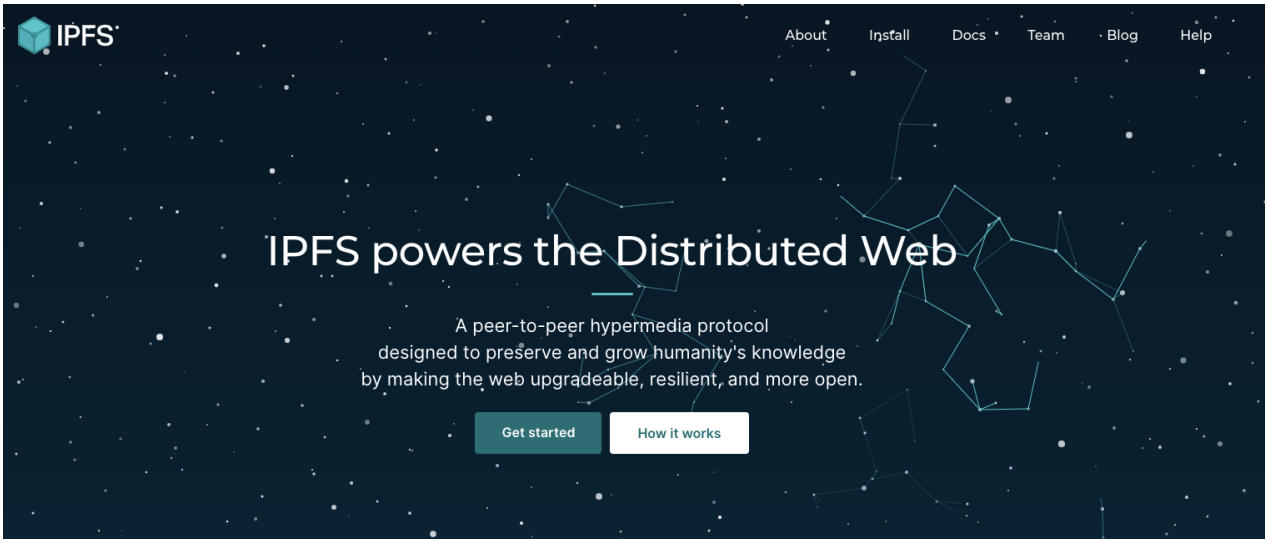
Lesson 14

Decentralised Storage

IPFS



- Distributed Hash Table, nodes can store & share data without central coordination
- IPNS allows exchanged data to be instantly pre-authenticated and verified using public key cryptography.
- The Merkle DAG enables uniquely identified, tamper-resistant and permanently stored data
- You can access past versions of edited data via the Version Control System

IPNS gives us

- Human Readable Links

E.g. /ipns/extropy instead of QmPrPmbbUKA3ZodhzPWZnpFgcPMFWF4QsxXbkWfEptTBKI

- A link to the latest version of content

IPFS based websites

- websites that are completely distributed
- websites that have no origin server
- websites that can run entirely on client side browsers

IPFS is built upon

- Distributed Hash Tables
- Block Exchange (BitSwap)
- Merkle DAGs (The central tenet of IPFS is modelling all data on a generalised merkle DAG)
- Version Control
- Self-certifying File System to give a naming system for files and nodes

Block Exchange - BitSwap Protocol
In IPFS, data distribution happens by exchanging blocks with peers using a BitTorrent inspired protocol: BitSwap.

Like BitTorrent, BitSwap peers are looking to acquire a set of blocks (want_list), and have another set of blocks to offer in exchange (have_list). Unlike BitTorrent, BitSwap is not limited to the blocks in one torrent.

BitSwap operates as a persistent marketplace where node can acquire the blocks they need, regardless of what files those blocks are part of. The blocks could come from completely unrelated files in the filesystem. Nodes come together to barter in the marketplace.

While the notion of a barter system implies a virtual currency could be created, this would require a global ledger to track ownership and transfer of the currency. This can be implemented as a BitSwap Strategy

Distributed Hash Tables

A distributed hash table (DHT) is a distributed system that provides a lookup service similar to a hash table: key-value pairs are stored in a DHT, and any participating node can efficiently retrieve the value associated with a given key. The main advantage of a DHT is that nodes can be added or removed with minimum work around re-distributing keys.

Properties

- Autonomy and decentralization: the nodes collectively form the system without any central coordination.
- Fault tolerance: the system should be reliable (in some sense) even with nodes continuously joining, leaving, and failing.
  Scalability: the system should function efficiently even with thousands or millions of nodes.**

The IPFS DHT in practice

The DHT is used in IPFS for routing, in other words:

1. to announce added data to the network
2. and help locate data that is requested by any node.

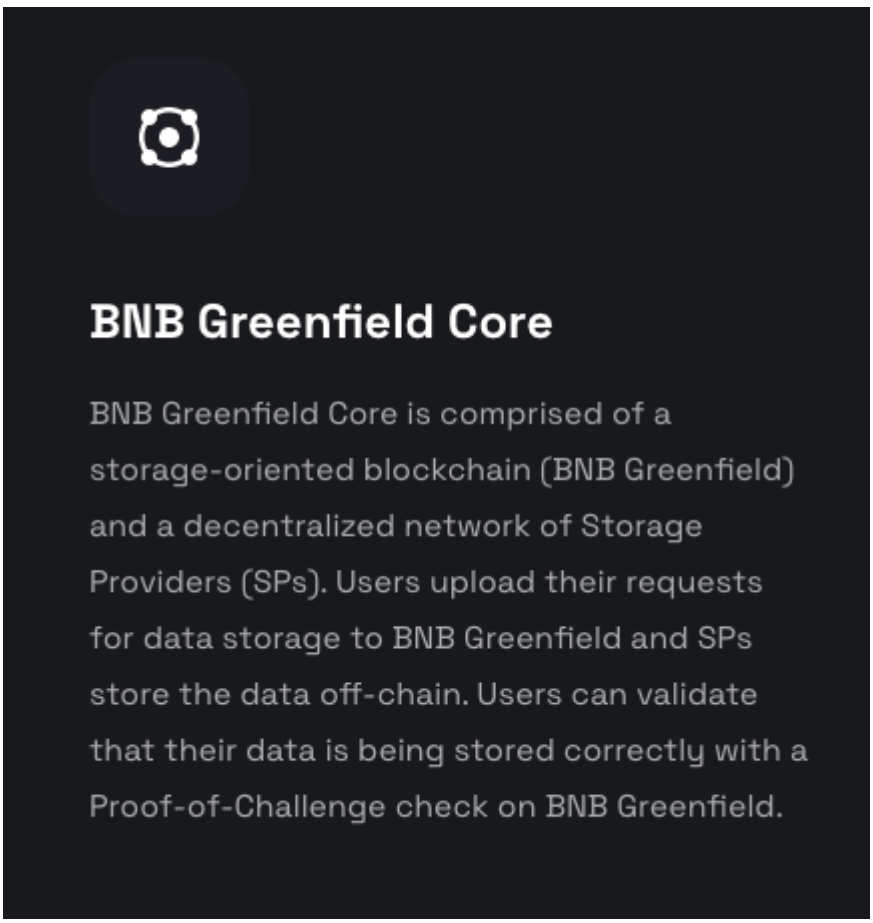
The white paper states:

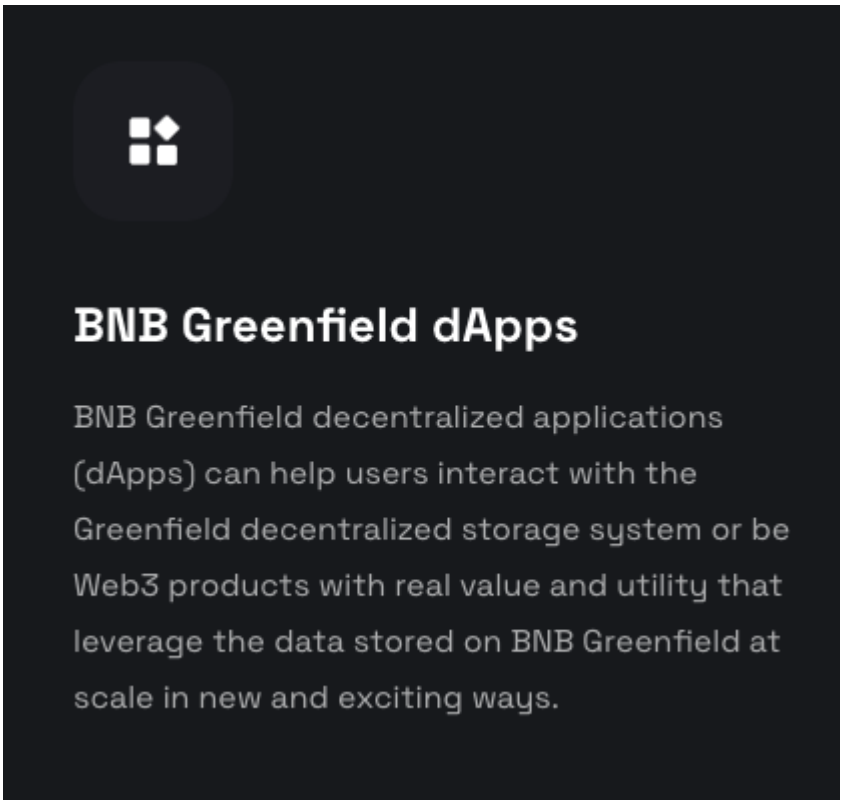
Small values (equal to or less than 1KB) are stored directly on the DHT. For values larger, the DHT stores references, which are the NodeIds of peers who can serve the block.

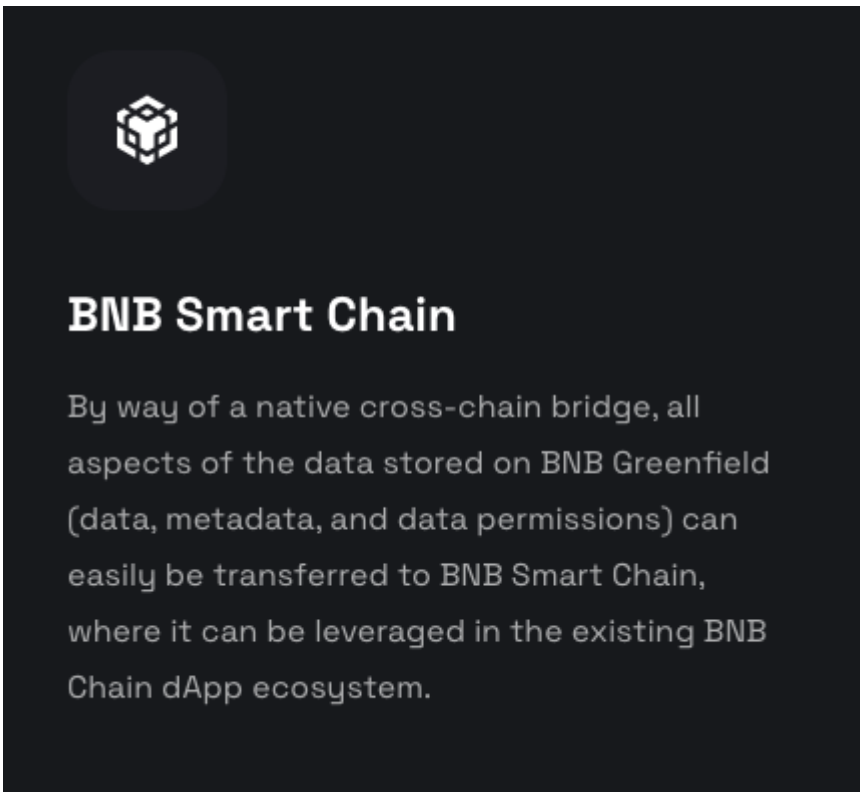
BNB Greenfield

White Paper

Components







Developer Docs

The aim is for users to be able to

1. "login" with anonymous cryptographic-based keys (IDs)
2. create, read, share, and even execute data with the user experience and cost close to the state-of-art cloud storage service today

3. fully own their data assets and control who can use them and how
4. easily put their data assets into a wide, smart-contract-based economic context to gain financial value with them.

Greenfield Core

BNB Greenfield Core consists of two components:

1. A blockchain focused on storage
2. A collection of "storage providers."

The BNB Greenfield blockchain preserves user and storage metadata records as shared blockchain state information. Third-party infrastructure services serve as Storage Providers (SPs).

These storage providers handle user requests for uploading and downloading data, acting as the custodian for user access and authentication.

In the beginning, several validators, managed by either the BNB community or SPs, participate in the genesis process to initiate BNB Greenfield. Simultaneously, some SPs will deploy the related storage infrastructure and register themselves on the Greenfield blockchain. These SPs establish a separate P2P network that offers a comprehensive suite of features to applications and users for creating, storing, accessing, and exchanging data, while utilising the Greenfield blockchain as the metadata and ledger layer.

User Identifiers

Every user possesses a unique address that serves as their account identifier. These addresses can generate objects for storage on Greenfield, manage permissions, and cover fees.

Greenfield's account structure is identical to BSC and Ethereum, utilizing the ECDSA secp256k1 curve for keys and adhering to EIP84 for complete BIP44 paths.

When displayed, a Greenfield address appears as a 42-character hex string, originating from the final 20 bytes of the public key of the managing account and prefixed with 0x.

Thanks to the compatible addressing system, users can repurpose existing accounts and infrastructure from BSC for Greenfield. For instance, they can employ TrustWallet and Metamask (or other

compatible wallets) to transfer their BNB from BSC to Greenfield and engage with dApps on Greenfield. Identifying the same owner is simplified by referencing identical addresses on both BSC and Greenfield.

Although the Greenfield blockchain is an application-specific chain without an EVM, its transaction data structure and API differ from BSC. Greenfield does not support all functions in existing wallets, such as Transfer and Send Transactions. However, it enables existing wallets to sign transactions using the EIP712 standard, which allows wallets to present data in signing prompts in an organised and legible manner.

Storage metadata model

The basic data models for Greenfield storage are:

- bucket
- object
- group
- permission

Bucket

Bucket is the unit to group storage "objects". BucketName has to be globally unique. Every user account can create a bucket. The account will become the "owner" of the bucket.
Each bucket should be associated with its own Primary SP, and the payment accounts for Read and Store. The owner's address will be the default payment account.

Object

Object is the basic unit to store data on Greenfield.

Object metadata is stored with the bucket name as the prefix of the key. It is possible to iterate through all objects under the same bucket, but it may be a heavy-lifting job for a large bucket with lots of objects.

Storage Process

1. Users contact the primary SP and ask for approval.
2. The SP signs and returns the message to indicate approval
3. Fees are locked for th user
4. Object metadata is checked to see if the object already exists
5. The user uploads the data
6. The primary SP synchronising with other SPs for redundancy
7. The seal transaction is created, the user can still cancel.
8. The object is sealed, this represents a contract between the user and the SP, the user agrees to the fees, and the SP promises data availability.

Use Cases

From the BNB blog

1. Web hosting
2. Recommendation Systems
3. Data Swapping / Data availability layer
4. Subscription-Based Decentralised Applications
5. Data Markets
6. Behavioural Analytics
7. Personal / Business Data Storage
8. Backup / Disaster Recovery / File sharing
9. Oracles

Status

The white paper has been produced but there is no hard ETA yet, but it **maybe** in the coming month.

Swarm

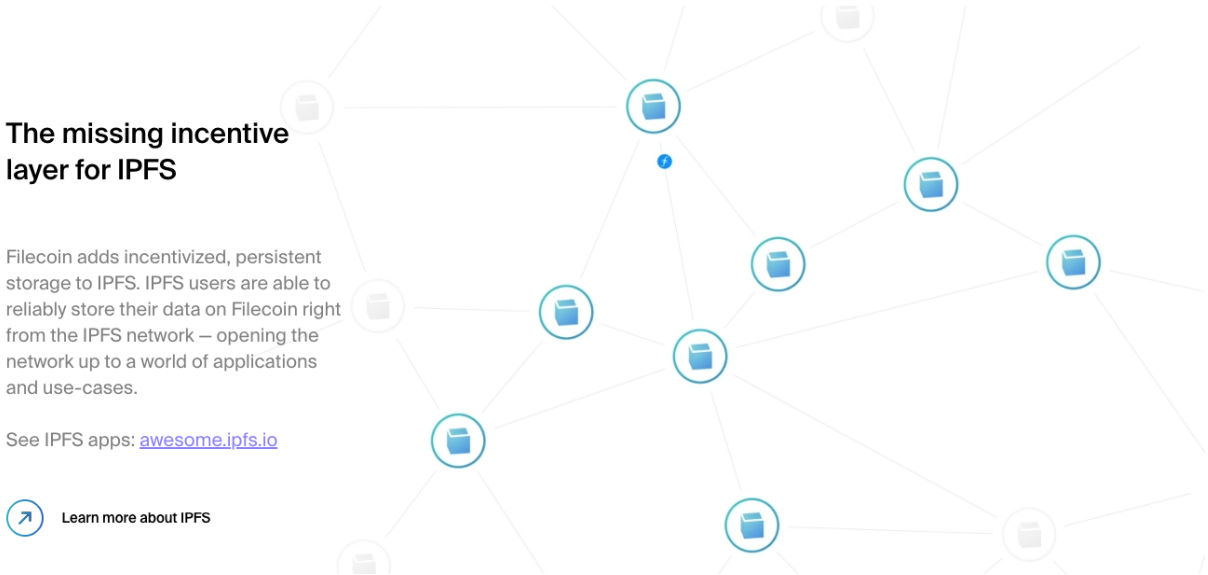
For Swarm to properly function as a decentralized p2p storage and communication infrastructure, on very basic terms there must be network participants who:

- contribute bandwidth for incoming and outgoing requests
- provide storage for users to upload and retrieve data
- forward incoming requests to peers who can fulfill them if they can not serve the request themselves

Swarm introduces its own incentives system for ensuring correct network behavior by rewarding nodes for serving these functions. The general swarm docs specify different functional layers of Swarm, which roughly correspond to its implementation roadmap:

- accounting system
- file insurance
- litigation

Filecoin



Filecoin is a digital storage and data retrieval method, made by Protocol Labs and builds on top of InterPlanetary File System, allowing users to rent unused hard drive space.

The project was launched in August 2017 and raised over $200 million within 30 minutes.

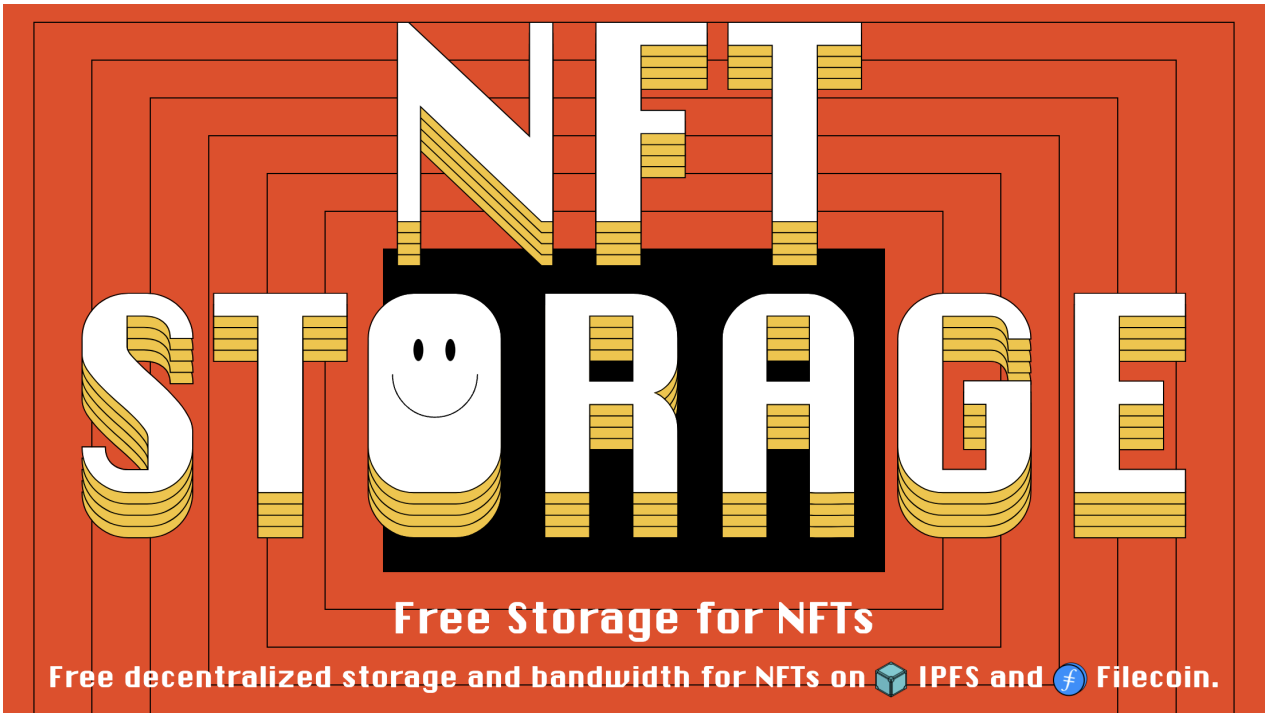
Proof of space

A proof-of-space is a piece of data that a prover sends to a verifier to prove that the prover has reserved a certain amount of space. For practicality, the verification process needs to be efficient, namely, consume a small amount of space and time. For soundness, it should be hard for the prover to pass the verification if it does not actually reserve the claimed amount of space. One way of implementing PoSpace is by using hard-to-pebble graphs.1(https://en.wikipedia.org/wiki/Proof_of_space#cite_note-DziembowskiFaust2015-1)[[3]] (https://en.wikipedia.org/wiki/Proof_of_space#cite_note-Ren2016-3) The verifier asks the prover to build a labelling of a hard-to-pebble graph. The prover commits to the labelling. The verifier then asks the prover to open several random locations in the commitment.

Proof of spacetime

Proof-of-spacetime differs from proof-of-capacity in that PoST allows network participants to prove that they have spent a "spacetime" resource, meaning that they have allocated storage capacity to the network over a period of time.

NFT Storage



See Documentation

See their quick start guide
The steps are

1. Create an account
2. Upload your assets
3. Get an API key
4. Create a client using the nft.storage package and the client library