pana
cloud

Programmable Money for the API Economy

Bootcamp 2021

https://panacloud.github.io/bootcamp-2021/

What is an Programmable Money?

- Programmable money is money represented in digital form, also known as tokens.
- This digital currency is tracked with corresponding electronic ledgers, most popularly blockchains, enabling a transactional record that is publicly and securely shared.
- This ledger should preferably be based on open source software to promote sound governance, and to keep the programming interfaces equally available to banks and other participants.

It is Not Automated Payments

- Programmable money is not the ability to write arbitrary code that can move money using the Banking API. Even if it includes complex business logic and external data as part of the decision making.
- Nor is it if I instruct my bank to make an automated monthly payment of $4,000 to my landlord.
- In each of the above cases, of course the bank could actually hold back the payment, even after they have received the payment instruction.

Then what is it?

- One main characteristic of programmable money mean that no matter what, the code’s instructions will be carried out, and no bank or intermediary can stop it.
- Stable Coin like DAI which are controlled by Smart Contract on Ethereum achieves this.
- Stable Coin like Diem on Permissioned blockchain do not achieve this.

Designer Money

- Now you can create money where the money itself has control logic built into it. This is done at the smart contract level.
- A smart contract is typically a bunch of code that is run by all participants in a blockchain network.
- It defines:

1. The characteristics of the money (how many units there are, who initially owns it, etc)

2. How users can interact with the money (ask for a balance, make a payment, etc). The constraints are coded here.

DAI the Ideal Currency for the API Economy

- Dai (or DAI) is a stablecoin cryptocurrency which aims to keep its value as close to one United States dollar (USD) as possible through an automated system of smart contracts on the Ethereum blockchain.
- Dai is already powering the digital economy where people can bypass banks and other financial firms to transact directly with one another.
- Dia can also power the API Economy where it will allow the developers to directly transact with one another.
- Dia is programmable money.

DAI Details

- The protocol allows lenders to lock up their funds in ETH or any other crypto asset through smart contracts on Ethereum, guaranteeing the issuance of new DAI tokens in the form of loans.
- The required deposit is greater than the value of the token.
- In contrast to other crypto assets, it minimizes price volatility and traded on Crypto Exchanges.
- Anyone can send and receive DAI without the need for third parties, in a peer-to-peer (P2P) method.
- It is decentralized, resistant to censorship, and trustworthy.

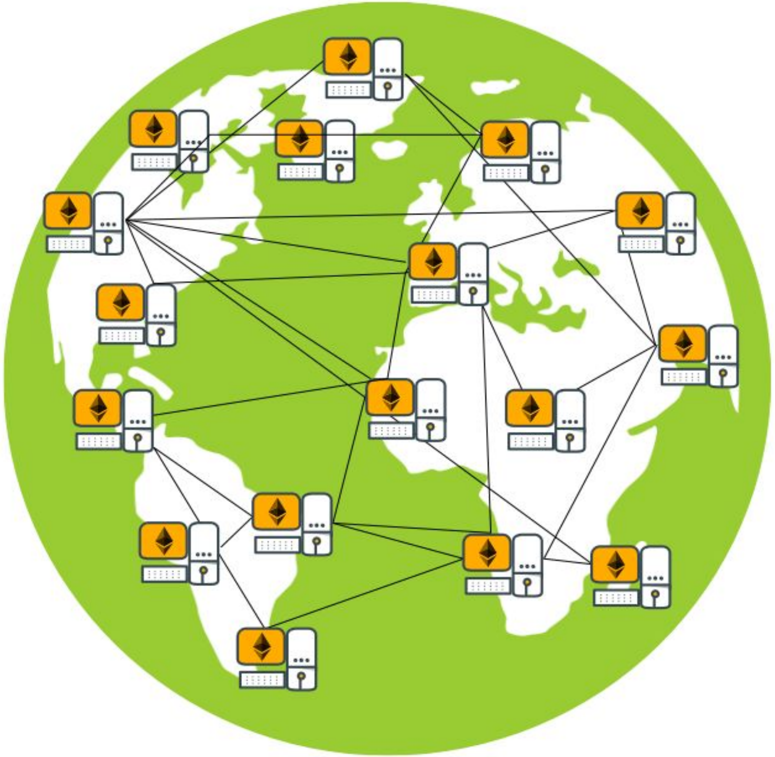
DAI Benefits for API Economy

- Stable: Has value that is designed to be stable over any period
- Transparent: Auditable Payments to API Developers
- Borderless: Allows cross-border payments globally
- Faster Speed: Instant settlement
- Lower Fees: after Ethereum 2 it will be even lower, less than credit card payment of 2%
- Programmable: Feathers can be added on top of Dai which are useful for the API Economy

DAI Benefits for API Economy

- Stable: Has value that is designed to be stable over any period
- Transparent: Auditable Payments to API Developers
- Borderless: Allows cross-border payments globally
- Faster Speed: Instant settlement
- Lower Fees: after Ethereum 2 it will be even lower, less than credit card payment of 2%
- Programmable: Feathers can be added on top of Dai which are useful for the API Economy

The World Computer



Ethereum is a decentralized platform that runs smart contracts applications that run exactly as programmed without any possibility of downtime, censorship, fraud or third party interference

The Birth of Ethereum

1. **A blockchain without a specific purpose, that could support a broad variety of applications by being *programmed*.**
2. **The idea was that, a developer could program their particular application without having to implement the underlying mechanisms of**
   a. **peer-to-peer networks,**
   b. **blockchains,**
   c. **consensus algorithms, etc.**
3. **The Ethereum platform was designed to abstract these details and provide a deterministic and secure programming environment for decentralized blockchain applications.**

What Is Ethereum?

From a computer practical perspective:

Ethereum is an open source, globally decentralized computing infrastructure that executes programs called smart contracts. It uses a blockchain to synchronize and store the system’s state changes, along with a cryptocurrency called ether to meter and constrain execution resource costs.

How does one get gas

1. You won’t find gas on any exchanges.
2. Only be purchased as part of a transaction, and can only be bought with ether.
3. Ether needs to be sent along with a transaction and it needs to be explicitly earmarked for the purchase of gas, along with an acceptable gas price.
4. Just like at the pump, the price of gas is not fixed.
5. Gas is purchased for the transaction, the computation is executed, and any unused gas is refunded back to the sender of the transaction.

Decentralized Applications (DApps)

DApp is a web application that is built on top of open, decentralized, peer-to-peer infrastructure services.

A DApp is composed of at least:

- Smart contracts on a blockchain
- A web frontend user interface

In addition, many DApps include other decentralized components, such as:

- A decentralized (P2P) storage protocol and platform
- A decentralized (P2P) messaging protocol and platform

Choosing an Ethereum Wallet

1. "Wallet" means a software application that manage your Ethereum account.
2. It holds your keys & can create and broadcast transactions on your behalf.
3. If you choose a wallet and don’t like you can change wallets quite easily.
4. Make a transaction to sends your funds from the old wallet to the new
5. Export your private keys and import them into the new one.

Types of Wallet

1. Mobile wallet,
2. Desktop wallet, and
3. Web-based wallet.

Generally the more popular a wallet application is, the more trustworthy it is likely to be.

It is good practice to avoid "putting all your eggs in one basket" and have your Ethereum accounts spread across a couple of wallets.

A good starter wallet

MetaMask

1. Browser extension wallet that runs in your browser.
2. It is easy to use and convenient for testing,
3. as it is able to connect to a variety of Ethereum nodes and test blockchains.
4. MetaMask is a web-based wallet.

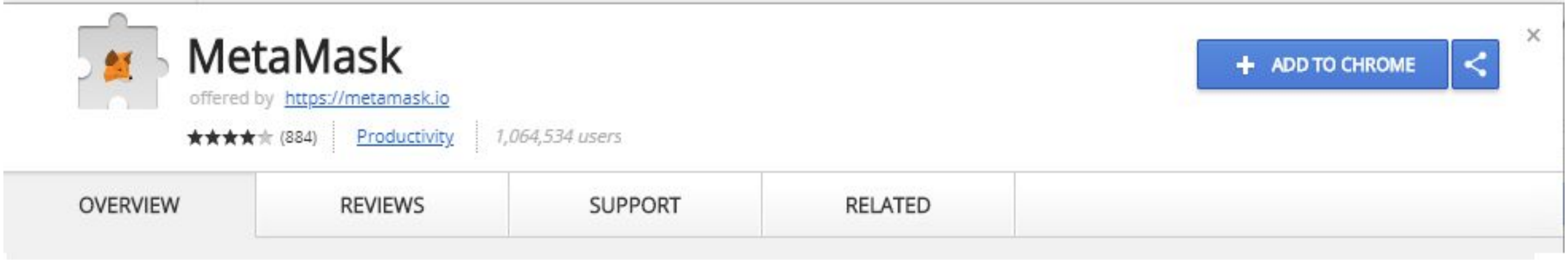
Control and Responsibility

1. **Open Blockchain & operate as Decentralize System**
2. **Each user can control their own private keys**
3. **That mean: control access to funds and smart contracts.**
4. **We sometimes call the combination of access to funds and smart contracts an "account" or "wallet."**

With control comes a big responsibility

1. **If you lose your private keys, you lose access to your funds and contracts.**
2. **No one can help you regain access—your funds will be locked forever.**
3. **Some users choose to give up control over their private keys by using a third-party custodian, such as an online exchange.**

Getting Started with MetaMask



MetaMask

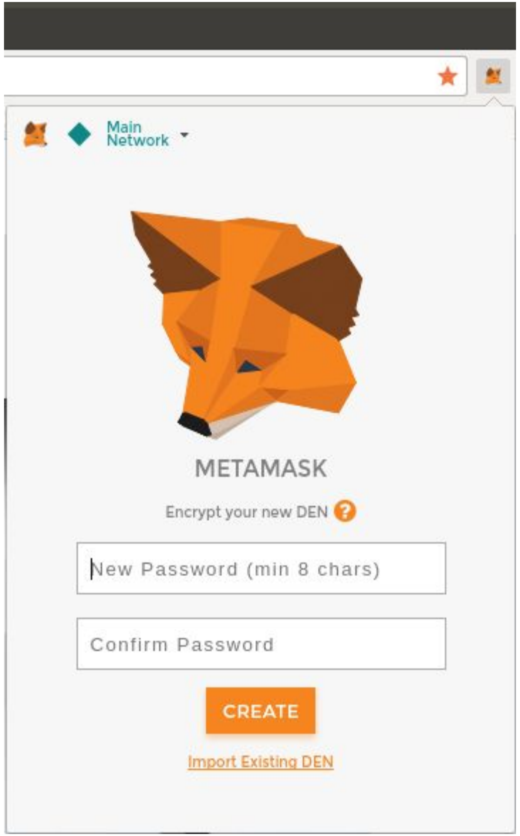
offered by https://metamask.io

★★★★☆ (884) | Productivity | 1,064,534 users

https://metamask.io

Getting Started with MetaMask

1. **Once MetaMask is installed you should see a new icon in your browser's toolbar.**
2. **Click on it to get started.**
3. **You will be asked to accept the terms and conditions and then to create your new Ethereum wallet by entering a password**



Switching Networks

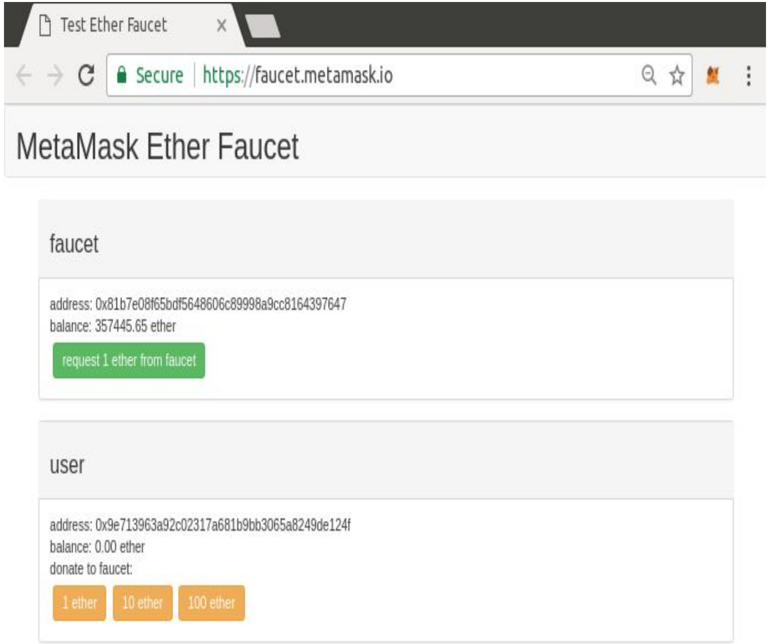
1. **Main Ethereum Network**
   a. By default, MetaMask will try to connect to the main public network.
   b. Real ETH, real value, and real consequences.
2. **Ropsten Test Network**
   a. Ethereum public test blockchain and network.
   b. ETH on this network has no value.
3. **Kovan Test Network**
   a. Ethereum public test blockchain and network using consensus protocol - Proof of authority ETH on this network has no value. The Kovan test network is supported by Parity only.

Switching Networks

4. **Rinkeby Test Network**
    a. Ethereum public test blockchain and network, using consensus protocol - proof of authority
    b. ETH on this network has no value.
5. **Localhost 8545**
    a. Connects to a node running on the same computer as the browser.
    b. The node can be part of any public blockchain (main or testnet), or a private testnet.
6. **Custom RPC**
    a. Allows you to connect MetaMask to any node with a Geth-compatible Remote Procedure Call (RPC) interface. The node can be part of any public or private blockchain

Getting Some Test Ether

1. New transaction will be mined and your MetaMask wallet will show a balance of 1 ETH.
2. Click on the transaction ID it take you to a *block explorer*. MetaMask uses the [Etherscan block explorer](),
3. The transaction containing the payment from the Ropsten Test Faucet is shown in [Etherscan Ropsten block explorer]().



MetaMask Ether Faucet

faucet

address: 0x81b7e08f65bdf5648606c89998a9cc8164397647
balance: 357445.65 ether

user

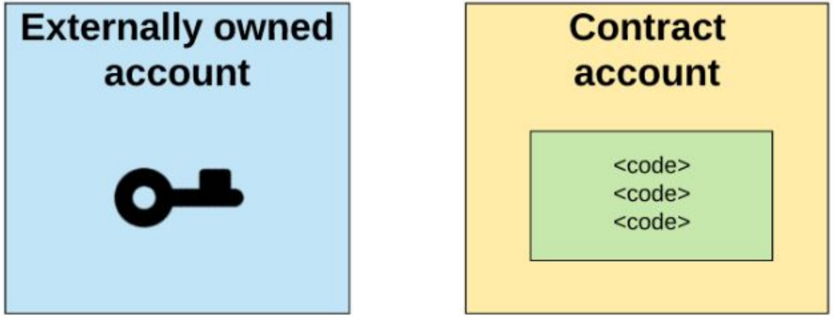
address: 0x9e713963a92c02317a681b9bb3065a8249de124f
balance: 0.00 ether
donate to faucet:

Addresses

One of Ethereum’s foundational technologies is *cryptography*

Two types of Accounts

1. Externally owned Accounts (EoA)
2. Contract Account

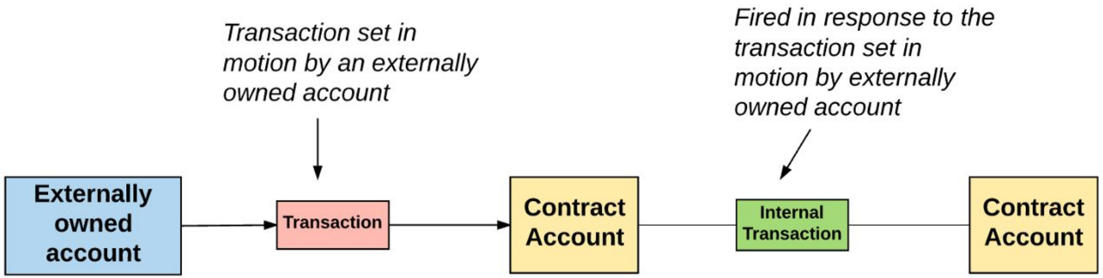


Ownership of ether by EOAs is established through

1. Digital private keys,
2. Ethereum addresses, and
3. Digital signatures

Addresses

- **An externally owned account can send messages to other externally owned accounts OR to other contract accounts by creating and signing a transaction using its private key.**



Transaction set in motion by an externally owned account
Fired in response to the transaction set in motion by externally owned account
Externally owned account
Transaction
Contract Account
Internal Transaction
Contract Account

- **Unlike externally owned accounts, contract accounts can't initiate new transactions on their own**

Keys

Private keys are not used directly in the Ethereum system.

- **Account addresses are derived directly from private keys.**
- **Private key uniquely determines a single Ethereum address, also known as an account.**

Access and control of funds is achieved with digital signature

- **Ethereum transactions require a valid digital signature to be included in the blockchain.**
- **Anyone with a copy of a private key has control of the corresponding account and any ether it holds.**
- **Contracts account are not backed by public-private key paris**

Public Key Cryptography and Cryptocurrency

In Ethereum, we use public key cryptography (also known as asymmetric cryptography) to create the public–private key pair.

Together, they represent an Ethereum account by providing, respectively, a publicly accessible account handle (the address) and private control over access to any ether in the account and over any authentication the account needs when using smart contracts.

The private key controls access by being the unique piece of information needed to create *digital signatures*, which are required to sign transactions to spend any funds in the account. Digital signatures are also used to authenticate owners or users of contracts

Ethereum transaction is basically a request to access a particular account with a particular Ethereum address

Wallets

- A wallet is a software application that serves as the primary user interface to Ethereum.
- The wallet controls access to a user’s money, managing keys and addresses, tracking the balance, and creating and signing transactions.
- Ethereum wallets can also interact with contracts, such as ERC20 tokens.
- Interfaces to Ethereum-based decentralized applications, or *DApps*

ERC20

What are Tokens

Commonly used to refer to **privately issued special-purpose coin**

- like items of **insignificant intrinsic value,**
- i. e. Transportation, food courts, and game tokens.

Nowadays, "tokens" administered on blockchains are **redefining the word** to mean **blockchain-based abstractions** that can be owned and that represent

- Assets,
- Currency, or
- Access rights.

Traditional Vs Blockchain Tokens

Traditional Tokens

Restricted to specific businesses, organizations, or locations,

Physical tokens are **not easily exchangeable** and **typically have only one function**.

Blockchain Tokens

these restrictions are lifted

Many **blockchain tokens serve multiple purposes globally** &

Can be traded for each other / for other currencies

How are Token Used

Most obvious use of tokens is as digital private currencies

This is only one possible use, **Currency is just the first application:**

Tokens can be programmed to serve many different functions

What can be those difference Functions?

Token Standards - ERC20

First standard was introduced in November 2015 by Fabian Vogelsteller as an Ethereum Request for Comments (ERC). It was automatically assigned GitHub issue number 20, giving rise to the name "ERC20 token." The vast majority of tokens are currently based on the ERC20 standard.

ERC20 is a standard for fungible tokens, meaning that different units of an ERC20 token are interchangeable and have no unique properties.

The ERC20 standard defines a common interface for contracts implementing a token, such that any compatible token can be accessed and used in the same way.

Token Standards - ERC20

Here's what an ERC20 interface specification looks like in Solidity:

```
contract ERC20 {
   function totalSupply() constant returns (uint theTotalSupply);
   function balanceOf(address _owner) constant returns (uint balance);
   function transfer(address _to, uint _value) returns (bool success);
   function transferFrom(address _from, address _to, uint _value) returns
       (bool success);
   function approve(address _spender, uint _value) returns (bool success);
   function allowance(address _owner, address _spender) constant returns
       (uint remaining);
   event Transfer(address indexed _from, address indexed _to, uint _value);
   event Approval(address indexed _owner, address indexed _spender, uint _value);
}
```

Consists of a number of functions that must be present in every implementation of the standard, as well as some optional functions and attributes that may be added by developers.

ERC20 required functions and events

An ERC20-compliant token contract must provide at least the following functions and events:

totalSupply - Returns the total units of this token that currently exist. ERC20 tokens can have a fixed or a variable supply.

balanceOf - Given an address, returns the token balance of that address.

transfer - Given an address and amount, transfers that amount of tokens to that address, from the balance of the address that executed the transfer.

transferFrom - Given a sender, recipient, and amount, transfers tokens from one account to another. Used in combination with approve.

ERC20 required functions and events

approve - Given a recipient address and amount, authorizes that address to execute several transfers up to that amount, from the account that issued the approval.

allowance - Given an owner address and a spender address, returns the remaining amount that the spender is approved to withdraw from the owner.

Transfer - Event triggered upon a successful transfer (call to transfer or transferFrom) (even for zero-value transfers).

Approval - Event logged upon a successful call to approve.

Smart Contract

When smart people hear the term “smart contracts”, their imaginations tend to run wild

A smart contract is just a fancy name for code that runs on a blockchain, and interacts with that blockchain’s state.

A smart contract is a piece of code that is stored on an blockchain, triggered by blockchain transactions and which reads and writes data in that blockchain’s database.

Each Language has its strengths and weaknesses – you’d be crazy to build a website in C or compress HD video in Ruby.

But in principle at least, you could if you wanted to. You’d just pay a heavy price in terms of convenience, performance, and quite probably, your hair.

DAI Lab

1. **Install Metamask (Ethereum Wallet)**

   https://metamask.io/

2. **Select Test network for development in Metamask, we will be using Ropsten**

3. **Generate Ethereum on a Facet and send it to your Metamask account address:**

   https://faucet.ropsten.be/

   https://faucet.dimensions.network/

   https://faucet.metamask.io/

DAI Lab

4. Generate Dai from a Dai Faucet and send it to your Metamask Account Address:

https://app.compound.finance/

Watch this video to learn:

https://youtu.be/5GLh1_zAftk

5. To show received Dai in Matamask push Add Token button using the following DAI contract address:

0xad6d458402f60fd3bd25163575031acdce07538d

DAI Lab

6. Send one Dai or more to my address:

0xA31Fa5D96441645FCeD6F09743cA5703605817Ca

Thank You

pana
cloud

Platform for the Serverless API Economy

Fusing Serverless, AI, IoT, Blockchain, and Quantum Technologies
in Next-Gen APIs