

Decentralized Platforms: Comprehensive Comparative Analysis

Section 7 – Platform Deep Profile: Secure Scuttlebutt (SSB)

Version 3.0 | 16 August 2026 | Segment 07

7.1 Overview

Attribute	Value
Full name	Secure Scuttlebutt (SSB)
Type	Local-first, offline-capable peer-to-peer social protocol
Primary identity	Ed25519 keypair (public key = feed ID)
Core primitive	Signed append-only feed (sigchain) per identity
Official site	https://scuttlebutt.nz
Protocol guide	https://ssbc.github.io/scuttlebutt-protocol-guide
Primary design goal	Unforgeable, offline-friendly social data with gossip replication

SSB is local-first: each user owns an append-only log of signed messages. Peers gossip the feeds they care about. No global consensus and no required always-on server for core operation. Pubs/Rooms improve connectivity.

7.2 History and Design Philosophy

Created by Dominic Tarr and collaborators. Rejects global singletons. Every identity has its own signed append-only log; replication follows the social graph; works offline and across intermittent links. Principles: unforgeability, local-first, gossip, subjectivity (no single global state).

7.3 Architecture

[illegible]

Identity/Feed = Ed25519 keypair. Message = signed, sequenced, hash-linked. Peer = any device storing and gossiping feeds. Pub/Room = always-on connectivity helpers. Replication is efficient because feeds are linear (exchange latest sequence numbers).

7.4 Identity, Feeds and Messages

Public key is the stable identifier; no global name registry. Feed is append-only; only key holder can append; history immutable. Classic message includes previous hash, author, sequence, timestamp, content (schemaless, often typed), and signature. Private messages can be encrypted to recipients. Metafeeds later allow multiple subfeeds per identity.

7.5 Clients and Current Status (mid-2026)

Client	Platform	Notes
Manyverse	Android+	Major mobile client; maintainer transition ~2024
Patchwork	Desktop	Classic; community forks active (e.g. Poncho Wonky)
Patchbay / Patchfox	Desktop / Browser	Power-user and extension clients
Others	Various	Tilde Friends, ait-ssb, tools; multi-language impls

Core protocol stable. Flagship clients have seen maintainer transitions; community forks keep the network usable. Modest scale compared with ActivityPub/Nostr; strength is resilient offline social graphs. Metafeeds and newer formats improve partial replication.

7.6 Strengths and Limitations

Strengths: excellent offline behaviour; strong feed integrity; simple replication; no central server required; fit for local-first communities; multiple clients historically and via forks.

Limitations: multi-device/key-recovery UX hard; true deletion impossible; scaling and discovery non-trivial; client maintenance uneven; smaller ecosystem; onboarding can feel opaque.

7.7 Comparative Position

Dimension	vs Nostr	vs Peergos	vs ActivityPub	vs Syncthing
Offline capability	Stronger	Stronger	Stronger	Comparable
Identity	Keypair (similar)	Keypair	Domain/Actor	Device cert
Data model	Append-only feed	Encrypted FS	Activities	File sync
Default social model	Follow + gossip	Capability sharing	Federation	None
Deletion / edit	Impossible	Possible	Server-dep.	File-level
Primary strength	Offline social	Private storage	Network effect	Reliable sync

SSB is the strongest pure local-first social design in this survey. It trades global scale and easy multi-device convenience for integrity, offline operation, and freedom from always-on servers.

7.8 Key Primary Sources

- <https://scuttlebutt.nz> • <https://ssbc.github.io/scuttlebutt-protocol-guide> • <https://spec.scuttlebutt.nz>
- <https://github.com/ssbc> • Handbook and application lists on scuttlebutt.nz

- Community client repositories (Manyverse, Patchwork forks, etc.)

End of Section 7 – SSB • Next: Section 8 – Syncthing Deep Profile