

Proof Is the Product

Routing, repairing, and verifying legacy software from the binary — a self-evolving compatibility engine that perceives, diagnoses, fixes, verifies, and learns, where the industry only looks up.

Luke Kist
kist × Perslis

2026-07-11 **PILOT-READY**

Provenance. [MEASURED] = observed from a real run or a passing test on this machine. [PROTOTYPE] = built to a point, boundary stated. [ILLUSTRATIVE] = schematic, not a result. Every claim below carries its tag; nothing is labelled above what it has earned.

Abstract

Every mainstream compatibility tool examined here — Lutris, Bottles, GOG, protonfixes, and winetricks — starts from a human-authored recipe or known identifier. The kist × Perslis loop instead reads the binary, routes it to a runtime tier, diagnoses a failure, compiles a shell-free remediation plan, verifies the rendered result, and records the outcome as evidence for the next decision. This report separates measured results from prototypes and illustrative material at every evidence-bearing surface. [MEASURED]

Source. Synthesis of T-core, D-loop, H-autofix, H-ff7, and L-vm; each underlying fact is separately sourced in this registry.

1 The wedge

The entire game-preservation and compatibility industry — Lutris, Bottles, GOG, Proton/protonfixes, winetricks — is a set of static, human-curated recipe databases. Their only automation is looking up a pre-authored fix by a known identifier. None of them routes from the binary, diagnoses an unseen failure, verifies that a fix actually worked, or learns the mapping. That detect→route→diagnose→fix→verify→learn loop is the wedge. Their databases are our seed corpus. [MEASURED]

Source. Mechanism-level teardown of umu-protonfixes (key→module lookup on Steam AppID/UMU-ID) and winetricks (declarative w_metadata + imperative load_verbs); cross-compare matrix (2026-07-07).

2 The mechanism: one loop, six stages

The system is a Darwin–Gödel-style engine: it does not merely apply fixes; it accumulates which fix worked for which failure signature and selects the next route on measured fitness. The six-stage loop is the product. [MEASURED]

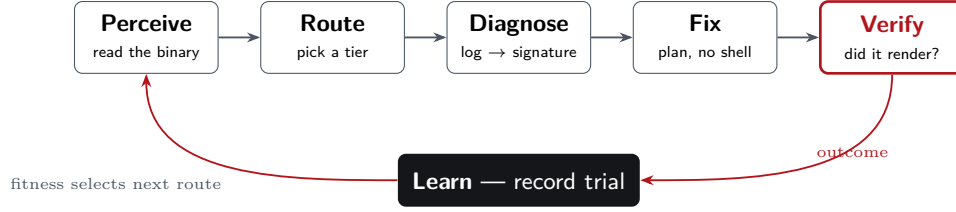


Fig 1. The loop. Verify is the load-bearing stage: a fix counts only when a captured frame is non-black by luminance, not when a process exits zero. Every outcome becomes a fitness-weighted trial in a self-evolving archive. [MEASURED]

Stage	What it does	Implementation
Perceive	Reconstruct a stripped binary’s architecture from the binary itself: header class (MZ/PE/NE/LE), import dependency tree by OS subsystem, sections, resources, and a capstone call-graph trace-back to key OS APIs.	map + compatScanImports + detectCompatSymptoms
Route	From the fingerprint alone, pick a runtime tier: NE→unwrap, DOS/LE→DOSBox-X, Glide→Perslis accelerated legacy graphics, legacy-D3D→wine+dgVoodoo2, DDraw-only→wine+cnc-ddraw, modern-D3D→wine+DXVK.	compat-route (compatRouteTier)
Diagnose	Turn a crash log + display probe into a failure signature, then match it to a remediation rule.	compat-diagnose
Fix	Compile a rule into a runnable, shell-quoted, exec+argv plan (no shell) with resolved, guarded shim assets.	compat-apply / compat-autofix
Verify	Re-launch supervised; a fix counts only if the display actually maps and renders — checked by luminance on a captured frame, not by an exit code.	liveWineRerun + compatProbeDisplay + compatImageIsBlack
Learn	Every fix application is an empirical trial; Laplace-smoothed fitness drives selection; failed rules are kept as stepping-stones and can breed variants (parent/generation lineage).	recordCompatTrial + compatFitness + proposeCompatCandidate

Source. Each stage maps to a live CLI command; the full loop is exercised by compat-autofix –execute (live-proven on Heavy Gear) and 70 green compat tests.

3 What is actually measured

Zero-human repair, end to end. compat-autofix –execute took a broken Heavy Gear runtime, read glide2x.dll not found from its own launch, deployed the dgVoodoo2 glide2x wrapper, re-launched, verified the display, and recorded a worked trial — with no human in the loop. [MEASURED]

Boundary: This is the wine tier. The 3D-mission wall of the same game is NOT solved under wine — it needs the VM tier (see limitations).

Source. Live run of compat-autofix –execute on the Heavy Gear runtime. The seed knowledge came from the Heavy Gear wine campaign (/Volumes/PCGAMER3_15/HGEAR).

3.1 The hardware-3D wall, proven three ways

For Final Fantasy VII, the capstone call-graph trace-back independently re-derived the initialisation path: `sub_402275` reads the registry keys and calls `sub_41CD14`, which calls `DirectDrawCreate`. The same hardware-3D dependency was confirmed by three independent readings of the binary — strings, the import table, and the reconstructed call graph. [MEASURED] Perception is grounded in the binary, not in a title lookup. The same conclusion reached three independent ways is what makes a route trustworthy.

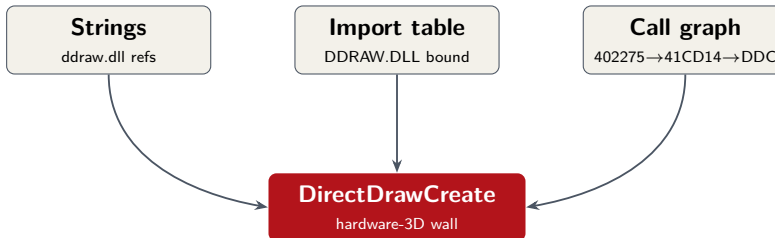


Fig 2. Triangulation. The FF7 hardware-3D dependency was reached three independent ways. The call-graph path was re-derived by the trace-back, not read from a database. [MEASURED]

Source. perslis map with capstone trace-back (656a4a1), rendered to PNG via graphviz.

Quantity	Value	Provenance
winetricks verbs parsed	469	[MEASURED]
of which DLL-category	328	[MEASURED]
compat tests green	70	[MEASURED]
titles mapped from one CD	5	[MEASURED]
remediation rules in archive	11	[MEASURED]
fitness of the top proven rule	0.90	[MEASURED]
adversarial review models	4	[MEASURED]

Statistic sources.

- **winetricks verbs parsed.** `compat-verbs -fetch (73f279f)`. Official winetricks parsed live.
- **of which DLL-category.** `compat-verbs -category dlls`. Matches the independent protonfixes teardown count exactly.
- **compat tests green.** swift test on the compat subsystem (fb1e24f).
- **titles mapped from one CD.** perslis map over `/Volumes/PCGAMER3_15` (PC Gamer 3/15).
- **remediation rules in archive.** `compat-archive (ebf1549)` + Phase A seeds (61ddc7b).
- **fitness of the top proven rule.** `compat-archive: ddraw-blackscreen→cnc-ddraw`, Laplace-smoothed over recorded trials.
- **adversarial review models.** 3-model council (Codex+Grok+DeepSeek) + a separate 4-model council; multiple Codex stop-gates.

3.2 Ownership, stated plainly

The loop, the routing, the trace-back algorithm, and the self-evolving archive are ours. pefile, capstone, and graphviz are commodity third-party oracles — the same relationship the COBOL vertical has with GnuCOBOL. We do not claim to have written a disassembler; we claim to have built the agent that uses one to decide, fix, verify, and learn. [MEASURED]

Source. Dependency audit: pefile/capstone/graphviz are the only third-party analysis engines; the loop, routing, trace-back, and archive live in kist/Perslis source.

4 The field, side by side

Read from each project’s source, not its marketing. The columns are the four things that separate a loop from a lookup. [MEASURED]

System	Route	Diagnose	Verify	Learn
kist × Perslis	Yes	Yes	Yes	Yes
protonfixes / umu	No	No	No	No
winetricks	No	No	No	No
Lutris	No	No	No	No
Bottles	No	No	No	No
GOG	No	No	No	No

Source. Mechanism-level teardown, 2026-07-07. Capability presence read from each project’s source. We hold ~11 rules against winetricks’ 556 verbs, and ~12 curated titles against thousands. The moat is the loop; the gap is their data. Corpus ingestion (winetricks verbs + protonfixes modules as seed rules) is the closing move — 469 verbs are already parsed and addressable.

5 Adversarial review

The recurring lesson: another tool’s shell string is not your argv [MEASURED]

- **HIGH — found by Codex rescue review.** compatStageFromPath fed a provider-script’s shell-string staging commands to a naive splitter; a /Volumes path emitted `rm -rf X && mkdir Y && find...|cpio...`, whose unparsed `&&` turned staging roots into literal operands of `/bin/rm` — a delete-outside-workspace on writable media. Fixed: fail-closed rejection of shell control operators outside quotes, plus an exec allowlist (`mkdir/ditto/cp/mv/rsync`; `rm` excluded).
- **HIGH — found by Codex (Grok + DeepSeek missed it).** legacy_app allowlist-checked only absolute paths, so a relative `../../etc/x` escaped (Perslis `cwd` = projectRoot). Fixed: reject any `..` component; map output via `ConfinedRoot.makeDirectory`; fail-closed on non-canonicalizable roots.
- **MED — found by Codex rescue review.** compatLaunchDetached reported success on `pid>0`, so a crash-on-start legacy runtime process faked ‘launched’. Fixed: settle, then require the process is still running.

Every subprocess in the fix path is `exec+argv`. Writes stay confined to the project root by construction; external roots (mounted media) are read-only. The review gate is on.

Source. Codex rescue review a70104e + 3-model council; every finding fixed with a regression test.

6 Limitations, without softening

- [PROTOTYPE] VM (T2) tier boots but does not yet self-install. The Win98 VM escalation stages a copy-on-write overlay and install media and boots a real guest under private Perslis control, fail-closed on any unknown tier. It does not yet drive the in-guest

installer unattended — that is a separate sub-project. Escalation to an unavailable tier returns `launched=false` with a reason and stops; it never fakes a fix.

- **[MEASURED] 3Dfx/Glide on Apple Silicon is CPU-bound under wine.** The durable path is the Perslis accelerated legacy graphics runtime: accelerated Glide passthrough from guest to host GPU/Metal. It is gated on a golden VM image and is not yet built. Routing Glide→OpenGL on ARM is a known dead end and the router explicitly avoids it.

Source. Heavy Gear wine campaign: dgVoodoo2 D3D11 rejected by wined3d on M4; RUN_ANYTHING_PLAN.md research sweep.

- **[MEASURED] Thin where the industry is thick.** We hold ~11 rules against winetricks’ 556 verbs, and ~12 curated titles against thousands. The moat is the loop; the gap is their data. Corpus ingestion (winetricks verbs + protonfixes modules as seed rules) is the closing move — 469 verbs are already parsed and addressable.

Source. compat-archive rule count (11) vs winetricks w_metadata count (556); curated dogfood corpus table in RUN_ANYTHING_PLAN.md.

This section exists because a label is worthless the moment it is aspirational. The system is **PILOT-READY**: The perception→diagnose→fix→verify→learn loop is live and measured on the wine tier. The VM (T2) tier is a booting-but-not-unattended PROTOTYPE. This paper is labelled down to what is actually earned.

Accessibility disclosure

The HTML microsite has source-level accessibility gates plus an axe-core 4.10.3 WCAG 2.2 AA browser audit at desktop and 320 CSS pixels. **[MEASURED]** The companion PDF is not yet PDF/UA-tagged. The HTML microsite is the canonical accessible edition.

Source. node scripts/verify_accessibility.mjs; axe-core 4.10.3 browser audits at 1440px and 320px; pdfinfo reports Tagged: no for the companion PDF.

References

1. Open-Wine-Components / umu-protonfixes. Per-AppID Python compatibility fixes. <https://github.com/Open-Wine-Components/umu-protonfixes>
2. Winetricks. Declarative w_metadata verbs with pinned URL+SHA. <https://github.com/Winetricks/winetricks>
3. Lutris. Community-authored install scripts. <https://lutris.net>
4. Bottles. Curated dependency programs for wine prefixes. <https://usebottles.com>
5. dgVoodoo2. Glide / DirectX 1-9 to Direct3D 11 wrapper. <http://dege.freeweb.hu>
6. cnc-ddraw. DirectDraw replacement. <https://github.com/FunkyFr3sh/cnc-ddraw>
7. Capstone disassembly framework. <https://www.capstone-engine.org>
8. pefile. Portable-Executable parsing. <https://github.com/erocarrera/pefile>
9. Zhang et al., “Darwin Gödel Machine: Open-Ended Evolution of Self-Improving Agents,” 2025; after Schmidhuber’s Gödel Machine. <https://arxiv.org/abs/2505.22954>

Fact-registry SHA-256 (first 64 bits): `f48c04536db7d954`