Cryptocurrency: A Technological Perspective

Dr. Bambang Purnomosidi D. P.
STMIK AKAKOM

Agenda

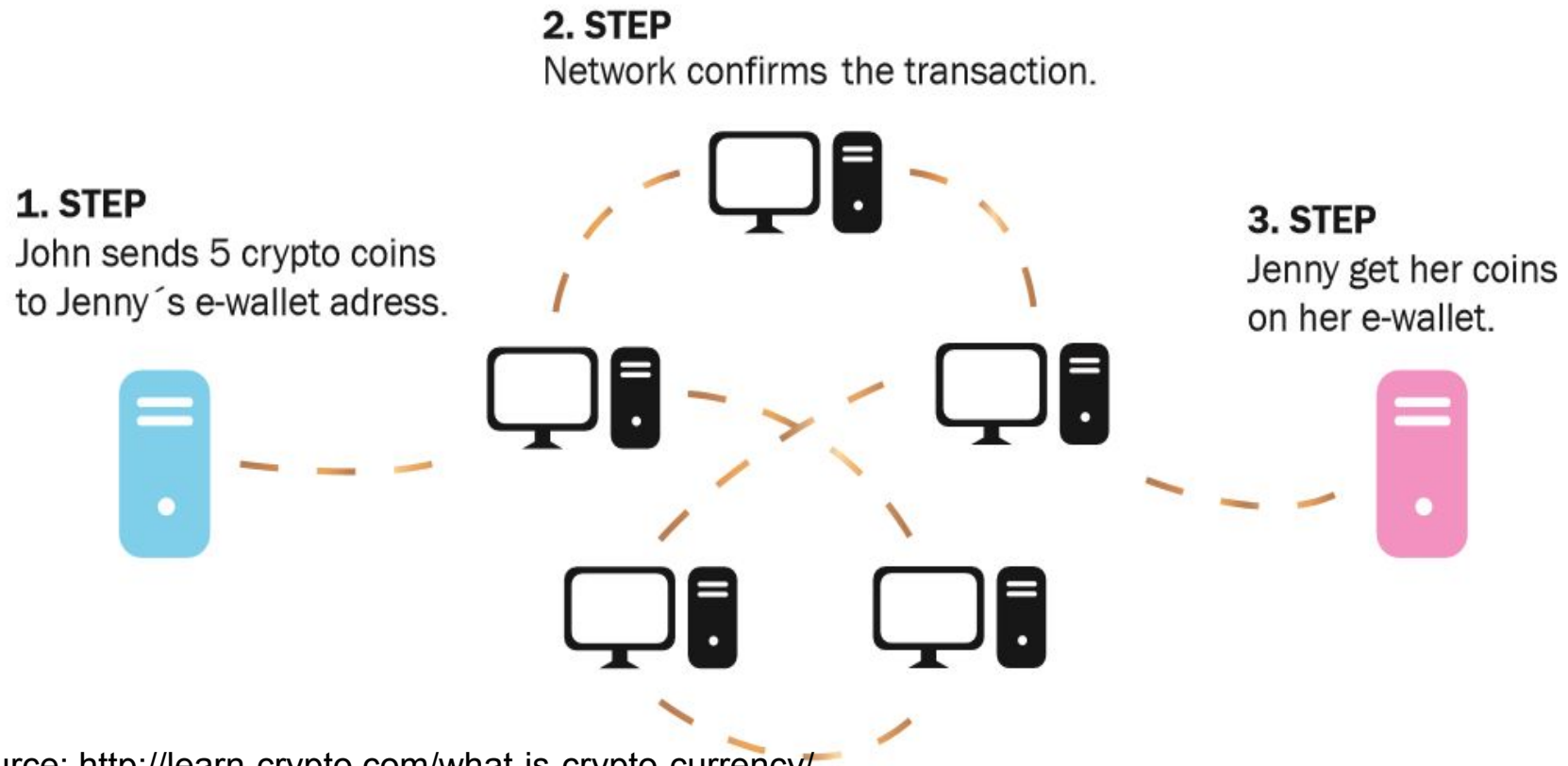
1. Understanding Cryptocurrency
    a. What is Cryptocurrency?
    b. How Does Cryptocurrency Work?
    c. List of Cryptocurrencies
    d. Cryptocurrency Mining
2. Technology Behind Cryptocurrency: Blockchain and DApp
    a. What Is Blockchain?
    b. How Does Blockchain Work?
    c. Blockchain Usage
    d. Types of Blockchain
    e. Understanding DApp
    f. DApp Architecture
    g. Ecosystem for DApp Development

Understanding Cryptocurrency

What is Cryptocurrency?

- A digital asset, a type of digital currency.
- Transferred from peer-to-peer, no middlemen (no bank, no financial institution).
- Just as currency (money), its main function is as medium of exchange.
- Uses cryptography to:
  - secure transaction
  - control the creation of additional units
  - verify the transfer of assets
- Decentralized, as opposed to central banking system, made possible by blockchain
- First cryptocurrency: bitcoin (2009). Other than bitcoin, they are called **altcoin**.

How Does Cryptocurrency Work?



source: http://learn-crypto.com/what-is-crypto-currency/

List of Cryptocurrencies

1. First cryptocurrency: **Bitcoin** (2009)
2. **Altcoin**: coin other than Botcoin: litecoin, ether, ada, etc
3. Since Bitcoin, many cryptocurrencies emerge. As of April 11, 2018, there are Litecoin, Ether, and more (1566 cryptocirrency). See: https://coinmarketcap.com/all/views/all/
4. Why are there so many cryptocurrencies?
    a. Because that is possible.
    b. One may use already available blockchain system (such as Ethereum) to create cryptocurrency.

Cryptocurrency Mining

- Mining: a validation of transactions
- It’s a process of adding transaction records to cryptocurrency public ledger of past transactions.
- See https://www.buybitcoinworldwide.com/mining/

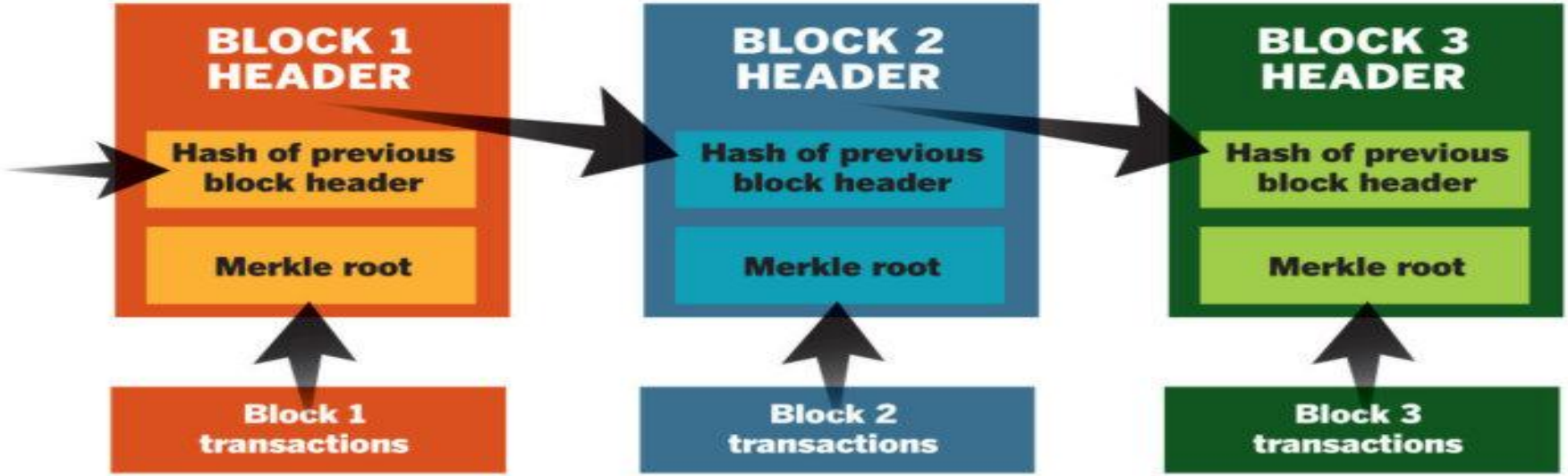
Technology Behind Cryptocurrency: Blockchain and DApp

What Is Blockchain?

- “We can define the blockchain as a system that allows a group of connected computers to maintain a single updated and secure ledger.” (Michele D’Aliessi, 2016).
- “A blockchain is a decentralized and distributed digital ledger that is used to record transactions across many computers so that the record cannot be altered retroactively without the alteration of all subsequent blocks and the collusion of the network”. (Stephen Armstrong, 2018).
- Preliminary research: Cryptographically secured chain of blocks (Stuart Haber dan W. Scott Stornetta, 1991), Ross J. Anderson (1996), Michael Doyle (1997), Bruce Schneier and John Kelsey (1998), Nick Szabo (1998, bit gold), Stefan Konst (2000).
- Satoshi Nakamoto (2008): Bitcoin

How does blockchain work?

With blockchain technology, each page in a ledger of transactions forms a block. That block has an impact on the next block or page through cryptographic hashing. In other words, when a block is completed, it creates a unique secure code, which ties into the next page or block, creating a chain of blocks, or blockchain.



SIMPLIFIED BITCOIN BLOCK CHAIN

source: https://www.cio.com/article/3055847/security/what-is-blockchain-and-how-does-it-work.html

Blockchain usage

1. Distributed ledger for cryptocurrency (FinTech)
2. Land / Property registeration (Sweden and Georgia)
3. Decentralized Library (Alexandria project)
4. Govenment and National Currency: e-Dinar (Tunisia) and eCFA (Senegal)
5. Digital assets tracking
6. Identity
7. Digital Voting
8. Distributed Storage
9. … and many more.

Types of Blockchain

1. **Private blockchains**: a fully private blockchain is a blockchain where write permissions are kept centralized to one organization.
2. **Consortium blockchains**: a consortium blockchain is a blockchain where the consensus process is controlled by a pre-selected set of nodes; for example, one might imagine a consortium of 15 financial institutions, each of which operates a node and of which 10 must sign every block in order for the block to be valid.
3. **Public blockchains**: a public blockchain is a blockchain that anyone in the world can read, anyone in the world can send transactions to and expect to see them included if they are valid, and anyone in the world can participate in the consensus process.

Understanding DApp

- DApp is an abbreviated form for decentralized application. A DApp has its backend code running on a decentralized peer-to-peer network. Contrast this with an app where the backend code is running on centralized servers.

DApp Architecture

- Just as other common pattern in software development, we also need to separate some components in DApp.
- There is front-end (could be anything: Web, mobile, etc.
- There is also back-end code which access immutable database / distributed ledger.

Ecosystem for DApp Development

- Blockchain Platform:
  - Public?
  - Private?
  - Consortium?
- Development Tools
- See https://www.stateofthedapps.com/ for inspiration.

DApp Development for Public Blockchain (Ethereum)

- The development process of a ĐApp often entails a Whitepaper and a working prototype, a token sale, an initial coin offering (ICO), its implementation and launch.
- From the workflow, developer tasks are:
  - Create token
  - Create application
  - Deploy application to EVM

Ethereum DApp Development Without Framework

- Never use public blockchain for test.
- What you have to learn:
  - Smart contract and how to use Solidity programming language. Get Solidity compiler (https://github.com/ethereum/solidity/releases). Some Linux distros provide official package (ex: Arch Linux pacman: **community/solidity 0.4.18-1**)
  - How to deploy your smart contract to EVM
  - Documentation for Solidity: http://solidity.readthedocs.io/en/latest/
- Development environment: Atom, Emacs, Vim, Intellij, VSCode, See Awesome Solidity (https://github.com/bkrem/awesome-solidity)
- Step:
  - Build everything in local computer. Use **testrpc** to mimick blockchain (https://github.com/ethereumjs/testrpc).
  - Test on the staging blockchain (Ether and any other resource fees are fake).: Ropsten. See https://github.com/ethereum/ropsten and https://ropsten.etherscan.io/ . Also *--morden* when run Ethereum clients.
  - Real Ethereum deployment. Mainnet - Homestead.

DApp Development With Framework in Ethereum

- Truffle Framework (http://truffleframework.com/) - A Swiss army knife for smart contract development and deployment.
- DApp (https://dapp.readthedocs.io/en/latest/) - simple command line tool for smart contract development for package management, source code building, unit testing, simple contract deployments.

Deployment in Ethereum

- In deployment, understanding **Gas** is important.
- Transactions on the Ethereum network require fees in the form of **gas**. The amount of gas depends on the amount of computation required to complete the transaction.
- Estimating transaction costs: http://ethdocs.org/en/latest/contracts-and-transactions/account-types-gas-and-transactions.html

DApp Development for Private Blockchain (Multichain)

- https://www.multichain.com
- Source code: https://github.com/MultiChain/multichain
- Steps:
  - Download **multichain** from https://www.multichain.com/download-install/
  - There are 4 *binary executables*: multichain-cli, multichaind, multichaind-cold, multichain-util
  - Create blockchain using multichain-util, initializing blockchain, also can be used to connect to existing blockchain with multichaind, manipulate blockchain using multichain-cli
  - Explore blockchain using multichain explorer: https://github.com/MultiChain/multichain-explorer (Python)
  - See https://github.com/MultiChain/multichain-web-demo if you need an example of multichain blockchain with PHP and all front-end Web

DApp Development for Consortium Blockchain (Corda)

- https://corda.net
- Source code: https://github.com/corda/corda
- Build with Kotlin (https://kotlinlang.org)
- Needs JDK (also Kotlin if you want to use Kotlin), JDK 8 is the only supported JDK at the moment.
- Also need to understand how Gradle works. See https://gradle.org.
- See example at https://github.com/corda/cordapp-example

- A Corda network is made up of nodes running Corda and CorDapps
- The network is permissioned, with access controlled by a doorman
- Communication between nodes is point-to-point, instead of relying on global broadcasts
- CorDapps (Corda Distributed Applications) are distributed applications that run on the Corda platform. The goal of a CorDapp is to allow nodes to reach agreement on updates to the ledger. They achieve this goal by defining flows that Corda node owners can invoke through RPC calls

The End