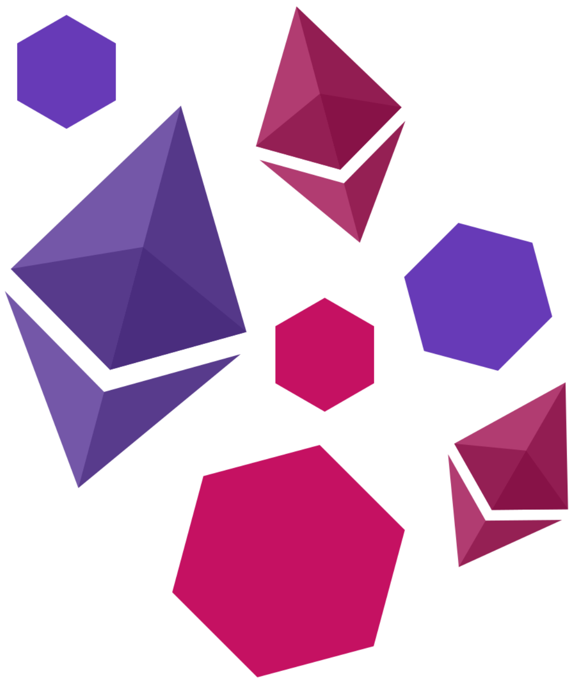
Mastering Smart Contracts

A no-nonsense guide to writing smart contracts for **Ethereum Blockchain**



Sandeep Panda

Mastering Smart Contracts

A no-nonsense guide to writing smart contracts for Ethereum blockchain

Sandeep Panda

This book is for sale at http://leanpub.com/smart-contracts

This version was published on 2018-03-18

Tweet This Book!

Please help Sandeep Panda by spreading the word about this book on Twitter!

The suggested hashtag for this book is #Ethereum.

Find out what other people are saying about the book by clicking on this link to search for this hashtag on Twitter:

#Ethereum

To my Mom and Dad who taught me to love books. To my brother Preetish for being a constant source of inspiration. And to Ipseeta and Fazle for always believing in me.

Contents

Introduction

6 months back when people used to mention terms like Blockchain, Ethereum, Smart Contracts etc, I used to picturize a giant blackhole. That is mainly because the concepts were seemingly complex and I didn't have enough time to read more about them. But the moment I started digging into Ethereum blockchain and Smart Contracts my perception changed completely. The trick is:

> Accept certain things as they are. Don't worry about low level stuff until you really need to know them. Do you worry about how AWS works internally when you host your websites with them?

So, without wasting any time let's try to understand various buzzwords and commonly used terminologies. It's okay if you don't understand everything.

What's a Blockchain

As the name suggests, a blockchain is just a chain of blocks arranged sequentually. Do note that blockchain is just a concept/idea and there are various implementations such as Bitcoin blockchain, Ethereum blockchain and various others (the first two being tremendously successful). So, our first lesson of the chapter is:

> Blockchain is a concept. Bitcoin, Ethereum etc are the actual implementations.

I won't go deeper into general blockchain concepts, but here are 5 key points that will get you up to speed.

- A blockchain network consists of thousands of computers (referred to as nodes). Each computer contains all the records of the blockchain so that no single person can alter or tampter the data. If someone wanted to tamper a particular record, they would have to change it across all of the nodes. That's why blockchain networks are known as decentralized i.e. no single authority controls it.

- Each block in the blockchain wraps multiple incoming transactions and once the block is validated it's added to the chain permanently.

- Every block in the blockchain stores the unique hash of the previous block. This guarantees that no single block can be modified or tampered by any malicious user.

- Some members of the blockchain (a.k.a miners) solve complex mathematical problems and compete against each other to validate a block (which has several transactions). This requires computing power and in return the miners get rewarded with some digital currencies (such as bitcoin or ether). Don't worry about the details of the mathematical problems. Just know that this is how new blocks are added to the chain. This is also known as Proof of Work model (PoW).

- PoW protects entire blockchain against DDOS (Denial of service) attacks and makes sure malicious users can't bring the network down by issuing useless transactions too frequently.

If you need to learn more about blockchain concepts check out this guide by CoinDesk[1].

Ethereum Blockchain

Ethereum is an implementation of blockchain concepts and is known as the platform for "Decentralized Apps". Bitcoin blockchain just records transactions between users. But in Ethereum blockchain you can store sophisticated programs called "Smart Contracts" which are then executed when requests come in. These contracts are immutable (can never be altered) and can't be controlled by a single person. That's why the apps that we build on the Ethrereum Blockchain are known as Decentralized Apps or simply Dapps.

The currency of Ethereum Blockchain is called "Ether". Miners are rewarded with Ethers when they successfully validate new blocks.

Smart Contracts

Smart Contracts are pieces of code that are stored in the blockchain and are used to provide trustless transactions. Imagine the following scenarios:

- You want to sell a domain or website to a stranger, but you are also worried that the person may not pay you after you transfer the domain.

- You want to perform an auction, but don't know if the winner will really pay after you transfer the item.

- Imagine high value property transactions where the risk is too high.

In all of the above cases, both the parties don't trust each other and usually involve a third party known as Escrow Agent. Then again you have to trust the agent with your money and pay them a certain fee.

Now imagine the following modified scanarios:

- What if there exists a piece of code which can guarantee that once you receive your payment, the domain will automatically be transferred to the buyer?

- Imagine a code block that can automatically collect and store bids from users and take necessary actions (transfer asset, refund bids and transfer winning bid to seller etc) upon the completion of auction?

[1] https://www.coindesk.com/information/

- A code block that helps both seller and buyer complete a property deal without the involvement of any third-party?

Well this is what Smart Contracts are all about. They are pieces of code that help establish trustless transactions between two parties. Now you may ask how can we trust the contract itself? The answer is:

> Before agreeing to a Smart Contract both the parties can read and verify the entire source code. Further, the nature of Ethereum Blockchain guarantees that the contract will do exactly as it says once it's executed. No one (even the contract creator) can alter it after the said contract is deployed to the blockchain.

Isn't it amazing? Imagine peforming transactions with peace of mind and without the involvement of any kind of intermediaries or agencies that charge hefty fees.

Solidity

So, what language do we use to code these Smart Contracts? Well, the good news is that you have a turing-complete and fully fledged programming language called Solidity at your disposal.

> Turing Complete simply means it's a programming language capable of solving any reasonable computational problem.

In simple terms Solidity is a statically typed language designed to write Smart Contracts. The syntax is inspired by C++ and JavaScript and the compiled output runs on Ethereumm Virtual Machine (EVM). As we move further we'll understand more about Solidity language and how Smart Contracts work in general.

Now that you know the high-level overview of Smart Contracts and Solidity, let's see how we can create an Ether wallet for us and start sending/receiving ETH.

Wallets, Ethers & Wei

Just like Bitcoin is the currency in Bitcoin blockchain, Ether (ETH) is the currency in Ethereum blockchain. In some places you will also come across something called Wei. It's the smallest denomination or the base unit of ETH. Note that `1 ETH = 10^18` Wei.

In order to work with Ethereum Blockchain, you need an Ether wallet. The following are some good options to get your own wallet:

- My Ether Wallet (MEW)[2]: A web based service. It gives you a public address for receiving ETH. It also gives you a private key which you can use to unlock your wallet anytime. Be careful! If you lose your private key, you can't recover your wallet.

- Ethereum Desktop Wallet (Mist)[3]: This is the official wallet app (A.K.A Mist) released by Ethereum team.

- MetaMask[4]: It's an extension that you can install on Chrome, FireFox, Opera and Brave browser. Do note that MetaMask is much more than a wallet. It also lets you interact with Smart Contracts directly from your web browser - but more on that later.

Once you create a wallet you will get a public address and a private key that controls this address. Never reveal your private key or you introduce the risk of losing your ETH. Never store huge amounts of ETH in these wallets. First play around with these tools and familiarize yourself with the ecosystem.

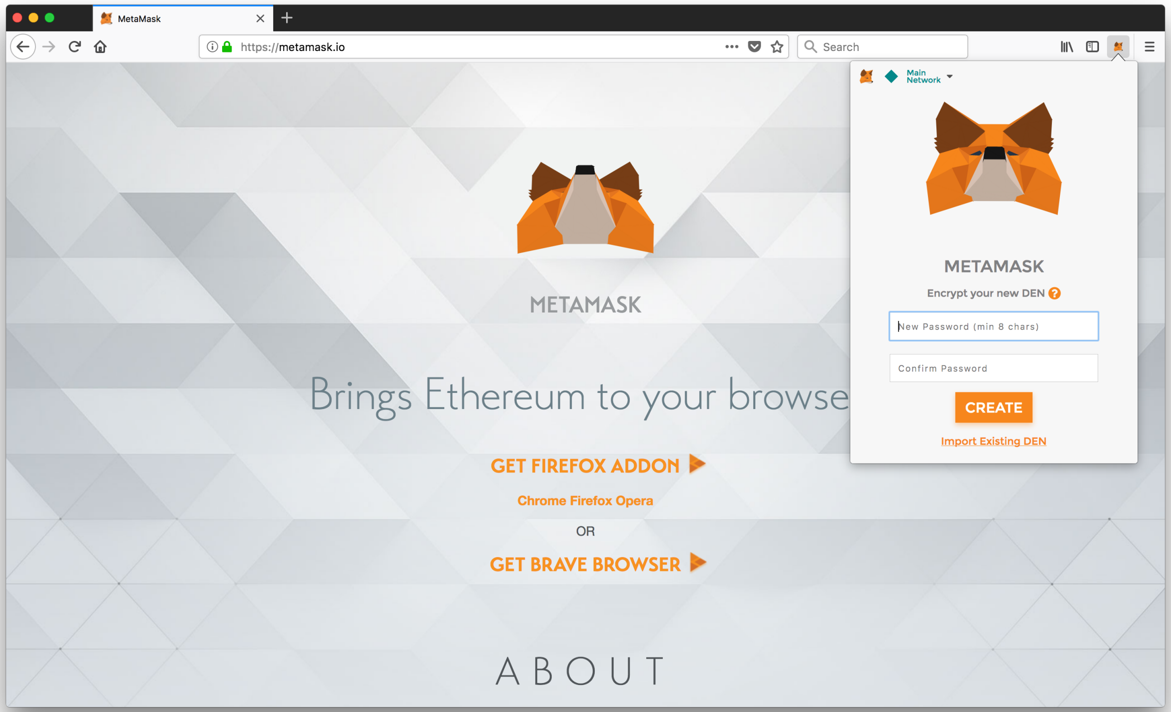
For the sake of simplicity, let's create a wallet on MetaMask and use it throughout the book. Here are the steps to create a MetaMask wallet:

- Open MetaMask website[5] and install the add-on/extension. Currently, it's supported in Chrome, Opera, Brave and Firefox.

- Once MetaMask is installed, click on the icon present at the top right corner of the browser. Accept the terms and conditions and click proceed.

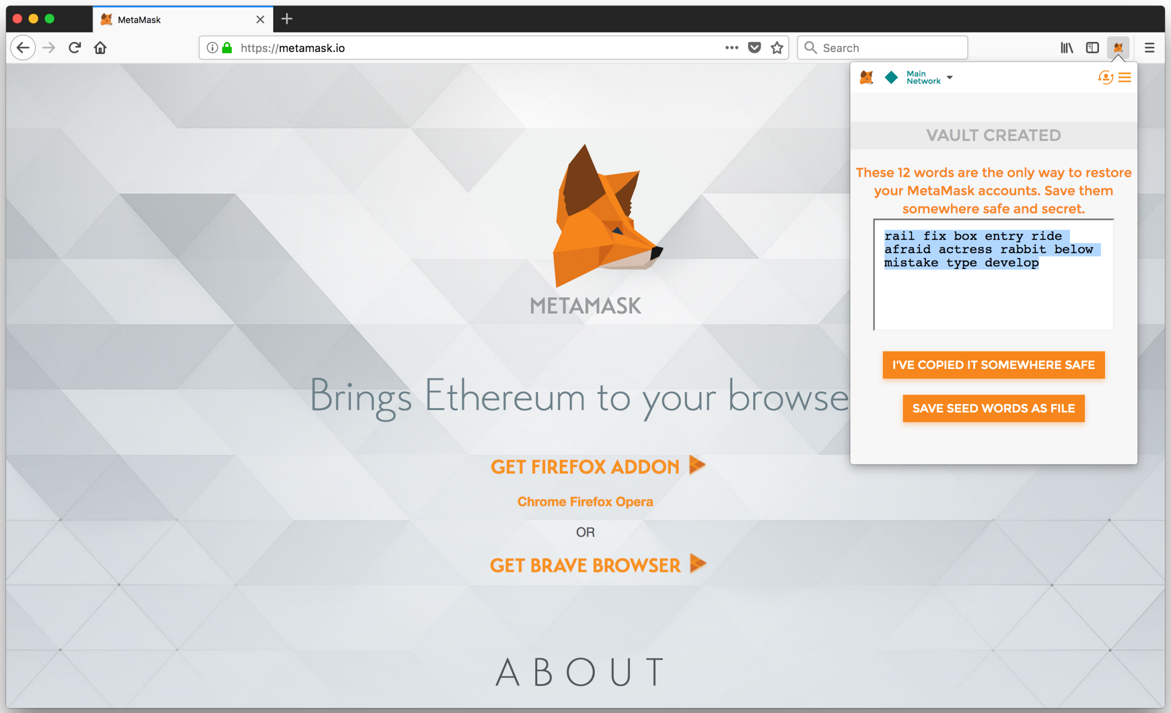
- The next screen will ask you to set a password.

[2]https://www.myetherwallet.com/
[3]https://github.com/ethereum/mist/releases
[4]https://metamask.io/
[5]https://metamask.io/



Set up MM Password

- Once you set the password, the next screen will show you a 12-word seed phrase. Please write it down somewhere and don't lose it. It's the only way to recover your wallets (you are about to create) if something goes wrong. Also, keep it handy as we are going to need this seed phrase in the next chapter.



These 12 words are the only way to restore your MetaMask accounts. Save them somewhere safe and secret.

```
rail fix box entry ride
afraid actress rabbit below
mistake type develop
```

MM seed words

- Viola! Your wallet is now ready. You should see an account named "Account 1". Feel free to edit it and give a meaningful name. You can also create more wallet addresses by clicking on "Create Account" in MetaMask.

Note that these wallets aren't just meant for sending and receiving ETH. They are also needed to deploy and interact with Smart Contracts.

Now that you are familiar with the basics, let's go ahead and write our first Smart Contract.

Chapter 1: Writing a basic Arithmetic Contract

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/smart-contracts.

Tools we'll use

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/smart-contracts.

Project Structure

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/smart-contracts.

Code

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/smart-contracts.

Compile

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/smart-contracts.

Unit Testing

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/smart-contracts.

Final Deployment

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/smart-contracts.

Deployment using online Solidity Compiler

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/smart-contracts.

Next Steps

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/smart-contracts.

Chapter 2: Building a Voting Smart Contract

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/smart-contracts.

Concepts to be covered

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/smart-contracts.

Code

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/smart-contracts.

Poll.sol

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/smart-contracts.

Understanding the concepts

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/smart-contracts.

Data Types in Solidity

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/smart-contracts.

Visibility of Members

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/smart-contracts.

External

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/smart-contracts.

Public

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/smart-contracts.

Internal

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/smart-contracts.

Private

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/smart-contracts.

Function Modifiers, View and Pure Functions

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/smart-contracts.

A note on view and pure functions

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/smart-contracts.

Storage and Memory variables

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/smart-contracts.

Events

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/smart-contracts.

Final Code

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/smart-contracts.

Conclusion

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/smart-contracts.

Chapter 3: Writing an ICO Contract (a.k.a launch your own cryptocurrency)

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/smart-contracts.

What are tokens?

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/smart-contracts.

What is an ICO?

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/smart-contracts.

What's an ERC20 token?

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/smart-contracts.

Writing a basic ICO contract

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/smart-contracts.

Step 1

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/smart-contracts.

Step 2

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/smart-contracts.

Step 3

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/smart-contracts.

Deployment

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/smart-contracts.

Step 1

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/smart-contracts.

Step 2

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/smart-contracts.

Step 3

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/smart-contracts.

Writing a real-world ICO contract

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/smart-contracts.

Project Setup

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/smart-contracts.

Introducing Open Zeppelin

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/smart-contracts.

Code

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/smart-contracts.

Testing & Deployment

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/smart-contracts.

Deploy to Ropsten TestNet

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/smart-contracts.

Testing

This content is not available in the sample book. The book can be purchased on Leanpub at http://leanpub.com/smart-contracts.

Chapter 4: Building a Simple Auction Contract

Chapter 5: Writing a Gift Card Smart Contract

Security Considerations

Best Practices, Dos and Don’ts

Conclusion and what's next?