Tools available to develop and test contracts

repository for Ethereum dev tools @Github

To develop and test contracts

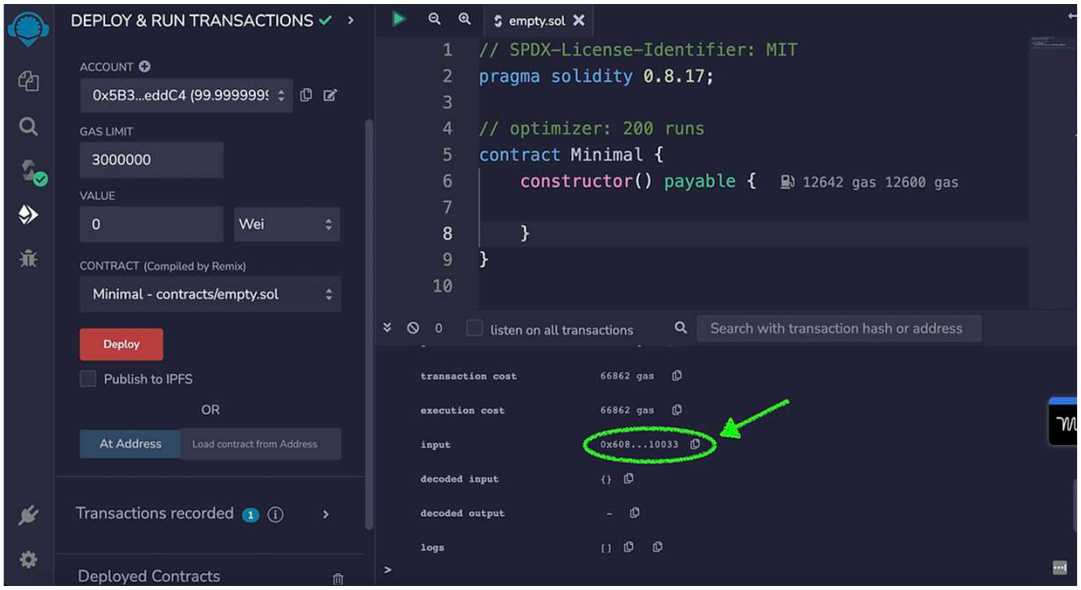
Solidity

docs: https://docs.soliditylang.org/en/develop/

IDE: https://remix-ide.readthedocs.io/en/latest/

The smart contract **Programming** language

- **Solidity :**Solidity is an object-oriented programming language for implementing smart contracts on various blockchain platforms, most notably, Ethereum
  - is a statically-typed curly-braces programming language designed for developing smart contracts that run on Ethereum
  - most popular language for contact
- **Vyper** — relatively new, pythonic programming language used to write smart contracts.
  - Vyper targets Ethereum Virtual Machine (EVM) and has very simple/intelligible syntax;
- **Bamboo —** A morphing smart contract language, Bamboo makes state transition explicit and avoids reentrance problems by default.
- **LLL** — Low-level Lisp-like Language
- **Flint —** New language under development with security features including asset types, state transition, and safe integers



Frameworks

- Building a full-fledged dapp requires different pieces of technology.
- development and testing frameworks offer great support in deploying and testing smart contracts efficiently.
- Software frameworks include many of the needed features or provide easy plugin systems to pick the tools desire.

Popular frameworks for creating, compiling, testing, debugging and deploying smart contracts.

TRUFFLE

Ganache

drizzle

Truffle suite

Truffle suite: Frameworks

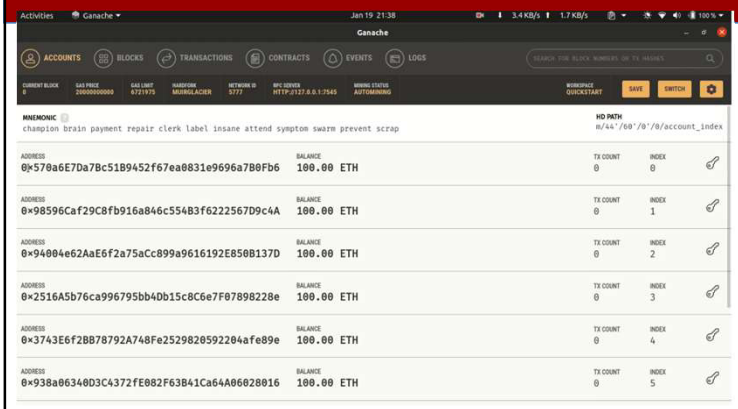
- **Truffle suite:** Most popular smart contract development, testing, debugging and deployment framework.
- Truffle suite includes **Truffle, Ganache, and Drizzle.**
- For Details refer @ https://trufflesuite.com/docs/

➢ **Truffle :** A world class development environment, testing framework and asset pipeline for blockchains using the Ethereum Virtual Machine (EVM).

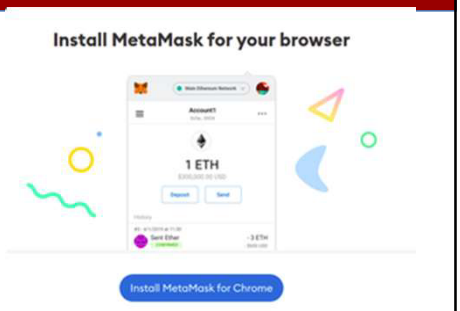
➢ **Ganache** : a personal blockchain for rapid Ethereum and Filecoin distributed application development.
  - It comes in two flavors: a UI and CLI. Ganache UI is a desktop application supporting Ethereum and Filecoin technology

➢ **Drizzle:** a collection of front-end libraries that make writing dapp front-ends easier and more predictable. The core of Drizzle is based on a Redux store.
  - Redux is an open-source JavaScript library for managing and centralizing application state.
  - It is most commonly used with libraries such as React or Angular for building user interfaces. Similar to Facebook's Flux architecture,

Truffle Suite used with Ethereum, Quorum,Filecoin, Hyperledger EVM, and other EVM-based chains



Ganache GUI



Ganache CLI

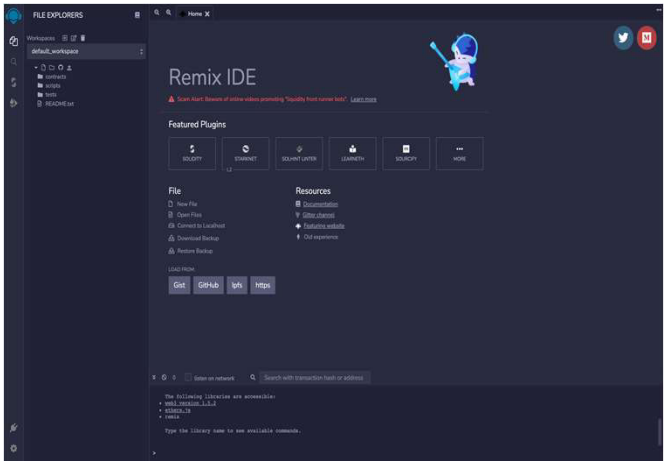
Other frameworks

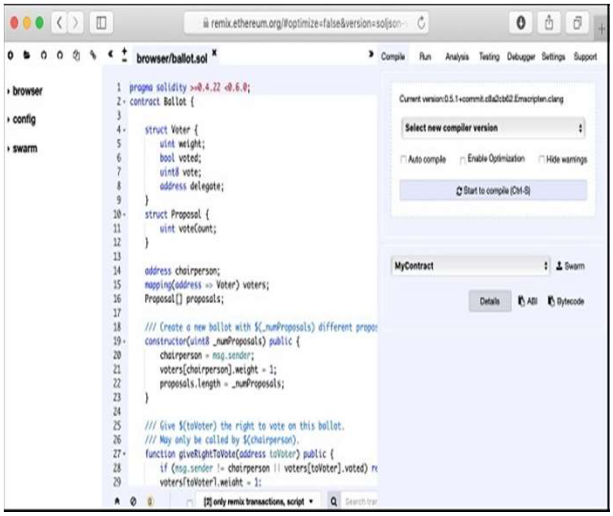
- Other options in the development and testing frameworks for smart contracts are:
  - Waffle(based on ethers.js),
  - Embark, and
  - web3j (modular, reactive, type safe Java and Android library, portable for blockchain library).
- In addition, frameworks without EVM support are
  - TerraSDK and
  - Anchor
- **Hardhat - *Ethereum development environment for professionals.***

Integrated Developer Environments

- **Remix** — is **an open-source Ethereum IDE you can use to write, compile and debug Solidity code**.
- Remix IDE, is a no-setup tool with a GUI for developing smart contracts.
- Remix IDE **is an open source web and desktop application**
- plays well with other tools, and allows for a simple deployment process to the chain of your choice.
- Remix is famous for our visual debugger.
- Web IDE with built in static analysis, test blockchain VM.
- For details refer @ https://remix.ethereum.org/
  @ https://remix-project.org/.

Remix IDE



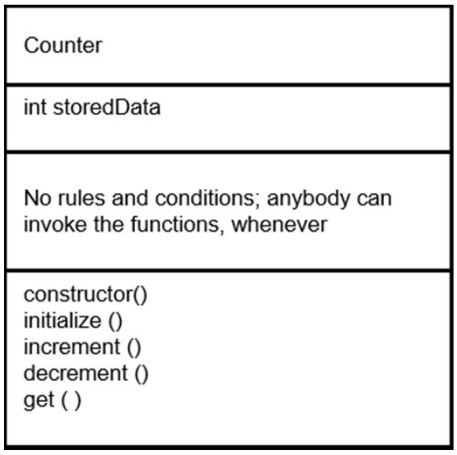


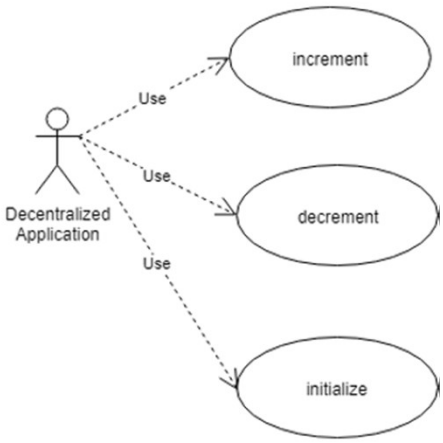
Test blockchain networks

- Ganache :App for test Ethereum blockchain with visual UI and logs
- Kaleido — Use Kaleido for spinning up a consortium blockchain network. Great for PoCs and testing
- Ethereum on Azure — Deployment and governance of consortium Ethereum PoA networks
- Ethereum on Google Cloud — Build Ethereum network based on Proof of Work
- Go Ethereum /Geth (github.com/ethereum/go-ethereum/wiki/geth)
  - Geth is the command line interface for running a full Ethereum node implemented in Go programming language.
- Metamask Firefox plugin (metamask.io/): **cryptocurrency wallet that enables users to store Ether and other ERC-20 tokens**
  - Metamask allows you to run Ethereum dApps right in your browser without running a full Ethereum node.
  - includes a secure identity vault, providing a user interface to manage your identities on different sites & sign blockchain transactions.
- **Quorum private blockchain (jpmorgan.com/global/Quorum)**
  - Quorum is a blockchain protocol specially designed for use in a private blockchain network,
  - Quorum™ is an enterprise-focused version of Ethereum.
  - Quorum is ideal for application requiring high speed & high throughput processing of private transactions within a permissioned group of known participants.
  - addresses specific challenges to blockchain technology adoption within the financial industry, & beyond.

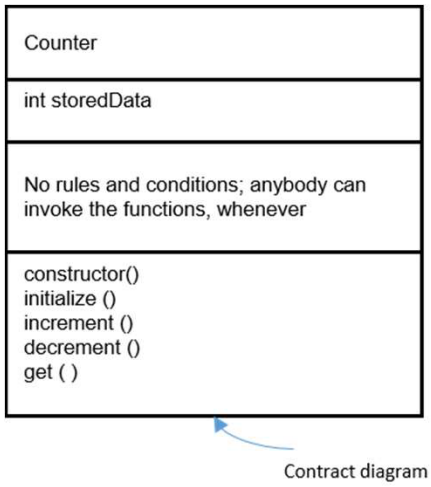
- Solidity — popular smart contract language.
- Truffle — Most popular smart contract development, testing, and deployment framework. Install the CLI via NPM(**the package manager for the Node JavaScript platform)** and able to write smart contracts.
- Metamask — Chrome extension wallet to interact with Dapps.
- Truffle boxes — Packaged components for the Ethereum ecosystem
- EthHub.io — Comprehensive crowdsourced overview of Ethereum- its history, governance, future plans and development resources.
- Infura — Scalable, secure, and reliable access to the Ethereum network.

Contact of a simple counter





Obtain the pseudo code




```
contract Counter;

int storedData;

constructor Counter (int initValue) { }

initialize (int initValue) { }
increment (int value) { }
decrement (int value) { }
int get ()
```

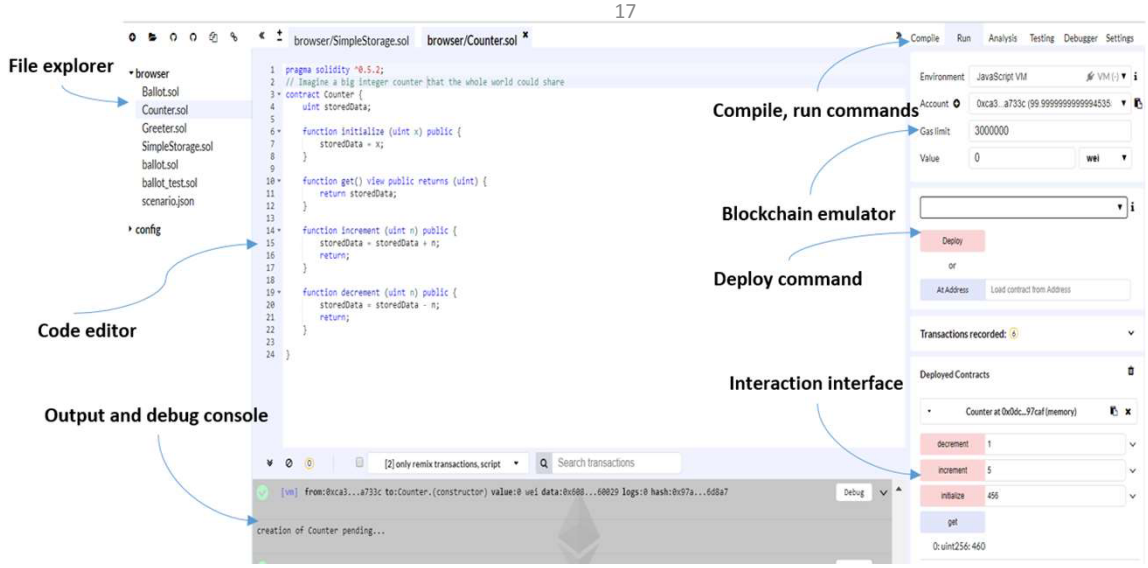
Pseudocode

Code it in Solidity

```
pragma solidity 0.8.7;
contract Counter {
   uint storedData;
   function initialize (uint x) public {
      storedData = x;
   }
   function get() view public returns (uint) {
      return storedData;
   }
   function increment (uint n) public {
      storedData = storedData + n;
      return;
   }
   function decrement (uint n) public {
      storedData = storedData - n;
      return;
   } }
```

Test it in Remix IDE

17



Blockchain programming (Ethereum)