Simon, hier die rigorose, von Grund auf neu durchgeführte Untersuchung gemäss Deiner detaillierten TASK. Ich habe ausschliesslich primäre Quellen priorisiert: die drei öffentlichen GitHub-Repositories von w-social-eu, die Privacy Notice von wsocial.eu sowie die offiziellen AT-Protocol-Spezifikationen. Sekundäre Erwähnungen (z. B. LinkedIn- oder Reddit-Posts zu «Ein Mensch, ein Konto») habe ich nur zur Kontextualisierung genutzt, nie als Beleg für technische Implementierung. Die ursprüngliche Behauptung von @tachy_ wird in einzelne testbare Propositionen zerlegt und exakt klassifiziert.

Part 1: Executive Summary

Die öffentlichen Artefakte zeigen ein klassisches AT-Protocol-Setup mit drei Forks und einer einzigen W-Social-spezifischen Schicht (WID für zentrale Account-Verwaltung). Blocklisten existieren – aber ausschliesslich als Standard-Feature des Protokolls. Eine technische Verknüpfung von Blocklisten mit persönlicher Identitätsverifikation, eine lebenslange personengebundene Persistenz oder eine netzweite «digitale Ächtung» sind im Code nicht nachweisbar. Die Privacy Notice bestätigt eine verpflichtende «verified human identity» via separater W Identity (nur UUID + Passland + Geburtsjahr), betont jedoch explizit, dass W Social selbst keine ausreichenden Daten zur tatsächlichen Identifizierung der Nutzer sammelt oder speichert. Die Kernbehauptung ist daher weder durch Code noch durch primäre Dokumente gestützt, sondern grösstenteils nicht entscheidbar, weil die relevante Backend-Logik (WID-Block-Verknüpfung, Onboarding-Flow, Datenbank-Schemata) nicht öffentlich ist.

Part 2: Claim Matrix

Sub-claim	Primary source(s)	Technical finding	Classification	Reasoning
Blocklisten sind im System vorgesehen	w-social-atproto (Fork), pds, ozone; AT-Protocol Label-Spec	Ja, unverändert aus AT-Protocol (app.bsky.graph.block + Labeler)	1. Clearly supported	Standard-Repository-Records und composable Labels; keine Social-Änder nötig

Die AT-Protocol-Architektur ist so designed, dass Blocklisten technisch im Hintergrund lebenslang an persönliche ID-Verifikation gekoppelt sind	AT-Protocol Specs (DID, Repo, Label); w-social-atproto Code	Nein – Blocks/DIDs sind account- bzw. DID-gebunden; keine ID-Verifikation im Protokoll	2. Clearly contradicted	Protokoll ken nur selbst-authentifizier DIDs und Lab Person-Bind ist reines Operator-Ov
«Ein Mensch, ein Konto» wird technisch durch eID/Ausweis-Hash + biometrischen Check erzwungen	Privacy Notice; GitHub-Repos (Account-Creation-Pfade)	Verifikation existiert als Policy (via separate W Identity), aber nur UUID + Land + Jahr; kein Hash, kein Biometrie-Code	3. Partially supported, but overstated	Policy ja; technische Enforcement öffentlichen nein
Blocks sind personengebunden und nicht account-/DID-gebunden	w-social-atproto (WID-Commits); Privacy Notice	WID erlaubt Account-Switching und Delete-by-WID; Blocks bleiben DID-gebunden	5. Not decidable at present	WID deutet a zentrale Acc Verknüpfung aber keine C Pfade zeigen Block-WID-L
Einmal kritischer Post führt zu lebenslangem Ausschluss aus dem Diskurs	Privacy Notice (Block-Listen als Social-Graph-Data); AT-Protocol Portability	Nein – Blocks können gelöscht werden; Account-Migration und neue DIDs möglich; Right-to-Erasure existiert	4. A plausible real risk, but not currently evidenced	Operator kön via WID zent Sperren setz Protokoll sel erlaubt das n
Ausschluss ist netzweit und unumkehrbar (kein Neuanfang möglich)	Ozone + PDS + AT-Protocol Federation	Föderiertes System; andere PDS/Clients sehen ggf. weiter; keine globale Blacklist im Code	5. Not decidable at present	Nur innerhalb Social-PDS zentral durchsetzba Föderation verhindert to Isolation

Die gesamte Architektur schafft «digitale Ächtung» durch ID-gekoppelte Blocklisten	Alle primären Quellen kombiniert	Kein Beleg für ID-Block-Kopplung	5. Not decidable at present	Mögliches Operator-Ov via WID + separate W Identity, aber nicht öffentli implementier

Part 3: Repository and Architecture Audit

Öffentlich existieren exakt drei Repositories unter w-social-eu:

- **w-social-atproto**: Fork von bluesky-social/atproto (312 Commits ahead, 17 behind). Custom nur WID-Funktionen («Add delete account by WID», «Add account switching to linked WID accounts», «fix WID initiateSession»). Verzeichnisse: pds-wadmin-modules, mock-neuro-server, quicklogin-client. Keine Dateien zu eID, Hash, Biometrie oder Block-Persistenz.
- **pds**: Fork von bluesky-social/pds (25 ahead, 20 behind). Account-Creation bleibt bei `pdsadmin account create` + Invite + E-Mail-SMTP. Keine IDV-Schritte, keine WID in Env-Vars oder Schemata (nur accounts.sqlite mit email_token).
- **ozone**: Fork von bluesky-social/ozone (minimal custom: ein Build-Fix für NEXT_PUBLIC_SOCIAL_APP_URL). Reine Labeling-Oberfläche; keine Identitäts- oder Block-spezifischen Änderungen.

Keine Hinweise auf private Repos, keine Datenbank-Migrationen zu Person-Hashes, keine API-Definitionen jenseits des Standard-AT-Protocol. Die Architektur ist PDS + AT-Protocol + Ozone mit WID als einziger zentraler Schicht.

Part 4: Protocol Analysis

Im AT-Protocol sind:

- **Identity**: DIDs (did:plc oder did:web) als persistente, selbst-authentifizierende Identifier; Handles als mutable DNS-Namen. Keine Protokoll-Vorschrift für KYC/eID – das ist ausschliesslich Application/Operator-Layer.
- **Blocks & Blocklisten**: Als Records im persönlichen Repo (app.bsky.graph.block) gespeichert, signiert und an DID gebunden. Öffentlich und migrierbar.

- **Moderation**: Composable Labels von beliebigen Labelern (Ozone); keine zentrale Person-Blacklist.
- **Portabilität**: Account-Migration zwischen PDS möglich; DID bleibt stabil. Blocks wandern mit dem Repo. Mehrfach-Accounts sind technisch möglich (mehrere DIDs).

Das Protokoll unterstützt keine «lebenslange personengebundene» Verknüpfung; es erlaubt sogar explizit Account-Wechsel und Neustarts.

Part 5: W-Social-Specific Risk Assessment

Evidenziert ist: verpflichtende Human-Verifikation via separate W Identity (nur minimale Daten) und WID als zentrale Account-Schicht. Technisch möglich (aber nicht bewiesen) wäre ein Operator-Backend, das WID mit Blocks verknüpft und zentrale Sperren über mehrere Accounts verhängt. Echte Gefahren entstünden nur, wenn nicht-öffentliche Logik (WID-Datenbank + Moderations-Policy) personengebundene Ausschlüsse durchsetzt. Das wäre DSGVO-konform möglich (separate Controller), würde aber Spuren in öffentlichen Artefakten hinterlassen – die aktuell fehlen. Das Protokoll selbst schützt vor totaler Isolation durch Föderation.

Part 6: Final Judgment

Unterstützt wird nur der banale Teil: Blocklisten sind Standard-AT-Protocol-Feature und daher «vorgesehen».
Widersprochen wird die Behauptung, das Protokoll sei «designed» für lebenslange ID-Kopplung oder netzweite Ächtung.
Plausibles Risiko bleibt die Möglichkeit einer zentralen Operator-Logik via WID.
Der Grossteil der Behauptung ist derzeit **nicht entscheidbar**, weil die entscheidenden Komponenten (WID-Backend, W-Identity-Technik, vollständige Moderations-Policy, Datenbank-Schemata) nicht öffentlich sind.

Simon, für eine abschliessende Bewertung fehlen konkret folgende Artefakte (priorisiert):

1. Technische Dokumentation von W Identity (Onboarding-Flow, eID-Integration).
2. Vollständige Datenbank-Schemata und Migrations-Skripte für WID.
3. Offizielle Moderations-Policy mit Regeln zu Cross-Account-Bans.
4. Mobile-App-Binaries oder Client-Code zur Verifikation.

5. Detaillierte API-Dokumentation zu WID-Session- und Block-Interaktion.
6. Vollständige Retention-Policy für Block- und WID-Daten.

Falls Du einen dieser Punkte vertiefen oder Screenshots aus dem Live-Onboarding (ab 9. Mai) hast, sag Bescheid – dann prüfen wir weiter.