

Hi all,

Immutablelawyer here with yet another boring technical-legal post about progressively decentralising the dYdX Protocol. This post is meant to initiate a well-needed discussion with regard to decentralising the front-end layer of the protocol. This is a layer which is regularly disregarded when discussing the progressive decentralisation of a particular tech-stack yet, from a regulatory point of view, having a decentralised front-end could make or break a protocol. Firstly, let's get into the different layers that actually make up a decentralised system (or that would make up a potentially decentralised one).

[

image

1024x513 101 KB

](https://europe1.discourse-cdn.com/standard21/uploads/dydx/original/1X/3a5600f6c97e716a371222da992ee84d8471a9f4.png)

The image posted above is a clear example of what currently makes up the majority of tech-stack architecture within our industry (specifically with respect to on-chain finance i.e. DeFi, but also applicable to other sub-sets such as storage protocols, oracle protocols etc.). As you can see there are different elements present here with these different elements contributing (in a joint manner) to the overall decentralisation metric of a particular protocol. What is quite interesting about V4 is that the second layer within the tech-stack (the composable smart contract protocol), will be eliminated and thus, the dYdX V4 (high-level), tech-stack architecture will look more like this:

[

image

910x489 19.4 KB

](https://europe1.discourse-cdn.com/standard21/uploads/dydx/original/1X/508345a410a8129c934029e8dc43559fa964ebd1.png)

For the purposes of this discussion, the main focus shall be on the Client (i.e. Frontend). As previously stated, to reach a level of full decentralisation the different components here have to be decentralised (naturally eliminating users – however, with regard to the Token and User Data, decentralisation here takes the form of ensuring economic decentralisation re. the token and ensuring technical and legal decentralisation re. collection, storage and safekeeping of user data – I potentially plan to post on all these different facets, but let's get back to the topic at hand – the Frontend).

Why decentralise the Frontend?

The first question one might ask is why, after all, do we want a decentralised frontend layer within the dYdX Ecosystem? The two main points (among numerous others), to be addressed here are censorship and the concept of a 'Single-Point-of-Failure'.

Censorship and Single-Points-of-Failure

Censorship is an important phenomenon to tackle when wanting to move towards full or what I more adequately call, optimal level, of decentralisation. To explain this I shall provide a very recent example. Uniswap is one of the largest DEXs within our industry. The structure is made up of the Uniswap Foundation, Uniswap Labs and the Protocol (the tech-stack). Now, Uniswap Labs depicts itself as a mere frontend provider of the Protocol (we know that they are also the main active software developers of the Protocol – but that's a separate argument altogether) however, they developed both the Frontend and the Protocol itself (they have since decentralised many functions).

Uniswap had 3 frontend related events in 2022. The first related to the Privacy Policy (and this relates to the User Data component of the above diagram). The updated Privacy Policy stated that the DEX collects certain on-chain and even off-chain data connected to users' wallets. As pointed out by the block ([The Block: Uniswap's new privacy policy says it collects data tied to user wallets](#)) :

"It (Uniswap) clarified that publicly-available on-chain data is analyzed to help make informed decisions. As far as off-chain data goes, Uniswap claimed it does not gather sensitive personal data like names, emails or IP addresses. However, other off-chain web identifiers related to users' activity on front end website are still scraped, the exchange noted."

Our main concern here relates to the off-chain data i.e. data collected from the frontend. The centralisation of the Uniswap frontend (hosted by Uniswap Labs), led to a high degree of centralisation. Due to the amassed simultaneous on-chain and off-chain data collected by Uniswap Labs, it could theoretically link and IP Address to a real address to the on-chain data being a wallet and the transactions of that wallet. Now, some might say, "But IL, their frontend code is open-source, they're doing all they can!"-> I'll get to this in a bit.

With regard to on-chain data, we cannot make any arguments, on-chain data is transparent and immutable in nature – this is the beauty of the blockchain. What we can do, is providing for and catering for mechanisms which make sure that off-chain data (User Data), is not merely funnelled through one front-end which, if compromised, could lead to a data-leak like no other – leading to IP Addresses being linked to large wallet and god knows what follows.

The second frontend related matter Uniswap experienced regarded the main frontend (which funnels almost all of the Protocol's volume) going down in November of 2022. Again, here we have a clear example of a Single-Point-of-Failure. The front-end went down, the token crashed until it was back on, volume went down, a lot of people publicly stated that they'd move to use other DEXs. This is the result of having a Single-Point-of-Failure. Were Uniswap to promote and incentivise multiple front-ends, the aforementioned negative results would not have been an issue or would have been greatly diminished.

The last, and most important, Frontend incident dates back to August of 2022 when Uniswap Labs blocked 253 addresses from accessing its Frontend. The 253 blocked crypto addresses could continue to use Uniswap's smart yet, they were blocked from accessing the popular Uniswap website, which is a frontend managed and maintained by Uniswap Labs, a New York-based company.

These address blocks were the result of US Sanctions (kudos to them as these were some bad people for the most part – sanctions relating to heinous crimes were linked to most wallets [uniswap-trm.csv - GitHub](#)). Here's the issue: as pointed out by @banteg

([https://twitter.com/banteg/status/1560711571952865285?ref\\_src=twsrc%5furl%5fembed%5ftweet%5f1560711585190100992%5ftwgr%5fd45096ff27b874eb31e10afc4ab32d48426266b1%5ftwcon%5fs2\\_&ref\\_url=https%3A%2F%2Ffinance.yahoo.com%2Fnews%2Fpoq-uniswap-frontend-blocks-over-122401481.html](https://twitter.com/banteg/status/1560711571952865285?ref_src=twsrc%5furl%5fembed%5ftweet%5f1560711585190100992%5ftwgr%5fd45096ff27b874eb31e10afc4ab32d48426266b1%5ftwcon%5fs2_&ref_url=https%3A%2F%2Ffinance.yahoo.com%2Fnews%2Fpoq-uniswap-frontend-blocks-over-122401481.html)), circa 30 out of the 253 blocked addressed (12%), merely had associated ENS names to the address holders on the US Sanctions List and were thus, collateral damage. As pointed out by ZachXBT – some of these addresses were NomadWhiteHats (<https://twitter.com/zachxbt/status/1560713889851052033>).

What does the above mean?

The fact that Uniswap Labs has the capacity to censor sanctioned users also means that Uniswap Labs also has the capacity to censor (remove access to the frontend), to any user based on an arbitrary choice (case in point re. those caught in the crossfire above). Hence, where a frontend is centralised, we have a situation where one may abuse of this centralisation or rather, misuse it. Furthermore, this also gives rise to a situation where Uniswap Labs might be ordered to remove frontend access to Person X for an act performed in the United States – but maybe that act is not a crime within the European Union – thus, that person would be able to use an EU-Hosted frontend (this is just a high-level example – other examples could relate to the US party-in-power wanting to infringe on an opposition party's financial liberties and ordering Uniswap Labs to comply with blocking such a person from accessing the Frontend).

Potential Solutions

Dappnet

(P.S. I used Uniswap as an example due to the substantial similarity of its corporate structure to that of dYdX – specifically in relation to having the private company behind the Protocol also running the main frontend in the case of dYdX).

Following that example-giving section of the post we'll now move on to a potential solution that could be used to avoid any of the three scenarios experience by Uniswap – I am a big believer that those who do not learn from history are doomed to repeat it – so let's not repeat it!

I will now enlist a cool concept that could potentially be developed on Cosmos, and also a potential solution that could be leveraged to progressively decentralise the frontend layer of the dYdX Ecosystem. The first is not currently live and not available on Cosmos due to its reliance on ENS Domains - Dappnet.

Dappnet is a permissionless application network built on IPFS and ENSDomains. Dappnet addresses the issue of the current predominant access to dAPPs via DNS and servers (which are centralised in nature), and leverages ENS and IPFS hosting to create a 'BitTorrent-like P2P Network' (in fact Liam is currently discussing moving from IPFS to BitTorrent). More details on Dappnet can be found in Liam's tweet here ([https://twitter.com/liamzebbedee/status/1578127982173908992?s=20&t=E-U\\_vw4ibONdXpjkdUZocA](https://twitter.com/liamzebbedee/status/1578127982173908992?s=20&t=E-U_vw4ibONdXpjkdUZocA)), and in the GitHub Repo here ([issues - gliss-co/dappnet-features - GitHub](#)) – this is currently a work in progress. I've been looking forward to this for a while. However, unfortunately, this does not solve the issue with dYdX as it functions through ENS Domains. So let's move on to our next solution.

The Liquity-Model

Previously, I mentioned the following: "Now, some might say, "But IL, their frontend code is open-source, they're doing all they can!"-> I'll get to this in a bit." – let's get into this. Open-sourcing the frontend code is not enough to decentralise your front. Merely open-sourcing the codebase of the frontend is the equivalent of stating that one needs to use electric cars without adequately incentivising electric car usage (in my home country this is done through subsidies, government grants etc. – an interesting parallel to draw).

Liquity did just this. Liquity itself does not run its own frontend at all – it actually instructs users intending to use Liquity's features to choose from a list of Frontend Operators ([Liquity | Frontends List - Use Liquity](#)). Hence, to open loans, make deposits etc., users thus have to use one of the frontends provided by third parties (an explanation via video material can be found here: [Liquity Runs on Decentralized Frontends - YouTube](#)).

Frontend Operators provide a web interface to the end-user enabling them to interact with the Liquity protocol. For that service, they will be rewarded with a share of the LQTY tokens their users generate.

LQTY rewards are awarded to Stability Pool depositors and then proportionally shared between the users themselves and the Frontend Operator. How much each party gets is determined by the

Kickback Rate which is set by the Frontend Operator and can range between 0% and 100%. Each Stability Pool deposit is tagged by the Ethereum address of the front end through which the deposit was made. This address is where the frontend's LQTY rewards accrue.

Setting a high Kickback Rate will make the Frontend Operator attractive to users, but offering a nice interface and additional functionalities might allow for a lower kickback rate while still garnering user interest.

To incentivise this users to act as Frontend Operators, Liquity allows these Operators to set a rate between 0 and 100% that determines the fraction of LQTY that will be paid out as a kickback to the Stability Providers using the frontend. If a frontend set the kickback rate to 60%, their users would receive 60% of their earned rewards while the frontend receives the remaining 40%.

This is a simple, yet effective way of making sure that users are adequately incentivised to run frontends. In the case of Uniswap, the lack of incentivisation re. the operation of alternative frontends has naturally led to a stagnation in alt-frontends, and thus a centralisation and Single-Point-of-Failure in the Uniswap Labs operated frontend.

I firmly believe that this model should be put up as a Request For Proposal on the dYdX Grants Page so as to request research to be conducted on the possibility of adopting this model on Cosmos – or alternatively, set up a working group to tackle this optimisation. This, naturally, will lead to other issues that need to be taken into account – namely, ensuring that the economic incentives distributed to frontend operators are distributed in an equitable manner so as to not hinder the Chain's economic decentralisation. There are a myriad of ways to achieve this and one may not merely incentivise users to operate frontends via dYdX Tokens. I for one, would be rather in favour of incentivising Frontend Operators via USDC with the Frontend Operator getting a USDC kickback based on the volume traded by users through that particular frontend. This model would also bring in another set of Ecosystem Participants (additional to market makers and node operators), as users who cannot afford to meet the hardware requirements to set up a node due to high barrier to entry, would still be able to participate within the Ecosystem and contribute to its progressive decentralisation by operating a frontend (which naturally entails a substantially lower barrier to entry from a cost perspective).

Disclaimer: The operation of a Frontend could potentially trigger regulatory implications based on where the frontend is being operated from – the operation a frontend operator within the EU could be deemed to constitute the unregulated/unlicensed solicitation of users to use dYdX (which would breach current provisions in relation to financial services and crypto asset services). Hence, the research conducted should also take into account an assessment of the regulatory implications that could potentially be incurred by virtue of operating a frontend so as to make sure that frontend operators are aware of the risks and pitfalls associated therein.

Thankyou for taking the time to read through this post – I hope this serves as a catalyst to initiate discussions with regard to decentralising the frontend layer on dYdX v4.

Any feedback or discussion is always appreciated!