

Dock.io: The Middleground to Data Silo Liberation

Much has been made in the crypto world about replacing the data silos of the world, the big bad Facebooks and Linkedins that connected us and then got us [hooked](#) and now make \$\$\$ by [extracting from us](#). A handful of projects have launched what promise to be the decentralized versions of these platforms: earn.com (Linkedin), Afari (Twitter), Steemit (Reddit). There's one problem with the decentralized versions of these platforms: they're pretty empty. With minimal network size, they have minimal value (see [Metcalfe's Law](#)), and for the time being, those shy of the die hards will continue to spend their time where other people are. Every crypto investor and developer I know still gets their news from Twitter, a centralized data silo, and posts thoughts on Medium, another centralized data silo. These decentralized data sharing platforms, should they have any hope of dethroning the giants, will need a way to allow users to easily port data from the current data silos that be. Enter dock.io.

The Quick Pitch

Dock is a platform that allows people to own their data. Individual's data that has not yet been shared with applications is encrypted by default using the user's public key and stored in a decentralized file system (called [IPFS](#)). For an application to access a user's data, the user must give the application access by decrypting the data using his/her private key and then re-encrypting the data using that application's public key. Encrypting information with a public key allows only for the person that possesses the corresponding private key to decrypt the ciphertext. Thus, a person can now post to Facebook from dock.io and be in full control of what exactly he/she is giving Facebook. Additionally, that person could also post to LinkedIn the exact same content he/she is posting to Facebook by encoding the same data with LinkedIn's public key.

Even better, this process allows users to port their data into new applications easily. Right now, to port all of my data into a Facebook competitor, the competitor would need access an API from Facebook that revealed all of my data. There is no way that Facebook would ever give developers that level of access, as it would severely affect their business model that has given them a monopoly on social graphs and digital identity. It could also be illegal for Facebook according to some privacy laws. Using dock, I could port all of my connections on Facebook into Snapchat, LinkedIn, or any other data loving application. In an environment where network size is everything, this ability is massive. It empowers competition to data silos that has not been possible in the Web 2.0 thus far.

*Quick side note: Facebook isn't the most pressing example for dock's platform. It's just the easiest one for me to poke at because of how big they are. Dock is initially focusing on reputation-based sites that influence job searches, specifically within the gig economy. A better example would be the ability to port one's Uber driver reviews and Doordash delivery reviews onto LinkedIn, or to push all of their LinkedIn data onto another more general resume site that targets a different demographic of employers.

Dock also allows applications to push updates of a user's data to that user's profile. The user could then share this same data with another application, with the data verified by the original application. Some applications, such as a site connecting employers with employees, will want to only show verified data, so this is actually pretty useful.

Once an application has obtained a user's data, it is still impossible to prevent that application from doing whatever it wants with the data (i.e. selling it or using it to generate best-in-class advertisements). Some may view this negatively, but I think it actually helps dock because it works well with legacy applications' existing business models. It's already the case that when we post something to the internet, we should never expect to have control over it again. In the words of an anonymous Fortune 500 CEO: whenever you have a new design, it's always good to have some element that is familiar.

The Tech

Dock is a second layer protocol whose smart contracts run on Ethereum but leverage IPFS and [Chainpoint](#) to securely store data off chain. Ethereum allows people from all over the world to execute exchanges of value contingent on terms written in agreed upon code. Despite everything you've read recently, Ethereum is still the leader among smart contract platforms in just about every regard (core team talent, developer tools, etc.). IPFS allows users to store their data in a decentralized network and thus allows applications to obtain data even when the originator of the data is offline. IPFS achieves this through [content-addressed storage](#) as opposed to location-addressed storage--quick side note: [IPLD is awesome](#) and the reason IPFS is going to become the standard for decentralized file storing. When compared to Ethereum and IPFS, Chainpoint isn't that revolutionary but that's because it doesn't need to be complex to get its job done. Chainpoint leverages Merkle trees to allow users to show receipts of when data was last updated, thus giving applications some notion of the time when a user's data was created.

The dock Ethereum smart contracts allow for users and applications to exchange in a trustless manner. The user and application are the only ones with read-write capabilities on each smart contract they use, preventing any third party meddling. There are two scenarios, either the application is receiving data from the user, or the application is pushing data to the user's profile.

Scenario 1, An application receives data from a user:

The user sends the IPFS hash of the encrypted data to the user-application smart contract. The application, in order to access the hash, must send dock tokens to the smart contract (which then burns the tokens by sending them to a burn contract). It is unclear how dock prevents applications from obtaining IPFS hashes before they have paid. Wouldn't it be great for LinkedIn if LinkedIn could keep getting your data at 0 cost? Dock will probably use a [commit-reveal](#)

scheme, where the reveal is generated only after the tokens have been burned. Now that the application has your data, it can do whatever it wants with it, like publish it onto its feed or sell it. There are a few problems with this model, which are addressed in **Critiques**.

Scenario 2, An application pushes data to a user's account:

The application sends data about a user to a user-application smart contract. The user cannot reject data being sent by the application, although the application can only execute this behavior if the user has already opted into a smart contract with that application (which means the user-application pair has had some previous data exchange or have mutually agreed for this to be their first exchange). The data pushed contains the application's digital signature, which may be linked to the application's digital identity, thus allowing other applications who later will receive that same data to know that the data came from another application. As stated previously, this allows for verification of data by applications, which will be useful in applications that have strong data verification requirements. There are a few problems with this model too, which are also addressed in **Critiques**.

Critiques

Critiques in scenario 1 (when applications pay for user data):

There is no easy way for the application to know that they are receiving the most updated version of a user's data, and there is no way for them to know what exactly is in the data. Applications are burning dock tokens in return for data, and consequently need to know that they are not wasting money. This is a fairly gaping flaw.

Dock attempts to counter this problem slightly, as users receive no direct benefit from the token burn, as they are not being paid for the data. However, if that user is a dock token holder, they would benefit from the token burn due to the increase in price from a supply sink of a scarce commodity. The counter-argument to this point would be that a dock token holder doesn't want to adversely affect the network, and thus won't repeatedly send crappy data for the sake of having applications burn tokens, as that will scare the applications away and hurt the network. That argument is very weak though, as an attacker could send crappy data and immediately sell his/her dock tokens right after.

Another problem in scenario 1 is that dock requires that applications receive all updates to a data file in order to receive the next update. This is inefficient and also increases costs for businesses, which could deter businesses from using the platform. This is further compounded by the fact that, while applications could see all the updates before an update they are paying for (if dock links a user's Chainpoint receipts through hash pointers to previous Chainpoint receipts), applications still cannot see the updates that may be coming after. Users could hide future updates from applications, in order to exploit sunk cost fallacies to get applications to pay more.

To summarize, applications could end up burning a fair amount of dock for receiving crappy, worthless data.

Critique in scenario 2 (when applications get paid for sharing user data):

A lot of the problems that apply in scenario 1 also apply in scenario 2, but with a different attacker. In this scenario, the application is the malicious actor. The protocol does not specify the factors that determine how much applications get paid for pushing data to a user's account. What if the data being pushed already exists on IPFS in a slightly different format? What if the data is faulty or completely worthless?

To summarize, applications could end up being paid a fair amount of dock for producing crappy, worthless data. To clarify, the payment to applications comes from the same burn contract that is called when applications pay for data, not from dock users themselves. An additional problem with this model may be that it is seldom beneficial for a centralized data silo to share that data, unless the compensation exceeds the discounted net value of the data over some time period. Thus, dock would either need to pay applications well or rely on user demand that applications push data updates regardless of a loss of profit. Users may have that power, but who knows.

Other critiques:

IPFS doesn't allow users to delete data that has been stored. A counterargument to this point is that only select, approved parties will be able to decrypt that data anyway.

The whole platform would be extremely slow to run today (although [the dock CTO recently dropped a nice article explaining the scaling options they are exploring](#)).

The whole platform would be extremely expensive to run today between the gas required for smart contract execution and fees for Chainpoint anchoring, although a large part of the Ethereum scalability roadmap also leads to cost reduction.

At the end of the day, the project is not even a year old. None of these critiques have been proven to be unsolvable, and I'd bet they will be solved because human ingenuity is a pretty cool thing, especially when applied to malleable software. In the meantime, [dock.io](#) provides the best user experience (that I'm guessing is still pretty much entirely centralized) I've seen on any DApp.

Why The Token Model Works

Most DApps have tokens that act as payment and governance tokens. Dock isn't that different. You can already stake dock to vote on proposals for the network, such as whom to partner with next. However, dock's payment token model is unique because it requires no escrow within the

user-application smart contract and therefore businesses can use price oracles to purchase and burn (or acquire and sell) dock tokens immediately. Yes, this will create a high velocity of dock tokens, barring spectators, but it won't deter businesses from using the platform.

[High velocities can significantly lower the value of "money"](#), but I'm still bullish that the dock token will accrue value as the network grows for the following three reasons (in order):

- 1) The supply sink presented in Scenario One coupled with the supply inflation presented in Scenario Two creates a novel [burn-mint token model](#), where the inflation rate of the dock token will be determined by the proportion of the value of the data provided by users to the value of the data provided by applications. If I had to bet, I'd bet that the users provide significantly more valuable data than applications in the long term (maybe not in the short term, since Facebook could give dock basically everything they know about me, although I already got a fair amount of data off LinkedIn onto dock for free). Thus, the supply of dock tokens should decrease over time, despite its high velocity. Applications will still be able to monetize data, as said before. Now, they just have to pay a small price to the dock token holders to do so.
- 2) Dock [will be a platform worth governing](#) (especially given the compliance laws regarding data that are going to become increasingly more important and ubiquitous, such as the EU's GDPR). Strict data compliance laws will incentivize applications who are using the dock protocol to desire control over the dock protocol. The cost of changing protocols/forking is likely to be higher with a use case like dock's, given that it is (and will be increasingly) difficult to obtain users' trust over personal data and please regulators simultaneously.
- 3) The supply sink makes it likely that there will be more speculators than traditional payment token models, and more speculators lead to a lower token velocity. That said, it isn't too smart to make a speculative prediction based on other speculators' behavior.

Competition

There are two other platforms that may be considered to be competing directly with dock: [GXChain](#) and [Metadium](#). GXChain and Metadium have very similar technical and go-to-market approaches as each other, but both are very different from dock's. Both GXchain and Metadium are creating their own native blockchains and are geared toward Eastern markets, while dock is building on Ethereum and is geared toward Western markets.

GXChain

GXChain is creating its own smart contract platform. Their goal is to support DApps that seek to interact with user data, which would account for most types of DApps. The team has already launched a data exchange service that has accumulated over 7 mil USD in throughput and just

recently launched LendChain, which uses credit scores to connect borrowers and lender in a peer-to-peer marketplace on GXChain. The team is moving quickly and the project is promising. Their iOS application, Blockcity, reports 2.1 million registered users.

From a technical perspective, GXChain is building something much more complex than dock, trying to leverage user data to create a first layer protocol that attracts DApp developers. This aspect of the platform makes it a competitor moreso to Ethereum than to dock. For GXChain to not be a direct competitor to Ethereum, projects like [Polkadot](#) that promise to allow for cross-chain interoperability will need to launch successfully and in the near future. Polkadot isn't scheduled to launch until the middle of 2019, and there's no telling how scalable it will be at the time of launch. In the meantime, It is unlikely that GXChain can poach a meaningful amount of developers from existing smart contract platforms.

Their approach is strong to attract developers is though, reflected in their decision to create smart contracts compatible with WASM, which promises to be the standard virtual machine (in some form or another) across most smart contract platforms. They should also be able to hire plenty of mercenary developers due to low costs in the region and their well-sized war chest. If GXChain continues to attract users, and there is early signal that they can, the platform could be able to attract non-mercenary developers too.

In summary, building the full-stack is a much greater challenge than building a modular part of the stack, like what dock is doing, although the reward could be greater if [the fat protocol thesis](#) proves to be absolute.

Metadium

Metadium is also creating their own native chain, but they are significantly behind GXChain in their development. Additionally, the Metadium team has released very sparse information (besides a patent) outlying how developers would built smart contracts on their chain. Their consensus mechanism is Proof of Authority (PoA), which is extremely centralized and isn't much of an improvement over legacy systems in terms of decentralization. I don't think there are any blockchain developers itching to build on a PoA platform, let alone one that does not provide a developer tool kit. Despite being backed by well-known Eastern investors, these factors lead me to believe that Metadium is the least likely to win out of the three platforms. Because of their location/investors, Metadium will likely have to compete with GXChain directly in the East.

Dock versus GXChain

Because dock is more geared toward Westerners, it is unclear whether dock will ever have to compete with either of these two platforms. As mentioned above, GXChain is targeting the Asian market, with a heavy focus on China. Their application, Blockcity, is not navigatable for a non-Chinese speaking person. Additionally, GXChain is much more susceptible to an attack given its small market cap for a public chain.

Still, GXChain is about 10x the value of dock given today's circulating supply, and about 7x the value of dock adjusting for non-circulating supply. GXChain is significantly stronger than Metadium today, but is difficult to compare directly to dock because of how different both their go-to-market and technical approaches are. Dock's partnerships are all based in the US, and it does not seem that they have released any materials that would be helpful for Chinese users or developers. The road goes both ways.

Recap

Dock needs to need to make sure that applications feel comfortable paying (through burning) for data that they cannot see before they have paid. This is tricky. Dock also needs to make sure that applications are pushing data whose value is proportional to the dock tokens received for providing the data. This is also tricky. The scaling/cost/IPFS problems are being worked on by other teams that have the best minds in the space, and [the dock's team and expertise](#) puts them in a decent position to solve their application-specific problems. For example, their work using an in-house Plasma Cash solution allowed for [250k transactions per second](#) in a testing environment, which is pretty impressive.

Once all of these issues are sorted out, which may take some time, the protocol will be what internet users are increasingly calling for, and the application's front-end will have iterated from the current model to something strong enough to go viral. Because dock has gone to market so quickly, they have given themselves a long runway to iterate and acquire users. Their partnerships are strong because their executives are coming from inside the industry. There are a lot of promising blockchain projects that could achieve real world usage soon--dock could be one of them.

Dock's primary competition is strong but is targeting a different market with a different technical approach, leading me to believe that they don't have any competition other than what they will face from legacy data silos. However, legacy data silos will have problems competing with dock, as creating a similar platform would directly interfere with their current business model, killing profits in the short-term and pissing off a lot of shareholders.

Conclusion

Dock gives people an exit from the legacy data silos that will continue to extract from them until an exit is possible. Users are already showing an appetite for an exit, following the high profile attacks on Equifax, Facebook, Uber, etc. It is not currently possible to aggregate and control our digital identity, which is basically our data shared online. Our friends, shopping preferences, and google searches say more about us than our passports. Without a viable exit strategy (i.e. a way to aggregate and control this data), our digital identities are fragmented across profit-seeking, addiction-creating C-corps that don't care about much more than shareholder returns.

Other Notes About Other Parts of the Data Stack

There are some other cool parts of the data stack, such as data marketplaces geared for AI like [Ocean Protocol](#). Ocean's goal is to break up the oligarchies on AI that the biggest data silos are developing. Currently, data silos have no incentive to share their data; they are more incentivized to develop powerful AI using the data they accumulate, maximizing shareholder returns. Ocean's goal is to allow for more AI developers to enter the AI market by lowering the barrier to entry for them to acquire data necessary to train their algorithms. Other parts of this stack, like [Raven protocol](#), are creating decentralized services for modeling training, due to the rising cost of computational resources needed to run competitive AI modeling algorithms. Just how decentralized we can get remains to be seen, but the modularity with clear points of tangency between projects like Dock, Ocean, and Raven could accumulate into something very powerful, at least in my head.

There are also lots of [digital identity attestation/namespace solutions](#) being created, some of which have overlaps with dock, such as Uport's [3box architecture](#). I do not view projects like Uport as direct competitors today (due to their initial focus on identity attestation solutions), although should Uport's identity attestation solution become widespread, they could consolidate other parts of digital identity, such as personal data, down the road. How modular digital identity becomes remains to be seen.

Special thanks to Sandy Peng (Fission Capital), Dean Patrick (G2H2 Capital), March Zheng (Bizantine Capital), and Faris Nathoo for their input.

***This article is in no way investment advice from Andrew Bakst or Bizantine Capital, only the opinions of Andrew Bakst. Your investment decisions should be made by you, and you alone.**