Blockchain Storage

Tim Sigl

The first generation of P2P data networks started in 1999 with Napster and BitTorrent. With the first Blockchains new solutions for P2P data storage were found. The feasibility of a purely DLT-based data network is limited, due to scalability problems and limited on-chain storage. Therefore in 2014 with the introduction of the Interplanetary File System (IPFS) a third generation of P2P data networks started. BitTorrent works by the user having to first download a torrent file from a web server. The torrent file contains important meta information about the file. With the information of the torrent file the user contacts a tracker to receive a list of peers who own data packages of the requested file. While the user has not fully downloaded the file he/she acts as a leecher. In the process of leeching a file exchange technique called swarming is used seperating a file into fixed-sized pieces each of which is usually 256 KB. When a piece of data is downloaded other peers can also download the piece of data from the user. After the file is fully downloaded the user acts as a seeder and provides the data for other users.
IPFS works with content-based addressing, where the content is not addressed via a location (location-based addressing) but via its content. The main focus of IPFS is the storage of immutable data. The addressing changes when the content changes so users must be careful when wanting to update their data. Addressing mutable data can be done via an IPFS name resolution service. IPFS uses its own coin Filecoin to ensure file availability but works independently from Filecoin. IPFS objects (like files) are split into chunks or blocks with an ID and arranged into a merkle DAG. Exchange of blocks is handled by the Bitswap protocol where each node has a “want”, “have” and a “do not want” list. Nodes send requests with the IDs of their want list to their swarm (connected nodes). If no node has the requested data IPFS performs a hash table lookup and adds a node to the swarm that has the requested data. IPFS does not have any implicit mechanisms for repairing and maintaining files or ensuring redundancy. For storage guarantees the Filecoin exists which employs a storage and retrieval market. Also a user could deploy an IPFS cluster which is an additional overlay network for data redundancy. Swarm is a different P2P data sharing protocol developed by the Ethereum Foundation. Swarm ensures data availability by areas of responsibility. Other P2P data solutions concentrate on different aspects. Storj concentrates on high durability of data, low latency and high security for stored data. Arweave utilizes a blockchain-like structure called blockweave to provide a mechanism for permanent on-chain data storage. The authors of *When Blockchain Meets Distributed File Systems* divide blockchain-based distributed file systems into 7 layers. The identity layer is responsible for identifying every single node. In IPFS an encrypted hash of the public key is used as the NodeId. The routing layer maintains a peer-connection topology between nodes, responds to queries from local and remote peers and communicates with distributed hash tables. The network layer defines the protocols used on the network layer. IPFS uses a solution named libP2P. On the data layer IPFS has four levels of a data model. A block, an arbitrary-sized piece of data. A list, a collection of blocks or other lists. A tree, a collection of blocks, lists or other trees. A Commit, a snapshot in the version history of a tree. As previously mentioned IPFS uses Filecoin on the incentive layer. On the storage market data is stored directly on the Filecoin blockchain. For the retrieval market data is recorded off-chain. The data-swap layer IPFS is realized through the BitSwap algorithm which is

based on BitTorrent. The last layer is the consensus layer. Filecoin employs three proofs. A transaction proof, a proof-of-replication and a proof-of-spacetime. The consensus is called expected consensus and is a round-based leader selection consensus protocol.