Implementing
Ethereum Smart Contracts

Benjamin Mawson // ben@benmawson.com
22 December 2022

Table of Contents

Introduction 3
- What are Ethereum Smart Contracts?
- How does Ethereum run smart contracts?
- Benefits and risks of Ethereum Smart Contracts
  - Benefits
  - Risks

Implementation guide 5
- Create and deploy a smart contract on Ethereum
  - Pre-Requisites
  - Connect to Ethereum
  - Select a Test Network
  - Fund Wallet
  - Write First Contract
  - Deploy to Test Network
  - Test and Audit
  - Deploy to Main Network

Further Reading 8

INTRODUCTION

What are Ethereum Smart Contracts?

The term ‘smart contract’ was coined in 1994 by programmer Nick Szabo, who wrote:

> *I call these new contracts “smart” because they are far more functional than their inanimate paper-based ancestors. No use of artificial intelligence is implied. A smart contract is a set of promises specified in digital form, including protocols within which the parties perform on these promises.*

An Ethereum smart contract can simply be defined as a collection of code deployed to the Ethereum platform that defines conditions to which all parties within the contract should agree.

How does Ethereum run smart contracts?

The Ethereum Virtual Machine (EVM) is a software layer on top of the Ethereum blockchain that serves as a runtime environment to execute smart contracts.

Although there are many toolsets used to create, test and deploy smart contracts to the Ethereum platform, this guide will focus on using the Remix IDE which outputs to the Solidity programming language, developed specifically for the creation of Ethereum smart contracts.

Solidity is based on existing programming languages like C++, JavaScript and Python and runs on the EVM.

A smart contract is written in Solidity and then compiled into the EVM to be executed.

In the EVM, 'gas' is a measurement unit used for assigning fees to each smart contract transaction. The greater the computational complexity, the more 'gas' is required.

The cost of a given transaction is proportional to the complexity (the size of the data it contains), calculated by multiplying the current price of 'gas' by the total 'gas' used.

ETHEREUM GAS PRICE EXTENSION FOR CHROME BROWSER

HOW MUCH DOES IT COST TO DEPLOY A SMART CONTRACT ON ETHEREUM?

A contract at a particular address is executed by sending it data that calls a function written into the contract to create a transaction.

A transaction is a cryptographically signed instruction from an account that changes the state of the blockchain.

Block explorers track the details of all transactions in the network. All transactions may be viewed at Etherscan, a site to explore the state of the chain:

ETHERSCAN.IO

Smart contracts specify the rules of decentralized finance engagement and the EVM executes algorithms to enact them.

For each interaction with a decentralized exchange like Uniswap or a DeFi lending protocol like Aave, EVM executes scripts (smart contracts) based on user inputs.

Benefits and risks of Ethereum Smart Contracts

Benefits

Unlike Bitcoin's lack of scripting flexibility, Ethereum's Solidity makes smart contracts highly programmable which more easily facilitates mainstream adoption.

Being decentralized digital agreements between two or more parties there is no requirement for intermediaries.

No single party is in control of the contract and once deployed to the blockchain, it cannot be modified.

Decentralized finance protocols are valued for their 'composability' meaning that smart contracts can work together to create a new service. One protocol can be used to control another, permitting the creation of powerful automated toolsets.

Risks

While providing security, speed and transparency for transactions, the visibility to all of the contracts themselves makes their vulnerabilities exploitable.

These could include, for example, bugs that trigger unintended tasks which result in losses for investors.

The Genesis DAO cyberattack resulted from exploitation of a bug where fund requests could be made multiple times before the contract had registered it and updated the balance, leading to the loss of $millions in Ether.

While blockchains are effectively tamper-proof (if a single data block in a distributed database is maliciously altered, it is rejected as false and dismissed by the network as a whole), this is only true of data stored as completed transaction records and is not applicable to active smart contracts.

Flawed smart contracts therefore are vulnerable to potential exploits if poorly written or incompletely audited.

As smart contracts cannot be modified, to amend or repair faulty functionality an additional contract is required to interact with the first in order to fix problems that were overlooked during the build.

Smart contracts are not yet subject to legal governance and legal liability has often been found to be indeterminate, with no single party holding responsibility for losses.

The key to safeguarding against malicious activity is to run comprehensive code audits and automated security scans to identify bugs before deploying the contract to the mainnet.

Smart contract use cases are numerous but the current state of immutability carries both risk and remedial cost not associated with traditional code development.

Validation systems are needed to remove potential loopholes creating vulnerabilities. Legal frameworks to provide governance and security are urgently required to protect investors from the results of attacks.

IMPLEMENTATION GUIDE

Create and deploy a smart contract on Ethereum

Pre-Requisites

1. Download MetaMask browser extension for Chrome and enable a MetaMask wallet.

HTTPS://METAMASK.IO/

Set up a password and securely store the auto-generated 'secret backup phrase' (essential for data recovery).

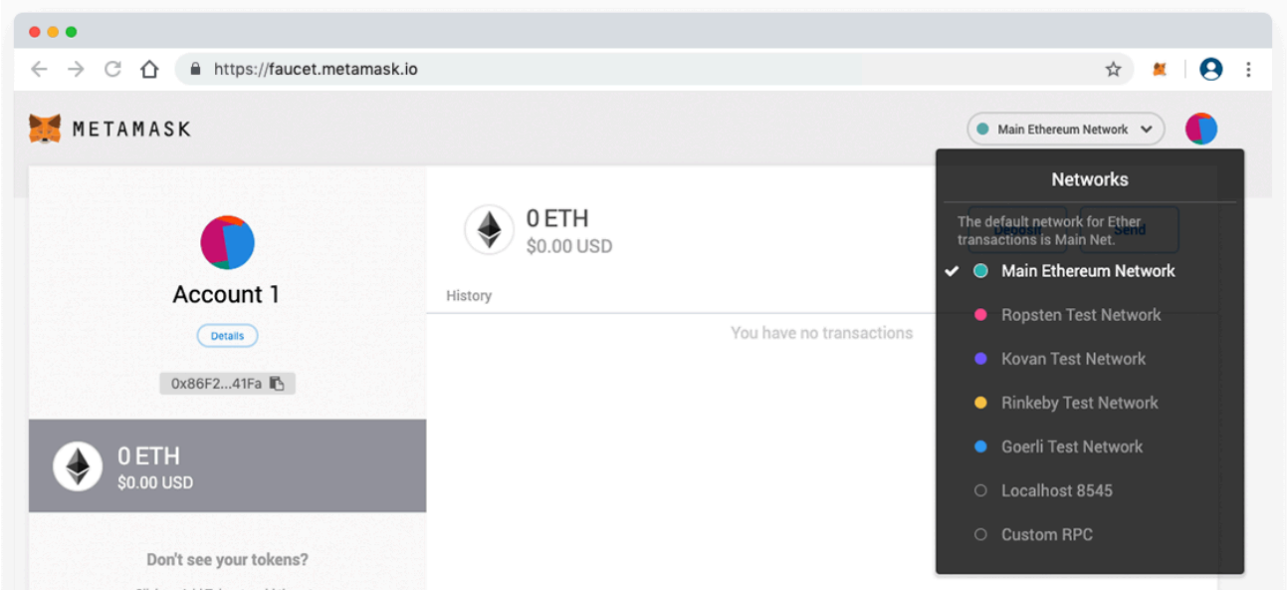
Connect to Ethereum

2. Connect the MetaMask wallet to the Main Ethereum Network or 'Ethereum mainnet' The dropdown on screen top-right confirms connection to the Main Ethereum Network.

Select a Test Network

3. Choose a test network.

Test networks are listed below the Main Ethereum Network. Select one to use for testing the smart contract you are building.



Fund Wallet

4. Fund wallet with dummy Ethers.

To test the smart contract, add dummy ethers to MetaMask wallet.

To test a contract using e.g. the Robsten test network, select it and you will find 0 ETH as the initial balance in your account.

Click on the "Deposit" and "Get Ether" buttons under the Test Faucet and follow the instructions shown.

Once the dummy ethers are added to the wallet, you can start writing smart contracts on the Remix Browser IDE in the Solidity programming language.

Write First Contract

5. Download the Remix browser to write a smart contract in Solidity.

HTTPS://REMIX-PROJECT.ORG/

Remix is appropriate for writing small contracts and its features include:

- Warnings [gas cost, unsafe code, overlapping variable names, whether functions can be constant]
- Syntax and error highlighting
- Functions with injected Web3 objects
- Static analysis
- Integrated debugger
- Integrated testing and deployment environment
- Deploy directly to Mist or MetaMask

Activate two plugins using the 'plug' icon in the left menu:

- Deploy and run transactions
- Solidity compiler

6. Create a .sol extension file.

In Remix, click on the **+** icon to create a .sol extension to make a programmed file solidity-compatible.

Full instructions on creating smart contracts:

HOW TO WRITE AN ETHEREUM SMART CONTRACT USING SOLIDITY

7. Complete your smart contract code.

Choose a version of the compiler from the Remix browser and compile the Solidity smart contract code.

REMIX IDE REMIX DOCUMENTATION

Deploy to Test Network

8. Deploy the smart contract to test network.

Click 'deploy' in Remix. Wait until the transaction is complete. On completion, the smart contract address appears on the right of the Remix window.

To check wallet tokens, in MetaMask select 'add tokens', enter the smart contract address and click 'ok'. The number of available tokens is shown.

Test and Audit

9. Prepare smart contract to go live.

Test and audit the smart contract prior to go live: once the smart contract is ready, deploy to Ethereum mainnet.

- Run each method [e.g. transfer, total supply, balance] in Remix
- Transfer some tokens to other ethereum wallet addresses
- Check the balance of that address by calling the balance method
- Get total supply by running the total supply method

Deploy to Main Network

10. Deploy the smart contract to Main Ethereum Network

Switch to Main Ethereum Network at MetaMask and add real ethers.

Again deploy smart contract using Remix as above steps.

After successful smart contract deployment, search

ETHERSCAN.IO

for the smart contract address.

- Click “verify the contract”
- Copy smart contract code and paste it at Etherscan
- Select the same compiler version as previously selected in Remix to compile the code
- Check “optimization” to Yes, if optimization was selected in Remix; otherwise, select No
- Click Verify
- Wait until the verification is complete
- Run required smart contract methods at Etherscan

FURTHER READING

ETHEREUM LIVE MARKET UPDATES ON COINBASE

SMART CONTRACT MONITORING

BLOCKCHAIN RISK MANAGEMENT

Three brief Smart Contract Guides

HOW TO CREATE A SMART CONTRACT ON ETHEREUM

STEPS TO CREATE, TEST AND DEPLOY ETHEREUM SMART CONTRACTS

HOW TO WRITE AN ETHEREUM SMART CONTRACT USING SOLIDITY