



Tracking Payment Flows in Ethereum

Michael Fröwis

Overview

Content:

This talk explains the differences of money-flow tracking between Bitcoin and Ethereum.

We outline challenges for forensic investigations in Ethereum.

What is Ethereum ?



- Second most relevant blockchain system by market valuation

What is Ethereum ?



- Second most relevant blockchain system by market valuation
- Built-in currency called Ether (ETH)

What is Ethereum ?



- Second most relevant blockchain system by market valuation
- Built-in currency called Ether (ETH)
- Online since July 30th, 2015

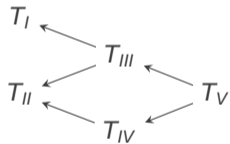
What is Ethereum ?



- Second most relevant blockchain system by market valuation
- Built-in currency called Ether (ETH)
- Online since July 30th, 2015
- \approx 14 second block time

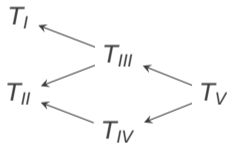
Tracking Payment Flows: From Bitcoin to Ethereum

Transaction graph

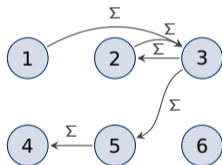


Tracking Payment Flows: From Bitcoin to Ethereum

Transaction graph

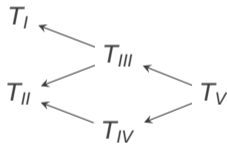


Address graph

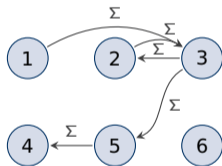


Tracking Payment Flows: From Bitcoin to Ethereum

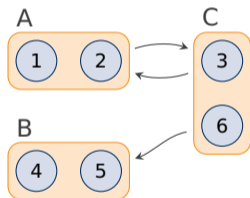
Transaction graph



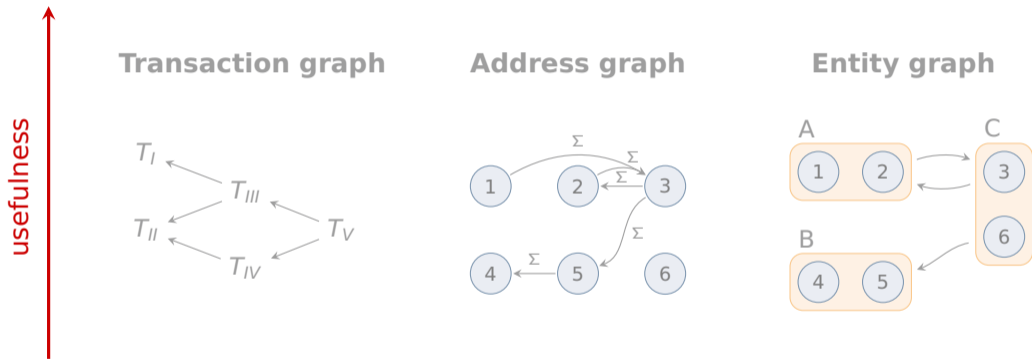
Address graph



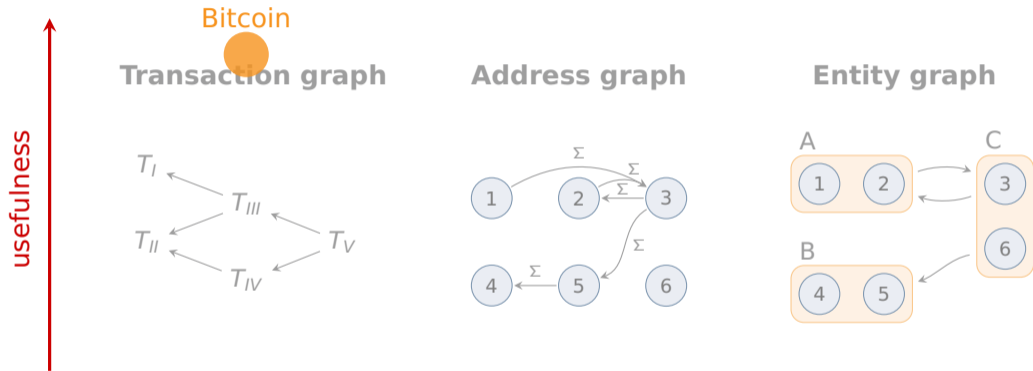
Entity graph



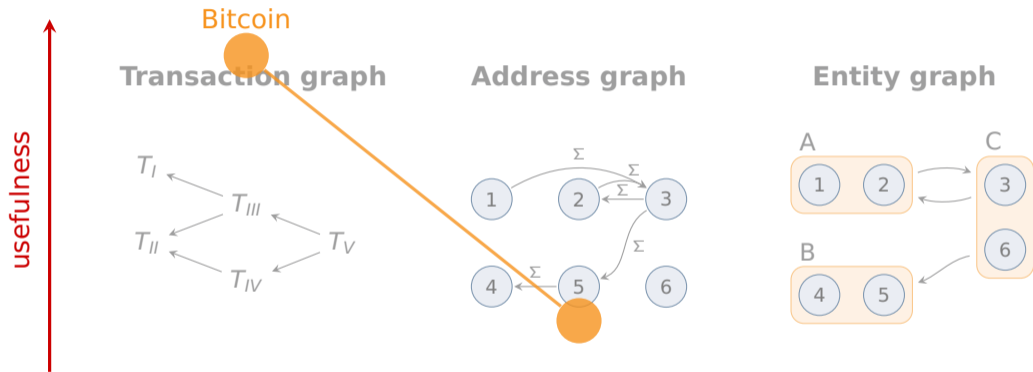
Tracking Payment Flows: From Bitcoin to Ethereum



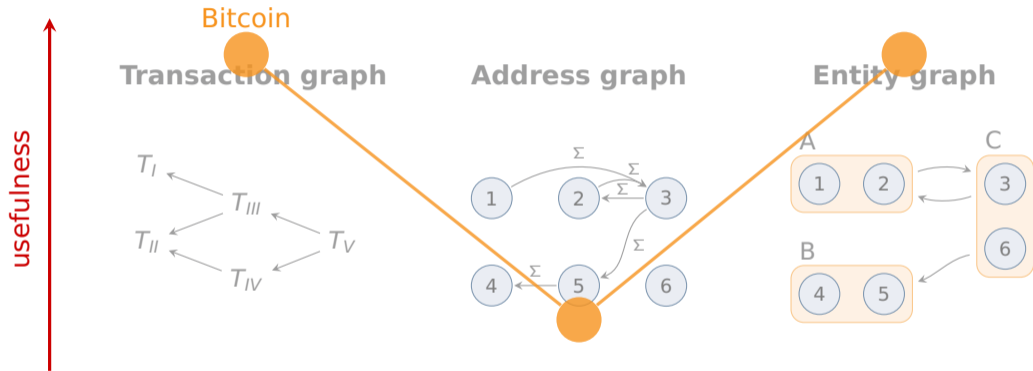
Tracking Payment Flows: From Bitcoin to Ethereum



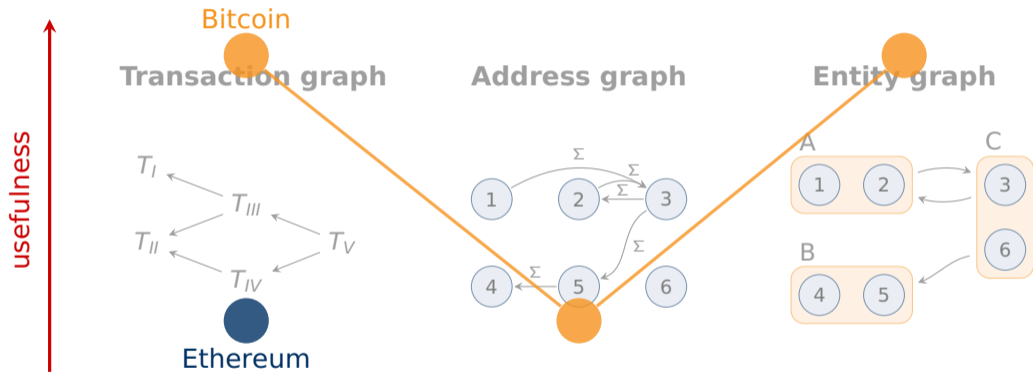
Tracking Payment Flows: From Bitcoin to Ethereum



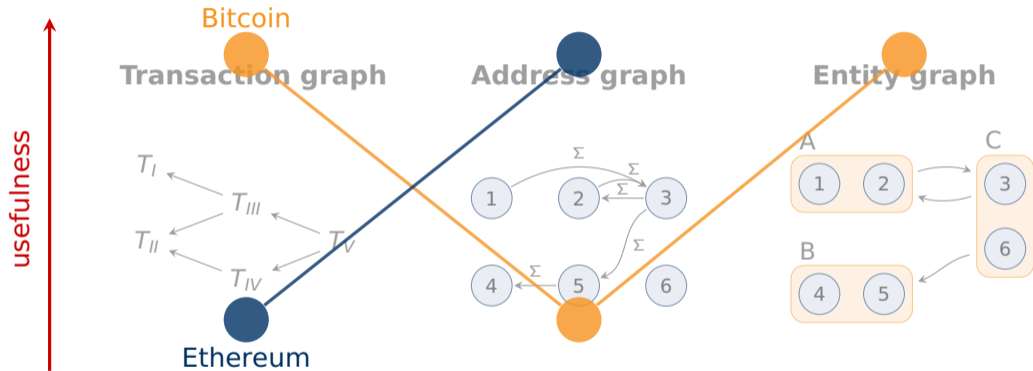
Tracking Payment Flows: From Bitcoin to Ethereum



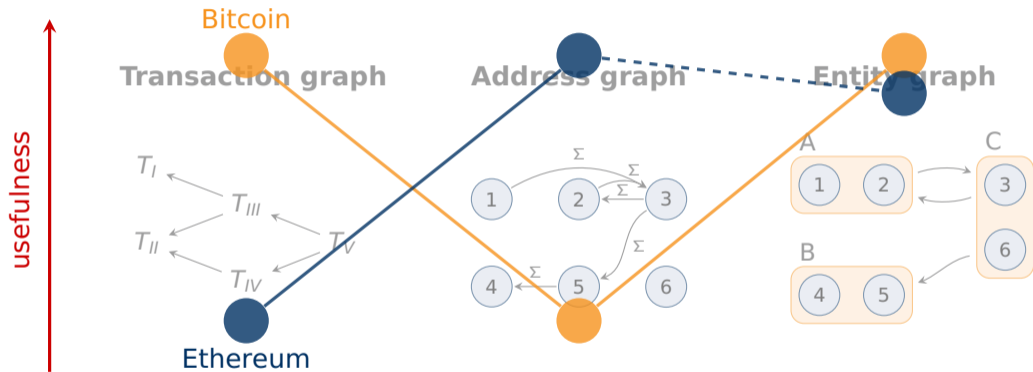
Tracking Payment Flows: From Bitcoin to Ethereum



Tracking Payment Flows: From Bitcoin to Ethereum



Tracking Payment Flows: From Bitcoin to Ethereum



Ethereum

Account model: Analogy std. bank transfer, accounts have balances

Externally Owned Account (EOA):

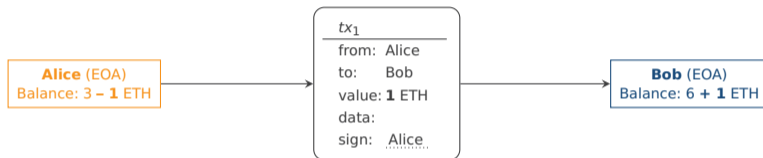
Alice (EOA)
Balance: 3 ETH

Bob (EOA)
Balance: 6 ETH

Ethereum

Account model: Analogy std. bank transfer, accounts have balances

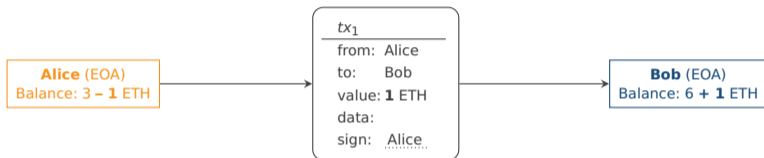
Externally Owned Account (EOA):



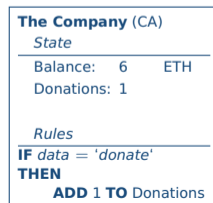
Ethereum

Account model: Analogy std. bank transfer, accounts have balances

Externally Owned Account (EOA):



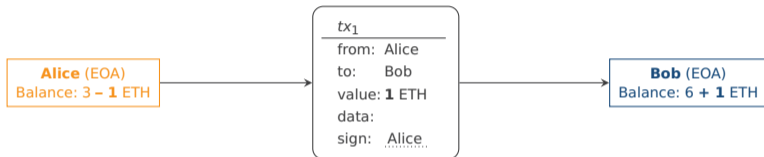
Code Account (CA) aka „Smart Contract“:



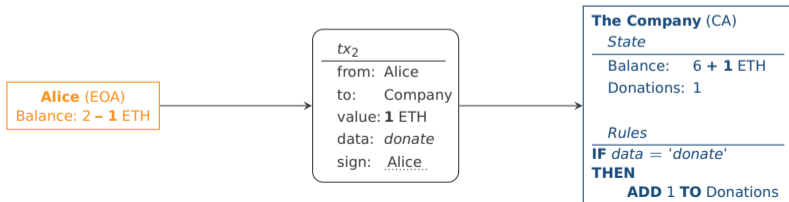
Ethereum

Account model: Analogy std. bank transfer, accounts have balances

Externally Owned Account (EOA):



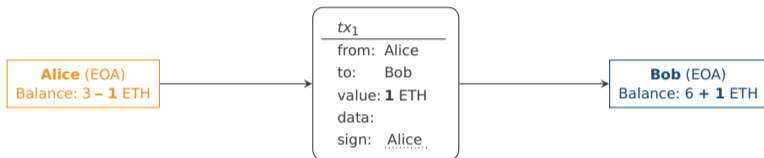
Code Account (CA) aka „Smart Contract“:



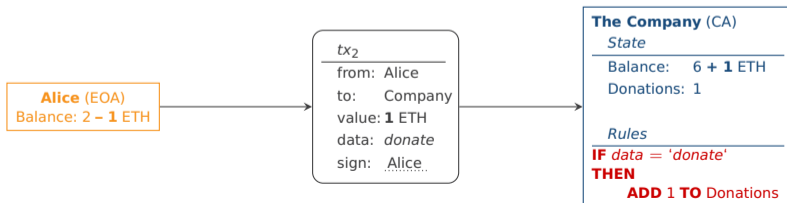
Ethereum

Account model: Analogy std. bank transfer, accounts have balances

Externally Owned Account (EOA):



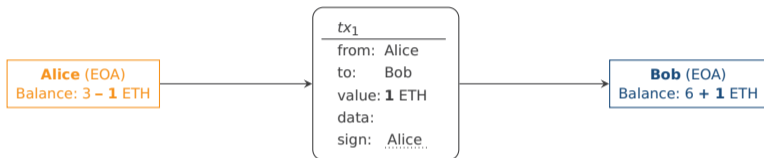
Code Account (CA) aka „Smart Contract“:



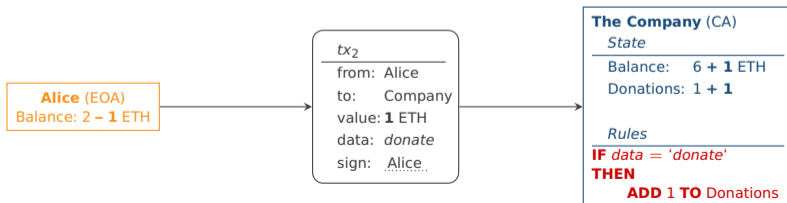
Ethereum

Account model: Analogy std. bank transfer, accounts have balances

Externally Owned Account (EOA):



Code Account (CA) aka „Smart Contract“:



What Can Code Accounts Do ?

- Activated on receipt of a transaction

What Can Code Accounts Do ?

- Activated on receipt of a transaction
- Store and modify local state

What Can Code Accounts Do ?

- Activated on receipt of a transaction
- Store and modify local state
- Arbitrary computations

What Can Code Accounts Do ?

- Activated on receipt of a transaction
- Store and modify local state
- Arbitrary computations
- Create transactions: communicate, transfer Ether

What Can Code Accounts Do ?

- Activated on receipt of a transaction
- Store and modify local state
- Arbitrary computations
- Create transactions: communicate, transfer Ether

Why is this useful?

Gambling services, decentralized exchanges, prediction markets, wallets, state channels and payment channels, . . .

But most prominently token systems.

What Is a Token System?

Token: Jargon for exchangeable virtual asset, fungible or non-fungible.

What Is a Token System?

Token: Jargon for exchangeable virtual asset, fungible or non-fungible.

What Is a Token System?

Token: Jargon for exchangeable virtual asset, fungible or non-fungible.

Token system: CA keeping track of token ownership and transfers.

What Is a Token System?

Token: Jargon for exchangeable virtual asset, fungible or non-fungible.

Token system: CA keeping track of token ownership and transfers.

Use-cases: Sub-VCs, crowdfunding, shares, votes, reward systems, ...

How Does a Token System Work ?

Alice (EOA)
Balance: 3 ETH

My Token (CA)

State

Balance: 0 ETH

Alice: 2

Bob: 0

Rules

IF *data = 'send';to;val*

AND *from \geq val*

THEN

SUB *val* **FROM** *from*

ADD *val* **TO** *to*

How Does a Token System Work ?

Alice (EOA)
Balance: 3 ETH

My Token (CA)

State

Balance: 0 ETH

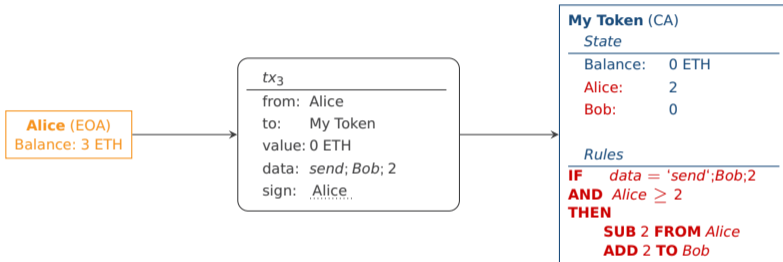
Alice: 2

Bob: 0

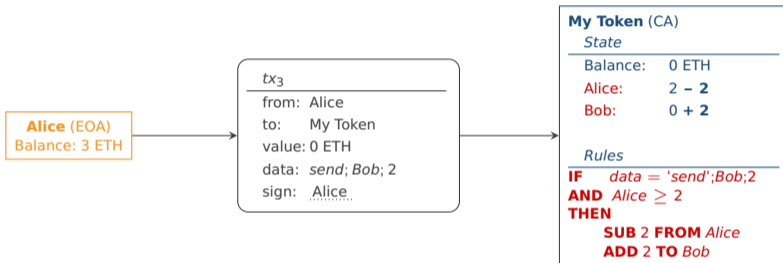
Rules

IF *data = 'send';to;val*
AND *from ≥ val*
THEN
 SUB *val* **FROM** *from*
 ADD *val* **TO** *to*

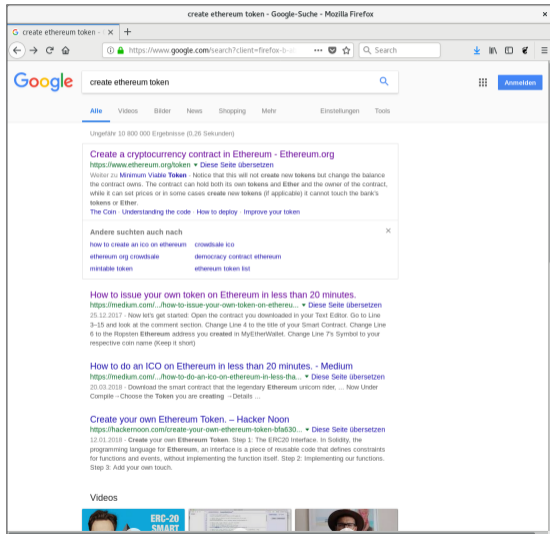
How Does a Token System Work ?



How Does a Token System Work ?



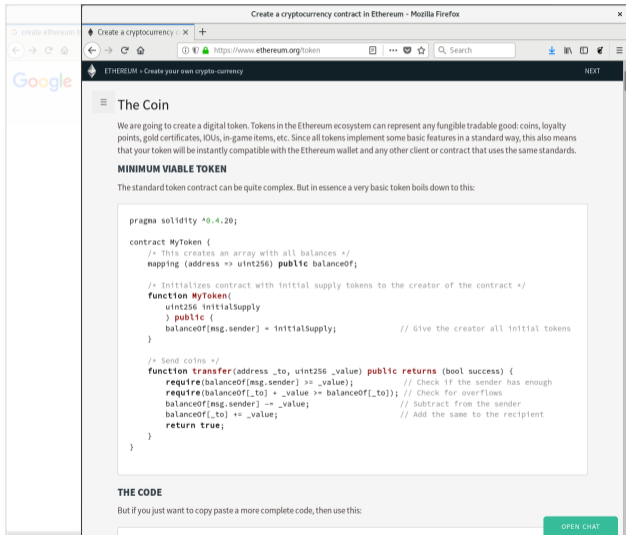
What Does It Take to Create a Token System ?



The screenshot shows a Mozilla Firefox browser window with the search query "create ethereum token". The search results are as follows:

- Search Results:**
 - Create a cryptocurrency contract in Ethereum - Ethereum.org**
URL: <https://www.ethereum.org/token> | [Diese Seite übersetzen](#)
Wolfram 26. Minimum Viable Token. Notice that this will not create new tokens but charge the balance the contract owns. The contract can hold both its own tokens and Ether and the owner of the contract, while it can set prices or in some cases create new tokens (if applicable) it cannot touch the bank's tokens or Ether.
The Coin - Understanding the code - How to deploy - Improve your token
 - How to issue your own token on Ethereum in less than 20 minutes.**
URL: <https://medium.com/.../how-to-issue-your-own-token-on-ethereu...> | [Diese Seite übersetzen](#)
25.12.2017 - Now let's get started: Open the contract you downloaded in your Text Editor. Go to Line 3-15 and look at the comment section. Change Line 4 to the title of your Smart Contract. Change Line 6 to the Ropsten Ethereum address you created in MyEtherWallet. Change Line 7's Symbol to your respective coin name (Keep it short)
 - How to do an ICO on Ethereum in less than 20 minutes, - Medium**
URL: <https://medium.com/.../how-to-do-an-ico-on-ethereum-in-less-tha...> | [Diese Seite übersetzen](#)
20.09.2018 - Download the smart contract that the legendary Ethereum unicorn rider, ... Now Under Complete--Choose the Token you are creating --Details ...
 - Create your own Ethereum Token. - Hacker Noon**
URL: <https://hackernoon.com/create-your-own-ethereum-token-bfa530...> | [Diese Seite übersetzen](#)
12.01.2018 - Create your own Ethereum Token. Step 1: The ERC20 Interface. In Solidity, the programming language for Ethereum, an interface is a piece of reusable code that defines constraints for functions and events, without implementing the function itself. Step 2: Implementing our functions. Step 3: Add your own touch.
- Other searches also found:**
 - how to create an ico on ethereum
 - ethereum.org/crowdsale
 - mintable token
 - crowdsale ico
 - democracy contract ethereum
 - ethereum token list
- Videos:**
 - ERC-20 SMART

What Does It Take to Create a Token System ?



The screenshot shows a web browser window titled "Create a cryptocurrency in Ethereum - Mozilla Firefox". The address bar shows "https://www.ethereum.org/token". The page content includes a Google search bar, a navigation menu, and a main heading "The Coin". Below the heading is an introductory paragraph about digital tokens in the Ethereum ecosystem. A section titled "MINIMUM VIABLE TOKEN" explains that a standard token contract can be complex but can be simplified. A code block contains Solidity code for a "MyToken" contract. Below the code is a section titled "THE CODE" with a note about copying the code and an "OPEN CHAT" button.

The Coin

We are going to create a digital token. Tokens in the Ethereum ecosystem can represent any fungible tradable good: coins, loyalty points, gold certificates, IOUs, in-game items, etc. Since all tokens implement some basic features in a standard way, this also means that your token will be instantly compatible with the Ethereum wallet and any other client or contract that uses the same standards.

MINIMUM VIABLE TOKEN

The standard token contract can be quite complex. But in essence a very basic token boils down to this:

```
pragma solidity ^0.4.20;

contract MyToken {
  /* This creates an array with all balances */
  mapping (address => uint256) public balanceOf;

  /* Initializes contract with initial supply tokens to the creator of the contract */
  function MyToken(
    uint256 initialSupply
  ) public {
    balanceOf[msg.sender] = initialSupply; // Give the creator all initial tokens
  }

  /* Send coins */
  function transfer(address _to, uint256 _value) public returns (bool success) {
    require(balanceOf[msg.sender] >= _value); // Check if the sender has enough
    require(balanceOf[_to] + _value >= balanceOf[_to]); // Check for overflows
    balanceOf[msg.sender] -= _value; // Subtract from the sender
    balanceOf[_to] += _value; // Add the same to the recipient
    return true;
  }
}
```

THE CODE

But if you just want to copy paste a more complete code, then use this:

[OPEN CHAT](#)

What Does It Take to Create a Token System ?

The screenshot shows the 'Create a cryptocurrency' page on Ethereum.org. The page title is 'HOW TO DEPLOY'. The main text reads: 'If you aren't there already, open the Ethereum Wallet, go to the contracts tab and then click "deploy new contract". Now get the token source from above and paste it into the "Solidity source field". If the code compiles without any error, you should see a "pick a contract" drop-down list on the right. Get it and select the "MyToken" contract. On the right column, you'll see all the parameters you need to personalize your own token. You can tweak them as you please, but for the purpose of this tutorial we recommend you to pick these parameters: 10,000 as the supply, any name you want, "%" for a symbol and 2 decimal places. Your app should be looking like this:

The Ethereum Wallet overlay shows the 'CONTRACTS' tab with 1,869.42 ETH. The 'SOLIDITY CONTRACT SOURCE CODE' is as follows:

```
47 event Transfer(address indexed from, address indexed to, uint256 value);
48
49 /* Initializes contract with initial supply tokens to the creator of the
50 function MyToken(uint256 _supply, string _name, string _symbol, uint8 _ok
51 */ /* If supply not given then generate 1 million of the smallest unit
52 if (_supply == 0) _supply = 1000000;
53
54 /* Unless you add other functions these variables will never change */
55 balanceOf[msg.sender] = _supply;
56 name = _name;
57 symbol = _symbol;
58
59 /* If you want a divisible token then add the amount of decimals the
60 decimals = _decimals;
61 }
62
63 /* Send coins */
64 function transfer(address _to, uint256 _value) {
65     /* If the sender doesn't have enough balance then stop */
66     if (balanceOf[msg.sender] < _value) throw;
67     if (balanceOf[_to] + _value > balanceOf[_to]) throw;
68
69     /* Add and subtract new balances */
70     balanceOf[msg.sender] -= _value;
71     balanceOf[_to] += _value;
72
73     /* Notify anyone (listening) that this transfer took place */
74     Transfer(msg.sender, _to, _value);
75
76 }
```

The 'CONTRACT BYTE CODE' section is empty. The 'SELECT CONTRACT TO DEPLOY' dropdown is set to 'MyToken'. The 'CONSTRUCTOR PARAMETERS' are:

- _supply: 256 bits unsigned integer: 10000
- _name: String: My DAO Shares
- _symbol: String: %
- _decimals: 8 bits unsigned integer: 2

An 'OPEN CHAT' button is visible at the bottom right of the wallet overlay.

What Does It Take to Create a Token System?

Ethereum Project - Mozilla Firefox

https://www.ethereum.org

Devcon 4 takes place between Oct 30th and Nov 2nd, 2018.

The project was bootstrapped via an ether presale in August 2014 by fans all around the world. It is developed by the [Ethereum Foundation](#), a Swiss non-profit, with contributions from great minds across the globe.

Smart money, smart wallet

The **Ethereum Wallet** is a gateway to decentralized applications on the Ethereum blockchain. It allows you to hold and secure ether and other crypto-assets built on Ethereum, as well as write, deploy and use smart contracts.

DOWNLOAD
Ethereum Wallet for Linux

[See all versions](#)

Easy template-based contract creation

1000000
Token name: ETH
My New Token

Learn Solidity, a new language for smart

What Does It Take to Create a Token System?

universität
innsbruck

What Does It Take to Create a Token System?

The image is a composite of two screenshots. The left screenshot shows a Google search for 'create ethereum token', displaying various search results including articles and videos about creating tokens on the Ethereum blockchain. The right screenshot shows the 'Ethereum Wallet' interface in 'Deploy contract' mode. The wallet has a balance of 1.00 ETH. The 'FROM' field is set to 'Account 1 - 1.00 ETH'. The 'AMOUNT' field is set to '0.0'. The 'SEND EVERYTHING' checkbox is checked. Below the form, there are two tabs: 'SOLIDITY CONTRACT SOURCE CODE' and 'CONTRACT BYTE CODE'. The Solidity code is visible, showing a contract named 'MyToken' with an initial supply of 10000. The 'SELECT CONTRACT TO DEPLOY' dropdown is set to 'My Token'. The 'CONSTRUCTOR PARAMETERS' field is set to '10000'. At the bottom, there is a 'SELECT FEE' section.

Ethereum Wallet Deploy Contract Interface:

- Account: Account 1 - 1.00 ETH
- Amount: 0.0
- Send everything:
- You want to send: 0 ETH
- Contract Name: My Token
- Constructor Parameters: 10000
- Contract Source Code (Solidity):

```
1 pragma solidity ^0.4.20;
2
3 contract MyToken {
4     /* This creates an array with all balances */
5     mapping(address => uint256) public balanceOf;
6
7     /* Initializes contract with initial supply tokens to the creator of the
8     contract */
9     uint256 initialSupply;
10    public {
11        balanceOf[msg.sender] = initialSupply; // Give the cre
12    }
13
14    /* Send coins */
15    function transfer(address _to, uint256 _value) public returns (bool suc
16        require(balanceOf[msg.sender] >= _value); // Check if the
17        balanceOf[_to] += _value; // Add the new
18        balanceOf[msg.sender] -= _value; // Subtract fr
19        balanceOf[_to] += _value; // Add the new
20        return true;
21    }
22 }
23
24
```

What Does It Take to Create a Token System?

The image is a collage illustrating the process of creating a token system. It features three main components:

- Search Results:** A browser window showing search results for "The Coin". The results include articles about creating a cryptocurrency token, with titles like "HOW TO DEPLOY The Coin" and "The Ethereum Wallet is a gateway to decentralized applications".
- Solidity Contract Source Code:** A code editor displaying the source code for a Solidity contract named "MyToken". The code includes a pragma statement for solc, an initial supply of 256, and a transfer function that updates the balance of the sender and receiver.
- Ethereum Wallet Interface:** The Ethereum Wallet application showing the "Create contract" screen. It displays the estimated fee consumption (0.0007482 ether), the gas price (0.004 ether per million gas), and the total amount to be sent (0.0007482 ether). The interface also shows a "SEND TRANSACTION" button and a "SENDING..." status.

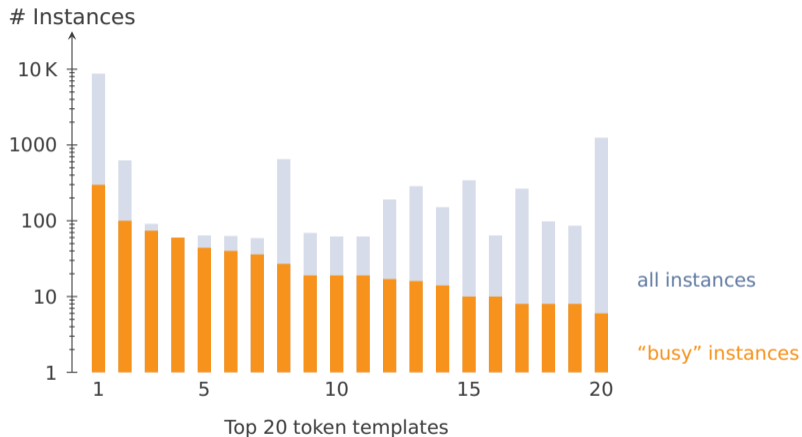
What Does It Take to Create a Token System?

The collage features several overlapping elements:

- A browser window with the title "HOW TO DEPLOY" and text: "The project was bootstrapped as another personal project. We aim to create a digital token taking in the Ethereum system can represent any fungible tradable good, commodity, points, gold certificates, etc. In general, tokens are used to represent any other client or contract that uses the same standard."
- A "Deploy contract" button from an "Ethereum Wallet" interface.
- Snippets of Solidity code, including:

```
prague solidity *0.4.20; function transfer(address to, uint value) public { require(balanced(msg.sender) >= value); _transfer(msg.sender, to, value); return true; }
```
- A "Create contract" dialog box with a "SEND TRANSACTION" button.
- A central white box with a black border containing the text: "You are now the owner of your own token system!"
- Other visible text includes "Smart money, smart wallet" and "Learn Solidity - a new language for smart".

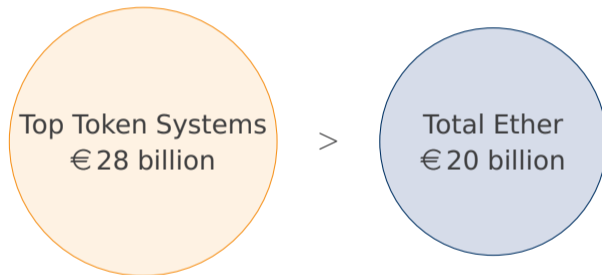
Code Reuse in Ethereum



Source: own research, data until May 30th, 2018

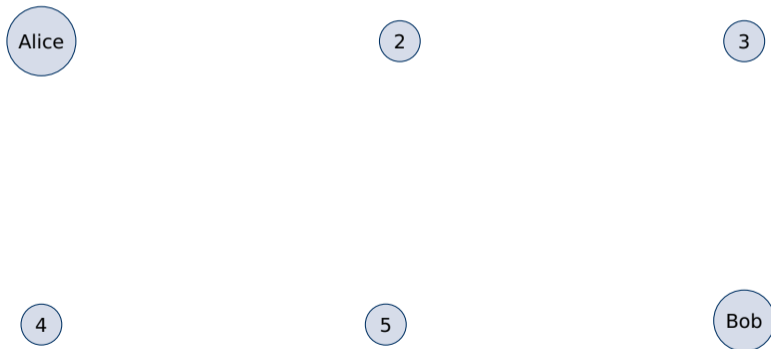
Why Are Token Systems Interesting ?

Market valuation:

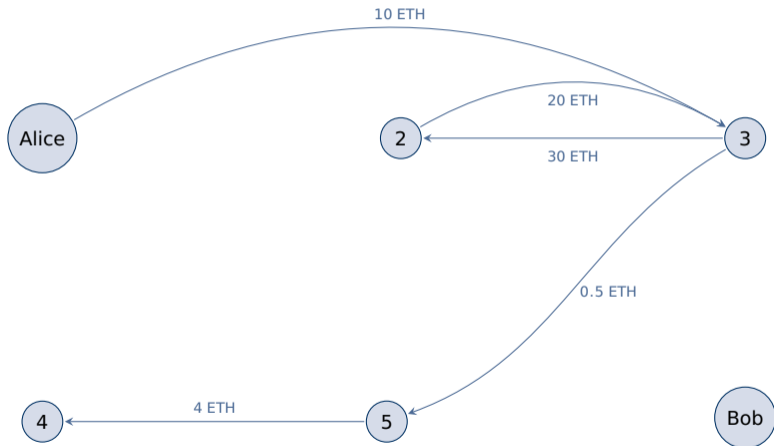


Sources: Etherscan, coinmarketcap.com, October 9th, 2018

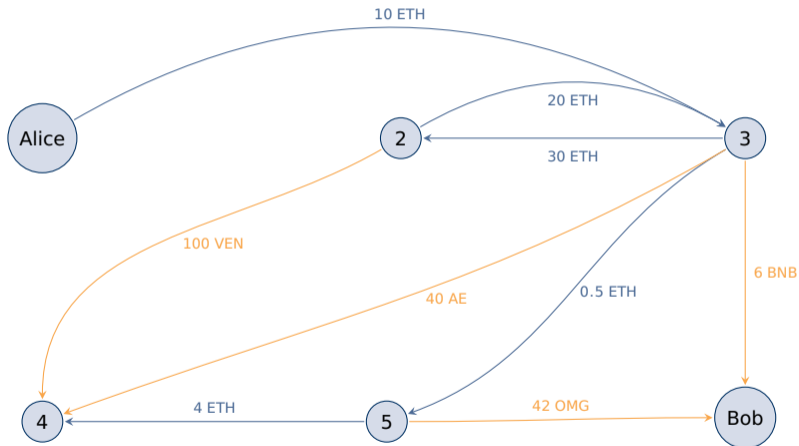
Why Are Token Systems Interesting?



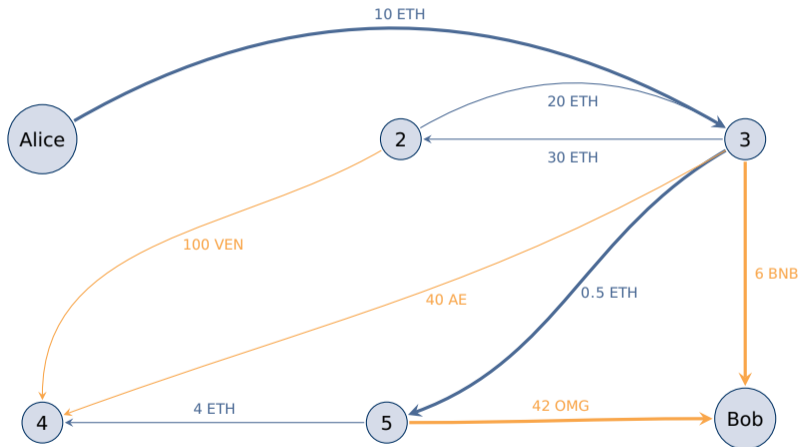
Why Are Token Systems Interesting?



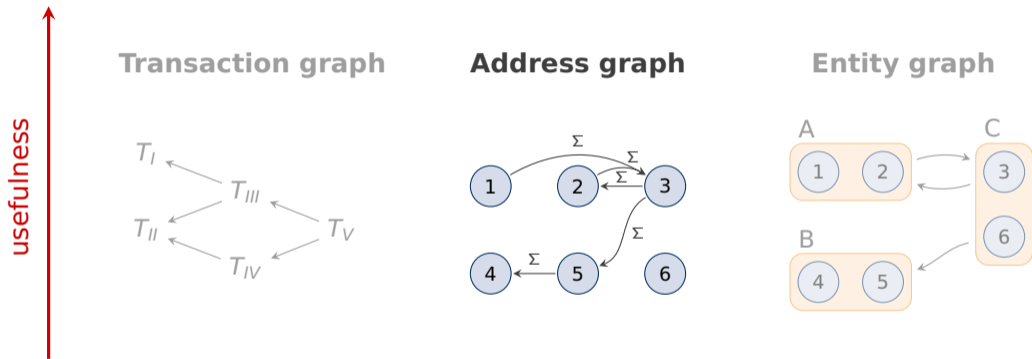
Why Are Token Systems Interesting?



Why Are Token Systems Interesting?



Tracking Payment Flows: From Bitcoin to Ethereum



Challenges

Only considering Ether flows gives an incomplete picture.

Challenges

Only considering Ether flows gives an incomplete picture.

Upside:

- Many CAs follow standards to make them more observable.

Challenges

Only considering Ether flows gives an incomplete picture.

Upside:

- Many CAs follow standards to make them more observable.

Challenges:

- But there is no obligation for CAs to be transparent.

Challenges

Only considering Ether flows gives an incomplete picture.

Upside:

- Many CAs follow standards to make them more observable.

Challenges:

- But there is no obligation for CAs to be transparent.
- Automatic identification of the purpose of a system is a research topic.

Take Home Messages

Privacy was not a design goal of Ethereum

Take Home Messages

Privacy was not a design goal of Ethereum

- Identifiers are more sticky than in Bitcoin.



Take Home Messages

Privacy was not a design goal of Ethereum

- Identifiers are more sticky than in Bitcoin.
- Ethereum wallets are built for one identifier per person.



Take Home Messages

Privacy was not a design goal of Ethereum

- Identifiers are more sticky than in Bitcoin.
- Ethereum wallets are built for one identifier per person.
- But there are efforts to improve privacy.



Take Home Messages (cont'd)

Tracking Ether is easy, but . . .

Take Home Messages (cont'd)

Tracking Ether is easy, but . . .

- incomplete.

Take Home Messages (cont'd)

Tracking Ether is easy, but . . .

- incomplete.
- tokens and other services must be considered.

Take Home Messages (cont'd)

Tracking Ether is easy, but . . .

- incomplete.
- tokens and other services must be considered.
- programmers of CAs decide what is easily observable.

Take Home Messages (cont'd)

Tracking Ether is easy, but . . .

- incomplete.
- tokens and other services must be considered.
- programmers of CAs decide what is easily observable.
- we lack adequate tools for investigations and monitoring.

Take Home Messages (cont'd)

Why should regulators care?

Ethereums is not criminals' first choice. But ...



Take Home Messages (cont'd)

Why should regulators care?

Ethereums is not criminals' first choice. But ...

- everyone can issue their own financial products. → Security regulation



Take Home Messages (cont'd)



Why should regulators care?

Ethereums is not criminals' first choice. But ...

- everyone can issue their own financial products.
- there are shady services and business practices.

→ Security regulation

→ Consumer protection

Take Home Messages (cont'd)



Why should regulators care?

Ethereums is not criminals' first choice. But ...

- everyone can issue their own financial products. → Security regulation
- there are shady services and business practices. → Consumer protection

Ethereum is relevant and different.

Regulators who take Bitcoin as a model may **miss** important aspects.



Tracking Payment Flows in Ethereum

Thank you for your attention

Michael Fröwis · michael.froewis@uibk.ac.at

Fungible vs. Non-fungible

Asset



Fungible



Non-fungible