Programming Decentralized Application

Dr. Bambang Purnomosidi D. P.
VP Curriculum at Refactory
Faculty at STMIK AKAKOM

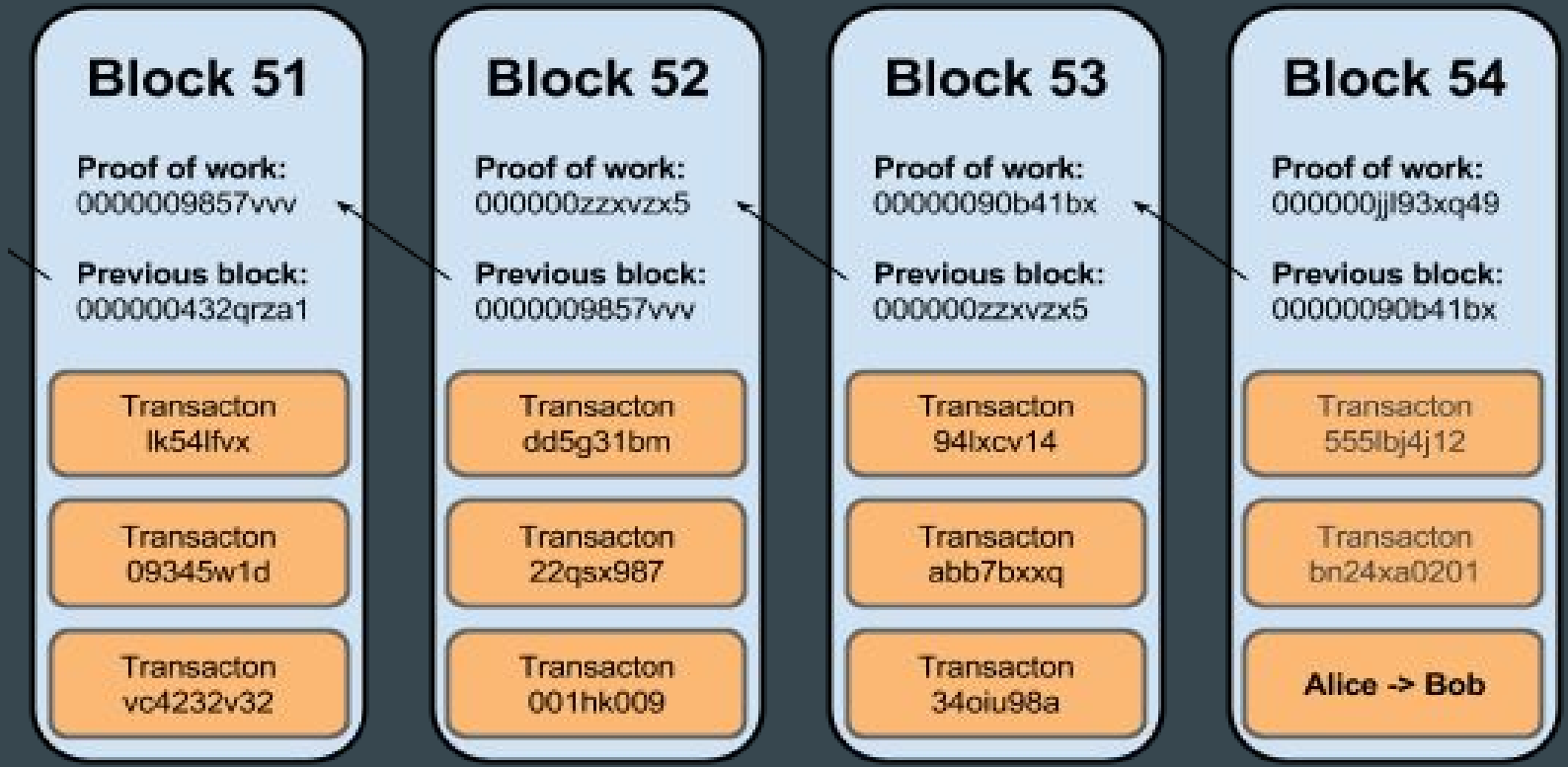
Road to DevSummit - STMIK AKAKOM - 28 Oktober 2017

Agenda

1. What is Blockchain?
2. What is Decentralized Application (DApp)?
3. What do I need to program a DApp?
4. Enter Ethereum
5. Ethereum Clients
6. Private or Public BlockChain?
7. Overview of Development Environment in Ethereum
8. Development Without Framework
9. Development With Framework
10. Deployment

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



Source: http://dataconomy.com/2015/10/wtf-is-the-blockchain-a-guide-for-total-beginners/

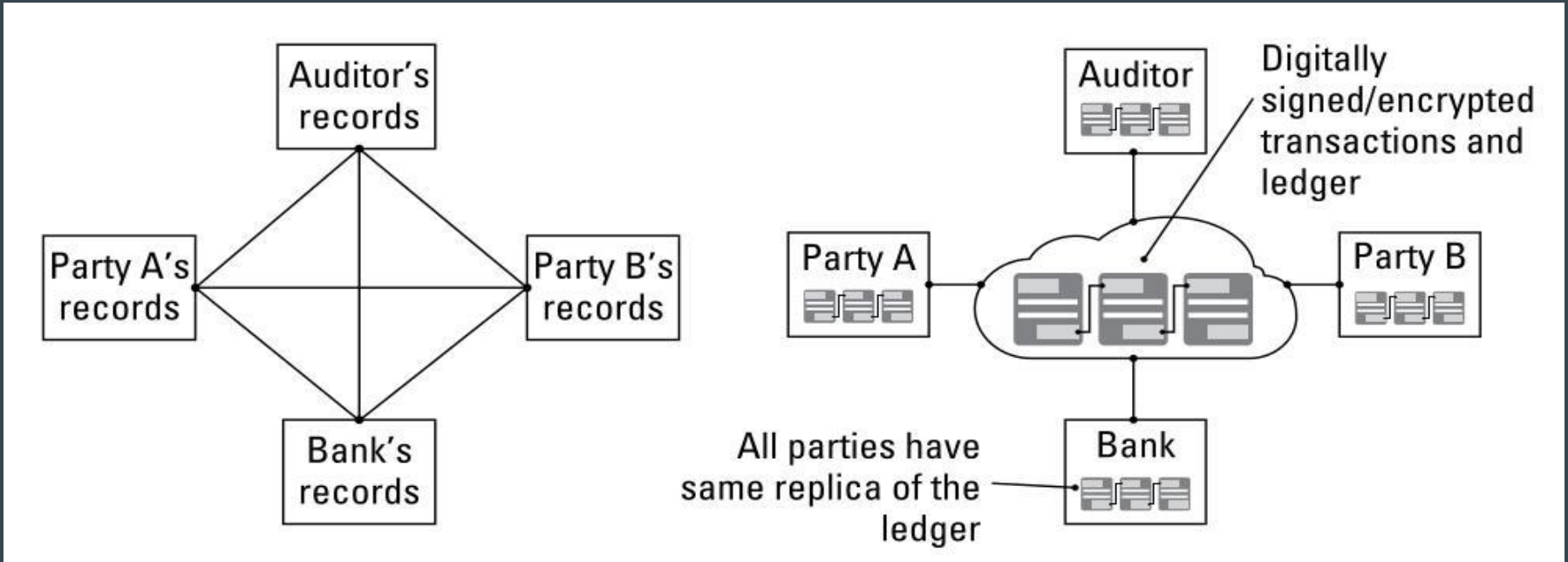
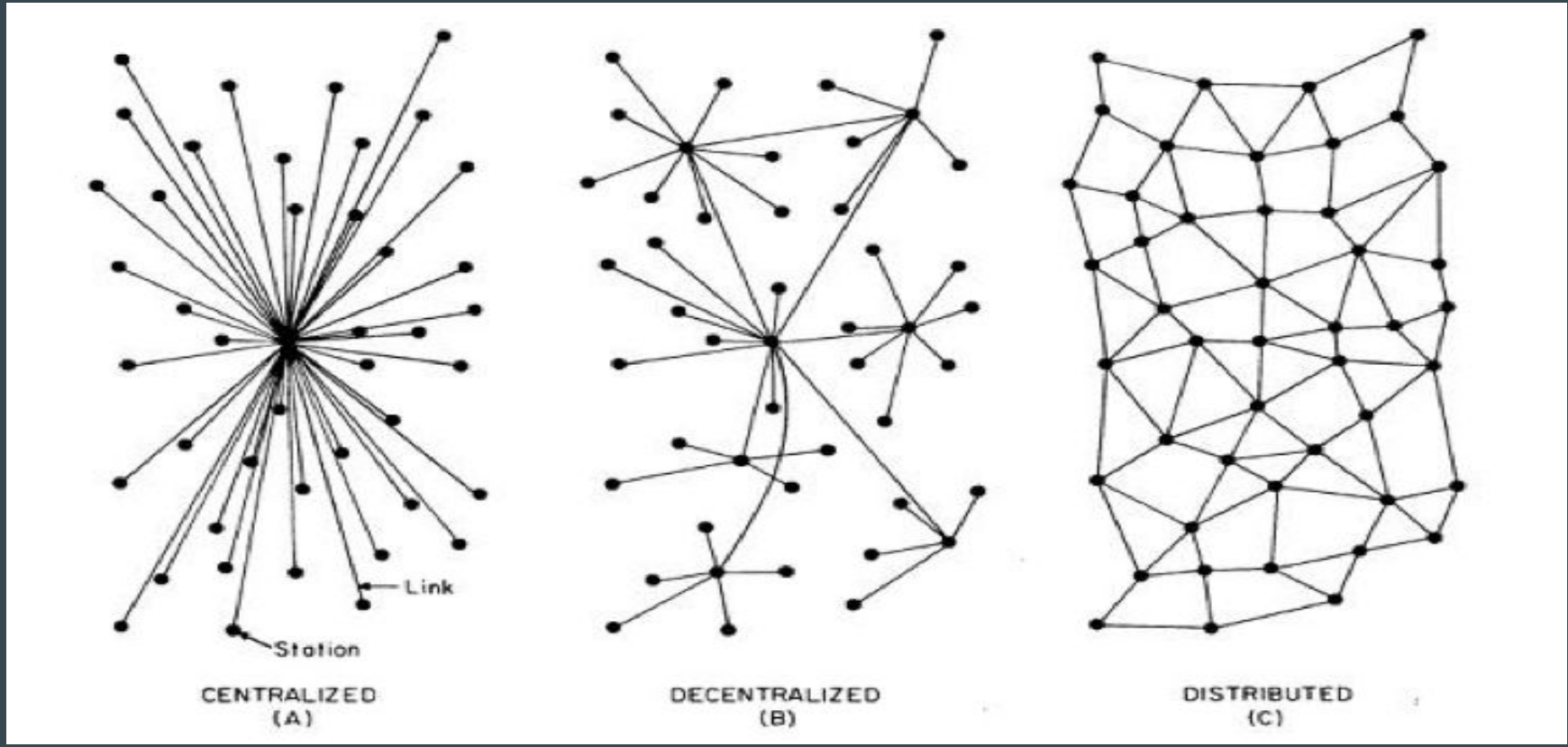


Image was taken from "Blockchain for Dummies - IBM"

What is Decentralized Application (DApp)?



What Do I Need to Program A DApp?

- Blockchain Platform:
  - Ethereum
  - Tezos
  - Corda
  - Exonum
  - Hyperledger Project
- Development Tools
- See https://www.stateofthedapps.com/ for inspiration.

Enter Ethereum

- Ethereum is a decentralized platform that runs smart contracts: applications that run exactly as programmed without any possibility of downtime, censorship, fraud or third party interference.
- It has **ether** as its cryptocurrency.
- Provides **smart contracts** for DApp development
- Provides **Ethereum Virtual Machine**. EVM is an environment where arbitrary code of smart contracts and other operations can be executed.
- There are some high level language for DApp development:
  - Solidity
  - Serpent
  - LLL

Ethereum Clients

- Ethereum client refers to any node able to parse and verify the blockchain, its smart contracts and everything related. It also allows you/provides interfaces to create transactions and mine blocks which is the key for any blockchain interaction.
- Official Reference Implementation:
  - Go: go-ethereum (geth):: https://geth.ethereum.org/
  - C++: cpp-ethereum (eth): http://www.ethdocs.org/en/latest/ethereum-clients/cpp-ethereum/
  - Python (pyethapp): https://github.com/ethereum/pyethapp
- 3rd party:
  - Rust: parity: https://parity.io
  - Java: https://github.com/ethereum/ethereumj
  - JS: https://github.com/ethereumjs/ethereumjs-vm
  - Ruby: https://github.com/janx/ruby-ethereum

Private, Consortium, or Public BlockChain?

- **Private blockchains**: a fully private blockchain is a blockchain where write permissions are kept centralized to one organization.
- **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.
- **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.

Overview of Development Environment / Workflow in 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

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.

Development With Framework

- 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 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

The End