PWOS at a glance

Protocol Wealth Operating System — v0.1 fact sheet

2026-05-19 snapshot ▪ v0.1 internal draft

Protocol Wealth Operating System (PWOS) is the integrated advisor IDE, client portal, and compliance engine Protocol Wealth built for its own SEC-registered investment advisory practice. The same system is now being prepared for licensing to other RIAs on a retainer + profit-interest model.

PWOS is architecturally an *integration wrapper* around open-source primitives. The open primitives (`pwos-core` Apache 2.0, `nexus-core` Apache 2.0) are inspectable without an NDA. The licensed wrapper is what partner firms actually pay for: institutional integration logic accumulated under live SEC examination.

The six structural layers

Layer	What it does	Where it lives
Zero Data Retention (ZDR)	Anthropic ZDR posture, enforced contractually at the API workspace level. Inputs and outputs are not retained beyond the duration of a single inference request; content is contractually excluded from training, fine-tuning, and model improvement. Inference runs on US-based infrastructure.	All AI-using surfaces; workspace ID on file; effective April 21, 2026
Schema-level PII tagging	Every client-data field is tagged `pii.high`, `pii.medium`, or `pii.low` at ingestion. The `pii.high` bucket (SSN, full account numbers, wallet seed phrases, biometrics, authentication artifacts) is structurally stripped from every LLM-bound payload by middleware — not by developer discipline.	`pw-api` + downstream BFFs; canonical map at `pw-api/src/lib/pii-tags.ts`; CI drift-check enforces alignment across repos

Layer	What it does	Where it lives
Independent PII egress canary	A second-layer guard runs at every Anthropic SDK call site. The pattern set is *deliberately* re-implemented byte-identical across surfaces so the two layers cannot share a single bug. Any residual PII raises `PIIEgressCanaryError`, aborts the call, and surfaces as a structured `pii_blocked` event with Cloud Logging alerts.	`pw-os-v2` + `pw-portal-v2`; `pw-api` wave queued
WORM audit retention	A 7-year retention-locked Google Cloud Storage bucket (`gs://pwllc-audit-archive`) mirrors every row written to the canonical `audit_log` table. Bucket-level retention lock means no PW operator — including those with full GCP organization-admin — can delete or modify an object within the window. SHA-256 content hash per object; source git revision in metadata.	Per SEC Rule 17a-4(b)(4) + 17a-4(f)(2)(ii); lock applied 2026-05-02
Sentinel-row reconciliation	Reconciliation of failed mirror writes cannot `UPDATE` the audit_log (the table is immutable at the database trigger level). Instead, a daily cron emits a *new* sentinel row referencing the failed row's ID. The pattern is architectural and applies to every immutable evidence table operated: audit_log, KYC verifications, signed-document archive.	`audit_log`, `kyc_sessions`, `signed_document_archive`

Canonical webhook receiver	Every vendor callback flows through the same six stages: verify (HMAC signature), dedup, parse, process, audit, dead-letter. The pattern is shared across vendor integrations; cross-component tracking is mechanical, not per-route judgment.	All vendor integrations (Veriff, Anvil, Quiltt, Scorechain, custodian APIs)

What's in production today (May 2026)

Component	Status	Notes
Component 1 — Portal authentication (passkey + Turnkey)	Live	Per-client passkey; Turnkey RP ID `pwportal.app`
Component 2 — KYC (Veriff identity + Scorechain AML two-layer)	Live	Webhook receiver primitive; sentinel-row reconciliation on `kyc_sessions`
Component 3 — Risk tolerance (MIT/Grable FRTS-13 + PW overlay)	Live	1-100 composite score; advisor-override HITL on cross-bucket changes
Component 4 — E-signature (Anvil 4-document envelope)	Live	IAA + Form ADV 2A + Privacy Notice + IPS; annual ADV re-delivery cron per Rule 204-3
Component 5 — Custodian-data (Quiltt primary + Schwab direct-API)	Scope landed; implementation next iteration	Quiltt is existing substrate; Schwab API is scaffolding (production deferred v1.5)
Component 6 — Onboarding status dashboard	Scope landed; implementation queued	Cross-component aggregator surface

The structural-over-disciplinary frame

Where a control can be enforced by code, PWOS enforces it by code. The PII egress guard does not rely on every developer remembering to scrub a payload; middleware enforces it and CI lints any LLM SDK call routed outside the middleware. The canonical action catalog is regex-enforced. The model alias map is ESLint-enforced. The PII tag map is drift-check-enforced across repos.

The credential-layer thesis — *PII never sent to any LLM* — is enforced by code that runs regardless of caller behavior, not by caller behavior itself.

How to verify any of this

Five concrete pathways are documented in the public security posture (`docs/compliance/security-posture-publi` § "How to verify any of this"):

1. **Confirm ZDR with Anthropic directly** — workspace ID and effective date attestable on request to credentialed reviewers under NDA.
2. **Review the substrate ADRs** — Architecture Decision Records covering PII tagging, WORM audit mirror, webhook receiver primitive, KYC state machine, signed-document state machine. Available under NDA for qualified institutional prospects.
3. **Sample anonymized audit log + WORM mirror rows** demonstrating action-verb structure, principal chain capture, WORM round-trip.
4. **Vendor due-diligence bundle** — SOC 2 reports, DPAs, vendor risk assessments under NDA.
5. **Written attestation on CCO letterhead** — formal attestation of current AI posture (subprocessor list, ZDR status, inference region, PII controls in effect at time of request) available on qualified request.

Open source

- `pwos-core` — Apache 2.0-licensed; compliance audit-logging patterns + PII redaction toolkit. Reference code, not deployed.
- `nexus-core` — Apache 2.0 defensive licensing; MCP server foundation; ~243 financial-data tools surfaced; live at nexusmcp.site.

The intent is not marketing surface; it is to demonstrate, in code, the structural-over-disciplinary patterns described above.

Cross-references

- **Public security posture (full):** `docs/compliance/security-posture-public-2026-05-19.md`
- **Architecture decisions:** `architecture/decisions/` (ADRs under `shared/`)
- **Form ADV Part 2A:** adviserinfo.sec.gov/firm/brochure/335298
- **Subprocessor list:** `protocolwealthllc.com/subprocessors`

Protocol Wealth, LLC · SEC-Registered Investment Adviser · CRD #335298 This document is an internal draft of public-facing collateral. Engineering substrate transparency · aggregate substrate material · not investment advice · not advisory performance.