

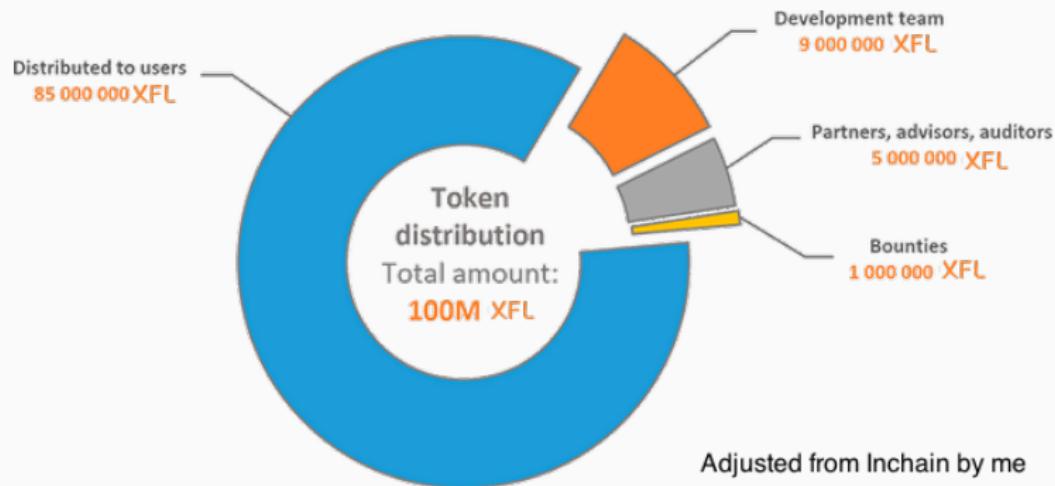
Blockchain, Smart Contracts, ClojureScript, Fleet

Erwin Rooijakkers

12-07-2017

Blockchain

Initial Coin Offering (ICO) announcement



Blockchain?

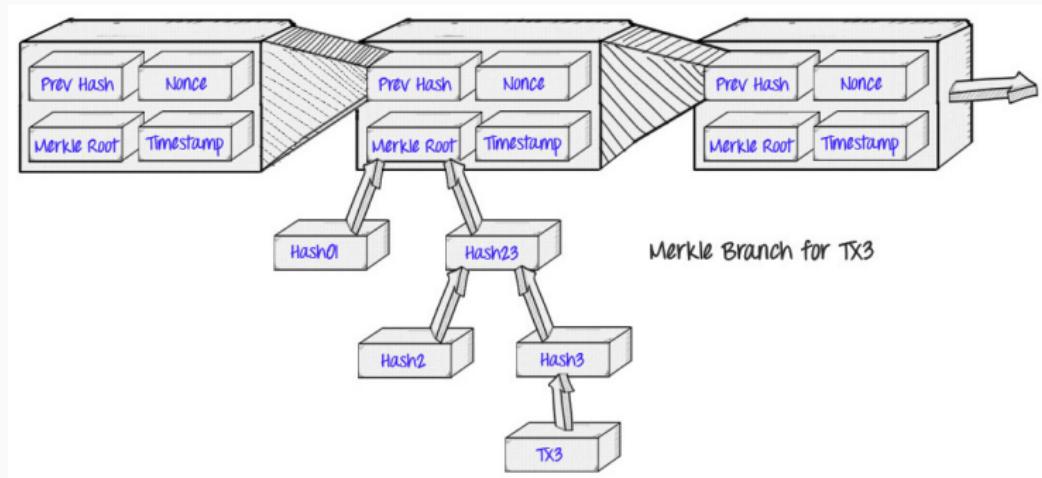
"A *shared*, programmable, cryptographically secure and therefore trusted ledger *which no single user controls* and *which can be inspected by everyone*."

– Klaus Schwab (Chairman World Economic Forum)

Four pillars

- Cryptographic Tokens and Addresses
- P2P Networking
- Consensus Formation Algorithm
- Virtual Machine

This.



Source: <https://blog.ethereum.org/2015/11/15/merkling-in-ethereum/>

Consensus

- Proof of Work (**PoW**)
- Proof of Stake (**PoS**)

Smart Contracts

DECENTRALISED SMART CONTRACTS



Source: ChainThat YouTube video

Model

- Stateless webservices
- Contract-oriented programming
- Gas fees

Stateless webservices



Source: ETH news

Stateless web services

A screenshot of a Reddit post from the self.ethereum subreddit. The post title is "Can ethereum be compared to aWS Lambda?". It has 2 upvotes and was submitted 11 months ago by user luandro. The post content discusses the similarities between Ethereum's smart contracts and AWS Lambda. A comment from user RaptorXP is highlighted with a red box, stating: "You can compare it to a slow, crippled and extremely expensive version of AWS Lambda." The background of the screenshot shows a blurred image of a modern building with a curved glass facade.

Can ethereum be compared to aWS Lambda? self.ethereum

Submitted 11 months ago by [luandro](#)

I've been trying to understand Ethereum and the concepts of Dapps, DAOs and contracts for the past week. I've understood that is has many uses, and one of them is in processing contracts. So in the case of creating a simple app, IPFS would be AWS S3, and could Ethereum be compared to AWS Lamda, which is responsible for resolving function calls?

[4 comments](#) [share](#)

[all 4 comments](#)
sorted by: top (suggested)

[–] [RaptorXP](#) 7 points 11 months ago
You can compare it to a slow, crippled and extremely expensive version of AWS Lambda.
[permalink](#) [embed](#)

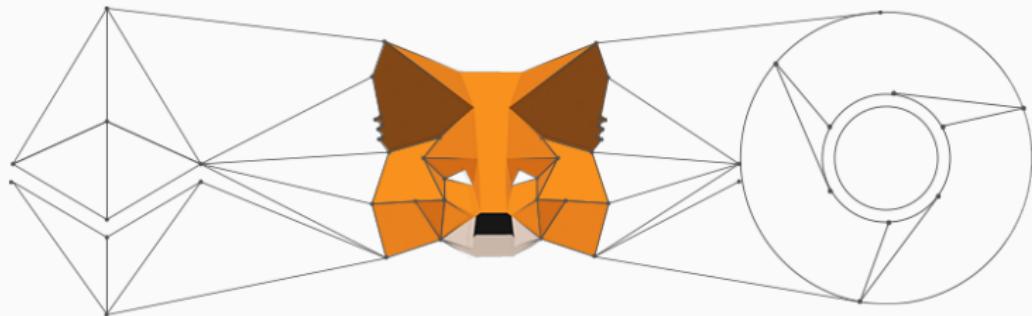
Contract-oriented programming

```
contract HelloSayerFactory {  
  
    function create() returns (address) {  
        return address(new HelloSayer());  
    }  
  
    function delete(address addr){  
        HelloSayer(addr).remove();  
    }  
}
```

Tools

- geth: (Go Ethereum) **cli** for running full Ethereum node, exposes RPC
- web3.js: Ethereum compatible **JavaScript API** which implements the Generic **JSON RPC spec**
- solc: JavaScript bindings for Solidity compiler (creates **ABI** and **BIN**)

MetaMask (also: Mist; Parity)



Fleet

Why ClojureScript + blockchain



Code inspiration and big help

- <https://medium.com/@matus.lestan>
- <https://github.com/district0x/ethlance>
- <https://ethlance.com/#/job/128>



erwin 8:40 AM

Hey I have a question related to Ethereum development

When I watch tutorials everybody is using Truffle

But in ClojureScript we don't need that?



madvas 8:41 AM

yeah, I don't use that. I've build my own tooling and with REPL you have much more power than with Truffle

8:45 ★ I started to build library which will be basis for all our dapps <https://github.com/district0x/d0xINFRA-frontend>
It's pretty opinionated, and not documented yet, but there's tons of code that can be reused in any dapp. I don't yet, but you can use it as inspiration. Especially events.cljs file.

A New “Perspective” for Circular Economy Finance

Tim Wright & Elisa Achterberg

Executive Summary

Circular Economy

exceptional opportunity

heavily in “design for failure”

current linear economy

invest

selling “almost-broken” products rather than new sustainable products
creating waste.

new way of doing business

Cornerstone is the shift *from ownership to use*

Linear economy designs for failure



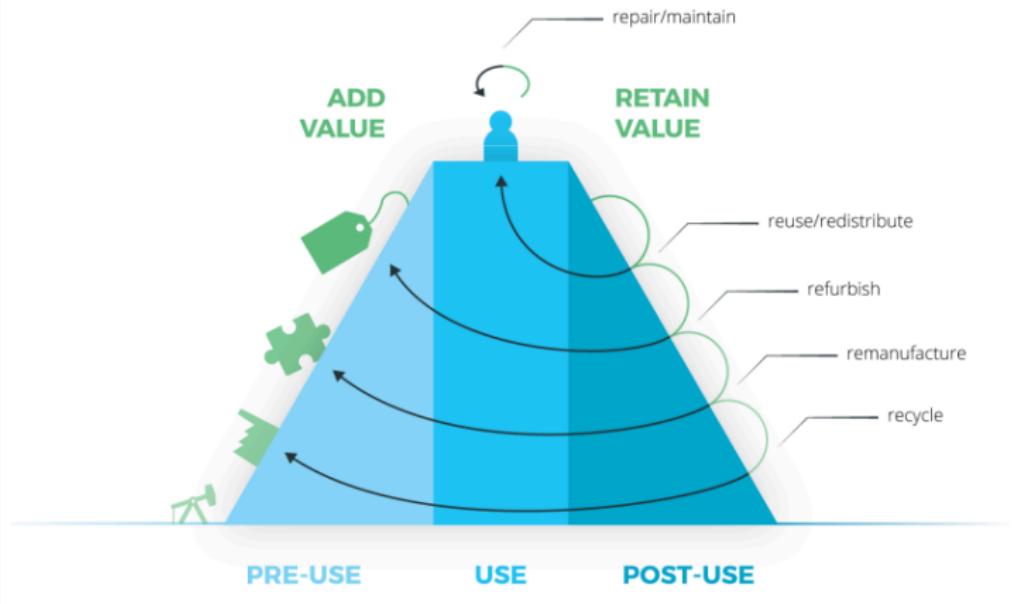
Source: Politico Europe

Smart assets

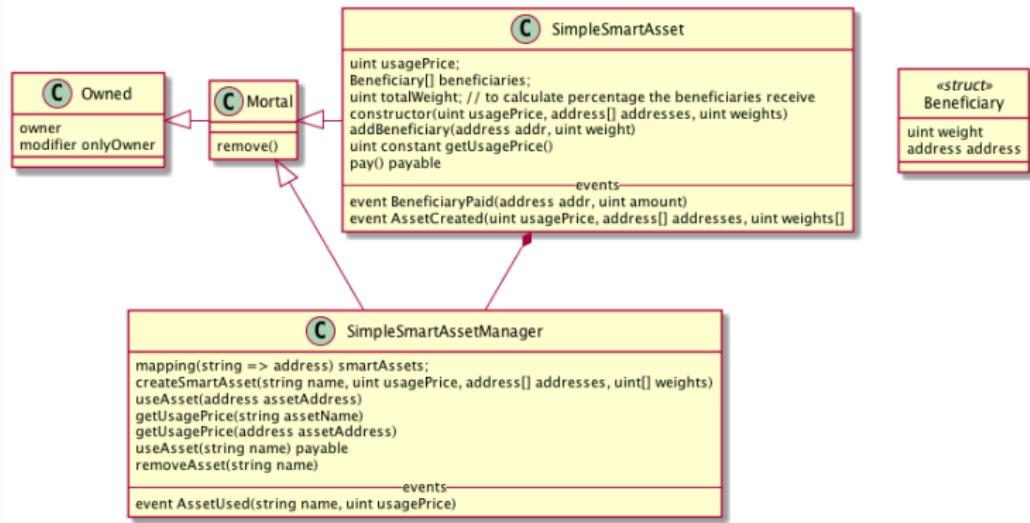
- When they are used, **smart assets** (a fleet of assets) pay parties involved in value chain (involved with design, commodities, creation, maintenance, et cetera)

Shift from ownership to use

CIRCULAR ECONOMY



Contract design



App design

- Reagent atom for **application state**:
 - Frontend state: (beneficiaries)
 - Blockchain state: Contract instance, abi and bin
 - Address of user
 - **queries.cljs** to interact with ratom
 - **views.cljs** reacts to its changes $v = f(S)$, or something
- **blockchain.cljs** *deploys or retrieves* contracts
 - Depending on network of web3 object (Ropsten or local)
- **smart_asset_manager.cljs** calls methods on instances
- Both use the **web3-cljs** web3.js wrapper
- Setup local blockchain for development with **fleet.el**
- TODO: **tests...**, **IPFS** deployment

Fleet demo
