ISSUE 01/SEPTEMBER 17

STORJ BYTES

The Quarterly Community Newsletter

INDEX

WELCOME to Storj Bytes, a quarterly newsletter from the Storj Labs team. Storj Bytes is your resource for the latest product and technology updates and a look ahead at our roadmap. It also details community feedback and how we're integrating it into our work.

Storj Labs is thriving with product and technology adoption because of you, the community. Every single community member is crucial and plays a role in advancing distributed storage in our industry and how Storj grows. Without our ecosystem of Farmers, Renters, Moderators, Contributors, and Developers, many of whom have been with us since the very beginning, Storj wouldn't be what it is today.

We're now, as a community, in a position to drive the future of blockchain-based storage for all. We've finalized our token sale and are poised to accelerate company growth and development of the platform. We've aggregated community feedback and are focusing in the near term on improved onboarding and consistent payment plans. Development and services are our priority in Q4, and we hope this newsletter addresses these topics and provides visibility into what we are currently building and what new capabilities are on the horizon.

TECHNOLOGY IMPROVEMENTS

Our C-library has increased the efficiency and reliability of file transfer and encryption

Community members have requested an easier way to integrate with our Node.js core. As a result, the engineering team has largely been focused on migrating the Node.js core and CLI to a C-library called libstorj, with Node bindings called node-libstorj. We wrote libstorj as a replacement for the older JavaScript library with the purpose of interacting with the Storj network. We chose to build the core in C because of its portability, performance and efficiency. The bindings provide a portable way for other languages to interact with the network. Prior to the bindings, developers had trouble implementing a library that can reliably download files. Since developers invested significant time in this implementation, it detracted from the overall main goal of what they were trying to achieve: incorporate the Storj network in their application. Furthermore, by migrating to C, and re-architecting the code base, overall performance has greatly increased. This has directly resulted in an increased efficiency of the network transfer connection initiation time, the general response time for commands, how files are encrypted, and how larger files are transferred.

Earlier this year, we improved the network reliability for file downloads, including shard mirroring and reconstructing a file from its parts

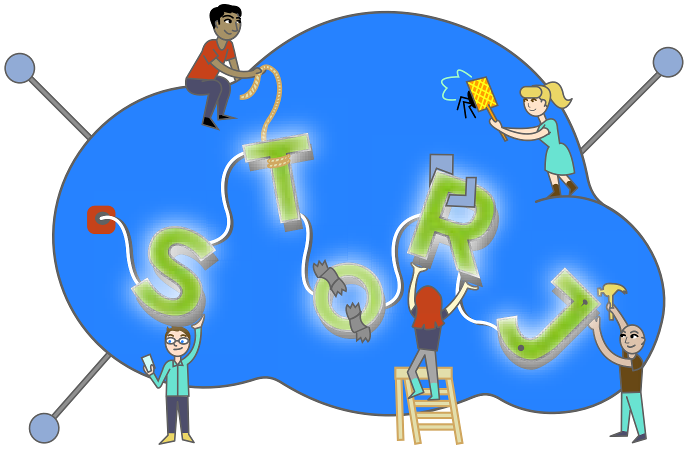
A distributed storage network utilizes its Farmer community and storage availability. When Farmers go offline, storage space decreases, resulting in an unacceptably low probability of file retrieval. We've made great strides to improve retrieval reliability by creating multiple copies of files and implementing an algorithm, "Reed-Solomon", that is able to reconstruct a file from just a fraction of its data shards.

Specifically, one of the issues with Clients receiving a file is that Farmers have had less than a 50 percent probability of being available to serve this data. The node´s (Farmer's) ability to retrieve data was dependent on the size of the file being received: a larger file size coincided with a decreased probability of data retrieval. Data retrieval issues are directly related to a node's performance and behavior. Nodes could go offline due to power failures, crashes, intermittent internet connections, and so on. The solution to these unfortunate circumstances was mirroring the node that's in a failed state, and adding erasure encoding. When a node is inactive, other nodes in its vicinity will voluntarily hold pieces of data that were originally stored on the node that failed. Erasure encoding comes into play in the event that too many nodes that hold data or parity shards have gone offline. The file's metadata is stored on the Bridge. The metadata contains the order and distribution of the shards stored at various node locations. To be able to reproduce the lost data, the metadata is requested from a Bridge. The missing shards can then be recreated from the remaining shards and parity shards. Up to two-thirds of the data can be lost and the file can still be reconstructed 100 percent. Parity shards are used to recalculate missing file shards and are stored on Farmer nodes along with file shards. This solution greatly increased the reliability of file downloads.

Bridges and Farmers will show an improvement in the scalability and reliability of file uploads

Like file downloads, file uploads rely on a pool of Farmers willing to accept a contract and hold the data for a specified time dictated by the contract. Originally, we implemented the Quasar/Kademlia structure to create a publish-subscribe model due to the promise of decentralized scalability. In this implementation, some Farmers are put at a disadvantage because other Farmers stay online but fail to propagate messages to their neighbors. These issues, along with timeouts, created an opportunity to mature from this model and create a replacement detailed in SIP6.

Quasar describes an event routing infrastructure for performing a publish-subscribe system on a structured peer-to-peer overlay network. However, the Quasar model is completely separate from the contract itself. Kademlia describes how a node is mapped in the network and where its neighbors are in relation to it. In this implementation, a Bridge advertises data that needs to be stored. Farmers (nodes) listen to the data and choose to subscribe to the advertisement. A Bridge sends a 'publish' message containing a contract to the network of subscribing nodes. The nodes review the contract and choose to accept or decline it. If a node accepts the contract, it sends an 'offer' message to the Bridge. If the Bridge accepts the 'offer' message, it sends a 'consignment' message to the node. The Bridge then receives a token from that node to upload data to it.



While this implementation works, with time, we found it to be unreliable. Community issues were also raised that confirmed the possibility of oversights in the design. In the described setup, a Bridge would wait for a certain amount of time to get a response from different nodes to see if any are interested in a contract. Nodes organized via the Kademlia setup relay this information to other nodes in their vicinity. However, not all nodes ideally performed this task. Timeouts increased due to some nodes having bad connections, refusing to relay the data to other nodes they are connected to, and so on. If a node is the last link in the sequence, it most likely will be delayed in sending an 'offer' message to the Bridge. This puts some nodes on the network at an unfair disadvantage and creates a setback for the network. We invested considerable engineering efforts to resolve this issue by creating SIP6.

The SIP6 implementation provides a different strategy of how a Bridge communicates to a Farmer. It facilitates direct communication between a Bridge and a Farmer where the Bridge will directly contact a Farmer based on qualitative data. Unlike the first solution described, Bridges are aware of Farmers in its network and their space availability information for holding data. When a Bridge asks a group of nodes to store data, a node has an option to accept or decline it. This communication is immediate and does not rely on relaying messages between nodes.

We are currently working on deploying these changes. For background information, please check out Problems with Quasar based publish-subscribe systems in peer-to-peer storage networks authored by one of our engineers.

A status page informs the community of any incidents

We feel it's crucial that the Storj community is aware of incidents when they happen. They should also know why they have happened, and how they were resolved. We've created a status page to enhance the community experience and provide updates around incidents that occur. The status page is our commitment to reinforce confidence in the product and provide an even greater level of transparency to the community.

STORJ NOW

The Bridge's GUI is now more informative

The Bridge's GUI is a key tool that enables users to simplify user account information on the Storj network by providing detailed information and a refined experience. The team is currently working on improving the experience and capabilities within the GUI, and updates will be rolled out in the coming months.

The updates will be centered on insights and payment. The improved GUI will add more useful features for Renters to work with their Storj account and files, like the ability to monitor data usage. There will now also be more insights into file storage usage statistics, which will replace the bucket tab on the current GUI.

The updated GUI will also deliver an improved experience around billing and payments with monthly invoices, clearer data about how your bill was calculated, and add more types of payment methods. Renters should see an improvement in how they can pay for storage on the network including STORJ tokens, Bitcoin, Ethereum and credit card payment options.

Front-end user experience will be enhanced

For the initial deployments around the Storj platform, we focused engineering efforts on what made the network better - the security, encryption, performance, and scalability. Largely, we've been concentrating on the backend. This leaves work to be done on how users and developers experience the overall product. We recognize that if Storj is going to continue to be successful, we need to have the best, most intuitive experience in the industry. We will be dedicating efforts to making that possible. Some initial changes to look forward to will be focused on how Farmers and developers use and interact with Storj.

STORJ FORWARD

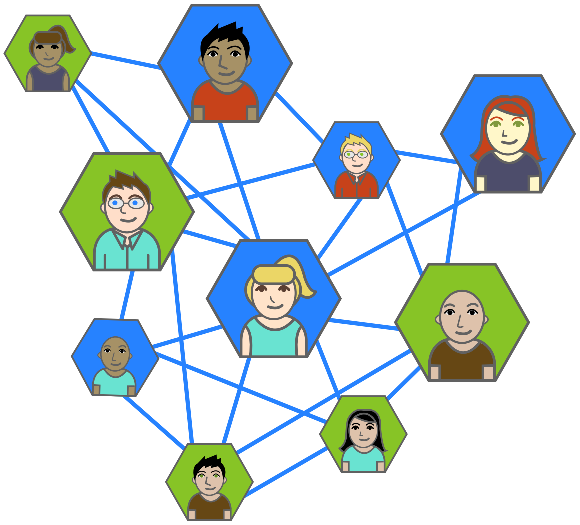
Documentation will be more substantial to help onboard users to the network

Product improvements you can expect to be implemented by December 2017 include crisper documentation. The goal of documentation is to have Farmers, developers, and everyone else onboard to the Storj network at a faster and easier pace. We will provide substantial resources such as detailed getting-started guides, screencasts, projects, and code snippets to enable the community to build and use the Storj platform. You should start to see improvements of how documentation is constructed and organized by the end of this year. Through 2018, you should see even bigger improvements. User-friendly software and documentation are one of our main priorities and will be an ongoing project late this year and into next year.

Part of our initiative with documentation is to also update the Storj GitHub repositories. We will be removing unnecessary files and repositories. Furthermore, we will adopt a consistent naming convention. These changes should help developers intuitively search through the Storj repos.

Ethereum integration

We will internally begin integrating with the Ethereum network for Farmer payments and other use cases. Some of the issues raised within the community, such as the monthly Farmer payouts, can be improved through this integration. This will enable Storj to potentially pay farmers in real time for the data contracts they are fulfilling. Ethereum smart contracts will also enable new Bridge operators to have a requirement to lock up STORJ tokens so they are there to be distributed to farmers at payout time.



OUR COMMUNITY REACHED OVER 10,000 MEMBERS

STORJ COMMUNITY

Community will have frequent company updates

Moving forward, watch for regular updates from the Storj Labs team. Storj Bytes will be a quarterly release that details updates on company direction and product improvements. Following the release of the newsletter, will be a townhall meeting with live Q&A sessions in an open forum.

You can also expect frequent new posts on our Storj blog. Here, we will continue to share detailed accounts about our engineering team developments, insights on the industry, and much more. Finally, we will continue to hold weekly developer meetings for discussion of ongoing projects. Developer meetings are held Wednesdays at 11am EDT in the #dev channel of our rocketchat. Our goal is to be as transparent as possible as we continue to work together to strengthen and drive network adoption.

Community members will have perks

Storj deeply values its community members. To show our appreciation to you, we will be developing a bounty program for those of you who are active and/or contribute to the community. You should see the bounty program roll out within the coming month. Also, we will be planning out contests, hackathons, and the likes where community members can participate and win prizes. Stay tuned- you should see announcements regarding this very soon.

Building the Storj community through partnerships

An important part of growing our community is leveraging partnerships to expand the Storj ecosystem. Through integrations with our partners, we can accelerate adoption of the platform. One of these partners is FileZilla. FileZilla is helping us by providing a GUI experience with data management and file transfer on the Storj network. Watch for further updates regarding this partner and future partnerships. Our partner ecosystem continues to grow stronger every day and as it does, it strengthens the Storj community and network. If you have questions about partnerships, please reach out to ***partnerships@storj.io*** .

The Storj team is growing

With our token sale completed, the Storj Labs team has the resources to accelerate product and technology development as we enter the next iteration of our product evolution. We are searching for smart individuals who are passionate about decentralization, blockchain and free and open-source software. Open positions include roles on our products, design, marketing and operations teams. To see the positions available, please visit the Storj Labs Careers page.

We've already hired several talented team members. James is our developer; Nadine is our developer evangelist; Garrett is our data analyst; Dan is our PR director; and John is our customer success director. These team members will support the development of the Storj platform as well as our community and partner ecosystems. Check out our team page to see the new members.

Lastly, we want to share that Tome Boshevski has resigned from the company. On behalf of the entire team at Storj Labs, we thank Tome for his contributions as Chief Design Officer. His state-of-the-art design has helped us stand out within the community and we wish him the best in his future endeavors.

Our team is excited for the opportunities on the horizon! We couldn't be more grateful to the community leaders and members who make up so much of what Storj Labs is. As we move forward, we want to assure you that we are prioritizing product features and improvements based on your feedback. As this community grows, we are excited to work together on our next phase of development of blockchain-based decentralized storage.