



# Document Platform Architecture

ReportLab kernel · layout v2 · Phase 4 expansion

Problem → architecture → proof.

ReportLab kernel — combination strategy.

---

**AUDIENCE** Engineers · architects · technical due diligence

**Part I — Problem**

# Problem

Why a document platform expansion on ReportLab instead of HTML-first rewrite.

---

**§1**

## Problem statement

Stakeholder PDFs outgrew ratio-only table layout.

---

The NLT document suite started as ten investor briefs with ratio-scaled tables and paragraph-only two-column grids. Jun 2026 research scored table auto-width at ~2.1/5 versus pdfmake-style intrinsic sizing. Chart + callout layouts failed because nested Platypus tables rejected non-paragraph left columns. Phase 4 addresses root causes in-engine rather than rewriting in WeasyPrint or Playwright.

- Vol 1-10 briefs must keep building from ReportLab/reports/\* — no consumer fork.
- New classes (guides, release notes, marketing, whitepapers) share nlt-v3.2 brand.
- QA must catch thin pages and broken portfolio links before briefs.nltlabs.ai deploy.
- Agents authoring stories need hybrid table mode and screenshot primitives.

HTML-first PDF generators (WeasyPrint, Playwright print) excel at web parity but would fork ten production briefs and every consumer story module. The combination strategy keeps Platypus as kernel, ports pdfmake-style width algorithms, and adds pdfplumber layout audit at the CI edge. This whitepaper documents that architecture for engineers evaluating Phase 4 templates or authoring new reports/\*/story.py modules.

**§2**

## Constraints and requirements

Non-negotiables for the NLT document suite.

---

## Requirements matrix

Constraint	Rationale	Implementation
Sole PDF kernel	Parity with 117 pp workspace bar	ReportLab Platypus
Brand voice	FUND not invest; outward names only	docs/NLT_BRAND.md
No hand-edited PDFs	Reproducible CI builds	report-studio build --audit
Sibling read-only	No cross-repo writes	*_ROOT env at build time
Security floor	tapps gate $\geq 50$ security	tapps_validate_changed

## Part II — Architecture

# Architecture

ReportLab kernel with borrowed algorithms and edge QA tools.

## §3

## Architecture

Four components, one rendering layer.

Four outward-facing components (Scout, Engine, AgentForge, Memory) produce artifacts consumed at PDF build time. Report Studio sits above as the rendering layer — templates/\*.yaml declare TOC parts; reports/\*/story.py assembles flowables.

### Platform ecosystem

The four outward-facing components share a workspace layout but enforce separate write boundaries. The table below summarizes who owns durable state; Vol 3 diagrams show how compose stacks and git remotes connect.

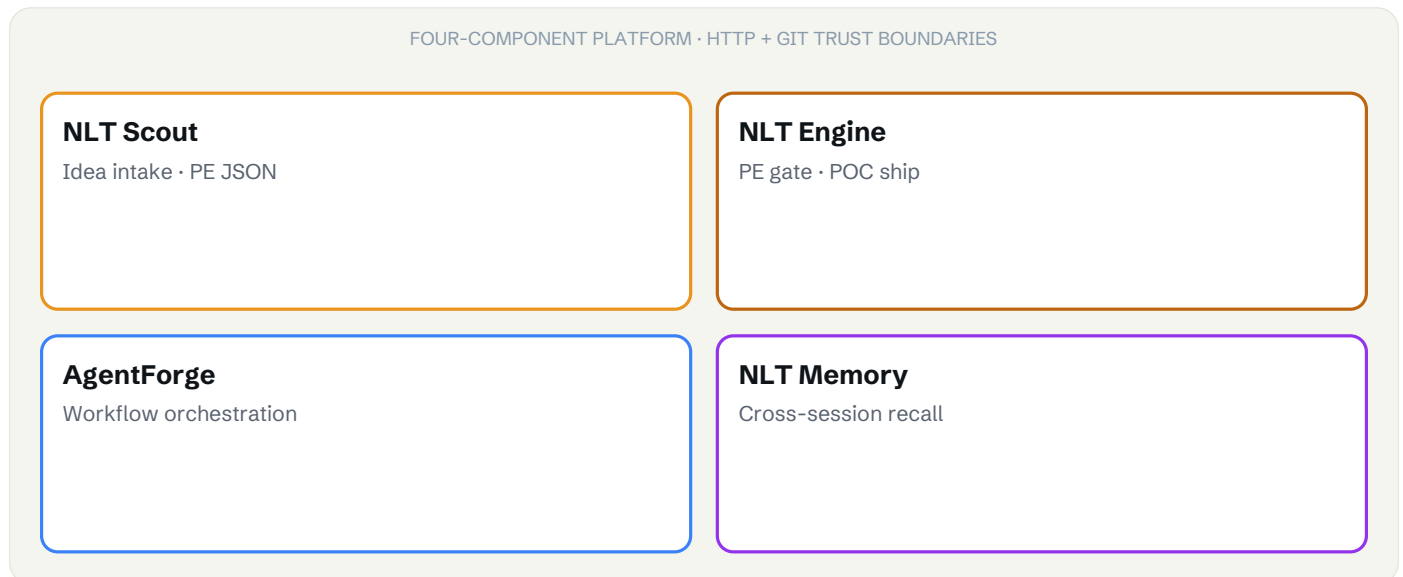


Figure 1 — Four outward-facing components and their primary roles. Deep dives: Vol 5-8.

## Write vs read scope

Component	Primary writes	Primary reads
NLT Scout	ideas/, PE JSON	Intake templates, brand packs
NLT Engine	Registry, workflows, agent defs	Scout artifacts, memory recall
AgentForge	Task audit, experience index	Engine invoke, memory bridge
NLT Memory	Tiers, graph, embeddings	All MCP memory tool calls

OPERATING PIPELINE



**Integration principle**  
 Each component owns writes inside its trust boundary. Cross-repo effects use HTTP APIs or git — never in-tree patches of consumer code inside AgentForge.

OPERATING PIPELINE



Figure 2 — Document platform pipeline from template manifest to deployed PDF.

### §4

# Layout engine

Content-aware tables and chart-friendly multi-column grids.

- `compute_col_widths()` — min token width, max line width
- `premium_table(mode=hybrid)` — author weights clamp
- `_is_grid_safe_flowable()` — charts with stable wrap() in
- `default fill_width=CONTENT_W (7.4 inches)` on `make_t`

**Width solver**  
 pdfmake 3-case min/max distribution ported to `compute_col_widths()`. Hybrid mode clamps author hints to readable minima.

## Table width modes

Mode	When to use	Entry point
ratio	Suite tables with author weights	<code>premium_table(mode="ratio")</code>
auto	Unknown content width	<code>premium_table(mode="auto")</code>
hybrid	Author hint + content floor	<code>premium_table(mode="hybrid")</code>

### §5

# Media and charts

Screenshots, sparklines, and template expansion.

- `screenshot_figure()` + `MAX_FIGURE_H` — GitHub Docs width/height guidance.
- `figure_in_rail()` — callout column width via `callout_width()`.
- `BrandedSparklineFlowable` — release-note trend lines without matplotlib.
- `ingest-screenshots.py` — 1000 px / 500 KB PNG normalization.
- `user_guide` + `release_notes` templates dogfood the media/chart stack.

Template manifests under `templates/*.yaml` declare running headers, audit profiles, and TOC parts. Phase 4 adds `user_guide`, `release_notes`, `marketing_brief`, `whitepaper`, and `video_artifact` scaffolds — each pairs with `reports//story.py` and registers in `docs/NLT_DOCS_SUITE.md`. Authors run `report-studio init` for consumer repos or copy the ReportLab dogfood modules as reference implementations.

## Phase 4 template catalog

Template	Purpose	Status
<code>user_guide</code>	Task-oriented how-to + screenshots	Shipped
<code>release_notes</code>	Keep a Changelog + sparkline	Shipped
<code>marketing_brief</code>	Partner memo ≤4 pp	Shipped
<code>whitepaper</code>	Problem → architecture → proof	Shipped
<code>video_artifact</code>	Storyboard + shot list	Scaffold

Part III — Proof

# Proof

Operational evidence and suite integration.

§6

## Proof and operations

Build-time audit, pytest judges, and live URLs.

<b>1</b> pytest judges run on every validate_changed for reports/ edits.	<b>2</b> Vol 3-8 depth gate enforces engineering-reference page counts.
<b>3</b> apps/docs build-pdfs.mjs runs report-studio build --audit per volume.	<b>4</b> Phase 4 expansion table indexed in docs/NLT_DOCS_SUITE.md.

Consumer repos pin nlt-report-studio by git tag and invoke report-studio build from CI with --audit. Engine maintainers run tapps\_validate\_changed on every reports/ edit; the PDF judge shell script guards regression on layout primitives. Sibling roots remain read-only — stories cite path::symbol anchors for verify, never mutate foreign repositories during build.

- Vol 1-10 briefs unchanged — Phase 4 is additive.
- Marketing brief ≤4 pp; whitepaper ≥8 pp acceptance gates in pytest.
- video\_artifact template follows in Story 2.7.

### Operational proof matrix

Artifact	Command	Gate
User guide PDF	report-studio build --template user_guide	reference audit
Release notes PDF	report-studio build --template release_notes	reference audit
Layout unit tests	uv run pytest tests/test_col_widths.py	9+ assertions
Suite deploy	cd apps/docs && npm run build	prebuild PDF step

### **Methodology**

Generated from nlt-report-studio with brand nlt-v3.2. Portfolio proof: <https://portfolio.nltlabs.ai> · Briefs: <https://briefs.nltlabs.ai>

Confidential — prepared for NLT Labs stakeholders. Outward names follow docs/NLT\_BRAND.md. Internal checkout names appear only in builder ops volumes (9-10).