Private Apps on Web5: Local-First Decentralized Cloud Applications

Zach Kelling

Lux Industries

April 2025

Abstract

Web5 is a local-first application architecture where mutable state lives in encrypted per-user SQLite vaults, synchronized via CRDTs, and anchored to a blockchain trust layer for identity, key policy, capability grants, and audit receipts. This paper defines the architecture, explains why blockchains are excellent trust planes but poor data planes, and describes the concrete implementation using Lux Network VMs and Hanzo Base.

1 Thesis

Put trust on-chain, keep state local, sync privately, make identity portable.

Web5 is not "apps fully on-chain." Web5 is local-first apps with a shared cryptographic trust layer that gives users ownership and speed, developers a better runtime, and operators a new model for decentralized cloud.

2 The Web5 Stack

Layer	Role	Stores
Client vault	Data plane	Encrypted SQLite, CRDT log
Sync layer	Replication	Ciphertext ops, snapshots
Trust layer	Control plane	DIDs, keys, policies, anchors
Provider layer	Utility market	Relays, storage, signers

3 Why SQLite + CRDT

SQLite is an ideal unit of tenancy: durable, compact, cross-platform, single-writer per file [1]. CRDTs solve merge after independent edits [3]. Together they give a clean runtime for mobile, desktop, browser, and edge.

The practical unit: one vault per user, one per organization, one per shared workspace. Horizontal scale comes from sharding by principal, not concentrating into one shared database.

4 Where Blockchain Belongs

On-chain: identity roots, key handles, wrapping policy, rotation history, capability grants, revocations, sync checkpoint anchors, provider discovery, payments, governance.

Not on-chain: application documents, messages, notes, settings, collaborative edits.

5 Security Model

ML-KEM establishes shared secrets for key wrapping (NIST FIPS 203 [2]). Symmetric encryption (AES-256-GCM) protects vault data. Threshold policies handle recovery and delegation. Chain receipts provide verifiable policy state.

6 Product Architecture

Hanzo Base becomes the Web5 app runtime (data plane). **Lux** becomes the trust and coordination layer (control plane). The synthesis: one encrypted CRDT-backed SQLite vault per principal, decrypted locally, synced as ciphertext, anchored to Lux.

7 Firebase Replacement

Firebase	Web5
Auth	DID + capability/session gateway
Firestore	Encrypted SQLite shard
Offline sync	CRDT (local-first by default)
Storage	Encrypted blob providers
Functions	Workers on sync/anchor events
Security Rules	Capability grants + chain policy

Migration: keep the app, change the backend shape.

8 Decentralized Cloud

Cloud providers become optional and replaceable. Users own keys. Providers rent sync, storage, indexing, signing, and recovery. The protocol is small enough that many operators can

implement it, strong enough that apps can rely on it for portability and trust.

9 Conclusion

Each user, org, or app gets its own encrypted CRDT-backed SQLite vault. Vaults execute locally, sync as ciphertext, and anchor trust to Lux. Providers relay, store, recover, and index without owning the data. That is the decentralized cloud model.

References

[1] R. Hipp, "SQLite: Past, Present, and Future," *Proc. VLDB Endow.*, 2022.

[2] NIST, "Module-Lattice-Based Key-Encapsulation Mechanism Standard," FIPS 203, 2024.

[3] M. Shapiro et al., "Conflict-Free Replicated Data Types," *SSS 2011*, LNCS 6976.

[4] M. Kleppmann et al., "Local-First Software," *Onward! 2019*.