Subject: alice's compacted fedi memoir
Status: simmering in editing hell
Creation: 2024-10
Last change: 2024-12-01

> starting maastodon is potentially the worst decision I've ever made
[~ @CobaltVelvet of the Octodon, 2018-05-02 03:04:42 UTC]

content:

this started with one post and is now at ten thousands words. lol.
i still have a lot on my heart it seems.

Chapter 1 : where i am coming from. a little mastodon story

i joined the fediverse in 2017. that's relatively early. many waves sinces.
made an instance, instance was one of the coolest spots around, mildly famous even, etc. but that's not the main story and i am tired of telling it. maybe later.

i was good at 'hosting things on the internet' so, i became at one point and for a while 'the mastodon hoster', in that quite a lot of popular instances were my responsibility to keep online; including the precious mastodon.social, that i hosted for a very low fee.
dont get me wrong, i was passionate about the good work, but also dont get me wrong, i was desperate for any income and eugen used that and i burnt up and he threw me out the minute i was not useful anymore, as it turned out to happen to quite a few people.

one of the most obvious pain points i was aware of in mastodon, and ultimately common in the fediverse entirely, is that the whole point of the thing is empowering individuals and small communities but it stopped short: communities|instances were strongly tied to hosting and means. to start a strong community, you'd need a team of moderators and administrators, and a technical team, and some funding to even start. hosting your own instance costs between 1 and 30$/month, that's often too much. one is pressured to at least have a dozen people per instance, and most likely to join a larger one, all of which imply a Lot of things. and we still don't have smooth account portability at all. it's frustrating and many are unsatisfied, and it keeps happening.
this is a failure and an important one that went fully ignored, or quietly accepted, and in my opinion the number one reason why the fedi is so biased towards the most insufferable tech dorks in the room: it's correlated to who is allowed space. and too many love it that way and are comfortable in this situation.

now let me tell you something that is very important to me specifically, because it was my work for years: already in 2020, we could have made it as easy as discord to start your own community|instance with no required tech skills or servers. it was always possible. and with economy of scale it would quickly become feasible to host instances for very cheap and even free. if mastodon was built differently we could have optimized out a whole order of magnitude of resource usage as well but that's mostly wishful thinking.

it's not just a silly little theory, it's what i went and built. it exists, right here and now. it's been over-automated and over-engineered twice over, it's fast and reliable and affordable and it even hosts people for free already. it's more successful than a lot of corporations and more ethical than a lot of non-profits. [https://fedi.monster/]
and it's also stuck closed at the moment because people aren't exactly queueing to do the work that i am tired of doing; and when you do something actually for public benefit, investors don't crawl all over you so you can go on a hiring spree, so i beg people to join and promise some money but they need real money and have no time for small things; most of everyone i know struggles for rent, how could i blame them, even i have had to prioritize other work to survive the whole

time.

and its so hard to let go! bc people really want this. and sure, i'd love to help them all get hosted comfortably, but as soon as they find people to take over and lead the way this time around. i said i am done and i am so fucking done for at least a decade now. i relentlessly gave them tools and knowledge and fixes and so so much time and care and maintenance. eight years of this is enough.
it's time for people to share the workload instead of waiting for some of the most broken people in the room to sacrifice themselves again and again. why dont you throw some emotional manipulation on top? (asshole) where are all the wealthy nerds with spare time now? (enjoying a comfier life and job than me) i only have left a kindly, heartfelt, compassionate, démerdez-vous. or i'll never get out.

it's the third time i try to let go, actually. 2018, 2023, probably once in the middle too. it gets all so blurry and i have to dig into old notes and posts. 'why didnt you just stop' i needed money and had nothing else. i needed to feel useful. i wanted to help. i couldn't take disappointing anyone. whatever.
some people did help and are still helping, and i love them for it. this time i even assembled a Team to take over things, and they're very good, which makes me think it's finally Time. but there's still a long way to go for everyone, and they will go at their own pace, not constantly pushed by the pressure like i was, i hope.
it's an exhausting hard process and it takes so much time and so much work. surviving small scale under capitalism surrounded by ppl who mostly don't give a shit is hard! who'd have thunk. and yet we persevere and refuse to betray ourselves.

now. can you imagine what we would have accomplished if i hadn't had to build this all alone for so long? if we had gotten one million dollars? or more volunteer co-conspirators than needed? or even simply the mastodon project's support, whether moral or financial or logistical, for a collective goal that would have helped everyone? i guess not.
when joining the free network, you might expect some actual community and collaboration, and it hits hard when you only get met with more self-centered individualist assholes.

did i tell you i hated hosting, the entire time? like, datacenters are mostly a failure case. it's the mainframes all over again, all due to decades of corporate control and constraints. corporations lead and fund tool development, and available tools shape developers' imagination.
a pencil-drawn construction line in wait for a fairer internet turning out etched in stone and kept forever due its unfortunate economic consequences.

and what tools these are anyway. i hate ruby on rails elasticsearch redis hashicorp docker kubernetes and google, i hate all the cloud shit stacking dodgy abstractions into orbit to pretend horizontal scaling was invented yesterday, i hate the hosting companies we can never trust, and i hate hate hate all those evil fucking tech corporations and their interests.
call me a devops again and i will burst into flames; i have been a system caretaker and a programmer at times, the ∞-stack freelance jack-off-all-trades of the radioactive-dumpster-crater you call a cyber age, pre-formatted from childhood and always there clean up after the big-money-professionals' filthy messes. a modern character for a modern tragedy. anyway. that's out of my system.

every day i get reminded of yet another way that mastodon has Failed, not an abstract promise to a vague customer or potential user base; but failed us, those who believed in it and committed to it because we knew its true potential.
yet another way in which eugen has ignored our needs and concerns for so many years. yet another person who gave up or was given up on or was not given the tools they needed to protect themselves. yet another important chunk of work rotting in mastodon's github for years in plain sight waiting for The CEO to deign look at it. yet another contributor who foolishly believed it would be a 'community project'. yet another time queerness was used as a selling point and then its people ignored. what a waste. and for what? branding, ego, market shares, the feeling of being on equal footing with corporate competitors neck deep in their own horrifying lack of ethics?
will i even be remembered as a footnote in the tiny shitty crumbling empire he built? do i even want to? will any of us whose back was broken to uplift him and so many others? who kept trying to nudge him in the right direction mistake after mistake before being erased and forgotten? probably not.
this is the world we built and live in. i dont want thanks and praise and to be told how strong i am; i want it all to mean something. for once in my miserable life, let our toil Mean Something

i had way enough of all that and need to move on, i am just too disabled depressed and burnt out to spend any of the short time i have left in this world fighting people who don't want to listen to me. it all feels like hitting my head against a wall and it's already 100% bruise by volume. good luck with all the temporarily embarrassed millionaires, i'll be sharpening the guillotine.

i am on a tiny GoToSocial instance lately, 'present day fediverse' to me. it does a few things that i need and mastodon is gone too far to care about, like:
- 'scale down' and optimize for smaller, simpler, cheaper, and low-maintenance servers
- additional privacy, control, options, and customization for users

- experimental features like interaction controls (someone has to flesh it out before mastodon can decide what it wants).
its just a bit of my personal website that friends can subscribe to for the secret posts and slice of life. it's a start. not the destination

the point of a decentralized protocol is how much you can actually play with it. as an entity with your own needs and communities. the rest is bullshit and capitalist shitlords trying to split a pie they havent baked yet.

' only emperors will win at the empire building contest '

| (?) if you love the w3c so much why is this a crummy text file
|
| this is the future of blogging. i am joking. try it in firefox at least.
| i dont Want to format it like this, but Someone Someday made it so that one could
| use the Link HTTP header to add a stylesheet, which is a trick i Love and use here, but
| not a javascript document, which would have let me embed a markdown parser, for example.
| i have been frustrated about this for about a decade and i will make everyone suffer
| for it now.
| please let me distribute a plain text file and HTTP Link it to a renderer script

Chapter 1 Act 2: if you really must know about my time as an admin,
==========

/
  'i really do not want to do this for the rest of my life'
\

i have already said all that, but mostly in private, and ultimately it's a big chunk of juicy Context. have at it.
of course i was tired of mastodon and the project's leadership and didn't want to bear that weight every day or any day, but that's just one part.

i didnt trust or wished for anyone to replace me; this place was so tightly bound to me for the best and the worst.
a lot of things move in 8 years. there's plenty of time to reconsider your needs and priorities.
to realize there may be nothing left for you at that place you keep going to.

at the end, i didnt like the instance admin role. being everyone's mom and everyone's court appointed therapist and everyone's lawyer. ppl worship you, ppl hate you, a few manage to balance both. whatevs.

i didnt like the instance social dynamics. there's a running joke that it's all just cyber feudalism, and that's so absolutely true. but you rly have to consider all the terrible little lords dying on terrible little hills. the social groups picking sides and the local wannabe-influencers. it's painful to watch and be a part of.

i ended up disliking the 'online microcelebrity' thing as well. its fun its fun and then i just wanted a few dozen of my closest friends to talk to with some privacy and the occasional boost. having a hundred strangers in my mentions never made me happy. i take my fame for a moment at a time.

personal data is radioactive, and there are terabytes of the stuff. people dont appreciate it because it was all commodified by surveillance capitalism, but i take this very seriously.
over time, most of it, no one will delete, no one will use, it's just sticking here as a forgotten liability, neither public nor secret. every day there are more aggressive data harvesters and new security vulnerabilities being found and exploited. if it's on the internet, it will get poked and abused, and needs constant care.
all server operators should consider clearing the whole board every now and then, and demand to hold less sensitive information.

i met many good people, even a few i want to keep in my life, many of them helped me back in many ways or were just good company that i was glad to fight by their side.
but a few of them just considered me a tool in the background; someone who's defined by providing or owing them a service, who's easy to blame and easy to ignore. some didnt see me as a peer, volunteer, friend, coworker, or person with my own marginalizations; sometimes i even doubt that i was granted interiority at all, maybe the equivalent of a 'brand' at best.

this is something i see very frequently, actually. people sometimes would rather believe that the tool is not built and maintained by their peers.
they want the Silicon Valley Promise of the tool appearing from a cloud invented by two millionaires and maintained by no one in particular.
it terrifies me because it's a lie; the tool always ends up built and maintained by an exploited bunch of people that you might very well meet eventually.
and it terrifies me because people can be so attached to the passive consumer role. we cannot do this without their help, and yet they want us to do all the work and find all the money and glue it all together while they watch and stack constraints. 'are we there yet? (kicks your seat)'

a huge mistake in hindsight was the open registrations. never do that.
either you don't want ten thousands users, or you have a team of experienced and organized moderators on standby, or you don't really care and are going to be an irresponsible shit admin.

anecdote time: it reminds me of when some ethereum guy tried to get back to a mostly-empty account he signed up while registrations were initially open. i didnt like him and had closed registrations for a while since, so i quickly banned him. its my instance, its already a bit of an overflowing mess, move off 'or make a sizeable donation, you capitalist' (something like that).
he then complained about it so hard on twitter that elon fucking musk saw it. can you believe that shit! my mundane action triggered a reaction from 'lord biggest douchebag in da world' in person. absolutely surreal moment.
we had plenty of fun too. but i will agree that the instance premise is painful for everyone, at least.

anyway.
i quit because i was tired of the job.
my wish was for the Octodon to Explode, for everyone to disseminate and go build their own communities or join their friend's smaller servers.
i know this bunch, a third have the skills to host a small instance, a third has money that ought to cover it, some have community organization or moderation experience, etc. 'you all had eight years to avoid this question, now im sorry but figure it out. there are more beaten paths than ever before. it's your turn on the mastodon or the gotosocial or anything else.'

i was never going to hold the world forever. do i look like fuckin Atlas

sometimes i wonder if it would have been different if i started out saying, 'this instance will live for exactly eight years and blow up'.
but that would have taken actual long-term planning, and at that time i didn't even plan to live this long at all.
that's right, your most likely end was your admin off'ing herself or getting killed by a fascist state and no one paying the hosting bills anymore. Surprise!!
many of those closely involved knew that already, and trusted me anyway and wanted to be at my side throughout, whatever happens, and i thank them all again for this. you were a good lot and i'm glad to have met you.

be glad about those eight years. i feel fine about it. it's an impressive life for a website and a poetic end.
nothing lives forever, even your favorite corporate empire monolith. death is part of life

| (?) the matrix anecdote
|
| i had a very similar (but shorter) time on matrix actually, except the moderation tools
| were even worse. there were basically none and i had to script it all with bash and curl to
| the API. no domain blocks at all! come on. it was a horrible experience. horrible software.
| horrible memories. and the protocol isnt even novel or interesting, why was i ever hopeful?
|
| all of this really left me disgusted by the federated model. but once again a big
| aggravating factor was the open registrations. it only highlighted to me how silly
| it is when some think the fediverse 'just' needs many more big open instances.
| the web of trust is the most important part, you can't skip it that easily, or you become the
| yahoo dot com; too big to really notice or care about abuse.

Chapter 2 : my wacky ideas on protocol design

==========

> okay but How Would You Do It, then?

and its a whole thing with its own prelude, because after thinking about this question for so long, there is quite a lot of shit to unpack.

this is all technical bullshit - that's what i know. you need to interpret all this with creativity.
we are covered in mud and stacking bricks, and i am telling you that you need to imagine the fully-furnished house. my skill is in laying the foundation and i am not going to waste time pretending i'm good at something else. we deserve a serious foundation to build things on it, and only then can we get to the living room.

this is not a perfect final treatise of the subject, just a draft of however i would make things. it might have flaws, it might miss or forget. that's okay.
it's merely a collection of ideas that might solve a few problems in interesting ways and i think are usually underrated, unknown, or purposely ignored because they aren't exploitable in the classic ways.

lets throw out a few ideas out never to see them again because they do not interest me:
- the global audience promise : i will focus on building connections between individuals and safely sharing along those connections. an organic graph of people with mutual trust, not the emperor's public relations account or the marketplace of harassment.
- information permanence : social media is more of a private chat app than a scientific paper archive; information should have arbitrarily small scopes and posters the need to finely control how their posts and profiles are used and diffused. if you want a blog replicated on a blockchain then go do that without me.
- the client/server paradigm and mainframes : after spending two thirds of my existence managing servers i can confidently say that it feels like banging rocks together ad nauseam and it is a whole Problem. we all deserve so much better than what the silicon valley bros are forever stuck on

without servers, without a platform, without a corpo, where do people meet and who moderates and at what scopes? compared to the current fediverse, we actually got rid of the coaxed community to be moderated altogether.
we need to remodel social groups, explicitly and with intention, as structures of our network that isnt bound to a server or domain name: with their own members and roles, with their own timeline-s, with names or hashtags they're associated to, with their own policies and moderation and decision making processes. social groups need to be free.
and this more generalized tool could be used to recreate a mastodon-style local timeline, or organized fediverse groups, or be associated to twitter-style hashtags, or cover the twitter lists without skipping consent, and reach all the way to the possibilities of a facebook-style group, all with the same building blocks. all with so many scales and options of moderation tools and group decision-making. and still not constrained to a physical server, to an administrator, to a hosting provider. am i making you dream at least a tiny bit?

i will split the question into three sections:
- we have to solve the Identity Issue (how to prove who you are to the network?).
- its subplot, the Naming Conundrum (everyone wants the same usernames, and a flat namespace is an impasse).
- then the Location Issue (where are posts and profiles? where is your inbox and devices?).
- and the Asynchronicity Issue (i want p2p And to be allowed to close the app).

i will skip the encoding issues, eg how we actually represent all this, how we encode activities, what bytes nodes send eachother, because frankly it's implementation details and always gets figured out as things are built.
i would reuse ActivityStreams, and maybe even ActivityPub itself, because it's simply such an established and extremely wide set of standards that i have no reason to reinvent. but it also does not help much for our purpose. i trust you all to bikeshed these points until far after my death.

i will try to describe it as if it was built in a Cheap, Reasonable, Compact, way, because i think like someone who would never be given any funding by a millionaire+. of course there is a bigger way to build every single thing here if you have unlimited means, but you asked me.

and i also don't mind it being slightly slower than loading twitter dot com at its best, if we handle harder cases well enough. we are building the future that no one will bother using, you can have a little theoretical patience about it.
and this will require a bit more computing power for peers than a plain HTTP request; but if we take servers in the equation it will be pretty efficient and 'low tech' overall. personally i could take more progress bars and slower computers in my life, instead of displacing the complexity to a giant cloud that poisons the other side of the planet.
and also, 'how does it handle its first account with a billion followers?' i dont give a shit, no one should have this. they can have their own website with press releases or whatever. it will probably work, but this is about us and a humble tool.

we are building this in a way that considers globality and imagines a universal protocol;

but always prioritizing the locality and plurality of organic interactions.
while i try to design for everyone, reaching everyone is not part of it, and this imaginary network might seem eerily quiet when compared to twitter dot com.
that's on purpose. this is to social media what amateur pirate mesh radio is to music broadcasting.

also keep in mind that the fediverse, or matrix/email, and other federated and more-or-less decentralized networks have glued together some of those sections.
e g, a mastodon server is both an identity provider and a repository of user profiles, and also acts as a suffix to turn local names into global names; and joining the three has important user implications for the better and the worse.
it often makes sense to abstract the composition away; but as with all abstractions, i appreciate when they know to let you explore for yourself, what they abstract over. sometimes that's what you want. that's what we want here, a disassembled view.

the identity issue : presence in an unknowable set

if we don't want to use DNS and import its issues, we will need some kind of a universal registry somewhere that lets us look up any profile and authenticate, and find out what machine-s to communicate with. this flattening lets us clear the list of constraints and establish new ones. (full portability without requiring a domain name, for one)

let's talk about my favorite animals, public key cryptography (that i will stack like lego bricks) and the distributed hash map (or DHT), that attempts to fairly and efficiently spread a large database over a decentralized network, without the overwhelming issues of a structure replicated in full on every node (which is not useful for this purpose anyway).
i want to put all the profiles in there, like magnet links in bittorrent, with the least amount of public data we can get away with.

each row could look like this:
- a unique public key identifying your profile. you'd hold its associated private key somewhere.
- a location, as a small list of nodes that you are present on and will relay messages for you. this can be a server, your computer, a trusted third party, or any other node-s at random depending on things we'll discuss later.
- a signature, ensuring that you - or at least your key - authored this entry.
- a last update timestamp, for possible emphemerality.
- a more detailed profile, in a few kilobytes, with your main metadata. (everything required to authenticate you and update this row)
  this could be optionally encrypted with an additional key, shared to your contacts but not the whole transport network, for extra stealthy profiles.

it would always be replicated on multiple nodes (per row), and each row would function as an individual state machine, mutable with the key and following arbitrary constraints pre-established by the row. the hardest part in my opinion will lie in defining this state machine, and the ways we can make it cover all use cases and threat models.
a key point is that the identifying keypair is not so important, more of a signed UUID, and if lost or cracked should not be enough to take over an account, although it would be an annoyance and require switching to a new key. (the previous would be become an alias or redirect)

each record could specify a set of modular multi-factor authentication methods, one or any combination, to pair a new device, migrate keys, change locations, etc:
- your previously logged-in devices with their own keys already tied to your profile;
- a paper backup key or password, or hardware authenticator;
- a third-party OAuth or OpenID provider;
- any existing service that can provide signatures, such as an email address (SMTP with DKIM);
- any existing service that can provide a publicly accessible value (HTTPS page, DNS record)
etc. i dont Recommend logging into the decentralized network with GMail but im sure someone somewhere will be glad it's possible.

visually, one could imagine this system as: we are each picking a random point in a nearly infinite ring.
there, we place a tiny virtual machine that will point towards ourselves, holding our locations and protocols and arbitrary requirements for changing this record later. it can change, always determined by its previous state and rules; any node can run it and any other node can verify it.

the naming conundrum

there's this diagram going around, Zooko's triangle, one angle is 'humanly meaningful', one is 'secure', and the last is 'decentralized'. the point is that you can't have the three. you can have any two, but then you're stuck.
dont wikipedia it, the page stinks so bad it gave me a terrible gut reaction. the whole Solutions section is about blockchains and i want to stab someone. i keep saying your answers to such problems has heavy political implications, and yet again the page only mentions blockchains and market-inspired systems. it is Sad, Depressing, and not a coincidence at all, that an entire industry's imagination was brutally truncated by libertarian capitalists.

my favorite answer so far is : petnames.

we admit that humanly-meaningful naming is inherently not unique. 'no two people have the same name' would be a lie in most societies. government names, or even hyper-personalized usernames, have never been globally unique, and collide in many places.

if you want uniqueness, give people numbers, but you can't and shouldn't make them use them directly.
by forcing ourselves in the goal of building a unique human-meaningful naming system, we only build harsh conditions that will harm ourselves, and it is exactly what we can see with markets having rich people hoarde and trade usernames. i will not make my identity an entry on my portfolio 's nft section. over my dead body and the billions i will be in

notice that names are rarely the same between everyone you know. many people, individually, have more than one name, names changing in time, and names different between their correspondents. you call this person so, but someone else will call them something else. why would this not be an inherent and accepted part of the way we model social networks? computers are not cold and rigid; we made them that way, modelled after our governments and corporations, with hints of latent fascism.

our initial conundrum was a conflict between high-entropy identifiers [impractical but our only way towards global uniquess],
and low-entropy identifiers [practical and intuitive but only meaningful at the smaller local scale].
we need both. there is no solution that will do both at the same time.

to cover both security and decentralization, we might start with a randomly generated cryptographic keypair. it has nothing special to it, and can be changed later, provided that there is some way to follow continuity. (e g you publish a message saying that you are switching to a new key, and the previous one should not be used)
to cover the intuitive meaningful names, we might bring back contact names, which no one had time to forget anyway, and grant them as much fluidity as possible.

as far as names and identifiers go, you would have :

- the public key will be used by people to identify and reach you in a universally-scaled network;
  and you might have multiple for different identities, and changing it would be a technicality for
  the software to fully handle for you.

- a friendly name, provided by either side of any relationship, which has no need to be
  universally unique, or secure, and can be changed at any frequency.
  your most used name should be unconstrained by any kind of technical bullshit.

- optionally, a secret component, valid for a limited time and place. getting closer to capabilities,
  (or like a classic invite link), this lets you control who can practically reach you
  by always having a way to invalidate a harmful path from your side. once their link
  is locked, they would have 'lost your @ ' for most intents and purposes.

'that's a lot!' indeed, but we have already fully normalized linking to profiles by QR codes, or addresses so long and convoluted that they need to be copy/pasted. and meanwhile have found so many ways to share high-entropy identifiers - encoded as word lists, emoji, pictures, etc. but all secure schemes will ultimately require you to exchange or compare some kind of high-entropy value. (even signal, for example, will ask you to compare 'safety numbers' to be sure)

another important point about composability, is that whatever we've come up to so far, you can always point a regular old DNS name to it. we can easily imagine a TXT record on any domain or subdomain, pointing to your public key. and by this association, looking up any profile from a domain name.
imagine this: you add a DNS field, and xxx://yourdomain.example.org/s3cr3t becomes a secure one-time-link to reach you.
still not a server in sight beyond DNS.
this should quench the 'but how am i going to build a giant platform and have a lasting name' - the DNS pyramid won't save us, in fact it's already fully owned by fascists and renting you unique names at a profit, but you can still use it if you must;

but we have to build for more than this.

for comparison, both the current fedi implementations and bluesky currently use the DNS pyramid as a sole load-bearing component in looking up identifiers. it's more decentralized than twitter dot com, to be sure, but still entirely hinges on people being able to rent domain names and keep them long-term. which is not something i am comfortable with.

just out of completeness, my issues with DNS for this purpose are that domains are (at best) leased to you for a few years, by a shady investment company (most registries) through even scummier companies (most registrars), and the whole thing is managed by USA-centered organizations (ICANN) and a handful of DNS-related businesses (VeriSign & co). free options exist, but always unambiguously leave you in a precarious mess, yet companies can merrily go buy a million domains just to park them and speculate on their value. and finally, one has to ask why it is so easy for stormfr.nt to keep a domain name and so hard for sci-hub and the pirate bay.
i love DNS but it's not a good identifier. i will not be asking anyone to get their own domain name so they can exist on a communication tool. we all have too many landlords in our lives already.

my real proposal, is to destroy the myth of the globally unique name.
burn fame and notoriety. welcome a fluid identity.
where we're going, we don't need a name, and we have no brand to build.

the location issue

so far, i put the public part of identity into a large public structure, but we also have a lot of private data on our hands, and there might not be a unique good answer to where to store it.

i see basically four levels:
- your data is open In The Public Cloud, and all you need is a big passphrase. it's practical but vulnerable over time as it can eventually be cracked
- your data is hidden In The Public Cloud, protected by a passphrase and a list of locations. less vulnerable, but now you need a backup file or printed QR code.
- your data is entrusted to one or multiple arbitrary locations, like the fediverse or any centralized service does. a lot like the previous item, but the locations are meaningfully picked instead of randomized.
- your data is stored exclusively on your devices. less redundancy and paths for recovery, but the most private option.
i think we should consider and prepare for them all, as they have their use cases.

another aspect of location is which transport protocols to support, and again i think wide range is the way to go.
whether using a system of peer-to-peer relays, or when hosting profiles directly on a server, we can benefit a lot from using an overlay network like tor, and linking to a .onion address whenever a location is required:
- home hosting is a lot easier when you can't be surveilled by your ISP and when you don't have to bother with dynamic IP addresses or port forwarding.
- home hosting is a lot less scary when it doesn't involve sharing your home IP address too obviously.
- it also avoids us the hard problem of securing the transport layer between nodes and authenticating addresses. classic TLS certainly won't help us this time.
we should prepare to support many of such transports, and change recommendations over time. one could imagine an early version supporting any of [direct IPv4/6, HTTPS, tor, i2p] and we would be set for a long while.

for our purposes, tor would just give all our nodes a secure and reliable .onion address that we can include in our minimalist profiles, and not bother with the actual networking or encryption. it's pretty neat, and should be considered a basic building block of modern peer-to-peer architectures.

of course now that's secure and decentralized, but when you shut down your computer or lose connectivity on your phone, cannot get incoming messages.
i would not be satisfied by leaving you with 'simply point the decentralized registry at the server you rent or computer you leave running at home'.
this might be best for some use cases, but we're still missing the most important component.

the asynchronicity issue

people are doomed to not be online at the same time. connectivity comes and goes. servers are shut down. sometimes we lose any of those unexpectedly.

out of this, we have come to rely so much on intermediates, the mythical always-available always-secure Server component. and soon enough we're displacing computing resources and heat and power consumption and power dynamics and supporting a coup to dig fuel.

i want a network of asynchronous relays generalizing message passing. or the minimal building block we can replace our current 'servers' with, in a way that is fully decentralized, composable, and fluid, reducing the roles of those intermediates as far as we can.

an anonymous and undistinguished machine on the network among so many, holding small messages for everyone as a public service, that other nodes can retrieve at a later point. like a swarm of blended hosting providers where the content is opaque and randomly distributed among multiple redundant nodes. nodes would be picked depending on their uptime and available storage, and all could pick how much space they're comfortable sharing to the common pool.

a profile would have three or five 'habitual' relays where they keep their main inbox/outbox, and other peers could fetch or post messages into each.
one peer places data on the relay network and shares which nodes hold it, and another fetches it by querying the relays.
and relays can always be skipped entirely if you can establish a direct connection instead, or replaced by another relay quickly.

- the message is stored temporarily, and will be automatically forgotten by the relay
  after a set time, after a number of reads, after a set inactivity time, or when it lacks resources.
- the message is fully encrypted, and cannot be read or interpreted by the relay
  beyond its (signed) enveloppe: destination or box identifier, time to keep, and priority.
  and additional metadata managed by the relay, like the time it was submitted, or who it is held for if that's relevant.
- the message is always stored on multiple relays, with a variable multiplicity factor.
  (highly important messages, or cases expecting a high traffic, could increase it)

'wont that be full of spam in a day and a half' i am hoping that by both putting an emphasis on emphemerality and harshly limiting the opportunities for spam to actually reach anyone, it wouldn't be too bad.

something that has inspired me lately, is NNCP and their 'refreshingly ancient' use of asynchronous communication.
https://www.complete.org/nncp/
in a world where connectivity metrics get more granular in every dimension by the year,
asynchronicity and the act of holding data are bound to be more and more relevant.
but it is also generally one of the harder problems of designing p2p protocols, most examples requiring a synchronous connection.

call it p2p2p or dweb 7 (please dont)

a practical example : if we wanted to add portable identities to activitypub

there really arent that many limitations to ActivityPub itself; and so i think most of this is fully doable as an extension to AP.
what is always more complicated, is getting the fediverse entirely to cooperate. but

1. someone should build the distributed index, and host one 'flagship' node on a
   well-known domain as a public query service.
   other fedi servers and end users could run such nodes and have the same identifier lookup service.
   it would expose:
   - a HTTP redirection of arbitrary paths to the destination instance association with a profile;
   - a webfinger resolver that redirects to the destination instance;
   - an API for profiles (or their instances) to update their location, signed with their actor key.
   'how exactly' a lot of work, indeed, i can't just plop down a github about it in a week end,
   but it's been done before. hire me about it i guess, but i want a good situation this time.
   i am holding the world hostage if no one else will do it. hire someone better and in working order

2. fedi implementations could start issuing identities on this namespace; eg
   https://public-relay.example.org/id/{public-key-fingerprint}
   https://public-relay.example.org/id/{public-key-fingerprint}/arbitrary/post-id-123
   and send it periodic updates containing the instance URL, signed with the actor key.
   this is also the opportune moment to bring in petnames.

3. other implementations would be able to query this URL and get a redirect to the home instance;
   while webfinger would resolve @fingerprint@public-relay.example.org

4. to avoid building a central dependency on public-relay.example.org, compatible fedi services
   would run a local node that takes priority over public-relay.example.org.
   identifiers on this specific domain would be considered a special case to be resolved out of DNS.
   public-relay.example.org would stay as a reference and example.
   we can also hope for parallel implementations at this point, to avoid a new central dependency.

5. we have:
   - fediverse where you can move instance and transfer your posts seamlessly between servers
   - not dependent on the domain name pyramid 😎not paying rent to united tld holdco anymor 😎
     ('united tld holdco' is a real company owning like 30 TLD's. not even joking you)
   - you don't even need for the previous server to cooperate much, except by not misusing
     your previous private key or refusing to let you use or export it.
     ideally they would not have it at all ofc, but that's for future steps.
   - everyone can then transform identifiers to the DID spec if they find it relevant, or just a
     bit shorter. it's a detail.

6. progressively bring tor and the asynchronous-cypher-cloud i guess.
   that step might as well take a hundred years but so far
   - everyone still has your end IP address. it discourages ppl from hosting at home for
     many reasons (privacy, dynamic addresses, evil isp's, etc) and hinders portability.
     ideally we'd support each profile being available on various transports.
   - we're still using servers that we have to individually maintain. its a nerd sport.
     i would like ppl to be able to start their community as easily as they start any app
     and everything else described in this document. (encryption, async relays, etc)

of course it's a Huge work and i dont think we can ever convince eugen or many people at all of this but like.
it's Possible. it's possible without compromising to centralization, without a blockchain, without vc funding,
and even without starting over, by bringing small incremental changes and waiting for everyone to catch up.
this same vague 'fediverse' could do so much.

my only point here is that what we have built is already extensible far beyond the crowd's imagination.
starting over is a strategically motivated move; and so is saying change is impossible and locking down into flawed designs.

Chapter 3 : hitting reality
==========

my main worries are harassment and the unavoidable, inescapable, surveillance capitalism.

i go out and stare at a surveillance camera.
i go on the web and get permanently recorded in a hundred log files.
i send a message to a friend and get archived by three corporations and seven states and thirteen private intelligence agencies.
i try to get medical help and have to worry about how many electronic devices are in the room because i know exactly how they work.
i watch porn and get my ghost profile for 300 advertising companies live updated with a new kink. facebook hears about it and throws it at its new heuristics model inspired by psychoanalysis, which misses my deal entirely, although it would have been obvious to a regular pervert.
i post a joke and a selfie and get permanently archived, analyzed by 70 language-heuristics-models, added to the training set for an image-generation model, seven overzealous data-scientists with no respect for consent are trying to figure out my exact 'mood' and 'gender', and i end up in three public datasets with a statistical guess of my genealogy, genitalia situation, age, and thought process. my left nipple becomes an accidental staple of image generation for the next decade but no one believes me, while the legend says my face will haunt you if you type the right cursed prompt. finally, my selfie arrives to a message board for heavily masturbating incels trying to guess my least favorite slur (no luck, i love them all). that thread is also permanently archived to more than a thousand backups around the globe.
i throw my phone into a volcano and jump after it. dear google, enjoy my last gps coordinates. i will finally know peace

okay. Okay. Why Dont They Build Your Good Ideas, then?

buddy they simply dont want any of this.
sure some of it is 'new' and 'risky' and i am such a visionary for thinking it and applying it like that. calm down before my head falls off. but none of this is really New. i learned about protocol building and cryptography like anyone else, by taking things apart until they made sense while reading wikipedia pages and books and manuals and technical references until my eyes caught on fire. this is all public knowledge and methods already in use.
i tried to tell eugen. i tried to tell thousands of people. i just end up looking more unhinged and disposable. they either don't understand or dont want it. usually they know what they want and it isnt this. big instance admins bet on an investment and user base. eugen wants to micromanage his project. everyone has business plans and funding sources and there are big sunk costs involved locking things down. innovation dies off so quickly.

bsky certainly doesnt want that. activitypub was live and already getting old and they decided not to extend it but to go a very different direction, all public data and keeping key bits of centralized control. (it stinks of cryptocurrency investors who also build scams around a promise of decentralization. oh look bsky got funding from cryptobros as i was editing this post. lol. sigh.)
in a way bluesky could be merged with activitypub very complementarily - and would stand for a lot of the features i consider a harassment vector and demand to opt-out of. the fedi's current interpretation of a Public post is already too much for me, to be honest.
the existence of a firehose is enough for me to flee, too; a built-in dragnet for unlimited surveillance by third parties, it was evil of twitter and it's still evil today. we should be very scared of a tool mainly designed around data-mining and social-mapping. this is the kind of design decision a system gets used to and never move away from, imo, the Public Repository Of User Data inevitably becomes the main product. (and we know how much demand there is for exploitable training data lately)
when i see that atm they let anyone put you on a public list without having you confirm it, i keep having to ask, why never consider consent first? (and here we go, our own local CEO considers the same, youve got to keep up with the supposed-competition's ethical violations)

-> more in-depth about bsky:
  https://dustycloud.org/blog/how-decentralized-is-bluesky/
  i mostly agree but i'm meaner about every point. i'm mean to
  corporations. genuine communist free-software bitch dork here

-> sorry i cannot make a list of 'orgs and projects going my way' because it takes a huge
  effort to actually understand how something works. not just read the big claims and
  papers - to dig into it until you know what it can and cannot do, understand the why's
  and how's of every decision, the full picture from the broadest strokes to the thinnest
  details, etc. a few vague references is all you get until i am paid full time to do
  this or stumble upon a unicorn who has all the good ideas and keeps telling me about it.

paranoia can often hint at what's coming to get you, and it's been coming for me for a long time. i know this threat and how many it has tortured and killed before me. i only give in to call it 'paranoia' to calm myself - technofascism haunts me every day.
take care friends because it's coming for you too, and i am already hidden deep in my weird little maze. i do often miss you.

of course the fediverse also has similar issues, like every web scraper can browse your instance and slurp a lot of things. and people kept trying to build global views again and again, and it was an exhausting fight to get them all to follow an opt-in process. but by normalizing opt-in behaviors and in-depth privacy features, the state of things as of now is not so bad. (even if it's not all E2EE, but one can dream. or build over)

i mostly post in private so it goes directly to my friends' instances. at worst a snoopy instance admin (that at least one of us trusts) gets to see it. sometimes i post unlisted and they can share it with their friends and a few more hops on a talkative day. i assume those posts will be used against me in a court of law but that's a habit.
maybe one of those days i will post in public again. maybe i'll never want to.

the last bit

the fediverse failed to push its vision to its conclusion yet. mastodon's narrow angle, and the rigid 'instance' and everything it implies, limiting our possibilities and defining our costs and keeping our gates.
but the fediverse is not an accomplished fact of life, and we are not done making with it. it is a dynamic that started a long time ago, outlives all its contributors, and will keep changing as long as we try.
this is the meaning of an open protocol. it's still true after my 8 years, and there is so much more to build.

our work, as builders of decentralized social tools, is to emancipate people from Private Platforms and Servers and the Domain Name System and all capitalists - to resist the centralization of power.

designing a decentralized protocol is easy - you simply have to be a communist. i am joking you can come to the same conclusions by being an anarchist. haha.
trust but verify. never put much responsibility on a single entity. if it's important make it redundant and share it among many. its common sense glued with old cryptographic tools.

data should not be accumulated forever. data must be let go of, it must be spread responsibly and allowed to fade out.
cherish not knowing everything or everyone, and not being known by everyone and everything.
cherish small purposeful scopes and groups and circles of trust.
to me there is no greater 'social network' than messaging your friends in a way that's comfortable for you both, and witnessing that these relationships reach everyone at their own pace.

i do love cryptography. i will never forgive the cryptomoney crowds from selling out 'crypto' like the latest magic stock. a honest and useful science turned into nonsense business sludge in the minds of so many people. ok the applications often were depressing i'll give you that.
cryptography is wasted on finance by capitalists and their petty selfish dreams.
everything is wasted on finance by capitalists and their petty selfish dreams.