Tonk: A Local-First System for Small Software

Eileen Wagner · 2026-10-09 · English

English transcript (with translations of the Chinese questions). Edited transcript of the presentation and discussion on 9 October 2026. Repetitions and clear recognition errors have been corrected while preserving the order, examples, qualifications and questions. Product status and plans reflect the talk date. Timestamps refer to the edited video.

Introducing Eileen

00:00:00 · Eileen Wagner

Hello everyone. I'm very happy to be here. I'm Eileen. As I said earlier, I can answer questions in Chinese, but I think English is better for explaining these technical terms. You can see which words we use and consider how best to translate them into Chinese. Questions later can be in Chinese or English—or German. I'm a designer from Berlin. I'm going to switch to English now, because the rest of my slides are in English.

00:00:35 · Eileen Wagner

So, yes, I'm a UX designer, in privacy and decentralization. So I work a lot with open-source projects that, care about user privacy and how to distribute computation and power in general. I am based in Berlin, as I already said, and I also co-organized the Local-First Conf in Berlin.

00:00:57 · Eileen Wagner

I was there the first year, gave a talk the second year, and the third time I was a co-organizer. Who knows what I'll do next year?

00:01:05 · Eileen Wagner

I'm also very active in a bunch of communities, among them are Open Source Design, which is a little group of people caring about design and open source, as well as Decent Patterns, which is a library that I started many years ago collecting patterns for decentralization projects. I can talk more about that too. And DWeb, like, basically there are a couple of communities that I'm active in.



Introducing Eileen · Frame 00:01:20

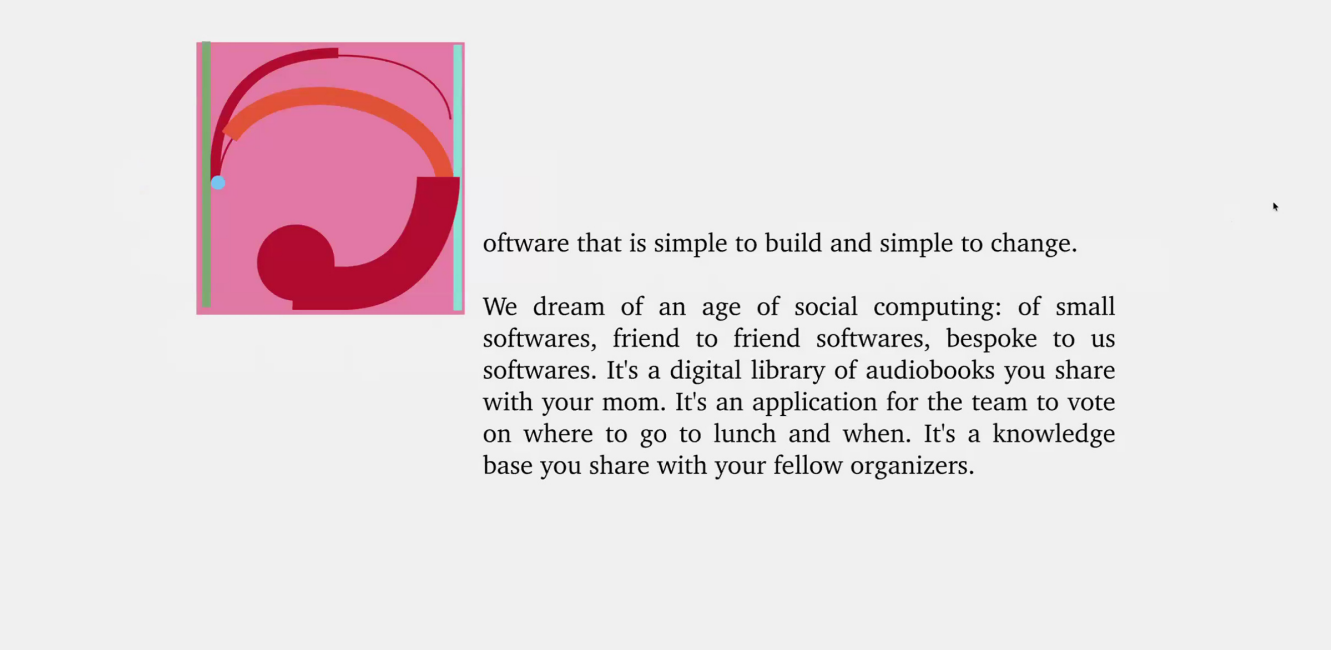
00:01:35 · Eileen Wagner

But right now I am mostly focused on building a product. I am the head of product at a small startup called Tonk, and that's what I would like to talk to you all about today. So, what is Tonk?

What is small software?

00:01:49 · Eileen Wagner

Tonk is a unified system for all of your small software. This is the basic idea, that we're trying to introduce here. I'll get to why we are at local-first, this talk series in a second, but for now, imagine it's mostly about small software. Small software we understand as software that is simple to build and simple to change.



Small software · Frame 00:02:20

00:02:15 · Eileen Wagner

I'm just going to read this little bit because I really like it. We really dream of an age of social computing, of small softwares, friend-to-friend softwares, bespoke-to-us softwares. It's a digital library of audiobooks you share with your mom. It's an application for the team to vote on where to go to lunch and when. It's a knowledge base that you share with your fellow organizers. It's really this, like, idea of small personal software.

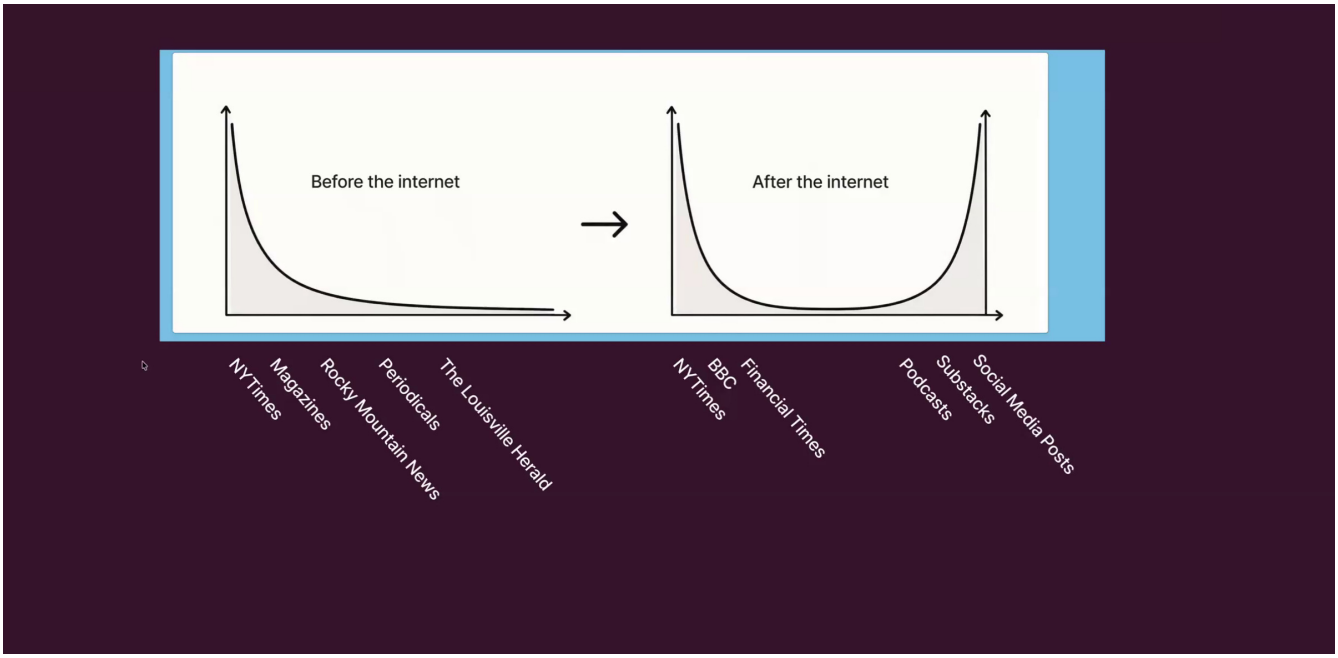
00:02:44 · Eileen Wagner

And it's really something that has been made possible very recently. This is a bit of a bigger theory. I promise I won't go too deep into this, but you can kind of imagine small software being kind of a parallel development as general news media.

From the internet to automated coding

00:03:02 · Eileen Wagner

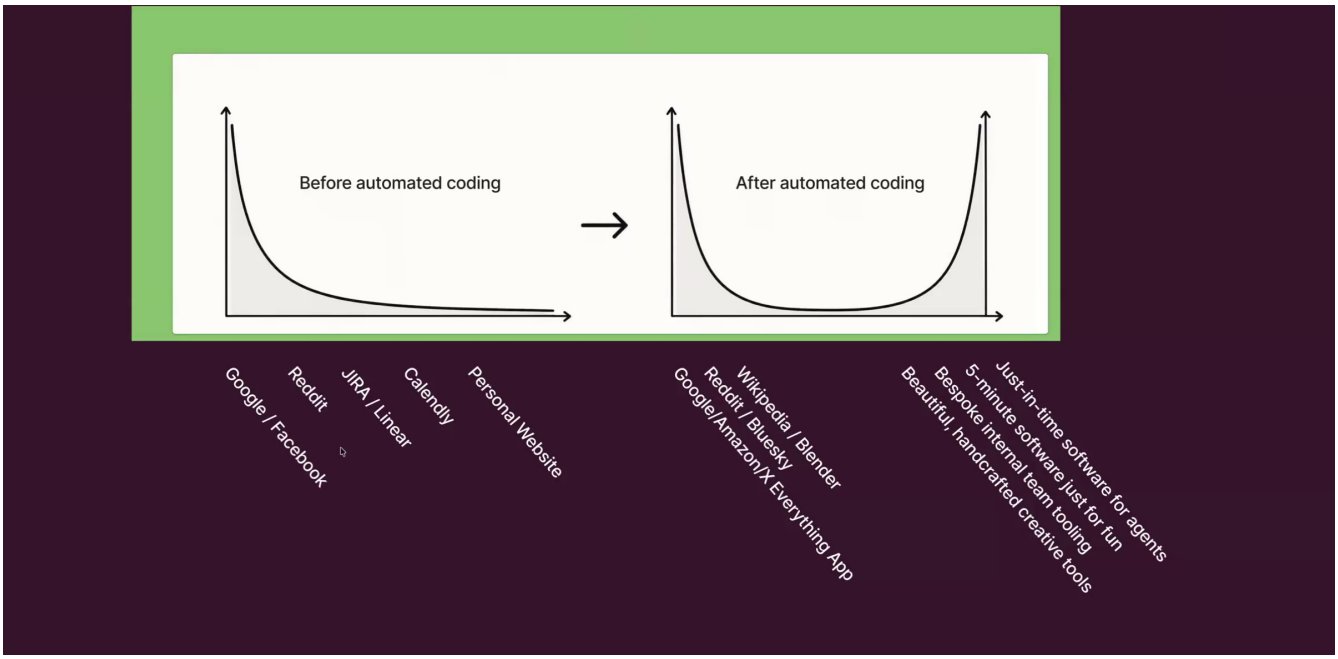
So news media before the internet was, you know, big players: New York Times, magazines. And then as you go into, like, smaller and more personal use cases, maybe you have the Rocky Mountain News, the periodicals, the Louisville Herald. This is very American, I apologize, but the idea is you go, you go smaller and you go more local. But then at the very tail end of this, there is basically nothing, right? There is nothing. There is no news magazine that serves exactly you and your friends and family.



Before and after the internet · Frame 00:03:20

00:03:36 · Eileen Wagner

After the internet, we see this, like, explosion of different types of media that serve your personal, context. For example, now we have podcasts, we have Substacks, we have small social media posts that really you can kind of cherry-pick and is really catered to you and your needs. This is what you might argue what the internet has done to media.



Before and after automated coding · Frame 00:04:20

00:04:03 · Eileen Wagner

I think there is a similar thing happening right now with coding. So coding is, you know, before AI-assisted automated coding, we had the big tech companies. We had Google, Facebook. We had people who made particular software for very specific goals, like company, you know, product teams collaborating. Calendly is whatever, like, scheduling. You can maybe use a building block software for your personal website. But basically there are things you can use for specific use cases.

00:04:39 · Eileen Wagner

But there is also a bit of a tail end here. Like, there is no money to be made on this end, and so people don't really have super bespoke software. And so if you really wanted to have something just for you and your friends, you have to basically write it yourself. You have to be a developer, or you have to know a very nice developer who would do it for you. So that is sort of the paradigm we had before automated coding.

00:05:06 · Eileen Wagner

And now I think we see an influx of different ideas. We have, really beautiful handcrafted creative tools that people build for one another. We have small internal team tooling that people are building on the fly. Maybe sometimes just, like, a five-minute software just for fun. Like, I thought I had a funny idea, I made it, and now it's a meme, but it, you know, took me five minutes to do it.

00:05:34 · Eileen Wagner

Or even, you know, if you think about agentic use cases, just-in-time software. So software that you use just in this context. You write a script for it, you use it, and five minutes later it's trash. You don't need it anymore, right? So there is a sense of, software that really doesn't have to live very long. And now this is exploding right now over here. And this is what I think Tonk is really interested in and what we also would call small software.

Home-cooked software and local-first

00:06:03 · Eileen Wagner

The idea is not new. Maggie Appleton already talked about this a little bit in the 2022 talk called Home Cooked Software and Barefoot Developers. This is basically another iteration, another articulation of small software. It's home-cooked apps.

Editorial note: the speaker says 2022 here; the slide gives 2024 for Maggie Appleton's talk. Both are retained as they appear in the recording.



Maggie Appleton, *Home-Cooked Software and Barefoot Developers (2024)*

Home-cooked apps · Frame 00:06:20

00:06:23 · Eileen Wagner

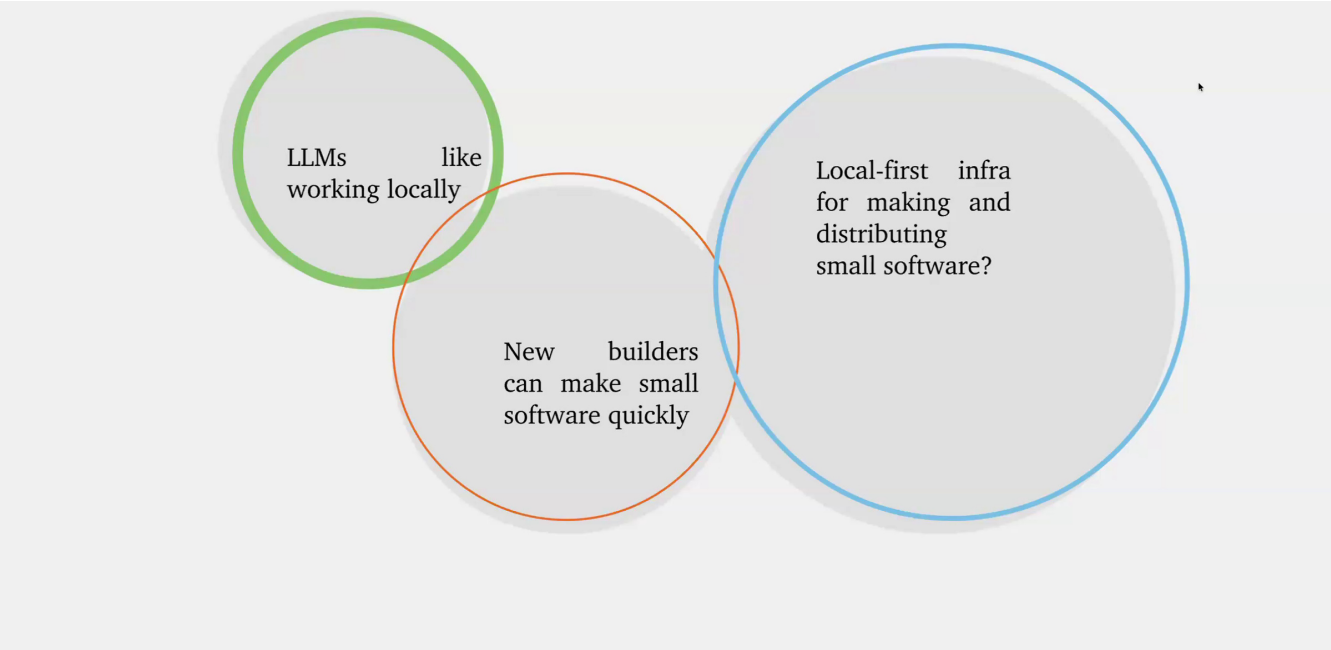
Now, why am I giving a talk about home-cooked apps? Well, we know some things about AI-assisted coding. We know that LLMs like working locally. We know that coding assistants really just want access to the CLI, and then they can do a lot of things. They can get their own tools. They can do all kinds of direct data manipulation.

00:06:45 · Eileen Wagner

We also know that new builders, and by new builders I mean people who maybe didn't code before but now are interested in learning about coding small things, are just making things because, you know, the AI does it for them. They may not understand all of the things underneath the code, but they're happily using them. But they really like building, right? They really like making things, for themselves and for each other.

00:07:12 · Eileen Wagner

And maybe more than a developer, they have an eye for design, or they're really creative about the ideas. So a new builder is a person that we're seeing more and more, and they can really make these small pieces of software quite quickly.



Local models and local-first · Frame 00:07:20

00:07:27 · Eileen Wagner

So now here's the big idea, right? Okay, you all just listened to a bunch of talks about local-first infrastructures. Well, can we create an infrastructure that is local-first and that can help people make but also distribute small software? So that's, that's what Tonk is trying to do.

Owning your data and changing your software

00:07:48 · Eileen Wagner

If this is indeed the world that we want, people distributing small software, then let's build it the way we like it, right? One is this idea of post-platform. You really can own your data. The tools come to you. You can bring your own agent. You don't have to stay with, you know, the big hyperscalers. You also don't need to have a platform lock in your data. It's all local to you. You remain in control.

00:08:18 · Eileen Wagner

It's local and private. We put in best-in-class cryptography for decentralized identities, right? So you really don't have a central registry anywhere of all of the information on the people. The people kind of retain, like, I mean, you all are already using DID keys, right? But the idea is just, let's have decentralized identities that you can mint yourself and control and manage yourself.

00:08:43 · Eileen Wagner

Let's make it open source. Let's make the data inspectable so, you know, it's not just a black box that the AI is coding into, but you can really see what's in the code.

00:08:55 · Eileen Wagner

Let's use formats that are really, forever formats. Things that are going to last forever, that even in 20 years, when people dig out a hard drive, you know, from, like, you know, the sand, they will find a data format that they can use and use, and understand and reuse.



A post-platform, open and malleable system · Frame 00:09:20

00:09:14 · Eileen Wagner

And of course, malleability. Malleability is an interesting idea that we're exploring. It's not necessarily, I think, a constraint in any of our designs, but it's just now possible because all of a sudden you can both make a thing, one time, change it another time, or you can constantly change it on the go. And that's also a really interesting thing for us to explore.

00:09:39 · Eileen Wagner

So before I go any further, I would just say, let's do a demo so that you can see what I'm talking about. I am going to just share with you literally what my, what my Tonk looks like, what my day looks like, all the things that I work on on a daily basis. Okay, so I was just here. All right, so, Tonk.

Live demo: a message board built by Claude

00:10:05 · Eileen Wagner

Let's go to Tonk. Tonk right now has a little bit of a home. Here are a lot of the Tonks that I own. What would I do? I guess I would make a new space. I would say hello. I guess you all don't say local-first; you say 'bendi youxian,' local-first in Chinese. Okay, hello, Local-First. I'm going to create this space. It's going to take a little while, but really it's just a blank canvas at this point. I can do whatever I want.

00:10:40 · Eileen Wagner

Here's a little menu bar that I can put on top, and I have an agent link that I can copy. It just generated a link, which is actually a UCAN delegation. It gives whoever has this link the capabilities to read and write into this space.

00:11:00 · Eileen Wagner

So I'm like, okay, great. So what am I going to do? I'm going to copy this link, and I'm going to go to my, cute little Claude over here. I'm going to go, hey, Claude, connect. Oh, there you go. Connect to this space. Here you go. It's a space link. And I say, what should we make? Build a message board for my call. It's live now.

00:11:33 · Eileen Wagner

And then I can say something else like, do I have a static link here? I don't. Okay, I'll just maybe just give them this link. Follow the design here. All right, that's my build flow, right? I just have a link. I have a location, and then I have whatever I want to give my coding agents as a prompt, and the coding agent will start doing things.

00:12:03 · Eileen Wagner

I'm going to put it on auto. It's been behaving fairly well, so I'm going to trust it with that. Claude's doing Claude. It gives me a little note saying, 'Agent connected.' Wonderful. I'll just let Claude do its thing.

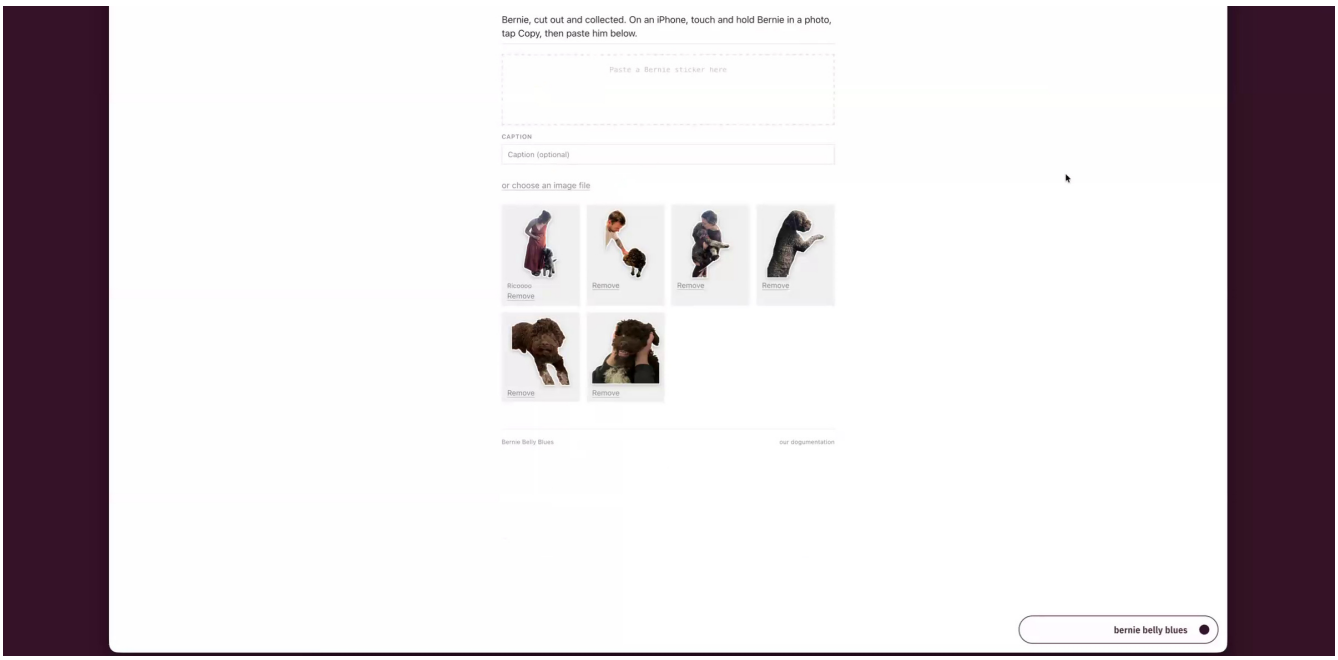
Dog photos, quartets and a shared soundboard

00:12:17 · Eileen Wagner

But I can show some of the Claude's, Claude design spaces that I've built so far. So this one is a little app that I made for me and my friends. I have a dog, and the dog travels around my friend group a lot, and we just want to see what the dog is up to. So this is just basically a little, collection of pictures of my dog, Bernie. And it also has a little manual. I thought it would be really nice to just document, like, when my dog eats and goes to bed and what the rules are, what the commands are. And so, you know, the humans can kind of deal with it.

00:12:56 · Eileen Wagner

And then I gave it to my friends, and someone was like, oh, it would be really cool if we had, like, a little sticker as an avatar. So I added the feature to have stickers. So you can just copy and paste a picture from your phone. You can drop it in here, and then you have a sticker of you and the dog, which you can use as your avatar. Personal software, small software.



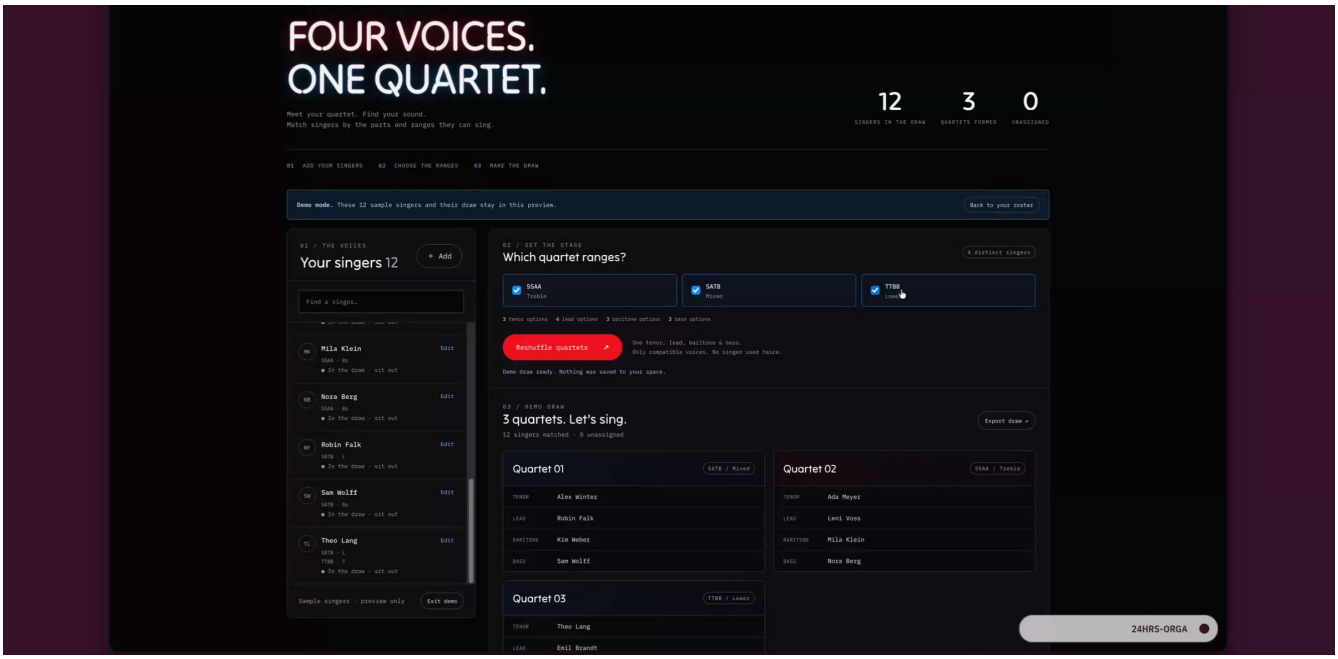
A dog photo album shared with friends · Frame 00:13:20

00:13:22 · Eileen Wagner

Here's another thing I made. I do a lot of singing in my free time. And then we had this crazy idea that we were going to get a, like, 20 people together. We're going to sing together, and, there's going to be randomly generated quartets. So we had to basically add some people in, you know, give them a name, and basically say, okay, what voice parts do they sing? And then, there you go. And then you can kind of generate quartets. There's a little demo here. So let's say I entered a lot of people's names. Generating quartet.

00:14:00 · Eileen Wagner

This is not software that anybody would sell you on the market, right? There's no money to be made here. And it would take probably a good developer half a day to really make it finished, you know, with all the constraints, with all of the people testing it. And now I can just code it. I coded it, like, it looks very sloppy. That's true. But it took about five minutes for this to be generated. So it's awesome.



Random quartet assignments · Frame 00:14:20

00:14:28 · Eileen Wagner

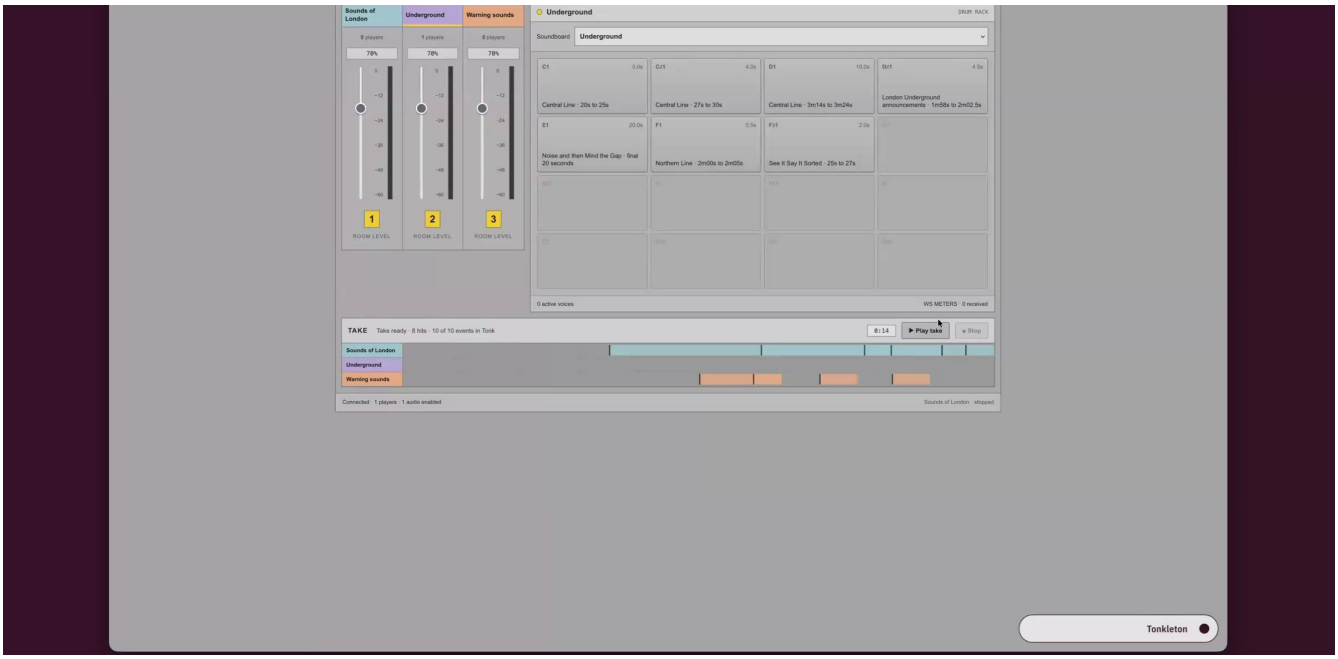
I have another one. Is Gavin in the call? Let me see.

00:14:36 · Gavin Owens

Yeah, I'm here.

00:14:37 · Eileen Wagner

Gavin. Hey, hi, Gavin. Okay, Gavin's in the call. Gavin made this, and it's a little, audio library of different sounds that you can play with. And the nice thing is, I don't know, I don't think audio is being shared here, but, you can basically try it yourself later on and, figure out different sounds that you can play with. You can remix them. You can play them at the same time. It's sort of like a collaborative soundboard experience and just exactly the way that, you know, Gavin likes things to be.



A collaborative soundboard · Frame 00:15:20

00:15:11 · Eileen Wagner

So very different people have their own apps, and it has their own sort of fingerprint, their own preferences, and exactly the way they want it to be. So that's, that's what Tonk is right now. It's a blank canvas that you can paint however you want to paint it.

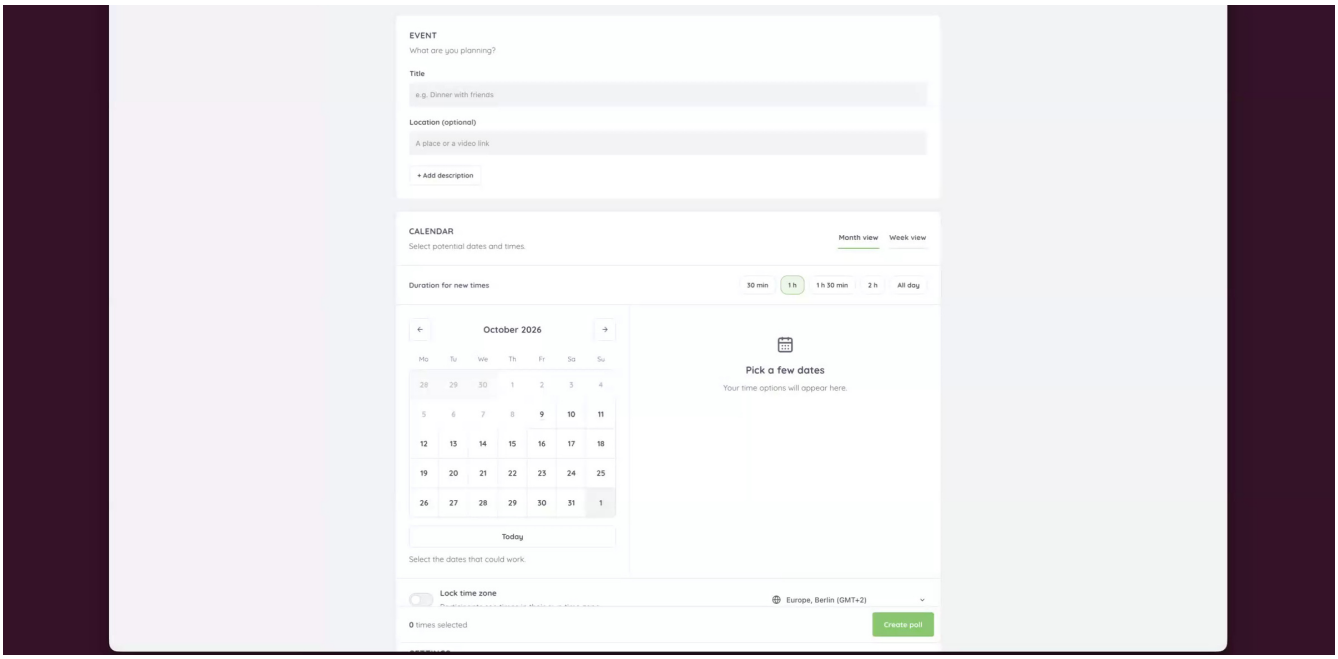
00:15:31 · Eileen Wagner

I'm going to see what happened here. There's anything? No? Oh, there's something there. Okay. It's still going. I think it's downloading some fonts right now.

Sharing apps and scheduling a get-together

00:15:45 · Eileen Wagner

But yeah, I think the interesting thing here is not just that you can build anything you want yourself, but you can also go and see what everybody else is making. So we have a little list over here of different apps that people have made, that they choose to publish. So, for example, you can have a little app called Canoodle. It's just an app that helps you, find a time to schedule an event. I think we all have seen some version of this when we try to, you know, find a date with friends or colleagues.



Choosing a time to meet · Frame 00:16:20

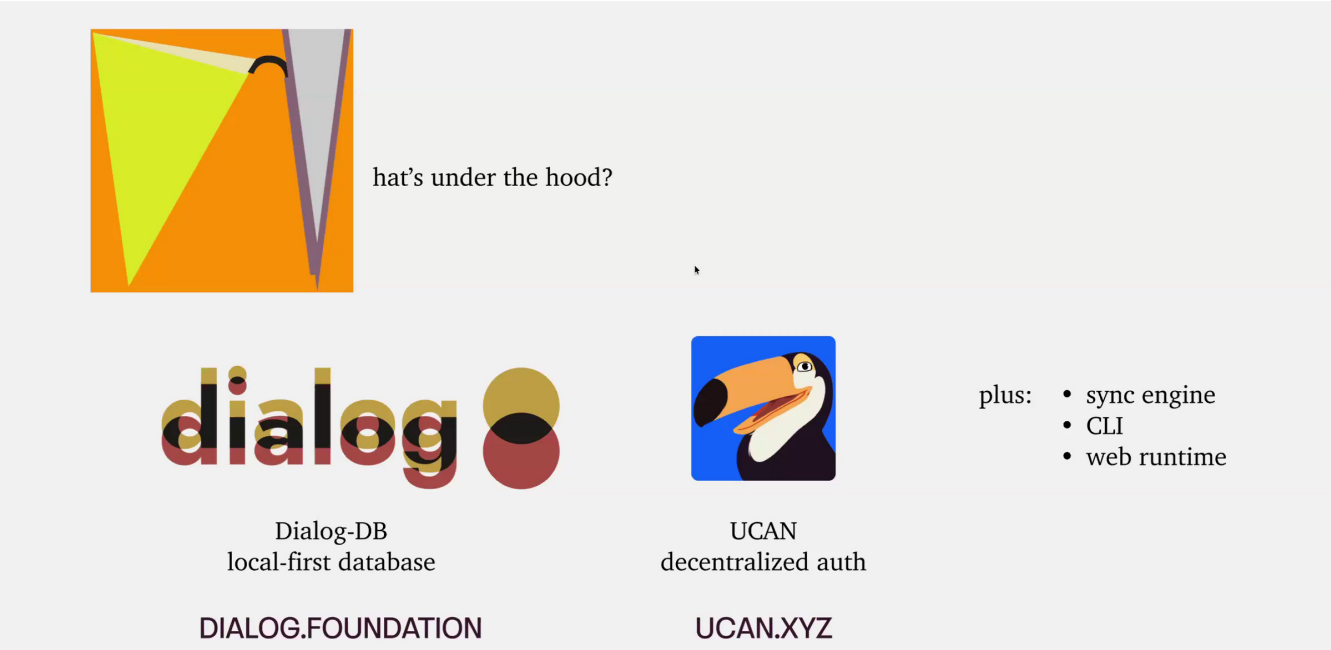
00:16:28 · Eileen Wagner

And this is just the same thing, except for anyone that I share this link with, which I do like this. I just say, copy, share, link. And then, I can share this link with someone, and then they will have immediate read and write access in this case, and they can put in their data, and then all the data comes back together, and we can determine when to meet. So that's just, like, very quickly what Tonk is at this moment.

Under the hood: Dialog and UCAN

00:17:01 · Eileen Wagner

Okay, I'm going to let this agent build just a little longer. I have way more things that I can show you, but maybe I'll go back to the presentation. What's actually under the hood? There are a couple of libraries that we're using and not just using, but also contributing back to, because our CTO is also one of the main maintainers and developers and authors of these libraries.



Dialog and UCAN · Frame 00:17:50

00:17:30 · Eileen Wagner

The first one is called Dialog, DialogDB. It's a local-first and CRDT-based database. You can see more information under dialog.foundation, but the basic idea is you have a semantic database that, writes everything in YAML. Yes, everything is good old YAML. It comes with both the application, the certain views on the data, but the data structures themselves. And it's just extremely malleable. It's what we call, like, it has an evolving schema, right? So the schema can evolve as you grow with it. It's not very strict like an SQL database.

00:18:14 · Eileen Wagner

It's really good for malleable, decentralized data sharing. We are also using UCAN. UCAN is decentralized auth for all kinds of applications. UCAN.xyz, if you want to learn more about UCAN. It uses DID keys. In our case, we are basing the DID keys on a passkey. Everyone can generate a passkey pretty easily these days. You have a passkey, and that's the basis of your identity.

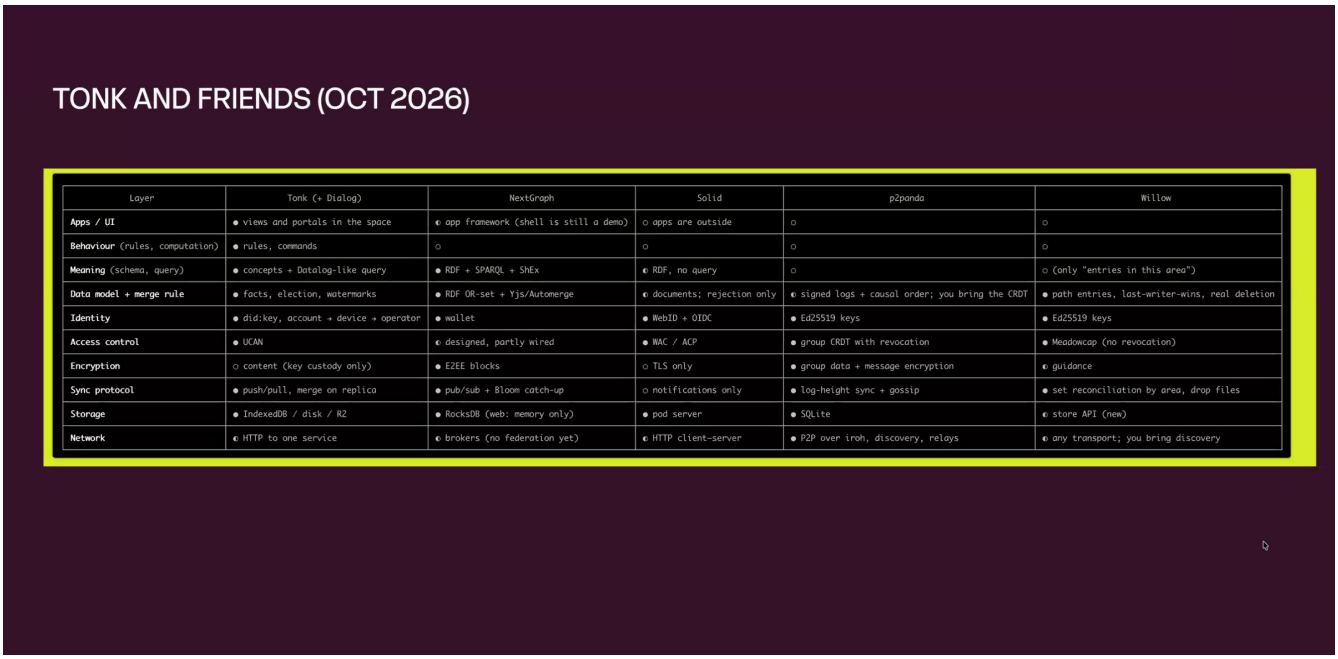
00:18:47 · Eileen Wagner

Beyond those two things, what we have built is really just a sync engine. We use a very basic Cloudflare R2 for syncing and storing, and we have a CLI for agents to read and write from the space. And of course, we have the web runtime. So there is a way for you to open up these spaces in the browser and then share links, et cetera. That's, that's what's under the hood.

Tonk and other local-first systems

00:19:13 · Eileen Wagner

And I also think it's worth, because I know all of the different projects that have presented in this series, I think it's worth maybe just having a quick look at what Tonk is as opposed to all of these other things. Because you have, you have talked about a lot of similar databases that are also local-first and CRDT-based. I think, okay, this is a very big and overwhelming table. I apologize. But the basic thing that I will say that Tonk and Dialog have, is we were focusing really on the apps and the UI level, right?



Tonk and other systems, as of October 2026 · Frame 00:19:50

00:19:51 · Eileen Wagner

So thinking about how the data model is stored, how it's compiling, what, how you can make the rules, how it actually bundles all into one single container that also contains the app and the UI is sort of the thing that we add over and above these other things that other people have built.

00:20:13 · Eileen Wagner

We currently don't have encryption, for example. Encryption is on our roadmap, but through UCAN we have pretty good auth. Every edit is signed, but it's not encrypted.

00:20:27 · Eileen Wagner

Same thing with networking. Those are also things that we could add in the future. We could easily imagine plugging in iroh, for example, for transport. But for now, it's just HTTP in the service that we are hosting, which, as you all know, comes with, you know, some benefits as well, because we can basically guarantee a certain type of experience because we're still running a dedicated version of Tonk on our services.

00:20:56 · Eileen Wagner

I'll leave it here for now. If there are more questions, I'm glad Gavin's here. He can also answer more questions. Okay, before I do this, though, I want to see if my Claude is done building. Oh, it's done building. Excellent. Okay, so, where are we? We're here. Oh, look at this. There's a space that just spawned.

00:21:17 · Eileen Wagner

This is a message board. What do we have? A message board for Local-First. You can write your message and have different ways of displaying it. You can have a big thing and a small thing. Beautiful. Great job. One-shot build, Claude.

Trying the message board together

00:21:38 · Eileen Wagner

So now what I can do is I can copy the share link, which is, again, a delegation of permission to this space. And what I can do is I can come to, you all over here, and I just paste it in the chat. So here you go. You can now access the space, and you can read and write.

00:22:02 · Eileen Wagner

You don't need an account to do so at this point. The right, the correct, account model is still like a little TBD, but for now, we're thinking every space has one owner, and the owner has an account, and everything else is as, you know, local as you want it.

00:22:23 · Eileen Wagner

So, oh, thank you. Someone's already writing. There you go. It's live too. So anything that you, write in here is going to be synced and is immediately displayed. Okay, I can also say, "Hi, there." here you go. Thank you. All right. Oh, wow.

00:22:50 · Eileen Wagner

I already talked for 25 minutes. I want to talk about one last thing, and then I would like to open this up for Q&A, and then I can answer some of your questions. So, this is a lot of infrastructure. You can do many things with it. As I said, it's a blank canvas, and we really just supply sort of the packaging distribution infrastructure. You, I mean, the builders, the builders would bring their agents to build things, and, you know, they will bring their ideas and their UIs. We have no control and no interest also in controlling that.

Three patterns: apps, knowledge bases and data containers



Pattern 1: small apps · Frame 00:23:20

00:23:33 · Eileen Wagner

But it is interesting to see that there are some patterns emerging, on how people want to use Tonk as a service. Well, Tonk as infrastructure, really. So the first one is just apps. Small software is a small app. So we've seen, or I don't know, I showed you a little bit of this already. You can just build apps with it. Like the Canoodle app is one where you can find a common, like a shared time with your friends and colleagues.

00:24:07 · Eileen Wagner

We also built one that we call Little Writer. It's just sort of a pad that you can write into, and it's all the edits are concurrent, and you can file that, in your registry. But really, it's just a, it's a pad. Nothing fancy.

00:24:24 · Eileen Wagner

I also built something that is called Trello. It's a Kanban board where you just put in little tasks, and you can assign them to people. It's small, it's reliable, you can share it with people, it's immediately useful. There's nothing frivolous about it. You can publish them, other people can use them. Small apps. Excellent.

00:24:56 · Eileen Wagner

There is another kind of usage, though, that we also observe and I would say also lead on our own, which is that you don't only need to have apps, on Tonk. You can also have, of course, all the data, be useful to a specific group, be tied to a specific group, and evolve with the group. And that's sort of a more, a knowledge basey, use of this.



Pattern 2: shared, malleable knowledge bases · Frame 00:25:20

00:25:24 · Eileen Wagner

So I want to show you just a quick, give you a quick tour, on our internal knowledge base. This is our own Tonk team dashboard, basically. We have built many different things. We're only four people, but we have lots of ideas and lots of things we need to look at every day. So this is sort of our issue tracker that we just built on Tonk. And it is, you know, a lot of data coming in and out every day and changing as we go.

00:25:54 · Eileen Wagner

We have, you know, on the product side, we have certain hypotheses that we want to test, right? So we put in some ideas and, you know, make our own interface on how to evaluate the hypotheses. Of course, we have, you know, people. I'm going to take vacation soon. So that's also in this giant team headquarters, basically. And the whole thing has grown a lot over the many months we've been using it.

00:26:22 · Eileen Wagner

We've been doing research, so there's a lot of things that is, we just sort of dump in there. We don't really think too hard about it. It's sort of a big vault, almost like the way some people use Obsidian these days, like as a second brain. We just dump it in there, and the AIs that connect to these knowledge bases are also just much smarter because they have way more context. They know what's going on.

00:26:49 · Eileen Wagner

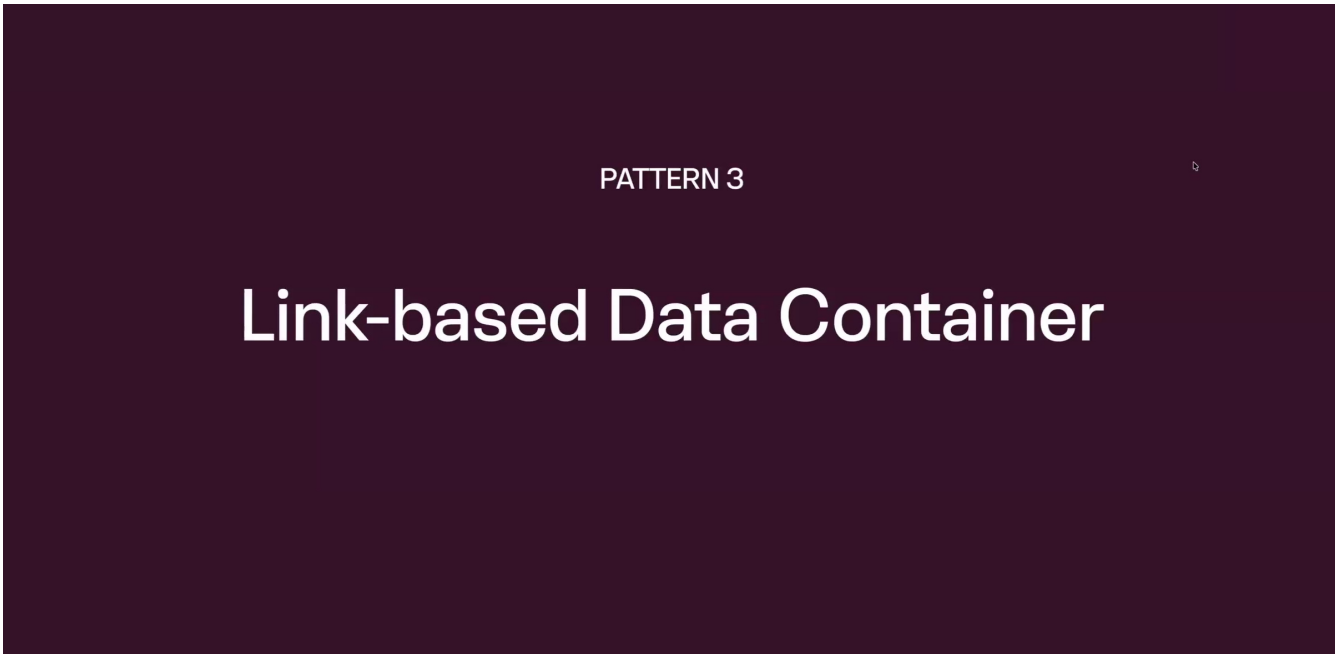
And what's more important, it's, when I don't want something to be a certain way, I can change it. Like, it's not one of those applications where, oh, you know, I build it once, and I, it has to stay that way. And if I build it, if I make any changes to it, then I have to basically do a migration of the entire database.

00:27:10 · Eileen Wagner

This is super easy to change. I don't think we have the time now, but I think I can make big and small changes to the space immediately, and nothing will be lost. Everything is still here, and it persists over lots of different changes.

00:27:29 · Eileen Wagner

And that's something that, we're beginning to see also, especially with, like, community organizing and whatnot, where people just sort of want a data repository, like a GitHub repository, but better, right? That they can share with their teams, especially if the teams are not, full of developers and coders.



Pattern 3: link-based data containers · Frame 00:27:55

00:27:49 · Eileen Wagner

Finally, there's a third type of pattern that is also emerging, which is what I call link-based data container. So you've just seen all of the things, contained within a link. And I think what, the future could look like is for these data containers to not even need a UI necessarily.

00:28:15 · Eileen Wagner

So what you can do is you can just paste the entire link into your agent and just have your agent do things, right? So, like, in my prompt, I'm giving my agent this project and just letting the agent work on it. And if the agent needs anything, they will ask me. Otherwise, you know, feel free to connect to, like, whatever, Gavin's agent for anything else.

00:28:39 · Eileen Wagner

Like, I mean, connecting, what does that mean? But basically, like, you can have a shared space. Like, agents can have a shared space, humans can have a shared space. It's super fascinating because it is, oh, well, everything, the data and the app are in one place. You don't need anything in addition to what's already contained in this one link.

00:28:58 · Eileen Wagner

It is multiplayer by default, right? Like, local-first and CRDT, so everybody has their own copy, but they can also concurrently write together. This also means you can have multiple agents at the same time writing to it and not get into weird conflict situations.

00:29:16 · Eileen Wagner

Because the delegation chain is through UCAN, you have provenance and attribution of agent changes for free. We're currently working on that interface as well.

00:29:25 · Eileen Wagner

And, oh, you can use it local only. If you have a local LLM, you have basically a really good local setup, or you can use it offline, meaning, you know, if you want to be online most of the time, but you're, you know, on the plane for a little bit, you can still use it and your changes would persist.

00:29:45 · Eileen Wagner

And I think also the YAML, so the format of the whole thing is inspectable and really fast to read and write. And just to show you what that means, I think we should, do we have? Basically, the data in all of these apps is basically YAML data. So you can really look at the, yeah, there we go. So, this is sort of the seed, so it's the basic things. The concepts are the concepts in the app that are defined.

00:30:20 · Eileen Wagner

This is all sort of like Datalog, like if you, if you speak semantic databases. But really, it's extremely readable. Agents have a really good time reading it. And, you can just read through the entries and inspect it yourself and also correct it yourself.

00:30:37 · Eileen Wagner

Okay. I think I'll stop here. But before I stop, I should probably check if there's anything else on the board. Ah, thank you. Wonderful. Okay, I'll stop sharing, and I, as promised, will switch back to Chinese.

Q&A: where the UI runs and Windows support

00:31:02 · Eileen Wagner

Okay, I can answer questions now. I will try to answer in Chinese.

00:31:06 · Host

Thank you, Eileen. That was amazing.

00:31:18 · Eileen Wagner

Thank you all for joining me and leaving all kinds of messages in there.

00:31:26 · Host

You can ask questions now. Here is one from the comments: is Tonk a program that puts all the interface design on the server?

00:31:54 · Eileen Wagner

Which interface do you mean? Tonk has its own interface for creating a new space or sharing a space with someone. Those interfaces are actually on the client, because it is local-first software.

00:32:21 · Eileen Wagner

Where your own interface lives is ultimately up to the user. You can put the whole design on your computer, especially if you use local AI coding tools such as DeepSeek or Qwen. You can work directly on your computer without a server. Did I understand the question?

00:33:13 · Eileen Wagner

There is also a question about Windows support. Gavin, do we have Windows support, or is that something we are considering?

00:33:25 · Gavin Owens

Like a Windows native executable?

00:33:30 · Eileen Wagner

Yeah.

00:33:30 · Gavin Owens

For the CLI, is that what we're asking?

00:33:32 · Eileen Wagner

I think that's what we're asking.

00:33:36 · Gavin Owens

I thought we added it. Let's take a look at the tags here. What do we have? We have Linux and macOS. I guess the thought is that you would be using some, like, node-based shell, which typically runs Linux binaries. Maybe that's wrong.

00:34:06 · Gavin Owens

If you can tell me specifically what binary you would like, we can make sure to add it. Like, how to, how your preferred way of running on Windows is.

00:34:15 · Eileen Wagner

Okay, thank you. We do not currently have that Windows support, but if you tell us which binary you need, we can try to make one.

Q&A: what Tonk adds to a web app

00:34:35 · Eileen Wagner

Someone says the website looks like a SaaS service. Do you mean our company, or something else? The idea is that browsers can now store data locally. You need a Tonk account when you want to share your space, because we currently provide the synchronization. We could also use iroh in the future. Are you asking about browser storage? Could you explain the question?

00:35:25 · Host

They mean the Tonk Network website.

00:35:36 · Eileen Wagner

Right. Tonk Network has not officially launched, but you can start using it. It is in beta testing, and you are welcome to play with it. Our data migration is generally quite mature, but I cannot guarantee everything will continue smoothly over the next two weeks, so please do not put important things in it yet. Trying it out with other things is fine.

00:36:14 · Eileen Wagner

Here is another question: compared with using AI to write a web app directly, what does Tonk provide? CRDT data structures, a registry and a container for running apps—is that a fair understanding?

00:36:28 · Eileen Wagner

Yes. First, there is the CRDT database. We have essentially standardized that structure. An app built on Tonk will be local-first and based on CRDTs, and it will use the Dialog language.

00:36:50 · Eileen Wagner

We also provide synchronization and our own service worker. You do not need to handle updates, deployment or complicated things such as DNS yourself. New builders in particular do not want to deal with that. Even if they can, they may not have the time. I can do my own hosting, for example, but I do not have time to manage it, so Tonk solves a very direct problem for me too.

00:37:33 · Eileen Wagner

Yes, if the domain is blocked, Tonk cannot currently be used. That is why we are considering something like iroh. Blocking the domain alone is a problem at the moment. There was also a point about a registry.

Q&A: desktop apps and clearing browser data

00:37:58 · Eileen Wagner

We are also considering a desktop app. From a design perspective, it offers several things. Someone has already mentioned IndexedDB: most users do not clear it for no reason, but it does happen. At the moment, we have a copy on R2, so the data is not lost, but that copy is no longer local. We think a local file system might be better, which would require a desktop app.

00:38:44 · Eileen Wagner

Another point is that packaging an app currently means giving you an iframe inside your browser. Tonk adds a layer between that iframe and the actual browser, so it can feel as though we are competing with the browser. Users may wonder why they have a browser, then a Tonk browser, and only then their app. Three layers can be confusing. A desktop app could itself become something like the browser.

00:39:49 · Eileen Wagner

Your browser has overlapped? If you mean duplication, we do not consider that a particularly serious problem. We often encounter duplicated databases or incomplete downloads in experiments. After using our system for a few weeks, you find that the data is very persistent; it can feel as though you cannot delete it even when you want to.

00:40:40 · Discussion

The question means that the data stored in the browser has been cleared.

00:40:51 · Eileen Wagner

Ah, cleared, rather than duplicated. I understand. That is a problem. Users need to understand where their data is stored to ensure it is not lost locally. We currently keep a server copy, but that may not always be the case. Users might choose whether they want a server backup. Without any backup, I do not think IndexedDB is a very good long-term solution.

Q&A: syncing agents, the CLI and the browser

00:41:41 · Eileen Wagner

How is this connected to the agent? I did not understand how it suddenly went from the cloud to local.

00:41:59 · Eileen Wagner

Let me try to explain. You have a local CLI, and your agent can use it directly to read, write and modify data. For synchronization, the CLI agent session and the browser session currently connect through the cloud. We considered connecting them directly with WebRTC, but using our cloud was the quickest implementation for now.

00:42:47 · Eileen Wagner

We are trying to make the whole process work locally with a local-only space; that is possible too. Gavin can explain more if you like. I have not heard of egg.js. Is there anything else you would like to understand about the connection between the agent and the browser space? Let me ask Gavin.

00:43:37 · Eileen Wagner

Gavin? Hello, thank you. Can we say more about the connection between the agent session, the CLI sessions with the web runtime? I said as much as, you know, we thought about using WebRTC, but right now it's just going through the cloud after all. But we also have some plans to really think about a local-only build experience. Is that right?

00:44:04 · Gavin Owens

Yeah, it's we have a prototype internally of it. Actually I think it'd be a much better experience because the updates will happen instantly. I don't know if you mentioned we're also prototyping the native desktop experience.

00:44:20 · Gavin Owens

So all of the syncs, all of the sync behavior goes through dialog. So we're just, like, when you write locally, you just write to the local instance of Dialog, and then Dialog is configured to have a remote. That remote, by default, is the Cloudflare R2 that we're using, but you could in theory set it to be anything. You could set it to be something you set up yourself, like you could self-host. You can actually do that today. It's not documented, but you can do it. And, you could also imagine setting up like a peer-to-peer remote.

00:44:57 · Gavin Owens

And, yeah, so there's no reason why we couldn't have our runtime just read from a local database. We just have to set up kind of the loop, and that's one of the things that we're trying to do right now, but it's just, we wanted to start with the web so that we could test the shareability first and just kind of understand how people want to use it before we build out, like a more rich build experience and, like more full-featured native application.

Q&A: frameworks and self-hosting

00:45:33 · Eileen Wagner

Thank you, Gavin. That is similar to what I said, but more detailed. Another question: is egg.js suitable for local-first applications? Which Node.js framework is better?

00:45:52 · Eileen Wagner

I do not really know, and I am not sure Gavin does either. We now leave this kind of decision to the AI coding tools. They can choose whichever framework they want. We use React and Node; I do not know egg.js, sorry. The AI generally makes these decisions for the small programs, and so far we have not really had failures. The software ultimately gets packaged with Dialog, and changes go through Dialog too.

00:46:50 · Eileen Wagner

We are also writing components ourselves. If someone just needs a place to write, a small database or a table, we have already built some components that work well with Dialog. People can use them directly instead of everyone rewriting them from scratch.

00:47:16 · Eileen Wagner

As Gavin said, we first want to understand these new builders: what they want to create, how they want to share it and how they want to interact. That is why we prioritized the web. Fully local applications are possible later. From the start, we considered whether other people could host the system or run it peer to peer.

00:48:07 · Eileen Wagner

Another question asks whether Tonk could be an intermediary, using host and SRV to redirect the browser to a local instance. I cannot answer that directly. Do you mean using Tonk as a middle layer? Could you explain a little more?

00:48:56 · Eileen Wagner

As a middle layer, yes, that is how we think about it. Different coding tools can use the Tonk CLI to read and write the underlying Dialog data, and you can build anything on top. But the CLI is only a small part of it. I think the more interesting part is how you interact with end users and share things with them.

00:49:54 · Eileen Wagner

Thank you, Gavin. Right, there was already a P2P test yesterday.

Q&A: local services, remotes and replaceable frontends

00:50:17 · Eileen Wagner

Gavin, I think there's a question about whether we can use Tonk CLI as a middle-layer tool. After launching Tonk locally, could we point the domain name to a local Tonk service, and have that service sync with the cloud, maybe using host and SRV? Does that mean anything to you?

00:50:55 · Gavin Owens

No, I'm not, I'm a bit confused about that one. But I mean, like, we do local dev already, so you can just run, like, the Tonk GitHub repo and start a local instance, and it, and it basically hits, like, Dialog locally.

00:51:16 · Gavin Owens

One thing Irakli, our CTO, likes to do is point the remote. Again, a remote is another peer it wants to synchronize with. He sets up remotes for our staging or production instances, so it starts syncing into R2 instead of just running locally. He has the web runtime on localhost, but it syncs through R2 and uses some of our production auth code. An auth service sits in front of the R2 sync and checks the UCAN delegations.

00:52:01 · Gavin Owens

So, I think that that what we're trying to do is we're just, we have yet to make it really easy for you to have the web runtime and all of this stuff, like, sit locally. But I'm sure, I'm sure if you're a technical person and you have a little bit of help from an LLM to speed things up, you could probably get something working relatively quickly with the code that already exists. And Irakli, if you jump into our Discord, Irakli would be very happy to help you. That's his favorite thing to do, is to help people set up other remotes. So.

00:52:41 · Eileen Wagner

I can testify to that. There's also, are you aware of something called Clash, Gavin, where you have a web app and a configurable back end? Like, is that something?

00:52:53 · Discussion

—I'm not aware of it, so I can't really speak to it. —But this is the model being invoked here. —I've never heard of it either. I'm curious.

00:53:06 · Eileen Wagner

We haven't heard of Clash, but we'd really like to learn about it and how its model could be used.

00:53:20 · Audience

I have experience with Clash. It's software people in China use to get around the network firewall. Its core is a backend that handles the network, configuration and things like that. It doesn't provide a GUI by default, but it exposes APIs and endpoints you can use. You can build your own frontend and call those APIs.

00:54:04 · Discussion

Okay. I think that's exactly something we can do.

00:54:13 · Eileen Wagner

Dialog itself comes with views, but you can actually make any kind of frontend, however you like. It's effectively a backend because it's the database. Our idea is also that everyone can use their own frontend with the same backend, and change it at any time. Those are all possible scenarios.

00:54:49 · Eileen Wagner

At the moment, I don't think Tonk Network is blocked, so you can try it. I don't know how well it works where you are.

00:55:00 · Eileen Wagner

You're thinking about switching backends. What I mean is that your data, frontend and authentication are all in the same place, so you can send it directly to someone and pass it on again. We call it portable software: software you can carry to different places and use. There isn't such a clear frontend/backend boundary, because the database carries its own views.

00:55:48 · Eileen Wagner

'Without configuration, use SaaS; with configuration, use CRDT.' Someone is also asking for the product link. Yes, this is our homepage.

Q&A: building useful software

00:56:12 · Eileen Wagner

There was another question: 'How can AI package it elegantly?' That's a fundamental question. Many people enjoy writing all kinds of programs, but few people enjoy receiving all kinds of programs, because some of the things we write really are a bit too crazy.

00:56:38 · Eileen Wagner

We're learning how to make higher-quality software, and we want to build things that are actually useful.

00:57:02 · Eileen Wagner

As a designer, my first question is: who are you writing this software for? Who are your users? In what situations do they need it? That's the most important question. Otherwise, you may end up with things no one needs.