Blocklime
<ACADEMY/>

Solidity Development Tools

Blocklime

1260889-P
Blocklime Technologies Sdn. Bhd.

1

It's complicated..

Lots of components:

- The programming language.. Solidity
- Compiler versions
- Blockchain tools
  - wallet?
  - how to call functions?
  - how to deploy?
  - how to see results?
  - debugger?
- Test network
- Unit tests

2 The parts

1. Prerequisite: node.js and npm
2. Solidity
    a. solc compiler
    b. truffle framework
3. Ethereum network
    a. local test node
    b. test network
    c. main net
4. Metamask
5. IDE & GUI tools

Solidity Compiler

Blocklime

1260889-P
Blocklime Technologies Sdn. Bhd.

3

Solidity compiler: solc

- There are two main versions - solc and solc-js
  - The arguments are not compatible
- Documentation on solc: http://solidity.readthedocs.io/en/v0.4.24/installing-solidity.html
- Documentation on solc-js: https://www.npmjs.com/package/solc
  - Confusingly solc-js is known as solc in npm

3 Solidity compiler: Truffle framework

- The most popular way to develop smart contracts
- Command-line driven development environment
- Install via npm:
  - `npm install -g truffle`
- Automatically installs **solc-js** internally
- Since different versions of Solidity may not be compatible it is a good idea to specify a version when installing truffle:
  - npm install -g truffle@4.1.13
- Can connect to local test network, public test network and main net
- Also works with other Ethereum-like blockchains such as Quorum, Ellaism, Expanse, Qtum etc.

3 Solidity compiler: Truffle framework

Let's create a Truffle project

```
#install truffle
npm install -g truffle@4.1.13

#create our project directory
mkdir mySmartContract
cd mySmartContract

#initialize a truffle project
truffle init

#also, make this a proper npm project
npm init

#as long as we're at it, put this all under git
git init
```

3

Solidity compiler: Truffle framework

Create a file called .gitignore in the project directory (note the dot!)

```
node_modules
build
```

3 Solidity compiler: Truffle framework

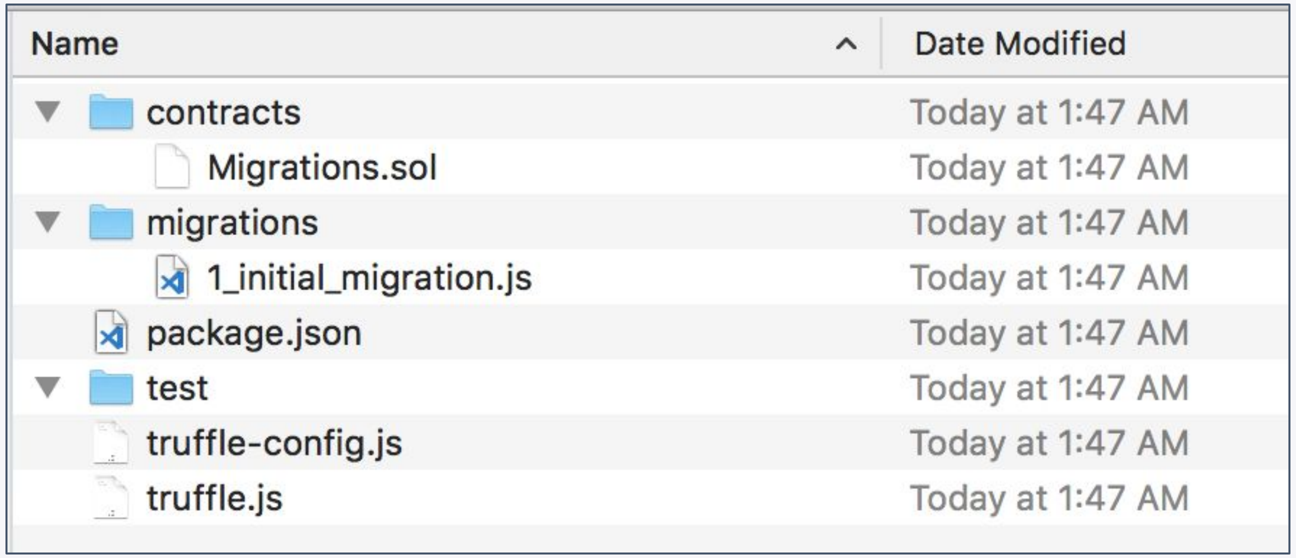
Commit the code!

```
git add .
git commit -m "Initial Commit"
```

3

Solidity compiler: Truffle framework

Notice that Truffle creates a project structure for us.



Name	Date Modified
▼ contracts	Today at 1:47 AM
Migrations.sol	Today at 1:47 AM
▼ migrations	Today at 1:47 AM
1_initial_migration.js	Today at 1:47 AM
package.json	Today at 1:47 AM
▼ test	Today at 1:47 AM
truffle-config.js	Today at 1:47 AM
truffle.js	Today at 1:47 AM

Note: truffle.js and truffle-config.js serve the same purpose but there is a compatibility issue with truffle.js for some Windows users - so delete truffle.js.

3

Solidity compiler: Truffle framework

Now let's try to compile the Migrations.sol smart contract

```
truffle compile
```

Truffle will generate some messages on screen that ends with:

```
    Writing artifacts to ./build/contracts
```

If you look at your project folder you will see a new directory called build.

3

A word on migrations

Truffle uses a migration system to keep track of which contracts are deployed and to order smart contract deployments.

The migration system is itself a smart contract - so all data about migrations are stored on the blockchain itself.

Truffle actually has no built in mechanism to upgrade smart contracts - contracts are immutable

More on migrations: https://truffleframework.com/docs/getting_started/migrations

3 Solidity compiler: Truffle framework

Further reading :-

Truffle documentation: https://truffleframework.com/docs

Tutorials: https://truffleframework.com/tutorials

Ethereum Network

4

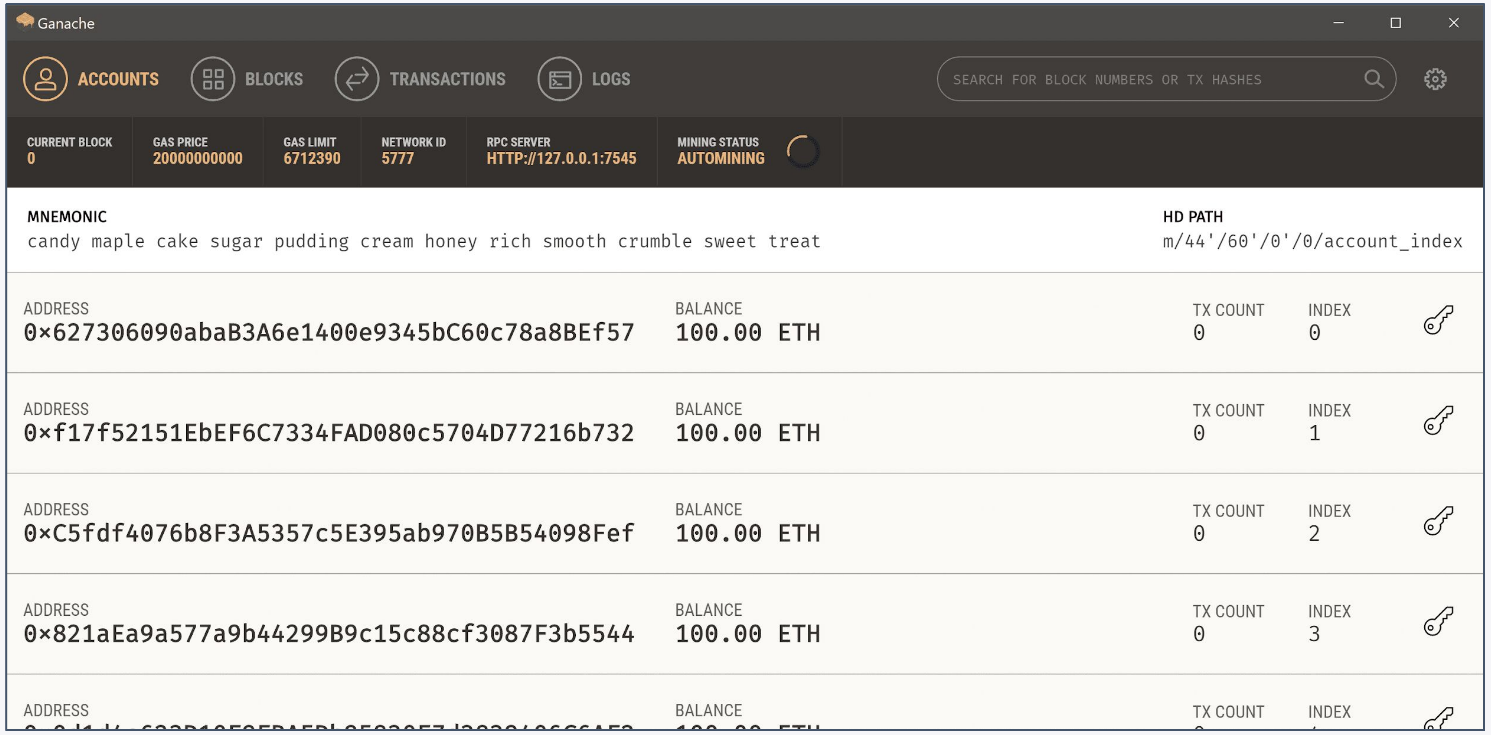
Ethereum: testrpc/ganache-cli

ganache-cli (previously testrpc) is the simplest test environment for Ethereum

- Is part of the Truffle project
- Is a fake Ethereum node - not connected to any network
- Automatically creates 10 temporary wallets pre-loaded with Ether
- Does not fully test gas limits
- Install via npm:
  - `npm install -g ganache-cli`

4 Ethereum: Ganache

Ganache uses the exact same back-end as ganache-cli but has a really nice GUI.



CURRENT BLOCK	GAS PRICE	GAS LIMIT	NETWORK ID	RPC SERVER	MINING STATUS
0	20000000000	6712390	5777	HTTP://127.0.0.1:7545	AUTOMINING

MNEMONIC
candy maple cake sugar pudding cream honey rich smooth crumble sweet treat

HD PATH
m/44'/60'/0'/0/account_index

ADDRESS	BALANCE	TX COUNT	INDEX
0×627306090abaB3A6e1400e9345bC60c78a8BEf57	100.00 ETH	0	0
0×f17f52151EbEF6C7334FAD080c5704D77216b732	100.00 ETH	0	1
0×C5fdf4076b8F3A5357c5E395ab970B5B54098Fef	100.00 ETH	0	2
0×821aEa9a577a9b44299B9c15c88cf3087F3b5544	100.00 ETH	0	3
[illegible]	[illegible]	[illegible]	[illegible]

4 Ethereum: Ganache

Ganache uses the exact same back-end as ganache-cli but has a really nice GUI.

- Works exactly like ganache-cli
- Has built-in block explorer
- Download from the Truffle website: https://truffleframework.com/ganache

4 Ethereum: geth

geth is the short name for the go-ethereum project

- It is the official client from the Ethereum project (there is a c++ client but all new features are first tested in geth
- "geth" is the name of the command-line executable
- There are several ways to install geth: https://github.com/ethereum/go-ethereum/wiki/Installing-Geth
- For the purpose of this lecture just download the executable: https://ethereum.github.io/go-ethereum/downloads/

4 Ethereum: geth

- geth can be used to:
  - Connect to Ethereum main net
  - Connect to Ropsten or Rinkeby test networks
  - Connect to a custom test network
  - Create your own private Ethereum network
  - Mine Ethereum
  - Act as a wallet
- Truffle does not include a node except for Ganache. If you use Truffle you need geth to connect to the real blockchain
- Documentation: https://github.com/ethereum/go-ethereum/wiki

Metamask

5 Metamask

- Is a browser plugin allowing the browser to connect to Ethereum
- Is required by a lot of online tools
  - Online Solidity IDEs
  - Test network faucets
- Available on Chrome, Firefox, Opera and Brave
  - (sorry Safari & Edge users)

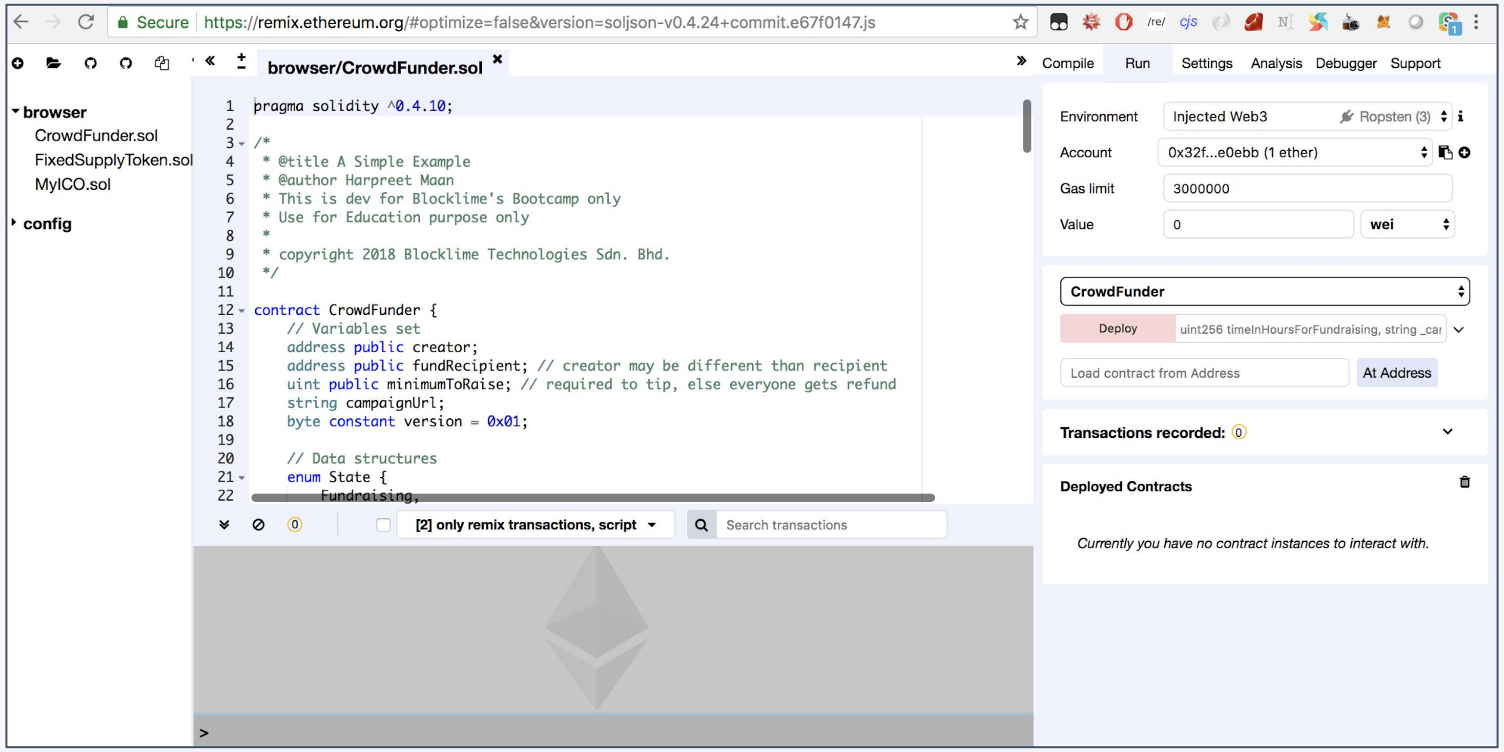
Solidity IDE

Blocklime

6

IDE: Remix

https://remix.ethereum.org/



https://remix.ethereum.org/#optimize=false&version=soljson-v0.4.24+commit.e67f0147.js

```
pragma solidity ^0.4.10;

/*
 * @title A Simple Example
 * @author Harpreet Maan
 * This is dev for Blocklime's Bootcamp only
 * Use for Education purpose only
 *
 * copyright 2018 Blocklime Technologies Sdn. Bhd.
 */

contract CrowdFunder {
    // Variables set
    address public creator;
    address public fundRecipient; // creator may be different than recipient
    uint public minimumToRaise; // required to tip, else everyone gets refund
    string campaignUrl;
    byte constant version = 0x01;

    // Data structures
    enum State {
        Fundraising,
```

Environment: Injected Web3 — Ropsten (3)
Account: 0x32f...e0ebb (1 ether)
Gas limit: 3000000
Value: 0 wei

CrowdFunder
Deploy: uint256 timeInHoursForFundraising, string _car
Load contract from Address

Transactions recorded: 0

Deployed Contracts

Currently you have no contract instances to interact with.

Blocklime

6

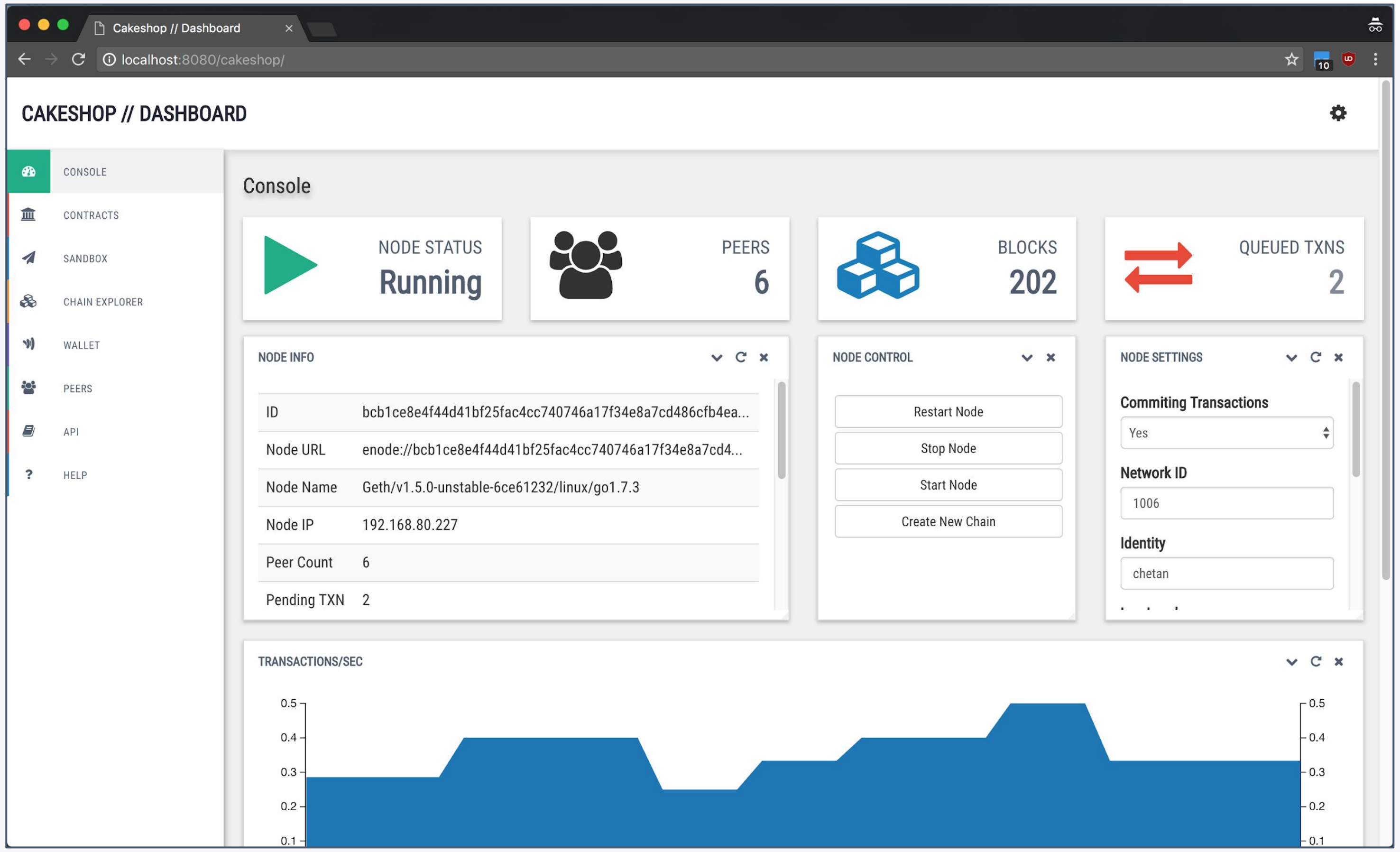
IDE: Remix

https://remix.ethereum.org/

- Is a browser based IDE
- Integrates with Metamask
- Can actually be used to deploy on main net
- Good for small test contracts
- Not so good for real work
- Includes built-in compiler and linter
- Highly configurable
- Debugging is a bit confusing

6 IDE: Cakeshop

JPMorganChase



CAKESHOP // DASHBOARD

Console

NODE STATUS Running

PEERS 6

BLOCKS 202

QUEUED TXNS 2

NODE INFO

ID	bcb1ce8e4f44d41bf25fac4cc740746a17f34e8a7cd486cfb4ea...
Node URL	enode://bcb1ce8e4f44d41bf25fac4cc740746a17f34e8a7cd4...
Node Name	Geth/v1.5.0-unstable-6ce61232/linux/go1.7.3
Node IP	192.168.80.227
Peer Count	6
Pending TXN	2

NODE CONTROL

NODE SETTINGS

Commiting Transactions: Yes

Network ID: 1006

Identity: chetan

TRANSACTIONS/SEC

Blocklime

1260889-P

Blocklime Technologies Sdn. Bhd.

6

IDE: Cakeshop

JPMorganChase

- Developed by JPMorganChase & Co.
- Java app
- For Quorum, not Ethereum
  - ..but Quorum also uses Solidity
  - can directly interface with Ethereum's version of geth
- All-in-one blockchain IDE
  - includes block explorer (chain explorer)
  - includes Solidity IDE
  - includes wallet
- Download from: https://github.com/jpmorganchase/cakeshop/releases
- Documentation: https://github.com/jpmorganchase/cakeshop/wiki/Cakeshop-Overview

Honorable Mentions

8 Other useful things

1. Openzeppelin
   https://openzeppelin.org/
   a. A really useful library of tested Solidity code
   b. Significantly reduce development time

2. Infura
   https://infura.io/
   a. A cloud Ethereum node service
   b. Saves you the hassle of configuring your own server