can use this contract for only the most basic simulation
/// @dev All function calls are currently implemented without side effects
contract Tree {
/// @author Mary A. Botanist
/// @notice Calculate tree age in years, rounded up, for live trees
/// @dev The Alexandr N. Tetearing algorithm could increase precision
/// @param rings1 The number of rings from dendrochronological sample
/// @param rings2 The number of rings from dendrochronological sample
/// @return age1 in years, rounded up for partial years
/// @return age2 in years, rounded up for partial years // shown separately with solidity 0.6.0
    function age(uint256 rings1, uint256 rings2) external pure returns (uint256 age1,uint256 age2) {
        return (rings1 + 1,rings2 + 1);
    }
}
```

https://solidity.readthedocs.io/en/latest/natspec-format.html#documentation-example

https://github.com/web3examples/ethereum/blob/master/solidity_examples/sol6_natspec.sol

PD-9.12 Natspec Tags

Tag		Context
`@title`	A title that should describe the contract/interface	contract, interface
`@author`	The name of the author	contract, interface, function
`@notice`	Explain to an end user what this does	contract, interface, function
`@dev`	Explain to a developer any extra details	contract, interface, function
`@param`	Documents a parameter just like in doxygen (must be followed by parameter name)	function
`@return`	Documents the return variables of a contract's function	function

https://solidity.readthedocs.io/en/latest/natspec-format.html

PD-9.12 SOLC (solidity compiler)

solidity-windows.zip

Name	Size
msvcp140.dll	450 024
msvcp140_1.dll	29 160
msvcp140_2.dll	173 544
solc.exe	5 788 672
soltest.exe	14 059 008

Download
Unzip
add to path

https://github.com/ethereum/solidity/releases

PD-9.12 Userdoc

```
>solc sol6_natspec.sol --userdoc

======= sol6_natspec.sol:Tree =======
User Documentation
{  "methods":
  {   "age(uint256,uint256)":
    {    "notice": "Calculate tree age in years, rounded up, for live trees"
    }
  },
  "notice": "You can use this contract for only the most basic simulation"
}
```

PD-9.12 Devdoc

```
>solc sol6_natspec.sol --devdoc
======= sol6_natspec.sol:Tree =======
Developer Documentation
{ "author": "Larry A. Gardner",
  "details": "All function calls are currently implemented without side effects",
  "methods":
  { "age(uint256,uint256)":
    { "author": "Mary A. Botanist",
      "details": "The Alexandr N. Tetearing algorithm could increase precision",
      "params":
      { "rings1": "The number of rings from dendrochronological sample",
        "rings2": "The number of rings from dendrochronological sample "
      },
      "returns":
      { "age1": "in years, rounded up for partial years",
        "age2": "in years, rounded up for partial years // shown separately now"
      }
    }
  },
  "title": "A simulator for trees"
}
```

PD-9.13 Play editor

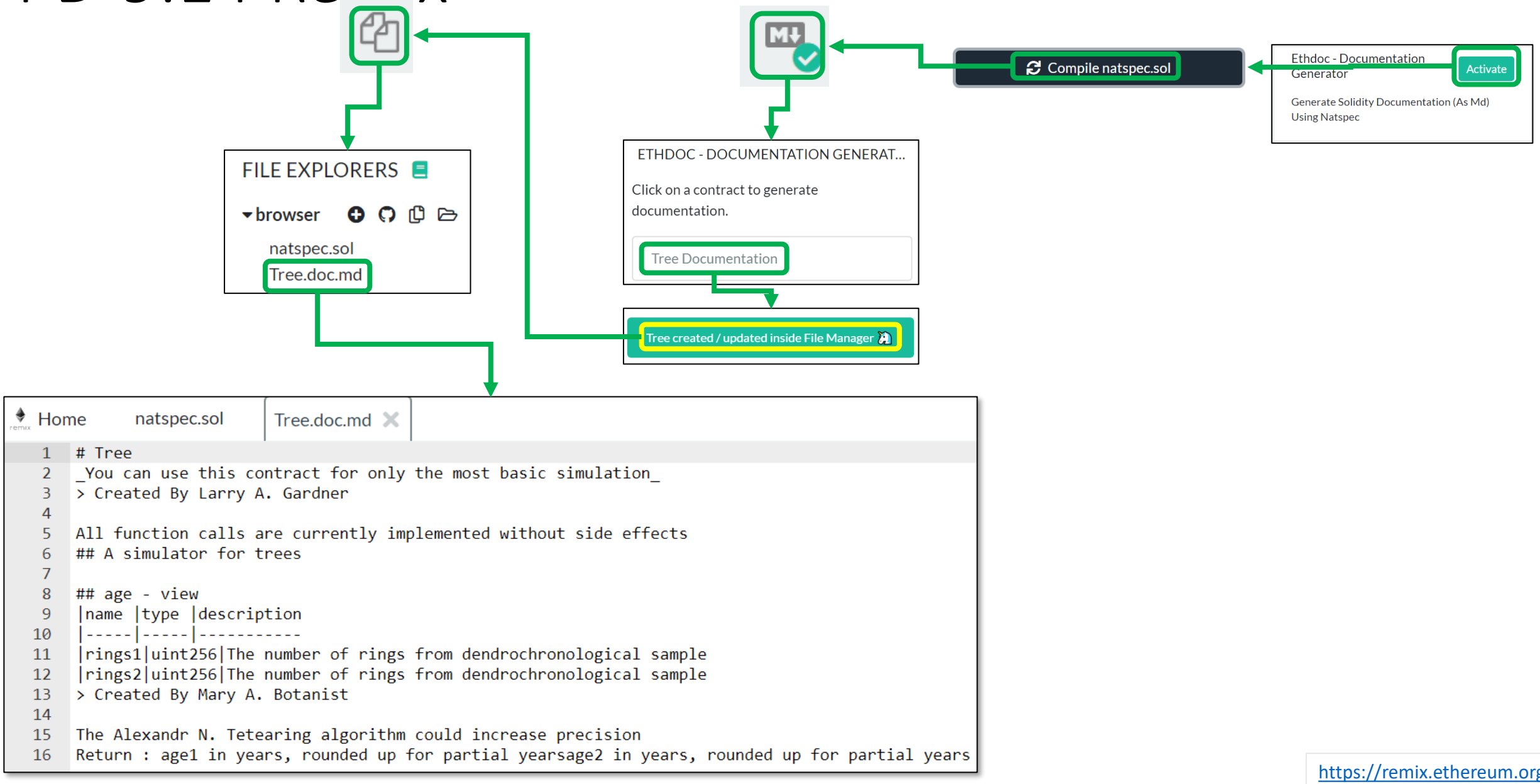
ethereum-play.github.io/editor-solidity/

contract.sol | output.json


```
"devdoc": {
  "author": "Larry A. Gardner",
  "details": "All function calls are currently implemented without side effects",
  "methods": {
    "age(uint256,uint256)": {
      "author": "Mary A. Botanist",
      "details": "The Alexandr N. Tetearing algorithm could increase precision",
      "params": {
        "rings1": "The number of rings from dendrochronological sample",
        "rings2": "The number of rings from dendrochronological sample "
      },
      "return": "age1 in years, rounded up for partial yearsage2 in years, rounded up for
    }
  },
  "title": "A simulator for trees"
},
"userdoc": {
  "methods": {
    "age(uint256,uint256)": {
      "notice": "Calculate tree age in years, rounded up, for live trees"
    }
  },
  "notice": "You can use this contract for only the most basic simulation"
},
```


https://ethereum-play.github.io/editor-solidity/

PD-9.14 Remix



https://remix.ethereum.org/

PD-9.14 Tree.doc.md

Tree

You can use this contract for only the most basic simulation

> Created By Larry A. Gardner

All function calls are currently implemented without side effects

A simulator for trees

age - view

name	type	description
rings1	uint256	The number of rings from dendrochronological sample
rings2	uint256	The number of rings from dendrochronological sample

> Created By Mary A. Botanist

The Alexandr N. Tetearing algorithm could increase precision Return : age1 in years, rounded up for partial yearsage2 in years, rounded up for partial years

https://github.com/web3examples/ethereum/blob/master/solidity_examples/Tree.doc.md

PD-9.15 SafeMath

safemath_underflow.sol

```
// SPDX-License-Identifier: MIT
pragma solidity ^0.7.0;

import "https://github.com/OpenZeppelin/openzeppelin-contracts/blob/master/contracts/math/SafeMath.sol";

contract ContractError {
    using SafeMath for uint256;

    function UncheckedUnderflow() public pure returns (uint) {
        uint x = 0;
        x = x-1; // this will generate an underflow
        return x;
    }

    function Underflow() public pure returns (uint) {
        uint x = 0;
        x = x.sub(1); // this will generate an underflow
        return x;
    }
}
```

https://github.com/web3examples/ethereum/blob/master/pattern_examples/safemath_underflow.sol

https://github.com/OpenZeppelin/openzeppelin-contracts/blob/master/contracts/math/SafeMath.sol

PD-9.15 Solidity 8

sol8_underflow.sol

```
// SPDX-License-Identifier: MIT
// Based on https://solidity.ethereum.org/2020/10/28/solidity-0.8.x-preview
// https://solidity-blog.s3.eu-central-1.amazonaws.com/data/08preview/soljson.js

pragma solidity >0.7.0;

contract ContractError {
    function Underflow() public pure returns (uint) {
        uint x = 0;
        x--; // this will generate an underflow
        return x;
    }
    function UncheckedUnderflow() public pure returns (uint) {
        uint x = 0;
        unchecked { x--; } // this will generate an underflow
        return x;
    }
}
```

https://solidity.ethereum.org/2020/10/28/solidity-0.8.x-preview

https://github.com/web3examples/ethereum/blob/master/pattern_examples/sol8_underflow.sol

PD-9.15 Solidity 8 error handling

```
contract C {
    ContractError e = new ContractError();

    function TestUnderflow() public view returns (string memory) {
        try e.Underflow() returns (uint) {
            return "Ok";
        } catch Error(string memory reason) {
            return reason;
        } catch (bytes memory reason) {
            uint x=0;
            for (uint i=0;i<4;i++) //get first 4 bytes
                x = (x<<8) + uint(uint8(reason[i]));
            byte b4=reason[reason.length-1]; // get last byte
            if (x == 0x4e487b71) { // abi.encodeWithSignature("Panic(uint256)")
                if (b4 == hex'11')
                    return "Panic: underflow or overflow";
                return "Panic";
            }
            if (x == 0x08c379a0) // abi.encodeWithSignature("Error(string)")
                return "Error";
            return "Unknown";
        }
    }
}
```

https://solidity.ethereum.org/2020/10/28/solidity-0.8.x-preview

https://github.com/web3examples/ethereum/blob/master/pattern_examples/sol8_underflow.sol