CREATION OF A CRYPTO CURRENCY

CRYPTO CRASH COURSE

PART I

WHAT IS A CRYPTO CURRENCY

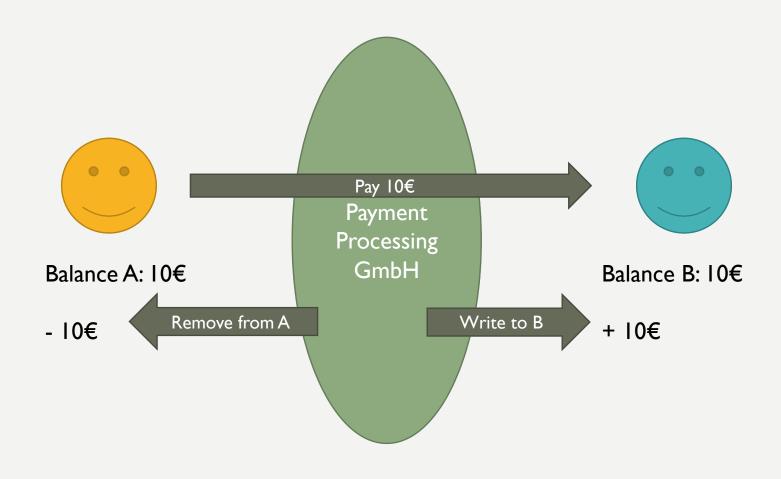
- Digital
- Based on cryptography
- Decentralized
- Public
- Exists in a distributed system known as a "blockchain"

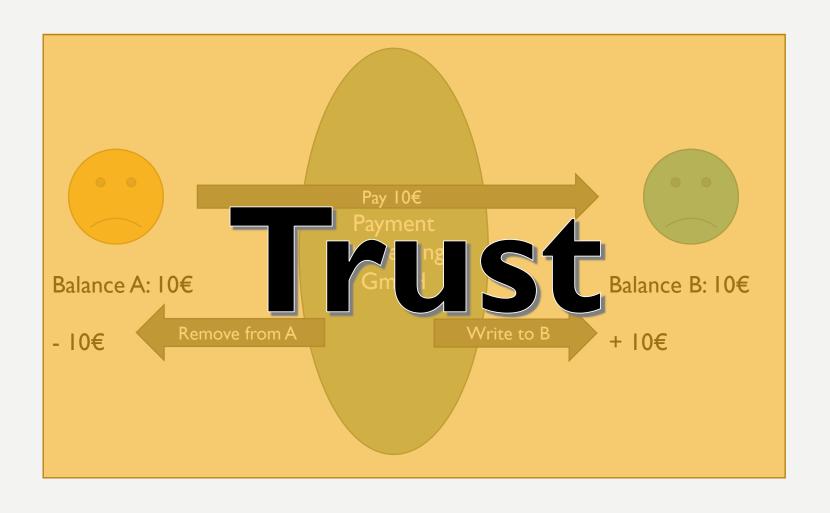
10€ of ...









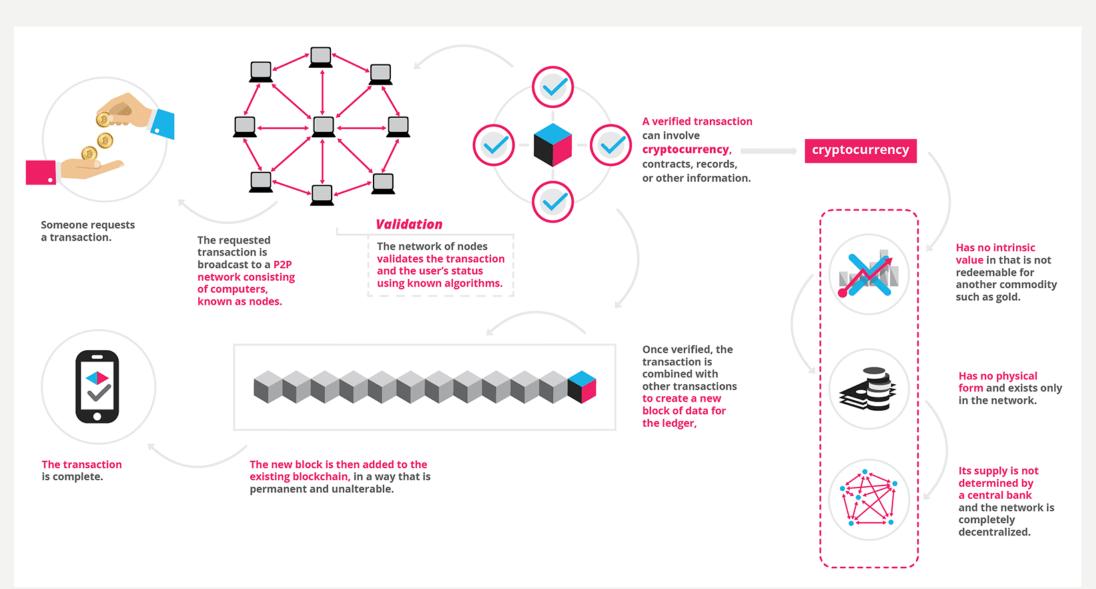


SOLUTION: THE BLOCKCHAIN

A blockchain can be defined as a persistent, transparent, public, append-only ledger.

- You can add but not change previous data
- Distributed network of actors
- Peer to peer
- Mechanism to reach a consensus between them
- No trust required between actors. Actors only have to trust the mechanism

SOLUTION: THE BLOCKCHAIN



EXAMPLES OF CRYPTO CURRENCIES

	Bitcoin	Ether
	#	
Created	2009	2015
Market cap	Over 10 billion	Under 1 billioin
Popular support	High	Low
Blockchain	Proof of work	Proof of work
Scalable	Not at the moment	Yes
Mining	ASIC miners	GPUs
Supply	21 million	18 million
Development	over 100 contributers	Small core team
Hash rate	1.8 ExaHash	3 TeraHash
Initial distribution	Mining	ICO







NEW COIN, THE THEORY

PART 2

WHITEPAPERS

- Bitcoin: https://bitcoin.org/bitcoin.pdf
- Ethereum: https://github.com/ethereum/wiki/White-Paper
- Ripple consensus algorithm: https://ripple.com/files/ripple_consensus_whitepaper.pdf
- Ripple Solution Guide: https://ripple.com/files/ripple_solutions_guide.pdf

WE NEED A BLOCKCHAIN

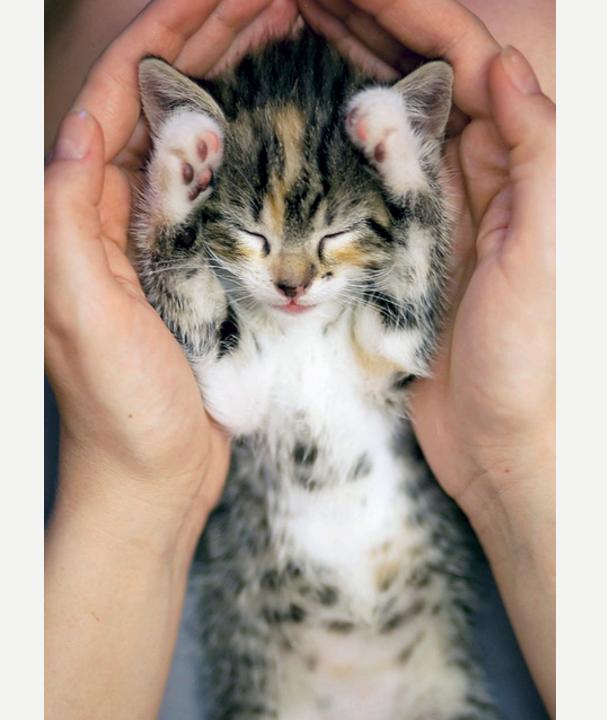
- Fork existing project
- Create smart contracts on top of existing blockchains
- Start from scratch

SMART CONTRACT

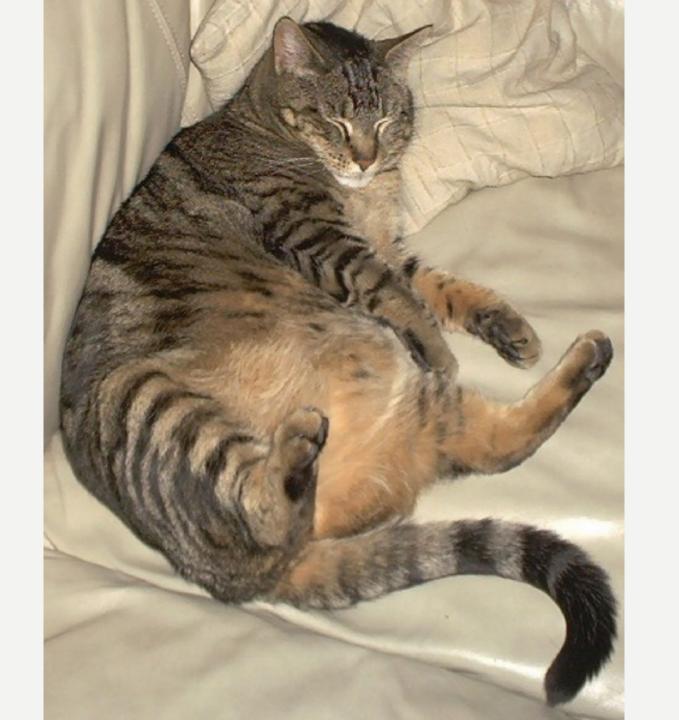
- Program living in a blockchain
- Allow performance of transaction without third parties

SMART CONTRACT

```
pragma solidity ^0.4.18;
contract MyToken {
    mapping (address => uint256) public balanceOf;
    string public name;
    string public symbol;
   uint8 public decimals;
    event Transfer(address indexed from, address indexed to, uint256 value);
    function MyToken(uint256 initialSupply, string tokenName, string tokenSymbol, uint8 decimalUnits) public {
        balanceOf[msg.sender] = initialSupply;
       name = tokenName;
        symbol = tokenSymbol;
        decimals = decimalUnits;
    function transfer(address to, uint256 value) public {
      require(balanceOf[msg.sender] >= value && balanceOf[to] + value >= balanceOf[to]);
      balanceOf[msg.sender] -= value;
      balanceOf[to] += value;
     Transfer(msg.sender, to, value);
    function changeName(string newName) {
     name = newName;
```







OPTIOCOIN, GENESIS

PART 3



DESIGN - LET'S KEEP IT SIMPLE

- Basic info: name, symbol, ...
- Balances
- Can transfer money