Using Blockchain technology to construct a Decentralized Application

Karthik Hubli

May 8th, 2018

Abstract

This paper discusses the research done as part of directed research under Prof. Haim Levkowitz in the Spring 2018 semester. The paper discusses, and outlines distributed and decentralized applications and compares them to traditional web applications. The paper contains brief explanations of the underlying technologies and try to create a sample architecture to build a simple application. As with any paper, it contains sections dedicated to the limitations and possible improvements of the systems. Since the topic chosen here is very vast and extensive, the paper cannot be treated as a comprehensive discussion, but will act as simplified outline to the technology. In this paper, I will be primarily be using 'Ethereum' and 'Bitcoin' as two example cases, though there are several other implementations available.

Introduction

The web as we know today, began as a decentralized system. The users or the content creators would own personal servers which would host their website or application. All the data pertaining to the application would flow only between the server and user. The data generated and collected during the process of interaction was owned collectively by the user and the service provider. But once there were third party service providers and web hosting services, the data was being collected and monetized by these few service providers. This created monopoly of few large service providers as they could offer better service for cheaper rates compared to owning and maintaining a personal server. This has altered the structure of the web from a network of millions of nodes connected to each other, to a network of few centralized hubs.

The above scenario, though provides great service at an affordable cost to us, it is harmful in the long run. The centralized system creates a single point of failure. If a larger service provider like Google or Heroku is compromised, several million applications using them may be rendered

useless and has the potential to be damaged irreversibly. But the biggest problem that is created by centralized web is monetization. Today, data is the most valued resource and centralized web architecture concentrates the data with a few large service providers. Small users who create the data do not own it or benefit from it. Instead, the service providers own and in certain cases extract data without the consent of user for their profit.

A decentralized application (D-App) solves all the above-mentioned problems. It removes the single point of failure by spreading or distributing the application over several nodes. Since, there is no central service provider, the data is owned and profited by all the users collectively.

Classic web 2.0 (Centralized) app vs Decentralized Application

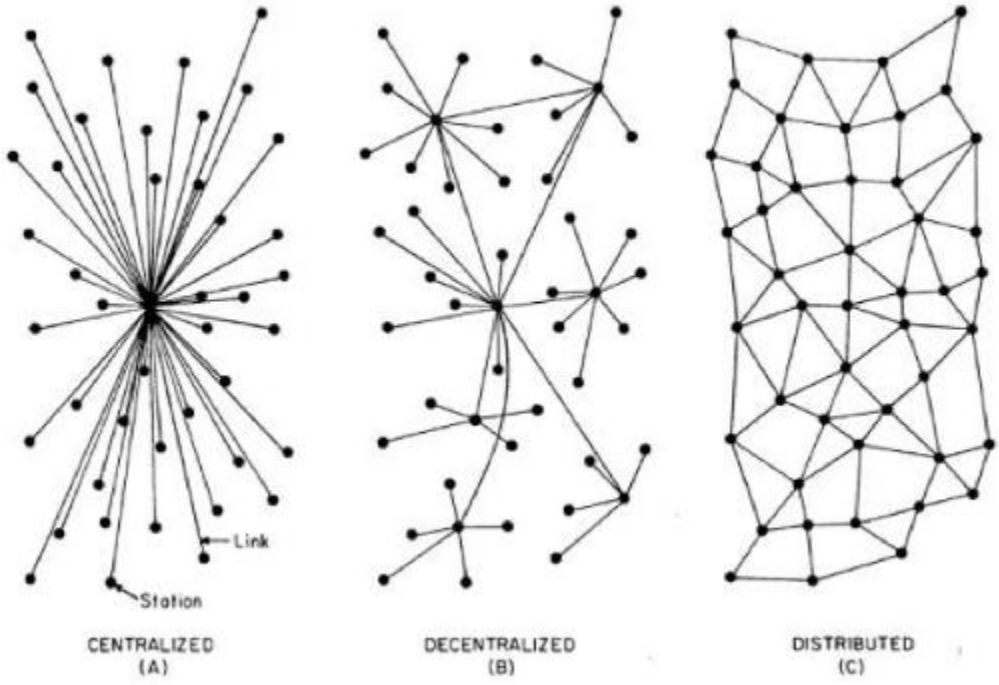

fig. 1

The above figure shows a high-level architecture of three different types of web architecture. Both centralized and decentralized systems can be distributed internally, i.e. the computation and work load can be spread across several nodes, but to the external user, it appears as one single node.

		Web 2.0	D-App
1	Computation units	Amazon EC2, Heroku	Ethereum, Truebit
2	File Storage	Amazon S3, MongoDB, SQL DB	IPFS, Storj
3	External Data	3rd Party API	Oracles (smart contracts)
4	Monetization	Ads, goods and merchandise	Token model
5	Payments	Credit cards, PayPal, G-Pay	Bitcoin, Ether,

Enabling Technologies

- The Blockchain: A Blockchain can be viewed as an immutable list. It is a ledger of records organized in 'blocks' that are linked together by cryptographic validation, mainly hashing. It is a digital storage couple with consensus from other participants. The key is that this ledger is neither stored in a centralized location nor managed by any single entity, hence it forms an important aspect of the decentralized and distributed network. The block validation system results in new transactions being added irreversibly and old transactions preserved forever for all to see, hence it is transparent and resilient. The network is almost fraud proof. For a fraud to be committed, at least 51% of the users or blocks must come to consensus and allow the fraud to take place. This is highly unlikely as its is virtually impossible for one person to have control over 51% or more of the blocks once the system has reached a stable size.

- Proof of Work: Proof of work is a mathematical algorithm used to arrive at a consensus in a Blockchain. In case of 'Ethereum' which we are using as an example, is implemented using a cryptographic hash function. Given an arbitrary piece of data, a user must find a second piece of data which, when combined with the first, produces a hash that has certain characteristics. Since hash is a one-way function, it is impossible to predict what second piece of data will produce the required hash. Only way to find the data is through brute force. Hence, when one finds the appropriate data, it acts as 'proof of work' done in computing the value. In most of the crypto currency system, a proof of work leads to addition of a block to the existing network and is rewarded with a 'crypto coin'.

System Architecture

For a system or application to be classified as Decentralized Application, it must satisfy all the below mentioned qualities.

Open Source: Ideally, it should be governed by autonomy and all changes must be decided by the consensus, or a majority, of its users. The code base should be available for scrutiny.

Decentralized: All records of the application's operation must be stored on a public and decentralized Blockchain to avoid pitfalls of centralization.

Incentivized: Validators of the Blockchain should be incentivized by rewarding them accordingly with cryptographic tokens.

Protocol/Algorithm: The application community must agree on a cryptographic algorithm to show proof of work.

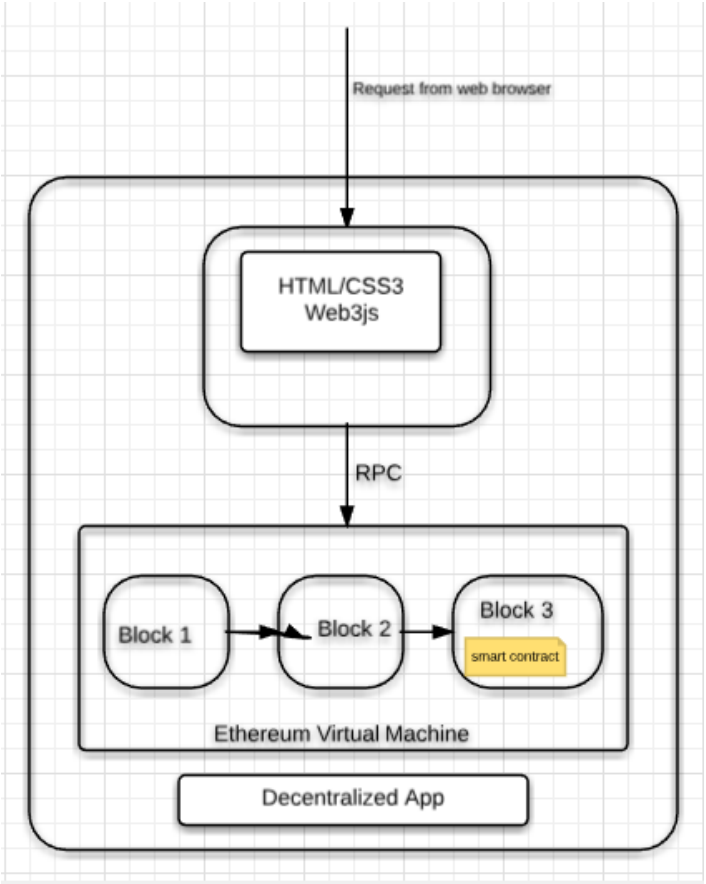

fig. 2

- Smart Contracts: A smart contract is a protocol used to digitally allow, verify, or enforce the execution of a process agreed by consensus. This process is known as 'contract'. Smart contracts conduct secure and credible transactions without third parties, thus eliminating the need for a centralized authority. Smart contracts are small chunks of code which reside on a block of Blockchain. The contract is distributed and copied multiple times between the nodes of the platform. When triggered, usually by a digital signature of the block, the contract is executed in accordance with the contract terms. The program checks the implementation of the commitments automatically. Oracles is Ethereum's implementation of a smart contract. The programming language of Oracles is 'Solidity'. The language is similar to Java in syntax and '.sol' is the file extension.
- Initial Coin Offering/Crypto Coin: Initial Coin Offering (ICO) is a way of revenue generation and providing a mean of payments for the service offered by the D-App. The D-App allocates crypto coin either by proof of work (mining) or in exchange for actual money. Theses crypto coins can be used as payments for the service offered by the D-App or can be traded for other monetary benefits. The quantity of coin is limited, thus creating a demand and value for it. If the service provided by the D-App is valued and highly desirable, the value of the crypto coin increases. Ether, Bitcoin, Litecoin, Iota etc. are few examples of crypto currency issued as reward for mining.
- Distributed Hash Table (IPFS): Blockchain can hold only small data as larger data sizes lead to longer time in validation through consensus. Hence, we use Distributed Hash Tables (DHT). A block of Blockchain only holds the address to the DHT and key to the particular file or hash-block. Interplanetary File System (IPFS) is the most popular DHT currently used for storage in D-App. IPFS is content addressable and peer-to-peer connected. This again is in line with decentralization as traditionally all data is stored in a central repository in web 2.0 applications. IPFS also duplicates data to increase redundancy in case of failure of one node. IPFS uses a 'Merkel Tree' structure to store and distribute files. This is a tree with every l eaf node is labelled with the hash of a data block and every non-leaf node is labelled with the cryptographic hash of the labels of its child nodes.

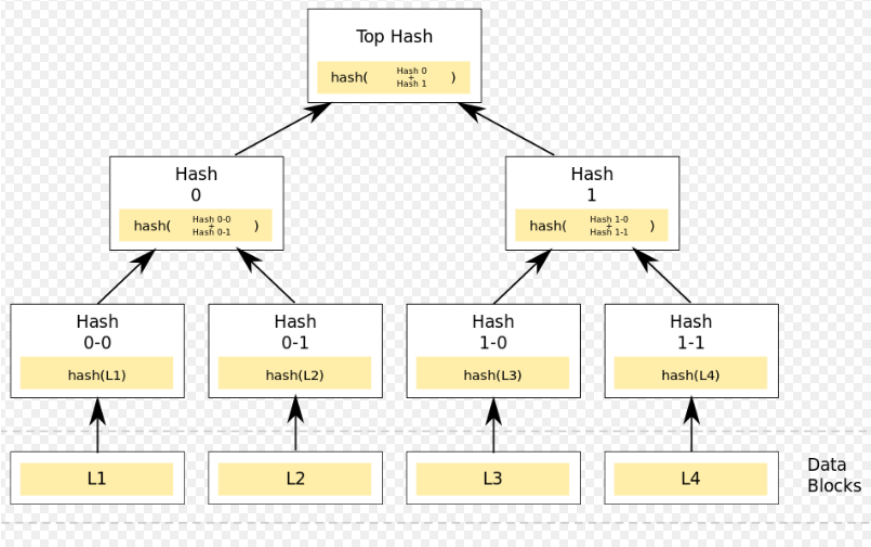

fig. 3

Limitations of the system

All the systems and technologies mentioned above are in an infant stage and not ready for large scale application. There is not a lot of community and industry support currently and all the implementations are either prototypes or in early stage of development. The architecture proposed above is paradigm shift the way we see the web today. The learning curve for such a technology is very steep and need a new way to envision the web. With current technology, the bulk of technical challenges are mitigated to the service providers which act as central hubs. But when it comes to D-Apps, all the infrastructure, technical and design issues are to be handled and solved by the D-App developer. Since all the technologies mentioned above are fairly new, the long-term reliability of the individual components as well as the entire system is not known. One of the major hindrance in implementation completely decentralized application is the delay in validation over the Blockchain. And the delay increases as the size of the Blockchain increases. Currently, in Bitcoin which is the most popular Blockchain, validation of a transaction can take several minutes. This delay is unacceptable in several day to day applications such as stock trading and banking. At the same time, decentralized validation being one of the main advantages of D-Apps, it cannot be bypassed.

Conclusion and Possible Upgrades

The adoption of D-Apps over traditional web 2.0 applications is negligible. But there definitely is a rise in the numbers recently. Once the technology is stable and simplified to implement, adoption would be lot faster and easier than it was to migrate to centralized hubs. As far as improving the speed of validation, there have been efforts made to migrate from a Blockchain architecture to Block Directed Acyclic Graph (Block-DAG). With Block-DAG architecture, the speed of validation will increase with increase in the size of network.

D-Apps may be the beginning of complete restructuring of the web as we know today. It will replace several traditional protocols like http, REST, socket, TCP with RPC, TPFS. This is the beginning of shift from web 2.0 architecture to web 3.0.

Reference

[1] Wright, Aaron and De Filippi, Primavera. *Decentralized Blockchain technology and the rise of lex cryptographia* (2015)

[2] Wood, Gavin. Ethereum: *A secure decentralised generalised transaction ledger.* (Ethereum Project Yellow Paper – 2014)

[3] Nakamoto, Satoshi. *Bitcoin: A peer-to-peer electronic cash system.* (2008)

[4] Benet, Juan. *IPFS-content addressed, versioned, P2P file system* (2014)

[5] Zhang, Fan and Cecchetti, Ethan and Croman, Kyle and Juels, Ari and Shi, Elaine. *Town crier: An authenticated data feed for smart contracts* (2016)

[6] Bhargavan, Karthikeyan and Delignat-Lavaud, Antoine and Fournet, C'edric and Gollamudi, Anitha and Gonthier, Georges and Kobeissi, Nadim and Kulatova, Natalia and Rastogi, Aseem and Sibut-Pinote, Thomas and Swamy, Nikhil and others. *Formal verification of smart contracts* (2016)

Image Source

fig. 1 - https://www.safaribooksonline.com/library/view/decentralized-applications/9781491924532/assets/dcap_0101.png
fig. 3 - https://upload.wikimedia.org/wikipedia/commons/thumb/9/95/Hash_Tree.svg/1024px-Hash_Tree.svg.png