Freehold.is

the internet without landlords.

A framework for creator-owned content hosting

BY JORDAN KRUEGER · PUBLISHED APRIL 27, 2026 · V3

Abstract

Social media was supposed to connect the world. Instead, it built a handful of trillion-dollar companies that own everything you've ever posted, manipulate what you see, and answer to shareholders rather than you. Every "decentralized" alternative has tried to fix this with better protocols. None of them have fixed the actual problem: regular people still can't own their digital lives without a computer science degree.

Freehold is the internet without landlords. It's an open ecosystem where you create and consume content in a single app, own your complete archive, and can move between hosting providers without breaking anything. Your content lives on your own domain, hosted by whoever you trust, syndicated out to whatever networks you choose. No algorithms deciding what people see. No oligarchs deciding what you're allowed to say. If you stop trusting your host, you move — with everything intact.

This whitepaper describes the architecture, adoption strategy, and sustainability model for Freehold.

A note on the name

A *freehold* is property held outright — land owned free of any feudal obligation to a lord. The word describes, precisely, what this proposal is trying to give back to people online: the ability to hold your digital life on your own terms, not as a tenant renting space from a platform that can change the rules, raise the rent, or evict you at any moment.

This whitepaper was originally drafted in February 2026 under the working name *Self-Hosted Everything* (SHE). It was renamed to Freehold in April 2026. The old name attracted the wrong audience — "self-hosted" is a term of art for hobbyist sysadmins who already know how to run a VPS, and the entire point of this proposal is to make ownership accessible to everyone else. The new name says what matters: you hold it; it's yours; no landlord gets a cut.

The architecture is unchanged. The framing is sharper.

1. Introduction: the problem

In the last few years, the dream of social media — as a place where people around the world could stay in touch, have meaningful discussions, and share ideas that matter — has been corrupted beyond recognition.

It's not subtle. The platforms we built our digital lives on are controlled by corporate oligarchs who've gotten obscenely wealthy by exploiting the content and data their users create. Elon Musk — apparently a Nazi — ruined Twitter by promoting right-wing content, meddling in the platform's systems for recognizing important voices, and using it as a weapon against his personal enemies. Mark Zuckerberg, long a weak-willed man driven by a need to be popular, goes wherever the political wind takes him, with little consideration for making his platform a force for good. The list goes on, and we've got insults to spare, but the bottom line is simply this: **we can't trust these people anymore.**

And it's not just bad leadership. The business model itself is broken.

Engagement-based ranking doesn't surface what's true or important. It surfaces what's outrageous. Rage gets clicks. Misinformation gets shares. The algorithm cares about keeping you scrolling, and everything else is secondary.

Cory Doctorow coined a term for the inevitable arc: enshittification. First the platforms are good to users to attract them, then they exploit users to attract advertisers, then they exploit advertisers to extract maximum profit. Facebook did it. Twitter did it. Instagram did it. TikTok is doing it. The pattern is as predictable as gravity.

Meanwhile, privacy is basically a fiction. Your location, contacts, browsing habits, private messages, face — all of it gets harvested, packaged, and sold. Every "privacy update" is a PR exercise wrapped around a new data collection strategy.

And when platforms moderate content, they do it to protect ad revenue, not users. LGBTQ content gets flagged as "sensitive" while actual hate speech thrives in the gaps. Political speech gets suppressed when it

threatens advertiser relationships. The people making these decisions are unaccountable and motivated by profit.

Marginalized communities bear the worst of it. They're forced to share space with people who despise them, subjected to algorithmic amplification of content that targets them, and given no meaningful tools to build safe spaces they control. When platforms do act, it's usually too little, too late, and performed for PR rather than protection.

This is an ownership problem. The platforms own everything — your content, your identity, your audience, your data. And as long as that's true, their incentives will never align with yours.

Remember all those posts you made on Facebook over the last fifteen years? The baby photos, the vacation albums, the life updates, the memories of people who aren't here anymore? Right now, Facebook owns the only copy of that timeline. If they shut down, change their terms, or just mess with your feed — it's gone. Fifteen years of your life, stored on someone else's servers, subject to someone else's rules.

It doesn't have to be this way.

---

2. Why current solutions fail

The problems with centralized social media are obvious enough that many smart people have tried to solve them. Every attempt has taught us something. None of them have solved the whole problem.

Mastodon and the Fediverse

Mastodon proved that federated social media can work and that people will choose community-run servers over corporate platforms. Its ActivityPub foundation powers a real, growing ecosystem. But Mastodon has a core flaw: your server owns your identity.

Your Mastodon handle is `@you@server.social`. If that server shuts down (and many have), your identity, your followers, and your post history vanish with it. Mastodon has a migration feature, but it only moves your follower list. Your posts stay behind. In practice, almost nobody migrates even when they can.

Mastodon also only handles one content type: micro-posts. If you want photos, you need Pixelfed. Long-form writing? There's no good answer. Podcasting? Video? Not covered. The Fediverse is a collection of single-purpose tools, not an integrated solution.

Bluesky and AT Protocol

Bluesky built the most technically advanced portability system in social media. AT Protocol's portable data repositories are well-designed. On paper, it solves the lock-in problem.

In practice, Bluesky is centralized. One company runs the relay. One company runs the app view. The PLC directory, which is the root of all identity resolution, is a single service owned by Bluesky PBC, a venture capital-backed company. AT Protocol is nominally open, but Bluesky controls every piece of infrastructure that actually makes it work.

History has a lesson here. Twitter's third-party ecosystem was thriving until Twitter decided it wasn't. Bluesky's investors will eventually need returns. Building your digital life on infrastructure controlled by a VC-backed company is the same bet that failed before, dressed up in better technology.

Bluesky also handles only one content type: short posts. The protocol could theoretically support more through its Lexicon system, but in practice, Bluesky decides which content types its clients render. If Freehold built on AT Protocol, it would be a second-class citizen, dependent on Bluesky's goodwill for its content to appear anywhere.

Nostr

Nostr went maximalist on sovereignty. Your identity is a cryptographic keypair. No servers own you. No company can revoke your access. It's the purest expression of self-sovereign identity in social media.

It's also completely unusable for normal people. Lose your private key (a 64-character hexadecimal string) and your identity is gone forever. No recovery, no reset, no help desk. Password managers help, but the fundamental ask is unreasonable: "memorize or safely store this string of random characters, because if you lose it, everything is gone." Regular people *will* lose keys.

Nostr's relay model also creates fragmentation and spam problems that haven't been solved. Discovery is unreliable. The user experience is rough. It's a protocol for cryptography enthusiasts, not for someone who just wants to share photos with their family.

Ghost

Ghost is the closest thing to what Freehold envisions: a publishing platform with a sustainable business model and, as of 2025, ActivityPub federation. Ghost proves that people will pay for good publishing tools and that a small team can build a viable alternative to corporate platforms.

But Ghost only handles long-form writing. No micro-posts, no photo galleries, no audio, no video. It's a blogging platform that added federation, not a social media replacement.

IndieWeb

IndieWeb has the right philosophy: own your content, publish on your domain, syndicate everywhere. It pioneered POSSE (Publish on Own Site, Syndicate Elsewhere), which is core to Freehold's approach.

But IndieWeb is a collection of protocols and standards, not a product. Setting up an IndieWeb presence requires real technical skill — choosing hosting, configuring webmentions, setting up micropub endpoints, wiring together a dozen different tools. It's a movement for developers, not a solution for people who make things.

The common thread

Every alternative has attacked the problem from the protocol layer: build a better protocol, and the right products will follow. But protocols don't solve the adoption problem. Regular people don't choose

protocols. They choose products. The gap is a missing product ecosystem.

Project	What it does well	What's missing
Mastodon / Fediverse	Federated micro-blogging, large community	Post portability broken, single content type, fragmented ecosystem
Bluesky / AT Protocol	Best-designed portability, clean UX	Centralized in practice (VC-controlled), single content type
Nostr	Key-owned identity, censorship resistance	Unusable key management, spam, rough UX
Ghost	Good publishing, sustainable business	Long-form only — no micro-posts, photos, video
IndieWeb	Right philosophy (POSSE, own your domain)	Too technical, no unified product
Micro.blog	Multi-content hosting, POSSE syndication	Single provider, no hosting ecosystem
Known CMS	All-in-one IndieWeb publishing	Effectively dead

Nobody has built what Freehold proposes: a complete, multi-content publishing ecosystem with commoditized hosting, progressive identity sovereignty, and real portability, wrapped in a product that non-technical people can actually use.

3. The Freehold vision

From a user's perspective, Freehold is one app. You open it, you see what the people you follow have posted, and there's a button to post something yourself. That's the entire learning curve on day one. Everything else — feeds, syndication, import tools, hosting settings — is available when you need it, invisible when you don't.

Under the hood, Freehold has three components. Most users will never think about them separately, and that's the point.

The Freehold App is what you use. It's where you scroll through posts from people you follow, discover new creators, and create your own content. The reference app is a web app that works in any browser, and it handles both sides naturally: consuming and creating in the same interface, the way every social media app works. Behind the social experience is a management layer — your dashboard — where you configure feeds, manage imports, set up syndication, and handle the power-user stuff. But the front door is a timeline and a compose button, not a control panel.

Freehold Hosts are companies, organizations, or individuals that run Freehold instances on their servers. Hosting is a commodity service. You pick a host like you'd pick a web host, based on price, features, reliability, and community values. If your host disappoints you, pack up your archive and move. Your links don't break. Your followers come with you.

Third-party apps are optional alternatives to the reference app, built by anyone for any purpose. A photo-focused app, a long-form reader, a podcast player, a power-user app with columns and keyboard shortcuts. The open architecture means the same content can be displayed in

wildly different ways. You don't need a third-party app (the reference app is designed to be good enough), but the ecosystem is open for people who want specialized experiences.

Why separate "Host" from "App"? Because that's how portability works. Your host stores your data and serves it to the network. Your app is how you interact with it. If you switch hosts, your app still works — it just points at your new host. If you switch apps, your data stays put. The separation is what prevents lock-in, even though from your perspective it feels like one thing.

Core principles

Extreme portability. Your content, your identity, your feeds — all of it can move between hosts. Not theoretically, not with caveats. Actually, fully, without breaking anything. This is the non-negotiable foundation.

Commoditized hosting. No single host has leverage over you. The WordPress managed hosting industry is worth billions of dollars and built entirely on free, open-source software that you can run on any server. That's the model. Hosts compete on service quality, not lock-in.

One app, not two. Freehold doesn't ask people to use one tool for creating and a different tool for reading. The reference app does both. You post and you browse in the same place. The "creator dashboard" exists for people who want to dig into settings, analytics, and advanced features — but it's a layer behind the main experience, not a separate product.

Progressive disclosure. Freehold should be as simple as signing up for Instagram on day one. Advanced features (multiple feeds, custom domains, self-sovereign identity, syndication configuration) appear at natural moments when you need them. Complexity is always available, never mandatory.

The WordPress analogy

WordPress powers 40% of the web. It's free, open-source software that anyone can install on any server. Around it, an entire industry emerged: managed hosting companies (WP Engine, SiteGround, Kinsta), theme developers, plugin builders, freelancers, agencies. WordPress succeeded not because it was the best technology, but because it created an ecosystem where everyone from hobbyist bloggers to Fortune 500 companies could find a solution that fit.

Freehold applies that same model to your entire social media presence. The software is open and free. Hosting is a competitive marketplace. Third-party developers build apps, plugins, and tools. Your content is yours, stored in a standard format that any host can import. You are never locked in.

The day-one pitch

"One app to create and publish everywhere. Write once, and it goes to your blog, your Bluesky, your Mastodon, your RSS feed, your newsletter. Everything lives on your domain, under your control."

That pitch works on day one with zero other Freehold users. You don't need to wait for a network to exist. The app and syndication are useful immediately.

---

4. Architecture: protocol foundation

We decided early: don't build a new protocol. Don't adopt someone else's protocol as the foundation. Use a boring, owned stack with phased multi-protocol syndication.

Why not a new protocol

Building a new protocol means years of plumbing before a working product exists. Freehold's innovation is the product and ecosystem, not the wire format. The world doesn't need another protocol. It needs a product that uses existing protocols well.

Why not AT Protocol as foundation

AT Protocol is the most technically advanced thing happening in social media right now. But building Freehold on top of it would mean building on infrastructure controlled by a VC-backed company. The PLC directory (the root of identity resolution) is a single service owned by Bluesky PBC. AT Protocol's PDS (Personal Data Server) is operationally

heavy: a Go binary with a custom DAG store, CBOR encoding, cryptographic signing, and a relay firehose. That's a long way from "anyone can set up a host."

The WordPress analogy argues against it. WordPress succeeded by building on HTTP, a protocol no single company controls. Freehold should build on infrastructure that has no owner.

Boring stack + phased syndication

Freehold's canonical data store is its own: PostgreSQL for structured data, clean file storage for media, a well-defined export/import format. No existential dependency on any external protocol. If every other protocol disappeared tomorrow, Freehold would keep working.

Syndication outputs are phased by priority:

- **Phase 1 (launch):** RSS/Atom + ActivityPub. Universal syndication for any reader, plus Fediverse interop. Covers the largest reachable audience on day one.
- **Phase 2:** AT Protocol as a syndication target. Bluesky ecosystem reach without Bluesky dependency.
- **Phase 3:** WebMention and additional protocols as the community needs them.

The principle: AT Protocol, ActivityPub, and RSS are all *output targets.* None of them is the foundation. If any protocol dies, pivots, or gets acquired, Freehold keeps working. External protocols are how Freehold talks to the rest of the internet, not what Freehold is built on.

Protocol assessment

Protocol	Portability	Multi-content	Identity UX	Hosting ease	Freehold role
AT Protocol	Best designed-in	Good (Lexicon)	Moderate	Heavy (Go, CBOR, DAG)	Phase 2 syndication
ActivityPub	Broken (LOLA unshipped)	Good but fragmented	Weak (server-owned)	Most mature	Phase 1 syndication
Nostr	Best inherent	Good (NIPs)	Hard (raw keys)	Early	Watch, maybe Phase 3
RSS/Atom	Excellent	Any content type	Domain-based	Mature (it’s the web)	Phase 1 core output

5. Architecture: identity

Every decentralized system has an identity problem. Mastodon ties your identity to your server. Nostr ties it to a cryptographic key you’d better not lose. IndieWeb ties it to a domain you have to manage. Each approach works for some people and fails for others.

Freehold doesn't pick one model. It layers all three, and lets users move between them as their comfort and needs evolve. We call this "training wheels to full sovereignty."

Level 1 — Managed (default)

Sign up on a host, pick a username, get `jordan.freeholdhost.example`. Email and password login, like every other service you use. Under the hood, the host generates a keypair and signs your content, but you never see it or think about it.

This is no worse than email or social media today in terms of trust. Your host can theoretically lock you out, but portability means you can leave for another host through a transfer protocol, the same way you'd transfer a domain name.

Most people will stay at Level 1 forever, and that's fine. The goal isn't to push people toward complexity. It's to make the simple path good.

Level 2 — Domain-based

Bring your own domain. Your identity becomes `jordan.com` — human-readable, memorable, and yours. Your domain survives host migration. Repoint your DNS to a new host, import your archive, and you're back online. The old host can't stop you.

This is for creators who want a professional presence, anyone who's been burned by a platform, and people who already own domains.

Level 3 — Self-sovereign

“Self-sovereign” means you hold the proof of who you are, and nobody else needs to confirm it for you.

Here’s how it works: when you create a Freehold account, the system generates a cryptographic keypair — a private key and a public key. Think of the public key as your verified signature that anyone can check, and the private key as the stamp that creates that signature. Every post you publish is signed with your private key, and anyone can verify it came from you by checking against your public key. At Levels 1 and 2, your host holds that private key on your behalf, the same way Gmail holds access to your email. You trust them not to lose it or misuse it.

At Level 3, you take custody of that key yourself. You export it from your host and store it somewhere you control — a password manager like 1Password, a hardware security key like a YubiKey, or even a piece of paper in a safe. Your host still has a copy (they need it to sign content you create through them), but you now have an independent copy that no host can revoke.

Why does this matter? Because the private key is the ultimate proof of identity in Freehold. Whoever holds it can:

- **Prove authorship.** Every piece of content you’ve ever published is cryptographically tied to your key. No one can forge your posts, and no one can deny you wrote yours.
- **Authorize a migration without asking permission.** If you want to move to a new host, you sign a migration attestation — a statement that says “I’m moving from Host A to Host B” — with your own key.

The new host can verify it's really you. The old host's cooperation is irrelevant.

- **Survive a hostile host.** If your host shuts down, gets acquired, or decides to lock you out, your identity isn't trapped inside their systems. You take your key, your exported archive, and your domain (if you have one) to any other host and pick up where you left off. Your followers can cryptographically verify it's still you, not an impersonator.

This is the same core idea behind Nostr, but without Nostr's brutal tradeoff of making key management mandatory from day one. In Freehold, you can use the platform for years at Level 1 or 2 and never think about cryptographic keys. Level 3 is there when you want it — or if you ever need it.

This level is for technical users, privacy-focused individuals, journalists operating in hostile environments, and anyone who wants the absolute guarantee that no third party can ever take their identity.

Moving between levels never breaks anything

Same identity, same followers, same links. A Level 1 user who adds a custom domain becomes Level 2 without changing their experience. A Level 2 user who exports their key becomes Level 3 without disrupting their audience. It's a smooth ramp, not a fork in the road.

How migration works

- **Level 1:** Host-to-host transfer protocol. Requires cooperation from both hosts, like a domain transfer. Freehold's host certification program requires hosts to honor transfer requests — a host that refuses loses certification and standing in the ecosystem. The protection here is contractual and reputational rather than cryptographic, which is one reason to consider upgrading to Level 2 or 3.
- **Level 2:** Repoint your DNS to the new host. Old host can't stop you. Content moves via export/import.
- **Level 3:** Sign a migration attestation with your own key. Zero host cooperation needed.

What about interactions?

Your Freehold instance stores *your* content: posts, media, feeds. Other people's reactions (likes, replies, boosts) live on *their* instances. Your instance may cache reactions for display, but they aren't your canonical data.

This keeps the portable data package clean. When you migrate, you're moving your stuff, not trying to bring other people's data along. Your social graph (who you follow) is your data. Mutual follows are inferred, not stored redundantly.

---

6. Architecture: content types and feeds

Freehold defines seven content primitives. Feeds are collections of those primitives. Apps decide how to display them. Data is JSON. Media is content-addressed.

Seven core primitives

Instead of defining content types as “the Twitter-like one” or “the Instagram-like one,” which bakes in assumptions from the platforms Freehold is replacing, Freehold defines primitives that apps can present however they choose:

1. **Note** — Short-form text with optional media attachments. No protocol-level character limit. Covers tweets, status updates, quick thoughts.
2. **Article** — Long-form text with rich formatting: headings, links, inline images, blockquotes. Covers blog posts, essays, newsletters.
3. **Gallery** — Ordered collection of images with optional captions. One photo or fifty. Covers albums, portfolios, visual storytelling.
4. **Audio** — Audio file with metadata (title, description, duration, chapters). Covers podcasts, music, voice notes.
5. **Video** — Video file with metadata. Same idea as Audio, but heavier. Cost implications are real (more in the Media section).
6. **Link** — URL with creator commentary. The protocol fetches and stores a snapshot/preview. Covers link-sharing, bookmarking, curation.

7. **Event** — Date, time, location, and description. Covers meetups, shows, deadlines. In the reference app's timeline, an Event appears as a card with the date, time, and location displayed prominently, plus an "add to calendar" action. A specialized calendar app could render the same data as traditional calendar entries.

New primitives can be proposed through an open registry managed by the Freehold Foundation (Polls, Recipes, Reviews, Listings, Check-ins, etc.). The core seven should cover 95% of launch use cases.

Feeds as collections

A feed is an ordered collection of content items. Each feed has a name, description, content type filter (single type or mixed), access level (public, private, invite-only), its own URL, its own RSS output, and visual identity (icon, color).

Creators can have as many feeds as they want, and feeds can mix content types:

- `jordan.com/notes` — public micro-posts (Notes only)
- `jordan.com/blog` — public articles (Articles only)
- `jordan.com/photos` — public gallery (Galleries only)
- `jordan.com/family-photos` — private gallery, shared with 30 people
- `jordan.com/podcast` — public audio feed with show notes (Audio + Articles — like a podcast website where episodes and written companion pieces live together)
- `jordan.com/travel` — trip journals mixing photos, short updates, and long-form write-ups (Galleries + Notes + Articles)

- `jordan.com/music` — a musician's feed with tracks, show dates, and announcements (Audio + Events + Notes)

Single-type feeds are the simplest case, but mixed feeds are where things get interesting. A local venue could run one feed combining Events, Galleries from past shows, and Notes for announcements. A journalist could mix Articles and Links into a curated beat feed. The content types are building blocks — feeds are how creators combine them to match what they actually do.

A creator's "profile" is the collection of their public feeds. Think of feeds like TV channels for your content — you run all the channels, each has its own audience.

Your profile page

When someone visits your Freehold URL — whether that's `jordan.com` or `jordan.freeholdhost.example` — they see your profile page. This is your home on the internet: your name, bio, avatar, banner, and your public feeds presented the way you want them.

Creators control how this page looks. You choose which feeds are featured, what order they appear in, and which one is primary. A musician might lead with their Audio feed and put Events (upcoming shows) second. A photographer leads with Galleries. A writer leads with Articles and tucks Notes below.

The reference app ships with a set of free profile templates — clean, opinionated layouts that work well out of the box. Beyond those, a template marketplace lets designers sell or share templates, the same way WordPress themes work. Templates control layout, typography,

colors, and how feeds are arranged on the page, but they all render the same underlying content. Switch templates and nothing breaks — your posts, feeds, and followers stay exactly the same.

The ambition here is to bring back something the internet lost: the ability to make your page *yours*. Early social media understood this — the MySpace era proved that people want creative control over how they present themselves online. Modern platforms stripped that away in favor of uniformity. Freehold brings it back, with better defaults and without the blinking GIFs (unless you want them).

App presentation

Different apps present the same primitives differently. A “Twitter-like” app renders Notes in a timeline. A “blog reader” renders Articles in long-form layout. A “photo browser” renders Galleries in a visual grid. A podcast app shows only Audio feeds. An “everything” app mixes all primitives in a unified stream.

Creators control how their profile page looks (templates, feed ordering, visual identity). What they don’t control is how other apps choose to render their content in someone else’s timeline — that’s up to the app developer. Same content, many possible displays.

Why JSON

JSON is the right data format despite being “heavier” than binary alternatives. The reasoning: content dwarfs metadata (a single photo is 2-5 MB; the JSON describing it is ~500 bytes). Gzip compression eliminates most verbosity overhead on the wire. Host admins can open

files in a text editor and understand them, which matters a lot for an ecosystem that wants hobbyist hosts. Every language, database, and framework speaks JSON natively. And JSON has been the web's interchange format for twenty years — it's going nowhere.

WordPress's WXR export format isn't efficient either, but every tool reads it. Interoperability beats performance.

Binary formats like CBOR or Protocol Buffers matter for high-frequency streaming (Bluesky's firehose handles millions of events per second) or resource-constrained devices. Freehold has neither use case.

Export/import format

The portability contract is a zip archive containing: a JSON manifest listing all feeds and content items, all media files organized by content-addressed hash (the same image in two posts isn't duplicated), and account metadata (identity, settings, feed configurations).

Any Freehold host imports this archive and reconstructs the entire account. That's the promise.

---

7. Architecture: media, storage, and bandwidth

Not all content costs the same to host, and pretending otherwise will bankrupt small hosts before the ecosystem gets off the ground.

The cost reality

- **Text** is cheap. A million Notes is a few gigabytes. A $5/month VPS handles it.
- **Photos** are moderate. An active photographer posting 10 photos a week uses ~2 GB per year in storage.
- **Audio** is moderate. A weekly podcaster uses ~2 GB per year. Audio gets consumed once (listen and done), so bandwidth is manageable.
- **Video** is brutal. A single 10-minute 1080p video is 500 MB to 1 GB. A weekly video creator generates 25-50 GB per year. A *daily* creator: 200-350 GB. And a viral video viewed 100,000 times could mean 50-100 TB of bandwidth in a single day.

Bandwidth is the bigger cost. Mastodon instances regularly shut down because a popular post generates unexpected bandwidth bills. YouTube solves this by owning data centers. Nobody else has solved it.

Freehold takes a tiered approach, because “anyone can run a host” shouldn’t mean “anyone can accidentally bankrupt themselves.”

Three tiers

Tier 1: text + light media is the baseline every host supports. Notes, Articles, Links, Events, Galleries with reasonable image counts. Object storage (Backblaze B2, Wasabi, Cloudflare R2) at pennies per GB. A small host with 100 creators doing mostly text and photos costs $20-50/month in infrastructure.

Tier 2: audio is mid-weight. Storage is moderate, bandwidth is manageable. Can be included in baseline hosting or offered as a premium tier.

Tier 3: video needs a fundamentally different approach. Two strategies combined: a shared CDN layer (the Freehold Foundation or a host consortium operates shared media delivery, pooling costs across the ecosystem) and progressive quality (video stored at a single reasonable quality, 720p or 1080p, without multi-bitrate transcoding). Premium hosts can offer more. Baseline is "store and serve."

Freehold isn't trying to replace YouTube. Most Freehold video will be casual — short clips, personal recordings, small-audience content. Heavy video hosting is a premium service, positioned as such.

Content-addressed storage

Every media file is stored by its content hash. This means the same meme shared by 500 creators is stored exactly once. Content-addressed URLs are infinitely cacheable since the content at a given hash never changes, giving near 100% cache hit rates. And if a media file already exists on a new host (uploaded by some other creator), it doesn't need to be transferred during migration.

Cost protection

This matters more than almost anything in the early ecosystem. If a hobbyist host operator gets a surprise $2,000 bandwidth bill, that story kills the hosting marketplace before it starts.

The principle: the Host software makes it harder to accidentally spend money than to stay within budget. Cost protection isn't an add-on. It's built into the core Host experience.

The **budget dashboard** is a central part of the host admin experience. Current storage and bandwidth vs. limits. Projected monthly costs based on current trajectory. Per-creator usage alerts. A clear "you are currently spending ~$X/month" number, always visible.

Defaults are conservative, and raising them requires conscious effort. Per-creator storage quotas (1 GB default), monthly bandwidth quotas (10 GB default), per-file size limits (100 MB, which prevents raw 4K uploads). Total host storage limits. Raising any limit requires a conscious choice, and the software explains cost implications: "Raising to 5 GB means max cost at 100 creators would be ~$X/month on your detected storage provider."

Circuit breakers activate automatically for viral content. First threshold: CDN caching kicks in harder. Second: admin alert ("jordan's photo is generating 50x normal traffic"). Third: delivery auto-throttles to a sustainable rate, so content is still available, just slower. Fourth: admin-set hard ceiling, delivery pauses until manual intervention.

Pre-built hosting tier templates prevent bad defaults. "Community" is text + images only with conservative limits, cheap. "Standard" adds audio with moderate limits. "Media" enables all content types including video, requires CDN, and warns about cost during setup. Admins pick a template and customize from there, so nobody accidentally configures "free unlimited video hosting" on a $10/month VPS.

What “anyone can run a host” actually means

Anyone can run a host for text and image content. A family photo-sharing server, a writers’ collective, a local community hub — all viable on cheap infrastructure. Video-heavy hosting is a premium service. Not every host offers it. The marketplace lets hosts differentiate on media capabilities, the same way not every WordPress host supports WooCommerce.

---

8. Architecture: discovery

In centralized social media, discovery *is* the product — TikTok’s For You Page, Instagram’s Explore tab. That works because one company has all the data. In decentralized systems, no one has all the data. Mastodon essentially gave up (local and federated timelines, no real search). Bluesky’s discovery works well, but its relay and app view are completely centralized.

Freehold treats discovery as a service layer, not a protocol feature. There’s an open crawl standard, competing services, and a strong anti-algorithm default.

Direct discovery (you know what you want)

Every feed has a stable, human-readable URL. For Level 2 users with their own domain, that's `jordan.com/photos` or `jordan.com/blog`. For most people at Level 1, it's a subdomain on their host: `jordan.freeholdhost.example/photos`. Either way, every feed is an RSS feed, so any reader or aggregator can subscribe. Your URL — whether it's a domain you own or a subdomain your host provides — is your directory. Know someone's address, find everything they publish. This handles the most common case: following people you already know about.

Indexed discovery (browsing and searching)

Freehold defines a crawl-friendly standard — a sitemap-like manifest that every host publishes, describing public feeds and metadata (content type, tags, language, last updated). Anyone can build discovery services by crawling these manifests.

The Freehold Foundation runs a default reference discovery service: basic, transparent, open-source, no algorithmic ranking. Chronological plus tag-based. Think Craigslist, not Google. It ships as the default in the reference app.

Hosts can run discovery scoped to their own users or federated hosts. A photography host might build an Explore page for photo feeds across trusted hosts. Third parties build whatever they want — algorithmic

engines, curated directories, niche topic search. Community directories offer human-curated lists by interest group, the same blogrolls and webrings that already work on the web.

Social discovery (word of mouth)

Boosts and reshares in the app. Feed rolls where creators publish lists of feeds they recommend, built into the dashboard. This is how most discovery actually happens: someone you trust recommends something.

The anti-algorithm position

The Freehold Foundation takes a firm stance: the default discovery experience is not algorithmically ranked.

The reference discovery service and reference app surface content chronologically, by tag, and by explicit subscription. No engagement-based ranking. No “you might like this.” No optimizing for time-on-site.

Why? Algorithmic discovery is the mechanism that created the problems Freehold exists to solve. Rage content gets amplified because it drives engagement. Extremism spreads because outrage keeps people scrolling. The Foundation won’t build that.

Third parties are free to build algorithmic apps (Freehold is open, and people who want recommendation engines can have them), but the default experience is chronological and user-directed.

9. Architecture: moderation

Decentralization and content moderation are fundamentally at odds. Every decentralized protocol has punted on this: AT Protocol says "moderation is at the AppView layer," ActivityPub says "each server sets its own rules," Nostr says "clients filter." These are all polite ways of saying "not our problem."

But it *is* your problem the moment a host serves CSAM or becomes a harassment hub. Pretending otherwise is irresponsible.

Freehold addresses the tension directly with three layers.

Layer 1: foundation-enforced (tiny, non-negotiable)

The mandatory baseline is deliberately minimal — only things with clear, demonstrable harm and a high evidentiary bar:

- CSAM / child sexual exploitation material
- Content that exists solely because a crime was committed to create it (non-consensual intimate images, etc.)
- Active operational coordination of imminent physical harm — not rhetoric, but actual "here's the address, go now"

That's it. The list is intentionally short.

Why not more? Because "obvious" rules aren't obvious. "Incitement to violence" — MLK and Malcolm X were accused of this by the people in power. Every liberation movement in history was characterized as violent by those who opposed it. "Comply with local laws" — local laws

in Hungary criminalize LGBTQ content, and Russian law bans political opposition as “extremism.” Law is not a moral framework. “No hate speech” — the most contested term in content moderation. The line is always drawn by whoever holds power, which is exactly what Freehold is trying to escape.

Expanding Layer 1 is structurally hard by design: supermajority requirement, independent review board, sunset clauses on any new additions.

Layer 2: host-enforced (the values marketplace)

Everything beyond Layer 1 is a host-level decision. Hosts publish their own moderation policies. A progressive host can ban fascist organizing. A free-speech-focused host can allow almost anything legal. An artist community sets its own standards. So does a religious community.

Users pick hosts whose values match theirs. Portability means they can re-sort at any time — if your host’s moderation starts disappointing you, leave. This creates a financial incentive for hosts to maintain policies their users actually want.

Layer 3: app-enforced (personal)

Blocking, muting, content warnings, filtering. Moderation lists you subscribe to. Your experience, your rules.

This is where most people actually experience moderation day to day. Not through policy documents or foundation statements, but through the concrete question: “What do I see, and what don’t I see?” Layer 3 is where that question gets answered.

Moderation lists: a first-class feature

Bluesky’s moderation lists were one of the most popular features the platform ever shipped. When they launched, people loved them — and the reason was simple: someone else had already done the work of identifying the accounts you didn’t want to deal with. One click, and hundreds of bad actors disappeared from your experience. No configuration, no decision fatigue.

Freehold builds on that model and fixes what Bluesky got wrong.

How lists work in Freehold:

Anyone can create a moderation list. A list is a collection of accounts and/or hosts, with a type (block, mute, or trust), a description of what it’s for, and a public changelog showing when entries were added or removed and why. Lists are published on your Freehold instance like any other content — they have URLs, they’re portable, and they’re part of your data if you migrate.

Subscribing to a list is one click in any Freehold app. The app applies the list’s rules (blocking, muting, or trusting the listed accounts/hosts) immediately. You can browse the list contents before subscribing. You can unsubscribe at any time. You can subscribe to multiple lists from

different maintainers. If lists conflict (one trusts an account another blocks), the more restrictive rule wins by default, and you can override per-account.

Types of lists:

- **Block lists** hide content from listed accounts/hosts entirely. You don't see their posts, and they can't interact with yours.
- **Mute lists** suppress content from listed accounts/hosts in your timeline without fully blocking interaction. Useful for "I don't want to see this, but I'm not trying to cut off all contact."
- **Trust lists** work in the other direction — they mark accounts or hosts as vetted and reliable. Apps can use trust lists to surface content more prominently, skip content warnings, or prioritize in discovery. Trust is the more interesting framing (more on this below).

What makes lists portable and open:

Lists are just data — JSON, stored on your instance, with a stable URL. Any Freehold app can read any list. Lists aren't controlled by the app developer or the host. If you switch apps, your list subscriptions come with you. If a list maintainer switches hosts, the list moves with them. This is fundamentally different from Bluesky, where moderation lists are tied to the Bluesky app and infrastructure.

Trust lists, not just blocklists

Block lists and mute lists are reactive — someone does something bad, they get added to a list. Trust lists work the other direction. Instead of cataloging who to avoid, they identify who meets a standard you care

about. Both are useful, but trust lists deserve special attention because they're less common in existing platforms.

In Freehold, trust lists primarily apply to **hosts**, not individual accounts. The question they answer is: "Which hosting providers have I vetted and feel good about?" This matters because your host determines the moderation environment you're in (Layer 2), and new users need a way to find hosts that share their values.

The Foundation maintains two official trust lists for hosts:

A **"Freehold Core Values" list** includes hosts committed to anti-discrimination, LGBTQ-affirming, accessible, pro-democratic norms. Foundation-vetted through a published evaluation process. Ships as a default subscription in new apps. A **"Freehold Safe Spaces" list** is a stricter tier for hosts with active moderation, proactive content review, and strong anti-harassment enforcement — for communities serving vulnerable populations.

Being on a trust list is aspirational. Being off one isn't punishment — it's the absence of endorsement. "We trust these hosts" is an easier position to maintain than "we banned that host."

Anyone can create and publish community trust lists too. A photography community might maintain a trust list of hosts with strong anti-harassment policies. An LGBTQ organization might maintain a safe-spaces list with stricter criteria than the Foundation's. A journalism collective might maintain a list of hosts with strong editorial standards. These lists compete on reputation and usefulness, the same way spam filter services compete in email.

Learning from Bluesky's mistakes

Bluesky's moderation lists had real problems that became visible quickly. Lists were sometimes inaccurate — people got added based on guilt-by-association, misidentification, or personal grudges. There was no appeals process. List maintainers had no accountability. And because subscribing was easy and invisible, people often didn't realize they were blocking hundreds of accounts they'd never actually evaluated.

Freehold addresses this by setting quality standards for lists:

Transparency is mandatory. Every list entry must include a reason. "Added 2026-03-15: repeated harassment in replies" is a valid reason. A bare account name with no explanation is not. Apps display reasons when you browse a list before subscribing.

Changelogs are public. Every addition and removal is logged with a timestamp and explanation. Anyone can review a list's history and judge whether the maintainer is acting in good faith.

Appeals are built into the infrastructure. Every list has a standardized way to request removal. List maintainers can set their own response timeline, but the mechanism exists by default. Apps can surface "this account has a pending appeal on 3 lists you subscribe to" so subscribers stay informed.

Quality signals are visible. How many people subscribe to this list? How often does the maintainer review entries? What's the appeal response rate? How many entries have been removed after appeal? Apps surface these signals so users can judge list quality before subscribing.

The Foundation's official lists set the standard here — they commit to reviewing every appeal within 7 days and publishing quarterly transparency reports.

Foundation-promoted lists meet higher standards. The Foundation maintains a "featured lists" directory — community lists that meet published quality criteria (transparency, active maintenance, appeals process, reasonable accuracy). Being featured is an endorsement of the list's process, not every entry on it. Featured lists get prominent placement in the app's discovery interface.

The Mastodon/Gab blueprint

Mastodon's handling of Gab is the model for how organic community moderation works at scale. When Gab (a far-right platform) forked Mastodon and joined the Fediverse, the response was swift: almost every major server preemptively defederated from Gab. Shared blocklists circulated among admins. A cultural norm emerged: if you federate with Gab, other servers defederate from *you.*

Nobody took Gab's server down. They just chose not to connect. Gab exists but is effectively isolated, talking to itself. Defederation is social pressure, not censorship.

Freehold formalizes what Mastodon does informally. Hosts publish moderation policies in a standardized, machine-readable format. Apps can auto-filter based on policy alignment. When a host is widely blocked, that signal is visible — apps can surface warnings ("this host is defederated by 94% of the network"). Easy migration means users on a host that starts tolerating bad behavior can leave in minutes.

Moderation lists make this process accessible to regular users, not just server admins. In Mastodon, defederation is an admin-level decision that most users never see or influence. In Freehold, list subscriptions put that power in every user's hands.

The Foundation's role in moderation

The Freehold Foundation is not value-neutral. It has a mission statement explicitly naming its values: freedom of expression, human dignity, LGBTQ rights, racial justice, democratic norms.

The Foundation's moderation role has three parts:

It maintains official lists. The Core Values and Safe Spaces trust lists are Foundation-run, Foundation-staffed, and held to the highest transparency and accountability standards. These lists represent the Foundation's values in practice, not just in a mission statement. Maintaining them well — accurately, fairly, and responsively — is one of the Foundation's most important jobs.

It maintains a third official list for spam, scams, and coordinated inauthentic behavior. Unlike the trust lists (which are about hosts), this is a block list targeting individual accounts — spam accounts, phishing operations, and bot networks. It's less about values and more about hygiene. These are problems for everyone regardless of political orientation. The Foundation maintains this list as a baseline utility, the way email providers maintain shared spam databases.

It builds and maintains list infrastructure. The tools for creating, publishing, subscribing to, and managing moderation lists are part of the core Freehold software. The Foundation develops and maintains

these tools, including the appeals mechanism, the quality signals, and the featured-lists directory. Good tooling is what makes community-driven moderation possible at scale.

It promotes, but doesn't mandate. The Foundation actively promotes high-quality community lists through the featured directory. It publishes statements when hosts or communities violate its stated values. It runs programs, grants, and partnerships advancing those values. But it doesn't wield a ban hammer. It can't force a host offline or compel users to subscribe to any list. It leads with persuasion, infrastructure, and the power of good defaults.

The Foundation's leverage is that its official lists ship as default subscriptions in the reference app. Most users will never change those defaults. That's significant power, and the Foundation takes it seriously — which is why the official lists have the strictest transparency, accuracy, and appeals requirements in the ecosystem.

---

10. The user experience

Most people who use Freehold will never think of themselves as "creators." They'll think of themselves as people who post things and read things other people posted. The app needs to work for them first.

The front door: a social app

When you open Freehold, you see a timeline — posts from people you follow, displayed chronologically. Notes show as compact text cards. Articles show as title-plus-excerpt. Galleries show as thumbnail grids. Audio has an inline player. It looks and feels like a social media app, because it is one.

There's a compose button. Tap it, write something, post it. For most people, most of the time, that's the entire experience: read, post, read, close.

This is non-negotiable. If the first thing a new user sees is a dashboard with six menu items and a "choose your content type" prompt, we've already lost them. The front door is familiar, because every social app in the world has trained people to expect a timeline and a compose button.

Inline creation

Creating content happens inside the social experience, not in a separate tool.

Notes are the default. Tap compose, type, optionally attach a photo or video, post. The simplest path does the most common thing.

Articles start as Notes and expand. Start typing, and if you keep going past a few sentences, a subtle prompt appears: "This is getting long — want to switch to the article editor?" The article editor is a distraction-

free rich text environment (Ghost/Notion-style block editing). You can also go directly to the article editor from the compose menu if you know you're writing something long.

Galleries are created by attaching multiple photos. Attach one photo and it's a Note with an image. Attach several and Freehold automatically presents it as a Gallery with drag-to-reorder, batch upload, and captions.

Audio, Video, Links, and Events are available from an expanded compose menu — a row of content type icons that appears when you tap a "more" button next to the compose field. Each gets a tailored form, but it's still inside the app, not in a separate management interface.

On publish, content goes to your default feed. That's it. No feed selection, no syndication configuration, no decisions. Those options exist for people who want them — accessible via an "advanced" toggle on the publish screen — but the default path is: write, post, done.

The dashboard: behind the scenes

Behind the social experience is a management layer for people who want more control over their feeds, syndication, imports, analytics, and moderation settings.

Activity shows what happened since your last login: boosts, new followers, view counts, import progress. Storage and bandwidth usage at a glance.

Feeds lets you manage your feeds: create, edit settings, reorder, set access levels. Preview recent items and basic stats per feed.

Archive is your complete digital history — everything created or imported. Searchable and browsable by content type, date range, source platform, and published/unpublished status. This is where you rediscover old content and optionally surface it to feeds.

Syndication shows connected platforms, publish history, failure handling ("Bluesky post failed — image too large. Retry compressed?"), and lets you connect new platforms.

Moderation shows your list subscriptions, lets you browse and subscribe to new lists, and manage your personal blocks and mutes.

Settings covers identity management, hosting details, import/export, and sovereignty level controls.

The dashboard is available from a menu icon in the app. It's not hidden, but it's not the first thing you see. The target feel for the dashboard is Notion or Bear: clean, calm, and layered. WordPress has 47 menu items. Freehold's dashboard has six.

Feeds: a gentle on-ramp

Feeds are the most novel concept in Freehold, and they need careful introduction. Start simple, introduce complexity at natural moments.

Every new user gets a single auto-created feed — "My Posts" or whatever they name it. Everything goes there. Freehold feels like any other platform. Nobody needs to understand feeds on day one.

Second feeds get introduced through contextual prompts at natural moments:

- Import mixed content → “These are pretty different. Want photos in a separate feed so people can subscribe to just photos or just your blog?”
- Try to create a private post → “Your feed is public. Want a private feed for certain people?”
- First Gallery after mostly Notes → “Some followers might want just your photos. Give galleries their own feed?”

And if you never want multiple feeds? Fine. Everything goes to “My Posts.” That works perfectly, forever.

Import: dump truck, not curate

Your Freehold instance is your private vault before it’s your public presence. The import model reflects this:

1. Upload a platform export zip (Facebook, Instagram, Twitter, WordPress, etc.)
2. Freehold ingests everything into your private archive
3. Done. Go about your day.
4. Whenever you want, browse your archive, rediscover old content, selectively surface things to public feeds — or leave it all private forever

You never *have* to curate. The archive is valuable on its own as your personal record. Import tools handle the messiness: deduplication, date preservation, quality transparency (“These photos were exported from Instagram at reduced quality. If you have originals on your phone, you can upload those instead.”), and volume management (“Found 4,312 posts, 12,847 photos, 234 videos — import all or filter by date/type?”).

First-class platform importers

Built into the dashboard:

- **WordPress** → Articles + Gallery items, preserving categories/tags (well-documented export format, gold standard)
- **Ghost / Substack** → Direct content mapping (clean JSON/HTML)
- **Twitter/X** → Tweets become Notes. Threads become linked Notes or merged Articles. Likes/bookmarks become an optional private feed.
- **Instagram** → Photos become Gallery items. Stories become a separate feed or Gallery. Reels become Video items.
- **Facebook** → Posts become Notes or Articles (by length). Photos become Gallery items. Messiest format — import wizard shows what was found, user picks what to bring in.
- **Bluesky** → Direct repo import
- **RSS/Atom** → Point at any feed, import entries. Covers any platform with RSS output.

Generic importers for power users: Markdown folders become Articles (for Obsidian/Bear users), photo folders become Gallery items with EXIF auto-organization, CSV becomes structured import for anything. Community-contributed importers via a plugin API cover everything else.

---

11. The ecosystem: hosts and apps

The host marketplace

Freehold Hosts run the Freehold software and provide infrastructure for creators. The marketplace works like WordPress hosting:

- Some hosts are cheap and minimal — $3/month for text and photos
- Some are premium — $15-50/month with video support, priority support, analytics, and custom domain setup
- Some are free — nonprofits, universities, community organizations running hosts for their members
- Some serve niche communities — a photography collective, a writers' group, an LGBTQ safe space

Hosts compete on price, reliability, features, community, and values. Portability ensures no host can coast on lock-in. If your host gets worse, you leave, and your content, identity, and audience come with you.

Hosts do not inject ads into creator content or sell user data (both are trust list requirements). And they can't lock you in, because the architecture makes that impossible.

The host package: easy to start, easy to run

The host marketplace only works if becoming a host is accessible. If standing up a Freehold instance requires a systems administrator and a weekend of configuration, the hosting ecosystem won't grow. WordPress succeeded partly because installing it takes five minutes and every hosting company on earth offers it pre-installed. Freehold needs the same.

The Freehold host package is a single, self-contained application — the software that runs a Freehold instance. The goal is to make it available everywhere people already get hosting:

- **One-click installs** on platforms like DigitalOcean (App Marketplace), Hostinger (Auto Installer), Cloudflare, and Railway
- **Pre-installed options** through traditional hosting companies that include Freehold alongside WordPress, Ghost, and other CMS options in their control panels (cPanel, Plesk, Coolify)
- **Docker image** for people who want to run it on their own server
- **Manual install** for the handful of people who want to do it from scratch

A new host operator picks a platform, clicks install, and gets a working Freehold instance with sensible defaults. The setup wizard walks through the basics: host name, admin account, storage provider connection, and moderation policy selection (start from a template or write your own). From there, creators can start signing up.

Storage is offloaded by default. The host package doesn't expect operators to serve media from the same server running the application. During setup, the wizard connects to an external object storage provider — Cloudflare R2, Backblaze B2, AWS S3, Wasabi, or any S3-compatible service. Media uploads go directly to the storage provider, and the host serves lightweight JSON and metadata. This is what keeps hosting costs predictable: a small host with 100 creators doing text and photos might use $5/month in object storage, not hundreds of dollars in server disk and bandwidth. The host software handles the integration — operators don't need to understand CDN configuration or bucket policies.

Once a host is running, the admin tools are designed for non-technical operators. The host admin interface is a web UI — not a terminal, not config files. It covers:

- **Creator management** — who's signed up, storage usage per creator, account status
- **Cost monitoring** — the budget dashboard described in Section 7, with projected costs, usage trends, and alerts before anything gets expensive
- **Moderation tools** — review reported content, manage host-level policies, subscribe to Foundation block lists
- **Updates** — one-click software updates with changelogs that explain what changed in plain language, automatic security patches
- **Backups** — automated backup scheduling to the storage provider of your choice, one-click restore
- **Health checks** — is everything running, are there errors, is storage getting full. Simple green/yellow/red status, not server logs

The target operator is someone comfortable managing a WordPress site or a Shopify store — not a DevOps engineer. Freehold's hosting ecosystem needs thousands of small hosts to thrive, and that only happens if the barrier to entry is low and the ongoing maintenance isn't intimidating.

The reference app

The reference app is the "Gmail" of Freehold — a good default that proves the concept and gives new users a complete experience from day one. Web app first (any browser, no install), mobile-responsive from launch.

As described in Section 10, the reference app handles both creating and consuming in a single interface. The **timeline** is the front door: a chronological feed of followed creators, all content types mixed and displayed appropriately. **Inline creation** means you post without leaving the social experience. The **dashboard** lives behind a menu for people who want feed management, syndication config, import tools, and analytics.

Creator profiles show all public feeds for a creator, browsable individually or mixed. Per-feed follow granularity lets you subscribe to someone's photos but not their political commentary.

Discover connects to discovery services. Browse by tag, content type, recency. Community directories and curated lists. Moderation lists are discoverable here too — browse featured lists, see what's popular, subscribe.

Interactions include follow/unfollow (per-feed), boost, like/react, comment (if enabled), and save/bookmark (private, on your own instance).

Moderation is built into the social experience, not buried in settings. Block and mute actions are available on every post and profile. List subscriptions are accessible from a dedicated tab. When you block someone, the app suggests relevant moderation lists: “3 lists you subscribe to also include this account” or “Want to report this account to list maintainers?”

The reference app is deliberately low-key. No infinite scroll. No notification guilt trips. Open, see what’s new, close. It’s the Honda Civic of social media apps — reliable, does the job, isn’t trying to impress you.

Third-party apps

The open primitive/feed architecture makes room for specialized apps:

- A **photo app** focused on Galleries, visual grid, Instagram-like browsing of Freehold photo feeds
- A **reader app** for Articles only, Instapaper/Pocket-like reading experience
- A **micro-post app** for Notes only, Twitter-like timeline, text-first
- A **podcast app** for Audio feeds in a standard podcast player UX
- A **community app** scoped to specific hosts, forum/subreddit feel
- A **power app** with columns, filters, keyboard shortcuts, multi-account — TweetDeck for Freehold

Third-party apps can focus on consumption only, creation only, or both. A podcast app probably doesn't need a compose button. A power-user app probably wants a richer editor than the reference app provides. The architecture supports all of these because the data layer (hosted on your instance) is separate from the presentation layer (the app you use).

The reference app should be good enough that you don't *need* a third-party app, and modest enough that there's clear room for better, specialized options.

---

12. Monetization and sustainability

No crypto. No ads. No data selling. Boring, proven models only.

Hosts: the WordPress hosting model

WP Engine, Kinsta, and SiteGround all built profitable businesses on free WordPress software. This is a proven model worth billions. Freehold Hosts work the same way:

- Monthly creator subscriptions, tiered by storage/bandwidth/features. Community tier: $0-3/month. Standard: $5-15/month. Media (with video): $15-50/month.
- Premium features: custom domains, priority support, analytics, extra storage, video transcoding, backup frequency

- Plugins and integrations: email newsletter delivery, e-commerce tools, SEO tools, custom themes

Hosts compete on price, reliability, features, and community values. Some are cheap and minimal. Some are premium. Some are free. Same as web hosting.

Creators: optional, not expected

Most creators don't monetize and don't want to. Freehold works perfectly with no money changing hands.

For creators who want to earn: **paid feeds** let you mark a feed as paid, where subscribers pay you directly (via Stripe or similar) and the host takes a small transaction fee (5-10%). This covers newsletters, exclusive content, paywalled articles. **Tips and one-time payments** work through a "support this creator" button with direct payment. And you can always bring your own monetization — affiliate links, sponsorships, whatever. Your content, your choice.

Freehold doesn't build an ad marketplace or a "promote this post" feature. There's no algorithm to promote into.

The Freehold Foundation

Host certification fees are the primary revenue source — annual, scaled by size, same model as the Linux Foundation or Wi-Fi Alliance. Small community hosts pay little or nothing; commercial hosts pay more. The shared CDN charges hosts for actual usage rather than levying a tax, and hosts can use their own CDN instead. Early on, grants from tech-for-good foundations (Mozilla, Ford, Knight) buy time while the

ecosystem develops. Corporate sponsorships from hosting, storage, and CDN companies add another layer. Down the road, paid enterprise features (compliance reporting, bulk admin, SLA-backed services) serve universities and large nonprofits.

The Foundation does not take equity. It's a nonprofit — no investors, no exit. It doesn't monetize user data or charge mandatory fees for basic participation. You can run a host without paying the Foundation. You just don't get trust list certification or access to the shared CDN.

Money flow

```
Creators  →  pay Hosts (subscription)
Consumers →  pay Creators (optional, for paid feeds / tips)
Hosts     →  pay Foundation (certification, CDN usage)
Grants    →  fund Foundation
```

Nobody has to pay to consume. Nobody has to pay to create (free hosts will exist, just as free WordPress hosting does today). Money comes from hosting infrastructure and optional premium features — the same model as WordPress, email, and the web.

---

13. Adoption strategy

The cold start problem

Portability is meaningless with one host. Network effects are meaningless with zero users. “You can leave someday” isn’t a reason to show up today. Freehold needs a standalone value proposition that works before the ecosystem exists.

The wedge: dashboard + POSSE + import

The day-one pitch doesn’t depend on the Freehold ecosystem existing at all:

“One app to create and publish everywhere. You write once, and it goes to your blog, your Bluesky, your Mastodon, your RSS feed, your newsletter. And everything lives on your domain, under your control.”

A creator signs up, connects their accounts, and immediately has a personal website on their domain with all their content, automatic syndication to Bluesky and Mastodon, an RSS feed for everything, a single app replacing five, and a complete archive they own and can download at any time.

The ecosystem grows from there. Portability (the long-term moat) isn’t the launch feature. It’s the foundation that makes everything else trustworthy.

Target user tiers

Tier 1: indie bloggers and writers who already care about ownership. IndieWeb community members, existing WordPress and Ghost users, newsletter authors frustrated with Substack's direction. A small audience, but passionate and vocal. They set the tone.

Tier 2: creators burned by platforms. Every few months, a platform does something that drives a migration wave — algorithm changes, policy reversals, hostile acquisitions. These moments create energy. Freehold needs to be ready when they hit: "Bring your archive. Set up in 10 minutes. Never worry about this again."

Tier 3: communities with specific needs. One decision-maker brings many users. A university giving professors a professional web presence. A nonprofit wanting a private content platform for members. A creative collective wanting a shared gallery without Instagram's algorithm. LGBTQ communities wanting a safe space they actually control.

Tier 4: mainstream. Much later. Only when "my friend set it up for me" is a real path.

Ghost's lesson

Ghost launched in 2013 as "a better blogging platform." For years, that's all it was — writers used it because the writing experience was excellent. It didn't try to be a social network. Didn't need network effects. Just needed to be a good product.

In 2025, Ghost added ActivityPub and became federated. By then, it had a solid user base, a sustainable business, and credibility. Network features were additive, not foundational.

The lesson: be a great product first, and worry about being a network later.

Development priority order

1. **Reference app** — The unified social + creation experience. Timeline, compose, inline creation for all seven primitives. This is the product.
2. **POSSE syndication** — Publish to Bluesky, Mastodon, and RSS from day one
3. **Personal website generation** — Freehold instance *is* your website, with good defaults and custom domain support
4. **Import tools** — Bring your WordPress, Substack, Instagram, Facebook, and Twitter archives
5. **Dashboard layer** — Feed management, syndication config, archive browsing, analytics. Available from day one but not the focus of early marketing.
6. **Moderation lists** — List creation, subscription, and the Foundation's official lists. Ship alongside or shortly after the app launch.
7. **Host package** — Easy to stand up a host
8. **Portability and migration** — Architected from the start, but not the marketing message until multiple hosts exist

---

14. Roadmap

Phase 1: foundation (months 1-6)

- Publish whitepaper, gather feedback from trusted builders and potential early adopters
- Define JSON schema spec for all seven content primitives
- Define export/import archive format spec
- Define moderation list format spec (list types, changelog format, appeals mechanism)
- Build proof-of-concept reference app (single-user, timeline + inline creation)
- Begin POSSE syndication implementation (RSS + Bluesky + Mastodon)

Phase 2: alpha (months 6-12)

- Functional reference app with timeline, creation for all seven content types, and dashboard layer
- Working syndication to RSS, Bluesky, Mastodon
- Personal website generation from Freehold instance
- Platform import tools (WordPress, Twitter/X, Instagram first)
- Moderation list infrastructure: create, publish, subscribe, browse
- Foundation’s initial official lists (Core Values trust list, spam/scam block list)
- Single-host deployment: one operator, multiple creators

- Invite-only alpha testing with Tier 1 users (indie bloggers, IndieWeb community)

Phase 3: beta (months 12-18)

- Host package: documented, tested, deployable by someone with basic server skills
- Host-to-host migration (the portability promise, tested and working)
- ActivityPub federation (Phase 1 syndication complete)
- Profile page customization: template system, free built-in templates, template marketplace for third-party designers (the MySpace energy, but with good defaults)
- Full moderation tooling: appeals workflow, quality signals, featured lists directory
- Foundation's Safe Spaces trust list launched
- Cost protection suite fully implemented in Host software
- Open beta with Tier 2 and Tier 3 users

Phase 4: launch (months 18-24)

- AT Protocol syndication (Phase 2 syndication)
- Multiple hosts operating independently
- Foundation formation (nonprofit structure, governance, certification program)
- Shared CDN layer operational
- Third-party app API stabilized

- Public launch

Long-term

- Native mobile apps (reference app)
- Plugin ecosystem for hosts and creators
- Additional platform importers (community-contributed)
- Enterprise features for universities, nonprofits, organizations

---

15. Conclusion

The internet was supposed to be ours. Somewhere along the way, a handful of companies convinced us that the right way to share our lives was to hand everything over to them — our photos, our words, our friendships, our data — and hope they'd treat it well. They didn't.

Freehold doesn't ask you to trust a new platform. It asks you to trust a model that's already been proven: open-source software, commoditized hosting, extreme portability, and the principle that your content belongs to you. It's the model that built the web. It's the model that powers 40% of all websites through WordPress.

The architecture is ready. The design decisions are made. What comes next is building, and building well enough that the first person who tries Freehold doesn't need to understand protocols or federation or content-addressed storage. They just need to feel, immediately, that this is *theirs.*

Whether you're a creator tired of platforms that treat you as a product, a developer who thinks the web should be open and owned by its users, or just someone who wants to know those fifteen years of posts and photos will still be there on your terms — we're building this, and we could use your help.

Read more and follow progress: freehold.social **Get in touch:** jordan@jordankrueger.com

---

References and prior art

Projects and platforms

- **AT Protocol / Bluesky** — Portable micro-blogging with lexicon-based content types. Informs Freehold's syndication strategy and identity model. atproto.com
- **ActivityPub / Mastodon** — Federated social networking with the largest decentralized installed base. Freehold Phase 1 syndication target. Defederation model directly informs Freehold's moderation architecture. activitypub.rocks
- **RSS/Atom + IndieWeb** — The original and most durable syndication layer. IndieWeb pioneered POSSE (Publish on Own Site, Syndicate Elsewhere). indieweb.org
- **WordPress** — Model for commoditized hosting marketplace. Proves that open-source software can power a multi-billion-dollar hosting ecosystem. wordpress.org

- **Ghost** — Creator-focused publishing with sustainable business model. Added ActivityPub federation in 2025. Closest product-level comparison. ghost.org
- **Micro.blog** — Multi-content POSSE publishing. Closest concept-level comparison. Limited to single provider (no hosting ecosystem). micro.blog
- **Nostr** — Key-owned identity and censorship resistance. Informs Freehold Level 3 (self-sovereign) identity. nostr.com
- **Known CMS** — All-in-one IndieWeb publishing platform. Effectively abandoned but validated the multi-content personal publishing concept. withknown.com

Concepts and movements

- **Enshittification** — Cory Doctorow's framework for how platforms degrade: attract users, exploit users, exploit advertisers, extract maximum value.
- **POSSE (Publish on Own Site, Syndicate Elsewhere)** — IndieWeb principle. Your canonical content lives on your domain; copies go to platforms for reach.
- **Web Revival movement** — A grassroots cultural movement promoting user ownership, creative freedom, and anti-corporate values on the web. Freehold provides the technical infrastructure the Web Revival's philosophy calls for, making ownership accessible to people who will never hand-code HTML.
- **Defederation as social pressure** — The Mastodon/Gab model of community self-organization. Not censorship, but the right to choose who you connect with. Formalized in Freehold's trust list architecture.

Technical foundations

- **PostgreSQL** — Freehold's canonical data store. Open-source, no vendor lock-in.
- **JSON** — Freehold's data interchange format. Universal tooling, human-readable, 20-year track record.
- **Content-addressed storage** — Media files stored and referenced by content hash for deduplication, cacheability, and migration-friendly media handling.

License

The architecture described here is meant to be implemented. If you're a developer, founder, or organization who wants to build on it, you don't need permission. Build it.

This whitepaper is a living document. Architecture decisions are based on the comprehensive design session of February 22, 2026. Feedback, critique, and collaboration are welcome.

Freehold.is *the internet without landlords.*