Decentralized Applications

Saravanan Vijayakumaran
sarva@ee.iitb.ac.in

Department of Electrical Engineering
Indian Institute of Technology Bombay

August 22, 2019

DApps

- Applications that let users own their data and run without a single centralized operator (Source: `https://app.co/faq`)
- Decentralized vs Distributed
  - Distributed does not mean decentralized
  - A single entity could be controlling a distributed system
- Pros
  - Uncensorable
  - Transparency, Privacy (DApps are typically open source)
- Cons
  - Usability (slow, error-prone)
  - Difficult to build and/or maintain

> *P2P systems are hard. The only thing harder than a distributed system is a distributed system you don't control. A system that will attack you, and that is running stuff you don't want it to run.*
>
> Juan Benet

- Pre-Bitcoin examples
  - Email (if everyone doesn't use GMail)
  - BitTorrent

DApp Frameworks

- Permissionless
  - Ethereum
  - Blockstack
  - IPFS (libp2p)
  - . . . and many more
- Permissioned
  - Quorum
  - Corda
  - Hyperledger Fabric
  - . . . and many more
- DApp directories
  - `https://app.co/`
  - `https://www.stateofthedapps.com/`

Ethereum

Ethereum

- A blockchain platform for building decentralized applications
  - Application code and state is stored on a blockchain
- Two types of transactions
  - Contract creation
  - Message calls
- Contract creation transactions create new contracts on the blockchain
- Message call transactions call methods in an existing contract
  - Input data to contract methods is specified

Storage Contract

```
    pragma solidity ^0.4.0;

    contract SimpleStorage {
        uint storedData;

        function set(uint x) public {
            storedData = x;
        }

        function get() public view returns (uint) {
            return storedData;
        }
    }
```

https://solidity.readthedocs.io/en/v0.4.24/introduction-to-smart-contracts.html#storage

Currency Example

```
pragma solidity ^0.4.7;

contract Coin {
    address public minter;
    mapping (address => uint) public balances;

    event Sent(address from, address to, uint amount);

    constructor() public {
        minter = msg.sender;
    }

    function mint(address receiver, uint amount) public {
        if (msg.sender != minter) return;
        balances[receiver] += amount;
    }

    function send(address receiver, uint amount) public {
        if (balances[msg.sender] < amount) return;
        balances[msg.sender] -= amount;
        balances[receiver] += amount;
        emit Sent(msg.sender, receiver, amount);
    }
}
```

Initial Coin Offerings

- Also called token sales
- Ethereum is the most popular platform for ICOs
  - Each ICO implements a ERC-20 token contract (link)
  - Investments in ICOs was about $7 billion in 2017
- Some notable ICOs
  - Basic Attention Token, May 2017, $35 million in 30 seconds
  - Kik, Sep 2017, $100 million
  - Filecoin, Jan 2018, $257 million
- Many of the ICO-funded projects have failed
- Used to execute “pump-and-dump” schemes

Ethereum DApp Examples

- CryptoKitties
  - Allows players to purchase, breed, and sell virtual cats
  - Each CryptoKitty is a non-fungible token using the ERC-721 standard
  - Game popularity caused network congestion in Dec 2017
  - The highest selling cat cost 246 ETH in Dec 2017 ($\approx$ $117,000)
- Fomo3D (`https://fomo3d.hostedwiki.co/`)
- Decentralized exchanges (`https://idex.market`)

Other DApp Examples

- Graphite Docs (`https://www.graphitedocs.com/about`)
  - Decentralized version of Google Docs
  - Why? Privacy, Censor resistance
  - Built using Blockstack
- Textile (`https://www.textile.photos/`)
  - Decentralized photo sharing built on IPFS
- Peerpad (`https://peerpad.net/`)
  - A P2P realtime collaborative editing tool built using IPFS
- Radicle (`http://radicle.xyz/`)
  - IPFS-based replacement for GitHub

Bitmessage

- Decentralized, encrypted, P2P communications protocol
- Released by Jonathan Warren in Nov 2012
- Downloads increased fivefold in June 2013 after news of NSA email surveillance
- Inspired by the Bitcoin protocol
  - Identities are hashes of public keys
  - Messages are broadcast over a network instead of blocks
  - Each message needs PoW attached (to prevent spam)
  - Messages live only for two days (by default)
- Source `https://github.com/Bitmessage/PyBitmessage`

Permissioned Blockchains

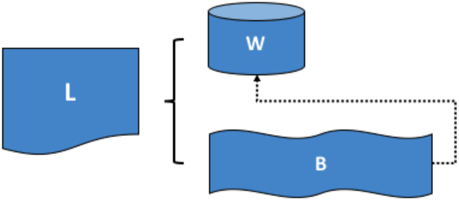
Permissioned Blockchains

- Private network of nodes which create and maintain a blockchain
- Proof-of-authority consensus is used instead of PoW
    - A valid block is one with a certain number of approvers
- **Motivation:** A shared ledger of facts about assets
- Popular frameworks
    - Hyperledger Fabric
    - Corda
    - Quorum

Hyperledger Fabric

- Hyperledger
  - Collaborative blockchain effort hosted by Linux Foundation
  - Mission: Create enterprise grade, open source distributed ledger frameworks
  - Launched in 2016
- Fabric
  - Permissioned distributed ledger framework with smart contracts
  - Originated in IBM in mid-2015 as Open Blockchain (OBC) project
  - Initial implementation completed in Dec 2015
  - IBM joined Hyperledger in Feb 2016 and donated OBC code

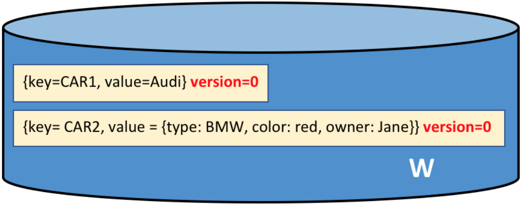
Ledger



L	Ledger
W	World State
B	Blockchain
L { B W	L comprises B and W
W ← B	B determines W

Image credit: `https://hyperledger-fabric.readthedocs.io/en/release-1.3/ledger/ledger.html`

World State


{key=CAR1, value=Audi} version=0
{key= CAR2, value = {type: BMW, color: red, owner: Jane}} version=0
W

W	Ledger world state
{key=**K**, value = **V** } **version=0**	A ledger state with **key=K**. It contains a set of facts expressed as a simple value, **V**. The state is at version 0.
{key=**K**, value = {**KV**} } **version=0**	A ledger state with **key=K**. It contains a set of facts expressed as a set of key-value pairs {**KV**}**.** The state is at version 0.

Image credit: `https://hyperledger-fabric.readthedocs.io/en/release-1.3/ledger/ledger.html`

Blockchain

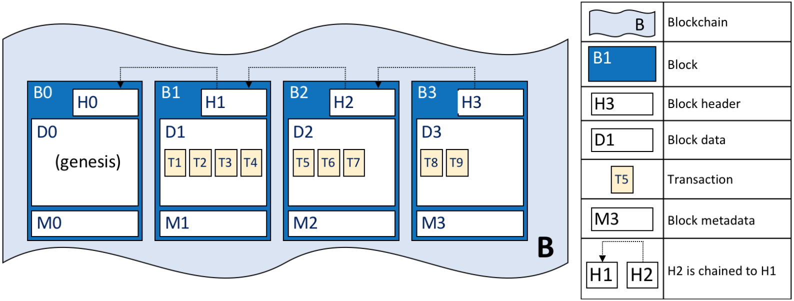
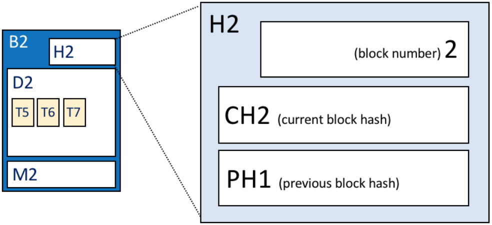


Image credit: `https://hyperledger-fabric.readthedocs.io/en/release-1.3/ledger/ledger.html`

Blocks



H2	Block header
2	Block number
CH2	Hash of current block transactions
PH1	Copy of hash from previous block
H2 ◁ V2	V2 is detailed view of H2

Image credit: `https://hyperledger-fabric.readthedocs.io/en/release-1.3/ledger/ledger.html`

Ledger Updates

Phase 1: Proposal

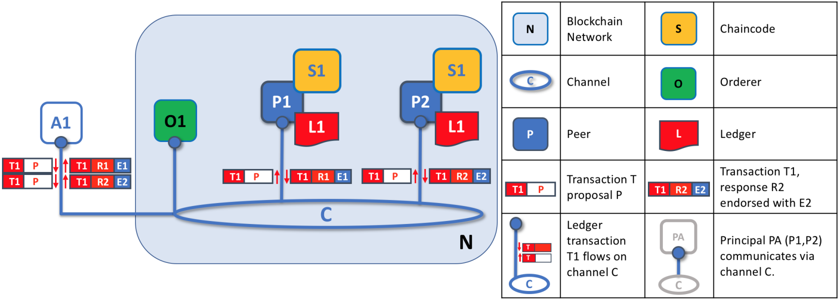
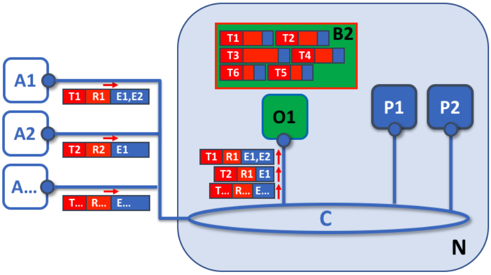


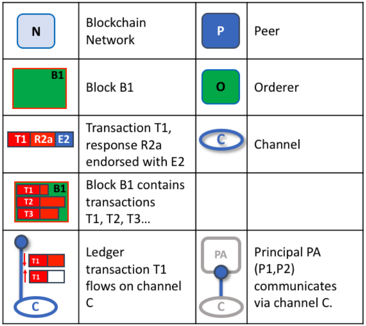
Image credit: `https://hyperledger-fabric.readthedocs.io/en/release-1.3/peers/peers.html`

- Application sends transaction proposal to some peers for endorsement
- Peers execute the transaction and append signatures endorsing the proposal
- Phase 1 ends when application receives sufficient responses

Ledger Updates

Phase 2: Packaging





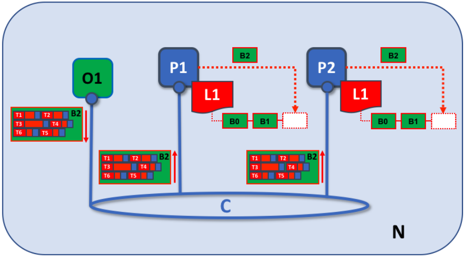
N	Blockchain Network	P	Peer
B1	Block B1	O	Orderer
T1 R2a E2	Transaction T1, response R2a endorsed with E2	C	Channel
T1 T2 T3 B1	Block B1 contains transactions T1, T2, T3...		
T1 T1 C	Ledger transaction T1 flows on channel C	PA C	Principal PA (P1,P2) communicates via channel C.

Image credit: `https://hyperledger-fabric.readthedocs.io/en/release-1.3/peers/peers.html`

- Endorsed transaction proposals are packaged into a block by the orderer

Ledger Updates

Phase 3: Validation



N	Blockchain Network	P	Peer
C	Channel	O	Orderer
L	Ledger	B	Block B
L1 B0 B1	Ledger L1 has blockchain with blocks B0, B1	T1 B1 T2 T3	Block B1 contains transactions T1, T2, T3...
B1 C	Block B1 flows on channel C	PA C	Principal PA (P1, P2) communicates via channel C.

Image credit: `https://hyperledger-fabric.readthedocs.io/en/release-1.3/peers/peers.html`

- Orderer distributes blocks to all peers
- Each peer checks that a block satisfies the organizational endorsement policy and applies to ledger

References

- Chapter 1 of *Decentralized Applications* by Siraj Raval, https://www.oreilly.com/library/view/decentralized-applications/9781491924532/ch01.html
- Ethereum Wikipedia Article https://en.wikipedia.org/wiki/Ethereum
- ICO Wikipedia Article https://en.wikipedia.org/wiki/Initial_coin_offering
- CryptoKitties Wikipedia Article https://en.wikipedia.org/wiki/CryptoKitties
- Solidity Documentation https://solidity.readthedocs.io
- Graphite Docs https://www.graphitedocs.com/about
- Textile https://www.textile.photos/
- Peerpad https://peerpad.net/
- Radicle http://radicle.xyz/
- Fabric Documentation https://hyperledger-fabric.readthedocs.io/