Technical Architectural Blueprint for a Sovereign Web Decentralized Overlay Network

Zanno Jacklin (Creepybits)
https://zanno.se
2026-07-10
Github Repo: https://github.com/Creepybits/iroh-socks-proxy

Socio-Political Underpinnings and Threat Landscape

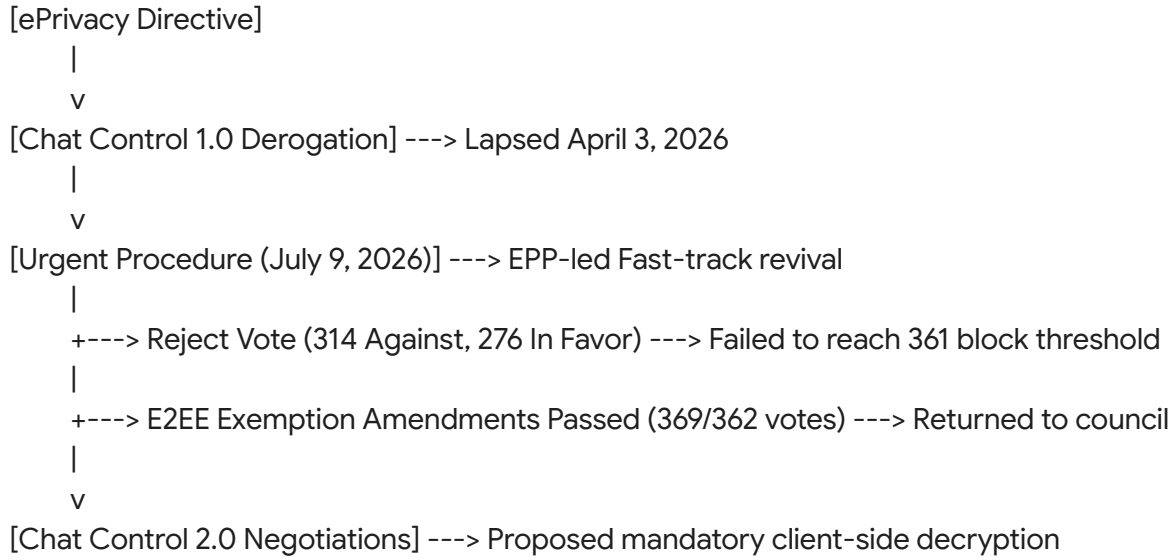
The modern digital landscape within the European Union is undergoing a rapid transition toward systemic state-level monitoring of both virtual communications and physical mobility[1]. The legislative trajectory of the Regulation to Prevent and Combat Child Sexual Abuse, colloquially known as "Chat Control," represents a significant push toward mass surveillance[2]. On July 9, 2026, the European Parliament voted on the renewal of the temporary derogation known as "Chat Control 1.0"—which permits platforms to voluntarily scan private, unencrypted digital communications[2]. This framework had previously lapsed on April 3, 2026, after a proposal for its extension was defeated [2]. The center-right European People's Party (EPP) utilized an urgent, fast-tracked legislative procedure to revive the framework before the summer recess, circumventing typical committee reviews [4].

Although a majority of voting Members of the European Parliament (MEPs) opposed the renewal (314 against, 276 in favor), the absolute majority of 361 required to block the Council's proposal was not met [2]. Consequently, the temporary scanning rules were extended [2]. However, Parliament successfully passed separate amendments with 369 and 362 votes to temporarily exclude end-to-end encrypted (E2EE) services like WhatsApp and Signal from scanning requirements [2]. This returned the legislation to the Council of the EU for a three-month review [2].

While E2EE platforms remain temporarily exempted, the broader legislative push under the

proposed "Chat Control 2.0" continues to seek mandated, suspicionless scanning of all private communications through judicial or regulatory detection orders [2]. This strategy would effectively undermine the core principles of encryption [3].

COMMUNICATION SURVEILLANCE CRITICAL PATH



Simultaneously, transportation infrastructure is incorporating deeper monitoring capabilities. National implementations, such as Swedish transport mandates aligned with European safety directives, dictate that all new passenger vehicles must feature driver-monitoring cameras and gaze-tracking sensors [8]. While framed as mitigation strategies for driver distraction and fatigue, these regulations introduce persistent interior cameras that continually capture and evaluate driver behavior [8].

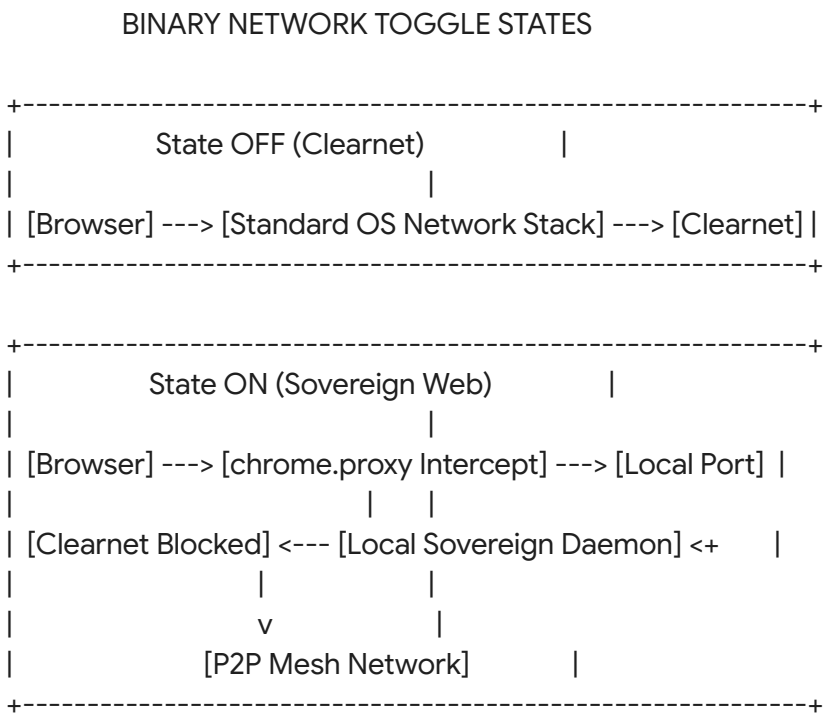
These parallel developments in communication and physical mobility highlight a broader shift: the erosion of default digital privacy through systemic, automated observation. To preserve unmonitored digital interaction, there is an urgent need for an alternative network paradigm. This paradigm must bypass centralized DNS hierarchies, avoid state-sanctioned deep packet inspection (DPI) checkpoints, and operate entirely beyond the control of corporate and sovereign authorities [1].

Evaluation of the Sovereign Toggle Logic and Structural Feasibility

The concept of a "Sovereign Web" toggle is built on a binary network switch [1]. When toggled "ON," the browser isolates itself from the standard operating system TCP/IP stack and routes all traffic through a local daemon [10]. This architecture is technically feasible and highly logical, but it faces key implementation challenges in modern browser environments [10].

Logical Soundness of the Toggle Mechanism

The core logic of using a browser-level proxy switch to toggle between the clearnet and a private overlay is sound [1]. By configuring the browser's proxy settings to point exclusively to a local loopback port (such as localhost:9999), the browser is forced to send all HTTP/HTTPS requests to the local daemon [10]. When the toggle is activated, the local daemon intercepts these requests, evaluates the destination domain, and either resolves it using a decentralized database or drops the packet entirely [10]. This approach blocks leaks to the public clearnet and prevents standard DNS queries from reaching Internet Service Providers (ISPs) [10].



Browser Extension Constraints Under Manifest V3

While the abstract network logic is sound, its execution within modern web browsers is constrained by Google's Manifest V3 (MV3) extension platform [12]. Chrome's full enforcement of MV3 disables the long-lived background pages that previously powered persistent proxy connections [12].

The standard chrome.proxy API remains functional under Manifest V3, allowing an extension to dynamically apply proxy configurations to the browser profile [14]. However, the extension's background service worker is ephemeral and subject to aggressive termination [12]. Chrome terminates an idle extension service worker after 30 seconds of inactivity, or if an event or API call takes longer than 5 minutes to resolve [16].

Because service workers run in an event-driven, non-persistent environment, they cannot host persistent TCP/UDP sockets, manage long-running peer-to-peer swarms, or maintain continuous data synchronization streams [16].

Furthermore, the replacement of the blocking webRequest API with the declarative declarativeNetRequest API restricts real-time, dynamic network interception [12]. Under MV3, the browser—rather than the extension—evaluates network requests against pre-declared rules [12]. While this improves performance and privacy, it prevents an extension from intercepting raw, arbitrary data streams on the fly to assemble peer-to-peer file shards [12].

Resolving Architectural Constraints

To overcome these limitations and make the proposed architecture viable, developers should adopt several key structural adjustments:

- **Decouple Network Operations:** The browser extension should not attempt to run peer-to-peer protocols directly [10]. Instead, it should act purely as a user interface controller and toggle switch [10]. The actual P2P stack, connection routing, and content caching are handled by a dedicated native background daemon written in Rust or Go [10].
- **Implement Local PAC (Proxy Auto-Configuration) Scripts:** The extension uses chrome.proxy to execute a PAC script [14]. This script selectively routes traffic based on destination domains [14]. For example, requests for custom top-level domains like .anon or .zil are routed directly to the local daemon at 127.0.0.1:9999 [10]. Standard clearnet traffic is either bypassed directly to the internet (when toggled "OFF") or blocked entirely (when toggled "ON" in an isolated sandbox mode) [10].
- **Maintain Persistent Service Connections:** To prevent the background service worker from terminating, the extension should establish a native messaging port using chrome.runtime.connectNative to link directly with the local daemon [16]. Connecting to a native messaging host keeps the service worker active for longer periods [16]. If the

connection drops, the service worker can be re-instantiated dynamically on subsequent browser events [16].

Alternative Paradigms and Architectural Precedents

To build a decentralized network that is more user-friendly than the Tor Network, developers can learn from several historical and modern P2P protocols [10].

Peer-to-Peer Protocol Comparisons

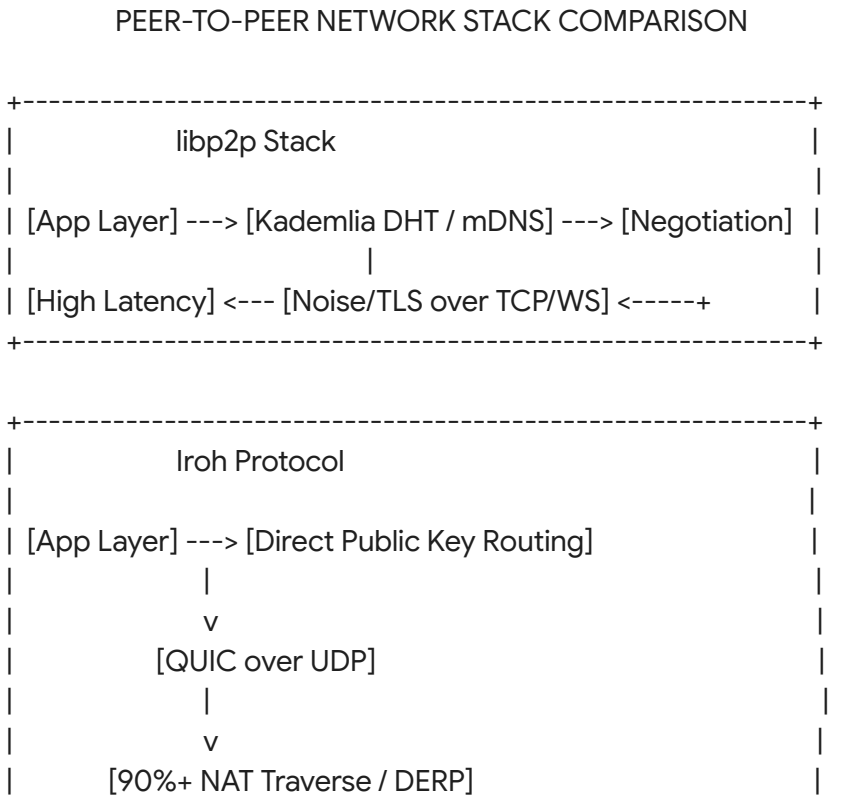
Protocol Paradigm	Routing & Resolution	Core Strengths	Technical Vulnerabilities
Tor Network	Multi-hop onion routing via central directory authorities[22].	High anonymity; robust against local eavesdropping[13].	High latency; vulnerable to exit-node monitoring and government-run entry-node correlation[13].
ZeroNet	BitTorrent trackers and Namecoin .bit registries[24].	Zero-configuration setup; offline access via local SQLite caching[25].	Lacks built-in anonymity (requires external Tor integration); vulnerable to tracker blocking[24].
Beaker (Dat/Hypercore)	Public-key addressing and Hyperswarm discovery[28].	Native versioning logs; highly optimized for mutable content and low latency[28].	Development halted (archived in December 2022); requires dedicated browser engines[28].
Proposed Sovereign Web	Decentralized Name Resolution via local SOCKS5 loopback[10].	Low latency; instant toggling; decoupled browser sandbox[10].	Requires manual trust store installation; vulnerable to extension store delisting[10].

ZeroNet and the Dat Protocol

ZeroNet demonstrates how a peer-to-peer web can run locally [22]. It uses Bitcoin-derived cryptography for content signing and BitTorrent trackers to gather peer IP addresses [24].
When a user visits a site (e.g., http://127.0.0.1:43110/{Bitcoin_Address}), the client downloads a manifest file called content.json from its peers [24]. This manifest contains the file names, cryptographic hashes, and the site owner's digital signature [24]. The client verifies the manifest's signature against the site's address, downloads the corresponding HTML/CSS resources from peers, and serves them locally [24]. Once loaded, the client seeds the content to other peers [24].
Namecoin integrates directly into this process to resolve .bit domains trustlessly [24].
The Dat Protocol (which powered the archived Beaker Browser) used an append-only log architecture called Hypercore [28]. Unlike IPFS, which addresses content by its immutable hash, Dat addressed content using public keys [28]. This allowed users to pull the latest updates directly from a writer's key-addressed drive, making the protocol highly optimized for mutable web content [28]. However, both networks require specific desktop clients or custom browser environments, which limits mainstream adoption [24].

Choosing the Network Stack: libp2p vs. Iroh

For the proposed Sovereign Web overlay, the selection of the underlying network library determines connection speed and stability [10].



+---+

Historically, libp2p has been the standard modular framework for decentralized applications [31]. It supports multiple transport protocols (TCP, WebSockets, QUIC) and manages peer routing via Kademlia DHT tables [33].

However, libp2p's modularity introduces significant complexity [32]. Before two nodes can establish a data connection, they must negotiate their encryption layers (e.g., Noise or TLS) and stream multiplexers (e.g., Yamux) [23]. This negotiation overhead increases connection latency [31]. Furthermore, libp2p's hole-punching success rate behind complex symmetric NAT routers is limited to approximately 60% to 70%, requiring costly fallback relay servers [32].

The **Iroh** protocol addresses these limitations by focusing purely on QUIC over UDP [36]. Iroh bypasses complex multi-layered protocol negotiation by using public keys directly as network addresses [40].

Additionally, Iroh incorporates a Tailscale-inspired connection engine [32]. By utilizing direct connection mapping (STUN) and falling back to a globally distributed network of secure DERP (Designated Encrypted Relay for Packets) relays, Iroh achieves a NAT traversal success rate exceeding 90% [32].

Iroh also provides three vertical, optimized P2P primitives:

- **iroh-blobs:** A high-performance content-addressable storage engine built on BLAKE3 hashing [40].
- **iroh-docs:** A synchronization layer that manages mutable key-value storage using Last-Write-Wins (LWW) CRDTs [40].
- **iroh-gossip:** An ephemeral messaging system designed for real-time signaling between peers [40].

This vertical integration makes Iroh an exceptionally fast, reliable, and lightweight backbone for the Sovereign Web prototype [32].

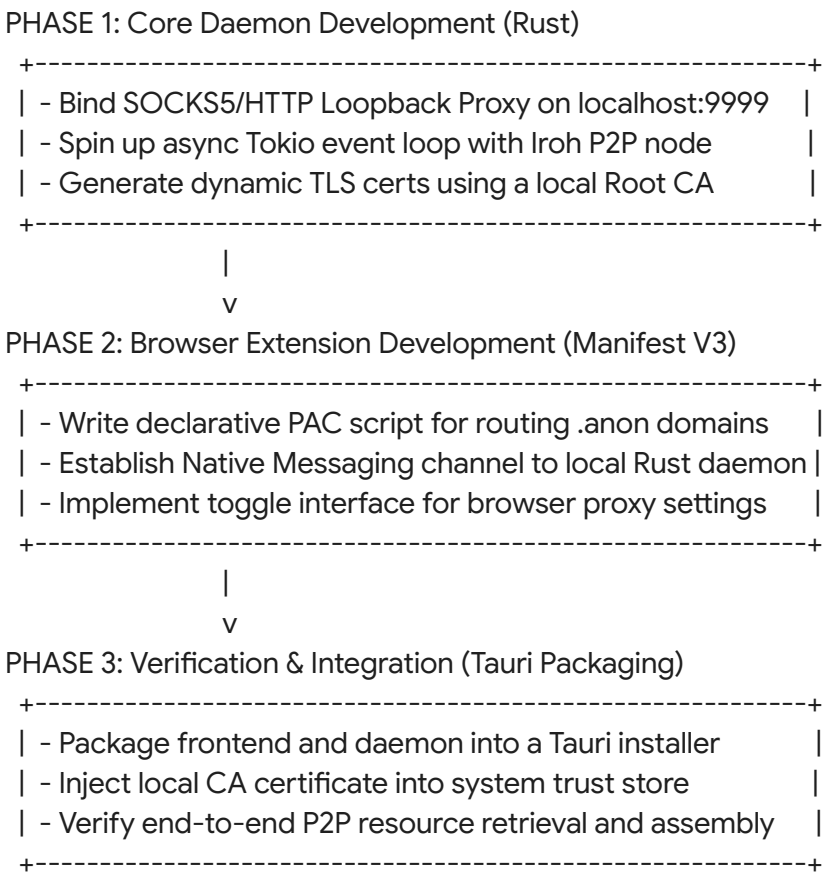
Step-by-Step Implementation and Tooling Requirements

Constructing the Sovereign Web prototype requires specific technical competencies, software development kits, and a structured implementation phase [10].

Technical Skill and Tooling Matrix

System Layer	Required Languages	Frameworks & Libraries	Key Concepts
Interface Layer	JavaScript, HTML, CSS[10].	Svelte, Tailwind CSS, Vite[19].	Chrome Manifest V3 APIs (chrome.proxy, chrome.runtime)[10].
Local Daemon	Rust[10].	tauri v2[20], iroh[40], tokio[44].	Asynchronous systems programming, custom loopback proxy routing[10].
Name Resolution	Rust[10].	hnsd / hickory-proto[45].	Alternative Root DNS, Compact Cryptographic Proofs[46].
Security & TLS	Rust, C[23].	rustls[45], rcgen[23].	Self-Signed CA Generation, SNI Filtering, Ephemeral Key Exchanges[23].

Execution Phase Roadmap



Phase 1: Core Daemon Development

The daemon is built as a background service using Rust [10]. It initiates an asynchronous runtime (using the tokio crate) to host two primary services [44]:

1. **Loopback Proxy Server:** A SOCKS5/HTTP proxy bound to 127.0.0.1:9999 [10].
2. **P2P Node Instance:** An integrated Iroh endpoint that connects to the decentralized swarm using a local cryptographic key pair [40].

The daemon also handles dynamic TLS generation [46]. Upon its first launch, it generates a unique, self-signed local Root Certificate Authority (CA) key pair [46]. The daemon exposes an endpoint or installer hook that registers this Root CA in the operating system's trusted root storage (or the browser's internal certificate store) [1].

When the user's browser requests an HTTPS resource under an alternative domain (e.g., https://example.anon), the daemon intercepts the TLS ClientHello [10]. It parses the Server Name

Indication (SNI) field [48], dynamically generates a domain-specific TLS certificate for example.anon signed by the local Root CA, and terminates the TLS handshake [46]. This allows the browser to connect securely over the loopback interface without showing certificate warnings [46].

Phase 2: Browser Extension Development

The browser extension is built using the standard WebExtension APIs under Manifest V3 [15]. The primary component is a background service worker that interacts with the user interface and the local daemon [15].

The user interface features a simple switch that sets the proxy configuration via the chrome.proxy API [10].

When toggled "ON," the extension loads a Proxy Auto-Configuration (PAC) script [14]:

JavaScript

```
JavaScript
function FindProxyForURL(url, host) {
    if (shExpMatch(host, "*.anon")) {
        return "PROXY 127.0.0.1:9999";
    }
    return "DIRECT";
}
```

This ensures that only requests destined for the alternative namespace are routed to the local daemon [10].

When the browser sends a request for a .anon domain to 127.0.0.1:9999 [10], the daemon intercepts the request [10]. It queries its local iroh-docs database to retrieve the BLAKE3 content identifier (CID) associated with that domain [40].

The daemon then streams the requested file blocks from the peer-to-peer network using iroh-blobs [40]. It verifies the integrity of the downloaded data blocks, assembles them into standard web resources (HTML, CSS, JS), and streams the assembled response back to the browser [10].

Phase 3: Integration and Desktop Packaging

To ensure a user-friendly installation process, the daemon and control interface are compiled into a unified desktop application using **Tauri v2** [19].

Tauri packages the Rust-based background daemon alongside a lightweight user interface that runs inside the operating system's native web view [19].

During installation, the Tauri installer runs script hooks with administrator privileges to automatically register the daemon's Root CA into the system's trusted root certificates[1]. This automation removes the need for complex manual network configuration, making the application accessible to non-technical users [25].

Minimum Viable Network Topology and Bootstrapping Protocol

To establish a functional testbed for the Sovereign Web prototype, the network requires a minimum physical architecture and bootstrapping strategy [10].

Minimum Node Profile Specifications

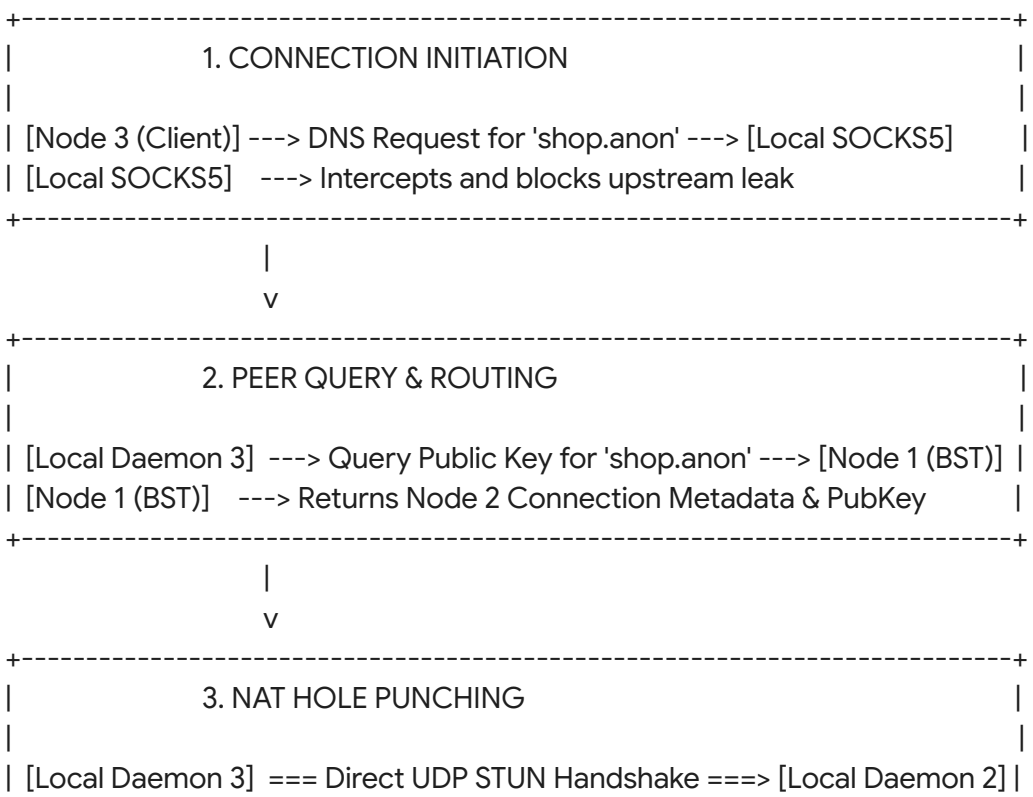
Node Persona	Network Configuration	Core Responsibilities	Resource Requirements
Node 1: Public Bootstrap	Public IP address; open ports 19301 to 19305[51].	Peer discovery; STUN coordination; DERP relay fallback[32].	Lightweight cloud VPS (1 vCPU, 1 GB RAM, unmetered bandwidth)[51].
Node 2: Host Node A	Private IP behind residential NAT; outbound UDP open[51].	Store and seed encrypted content shards[10].	Standard home PC or Raspberry Pi; continuous uptime[10].
Node 3: Client Node B	Private IP behind symmetric NAT[51].	Fetch, verify, and render content in the browser[10].	End-user laptop, desktop, or mobile device[10].

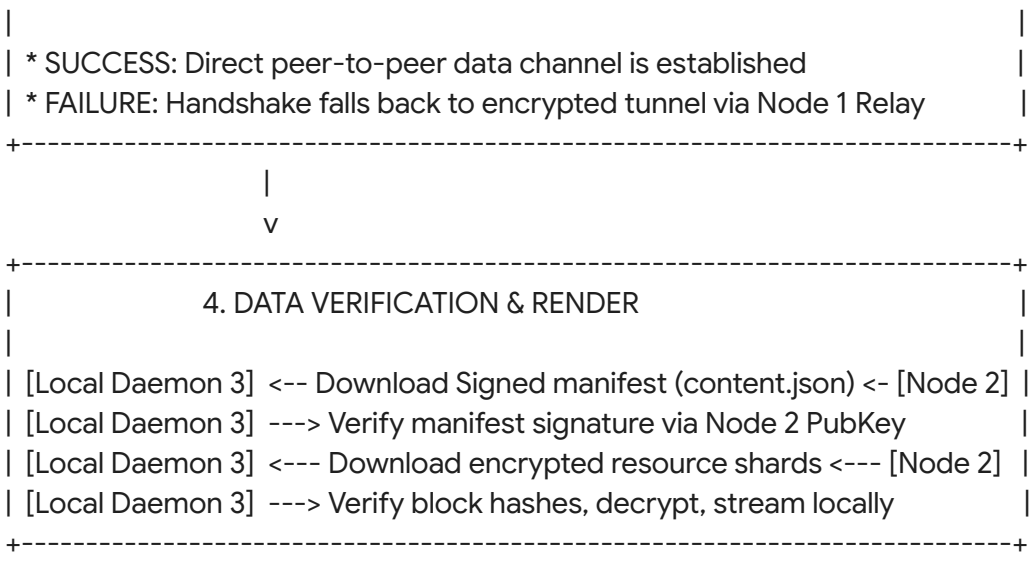
The absolute minimum requirement to verify the network is **three independent nodes** [10]. A two-node topology is insufficient because it cannot adequately test peer routing, multi-hop discovery, or fallback relay transitions [32].

- **Node 1** acts as the anchor point of the network [33]. It must be globally reachable, meaning it cannot sit behind a symmetric residential firewall or NAT [33]. It hosts the primary discovery registry and a fallback packet relay [32].
- **Node 2** serves as the publisher [24]. It registers a site (e.g., shop.anon) by publishing a signed key-value mapping to Node 1's discovery registry [24].
- **Node 3** represents the consumer [24]. It queries Node 1 to find shop.anon [25]. Once it receives the connection metadata, it attempts a direct UDP hole-punching handshake with Node 2 [32]. If hole punching fails due to restrictive firewalls, both clients fallback to communicating via Node 1's relay [32].

Node Connection and Data Exchange Lifecycle

When Node 3 attempts to access a resource hosted on Node 2, the client and daemon proceed through a multi-step networking sequence [10]:





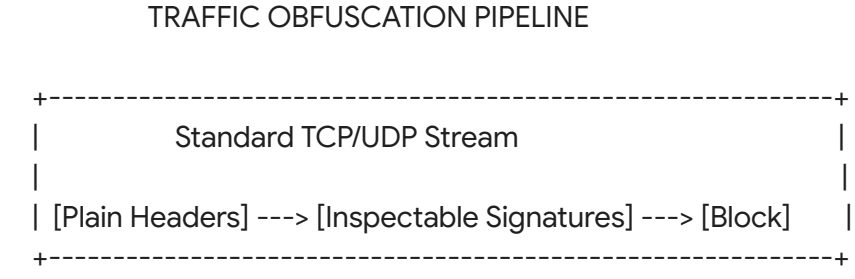
Threat Mitigation, Censorship Resistance, and Security Hardening

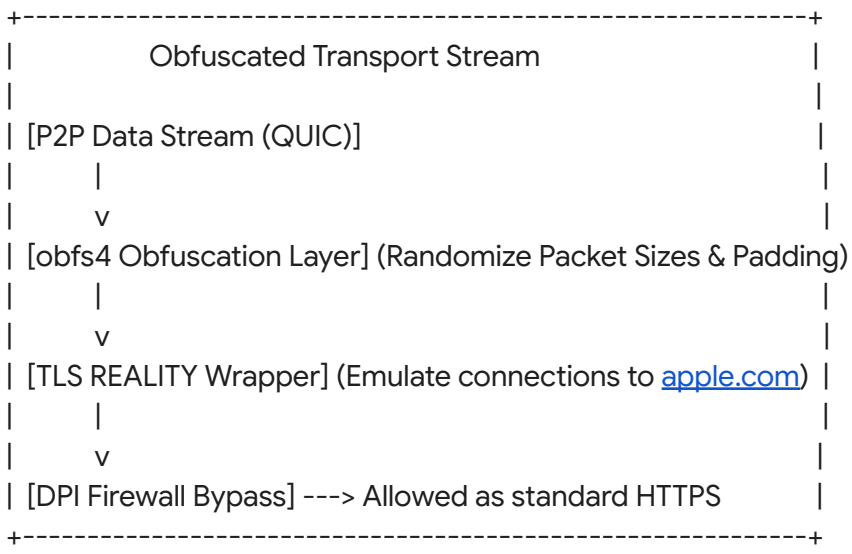
To ensure the Sovereign Web remains resilient against state-level blocklists and censorship attempts, developers must incorporate advanced traffic security measures [11].

Evading Deep Packet Inspection (DPI)

State-run firewalls monitor network traffic using deep packet inspection (DPI) to identify and block known cryptographic protocol handshakes [11].

Standard P2P protocols are highly vulnerable to DPI because they present recognizable packet sizes, connection patterns, and Application-Layer Protocol Negotiation (ALPN) headers [11].
To bypass this monitoring, the Sovereign Web local daemon must obfuscate its outbound traffic [11]:





1. **obfs4 Integration:** The daemon routes its outbound QUIC connections through an obfs4 obfuscation wrapper [11]. This wrapper scrambles packet payloads and introduces randomized byte padding and timing jitter, removing recognizable structural fingerprints from the wire [11].
2. **TLS REALITY Protocol Mimicry:** The connection layer can adopt shadowsocks or vless transports featuring the REALITY handshake [11]. REALITY intercepts the client-server TLS handshake and reuses the public security certificates of unblocked, mainstream domains (e.g., Apple, Microsoft, or cloud providers) [11]. The censor's DPI system sees a standard, secure HTTPS session with a trusted domain, allowing the connection to pass [11].
3. **Encrypted ClientHello (ECH):** Operating over TLS 1.3, the local daemon implements Encrypted ClientHello to encrypt the Server Name Indication (SNI) field [59]. This prevents on-path network observers from seeing the destination hostname of the P2P bootstrap or peer nodes [48].

Addressing Domain Hijacking and Squatting

Because the network operates outside the jurisdiction of ICANN and traditional registrars, it avoids the risks of domain seizures [1]. However, this lack of centralized control introduces risks of name squatting and domain spoofing [49]. In a decentralized network, security is maintained by separating domain resolution from content verification [49].

When a domain name is registered, it is cryptographically bound to the owner's public key [22]. When a user visits a site, the local daemon resolves the domain to this public key and verifies that all downloaded resources are signed by the corresponding private key [24].

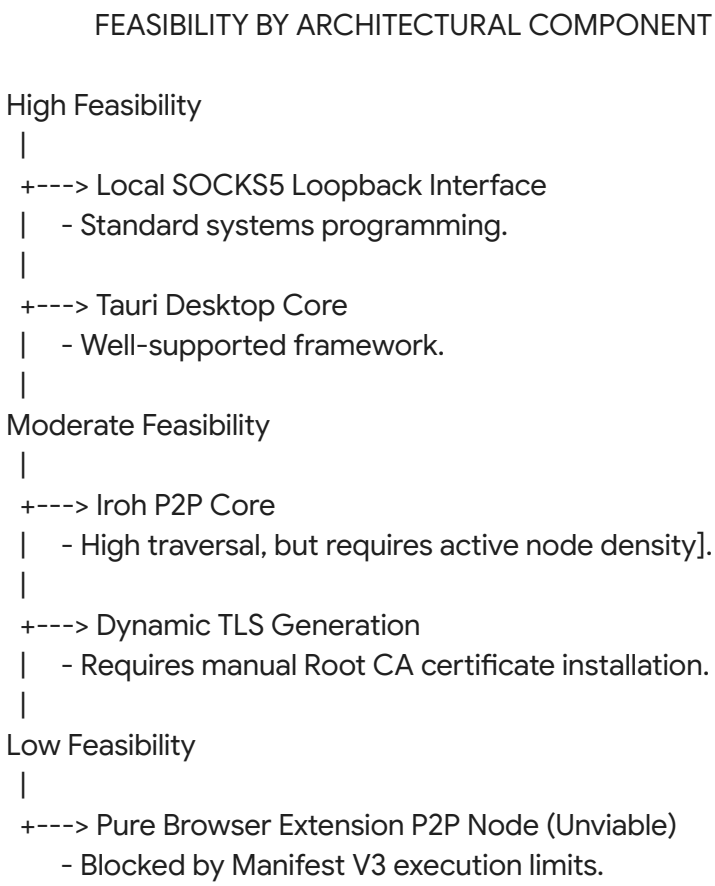
Even if an attacker attempts to hijack a domain record within the DHT, they cannot forge the

owner's cryptographic signature on the site's manifest file [13]. The browser will detect the invalid signature and immediately block the connection, protecting users from spoofing attacks [13]. To manage name squatting without a centralized registrar, developers can implement a decentralized Web of Trust (WoT) [54]. In a WoT system, domain name mappings can be validated and signed by trusted community nodes [13].

Users can import preferred trust anchors, prioritizing domains that have been cryptographically verified by trusted parties [13].

Technical Feasibility Assessment

To determine the overall viability of the proposed Sovereign Web, its technical components are evaluated based on their feasibility and complexity [10].



To achieve an optimal balance between user-friendliness and strict privacy, developers should build the Sovereign Web overlay as a **decoupled application** [10].

Rather than running as a pure browser extension (which is blocked by Manifest V3) [12], the core networking engine runs as a lightweight, native Rust daemon packaged within a Tauri-controlled system app [10].

The companion browser extension serves simply as an interface controller and domain-routing switch [10].

This hybrid architecture bypasses standard browser resource limits, avoids state-level DNS censorship [12], and delivers a secure, zero-configuration installation process that makes decentralized networking highly accessible to everyday users [19].

Sources

1. Thoughts around internet protocols and privacy.md
2. EU Chat Control 2026: The Vote That Passed and Failed at the Same Time - Hive Security, https://hivesecurity.gitlab.io/blog/eu-chat-control-2026-vote-against-majority/
3. Chat Control - Wikipedia, https://en.wikipedia.org/wiki/Chat_Control
4. EPP revives EU 'chat control' bill; WhatsApp, Signal exempt | AI Weekly, https://aiweekly.co/alerts/epp-revives-eu-chat-control-bill-whatsapp-signal-exempt
5. EU to again vote to extend 'chat control' rules, https://www.tradingview.com/news/cointelegraph:755512565094b:0-eu-to-again-vote-to-extend-chat-control-rules/
6. EP Plenary Session Voting session - EC Audiovisual Service, https://audiovisual.ec.europa.eu/de/media/video/I-292172
7. European Parliament greenlights Chat Control 1.0, will now become law. 276 In Favour, 314 Against, 17 Abstentians. - Reddit, https://www.reddit.com/r/europe/comments/1urnadd/european_parliament_greenlights_chat_control_10/
8. Nya regeln: Obligatorisk kamera på bilföraren : r/sweden - Reddit, https://www.reddit.com/r/sweden/comments/1upq2h7/nya_regeln_obligatorisk_kamera_p%C3%A5_bilf%C3%B6raren/
9. Post by @thelostswede.com — Bluesky, https://bsky.app/profile/thelostswede.com/post/3mq2go5l7w22l
10. The _Sovereign Web_ Prototype.md
11. VPN Obfuscation: Techniques, Tiers, and When You Need It - encapsulated.network, https://encapsulated.network/what-is-an-obfuscated-vpn/
12. How Users Are Adapting To Manifest V3 Changes in 2025 - TechRound, https://techround.co.uk/tech/how-users-adapting-manifest-v3-changes-2025/
13. Why Decentralization is Essential for DNS Security - Edge Network, https://edge.network/blog/decentralization-and-dns-security
14. BP Proxy Switcher & Best Proxy Extensions for Chrome 2026 - Coronium.io, https://www.coronium.io/blog/bp-proxy-switcher-usage-guide
15. chrome-dns-extension/EXTENSION_ANALYSIS.md at main - GitHub, https://github.com/hashemabad/chrome-dns-extension/blob/main/EXTENSION_ANALYSIS.md
16. The extension service worker lifecycle - Chrome for Developers, https://developer.chrome.com/docs/extensions/develop/concepts/service-workers/lifecycle
17. ServiceWorker is shut down every 5 minutes for manifest V3 extension [40733525] - Chromium, https://issues.chromium.org/40733525
18. Building persistent Chrome Extension using Manifest V3 | by Rahul Negi - Medium, https://rahulnegi20.medium.com/building-persistent-chrome-extension-using-manifest-v3-198000bf1db6

19. Tauri vs Electron: The Complete Developer's Guide (2026) - NishikantaRay, https://blog.nishikanta.in/tauri-vs-electron-the-complete-developers-guide-2026
20. Tauri vs Electron 2026: Tauri Wins: Here's the Real Data - Rustify, https://rustify.rs/articles/rust-tauri-vs-electron-2026
21. Free Communication Strategy: From Censorship to Shutdown - MindMap AI, https://mindmapai.app/mind-mapping/%D1%81%D1%82%D1%80%D0%B0%D1%82%D0%B5%D0%B3%D0%B8%D1%8F-%D1%81%D0%B2%D0%BE%D0%B1%D0%BE%D0%B4%D0%BD%D0%BE%D0%B9-%D1%81%D0%B2%D1%8F%D0%B7%D0%B8-%D0%BE%D1%82-%D1%86%D0%B5%D0%BD%D0%B7%D1%83%D1%80%D1%8B-%D0%B4%D0%BE-%D1%88%D0%B0%D1%82%D0%B4%D0%B0%D1%83%D0%BD%D0%B0
22. ZERONET: AN OVERVIEW M.Pavithra1, S.Vasanth2, R.Rajmohan 3, D.Jayakumar 1,2Departmentof CSE, PEC, Pondicherry, India 3,4Departme, https://acadpubl.eu/hub/2018-119-14/articles/3/3.pdf
23. Cryptographic Security Architecture | PDF | Public Key Cryptography - Scribd, https://www.scribd.com/document/860198384/Security-Architecture
24. Frequently asked questions - ZeroNet, https://zeronet.io/docs/faq/
25. Introduction - ZeroNet, https://zeronet.io/docs/
26. Decentralized Server Using Bitcoin Cryptography and Bittorret Network - International Journal of Engineering Trends and Technology, https://ijettjournal.org/assets/year/2017/volume-45/number-1/IJETT-V45P203.pdf
27. Chapter 5: ZeroNet - Navigating the Dark Web, https://navigating-the-darkweb.readthedocs.io/en/latest/chapter5_zeronet.html
28. Comparing IPFS and Dat - by Jay Graber - Medium, https://medium.com/decentralized-web/comparing-ipfs-and-dat-8f3891d3a603
29. Hyperdrive v10 – a peer-to-peer filesystem | Hacker News, https://news.ycombinator.com/item?id=23180572
30. We Need A New Browser, as Firefox is Funded by Google - Humane Tech Community, https://community.humanetech.com/t/we-need-a-new-browser-as-firefox-is-funded-by-google/3707
31. A libP2P Implementation of the Bitcoin Block Exchange Protocol, https://dicg-workshop.github.io/2021/papers/guidi.pdf
32. Comparing Iroh & Libp2p: Simplifying P2P Connectivity, https://www.iroh.computer/blog/comparing-iroh-and-libp2p
33. The DHT - libp2p, https://libp2p.io/docs/dht/
34. WebRTC with js-libp2p, https://libp2p.io/docs/webrtc-browser-connectivity/
35. Otaberu - Open-Source Tool to Manage P2P Networks for Edge Devices and Robotics, https://robonomics.network/blog/otaberu-part-1/
36. The Wisdom of Iroh - LambdaClass Blog, https://blog.lambdaclass.com/the-wisdom-of-iroh/
37. LP2P / Lite - GitLab, https://gitlab.com/lp2p/lite
38. Research | ProbeLab Analytics, https://probelab.io/research/
39. Better NAT traversal so that Relay servers are a last (not first) resort - GitHub, https://github.com/ipfs/camp/blob/master/DEEP_DIVES/40-better-nat-traversal-s

o-that-relay-servers-are-a-last-not-first-resort.md
40. guardian-db - crates.io: Rust Package Registry, https://crates.io/crates/guardian-db
41. My first Rust Libp2p based VPN utility under 1000 lines - Reddit, https://www.reddit.com/r/rust/comments/1kj6lep/my_first_rust_libp2p_based_vpn_utility_under_1000/
42. Iroh: peer-2-peer, but it works : r/rust - Reddit, https://www.reddit.com/r/rust/comments/1hxlen4/iroh_peer2peer_but_it_works/
43. The Complete Manifest V3 Reference for Chrome Extensions - PlugThis.ai, https://plugthis.ai/guides/manifest-v3-reference
44. Iroh: p2p chat, in rust, from scratch - Reddit, https://www.reddit.com/r/rust/comments/1i94oxv/iroh_p2p_chat_in_rust_from_scratch/
45. Network programming — list of Rust libraries/crates // Lib.rs, https://lib.rs/network-programming
46. HandyOSS/HandyBrowser: A Handshake enabled Chromium web browser and reference client; works with HSD and the HNSD light client. Use our example to build or integrate Handshake into any browser. · GitHub, https://github.com/HandyOSS/HandyBrowser
47. A Deeper Grasp of Handshake: A Thorough Analysis of Blockchain-based DNS Records, https://www.researchgate.net/publication/385668511_A_Deeper_Grasp_of_Handshake_A_Thorough_Analysis_of_Blockchain-based_DNS_Records
48. Advanced Techniques for Obfuscation with VLESS and Reality - NEXGAP, https://www.nexgap.com/post/advanced-techniques-for-obfuscation-with-vless-and-reality
49. FAQ - Namecoin, https://www.namecoin.org/docs/faq/
50. Chrome incompatibilities - Mozilla - MDN Web Docs, https://developer.mozilla.org/en-US/docs/Mozilla/Add-ons/WebExtensions/Chrome_incompatibilities
51. Minimal public network - go - libp2p, https://discuss.libp2p.io/t/minimal-public-network/1651
52. libp2p-pubsub Peer Discovery with Kademlia DHT - Medium, https://medium.com/rahasak/libp2p-pubsub-peer-discovery-with-kademlia-dht-c8b131550ac7
53. How to setup V1 Relay in the new config - Help - IPFS Forums, https://discuss.ipfs.tech/t/how-to-setup-v1-relay-in-the-new-config/14019
54. Load-balanced bootstrapping · Issue #574 · libp2p/go-libp2p-kad-dht - GitHub, https://github.com/libp2p/go-libp2p-kad-dht/issues/574
55. slskdN/docs/anonymity/obfuscated-transports-user-guide.md at main - GitHub, https://github.com/snapetech/slskdn/blob/main/docs/anonymity/obfuscated-transports-user-guide.md
56. Extended Abstract: Traffic Splitting for Pluggable Transports - Privacy Enhancing Technologies Symposium, https://www.petsymposium.org/foci/2024/foci-2024-0004.pdf

57. Edge Exemplars Enhanced Incremental Learning Model for Tor-Obfuscated Traffic Identification - MDPI, https://www.mdpi.com/2079-9292/14/8/1589
58. vpn ssh portable free download - SourceForge, https://sourceforge.net/directory/?q=vpn%20ssh%20portable
59. Handshake Encryption: Endgame (an ECH update) - The Cloudflare Blog, https://blog.cloudflare.com/handshake-encryption-endgame-an-ech-update/
60. What is Domain Hijacking and How Do I Prevent it? - SecurityScorecard, https://securityscorecard.com/blog/what-is-domain-hijacking-and-how-to-prevent-it-securityscorecard/
61. Securing the Foundation: Advanced Domain Hijacking Prevention Techniques for 2025, https://expiring.at/blog/securing-the-foundation-advanced-domain-hijacking-prevention-techniques-for-2025/