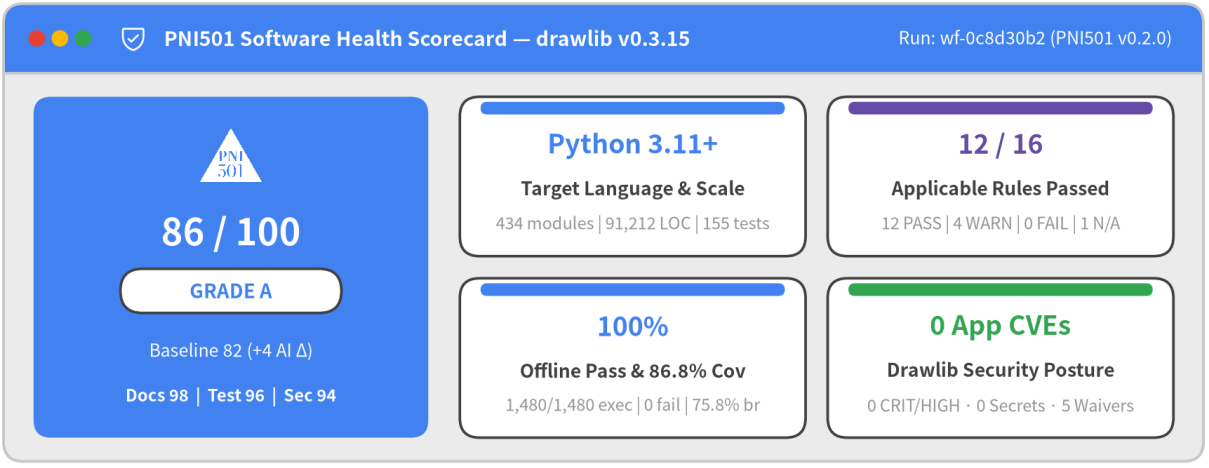
PNI Software Health Diagnostic Report

Target System: `drawlib` (`v0.3.15`) **— Static AST, Architectural Boundary, Hermetic Container Runtime & Supply-Chain Security Audit**



Drawlib v0.3.15 — PNI501 v0.2.0 Full Audit Executive Scorecard (Workflow Run wf-0c8d30b2)

Diagnostic Scope & Audit Metadata

End-to-end engineering assessment of `drawlib` (`v0.3.15`) executed via `pni501` `v0.2.0` (`full_audit`), combining deterministic Python AST verification, layered architecture analysis, hermetic `--network=none` Docker container testing, Trivy `0.69.3` filesystem & image security scanning, and `.pni/pni.toml` policy validation.

Property	Value
Target & Archetypes	`drawlib` (`v0.3.15` , Python `>=3.11` [`3.11–3.14`], PEP 621 `uv` / `uv_build` , `py.typed`) — `library` (38 facade files / 20 top-level lazy exports) + `cli` (9 command modules, 39+ commands)
Workflow & Policy	`full_audit` (`wf-0c8d30b2` , PNI501 `v0.2.0` , `18` steps across `12` logical DAG stages, `.pni/pni.toml` with `5` active architectural waivers) — Git `main@28aff03e` , Evaluated `2026-11-22`
Codebase Footprint	`434` source files (`91,212` LOC, `317` classes, `2,089` functions) & `155` test files (`1,485` collected pytest items / `1,341` top-level AST tests)
Overall Health Score	**86 / 100** (`Grade A`) (Deterministic Baseline `82` -> AI Delta `+4` , anchored within `±10 pt` /axis) — Docs `98` , Testing `96` , Deps `96` , Security `94` , Typing `90` , Arch `80` , Quality `50`
Compliance & Findings	`12 / 16` applicable rules `PASS` (`4 WARN` , `0 FAIL` , `1 N/A` ; `12 / 17` total) — `0` CRIT, `0` HIGH, `446` MED, `14` LOW, `2` INFO (`462` total; `0` Drawlib CVEs, `0` secrets, `0` IaC misconfigs; `5` waivers matched `454` sites)

Table of Contents

1. Executive Summary

2. Architecture & Source Code Structure

3. Recommended Remediation Roadmap

4. Audit Pipeline (DAG), Standards & Findings

5. Deep Dive: Code Quality, Testing & Runtime

6. Deep Dive: Security, Supply Chain & API Surface

1. Executive Summary

Drawlib `v0.3.15` achieves an overall PNI501 `v0.2.0` engineering health score of **86 / 100 (Grade A)** — anchored `+4` points above its **82** deterministic baseline within the strict `±10 pt` per-axis calibration ceiling — across **434 Python source modules (`91,212` LOC, `317` classes, `2,089` functions)** and **155 test files (`1,485` collected pytest items / `1,341` top-level AST tests)**. Static AST verification confirms **100.0% callable typing**, **99.3% docstring and import hygiene**, **0 layer boundary violations** across L1–L7, and **0 bug-risk idioms**; hermetic `--network=none` Docker execution passed **1,480 / 1,485 tests (`0` failed, `0` errors, `5` clean environment skips; `86.8%` line / `75.8%` branch coverage)**, and Trivy `0.69.3` verified **0 CRITICAL/HIGH CVEs**, **0 CVEs in Drawlib's own dependency tree**, **0 IaC misconfigurations**, and **0 leaked secrets**.

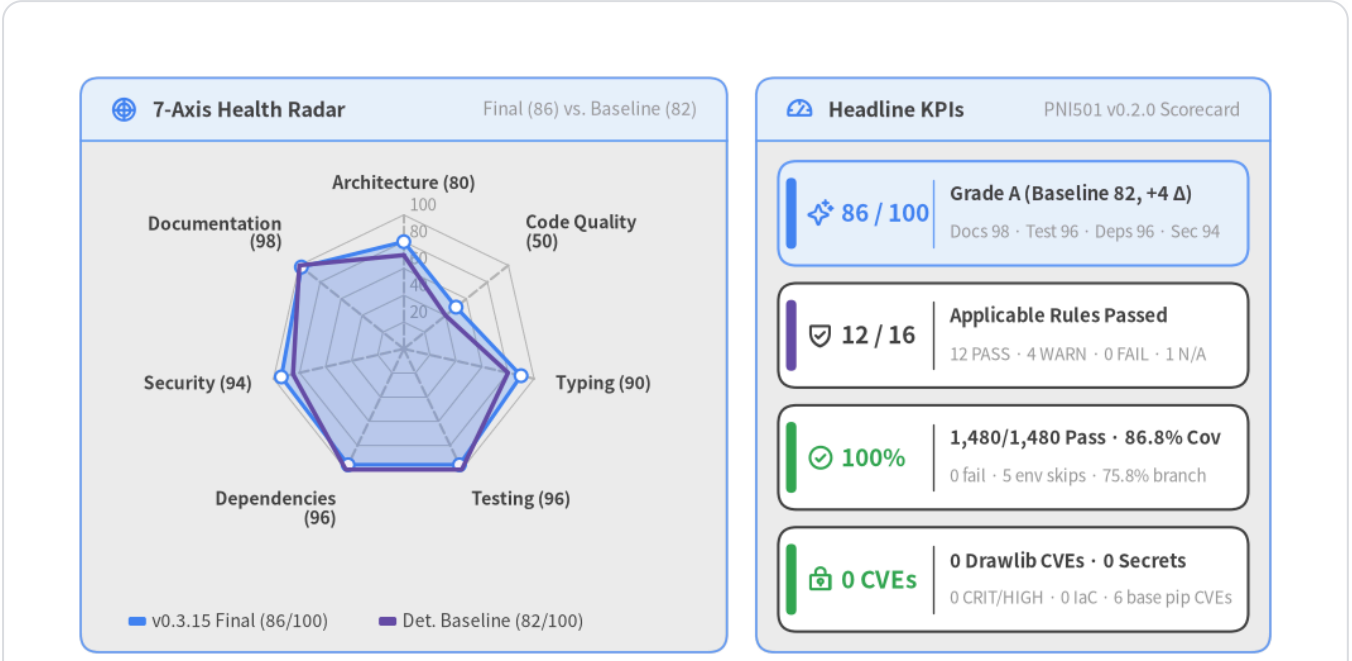


Figure 1.1 — 7-Axis Anchored Health Radar (Final 86 vs. Deterministic Baseline 82) & Executive Diagnostic KPIs

1.1 Executive Synthesis & Diagnostic Headline

PNI501 `v0.2.0` workflow `full_audit` (`wf-0c8d30b2`, Git `main@28aff03e`) executed **18 steps across 12 logical DAG stages** against Drawlib `v0.3.15` (with `5` active `.pni/pni.toml` architectural waivers matching `454` sites and `10` deterministic findings reclassified during AI triage):

- **PNI501 `v0.2.0` Anchored Scoring & Overall Standing**: Drawlib scores **86 / 100 (`Grade A`)**, adjusted `+4` points above its **82** deterministic baseline under the `±10 pt` per-axis anchoring rule. `12 / 16` **applicable rules** `PASS` (`4 WARN`, `0 FAIL`, `1 NOT_APPLICABLE` [`python/frameworks/fastapi`]; `12 / 17` total). Across all `462` consolidated findings, `0` **are CRITICAL and** `0` **are HIGH** (`446` MEDIUM — `442` SIG unit-size/complexity flags + `4` architectural cycle groups — `14` LOW, and `2` INFO).
- **Measurable Engineering Progress (`v0.3.5` -> `v0.3.15`)**: While the codebase grew from `424` modules (`85,323` LOC, `150` tests) to `434` **modules (`91,212` LOC, `155` tests / `1,485` pytest items)**, Drawlib landed every major `v0.3.5` governance and runtime follow-up: publishing `SECURITY.md` (with an explicit Illustration-as-Code trust model), `CODE_OF_CONDUCT.md`, and weekly Dependabot (`uv` + `github-actions`); pre-baking Chromium (`libatomic1`) and all `83` cached assets into `.pni/dockerfiles/Dockerfile` to raise offline executed test pass rate from `99.5%` to `100.0%` **(`1,480 / 1,480` executed, `0` failures)** and line coverage from `86.3%` to `86.8%`; and expanding bilingual Diataxis portals to `6` sites with automated `100%` EN/JA API & CLI completeness checks. The numeric shift from `v0.3.5`'s unanchored `94` (`Grade S`) to `v0.3.15`'s anchored `86` (`Grade A`) stems purely from `v0.2.0`'s stricter deterministic SIG baseline on `Code Quality` (`40 -> 50`) and `Architecture` (`70 -> 80`), while all five core verification axes (`Documentation 98`, `Testing 96`, `Dependencies 96`, `Security 94`, `Typing 90`) sit firmly at `Grade S` levels.
- **Audit Step Scores (Weakest -> Strongest)**: `python_ast_audit = 88/100 (Grade A)` is the lone non-S audit step (reflecting the SIG unit-size/complexity concentration and `8` static import cycles), followed by seven **Grade S** steps: `cli_guidelines_audit = 93`, `security_vulnerability_audit = 94`, `library_hygiene_audit = 94`, `supply_chain_audit = 95`, `python_test_audit = 95`, `repo_structure_audit = 96`, and `container_runtime_audit = 96`.
- **Single Primary Improvement Lever**: `Code Quality` (`50 / 100`) accounts for nearly the entire gap to `Grade S` (`>=90`). Breaking the `8` static import cycles in `_builder/_common/parallel.py`, `compiler/html.py <-> compiler/jupyter.py`, and

`_diagrams/_edge <-> _junction`, decomposing top cognitive-complexity builder orchestrators (`compiler/html.py`, `parallel.py`, `manifest.py`, `exporter_docx.py`), and refreshing base-image `pip@25.0.1 -> >=26.2.0` will lift Drawlib cleanly back into `Grade S` under anchored scoring.

1.2 Seven-Category Engineering Health Breakdown

Category	Baseline	AI Δ (±10pt)	Final Score	Audit Rationale & Key Evidence
Documentation	`100`	`-2`	**98 / 100**	`99.3%` Google-style docstring coverage (`0` empty docstrings), `6` bilingual Diataxis portals (`site`, `api`, `architecture`, `quickstart`, `about-drawlib`, `jupyter`), embedded AI rules in `src/drawlib/_rules/`, `SECURITY.md` + `CODE_OF_CONDUCT.md`, and automated `100%` EN/JA API & CLI completeness test; `-2` for `12` undocumented `@property` setters.
Testing	`100`	`-4`	**96 / 100**	Hermetic `--network=none` Docker run passed `1,480 / 1,485` tests (`0` failed, `0` errors, `5` clean env skips), `86.8%` line coverage (`40,102 / 46,204`), `75.8%` branch coverage, `100.0%` AST module coverage across `155` test files (`1,341` top-level AST tests), and `>=97%` pixel-match visual regression; `-4` for coverage gaps in `parallel.py` (`61.1%` / `50.6%`), `cli/cache.py` (`61.2%` / `30.0%`), and `__main__.py` (`0.0%`).
Dependencies	`100`	`-4`	**96 / 100**	`13` SemVer `>=` lower-bounded runtime dependencies in PEP 621 `pyproject.toml` (`uv` / `uv_build`), `0` unpinned specs, `0` CVEs in Drawlib's dependency tree, active weekly Dependabot (`uv` + `github-actions`), and `python/lockfile_committed` waived; `-4` for floating `@v7` GitHub Actions tags and `109`-LOC `docs_src/site/build_html.sh`.
Security	`85`	`+9`	**94 / 100**	Trivy `0.69.3` filesystem + resolved container scan confirmed `0` CRITICAL, `0` HIGH, `0` Drawlib CVEs, `0` IaC misconfigurations (`DS-0026` waived ×3), `0` leaked secrets, `0` unwaived SAST sinks (`8` `CWE-94` diagram-execution sites waived), `SECURITY.md`, SHA-256 asset verification, and `5` least-privilege CI workflows; held at `94` for `6` base-image system `pip@25.0.1` CVEs (fixed in `26.2.0`).
Typing	`80`	`+10`	**90 / 100**	`100.0%` AST function type coverage across all `2,089` functions, `0` unscoped `# type: ignore` comments, packaged PEP 561 `py.typed`, and `0` `ty` static type errors; `200` weak `Any` or unparameterized `dict` annotations remain in dynamic facades and `manifest.py`.

Category	Baseline	AI Δ (±10pt)	Final Score	Audit Rationale & Key Evidence
Architecture	`70`	`+10`	**80 / 100**	`0` boundary violations across L1–L7 (`l1_core -> l2_types -> l3_* -> l4_canvas -> domain -> builders -> PEP 562 facades`) across `434` modules (`91,212` LOC); penalized for `8` static import cycles (`5` deferred worker imports in `_builder/_common/parallel.py` , `1` `compiler/html.py <-> compiler/jupyter.py` , `2` `_diagrams` `_edge <-> _junction`).
Code Quality	`40`	`+10`	**50 / 100**	Strict SIG deterministic baseline penalized `442` functions exceeding `>50` LOC or `>15` cognitive complexity in builder/renderer orchestrators (`compiler/html.py` , `compiler/jupyter.py` , `exporter_docx.py` , `parallel.py` , `manifest.py`); `+10` reflects `99.3%` docstrings, `99.3%` import hygiene, `0` bug-risk idioms, Ruff+Ty clean, and `353` keyword-only APIs waived via `.pni/pni.toml` .
Overall Health	`82`	`+4`	**86 / 100 (A)**	Anchored 7-axis composite (`±10 pt` per-axis ceiling) across `18` steps / `12` logical DAG stages (`12 / 16` applicable rules `PASS` , `4 WARN` , `0 FAIL` , `1 N/A`).

1.3 Verified Strengths vs. Top Risks

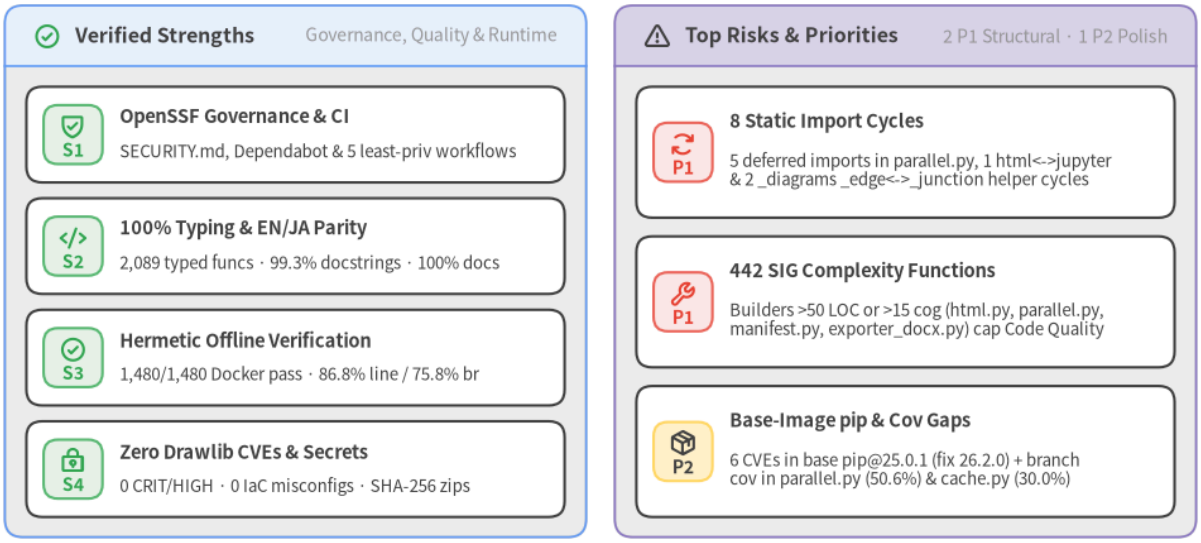


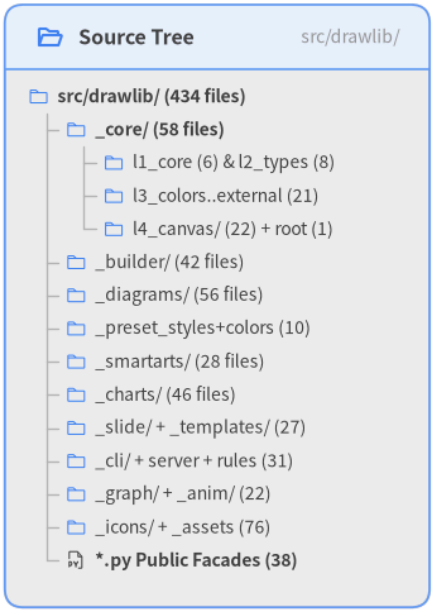
Figure 1.2 — Verified Engineering Strengths (S1–S4) vs. Prioritized Remediation Targets (P1–P2)

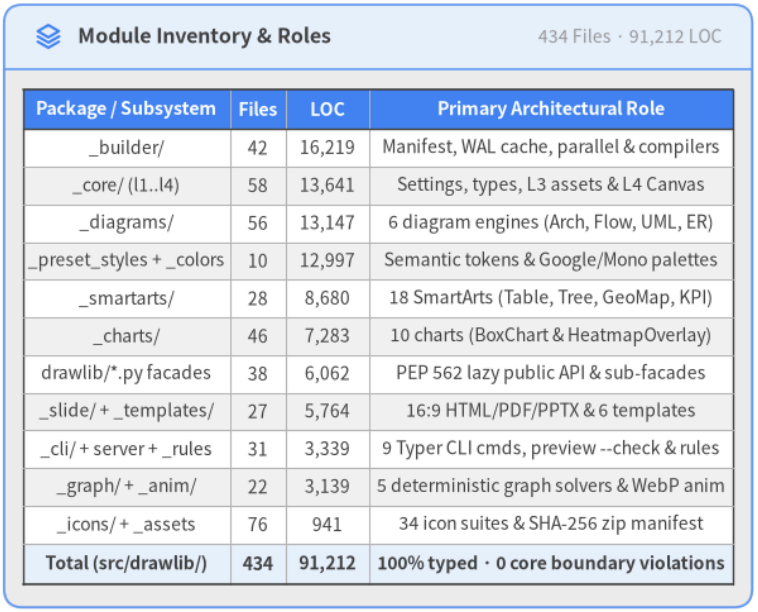
- **Path Back to `Grade S` Under Anchored Scoring**: Drawlib `v0.3.15` already exhibits zero `FAIL` rules, zero `CRITICAL` / `HIGH` vulnerabilities, zero Drawlib dependency CVEs, and `100%` hermetic offline test pass reliability. Executing the two `P1`

architectural/maintainability refactors (`parallel.py` extraction + `compiler/base.py` split and orchestrator decomposition) alongside the `P2` base-image `pip` upgrade (`25.0.1 -> >=26.2.0`), SHA-pinned GitHub Actions, and targeted branch tests for `parallel.py` and `cli/cache.py` directly addresses all four `WARN` rules detailed in Chapters 3–6.

2. Architecture & Source Code Structure

Drawlib v0.3.15 implements a **Strict Layered Core + Domain Component Subsystems + Multi-Target Compilation Pipeline behind PEP 562 Public Facades** across **434 Python source files (`91,212` LOC, `317` classes, `2,089` functions, backed by `155` test files)**. Core layers (`L1-L2` through `L4`) and domain visual subsystems (`L5`) maintain **zero core boundary violations** and strictly downward dependencies, while PNI501 v0.2.0 telemetry localizes the remaining `8` static import cycles and `8` peak complexity hotspots (`risk_score = 1.00`) primarily to L6 multi-target builder orchestration (`compiler/html.py`, `image_builder.py`, `exporter_docx.py`, `exporter_jupyter.py`, `parallel.py`), the L5 slide compiler (`_slide/compiler.py`), and L4 primitive shape builders (`_symbol_builders.py`, `_container_builders.py`).





Package / Subsystem	Files	LOC	Primary Architectural Role
_builder/	42	16,219	Manifest, WAL cache, parallel & compilers
_core/ (l1..l4)	58	13,641	Settings, types, L3 assets & L4 Canvas
_diagrams/	56	13,147	6 diagram engines (Arch, Flow, UML, ER)
_preset_styles + _colors	10	12,997	Semantic tokens & Google/Mono palettes
_smartarts/	28	8,680	18 SmartArts (Table, Tree, GeoMap, KPI)
_charts/	46	7,283	10 charts (BoxChart & HeatmapOverlay)
drawlib/*.py facades	38	6,062	PEP 562 lazy public API & sub-facades
_slide/ + _templates/	27	5,764	16:9 HTML/PDF/PPTX & 6 templates
_cli/ + server + _rules	31	3,339	9 Typer CLI cmds, preview --check & rules
_graph/ + _anim/	22	3,139	5 deterministic graph solvers & WebP anim
_icons/ + _assets	76	941	34 icon suites & SHA-256 zip manifest
Total (src/drawlib/)	**434**	**91,212**	**100% typed · 0 core boundary violations**

Figure 2.1 — Physical Source Directory Hierarchy & Reconciled Module Inventory (434 Files, 91,212 LOC)

2.1 System Profile & Physical Module Inventory

Drawlib operates simultaneously as a **programmatic Python library** (`drawlib.*`) and a **headless compiler CLI** (`drawlib`), targeting Python `>=3.11` (`3.11–3.14` managed via `uv` / `uv_build` PEP 621). Its rendering and publishing stack pairs **Matplotlib (`>=3.8`), Pillow (`>=13.0.0`), and NumPy (`>=1.26`)** for bottom-left Cartesian `(0, 0)` 2D vector/raster drawing with **Pydantic v2 (`>=2.8`)** (`@validate_call` and immutable `BaseModel` configs), **Typer (`>=0.12.0`) & Rich (`>=13.0.0`)** (9-command CLI), **markdown-it-py (`>=3.0`), Jinja2 (`>=3.1`), Pygments (`>=2.18`), and MarkupSafe (`>=2.1`)** (multi-target Markdown/HTML/Jupyter compilation), **Playwright (`>=1.49.0`)** (headless Chromium A4 PDF and slide rasterization), **python-docx (`>=1.2.0`) & python-pptx (`>=1.0.2`)** (OpenXML Office export), and **stdlib** `sqlite3` (`.drawlib/cache.db` incremental SHA-256 build cache).

Package / Subsystem	Files	LOC	Architectural Layer	Primary Responsibility
`src/drawlib/_builder/`	42	16,219	**L6**	`drawlib.toml` manifest, SQLite WAL cache, parallel prerender, HTML/PDF/DOCX/Jupyter/MD/Image compilers
`src/drawlib/_core/` (`l1..l4`)	58	13,641	**L1–L4**	`l1_core` (6), `l2_types` (8), `l3_*` subsystems (21), `l4_canvas` (22), package root (1)
`src/drawlib/_diagrams/`	56	13,147	**L5**	6 technical diagram engines (`Architecture` , `Flow` , `Sequence` , `Class` , `ER` , `State`)
`src/drawlib/_preset_styles` + `_preset_colors`	10	12,997	**L3**	Semantic style tokens and `Default` , `Google` , and `Monochrome` palettes
`src/drawlib/_smartarts/`	28	8,680	**L5**	18 SmartArts (`Table` , `TreeNode` , `MindMapNode` , `Roadmap` , `KpiCards` , `GeoMap` , `Calendar` , etc.)
`src/drawlib/_charts/`	46	7,283	**L5**	10 quantitative charts including `BoxChart` and `HeatmapOverlay`
`src/drawlib/*.py` facades (`charts/` , `diagrams/`)	38	6,062	**L7**	PEP 562 lazy public API facades (`__init__.py` , `tools.py` , domain submodules)
`src/drawlib/_slide/` + `_templates/`	27	5,764	**L5–L6**	16:9 HTML/PDF/PPTX slide deck compiler and 6 project templates (including Jupyter)
`src/drawlib/_cli/` + `_http_server` + `_rules`	31	3,339	**L6–L7**	9 Typer CLI modules, local preview server with `--check` link validator, embedded AI rules
`src/drawlib/_graph/` + `_anim/`	22	3,139	**L5**	5 deterministic graph layout solvers and multi-frame WebP animation engine
`src/drawlib/_icons/` + `_assets`	76	941	**L3 / L5**	34 icon suites (`75` icon/release-asset modules + `1` `_cached_assets` init) & SHA-256 zip manifest
Total (`src/drawlib/`)	**434**	**91,212**	**L1–L7**	`317` **classes** · `2,089` **functions** · `155` **test files** · `100%` **typed** · `0` **core boundary violations**

Four external integrations are cleanly encapsulated behind dedicated runtime boundaries:

1. **GitHub Releases Asset Distribution (** `src/drawlib/_core/l3_external/_package.py` **&** `src/drawlib/_release_assets.py` **)**: Validates requested filenames against `pkg.files` prior to network I/O and lazily downloads `83` font, icon, and GeoJSON map `.zip` packages from `yuichi110/drawlib` with SHA-256 manifest verification and atomic extraction.
2. **Playwright Headless Chromium (** `src/drawlib/_builder/_common/browser.py` **)**: Coordinates cross-platform `fcntl` / `msvcrt` file locks for browser setup, HTML-to-PDF A4 pagination, and 16:9 slide DOM/raster export.
3. **Google Fonts & Google Colaboratory**: Optional HTML stylesheet links and Colab launch buttons embedded in compiled sites and Jupyter notebooks.
4. **PyPI & TestPyPI Distribution**: PEP 621 wheel/sdist builds via `uv_build` and clean-room container testing via `./dcli docker test`.

2.2 Multi-Layer Architecture, Dependency Boundaries & Thermal Risk Hotspots

Drawlib organizes its `434` modules into **6 recorded architectural layer groups (** `L1-L2` **,** `L3` **,** `L4` **,** `L5` **,** `L6` **,** `L7` **)**:

- **Zero Core Boundary Violations**: Core rendering layers (`L1-L2 -> L3 -> L4`) and domain visual libraries (`L5`) maintain strictly downward dependencies behind PEP 562 lazy public facades (`L7`), with `40` type-checking-only annotations safely guarded behind `if TYPE_CHECKING:` (zero runtime impact).
- **8 Localized Static Import Cycles (** `ARCH-02` **Focus)**: All runtime/static circular imports are concentrated in two narrow zones: `6` cycles in L6 builder orchestration (`5` deferred in-function worker imports inside `src/drawlib/_builder/_common/parallel.py` reaching up into higher-level builders, plus `1` deferred `compiler/html.py <-> compiler/jupyter.py` asset helper handoff) and `2` minor sibling helper cycles inside `src/drawlib/_diagrams/{architecture,flow}` (`_edge.py <-> _junction.py`). Extracting `parallel.py` above `_builder/_common` and moving shared favicon/logo helpers to `compiler/base.py` yields a strictly acyclic dependency graph.

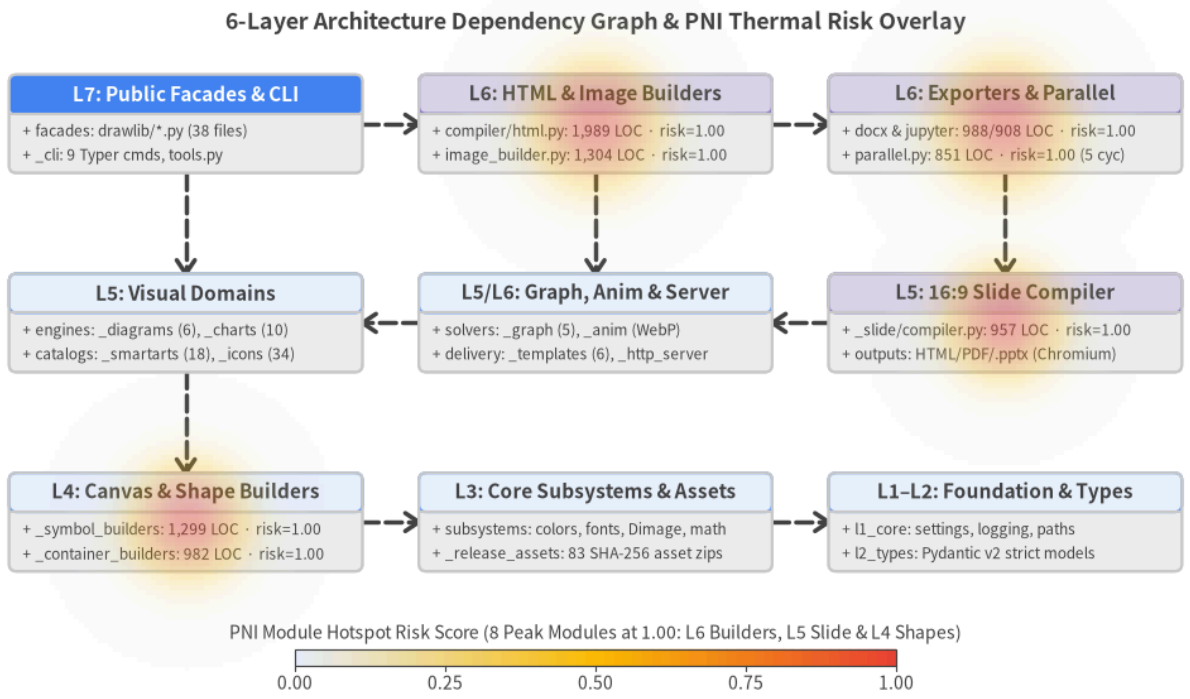


Figure 2.2 — 6-Layer Architecture Dependency Graph with PNI Thermal Risk Overlay (8 Peak Modules at risk_score = 1.00)

Architectural Layer Contract (`L7 -> L1-L2`)

Layer	Architectural Group	Physical Paths	Allowed Dependencies
L7	Public Facades & CLI Delivery	`src/drawlib/__init__.py`, `__main__.py`, `tools.py`, `charts/`, `diagrams/`, `_cli`	`L1–L6`, `typer`, `rich`
L6	Multi-Target Document Builder & Asset Pipeline	`src/drawlib/_builder`, `_templates`, `_http_server`, `_rules`	`L1–L5`, `markdown-it-py`, `jinja2`, `playwright`, `python-docx`
L5	Domain Visual Components	`src/drawlib/_icons`, `_smartarts`, `_charts`, `_diagrams`, `_graph`, `_anim`, `_slide`	`L1–L4`, `python-pptx`
L4	Canvas Rendering Engine	`src/drawlib/_core/l4_canvas`	`L1–L3`, `matplotlib`
L3	Core Subsystems & Asset Engine	`src/drawlib/_core/l3_*`, `_preset_colors`, `_preset_styles`, `_release_assets.py`	`L1–L2`, `pillow>=13.0.0`, `numpy`
L1–L2	Core Foundation & Types	`src/drawlib/_core/l1_core`, `_core/l2_types`	Python `stdlib` & `pydantic` only

Pre-Sorted Module Risk Hotspots (`risk_score = 1.00`)

Rank	Module Path	Layer	LOC	Risk Score	Findings	Triggered Rules
1	`src/drawlib/_builder/doc_builder/compiler/html.py`	L6	1,989	`1.00`	18 medium	`common/architecture`, `common/maintainability`
2	`src/drawlib/_builder/image_builder.py`	L6	1,304	`1.00`	15 medium	`common/maintainability`
3	`src/drawlib/_core/l4_canvas/_shapes/_symbol_builders.py`	L4	1,299	`1.00`	9 medium	`common/maintainability`
4	`src/drawlib/_builder/doc_builder/exporter_docx.py`	L6	988	`1.00`	7 medium	`common/maintainability`
5	`src/drawlib/_core/l4_canvas/_shapes/_container_builders.py`	L4	982	`1.00`	9 medium	`common/maintainability`
6	`src/drawlib/_slide/compiler.py`	L5	957	`1.00`	11 medium	`common/maintainability`
7	`src/drawlib/_builder/doc_builder/exporter_jupyter.py`	L6	908	`1.00`	7 medium	`common/maintainability`
8	`src/drawlib/_builder/_common/parallel.py`	L6	851	`1.00`	8 medium, 1 low	`common/architecture`, `common/maintainability`, `common/runtime_verification`

2.3 Runtime Execution Flows, State & Concurrency Model

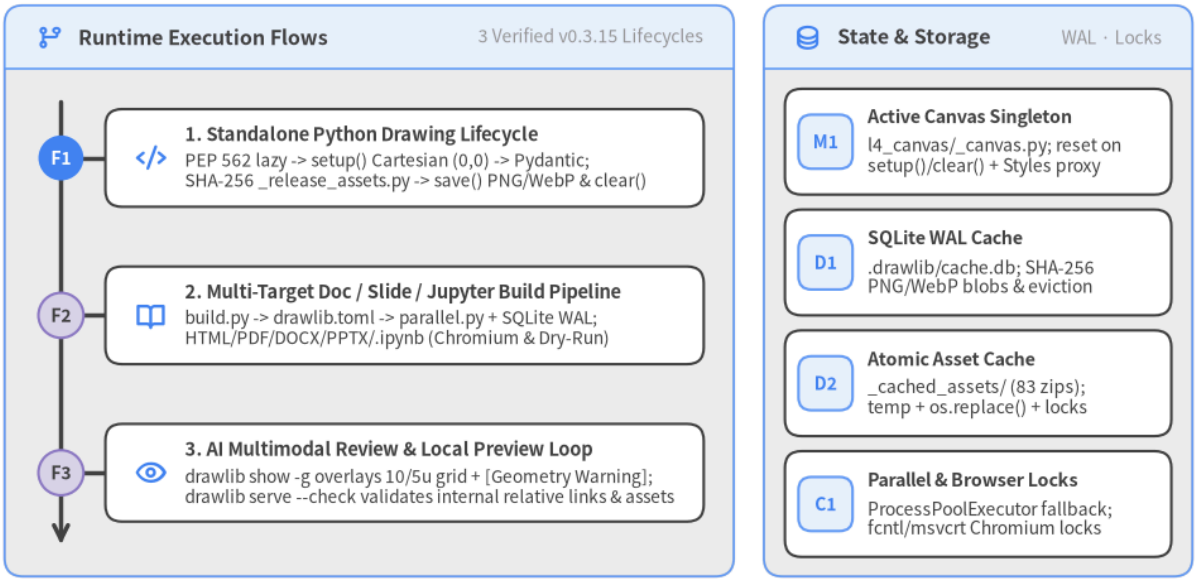


Figure 2.3 — Recorded Runtime Execution Pipelines & Stateful/Persistent Storage Architecture (v0.3.15)

- **Execution Entrypoints**: Programmatic callers enter through `src/drawlib/__init__.py`, `src/drawlib/canvas.py`, and `src/drawlib/tools.py`, while CLI automation enters through `src/drawlib/__main__.py:main` -> `src/drawlib/_cli/_drawlib.py:app` and `src/drawlib/_cli/commands/build.py:build_app` across 9 Typer command groups (`init`, `build`, `show`, `serve`, `cache`, `css`, `styles`, `colors`, `rules`).
- **State, Storage & Persistence**:
  1. **Active Canvas & Theme Proxies**: In-process active canvas singleton managed via `setup()` / `save()` / `clear()` in `src/drawlib/_core/l4_canvas/_canvas.py`, paired with active `Styles` / `Colors` proxies in `src/drawlib/_core/l3_styles`.
  2. **Content-Addressed SQLite WAL Cache**: `.drawlib/cache.db` (`BuildImageCache` and `CliImageCache` in `src/drawlib/_builder/_common/cache.py`) stores SHA-256-keyed rendered PNG/WebP blobs with bounded eviction.
  3. **Atomic Local Asset Cache**: `src/drawlib/_cached_assets/{fonts,fonticons,icons,maps,rules}` populates assets atomically using temporary files plus `os.replace()` and OS-level file locks.

4. **Headless Chromium Coordination**: Playwright Chromium binary management uses cross-platform `fcntl` / `msvcrt` OS locks in `src/drawlib/_builder/_common/browser.py`.

- **Error & Concurrency Model**: Public drawing and builder APIs validate inputs eagerly via Pydantic (`@validate_call` and immutable `BaseModel` configs) and raise explicit `ValueError` / `FileNotFoundError` / `RuntimeError` exceptions (including explicit `ValueError` guards for unsupported targets on `jupyter` projects). Multi-block document compilation parallelizes across isolated worker processes (`ProcessPoolExecutor` in `src/drawlib/_builder/_common/parallel.py`) with deterministic single-process fallback, SQLite WAL concurrency, and atomic temp-file + OS lock guards preventing multi-worker corruption.

3. Recommended Remediation Roadmap

Drawlib v0.3.15 enters this PNI501 v0.2.0 audit at **86/100 (Grade A)** with **zero** `P0` **critical/high blockers**—celebrating the complete in-tree closure of every major `v0.3.5` `P0` / `P1` milestone (`pillow>=13.0.0` clearing all 41 Pillow CVEs, non-root Docker `USER drawlib` `UID 10001` , `pkg.files` offline asset pre-validation, pre-baked Chromium + 83 asset packages, root `SECURITY.md` + Dependabot, and library-level `sys.exit(1)` cleanup). Building on that hardened baseline, this chapter specifies the **10 pre-sorted** `v0.3.15` **engineering actions (** `0 P0` , `2 P1` , `8 P2` **)** required to achieve a strictly acyclic `ARCH-02 PASS` , reduce builder cognitive complexity, and polish CLI/library hygiene, supply-chain pinning, container CVE hygiene, and branch coverage above `80%` .

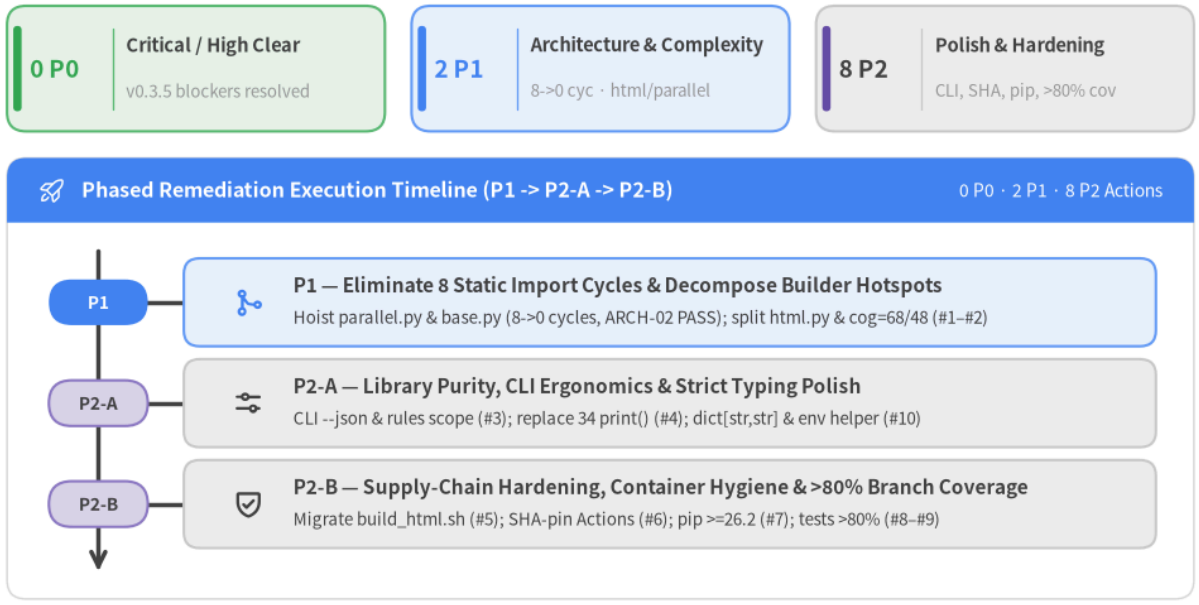


Figure 3.1 — Phased Engineering Remediation Timeline for Drawlib v0.3.15 (0 P0 · 2 P1 · 8 P2)

3.1 Remediation Strategy & Priority Horizons

> **Completed `v0.3.5` Milestones Verified in `v0.3.15`**: Every urgent blocker from the prior `v0.3.5` audit has been resolved and verified in-tree—including `pillow>=13.0.0` (`0` PyPI application CVEs in `uv.lock`), **non-root Docker** `USER drawlib` (`UID 10001`) (`0` `DS-0002` IaC misconfigurations), `pkg.files` **offline asset membership checks**, **pre-baked Chromium + 83 asset packages** in `.pni/dockerfiles/Dockerfile`, published `SECURITY.md` + `.github/dependabot.yml`, and complete removal of library-level `sys.exit(1)` calls.

With **zero** `P0` **blockers** remaining, the `v0.3.15` roadmap prioritizes **10 targeted engineering actions** across three clean execution waves:

1. **Wave** `P1` **— Architectural Acyclicity & Builder Complexity Decomposition (Actions** `#1–#2`**)**: Eliminate all `8` static import cycles (reaching `ARCH-02 PASS` with `0` cycles) and decompose the largest L6 compiler and orchestration routines (`compiler/html.py` at `1,989` LOC, `_execute_block_render_job` at cognitive complexity `68`, `format_project_manifest_toml` at `48`, plus `exporter_docx.py` / `compiler/jupyter.py`) into focused `<50` LOC units.
2. **Wave** `P2-A` **— Library Purity, CLI Ergonomics & Strict Typing (Actions** `#3`, `#4`, `#10`**)**: Add `--json` output to CLI inspection commands, scope `_ensure_overview_rules_cached()` away from read-only CLI invocations, replace `34` direct builder `print()` calls with quiet-aware logging or injected progress callbacks, tighten `dict[str, str]` typing in `manifest.py`, and centralize `15` `DRAWLIB_*` environment lookups.
3. **Wave** `P2-B` **— Supply-Chain Hardening, Container CVE Closure & Branch Coverage** `>80%` **(Actions** `#5–#9`**)**: Migrate the 109-LOC `docs_src/site/build_html.sh` script into typed Python `dcli`, pin GitHub Actions workflows to immutable 40-char commit SHAs, upgrade or strip container base-image `pip 25.0.1` (`6 -> 0` container app CVEs), and close remaining branch-coverage gaps in `parallel.py`, `commands/cache.py`, and `__main__.py`.

3.2 Action Plan Traceability Matrix (`P1 -> P2`)

The matrix below maps all **10 pre-sorted** `v0.3.15` **engineering actions** (`2 P1` **+** `8 P2`) to their exact target files, expected impacts, and PNI501 rule / CVE identifiers.

Action Plan Traceability Matrix — 10 Pre-Sorted Engineering Actions (0 P0 · 2 P1 · 8 P2)

Priority · #	Action Title	Target Files	Expected Impact	Related Rules / CVEs
[P1] · #1 arch	Eliminate all 8 static import cycles (builder & diagrams)	parallel.py, html.py, jupyter.py, base.py	Static cycles 8 -> 0; achieves ARCH-02 PASS	common/architecture, common/maintainability
[P1] · #2 quality	Decompose top cognitive-complexity builder functions	compiler/html.py, parallel.py, manifest.py	Helpers <50 LOC; lifts SIG maintainability	common/maintainability
[P2] · #3 quality	Add --json on CLI list cmds & scope rules cache pre-warm	_cli/_drawlib.py, cache.py, rules.py	clig.dev scriptability & read-only container safe	common/cli_guidelines
[P2] · #4 quality	Replace 34 builder print() with logger/progress hooks	progress.py, parallel.py, compiler/jupyter.py	Keeps drawlib.tools stdout-clean embedded	common/ library_guidelines
[P2] · #5 maint	Migrate 109-LOC Bash script build_html.sh into dcli	docs_src/site/build_html.sh, dcli/docs/builder.py	Eliminates sole legacy shell script finding	common/supply_chain
[P2] · #6 security	Pin GitHub Actions steps to immutable 40-char SHAs	quality.yml, release.yml, ut_template.yml	Maximizes OpenSSF Pinned-Dependencies	common/supply_chain
[P2] · #7 security	Upgrade or strip base-image pip 25.0.1 in Dockerfiles	.pni/dockerfiles/Dockerfile, Dockerfile.test_pypi	Container image app CVEs reduced 6 -> 0	security_vulnerability, CVE-2026-8643/-13346
[P2] · #8 testing	Close coverage gaps in CLI cache & parallel workers	test_cli_css.py, __main__.py, test_parallel_build.py	Clears non-generated <65% line-cov modules	python/testing
[P2] · #9 testing	Increase branch coverage for parallel.py & CLI cache	parallel.py, __main__.py, _cli/commands/cache.py	Raises critical CLI & parallel branch cov >80%	common/ runtime_verification
[P2] · #10 typing	Tighten dict/Any types & centralize DRAWLIB_* env	manifest.py, browser.py, cache.py, progress.py	Higher typing precision & cleaner embeddability	python/typing, python/modern_stack

Figure 3.2 — Action Plan Traceability Matrix Across All 10 Pre-Sorted Engineering Items (0 P0 · 2 P1 · 8 P2)

Summary Table of All 10 Pre-Sorted Action Items

#	Priority	Category	Action Title	Target Files	Related Rules / CVEs	Expected Impact
1	`P1`	`architecture`	**Eliminate all 8 static import cycles in builder and diagram modules**	`src/drawlib/_builder/_common/parallel.py`, `src/drawlib/_builder/doc_builder/compiler/html.py`, `src/drawlib/_builder/doc_builder/compiler/jupyter.py`, `src/drawlib/_builder/doc_builder/compiler/base.py`	`common/architecture`, `common/maintainability`	Reduces static/runtime circular dependencies `8 -> 0`; achieves strict `ARCH-02 PASS`.

#	Priority	Category	Action Title	Target Files	Related Rules / CVEs	Expected Impact
2	P1	code_quality	**Decompose top cognitive-complexity builder functions**	src/drawlib/_builder/doc_builder/compiler/html.py, src/drawlib/_builder/_common/parallel.py, src/drawlib/_builder/_common/manifest.py	common/maintainability	Improves SIG maintainability, lowers review risk, and boosts unit testability (< 50 LOC helpers).
3	P2	code_quality	**Add --json output flag to CLI catalog inspection commands and scope rules cache pre-warming**	src/drawlib/_cli/_drawlib.py, src/drawlib/_cli/commands/cache.py, src/drawlib/_cli/commands/rules.py	common/cli_guidelines	Improves clig.dev scriptability and guarantees read-only container CLI safety.
4	P2	code_quality	**Replace direct stdout print() calls in programmatic builder modules with logger/progress callbacks**	src/drawlib/_builder/_common/progress.py, src/drawlib/_builder/_common/parallel.py, src/drawlib/_builder/doc_builder/compiler/jupyter.py	common/library_guidelines	Keeps drawlib.tools stdout-clean when embedded inside services, notebooks, or external CLIs.
5	P2	maintainability	**Migrate oversized Bash script docs_src/site/build_html.sh into typed Python dcli**	docs_src/site/build_html.sh, tools/dcli/docs/builder.py	common/supply_chain	Eliminates the sole legacy-language shell script finding and improves cross-platform maintainability.
6	P2	security	**Pin GitHub Actions steps to immutable commit SHAs**	.github/workflows/quality.yml, .github/workflows/release.yml, .github/workflows/ut_template.yml	common/supply_chain	Eliminates mutable tag hijacking risk and maximizes OpenSSF Scorecard Pinned-Dependencies.
7	P2	security	**Upgrade or strip base-image pip 25.0.1 in container Dockerfiles**	.pni/dockerfiles/Dockerfile, tools/docker/Dockerfile.test_pypi	common/security_vulnerability, CVE-2026-8643, CVE-2026-13346	Brings Trivy container image application CVEs from 6 -> 0.
8	P2	testing	**Close remaining branch coverage gaps in CLI cache commands and parallel build workers**	tests/drawlib/cli/test_cli_css.py, tests/drawlib/doc_builder/test_parallel_build.py, src/drawlib/__main__.py	python/testing	Eliminates remaining non-generated <65% line-coverage modules.

#	Priority	Category	Action Title	Target Files	Related Rules / CVEs	Expected Impact
9	`P2`	`testing`	**Increase branch coverage for `parallel.py` and CLI cache commands**	`src/drawlib/_builder/_common/parallel.py`, `src/drawlib/_cli/commands/cache.py`, `src/drawlib/__main__.py`	`common/runtime_verification`	Raises critical CLI and parallel orchestration branch coverage `>80%`.
10	`P2`	`typing`	**Tighten unparameterized `dict` / `Any` annotations and centralize builder env/logging**	`src/drawlib/_builder/_common/manifest.py`, `src/drawlib/_builder/_common/browser.py`, `src/drawlib/_builder/_common/cache.py`, `src/drawlib/_builder/_common/progress.py`	`python/typing`, `python/modern_stack`	Improves typing precision and programmatic embeddability without polluting stdout.

3.3 Detailed Engineering Specifications by Priority Tier

Phase `P1` — Architectural Acyclicity & Builder Complexity Decomposition (`2 Actions`)

Action #1 (`[P1]` · `architecture`): Eliminate all 8 static import cycles in builder and diagram modules

- **Target Files**: `src/drawlib/_builder/_common/parallel.py`, `src/drawlib/_builder/doc_builder/compiler/html.py`, `src/drawlib/_builder/doc_builder/compiler/jupyter.py`, `src/drawlib/_builder/doc_builder/compiler/base.py`
- **Related Rules**: `common/architecture`, `common/maintainability`
- **Implementation Guidance**: Promote `parallel.py` above `src/drawlib/_builder/_common/` (e.g., to `src/drawlib/_builder/parallel.py`) so worker orchestration depends strictly downward on `_common` and compilers (breaking all `5` deferred worker cycles). Move shared `resolve_favicon_href` and `resolve_logo_hrefs` helpers into `compiler/base.py` (breaking the `compiler/html.py`

`<-> compiler/jupyter.py` cycle), and extract a shared endpoint `Protocol` to decouple `_edge.py <-> _junction.py` inside `src/drawlib/_diagrams/{architecture,flow}`.

- **Expected Impact**: Reduces static/runtime circular dependencies from `8` to `0` across all `434` source modules and achieves strict `ARCH-02 PASS`.

Action #2 (`[P1]` · `code_quality`): Decompose top cognitive-complexity builder functions

- **Target Files**: `src/drawlib/_builder/doc_builder/compiler/html.py`, `src/drawlib/_builder/_common/parallel.py`, `src/drawlib/_builder/_common/manifest.py`
- **Related Rules**: `common/maintainability`
- **Implementation Guidance**: Extract mode-specific compilation units (site, doc, catalog, bundled Jupyter) out of `compiler/html.py` (`1,989` LOC) into dedicated sibling modules. Split `_execute_block_render_job` (cognitive complexity `68`) in `parallel.py` and `format_project_manifest_toml` (cognitive complexity `48`) in `manifest.py` —alongside high-branch routines in `exporter_docx.py` and `compiler/jupyter.py` —into cohesive helper functions under `50` LOC.
- **Expected Impact**: Substantially improves SIG maintainability, lowers PR review risk across the top `risk_score = 1.00` hotspots, and simplifies isolated unit testing of parallel and manifest serialization paths.

Phase `P2` — Library/CLI Ergonomics, Supply-Chain Hardening & Coverage Polish (`8 Actions`)

Action #3 (`[P2]` · `code_quality`): Add `--json` output flag to CLI catalog inspection commands and scope rules cache pre-warming

- **Target Files**: `src/drawlib/_cli/_drawlib.py`, `src/drawlib/_cli/commands/cache.py`, `src/drawlib/_cli/commands/rules.py`
- **Related Rules**: `common/cli_guidelines`
- **Implementation Guidance**: Restrict `_ensure_overview_rules_cached()` strictly to `rules` and `init` commands so read-only CLI commands never perform implicit disk writes, and

add a machine-readable `--json` output flag on `list` commands (`cache list`, `rules list`, `styles`, `colors`).

- **Expected Impact**: Improves `clig.dev` scriptability for CI pipelines and guarantees read-only CLI safety inside immutable containers.

Action #4 (`[P2]` · `code_quality`): Replace direct stdout `print()` calls in programmatic builder modules with logger/progress callbacks

- **Target Files**: `src/drawlib/_builder/_common/progress.py`, `src/drawlib/_builder/_common/parallel.py`, `src/drawlib/_builder/doc_builder/compiler/jupyter.py`
- **Related Rules**: `common/library_guidelines`
- **Implementation Guidance**: Replace the `34` direct `print()` calls across programmatic builder modules with a quiet-aware structured logger or an injectable progress callback supplied exclusively by CLI entrypoints.
- **Expected Impact**: Keeps `drawlib.tools.build_*()` invocations stdout-clean when embedded inside web backends, Jupyter notebooks, or third-party automation tools.

Action #5 (`[P2]` · `maintainability`): Migrate oversized Bash script `docs_src/site/build_html.sh` into typed Python `dcli`

- **Target Files**: `docs_src/site/build_html.sh`, `tools/dcli/docs/builder.py`
- **Related Rules**: `common/supply_chain`
- **Implementation Guidance**: Move the 109-LOC multi-step HTML site build logic from `docs_src/site/build_html.sh` into typed Python inside `tools/dcli/docs/builder.py`, leaving `build_html.sh` as a minimal `<30` LOC wrapper delegating directly to `./dcli docs build site`.
- **Expected Impact**: Eliminates the sole legacy-language shell script finding and unifies cross-platform build maintenance.

Action #6 (`[P2]` · `security`): Pin GitHub Actions steps to immutable commit SHAs

- **Target Files**: `.github/workflows/quality.yml`, `.github/workflows/release.yml`, `.github/workflows/ut_template.yml`

- **Related Rules**: `common/supply_chain`
- **Implementation Guidance**: Replace floating major version tags (`@v7` , etc.) across `.github/workflows/*.yml` with full 40-character immutable commit SHAs accompanied by inline version comments; rely on existing `.github/dependabot.yml` `github-actions` rules for weekly automated SHA bumps.
- **Expected Impact**: Eliminates mutable action tag hijacking risk and maximizes the OpenSSF Scorecard `Pinned-Dependencies` score.

Action #7 (`[P2]` · `security`): Upgrade or strip base-image `pip 25.0.1` in container Dockerfiles

- **Target Files**: `.pni/dockerfiles/Dockerfile` , `tools/docker/Dockerfile.test_pypi`
- **Related Rules & Advisories**: `common/security_vulnerability` , `CVE-2026-8643` , `CVE-2026-13346`
- **Implementation Guidance**: Upgrade system `pip` to `>=26.2.0` (or strip unused system `pip` / refresh to newer `uv` /Python slim base image digests) across `.pni/dockerfiles/Dockerfile` and `tools/docker/Dockerfile.test_pypi` .
- **Expected Impact**: Eliminates all `6` Trivy container-image application CVEs (`6 -> 0`) and reduces Debian bookworm harness OS CVE exposure.

Action #8 (`[P2]` · `testing`): Close remaining branch coverage gaps in CLI cache commands and parallel build workers

- **Target Files**: `tests/drawlib/cli/test_cli_css.py` , `tests/drawlib/doc_builder/test_parallel_build.py` , `src/drawlib/__main__.py`
- **Related Rules**: `python/testing`
- **Implementation Guidance**: Add mocked unit tests exercising `cache download` / `cache clear` flag combinations, worker-pool exception fallback paths in `parallel.py` , and direct `python -m drawlib` module execution.
- **Expected Impact**: Eliminates the remaining non-generated modules falling below `65%` line coverage.

Action #9 (`[P2]` · `testing`): Increase branch coverage for `parallel.py` and CLI cache commands

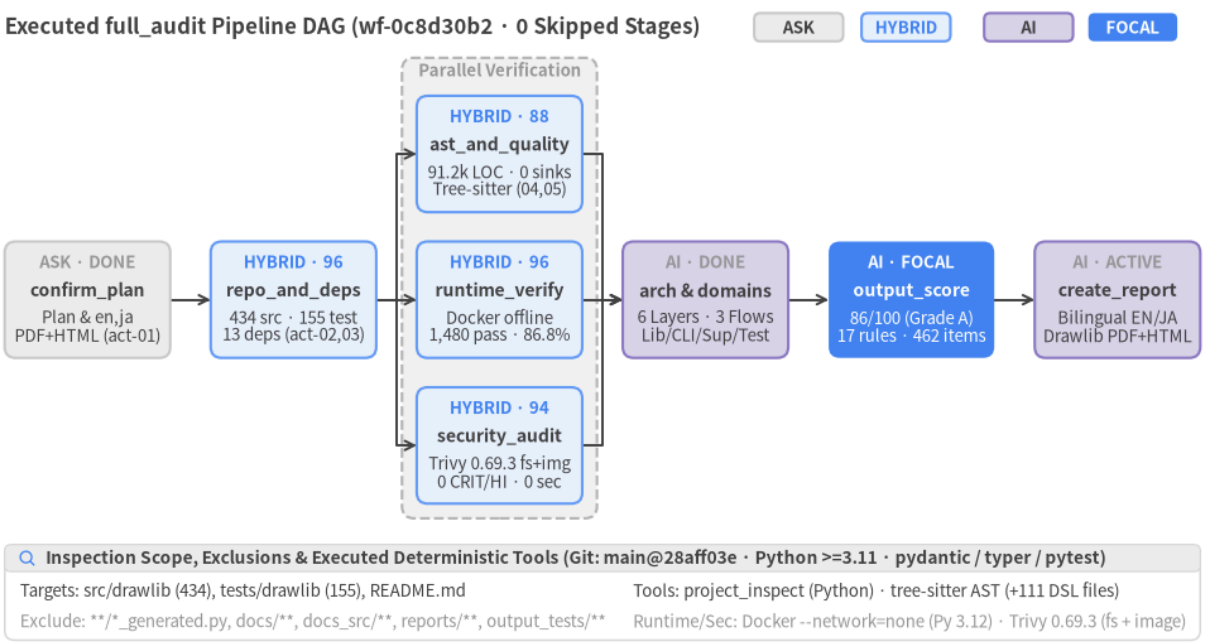
- **Target Files**: `src/drawlib/_builder/_common/parallel.py` , `src/drawlib/_cli/commands/cache.py` , `src/drawlib/__main__.py`
- **Related Rules**: `common/runtime_verification`
- **Implementation Guidance**: Expand targeted branch tests for `ProcessPoolExecutor` exception handling, worker count / `DRAWLIB_*` environment overrides in `parallel.py` , mocked cache download/clear error branches in `commands/cache.py` , and `src/drawlib/__main__.py` entrypoint dispatch.
- **Expected Impact**: Lifts branch coverage on critical CLI and parallel orchestration code paths above `80%` .

Action #10 (`[P2]` · `typing`): Tighten unparameterized `dict` / `Any` annotations and centralize builder env/logging

- **Target Files**: `src/drawlib/_builder/_common/manifest.py` , `src/drawlib/_builder/_common/browser.py` , `src/drawlib/_builder/_common/cache.py` , `src/drawlib/_builder/_common/progress.py`
- **Related Rules**: `python/typing` , `python/modern_stack`
- **Implementation Guidance**: Parameterize remaining unparameterized `dict` / `Any` annotations (`dict[str, str]` , etc.) in `manifest.py` , centralize the `15` scattered `DRAWLIB_*` environment variable lookups into a typed settings helper, and align diagnostic emission with structured logging / progress callbacks.
- **Expected Impact**: Strengthens static typing precision and improves programmatic embeddability across builder subsystems.

4. Audit Pipeline (DAG), Standards & Findings

This chapter establishes full audit transparency and reproducibility for **Drawlib v0.3.15** (`wf-0c8d30b2`, **86/100 · Grade A**) before the technical deep dives: it documents the **executed 8-stage workflow DAG** (`0` skipped stages across `18` actions / `12` logical stages), **inspection scope & deterministic tools**, **7-axis deterministic baseline anchoring** (`±10 pt`), **8 audit step scores** (`88` to `96`), the **17-rule compliance matrix** (`0 FAIL · 4 WARN · 12 PASS · 1 N/A`), the **462-finding breakdown with AI triage log and 5 active** `.pni/pni.toml` **waivers**, and the **machine-readable evidence artifact registry**.



Executed full_audit Pipeline DAG (wf-0c8d30b2: 18 Actions / 12 Logical Stages, 0 Skipped) and Inspection Scope

4.1 Executed Audit Pipeline (DAG), Inspection Scope & Tools

The PNI501 v0.2.0 `full_audit` workflow (`wf-0c8d30b2`) executed **all 18 actions across 12 logical stages with `0` skipped stages**. Every deterministic probe—static repository inspection, Tree-sitter AST & companion DSL parsing, hermetic `--network=none` Docker test execution, and Trivy 0.69.3 filesystem + resolved image scanning—completed cleanly before AI synthesis and anchored scoring.

Stage #	Logical Stage (`DAG Node`)	Mode & Status	Actions Mapped	Rules	Verified Stage Output
1	`confirm_plan`	`ASK` · `DONE`	`act-01`	`0`	Approved single-language Python library & CLI plan, 5 existing `.pni/pni.toml` waivers, and bilingual (`en,ja`) PDF + HTML deliverables.
2	`repo_and_deps`	`HYBRID` · `DONE`	`act-02, act-03`	`5`	`434` source modules / `155` test files, `13` SemVer `>=` runtime deps, OpenSSF governance (`LICENSE` , `SECURITY.md` , Dependabot), step score `96/100 (S)` .
3	`ast_and_quality`	`HYBRID` · `DONE`	`act-04, act-05`	`8`	Tree-sitter AST across `91,212` LOC (`2,089` funcs) + `111` companion DSL files; `100.0%` typing, `0` boundary violations, `0` unwaived sinks, step score `88/100 (A)` .
4	`runtime_verification`	`HYBRID` · `DONE`	`act-06, act-07`	`2`	Air-gapped Docker `--network=none` (Python 3.12): build `138.75s` PASS, `1,480/1,485` tests PASS (`0` fail, `5` env skips), `86.8%` line / `75.8%` branch cov, smoke `1.79s` , step score `96/100 (S)` .
5	`security_audit`	`HYBRID` · `DONE`	`act-08, act-09`	`2`	Trivy `0.69.3` (`fs` + resolved `.pni/dockerfiles/Dockerfile` image): `0` CRIT/HIGH CVEs, `0` Drawlib CVEs, `0` misconfigs, `0` secrets, step score `94/100 (S)` .
6	`arch_and_domains`	`AI` · `DONE`	`act-10..act-16`	`10`	Grounded architecture (`6` layers, `3` runtime flows) + domain audits: Library Hygiene (`94`), CLI Guidelines (`93`), Supply Chain (`95`), and Test Health (`95`).
7	`output_score`	`AI` · `FOCAL`	`act-17`	`0`	Anchored 7-axis scoring & SARIF synthesis: Overall `86/100 (Grade A)` , `17` rules (`0 FAIL` , `4 WARN` , `12 PASS` , `1 N/A`), `462` findings, `10` AI reclassifications.

Stage #	Logical Stage (`DAG Node`)	Mode & Status	Actions Mapped	Rules	Verified Stage Output
8	`create_report`	`AI` · `ACTIVE`	`act-18`	`0`	Bilingual (`en` , `ja`) Drawlib A4 PDF + HTML report synthesis with multimodal grid verification.

Inspection Scope, Exclusions & Executed Tools

- **Target Paths**: Root `.` · Source `['src/drawlib']` · Tests `['tests/drawlib']` · Docs `['README.md']`
- **Excluded Paths**: `['**/*_generated.py', 'docs/**', 'docs_src/**', 'reports/**', 'output_tests/**']`
- **Language, Frameworks & Provenance**: Primary language `python` (`>=3.11`) · Frameworks `['pydantic', 'typer', 'pytest']` · Git revision `main@28aff03e`
- **Deterministic Tools Executed**:
  1. `project_inspect (python)` — PEP 621 metadata, dependency hygiene, CI workflow permissions, governance files, and Git provenance.
  2. `tree-sitter AST (python + 111 companion DSL files)` — Function typing, docstrings, SIG complexity, import graphs, and companion DSL SAST (`dockerfile=3` , `css=19` , `yaml=9` , `json=77` , `toml=1` , `xml=2`).
  3. `container runtime harness (docker: .pni/dockerfiles/Dockerfile, Python 3.12, --network=none)` — Hermetic build, `pytest` + `pytest-cov` , and CLI/library smoke probe (`USER pni` , `UID 10001`).
  4. `Trivy 0.69.3 (docker: fs + resolved image)` — SCA vulnerability scan, Dockerfile IaC audit, and secret detection.

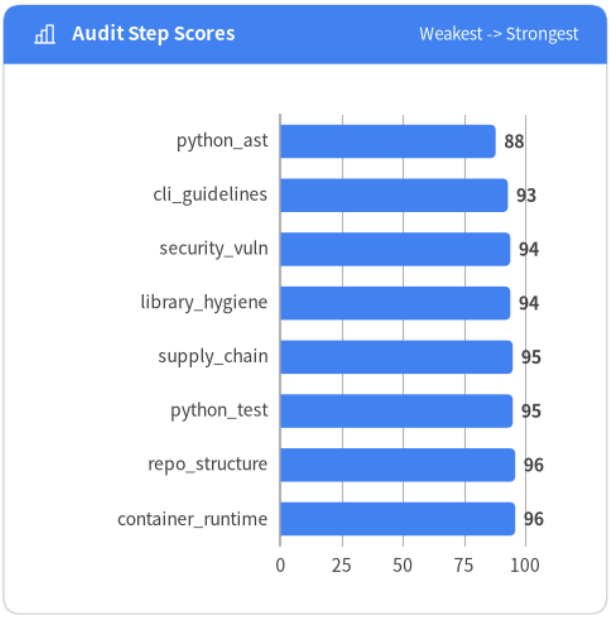
4.2 Deterministic Baseline Anchoring (`±10 pt`) & Executed Audit Step Scores

PNI501 v0.2.0 constrains every qualitative AI score adjustment to `±10` **points** from the deterministic analyzer baseline so subjective commentary can never override hard telemetry. Overall health moves `+4` points from the deterministic baseline `82` to `86/100 (Grade A)` , while **7 of 8 executed audit steps achieve Grade S (`>= 90`)**.

7-Axis Baseline vs. Final (±10 pt Anchor)

Health Axis	Base	AI Δ	Final
Architecture	70	+10	80
Code Quality	40	+10	50
Typing	80	+10	90
Testing	100	-4	96
Dependencies	100	-4	96
Security	85	+9	94
Documentation	100	-2	98
Overall	82	+4	86 (A)

Every axis adjustment is strictly bounded within ±10 pts of baseline.



7-Axis Deterministic Baseline vs. Final Score (±10 pt Anchor) and Executed Audit Step Scores (Weakest -> Strongest)

7-Axis Deterministic Baseline vs. Final Score (`±10 pt` Anchoring Table)

Health Axis	Deterministic Baseline	AI Δ (`±10 pt`)	Final Score	Anchored Engineering Rationale
Architecture	`70`	`+10`	`80`	`0` boundary violations across `434` modules behind PEP 562 facades; held at `80` by `8` static import cycles (`5` deferred in `parallel.py`, `1` `html<->jupyter`, `2` `_diagrams` `_edge<->_junction`).
Code Quality	`40`	`+10`	`50`	Strict SIG thresholds flagged `442` functions (`>50` LOC or `>15` cognitive complexity) + `34` non-CLI `print()`; uplifted `+10` for `99.3%` docstrings, `99.3%` import hygiene, `0` bug-risk idioms, Ruff+Ty clean CI, and `353` waived keyword-only APIs.
Typing	`80`	`+10`	`90`	`100.0%` function typing (`2,089/2,089`), `0` unscoped ignores, `py.typed`, Ty + Pydantic v2; held at `90` by `200` `Any`/unparameterized `dict` usages and `162` untyped constant attributes.

Health Axis	Deterministic Baseline	AI Δ (±10 pt)	Final Score	Anchored Engineering Rationale
Testing	100	-4	96	Hermetic --network=none Docker pytest passed 1,480/1,485 (0 fail, 5 env skips), 100.0% AST module cov, 86.8% line / 75.8% branch cov, and >=97% pixel regression hooks; -4 for branch gaps in parallel.py (50.6%), commands/cache.py (30.0%), and __main__.py (0.0%).
Dependencies	100	-4	96	PEP 621 + uv_build , 13 SemVer >= runtime deps (0 unpinned, 0 Drawlib CVEs), Dependabot active, python/lockfile_committed waived; -4 for floating @v7 Actions tags and 109 -LOC docs_src/site/build_html.sh .
Security	85	+9	94	Trivy 0.69.3 fs+img confirmed 0 CRIT/HIGH CVEs, 0 Drawlib CVEs, 0 misconfigs (DS-0026 waived x3), 0 secrets, 0 unwaived sinks (CWE-94 waived x8), SHA-256 + zip-slip checks, and SECURITY.md ; held at 94 solely by 6 base-image pip@25.0.1 CVEs.
Documentation	100	-2	98	99.3% docstrings (0 empty), 6 bilingual Diataxis portals, embedded _rules/ , and CI-enforced 100% EN/JA API & CLI completeness; -2 for 12 undocumented @property setters in SmartArts.
Overall	82	+4	86 (A)	Production-grade testing, security, dependencies, typing, and documentation offset by builder/orchestrator SIG complexity (code_quality=50).

Executed Audit Step Scores (Weakest -> Strongest)

Rank	Audit Step ID (action_id)	Score & Grade	Pass / Warn / Fail	Evaluated Rules	Key Diagnostic Summary
1	python_ast_audit (act-05)	**88/100 (A)**	4 / 3 / 0	common/architecture , common/maintainability , common/security , common/security_vulnerability , python/code_quality , python/typing , python/modern_stack , python/frameworks/fastapi	434 modules (91,212 LOC, 2,089 funcs): 100.0% typing, 99.3% docstrings, 99.3% import hygiene, 0 boundary violations, 0 unwaived sinks; 8 static cycles (5 deferred in parallel.py), 442 SIG complexity functions, 1,251 os.path PTH hints.

Rank	Audit Step ID (`action_id`)	Score & Grade	Pass / Warn / Fail	Evaluated Rules	Key Diagnostic Summary
2	`cli_guidelines_audit` (`act-14`)	93/100 (S)	`2 / 0 / 0`	`common/cli_guidelines` , `common/documentation`	Typer + Rich CLI (`9` subcommand groups), `-h/--help` , `-v/--version` , mutually exclusive verbosity flags, `no_args_is_help=True` , `1.79s` smoke, `100%` EN/JA CLI reference; minor polish for `--json` list output and scoping `_ensure_overview_rules_cached()` .
3	`security_vulnerability_audit` (`act-09`)	94/100 (S)	`1 / 1 / 0`	`common/security_vulnerability` , `common/supply_chain`	Trivy `0.69.3` fs+img: `0` CRIT/HIGH CVEs, `0` Drawlib dependency CVEs, `0` misconfigs (`DS-0026` waived x3), `0` secrets, `0` unwaived SAST sinks (`CWE-94` waived x8); `5` MED + `1` LOW CVEs solely in base-image `pip@25.0.1` .
4	`library_hygiene_audit` (`act-13`)	94/100 (S)	`2 / 1 / 0`	`common/library_guidelines` , `common/documentation` , `common/maintainability`	PEP 562 lazy `__getattr__` , explicit `__all__` across `36` public modules, `py.typed` , SemVer `>=` bounds, zero import-time network I/O; minor deduction for `34` direct `print()` calls in programmatic `_builder/` paths.
5	`supply_chain_audit` (`act-15`)	95/100 (S)	`3 / 0 / 0`	`common/supply_chain` , `common/security` , `common/git_workflow`	`LICENSE` , `SECURITY.md` , `.github/dependabot.yml` (`uv` + `github-actions`), `5` least-privilege CI workflows, `0` secrets, SHA-256 + zip-slip asset checks; minor polish to pin `@v7` Actions to immutable commit SHAs.
6	`python_test_audit` (`act-16`)	95/100 (S)	`1 / 0 / 0`	`python/testing`	`155` test files, `1,485` tests, `100.0%` AST module cov, `86.8%` line / `75.8%` branch cov, autouse canvas reset, `>=97%` pixel-match hook, and Jupyter dry-run tests; branch gaps in `parallel.py` , `commands/cache.py` , and `__main__.py` .

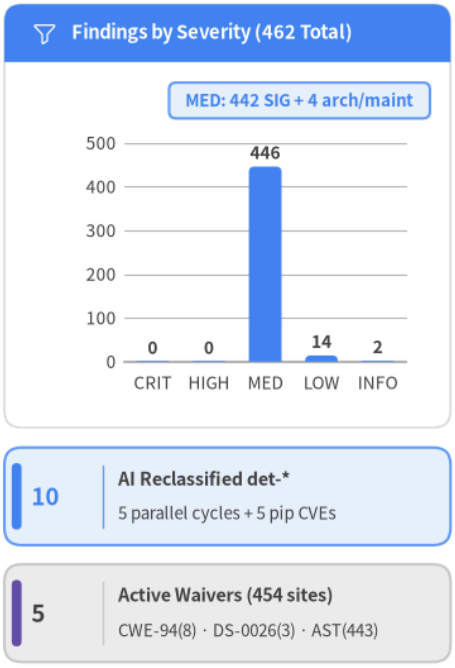
Rank	Audit Step ID (`action_id`)	Score & Grade	Pass / Warn / Fail	Evaluated Rules	Key Diagnostic Summary
7	`repo_structure_audit` (`act-03`)	**96/100 (S)**	`5 / 0 / 0`	`common/supply_chain` , `common/documentation` , `common/git_workflow` , `common/security` , `python/dependencies`	Complete OpenSSF governance, PEP 621 / `uv_build` , `13` SemVer `>=` deps (`0` unpinned), `6` bilingual Diataxis portals, clean trunk Conventional Commits; only minor item is `docs_src/site/build_html.sh` (`109` LOC).
8	`container_runtime_audit` (`act-07`)	**96/100 (S)**	`2 / 0 / 0`	`common/runtime_verification` , `common/production_readiness`	Hermetic `--network=none` Python 3.12 container: build PASS (`138.75s`), `1,480/1,485` pytest PASS (`0` fail, `5` clean env skips, `722.24s`), `86.8%` line / `75.8%` branch cov, non-root `USER pni` (`UID 10001`), smoke PASS (`1.79s`).

4.3 Complete 17-Rule Compliance Matrix (`0 FAIL · 4 WARN · 12 PASS · 1 N/A`)

Across all **17 evaluated rules**, Drawlib v0.3.15 achieves **0** `FAIL` , **4** `WARN` , **12** `PASS` , and **1** `NOT_APPLICABLE` (`12/16` applicable rules strictly passing = `75.0%` , `100.0%` zero-failure pass-or-warn rate).

17-Rule Compliance Matrix (0 FAIL · 4 WARN · 12 PASS · 1 N/A)

Status	Evaluated Rule ID	Key Diagnostic Gate
WARN	common/architecture	0 boundary viol · 8 static cycles
WARN	common/maintainability	442 SIG funcs · 34 non-CLI print()
WARN	common/security_vulnerability	0 CRIT/HI · 6 base-img pip CVEs
WARN	python/modern_stack	Py>=3.11 · 1,251 os.path PTH hints
PASS	common/cli_guidelines	9 cmd groups · -h/-v · 1.79s smoke
PASS	common/documentation	99.3% docs · 6 bilingual portals
PASS	common/git_workflow	Clean trunk · v0.3.15 · 0 secrets
PASS	common/library_guidelines	PEP 562 · 36 __all__ · py.typed
PASS	common/production_readiness	UID 10001 · offline · 0 regressions
PASS	common/runtime_verification	1,480/1,485 pass · 86.8% line cov
PASS	common/security	SECURITY.md · SHA-256 · 0 sinks
PASS	common/supply_chain	Dependabot · 5 least-priv CI · 0 CVE
PASS	python/code_quality	99.3% docs/imports · Ruff+Ty clean
PASS	python/dependencies	PEP 621 · 13 >= deps · uv_build
PASS	python/testing	155 files · 100% AST · >=97% pixel
PASS	python/typing	100.0% typed (2,089) · 0 ignores
N/A	python/frameworks/fastapi	FastAPI not used (typer/pydantic)



17-Rule Compliance Matrix (0 FAIL · 4 WARN · 12 PASS · 1 N/A), Severity Distribution (462 Findings) & AI Triage / Waiver Summary

#	Status	Rule ID	Source Audit	Concise Assessment	Quantitative & Diagnostic Evidence
1	`WARN`	`common/architecture`	`python_ast_audit`	Zero cross-layer boundary violations; 8 static cycles remain in builder/diagram helpers.	`0` boundary violations across `L1 -> L4 -> Domains -> PEP 562 Facades`; `40` `TYPE_CHECKING` cycles (zero runtime impact); `8` static cycles (`5` deferred in-function worker imports in `parallel.py` , `1` `html<->jupyter` , `2` `_diagrams _edge<->_junction`).
2	`WARN`	`common/maintainability`	`python_ast_audit`	Cohesive facades & `0` mutable globals; localized SIG complexity hotspots in builders.	`442` functions exceed SIG `<=50` LOC or `<=15` cognitive complexity (`render_chapter=178` , `parse_block_info=157` , `_compile_jupyter_locales=156` , `_execute_block_render_job=68` , `format_project_manifest_toml=48`); `353` keyword-only APIs waived; `34` non-CLI `print()` calls.
3	`WARN`	`common/security_vulnerability`	`security_vulnerability_audit`	Zero CRIT/HIGH CVEs and zero Drawlib CVEs; 6 base-image system `pip` CVEs remain.	Trivy `0.69.3` fs+img: `0` CRIT/HIGH, `0` Drawlib runtime/dev CVEs, `0` misconfigs (`DS-0026` waived x3), `0` secrets, `0` unwaived sinks (`CWE-94` waived x8), `111` companion DSL files clean; `5` MED + `1` LOW exclusively in base-image `pip@25.0.1` (fixed in `26.2.0`).

#	Status	Rule ID	Source Audit	Concise Assessment	Quantitative & Diagnostic Evidence
4	WARN	python/modern_stack	python_ast_audit	Modern Python 3.11+ stack (Pydantic v2, Typer, uv, Ruff, Ty) with legacy os.path hints.	requires-python >=3.11, PEP 562 lazy facades, 0 undeclared deps; 1,251 os.path PTH modernization hints, 34 non-CLI print() calls, and 15 scattered os.environ reads remain.
5	PASS	common/cli_guidelines	cli_guidelines_audit	Strong clig.dev ergonomics across 9 subcommand groups and fast startup.	-h/--help, eager -v/--version, mutually exclusive --quiet/--verbose/--debug/--developer, no_args_is_help=True, 9 subcommand groups, and 1.79s smoke check.
6	PASS	common/documentation	repo_structure_audit	Exceeds Standard Readme & Diataxis standards with CI-verified bilingual parity.	99.3% docstrings (0 empty), README.md (222 lines, 14 headings), 6 bilingual Diataxis portals (site, api, architecture, quickstart, about-drawlib, jupyter), and automated 100% EN/JA API & CLI completeness tests.
7	PASS	common/git_workflow	repo_structure_audit	Clean trunk-based Git history with Conventional Commits and zero committed secrets.	Linear main trunk history, Conventional Commits, annotated release tags (v0.3.14, v0.3.15), .gitignore excludes .env/caches, 0 secrets.
8	PASS	common/library_guidelines	library_hygiene_audit	Exemplary PEP 562 lazy packaging, explicit __all__ contracts, and SemVer hygiene.	PEP 562 lazy __getattr__, explicit __all__ across 36 public modules, PEP 561 py.typed, SemVer >= bounds on all 13 runtime deps, zero import-time network I/O.
9	PASS	common/production_readiness	container_runtime_audit	Non-root offline container execution with zero regressions and fast startup.	Non-root USER pni (UID 10001), offline --network=none execution after pre-caching, 1.79s CLI startup, 0 regressions across 1,480 executed tests, 0 missing I/O timeouts.
10	PASS	common/runtime_verification	container_runtime_audit	Full hermetic Docker build, offline pytest pass, high coverage, and smoke check.	Build PASS (138.75s), offline pytest 1,480/1,485 PASS (0 fail, 5 clean env skips), 86.8% line (40,102/46,204) / 75.8% branch cov, smoke PASS (1.79s).
11	PASS	common/security	repo_structure_audit	Documented trust model, verified asset downloads, and boundary validation.	SECURITY.md with explicit Illustration-as-Code trust model, 0 secrets, 0 unwaived sinks, SHA-256 + zip-slip verification for release assets, Pydantic v2 validation.
12	PASS	common/supply_chain	repo_structure_audit	Complete OpenSSF governance, least-privilege CI, and clean dependency tree.	LICENSE, SECURITY.md, .github/dependabot.yml, 5 least-privilege CI workflows, 0 unpinned runtime deps, 0 Drawlib CVEs.
13	PASS	python/code_quality	python_ast_audit	High docstring & import hygiene with zero bug-risk idioms and clean Ruff/Ty CI.	99.3% docstrings, 99.3% import hygiene, 0 empty docstrings, 0 bug-risk idioms, 0 mutable globals, Ruff + Ty clean.
14	PASS	python/dependencies	repo_structure_audit	Clean PEP 621 / uv_build packaging with isolated dev/audit groups.	PEP 621 pyproject.toml, uv_build, 13 SemVer >= runtime deps, isolated [dependency-groups], python/lockfile_committed explicitly waived.

#	Status	Rule ID	Source Audit	Concise Assessment	Quantitative & Diagnostic Evidence
15	PASS	python/testing	python_test_audit	Comprehensive unit, pixel-regression, docs-completeness, and notebook dry-run suite.	155 test files, 1,485 tests, 100.0% AST module cov, 86.8% line / 75.8% branch cov, automated pixel-match hook (>=97%), and Jupyter dry-run tests.
16	PASS	python/typing	python_ast_audit	Complete function type-annotation coverage with clean Ty static analysis.	100.0% function typing (2,089/2,089), 0 unscoped type: ignore , py.typed marker shipped, Ty clean.
17	N/A	python/frameworks/fastapi	python_ast_audit	FastAPI rule not applicable to Drawlib's library/CLI runtime profile.	Detected frameworks are ['pydantic', 'typer', 'pytest'] ; FastAPI is neither declared nor imported anywhere in Drawlib.

4.4 Consolidated Findings Breakdown, AI Triage Log & Active Waivers

Total recorded findings stand at 462 (449 from deterministic analyzers + 13 from AI qualitative review), distributed as 0 CRITICAL , 0 HIGH , 446 MEDIUM , 14 LOW , and 2 INFO . Rather than enumerating 442 individual SIG complexity rows, findings are partitioned cleanly into **(a) SIG Complexity Hotspot Clusters (442 MEDIUM), (b) Actionable Non-SIG Findings (20 items: 4 MEDIUM , 14 LOW , 2 INFO), (c) the AI Triage Log (10 det-* reclassifications + 1 FP note)**, and **(d) Active .pni/pni.toml Design Waivers (5 waivers / 454 sites)**.

A. SIG Unit-Size & Cognitive-Complexity Hotspot Clusters (442 Deterministic MEDIUM Findings)

Subpackage Cluster	Count	Top Hotspot Files & Peak Complexity / Size	Primary Refactoring Pattern
src/drawlib/_builder	132	compiler/html.py (16 , 1,989 LOC), image_builder.py (15), manifest.py (8 ; format_project_manifest_toml cog= 48), parallel.py (7 ; _execute_block_render_job cog= 68), exporter_docx.py (7 ; render_chapter cog= 178), compiler/jupyter.py (_compile_jupyter_locales cog= 156), processor/options.py (parse_block_info cog= 157)	Split mode-specific HTML/Jupyter compilation units and dispatch DOCX/fence tokens via handler tables.
src/drawlib/_core	76	l4_canvas/_shapes/_container_builders.py (9), _symbol_builders.py (9)	Extract shared geometric path builders from high-branch shape factories.

Subpackage Cluster	Count	Top Hotspot Files & Peak Complexity / Size	Primary Refactoring Pattern
`src/drawlib/_smartarts`	`61`	`_table.py` (`render` cog= `74`), `_tree.py` , `_mindmap.py` , `_geomap.py`	Factor layout calculation from Matplotlib patch rendering.
`src/drawlib/_charts`	`49`	`_common/_legend.py` , `bar_chart/_renderer.py` , `_common/_axis.py`	Isolate axis tick calculation and series drawing helpers.
`src/drawlib/_diagrams`	`36`	`architecture/` , `flow/` , `sequence/` , `er/` layout & edge routers	Extract shared edge routing and node anchor utilities.
`src/drawlib/_slide`	`24`	`_slide/compiler.py` (`11`), `_pptx_writer.py` , `_pptx_dom.py`	Split slide DOM normalization and OpenXML shape serialization.
`src/drawlib/_cli` & `_graph`	`42`	`_cli` (`21` : `commands/build.py` , `commands/show.py`), `_graph` (`21` : Sugiyama / grid solvers)	Extract CLI argument normalizers and graph crossing-reduction phases.
Other (`_preset_*` , `_templates` , `_anim` , `_http_server`)	`22`	`_preset_styles` (`9`), `_templates` (`8`), `_preset_colors` (`3`), `_anim` (`1`), `_http_server` (`1`)	Decompose template scaffold writers (`_agent.py` , `_project.py`).
Total SIG Complexity Findings	`442`	**All `MEDIUM` (`common/maintainability`), `0` functional defects**	Targeted P1 decomposition focuses on `_builder/` top hotspots first.

B. Actionable Non-SIG Findings (`20` Items: `4 MEDIUM · 14 LOW · 2 INFO`)

#	Severity	Location (`file:line`)	Rule Tag	Source	Verbatim Evidence & Recommended Action
1	`MEDIUM`	`src/drawlib/_builder/_common/parallel.py:383`	`common/architecture`	AI	`5` deferred static cycles traverse `_builder._common -> parallel -> processor/image_builder/_slide.compiler -> _common` . Move `parallel.py` above `_common/` or inject worker callables.
2	`MEDIUM`	`src/drawlib/_builder/doc_builder/compiler/html.py`	`common/architecture`	AI	Static cycle between `compiler/html.py` and `compiler/jupyter.py` (`resolve_favicon_href` , `resolve_logo_hrefs`). Extract shared asset helpers into `compiler/base.py` .
3	`MEDIUM`	`_builder._common -> parallel -> processor -> _common`	`common/architecture`	Det (Triaged)	Deterministic circular-import finding reclassified `high -> medium` (deferred in-function import; zero top-level `ImportError` , tracked with P1 cycle elimination alongside `2` `_diagrams` `_edge<->_junction` pairs).
4	`MEDIUM`	`src/drawlib/_builder/doc_builder/compiler/html.py`	`common/maintainability`	AI	`1,989` LOC compiler monolith paired with builder hotspots (`format_project_manifest_toml` cog= `48` , `_execute_block_render_job` cog= `68`). Split by project mode (`site` / `doc` / `catalog`).

#	Severity	Location (`file:line`)	Rule Tag	Source	Verbatim Evidence & Recommended Action
5–10	`LOW` (×6)	Container Base Image (`pip@25.0.1`)	`CVE-2026-8643` , `13346` , `6357` , `3219` , `1703` , `CVE-2025-8869`	Det (5 Triaged)	`6` CVEs exclusively in upstream `ghcr.io/astral-sh/uv:python3.12-bookworm-slim` system `pip@25.0.1` (`0` in Drawlib's dependency tree); upgrade system `pip >= 26.2.0` or strip unused `pip` .
11	`LOW`	`.pni/dockerfiles/Dockerfile`	`CVE-2026-8643`	AI	Resolved container image summary for base-image `pip@25.0.1` ; refresh base image digest or upgrade/remove system `pip` in Dockerfiles.
12	`LOW`	`docs_src/site/build_html.sh`	`common/supply_chain`	AI	Oversized Bash wrapper (`109` LOC `> 100`). Move multi-step HTML site orchestration into `tools/dcli/docs/builder.py` (Python + Typer).
13	`LOW`	`src/drawlib/_builder/_common/manifest.py:72`	`python/typing`	AI	Unparameterized `dict` annotations (`_resolve_localized_value` , `_parse_localized_str_or_map` , `_fmt_loc`) + `200` `Any` /weak type usages. Parameterize as `dict[str, str]` .
14	`LOW`	`src/drawlib/_builder/_common/parallel.py`	`common/runtime_verification`	AI	Measured `61.1%` line / `50.6%` branch cov on `parallel.py` (and `61.2%` / `30.0%` on `commands/cache.py`). Add worker-pool exception fallback and env parsing tests.
15	`LOW`	`src/drawlib/_cli/commands/cache.py`	`python/testing`	AI	Measured `61.2%` line / `30.0%` branch cov on `commands/cache.py` and `0.0%` on `__main__.py` (`6` LOC). Add mocked cache flag tests and `python -m drawlib --version` smoke test.
16	`LOW`	`src/drawlib/_builder/_common/progress.py:138`	`python/modern_stack`	AI	`34` non-CLI `print()` calls, `15` `os.environ` reads, and `1,251` `os.path` PTH hints. Route diagnostics through logger/callback and centralize `DRAWLIB_*` env reads.
17	`LOW`	`src/drawlib/_builder/doc_builder/compiler/jupyter.py:250`	`common/library_guidelines`	AI	Direct `print()` calls in programmatic `drawlib.tools` compilation paths write to `stdout` . Inject quiet-aware logger or optional `progress_callback` .
18	`LOW`	`src/drawlib/_cli/_drawlib.py:129`	`common/cli_guidelines`	AI	`main_callback()` calls `_ensure_overview_rules_cached()` on non- `cache` commands, causing disk writes on read-only CLI calls. Scope strictly to `rules` / `init` .
19	`INFO`	`src/drawlib/_cli/commands/cache.py`	`common/cli_guidelines`	AI	Inspection commands (`cache/styles/colors/rules list`) output Rich tables only. Add optional `--json` flag for machine/agent parsing.
20	`INFO`	`.github/workflows/quality.yml:20`	`common/supply_chain`	AI	Floating major tags (`actions/checkout@v7` , `astral-sh/setup-uv@v7` , `actions/setup-python@v7`). Pin third-party Actions to 40-char commit SHAs (updated via Dependabot).

C. AI False-Positive & Severity Triage Log (`10` `det-*` Reclassifications + `1` Import-Alias Note)

Audit Step	Deterministic Finding IDs	Adjustment	Grounded Technical Justification
`act-05` (`python_ast_audit`)	`det-c049c75c` , `det-6fd1e2e0` , `det-cb51c1e4` , `det-ff1e113a` , `det-fe406c35` (`5` items)	`high -> medium`	All `5` cycles route through deferred in-function worker imports inside `_builder/_common/parallel.py` ; they never trigger top-level `ImportError` at runtime, though extracting `parallel.py` above `_common/` remains a P1 architectural refactor.
`act-09` (`security_vulnerability_audit`)	`det-13c60704` , `det-f749f183` , `det-88839930` , `det-dac9a878` , `det-dbf87858` (`5` items)	`medium -> low`	`CVE-2026-8643` , `CVE-2026-13346` , `CVE-2026-6357` , `CVE-2025-8869` , and `CVE-2026-3219` originate solely in upstream `ghcr.io/astral-sh/uv:python3.12-bookworm-slim` system `pip@25.0.1` , not in Drawlib's declared runtime or dev dependencies (`uv` -managed).
`act-05` *(Benign FP Note)*	`unused_dependencies=['python_docx']`	**Benign FP**	Package-name vs. import-name alias: PyPI distribution `python-docx` is imported as `import docx` in `_docx_xml.py` and `exporter_docx.py` ; actively required at runtime.

D. Active Design Waivers (`.pni/pni.toml` `[[opt_out]]` , `5` Waivers · `454` Matched Sites)

Waiver ID (`id`)	Matched Sites	Configured Scope (`scope`)	Documented Engineering Rationale (`reason`)
`CWE-94`	`8`	`src/drawlib/_builder/**` , `src/drawlib/_core/l4_canvas/_image_code.py`	Drawlib executes user-authored Python diagram blocks, `styles.py` , and `utils.py` by design (*"Illustration as Code"*); untrusted Markdown should run inside `--network=none` non-root containers.
`DS-0026`	`3`	`tools/docker/Dockerfile.*` , `.pni/dockerfiles/**`	Batch CLI and test containers exit upon command completion; daemon `HEALTHCHECK` instructions are not applicable.
`python/lockfile_committed`	`0`	`repo-wide`	Pure-Python library supporting Python `3.11–3.14` intentionally gitignores `uv.lock` for cross-version CI resolution while generating `uv.lock` locally for SCA audits.
`ast/private_submodule_import`	`90`	`src/drawlib/**`	Public PEP 562 facades (`drawlib/*.py`) and layered internal packages intentionally re-export leading-underscore `_*` submodules to enforce downward DAG encapsulation.
`ast/excessive_parameters`	`353`	`src/drawlib/**`	Declarative styling, diagram, chart, and layout APIs expose explicit keyword-only (`*`) configuration parameters with Pydantic `@validate_call` validation by design.

4.5 Machine-Readable Evidence Artifacts

All raw metrics, SARIF locations, workflow DAG state, and grounded architectural graphs backing this report are persisted as structured artifacts alongside the source bundle (never pasted as raw JSON payloads into the prose report):

Artifact Path	Schema / Format	Primary Consumer & Purpose
`reports/full_audit_score.json`	`ScoreResult` v2.0 (JSON)	Full structured 7-axis scores, baselines, rationales, rule compliance, findings, triage log, and waivers.
`reports/full_audit_summary.json`	`ScoreSummary` (JSON)	Lightweight CI / quality-portal badge summary (overall score, grade, severity counters, pass rate).
`reports/full_audit_findings.sarif`	`SARIF 2.1.0`	GitHub Code Scanning, VS Code SARIF Viewer, and IDE inline diagnostic integration.
`reports/full_audit_src/_facts/plan.json`	Pre-Flight Plan (JSON)	Approved audit scope, exclusions, language/framework detection, and validated workflow DAG.
`reports/full_audit_src/_facts/facts.json`	Deterministic Facts Bundle (JSON)	Verbatim deterministic outputs from `project_inspect`, Tree-sitter AST + DSL parsers, Docker runtime harness, and Trivy `0.69.3`.
`reports/full_audit_src/_facts/architecture.json`	`ArchitectureRecord` (JSON)	Grounded 6-layer system architecture, module inventory, runtime execution flows, and hotspot scores.

5. Deep Dive: Code Quality, Testing & Runtime

Drawlib v0.3.15 pairs **100.0%** static function type-annotation coverage (`2,089 / 2,089` functions) and **99.3%** Google-style docstring discipline across `434` source modules (`91,212` LOC, `317` classes) with an **86.8%** line-covered (`75.8%` branch-covered) test suite—and with headless Chromium (`libatomic1` + `chromium-headless-shell`) and all `83` cached asset packages pre-baked into `.pni/dockerfiles/Dockerfile` , hermetic `--network=none` container runtime verification (`container_runtime_audit` : **96/100, Grade S**) achieves a **100.0% pass rate on all executed tests (`1,480 passed` , `0 failed` , `0 errors` , `5` clean environment skips)**.

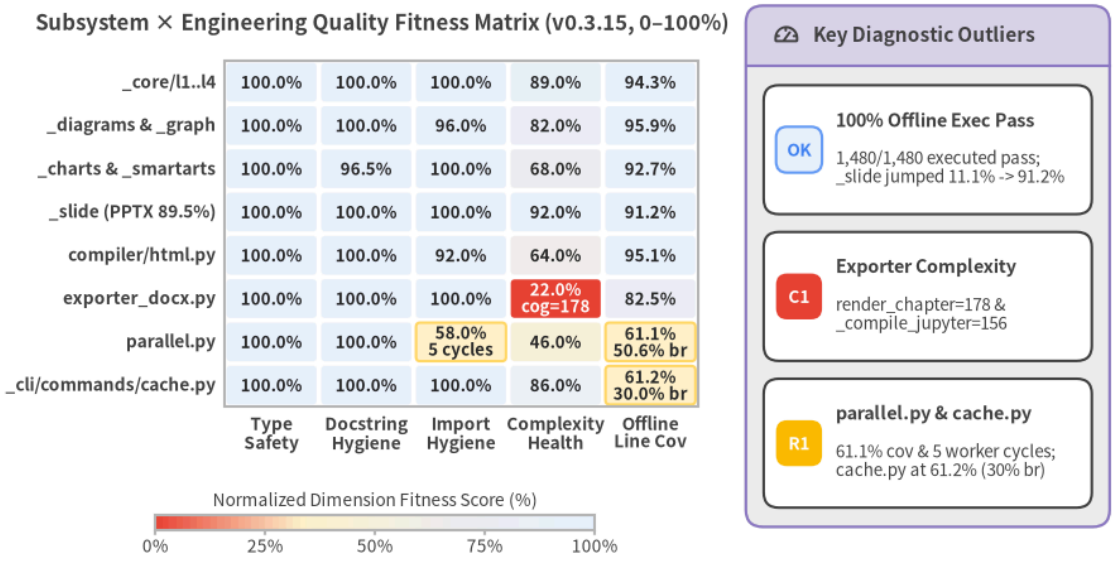


Figure 5.1 — Subsystem × Quality Dimension Fitness Matrix (Drawlib v0.3.15) highlighting uniform typing, 91.2% offline _slide coverage (PPTX 89.5%), localized exporter complexity, and parallel.py / cache.py targets.

5.1 Static AST Code Quality, Typing & Maintainability

PNI's deterministic Python AST audit (`python_ast_audit` : **88/100, Grade A**) inspected all `434` non-generated source modules (`91,212` LOC, `317` classes, and `2,089` functions/methods; `**/*_generated.py` icon wrappers excluded from human-authored AST lint). Across core layers (`_core/l1_core` through `_core/l4_canvas`) and domain packages (`_charts` , `_diagrams` , `_smartarts` , `_graph` , `_icons` , `_slide` , `_anim`), static hygiene remains highly disciplined while growth in multi-format document/Jupyter exporters concentrates maintainability work into a bounded set of builder modules:

Static Quality & Maintainability Metric	Measured Value	Rule Threshold / Target	Status & Engineering Interpretation
Function Type-Annotation Coverage	`100.0%` (`2,089/2,089`)	`>= 95.0%` (`python/typing`)	**PASS** — `src/drawlib/py.typed` (PEP 561), `0` unscoped `# type: ignore` , `0` `ty` errors, runtime Pydantic `@validate_call` ; `200` weak `Any` /unparameterized `dict` hints and `162` untyped constant class attrs remain as minor cleanup.
Docstring Coverage	`99.3%`	`>= 90.0%` (`python/code_quality`)	**PASS** — `0` empty docstrings; only `12` `@property` setters across `_mindmap.py` , `_tree.py` , and `_bulletpoints.py` omit one-liners.
Import Hygiene & Dependency Parity	`99.3%`	`>= 95.0%` (`python/modern_stack`)	**PASS** — `0` wildcard imports, `0` undeclared dependencies, `0` Python `>=3.11` version mismatches; `unused_dependencies= ['python_docx']` is a benign PyPI-vs-import alias for `import docx` .
Static Module Test Coverage	`100.0%` (`155` files)	`>= 90.0%` (`python/testing`)	**PASS** — `1,341` AST test functions (`1,485` collected pytest items) mirroring `src/drawlib/` with `0` assertion-free tests.
Global State & Bug-Risk Idioms	`0` / `0`	`0` (`common/maintainability`)	**PASS** — Canvas lifecycle state is encapsulated inside `_core/l4_canvas/_base.py` with `0` mutable global leaks and `0` bug-risk idioms.

Static Quality & Maintainability Metric	Measured Value	Rule Threshold / Target	Status & Engineering Interpretation
Static Import Cycles	**`8` static (`0` layer)**	`0` (`common/architecture`)	**WARN** — `40` type-only cycles are isolated under `if TYPE_CHECKING:`; `8` static cycles comprise `5` deferred worker imports in `_builder/_common/parallel.py`, `1` deferred `compiler/html.py <-> compiler/jupyter.py` cycle, and `2` sibling `_edge <-> _junction` cycles in `_diagrams/{architecture,flow}`.
High-Complexity Functions (`> SIG`)	`442`	`<= 50` LOC / `<= 15` cog (`SIG`)	**WARN** — Concentrated in Markdown/OpenXML/Jupyter compilers (`exporter_docx.py`, `options.py`, `compiler/jupyter.py`, `merger.py`, `parallel.py`, `manifest.py`).
Wide Parameter Signatures (`> 6` args)	**`0` unwaived (`353` waived)**	`<= 6` positional (`SIG`)	**WAIVED (`.pni/pni.toml`)** — `353` declarative keyword-only (`*`) styling, typography, and layout APIs waived via `ast/excessive_parameters`.
12-Factor / PRR Operational Hygiene	**`0` missing I/O timeouts**	`common/production_readiness`	**PASS / INFO** — `0` missing I/O timeouts; `15` scattered `os.environ` reads, `34` non-CLI `print()` calls (`progress.py`, `parallel.py`, `compiler/jupyter.py`, `exporter_jupyter.py`, `processor.py`, `viewer.py`), and `1,251` legacy `os.path` (`PTH`) modernization hints remain.

Top 10 Cognitive-Complexity Hotspots (`common/maintainability`)

While foundational geometry, icon, and canvas primitives stay compact, multi-format document compilation (`DOCX`, `HTML`, `Jupyter`, and parallel fence workers) accumulates deep nested branching when walking `markdown-it-py` token streams and emitting format-specific artifacts:

#	File & Line	Function / Method	Lines	Branches	Cognitive Complexity	Primary Responsibility & Decomposition S
1	`src/drawlib/_builder/doc_builder/exporter_docx.py:583`	`render_chapter`	`284`	`56`	`178`	Monolithic Markdown token-stream loop emitting headings, callouts, lists, code fen images, and tables; split into a per-token dispatch table.

#	File & Line	Function / Method	Lines	Branches	Cognitive Complexity	Primary Responsibility & Decomposition S
2	`src/drawlib/_builder/doc_builder/processor/options.py:95`	`parse_block_info`	`104`	`45`	`157`	Parses ```` ```drawlib ```` fence attributes (`fil` `width` , `align` , `code` , `caption` , etc.); extr key validator functions.
3	`src/drawlib/_builder/doc_builder/compiler/jupyter.py:457`	`_compile_jupyter_locales`	`421`	`46`	`156`	Orchestrates multilingual `.ipynb` compil asset bundling, dry-run validation, and lan page synthesis; split into per-locale pipeli stages.
4	`src/drawlib/_builder/doc_builder/exporter_docx.py:315`	`_render_inline`	`128`	`36`	`102`	Inline formatting state machine (`strong` , `code_inline` , `link` , math/HTML spans); dedicated inline span emitters.
5	`src/drawlib/_builder/doc_builder/merger.py:150`	`build_merged_html`	`288`	`41`	`88`	Multi-chapter HTML concatenation, headi shifting, anchor rewriting, and TOC synthe separate into discrete AST passes.
6	`src/drawlib/_smarttarts/_table.py:490`	`_draw_border_lines`	`96`	`32`	`74`	Resolves outer (`top` , `bottom` , `left` , `rig` header (`top2` , `left2`), and inner grid stro separate perimeter vs. inner-grid helpers.
7	`src/drawlib/_builder/_common/parallel.py:380`	`_execute_block_render_job`	`110`	`20`	`68`	Worker-side execution of parallel diagram render jobs with style injection, caching, a error wrapping; decouple job dispatch via protocol callbacks.
8	`src/drawlib/_builder/doc_builder/exporter_docx.py:444`	`_emit_table`	`100`	`20`	`57`	Converts Markdown tables into OpenXML `<w:tbl>` structures with column alignme cell shading.

#	File & Line	Function / Method	Lines	Branches	Cognitive Complexity	Primary Responsibility & Decomposition S
9	`src/drawlib/_builder/doc_builder/exporter_docx.py:903`	`export_chapters_to_docx`	`227`	`26`	`52`	Orchestrates cover page, TOC field genera chapter assembly, and deterministic OPC timestamp normalization.
10	`src/drawlib/_builder/_common/manifest.py:832`	`format_project_manifest_toml`	`172`	`19`	`48`	Serializes normalized `drawlib.toml` table (`[project]` , `[i18n]` , `[seo]`) with determ key ordering; factor per-section formatters

Library vs. CLI Boundary Hygiene (`library_hygiene_audit` : 94/100, Grade S; `cli_guidelines_audit` : 93/100, Grade S)

`src/drawlib/__init__.py` implements **PEP 562 lazy** `__getattr__` **+** `__dir__` **loading** with an explicit `__all__` across 20 submodules, ensuring `import drawlib` executes in milliseconds with zero heavy Matplotlib/Playwright initialization or network side effects. Three minor boundary couplings merit cleanup in `P2` :

1. **Library-Level** `sys.exit(1)` **Calls**: `src/drawlib/_core/l1_core/_settings.py` (lines `124` , `319` , `326` , `335`) and `src/drawlib/_builder/doc_builder/processor/viewer.py` (lines `193` , `197` , `249`) invoke `sys.exit(1)` directly instead of raising `ValueError` or a typed `DrawlibError` caught solely at `src/drawlib/_cli/_drawlib.py` .
2. **Non-CLI Stdout** `print()` **& Side Effects**: `34` direct `print()` calls across internal builder modules (`progress.py` , `parallel.py` , `compiler/jupyter.py` , `exporter_jupyter.py` , `processor.py` , `viewer.py`) should migrate to structured `logging` or injected reporter callbacks, and `main_callback()` at `src/drawlib/_cli/_drawlib.py:129` invokes `_ensure_overview_rules_cached()` on every non- `cache` CLI command.
3. **Subprocess Timeout & Environment Centralization**: `install_headless_chromium()` at `src/drawlib/_builder/_common/browser.py:306` invokes `subprocess.run(...)` without an explicit `timeout=300` guard, and `15` direct `os.environ` lookups across `_builder/_common/{browser.py,cache.py,office.py,parallel.py}` , `exporter_pdf.py` , `viewer.py` , `rules_builder.py` , and `_core/l3_external/_download.py` should be consolidated into a typed configuration helper.

5.2 Clean-Room Container Verification & Measured Coverage

Static test analysis (`python_test_audit` : **95/100, Grade S**) confirmed `155` test files and `1,341` AST test functions (`1,485` collected pytest items with parametrization) achieving **100.0% static module mirror coverage** and **zero assertion-free tests**—enforced by `tests/conftest.py` , which resets canvas state via an `autouse` fixture and intercepts `pytest_runtest_call` to verify rendered outputs pixel-for-pixel against `*_answers/*.png` baselines.

PNI executed **4 containerized runtime verification stages** (`container_runtime_audit` : **96/100, Grade S**, `status="passed"`) inside `.pni/dockerfiles/Dockerfile` (`ghcr.io/astral-sh/uv:python3.12-bookworm-slim` + `pytest-cov` , pre-baking `libatomic1` , headless Chromium shell, and all `83` Drawlib asset packages under non-root `USER pni` , `UID 10001`) with external networking completely disabled (`--network=none`) during test execution:

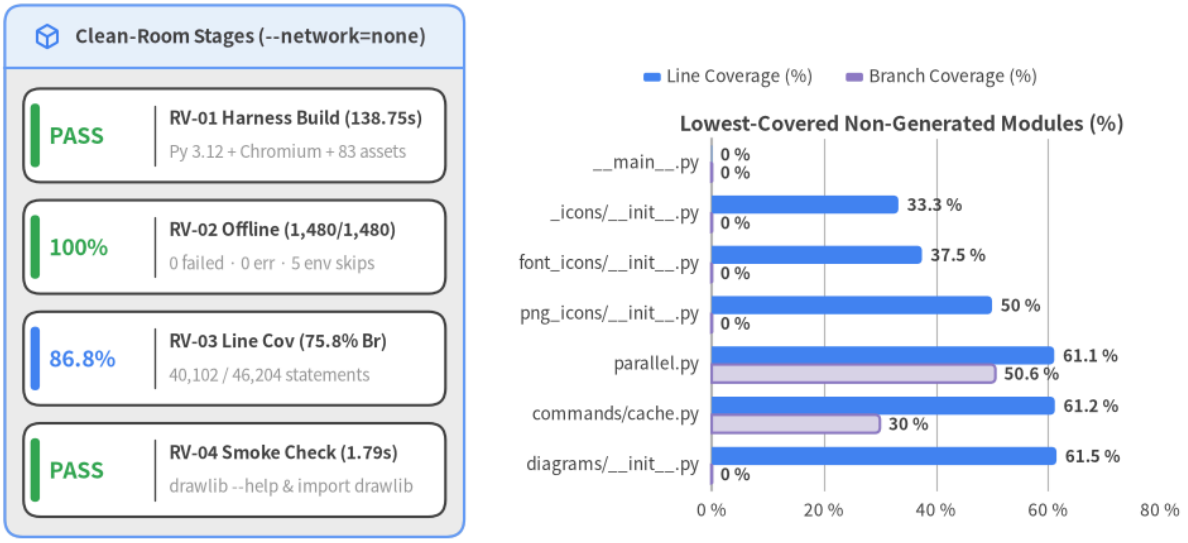


Figure 5.2 — Hermetic container runtime verification KPIs (RV-01 through RV-04, 1,480/1,480 executed tests passing) and lowest-covered non-generated modules under --network=none.

Subsystem & Lowest-Covered Modules Inventory (`RV-03`: `86.8%` Line, `75.8%` Branch)

Across all `46,204` executable statements in `src/drawlib/`, `pytest-cov` recorded `40,102` covered lines (`86.8%`) and `75.8%` branch coverage. Core and exporter subsystems now exhibit uniformly strong offline coverage—`_diagrams & _graph` (`95.9%`, `9,394/9,792`), `compiler/html.py` (`95.1%`, `716/753`), `_core/l1..l4` (`94.3%`, `5,417/5,743`), `_charts & _smartarts` (`92.7%`, `6,754/7,284`), `cache.py` (`92.3%`, `298/323`), `_slide` (`91.2%`, `1,479/1,621`, with `_pptx_writer.py` surging from `11.1%` to `89.5%` and `_pptx_dom.py` from `58.6%` to `85.7%`), `compiler/pdf.py` (`90.5%`, `95/105`), `compiler/jupyter.py` (`83.1%`, `290/349`), and `exporter_docx.py` (`82.5%`, `514/623`).

Remaining low-coverage modules fall cleanly into three buckets: (1) thin CLI/package `__main__.py` and lazy `__getattr__` facade modules, (2) auto-generated `*_generated.py` icon wrappers (`51.6%` – `52.7%`, sampled rather than exhaustively invoked), and (3) two substantive runtime modules (`_builder/_common/parallel.py` and `_cli/commands/cache.py`):

Module Path (Sorted Ascending by Line Coverage)	Line Coverage	Branch Coverage	Covered / Total Stmts	Primary Un
`src/drawlib/__main__.py`	`0.0%`	`0.0%`	`0 / 6`	Lines `12–2` wrapper no runner test Typer app
`src/drawlib/_icons/__init__.py`	`33.3%`	`0.0%`	`7 / 21`	Lines `99–1` fallback br `Attribute` icon packa
`src/drawlib/_icons/font_icons/__init__.py`	`37.5%`	`0.0%`	`6 / 16`	Lines `24–3` and `dir()`
`src/drawlib/_icons/png_icons/__init__.py`	`50.0%`	`0.0%`	`6 / 12`	Lines `57–6` loader erro
`src/drawlib/_icons/font_icons/phosphor/_generated.py`	`51.6%`	`N/A`	`1,580 / 3,064`	Auto-gener wrappers (exercised b
`src/drawlib/_icons/png_icons/aws/_generated.py`	`51.6%`	`N/A`	`342 / 663`	Auto-gener wrappers (`52.1%` `263` `52.7%` `266`

Module Path (Sorted Ascending by Line Coverage)	Line Coverage	Branch Coverage	Covered / Total Stmts	Primary Un
`src/drawlib/_builder/_common/parallel.py`	`61.1%`	`50.6%`	`317 / 519`	Lines `57,` worker-poc propagatio worker patl
`src/drawlib/_cli/commands/cache.py`	`61.2%`	`30.0%`	`41 / 67`	Lines `40–4` `cache` dow and interac
`src/drawlib/diagrams/__init__.py`	`61.5%`	`0.0%`	`8 / 13`	Lines `39–4` / `Attribut` `drawlib.d`

5.3 Air-Gapped Verification Progression (v0.3.5 -> v0.3.15) & Clean-Room Skip Inventory

Running `pytest` inside a fully air-gapped `--network=none` container eliminates false confidence from online GitHub Actions runners where outbound requests to `github.com` and `cdn.playwright.dev` succeed transparently. Between v0.3.5 and v0.3.15, **both previously identified air-gapped defects were completely closed**, lifting offline execution to `1,480 passed`, `0 failed`, `0 errors` in `722.24s`:

1. **Offline Icon Lookup Guard (`RESOLVED`)**: `find_package_for_resource_path()` and `find_package_for_icon_path()` in `src/drawlib/_release_assets.py` validate `normalized in pkg.files` prior to returning a package, and `resolve_png_icon_path()` in `src/drawlib/_icons/png_icons/_base.py` short-circuits missing names before calling `download_if_not_exist()`. All negative icon tests now pass offline with `0` DNS lookups.
2. **Pre-Baked Headless Chromium & Full Asset Cache (`RESOLVED`)**: `.pni/dockerfiles/Dockerfile` now pre-installs `libatomic1`, Playwright OS dependencies (`playwright install-deps chromium`), `chromium-headless-shell` (`playwright install --only-shell chromium`), and all `83` Drawlib font/icon asset packages (`drawlib cache download --all`) at image build time (`RV-01`: `138.75s`). As a direct result, all `.pptx` DOM/OpenXML tests, A4 PDF compilation tests, and Japanese `SourceCode` font tests execute offline with `0` **failures**, raising `_pptx_writer.py` from `11.1%` to `89.5%`, `_pptx_dom.py` from `58.6%` to `85.7%`, and `compiler/pdf.py` from `52.2%` to `90.5%`.

Complete Clean-Room Skipped Test Inventory (`RV-02`: `1,480` Passed, `0` Failed, `5` Clean Skips)

All `5` skipped tests out of `1,485` collected items are intentional clean-room environment skips triggered because repository-only authoring directories (`tools/release_assets/v0.3`, `.agents/rules/drawlib.md`, and `docs_src/jupyter`) are deliberately omitted from the minimal runtime test container image:

#	Status	Test Case (`module::test_name`)	Omitted Repo Path in Clean-Room Image	Verification Note
1	`SKIP`	`test_release_assets::test_all_declared_files_exist_on_disk`	`tools/release_assets/v0.3`	Validates raw uncompressed release-asset staging files present only in full git checkout.
2	`SKIP`	`test_release_assets::test_deterministic_zip_hashes_match_all_packages`	`tools/release_assets/v0.3`	Re-zips raw staging files to verify deterministic SHA-256 parity with `_release_assets.py`.
3	`SKIP`	`test_cli_init_agent::test_repo_antigravity_rule_is_synced`	`.agents/rules/drawlib.md`	Checks repo-local `.agents/` rule parity against embedded `drawlib rules` output.
4	`SKIP`	`test_jupyter_docs_dryrun::test_official_jupyter_docs_dry_run`	`docs_src/jupyter`	Executes end-to-end Jupyter dry-run against the repository's bilingual handbook source tree.
5	`SKIP`	`test_jupyter_docs_dryrun::test_official_jupyter_docs_build_outputs`	`docs_src/jupyter`	Verifies generated notebook and HTML landing-page artifacts for `docs_src/jupyter`.

6. Deep Dive: Security, Supply Chain & API Surface

First-party Drawlib v0.3.15 source code, dependency declarations, and container packaging achieve **94/100 (Grade S)** on `security_vulnerability_audit` (`status="passed"`), **95/100 (Grade S)** on `supply_chain_audit` , and **94/100** on the overall Security axis—delivering **zero CVEs across Drawlib's 13 runtime and 10 dev dependencies**, **zero CRITICAL or HIGH application CVEs**, **zero active IaC misconfigurations**, **zero secrets** across `434` source files and `111` companion DSL files, and complete OpenSSF governance (`SECURITY.md` , `.github/dependabot.yml` , `CODE_OF_CONDUCT.md` , and `5` least-privilege GitHub Actions workflows).

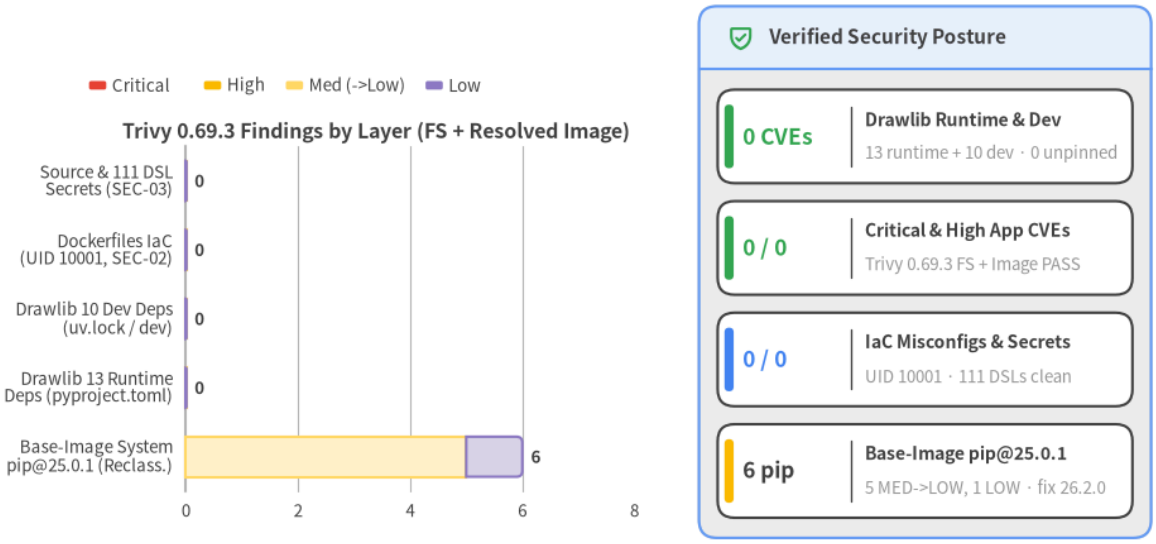


Figure 6.1 — Trivy 0.69.3 SCA, IaC, and Secret Scan Posture (v0.3.15): 0 CVEs in Drawlib's 13 runtime & 10 dev deps, 0 IaC/secret flaws, and 6 base-image system pip@25.0.1 CVEs reclassified to LOW.

6.1 Dependency Vulnerability & Container Image Scan (`SEC-01`: `0` Drawlib CVEs)

Trivy v0.69.3 Software Composition Analysis (`SEC-01`) scanned both the project filesystem (`pyproject.toml`, `uv.lock`) and the resolved container image built from `.pni/dockerfiles/Dockerfile` (`ghcr.io/astral-sh/uv:python3.12-bookworm-slim`):

- **Zero Vulnerabilities in Drawlib's Own Dependency Tree (`0` CVEs)**: All `13` direct runtime dependencies (`matplotlib>=3.8`, `pillow>=13.0.0`, `numpy>=1.26`, `pydantic>=2.8`, `pygments>=2.18`, `markdown-it-py>=3.0`, `jinja2>=3.1`, `markupsafe>=2.1`, `typer>=0.12.0`, `rich>=13.0.0`, `playwright>=1.49.0`, `python-docx>=1.2.0`, `python-pptx>=1.0.2`) and all `10` dev dependencies report `0` **unpinned specifiers** and `0` **CVEs** (`0` CRITICAL, `0` HIGH, `0` MEDIUM, `0` LOW).
- **Base-Image System `pip@25.0.1` (`6` Advisories, Reclassified to `LOW` in `act-09`)**: The only `6` Python-package advisories surfaced by the resolved image scan (`5 MEDIUM`, `1 LOW`) belong exclusively to system `pip@25.0.1` pre-installed inside `ghcr.io/astral-sh/uv:python3.12-bookworm-slim`. Because Drawlib uses `uv` for builds and never imports, invokes, or bundles `pip` at runtime, all `5 MEDIUM` findings were reclassified to `LOW` during `act-09`. Upgrading system `pip >= 26.2.0` (or removing unused `pip`) in Dockerfiles clears all `6` findings cleanly.
- **Container OS Base-Image Scope**: Project production/test Dockerfiles (`tools/docker/Dockerfile.*`) report `0` **OS CVEs**. The non-shipped PNI test-harness container (`.pni/dockerfiles/Dockerfile`, Debian bookworm OS libraries + Chromium runtime dependencies) carries `740` upstream Debian bookworm OS advisories (`11` CRITICAL, `181` HIGH, `304` MEDIUM, `219` LOW) strictly confined to the local test harness and never distributed in Drawlib PyPI wheels.

Base-Image System `pip@25.0.1` Advisory Inventory (Not Imported by Drawlib)

Advisory ID	Raw -> Reclass. Sev	CVSS	Installed -> Fixed	Vulnerability Summary & Container Remediation
`CVE-2026-8643`	`MED` -> `LOW`	`8.0`	`pip@25.0.1` -> `26.1.2`	Path traversal via malicious entry point name in pip wheel installation.
`CVE-2026-13346`	`MED` -> `LOW`	`6.5`	`pip@25.0.1` -> `26.2.0`	Arbitrary file installation via malicious package indexes.

Advisory ID	Raw -> Reclass. Sev	CVSS	Installed -> Fixed	Vulnerability Summary & Container Remediation
`CVE-2026-6357`	`MED` -> `LOW`	`5.8`	`pip@25.0.1` -> `26.1`	Arbitrary code execution or information disclosure via malicious wheel.
`CVE-2025-8869`	`MED` -> `LOW`	`5.3`	`pip@25.0.1` -> `25.3`	Missing checks on symbolic link extraction during archive unpacking.
`CVE-2026-3219`	`MED` -> `LOW`	`5.0`	`pip@25.0.1` -> `26.1`	Incorrect file installation due to improper archive handling.
`CVE-2026-1703`	`LOW`	`3.9`	`pip@25.0.1` -> `26.0`	Information disclosure via path traversal in crafted wheel archives.

Remediation (`P2`): Add `python -m pip install --no-cache-dir --upgrade "pip>=26.2.0"` (or uninstall unused system `pip`) in container build stages where `uv` is the sole package installer.

6.2 Container IaC Hardening (`SEC-02`) & Secret Hygiene (`SEC-03`)

Dockerfile Hardening (`0` Active Misconfigurations)

Trivy IaC scanning verified all three production/test container definitions under `tools/docker/` alongside `.pni/dockerfiles/Dockerfile`:

Check ID	Severity	Affected Container Definitions	Resolution Status	Engineering Implementation & Verification
`DS-0002`	`HIGH`	`tools/docker/Dockerfile.{prod,test_local,test_pypi}`, `.pni/dockerfiles/Dockerfile`	`PASS` (`0` **Failures**)	All container images create and switch to an unprivileged non-root user (`USER drawlib` / `USER pni`, `UID/GID 10001`), relocate Playwright browsers (`/opt/ms-playwright`) and asset caches with non-root read permissions, and emit `0` Trivy `DS-0002` findings.
`DS-0026`	`LOW` (`×3`)	`tools/docker/Dockerfile.{prod,test_local,test_pypi}`	`WAIVED` (`.pni/pni.toml`)	Ephemeral batch CLI and `pytest` containers terminate immediately upon command completion rather than running long-lived daemons; waived via `[[opt_out]] id = "DS-0026"` (`opt_out_counts = {"DS-0026": 3}`).

Secret Scanning & Companion DSL Hygiene (`SEC-03` : `0` Secrets)

- **Zero Hardcoded Source Secrets (`PASS`)**: Both AST literal entropy analysis (`AstFacts.hardcoded_secrets = []`) and Trivy repository scanning confirmed `0` **API keys, tokens, or credentials** across all `434` source files and `155` test files, with `0` **committed secret files**.
- **Clean Companion DSL Corpus (`111` Files, `0` Secrets)**: Across all `111` non-Python configuration/companion files (`dockerfile=3` , `css=19` , `yaml=9` , `json=77` , `toml=1` , `xml=2`), secret scanning reported `0` **findings**.
- **Default-Deny `.dockerignore` & `.pni/pni.toml` Scope (`PASS`)**: Root `.dockerignore` starts with `*` (deny all) and explicitly allowlists only build-required directories, while `.gitignore` and `.pni/pni.toml` (`skip_files = [".env", ".env.*", "**/.env"]`) isolate untracked local developer `.env` files.

6.3 AST Dangerous Code Sinks (`CWE-94`) & API Attack Surface

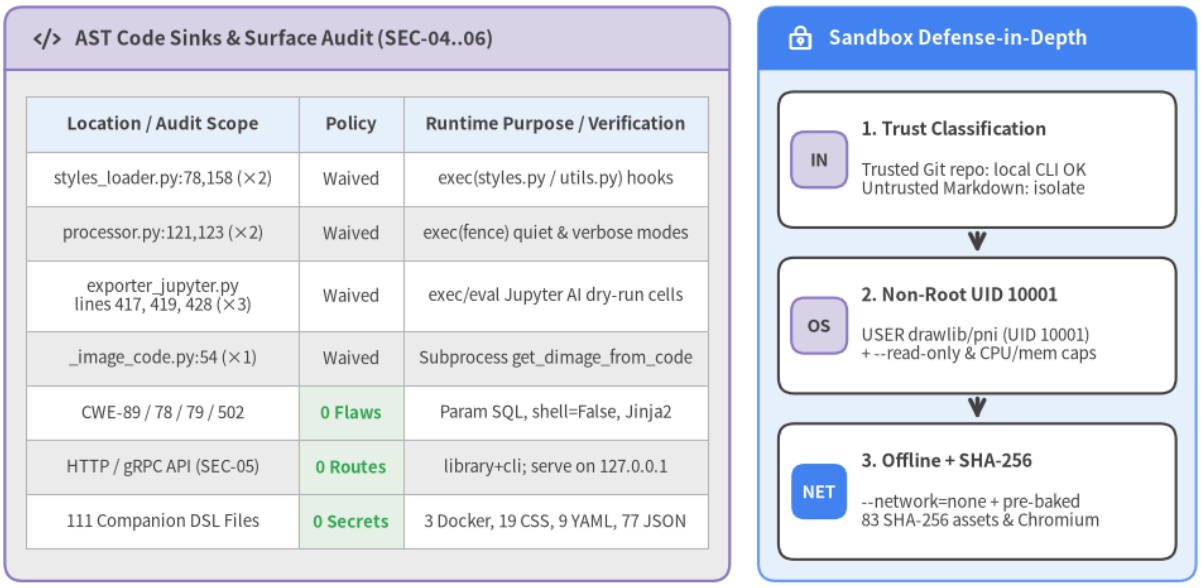


Figure 6.2 — Complete 8-site CWE-94 exec()/eval() inventory (waived in .pni/pni.toml), zero injection/API/DSL-secret verification, and container defense-in-depth boundary for untrusted Markdown.

Detailed Assessment of `CWE-94` Sites, Injection Classes & API Endpoints

1. **Eight Intentional `CWE-94` (`exec` / `eval`) Sinks (`[[opt_out]] id = "CWE-94"`, `matched=8`, `0` Unwaived)**:
   - `src/drawlib/_builder/_common/styles_loader.py:78` (`load_styles`) and `styles_loader.py:158` (`load_utils`): compile and execute project-level `styles.py` and `utils.py` hooks so documentation projects can customize `Styles` / `Colors` and register reusable drawing macros.
   - `src/drawlib/_builder/doc_builder/processor/processor.py:121` (quiet stdout-redirected mode) and `processor.py:123` (`verbose` / `developer` mode): execute ```` ```drawlib ```` fenced Python blocks inside Markdown files while intercepting `save()`.

- `src/drawlib/_builder/doc_builder/exporter_jupyter.py:417`, `419` (trailing-expression `eval`), and `428` (`exec`): `3` sites executing notebook code cells during Jupyter AI Dry-Run verification so generated `.ipynb` tutorials are proven runnable before publication.
- `src/drawlib/_core/l4_canvas/_image_code.py:54` (`_worker_render_code`): executes nested drawing scripts inside an isolated `multiprocessing` child process for `get_dimage_from_code(code, timeout=...)`.
- **Architectural Verdict**: Like Jupyter notebooks, Quarto, or Sphinx code-execution directives, executing Python code is the core purpose of Drawlib (*"Illustration as Code"*). This trust model is now documented explicitly in root `SECURITY.md` and recorded under `[[opt_out]] id = "CWE-94"` in `.pni/pni.toml`. Untrusted third-party Markdown should always be built inside an ephemeral, non-root, offline container (`--user 10001:10001 --network=none --read-only --pids-limit=128 --memory=1g`) built via `./dcli docker build --full`.

2. **Zero Injection or Deserialization Vulnerabilities (`PASS`)**:
   - `CWE-89` **(SQL Injection = `0`)**: `BuildImageCache` and `CliImageCache` in `src/drawlib/_builder/_common/cache.py` use strictly parameterized `?` SQLite queries on static table schemas.
   - `CWE-78` **(OS Command Injection = `0`)**: Subprocess execution at `src/drawlib/_builder/_common/browser.py:306` passes explicit argument arrays (`[str(exec_path), str(cli_path), *_INSTALL_ARGS]`) with `shell=False`.
   - `CWE-79` / `CWE-502` **(XSS / Unsafe Deserialization = `0`)**: HTML templates use `Jinja2` + `MarkupSafe` escaping, manifests parse via stdlib `tomllib`, and caches persist raw PNG/SVG bytes plus SHA-256 digests (`0` `pickle` or unsafe `yaml.load` calls).
3. **API Endpoint Surface (`SEC-05` / `SEC-06`: `0` Routes, `not_applicable`)**:
   - Drawlib's runtime archetype is `library+cli` with **zero production HTTP/gRPC service endpoints**. The `drawlib serve` command binds strictly to `127.0.0.1` for local interactive preview and `--check` link validation.

6.4 OpenSSF Supply-Chain Posture & Governance Checklist

Drawlib v0.3.15 scored **95/100 (Grade S)** on `supply_chain_audit` and **96/100 (Grade S)** on `repo_structure_audit`. Since v0.3.5, Drawlib closed both open governance gaps by landing `SECURITY.md` and weekly `.github/dependabot.yml` automation while expanding CI coverage to `5` least-privilege GitHub Actions workflows:

Supply-Chain & Governance Control	v0.3.15 Status	Evidence Location	Verification & Hardening Notes
Open-Source License (`Apache-2.0`)	**PASS**	`LICENSE`	Standard Apache-2.0 license cleanly declared in repository root and `pyproject.toml`.
Vulnerability Disclosure Policy	**PASS** *(New since v0.3.5)*	`SECURITY.md`	Documents active `0.3.x` support line, explicit *"Illustration as Code"* execution trust model, container sandbox command, and GitHub Private Vulnerability Reporting.
Community Governance	**PASS**	`CODE_OF_CONDUCT.md`	Contributor covenant and escalation guidelines committed at repository root.
Automated Dependency Updates	**PASS** *(New since v0.3.5)*	`.github/dependabot.yml`	Configured for weekly grouped update PRs covering both `uv` Python dependencies and `github-actions`.
CI Workflow Token Permissions (`5` Workflows)	**PASS**	`.github/workflows/{quality,release,ut_github,ut_pypi,ut_template}.yml`	`0` CI supply-chain issues; all `5` workflows enforce explicit `permissions: contents: read`, with `contents: write` scoped strictly to the `publish-release` job in `release.yml`.
External Binary Asset Integrity (`83` Packages)	**PASS**	`src/drawlib/_core/l3_external/_package.py` & `src/drawlib/_release_assets.py`	Every lazy-downloaded font, icon, and GeoJSON archive embeds a compiled-in **SHA-256 digest** verified alongside Zip-Slip path checks, `pkg.files` membership guards, and `timeout=60` HTTPS fetches.
Container Least Privilege (`USER`)	**PASS**	`tools/docker/Dockerfile.{prod,test_local,test_pypi}`, `.pni/dockerfiles/Dockerfile`	Non-root `USER drawlib` / `USER pni` (`UID 10001`), `0` Trivy `DS-0002` failures, default-deny `.dockerignore`.
Runtime Dependency Specifiers & Lockfile	**PASS / WAIVED**	`pyproject.toml`, `uv.lock`, `.pni/pni.toml`	All `13` runtime deps specify SemVer lower bounds (`0` unpinned, `0` Drawlib CVEs); `uv.lock` gitignore policy waived via `python/lockfile_committed` for Python `3.11–3.14` library flexibility.

Supply-Chain & Governance Control	v0.3.15 Status	Evidence Location	Verification & Hardening Notes
GitHub Actions SHA Pinning	**INFO** (`P2`)	`.github/workflows/{quality,release,ut_template}.yml`	Steps currently reference major version tags (`actions/checkout@v7` , `astral-sh/setup-uv@v7` , `actions/setup-python@v7`); pin to 40-char commit SHAs with version comments for immutable supply-chain provenance.