Let's Decentralize

Alternatives to the standard internet stack

This is a website dedicated to methods of decentralized publishing on the web. The name is a reference to Let's Encrypt: what they did for securing network transmissions by offering free TLS certificates, making HTTPS possible for small indie servers, this website hopes to do for web hosting by offering simple instructions on how to host a website at home as opposed to spending exorbiant amounts of money on hosting on someone else's server. Other than the (optional) purchase of a Raspberry Pi or other small server if one wants to go the client-server route (or have a seedbox for their peer-to-peer sites), **none of these options require any money to be spent**, meaning there will never be any "web3" nonsense on this page and there never will.

Quick guide to: [GPG] [W3M]

[Tutorials]

Routing networks

These are ways to set up a standard client-server connection for applications.

Tor is "free and open-source software for enabling anonymous communication by directing Internet traffic through a free, worldwide, volunteer overlay network consisting of more than seven thousand relays in order to conceal a user's location and usage from anyone conducting network surveillance or traffic analysis." [Wikipedia]

Tor is not a silver bullet (many websites block visitors coming from known Tor exit nodes, and Google's captchas are notoriously slow, although this is due to Google being assholes), but it can be a massively helpful tool for breaking through firewalls or concealing one's browsing habits from traffic sniffers. Programs can be configured to use Tor via a SOCKS proxy or by a wrapper like torsocks.

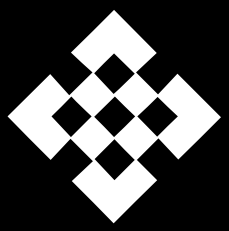
Although it is not Tor's primary purpose, as a side effect of its routing methods, it can also be used to set up "hidden services", or websites/services that can only be accessed through Tor.

[Set up a Tor hidden service] [This site on Tor] [List of Tor sites]

I2P is "an anonymous network layer (implemented as a Mix Network) that allows for censorship resistant, peer to peer communication. Anonymous connections are achieved by encrypting the user's traffic (by using end-to-end encryption), and sending it through a volunteer-run network of roughly 55,000 computers distributed around the world." [Wikipedia]

Unlike Tor, I2P's primary purpose is to facilitate hidden services. While "outproxies" exist to route clearnet traffic through I2P, these are slow, rare, and cannot guarantee any amount of privacy. Therefore, programs configured for I2P should ONLY be used for I2P. Because I2P's routing is peer-to-peer instead of through dedicated "guard" and "entry" nodes like Tor, sometimes it can fail to create routes to servers, meaning sites that are up may appear to be down on occasion.

[Set up a I2P eepsite] [This site on I2P] [List of I2P sites]



Lokinet is a decentralized onion router that uses the same service nodes as the Loki/Oxen blockchain (developed by the same team) for routing. Because servers that want to be service nodes are required to meet a minimum standard for bandwidth and processing power, Lokinet is (at least, according to the documentation) near-guaranteed to be fast and low-latency. [Lokinet homepage]

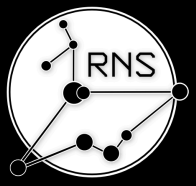
It uses a system-wide local DNS server that only handles domains ending in .loki, so it neither interferes with clearnet traffic nor requires programs to be specially configured for Lokinet. While Tor can only handle TCP traffic, Lokinet can handle any IP-based protocol, including UDP, ICMP, *and* TCP. Lokinet seems to only be able to grant each device *one* address, meaning that hosting more than one hidden service (or "snapp" as the official documentation calls them) requires configuring subdomains in one's web server configuration.

We (the webmasters) are not going to host a mirror of this site on Lokinet or keep a list of known sites maintained until the developers get their shit together and fix the source code so that it can be compiled on FreeBSD without any fatal make errors. **If one cannot compile from source, then the binaries provided might as well be proprietary.**

Yggdrasil is "an early-stage implementation of a fully end-to-end encrypted IPv6 network. It is lightweight, self-arranging, supported on multiple platforms and allows pretty much any IPv6-capable application to communicate securely with other Yggdrasil nodes. Yggdrasil does not require you to have IPv6 Internet connectivity - it also works over IPv4." [Yggdrasil homepage]

Unlike the other routing networks listed here, Yggdrasil uses IPv6 addresses instead of public keys. Thus, while not particularly anonymous, it can coexist with standard DNS resolvers; the IP Yggdrasil gives can be assigned to any standard domain or subdomain's AAAA record (although obviously it will require Yggdrasil to access). From personal experience, Yggdrasil does not seem to play nicely with Linux distributions not using systemd as their init system.

[Set up a Yggdrasil hidden service] [This site on Yggdrasil] [List of Yggdrasil sites]



Reticulum is "the cryptography-based networking stack for building local and wide-area networks with readily available hardware. Reticulum can continue to operate even in adverse conditions with very high latency and extremely low bandwidth." [Reticulum homepage]

Reticulum can connect peers over anything from traditional TCP/IP networks to darknets like Yggdrasil and I2P to exotic networking interfaces like LoRa and packet radio. All packets are encrypted; unencrypted packets are forcibly dropped from the network.

You can't (currently) directly host webpages over the Reticulum network, but a tool for both chat and hosting pages already exists: NomadNet.

[Set up a NomadNet node] [Find a list of NomadNet sites]

Peer-to-peer website sharing

These are ways to publish documents or, well, *websites* without the need for a centralized server.

ZeroNet is "a decentralized web-like network of peer-to-peer users... Instead of having an IP address, sites are identified by a public key (specifically a bitcoin address)." [Wikipedia]

ZeroNet is basically BitTorrent for websites, where instead of domains, websites are identified using a Bitcoin public key (although ZeroNet supports a few ways to link a public key to a ZeroNet-specific domain name). Unlike traditional BitTorrent, however, "zites" (ZeroNet sites) can be updated after they have been originally published while retaining the same key and peers.

Because ZeroNet is not a client-server network, traditional website applications like WordPress that require server-side languages like PHP will not work on it. ZeroNet works best with static sites (HTML/CSS/client-side JavaScript), or you can use CoffeeScript and ZeroNet's special APIs to create decentralized applications.

Please note that development in the official repository seems to have halted completely. The main developers are AWOL, and (almost?) all attempts to contact them have failed. There are multiple forks of the ZeroNet code by people seeking to continue development, but all the ones we (the webmasters) have seen have either also slowed/stopped development-wise or are developed by individuals we do not consider trustworthy enough to keep such a high-risk application as ZeroNet secure. If you *must* run ZeroNet, run it in a virtual machine, preferably also with a VPN and an isolated network.

IPFS is "a protocol and peer-to-peer network for storing and sharing data in a distributed file system. IPFS uses content-addressing to uniquely identify each file in a global namespace connecting all computing devices." [Wikipedia]

Similar to BitTorrent, IPFS allows its users to both receive data from other users who are hosting the file and to share that data in turn with other users looking for that file. Unlike BitTorrent, it seeks to create a unified global network. Files use hashes, meaning if two users publish the same file, that file will be available under the same hash.

While IPFS hashes themselves are immutable, meaning they cannot be changed once published, IPFS supports a system called IPNS where the hash can be of a peer themselves instead of the file, enabling mutable (re-writable) files and folders and ultimately websites.

[Set up an IPFS node and make a site] [List of IPFS sites]

Hyphanet is "a peer-to-peer platform for censorship-resistant communication. It uses a decentralized distributed data store to keep and deliver information, and has a suite of free software for publishing and communicating on the Web without fear of censorship." [Wikipedia]

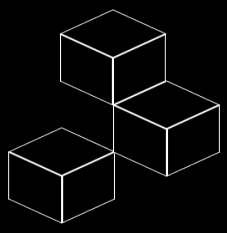
Hyphanet's first release (as Freenet) was on March 2000. It is (probably) the oldest project on this page (other than Gopher), and is still receiving regular updates to this day.

Unlike ZeroNet and IPFS, you do not necessarily have control over the data that is seeded on your device. Data is split into several small blocks, which are replicated to multiple nodes. You designate a set amount of disk space to give Hyphanet (usually between fifteen and fifty gigabytes), and it caches the most popular information on the network. The more frequently accessed a Hyphanet site is by Hyphanet users, the more users that will seed the content. Data is encrypted on disk, and can only be accessed through Hyphanet's web interface.

Also one of the devs endorsed us! Thank you!

(GUIDE TO BE WRITTEN)

[This site on Hyphanet] [List of Hyphanet sites]



Hyperdrive is a "secure, real-time distributed file system designed for easy P2P file sharing." [Hyperdrive homepage]

Hyperdrive is the successor to the Dat protocol. You can deploy static webpages, which can be viewed in a browser that supports hyper:// such as Beaker-ng or Agregore. The main implementation is unfortunately written in Node.js.

[Set up a Hyperdrive site]

Protocols

Sometimes HTTP just isn't enough (or is too much). These can be combined with routing networks to provide additional anonymity or transport security.

Gopher is "a communications protocol designed for distributing, searching, and retrieving documents in Internet Protocol networks. The design of the

Gopher protocol and user interface is menu-driven". [Wikipedia]

The experience of browsing a Gopherhole (a site on Gopher) is essentially the same as browsing a folder on your local hard drive: content is hierarchical (organized into a folder structure). While HTML files can be shared, the vast majority of Gopher clients either do not support viewing HTML in-browser or do not support CSS.

Because Gopher was created before the advent of SSL/TLS, it has *no transport security*. Someone eavesdropping on your internet connection, like your ISP or network administrator, can know exactly what files you access on a Gopherhole and what the contents of those are, and modify them in transit (a man-in-the-middle attack). This can be mitigated by serving a Gopherhole over an encrypted network like the ones in the first section of this webpage, but unfortunately the vast majority of Gopher administrators do not offer their holes on these.

Gemini is "a new, collaboratively designed internet protocol, which explores the space inbetween [sic] gopher and the web, striving to address (perceived) limitations of one while avoiding the (undeniable) pitfalls of the other." [Project Gemini homepage]

Like HTTPS, it uses TLS to secure connections (required by the protocol spec); like Gopher, it serves documents in one-off connections with minimal traffic overhead. Most Gemini clients support a new file type called "gemtext" (.gmi, mimetype "text/gemini"), which is like a stripped-down Markdown with only bullet lists, a few levels of document headers, and links that can only be on their own line. Thus it offers a bit more customization than Gopher's gophermaps, but not by much.

Nex is "a simple internet protocol designed for distributed document retrieval." [Nex homepage]

Nex has a similar goal to Gemini, that being to serve plain text and the occasional file without CSS stylesheets; unlike Gemini, and like Gopher, Nex has *no transport security* as it does not support TLS. Thus a barebones browser can be put together using nc / ncat / netcat, whichever one your operating system has available, but it should not be trusted to reliably deliver information in hostile environments where an adversary would have an interest in modifying data during transit. Again, like with Gopher, this can be mitigated by setting up the Nex server as a Tor hidden service, an I2P eepsite, etc.

Scroll is "a new document-retrieval protocol on port 5699, inspired by Gemini and Gopher+. It focuses on quality-of-life additions, particularly internationalization. It adds more metadata to responses, the ability to request pages in a specific language, and a new metadata request, similar to Gopher+'s metadata request. It tries to offer a power-to-weight balance by providing basic metadata and relational data in documents with minimal effort and additions." [Scroll homepage]

The Scroll protocol was developed to address four frustrations the designer had with Gemini:

1. lack of metadata in regards to publication/modification dates;
2. differentiating different types of links, such as citations, references, and other external resources meant to provide context, along with indicating a positive/negative sentiment towards the thing being linked to;
3. lack of formatting such as *italics* and **bold** critical for typography;
4. and poor support for internationalization and offering documents or other resources in multiple languages.

Scroll is designed specifically for documents; it is not intended for building web applications, unlike how Gemini turned out with its user certificate system.

[This site on Scroll]

Miscellaneous

Tor and friends are good and all, but even the best anonymizing network won't help if you're using services that don't respect you like Google or Facebook. These are softwares you can self-host on a server of yours, like a VPS or an old computer at home, to take back control of your data.

The **"fediverse"** is "an ensemble of federated (i.e. interconnected) servers that are used for web publishing (i.e. social networking, microblogging, blogging, or websites) and file hosting, but which, while independently hosted, can communicate with each other." [Wikipedia]

In simpler terms, the fediverse is the colloquial term for servers running publishing software (like social media or a blog) that speaks to other servers using the ActivityPub protocol. Like how Gmail users can email users on Outlook and both can (theoretically) email users running their own servers, users on a Misskey (think decentralized Twitter) server can talk to users on a Pixelfed (think decentralized Instagram) server, and both can talk to users on servers running Friendica

(think... decentralized Facebook). Many other software packages speak ActivityPub than just these three: Funkwhale, for instance, is for audio streaming like Soundcloud, and PeerTube is for video sharing like YouTube.

Many servers exist already, and each one has its own moderation policy set by its admins. Not all servers federate with each other: for example, it is often common practice for admins to block the instances of spammers, preventing either server from talking to each other, or to reject incoming media (like pictures) from NSFW-themed servers.

XMPP is "an open communication protocol designed for instant messaging (IM), presence information, and contact list maintenance. Based on XML (Extensible Markup Language), it enables the near-real-time exchange of structured data between two or more network entities. Designed to be extensible, the protocol offers a multitude of applications beyond traditional IM in the broader realm of message-oriented middleware, including signalling for VoIP, video, file transfer, gaming and other uses." [Wikipedia]

XMPP is to chat what email is to... well, mailing. Users on one server can talk to users on other servers. Many clients exist for every mainstream operating system and a great deal of obscure ones. We recommend Conversations for Android and Gajim for Linux, as these both support OMEMO for end-to-end encryption.

Git is "software for tracking changes in any set of files, usually used for coordinating work among programmers collaboratively developing source code during software development. Its goals include speed, data integrity, and support for distributed, non-linear workflows (thousands of parallel branches running on different systems)." [Wikipedia]

Git is a version control system. In other words, it's an extended undo/redo history log for files. This makes it incredibly useful for software development: end users can see exactly what parts of the code have changed between commits, and developers can see what contributors wrote what parts in case something goes horribly wrong. It can also be used for hosting websites or as a general system for distributing files to download.

Many options to host a personal Git instance exist. We personally recommend Gogs as it can be fully set up in less than five minutes and does not require a complicated database setup. Other options include GitLab and Pagure.

[List of known open-registration Git instances]

Webrings are an algorithm-free way of discovering new websites. Webmasters with similar interests voluntarily agree to share mutual links with (at least some of) the other sites in the webring, whether through banner exchanges where one hosts an image with a simple link to said website or through a fancy server-side program that one of the webmaster hosts which provides a "previous" and "next" button to cycle through all the members of the webring (which is where the "ring" part comes from).

Several webrings are open for registration: Sidebar, TheOldNet, and Hotline Webring, to name a few.

Webrings should not be confused with blogrolls, which are collections of websites that a webmaster finds interesting but where the members of the list are not necessarily aware of their inclusion and were not contacted beforehand.

Further reading

- awesome-alternative-frontends: a list on GitHub of frontends and proxies for various services, like Twitter and YouTube, many of which are self-hostable
- The Tragedy of IPFS in a series of links: an ongoing Twitter thread

[Need to contact me?]

Made with <3 by your friends at Dead End Shrine Online.

Remember. Resist. Do not comply.