# Open-source RIA tooling — a reading list

12 projects worth knowing if you want to audit your own stack

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

This is a curated starter map. The intent is not to recommend any of these for use in a specific advisory practice; it is to make the *shape* of the open-source RIA infrastructure landscape visible. Use it to evaluate which patterns your existing closed-source vendors are quietly built on, and where the lock-in actually lives.

The list is organized into three tiers: foundational infrastructure, RIA-adjacent primitives, and protocol-layer infrastructure that's about to matter.

---

## Tier 1 · Foundational infrastructure

Software your vendors are already running, whether they tell you or not.

**1. PostgreSQL** (postgresql.org) The relational database underneath roughly every serious RIA back-office tool. PostgreSQL License (BSD-style). Row-level security, write-ahead logging, point-in-time recovery, and the audit-logging pattern most compliance tooling is built on. Worth understanding because most "AI-powered" advisor tools are 90% Postgres + 10% AI veneer.

**2. OpenTelemetry** (opentelemetry.io) The open standard for distributed-tracing, metrics, and logs. Apache 2.0. The substrate for the `trace_id` and `request_id` fields that should appear in any vendor's audit log. If a vendor cannot produce a trace_id when you ask, they aren't running OpenTelemetry (or anything equivalent), and your incident response is structurally weaker.

**3. SQLite** (sqlite.org) The most-deployed database in the world. Public domain. Relevant because every major financial-data export format eventually becomes a SQLite question, and because edge AI is increasingly running on local SQLite stores.

## Tier 2 · RIA-adjacent OSS primitives

Open-source pieces that are directly load-bearing for RIA tooling.

**4. `pwos-core`** (Apache 2.0) · Protocol Wealth Compliance audit-logging patterns and a PII redaction toolkit. Public reference code; not deployed as a production service. The patterns here — canonical action verbs, principal chain, sentinel-row reconciliation — are demonstration material for the same shape any RIA's audit log should take. *Disclosure: this is Protocol Wealth's repo.*

**5. `nexus-core`** (Apache 2.0) · Protocol Wealth An open-source Model Context Protocol (MCP) server

foundation. Live at nexusmcp.site with ~243 financial-data tools surfaced. Useful for understanding what an MCP integration substrate looks like in production. *Disclosure: this is Protocol Wealth's repo.*

**6. dbt-core** (github.com/dbt-labs/dbt-core) Apache 2.0 data-transformation framework. The commercial parallel for the OS-licensing thesis: dbt-core is open; dbt Cloud is the paid orchestration / observability / governance layer. Worth studying as the canonical example of how an open primitive + a licensed wrapper produces a defensible commercial business.

**7. Supabase** (github.com/supabase/supabase) Apache 2.0 Postgres + Auth + Realtime + Storage primitives, with a proprietary cloud platform on top. Another commercial parallel for the OSS-wrapper pattern. The auth and row-level-security implementations are particularly worth reading.

**8. Tailscale** (tailscale.com · WireGuard underneath) Open-source WireGuard primitives with a proprietary control plane and admin console. The model for "use the open protocol, charge for the institutional logic." Relevant if you've ever wondered how an RIA would build its own zero-trust networking without sending traffic to a third-party SaaS.

## Tier 3 · Protocol-layer infrastructure that's about to matter

If you read nothing else on this list, read these.

**9. Model Context Protocol (MCP)** (modelcontextprotocol.io) The open protocol (originated by Anthropic) for LLMs to interact with external tools and data sources. Apache 2.0. The substrate underneath every "AI agent" announcement for the next 24 months. If you understand MCP, you understand which vendors are building on a real protocol versus which are shipping proprietary wrappers around closed APIs.

**10. SLSA — Supply-chain Levels for Software Artifacts** (slsa.dev) A framework for software-supply-chain security. Not a tool, a posture. Relevant because the next round of cybersecurity examination questions will be supply-chain questions, and SLSA gives you a vocabulary.

**11. SPIFFE / SPIRE** (spiffe.io) The open standard for workload identity. The substrate underneath any production-grade "who is calling whom" enforcement in a multi-service system. Worth understanding because vendor-side IAM is increasingly the access-control story.

**12. Sigstore** (sigstore.dev) Open-source software-signing infrastructure. Apache 2.0. The shape of "did this code actually come from where it says it came from." Relevant for any RIA that runs custom integrations or relies on automated CI/CD pipelines.

---

## How to use this list

- **Pick two.** Reading all twelve is a graduate degree. Reading two is a useful afternoon. Start with MCP and dbt-core; everything else extends from those.
- **Run the diff.** For each closed-source tool in your stack, ask: which of these open primitives is the vendor building on, and what does that imply about my exit cost?
- **Ask your vendors.** "Are you running OpenTelemetry? Postgres row-level security? MCP-conformant tools?" Their answers reveal whether they're building on protocols or building moats around cus-

tomers.

---

*Contributed by Protocol Wealth, LLC (CRD #335298). This is a curated educational list and not a recommendation to adopt any specific project. Disclosed:* `pwos-core` *and* `nexus-core` *are repositories of the contributor. Engineering substrate transparency posture · aggregate substrate material · not investment advice. Suggestions or corrections welcome to nick@protocolwealthllc.com.*