A Deal with the Devil:
Breaking Smart Contracts

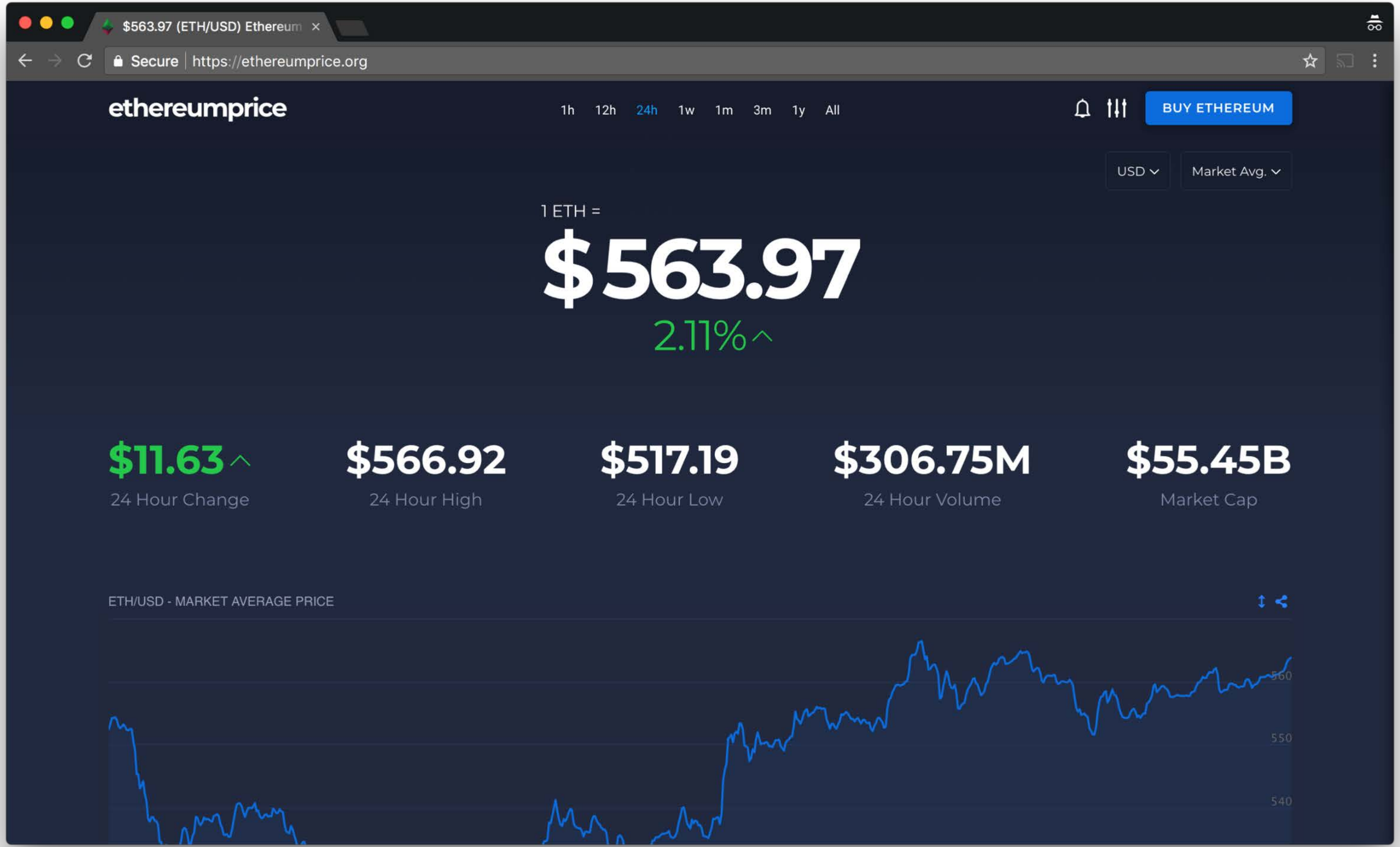
David Wong, Mason Hemmel

nccgroup

History

- 2009 - Bitcoin
- 2014/2015 - Ethereum

Ethereum is a computer
— Gavin Wood


$563.97 (ETH/USD) Ethereum
Secure
https://ethereumprice.org
ethereumprice
1h
12h
24h
1w
1m
3m
1y
All
BUY ETHEREUM
USD
Market Avg.
1 ETH =
$563.97
2.11%
$11.63
24 Hour Change
$566.92
24 Hour High
$517.19
24 Hour Low
$306.75M
24 Hour Volume
$55.45B
Market Cap
ETH/USD - MARKET AVERAGE PRICE
560
550
540

Source Vitalik Buterin's Style -

David

Secure https://www.vitalikwears.com

Vitalik**WEARS**

What on earth is Vitalik Buterin wearing now? **Something awesome, that's what!** Scroll down to sauce these badboys...

NOTE: Anyone could have ripped off these designs & sold them using print on demand. Instead, I have done my best to find and support the original artists. |

TIPJAR: 0x953dD20Ee7d304B3DcB5D9e9C3bb261644C2F14A

White Paper

Ammar Husain edited this page 20 hours ago · 113 revisions

A Next-Generation Smart Contract and Decentralized Application Platform

Satoshi Nakamoto's development of Bitcoin in 2009 has often been hailed as a radical development in money and currency, being the first example of a digital asset which simultaneously has no backing or "intrinsic value" and no centralized issuer or controller. However, another - arguably more important - part of the Bitcoin experiment is the underlying blockchain technology as a tool of distributed consensus, and attention is rapidly starting to shift to this other aspect of Bitcoin. Commonly cited alternative applications of blockchain technology include using on-blockchain digital assets to represent custom currencies and financial instruments ("colored coins"), the ownership of an underlying physical device ("smart property"), non-fungible assets such as domain names ("Namecoin"), as well as more complex applications involving having digital assets being directly controlled by a piece of code implementing arbitrary rules ("smart contracts") or even blockchain-based "decentralized autonomous organizations" (DAOs). What Ethereum

ETHEREUM: A SECURE DECENTRALISED GENERALISED TRANSACTION LEDGER

EIP-150 REVISION

DR. GAVIN WOOD
FOUNDER, ETHEREUM & ETHCORE
GAVIN@ETHCORE.IO

ABSTRACT. The blockchain paradigm when coupled with cryptographically-secured transactions has demonstrated its utility through a number of projects, not least Bitcoin. Each such project can be seen as a simple application on a decentralised, but singleton, compute resource. We can call this paradigm a transactional singleton machine with shared-state.

Ethereum implements this paradigm in a generalised manner. Furthermore it provides a plurality of such resources, each with a distinct state and operating code but able to interact through a message-passing framework with others. We discuss its design, implementation issues, the opportunities it provides and the future hurdles we envisage.

1. INTRODUCTION

information is often lacking, and plain old prejudices are difficult to shake.

Beigepaper:
An Ethereum Technical Specification

Micah Dameron

Abstract

The Ethereum Protocol is a deterministic but practically unbounded state-machine with two basic functions; the first being a globally accessible singleton state, and the second being a virtual machine that applies changes to that state. This paper explains the individual parts that make up these two factors.

1. Imagining Bitcoin as a Computer

the network with excessive computational expenditures. The smallest unit of currency in Ethereum is the Wei, which is equal to $\Xi 10^{-18}$, where Ξ stands for

Ethereum 2.0 Mauve Paper

Because mauve has the most RAM...

Over the past decade, projects such as Bitcoin, Namecoin and Ethereum have shown the power of cryptoeconomic consensus networks to bring about the next stage in evolution of decentralized systems, potentially expanding their scope from simply providing for the storage of data and messaging services to managing the "back-end" of arbitrary stateful applications. Proposed and implemented applications for such systems range from globally accessible cheap payment systems to financial contracts, prediction markets, registration of identity and real world property ownership, building more secure certificate authority systems and even keeping track of the movement of manufactured goods through the supply chain.

However, there remain serious efficiency concerns with the technological underpinnings of such systems. Because every full node in the network must maintain the entire state of the system and process every transaction, the network can never be more powerful than a single computer. The consensus mechanism most often used in existing systems, proof of work, consumes a very large amount of electricity in order to operate; the largest working blockchain using this mechanism, Bitcoin, has been shown to consume as much electricity as the entire country of Ireland.

This document proposes a solution to these problems based on the combination of proof of stake and sharding. Proof of stake itself is not a novel idea, having existed since 2011, but this new algorithm presents substantial benefits, both solving flaws in previous systems and even introducing new properties that are not present in proof of work. Proof of stake can be thought of as a kind of "virtual mining": whereas in proof of work, users can spend real-world dollars to buy real computers which expend electricity and stochastically produce blocks at a rate roughly proportional to the cost expended, in proof of stake, users spend real-world dollars to buy virtual coins inside the system, and then use an in-protocol mechanism to convert the virtual coins into virtual computers, which are simulated by the protocol to

MELON PROTOCOL: A BLOCKCHAIN PROTOCOL FOR DIGITAL ASSET MANAGEMENT DRAFT

RETO TRINKLER AND MONA EL ISA

ABSTRACT. The Melon protocol is a blockchain protocol for digital asset management on the Ethereum platform. It enables participants to set up, manage and invest in digital asset management strategies in an open, competitive and decentralised manner.

1. INTRODUCTION

The value and importance of a wide range of digital assets[1] has risen dramatically over the last few years. Hence the question naturally arises how to manage this new and fast-growing asset class in the most advantageous way.

This could be done by investing in a hedge fund which

Digital assets which do not gain their value from collateralisation, called *un-collateralised assets*. Finally, digital assets which are derived from existing digital assets called *derivatives*.

2.1. **Collateralised Assets.** Collateralised Assets, are assets which gain their value from the collateralisation of

POLKADOT: VISION FOR A HETEROGENEOUS MULTI-CHAIN FRAMEWORK

DRAFT 1

DR. GAVIN WOOD
FOUNDER, ETHEREUM & PARITY
GAVIN@PARITY.IO

ABSTRACT. Present-day blockchain architectures all suffer from a number of issues not least practical means of extensibility and scalability. We believe this stems from tying two very important parts of the consensus architecture, namely *canonicality* and *validity*, too closely together. This paper introduces an architecture, the *heterogeneous multi-chain*, which fundamentally sets the two apart.

In compartmentalising these two parts, and by keeping the overall functionality provided to an absolute minimum of *security* and *transport*, we introduce practical means of core extensibility in situ. Scalability is addressed through a divide-and-conquer approach to these two functions, scaling out of its bonded core through the incentivisation of untrusted public nodes.

The heterogeneous nature of this architecture enables many highly divergent types of consensus systems interoperating in a trustless, fully decentralised "federation", allowing open and closed networks to have trust-free access to each other.

We put forward a means of providing backwards compatibility with one or more pre-existing networks such as Ethereum [6, 22]. We believe that such a system provides a useful base-level component in the overall search for a practically implementable system capable of achieving global-commerce levels of scalability and privacy.

what are

smart contracts

?

Solidity — Solidity 0.4.20 docu ×

David

Secure https://solidity.readthedocs.io/en/develop/

Solidity

Solidity is a contract-oriented, high-level language for implementing smart contracts. It was influenced by C++, Python and JavaScript and is designed to target the Ethereum Virtual Machine (EVM).

Solidity is statically typed, supports inheritance, libraries and complex user-defined types among other features.

As you will see, it is possible to create contracts for voting, crowdfunding, blind auctions, multi-signature wallets and more.

Note

The best way to try out Solidity right now is using Remix (it can take a while to load, please be patient).

```
pragma solidity 0.4.19;

contract BlackHat {

    string public conferanceName = "Black Hat";
    string public country = "Asia";
    uint256 public numberOfAttendees;
    uint256 public ticketPrice

    mapping(address => uint256) signedUp;

    function BlackHat(uint256 _ticketPrice) public {
        ticketPrice = _ticketPrice;
    }

    function signUpToBlackHat() public payable {
        require(msg.value > ticketPrice);

        signedUp[msg.sender] = true;
    }

    event AttendingTalk(address who, uint16 talkId);

    function attendBlackHat(uint16 _talkId) public {

        AttendingTalk(msg.sender, _talkId);
    }
```

david — _zsh_tmux_plugin_run — tmux


```
6060604052604080519081016040528060088152602001 7f43697479436f696e000000000000000000000000000000000000000000000000815250600090805190602001906200
0051929190620002d6565b506040805190810160405280600481526020017f434954590000000000000000000000000000000000000000000000000000008152506001908051
90602001906200009f929190620002d6565b506000600260006101000a81548160ff021916908360ff160217905550611388600660146101000a81548161ffff021916908361ff
ff160217905550624c4b406007556113886008556003600955606 4600a556003600b556003600c5534156200010857600080fd5b600033600660006101000a81548173ffffffff
ffffffffffffffffffffffffffffffff021916908373ffffffffffffffffffffffffffffffffffffffff16021790555060075460036000600660009054906101000a900473ffff
ffffffffffffffffffffffffffffffffffff1673ffffffffffffffffffffffffffffffffffffffff1673ffffffffffffffffffffffffffffffffffffffff168152602001908152
602001600020819055506207a1209050806003600073e1811ec49f493afb1f4b42e3ef4a3b9d62d9a01b73ffffffffffffffffffffffffffffffffffffffff1673ffffffffffff
ffffffffffffffffffffffffffff1681526020019081526020016000208190555080600360 00731e4f1275bb041586d7bec44d2e3e4f30e0da7ba473ffffffffffffffffffffff
ffffffffffffffffff1673ffffffffffffffffffffffffffffffffffffffff16815260200190815260200160002081905550806003600073d5d6301de62d82f461dc29824fc597
d38d80c42473ffffffffffffffffffffffffffffffffffffffff1673ffffffffffffffffffffffffffffffffffffffff1681526020019081526020016000208190555080600302
600754016004819055505062000385565b828054600181600116156101000203166002900490600052602060002090601f016020900481019282601f106200031957805160ff19
1683800117855562000034a565b8280016001018555821562000034a579182015b828111156200034957825182559160200191906001019062000032c565b5b5090506200035991 90
6200035d565b5090565b6200038291905b808211156200037e576000816000905550600101620003645 65b5090565b90565b615b938062000395600039 6000f300606060405260
0436106101e3576000357c0100000000000000000000000000000000000000000000000000000000900463ffffffff16806306fdde03146101e857806308e9daa5146102765780
63095ea7b31461032f57806318160ddd146103895780 6323b872dd146103b2578063313ce5671461042b57806337 f832911461045a5780634a87e08a1461055c5780634c34d7e9
146105855780635a768e4a146105ae578063661884631461 05d75780636a51b918146106315780636e2c10a114610745578063 70a082311461079a5780637c5b4a37146107e757
806382b57b961461080a578063839acf14146108335780638973e3161461085f5780638a72ea6a1461087b57806395d89b41146108f457806396dbad1e146109825780639e65a0
3a146109b2578063a0bde945146109d9578063a25fc8dc146109f5578063a2741bdb14610a7c578063a27c175614610aad578063a9059cbb14610ad6578063ae93e51314610b30
578063b2bdc6b314610b53578063bd1de90014610bda578063d02e8f7014610c03578063d73dd62314610c2c578063db5a277814610c86578063dc1c138914610cf7578063dd62
ed3e14610d30578063e3b5527614610d9c578063e6c89d4a14610de2578063ee5845cc14610e26578063f93e2d3114610e4b575b600080fd5b34156101f357600080fd5b6101fb
610edd565b6040518080602001828103825283818151815260200191508051906020019080838360005b8381101561023b5780820151818401526020810190506102205 65b5050
5050905090810190601f1680156102685780820380516001836020036101000a031916815260200191505b509250505060405180910390f35b3415610281576000 80fd5b610297
6004808035906020019091905050610f7b565b6040518080602001828103825283818154600181600116156101000203166002900481526020019150805460018160011615 6101
000203166002900480156103205780601f106102f557610100808354040283529160200191610320565b820191906000526020600020905b815481529060010190602001808311
61030357829003601f168201915b505092505050604051809103 90f35b341561033a57600080fd5b61036f600480803573ffffffffffffffffffffffffffffffffffffffff1690
602001909190803590602001909190505050610fa7565b604051808215151515815260200191505060405180910390f35b341561039457600080fd5b61039c611099565b60405180
8281526020019150506040518091039 0f35b34156103bd57600080fd5b610411600480803573ffffffffffffffffffffffffffffffffffffffff16906020019091908035 73ffff
ffffffffffffffffffffffffffffffffffff169060200190919080359060200190919050506110a3565b604051808215151515815260200191505060405180910390f35b341561
043657600080fd5b61043e611462565b604051808260ff1660ff16815260200191505060405180910390f35b341561046557600080fd5b61047b600480803590602001909190 50
50611475565b6040518080602001858152602001847 3ffffffffffffffffffffffffffffffffffffffff1673ffffffffffffffffffffffffffffffffffffffff16815260200183
61ffff1661ffff16815260200182810382528681815460018160011615610100020316600290048152602001915080546001816001161561010002031660029004801561054a57
80601f1061051f576101008083540402835291602001916105 4a565b820191906000526020600020905b815481529060010190602001808311610 52d57829003601f168201915b
505095505050505050604051809103 90f35b341561056757600080fd5b61056f6114e1565b6040518082815260200191505060405180910390f35b341561059057600080fd5b61
```

david — _zsh_tmux_plugin_run — tmux

```
EVM assembly:
    /* "SafeMath.sol":26:1060  library SafeMath {... */
  mstore(0x40, 0x60)
  jumpi(tag_1, iszero(callvalue))
  0x0
  dup1
  revert
tag_1:
  dataSize(sub_0)
  dup1
  dataOffset(sub_0)
  0x0
  codecopy
  0x0
  return
stop

sub_0: assembly {
        /* "SafeMath.sol":26:1060  library SafeMath {... */
      mstore(0x40, 0x60)
      0x0
      dup1
      revert

    auxdata: 0xa165627a7a723058206c9bd66ef0efe163a41f5663b86835819321d24be410768f14bb77ef7c2f8e2d0029
}


======= TokenCity.sol:TokenCity =======
EVM assembly:
    /* "TokenCity.sol":52:19141  contract TokenCity {... */
  mstore(0x40, 0x60)
    /* "TokenCity.sol":128:159  string public name = "CityCoin" */
  0x40
```

what are
smart contracts
used for?

IPO

~~IPO~~

ICO

Preamble

```
EIP: 20
Title: ERC-20 Token Standard
Author: Fabian Vogelsteller <fabian@ethereum.org>, Vitalik Buterin <vitalik.buterin@ethereum.org>
Type: Standard
Category: ERC
Status: Accepted
Created: 2015-11-19
```

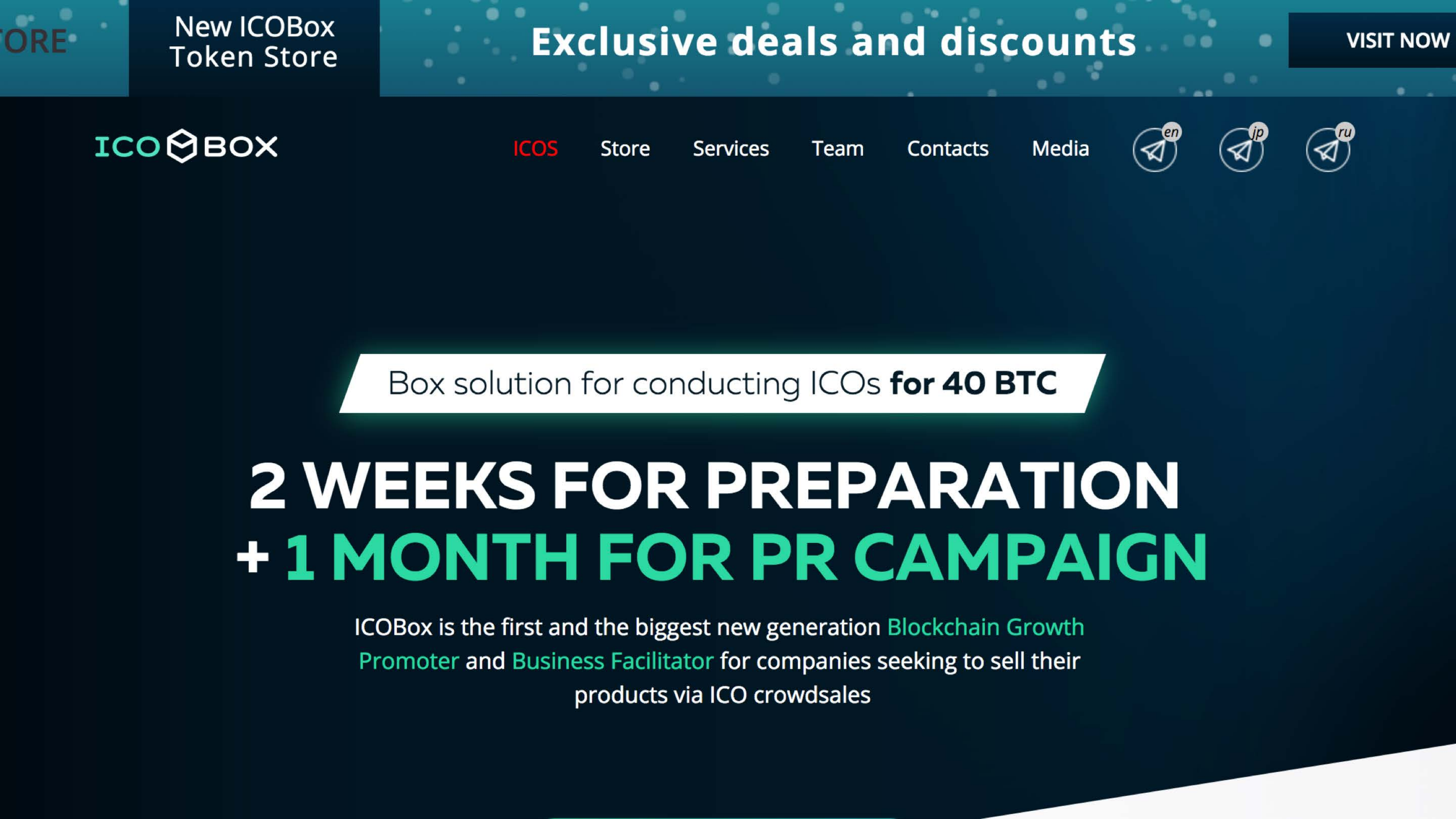
Simple Summary

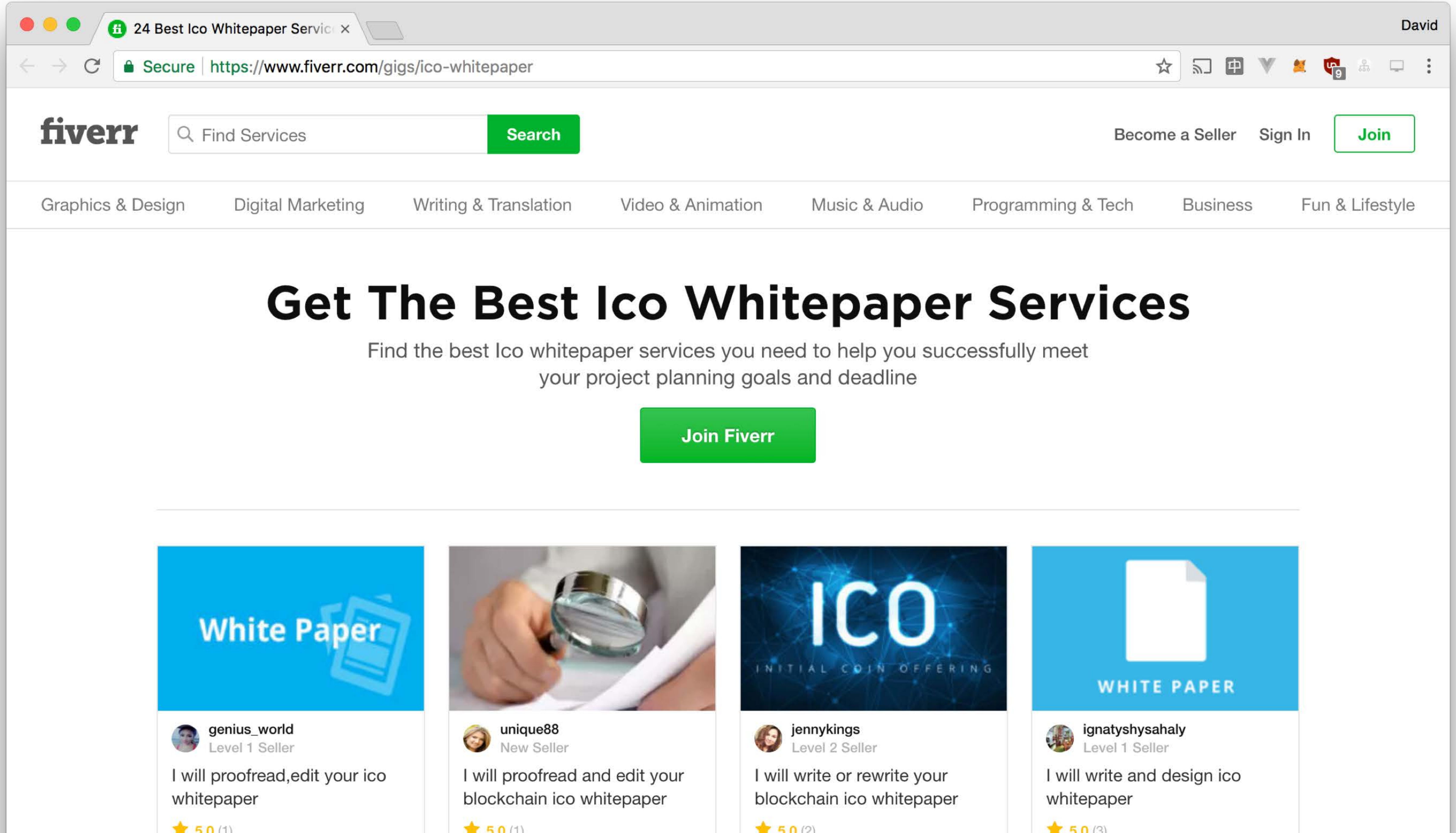
A standard interface for tokens.

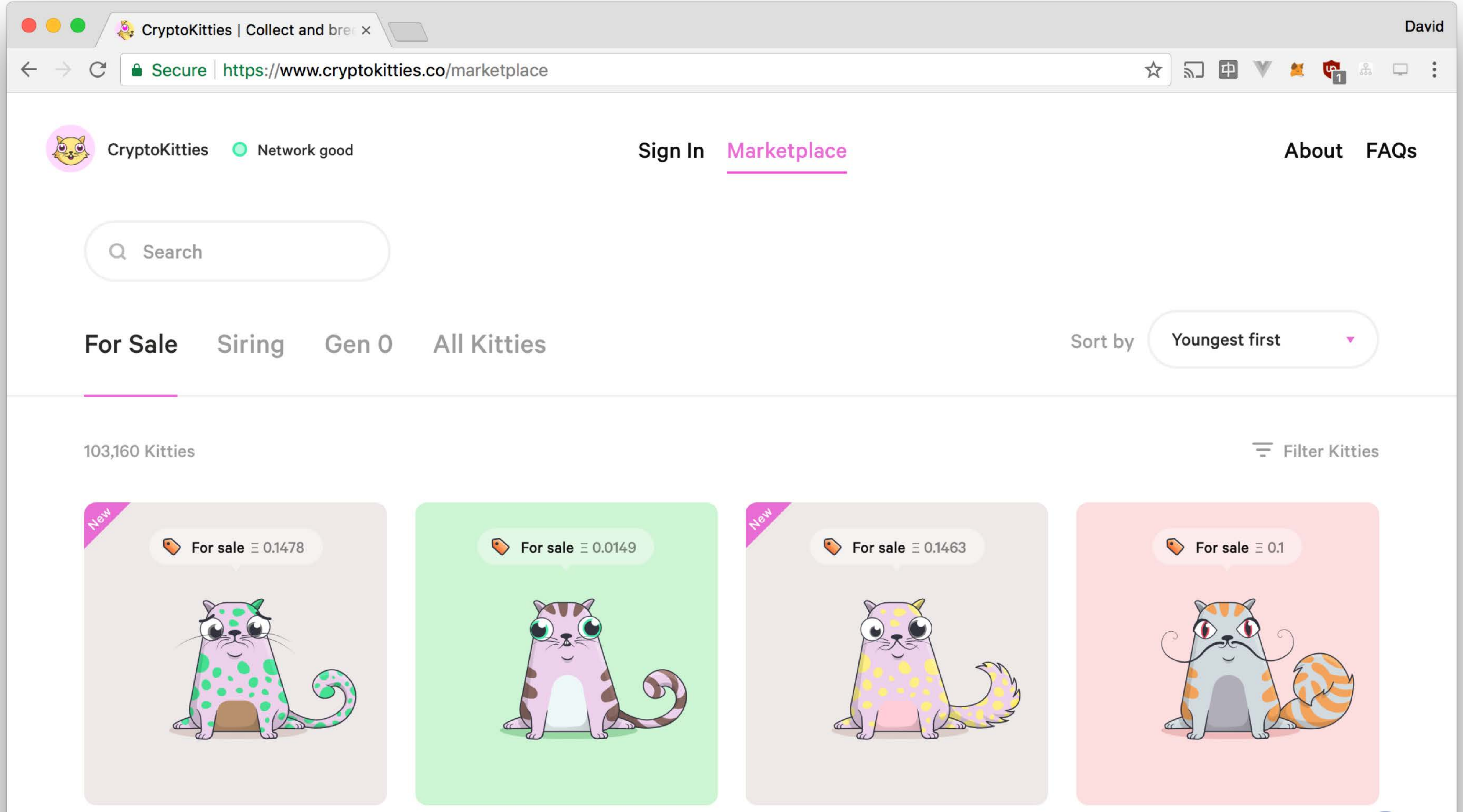
Abstract

The following standard allows for the implementation of a standard API for tokens within smart contracts. This standard provides basic functionality to transfer tokens, as well as allow tokens to be approved so they can be spent by another on-chain third party.

Motivation

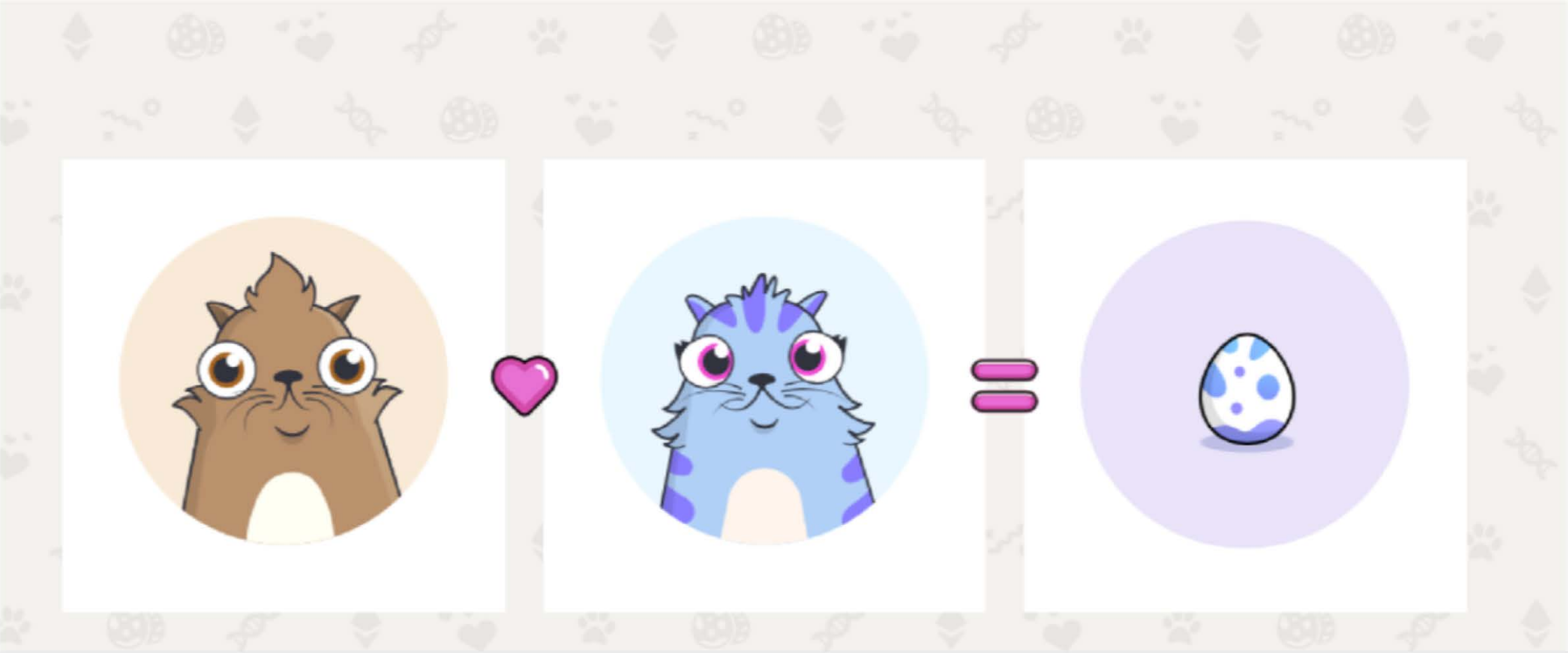

ORE
New ICOBox Token Store
Exclusive deals and discounts
VISIT NOW
ICO BOX
ICOS
Store
Services
Team
Contacts
Media
en
jp
ru
Box solution for conducting ICOs **for 40 BTC**
2 WEEKS FOR PREPARATION
+ 1 MONTH FOR PR CAMPAIGN
ICOBox is the first and the biggest new generation Blockchain Growth Promoter and Business Facilitator for companies seeking to sell their products via ICO crowdsales

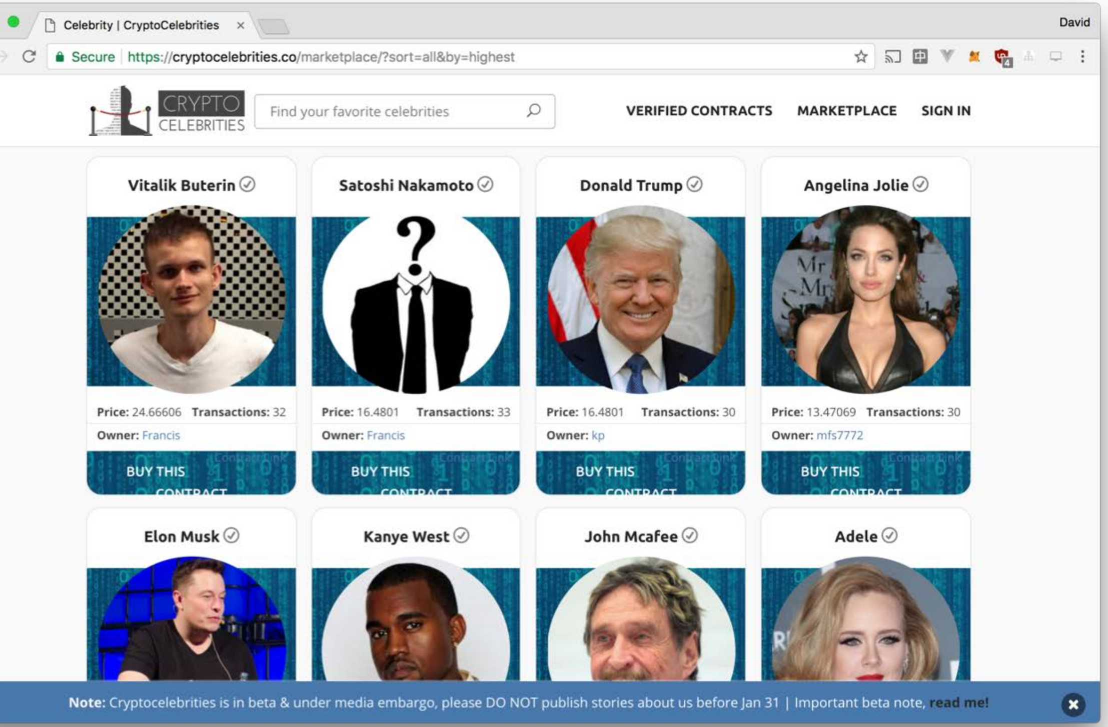

24 Best Ico Whitepaper Servic
David
Secure
https://www.fiverr.com/gigs/ico-whitepaper
fiverr
Find Services
Search
Become a Seller
Sign In
Join
Graphics & Design
Digital Marketing
Writing & Translation
Video & Animation
Music & Audio
Programming & Tech
Business
Fun & Lifestyle
Get The Best Ico Whitepaper Services
Find the best Ico whitepaper services you need to help you successfully meet your project planning goals and deadline
Join Fiverr
White Paper
ICO
INITIAL COIN OFFERING
WHITE PAPER
genius_world
Level 1 Seller
I will proofread,edit your ico whitepaper
unique88
New Seller
I will proofread and edit your blockchain ico whitepaper
jennykings
Level 2 Seller
I will write or rewrite your blockchain ico whitepaper
ignatyshysahaly
Level 1 Seller
I will write and design ico whitepaper

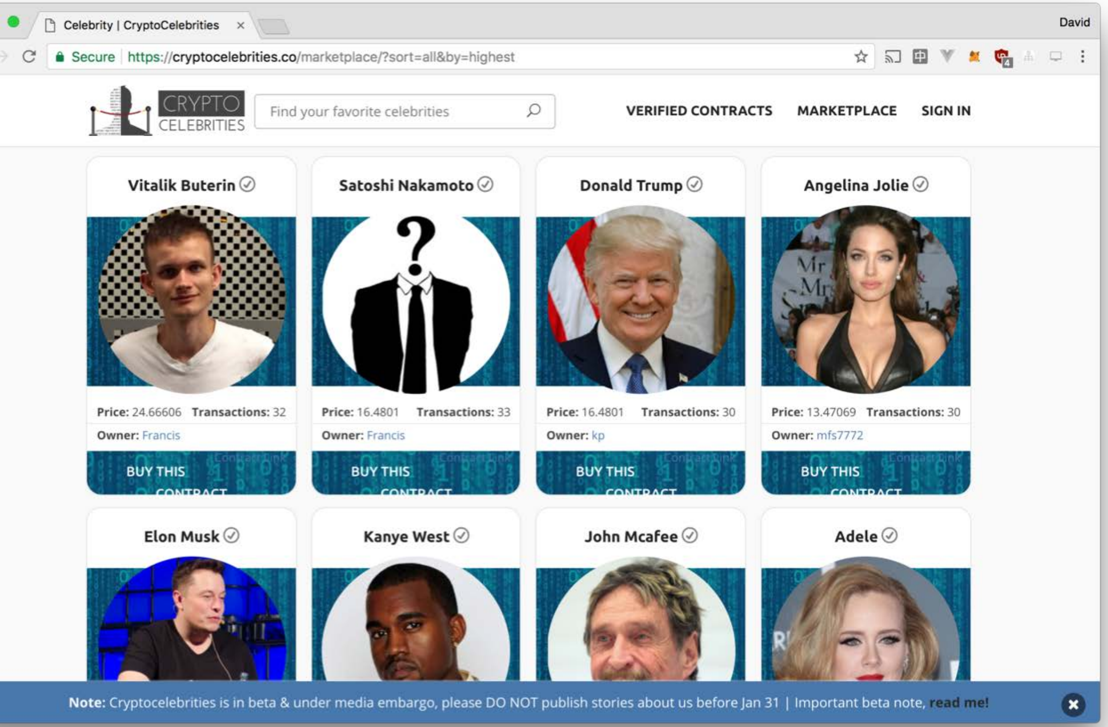

CryptoKitties | Collect and bre
David
Secure
https://www.cryptokitties.co/marketplace
CryptoKitties
Network good
Sign In
Marketplace
About
FAQs
Search
For Sale
Siring
Gen 0
All Kitties
Sort by
Youngest first
103,160 Kitties
Filter Kitties
New
For sale Ξ 0.1478
For sale Ξ 0.0149
New
For sale Ξ 0.1463
For sale Ξ 0.1

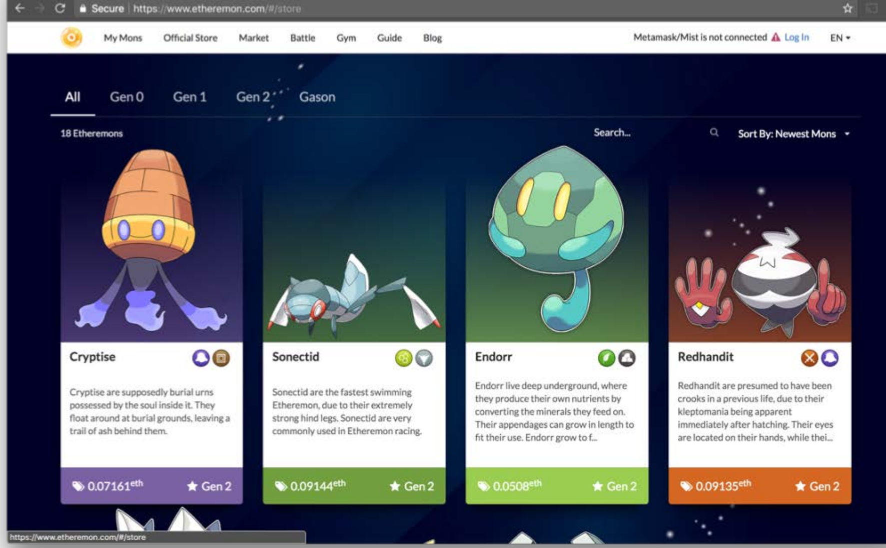
CATS RULE THE BLOCKCHAIN, TOO

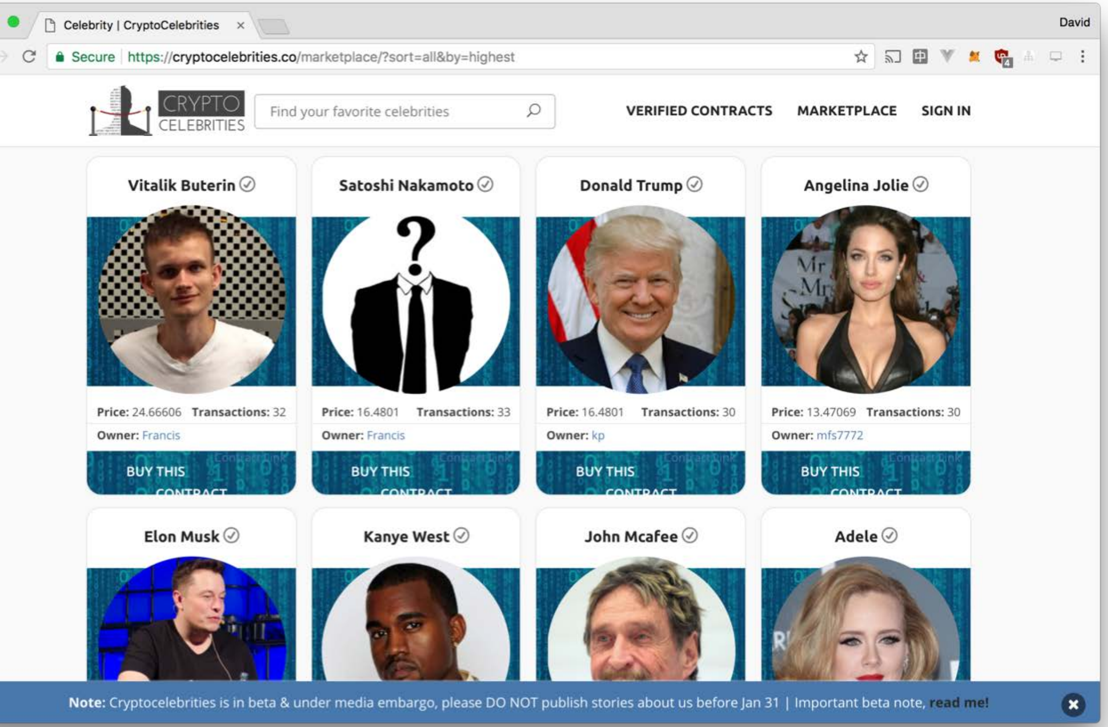
The ethereum network is getting jammed up because people are rushing to buy cartoon cats on its blockchain

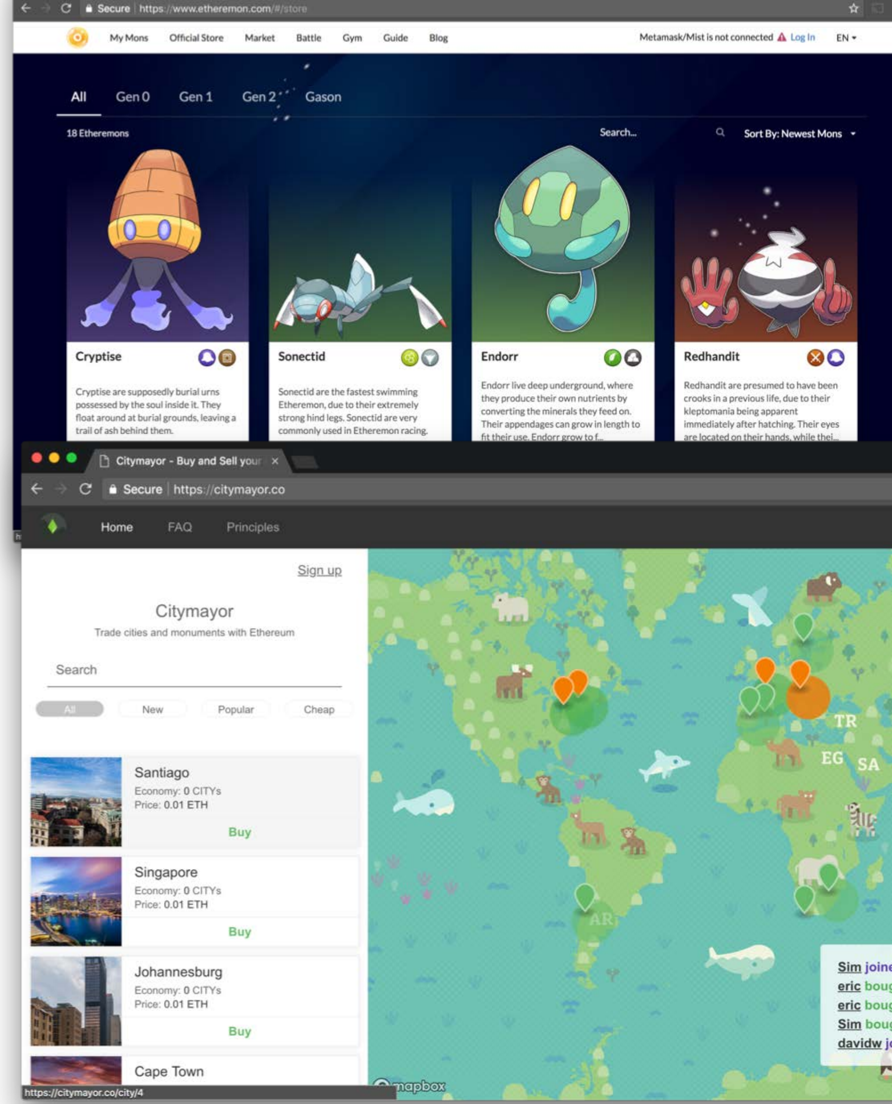


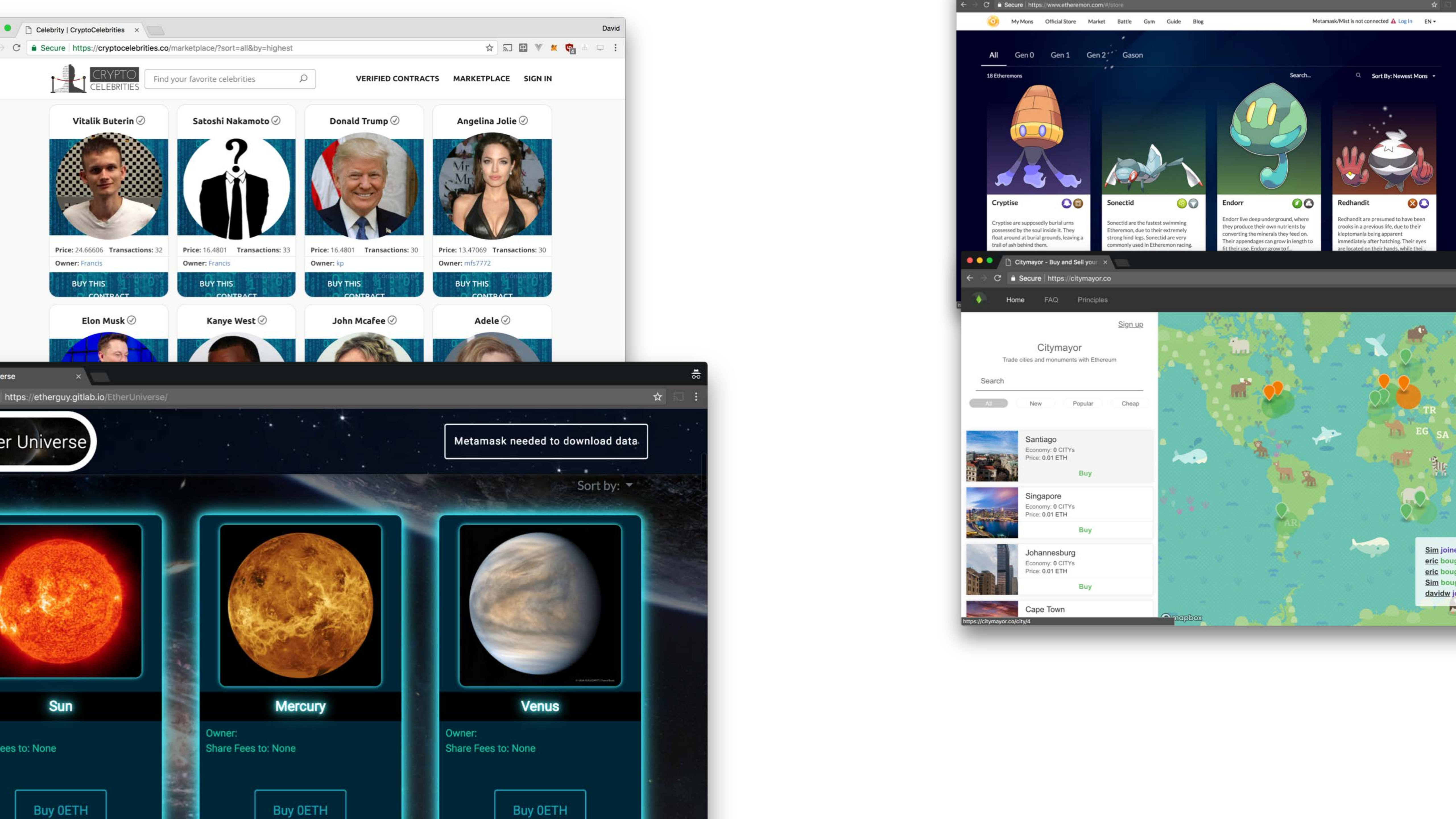

Celebrity | CryptoCelebrities
David
Secure https://cryptocelebrities.co/marketplace/?sort=all&by=highest
CRYPTO CELEBRITIES
Find your favorite celebrities
VERIFIED CONTRACTS
MARKETPLACE
SIGN IN
Vitalik Buterin
Price: 24.66606 Transactions: 32
Owner: Francis
BUY THIS CONTRACT
Satoshi Nakamoto
Price: 16.4801 Transactions: 33
Owner: Francis
BUY THIS CONTRACT
Donald Trump
Price: 16.4801 Transactions: 30
Owner: kp
BUY THIS CONTRACT
Angelina Jolie
Price: 13.47069 Transactions: 30
Owner: mfs7772
BUY THIS CONTRACT
Elon Musk
Kanye West
John Mcafee
Adele
Note: Cryptocelebrities is in beta & under media embargo, please DO NOT publish stories about us before Jan 31 | Important beta note, read me!


Celebrity | CryptoCelebrities
David
Secure https://cryptocelebrities.co/marketplace/?sort=all&by=highest
CRYPTO CELEBRITIES
Find your favorite celebrities
VERIFIED CONTRACTS
MARKETPLACE
SIGN IN
Vitalik Buterin
Price: 24.66606 Transactions: 32
Owner: Francis
BUY THIS CONTRACT
Satoshi Nakamoto
Price: 16.4801 Transactions: 33
Owner: Francis
BUY THIS CONTRACT
Donald Trump
Price: 16.4801 Transactions: 30
Owner: kp
BUY THIS CONTRACT
Angelina Jolie
Price: 13.47069 Transactions: 30
Owner: mfs7772
BUY THIS CONTRACT
Elon Musk
Kanye West
John Mcafee
Adele
Note: Cryptocelebrities is in beta & under media embargo, please DO NOT publish stories about us before Jan 31 | Important beta note, read me!

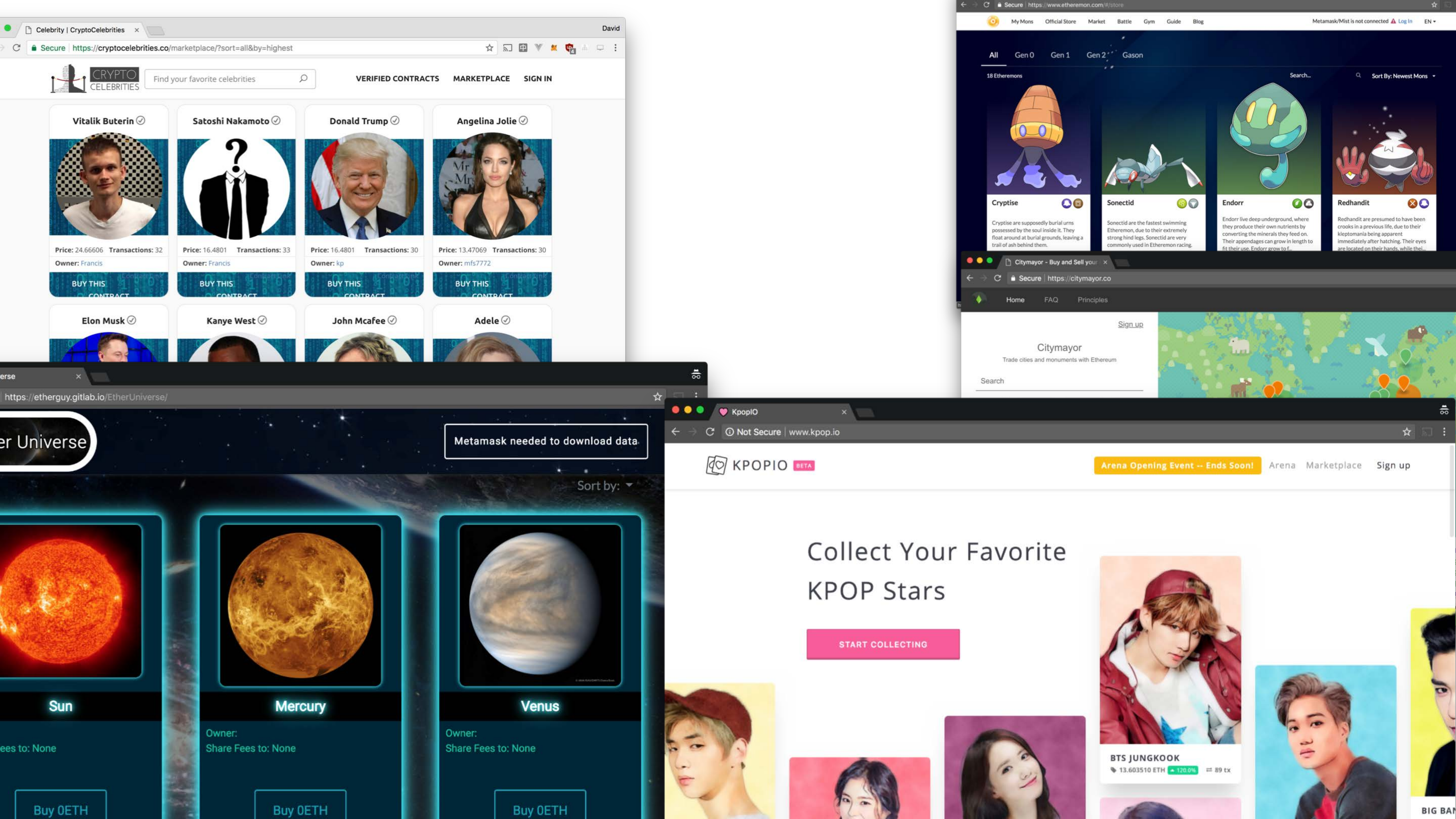

Secure https://www.etheremon.com/#/store
My Mons
Official Store
Market
Battle
Gym
Guide
Blog
Metamask/Mist is not connected
Log In
EN
All
Gen 0
Gen 1
Gen 2
Gason
18 Etheremons
Search...
Sort By: Newest Mons
Cryptise
Cryptise are supposedly burial urns possessed by the soul inside it. They float around at burial grounds, leaving a trail of ash behind them.
0.07161eth
Gen 2
Sonectid
Sonectid are the fastest swimming Etheremon, due to their extremely strong hind legs. Sonectid are very commonly used in Etheremon racing.
0.09144eth
Gen 2
Endorr
Endorr live deep underground, where they produce their own nutrients by converting the minerals they feed on. Their appendages can grow in length to fit their use. Endorr grow to f...
0.0508eth
Gen 2
Redhandit
Redhandit are presumed to have been crooks in a previous life, due to their kleptomania being apparent immediately after hatching. Their eyes are located on their hands, while thei...
0.09135eth
Gen 2


Celebrity | CryptoCelebrities
David
Secure https://cryptocelebrities.co/marketplace/?sort=all&by=highest
CRYPTO CELEBRITIES
Find your favorite celebrities
VERIFIED CONTRACTS
MARKETPLACE
SIGN IN
Vitalik Buterin
Price: 24.66606 Transactions: 32
Owner: Francis
BUY THIS CONTRACT
Satoshi Nakamoto
Price: 16.4801 Transactions: 33
Owner: Francis
BUY THIS CONTRACT
Donald Trump
Price: 16.4801 Transactions: 30
Owner: kp
BUY THIS CONTRACT
Angelina Jolie
Price: 13.47069 Transactions: 30
Owner: mfs7772
BUY THIS CONTRACT
Elon Musk
Kanye West
John Mcafee
Adele
Note: Cryptocelebrities is in beta & under media embargo, please DO NOT publish stories about us before Jan 31 | Important beta note, read me!

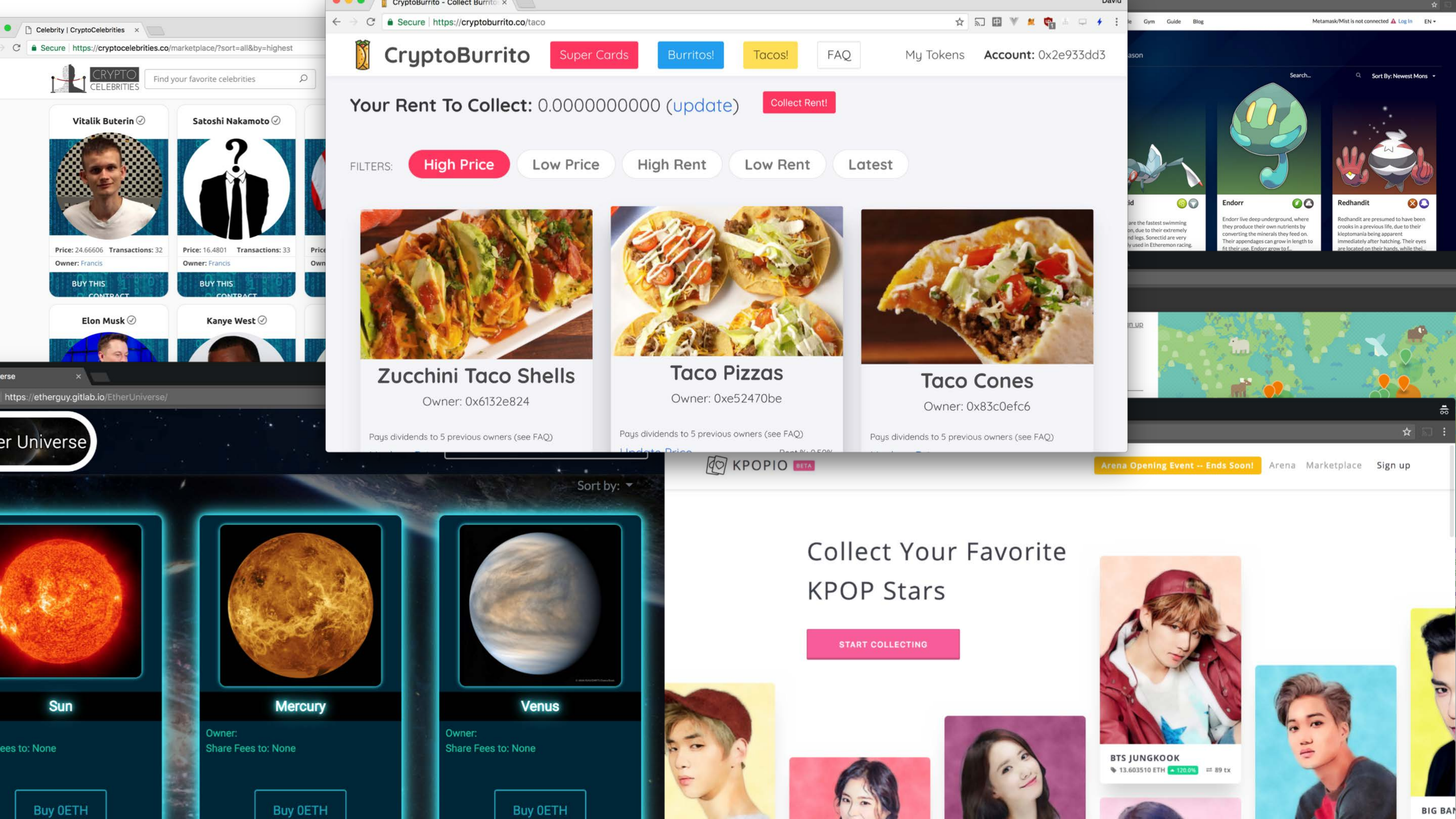

Secure https://www.etheremon.com/#/store
My Mons
Official Store
Market
Battle
Gym
Guide
Blog
Metamask/Mist is not connected
Log In
EN
All
Gen 0
Gen 1
Gen 2
Gason
18 Etheremons
Search...
Sort By: Newest Mons
Cryptise
Sonectid
Endorr
Redhandit
Citymayor - Buy and Sell your
Secure https://citymayor.co
Home
FAQ
Principles
Sign up
Citymayor
Trade cities and monuments with Ethereum
Search
All
New
Popular
Cheap
Santiago
Economy: 0 CITYs
Price: 0.01 ETH
Buy
Singapore
Economy: 0 CITYs
Price: 0.01 ETH
Buy
Johannesburg
Economy: 0 CITYs
Price: 0.01 ETH
Buy
Cape Town
https://citymayor.co/city/4

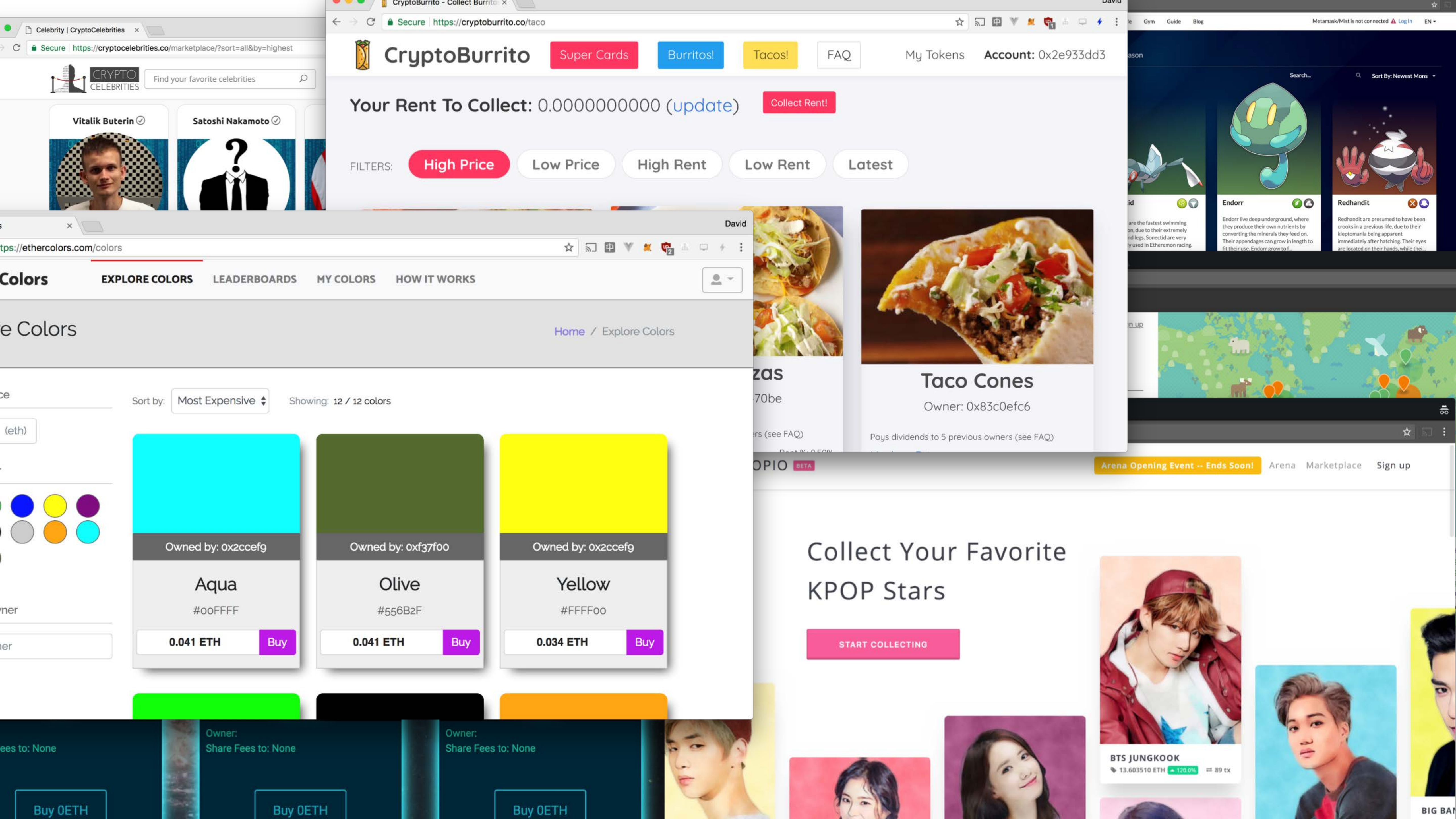


Celebrity | CryptoCelebrities
David
Secure https://cryptocelebrities.co/marketplace/?sort=all&by=highest
CRYPTO CELEBRITIES
Find your favorite celebrities
VERIFIED CONTRACTS
MARKETPLACE
SIGN IN
Vitalik Buterin
Price: 24.66606 Transactions: 32
Owner: Francis
BUY THIS CONTRACT
Satoshi Nakamoto
Price: 16.4801 Transactions: 33
Owner: Francis
BUY THIS CONTRACT
Donald Trump
Price: 16.4801 Transactions: 30
Owner: kp
BUY THIS CONTRACT
Angelina Jolie
Price: 13.47069 Transactions: 30
Owner: mfs7772
BUY THIS CONTRACT
Elon Musk
Kanye West
John Mcafee
Adele
https://etherguy.gitlab.io/EtherUniverse/
er Universe
Metamask needed to download data
Sort by:
Sun
ees to: None
Buy 0ETH
Mercury
Owner:
Share Fees to: None
Buy 0ETH
Venus
Owner:
Share Fees to: None
Buy 0ETH
Secure https://www.etheremon.com/#/store
My Mons
Official Store
Market
Battle
Gym
Guide
Blog
Metamask/Mist is not connected
Log In
EN
All
Gen 0
Gen 1
Gen 2
Gason
18 Etheremons
Search...
Sort By: Newest Mons
Cryptise
Cryptise are supposedly burial urns possessed by the soul inside it. They float around at burial grounds, leaving a trail of ash behind them.
Sonectid
Sonectid are the fastest swimming Etheremon, due to their extremely strong hind legs. Sonectid are very commonly used in Etheremon racing.
Endorr
Endorr live deep underground, where they produce their own nutrients by converting the minerals they feed on. Their appendages can grow in length to fit their use. Endorr grow to f...
Redhandit
Redhandit are presumed to have been crooks in a previous life, due to their kleptomania being apparent immediately after hatching. Their eyes are located on their hands, while thei...
Citymayor - Buy and Sell your
Secure https://citymayor.co
Home
FAQ
Principles
Sign up
Citymayor
Trade cities and monuments with Ethereum
Search
All
New
Popular
Cheap
Santiago
Economy: 0 CITYs
Price: 0.01 ETH
Buy
Singapore
Economy: 0 CITYs
Price: 0.01 ETH
Buy
Johannesburg
Economy: 0 CITYs
Price: 0.01 ETH
Buy
Cape Town
https://citymayor.co/city/4
mapbox

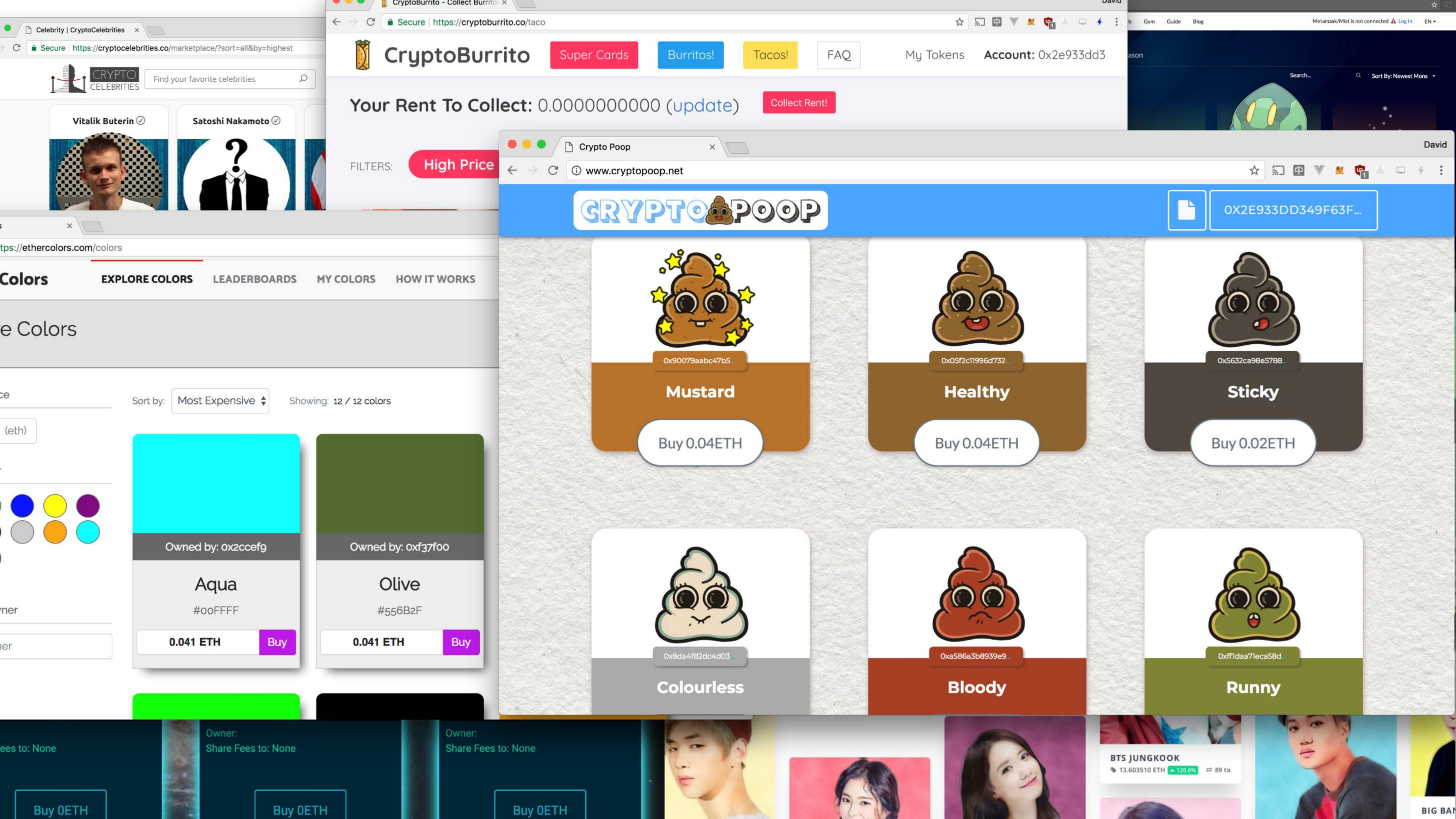


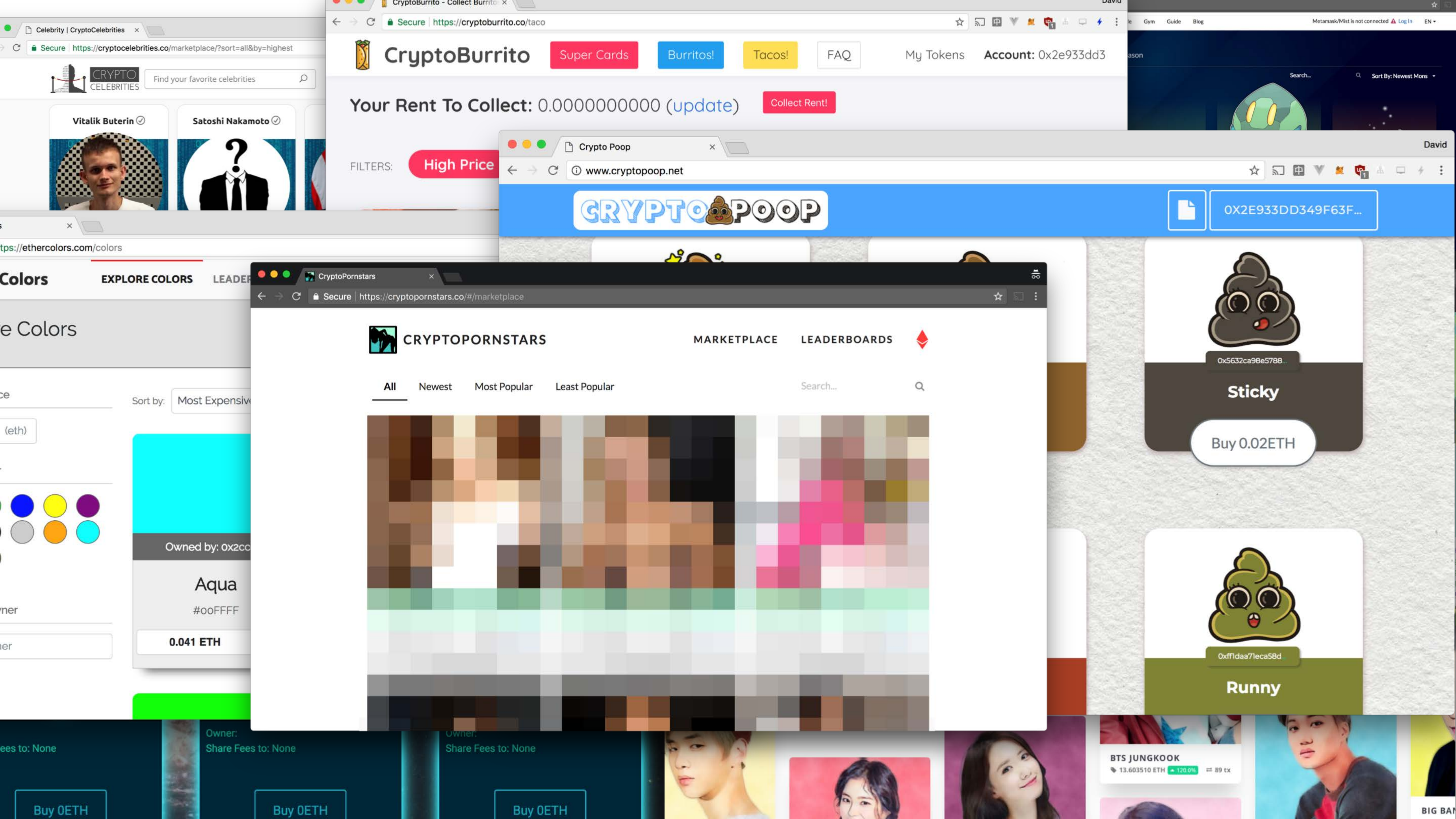
Celebrity | CryptoCelebrities
Secure https://cryptocelebrities.co/marketplace/?sort=all&by=highest
CRYPTO CELEBRITIES
Find your favorite celebrities
VERIFIED CONTRACTS
MARKETPLACE
SIGN IN
Vitalik Buterin
Price: 24.66606 Transactions: 32
Owner: Francis
Satoshi Nakamoto
Price: 16.4801 Transactions: 33
Owner: Francis
Donald Trump
Price: 16.4801 Transactions: 30
Owner: kp
Angelina Jolie
Price: 13.47069 Transactions: 30
Owner: mfs7772
BUY THIS CONTRACT
Elon Musk
Kanye West
John Mcafee
Adele
Secure https://www.etheremon.com/#/store
My Mons
Official Store
Market
Battle
Gym
Guide
Blog
Metamask/Mist is not connected
Log In
All
Gen 0
Gen 1
Gen 2
Gason
18 Etheremons
Sort By: Newest Mons
Cryptise
Cryptise are supposedly burial urns possessed by the soul inside it. They float around at burial grounds, leaving a trail of ash behind them.
Sonectid
Sonectid are the fastest swimming Etheremon, due to their extremely strong hind legs. Sonectid are very commonly used in Etheremon racing.
Endorr
Redhandit
Citymayor - Buy and Sell your
Secure https://citymayor.co
Home
FAQ
Principles
Sign up
Citymayor
Trade cities and monuments with Ethereum
Search
https://etherguy.gitlab.io/EtherUniverse/
er Universe
Metamask needed to download data.
Sort by:
Sun
ees to: None
Buy 0ETH
Mercury
Owner:
Share Fees to: None
Buy 0ETH
Venus
Owner:
Share Fees to: None
Buy 0ETH
KpopIO
Not Secure www.kpop.io
KPOPIO BETA
Arena Opening Event -- Ends Soon!
Arena
Marketplace
Sign up
Collect Your Favorite KPOP Stars
START COLLECTING
BTS JUNGKOOK
13.603510 ETH
120.0%
89 tx


Celebrity | CryptoCelebrities
Secure https://cryptocelebrities.co/marketplace/?sort=all&by=highest
CRYPTO CELEBRITIES
Find your favorite celebrities
Vitalik Buterin
Price: 24.66606 Transactions: 32
Owner: Francis
BUY THIS CONTRACT
Satoshi Nakamoto
Price: 16.4801 Transactions: 33
Owner: Francis
BUY THIS CONTRACT
Elon Musk
Kanye West
CryptoBurrito - Collect Burrito
Secure https://cryptoburrito.co/taco
CryptoBurrito
Super Cards
Burritos!
Tacos!
FAQ
My Tokens
Account: 0x2e933dd3
Your Rent To Collect: 0.0000000000 (update)
Collect Rent!
FILTERS:
High Price
Low Price
High Rent
Low Rent
Latest
Zucchini Taco Shells
Owner: 0x6132e824
Pays dividends to 5 previous owners (see FAQ)
Taco Pizzas
Owner: 0xe52470be
Pays dividends to 5 previous owners (see FAQ)
Taco Cones
Owner: 0x83c0efc6
Pays dividends to 5 previous owners (see FAQ)
Gym
Guide
Blog
Metamask/Mist is not connected
Log In
EN
Search...
Sort By: Newest Mons
Endorr
Endorr live deep underground, where they produce their own nutrients by converting the minerals they feed on. Their appendages can grow in length to fit their use. Endorr grow to f...
Redhandit
Redhandit are presumed to have been crooks in a previous life, due to their kleptomania being apparent immediately after hatching. Their eyes are located on their hands, while thei...
https://etherguy.gitlab.io/EtherUniverse/
er Universe
Sort by:
Sun
ees to: None
Buy 0ETH
Mercury
Owner:
Share Fees to: None
Buy 0ETH
Venus
Owner:
Share Fees to: None
Buy 0ETH
KPOPIO BETA
Arena Opening Event -- Ends Soon!
Arena
Marketplace
Sign up
Collect Your Favorite
KPOP Stars
START COLLECTING
BTS JUNGKOOK
13.603510 ETH
120.0%
89 tx



Celebrity | CryptoCelebrities
https://cryptocelebrities.co/marketplace/?sort=all&by=highest
CRYPTO CELEBRITIES
Find your favorite celebrities
Vitalik Buterin
Satoshi Nakamoto
CryptoBurrito - Collect Burrito
https://cryptoburrito.co/taco
CryptoBurrito
Super Cards
Burritos!
Tacos!
FAQ
My Tokens
Account: 0x2e933dd3
Your Rent To Collect: 0.0000000000 (update)
Collect Rent!
FILTERS:
High Price
Low Price
High Rent
Low Rent
Latest
Taco Cones
Owner: 0x83c0efc6
Pays dividends to 5 previous owners (see FAQ)
Gym
Guide
Blog
Metamask/Mist is not connected
Log In
Sort By: Newest Mons
Endorr
Redhandit
https://ethercolors.com/colors
David
Colors
EXPLORE COLORS
LEADERBOARDS
MY COLORS
HOW IT WORKS
Home / Explore Colors
Sort by:
Most Expensive
Showing: 12 / 12 colors
Owned by: 0x2ccef9
Aqua
#00FFFF
0.041 ETH
Buy
Owned by: 0xf37f00
Olive
#556B2F
0.041 ETH
Buy
Owned by: 0x2ccef9
Yellow
#FFFF00
0.034 ETH
Buy
Owner:
Share Fees to: None
Buy 0ETH
Buy 0ETH
Buy 0ETH
BETA
Arena Opening Event -- Ends Soon!
Arena
Marketplace
Sign up
Collect Your Favorite
KPOP Stars
START COLLECTING
BTS JUNGKOOK
13.603510 ETH
120.0%
89 tx


Celebrity | CryptoCelebrities
Secure https://cryptocelebrities.co/marketplace/?sort=all&by=highest
CRYPTO CELEBRITIES
Find your favorite celebrities
Vitalik Buterin
Satoshi Nakamoto
CryptoBurrito - Collect Burrito
Secure https://cryptoburrito.co/taco
CryptoBurrito
Super Cards
Burritos!
Tacos!
FAQ
My Tokens
Account: 0x2e933dd3
Your Rent To Collect: 0.0000000000 (update)
Collect Rent!
FILTERS:
High Price
Crypto Poop
www.cryptopoop.net
CRYPTO POOP
0X2E933DD349F63F...
0x90079aabc47b5
Mustard
Buy 0.04ETH
0x05f2c11996d732
Healthy
Buy 0.04ETH
0x5632ca98e5788
Sticky
Buy 0.02ETH
0x8da4f82dc4d03
Colourless
0xa586a3b8939e9
Bloody
0xff1daa71eca58d
Runny
ethercolors.com/colors
Colors
EXPLORE COLORS
LEADERBOARDS
MY COLORS
HOW IT WORKS
Sort by: Most Expensive
Showing: 12 / 12 colors
Owned by: 0x2ccef9
Aqua
#00FFFF
0.041 ETH
Buy
Owned by: 0xf37f00
Olive
#556B2F
0.041 ETH
Buy
Owner:
Share Fees to: None
Buy 0ETH
Owner:
Share Fees to: None
Buy 0ETH
Buy 0ETH
BTS JUNGKOOK
13.603510 ETH
120.0%
89 tx
Metamask/Mist is not connected
Log In
Sort By: Newest Mons
David



Celebrity | CryptoCelebrities
Secure https://cryptocelebrities.co/marketplace/?sort=all&by=highest
CRYPTO CELEBRITIES
Find your favorite celebrities
Vitalik Buterin
Satoshi Nakamoto
CryptoBurrito - Collect Burrito
Secure https://cryptoburrito.co/taco
CryptoBurrito
Super Cards
Burritos!
Tacos!
FAQ
My Tokens
Account: 0x2e933dd3
Your Rent To Collect: 0.0000000000 (update)
Collect Rent!
FILTERS:
High Price
Metamask/Mist is not connected
Log In
Gym
Guide
Blog
Sort By: Newest Mons
Crypto Poop
David
www.cryptopoop.net
CRYPTOPOOP
0X2E933DD349F63F...
0x5632ca98e5788...
Sticky
Buy 0.02ETH
0xff1daa71eca58d
Runny
tps://ethercolors.com/colors
Colors
EXPLORE COLORS
LEADER
e Colors
Sort by:
Most Expensiv
(eth)
Owned by: 0x2cc
Aqua
#00FFFF
0.041 ETH
CryptoPornstars
Secure https://cryptopornstars.co/#/marketplace
CRYPTOPORNSTARS
MARKETPLACE
LEADERBOARDS
All
Newest
Most Popular
Least Popular
Search...
ees to: None
Owner:
Share Fees to: None
Buy 0ETH
BTS JUNGKOOK
13.603510 ETH
120.0%
89 tx
BIG BAN

watch the statistics

The DAO has been created

968.42 M

DAO TOKENS CREATED

9.68 M

TOTAL ETH

97.81 M

USD EQUIVALENT

1.00

LAST EXCHANGE RATE
ETH / 100 DAO TOKENS

22 hours

NEXT PRICE PHASE

13 days

SINCE CREATION PERIOD ENDED
CREATED 28 MAY 09:00 GMT

Thank you all for your contribution

The DAO Raises More Than $1 ×

David

Secure https://bitcoinmagazine.com/articles/the-dao-raises-more-than-million-in-world-s-largest-crowdfunding-to-date-1463422...

/ ETHEREUM

by **Giulio Prisco**

May 16, 2016 2:09 PM EST

Pending on po.et

What is Po.et?

Tweet

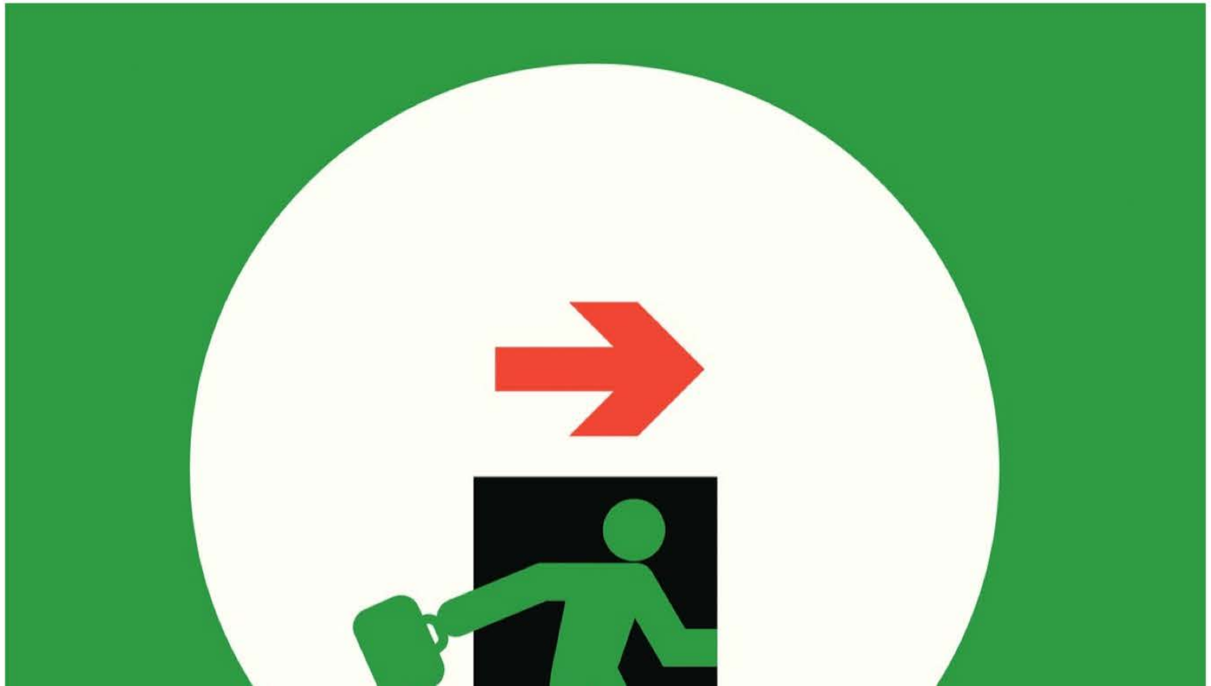
The DAO Raises More Than $117 Million in World's Largest Crowdfunding to Date

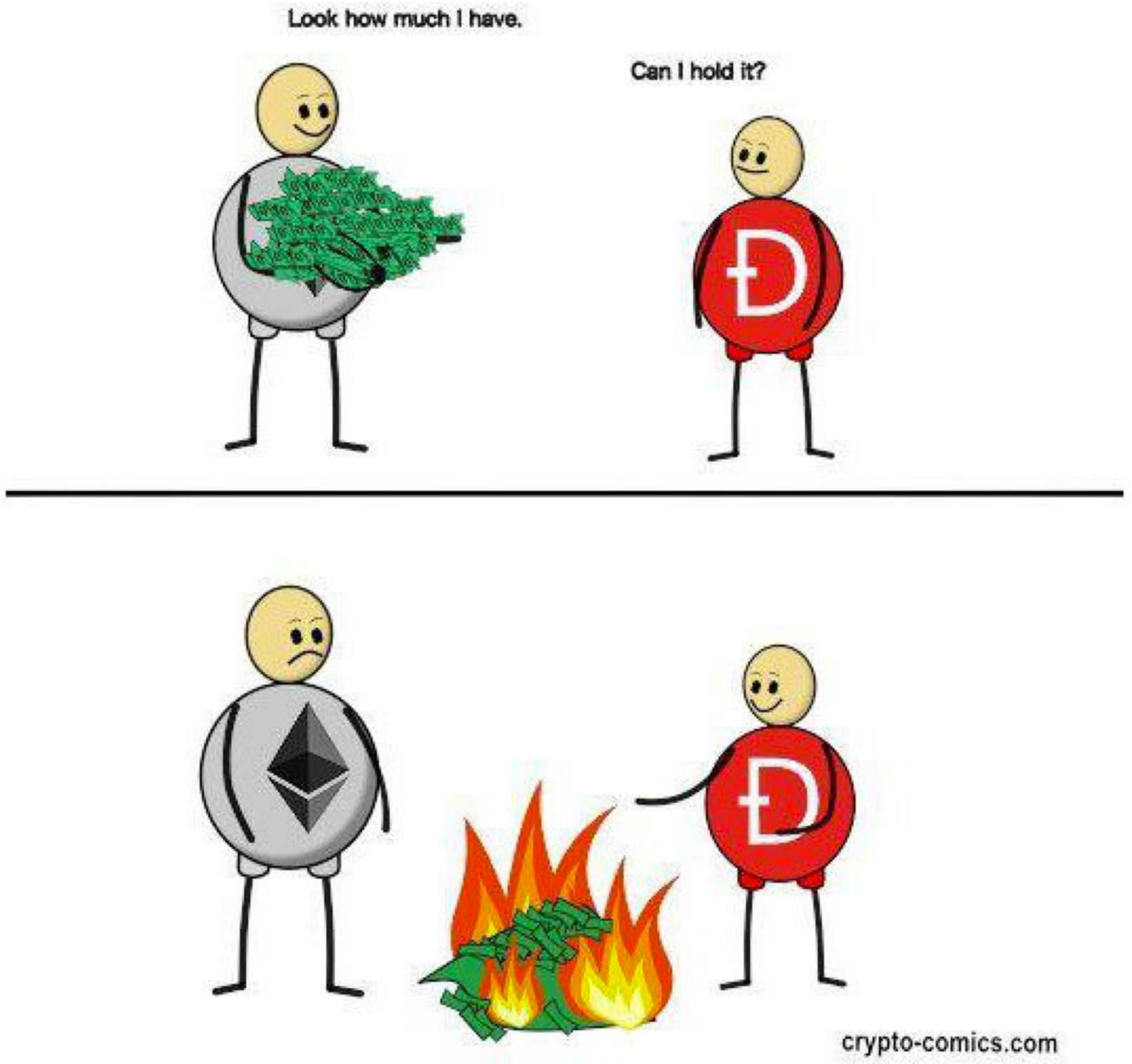
Trending Now

- South Korea Allows Cryptocurrency Trading for Real-Name Registered Accounts
- Edge's Paul Puey: "Digital Security Will Take Place on the Edges"
- Physical Bitcoins: Our Hands-On, End-to-End Review of Opendime
- Driving Blockchain Adoption in the Developing World With Spire
- Study Suggests 25 Percent of Bitcoin Users Are Associated With Illegal Activity

KLINT FINLEY BUSINESS 06.18.16 04:30 AM

A $50 MILLION HACK JUST SHOWED THAT THE DAO WAS ALL TOO HUMAN




Look how much I have.
Can I hold it?
crypto-comics.com

```
pragma solidity ^0.4.20;

contract Reentrance {

    mapping (address => uint256) public balances;

}
```

```
pragma solidity ^0.4.20;

contract Reentrance {

    mapping (address => uint256) public balances;

    function donate() public payable {
        balances[msg.sender] += msg.value;
    }

}
```

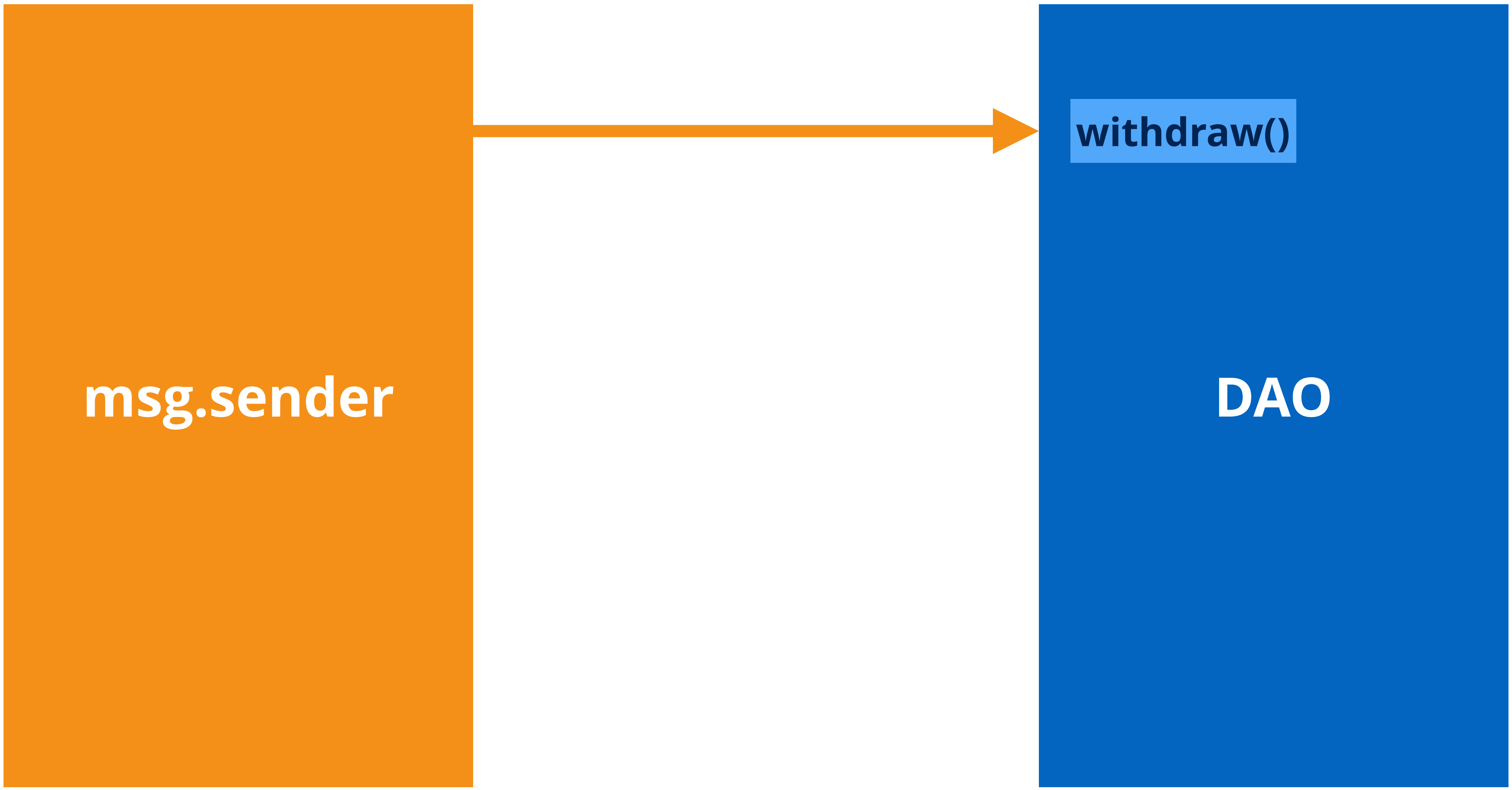
```
pragma solidity ^0.4.20;

contract Reentrance {

    mapping (address => uint256) public balances;

    function donate() public payable {
        balances[msg.sender] += msg.value;
    }

    function withdraw(uint256 amount) public {
        require(balances[msg.sender] >= amount);
        msg.sender.call.value(amount)();
        balances[msg.sender] -= amount;
    }

}
```

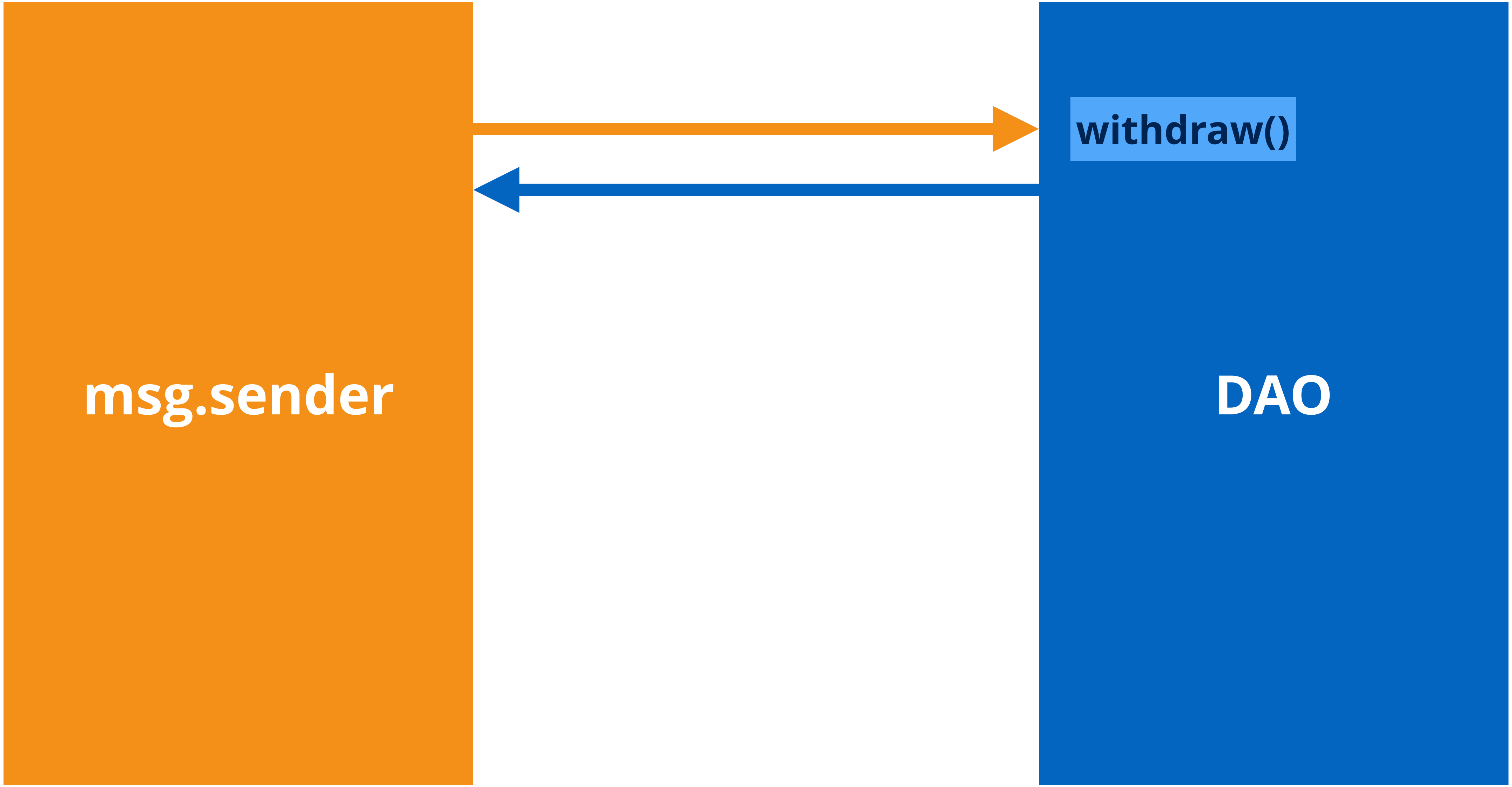
```solidity
pragma solidity ^0.4.20;

contract Reentrance {

    mapping (address => uint256) public balances;

    function donate() public payable {
        balances[msg.sender] += msg.value;
    }

    function withdraw(uint256 amount) public {
        require(balances[msg.sender] >= amount);
        msg.sender.call.value(amount)();
        balances[msg.sender] -= amount;
    }

}
```

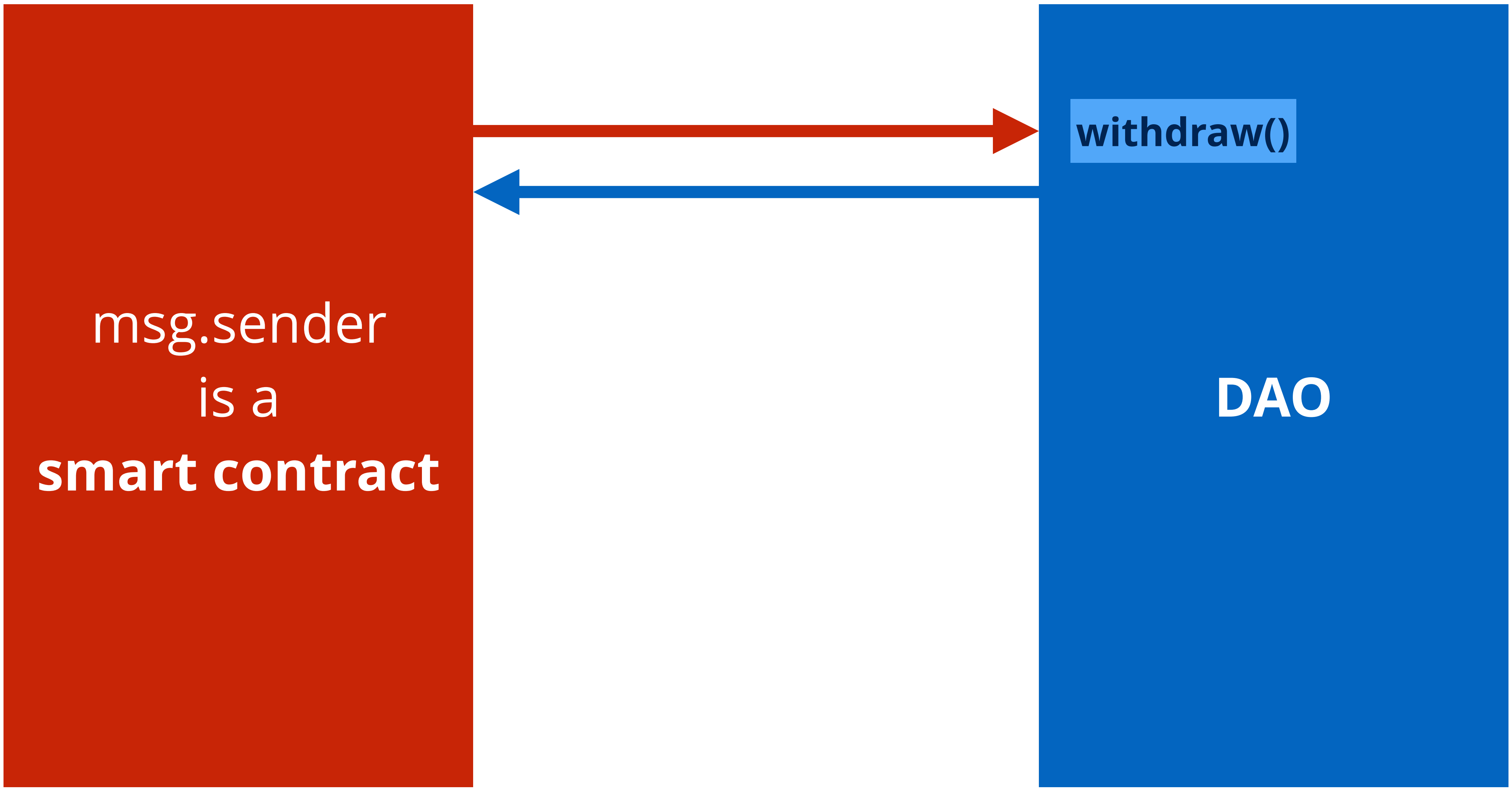
```
pragma solidity ^0.4.20;

contract Reentrance {

    mapping (address => uint256) public balances;

    function donate() public payable {
        balances[msg.sender] += msg.value;
    }

    function withdraw(uint256 amount) public {
        require(balances[msg.sender] >= amount);
        msg.sender.call.value(amount)();
        balances[msg.sender] -= amount;
    }

}
```

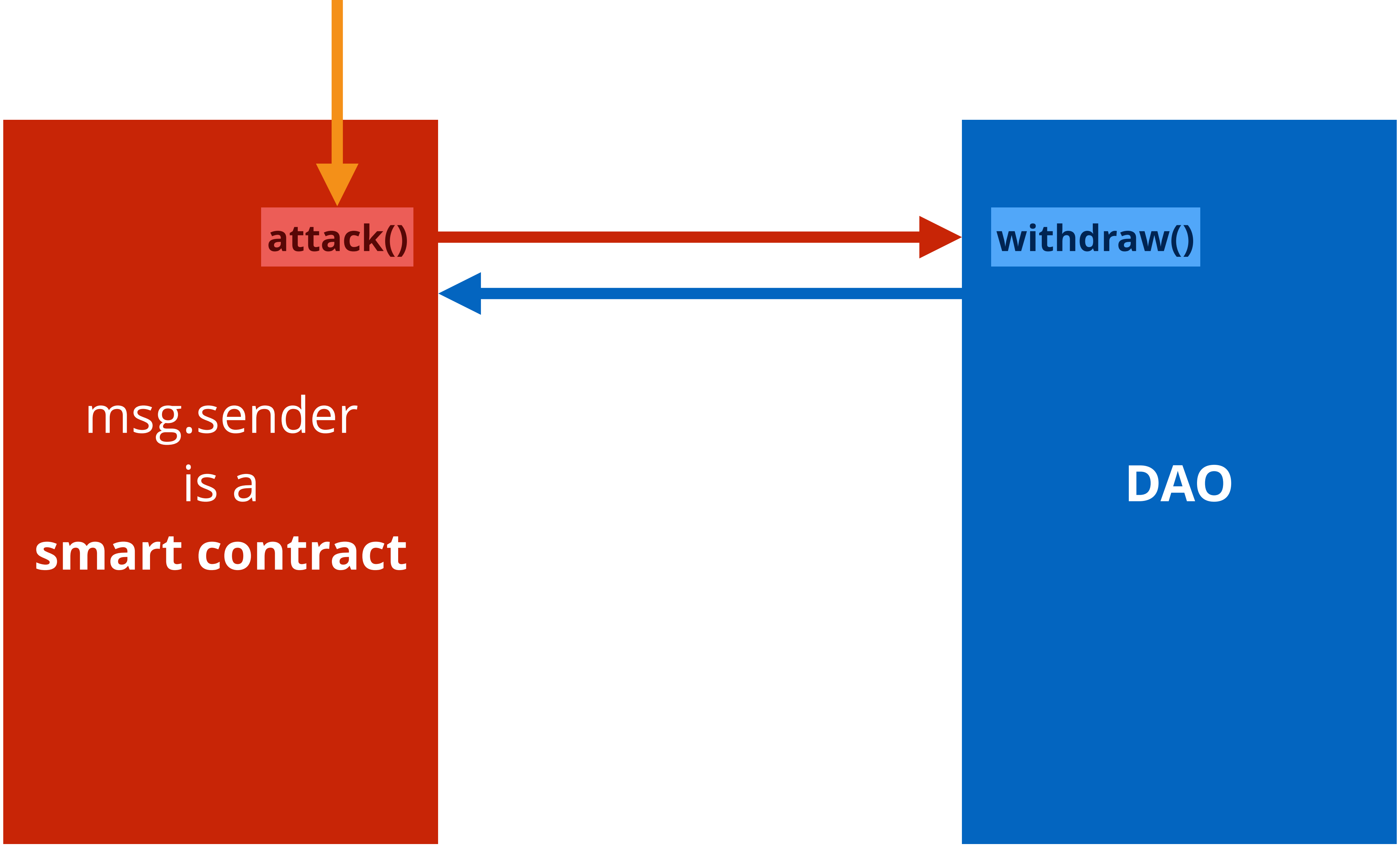
```
pragma solidity ^0.4.20;

contract Reentrance {

    mapping (address => uint256) public balances;

    function donate() public payable {
        balances[msg.sender] += msg.value;
    }

    function withdraw(uint256 amount) public {
        require(balances[msg.sender] >= amount);
        msg.sender.call.value(amount)();
        balances[msg.sender] -= amount;
    }

}
```

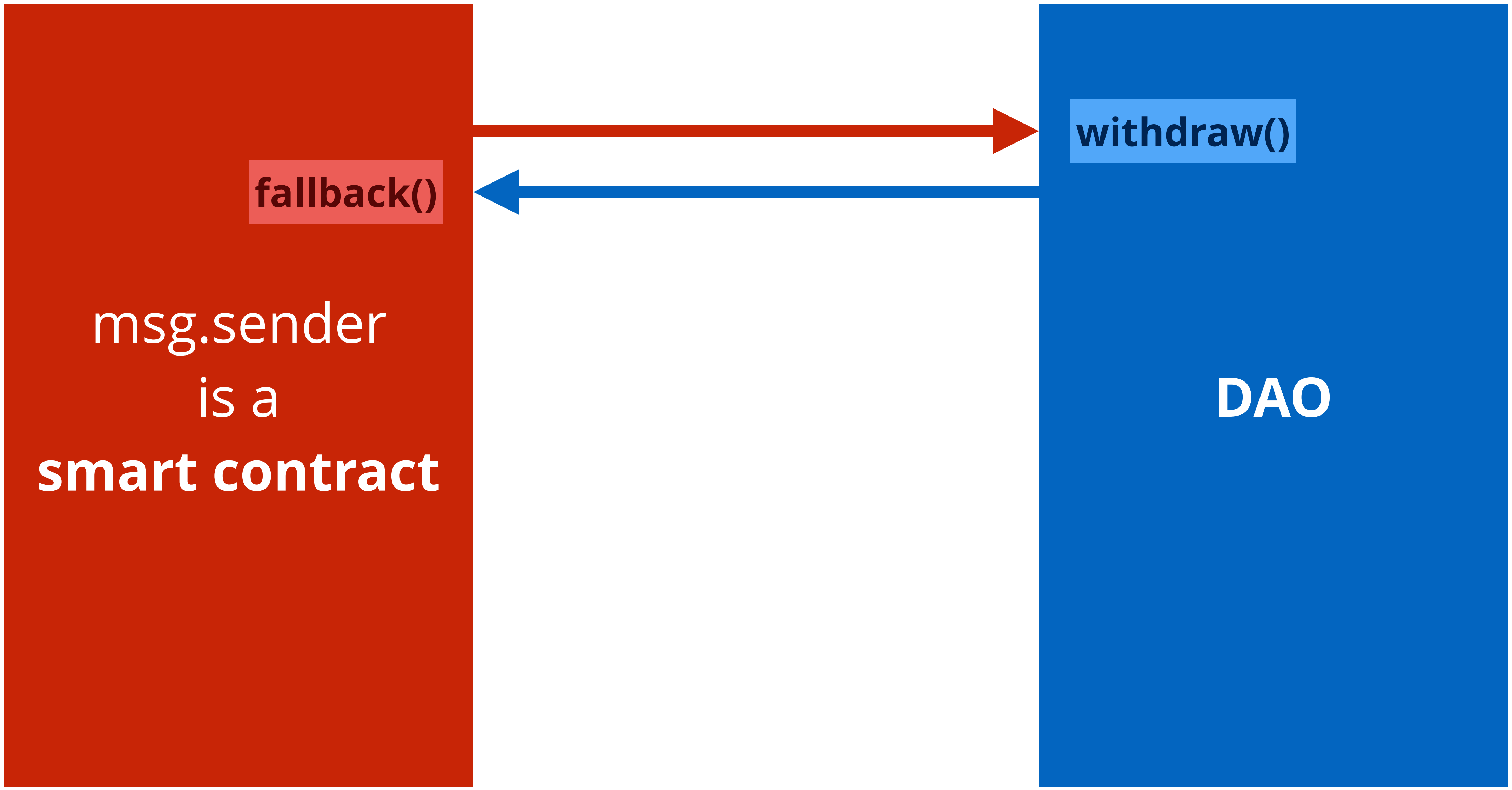
```
pragma solidity ^0.4.20;

contract Reentrance {

    mapping (address => uint256) public balances;

    function donate() public payable {
        balances[msg.sender] += msg.value;
    }

    function withdraw(uint256 amount) public {
        require(balances[msg.sender] >= amount);
        msg.sender.call.value(amount)();
        balances[msg.sender] -= amount;
    }

}
```

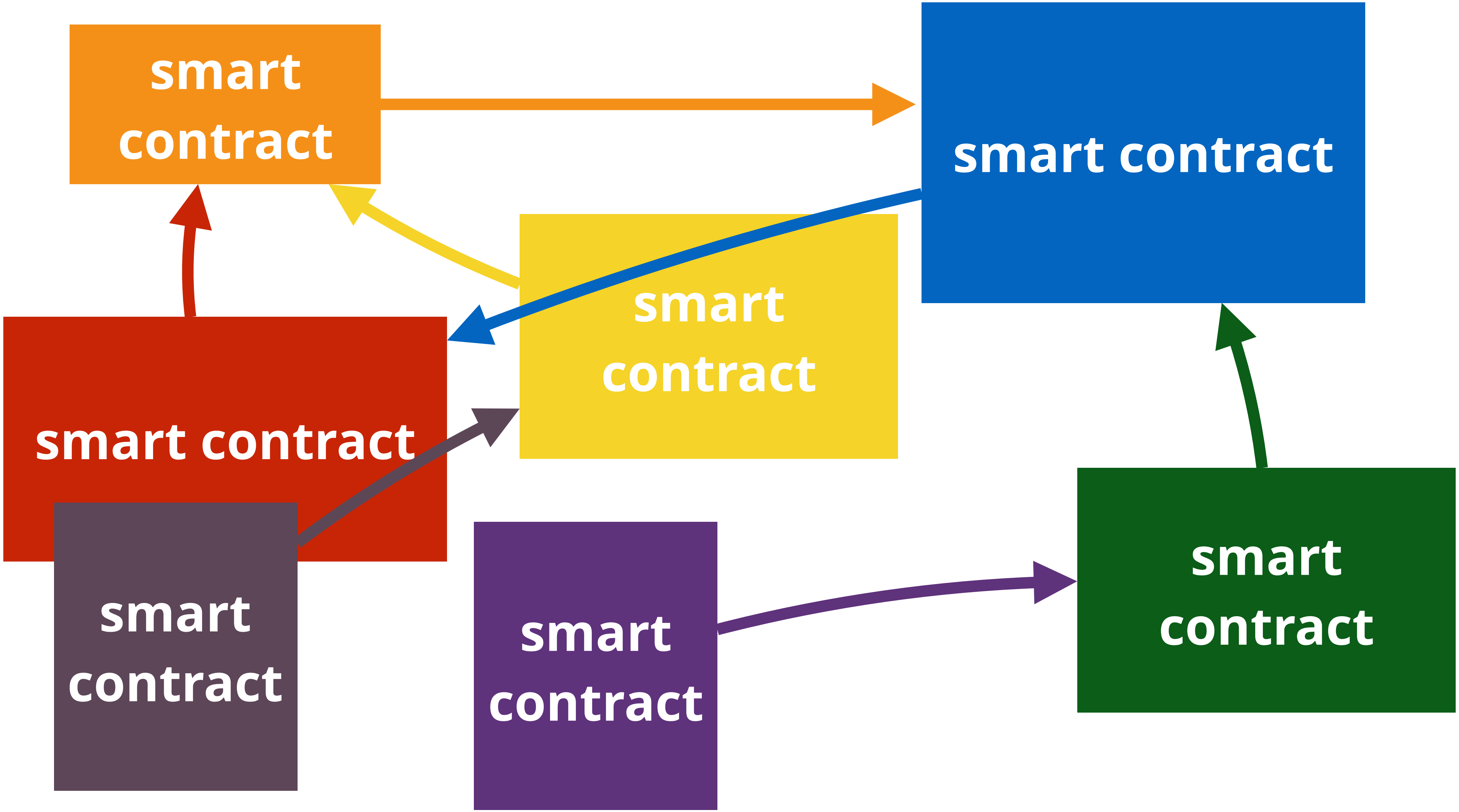

msg.sender
withdraw()
DAO


withdraw()
msg.sender
DAO

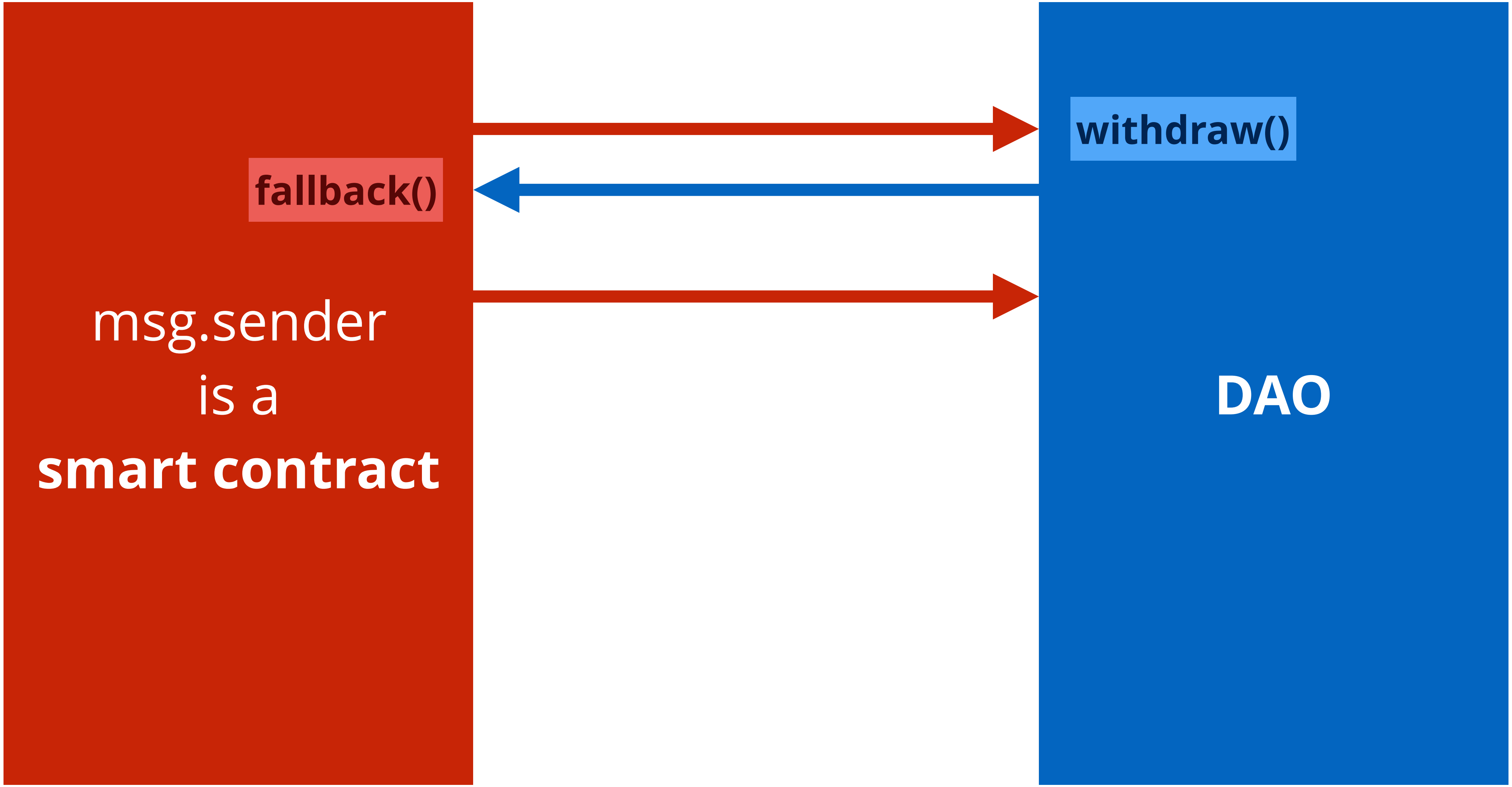

msg.sender
is a
smart contract
withdraw()
DAO


attack()
withdraw()
msg.sender
is a
smart contract
DAO

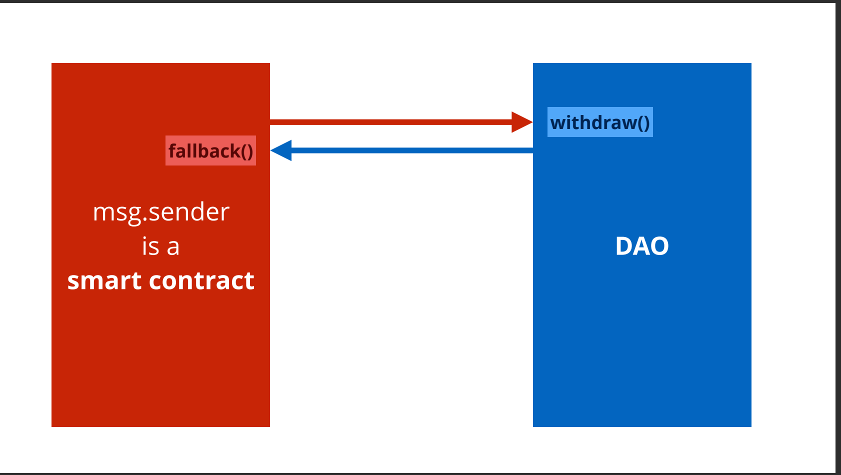

fallback()
msg.sender
is a
smart contract
withdraw()
DAO



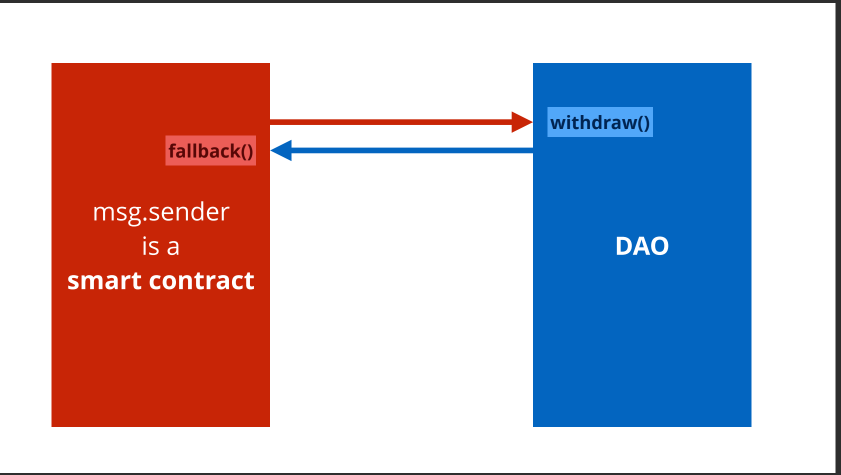
smart contract
smart contract
smart contract
smart contract
smart contract
smart contract
smart contract


fallback()
msg.sender
is a
smart contract
withdraw()
DAO

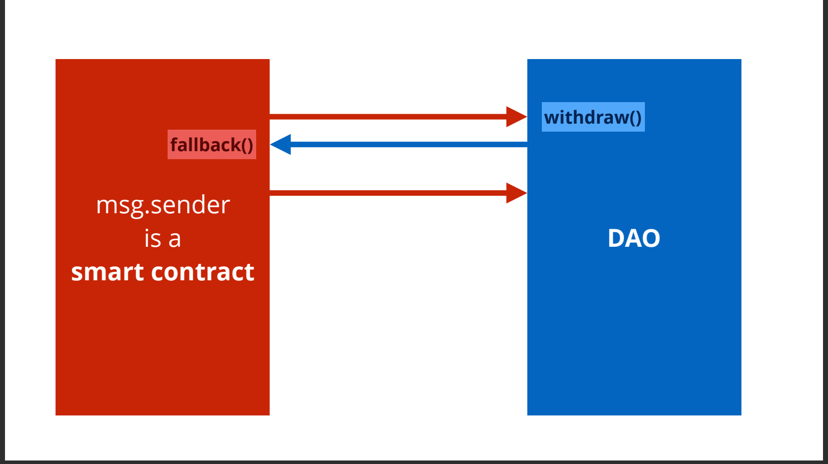
```
pragma solidity ^0.4.20;

contract Reentrance {

    mapping (address => uint256) public balances;

    function donate() public payable {
        balances[msg.sender] += msg.value;
    }

    function withdraw(uint256 amount) public {
        require(balances[msg.sender] >= amount);
        msg.sender.call.value(amount)();
        balances[msg.sender] -= amount;
    }

}
```



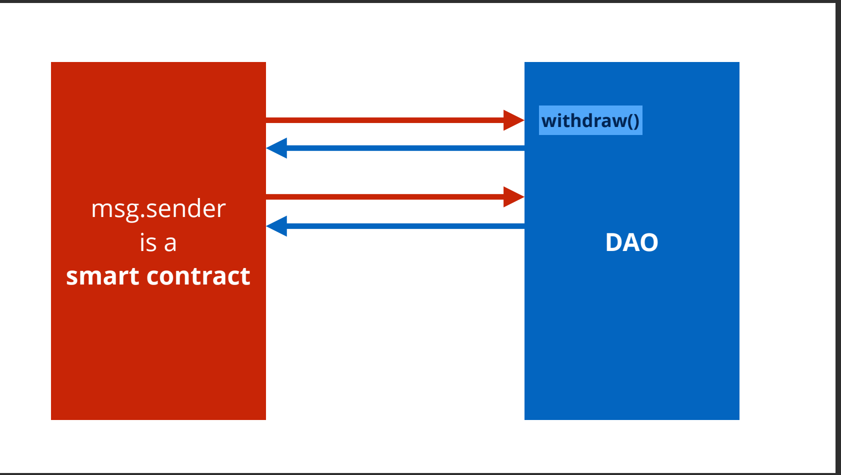
```
pragma solidity ^0.4.20;

contract Reentrance {

    mapping (address => uint256) public balances;

    function donate() public payable {
        balances[msg.sender] += msg.value;
    }

    function withdraw(uint256 amount) public {
        require(balances[msg.sender] >= amount);
        msg.sender.call.value(amount)();
        balances[msg.sender] -= amount;
    }

}
```

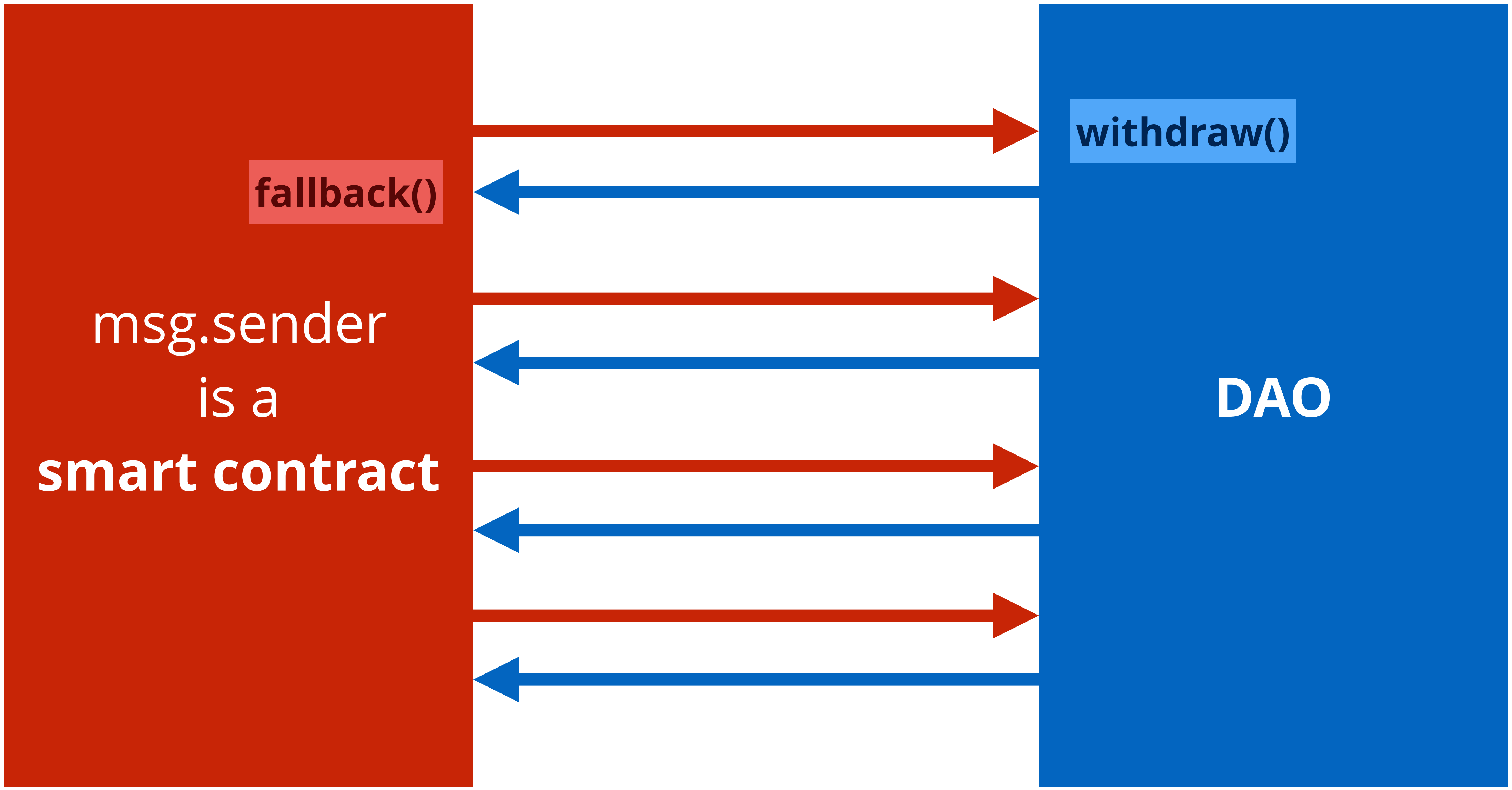


```
pragma solidity ^0.4.20;

contract Reentrance {

    mapping (address => uint256) public balances;

    function donate() public payable {
        balances[msg.sender] += msg.value;
    }

    function withdraw(uint256 amount) public {
        require(balances[msg.sender] >= amount);
        msg.sender.call.value(amount)();
        balances[msg.sender] -= amount;
    }

}
```



```
pragma solidity ^0.4.20;

contract Reentrance {

    mapping (address => uint256) public balances;

    function donate() public payable {
        balances[msg.sender] += msg.value;
    }

    function withdraw(uint256 amount) public {
        require(balances[msg.sender] >= amount);
        msg.sender.call.value(amount)();
        balances[msg.sender] -= amount;
    }

}
```




withdraw()
fallback()
msg.sender
is a
smart contract
DAO

Reentrancy

Race-to-empty

- Sending ether to an address might **lead to code execution**
- **Changing the state after calling another contract**

parity

The **fastest** and **most secure** way of interacting with the Ethereum blockchain. Our client powers much of the infrastructure of the public Ethereum network and is used by companies and users alike.

Wallet

- function **initWallet**(address[] _owners, …)
- function confirm(bytes32 _h)
- function kill(address _to)
- …

Wallet

- function **initWallet**(address[] _owners, …)
- function confirm(bytes32 _h)
- function kill(address _to)
- …

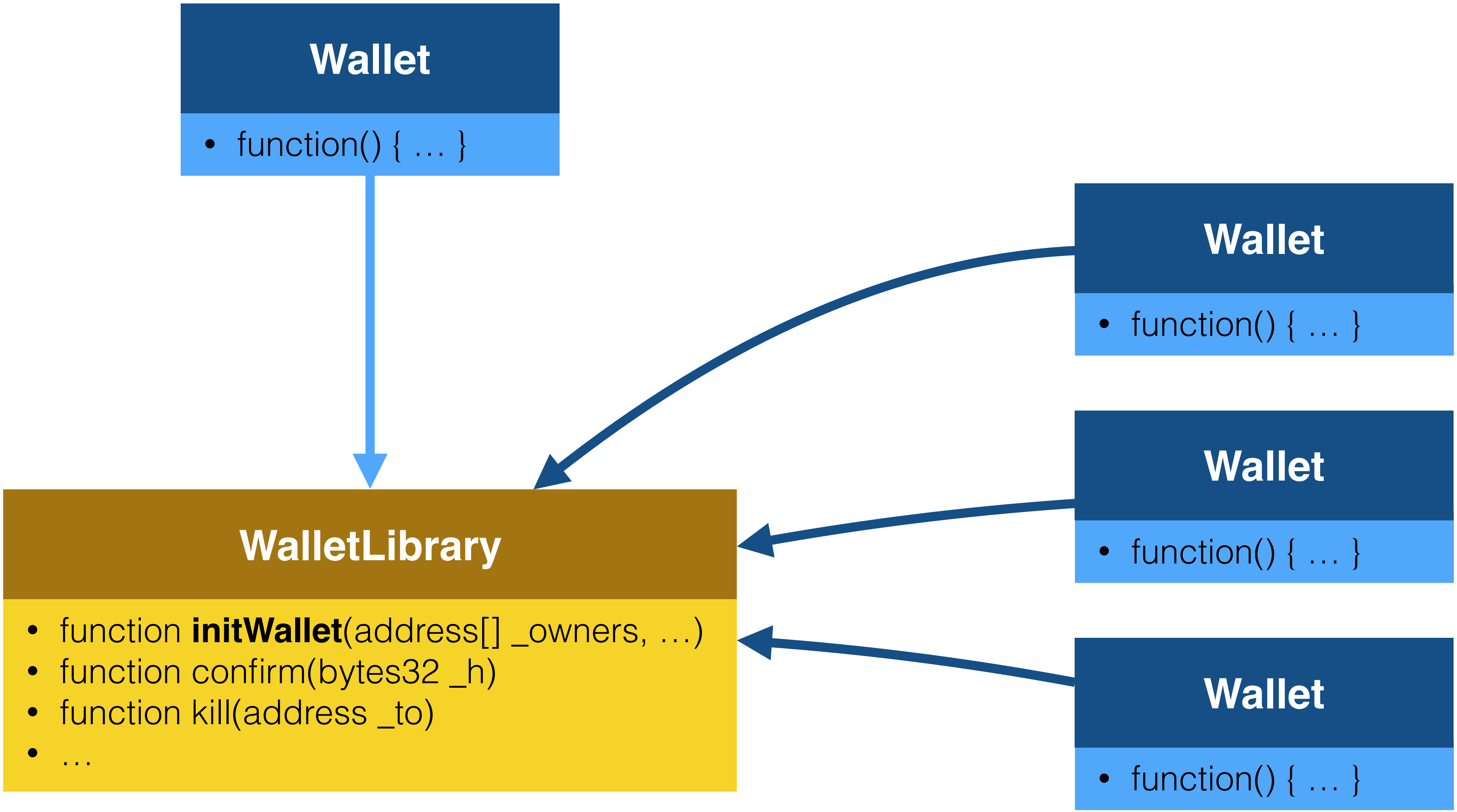
owners: David, Mason and Thomas
required: 2

Wallet

- function() { … }

WalletLibrary

- function **initWallet**(address[] _owners, …)
- function confirm(bytes32 _h)
- function kill(address _to)
- …


Wallet
• function() { … }
Wallet
• function() { … }
Wallet
• function() { … }
Wallet
• function() { … }
WalletLibrary
• function initWallet(address[] _owners, …)
• function confirm(bytes32 _h)
• function kill(address _to)
• …

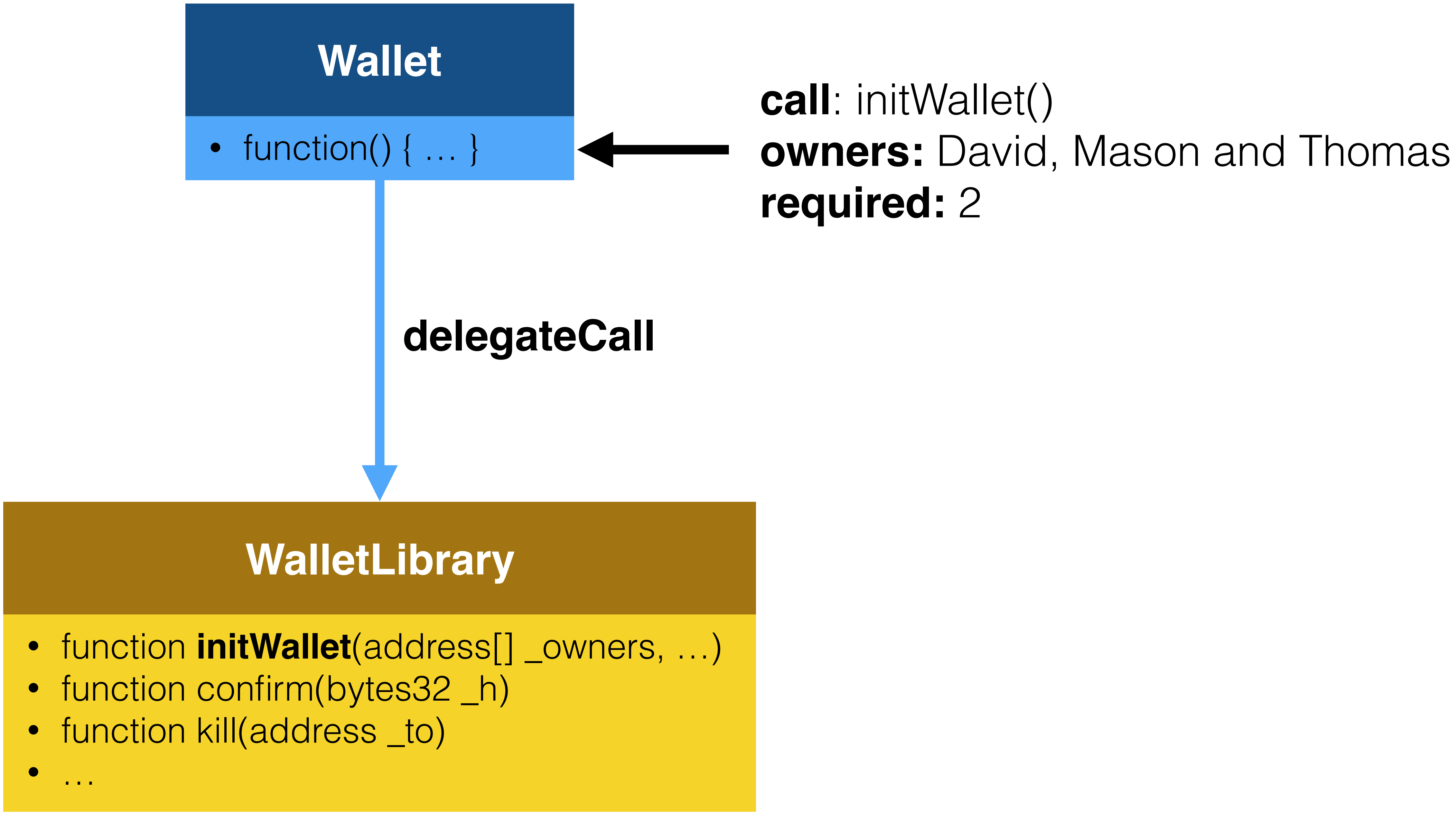
Wallet

- function() { … }

call: initWallet()
owners: David, Mason and Thomas
required: 2

WalletLibrary

- function **initWallet**(address[] _owners, …)
- function confirm(bytes32 _h)
- function kill(address _to)
- …


Wallet
• function() { … }
call: initWallet()
owners: David, Mason and Thomas
required: 2
delegateCall
WalletLibrary
• function initWallet(address[] _owners, …)
• function confirm(bytes32 _h)
• function kill(address _to)
• …

Wallet

- owners: David, Mason, Thomas
- function() { … }

WalletLibrary

- function **initWallet**(address[] _owners, …)
- function confirm(bytes32 _h)
- function kill(address _to)
- …

multisig-wallet-vuln.sol — contracts

OPEN FILES

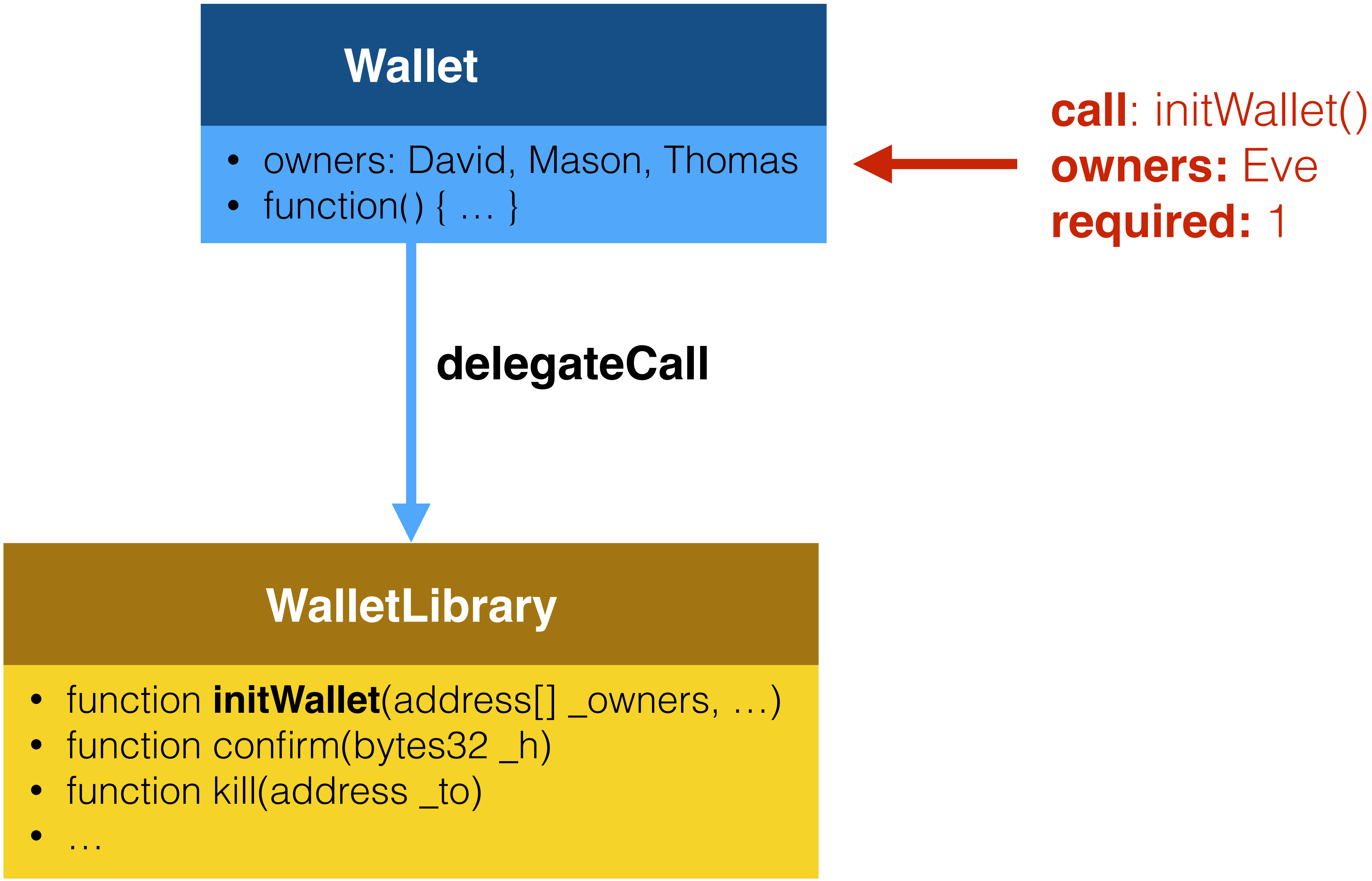
FOLDERS

- contracts
  - parity_multisig1
    - multisig-wallet-vuln.sol
  - parity_multisig2

multisig-wallet-vuln.sol

```
  function today() private constant returns (uint) { return now / 1 days; }

  function clearPending() internal {
    uint length = m_pendingIndex.length;

    for (uint i = 0; i < length; ++i) {
      delete m_txs[m_pendingIndex[i]];

      if (m_pendingIndex[i] != 0)
        delete m_pending[m_pendingIndex[i]];
    }

    delete m_pendingIndex;
  }

  // FIELDS
  address constant _walletLibrary = 0xa657491c1e7f16adb39b9b60e87bbb8d93988bc3;

  // the number of owners that must confirm the same operation before it is run.
  uint public m_required;
  // pointer used to find a free slot in m_owners
  uint public m_numOwners;

  uint public m_dailyLimit;
  uint public m_spentToday;
  uint public m_lastDay;

  // list of owners
  uint[256] m_owners;

  uint constant c_maxOwners = 250;
  // index on the list of owners to allow reverse lookup
  mapping(uint => uint) m_ownerIndex;
  // the ongoing operations.
  mapping(bytes32 => PendingState) m_pending;
```

W: 0 E: 0, Line 375, Column 27 Spaces: 2 Solidity



Wallet
owners: David, Mason, Thomas
function() { … }
call: initWallet()
owners: Eve
required: 1
delegateCall
WalletLibrary
function initWallet(address[] _owners, …)
function confirm(bytes32 _h)
function kill(address _to)
…

Wallet

- owners: **Eve**
- function() { … }

WalletLibrary

- function **initWallet**(address[] _owners, …)
- function confirm(bytes32 _h)
- function kill(address _to)
- …

$30 Million: Ether Reported Stolen Due to Parity Wallet Breach

Wolfie Zhao

Jul 19, 2017 at 21:44 UTC | Updated Jul 20, 2017 at 13:05 UTC

NEWS

Smart contract coding company Parity has issued a security alert, warning of a vulnerability in version 1.5 or later of its wallet software.

So far, 150,000 ethers, worth $30 million, have been reported by the company as stolen, data confirmed by Etherscan.io. As reported by the startup, the issue is the result of a bug in a specific multi-signature contract known as wallet.sol. Data suggests the issue was mitigated, however, as 377,000 ethers that were potentially vulnerable to the issue were recovered by white hat hackers.

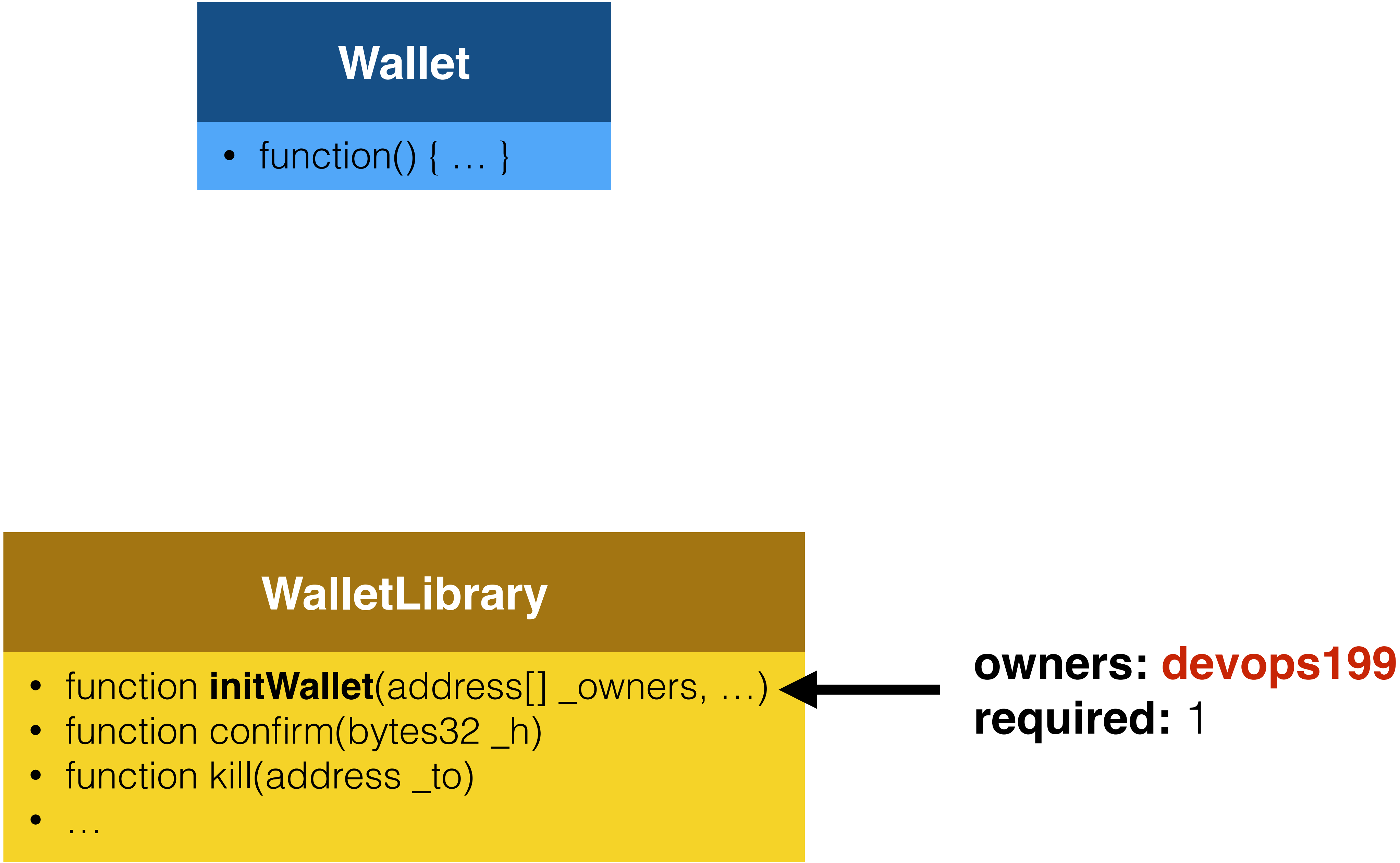
Part 2: Electric Boogaloo

Wallet

- function() { … }

WalletLibrary

- function **initWallet**(address[] _owners, …)
- function confirm(bytes32 _h)
- function kill(address _to)
- …


Wallet
• function() { … }
WalletLibrary
• function initWallet(address[] _owners, …)
• function confirm(bytes32 _h)
• function kill(address _to)
• …
owners: devops199
required: 1

Wallet

- function() { … }

WalletLibrary

- owner: **devops199**
- function initWallet(address[] _owners, …)
- function confirm(bytes32 _h)
- function kill(address _to)

paritytech / parity

Watch 315 | Star 3,840 | Fork 744

Code | Issues 367 | Pull requests 21 | Projects 5 | Wiki | Insights

anyone can kill your contract #6995

New issue

Closed **ghost** opened this issue on 6 Nov 2017 · 16 comments

ghost commented on 6 Nov 2017 • edited by ghost

I accidentally killed it.

https://etherscan.io/address/0x863df6bfa4469f3ead0be8f9f2aae51c91a907b4

👍 23 👎 1 😄 59 🎉 29 😕 14 ❤️ 24

jtakalai commented on 7 Nov 2017

Hmmh, clearly the kill came from registered owner, and required signatures was 0, see initWallet transaction arguments https://etherscan.io/tx/0x05f71e1b2cb4f03e547739db15d080fd30c989eda04d37ce6264c5686e0722c9

ghost commented on 7 Nov 2017

Will it effect the dependent multisig wallets? When i query " isowner(<any_addr>)" the multisig wallets returns TRUE.

Assignees

No one assigned

Labels

F1-security
M8-contracts
P0-dropeverything

Projects

None yet

Milestone

1.9

Notifications

Subscribe

You're not receiving notifications

Someone Triggered a Bug That Has Frozen Over 280 Million in Ethereum - Google Chrome

Secure | https://wccftech.com/bug-frozen-280-million-ethereum/

FINANCE : SECURITY

More ETH Troubles: Someone Triggered a Bug That Has Frozen Over 280 Million in Ethereum

By Rafia Shaikh

Nov 7, 2017

82 SHARES

SHARE TWEET SUBMIT

Other vulnerabilities

Randomness

Nothing is secret

game_random.sol

game_random.sol

```
pragma solidity ^0.4.18;

contract GameRandom {

    uint256 private seed;
    uint256 private iteration = 0;

    function play() public payable {
        require(msg.value >= 1 ether);

        iteration++;

        uint randomNumber = uint(keccak256(seed + iteration));
        bool won = (randomNumber % 2) == 0;

        if (won) {
            msg.sender.transfer(2 ether);
        }
    }
}
```

Line 20, Column 2 Tab Size: 4 Solidity

game_random.sol

game_random.sol

```
pragma solidity ^0.4.18;

contract GameRandom {

    uint256 private seed;
    uint256 private iteration = 0;

    function play() public payable {
        require(msg.value >= 1 ether);

        iteration++;

        uint randomNumber = uint(keccak256(seed + iteration));
        bool won = (randomNumber % 2) == 0;

        if (won) {
            msg.sender.transfer(2 ether);
        }
    }
}
```

Line 20, Column 2 Tab Size: 4 Solidity

game_random.sol

```
pragma solidity ^0.4.18;

contract GameRandom {

    uint256 private seed;
    uint256 private iteration = 0;

    function play() public payable {
        require(msg.value >= 1 ether);

        iteration++;

        uint randomNumber = uint(keccak256(seed + iteration));
        bool won = (randomNumber % 2) == 0;

        if (won) {
            msg.sender.transfer(2 ether);
        }
    }
}
```

Line 20, Column 2   Tab Size: 4   Solidity

game_random.sol

```
pragma solidity ^0.4.18;

contract GameRandom {

    uint256 private seed;
    uint256 private iteration = 0;

    function play() public payable {
        require(msg.value >= 1 ether);

        iteration++;

        uint randomNumber = uint(keccak256(seed + iteration));
        bool won = (randomNumber % 2) == 0;

        if (won) {
            msg.sender.transfer(2 ether);
        }
    }
}
```

Line 20, Column 2 Tab Size: 4 Solidity

NOTHING IS SECRET

game_random.sol

game_random.sol

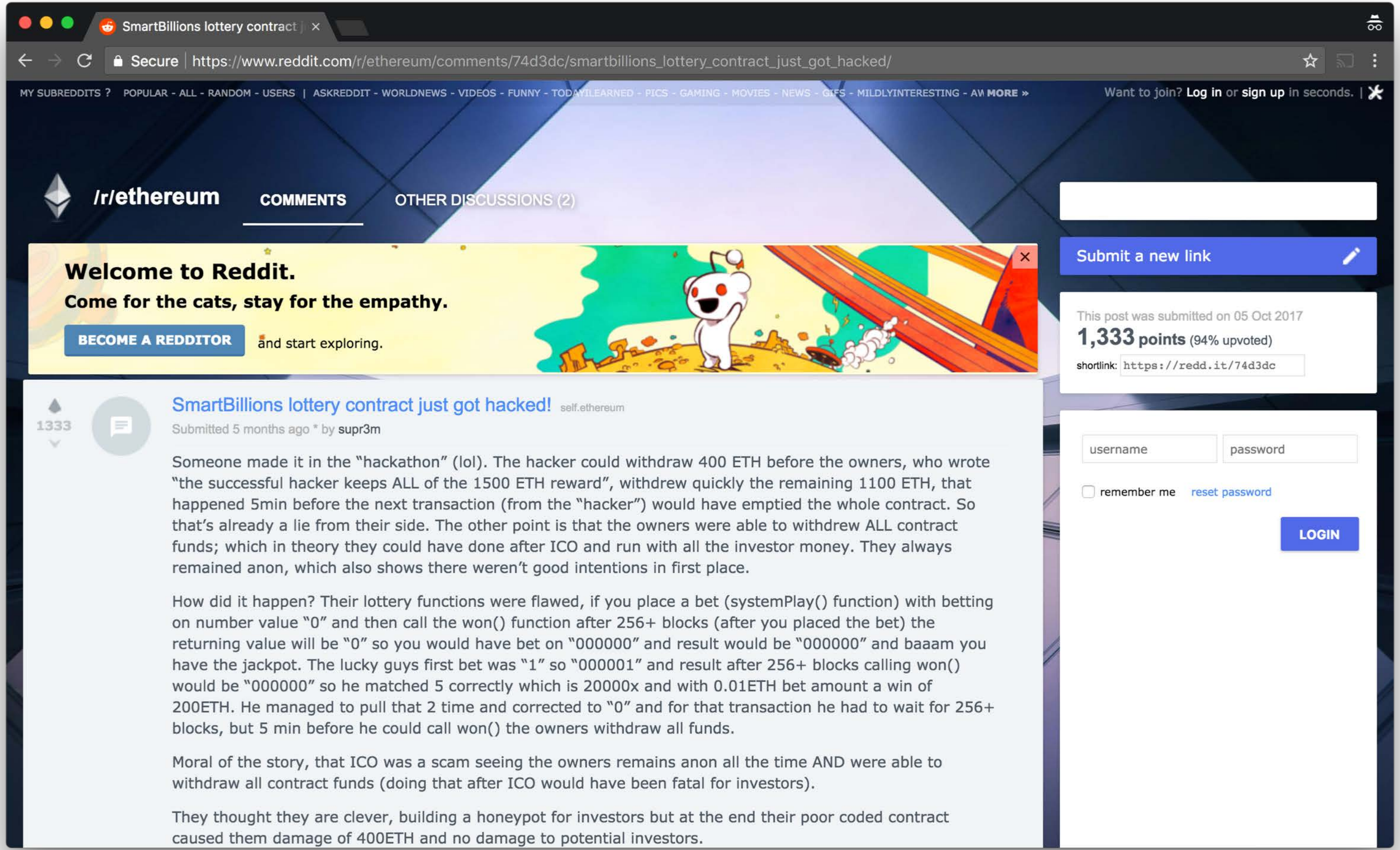
```solidity
pragma solidity ^0.4.18;

contract GameRandom {

    uint256 private seed;
    uint256 private iteration = 0;

    function play() public payable {
        require(msg.value >= 1 ether);

        iteration++;

        uint randomNumber = uint(keccak256(seed + iteration));
        bool won = (randomNumber % 2) == 0;

        if (won) {
            msg.sender.transfer(2 ether);
        }
    }
}
```

Line 20, Column 2 | Tab Size: 4 | Solidity

Block and Transaction Properties

- `block.blockhash(uint blockNumber) returns (bytes32)`: hash of the given block - only works for 256 most recent blocks excluding current
- `block.coinbase` (`address`): current block miner's address
- `block.difficulty` (`uint`): current block difficulty
- `block.gaslimit` (`uint`): current block gaslimit
- `block.number` (`uint`): current block number
- `block.timestamp` (`uint`): current block timestamp as seconds since unix epoch
- `gasleft() returns (uint256)`: remaining gas
- `msg.data` (`bytes`): complete calldata
- `msg.gas` (`uint`): remaining gas - deprecated in version 0.4.21 and to be replaced by `gasleft()`
- `msg.sender` (`address`): sender of the message (current call)
- `msg.sig` (`bytes4`): first four bytes of the calldata (i.e. function identifier)
- `msg.value` (`uint`): number of wei sent with the message
- `now` (`uint`): current block timestamp (alias for `block.timestamp`)
- `tx.gasprice` (`uint`): gas price of the transaction
- `tx.origin` (`address`): sender of the transaction (full call chain)

12.2. **Random Numbers.** Providing random numbers within a deterministic system is, naturally, an impossible task. However, we can approximate with pseudo-random numbers by utilising data which is generally unknowable at the time of transacting. Such data might include the block's hash, the block's timestamp and the block's beneficiary address. In order to make it hard for malicious miner to control those values, one should use the BLOCKHASH operation in order to use hashes of the previous 256 blocks as pseudo-random numbers. For a series of such numbers,



SmartBillions lottery contract j ×

Secure | https://www.reddit.com/r/ethereum/comments/74d3dc/smartbillions_lottery_contract_just_got_hacked/

MY SUBREDDITS ? POPULAR - ALL - RANDOM - USERS | ASKREDDIT - WORLDNEWS - VIDEOS - FUNNY - TODAYILEARNED - PICS - GAMING - MOVIES - NEWS - GIFS - MILDLYINTERESTING - AW MORE »

Want to join? Log in or sign up in seconds. |

/r/ethereum COMMENTS OTHER DISCUSSIONS (2)

Welcome to Reddit.

Come for the cats, stay for the empathy.

BECOME A REDDITOR and start exploring.

1333

SmartBillions lottery contract just got hacked! self.ethereum

Submitted 5 months ago * by supr3m

Someone made it in the "hackathon" (lol). The hacker could withdraw 400 ETH before the owners, who wrote "the successful hacker keeps ALL of the 1500 ETH reward", withdrew quickly the remaining 1100 ETH, that happened 5min before the next transaction (from the "hacker") would have emptied the whole contract. So that's already a lie from their side. The other point is that the owners were able to withdrew ALL contract funds; which in theory they could have done after ICO and run with all the investor money. They always remained anon, which also shows there weren't good intentions in first place.

How did it happen? Their lottery functions were flawed, if you place a bet (systemPlay() function) with betting on number value "0" and then call the won() function after 256+ blocks (after you placed the bet) the returning value will be "0" so you would have bet on "000000" and result would be "000000" and baaam you have the jackpot. The lucky guys first bet was "1" so "000001" and result after 256+ blocks calling won() would be "000000" so he matched 5 correctly which is 20000x and with 0.01ETH bet amount a win of 200ETH. He managed to pull that 2 time and corrected to "0" and for that transaction he had to wait for 256+ blocks, but 5 min before he could call won() the owners withdraw all funds.

Moral of the story, that ICO was a scam seeing the owners remains anon all the time AND were able to withdraw all contract funds (doing that after ICO would have been fatal for investors).

They thought they are clever, building a honeypot for investors but at the end their poor coded contract caused them damage of 400ETH and no damage to potential investors.

Submit a new link

This post was submitted on 05 Oct 2017

1,333 points (94% upvoted)

shortlink: https://redd.it/74d3dc

username password

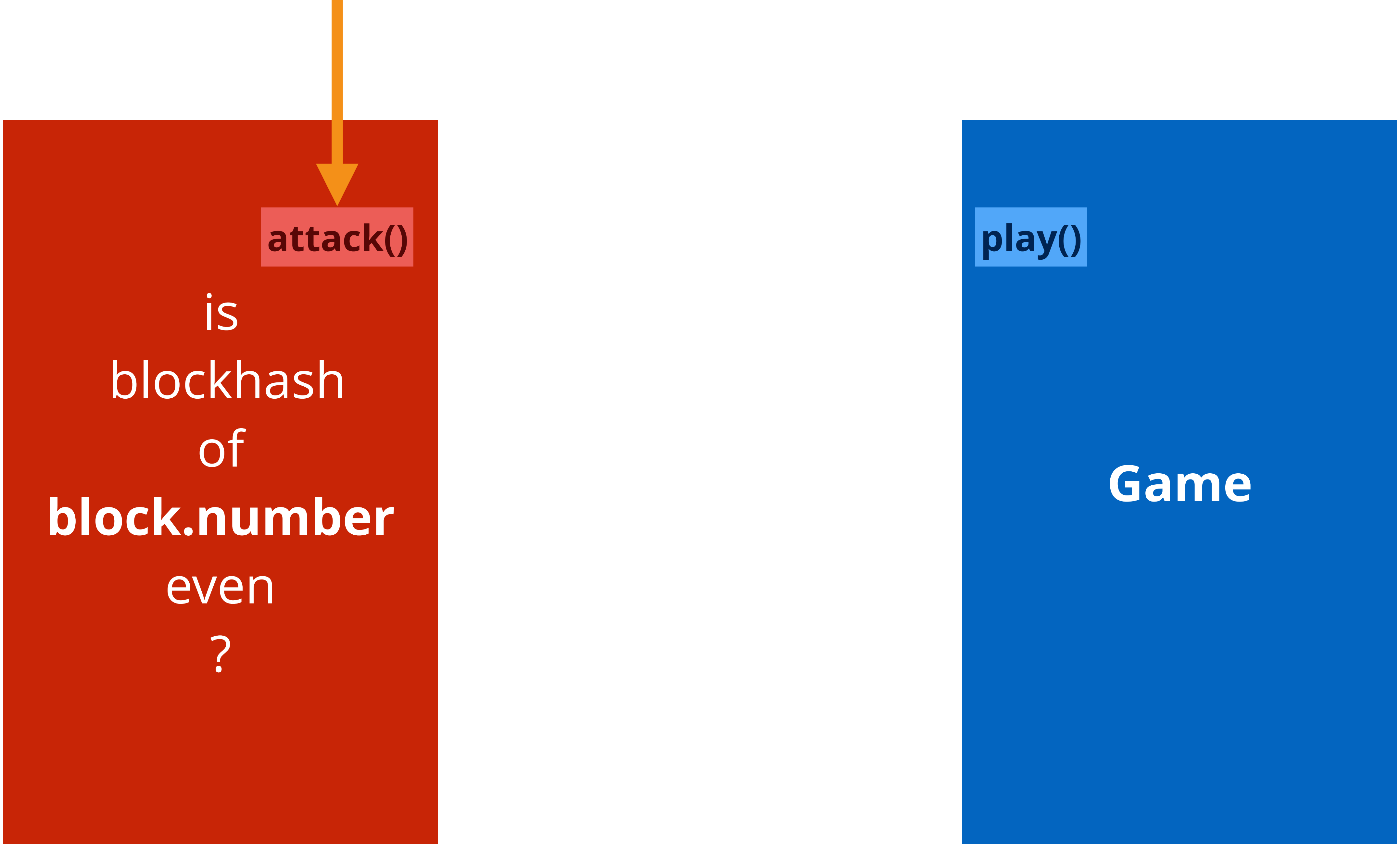
remember me reset password

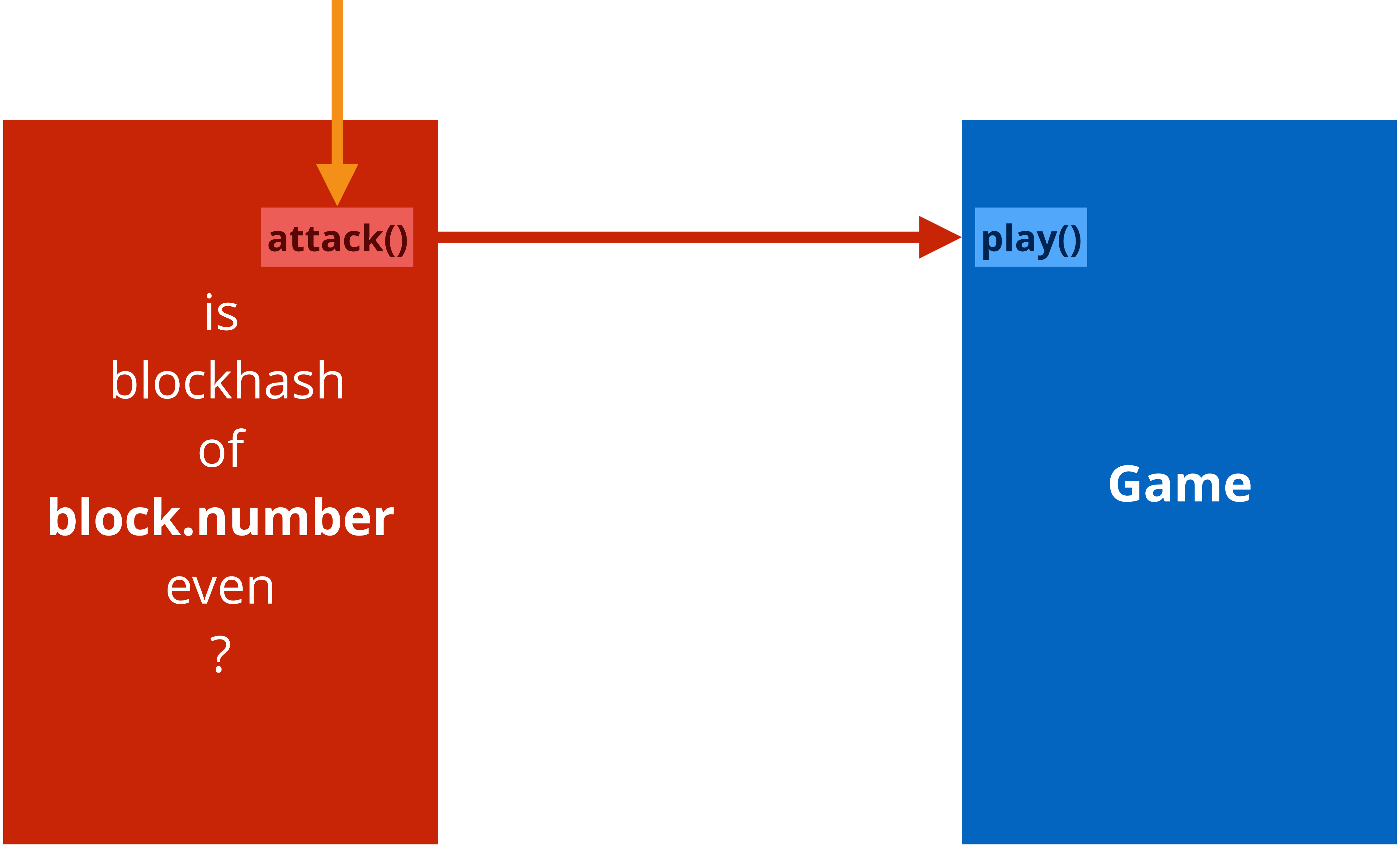
LOGIN

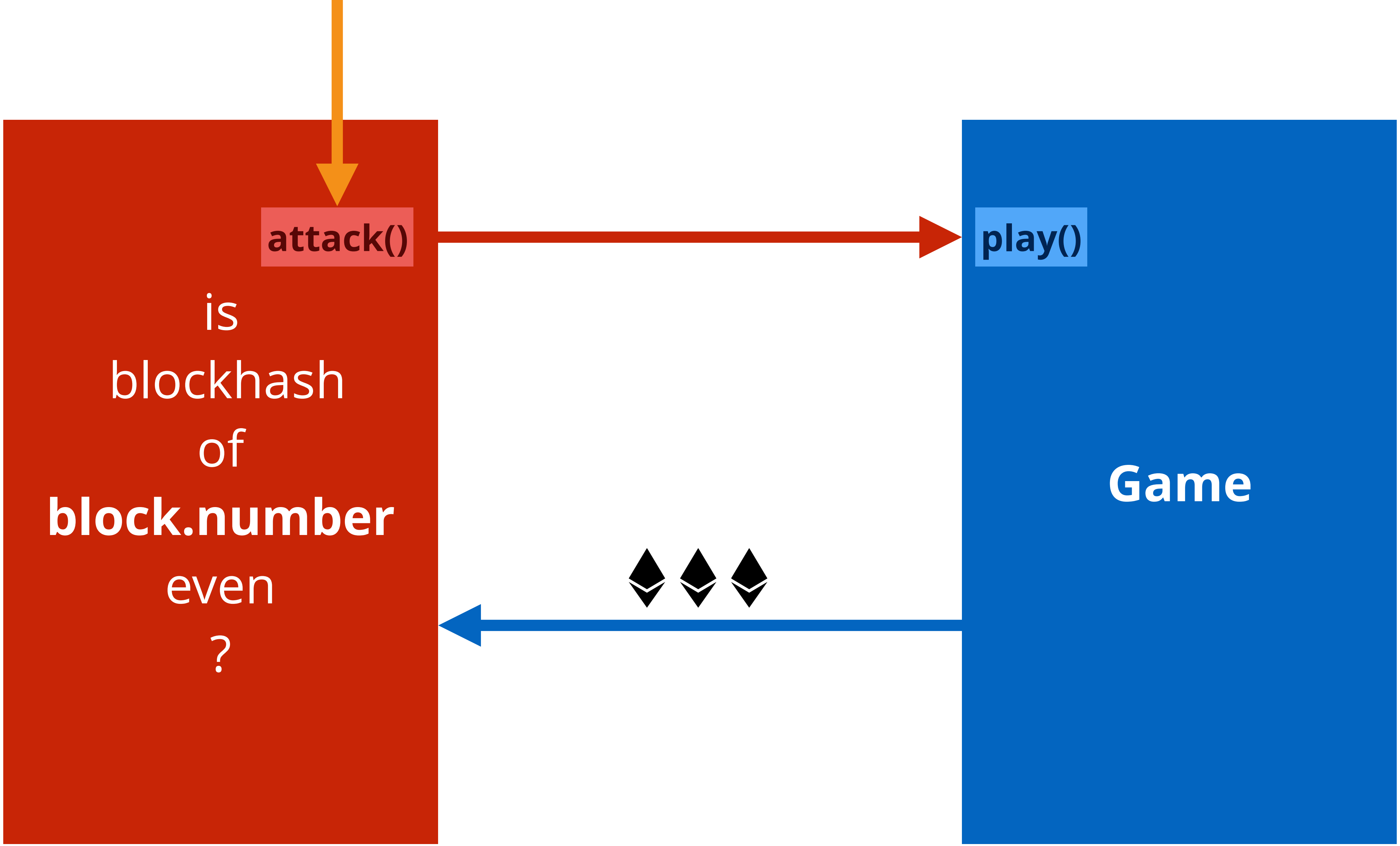
game_random.sol

```
pragma solidity ^0.4.18;

contract GameRandom {

    function play() public payable {
        require(msg.value >= 1 ether);

        bool won = (block.blockhash(block.number) % 2) == 0;

        if (won) {
            msg.sender.transfer(2 ether);
        }
    }

}
```

Line 10, Column 19 Tab Size: 4 Solidity


attack()
is
blockhash
of
block.number
even
?
play()
Game


attack()
is
blockhash
of
block.number
even
?
play()
Game


attack()
is
blockhash
of
block.number
even
?
play()
Game

game_random.sol

block.timestamp

```
pragma solidity ^0.4.18;

contract GameRandom {

    function play() public payable {
        require(msg.value >= 1 ether);

        bool won = (block.blockhash(block.number) % 2) == 0;

        if (won) {
            msg.sender.transfer(2 ether);
        }
    }

}
```

Line 10, Column 19   Tab Size: 4   Solidity

Randomness

Nothing is secret

- Be wary of ”randomness”
- **Strong reliance on time** is bad

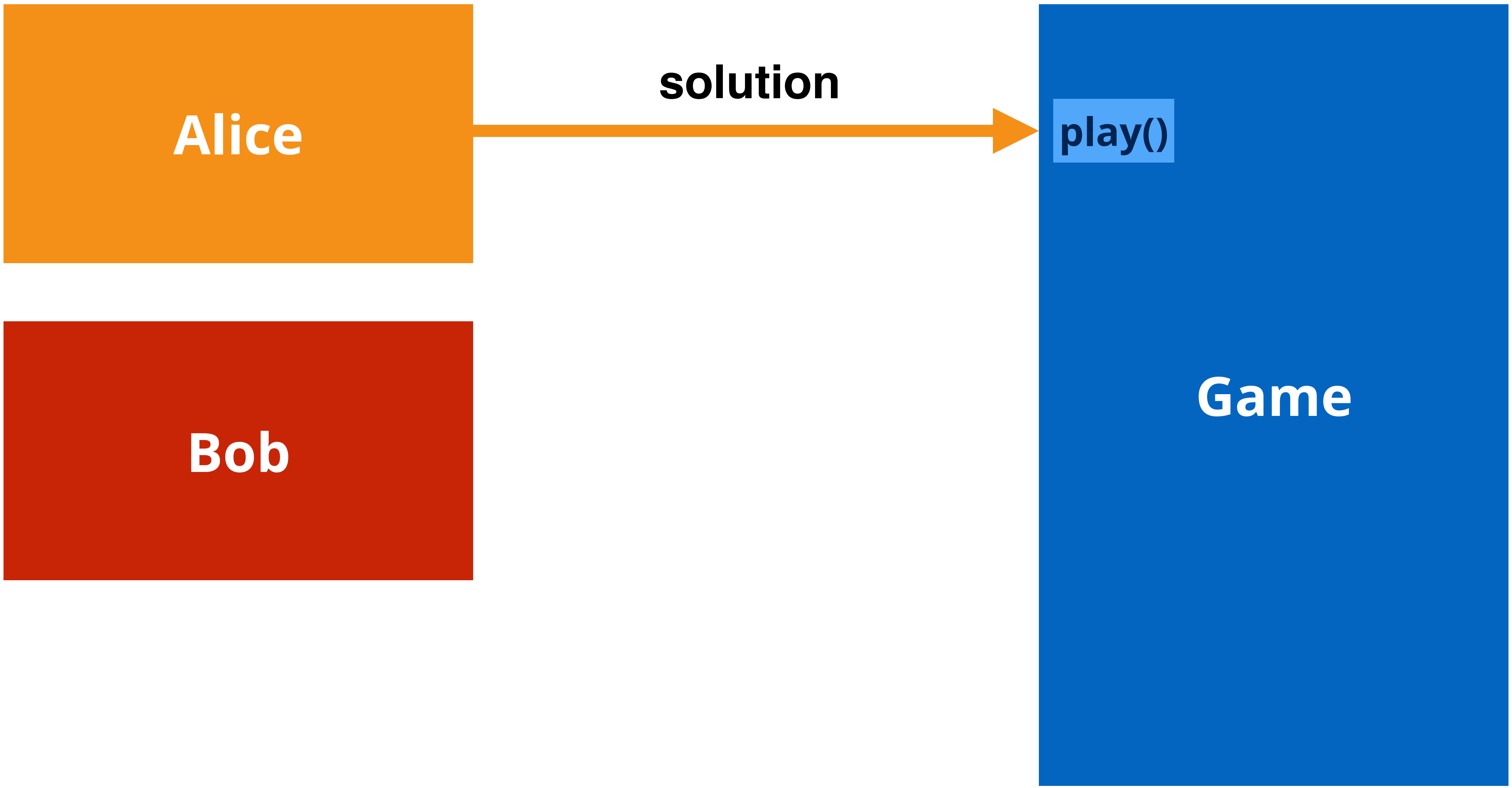
Front-running

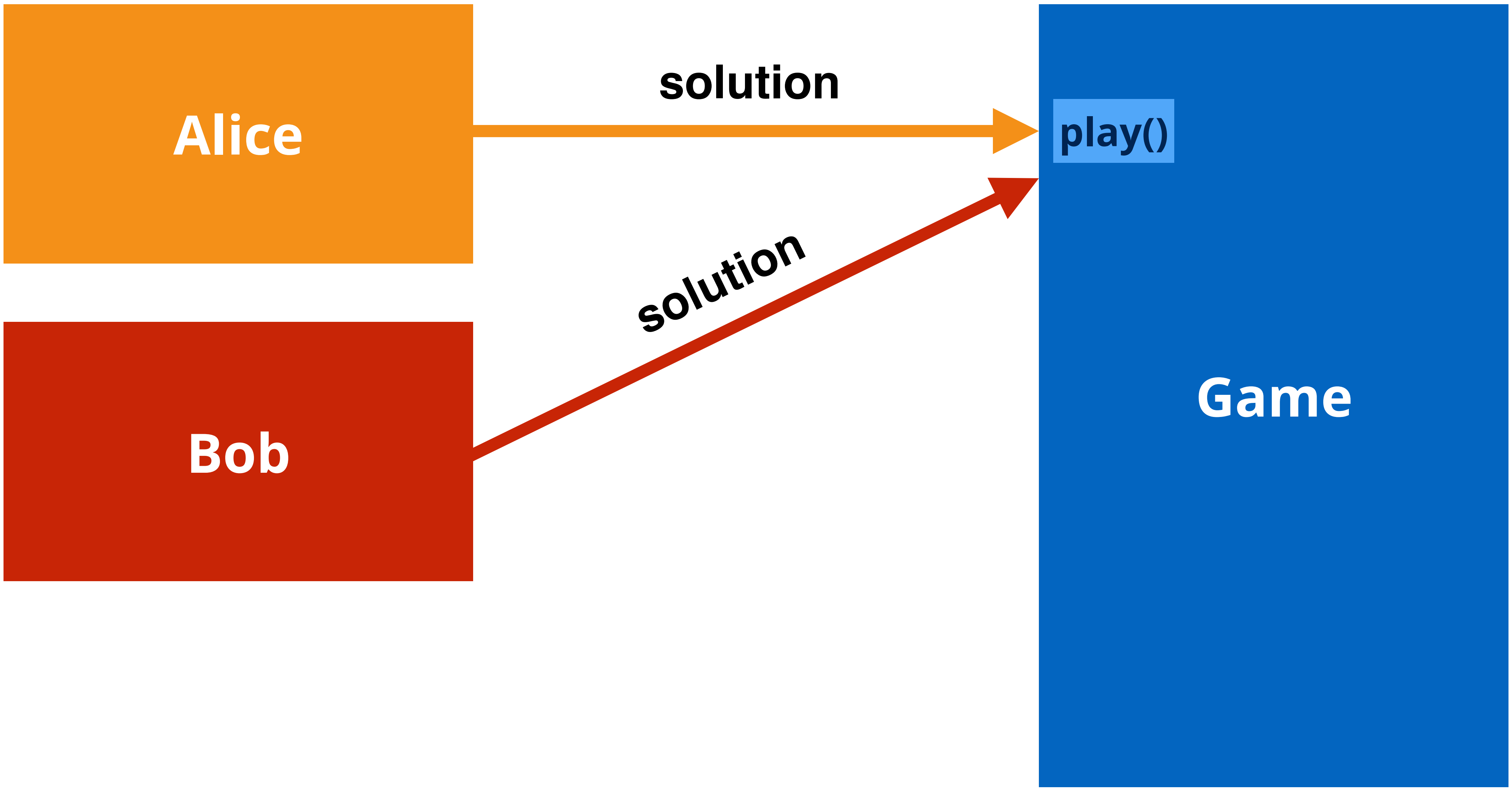
also known as race condition, TOCTOU, TOD

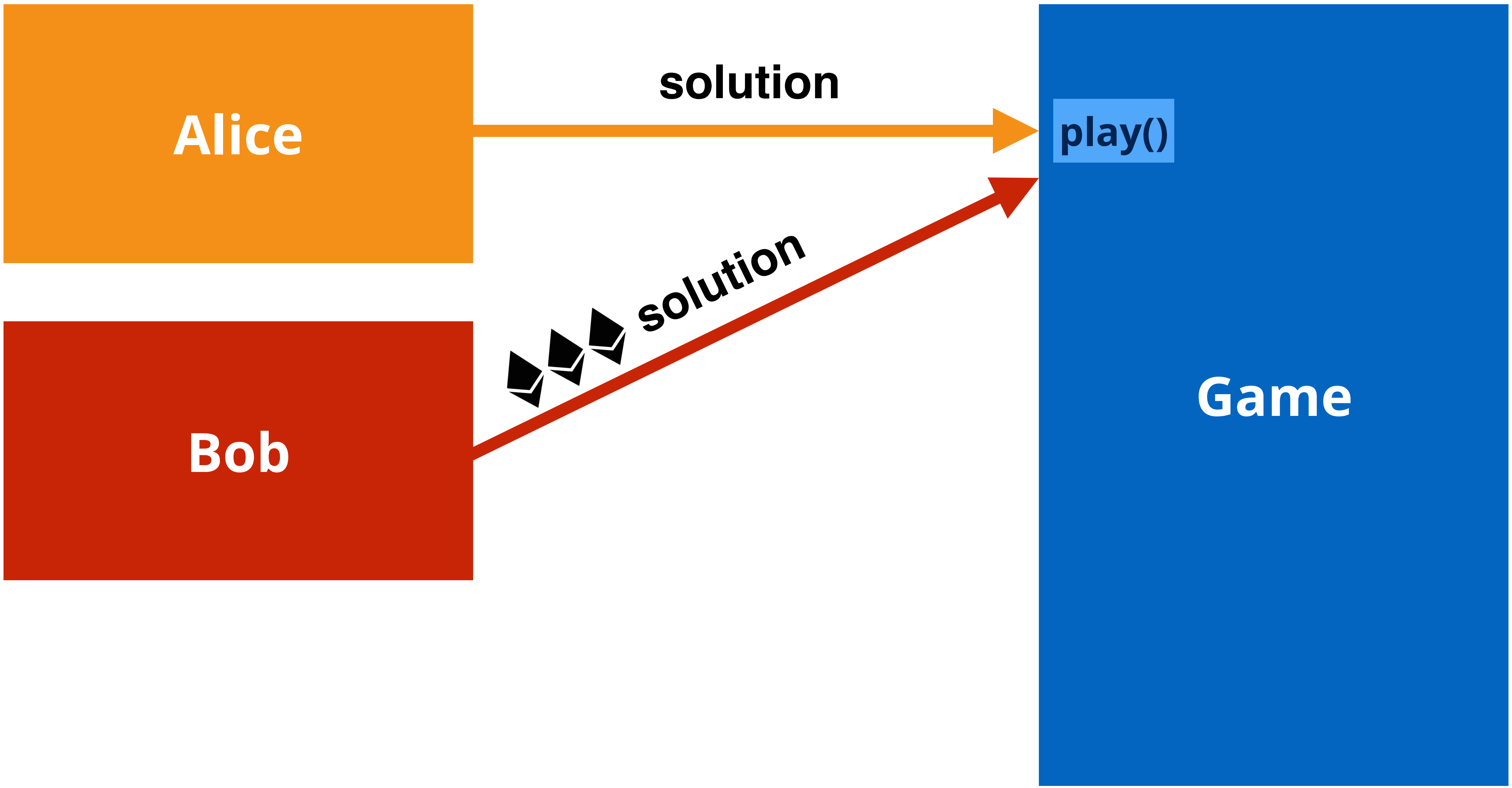
game_random.sol

```
pragma solidity ^0.4.18;

contract GameRandom {

    uint rsaNumber = 21940124219032193011 51; // N = p * q

    function play(uint p, uint q) public payable {

        if (p * q == rsaNumber) {
            msg.sender.transfer(1000 ether);
        }
    }
}
```

Line 13, Column 2   Tab Size: 4   Solidity


solution
Alice
play()
Bob
Game


Alice
solution
play()
solution
Bob
Game


Alice
solution
play()
Bob
solution
Game

EIPs/eip-20.md at master · eth ×

GitHub, Inc. [US] | https://github.com/ethereum/EIPs/blob/master/EIPS/eip-20.md

transferFrom

Transfers `_value` amount of tokens from address `_from` to address `_to`, and MUST fire the `Transfer` event.

The `transferFrom` method is used for a withdraw workflow, allowing contracts to transfer tokens on your behalf. This can be used for example to allow a contract to transfer tokens on your behalf and/or to charge fees in sub-currencies. The function SHOULD `throw` unless the `_from` account has deliberately authorized the sender of the message via some mechanism.

Note Transfers of 0 values MUST be treated as normal transfers and fire the `Transfer` event.

```
function transferFrom(address _from, address _to, uint256 _value) returns (bool success)
```

approve

Allows `_spender` to withdraw from your account multiple times, up to the `_value` amount. If this function is called again it overwrites the current allowance with `_value`.

NOTE: To prevent attack vectors like the one described here and discussed here, clients SHOULD make sure to create user interfaces in such a way that they set the allowance first to `0` before setting it to another value for the same spender. THOUGH The contract itself shouldn't enforce it, to allow backwards compatibility with contracts deployed before

```
function approve(address _spender, uint256 _value) returns (bool success)
```

allowance

Returns the amount which `_spender` is still allowed to withdraw from `_owner`.

```
function allowance(address _owner, address _spender) constant returns (uint256 remaining)
```

Front-running

also known as race condition, TOCTOU, TOD

- Can **transactions re-ordering** strongly affect the outcome?
- Do some transactions contain **secret** information?

Arithmetic Errors

Integer overflows and underflows

uint8

1	1	1	1	1	1	1	1

+1

0	0	0	0	0	0	0	0

```
pragma solidity ^0.4.15;

contract Overflow {
    uint private sellerBalance=0;

    function add(uint value) returns (bool){
        sellerBalance += value; // possible overflow

        // possible auditor assert
        // assert(sellerBalance >= value);
    }

    function safe_add(uint value) returns (bool){
        require(value + sellerBalance >= sellerBalance);
        sellerBalance += value;
    }
}
```

https://github.com/trailofbits/not-so-smart-contracts/blob/master/integer_overflow/interger_overflow_1.sol

```
library SafeMath {

  /**
  * @dev Multiplies two numbers, throws on overflow.
  */
  function mul(uint256 a, uint256 b) internal pure returns (uint256) {
    if (a == 0) {
      return 0;
    }
    uint256 c = a * b;
    assert(c / a == b);
    return c;
  }

  /**
  * @dev Integer division of two numbers, truncating the quotient.
  */
  function div(uint256 a, uint256 b) internal pure returns (uint256) {
    // assert(b > 0); // Solidity automatically throws when dividing by 0
    uint256 c = a / b;
    // assert(a == b * c + a % b); // There is no case in which this doesn't hold
    return c;
  }

  /**
  * @dev Substracts two numbers, throws on overflow (i.e. if subtrahend is greater than minuend).
  */
  function sub(uint256 a, uint256 b) internal pure returns (uint256) {
    assert(b <= a);
    return a - b;
  }

  /**
  * @dev Adds two numbers, throws on overflow.
  */
  function add(uint256 a, uint256 b) internal pure returns (uint256) {
    uint256 c = a + b;
    assert(c >= a);
    return c;
  }
}
```

pirapira commented on Oct 13, 2016 • edited

Member

The main reason I opened separate issues is that each requires discussion if we want it or not, but if we are set to go over everything, here is a list.

- [ ] exception on overflow in unsigned->signed conversion
- [ ] exception on overflow in signed->unsigned conversion
- [ ] exception on overflow in size-decreasing implicit conversion
- [ ] exception on overflow in addition of two signed numbers
- [ ] exception on overflow in addition of two unsigned numbers
- [ ] exception on underflow in subtraction of two signed numbers
- [ ] exception on underflow in subtraction of two unsigned numbers
- [ ] exception on overflow in multiplication of two signed numbers
- [ ] exception on overflow in multiplication of two unsigned numbers
- [ ] exception on overflow in shifts
- [ ] exception on overflow in `++` on a signed number
- [ ] exception on overflow in `++` on an unsigned number
- [ ] exception on underflow in `--` on a signed number
- [ ] exception on underflow in `--` on an unsigned number
- [ ] exception on overflow in `+=`
- [ ] exception on overflow in `-=`
- [ ] exception on overflow in `*=`
- [ ] exception on overflow in `/=`
- [ ] make sure no optimizations are relying on `(a + b - b == a)`; lest they remove overflow exceptions
- [ ] compiler error on an out-of-range constant expression when the constant expression is interpreted into a type (`uint x = -1`)
- [ ] exception on overflow in `INT_MIN/-1` #1091

What to Look For

- Unchecked arithmetic operation
- Can be increased/decreased by large enough leaps (incrementing upward generally too expensive)
- Accessible from lower levels of privilege

Denial of Service

Etherscan
The Ethereum Block Explorer

LOGIN | Search by Address / Txhash / Block / Token / Ens | GO

HOME | BLOCKCHAIN | ACCOUNT | TOKEN | CHART | MISC

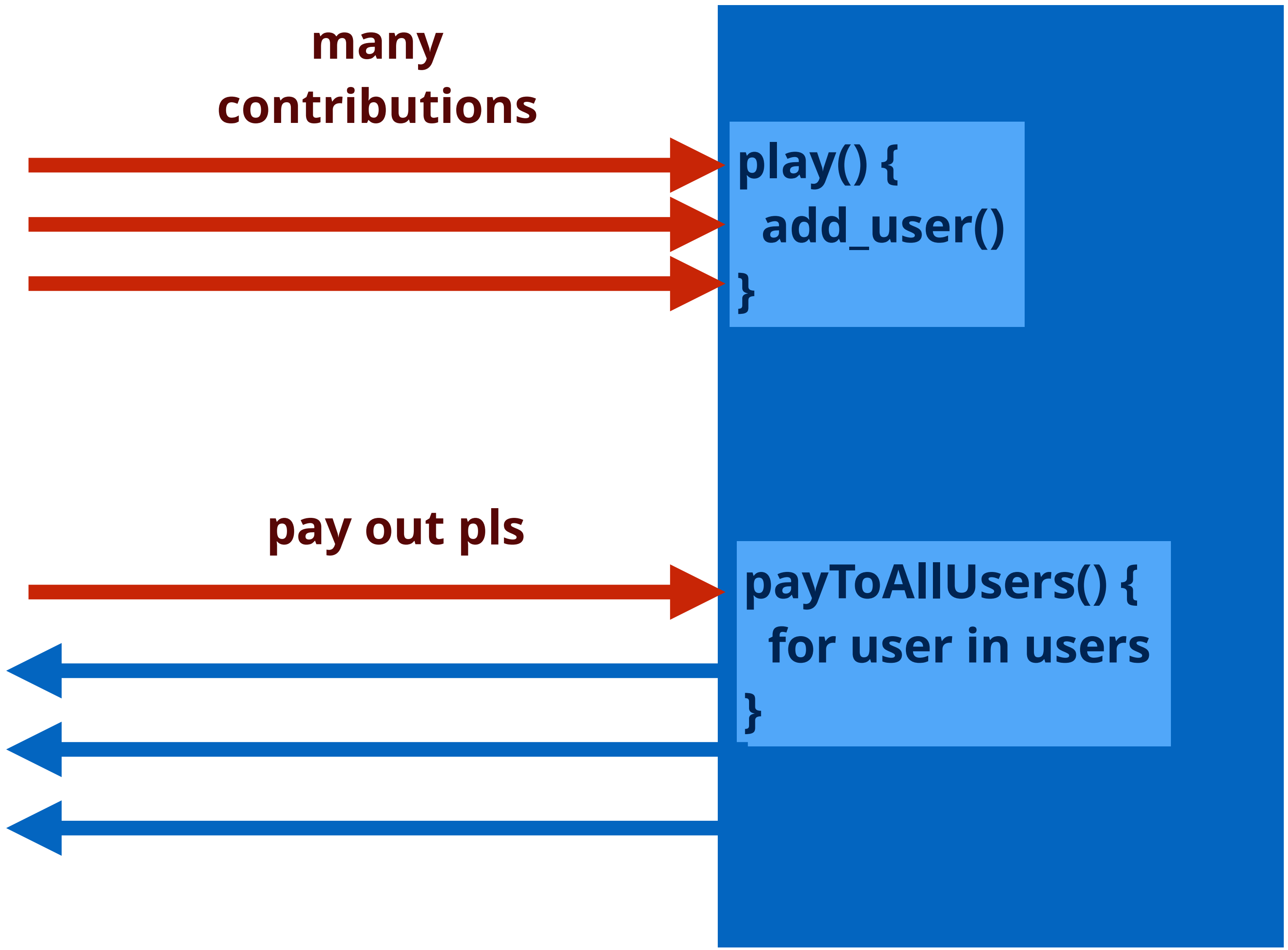
Blocks

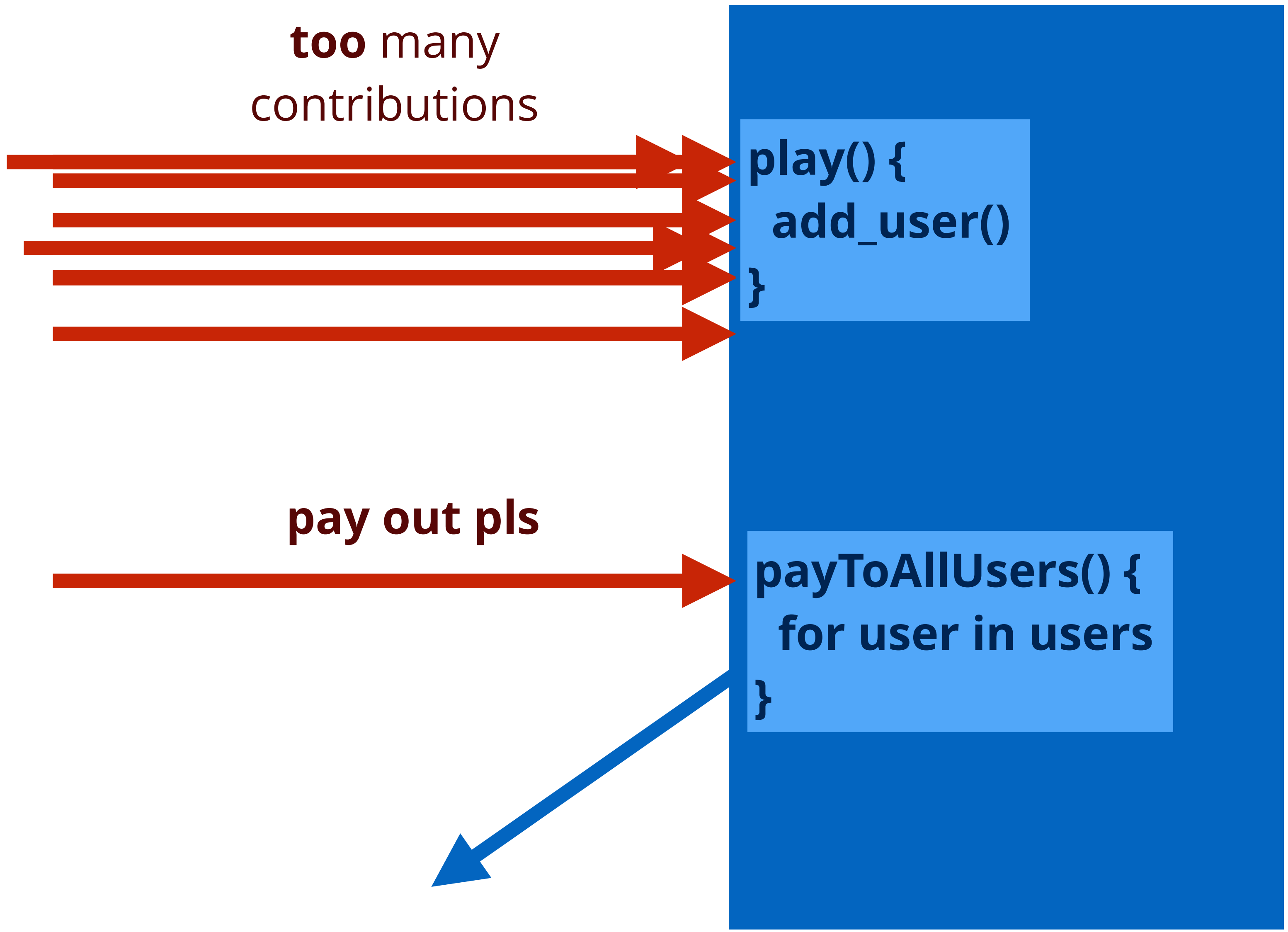
Home / Blocks

Showing Block (#5216258 to #5216234) out of 5216259 total blocks

First | Prev | Page **1** of **208651** | Next | Last

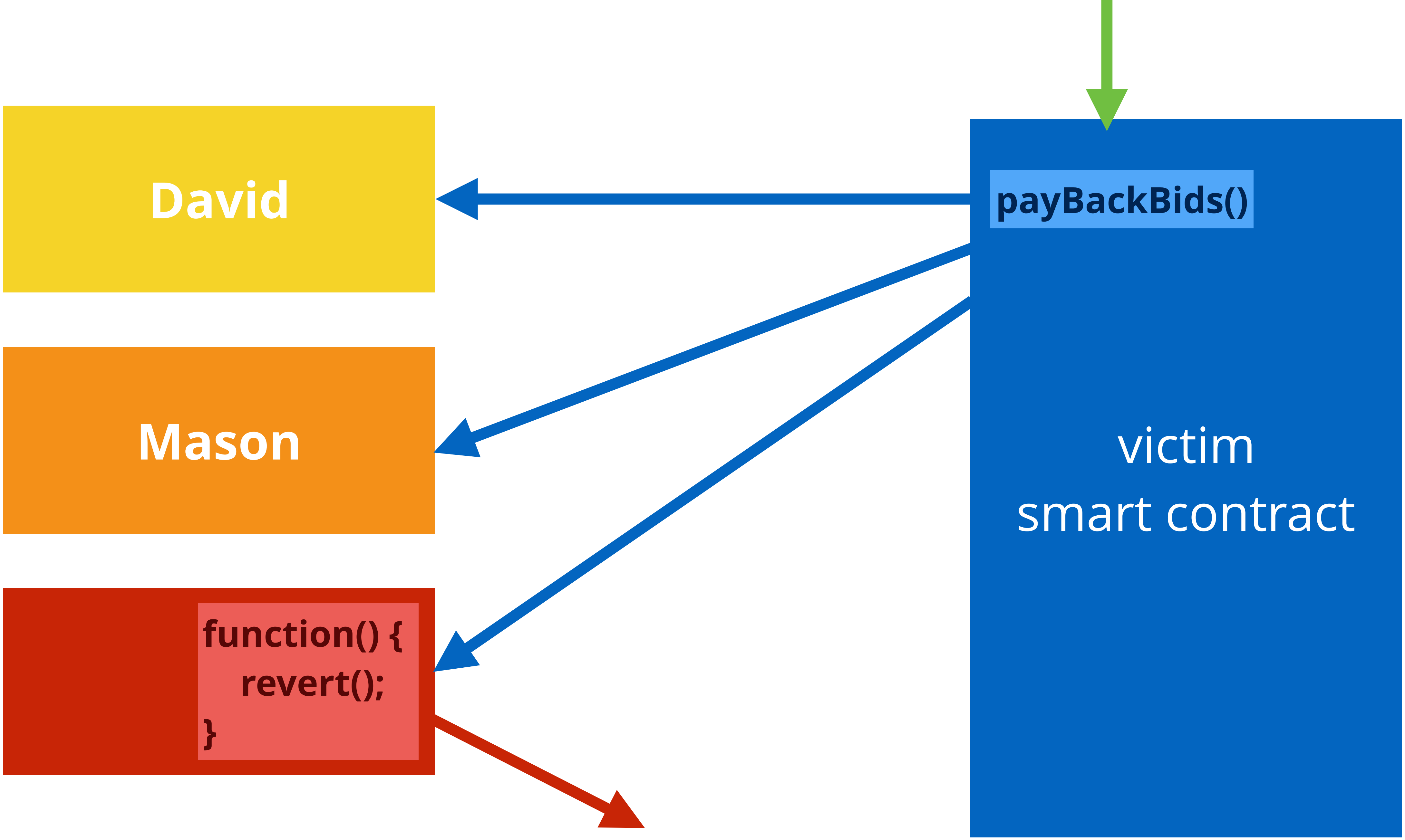
Height	Age	txn	Uncles	Miner	GasUsed	GasLimit	Avg.GasPrice	Reward
5216258	40 secs ago	146	0	DwarfPool1	7985670 (99.87%)	7996085	25.95 Gwei	3.20722 Ether
5216257	1 min ago	124	0	SparkPool	7989541 (99.97%)	7992185	12.52 Gwei	3.1 Ether
5216256	1 min ago	6	0	waterhole	126000 (1.58%)	7999959	62.17 Gwei	3.00783 Ether


many
contributions
play() {
add_user()
}
pay out pls
payToAllUsers() {
for user in users
}



too many
contributions
play() {
add_user()
}
pay out pls
payToAllUsers() {
for user in users
}

But Wait—There’s More!


David
Mason
function() {
revert();
}
payBackBids()
victim
smart contract

0x3b	EXTCODESIZE	1	1	Get size of an account's code. $\boldsymbol{\mu}'_{\mathbf{s}}[0] \equiv \|\boldsymbol{\sigma}[\boldsymbol{\mu}_s[0] \mod 2^{160}]_c\|$

Intro to Accounting for Solidity Programmers

```
contract BidThing {

    function bid(uint256 object) public payable {
        // place ethers
    }
    function cancelBid(uint256 object) public {
        // refund ethers
    }
    function acceptBid(uint256 object) public {
        // transfer ethers
    }

}
```

```
contract BidThing {

    function bid(uint256 object) public payable {
        // place ethers
    }
    function cancelBid(uint256 object) public {
        // refund ethers
    }
    function acceptBid(uint256 object) public {
        // transfer ethers
    }

    function withdraw(uint256 _amount) public {
        address owner = 0x32523459909435...
        owner.transfer(_amount);
    }

}
```

What to Look For

- For loops dependent on **storage values**
- Functions dependent on **transactions**
- Public functions affecting **contract balance**

Methodology Takeaways

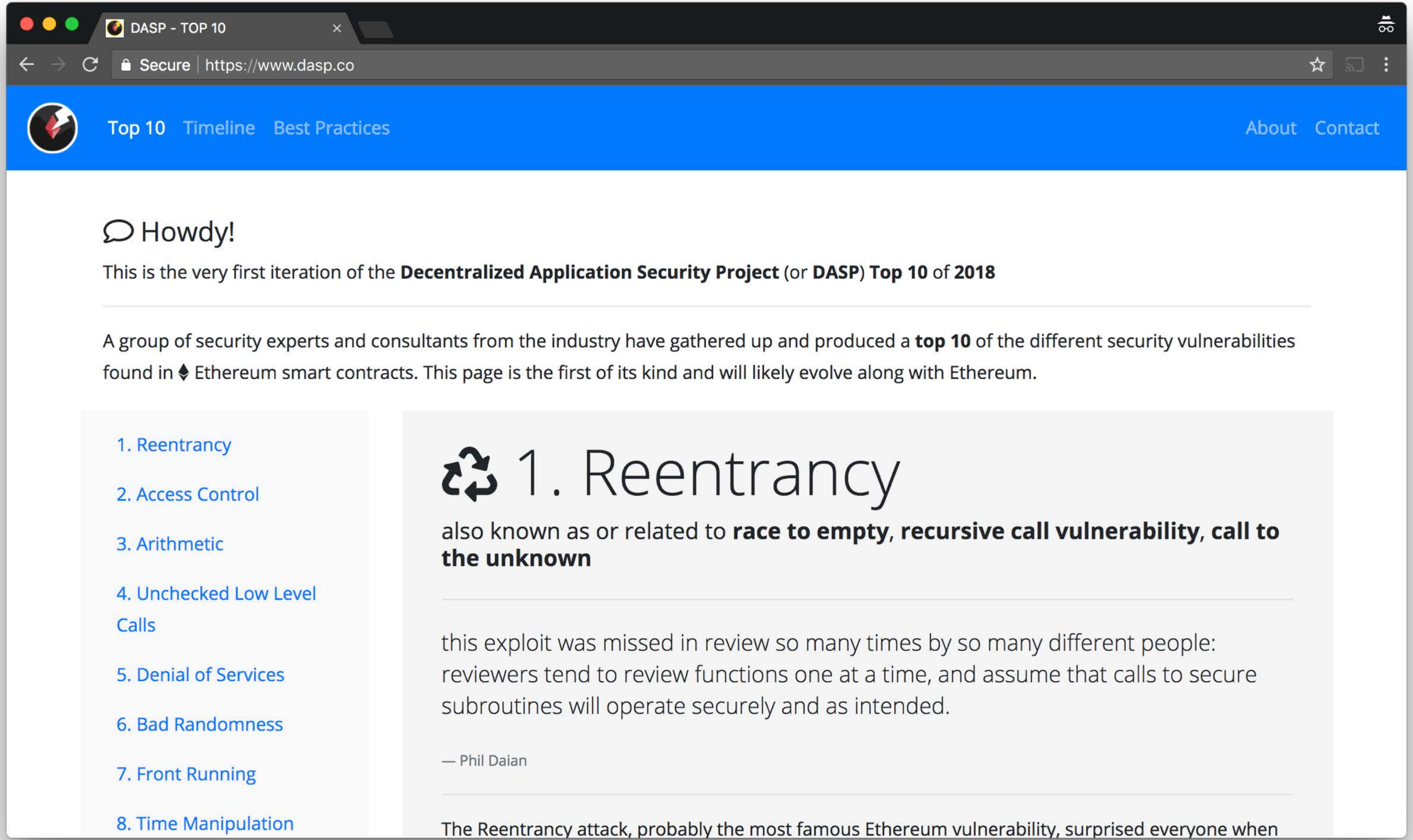
- Prerequisite: knowledge of common Ethereum **gotchas**
- **Understand** the contract
- Look for
  - Calls to external contracts and low level function calls
  - Denial of Services
  - Reliance on public or secret values
  - Logic relying on user input
  - Re-inventing the wheel
  - Excessive use of inline assembly
- Analyze any **publicly-accessible** functions in depth

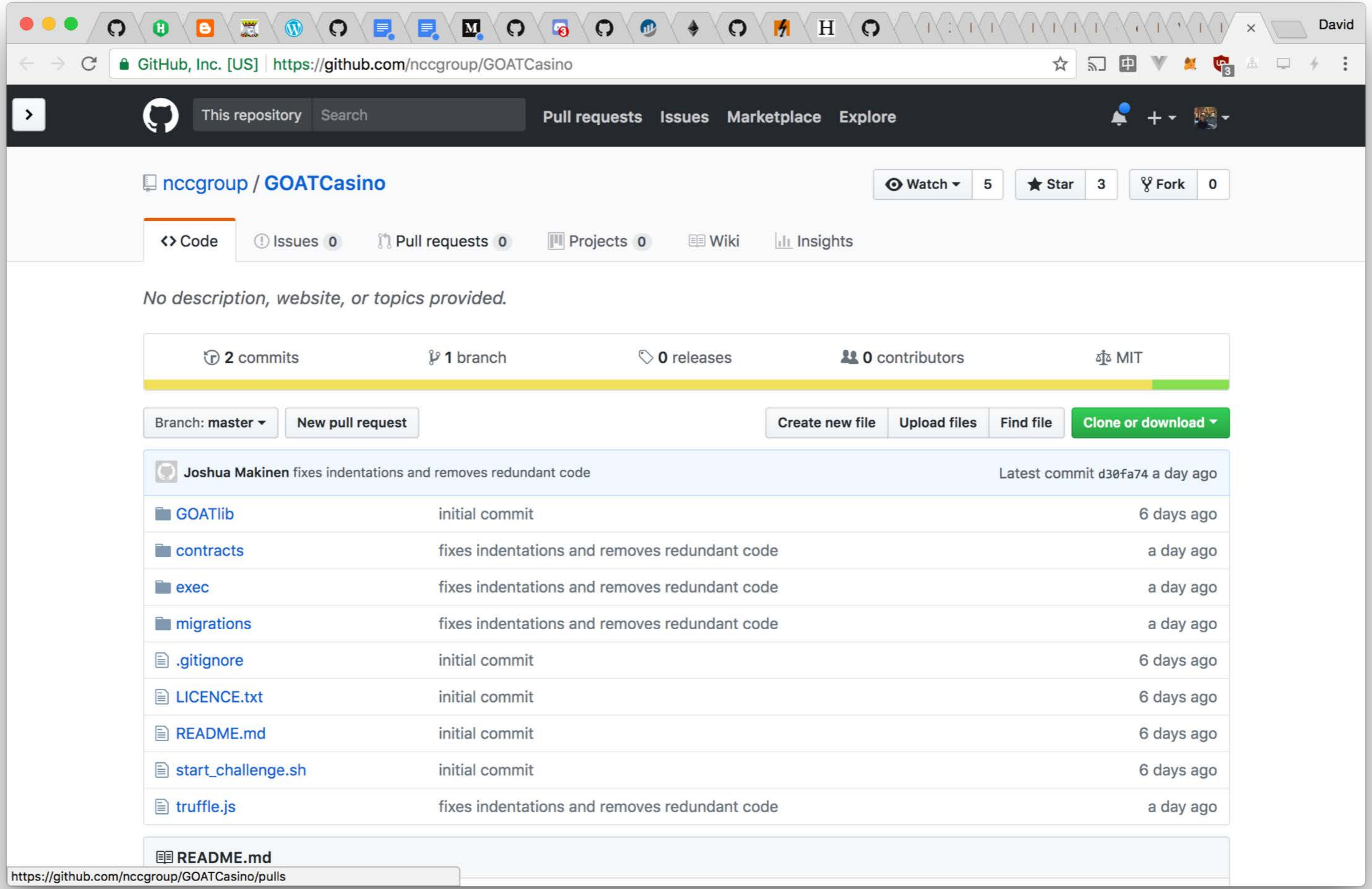
Tools

- Oyente, Mythril, Manticore, …
  - Run them in the beginning, but don't expect much unless there is an evident issue
- **Remix** and Linters
  - really useful to catch quick mistakes and bad behaviors that might lead to issues. Awesome to **test** and **debug** quickly.
- **Truffle**:
  - is OK
- Disassembler and decompilers

Future

- Formal verification
- Frameworks and re-usable code
- Ethereum rapidly evolves
- Solidity, Mist, Ethereum Wallet still have experimental versions.
- Audit process is immature
- Lots of money put in lightly-audited smart contracts
- Unknown unknowns ?


DASP - TOP 10
Secure https://www.dasp.co
Top 10 Timeline Best Practices
About Contact
Howdy!
This is the very first iteration of the Decentralized Application Security Project (or DASP) Top 10 of 2018
A group of security experts and consultants from the industry have gathered up and produced a top 10 of the different security vulnerabilities found in Ethereum smart contracts. This page is the first of its kind and will likely evolve along with Ethereum.
1. Reentrancy
2. Access Control
3. Arithmetic
4. Unchecked Low Level Calls
5. Denial of Services
6. Bad Randomness
7. Front Running
8. Time Manipulation
1. Reentrancy
also known as or related to race to empty, recursive call vulnerability, call to the unknown
this exploit was missed in review so many times by so many different people: reviewers tend to review functions one at a time, and assume that calls to secure subroutines will operate securely and as intended.
— Phil Daian
The Reentrancy attack, probably the most famous Ethereum vulnerability, surprised everyone when


David
GitHub, Inc. [US] https://github.com/nccgroup/GOATCasino
This repository
Search
Pull requests
Issues
Marketplace
Explore
nccgroup / GOATCasino
Watch 5
Star 3
Fork 0
Code
Issues 0
Pull requests 0
Projects 0
Wiki
Insights
No description, website, or topics provided.
2 commits
1 branch
0 releases
0 contributors
MIT
Branch: master
New pull request
Create new file
Upload files
Find file
Clone or download
Joshua Makinen fixes indentations and removes redundant code
Latest commit d30fa74 a day ago
GOATlib initial commit 6 days ago
contracts fixes indentations and removes redundant code a day ago
exec fixes indentations and removes redundant code a day ago
migrations fixes indentations and removes redundant code a day ago
.gitignore initial commit 6 days ago
LICENCE.txt initial commit 6 days ago
README.md initial commit 6 days ago
start_challenge.sh initial commit 6 days ago
truffle.js fixes indentations and removes redundant code a day ago
README.md
https://github.com/nccgroup/GOATCasino/pulls

the decentralized application security project:
www.dasp.co

follow us on
twitter.com/cryptodavidw
twitter.com/mah3mm

(we **work** here)