Uniswap Interface + IPFS

June 30, 2020

TL;DR

In an effort to continue decentralizing, we have created a mechanism for the community to host the Uniswap Interface

The open-source Uniswap Interface built by our team and community is automatically deployed daily to IPFS

Community members can pin the IPFS hashes to ensure availability

We use IPNS + DNSLink to point `/ipns/app.uniswap.org` to the latest IPFS release

app.uniswap.org is now served exclusively from the latest IPFS release, however any IPFS gateway can be used directly

The URL uniswap.exchange now forwards to app.uniswap.org

The ENS contenthash for `uniswap.eth` now points to the latest IPFS release allowing the URL uniswap.eth.link to be used

Interfaces and decentralization

The Uniswap **protocol** is trustless and decentralized because it lives entirely on-chain. Anyone running an Ethereum node can interact with the contracts directly which will perform as programmed for as long as Ethereum exists. However, not everyone wants to run a node. Instead, many users choose to interact with Uniswap through web interfaces, trade aggregators, wallets, or other DAPPs that have integrated Uniswap natively in their smart contracts.

When using an interface, users should verify the transactions they sign match the transaction presented by the interface. However, it is not always easy to do this. This is why it is important to use reputable interfaces. We're thrilled to see a growing ecosystem of high-quality interfaces for Uniswap, including Argent, 1inch, Rainbow, Paraswap, Zerion, Instadapp, and many more.

Open source interfaces (including many of the above) allow users to verify the code they are interacting with does what it claims. If a user runs the code locally, they can make transactions with confidence. However, as soon as the code is hosted, **it is difficult for users to verify the website they are interacting has not been modified**.

This is one of the problem that IPFS **aims to solve**.

What we did

Our team has always cared about decentralization, security, and accessibility. This is why we built an **open-source interface** for Uniswap that the community can directly run, verify and build upon. We've just taken another step forward by decentralizing the hosting of the Uniswap Interface using IPFS.

Using **GitHub Actions**, the **Uniswap Interface** is now deployed at least once per day to IPFS. Each release is automatically **pinned** using **pinata.cloud**, a free IPFS pinning service. The IPFS releases can be found **on GitHub**.

This means the Uniswap Interface can now be accessed via IPFS directly, through a gateway such as **cloudflare-ipfs.com**, or by an *alias* to the Cloudflare gateway at **app.uniswap.org**.

The domain uniswap.exchange is now redirected to app.uniswap.org, which is an alias to the Cloudflare IPFS gateway that serves the Uniswap Interface from IPFS.

How we did it

The `app.uniswap.org` subdomain is given a CNAME record pointing at `cloudflare-ipfs.com`.

When a user visits the domain `app.uniswap.org`, the browser first looks up the DNS record and finds a CNAME to `cloudflare-ipfs.com`. The server at `cloudflare-ipfs.com`, i.e. Cloudflare's IPFS gateway, looks up the DNSLink record for the subdomain. That TXT record contains the IPFS hash of the latest release.

Cloudflare's IPFS gateway then fetches the content using the IPFS protocol and serves the interface to your browser via HTTPS.

Some changes

Because IPFS gateways will not default to serving `/index.html` as is expected by many single page applications, the Uniswap Interface uses a "hash" based router.

This means that links that contain paths, such as app.uniswap.org/swap will no longer work, but app.uniswap.org/#/pool will work.

Security Unicorn

Some settings on the Uniswap Interface use localstorage, which is shared across users in some IPFS gateways.

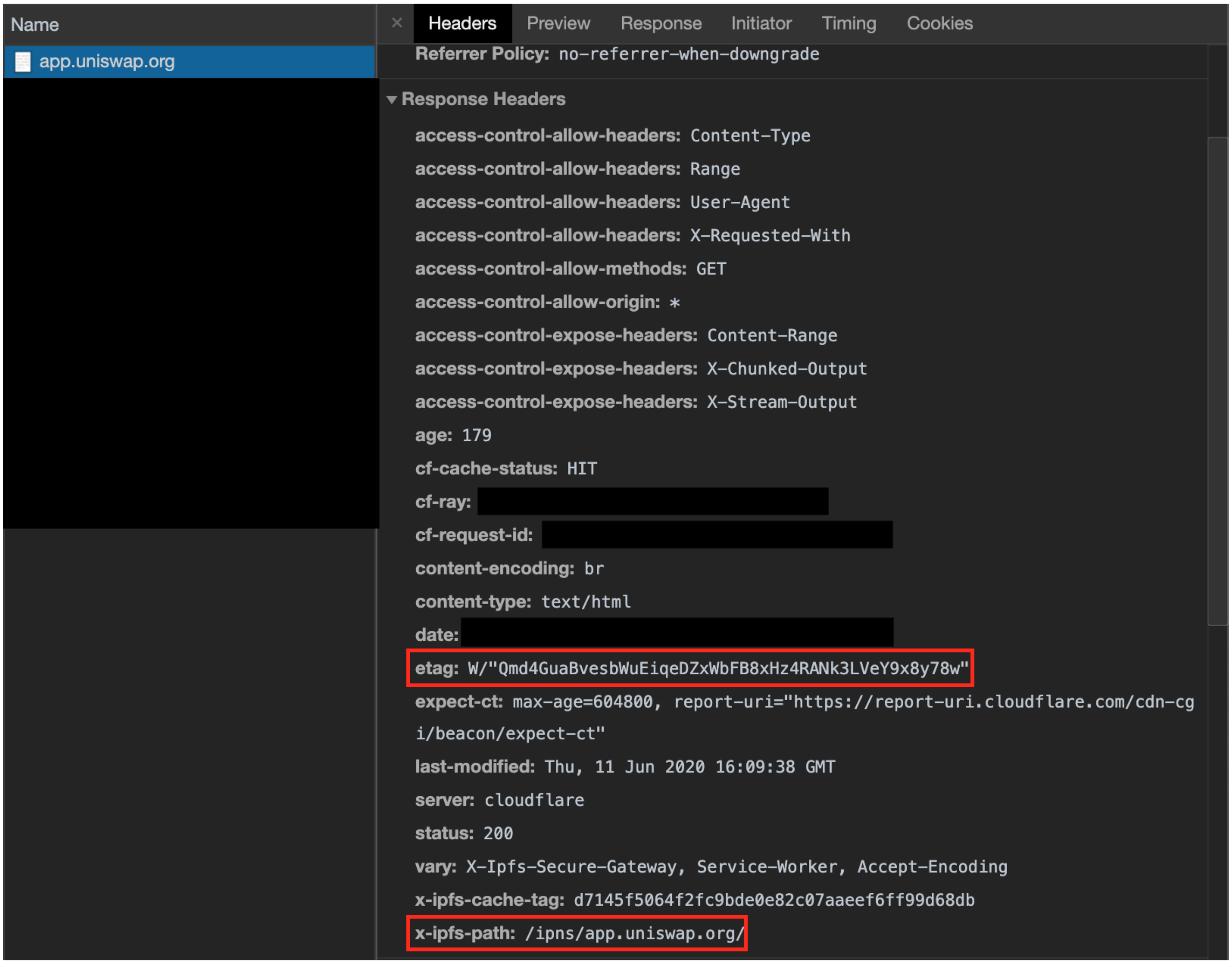
When using an IPFS gateway and referencing an IPFS hash or IPNS name by the *path* (e.g. cloudflare-ipfs.com/ipfs/QmdJApBwfsGua9v.../) rather than the *subdomain* (e.g. bafybeig6hsm....cf-ipfs.com), other sites accessed from the same IPFS gateway can view and change your settings in the Uniswap Interface.

To avoid this possibility, you can use the subdomain format of IPFS gateway URLs, which are contained in every release along with the path format.

Verifying a build

You can check what build you are being served from Cloudflare's IPFS gateway by looking at your browser's network console for the response headers sent directly from Cloudflare's IPFS gateway.

Cloudflare's gateway uses the IPFS hash of the deployment in the `etag` header, and reports the resolved IPNS path in the `x-ipfs-path` header.



[redacted]

```
Referrer Policy: no-referrer-when-downgrade
▾ Response Headers
access-control-allow-headers: Content-Type
access-control-allow-headers: Range
access-control-allow-headers: User-Agent
access-control-allow-headers: X-Requested-With
access-control-allow-methods: GET
access-control-allow-origin: *
access-control-expose-headers: Content-Range
access-control-expose-headers: X-Chunked-Output
access-control-expose-headers: X-Stream-Output
age: 179
cf-cache-status: HIT
cf-ray: [redacted]
cf-request-id: [redacted]
content-encoding: br
content-type: text/html
date: [redacted]
etag: W/"Qmd4GuaBvesbWuEiqeDZxWbFB8xHz4RANk3LVeY9x8y78w"
expect-ct: max-age=604800, report-uri="https://report-uri.cloudflare.com/cdn-cg
i/beacon/expect-ct"
last-modified: Thu, 11 Jun 2020 16:09:38 GMT
server: cloudflare
status: 200
vary: X-Ipfs-Secure-Gateway, Service-Worker, Accept-Encoding
x-ipfs-cache-tag: d7145f5064f2fc9bde0e82c07aaeef6ff99d68db
x-ipfs-path: /ipns/app.uniswap.org/
```

How you can help

To keep the Uniswap Interface available, you can pin the hash of the latest release.

If this sort of work sounds cool to you, we're hiring! Shoot us a message!