

EPISODE 1261

[INTRODUCTION]

[00:00:01] JM: The company Skynet Labs provides an open protocol for hosting data and web applications on the decentralized web. Skynet allows for decentralized censorship-resistant, highly redundant storage and applications that are available around the globe. Developers don't pay for their application storage. They can launch apps with access to a user's data right away, and they're free from corporations pulling access to the resources. They can maintain failover sites for when their primary site goes down. For users, you take your data with you without any corporate oversight, support developers and content creators by simply accessing their work, experience a web free of targeted ads, and never have to put your privacy or security at risk. Skynet is built on top of the Sia blockchain network, which is open source and guided by the Sia foundation. In this episode, we talk with Matthew Sevey, engineering director at Skynet Labs. Before joining Skynet, Matthew was a project manager at Procter and Gamble, and a web development fellow at Startup Institute before that. We discuss the decentralized internet, the Sia blockchain network and Skynet's mission to build a better and more equitable ecosystem.

A few announcements before we get started. One, if you like Clubhouse, subscribe to the Club for Software Daily on Clubhouse. It's just Software Daily. And we'll be doing some interesting Clubhouse sessions within the next few weeks. And two, if you're looking for a job, we are hiring a variety of roles. We're looking for a social media manager. We're looking for a graphic designer. And we're looking for writers. If you are interested in contributing content to Software Engineering Daily, or even if you're a podcaster, and you're curious about how to get involved, we are looking for people with interesting backgrounds who can contribute to Software Engineering Daily. Again, mostly we're looking for social media help and design help. But if you're a writer or a podcaster, we'd also love to hear from you. You can send me an email with your resume, jeff@softwareengineeringdaily.com. That's jeff@softwareengineeringdaily.com.

[INTERVIEW]

[00:02:09] JM: Matt, welcome to the show.

[00:02:10] MS: Thanks for having me.

[00:02:12] JM: You work on Skynet, and I want to get into what Skynet is. But first, I'd like to talk about the idea of a decentralized Internet. We've done several shows on this topic. We've done a bunch of Ethereum shows, did a show about DFINITY, plenty of IPFS coverage. So those are the infrastructure pieces I know of about decentralized Internet. Give me an overview for how the concept of the decentralized Internet has evolved over the last five years.

[00:02:43] MS: Yeah, so some of the projects that you talked about are – So if I take a step back, we as a company, first we're working on Sia, the decentralized data storage network. And that really is more of a comparison to some of the projects you've talked about in terms of IPFS and Filecoin and really offering decentralized data storage as a foundational piece of infrastructure for a decentralized Internet. About a year ago, we released a feature set on top of Sia, which is what we called Skynet. And that was the first time that we were able to really provide a new type of feature set to users and really create this whole new web ecosystem for applications and users. And so we kind of use Skynet pretty in that way. Or none of the projects that we know of in the space can offer what Skynet can offer right now. And something that I'm personally super excited about in terms of this evolution of the technology is that before, with Sia, and other decentralized data storage platforms, there is still a really big technical hurdle for users to kind of use the platform directly, or kind of get that benefit directly.

So like examples on Sia would be if you as a user wanted to store data on Sia, you still had to run your own Sia node, which requires a certain level of kind of technical understanding to know how to run it properly and maintain it. Or you would go through a third party company like Filebase, for example, is one of the projects that is built on top of Sia that offers nice kind of S3 compatible storage on the Sia network. And those are both great options. And what File offers is really great. But it's an abstraction that some of the benefits of Sia get kind of lost as soon as you have to rely on that third-party layer, because you're ultimately having to kind of trust your data to this platform.

So when we released the feature set that has become Skynet, it kind of took that technology to the next stage of providing a way for users to get the benefit of a decentralized data storage

network in the web context, and that users can truly own their data as they're surfing around the web. And at the same time, not have to worry about running their own node and kind of – So they get the benefit without the kind of technical hurdle of getting into things. And so the vast majority of users of Skynet never even have to worry about blockchain, or cryptocurrencies, or any of that kind of more technical or kind of new terminology that a lot of people don't quite understand or don't know how they feel about. So it allows anybody to really gain the benefits of decentralized Internet without having to jump through any technical hurdles or any long learning curves to kind of understand how to achieve that.

[00:06:14] JM: Can you give me a little bit more of a description for what Sia offered at launch?

[00:06:21] MS: Sure. So starting off, Sia was a decentralized data storage platform. So the idea being that users could run their own – So either a host could join the network, or an offer free disk capacity, or a renter could join the network and use that excess disk capacity. So kind of like Airbnb created an open marketplace for the exchange of physical space between renters that hosts, Sia created an open marketplace for renters and hosts to exchange digital space. And along with that, they'll basically giving users the scalability of the cloud while maintaining control of their data.

And so if a renter uploads their data to the network, it's encrypted client-side. So anything on the network is encrypted, and it's ratio-coated, and split across multiple hosts. So increasing the reliability and redundancy of their data, and giving that all to users directly. So they don't have to go through cloud providers. And additionally, it kind of gives that benefits to hosts and that, right now, if you're trying to get into the hosting space and trying to offer storage capacity to users, you have to provide enough storage for all of your users data. And with Sia, on the hosting side, it kind of brings that barrier down to allow any host of pretty much any size to start offering storage to the network.

A host could start with four terabytes of storage capacity. If you only had four terabytes of storage capacity, like you could not start your own like cloud providing service. That's not even close to the amount of data that you'd actually need. But on Sia, a little kind of one person shop can start with four terabytes of storage, start earning money off of those four terabytes and slowly start building out their capacity and adding it to the Sia network. And so really creating

that kind of open market for data storage and giving more power to the individual renter as well as the individual hosts and making it more kind of equitable ecosystem.

[00:08:39] JM: How did Sia differ from IPFS?

[00:08:44] MS: One of the big things that differentiated us was the incentives for the network. And so right now on IPFS, there're no incentives for nodes to pin content. It's kind of just a faith-based pinning, and they pin popular content because people are interested in it. But there's no guarantee that if I upload a video of my dog, that someone is going to care enough to keep pinning that on the IPFS network. Where on Sia, everything is economically incentivized and economically backed. So hosts are getting paid for every bite of storage that they have. So they're economically incentivized to keep storing that data for as long as I care to store it. And so it just makes it – I mean, additionally, Sia has from kind of day one, it was always a focus on being a live production network and kind of jumping past that stage of having kind of like a test kind of development network for kind of hackathons and projects. And so, the Sia network always prioritized being production-ready, production-first and economically incentivized to really support data being stored long-term and really making sure that if someone uploads a photo or a file today, if they keep paying for that storage, that storage is going to be there for as long as they need it.

[00:10:06] JM: So the obvious – Well, I guess the component that goes along with storage is you obviously need compute also. So storage and compute for a true fully functional application. So does the Sia network also handle the compute side of things? Or is it just the storage?

[00:10:27] MS: Sia is just the storage? And that's one of the powerful things about how we released Skynet, is that it changes the paradigm of how people think about cloud-based applications. And so as you said, when someone today thinks about building a cloud-based application on the web, they're going to think about, “Okay, where do I store my data? And how do I have servers to do all this compute to crunch all these numbers to like serve all these millions of users?”

With Skynet, it turns that paradigm on its head a little bit and that we only have to solve the problem of storage. And all the applications are completely client-side in the user's browser. And

so the compute is passed off to the client, which then solves a huge amount of infrastructure questions for developers when they're building applications. And so we've used Skynet, really, as a really powerful platform for developers for that reason. And that a developer today, if they wanted to build a web application, they got to figure out what they want to build and then they got to figure out where they're going to host it and pay for the infrastructure that's going to scale as their application scales with users.

With Skynet, a developer can build an application, deploy it on Skynet, and they're done. They don't have to worry about infrastructure ever again. And because the application is always loaded and handled on the client-side browser, if you have a Skynet application that serves 100 million users, that's just 100 million browsers that are open. And there's no compute load to the network. And it's just a storage, decentralized storage network at that point.

[00:12:15] JM: How does the performance of decentralized storage compare to typical centralized storage like S3?

[00:12:25] MS: Pretty good actually. So we are getting, or kind of building out our tooling around performance management and kind of getting more data on like how our network is providing. But in terms of time to first byte, for example, we can see sub-50 millisecond response times, kind of the P999 statistics is around 200 milliseconds. And so for a lot of the times, it's very comparable and, really, at that point, not a performance limitation in terms of choosing should I go with a centralized service versus Skynet?

[00:13:08] JM: Can you walk me through a read and or write to Skynet?

[00:13:14] MS: Sure. So there's a – We'll kind of step back a minute and kind of give some context. So when we talk about nodes on the network for Skynet called portals, it's kind of like that's how you access the Skynet network. And when a file is uploaded to Skynet, we have – We call it a Skyfile. And that file is immutable. And you get a Skylink back. And so that's one type of reading and writing to the network. But immutable data is great for some things, but also not great for other things. And so for a more mutable data, we use something that's called – We call the registry, which is a key value store on all the hosts on the network. And so I'll kind of walk through both of those examples.

And so currently, today, most people will access Skynet through our public web portal, which is siasky.net. Skynet is still very decentralized in the sense that the hardware requirements to run a portal are low enough that anyone could run a portal on their laptop. So if they're the type of user that is willing to kind of get into the weeds of like how to run a portal, they can still run the portal on their website and they don't have to rely on our public web portals to access Skynet. But we want to be able to offer that for the users that don't want to have to worry about dealing with portals.

So when a user uploads a file to siasky.net or does it through the Skynet application, that data will be uploaded by our portal to the Sia network. And so kind of the specifics around that as it's right now the files are broken into four megabyte sector. And so for a small file, so anything less than four megabytes, it's going to be kind of uploaded with 10X redundancy. So there's going to be 10 four megabyte sectors that get uploaded to 10 hosts on the Sia network. So any small file is going to be at that 10X redundancy. For larger files, or files larger than four megabytes, we get some kind of efficiency by using a 10 of 20, or 10 to 30 ratio coding scheme. So four, that kind of over four megabyte sector size. So it's a 3X redundancy, but spread across 30 hosts, so you get a lot more reliability from that standpoint. So you can upload it through a portal like that. And that data is then available to any other portal immediately. So like I said, someone could be running a portal on their laptop. And if they had uploaded a file through our public web portal, they get a Skylink back, and they could take that Skylink and download it through their portal on their laptop and vice versa. If they upload a file on their laptop through their personal portal, they could share that Skylink with somebody else, and that person could download it through our public web portal. So that is where we kind of talk about our immutable global data layer where anything uploaded through any portal is immediately accessible through any other portal and with no kind of propagation time. As soon as it hits the network, any other portal can access it. And so that's kind of the data flow of a simple kind of uploading a Skyfile and downloading a Skyfile.

And then I can talk about the registry aspect of it. And so through the – In our SDK that we offer for developers, we talk about it as SkyDB. It's a kind of a database on Skynet. And what it's doing is storing and kind of a key value pair of basically that Skylink in that key value store on the host. So it allows for updating the file that that registry entry is pointing to. And mostly how

people are using that is using the file that gets uploaded as a JSON kind of metadata file. So if you wanted to upload a profile information about someone, for example, that might change over time. So you want that to be mutable. So every time you would edit that file, it would update that registry entry, so that when applications are pulling the latest information about a user, they're getting that latest profile information. Did that help kind of clear things up, or does that bring up some more questions?

[00:17:31] JM: It does do both of those. So talking a little bit more about usability. Like I assume that there's – If I'm writing this data to servers that are distributed around the world, is my data encrypted on these different servers?

[00:17:52] MS: Great question. So it can be or cannot be. So that kind of gets us into the idea of one of the latest feature sets that we've released that we call MySky, and it's how we handle identity on Skynet. And so when someone creates a MySky profile, they have a seed that's generated that generates their public and private keys for reading and writing data from the registry. And what MySky enables is this concept of almost like a global file system. So a user has one MySky account, and any application can then use that minus guy account to get information or store information about that user. And so I kind of mentioned the idea of a global file system. And so Mysky will be able to basically have public files and hidden files for a user. And those files can then be encrypted client-side if they need to be.

So public files would be files that the user is fine with any application immediately being able to read in order to kind of give them the user experience that they want on that application. So this would be things like if they are visiting a social media site on Skynet. So like a Twitter style application. They did a bunch of posts, and they have a bunch of comments and likes, and they have followers, and they have people that follow them. That's all the type of information that would be public files of the user that they would want another type of social media platform on Skynet to also see. So we actually have an application called Skyfeed, which is a community member who develops as just kind of that Twitter style application. So if I posted a bunch of stuff to Skyfeed, and then someone comes out with Skynet Gram, which is actually another project that someone just did for a hackathon. If I go to Skynet Gram, they should be able to pull those posts that I just made on Skyfeed that have pictures and have comments. So if there are people

that I follow on Skyfeed, I'll immediately be able to see them on Skynet Gram, because that's all part of my public kind of files. I let applications see.

And then, additionally, you might have files that – Like a diary, for example. If you keep notes about things you want to be kept private, that can be stored in your hidden files. And so only application that you would give access to would be able to see that application. And because users are logged in with their MySky account, which has a secure seed and public and private keys, only that user with those public and private keys can access the data associated with their MySky login.

[00:20:33] JM: Alright. And another question around usability. So can I mount a database on this kind of storage?

[00:20:42] MS: Yeah. Another community member is actually working on Skye SQL. So basically a SQLite-esque database on top of Skynet. And I can share that link to that GitHub repo with you kind of after the notes if people are interested in checking that out on the show notes. But yeah, people are looking at how to build that type of database style access onto Skynet that they're used to kind of locally.

[00:21:11] JM: But I guess today, you're more focused on just the more basic, like blob storage usage?

[00:21:18] MS: Yeah. A lot of what we are building an offering through our SDK specifically is that kind of SkyDB tool set that is that key value store for user's information, and a lot of that being kind of JSON metadata right now, since that's kind of the standard format for a lot of data on the web. And because there's a lot that you can do with that – Because Skynet shifts the paradigm of how people are thinking about applications, and the application is very much now kind of single user focus more than it was before, a lot of the solutions that people might just immediately go and throw a bunch of information into a database in the past, we're learning new ways to kind of solve those problems in the Skynet context.

[00:22:11] JM: What are some of the hardest engineering problems you've encountered when building Skynet?

[00:22:16] MS: I think some of the biggest challenges that we're hitting right now is really kind of pushing these new paradigms of application design. So when I talked about MySky, MySky is something that we call a DAC, or a data access controller. And so because users have their data publicly available to all applications, we want to be able to make sure that any developer can develop on Skynet and that we also are supporting a new developer or any developer from creating a bug in their application that would then corrupt the user's data and impact other Skynet applications, right?

So there's a tricky balance there. And so these data access controllers is kind of our solution for that problem. And we're using a lot of kind of IFrames and web worker API to offer these types of solutions. So these DACs are kind of mini applications within Skynet that other applications and developers can use to build their application. So MySky is the identity DAC that applications can now use to authenticate the user to make sure that they're logged in. I talked about Skyfeed, and so that kind of Twitter style application. So that developer has developed a feed DAC. So if other applications want to use – Have a feed in their application for like posts and comments and things, they can use the feed DAC to make sure that they are writing content the same way that other applications are and then reading content the same way that other applications are. And that's really all to – One, it makes the developer's life easier. But it also then makes the users experience a lot more consistent across Skynet.

And that's really a new paradigm and a new type of technology that we're offering people. And so there's just a lot of learnings that we're having with that. And each week, we are learning things that make the decisions that we made last week seem almost irrelevant. And so it's just that kind of pushing the boundaries of how people are developing applications on the web and really learning as we go in that sense.

[00:24:30] JM: So if I need to build an application, I'm going to need like typically not just disk style storage, but also like caching, in-memory style caching storage. Are you working on anything in that domain, like a shared memory decentralised, kind of caching thing? Or is that kind of out of the scope of what you're working on?

[00:24:56] MS: I think that would be out of the scope of what we're working on for right now. Yeah, in terms of some of that caching layer, yeah, it's understanding what's the problem that the developers are trying to solve? And how they used to solve that in the past and how it makes sense to solve that on Skynet. And so instead of planning out our roadmap from a feature set to think about how do we one for one replicate what people can do on the centralized web, we see something that we've created as a totally new ecosystem and a totally new way for developers to make applications that's going to give users a totally new experience on the web.

And with that, we are pushing out new features as we are learning how people are using the ecosystem and kind of using that as a gauge of what are people interested in building? How are they thinking about building it? How does it make sense to build that on Skynet? And what's the right solution for that? And really kind of taking that approach to have the ecosystem and the tooling set really evolve together. We obviously have our views of what Skynet can do and what Skynet can offer. And we are encouraging users to kind of think to that future state and doing that through various hackathons where we highlight different feature sets that we've released and encourage the user. And so we're really excited about the way that the ecosystem and our tooling is evolving together and kind of pushing each other to get to where we want to go.

[00:26:44] JM: You've spent time working at more traditional companies. And I'd like to know like how do you think software engineering at a normal like centralized style company, how does it compare to a decentralized company?

[00:27:04] MS: So actually, my previous career was in mechanical and process engineering. And so when I switched into this offer space, I basically found what was then Nebulous, and now Skynet Labs kind of with that transition. So I actually have only worked at a decentralized kind of startup company. In my past experience of working in more traditional company was more in the process engineering kind of manufacturing realm.

[00:27:36] JM: Can I ask you to theorize then?

[00:27:39] MS: Sure. So something that we are excited about, because we see Skynet as a new ecosystem for a decentralized internet and decentralized application development, taking

that same ethos of decentralization, giving power back to users, is Skynet giving power back to developers and allowing developers to build out these really powerful, really unique applications without having to build out these monolithic companies at the same time.

And one of the things that I'm kind of specifically thinking about as I say that is, today, if someone wants to build a new social media platform, they have to get over the same network effect hurdles that every other social media platform had to get over. They have to get that network base that encourages other users to join the platform. And it's a huge kind of hustle in the beginning to kind of get that user base going. I think a lot of that work really drives these software companies kind of ballooning up in size. So they need a huge marketing team, or they need to use sales team. They need to like put all this effort and time and money into building a network.

On Skynet, because a nice side effect of it being user center and a user owning their data and a user being able to go from application to application, is that the Sia ecosystem has one network effect as a whole. So as the network effect, and as Sia and Skynet, or as Skynet grows, every application benefits from that. So say the application that really tips the scales for Skynet is a meme generator. People get super psyched up about generating these memes. It just hits the right tone. And then there's hundreds of millions of people on Skynet. Any new application that joins Skynet now has access to all of those users. And all of those users are already on the Skynet ecosystem. And they can go over to this new application. And any data that they have accumulated that relates this new application is immediately usable and viewable by that application and by that user. And it doesn't require any other users to even use the application. So a great example being we go back to the Twitter style application.

Say there's one Twitter style application on Skynet that explodes. Everybody loves it, and everyone is using it. And then it starts kind of taking a turn from a UX perspective that somebody doesn't like. They're like, "Oh, I don't like how they're changing the fonts. I don't like how they're changing the layout of things. I'm going to make a new application. Same style, but I'm going to change the UX and focus on different things and whatever the changes may be." One user could go over to that new platform – Or that not that new platform, that new application, and they would be able to interact with all the other users that are on the other Twitter. And so they don't need to be on the same Skynet application to interact with each other.

And I think that is something that is super powerful and allows for an application to be incredibly powerful and have a huge audience. And also, at the same time, be manageable by like a team of one, two, five engineers, and it enables creative minds and passionate developers to really push out a production-ready application and forget about all of the challenges that go along with scaling an application, when it comes to infrastructure, when it comes to network effect and all those other things.

[00:31:49] JM: Can you tell me about the different infrastructure pieces that go into Skynet. And I'd love to know about programming language decisions, framework decisions, just general engineering decisions and software architecture.

[00:32:04] MS: Sure. So I guess I'll start with Sia, because that's where we started. So the Sia network, we have, like I talked about the renters and the hosts and the different nodes on the network. And we programmed the Sia network in Golang. So it's basically 100% Golang. And that's something that our founders made that choice early on, and it's really proven to be a really – Time and again, a really beneficial choice for us in terms of the performance of Go and the tooling around Go and really how it works for this peer-to-peer network. And so the portals and kind of what we call the Skynet daemon, or SkyD. So that is a Golang project, a Golang product. And on top of that Skynet daemon – So if someone wants to run a portal on their own machine, all they need is the Skynet daemon. So they can download the binaries from our GitLab or our website, and run the binary, run the portal on their machine. Be able to upload and download and view content in their browser locally. And that would be all that they need. When we go to kind of the web, and the web portals, and the tooling for developers, that's where we get into more of the web languages in terms of our main focus from an SDK standpoint and a tooling standpoint is JavaScript SDK. And we write that in TypeScript for the benefits of type safety. And JavaScript allows us to take advantage of all of the Mozilla API tooling. And if we're going to build a powerful web application, JavaScript is the language of choice.

And so the web portal is another open source repo that we have. So anyone that wants to offer their own web portal can run it. And it's a number of Docker services that kind of comprise the web portal. One of the Docker services being SkyD. And then we have various other applications to help build the kind of website of Skynet. So we have our own homepage, siasky.net, but we also – We have kind of a landing page that we'd originally produced for

people. So if they want to run a web portal, they have kind of their own like webpage ready to go as well.

So like moving into the SDK, which is where a lot of the kind of power of Skynet comes from, and for application developers. So when you go to Skynet, or you go to an application on Skynet, most of the time you're going – If you're going to download a Skyfile and just a Skylink, you would see `siasky.net/` the 46 byte Skylink. And that's going to download or render the Skyfile that you had.

But because I said Skyfiles are immutable and the Skylinks are immutable. That doesn't work well for application development, right? Because if every time you make an update to your application, that would mean when you upload that application to Skynet is a different Skylink. And that's no good. You got to then point people to a new Skylink. And so the solution that we had for that was partnering with a project in the space Handshake, or Namebase. So Namebase has platform on top of the Handshake protocol, and so with decentralized domain. And so we built Handshake into our Skynet portal. So you can buy a Handshake domain and then set it up to point to your application. And so when you go to – I've kind of had mentioned a few names of applications that are built on Skynet today, like Skyfeed. To go to Skyfeed, you'd go to `skyfeed.hns.siasky.net`. And that's going to be the static URL that you can always go to. And so as the developers of Skyfeed updates their application, they can update the record settings of the Handshake domain. And so their application gets updated, and users don't have to change their link. So that was a really great partnership for us in terms of being able to have a really great solution to that problem or the challenge of Skyfiles and Skylinks being static and, obviously, applications changing all the time. And so when we get into the SDK, we have a lot of – I kind of mentioned some of the tooling around. There's obviously the basic upload and download features. There's the SkyDB toolset, which is kind of the key value store of how we're handling dynamic data. And then now this concept of DACs, and data access controllers. And so Mysky is part of our JavaScript SDK that we offer. And so that's a built in tooling. And you would actually access that by – A MySky DAC is hosted at `mysky.hns.siasky.net`. And so it is an application kind of living in Skynet to service other Skynet applications. And same thing with the profile DAC. So the DAC that manages the profile information of a user, like their avatar, their first name, their last name, their contact information, their social profiles, like that all can be found at `profile.hns.siasky.net`. And so we're super excited about kind of how that's progressing

and the types of architecture that our developer ecosystem is helping us to face and solve challenges for and kind of build out as we go. So that was kind of like a relatively high-level summary of what the stack and some of the components are. Happy to kind of answer some specific questions within there, if you have any.

[00:38:01] JM: Maybe we can dive a little bit deeper into what happens when I spin up a node, a Skynet node, on my server.

[00:38:12] MS: Sure. In terms of like what the node is doing, or what you can do with a node?

[00:38:21] JM: Exactly. Well, just what it what it does when it boots up and like allocates? And what kind of processes it starts to run to communicate with the overall network?

[00:38:31] MS: So the SkyD daemon is the main portal piece of software. And that is the the decentralized Sia network. And when we start up a portal, it's going to sync the Sia blockchain, so it's up to date. And once it's synced, we can then start forming contracts and file contracts. And so file contracts are one of the core things that makes the Sia network unique. And the smart contract that the Sia network uses for kind of proof of storage of hosts and between hosts and renters. And there are state channels that also support the payments between the hosts and the renter. So I can get more into detail on that in just a minute.

So let me backup. So after the portal is launched, it's a recognized node on the network, is synced to the blockchain. The portal can start forming contracts with hosts. And so we allocate what we call an allowance. So basically saying, "This is how much money we're willing to spend each contract period." A standard contract period being three months. And so every three months, the contracts with your hosts renew. And so it allows, which is beneficial for both the host and the renter in terms of it keeps the contracts fresh with the hosts so the hosts aren't being burdened with super long contracts with renters that might not be using it. But similarly, the renter is constantly, at the end of contract periods, is challenging the host and making sure that they're actually storing the data and moving, kind of renewing with the host that they see is as good.

And so in the forms contracts, it will end up forming contracts actually with every host on the network. And the reason it does that is to create payment channel between all the hosts in the network. And so, previously I had mentioned that any content uploaded through one Skynet portal is available to any other Skynet portal. And that is due to the fact that all portals are connected with all hosts. And so if a portal itself did not upload the data, it can send that request to access the data to all the hosts on the network through those file contracts. And for a portal that is uploading data and downloading data and servicing users, every byte that is transferred is paid for. And that's that economic incentive that we had also talked about earlier on. And all of that gets recorded in the file contracts. And so anytime a renter sends new data to a host, the host accepts it. They update the kind of revision of the file contract and record the new payment of the file contract. And that information is saved to disk on both the renters and the host. And that revision number helps to keep them in sync. And the reason it's saved to disk and all that payment information is stored in the file contract, it allows for us to not bloat the Sia network and the Sia blockchain with all these micro transactions of small transfers of data with whether it's upload, download or kind of look up requests.

And so each file contract is just two transactions on the network. Basically, the transaction that created the file contract, and then the transaction that finalized the file contract. And it doesn't matter how many transactions or interactions between the host happened in between them. It kind of boils down to just those two, which helps keep the Sia network as a whole nice, and lightweight, and clean. So that's kind of more into detail how the hosts and the renters, or the hosts in the portals are connected on the Sia network and how that portal then sends information to the hosts and how it receives that information from the hosts.

[00:42:21] JM: Something we kind of glossed over or didn't really touch on very much. We dove into the technical stuff pretty quickly. Applications of decentralization, like my sense is we don't really have many decentralized applications quite yet. Plenty people have talked about a decentralized Uber, a decentralized Airbnb, but it just hasn't really existed yet. When do we get to decentralized Internet? And what are going to be the first applications?

[00:42:50] MS: I think we would say the decentralized Internet is here with Skynet. And some of the first applications that we've seen have been things like Skyfeed, which is basically a

decentralized Twitter. Another team in our community just released Skychat, which is like a decentralized Discord server. And we have a lot – And so one of our big focuses has been building out our developer community to build out these applications to see where the ecosystem goes with this. And so people can start using Skyfeed today and kind of posting like they would on Twitter and interacting and following with people. People can jump onto Skychat and start making their own Discord servers, and having chats and channels and things there. And so the infrastructure for the decentralized Internet is really here with Skynet. And we are starting to see some of those early applications that are showing that it is possible to make a usable, like nice UX, like nice UI, decentralized application for the decentralized Internet.

[00:43:59] JM: But do you have any ideas about like when we're going to see something and like what those first applications might be? The ones that people like, really want to use?

[00:44:09] MS: I'm not quite sure what the application will be. And I think that's something that we as like a community and company have kind of been careful not to put in people's minds. We like to keep talking about what are the potentials and like what are people using it for and like what they can do with it and kind of really seeing where it comes from the developers. Where it comes from the users? Like what are they asking for and what do they really want?

I think somewhere that we see a big opportunity is going to be in that social media or content creating area for two reasons. One, because I think there's been a huge shift and in just kind of mind share of everyone in terms of some of the issues with today's social media, platforms, and really starting to struggle with the power that some of those platforms have. And so looking for alternatives to that, we see there being a huge opportunity for a social media platform on Skynet to be that catalyst to kind of pull people to the decentralized Internet.

And I think, also, too for content creators, there's a huge opportunity there as well. And that the example of a content creator on YouTube spends years generating content, generating videos, building a user base, and they're totally locked into YouTube. If YouTube cuts them off, like they have to start from zero again, or they don't have their followers anymore, or that's not necessarily transferable to another platform. So content creation on Skynet, because the user owns it, and because in that case, the content creator would own their content, they can produce that content and share that content on the Skynet application that works for them. But

again, all of their followers could be viewing and consuming their content on a totally different Skynet application. They don't have to be on the same one. I think that's a really big opportunity for the decentralized Internet and for Skynet to give that power back to content creators to be able to generate really pure, authentic content that isn't being degraded by the creator having to adjust their messaging to tweak how the algorithm picks up their videos. To be able to get it in front of most people to maybe get a couple of dollars in advertising revenue.

On Skynet, Skynet also allows for what we call recursive monetization. And so because every bit of data that is transferred on Skynet is paid for, it means that if we use the blog example, a writer writes a blog. They might use a bunch of stock photos, and they might choose a specific font. And then they have an application that they post the blog on. And so you have the blogger, the photographer, the designer of the font, and you have the developer of the application. On Skynet, anytime that that blog is read, all four of those people can get a direct payment for that blog being read. And so there is that direct transfer of value between the users and content creators or value generators. So whether it's content creators, whether it's developers, whether it's designers, whether it's artists, whatever it may be, there's that direct transfer of value from the moment the value is consumed, as opposed to relying on these tangential means of monetization that we use today with advertising. And I think that's a piece that is also going to be super powerful. And so we just need that Skyfeed or that decentralized medium to kind of come along and show people that you can get value directly for, or you can get paid directly for that value that you are providing people.

[00:44:09] JM: I think it's a bright future. I think it's pretty interesting. I await the earliest decentralized applications. And I think they'll be interesting to watch. Matt, thanks for coming on the show.

[00:48:30] MS: Thanks so much for having me. I really enjoyed speaking with you.

[END]