IP PROTECTION & DUAL-LICENSING STORY

OdbDesign

A documented chain of title, a CLA enforced on every pull request, and clean-room engineering — all verifiable from public evidence.

Artificial Intelligence Agency LLC · November 2026

How the OdbDesign codebase is made safe for closed-source commercial licensing — and why that license is fully self-contained, separate from the FOSS codebase, and verifiable from public evidence.

100%	281 / 281	16	0
owner-authored commercial tree	tests green after restoration	permissively-licensed dependencies	copyleft dependencies anywhere

The Thesis

OdbDesign ships under a **dual-license model**: the public repository remains free and open source under **AGPL-3.0**, forever — and a **fully self-contained commercial license**, granted solely by the copyright owner, authorizes closed-source use. Every action described below was taken so that the commercial grant derives from **one clean chain of title** with no dependence on anyone else's open-source contributions. The result: a prospective licensee's legal team can verify safety from public evidence alone.

Background: What Dual Licensing Is

Most software carries one license. **Dual-licensed software carries two**, and you choose which one governs your use:

- The **open-source license** (here AGPL-3.0) lets anyone use, study, and modify the code for free — with one core condition, *copyleft*: if you distribute the software, or run it as a network service for others, you must make your integration's source code available under the same terms. For an open project that is a fair trade; for a company whose product is closed-source, it is a non-starter — which is exactly why AGPL sits on many corporate deny-lists.
- The **commercial license** exists for that case: a standard fee-bearing license from the copyright owner that permits closed-source use and redistribution with **no copyleft obligations at all**. You keep your code, your product, and your IP entirely private.

The two never mix in one product — pick a lane per project, same as Qt. And to be clear about what this is: **not a novel legal experiment**. Dual licensing is a decades-old, mainstream structure. If you have used a car infotainment system (Qt), an appliance with an embedded database (MySQL), or enterprise software that produces PDFs (iText), you have already shipped or consumed products built on exactly this model. The rest of this paper shows the specific steps OdbDesign took so its commercial license is as clean as those household names' — because dual licensing only works if the copyright owner can actually grant the commercial side, which requires owning or controlling the rights to every line of the code.

The Model

Aspect	FOSS edition	Commercial edition
License	AGPL-3.0 (public repository, unchanged forever)	Fee-bearing, closed-source, per customer / OEM
Source code	Public repository	Binary-only delivery
Use trigger	Free for internal use; copyleft on distribution or network service	Exactly the use AGPL forbids: shipping in a closed product
Enforcement	None needed — the license is the grant	Offline license verification + per-customer watermarking (private pipeline)
Support	Community	Maintenance, response-time tiers, custom engineering

Two editions, one grantor: every line of the licensed tree is owner-authored, CLA-covered, or permissively-licensed third-party code.

The public repository contains **zero license-enforcement code** — an AGPL user could strip any check anyway, so the commercial model rests on copyright and contract, with technical enforcement (license files, watermarked builds) delivered from a separate private pipeline.

What Was Done — The Evidence Trail

All public, all verifiable:

1. **Dual-licensing framework published.** COMMERCIAL-LICENSE.md defines what the commercial license grants (closed-source OEM/SDK/server use) and what it does not grant (no ODB++ format or trademark rights — those remain with Siemens EDA; no source-code rights). Price-free by design. (PR #610, merged 2026-11-11.)
2. **Contributor License Agreement introduced — and enforced at the door.** .github/CLA.md is an Apache-style Individual + Corporate CLA whose grant explicitly includes commercial relicensing rights, so any future contribution strengthens rather than dilutes the chain of title. CLA Assistant is live: every pull request — the maintainer's own included — is gated by the license/cla check; merges are blocked until signed. (Live since 2026-11-11; first PR gated and signed: #623.)
3. **Chain-of-title audit.** A full git-history review identified the complete set of non-owner copyright interest: three code changes from two community contributors (a drill-tools parser, a matrix color feature, and a two-line fix). Everything else in the history is owner-authored or bot-generated. The ownership surface to clean was small, known, and fully enumerated.
4. **Excision of un-relicensed contributions. (PR #623, merged 2026-11-11)** The two features were removed from the tree with a full public rationale. The affected protobuf wire-field numbers were reserved so they can never silently return under different meaning. The removed code remains in the public AGPL history with credit — no contributor right is touched; it simply no longer feeds

commercial builds.

5. **Clean-room restoration of the functionality. (PR #624)** Both capabilities were reimplemented from scratch, authored exclusively from the ODB++ Specification (drill-tools grammar pp. 183–186; matrix COLOR p. 64) and sample design data — under a written protocol that prohibited reading the removed implementations. The replacements use brand-new wire contracts (new field numbers; the old ones stay reserved) and are proven by behavioral tests: real-design parses, save→re-parse round-trips, spec-legality negative cases — 281/281 tests green.
6. **Private commercial build pipeline.** A separate private repository superbuilds the public tree and adds what the FOSS edition deliberately lacks: offline Ed25519 license verification (evaluation licenses expire and can be loopback-locked), per-customer binary watermarking with stripped symbols, and packaging with SBOM and third-party notices. Nothing in the public repo depends on it, and nothing in it contaminates the public repo.
7. **Third-party dependency hygiene.** Every linked library (16 vcpkg ports) is permissively licensed — BSD, Zlib, Apache-2.0 families (libarchive, zlib, Crow, Protobuf, gRPC). No copyleft dependency exists anywhere in the build. THIRD_PARTY_NOTICES.txt is generated mechanically from the installed ports and refreshed per delivery.

Why the Closed-Source License Is Fully Self-Contained

1. **Sole grantor.** All commercial rights flow from one copyright holder. No committee, no contributor aggregation, no relicensing consent needed from anyone else.
2. **Every line in the licensed tree is one of exactly three things:** owner-authored; covered by a signed CLA; or permissively licensed third-party code with notices attached. The excision removed the only historical exception, and the clean-room replacement restored that functionality as owner authorship.
3. **The FOSS edition is unchanged.** Nothing restricts the public AGPL repository — same code visibility, same community rights. The two editions coexist exactly as designed.
4. **Enforcement travels with the binaries, not the repo:** licensed builds verify offline, are watermarked per customer, and carry their own compliance bundle (SBOM + notices).
5. **It is all public evidence.** PRs #610, #623, #624, the CLA, and the notices file let any licensee’s counsel audit the chain without an NDA.

A Model Proven at Scale

OdbDesign did not invent this structure — it adopted the most battle-tested licensing model in commercial open source:

• **Qt.** The GUI framework shipped in millions of vehicles, industrial HMI systems, and medical devices — GPL/LGPL (Community edition) plus commercial. Qt's own licensing page sells the commercial edition to OEMs on "retaining your IPR", and expressly prohibits mixing editions in one product: the same closed-source-product commercial-license trigger OdbDesign uses. For a hardware OEM, the closest precedent imaginable.

• **MySQL.** The database embedded in countless commercial appliances runs on the identical dual-license logic; Oracle's own OEM/ISV/VAR page states that embedding MySQL in a closed-source product requires the commercial license, freeing the OEM from GPL source disclosure. The model was MySQL AB's engine to a roughly one-billion-dollar acquisition.

• **iText PDF.** The closest license match: AGPLv3 plus commercial, exactly OdbDesign's pair. iText demonstrates both halves of the story enterprises care about — AGPL use is genuinely free if you meet its disclosure terms, and the commercial license exists precisely for those who cannot ("if you're open, we're open"). It also mirrors OdbDesign's shape: a single developer's open-source library grown into a global commercial SDK business on commercial-license revenue.

• **Elasticsearch and Redis — moving toward, not away.** In 2024 Elasticsearch **added** AGPLv3 as a licensing option, and Redis 8 (2025) did the same: two of the most-watched relocations in open-source business history both landed on AGPL as the OSI-accepted leg of a dual offering. AGPL plus commercial is a live, credible, current model — not a legacy quirk.

Further Reading

If you would like more information, these two links cover both ends of the spectrum — the license text itself, and a friendly plain-language walkthrough of what it means in practice:

• The GNU Affero General Public License v3.0 — full license text (gnu.org)

• AGPLv3 explained in plain language (choosealicense.com — a friendly, neutral guide to what the license permits and requires)

Evidence	Where to look
Commercial license overview	COMMERCIAL-LICENSE.md · repository root
Contributor License Agreement	.github/CLA.md · repository root
Third-party notices	THIRD_PARTY_NOTICES.txt · repository root
Dual-licensing framework, CLA landing	PR #610 (merged 2026-11-11)
Excision of community contributions	PR #623 (merged 2026-11-11)

Evidence	Where to look
Clean-room restoration	PR #624
Public repository	github.com/nam20485/OdbDesign

The complete diligence trail is public: no NDA required to verify any claim in this paper.

> ***For a prospective licensee, in one paragraph.*** *Using OdbDesign internally under AGPL costs nothing and always will. The moment your product ships with OdbDesign inside — or serves third parties over a network — you need the commercial license, and the grant you receive is clean: one rightsholder, a contributor-tested CLA, a documented excision-and-clean-room record, permissive-only dependencies, and per-customer licensed binaries with SBOM and watermark. The diligence trail is public. Welcome to the same licensing structure that ships Qt in your car.*