Application 2023-04-168 Project Name: DatDot received

The following submission was recorded by NLnet. Thanks for your application, we look forward to learning more about your proposed project.

Contact

name ▶	Nina Breznik
phone ▶	+447898348386
email ▶	ninabreznik@gmail.com
organisation name ▶	Vison Baker Ltd. / playproject.io
country ▶	Slovenia/UK
consent ▶	Erase my details when no longer needed

Project

code ▶	2023-04-168
project name ▶	Project Name: DatDot
fund ▶	Entrust_Fund
requested amount ▶	€ 21.120
website ▶	• https://github.com/datdotorg

synopsis

```
DatDot enables peer-to-peer sharing of storage
space and data seeding to make data sovereignity
```

and portability more accessible and reliable for users.

P2P networks are more stable and available for popular files, but accessing less popular or rare files can be unreliable due to the need for at least one node to have the requested data and to be able to connect to the peer requesting the data. Meeting this requirement can sometimes be difficult as users may have limited access to the internet or turn off their computers.

DatDot project aims to create a system that enables peer-to-peer sharing of storage space and data seeding, eliminating the need for users to rely on renting servers for data hosting or accept the potential unreliability of P2P data sharing. To achieve this goal, our protocol is designed to automate the matchmaking process and conduct periodic checks to ensure reliable hosting and serving of data to readers.

DatDot consists of two main building blocks:
- a dat logic for managing storing to and retreiving data from the peers in the network (written in JS)
- a ledger logic for managing incentivized relationship between hosting requests and offers (written in Rust)

experience

My team and I have been actively involved in the p2p ecosystem for several years as contributors and consortium members in the Dat ecosystem (https://dat-ecosystem.org/). During this time, we have prototyped many p2p apps, but have found that poor availability has been a persistent issue.

Initially, we attempted to run our own server to serve data as the main peer, but quickly realized that it would be too costly. We then tried to use the Hashbase solution, developed by the founder of the Beaker browser, but this service was centralized and was later discontinued by its founder.

Our project also includes a distributed ledger component. Rather than building on top of existing blockchain networks, we have created a custom chain specifically tailored to our needs. We are exploring ways to track the ratio between how much data a user offers to host versus how much data they ask others to host for them without using tokens.

Our team has gained valuable blockchain knowledge through our three-year contract work with the Ethereum Foundation and our three year-long work on this project so far, during which we participated in the Substrate builders program, a framework built with Rust, for building our ledger logic.

usage

The DatDot project has been in development for the past three years. During the first year, we received 30.000 EUR in support from the Web 3 Foundation to build the initial prototype. In the second year, we were awarded a 35.000 EUR grant from the Polkadot Treasury, which enabled us to focus on enhancing the matching, tracking, and checking logic for the ledger component.

Our next milestone is to update our Dat logic, which is based on three main components: the hypercore protocol, hyperswarm DHT-based peer discovery, and the protomux module for custom protocol extensions. These core components have undergone a complete rewrite, and the entire protocol introduced major breaking changes last year.

We paused our development of the Dat logic until all the components reached the beta stage, and now we are ready to update our improved logic to the current versions.

This task will require the full-time effort of two JavaScript engineers with extensive peer-to-peer knowledge for three months, or the part-time effort of two engineers for six months. Our calculation is based on an hourly rate of 44 EUR/h. Therefore, the cost for two full-time engineers for three months would be 16.800 EUR, calculated as follows: 44 EUR/h * 80h a month *

3 months * 2 engineers. We add to thes 25% for the administrative work, reporting, accounting, rent for office space, electricity, internet, amortized computing hardware costs, pay for statutory holidays etc.

comparison

1. Beaker Browser
There have been several attempts to address the availability issue in the Dat ecosystem, but none have utilized an independent network of peers. The Beaker browser, a p2p browser based the Electron framework, had a seeding service built into its system, but unfortunately, it was not widely used, in parts because it was a fully featured browser but compared to Firefox or Chrome had frequent crashes and drops, but also because users could not just run apps in a browser of their choice, but were instead required to use Beaker Browser, which was a significant adoption barrier. Our approach is to, on one hand add an incentivization layer and reward users for sharing disk space with the network, and on the other hand to work closely with the DataShell project, which will bring p2p to normal web browsers, lowering the adoption barrier.

2. Hashbase
Previously, there was a small hosting service, called Hashbase that provided a way for users to seed their data. Hashbase itself was open source

and anyone could host their own version, but in practice this did not happen as it also involved significant extra effort and skills. Our approach differs from Hashbase in that we enable peers to make multiple redundant copies, where many peers host only a few chunks of data. This creates a more censorship-resistant and reliable model for users. Moreover, users can also offer their storage to other peers, making hosting of their data free or more affordable than if they used a single centralized hosting.

3. Homebase
Next, there was the Homebase project, which enabled users to self host their data on a rented cloud server. The problems here was again the dependency on big tech (AWS etc.) and the fact that users had to be skilled enough to be able to manage their own hosting server.

challenges

1. Multiplexing the connections
Until now, our networking logic did not support multiplexing. As a result, we were unable to reuse existing connections between peers and had to open new ones for each task they performed.

2. Using latest Hypercore protocol version to create and check merklelized proofs
Next, after a peer has been selected to host the assigned chunks of data, our technology will perform randomized checks to ensure the data's

availability. This process will enable us to replace the hoster with a new peer when needed and this way maintain reliable access to the data.
To accomplish this, we will verify the merklelized proofs of hosting provided by the hoster. However, due to the changes in the hypercore protocol, the way proofs are created and checked has also changed. Thus, we will need to conduct research and rewrite the logic for this aspect as well.

3. Updating all connections to use hyperswarm
Furthermore, we have relied on a utility module named Hyperbeam to establish one-time connections using a customized DHT topic. However, this module has not been upgraded to the latest version of the Hypercore protocol, resulting in the need for a rewrite of this component as well.

To summarize, our plan is to develop a comprehensive module that will manage all peer-to-peer connections and avoid duplication by establishing multiple channels within each connection for specific tasks. In addition, we aim to enhance the implementation of the merkle proofs for each data chunk through regular data availability checks.

ecosystem

The DatDot team is collaborating closely with various Dat ecosystem projects, including

WizardAmigos, PicoStack, Sonar, Cabal, Sher, SSC and a few more. As there is currently no data availability service in the ecosystem, many of these projects are interested about our solution for this problem and are actively working to make their custom data types, built on top of dat, compatible with our hosting technology.

User-facing apps such as Cabal and Sher lack a robust backup and availability system for their users' data, primarily due to the absence of a standard for data vaults. To address this issue, we plan to collaborate closely with the DataShell project, which is being developed by a partner team. Our aim is to create a modular approach that includes a data vault and a data hosting service, providing a comprehensive solution to this problem. This will enable user-facing apps like Cabal and Sher to offer their users complete control over their data, including portability, backups, and availability.

One additional benefit of the datdot network is its public ledger, which allows for the creation of custom registries, search engines, and other data processing solutions. Because the data hosted on the datdot network is publicly recorded on the ledger, it is possible for developers and researchers (such as the Sonar Project) to create their own applications and tools that leverage this data. This can lead to new insights, discoveries, and innovations that would not have been possible otherwise. However, it's important to note that while public data on the datdot network is publicly available, users have the option to encrypt their private data. This means that only authorized parties will be able to access and view this data.

To promote responsible data storage and safeguard users' privacy and security, we plan to work with WizardAmigos, a global community focused on tech education, especially in the P2P technology space. Through this collaboration, we aim to raise awareness about DatDot and educate developers and end-users on best practices for utilizing the system to ensure data sovereignty, portability, and accessibility. Our ultimate goal is to enable individuals and organizations to have greater control over their data and enhance their digital privacy.

DatDot is an open-source project that supports the creation of custom data hosting networks within a single project or community, as well as one or many general networks. Projects around the Web3 foundation and the Polkadot community have expressed interest in our solution, which will give them more confidence in using dat and it's protocols to build fully-featured apps, beyond just decentralized finance solutions which don't involve a lot of data and can therefore be stored on blockchains. Being able to deal with massive amounts of data in a decentralized setting can for example empower the current movement withing the blockchain world, to create tooling for organisations, inspired by the traditional cooperatives. The decision making processes in these so called DAOs (decentralized autonomous organizations) can be supported by the blockchain (voting, proposing etc.), but because these processes are very data intense and include a lot of discussions, online meetings, documents and proposals, it becomes expensive to be stored on the chain, which is another reason for why the web3 ecosystem decided to support our work, namely in order to be able to start building on more reliable p2p infrastrucure for dealing with larger volumes of work-in-progress data.

pgp

attachments

Check

Please check that the above contact details are correct and that any attachments you have included have been uploaded. If you are in doubt, and near a deadline, don’t hesitate to resubmit - better safe than sorry. If you want to make changes to the proposal, do the same.

If you experience any technical problems, please contact the webmaster.

I checked the box but did not receive an email

Besides the obvious candidate for undelivered email (check your spam folder if you have it), some people run into their own outdated email configuration. Do you use a legacy forwarding mechanism for your mail, from me@example.com to theactualmailbox@another.example.org? In that case, the final mailserver may toss these out due the use of modern anti-spoofing techniques (notably DMARC, DKIM and SPF) at our side. Essentially, forwarding the original email as was done historically means that you can't satisfy the origin and integrity conditions - and thus our email to you will be discarded...

The structural solution is to do the forwarding with a mechanism like *Sender Rewriting Scheme*. Ask your service provider, or consult the documentation of your software how to do that.