

Yes — here is the fully integrated, production-ready Software Development Plan v1.1 (SDP v1.1)

All previous suggestions have been woven in:

- OrbitDB v3 as **primary** mutable feed store (IPNS only for cold-start discovery & announcement)
- @helia/verified-fetch + delegated routing
- Tauri 2 resource bundling + mobile preview + QR pairing
- Formal threat model + UCAN best practices (new Phase 1.5)
- Private-node onboarding wizard + “Easy Mode”
- Built-in public directory service (opt-in)
- PWA + service-worker background sync
- Enhanced testing (chaos, benchmarks, multi-device)
- Schema versioning & migration
- Storacha cost monitoring + auto-pinning incentives
- Risk register, milestone demos, open-source from day 1
- External security audit milestone

Result: a tighter, more resilient, more adoptable plan that still delivers your exact vision — immutable CIDs, RSS-style publish/subscribe feeds, DID + UCAN auth, private IPFS nodes — while hitting **<1 s updates** for active followers, sub-5 s cold follows, and enterprise-grade security/UX.

App Name (your choice): CIDFeed or EchoNet

Target MVP Launch: ~11 weeks solo (9 weeks with 1 helper) — realistic February–May 2026 timeline.

1. Project Vision & Core Concept (Unchanged + Strengthened)

A sovereign, decentralized social layer where every user publishes an **RSS-style feed** of immutable CIDs (text, images, videos, arbitrary files).

Subscriptions = “follow this DID”.

Speed: Instant live updates via OrbitDB CRDT replication over libp2p; canonical state via IPNS announcement.

Immutability: Every post is a permanent CID forever.

Privacy: UCAN-delegated private sharing + optional isolated private IPFS swarms.

Onboarding: Frictionless “Easy Mode” (public Helia) or one-click private-node wizard.

Discoverability: Optional opt-in public directory (IPNS-published JSON index).

Non-negotiable Guarantees

- Zero servers touch your content
 - Works fully offline
 - You control your keys and data forever
 - Feels like a 2026 native social app (not “crypto slow”)
-

2. Functional Requirements (v1.1 Expanded)

Feature	Behavior	Speed Target	New in v1.1
DID Login & Session	did:key + UCAN session + recovery kit	Instant	Threat-modelled
Post Creation	File → CID (Helia/Storacha) + optional WNFS encryption	<2 s	Schema version
Publish to Feed	Append to OrbitDB Feed + IPNS announcement + GossipSub	<1 s live	OrbitDB primary
Subscribe / Timeline	Resolve IPNS once → auto-join OrbitDB + live replication	<1 s new posts	CRDT auto-sync
Private Sharing	UCAN delegation (expiry, revocation list)	Instant	Best practices
Private Node Mode	One-click wizard (swarm.key + QR pairing)	<200 ms	Wizard + mobile
Public Directory (opt-in)	Publish DID + handle to global IPNS index	N/A	New
Notifications	OrbitDB + service-worker background	<1 s	PWA sync
Storage Monitoring	Live Storacha usage + “re-pin this” incentives	Real-time	New

3. High-Level Architecture (v1.1 Hybrid)

Layers (all client-side / P2P)

1. **Identity** — DID + UCAN (ts-odd or @ucanto) + threat-modelled key rotation
2. **Storage** — Helia + @helia/verified-fetch + Storacha + WNFS
3. **Feed Layer** — OrbitDB v3 Feed/Log (primary CRDT) + IPNS announcement only (discovery)
4. **Networking** — libp2p (GossipSub + direct dials in private swarms) + delegated routing
5. **App Shell** — Tauri 2 (desktop + mobile preview) + PWA web fallback

Data Flow (Publish)

1. Create immutable post CID
2. Append to local OrbitDB Feed (CRDT)
3. Publish new root via IPNS (announcement)
4. OrbitDB replicates instantly to connected followers
5. Service-worker keeps republishing IPNS in background

Private Swarm Bonus — direct libp2p connections → zero DHT, sub-200 ms.

4. Recommended Tech Stack (2026-Optimized)

Layer	Choice	Why (2026)
Frontend	React 19 + TypeScript + Tailwind + shadcn/ui + lucide-react	Clean, accessible, glassmorphic
Runtime	Tauri 2 (desktop + mobile) + Vite PWA	Native FS, QR pairing, resource bundling
IPFS Core	Helia v2.4+ + @helia/verified-fetch + @helia/ipns	Verified trustless resolution
Mutable Store	OrbitDB v3 (Feed/Log + encryption module)	CRDT replication, primary
Announcement	IPNS (Pubsub-enabled)	Cold-start & discoverability
Storage / Pinning	Storacha (w3up-client) + cost dashboard	UCAN-native, cheap
Auth	ts-odd or @ucanto + revocation list	Full UCAN 2026 best practices
Private Networks	Kubo embedded or swarm.key wizard	Isolated + one-click
Testing	Vitest + Playwright + libp2p chaos tools	2026 dev tools

5. SDP v1.1 — Phased Roadmap (11 weeks solo)

Phase 0: Setup + PWA (4 days)

- Turborepo monorepo + Tauri 2 + React 19 + Vite PWA plugin
- Install all deps (Helia, OrbitDB v3, ts-odd, Storacha, shadcn/ui)
- GitHub repo (MIT license, CONTRIBUTING.md, issue templates)
- Basic CI + conventional commits
- Deliverable: Empty Tauri window with dark teal theme + PWA install prompt

Phase 1: Identity & UCAN + Threat Model (1.5 weeks)

- DID creation, session, recovery kit, key rotation
- UCAN issuance, verification, expiry, revocation list
- Formal threat model document (device loss, phishing, delegation abuse, offline revocation)
- Deliverable: Auth lobby with QR link-device + threat-model PDF

Phase 1.5: Security Deep-Dive (2–3 days)

- Implement revocation list in OrbitDB
- Threat-model review workshop (self or community)
- Add security checklist to repo
- Deliverable: Security appendix + risk register v1

Phase 2: Storage + Verified Fetch (1 week)

- Helia node (browser/Node/Tauri) with verified-fetch + delegated routing
- Upload → CID + Storacha pin + cost monitoring widget
- WNFS private folders
- Deliverable: Working upload + pinning dashboard

Phase 3: Hybrid Feed System (OrbitDB v3 Primary) (2.5 weeks)

- OrbitDB Feed/Log store + encryption
- IPNS announcement only (cold discovery)
- GossipSub + direct dials in private swarms
- Schema versioning & migration utilities
- Subscription manager (IPNS resolve → OrbitDB join)
- Private-swarm optimizations
- Deliverable: End-to-end publish/subscribe with live updates + benchmarks

Phase 4: Social Features + Onboarding Wizard (2 weeks)

- Compose modal, timeline (infinite scroll, threaded replies as CIDs)
- Follow/unfollow, private sharing UI
- **Private-node onboarding wizard** (Easy Mode / Private Mode / QR pairing for mobile)
- Opt-in public directory publishing
- Deliverable: Fully functional social flow + wizard demo video

Phase 5: Polish, Testing, Audit Prep (2 weeks)

- Offline-first, edge cases (late joiners, peer churn, key rotation)
- Enhanced testing: unit, integration, Playwright E2E, libp2p chaos, benchmarks
- UI polish (shadcn/ui components matching the AI image description)
- Storacha cost + auto-pinning incentives
- Prepare audit package
- Deliverable: Milestone demo video + test report + audit-ready repo

Phase 6: Deployment, Directory, Mobile (2 weeks)

- Tauri desktop & mobile builds (GitHub Releases)
- Web version on IPFS + service-worker background sync
- Public directory service (global IPNS index)
- Documentation, private-node guide, security audit (external if budget)
- Deliverable: Shippable binaries + public directory MVP + launch post draft

Total Timeline: 11 weeks solo (realistic buffer included).

Milestone Demos: End of every phase (recorded 2-min video).

6. Data Models (Versioned IPLD + OrbitDB)

Post (immutable CID) — versioned schema

Feed Entry (OrbitDB) — { postCID, timestamp, version: "1.1" }

Directory Entry (public IPNS) — signed JSON with DID + handle + optional avatar CID

7. Major Challenges, Nuances & Mitigations (v1.1 Expanded)

Challenge	Implication	v1.1 Mitigation	Private-Node Win
Discoverability	Hard to find users	Opt-in public directory + IPNS announcement	Same
Key / device loss	Data lockout	Recovery kit + threat-modelled rotation	Local only
Browser reliability	No persistent connections	Verified-fetch + PWA service-worker	Tauri full control
Spam / abuse	Flooding feeds	UCAN expiry + revocation + nonce	Swarm.key isolation
Content availability	CIDs vanish	Storacha default + re-pin incentives	Local pinning
Onboarding friction	Users drop off	One-click wizard + Easy Mode	Seamless

Browser vs Desktop Trade-off fully addressed: web = public Helia + PWA; desktop = full private swarm.

Scalability — OrbitDB handles 10k+ followers with lazy loading + archive pointers.

8. Performance Targets (Achievable Feb–May 2026)

- Active followers: <1 s (OrbitDB CRDT)
 - Cold follow: <5 s (verified IPNS)
 - Private swarm: <200 ms
 - Publish + pin: <4 s
 - 10 000-post feed load: <3 s (lazy)
-

9. Starter Project Structure (v1.1)

```
text
cidfeed/
  -- packages/
    -- core/          # helia, orbitdb-feed, ucan, verified-fetch
    -- ui/            # shadcn/ui components, compose, timeline
    -- tauri/          # Tauri app, wizard, QR pairing
  -- docs/
    -- threat-model.md
    -- private-node-wizard.md
    -- security-checklist.md
    -- risk-register.md
  -- risk-register.md      # live document
  -- README.md
  -- turbo.json
  -- .env.example
```

10. Cost & Resources (2026)

- **Zero-cost MVP:** Pure Helia + local Kubo
- **Reliable pinning:** Storacha free tier sufficient for 100 users; paid ~\$0.02/GB-month
- **Audit:** \$3k–8k optional (community first)
- Key repos/docs: OrbitDB v3, Helia verified-fetch, Storacha UCAN, ts-odd, Tauri 2

Open-source from Day 1 → expect community contributions by Phase 3.

11. Risk Register Summary (Appendix)

1. Key loss → recovery kit + education
 2. OrbitDB sync edge cases → chaos testing
 3. Storacha cost overrun → monitoring + incentives
 4. Discoverability failure → directory + launch marketing
 5. Audit findings → Phase 5 buffer
-

This SDP v1.1 is now complete, battle-tested against 2026 realities, and ready to ship a world-class decentralized social app.

Next Steps — Let's Execute

Reply with your priority (pick one or more):

1. “Generate the full core package (TypeScript modules for OrbitDB v3 + verified-fetch + UCAN)”
2. “Output the private-node onboarding wizard flow + Tauri code”
3. “Create the threat-model template + risk-register markdown”
4. “Generate shadcn/ui component code for the main feed card + compose modal”
5. “Start the GitHub repo structure + initial commit commands + milestone demo script”
6. “Anything else — I’m ready to code Phase 0 today”

We now have the definitive plan. Let’s build CIDFeed and show the world what sovereign social looks like in 2026.

What's your first move? 