CONSENSYS

Breaking into Blockchain

Leveraging your current abilities & building in a new space

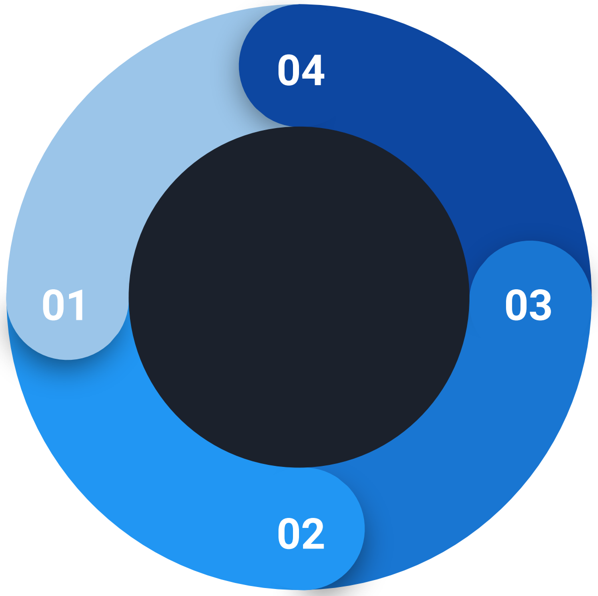
Written by: Robbie Kruszynski

Table of Contents

Lifecycle of a developer

Start to learn a language

Become wealthy with bugs
and frustration



Realize you need to learn
more to implement
features you want to add

Understand your mistakes,
fix them, and add
complexity to your project

The Blockchain Ecosystem

The Blockchain Ecosystem

The standard web vs the decentralized web

Tokens, fungible and non-fungible

Smart contracts

Nodes and Blocks

Mainnet, testnet

ConsenSys developer portal + other resources

Standard web vs Decentralized

Standard web

Client-server model, databases, cloud computing, content distribution networks. Architecture defines “owner” and single source-of-truth. User connects to server. Cloud and CDN networks operate behind the scenes but support a centralized architecture.

Decentralized web

While modern networks include decentralized CDNs and cloud infrastructure, they may not be *politically* decentralized, even if they are architecturally or logically decentralized. There is an owner, controller, or operator. Decentralized, blockchain-enabled web apps include connections to networks like Ethereum, to provide true decentralized processes.

Tokens

Fungible

“Fungible” means not-unique; you can trade one fungible thing for another, and it doesn’t matter. If you have one dollar bill, or a share of common Amazon stock, it’s worth the same as any other dollar bill or other share, and normally it doesn’t matter which one you have. Contrast this with, for example, baseball cards. Cryptocurrencies are fungible. Gold is fungible. Hay is fungible.

Non-Fungible

If it’s not fungible, it’s non-fungible. Collectibles are a good example of fungible things: if you have a signed Babe Ruth homerun ball, this is not the same as any other baseball. Cars, properties, artworks, are all examples of non-fungible things. Each one has unique qualities, so you can’t swap one for another.

Fungible Example

FEDERAL RESERVE NOTE
THE UNITED STATES OF AMERICA
THIS NOTE IS LEGAL TENDER FOR ALL DEBTS, PUBLIC AND PRIVATE
L 49560866 I
WASHINGTON, D.C.
ONE
SERIES 2003 A
Treasurer of the United States.
Secretary of the Treasury.
WASHINGTON
ONE DOLLAR

Non-Fungible Example

Ethereum & Smart Contracts

The Ethereum Virtual Machine

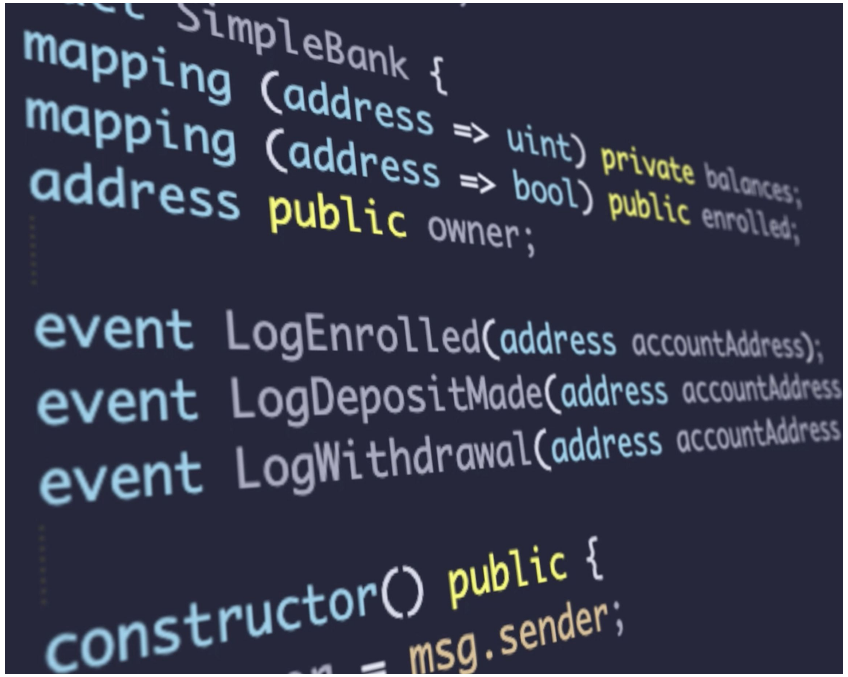
A key component of Ethereum is that it provides a Turing-complete virtual machine that can run code, entirely on the blockchain. Called the “world computer”, the EVM runs code that is deployed to the blockchain. Such code is referred to as a “smart contract”.

```
    SimpleBank {
mapping (address => uint) private balances;
mapping (address => bool) public enrolled;
address public owner;

event LogEnrolled(address accountAddress);
event LogDepositMade(address accountAddress
event LogWithdrawal(address accountAddress

constructor() public {
     = msg.sender;
```

Smart Contracts

Blockchain programs

Smart contracts are small pieces of executable code that are permanent and unchangeable. In Ethereum, smart contracts have addresses just like user accounts do. You can send Ether to a smart contract in the same way that you do to any other address.



```
SimpleBank {
mapping (address => uint) private balances;
mapping (address => bool) public enrolled;
address public owner;

event LogEnrolled(address accountAddress);
event LogDepositMade(address accountAddress
event LogWithdrawal(address accountAddress

constructor() public {
    = msg.sender;
```

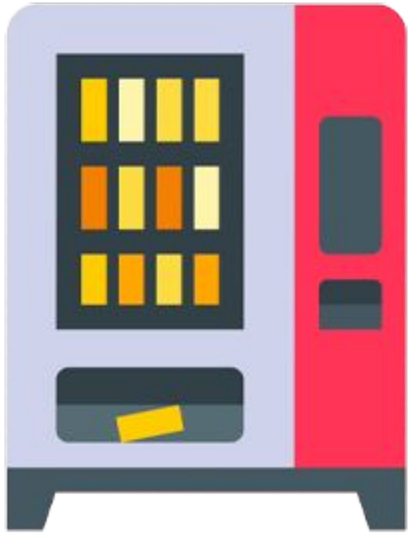
Solidity & Contracts

What is a smart contract?

Consider first a vending machine.

It contains something of value (a soda), and you can provide an input (coins) to release that value.

So, the vending machine keeps an asset safe and moves that asset once the input requirements are met. This kind of functionality is similar to how a smart contract works. The smart contract can protect and store a digital valuable and release it depending on user input.

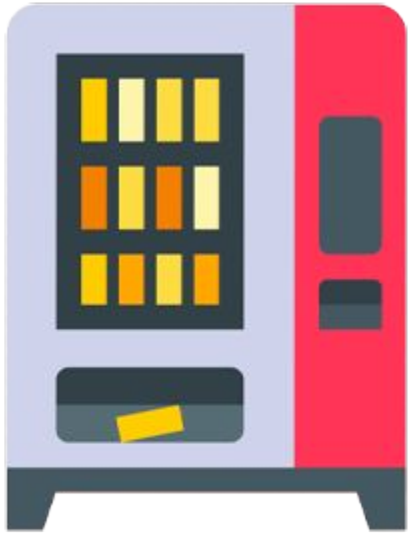


Solidity & Contracts

What is a smart contract?

Keep in mind that this is a high level example of a use a smart contract can provide.

They can be very powerful and programmed to execute a number of complex actions. This example should just help you wrap your head around basic functionality.



Nodes

Nodes make up the network

A blockchain network is *entirely* comprised of computers running node software and interacting with each other. There is no central authority. While the reality is more complicated, you can think of each node as storing a copy of the blockchain with protocols in place to ensure that new blocks are added to every node's chain in a consistent, organized, and agreed-upon way.

Blocks

What's inside a block?

While different blockchain implementations will have different block structures, fundamentally a block contains a list of transactions: values, senders, and receivers, and a variety of other information that we won't get into right now.

Mainnet, Testnet

Mainnet

In Ethereum, Mainnet is the primary, value-carrying network that is agreed to be the “official” place where real currency is traded. But Ethereum networks don’t have to be related to Mainnet. A developer could start a small Ethereum network on their own system—and many do, to develop applications.

Testnets

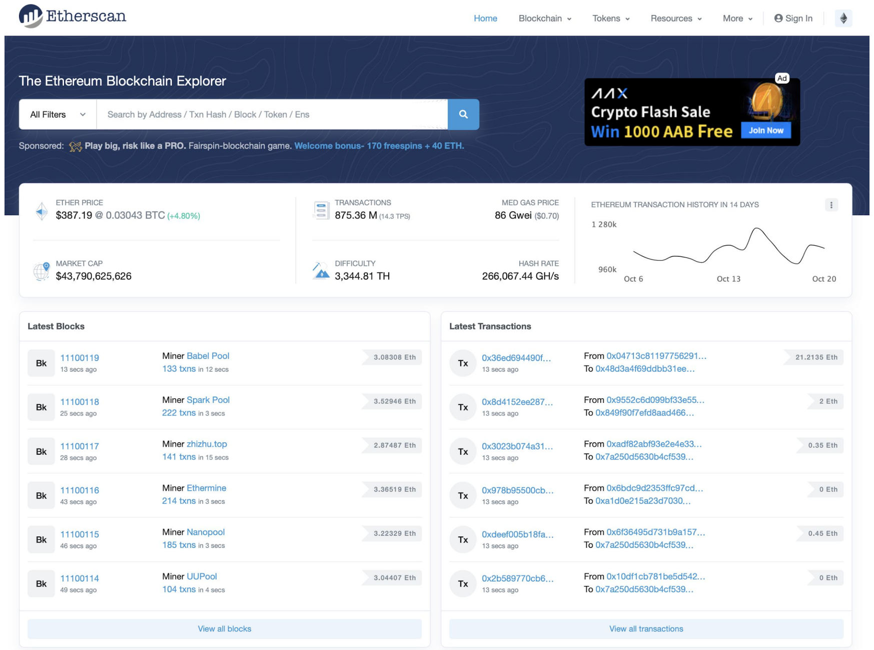
Testnets are smaller networks that devs can use to test their applications when they want to move beyond home-spun networks. This allows for almost-real-world testing of application functionality in an environment where real value isn’t at stake. Testnets include Ropsten, Kovan, Rinkeby, and others.

Blockchain Explorers

Etherscan.io

Etherscan allows you to explore and search the Ethereum blockchain for transactions, addresses, tokens, prices and other activities taking place on Ethereum.

There are many other services that let you explore Ethereum—not to mention other blockchains. Etherchain, Ethplorer, are some others.



The Ethereum Blockchain Explorer

Sponsored: **Play big, risk like a PRO.** Fairspin-blockchain game. **Welcome bonus- 170 freespins + 40 ETH.**

ETHER PRICE
$387.19 @ 0.03043 BTC (+4.80%)

MARKET CAP
$43,790,625,626

TRANSACTIONS
875.36 M (14.3 TPS)

MED GAS PRICE
86 Gwei ($0.70)

DIFFICULTY
3,344.81 TH

HASH RATE
266,067.44 GH/s

ETHEREUM TRANSACTION HISTORY IN 14 DAYS
1 280k
960k
Oct 6 Oct 13 Oct 20

Latest Blocks

Bk	11100119 13 secs ago	Miner Babel Pool 133 txns in 12 secs	3.08308 Eth
Bk	11100118 25 secs ago	Miner Spark Pool 222 txns in 3 secs	3.52946 Eth
Bk	11100117 28 secs ago	Miner zhizhu.top 141 txns in 15 secs	2.87487 Eth
Bk	11100116 43 secs ago	Miner Ethermine 214 txns in 3 secs	3.36519 Eth
Bk	11100115 46 secs ago	Miner Nanopool 185 txns in 3 secs	3.22329 Eth
Bk	11100114 49 secs ago	Miner UUPool 104 txns in 4 secs	3.04407 Eth

Latest Transactions

Tx	0x36ed694490f… 13 secs ago	From 0x04713c81197756291… To 0x48d3a4f69ddbb31ee…	21.2135 Eth
Tx	0x8d4152ee287… 13 secs ago	From 0x9552c6d099bf33e55… To 0x849f90f7efd8aad466…	2 Eth
Tx	0x3023b074a31… 13 secs ago	From 0xadf82abf93e2e4e33… To 0x7a250d5630b4cf539…	0.35 Eth
Tx	0x978b95500cb… 13 secs ago	From 0x6bdc9d2353ffc97cd… To 0xa1d0e215a23d7030…	0 Eth
Tx	0xdeef005b18fa… 13 secs ago	From 0x6f36495d731b9a157… To 0x7a250d5630b4cf539…	0.45 Eth
Tx	0x2b589770cb6… 13 secs ago	From 0x10df1cb781be5d542… To 0x7a250d5630b4cf539…	0 Eth

MetaMask & Transactions

MetaMask & Transactions

Transactions

The UX of Transactions

From the user perspective, a transaction is a transfer of value from one account to another. This account can be owned by an individual, or be the account of a smart contract. Transactions can include data that is processed by smart contracts and used in distributed apps.

Signing

MetaMask is a mobile wallet app and browser extension that can "store" Ethereum tokens, but also is an interface that gives users the ability to sign transactions. Signing is the process used by a blockchain network to authenticate a transaction.

Only the possessor of a private key can sign a transaction for a corresponding account. Nodes validate transactions mathematically to ensure they are legitimately signed.

How Security Works… Where are the passwords?

Private keys, cryptography, hash functions

Blockchains and accounts are secured largely through public key encryption. From the perspective of a blockchain, anyone who possesses a private key can control any corresponding addresses, which is why private keys must remain private. The details of security and cryptography on blockchains is fascinating, but outside the scope of this introduction.

Section 2

Let’s get technical

What we will achieve in this section

We are going to make our very own HelloWorld project!

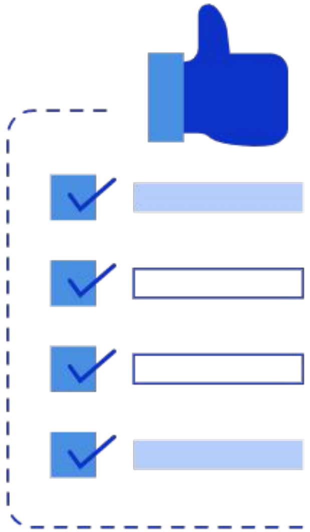
This project will explore some of the tools readily available to assist with your development and ensure your environment is safe for exploration!

We will create a smart contract that we will deploy to our local blockchain testing environment as well as interact with said contract on an IDE called Remix!

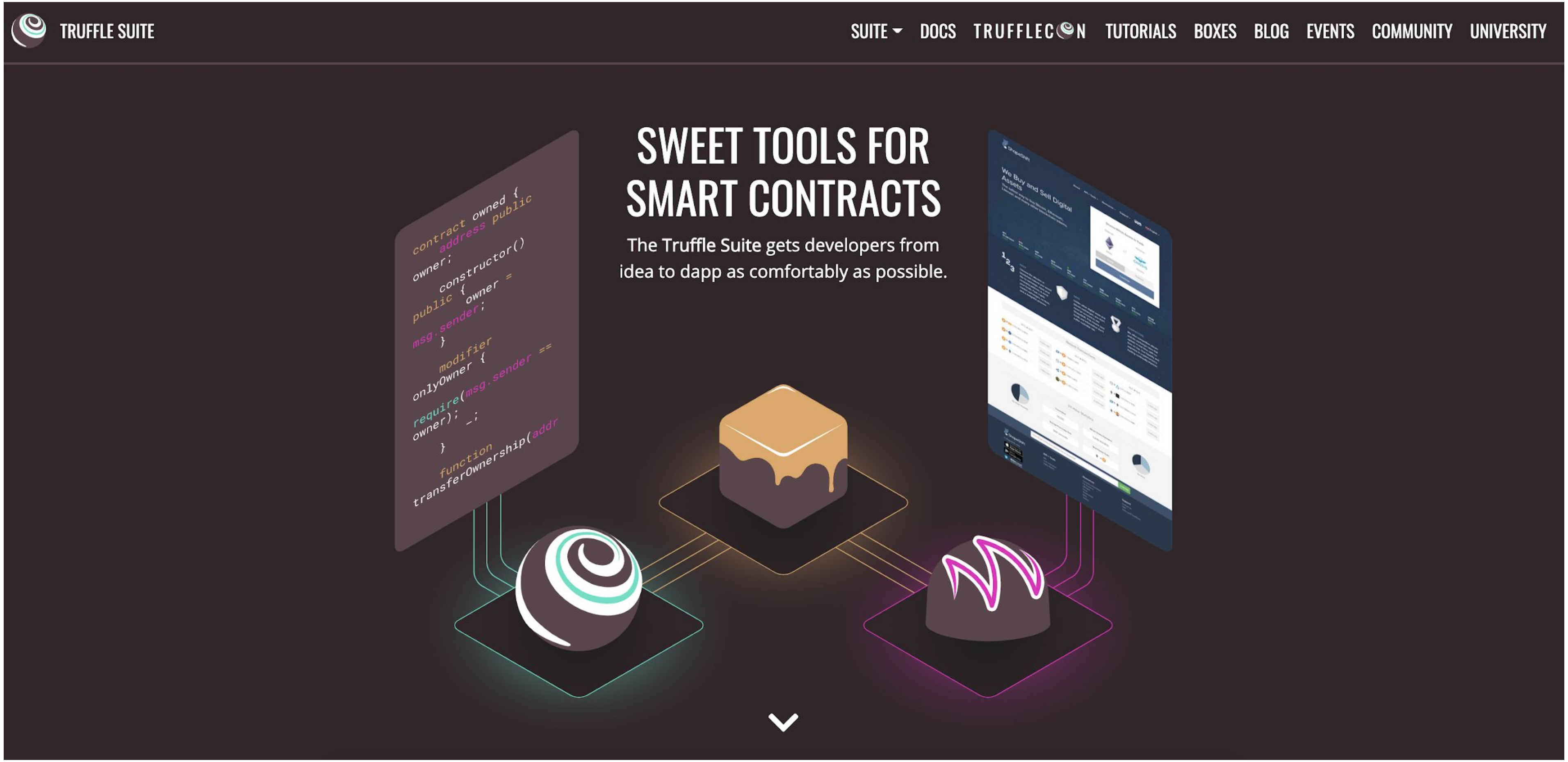
We will both cover and use bleeding edge technology to give you a better understanding of how all of these pieces fit together!

What we will cover

- The tools we will be using
- Creating a new project
- Taking a quick tour of the scaffolding
- Creating our HelloWorld.sol contract
- Updating our migrations directory
- Adjusting our config file and updating our network information
- Deploying locally using Ganache
- Deploying and interacting with our contract using Remix



Tools we will use



SWEET TOOLS FOR SMART CONTRACTS

The Truffle Suite gets developers from idea to dapp as comfortably as possible.

```
contract owned {
    address public
owner;
    constructor()
public {
        owner =
msg.sender;
    }
    modifier
onlyOwner {
        require(msg.sender ==
owner);
        _;
    }
    function
transferOwnership(addr
```

Truffle & Ganache

Truffle

Truffle is an excellent development environment that allows you to both connect with and test using the Ethereum Virtual Machine.

Truffle was created to make development easier, and with interactions occurring **locally** this helps reduce the stress of deployment on both a testnet (such as Ropsten or Rinkeby) and mainnet.

Install

```
npm install truffle -g
```

Truffle & Ganache

Ganache

A quick rundown on ganache is that it's a personal blockchain you can use locally to quickly spin up and test functionality of projects.

Ganache is a tool you can use throughout the entirety of the development cycle. Not only are you able to develop, but also deploy, and test your dApps. All of this happens locally on your machine so this is the lowest friction / risk environment to work on your projects!

Install

```
npm install -g ganache-cli
```

for the command line interface

Download

https://www.trufflesuite.com/ganache

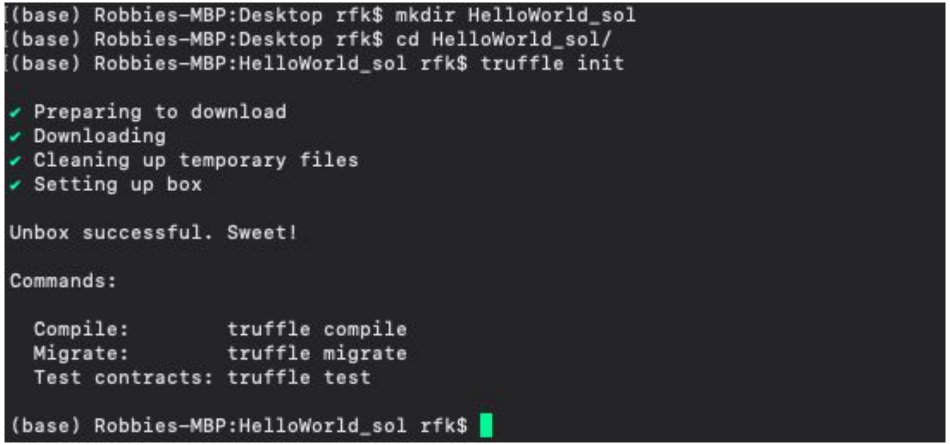
Time to spin up our project

Create a new Project

We will navigate to our Desktop now that we have truffle and ganache installed and make a new directory. Then we'll go ahead and run:

```
truffle init
```

This command will create three directories contracts, migrations, and test, along with three files: Migrations.sol, 1_initial_migrations.js, and truffle-config.js



```
(base) Robbies-MBP:Desktop rfk$ mkdir HelloWorld_sol
(base) Robbies-MBP:Desktop rfk$ cd HelloWorld_sol/
(base) Robbies-MBP:HelloWorld_sol rfk$ truffle init

✔ Preparing to download
✔ Downloading
✔ Cleaning up temporary files
✔ Setting up box

Unbox successful. Sweet!

Commands:

  Compile:        truffle compile
  Migrate:        truffle migrate
  Test contracts: truffle test

(base) Robbies-MBP:HelloWorld_sol rfk$ 
```

Let’s have a look under the hood

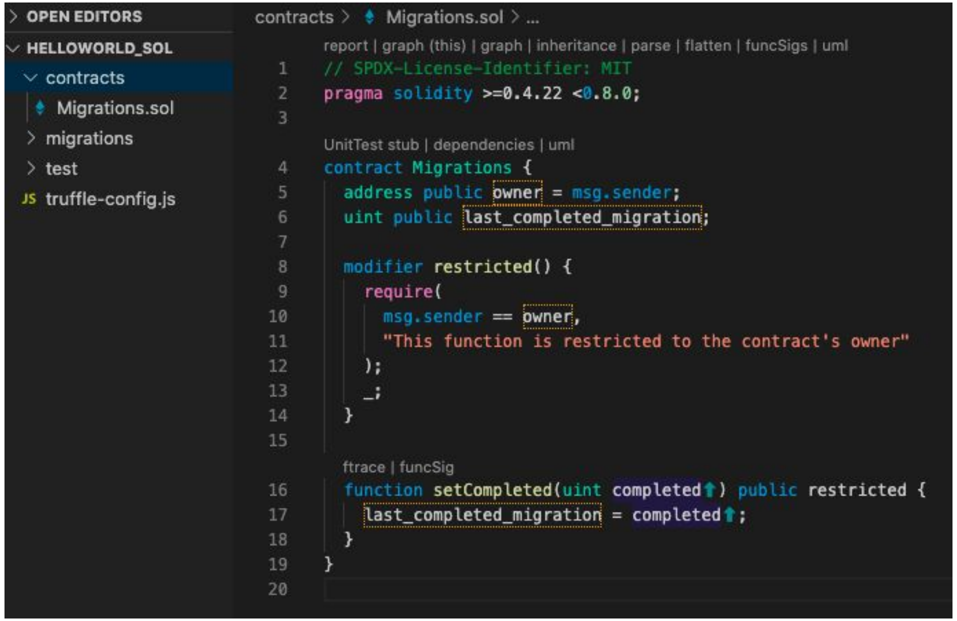
Contracts

Contract Directory

contracts/ will store all your Solidity (.sol files). This is where you will add your smart contracts that you need at compile time.

Migrations.sol is a complete, fully working smart contract written in Solidity.

It is used by truffle to ensure that your project's deployment to the blockchain is carried out in the proper sequence.


OPEN EDITORS
HELLOWORLD_SOL
contracts
Migrations.sol
migrations
test
truffle-config.js
contracts > Migrations.sol > ...
report | graph (this) | graph | inheritance | parse | flatten | funcSigs | uml
UnitTest stub | dependencies | uml
ftrace | funcSig

```
// SPDX-License-Identifier: MIT
pragma solidity >=0.4.22 <0.8.0;

contract Migrations {
  address public owner = msg.sender;
  uint public last_completed_migration;

  modifier restricted() {
    require(
      msg.sender == owner,
      "This function is restricted to the contract's owner"
    );
    _;
  }

  function setCompleted(uint completed) public restricted {
    last_completed_migration = completed;
  }
}
```

Migrations

Migrations

Migrations, generally speaking, are ways for developers to automate the deployment of data and its supporting structures.

Migrations are Javascript files that allow you to deploy your contracts to the Ethereum network.

Truffle migrations enable us to "push" the smart contracts to the Ethereum blockchain (either local, tesnet or mainnet) and to set up necessary steps for linking contracts with other contracts as well as populate contracts with initial data.

A way to think about migration files is they are predominantly responsible for the staging and deployment of your tasks.

Another way to think about them is migrations are a set of managed deployment scripts, as you update your work a log of your previously run migrations is recorded on-chain through a build-in Migrations contract.

Tests

Tests

test/ is were we will keep our tests! It can contain .js or .sol files, based on your choice of language.

Tests are conditionals you write against your contract to ensure it behaves as expected. You want to write tests that will check how your contract reacts to a range of user inputs.

If there was an exploit in your code, your tests should help you find it, not another person!

Truffle-config

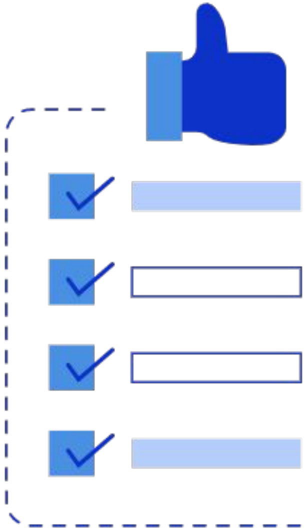
Truffle-config.js

truffle-config.js is the main configuration for your Truffle project. This is where we define what networks to use, gas usages, addresses to deploy with, and a few other variables.

This file acts as your walkie talkie to interact with the blockchain!

What we have covered

- ✓ The tools we will be using
- ✓ Creating a new project
- ✓ A quick tour of the scaffolding
- Creating our HelloWorld.sol contract
- Deploying and interacting with our contract using Remix



Hello World

Time to code

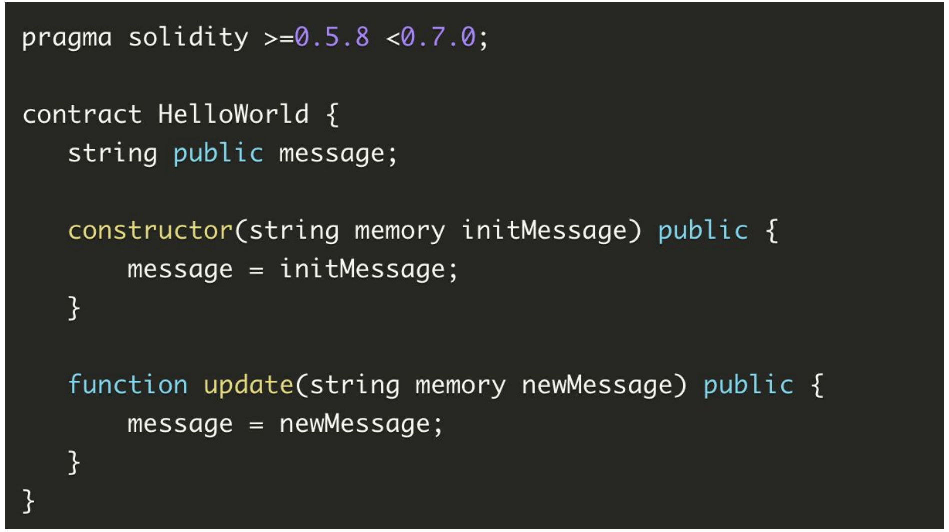
Solidity & Contracts

What is Solidity?

Solidity is an very popular object-oriented, high-level language for implementing smart contracts that run on the Ethereum Virtual Machine (EVM).

Smart contracts are programs which govern the behavior of accounts within the Ethereum state.

If you've never looked at a line of Solidity prior, but are familiar with C and or JavaScript you will notice more than a few similarities.



```
pragma solidity >=0.5.8 <0.7.0;

contract HelloWorld {
    string public message;

    constructor(string memory initMessage) public {
        message = initMessage;
    }

    function update(string memory newMessage) public {
        message = newMessage;
    }
}
```

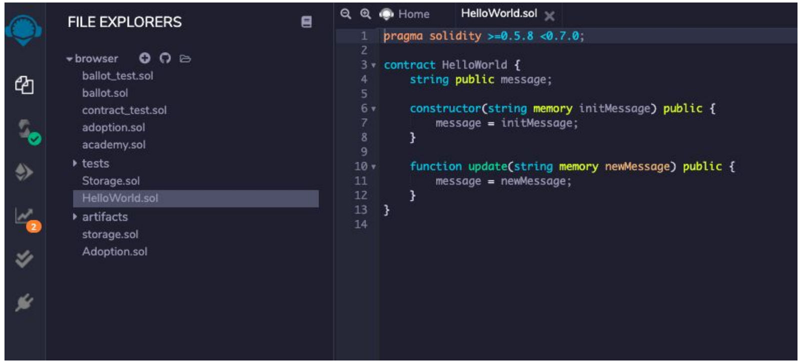
Remix

Let’s interact with our Contract

Remix

Remix is a powerful, open source tool that helps you write Solidity contracts straight from the browser and supports testing, debugging, and contract deployment to name a few key features.

Remix should be considered a staple tool for a developer in their building process. Starting with local deployment (as we did earlier) with Ganache is an excellent foundation, moving from local deployment we can experiment and interact with our contract on Remix.



```
pragma solidity >=0.5.8 <0.7.0;

contract HelloWorld {
    string public message;

    constructor(string memory initMessage) public {
        message = initMessage;
    }

    function update(string memory newMessage) public {
        message = newMessage;
    }
}
```

Resources

BUIDL Network

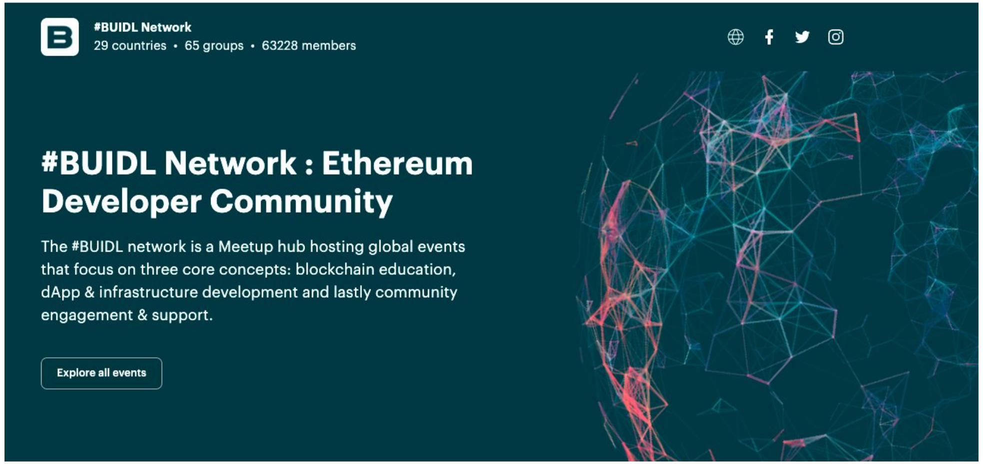
Stay up to date by keeping up with the BUIDL Network!

The BUIDL Network is a meetup hub hosting global events that focus on three core concepts

Blockchain Education

Infrastructure development

Community engagement


#BUIDL Network
29 countries • 65 groups • 63228 members

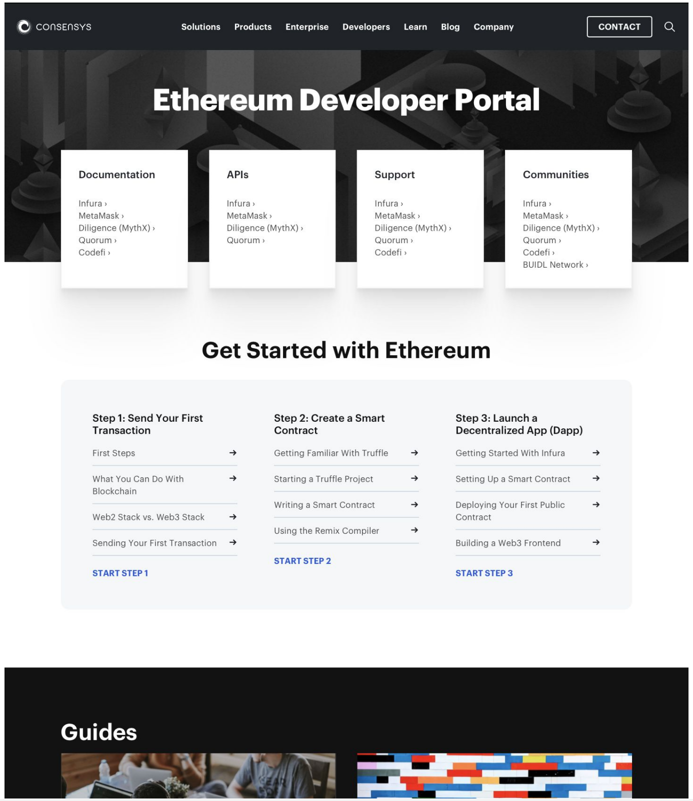
#BUIDL Network : Ethereum Developer Community

The #BUIDL network is a Meetup hub hosting global events that focus on three core concepts: blockchain education, dApp & infrastructure development and lastly community engagement & support.

ConsenSys.net/developers

Visit our developer portal

Our developer portal is evolving and is a great starting point for devs to connect with our products, APIs and documentation, training, events, and more.



Ethereum Developer Portal

Documentation
Infura ›
MetaMask ›
Diligence (MythX) ›
Quorum ›
Codefi ›

APIs
Infura ›
MetaMask ›
Diligence (MythX) ›
Quorum ›

Support
Infura ›
MetaMask ›
Diligence (MythX) ›
Quorum ›
Codefi ›

Communities
Infura ›
MetaMask ›
Diligence (MythX) ›
Quorum ›
Codefi ›
BUIDL Network ›

Get Started with Ethereum

Step 1: Send Your First Transaction
First Steps →
What You Can Do With Blockchain →
Web2 Stack vs. Web3 Stack →
Sending Your First Transaction →
START STEP 1

Step 2: Create a Smart Contract
Getting Familiar With Truffle →
Starting a Truffle Project →
Writing a Smart Contract →
Using the Remix Compiler →
START STEP 2

Step 3: Launch a Decentralized App (Dapp)
Getting Started With Infura →
Setting Up a Smart Contract →
Deploying Your First Public Contract →
Building a Web3 Frontend →
START STEP 3

Guides

CryptoZombies

Perhaps you have heard of / played

- Flexbox froggy

- Grid garden

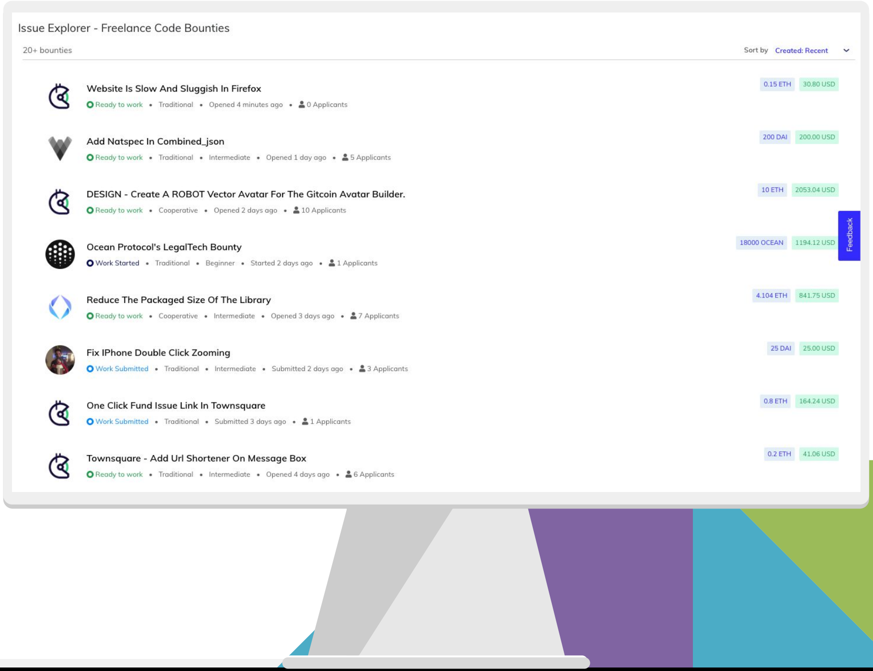
Well we have one too!

It’s called cryptozombies and it guides you through writing your very own smart contract!

GitCoin Open Source Contributions vs Bounties!

Contribute to open source, and get paid.
GitCoin is a company that is incentivising developers to contribute to open source projects.

Companies are creating bounties to expedite their needs by opening the door to a much larger community.



Issue Explorer - Freelance Code Bounties

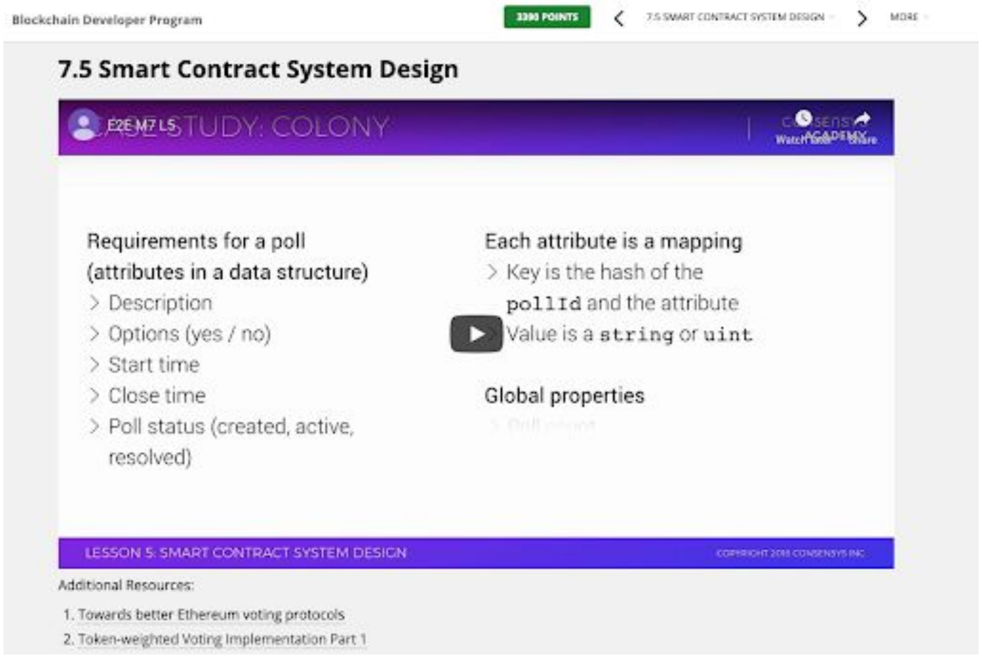
20+ bounties — Sort by Created: Recent

Bounty	Status	Details	Value	USD
Website Is Slow And Sluggish In Firefox	Ready to work	Traditional • Opened 4 minutes ago • 0 Applicants	0.15 ETH	30.80 USD
Add Natspec In Combined_json	Ready to work	Traditional • Intermediate • Opened 1 day ago • 5 Applicants	200 DAI	200.00 USD
DESIGN - Create A ROBOT Vector Avatar For The Gitcoin Avatar Builder.	Ready to work	Cooperative • Opened 2 days ago • 10 Applicants	10 ETH	2053.04 USD
Ocean Protocol's LegalTech Bounty	Work Started	Traditional • Beginner • Started 2 days ago • 1 Applicants	18000 OCEAN	1194.12 USD
Reduce The Packaged Size Of The Library	Ready to work	Cooperative • Intermediate • Opened 3 days ago • 7 Applicants	4.104 ETH	841.75 USD
Fix IPhone Double Click Zooming	Work Submitted	Traditional • Intermediate • Submitted 2 days ago • 3 Applicants	25 DAI	25.00 USD
One Click Fund Issue Link In Townsquare	Work Submitted	Traditional • Submitted 3 days ago • 1 Applicants	0.8 ETH	164.24 USD
Townsquare - Add Url Shortener On Message Box	Ready to work	Traditional • Intermediate • Opened 4 days ago • 6 Applicants	0.2 ETH	41.06 USD

ConsenSys Academy

There are real efforts / programs to help developers learn about the tech required to explore this space.

Built out courses to guide you through use- cases along with smart contract fundamentals.


Blockchain Developer Program
3390 POINTS
7.5 SMART CONTRACT SYSTEM DESIGN
MORE
E2E M7 L5
Watch later
Share

7.5 Smart Contract System Design

CASE STUDY: COLONY

Requirements for a poll (attributes in a data structure)
- Description
- Options (yes / no)
- Start time
- Close time
- Poll status (created, active, resolved)

Each attribute is a mapping
- Key is the hash of the `pollId` and the attribute
- Value is a `string` or `uint`

Global properties

LESSON 5: SMART CONTRACT SYSTEM DESIGN

COPYRIGHT 2018 CONSENSYS INC

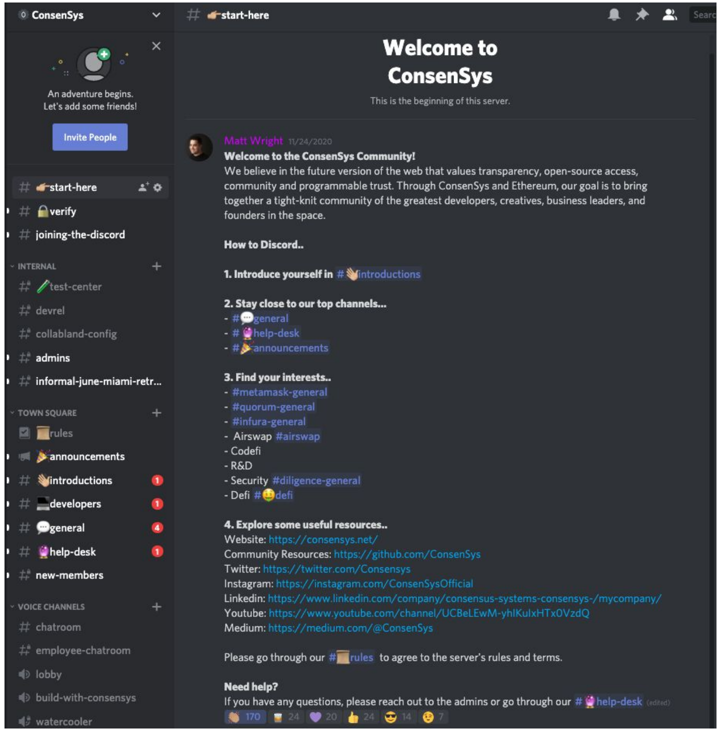
Additional Resources:

1. Towards better Ethereum voting protocols
2. Token-weighted Voting Implementation Part 1

ConsenSys Discord

Need help starting your journey as a developer in the Web3 space?

Want to chat with project members from MetaMask, Infura, Truffe + more? Looking to discuss DeFi, NFTs, upcoming events? Looking to share your project or and teammates for hackathons? Looking to be part of our weekly calls?



Welcome to ConsenSys

This is the beginning of this server.

Matt Wright 11/24/2020

Welcome to the ConsenSys Community!

We believe in the future version of the web that values transparency, open-source access, community and programmable trust. Through ConsenSys and Ethereum, our goal is to bring together a tight-knit community of the greatest developers, creatives, business leaders, and founders in the space.

How to Discord..

1. Introduce yourself in #introductions

2. Stay close to our top channels...
- #general
- #help-desk
- #announcements

3. Find your interests..
- #metamask-general
- #quorum-general
- #infura-general
- Airswap #airswap
- Codefi
- R&D
- Security #diligence-general
- Defi #defi

4. Explore some useful resources..
Website: https://consensys.net/
Community Resources: https://github.com/ConsenSys
Twitter: https://twitter.com/Consensys
Instagram: https://instagram.com/ConsenSysOfficial
Linkedin: https://www.linkedin.com/company/consensus-systems-consensys-/mycompany/
Youtube: https://www.youtube.com/channel/UCBeLEwM-yhlKulxHTx0VzdQ
Medium: https://medium.com/@ConsenSys

Please go through our #rules to agree to the server's rules and terms.

Need help?
If you have any questions, please reach out to the admins or go through our #help-desk (edited)

170 24 20 24 14 7

Questions