IPFS Support in Brave

January 19, 2021 | Announcements

Brian Bondy, CTO and co-founder of Brave

Over the past several months, the Brave team has been working with Protocol Labs on adding InterPlanetary File System (IPFS) support in Brave. This is the first deep integration of its kind and we're very proud to outline how it works in this post.

IPFS is an exciting technology that can help content creators distribute content without high bandwidth costs, while taking advantage of data deduplication and data replication. There are performance advantages for loading content over IPFS by leveraging its geographically distributed swarm network. IPFS is important for blockchain and for self described data integrity. Previously viewed content can even be accessed offline with IPFS! The IPFS network gives access to content even if it has been censored by corporations and nation-states, such as for example, parts of Wikipedia.

IPFS support allows Brave desktop users to download content by using a content hash, known as the Content identifier (CID). Unlike HTTP(S), there is no specified location for the content.

Each node in the IPFS network is a potential host for the content being requested, and if a node doesn't have the content being requested, the node can retrieve the content from the swarm of peers. The retrieved content is verified locally, removing the need to trust a third party's integrity.

HTTP(S) uses Uniform Resource Locators (URLs) to specify the location of content. This system can be easily censored since the content is hosted in specific locations on behalf of a single entity and it is susceptible to Denial

of Service Attacks (DDoS). IPFS identifies its content by content paths and/or CIDs inside of **Uniform Resource Identifier (URIs)** but not URLs.

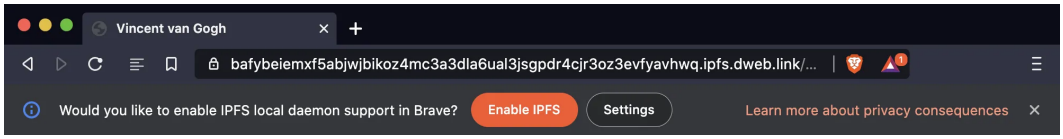
Overview of IPFS in Brave

Here's an example IPFS URI:

```
ipfs://bafybeiemxf5abjwjbikoz4mc3a3dla6ual3jsgpdr4cjr3oz3evfyavhwq/wiki/Vincent_van_Gogh.html
```

As of Brave 1.19.x, you can take this address, paste it in the Brave address bar, and load it.

By default, Brave will load the URI being requested via a public HTTP gateway; however, it will also show an infobar that asks you if you'd like to use a local node to resolve IPFS URIs. If you choose to use a local node, Brave will automatically download `go-ipfs` as a component and will route future traffic through this node. There is no need to manually manage an IPFS node or use an extension. A user can optionally install the IPFS Companion extension, and it will make a suggestion to use the Brave managed node.



Using a Gateway vs a Local Node

Brave allows you to configure what happens when IPFS URIs are encountered. You can select between:

- Using a local node for resolution
- Using a gateway of your choice for resolution
- Disabling IPFS resolution

When you have IPFS configured to use a local node, it will preserve the scheme (`ipfs:` or `ipns:`) in the address bar. You may want this option because you can always trust your local node to verify the content of CIDs

being accessed. You can also access previously viewed IPFS content offline when using a local node. A local node also contributes to the strength of the IPFS network.

If you're using a third party gateway, the address bar will show the redirected URL on the gateway server. You may want this option if you do not want to load an IPFS node on your local computer. You may have limited system resources and you just want access to IPFS content.

IPFS Privacy and Security Considerations

IPFS carries different privacy benefits and costs than sites loaded over traditional protocols such as HTTP(S). Some of these privacy considerations apply regardless of your IPFS configuration.

For example, typically browsers use the origin as a privacy and security boundary, called the same-origin policy (SOP). When loading sites over IPFS, Brave instead uses the CID as the origin boundary. Additionally, Brave only allows subresources to be loaded over IPFS when the main page is loaded from IPFS. A page can set a cookie for its own CID, but not that of another CID. An IPFS page can contain other IPFS images, stylesheets and iframes from any CID, and can fetch IPFS content within the same CID.

Other IPFS risks and benefits depend on how Brave is configured. If Brave is configured to use a local IPFS node, when accessing IPFS content, it also makes you a temporary host of that content. IPFS nodes use libp2p network-layer stack and have a PeerID which can be looked up in a distributed hash table (DHT), and that DHT can be observed by others. Both requests you make and content you serve are observable by network peers.

On the other hand, if Brave is configured to use a public IPFS gateway, the privacy risks are different. For example, that gateway can see the content you're asking it to load through IPFS requests. The gateway could potentially also lie about the content it is serving you. In the future, Brave plans to verify content retrieved through gateways by using its CID.

We do not allow IPFS URIs to be resolved in private windows and Tor windows. The go-ipfs component manages just one store of data and we configure it to run garbage collection every hour if the cache is at 90% of the storage max (1GB). When a user clears their browser data for cached images and files, we also trigger garbage collection on the stored IPFS content which go-ipfs manages. This deletes all IPFS content except for pinned content. We may decide to allow IPFS in private windows in the future by keeping a separate configuration and cache that gets automatically cleared when the session ends.

You can read more about privacy considerations here: https://support.brave.com/hc/en-us/articles/360051406452-How-does-IPFS-Impact-my-Privacy-

IPFS in Relation to Blockchains

IPFS and other decentralized protocols are important for blockchains and smart contracts that run in global decentralized networks, such as Ethereum.

What happens if you want to have a smart contract that addresses content? You wouldn't want to use a URL because the location of the content could change or become unavailable, and the smart contract is immutable. You could use a Uniform Resource Identifier (URI) though such as ones used to identify content in IPFS.

Storing content on Ethereum would be expensive, but storing a hash (CID) is feasible. The content of your data goes on the IPFS network, and the hash is stored in the smart contract. That hash can be used to access the content from any IPFS node.

Dapps can be served from IPFS as well and they can use web3 to talk to Ethereum. This enables users to have a fully decentralized browsing experience.

Implementation Details

System administrators can disable the functionality completely using an admin policy. There's also a global setting that can be used in `brave://flags`.

The *go-ipfs* binary is downloaded and kept up to date by Brave if a local node is enabled. It uses the same update mechanism as extensions.

Brave's IPFS support has been designed to minimize its effect on system resources. If Brave has been configured to use a local IPFS node, that node will be lazily loaded only when the first IPFS URI is accessed. If Brave has been configured to use the public gateway, the IPFS node is never loaded.

Users can troubleshoot IPFS by navigating to an internal page: `brave://ipfs`. This also shows users the gateway, API, and swarm ports. Users can also talk to the Brave managed instance using *go-ipfs* command line and by specifying an argument for the `--api=` parameter.

The configuration directory for the Brave managed go-ipfs node can be found in the browser's profile directory in a subfolder named `brave_ipfs`. You can find your profile directory listed here: `brave://version/`

Future Work

The work described above is just the start. We're actively working on more IPFS improvements, including:

- Support in our mobile browsers, Android is likely to come first.
- DNSLink which allows publishers to use DNS TXT records which point to an IPFS path. Brave specific implementation details
- Context menus for things like pinning content
- Website publishing
- More browser level UI
- Tor transport
- and much more…

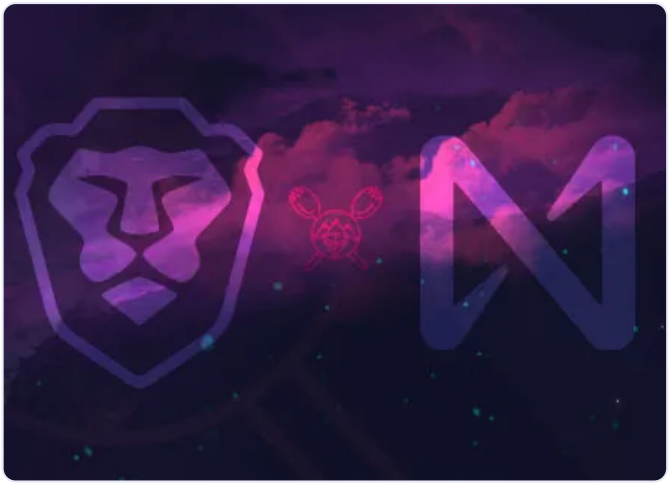
Reporting Issues and Learning More

You can read more about the implementation of IPFS in Brave from its spec here: github.com/brave/brave-browser/issues/10220

Future work is defined and managed here: github.com/brave/brave-browser/projects/32

We'd love to hear suggestions on how we can improve IPFS support in Brave. Issues can be posted here: github.com/brave/brave-browser/issues/new?labels=OS%2FDesktop&template=desktop.md

Related articles



NEAR, MetaBUILD 2, and Brave Head to ETHDenver

NEAR and Brave will be at ETHDenver 2022 for the MetaBUILD 2 Hackathon. The mission: Support the global community of developers in building a multi-bridge, multi-chain world!

Read this article →