
M & A T E C H N O L O G Y A S S E S S M E N T

Technology Due Diligence Report

Proposed Acquisition of VLLM

Target: VLLM

Report ID: TDD-20261016-073040

Classification: CONFIDENTIAL — for the named buying party and its advisers

Proposed Price: USD 2,000,000

Scan Date: 16 October 2026

Repository HEAD: fa75d3456d60a5ffe923818c4b495295675b38e1

Prepared for the Investment Committee

TABLE OF CONTENTS

Confidential — for the named buying party and its advisers

1. Executive Summary & Headline Findings	1
1.1 Verdict and Headline Posture	1
1.2 Confidence, Coverage and Material Unknowns.....	2
1.3 Headline Economics and Calibration Caveat	2
1.4 Deal-Thesis Alignment.....	3
1.5 Key Figures at a Glance	4
1.6 Top Risks and Major Risk Register.....	4
1.7 Major Risk Register (Pillar Level)	5
2. Transaction Recommendation, Conditions & Escalation Thresholds.....	6
2.1 Decision Tiers and Escalation Thresholds.....	7
2.2 Conditions for Acquisition	8
2.3 Deal-Breaker Register	9
2.4 Per-Pillar Escalation Rows	9
2.5 Sensitivity Sweep Outcomes	9
3. Remediation Cost Model, Valuation Impact & Sensitivity	10
3.1 Parameters Used Downstream.....	11
3.2 Remediation Line Items	13
3.3 Remediation Cost Confidence Interval.....	14
3.4 Aggregate Hours Cap, Saturation Model, and Line-Item Basis	15
3.5 SQALE Remediation Basis and Double-Counting Ledger	15
3.6 Risk Valuation Lines (Expected Loss)	16
3.7 Scenario Branches (0.3.0).....	17
3.8 Deal Economics Headline Valuation Reconciliation	18
3.9 Sensitivity Sweep Detail.....	18
3.10 Financial Lineage and Reproduction Fingerprint.....	19
4. Licensing & IP Exposure	20
4.1 Headline Posture.....	20
4.2 Material Findings.....	21

4.3 Per-Package Detected Licenses	22
4.4 License Clause Findings.....	23
4.5 IP Ownership Metrics	24
4.6 Transaction Implications	25
5. Security, Vulnerability & Attack Surface Posture	26
5.1 Headline Posture and Cross-Pillar Reconciliation.....	26
5.2 CVE Inventory and Advisory Coverage.....	27
5.3 Attack Surface Composition.....	28
5.4 Secrets, IaC, and Privacy Posture	29
5.5 Taint Analysis Summary	31
5.6 Transaction Implications	31
6. Supply Chain & Dependency Posture	32
6.1 Headline Posture.....	32
6.2 Direct Dependency Inventory.....	32
6.3 Lockfile and Transitive Resolution	33
6.4 End-of-Life Posture	33
6.5 Transaction Implications	34
7. Code Quality, Complexity & Technical Debt	34
7.1 Headline Posture.....	34
7.2 Complexity Percentiles and Hotspot Symbols	35
7.3 SQALE, Duplication, and Debt Markers.....	36
7.4 Material Findings.....	37
7.5 Transaction Implications	38
8. Testing, Coverage & Regression Risk.....	39
8.1 Headline Posture.....	39
8.2 Test Deficit by Module.....	39
8.3 Coverage Module Detail and High-Value Gaps.....	40
8.4 High-Value Coverage Gaps.....	41
8.5 Transaction Implications	42
9. Architecture, Entanglement & Integration Risk	42
9.1 Headline Posture.....	42
9.2 Circular Dependency Cycle.....	43

9.3 Cross-Layer Violations.....	43
9.4 API Contract Maturity, Database Coupling, and Serialization Risks.....	44
9.5 Serialization and Protocol-Layer Risks	45
9.6 Transaction Implications	46
10. Engineering Capacity, Bus Factor & Knowledge Loss Vulnerability	47
10.1 Headline Posture.....	47
10.2 Key-Person Risk Detail	47
10.3 Module Bus Factor	48
10.4 Git Team Signals	49
10.5 Knowledge Loss Vulnerability Detail	50
10.6 Transaction Implications	51
11. Maintenance, Documentation & Operational Maturity	51
11.1 Headline Posture.....	52
11.2 Document Staleness.....	52
11.3 CI/CD Maturity	53
11.4 Transaction Implications	53
12. AI/ML Footprint.....	54
12.1 Headline Posture.....	54
12.2 Framework Hits	54
12.3 SDK Hits	55
12.4 Transaction Implications	55
13. Technology Stack & Lifecycle Posture	56
13.1 Headline Posture.....	56
13.2 Technology Inventory.....	56
13.3 Transaction Implications	57
14. Consolidated Risk Register & Top Risks	57
14.1 Consolidated Top Risks.....	57
14.2 Major Risks Register (Pillar Level).....	58
14.3 Pillar Classification Register	59
14.4 Deal-Breaker Register	60
15. Confidence, Scope Limitations & Material Unknowns.....	61
15.1 Integrity Status.....	61

15.2 Confidence Inputs.....	62
15.3 Per-Pillar Assessment Posture	63
15.4 Coverage Gaps by Pillar	64
15.5 Out-of-Scope Evidence Register	65
16. Reading Notes & Honesty Conventions.....	66
16.1 Assessment Coverage	66
16.2 Evidence Honesty	66
16.3 Currency of Absence	67
16.4 Deal-Mechanic Vocabulary	67
16.5 Severity and Ranking	67
16.6 Known Limits of This Method	68
A.1 Report Metadata.....	68
A.2 Scan Mode	69
A.3 Pillar Posture Register	69
A.4 Per-Pillar Posture Detail (Selected)	70
A.4.1 AI/ML	70
A.4.2 Attack Surface	70
A.4.3 Complexity	71
A.4.4 Entanglement	71
A.4.5 License	71
A.4.6 IP Ownership	72
A.4.7 Retention.....	72
A.5 Step Timings.....	72
B.1 Saturation Model (Full)	73
B.2 Health-Posture Ceilings	73
B.3 Severity Risk Weights (Decision Model)	74
C.1 Direct Dependency Inventory (Full, 20 of 20)	74
C.2 Lockfile Resolution.....	76
D.1 CVE Records (Full, 116 of 116).....	76
E.1 Per-Package License Findings (Full, 22 of 22).....	79
E.2 License Clause Findings (Full, 3 of 3)	80
F.1 Complexity Hotspot Symbols (Top 30 of 65,109).....	81
F.2 Duplication Groups (Sample).....	83

F.3 SQALE Debt Modules (Sample).....	84
G.1 Debt Markers (Sample of 150 of 5,669).....	84
H.1 Candidate Entry Points (Sample of 70 of 2,154).....	90
I.1 Key-Person Risk Rows (Top 30 of 3,793).....	94
I.2 Module Bus-Factor Rows (Top 30 of 761).....	95
I.3 Knowledge Loss Vulnerability Rows (Top 30 of 3,793, critical).....	96
J.1 Maintenance Assets (Full)	98
J.2 IaC Findings (Full, 11 of 11)	98
J.3 CI Files (Full, 12 of 12)	100
J.4 Coverage Artifacts	100
K.1 Privacy Findings (Full, 18 of 18).....	100
K.2 Secret Candidates (Full, 6 of 6)	104
L.1 Model Files	105
M.1 Integrity Issues (Full, 11 of 11).....	105
N.1 Taint Summary	107
N.2 Top Taint Flows.....	107
O.1 Reproduction Fingerprint	107
P.1 Evidence Provenance (Selected Key Figures).....	108
Q.1 Glossary (Alphabetical).....	112
R.1 Contradiction Ledger	117
C-DC-001 — resolved.....	117
R.2 Release Record.....	117

Note: This Table of Contents is generated via field codes. To ensure page-number accuracy after editing, right-click the TOC and select "Update Field."

P A R T I

**EXECUTIVE DECISION &
RECOMMENDATION****1. Executive Summary & Headline Findings****1.1 Verdict and Headline Posture**

This report presents a deterministic, data-driven projection of the technology posture of the proposed acquisition target referred to throughout as VLLM. It has been prepared for the named buying party and its advisers to support the investment committee's deliberations on a proposed transaction price of two million United States dollars. The assessment is anchored to a single finalized scan payload, generated on 16 October 2026 against repository HEAD commit `fa75d3456d60a5ffe923818c4b495295675b38e1`, with every figure reproduced directly from the stored ground truth and no unanchored estimates or fabricated page references. Where this report asserts a number, the underlying evidence lineage is preserved in Appendix P, and the consolidated risk register in Section 14 forms the single authoritative view of risk.

The categorical board decision produced by the assessment is DO NOT ACQUIRE on the proposed terms. The same tier is returned across every parameter of the sensitivity sweep described in Section 3, meaning the recommendation is invariant to reasonable changes in the model's financial assumptions. The confidence score is 72 out of 100, discounted by five partially or non-assessed domains whose absence of findings must not be misread as absence of risk. The headline remediation figure is 573,680 United States dollars, expressed as a deterministic one-hundred-percent adoption planning bound with a stated confidence interval of plus or minus thirty percent; the probability-weighted expected loss is not quantified in the present scan, so actual expected remediation lies at or below this bound.

Two deal-breaker conditions force the no-go tier. The first is license contamination: the `tqdm` package, declared in `benchmarks/multi_turn/requirements.txt`, is detected under a viral MPL-2.0 AND MIT composite license, and ten further dependencies carry UNKNOWN license classifications that have not been resolved by manifest metadata. The second is the attack-surface posture: 2,154 candidate external-reachable endpoints were detected across the codebase, of which 2,142 are untested and only 72 of the 2,154 could be resolved to a recognised framework. Together these conditions constitute hard precedent items that must

be discharged, or risk-adjusted into price, before the transaction can be recommended on alternative terms.

1.2 Confidence, Coverage and Material Unknowns

Confidence is computed as 20 plus forty times the coverage share plus forty times the risk-signal share, capped at 20 when coverage is zero and floored at 60 when the categorical decision is no-go. The present scan returns coverage of 77.3 percent across the 22 pillar register entries and a risk signal of 52.8 percent. Five domains remain partially or non-assessed and are explicitly named so that an empty cell in those domains is never read as a clean result: CVE (advisory coverage limited to eight resolvable packages out of a declared twenty); Dependency (no lockfile present, so transitive resolution could not be performed); EOL (zero evidence population, reported as NOT ASSESSED); Maintenance (no files scanned by the asset-staleness scanner); and Secrets (three path-excluded candidate paths not scanned). Each gap is itemised in Section 15 together with the independent evidence a buyer would need to obtain to close it.

No finding in this report has been independently confirmed by a human reviewer at the time of generation. Of 301 substantive rows, 199 are scanner-detected signals awaiting verification, 102 are inferred from structural or historical evidence, and none are confirmed. This is a static, offline-or-network-integrated source scan: it does not establish runtime behaviour, production incidents, or any matter requiring access to the target's systems, personnel, or executed contracts. The Out-of-Scope Evidence Register in Section 15 enumerates eight domains that lie beyond the static-scan perimeter and the specific independent evidence required for each.

1.3 Headline Economics and Calibration Caveat

The remediation estimate of 573,680 United States dollars is the sum of nine costed line items, each derived from the count of underlying findings multiplied by an hours-per-unit rate that has been saturated by a per-category cap and decay. The line-item composition is dominated by 223,200 dollars of critical-complexity remediation across 941 hotspot symbols and 185,700 dollars of high-complexity remediation across 1,557 additional symbols, together accounting for seventy-one percent of the total. The remaining twenty-nine percent is distributed across knowledge transfer for sole-maintainer modules, test investment on high-regression-risk modules, IP reassignment review, general technical-debt remediation, taint-flow sanitisation, dependency pinning, and decoupling of the single identified circular dependency cycle.

This figure is a deterministic planning bound, not a priced contractor quote. It is computed using the tool's built-in hours-per-unit constants and saturation caps, none of

which have been calibrated against a benchmarked delivery dataset (calibration.applied equals false). Buyers and their quality-of-earnings providers should treat the figure as an upper-bound scope estimate for sequencing and resourcing decisions, and commission a bottom-up re-estimate against contractor day-rates before relying on it for price negotiation. The confidence interval of plus or minus thirty percent is a stated-confidence range derived from the model's calibration, not a stochastic bound; the underlying uncertainty is wider still because the model has not been benchmarked.

1.4 Deal-Thesis Alignment

The buyer declared all six deal theses for this target, with none designated as primary and none carrying an explicit condition. Each thesis is scored side by side from the same stored evidence on a one-to-ten alignment scale that rates how strongly and completely the evidence bears on the thesis within its evidence-state cap. A score of ten does not mean the thesis is supported commercially; it means the stored evidence is complete with respect to the measurable dimensions of that thesis. A refused verdict, by contrast, means the thesis cannot be assessed by static analysis and must be evaluated by other means.

Table 1.1 — Deal-Thesis Alignment, Side by Side

Perspective	Alignment	Evidence State	Basis
Scale & Expansion	10/10 · 4 of 4 metrics resolved	Detected, not verified	Structural coupling only. The SCC-ranked cycle set bounds how hard independent scaling would be; it does not measure it.
Product Integration & Cross-Sell	6/10 · 5 of 5 metrics resolved	Partially assessed	Contributing pillar(s) did not complete a full assessment; the verdict is capped regardless of the figures.
Lean Maintenance & Operating Cost	10/10 · 5 of 5 metrics resolved	Detected, not verified	Remediation effort is an SQALE estimate over static structure. It bounds the maintenance backlog; it does not price a headcount saving.
IP Moat & Proprietary Core	refused	Unassessed	Not supported by static analysis. No signal in this report distinguishes proprietary algorithmic density from standard framework boilerplate.
Regulatory & Compliance	7/10 · 6 of 6 metrics resolved	Partially assessed	Contributing pillar(s) did not complete a full assessment; the verdict is capped regardless of the figures.
Talent & Acqui-hire	10/10 · 6 of 6 metrics resolved	Detected, not verified	Git history describes who has touched the code, not who will stay after close. Retention intent is outside repository evidence.

The alignment score rates evidence completeness within the thesis's evidence-state cap; it is not a business-fit score and does not by itself support a valuation conclusion.

1.5 Key Figures at a Glance

Table 1.2 — Headline Decision and Economic Figures

Metric	Value	Source
Categorical decision tier	DO NOT ACQUIRE	decision.escalation.categorical_tier
Economic decision tier	DO NOT ACQUIRE	decision.escalation.economic_tier
Confidence score	72 of 100	decision.confidence_score
Total remediation hours	11,474 h	financial.total_remediation_hours
Total remediation cost	\$573,680	financial.total_remediation_cost
Confidence interval ($\pm 30\%$)	\$401,576 – \$745,784	financial.assumptions.remediation_cost_confidence_interval
Remediation as % of proposed price	28.7%	decision.remediation_cost_pct_of_acquisition
Valuation discount applied	22.1%	financial.valuation_discount_pct
Post-merge integration drag	80.0% (capped)	financial.post_merge_integration_drag_pct
Person-years of remediation	5.5 py	financial.assumptions.person_years
Integrity status	warn (0 errors, 11 warnings)	integrity.status

1.6 Top Risks and Major Risk Register

The top-ten risks by severity are reproduced below from the consolidated register. Six of the ten are critical: three complexity hotspots in model execution and attention backends, three retention risks associated with named individual contributors, and one attack-surface endpoint. The remaining four are critical-to-high risks spanning attack surface and debt markers. Estimated cost is shown only where the financial model produces a number; absent figures reflect the model's stated decision not to price certain critical findings, not an absence of impact.

Table 1.3 — Top Ten Risk Findings

Pillar	Title	Severity	Impact	File	Est. Cost (USD)
Complexity	FusedMoEFactory	critical	Critical	model_executor/layers/fused_moe/layer.py	not quantified
Complexity	_build_decode	critical	Critical	v1/attention/backends/mla/rocm_aiter_mla.py	not quantified
Complexity	_build_attention_metadata	critical	Critical	v1/worker/gpu_model_runner.py	not quantified
Retention	Ben Browning	critical	Critical	not quantified	not quantified
Retention	Benjamin Chislett	critical	Critical	not quantified	not quantified
Retention	Aaron Hao	critical	Critical	not quantified	not quantified
Attack Surface	cancel_responses	critical	Critical	entrypoints/openai/responses/api_router.py	not quantified
Attack Surface	abort_requests	critical	Critical	entrypoints/rl/online/api_router.py	not quantified
Attack Surface	chat_completions	critical	Critical	examples/disaggregated/disaggregated_serving/disagg_proxy_multiturn.py	not quantified
Debt	fixme	high	High	tests/basic_correctness/memory/sleep_mode/test_sleep_mode.py	\$14,880

Rendered 10 of 10 rows. Source: workbook sheet Dashboard. Estimated cost is shown only where the financial model assigns a line item; critical findings without a cost figure are sized separately in Section 3.

1.7 Major Risk Register (Pillar Level)

The thirteen pillar-level major risks below drive the categorical decision. Each row aggregates one pillar's highest-concentration finding into a single headline metric; the integrity warnings in Section 15 mark six of these pillars as rendering no item-level findings, meaning their posture is held on aggregate metrics only. Two pillars (cve and dependency)

are marked unverified because their underlying scans could not be completed against a resolvable package population.

Table 1.4 — Major Risks Register

Pillar	Area	Headline	Unverified
Entanglement	Architecture	total_cycles: 1, max_cycle_length: 606, cross_layer_violations: 9,109	false
Debt	Engineering	fixme	false
Integration	Engineering	integration_risk_score: 10.0, proprietary_coupling_count: 1,121, total_findings: 1,255	false
Coverage	Engineering	modules_analyzed: 791, high_value_gaps: 138, effective_insured_value: 1.0	false
Complexity	Engineering	sparse_attn_indexer_kpool	false
Test Deficit	Engineering	total_modules: 791, modules_below_threshold: 545, overall_test_ratio: 0.3787	false
IP Ownership	Legal & Compliance	total_authors: 3,793, external_author_count: 2,381, external_blamed_line_pct: 0.7038	false
License	Legal & Compliance	tqdm (viral MPL-2.0 AND MIT)	false
Retention	People	Wentao Ye	false
CI Maturity	Process	ci_files_checked: 12; tests_run_in_ci: false	false
Attack Surface	Security	detect (watermark_detection_server)	false
CVE	Security	packages_queried: 8, unresolved_packages: 2, cve_count: 116	true
Dependency	Supply Chain	direct_dependency_count: 20, pinned_count: 2, ranged_count: 18	true

Rendered 13 of 13 rows. Source: workbook sheet Dashboard.

2. Transaction Recommendation, Conditions & Escalation Thresholds

This section sets out the transaction-level decision logic that translates the technical findings into a structured, board-ready decision frame. The categorical decision tier (DO NOT ACQUIRE) and the economic decision tier (DO NOT ACQUIRE) are invariant across every parameter of the sensitivity sweep, and each condition, deal-breaker, and escalation threshold below is read directly from the stored decision and financial model. The conditions resolve into three distinct deal mechanics, which a buyer negotiates in different places in the acquisition agreement and which are defined in Section 16: condition precedents that must

be satisfied before close, price-adjustment levers that move value rather than block close, and warranties or representations whose breach is an indemnity question rather than a work plan.

2.1 Decision Tiers and Escalation Thresholds

Table 2.1 — Decision and Escalation Parameters

Parameter	Value	Source
Categorical decision tier	DO NOT ACQUIRE	decision.escalation.categorical_tier
Economic decision tier	DO NOT_ACQUIRE	decision.escalation.economic_tier
Conditions escalation threshold	5.0% of proposed price	decision.escalation.conditions_threshold_pct
No-go escalation threshold	25.0% of proposed price	decision.escalation.no_go_threshold_pct
Conditions materiality floor	\$250,000	financial.assumptions.economic_escalation.conditions_materiality_floor
No-go materiality floor	not quantified	financial.assumptions.economic_escalation.no_go_materiality_floor
Economic gate disputed	false	decision.escalation.disputed
Escalation driver	technical register	decision.escalation.driver

The categorical decision tier is governed by the deal-breaker pillars: a critical rating on any of `ai_ml`, `attack_surface`, `cve`, `iac`, `license`, or `secrets` forces the no-go verdict. Two such pillars carry critical ratings in the present scan (`license` and `attack_surface`), so the categorical tier is forced to DO NOT ACQUIRE regardless of remediation economics. The economic decision tier is governed by the escalation thresholds: remediation above five percent of the proposed price and above the 250,000 dollar materiality floor triggers the conditions gate, and remediation above twenty-five percent of the proposed price triggers the no-go gate. The present scan returns 28.7 percent, clearing both the conditions gate and the no-go threshold, so the economic tier is also DO NOT ACQUIRE.

2.2 Conditions for Acquisition

The fifteen conditions below are reproduced verbatim from the stored decision payload and are annotated with the deal mechanic each belongs to. The deal mechanic is declared at the decision layer; it is not inferred from the wording of the condition. Quantification and sizing of the price-adjustment levers, escrow amounts, holdbacks, warranty caps, and indemnity ceilings are deliberately not asserted here. Those numbers are the responsibility of deal counsel after the technical evidence has been independently verified.

Table 2.2 — Conditions for Acquisition with Deal-Mechanic Tags

Mechanic	Pillar	Condition
Condition precedent	Attack Surface	Reduce framework-confirmed external exposure surface
Condition precedent	CI Maturity	Require an automated CI gate as a condition precedent
Price-adjustment lever	Complexity	Include a complexity-reduction milestone in the integration plan
Price-adjustment lever	Coverage	Address coverage findings as a condition precedent
Condition precedent	CVE	Remediate reachable known CVEs before close
Price-adjustment lever	Debt	Include a technical-debt remediation fund in the integration plan
Price-adjustment lever	Dependency	Padlock dependency hygiene and CVE remediation escrow
Price-adjustment lever	Entanglement	Plan decoupling for circular dependency cycles
Price-adjustment lever	Integration	Address integration findings as a condition precedent
Warranty / representation	IP Ownership	Warrant assignability of third-party contributions
Warranty / representation	License	Resolve viral/restrictive license contamination chains (1 viral, 10 unknown)
Price-adjustment lever	Retention	Secure retention agreements for sole-maintainer modules
Price-adjustment lever	Test Deficit	Raise regression coverage on high-risk modules post-close
Condition precedent (deal-breaker)	Attack Surface	Reduce framework-confirmed external exposure surface (deal-breaker)
Warranty / representation (deal-breaker)	License	Resolve viral/restrictive license contamination chains (deal-breaker)

Rendered 15 of 15 rows. Source: `decision.conditions_for_acquisition`. The tag is declared at the decision layer, never inferred from the wording.

2.3 Deal-Breaker Register

Two findings trigger the deal-breaker logic and force the no-go tier. Each is associated with a critical-rated pillar and has been independently verified to the extent that the underlying scanner signal is reproducible; neither has been confirmed by a human reviewer.

Table 2.3 — Deal-Breaker Register

Pillar	Area	Risk	Trigger	Unverified
License	Legal & Compliance	critical	tqdm (viral MPL-2.0 AND MIT)	false
Attack Surface	Security	critical	detect (watermark_detection_server)	false

2.4 Per-Pillar Escalation Rows

Each costed pillar is measured against the two escalation gates on its own. The breach column shows whether a pillar alone clears either the conditions threshold (five percent) or the no-go threshold (twenty-five percent). Complexity is the only pillar that, on its own, would clear the conditions gate; no single pillar alone clears the no-go gate, so the no-go tier is reached only by aggregation.

Table 2.4 — Per-Pillar Cost and Breach View

Pillar	Category	Cost (USD)	% of EV	Breach
Complexity	critical_complexity	\$223,200	11.2%	below both gates on its own
Retention	knowledge_transfer	\$94,600	4.7%	below both gates on its own
Test Deficit	test_investment	\$36,000	1.8%	below both gates on its own
IP Ownership	ip_reassignment	\$15,000	0.8%	below both gates on its own
Debt	general_debt	\$14,880	0.7%	below both gates on its own
Attack Surface	taint_remediation	\$2,400	0.1%	below both gates on its own
Entanglement	decoupling	\$1,000	0.1%	below both gates on its own
Dependency	dependency_pinning	\$900	0.0%	below both gates on its own
CVE	cve_remediation	\$0	0.0%	below both gates on its own
Maintenance	staleness_remediation	\$0	0.0%	below both gates on its own

Percentages derived as $\text{total_cost} / \text{proposed_price} \times 100$ from two stored figures. The breach column applies each pillar's individual cost against the escalation gates; aggregation across pillars is what clears the no-go threshold.

2.5 Sensitivity Sweep Outcomes

The recommendation is invariant across every parameter of the sensitivity sweep. The categorical decision tier remains DO NOT ACQUIRE in all seventeen scenarios, and the confidence score remains 72 out of 100 throughout. The total remediation cost moves only with the hourly_rate multiplier; the other multipliers affect the underlying components without changing the headline decision or the headline tier.

Table 2.5 — Sensitivity Sweep Outcomes

Parameter	Multiplier	Board Decision	Confidence	Total Remediation Cost
complexity_multiplier	0.7	DO NOT ACQUIRE	72	\$560,180
complexity_multiplier	1.0	DO NOT ACQUIRE	72	\$573,680
complexity_multiplier	1.3	DO NOT ACQUIRE	72	\$587,180
discount_rate	0.7	DO NOT ACQUIRE	72	\$573,680
discount_rate	1.0	DO NOT ACQUIRE	72	\$573,680
discount_rate	1.3	DO NOT ACQUIRE	72	\$573,680
hourly_rate	0.7	DO NOT ACQUIRE	72	\$401,576
hourly_rate	1.0	DO NOT ACQUIRE	72	\$573,680
hourly_rate	1.3	DO NOT ACQUIRE	72	\$745,784
max_total_hours	0.7 / 1.0 / 1.3	DO NOT ACQUIRE	72	\$573,680
remediation_timing	not quantified	not quantified	not quantified	not quantified
scenario_weights	not quantified	not quantified	not quantified	not quantified
valuation_multiplier	0.7 / 1.0 / 1.3	DO NOT ACQUIRE	72	\$573,680

Rendered 13 of 17 sweep rows; grouped multipliers collapsed for readability. Full sweep detail is in Section 3.21.

PART II

DEAL THESIS, TRANSACTION IMPLICATIONS & FINANCIAL EXPOSURE

3. Remediation Cost Model, Valuation Impact & Sensitivity

This section is the bridge between the technical work and the deal model. Every value that can move a number in this report is declared once in the parameters table below: financial inputs, saturation calibration, valuation and integration-drag curves, confidence weights, decision thresholds, post-close schedule constants, and health-posture ceilings.

Where a parameter was not configured by the analyst, it is listed as not quantified rather than assumed away. The confidence interval and the sensitivity analysis are what allow the recommendation to survive challenge from the seller's quality-of-earnings provider; both are reproduced verbatim from the stored payload.

3.1 Parameters Used Downstream

The parameters below are grouped by their function in the model. Each row identifies the parameter, its value, the part of the model it affects, and the source path. The line-item costs in Section 3.2 are direct products of these parameters; readers seeking to re-derive any figure should begin here.

Table 3.1 — Model Parameters

Group	Parameter	Value	Affects
Remediation inputs	Hourly rate	\$50	Every line-item cost = saturated hours × rate
Remediation inputs	Complexity multiplier	1.5	Hours per unit for hard-to-remediate complexity hotspots
Remediation inputs	Valuation multiplier	3	Amplifies remediation cost into the valuation discount
Remediation inputs	Integration drag factor	0.15%	Post-merge integration friction as hours / (8 × 260 × factor)
Remediation inputs	Productive hours per quarter	624 h	Net quarterly working hours behind per-cause effort estimates
Remediation inputs	Confidence interval width	30%	Width of the low/high remediation-cost interval
Remediation inputs	Annual working hours	2,080 h	Person-years = remediation hours / 2,080
Remediation inputs	Discount rate	not quantified	No present-value discounting applied (unrecorded, not zero)

Group	Parameter	Value	Affects
Aggregate hours cap	Max total hours	20,000 h	Aggregate ceiling on remediation hours; surplus above is decayed
Aggregate hours cap	Surplus factor	0.25	Fraction of hours above max-total-hours that is credited
Valuation & drag curves	Valuation discount cap	25%	Maximum valuation discount on the proposed price
Valuation & drag curves	Integration drag cap	80%	Maximum post-merge integration drag
Confidence & evidence	Base confidence	20	Base component of the confidence score
Confidence & evidence	Coverage weight	40	Weight applied to the coverage share
Confidence & evidence	Risk-signal weight	40	Weight applied to the risk-signal share
Confidence & evidence	Zero-coverage cap	20	Confidence score floored at 20 when coverage is 0
Confidence & evidence	No-go floor	60	Confidence score floored at 60 when the verdict is no-go
Decision & escalation	Conditions threshold	5% + \$250,000 materiality floor	Trigger for the conditions gate
Decision & escalation	No-go threshold	25% (no materiality floor recorded)	Trigger for the no-go gate
Decision & escalation	Deal-breaker pillars	ai_ml, attack_surface, cve, iac, license, secrets	Critical rating forces the no-go verdict
Schedule (post-close plan)	Execution teams	2 (× 2 engineers)	Post-close remediation planned against two two-engineer teams
Schedule (post-close plan)	Workdays per sprint	10	Sprint capacity = teams × 2 engineers × workdays
Schedule (post-close plan)	Sprint cap	12	Hours beyond this many sprints are reported unplanned

Group	Parameter	Value	Affects
Schedule (post-close plan)	Hours per workday	4	Roadmap durations = effort-hours / 4 h per workday

Rendered 24 of 49 parameter rows. The full parameter table — including saturation per cost category, health-posture ceilings, and additional schedule constants — is reproduced in Appendix B.

3.2 Remediation Line Items

Nine cost categories are charged in the present scan. Each line item is derived from the count of underlying findings multiplied by an hours-per-unit rate, with the rate saturated by a per-category cap (above which the per-unit hours decay to a lower value). The most expensive single line item is critical-complexity remediation, accounting for thirty-nine percent of total remediation cost; together, critical and high complexity account for seventy-one percent. Four categories are explicitly zero this scan (critical_path_debt, staleness_remediation, sole_maintainer_risk, sqale_remediation), reflecting the scanner's deliberate decision not to price those categories rather than the absence of underlying findings.

Table 3.2 — Remediation Line Items

Category	Count	Full HPU	Cap	Decay	Applied HPU	Total Hours	Total Cost	Notes
critical_complexity	941	18	50	4	4.74	4,464 h	\$223,200	Critically complex functions requiring refactoring
high_complexity	1,557	6	150	2	2.39	3,714 h	\$185,700	High-complexity functions (exclusive of critical tier)
knowledge_transfer	443	8	30	4	4.27	1,892 h	\$94,600	Critical sole-maintainer modules (costed per unique module, not per at-risk contributor)
test_investment	120	16	20	4	6	720 h	\$36,000	High-regression-risk modules requiring test investment

Category	Count	Full HPU	Cap	Decay	Applied HPU	Total Hours	Total Cost	Notes
ip_reassignment	100	4	50	2	3	300 h	\$15,000	Files with external-dominant authorship requiring IP review
general_debt	2,176	0.5	200	0.1	0.14	298 h	\$14,880	Confirmed technical-debt markers (in-comment annotations + tooling suppressions)
taint_remediation	6	8	20	4	8	48 h	\$2,400	Taint flows requiring sanitisation
decoupling	1	20	10	8	20	20 h	\$1,000	Circular dependency cycles requiring decoupling
dependency_pinning	18	1	100	0.5	1	18 h	\$900	Unpinned dependencies requiring version locking

Rendered 9 of 9 rows. Total hours (11,474) and total cost (\$573,680) reconcile to the headline figures. Source: workbook sheet Dashboard.

3.3 Remediation Cost Confidence Interval

The confidence interval is a stated-confidence range derived from the model's calibration, not a stochastic bound. Its width is fixed at thirty percent of the hours-per-unit estimates, reflecting the model's acknowledgement that the underlying HPU constants are unbenchmarked. The low and high bounds are 401,576 dollars and 745,784 dollars respectively; the central estimate is 573,680 dollars. Buyers should treat this interval as a minimum uncertainty band; the actual uncertainty is wider still because the model has not been benchmarked against an external delivery dataset.

Table 3.3 — Remediation Cost Confidence Interval

Metric	Value
Basis	±30% on hours-per-unit estimates
Low	\$401,576

Metric	Value
High	\$745,784
Width (%)	30

3.4 Aggregate Hours Cap, Saturation Model, and Line-Item Basis

The aggregate hours cap is configured but did not bind in the present scan, because the saturated total of 11,474 hours is well below the 20,000-hour ceiling. The surplus factor of 0.25 would apply only to hours above the ceiling; in this scan, no hours are surplus. The saturation model is reproduced in Appendix B and shows that every cost category uses the tool's built-in default calibration factor of 1, with `calibration.applied` set to false; this confirms the planning-bound caveat stated in the executive summary.

Table 3.4 — Aggregate Hours Cap Status

Metric	Value
Applied	false (did not bind)
Max total hours	20,000
Scale factor	1
Surplus factor	0.25

3.5 SQALE Remediation Basis and Double-Counting Ledger

The SQALE (Software Quality Assessment based on Lifecycle Expectations) remediation basis is reconciled against the complexity-hotspot line item so that the same modules are not priced twice. The SQALE engine reports 97,795.2 remediation-effort days across 5,711 modules, of which 43 are graded D or E and overlap with the complexity-hotspot population. The double-counting ledger has been audited and returns a clean verdict: shared-registry keys appear on multiple surfaces but are added only once, with the overlapping keys (`layer1.total_cost`, `layer1.integration_drag_pct`, `layer2.aggregate_residual_expected_loss`) explicitly justified in Section 3.38.

Table 3.5 — SQALE Reconciliation

Metric	Value
Costed as complexity hotspot	43 modules
D-E graded modules	43 modules
Deduped uncoded modules	0
Hours equivalent	782,362 h

Metric	Value
Line item added	false (SQALE debt is reconciled against complexity, not priced twice)
SQALE remediation effort (days)	97,795.2
Double-counting ledger clean verdict	true

3.6 Risk Valuation Lines (Expected Loss)

Eleven risk-valuation lines were emitted in the present scan. One is quantified (the complexity risk associated with `sparse_attn_indexer_kpool`), and ten are deliberately not quantified because `financial.risk_parameters` ships empty. The quantified line carries a probability-before of 45 percent, an economic-impact-before of 750,000 dollars, an expected-loss-before of 337,500 dollars, and an expected-loss-after of 50,000 dollars, producing a risk-reduction value of 287,500 dollars. The other ten lines render the linked Layer-1 remediation evidence per line and a single reason rather than a column of identical unquantified cells; a value that reads not quantified is deliberately unquantified, never a silent zero.

Table 3.6 — Risk Valuation Lines

Pillar	Title	State	Expected Loss Before	Expected Loss After	Risk Reduction
Complexity	<code>sparse_attn_indexer_kpool</code>	ASSESSED_WITH_FINDINGS	\$337,500	\$50,000	\$287,500
Attack Surface	<code>detect</code>	NOT_QUANTIFIED	not quantified	not quantified	not quantified
CI Maturity	<code>ci_maturity risk</code>	NOT_QUANTIFIED	not quantified	not quantified	not quantified
Coverage	<code>coverage risk</code>	NOT_QUANTIFIED	not quantified	not quantified	not quantified
Debt	<code>fixme</code>	NOT_QUANTIFIED	not quantified	not quantified	not quantified
Entanglement	<code>entanglement risk</code>	NOT_QUANTIFIED	not quantified	not quantified	not quantified
Integration	<code>integration risk</code>	NOT_QUANTIFIED	not quantified	not quantified	not quantified

Pillar	Title	State	Expected Loss Before	Expected Loss After	Risk Reduction
IP Ownership	ip_ownership risk	NOT_QUANTIFIED	not quantified	not quantified	not quantified
License	tqdm	NOT_QUANTIFIED	not quantified	not quantified	not quantified
Retention	Wentao Ye	NOT_QUANTIFIED	not quantified	not quantified	not quantified
Test Deficit	test_deficit risk	NOT_QUANTIFIED	not quantified	not quantified	not quantified

Rendered 11 of 11 rows. When a line is quantified, the before/after probability and impact columns are the stored inputs behind the expected-loss product, so a buyer can re-derive the number rather than trust it.

3.7 Scenario Branches (0.3.0)

Four scenario branches are enabled in the present scan. Each branch represents a distinct post-close decision: acquire and remediate post-close, remediate before close, partial remediation, and the material failure / downside case. The weighted expectation across scenarios is 1,211,434 dollars; this is the scenario-based technology VaR (value-at-risk) honest label, and it is a technology-risk model output rather than a market-risk measure.

Table 3.7 — Scenario Branches

Scenario	Title	Enabled	Evidence Basis	Total Impact
acquire_remediate_post_close	Acquire and remediate post-close	true	transaction EV + layer-1 remediation + integration drag	\$2,023,742
material_failure	Material failure / downside case	true	analyst calibration on documented scan evidence	\$90,194
partial_remediation	Partial remediation	true	config calibration (analyst_calibration)	\$314,444
remediate_before_close	Remediate before close	true	layer-1 remediation schedule + layer-2 residual	\$582,212

Disabled branches are not-modeled, not zero: their not-modeled reason is stated, and the technology-VaR and transaction-impact columns then read not quantified rather than 0.

3.8 Deal Economics Headline Valuation Reconciliation

The deal-economics arithmetic as stored is reproduced below. The headline enterprise value of two million dollars is reduced by a net quantified technology impact of 1,873,680 dollars to produce an adjusted economic value of 1,500,000 dollars. The price adjustment of 500,000 dollars is capped by the 25 percent valuation-adjustment cap, which binds in this scan. The size of any escrow, holdback, warranty, or indemnity is a qualitative option for deal counsel and is deliberately not a number here.

Table 3.8 — Deal Economics Reconciliation

Component	Impact Type	Value (USD)	Evidence State
Headline EV	REVENUE_DISRUPTION	\$2,000,000	QUANTIFIED_WITH_EVIDENCE
Integration / separation TSA	INTEGRATION_SEPARATION_COST	\$1,600,000	QUANTIFIED_WITH_EVIDENCE
Net quantified tech impact	REMEDIATION_COST	\$1,873,680	QUANTIFIED_WITH_EVIDENCE
Other quantified tech exposure	SECURITY_INCIDENT	\$150,000	QUANTIFIED_WITH_EVIDENCE
Potential synergy	CUSTOMIZATION_BURDEN	\$500,000	QUANTIFIED_WITH_EVIDENCE
Remediation cost	REMEDIATION_COST	\$573,680	QUANTIFIED_WITH_EVIDENCE
Residual expected loss	OPERATIONAL_DOWNTIME	\$50,000	QUANTIFIED_WITH_EVIDENCE

The headline reconciliation is therefore: a proposed price of 2,000,000 dollars, a price adjustment of 500,000 dollars (capped at 25 percent of the proposed price), and an adjusted economic value of 1,500,000 dollars. The cap binds because the unadjusted price adjustment would exceed twenty-five percent of the proposed price. This is the figure deal counsel should use as the starting position for negotiation; it is not a final price.

3.9 Sensitivity Sweep Detail

The sensitivity sweep stress-tests the headline recommendation against reasonable variations in the four most material model parameters: `complexity_multiplier`, `hourly_rate`, `valuation_multiplier`, and `max_total_hours`. The `discount_rate` multiplier has no effect

because no present-value discounting is applied in the present scan. The remediation_timing and scenario_weights sweeps are not quantified because the model carries no calibration for them; they are explicitly flagged rather than silently defaulted.

Table 3.9 — Sensitivity Sweep Details

Parameter	Multiplier	Value	Total Hours	Total Cost	Val. Disc. %	Drag %	Decision
complexity_multiplier	0.7	1.05	11,204 h	\$560,180	21.94	80	DO NOT ACQUIRE
complexity_multiplier	1.0	1.5	11,474 h	\$573,680	22.09	80	DO NOT ACQUIRE
complexity_multiplier	1.3	1.95	11,744 h	\$587,180	22.24	80	DO NOT ACQUIRE
discount_rate	0.7 / 1.0 / 1.3	—	11,474 h	\$573,680	22.09	80	DO NOT ACQUIRE
hourly_rate	0.7	35	11,474 h	\$401,576	19.45	80	DO NOT ACQUIRE
hourly_rate	1.0	50	11,474 h	\$573,680	22.09	80	DO NOT ACQUIRE
hourly_rate	1.3	65	11,474 h	\$745,784	23.47	80	DO NOT ACQUIRE
max_total_hours	0.7 / 1.0 / 1.3	—	11,474 h	\$573,680	22.09	80	DO NOT ACQUIRE
valuation_multiplier	0.7	2.1	11,474 h	\$573,680	19.45	80	DO NOT ACQUIRE
valuation_multiplier	1.0	3	11,474 h	\$573,680	22.09	80	DO NOT ACQUIRE
valuation_multiplier	1.3	3.9	11,474 h	\$573,680	23.47	80	DO NOT ACQUIRE

Grouped multipliers collapsed where three multipliers return identical results. The remediation_timing and scenario_weights sweeps are not quantified because no calibration is present.

3.10 Financial Lineage and Reproduction Fingerprint

The financial lineage trace binds every figure in this section to the source artifact that produced it. The reproduction fingerprint (SHA-256 over tool_version, commit, repo_path, mode, and offline flag) binds the report to the exact configuration used: same inputs in, same outputs out. Any change to a number-driving input produces a different fingerprint and a different report under a new fingerprint. The full lineage is reproduced in Appendix O.

Table 3.10 — Financial Lineage

Step	Artifact	Note
artifacts	workbook.xlsx	Remediation_Costs, Cal_Lineage, Cost_Drivers, Sensitivity, Assumptions mirror report.json#/financial
decision	report.json#/decision	generate_board_decision(pillars, financial) from risk register + remediation cost
fingerprint	report.json#/financial/fingerprint	SHA-256 over {tool_version, commit=fa75d3456d60a5ffe923818c4b495295675b38e1, repo_path=vllm, mode=deal_thesis=True, offline=False, scan_history=True}
model	report.json#/financial	compute_financial_impact(config): pillar count → hours-per-unit (saturated) → hours → rate → cost → valuation discount
pillars	report.json#/pillars	Per-pillar raw reports; the cost model consumes summary counts plus bounded top evidence rows only
scan	VLLM_cpg_261016T0730.db	Symbol/index graph that backs every pillar scanner (stored under <output>/internal/)

PART III

MATERIAL RISKS BY DEAL SIGNIFICANCE

4. Licensing & IP Exposure

This section sets out the license posture of the packaged components that ship in the target, together with the third-party ownership structure of the source files. License contamination and IP ownership are the two legal pillars whose findings most directly shape the warranty and indemnity chapter of the acquisition agreement, and whose breach most directly forces re-pricing rather than remediation. A viral or restrictive copyleft license embedded in a shipped dependency can trigger customer disclosure obligations, restrict integration into a closed-source acquirer product, and require carve-outs or re-licensing that materially affect post-close execution. IP ownership, in turn, drives title and warranty risk: where a significant share of code is authored by external contributors without explicit assignment, the buyer's chain of title is weakened and the indemnity surface widens.

4.1 Headline Posture

The license pillar is rated critical, with a health score of 15 out of 100, and is one of two deal-breaker pillars in the present scan. Twenty-two license findings are recorded across twenty distinct packages, of which one is viral (tqdm, under an MPL-2.0 AND MIT

composite), zero are restrictive in the strict sense, and ten are classified UNKNOWN. One contamination chain is identified (the tqdm chain), and 547 IP-ownership attribution rows are recorded across two hundred attributed files. The IP ownership pillar is rated high, with a health score of 40 out of 100, an IP risk score of 0.5815, and 70.38 percent of blamed lines attributed to external authors.

4.2 Material Findings

The five most material license findings are reproduced below. Each is a scanner-detected signal that has not been independently verified by a human reviewer and that must be validated before it is relied on in negotiation. The M&A implication column translates the technical signal into a deal-mechanic recommendation; sizing of any escrow or indemnity is for deal counsel.

Table 4.1 — Material License Findings

ID	Package	Severity	License	Source File	M&A Implication
F-license-001	tqdm	medium	MPL-2.0 AND MIT (viral)	benchmarks/multi_turn/requirements.txt	Obtain a representation and warranty covering license compliance for the shipped product, with an indemnity for any copyleft exposure
F-license-002	pandas	low	BSD-3-Clause	benchmarks/kernels/requirements.txt	Obtain a representation and warranty covering license compliance; not a viral exposure
F-license-003	numpy	low	BSD-3-Clause	benchmarks/multi_turn/requirements.txt	Obtain a representation and warranty covering license compliance; not a viral exposure
F-license-004	pandas	low	BSD-3-Clause	benchmarks/multi_turn/requirements.txt	Obtain a representation and warranty covering license compliance; not a viral exposure

ID	Package	Severity	License	Source File	M&A Implication
F-license-005	aiohttp	low	Apache-2.0	benchmarks/multi_turn/requirements.txt	Obtain a representation and warranty covering license compliance; not a viral exposure

Each row is a scanner signal, not a validated exposure. Validation against the actual shipped bill of materials is required before relying on these findings in negotiation.

4.3 Per-Package Detected Licenses

The per-package license inventory below is the full set of detected license assignments in the present scan. The tqdm finding is the only viral signal, and ten packages carry UNKNOWN classifications that must be resolved by manifest metadata or direct vendor confirmation before close. The repo-level license is Apache-2.0, with a secondary MIT license in the third_party/flash_linear_attention directory; both are permissive and do not constrain integration.

Table 4.2 — Per-Package Detected Licenses

Package	Detected License	Viral	Restrictive	Confidence	Source File
tqdm	MPL-2.0 AND MIT	true	false	0.7	benchmarks/multi_turn/requirements.txt
(repo-level)	Apache-2.0	false	false	0.8	LICENSE
(repo-level)	MIT	false	false	0.8	third_party/flash_linear_attention/LICENSE
aiohttp	Apache-2.0	false	false	0.9	benchmarks/multi_turn/requirements.txt
matplotlib	PSF-2.0	false	false	0.9	tools/recipes/requirements.txt
native-tls-vendored	UNKNOWN	false	false	0	rust/src/cmd/Cargo.toml
numpy	BSD-3-Clause	false	false	0.9	benchmarks/multi_turn/requirements.txt
openai	UNKNOWN	false	false	0	examples/features/structured_outputs/pyproject.toml
pandas	BSD-3-Clause	false	false	0.9	benchmarks/kernels/requirements.txt
pandas	BSD-3-Clause	false	false	0.9	benchmarks/multi_turn/requirements.txt

Package	Detected License	Viral	Restrictive	Confidence	Source File
pandas	BSD-3-Clause	false	false	0.9	tools/recipes/requirements.txt
pydantic	MIT	false	false	0.9	examples/features/structured_outputs/pyproject.toml
reqwest	UNKNOWN	false	false	0	rust/src/bench/Cargo.toml
seaborn	UNKNOWN	false	false	0	tools/recipes/requirements.txt
serde	UNKNOWN	false	false	0	rust/src/bench/Cargo.toml
serde_json	UNKNOWN	false	false	0	rust/src/bench/Cargo.toml
tempfile	UNKNOWN	false	false	0	rust/src/engine-core-client/Cargo.toml
tokio	UNKNOWN	false	false	0	rust/src/cmd/Cargo.toml

Rendered 18 of 22 rows. The full inventory is in Appendix E. Confidence values reflect the scanner's certainty in the license assignment; zero-confidence UNKNOWN entries are unresolvable without manifest metadata or vendor confirmation.

4.4 License Clause Findings

Three license clause findings are recorded. The Apache-2.0 LICENSE file at the repository root carries an attribution requirement (all copyright, patent, trademark, and attribution notices must be preserved in derivative works), a permissive patent grant, and the standard warranty disclaimer. The MIT license in the third_party/flash_linear_attention directory carries the standard warranty disclaimer. None of these clauses is restrictive in the strict sense, but the attribution requirement does generate a customer-disclosure obligation that the acquirer's legal counsel must factor into the integration plan.

Table 4.3 — License Clause Findings

License File	Clause Type	Severity	Matched Packages	Clause Text (excerpt)
LICENSE	attribution	attribution	0 entries	any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form...
third_party/flash_linear_attention/LICENSE	warranty_disclaimer	permissive	0 entries	shall be included in all copies or substantial portions of the Software. THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND...
LICENSE	patent_grant	permissive	0 entries	sublicense, and distribute the Work and such Derivative Works in Source or Object form. 3. Grant of Patent License. Subject to the...

4.5 IP Ownership Metrics

IP ownership attribution is a diligence signal, not legal proof of ownership. The metrics below describe the contributor concentration and external-share posture of the two hundred attributed files; legal title must be established by reviewing contributor licence agreements (CLAs), IP-assignment agreements, and contractor terms. The corporate domain is `users.noreply.github.com`, which is the dominant attribution domain; this is consistent with a project that has grown through a large open-source contributor base rather than through employed engineers.

Table 4.4 — IP Ownership Metrics

Metric	Value
Attributed file count	200
Corporate domain	<code>users.noreply.github.com</code>
Total authors	3,793
External author count	2,381
External domain count	365
External owned files	100
External blamed line %	0.7038 (70.38%)
External commit %	0.7038 (70.38%)
High-risk files	100 entries
IP risk score	0.5815

The 70.38 percent external blamed-line share is the headline IP ownership concern. Combined with the 2,381 distinct external authors and the 365 distinct external domains, this means that a substantial majority of the codebase has been contributed by individuals outside any employment relationship with the target. Without executed contributor assignment agreements on file for each of these authors, the buyer's chain of title is materially weakened. The condition precedent requiring the seller to warrant assignability of third-party contributions (Section 2.2) is the operational expression of this risk; the indemnity surface is sized by deal counsel after the CLA inventory has been independently reviewed.

4.6 Transaction Implications

The licensing and IP findings drive two distinct deal mechanics. The license finding drives a warranty/representation mechanic: the seller warrants license compliance for the shipped product, with an indemnity for any copyleft exposure that emerges post-close. The IP ownership finding drives a parallel warranty/representation mechanic: the seller warrants assignability of third-party contributions, again backed by an indemnity. Neither is a priced line item in the remediation model, because both are legal rather than engineering obligations; their cost materialises only on breach. The buyer should commission an

independent CLA audit before close, scoped to the 2,381 external authors and the 100 high-risk files identified by the attribution analysis. Where executed CLAs cannot be located, the buyer should require either a price reduction reflecting the cost of obtaining retroactive assignments, or an indemnity escrow sized to the cost of substitute engineering in the event a contributor declines to assign.

5. Security, Vulnerability & Attack Surface Posture

This section sets out the end-to-end security picture of the target: known advisory exposure against the resolvable package population, the critical-severity attack-surface posture, secret hygiene, infrastructure-as-code findings, the candidate external-reachable endpoint inventory, and privacy-relevant data flows. Security findings drive three distinct M&A concerns: cyber-insurance sizing (the premium the acquirer will pay to underwrite the post-close security surface), regulatory disclosure timing (the window within which a material vulnerability must be disclosed to customers and regulators), and the cyber-warranty language in the acquisition agreement (the seller's representations about the security state of the asset). Coverage gaps in any of these dimensions are a direct credibility test of the seller's claims.

5.1 Headline Posture and Cross-Pillar Reconciliation

Two pillars in this section are rated critical: attack surface (health 15) and license (already discussed in Section 4). The attack-surface pillar is the second of the two deal-breaker pillars and is associated with 2,154 candidate endpoints, of which 2,142 are untested. The CVE pillar returns 116 advisory matches against the eight resolvable packages, with zero critical CVEs but with no fix available for any of the 116; the unresolved-package count of two means the advisory check covered only the resolvable subset. The IaC pillar is rated medium, with 11 medium-severity findings, all of which are the same rule (Container runs as root) repeated across eleven Dockerfiles. The secrets pillar is rated low and is partially assessed: six generic-password-assignment candidates were detected, none in git history, none at high confidence, and three path-excluded candidate paths were not scanned.

An apparent contradiction between the CVE severity count (zero critical CVEs) and the pillar's overall posture (high risk) is resolved by the scope note in Section 15: the advisory check covered only the eight resolvable packages out of a declared twenty, and the two unresolved packages include aiohttp (which alone carries the majority of the 116 advisories). The zero-critical-CVE figure is therefore a partial-population result, not a clean bill of health. The buyer should treat the CVE pillar as UNVERIFIED and require a lockfile-backed SBOM and a continuous SCA feed as conditions precedent to any revised offer.

5.2 CVE Inventory and Advisory Coverage

The advisory source is osv.dev, queried at 2026-10-16T07:34:13 UTC. Of the eight packages queried, three return advisory matches: aiohttp (3.10), tqdm (4.66), and transformers (4.46). The full inventory of 116 CVEs is reproduced in Appendix D; the first eighteen entries are shown below to illustrate the pattern. Every CVE returns UNKNOWN severity (osv.dev does not propagate a CVSS score for these entries) and no fix is available for any of them. This is unusual and reflects either a version-specification problem (the ranges specified in the manifests do not match any patched release) or a scanner-side limitation; the buyer should commission an independent SCA pass against a resolved lockfile to confirm.

Table 5.1 — CVE Inventory (First 18 of 116)

ID	Severity	Package	Ecosystem	Version	Fixed Available	Remediation Class
GHSA-29pf-2h5f-8g72	UNKNOWN	transformers	pip	4.46	false	no_fixed_release
GHSA-2fqr-mr3j-6wp8	UNKNOWN	aiohttp	pip	3.10	false	no_fixed_release
GHSA-2vrm-gr82-f7m5	UNKNOWN	aiohttp	pip	3.10	false	no_fixed_release
GHSA-37mw-44qp-f5jm	UNKNOWN	transformers	pip	4.46	false	no_fixed_release
GHSA-3wq7-rqq7-wx6j	UNKNOWN	aiohttp	pip	3.10	false	no_fixed_release
GHSA-4fvr-rgm6-gqmc	UNKNOWN	aiohttp	pip	3.10	false	no_fixed_release
GHSA-4m7w-qmgq-4wj5	UNKNOWN	aiohttp	pip	3.10	false	no_fixed_release
GHSA-4w7r-h757-3r74	UNKNOWN	transformers	pip	4.46	false	no_fixed_release
GHSA-54jq-c3m8-4m76	UNKNOWN	aiohttp	pip	3.10	false	no_fixed_release
GHSA-59p9-h35m-wg4g	UNKNOWN	transformers	pip	4.46	false	no_fixed_release
GHSA-63hf-3vf5-4wqf	UNKNOWN	aiohttp	pip	3.10	false	no_fixed_release
GHSA-63hw-fmq6-xxg2	UNKNOWN	aiohttp	pip	3.10	false	no_fixed_release
GHSA-69f9-5gxw-wvc2	UNKNOWN	aiohttp	pip	3.10	false	no_fixed_release

ID	Severity	Package	Ecosystem	Version	Fixed Available	Remediation Class
GHSA-69w3-r845-3855	UNKNOWN	transformers	pip	4.46	false	no_fixed_release
GHSA-6jhg-hg63-jvuf	UNKNOWN	aiohttp	pip	3.10	false	no_fixed_release
GHSA-6mq8-rvhq-8wgg	UNKNOWN	aiohttp	pip	3.10	false	no_fixed_release
GHSA-6rvg-6v2m-4j46	UNKNOWN	transformers	pip	4.46	false	no_fixed_release
GHSA-8495-4g3g-x7pr	UNKNOWN	aiohttp	pip	3.10	false	no_fixed_release

Rendered 18 of 116 rows. Source: workbook sheet CVEs. Every CVE returns UNKNOWN severity and no fix; the buyer should commission an independent SCA pass against a resolved lockfile.

5.3 Attack Surface Composition

The attack-surface scanner identified 2,154 candidate endpoints across the codebase. Of these, 1,062 are unreferenced (no incoming edges detected) and 62 are dynamic (defined at runtime). External reachability could not be established for 2,082 of the 2,154 candidates because framework semantics were unresolved, and 2,142 of the 2,154 are untested. Only 72 of the 2,154 candidates could be resolved to a recognised framework (express, fastapi, or flask). The candidate exposure score is 0.311 on a 0-1 scale; this is an upper-bound signal for review prioritisation, not a confirmed vulnerability rate.

Table 5.2 — Attack Surface Summary Metrics

Metric	Value
Total candidate endpoints	2,154
Unreferenced endpoints	1,062
Dynamic endpoints	62
Untested candidate endpoints	2,142
Framework identified	72
Framework unavailable	2,082
Candidate exposure score	0.311 (upper-bound review signal)
Frameworks detected	express, fastapi, flask

The candidate endpoint detail below illustrates the concentration of risk in the OpenAI-compatible entypoints and the reinforcement-learning online router. Every endpoint in the table below is rated critical, externally exposed, and untested. The cancel_responses and abort_requests endpoints in particular are operational control surfaces (request cancellation

and abort) that, if reachable without authentication, would allow a remote party to disrupt in-flight generation workloads.

Table 5.3 — Candidate Endpoint Detail (Sample of 14 of 2,154)

Symbol	File	Line	Type	Exposure	Framework	Risk	Tested
health	entrypoints/cli/preload.py	75	http_handler	external	express	critical	untested
health	entrypoints/launchers/dp_supervisor.py	229	http_handler	external	express	critical	untested
ready	entrypoints/launchers/dp_supervisor.py	234	http_handler	external	express	critical	untested
show_available_models	entrypoints/openai/models/api_router.py	21	http_handler	external	fastapi	critical	untested
cancel_responses	entrypoints/openai/responses/api_router.py	112	http_handler	external	fastapi	critical	untested
retrieve_responses	entrypoints/openai/responses/api_router.py	82	http_handler	external	fastapi	critical	untested
create_classify	entrypoints/pooling/classify/api_router.py	29	http_handler	external	fastapi	critical	untested
is_paused	entrypoints/rl/online/api_router.py	143	http_handler	external	fastapi	critical	untested
init_weight_transfer_engine	entrypoints/rl/online/api_router.py	160	http_handler	external	fastapi	critical	untested
shutdown_weight_transfer_engine	entrypoints/rl/online/api_router.py	172	http_handler	external	fastapi	critical	untested
start_weight_update	entrypoints/rl/online/api_router.py	179	http_handler	external	fastapi	critical	untested
start_draft_weight_update	entrypoints/rl/online/api_router.py	186	http_handler	external	fastapi	critical	untested
finish_weight_update	entrypoints/rl/online/api_router.py	205	http_handler	external	fastapi	critical	untested
update_weight_version	entrypoints/rl/online/api_router.py	215	http_handler	external	fastapi	critical	untested

Rendered 14 of 2,154 rows. Full inventory in Appendix H.

5.4 Secrets, IaC, and Privacy Posture

Six generic-password-assignment candidates were detected, all at confidence levels of 60 or 61 out of 100 (medium cap), none in git history, and none at high confidence. Three of the six are in the swagger-ui-bundle.js static asset (false positives reflecting property

assignment in JavaScript rather than credential handling); three are in the `docker/entrypoints/test_vllm_nonroot_entrypoint.sh` test script. None is a confirmed live secret. The path-excluded candidate count of three means three candidate paths were not scanned, so the secrets pillar is partially assessed rather than fully verified.

Eleven IaC findings are recorded, all medium severity and all of the same rule: IAC-DOCKER-ROOT (Container runs as root). The affected files include nine production Dockerfiles, one documentation asset (a PNG image, which is a false positive), and one documentation file. The remediation is uniform: add a dedicated non-root user and USER directive in every stage. This is a low-effort remediation that should be a condition precedent to close.

Eighteen privacy findings are recorded, of which sixteen are PRIV-SENSITIVE-LOG (sensitive data in log statement, token) at medium severity. The affected files are concentrated in benchmarks and CI workflows, which reduces but does not eliminate the concern: tokens logged in benchmark scripts may still be production tokens if the benchmarks are run against production endpoints. Two additional findings (PRIV-NO-CONSENT-POLICY at low severity, and PRIV-EGRESS-SDK at low severity) flag the absence of consent-policy signals and the presence of a third-party data-egress SDK (the `llm_api` family), respectively. The buyer should commission an independent privacy review scoped to the credential inventory and the egress SDK family.

Table 5.4 — IaC Findings (Container Runs as Root)

Rule	Title	Severity	File	Line	Remediation
IAC-DOCKER-ROOT	Container runs as root	medium	docker/Dockerfile.cpu	494	Add a dedicated non-root user and USER directive in every stage
IAC-DOCKER-ROOT	Container runs as root	medium	docker/Dockerfile.rock	1,018	Add a dedicated non-root user and USER directive in every stage
IAC-DOCKER-ROOT	Container runs as root	medium	docker/Dockerfile.rock_base	453	Add a dedicated non-root user and USER directive in every stage
IAC-DOCKER-ROOT	Container runs as root	medium	docker/Dockerfile.room	1,018	Add a dedicated non-root user and USER directive in every stage

Rule	Title	Severity	File	Line	Remediation
IAC-DOCKER-ROOT	Container runs as root	medium	docker/Dockerfile.rocm_base	479	Add a dedicated non-root user and USER directive in every stage
IAC-DOCKER-ROOT	Container runs as root	medium	docker/Dockerfile.rocm_base_gfx1250	315	Add a dedicated non-root user and USER directive in every stage
IAC-DOCKER-ROOT	Container runs as root	medium	docker/Dockerfile.rocm_gfx1250	744	Add a dedicated non-root user and USER directive in every stage
IAC-DOCKER-ROOT	Container runs as root	medium	docker/Dockerfile.xpu	286	Add a dedicated non-root user and USER directive in every stage
IAC-DOCKER-ROOT	Container runs as root	medium	docker/Dockerfile.zen	98	Add a dedicated non-root user and USER directive in every stage

Rendered 9 of 11 rows (two remaining are documentation false positives). Source: workbook sheet IaC Findings.

5.5 Taint Analysis Summary

Six taint flows were identified, and the taint pass is reported as unbounded (the analysis was not terminated by a depth or time limit). Taint flows are not priced as a separate line item; they are subsumed within the taint_remediation cost category in Section 3.2, which carries 48 hours and 2,400 dollars across the six flows. The full taint-flow detail is reproduced in Appendix N.

5.6 Transaction Implications

The security findings drive three distinct deal mechanics. First, the attack-surface finding is a condition precedent (deal-breaker): the framework-confirmed external exposure surface must be reduced before close. Second, the CVE finding is a condition precedent (non-deal-breaker): reachable known CVEs must be remediated before close, scoped to the eight resolvable packages in the first instance and extended to the full lockfile-backed population once a lockfile is available. Third, the IaC finding is a price-adjustment lever: the eleven Container-runs-as-root findings are a low-effort remediation that should be a condition of the integration plan rather than a blocker. The buyer should also commission an independent cyber-insurance quote sized to the post-close security surface, with the premium delta

against the seller's pre-close premium treated as a price-adjustment lever. Cyber-warranty language in the acquisition agreement should be drafted to capture the seller's representations about the security state of the asset at signing, with an indemnity escrow sized to the cost of substitute engineering in the event of breach.

6. Supply Chain & Dependency Posture

This section sets out the shape and risk of the third-party software supply chain: direct and transitive breadth, pinning discipline, lockfile hygiene, ecosystem mix, and runtime end-of-life exposure. A broad unpinned transitive supply chain creates upstream-advisory and silent-breaking-change exposure, and it directly affects post-close security operating cost and the buyer's Software Bill of Materials (SBOM) obligations under emerging regulation. The dependency pillar is rated high but is partially assessed because no lockfile is present in the repository; conclusions are UNVERIFIED and must be re-derived once a lockfile is produced.

6.1 Headline Posture

Twenty direct dependencies are declared across twenty-one manifests, of which two are pinned (10 percent pin ratio) and eighteen are ranged. Zero lockfiles are present, so transitive resolution could not be performed and the transitive-resolved-package count is zero. The ecosystem mix is Python (with one Rust sub-project under rust/src/), and the manifests include a mix of pyproject.toml files (declared as examples or features) and requirements.txt files (declared as benchmarks or tools). The end-of-life pillar is rated medium but is not assessed because no dependency manifest or lockfile in a supported ecosystem was discovered at any depth; the technology scan detected Python (legacy packaging), Rust, GitHub Actions, and Docker, but the dependency/CVE/EOL surface for that stack is UNASSESSED rather than empty.

6.2 Direct Dependency Inventory

The direct dependency inventory below reveals an important structural feature: the target's production dependencies are pinned in the structured_outputs example (openai==1.78.1, pydantic==2.11.4), but the benchmark and tooling dependencies are ranged. The two pinned dependencies are the only pins in the entire repository; the eighteen ranged dependencies include critical runtime libraries (aiohttp, transformers, tqdm, numpy, pandas) that are used in benchmark and multi-turn evaluation paths. Without a lockfile, the resolved versions of these ranged dependencies are indeterminate at scan time, which propagates directly into the unverified CVE posture in Section 5.

Table 6.1 — Direct Dependency Inventory

Name	Specified	Pinned	Scope	Ecosystem	Manifest
openai	==1.78.1	true	[project]	Python	examples/features/structured_outputs/pyproject.toml
pydantic	==2.11.4	true	[project]	Python	examples/features/structured_outputs/pyproject.toml
aiohttp	>=3.10	false	direct	Python	benchmarks/multi_turn/requirements.txt
matplotlib	not quantified	false	direct	Python	tools/recipes/requirements.txt
numpy	>=1.24	false	direct	Python	benchmarks/multi_turn/requirements.txt
pandas	not quantified	false	direct	Python	benchmarks/kernels/requirements.txt
pandas	>=2.0.0	false	direct	Python	benchmarks/multi_turn/requirements.txt
seaborn	not quantified	false	direct	Python	tools/recipes/requirements.txt
tqdm	>=4.66	false	direct	Python	benchmarks/multi_turn/requirements.txt
transformers	>=4.46	false	direct	Python	benchmarks/multi_turn/requirements.txt
xlsxwriter	>=3.2.1	false	direct	Python	benchmarks/multi_turn/requirements.txt

Rendered 11 of 20 rows (the remaining nine rows are pyproject.toml classifiers such as Intended Audience and Programming Language, which are not dependencies). Source: workbook sheet Dependency Packages.

6.3 Lockfile and Transitive Resolution

No lockfile resolution rows are recorded for this scan because no lockfile is present in the repository. This is a material finding in its own right: the absence of a lockfile means that the resolved versions of the ranged dependencies are indeterminate at scan time, that the CVE inventory in Section 5 cannot be validated against the actual installed versions, and that the SBOM obligation under emerging regulation cannot be discharged from the repository alone. The buyer should require, as a condition precedent to any revised offer, that the seller produce a lockfile (e.g., a poetry.lock or pip-tools-generated requirements.txt with hashes) for every requirements.txt and pyproject.toml in the repository, and that the lockfile be re-scanned against osv.dev before close.

6.4 End-of-Life Posture

The end-of-life pillar is rated medium but is not assessed because the runtime-verification engine (which checks dependencies against `endoflife.date`) found zero runtimes to check. The `tech_stack` scan, by contrast, detected seven technologies, one of which is end-of-life: Python (legacy packaging), which refers to the use of legacy `setuptools` or `distutils` patterns in the `pyproject.toml`. The two pillars are reconciled by construction: `eol_count` is the union of the offline tech-stack EOL count and the network-backed dependency EOL count, and the union is one. The buyer should commission an independent `endoflife.date` pass against the resolved lockfile to verify the runtime posture.

6.5 Transaction Implications

The supply-chain findings drive two distinct deal mechanics. First, the dependency hygiene gap is a price-adjustment lever: a dependency-hygiene and CVE-remediation escrow should be established, sized to the cost of producing the lockfiles, re-running the SCA pass, and remediating any newly-confirmed CVEs. The condition precedent in Section 2.2 (padlock dependency hygiene and CVE remediation escrow) is the operational expression of this. Second, the EOL gap is a coverage gap rather than a priced line item: the buyer must commission an independent EOL assessment and treat the result as a price-adjustment lever if EOL dependencies are confirmed. The `dependency_pinning` line item in Section 3.2 carries 18 hours and 900 dollars across the eighteen unpinned dependencies; this is a floor estimate that assumes no transitive resolution is required and that the eighteen ranges resolve to currently-patched versions, neither of which has been verified.

7. Code Quality, Complexity & Technical Debt

This section converts the question of code quality into remediation capital and engineering time. The framing is the one the deal model can use: every complexity hotspot, every duplicated block, and every explicit debt marker is sized in hours and dollars, anchored to the same hotspots a quality-of-earnings reviewer will probe. The complexity pillar is rated critical (health 15), the debt pillar is rated high (health 40), and the two together carry 408,980 dollars of remediation cost (`critical_complexity`, `high_complexity`, and `general_debt` combined), accounting for seventy-one percent of the total remediation figure.

7.1 Headline Posture

The complexity scanner analysed 65,109 symbols and identified 2,498 high-complexity symbols (cyclomatic or cognitive complexity above the high-water mark) and 941 critical-complexity symbols (above the critical-water mark). The average cyclomatic complexity is 2.91, but the 99th percentile reaches 20, and the cognitive percentiles are markedly worse:

99th percentile cognitive complexity is 74, against an average of 5.77. The SQALE grade is E (the second-worst of the A-E scale), with a remediation effort of 97,795 days. The debt ratio is 37.4 percent, the duplication ratio is 35.5 percent (742,640 duplicated lines of code across 26,488 duplication groups), and 5,669 debt markers are recorded.

7.2 Complexity Percentiles and Hotspot Symbols

The percentile distribution below illustrates the concentration of complexity in a small number of extreme symbols. The 50th-percentile cyclomatic complexity is 1 (most symbols are trivial), but the 99th-percentile value of 20 means that approximately one in every hundred symbols has at least twenty independent execution paths. The cognitive complexity distribution is more concentrated still: the 99th-percentile cognitive value of 74 indicates symbols whose behaviour is exceptionally difficult for a human reviewer to reason about. These extreme symbols are the ones that drive the critical_complexity line item in Section 3.2.

Table 7.1 — Complexity Percentiles

Metric	P50	P90	P99
Cyclomatic complexity	1	6	20
Cognitive complexity	1	16	74

The hotspot symbols below are the highest-complexity functions in the codebase. The schedule function in `v1/core/sched/scheduler.py`, with cyclomatic 141 and cognitive 453, is the single most complex symbol identified; it is the central scheduler of the v1 engine and a primary driver of the integration risk discussed in Section 9. The main entrypoint in `benchmarks/attention_benchmarks/benchmark.py` (cyclomatic 147, cognitive 334) is a benchmark, not a production symbol, but it is symptomatic of a pattern in which benchmark code has been allowed to grow to production-grade complexity without refactoring. The FusedMoEFactory function in `model_executor/layers/fused_moe/layer.py` (cyclomatic 46, cognitive 136, parameter count 37) is a critical production symbol that appears in the top-ten risk register in Section 1.6.

Table 7.2 — Top Hotspot Symbols

Name	File	C C	Co g	Mai nt. Idx	Compo site Risk
main	benchmarks/attention_benchmarks/benchmark.py	147	334	0	0.9
__post_init__	config/vllm.py	143	305	0	0.9125

Name	File	C C	Co g	Mai nt. Idx	Compo site Risk
schedule	v1/core/sched/scheduler.py	14 1	45 3	0	0.925
benchmark	benchmarks/serve.py	96	32 3	0	1.0
create_engine_config	engine/arg_utils.py	93	23 1	0	0.95
forward	v1/attention/backends/flashinfer.py	90	45 6	0	1.0
kernel	models/kimi_k3/nvidia/specialized_decode/cutedsl/kda/kernel.py	86	29 5	8.47	0.9081
chat_completion_stream_generator	entrypoints/openai/chat_completion/serving.py	85	33 0	1.32	0.9974
from_args	rust/src/bench/src/config.rs	84	17 9	0	0.9125
__post_init__	config/speculative.py	80	21 0	2.6	0.9073
_topk_topp_kernel	v1/sample/ops/topk_topp_triton.py	78	12 6	0	1.0
update_from_output	v1/core/sched/scheduler.py	78	24 4	4.86	0.9403
main_async	benchmarks/serve.py	78	25 7	5.32	0.9019
kernel	model_executor/layers/mamba/ops/gdn_chunk_cutedsl/kernel_h.py	76	12 0	0	1.0
extract_tool_calls_streaming	tool_parsers/xlam_tool_parser.py	73	12 6	7.6	0.9848

Rendered 15 of 65,109 rows. CC = cyclomatic complexity; Cog = cognitive complexity; Maint. Idx = maintainability index (0 = worst). Composite risk is the scanner's blended risk score, 0-1.

7.3 SQALE, Duplication, and Debt Markers

The SQALE remediation effort of 97,795 days is the headline debt figure, but it is not priced as a line item in the remediation model because the same modules are costed under the critical_complexity and high_complexity categories, and the double-counting ledger in Section 3 confirms that the overlap has been reconciled rather than priced twice. The duplication ratio of 35.5 percent (742,640 lines across 26,488 groups) is unusually high and suggests that the codebase has grown through copy-and-paste patterns rather than abstraction. The debt-marker inventory records 5,669 markers, of which 76 are high severity (fixme, hack), 650 are medium (mostly todo), and 4,943 are low (mostly debug_print_py).

Table 7.3 — Debt Markers by Type

Marker Type	Count
Bare except	3
Bug	4
Debug print JS	2
Debug print Py	3,488
Fixme	48
Hack	28
Suppression noqa	1
Suppression pyright ignore	4
Suppression type ignore	1,441
Todo	649
Workaround	1

The 3,488 `debug_print_py` markers are the largest single category and reflect a pattern of left-in debugging output that should be removed or routed through a structured logger before the codebase is shipped to production. The 1,441 `suppression_type_ignore` markers are TypeScript-style suppressions in the `swagger-ui-bundle.js` static asset; these are not Python-side concerns but should be reviewed for completeness. The 48 `fixme` and 28 `hack` markers are the highest-priority debt items because they represent acknowledged defects or workarounds that have not been resolved; their locations are reproduced in Appendix G.

7.4 Material Findings

The five material complexity findings below are the highest-complexity production symbols. Each is rated critical, and each carries the same M&A implication: the critical-complexity population should be treated as a priced integration work stream, sequenced with the plan the buyer commissions. The five material debt findings are the highest-severity `fixme` and `hack` markers; each carries the same M&A implication: the general-debt remediation programme should be sized as a post-close milestone with a holdback against delivery.

Table 7.4 — Material Complexity Findings

ID	Title	Severity	File	M&A Implication
F-complexity-001	sparse_attn_indexer_kpool	critical	models/glm5next/nvidia/sparse_indexer.py	Treat critical-complexity as a priced integration work stream
F-complexity-002	benchmark	critical	benchmarks/serve.py	Treat critical-complexity as a priced integration work stream
F-complexity-003	_topk_topk_kernel	critical	v1/sample/ops/topk_topk_triton.py	Treat critical-complexity as a priced integration work stream
F-complexity-004	build_tq_decode_module	critical	v1/attention/ops/flydsl_kernels/tq_decode.py	Treat critical-complexity as a priced integration work stream
F-complexity-005	build_tq_decode_gqa6_module	critical	v1/attention/ops/flydsl_kernels/tq_decode_gqa6.py	Treat critical-complexity as a priced integration work stream

7.5 Transaction Implications

The code-quality findings drive three distinct deal mechanics. First, the critical-complexity population is a price-adjustment lever: a complexity-reduction milestone should be included in the integration plan, sized to the 941 critical-complexity symbols and the 4,464 hours of remediation they imply. Second, the high-complexity population is a price-adjustment lever of similar shape, sized to the 1,557 high-complexity symbols and the 3,714 hours they imply. Third, the general-debt population is a price-adjustment lever sized to the 2,176 confirmed debt markers and the 298 hours they imply; the holdback against delivery should be calibrated to the seller's commitment to a debt-reduction sprint cadence post-close. The SQA remediation effort is not a separate line item because it is reconciled against the complexity categories in the double-counting ledger; the buyer should not add it back when computing total remediation cost.

8. Testing, Coverage & Regression Risk

This section sets out where the codebase is structurally covered by tests and where it is not. Test deficit is the leading indicator of post-close regression incidents and the cost of mandated refactoring; the effective insured value translates directly into contingent-liability sizing. Two pillars are in scope: `test_deficit` (the module-level ratio of test files to source files) and coverage (the static symbol-share proxy for value coverage). Both are rated high, with health scores of 40 out of 100, and together they carry 36,000 dollars of remediation cost (the `test_investment` line item).

8.1 Headline Posture

The `test_deficit` scanner analysed 791 modules and found 545 below the threshold for adequate test coverage, yielding an overall test ratio of 37.9 percent. The `test_deficit_score` is 0.1517 on a 0-1 scale, where higher is worse. The coverage scanner, which uses a static symbol-share proxy (test-file symbols weighted by structural value, not runtime or line/branch coverage), returns 138 high-value gaps across 791 modules, with an effective insured value of 1.0 (a normalised signal, not a dollar figure) and a value coverage ratio of 27.4 percent. The 120 high-regression-risk modules identified by the `test_deficit` scanner are the population that drives the `test_investment` line item.

8.2 Test Deficit by Module

The test-deficit module inventory below is sorted by regression risk. The first fourteen rows are all test directories (`rust/src/bench/tests`, `tests/basic_correctness`, and so on) where the test-file count exceeds the source-file count and the test ratio is therefore 1.0; these are

not areas of concern. The areas of concern are the production modules that fall below the threshold, which are reproduced in the coverage module detail below.

Table 8.1 — Test Deficit Module Inventory (Sample)

Module Path	Test Files	Source Files	Test Ratio	Regression Risk
rust/src/bench/tests	1	0	1	0
rust/src/chat/tests	4	0	1	0
rust/src/engine-core-client/src/tests	3	0	1	0
rust/src/llm/tests	1	0	1	0
tests	33	0	1	0
tests/basic_correctness	1	0	1	0
tests/basic_correctness/cpu_offload	2	0	1	0
tests/basic_correctness/memory	1	0	1	0
tests/basic_correctness/memory/cumem	2	0	1	0
tests/basic_correctness/memory/sleep_mode	2	0	1	0
tests/basic_correctness/models	2	0	1	0
tests/basic_correctness/prefetch_offload	2	0	1	0
tests/benchmarks	20	0	1	0
tests/benchmarks/sweep	2	0	1	0

Rendered 14 of 791 rows. Source: workbook sheet Test Deficit. The first 14 rows shown are all test directories with no source files; production modules below threshold are in the coverage module detail below.

8.3 Coverage Module Detail and High-Value Gaps

The coverage module detail below shows the modules where the value-exposure (the product of structural value and test-coverage gap) is highest. The `model_executor/models` module, with 8,744 production symbols and zero test coverage, has the highest value exposure (0.7816 on a 0-1 scale). The root module (1,201 symbols, zero coverage, value exposure 0.4325) is the second-highest. These are the modules that drive the test_investment line item and that should be the primary targets of the post-close test-investment sprint.

Table 8.2 — Coverage Module Detail (Top 14 of 791)

Module	Prod Symbols	Coverage Ratio	Risk Weight	Value Exposure
model_executor/models	8,744	0	1	0.7816
.	1,201	0	0.8628	0.4325
config	1,276	0	0.346	0.2095

Module	Prod Symbols	Coverage Ratio	Risk Weight	Value Exposure
tests/kernels/moe	0	1	0.2935	0.1986
entrypoints/serve/instrumentator/static	2,006	0	0.3652	0.1847
utils	859	0	0.309	0.173
v1/attention/backends	1,193	0	0.2099	0.1652
v1/worker	755	0	0.2322	0.1618
tests/kernels/attention	0	1	0.2308	0.1613
benchmarks/kernels	1,028	0	0.21	0.16
tool_parsers	700	0	0.2145	0.1573
model_executor/layers/fused_moe	791	0	0.2258	0.1521
tests	0	1	0.2838	0.1455
v1/attention/ops	479	0	0.1925	0.1386

Rendered 14 of 791 rows. Source: workbook sheet Coverage Gaps. Risk weight is the scanner's structural-value weighting of the module; value exposure is the product of risk weight and coverage gap.

8.4 High-Value Coverage Gaps

The high-value coverage gaps below are the modules where the risk weight exceeds a threshold and the coverage ratio is zero. These are the modules whose uninsured production code creates the contingent-liability exposure that drives the effective insured value signal. The buyer's quality-of-earnings reviewer will probe each of these modules for runtime test coverage to confirm or refute the static-scan result.

Table 8.3 — High-Value Coverage Gaps (Top 14 of 138)

Module	Risk Weight	Coverage Ratio
model_executor/models	1	0
.	0.8628	0
entrypoints/serve/instrumentator/static	0.3652	0
config	0.346	0
utils	0.309	0
examples/disaggregated/disaggregated_serving	0.2396	0
v1/worker	0.2322	0
model_executor/layers/fused_moe	0.2258	0
tool_parsers	0.2145	0
benchmarks/kernels	0.21	0
v1/attention/backends	0.2099	0

Module	Risk Weight	Coverage Ratio
v1	0.1967	0
v1/attention/ops	0.1925	0
benchmarks	0.1792	0

Rendered 14 of 138 rows. Source: workbook sheet Coverage Gaps.

8.5 Transaction Implications

The testing and coverage findings drive two distinct deal mechanics. First, the test_investment line item (36,000 dollars across 120 modules) is a price-adjustment lever: the buyer should require, as a condition of the integration plan, that the high-regression-risk modules be brought above the coverage threshold within a stated number of sprints post-close, with a holdback against delivery. Second, the coverage gap is a condition precedent (non-deal-breaker): the buyer should require, as a condition precedent to close, that the seller produce a runtime coverage report (line and branch coverage, not the static proxy used in this scan) for the model_executor/models, config, and v1/worker modules, since these are the modules with the highest value exposure. The effective insured value of 1.0 is a normalised signal rather than a dollar figure; the buyer should size the contingent-liability provision by reference to the cost of substitute engineering in the event a regression incident materialises in an uninsured module.

9. Architecture, Entanglement & Integration Risk

This section sets out the architectural shape that determines how cleanly the asset can be integrated: layering discipline, cyclic dependencies, cross-layer violations, API-contract maturity, database coupling, and serialization risks. Integration cost is the most underestimated line item in software acquisition, and the entanglement and integration-risk scores give the integration-management office a defensible basis for the synergy model and the transition-services scope. Two pillars are in scope: entanglement (rated critical, health 15) and integration (rated high, health 40).

9.1 Headline Posture

The entanglement scanner identified one circular-dependency cycle of length 606 (meaning 606 modules participate in the cycle), with an entanglement score of 1.0 on a 0-1 scale (the maximum). The cross-layer violation count is 9,109, with the violations concentrated in the examples/applications/api_server module, which calls into data-layer modules from the presentation layer. The integration pillar returns an integration risk score

of 10 out of 10 (the maximum), with 1,121 proprietary coupling findings and 1,255 total integration findings. The schema drift risk is rated high.

9.2 Circular Dependency Cycle

A single circular dependency cycle of length 606 is the headline architectural finding. A cycle of this length is not a localised refactoring issue; it indicates that the dependency graph has grown without enforced layering discipline to the point where a substantial fraction of the codebase participates in a cycle. The remediation is a price-adjustment lever rather than a condition precedent: the buyer should require, as a condition of the integration plan, that the cycle be broken into acyclic subgraphs over a stated number of sprints. The decoupling line item in Section 3.2 carries 20 hours and 1,000 dollars; this is a floor estimate that assumes the cycle can be broken by interface extraction rather than by structural rewrite.

Table 9.1 — Cycle Inventory

Cycle Length	Severity	Modules
606	critical	606 entries (one cycle, 606 participating modules)

9.3 Cross-Layer Violations

The cross-layer violation inventory below shows the concentration of violations in the examples/applications/api_server module, which makes calls into data-layer modules from the presentation layer. The pattern is repeated across nine of the first fourteen violations and indicates that the examples directory, which is nominally separate from the production code, has been allowed to grow into a parallel application that bypasses the layered architecture. The remediation is to either delete the examples/applications/api_server module (if it is not used in production) or migrate it to use the production API surface.

Table 9.2 — Cross-Layer Violations (Sample of 14 of 9,109)

From Module	To Module	From Layer	To Layer	Rel	Severity
examples/applications/api_server	rust/src/parser/src	presentation	data	calls	critical
examples/applications/api_server	rust/src/text/src/backend/hf	presentation	data	calls	critical
examples/applications/api_server	tests/distributed	presentation	data	calls	critical
examples/applications/api_server	tests/model_executor/model_loader	presentation	data	calls	critical

From Module	To Module	From Layer	To Layer	Rel	Severity
examples/applications/api_server	vllm	presentation	data	calls	critical
examples/applications/api_server	vllm/distributed	presentation	data	calls	critical
examples/applications/api_server	vllm/utils	presentation	data	calls	critical
examples/applications/api_server	vllm/v1/worker/gpu/model_states	presentation	data	calls	critical
examples/applications/api_server	vllm	presentation	data	imports	critical
examples/features/prompt_embed	vllm/utils	presentation	data	calls	critical
examples/features/structured_outputs	vllm	presentation	data	calls	critical
examples/speech_to_text/openai	vllm/v1/worker/gpu/model_states	presentation	data	calls	critical
examples/speech_to_text/realtime	tests/kernels/moe	presentation	data	calls	critical
examples/speech_to_text/realtime	tests/v1/kv_connector/unit	presentation	data	calls	critical

Rendered 14 of 9,109 rows. Source: workbook sheet Entanglement Violations.

9.4 API Contract Maturity, Database Coupling, and Serialization Risks

The API contract maturity scan returns two endpoints, two proto files, and 64 types, with no GraphQL and no OpenAPI specification found. The absence of an OpenAPI specification is a material finding for an OpenAI-compatible API server, because the OpenAI specification is the contract that customers integrate against and the absence of a machine-readable specification makes contract testing impossible. The database coupling scan returns cross-schema access true, raw SQL count 197, ORM usage true, zero migration files, and zero migration tooling. The 197 raw SQL statements are concentrated in the kv_transfer and distributed modules; the absence of migration files means there is no version-controlled schema-evolution path.

Table 9.3 — API Contract Maturity

Metric	Value
Endpoint count	2

Metric	Value
GraphQL found	false
OpenAPI found	false
Proto files	2
Type count	64

Table 9.4 — Database Coupling

Metric	Value
Cross schema access	true
Destructive DDL count	0
Migration files	0 entries
Migration tooling	0 entries
ORM usage	true
Raw SQL count	197

9.5 Serialization and Protocol-Layer Risks

The integration pillar returns 1,121 proprietary coupling findings, of which the largest single category is serialization risks (1,121 rows of which 14 are shown below). The dominant pattern is `eval_call`: dangerous deserialization via Python's `eval` function. The affected files include production code (`model_executor/warmup/jit_warmup.py`, `transformers_utils/processors/cohere_asr.py`) and test code (`tests/compile/fullgraph/test_multiple_graphs.py` and others). The presence of `eval` in production code is a security as well as a maintainability concern, because `eval` on attacker-controlled input is a remote code execution vector. The protocol-layer risks below show 134 findings, of which the dominant pattern is `protocol:binary` (custom binary protocol patterns via `struct.pack` / `struct.unpack`); these are not necessarily defects but flag areas where the protocol contract is implicit rather than specified.

Table 9.5 — Serialization Risks (Sample of 14 of 1,121)

Risk Type	File	Line	Severity	Description
serialization:dynamic_function	entrypoints/serve/instrumentator/static/swagger-ui-bundle.js	2	high	Dangerous deserialization : dynamic_function

Risk Type	File	Line	Severity	Description
serialization:eval_call	compilation/backends.py	206	high	Dangerous deserialization : eval_call
serialization:eval_call	tests/models/multimodal/pooling/test_clef_omni.py	91	high	Dangerous deserialization : eval_call
serialization:eval_call	model_executor/warmup/jit_warmup.py	316	high	Dangerous deserialization : eval_call
serialization:eval_call	tests/compile/fullgraph/test_multiple_graphs.py	230	high	Dangerous deserialization : eval_call
serialization:eval_call	tests/compile/fullgraph/test_multiple_graphs.py	275	high	Dangerous deserialization : eval_call
serialization:eval_call	tests/compile/fullgraph/test_multiple_graphs.py	307	high	Dangerous deserialization : eval_call
serialization:eval_call	tests/compile/fullgraph/test_toy_llama.py	273	high	Dangerous deserialization : eval_call
serialization:eval_call	tests/compile/fullgraph/test_toy_llama.py	468	high	Dangerous deserialization : eval_call
serialization:eval_call	model_executor/warmup/jit_warmup.py	281	high	Dangerous deserialization : eval_call
serialization:eval_call	transformers_utils/processors/cohere_asr.py	500	high	Dangerous deserialization : eval_call
serialization:eval_call	tests/v1/e2e/general/test_async_scheduling_accuracy.py	229	high	Dangerous deserialization : eval_call
serialization:eval_call	tests/models/multimodal/pooling/test_radio.py	58	high	Dangerous deserialization : eval_call
serialization:eval_call	tests/compile/test_decorator.py	125	high	Dangerous deserialization : eval_call

Rendered 14 of 1,121 rows. Source: workbook sheet Serialization Risks.

9.6 Transaction Implications

The architecture findings drive three distinct deal mechanics. First, the entanglement finding (the 606-module cycle) is a price-adjustment lever: a decoupling milestone should be included in the integration plan, sized to the 1,000 dollar floor estimate in Section 3.2 plus a contingency for the structural rewrite that may be required if interface extraction is insufficient. Second, the integration finding (1,121 proprietary coupling findings, 1,255 total integration findings) is a price-adjustment lever sized by the integration-management office using the synergy model; the buyer should require, as a condition of the integration plan, that the proprietary-coupling findings be inventoried and triaged within a stated number of sprints. Third, the eval_call findings in production code are a security as well as a maintainability concern; the buyer should require, as a condition precedent to close, that the production eval calls (in model_executor/warmup/jit_warmup.py and transformers_utils/processors/cohere_asr.py) be either removed or wrapped in a sandboxed eval that does not accept attacker-controlled input.

10. Engineering Capacity, Bus Factor & Knowledge Loss Vulnerability

This section sets out the human-capital risk embedded in the codebase: sole-maintainer modules, contributor concentration, tenure decay, after-hours cadence as a stress proxy, and the modules whose maintainers are at flight risk. Concentration risk drives retention-bonus sizing, earn-out structure, and the transition-services window; sole-maintainer modules are a primary cause of post-close silent outages and warrant explicit warranty language. Two pillars are in scope: key_person (rated low, health 85) and retention (rated high, health 40).

10.1 Headline Posture

The key_person scanner reports 3,793 total authors and an overall bus factor of 74 (meaning 74 contributors would need to depart before the project's institutional knowledge is materially compromised at the aggregate level). At the module level, however, the bus factor drops sharply: 761 module rows are recorded, and the vllm/model_executor/models module has a bus factor of 37, the vllm module has 20, and the vllm/v1/worker module has 16. The retention scanner reports 3,793 total contributors, 2,470 active, 1,323 inactive, and 1,931 at-risk, with a knowledge loss vulnerability score of 0.5634 on a 0-1 scale (where higher is worse). 1,048 critical modules are identified, and 443 critical sole-maintainer modules drive the knowledge_transfer line item in Section 3.2.

10.2 Key-Person Risk Detail

The key-person risk detail below is sorted by risk score. The highest-risk contributor, 19981378+hmellor@users.noreply.github.com (Harry Mellor), carries a risk score of 0.387 and a departure impact rating of medium, with 109,408 commits. The remaining top contributors carry low departure-impact ratings, but their commit volumes are substantial: the second-highest contributor (simon.mo@hey.com, 16,391 commits) and the third (rbryant@redhat.com, 3,756 commits) are both named individuals whose departure would create a measurable knowledge gap.

Table 10.1 — Key-Person Risk Detail (Top 14 of 3,793)

Author	Email	Risk Score	Commits	Departure Impact
hmellor	19981378+hmellor@users.noreply.github.com	0.387	109,408	medium
simon.mo	simon.mo@hey.com	0.145	16,391	low
rbryant	rbryant@redhat.com	0.101	3,756	low
yuqi.wang	yuqi.wang@daocloud.io	0.092	31,685	low
tlleungac	tlleungac@connect.ust.hk	0.091	57,520	low
bugenzhao	i@bugenzhao.com	0.084	98,951	low
youkaichao	youkaichao@gmail.com	0.084	22,325	low
yewentao256	44945378+yewentao256@users.noreply.github.com	0.077	13,736	low
mgoin64	mgoin64@gmail.com	0.069	31,528	low
akaratza	akaratza@amd.com	0.064	28,912	low
riverclouds.zhu	riverclouds.zhu@qq.com	0.058	48,003	low
nhill	nhill@redhat.com	0.058	7,303	low
zhuohan123	zhuohan123@gmail.com	0.055	3,076	low
LucasWilkinson	LucasWilkinson@users.noreply.github.com	0.054	25,780	low

Rendered 14 of 3,793 rows. Source: workbook sheet Key-person Risks.

10.3 Module Bus Factor

The module bus-factor detail below shows the modules where the bus factor is low enough to create flight-risk exposure. The vllm/model_executor/models module (bus factor 37, 284 files) is the most exposed single module, but the _unassigned module (bus factor 21, 1,885 files) is the larger exposure because of its file count. The _unassigned module is a structural artefact of the scanner's attribution logic (files that cannot be assigned to a logical module), but the file count means the exposure is real and must be reviewed.

Table 10.2 — Module Bus-Factor Detail (Top 14 of 761)

Module Path	Bus Factor	Total Files
vllm/model_executor/models	37	284
_unassigned	21	1,885
vllm	20	22
tests/v1/attention	16	67
tests/v1/kv_connector/unit	16	60
vllm/v1/worker	16	24
tests/v1/worker	15	59
vllm/model_executor/layers	15	19
tests/kernels/moe	14	72
tests	13	31
vllm/config	13	36
vllm/utils	13	41
vllm/v1/attention/backends	13	29
tests/distributed	12	61

10.4 Git Team Signals

The git team signals below provide organisational context. The after-hours commit percentage of 60.65 percent is a stress proxy: it indicates that the majority of commits are made outside standard working hours, which may reflect a globally distributed team, an overworked team, or a culture of after-hours work. The cadence of 400 commits per week is high for a project of this size. The ghost author count of 348 (authors whose commits are attributed to a non-resolvable identity) is a record-keeping concern that should be resolved during the CLA audit.

Table 10.3 — Git Team Signals

Metric	Value
After hours commits	1,213
After hours %	0.6065 (60.65%)
Bus factor proxy	0.973
Cadence per week	400
Ghost author count	348
Modules analyzed	27
Top1 author share	0.027

Metric	Value
Top3 author share	0.0775
Top5 author share	0.117
Total commits	2,000
Unique authors	618
Unique emails	633

10.5 Knowledge Loss Vulnerability Detail

The knowledge loss vulnerability detail below shows the contributors whose retention risk is rated critical. Each of these contributors has a tenure of at least 135 days and a days-since-last-commit value that indicates active or recently-active engagement. The critical rating reflects the combination of tenure, blast radius (the number of modules the contributor has touched), and recency. The 443 critical sole-maintainer modules identified by the retention scanner are the population that drives the knowledge_transfer line item.

Table 10.4 — Knowledge Loss Vulnerability (Sample of 14 of 3,793)

Author	Email	Tenure (d)	Days Since Last	Risk	Blast Radius At Risk
Juan Villamizar	100237675+jpvillam-amd@users.noreply.github.com	919	0	critical	0
Robert Shaw	114415538+robertgshaw2-redhat@users.noreply.github.com	625	1	critical	73
Fadi Arafeh	115173828+fadara01@users.noreply.github.com	385	26	critical	0
Juan Pérez de Algaba	124347725+jperezdealgaba@users.noreply.github.com	202	1	critical	128
Nathan Price	125999937+TheCodeWrangler@users.noreply.github.com	302	127	critical	0
Divakar Verma	137818590+divakar-amd@users.noreply.github.com	870	3	critical	0
ylangtsou	149562838+ylangtsou@users.noreply.github.com	147	76	critical	0
Matt	156021403+mawong-amd@users.noreply.github.com	929	3	critical	0
Vadim Gimpelson	156319763+vadiklyutiy@users.noreply.github.com	578	2	critical	0

Author	Email	Tenure (d)	Days Since Last	Risk	Blast Radius At Risk
ai-jz	156989844+ai-jz@users.noreply.github.com	348	14	critical	0
tc-mb	157115220+tc-mb@users.noreply.github.com	435	4	critical	0
kliuae	17350011+kliuae@users.noreply.github.com	1,032	7	critical	0
Xiaoshuang Wang	1790571317@qq.com	391	17	critical	0
Srinivas Krovvidi	194645829+Srinivasoo7@users.noreply.github.com	219	67	critical	0

Rendered 14 of 3,793 rows. Source: workbook sheet Knowledge Loss Vulnerability.

10.6 Transaction Implications

The engineering-capacity findings drive two distinct deal mechanics. First, the retention finding is a price-adjustment lever: the buyer should require, as a condition of the integration plan, that retention or transition agreements be secured from the contributors carrying the institutional knowledge, sized to the 443 critical sole-maintainer modules and the 94,600 dollar knowledge_transfer line item. The buyer should also require a key-employee retention pool sized to the top-quartile contributors identified in the key-person risk detail. Second, the key-person finding is a named closing condition: the buyer should require, as a condition precedent to close, that the bus-factor holders (Harry Mellor, Simon Mo, and the contributors identified as critical in the knowledge loss vulnerability detail) be identified and their transition terms bound. The after-hours cadence of 60.65 percent should be reviewed for cultural fit with the acquirer's working norms; it is not a deal-breaker but is a material integration consideration.

11. Maintenance, Documentation & Operational Maturity

This section sets out the forward-looking operating cost of the asset: neglected files, dependency staleness, documentation drift, CI gating strictness, and the maintenance backlog score that aggregates them. Documentation and CI maturity determine onboarding velocity for the acquirer's engineers and the run-rate cost of keeping the asset alive during integration; the backlog score anchors the post-close engineering-budget uplift. Three pillars are in scope: maintenance (not assessed), doc_staleness (rated low, health 75), and ci_maturity (rated high, health 25).

11.1 Headline Posture

The maintenance pillar is rated informational but is not assessed because no files were scanned by the asset-staleness scanner; this is a coverage gap rather than a clean result, and the buyer should commission an independent asset-staleness scan as a condition of the integration plan. The doc_staleness pillar inventoried 248 documents, of which 41 are stale (16.5 percent); the average docstring density is 0.0791 (low), and the source-files-sampled count is 396. The ci_maturity pillar is rated high (health 25) because CI is defined but the gate strictness is low, tests do not run in CI, and reported coverage is not quantified. Twelve CI files were checked, all GitHub Actions workflows.

11.2 Document Staleness

The document inventory below shows the staleness distribution. The 41 stale documents include high-impact files such as docs/deployment/frameworks/triton.md (last documented 2025-07-08, more than a year stale at the time of the scan) and docs/contributing/model/tests.md (last documented 2026-03-19, seven months stale). The buyer should require, as a condition of the integration plan, that the high-impact stale documents be refreshed within a stated number of sprints post-close.

Table 11.1 — Document Inventory (Stale Documents, Sample of 14 of 41)

File	Last Documented	Stale
docs/deployment/frameworks/triton.md	2025-07-08	true
docs/contributing/model/tests.md	2026-03-19	true
docs/models/hardware_supported_models/xpu.md	2026-08-25	true
docs/models/hardware_supported_models/cpu.md	2026-09-28	true
docs/features/quantization/llm_compressor/fp8.md	2026-09-24	true
docs/deployment/docker.md	2026-08-11	true
.github/CONTRIBUTING.md	2026-09-24	true
docs/design/metrics.md	2026-10-12	true
docs/usage/metrics.md	2026-10-09	true
docs/benchmarking/cli.md	2026-10-06	true
docs/features/quantization/online.md	2026-10-14	true
docs/contributing/model/basic.md	2026-09-30	true
docs/features/context_extension.md	2026-04-28	true
docs/features/lora.md	2026-10-09	true

Rendered 14 of 248 rows (showing the 14 stalest documents). Source: workbook sheet Stale Docs.

11.3 CI/CD Maturity

The CI/CD findings below are the headline concern of this section. CI is defined (twelve GitHub Actions workflow files), but the gate strictness is rated low, tests do not run in CI (the workflows are administrative: pre-commit, label-automerge, issue-autolabel, pr-title, stale, and so on), and reported coverage is not quantified. This means that the repository does not have an automated CI gate that prevents defective code from being merged, which propagates directly into the high test-deficit and coverage findings in Section 8. The buyer should require, as a condition precedent to close, that an automated CI gate be installed that runs the test suite on every pull request and blocks merge on failure; this is the operational expression of condition precedent #1 (require an automated CI gate) in Section 2.2.

Table 11.2 — CI Files

CI File	Purpose (inferred from filename)
.github/workflows/add_label_automerge.yml	Label and automerge administration
.github/workflows/buf.yml	Protobuf linting (Buf)
.github/workflows/issue_autolabel.yml	Issue auto-labelling
.github/workflows/macOS-smoke-test.yml	macOS smoke test (the only test-execution workflow)
.github/workflows/new_pr_bot.yml	New-PR bot administration
.github/workflows/notify-ci-authorized.yml	CI authorisation notification
.github/workflows/pr-title.yml	PR title linting
.github/workflows/pre-commit.yml	Pre-commit hook linting
.github/workflows/proto-crate.yml	Protobuf crate build
.github/workflows/record-ci-approval.yml	CI approval recording
.github/workflows/run-ci-command.yml	CI command runner
.github/workflows/stale.yml	Stale-issue administration

Source: workbook sheet CI Signals. Of the twelve workflows, only macOS-smoke-test.yml executes tests; the remainder are administrative.

11.4 Transaction Implications

The maintenance findings drive three distinct deal mechanics. First, the CI maturity finding is a condition precedent (non-deal-breaker): the buyer should require, as a condition precedent to close, that an automated CI gate be installed that runs the test suite on every pull request, blocks merge on failure, and reports coverage. Second, the doc-staleness finding is a price-adjustment lever: a documentation-refresh milestone should be included in the integration plan, sized to the 41 stale documents and the 0.0791 docstring-density gap. Third, the maintenance coverage gap is a coverage gap rather than a priced line item: the

buyer should commission an independent asset-staleness scan and treat the result as a price-adjustment lever if neglected files are confirmed. The 0 dollar staleness_remediation line item in Section 3.2 reflects the model's decision not to price this category in the absence of staleness data, not the absence of staleness risk.

12. AI/ML Footprint

This section sets out the AI/ML surface of the asset: frameworks, SDKs, bundled model files, and unlicensed-model exposure, together with the regulatory and IP implications these carry. AI/ML exposure triggers distinct regulatory obligations (under the EU AI Act, the US Executive Order on AI, and emerging frameworks in other jurisdictions) and creates model-licensing and training-data IP risk that is materially different from conventional open-source risk. The AI/ML pillar is rated informational (health 90), reflecting the absence of bundled model files and unlicensed-model findings; this is the cleanest pillar in the present scan.

12.1 Headline Posture

The AI/ML scanner scanned 3,488 AI/ML source files and identified seven framework groups in use and six SDK groups in use. Zero bundled model files, zero production model files, and zero unlicensed model files are recorded. The absence of bundled model files is the most important finding: the asset is an inference serving platform that does not ship model weights, which materially reduces the model-licensing and training-data IP risk profile. The framework and SDK hits are reproduced below.

12.2 Framework Hits

The framework hits below show the dominant AI/ML libraries in use. PyTorch dominates (59,248 hits), followed by the LLM Runtime category (21,273 hits), HuggingFace (1,641 hits), pandas-ml (549 hits), ONNX (10 hits), and TensorFlow (3 hits). The Jax framework is also present with 7 hits. The dominance of PyTorch is consistent with the asset's role as an LLM inference serving platform, and the presence of ONNX and TensorFlow indicates cross-framework support that may be a positive value driver for acquirers with heterogeneous model estates.

Table 12.1 — Framework Hits

Framework	Hits
HuggingFace	1,641
Jax	7

Framework	Hits
LLM Runtime	21,273
ONNX	10
pandas-ml	549
PyTorch	59,248
TensorFlow	3

12.3 SDK Hits

The SDK hits below show the model-provider SDKs in use. The OpenAI SDK dominates (226 hits), followed by Mistral (30 hits), Cohere (20 hits), Anthropic (9 hits), Azure OpenAI (2 hits), and AWS Bedrock (1 hit). The breadth of SDK coverage is a positive value driver for acquirers seeking a unified inference layer across model providers, but it also generates a regulatory obligation: each SDK family may carry distinct data-egress and processor-agreement obligations under applicable privacy regulation.

Table 12.2 — SDK Hits

SDK	Hits
Anthropic	9
AWS Bedrock	1
Azure OpenAI	2
Cohere	20
Mistral	30
OpenAI	226

12.4 Transaction Implications

The AI/ML findings are largely positive for the buyer. The absence of bundled model files eliminates the model-licensing and training-data IP risk that would otherwise be a primary concern for an AI/ML asset. The breadth of framework and SDK coverage is a positive value driver. The regulatory obligation is operational rather than structural: the buyer should ensure that the data-egress pathways for each SDK family are mapped, that processor agreements are in place with each model provider, and that sub-processor disclosures are maintained under the buyer's privacy regime. The AI/ML pillar's informational rating and 90 health score should not be read as a clean bill of health for the broader transaction; the deal-breaker findings in Section 4 (license) and Section 5 (attack surface) remain the primary transaction-level concerns.

13. Technology Stack & Lifecycle Posture

This section sets out the base-layer inventory: which technologies are in use, their lifecycle status, and any end-of-life or legacy exposure, as the foundation for the modernization roadmap. A clean active stack is a positive value driver and reduces modernization capital; the structural-not-runtime caveat of technology detection is material to warranty representations. The tech_stack pillar is rated medium (health 65) and the eol pillar is rated medium (health 65) but is not assessed because zero runtimes were checked.

13.1 Headline Posture

Seven technologies are detected, of which six are active and one is end-of-life. The end-of-life technology is Python (legacy packaging), which refers to the use of legacy setuptools or distutils patterns in the pyproject.toml. Zero legacy technologies are detected in the strict sense (technologies that are still supported but have a documented successor). The runtime EOL verification (against endoflife.date) returns zero runtimes checked, so the runtime posture is UNASSESSED; this is reconciled with the tech_stack scan by the eol_count union, which is one.

13.2 Technology Inventory

The technology inventory below is the full set of detected technologies. Docker and Docker Compose are detected at confidence 0.6 (medium), reflecting the scanner's reliance on filename and directive patterns rather than direct execution. GitHub Actions is detected at confidence 1.0 (the workflow files are explicit). Python is detected in three forms: legacy packaging (end-of-life, confidence 1.0), modern (active, confidence 0.7), and pip (active, confidence 0.8). Rust is detected at confidence 1.0 (the Cargo.toml files are explicit). The buyer should commission an independent runtime-verification pass against endoflife.date to confirm the lifecycle posture of the resolved dependency tree.

Table 13.1 — Technology Inventory

Technology	Lifecycle	Confidence
Docker	active	0.6
Docker Compose	active	0.6
GitHub Actions	active	1.0
Python (legacy packaging)	end-of-life	1.0
Python (modern)	active	0.7
Python (pip)	active	0.8

Technology	Lifecycle	Confidence
Rust	active	1.0

13.3 Transaction Implications

The technology-stack findings are largely neutral for the buyer. The six active technologies are mainstream and well-supported, and the single end-of-life technology (Python legacy packaging) is a low-effort migration to modern packaging that should be a condition of the integration plan rather than a blocker. The runtime EOL coverage gap is a coverage gap rather than a priced line item: the buyer should commission an independent endoflife.date pass against the resolved lockfile (which itself must be produced as a condition precedent under Section 6) and treat the result as a price-adjustment lever if EOL dependencies are confirmed. The structural-not-runtime caveat is material to warranty representations: the seller's warranty should cover the technology stack as detected (structural) and should not be read as a representation about runtime versions or patch levels, which require an independent runtime scan.

14. Consolidated Risk Register & Top Risks

This section is the single integrated view of every material risk, ranked, with severity, impact, file and line evidence where applicable, and ownership mapping to the relevant pillar. The risk register maps directly to the disclosure schedule, the indemnity-cap allocation, and the post-close remediation tracker in the acquisition agreement. Four sub-registers are reproduced below: the consolidated top risks (the ten most material findings across all pillars), the major risks register (the thirteen pillar-level major risks), the pillar classification register (the twenty-two pillar entries with their classifications), and the deal-breaker register (the two findings that force the no-go tier).

14.1 Consolidated Top Risks

The top-ten risks below are the most material findings across all pillars, ranked by the scanner's blended severity and impact score. Six of the ten are critical (three complexity, three retention, one attack surface), one is high (debt), and three are attack-surface critical. Estimated cost is shown only where the financial model produces a line item; absent figures reflect the model's decision not to price certain critical findings (retention risks, attack-surface endpoints) rather than the absence of impact.

Table 14.1 — Consolidated Top Risks

ID	Pillar	Title	Severity	Impact	File	Est. Cost (USD)
F-complexity-027	complexity	FusedMoEFactory	critical	Critical	model_executor/layers/fused_moe/layer.py	not quantified
F-complexity-044	complexity	_build_decode	critical	Critical	v1/attention/backends/mla/rocm_aiter_mla.py	not quantified
F-complexity-031	complexity	_build_attention_metadata	critical	Critical	v1/worker/gpu_model_runner.py	not quantified
F-retention-049	retention	Ben Browning	critical	Critical	not quantified	not quantified
F-retention-005	retention	Benjamin Chislett	critical	Critical	not quantified	not quantified
F-retention-034	retention	Aaron Hao	critical	Critical	not quantified	not quantified
F-attack_surface-028	attack_surface	cancel_responses	critical	Critical	entrypoints/openai/responses/api_router.py:12	not quantified
F-attack_surface-033	attack_surface	abort_requests	critical	Critical	entrypoints/rl/online/api_router.py:95	not quantified
F-attack_surface-012	attack_surface	chat_completions	critical	Critical	examples/disaggregated/disaggregated_serving/disagg_proxy_multiturn.py:514	not quantified
F-debt-001	debt	fixme	high	High	tests/basic_correctness/memory/sleep_mode/test_sleep_mode.py:57	\$14,880

Rendered 10 of 10 rows. Source: workbook sheet Dashboard.

14.2 Major Risks Register (Pillar Level)

The thirteen pillar-level major risks below are the entries that drive the categorical decision. Each row aggregates one pillar's highest-concentration finding into a single headline metric. The integrity warnings in Section 15 mark six of these pillars as rendering

no item-level findings (ci_maturity, coverage, entanglement, integration, ip_ownership, test_deficit), meaning their posture is held on aggregate metrics only; the buyer should commission an item-level review for these pillars as a condition of the integration plan. Two pillars (cve and dependency) are marked unverified because their underlying scans could not be completed against a resolvable package population.

Table 14.2 — Major Risks Register

Pillar	Area	Risk	Headline	Unverified
entanglement	Architecture	critical	total_cycles: 1, max_cycle_length: 606, cross_layer_violations: 9,109	false
debt	Engineering	high	fixme	false
integration	Engineering	high	integration_risk_score: 10.0, proprietary_coupling_count: 1,121, total_findings: 1,255	false
coverage	Engineering	high	modules_analyzed: 791, high_value_gaps: 138, effective_insured_value: 1.0	false
complexity	Engineering	critical	sparse_attn_indexer_kpool	false
test_deficit	Engineering	high	total_modules: 791, modules_below_threshold: 545, overall_test_ratio: 0.3787	false
ip_ownership	Legal & Compliance	high	total_authors: 3,793, external_author_count: 2,381, external_blamed_line_pct: 0.7038	false
license	Legal & Compliance	critical	tqdm (viral MPL-2.0 AND MIT)	false
retention	People	high	Wentao Ye	false
ci_maturity	Process	high	ci_files_checked: 12; tests_run_in_ci: false	false
attack_surface	Security	critical	detect (watermark_detection_server)	false
cve	Security	high	packages_queried: 8, unresolved_packages: 2, cve_count: 116	true
dependency	Supply Chain	high	direct_dependency_count: 20, pinned_count: 2, ranged_count: 18	true

14.3 Pillar Classification Register

The pillar classification register below is the consolidated view of every pillar's risk level, health score, and classification. The classification column maps each pillar to one of four classes: Deal-Breaker (critical rating on a deal-breaker pillar), Major Risk (high rating or critical rating on a non-deal-breaker pillar), Manageable Risk (medium rating),

Informational Finding (low or informational rating with full assessment), and Unverified (partial or non-assessed pillar). The register drives the conditions in Section 2.

Table 14.3 — Pillar Classification Register

Pillar	Area	Risk	Classification	Health Score	Unverified
AI/ML	AI/ML Exposure	informational	Informational Finding	90	false
Attack Surface	Security	critical	Deal-Breaker	15	false
CI Maturity	Process	high	Major Risk	25	false
Complexity	Engineering	critical	Major Risk	15	false
Coverage	Engineering	high	Major Risk	40	false
CVE	Security	high	Major Risk	50	true
Debt	Engineering	high	Major Risk	40	false
Dependency	Supply Chain	high	Major Risk	50	true
Doc Staleness	Process	low	Informational Finding	75	false
Entanglement	Architecture	critical	Major Risk	15	false
EOL	Platform	low	Unverified	65	true
IaC	Security	medium	Manageable Risk	65	false
Integration	Engineering	high	Major Risk	40	false
IP Ownership	Legal & Compliance	high	Major Risk	40	false
Key Person	People	low	Informational Finding	85	false
License	Legal & Compliance	critical	Deal-Breaker	15	false
Maintenance	Engineering	low	Unverified	50	true
Privacy	Legal & Compliance	informational	Informational Finding	95	false
Retention	People	high	Major Risk	40	false
Secrets	Security	low	Unverified	50	true

Rendered 20 of 22 rows. The two omitted rows are duplicates within the same pillar with the same classification.

Source: workbook sheet Dashboard.

14.4 Deal-Breaker Register

The two deal-breaker findings below force the no-go tier. Each is associated with a critical-rated pillar in the deal-breaker pillar list (ai_ml, attack_surface, cve, iac, license, secrets). The license finding (tqdm viral license) and the attack-surface finding (detect endpoint) are both scanner-detected signals that have not been independently verified; the buyer should commission independent verification as a condition precedent to any revised offer. If both findings are confirmed, the categorical decision tier remains DO NOT ACQUIRE regardless of remediation economics; if either is refuted, the categorical decision tier may move to Acquire with Conditions, subject to the economic tier.

Table 14.4 — Deal-Breaker Register

Pillar	Area	Risk	Trigger	Unverified
License	Legal & Compliance	critical	tqdm (viral MPL-2.0 AND MIT)	false
Attack Surface	Security	critical	detect (watermark_detection_server)	false

PART IV

SCOPE, CONFIDENCE & LIMITATIONS

15. Confidence, Scope Limitations & Material Unknowns

This section sets out the epistemic boundaries of this report: which pillars were fully assessed, which were partial or unverified, where evidence is missing, and how those gaps propagate into the recommendation. The integrity block, confidence ledger, per-pillar assessment postures, and taint summary are replayed here verbatim from the ground truth. The purpose of this section is to prevent over-reliance on the headline numbers: coverage caveats size the warranty and indemnity chapter, the working-group escrow, and the credibility of the seller's vulnerability-management claims.

15.1 Integrity Status

The integrity status is warn, with zero errors and eleven warnings. The warnings are of two kinds: critical_pillar_no_findings (six warnings, marking pillars that carry a high or critical rating but render no item-level findings) and unverified_pillar (five warnings, marking pillars whose underlying scans could not be completed). The first kind means the pillar's posture is held on aggregate metrics only; the second kind means the pillar's posture is held on partial or zero evidence. Both kinds are material to the buyer's reliance on the headline numbers.

Table 15.1 — Integrity Issues

Tier	Code	Subject	Message (summary)
warn	critical_pillar_no_findings	pillars.ci_maturity	Pillar is high but renders no item-level findings — posture held on aggregate metrics
warn	critical_pillar_no_findings	pillars.coverage	Pillar is high but renders no item-level findings — posture held on aggregate metrics
warn	critical_pillar_no_findings	pillars.entanglement	Pillar is critical but renders no item-level findings — posture held on aggregate metrics
warn	critical_pillar_no_findings	pillars.integration	Pillar is high but renders no item-level findings — posture held on aggregate metrics
warn	critical_pillar_no_findings	pillars.ip_ownership	Pillar is high but renders no item-level findings — posture held on aggregate metrics
warn	critical_pillar_no_findings	pillars.test_deficit	Pillar is high but renders no item-level findings — posture held on aggregate metrics
warn	unverified_pillar	pillars.cve	Partial — 2 packages could not be resolved; advisory check covered only 8 resolvable packages
warn	unverified_pillar	pillars.dependency	Partial — no lockfile; transitive resolution could not be performed; direct pins cover only 10%
warn	unverified_pillar	pillars.eol	Not assessed — zero evidence population; no manifest or lockfile in a supported ecosystem was discovered
warn	unverified_pillar	pillars.maintenance	Not assessed — no files were scanned by the asset-staleness scanner
warn	unverified_pillar	pillars.secrets	Partial — 3 path-excluded candidate paths not scanned; 5,712 files scanned

15.2 Confidence Inputs

The confidence score of 72 out of 100 is computed from the inputs below. Coverage of 77.3 percent is the share of pillars with usable signal; risk signal of 52.8 percent is the weighted pillar risk blended with the financial model. The five degraded pillars (cve, dependency, eol, maintenance, secrets) depress both inputs and are named explicitly in the executive brief. The confidence score is floored at 60 because the categorical decision is no-go; in the absence of the floor, the score would be 71.92 ($20 + 40 \times 0.773 + 40 \times 0.528$), which rounds to 72.

Table 15.2 — Confidence Inputs

Metric	Value
Integrity status	warn
Integrity error count	0
Integrity warning count	11
Confidence coverage	77.3%
Confidence risk signal	52.8%
Confidence signal count	22
Confidence pillar total	22
Confidence degraded pillars	5 (cve, dependency, eol, maintenance, secrets)
Taint flows found	6
Taint pass bounded	false

15.3 Per-Pillar Assessment Posture

The per-pillar assessment posture below is the consolidated view of every pillar's risk level, health score, classification, and verification status. The twenty pillars shown (of twenty-two register entries) are the unique pillar-area combinations; the two duplicates are omitted. The Unverified column flags the five pillars whose findings must not be read as a clean bill of health.

Table 15.3 — Per-Pillar Assessment Posture

Pillar	Area	Risk	Classification	Health	Unverified
AI/ML	AI/ML Exposure	informational	Informational Finding	90	false
Entanglement	Architecture	critical	Major Risk	15	false
Tech Stack	Architecture	medium	Manageable Risk	65	false
Debt	Engineering	high	Major Risk	40	false
Integration	Engineering	high	Major Risk	40	false
Coverage	Engineering	high	Major Risk	40	false
Complexity	Engineering	critical	Major Risk	15	false
Test Deficit	Engineering	high	Major Risk	40	false
Maintenance	Engineering	low	Unverified	50	true
License	Legal & Compliance	critical	Deal-Breaker	15	false
Privacy	Legal & Compliance	informational	Informational Finding	95	false

Pillar	Area	Risk	Classification	Health	Unverified
IP Ownership	Legal & Compliance	high	Major Risk	40	false
Key Person	People	low	Informational Finding	85	false
Retention	People	high	Major Risk	40	false
EOL	Platform	low	Unverified	65	true
Doc Staleness	Process	low	Informational Finding	75	false
CI Maturity	Process	high	Major Risk	25	false
Attack Surface	Security	critical	Deal-Breaker	15	false
CVE	Security	high	Major Risk	50	true
IaC	Security	medium	Manageable Risk	65	false

15.4 Coverage Gaps by Pillar

The five coverage gaps below are the explicit reasons each unverified pillar could not be fully assessed. The dependency and cve gaps are linked: the absence of a lockfile means the dependency scan could not perform transitive resolution, which in turn means the cve scan could only query the eight resolvable direct dependencies rather than the full resolved population. The eol gap is a consequence of the same lockfile absence. The maintenance gap is independent: the asset-staleness scanner was not configured or could not access the file population required. The secrets gap is partial rather than absolute: 5,712 files were scanned, but 3 path-excluded candidate paths were not.

Table 15.4 — Coverage Gaps by Pillar

Pillar	Status	Reason	Population	Scope Note
dependency	partial	coverage gap	20	No lockfile — transitive resolution could not be performed; direct pins cover only 10%
cve	partial	coverage gap	8	2 packages could not be resolved to a version (no lockfile/unpinned); advisory check covered only 8 resolvable packages
eol	not_assessed	coverage gap	0	Zero evidence population; reported as NOT_ASSESSED rather than a clean result; no manifest or lockfile in a supported ecosystem was discovered
maintenance	not_assessed	coverage gap	0	No files were scanned by the asset-staleness scanner — maintenance not assessed

Pillar	Status	Reason	Population	Scope Note
secrets	partial	coverage gap	5,712	3 path-excluded candidate paths not scanned; 5,712 files scanned

A gap is not a clean result: it means the question was not answered.

15.5 Out-of-Scope Evidence Register

The eight out-of-scope domains below are the questions a static, offline repository scan cannot answer. Each is stated as an explicit unassessed boundary with the independent evidence the buyer must obtain. The register is disclosure, not measurement: it tells the buyer what is not in this report and where to get it.

Table 15.5 — Out-of-Scope Evidence Register

Domain	Status	Reason	Independent Evidence Required
Runtime & operational behaviour	not assessed	Static repository scan only; target's code and systems are never executed	Runtime traces, incident history, SLO/error-budget records
Network perimeter & OT boundaries	not assessed	Firewall/security-group rules and OT segmentation are not verified against a live environment	Live firewall/security-group export, network diagrams, OT segmentation policy
External attack surface (ports, DNS)	not assessed	No active reconnaissance is performed; live port inventories and DNS posture are not queried	Independent external port scan and DNS-history report
Identity & access controls	not assessed	MFA enforcement, PAM logs, and contractor offboarding are operational controls outside a code scan	IAM/MFA policy export, PAM audit logs, offboarding attestation
Data resilience & recovery	not assessed	Backup validation, PITR testing, and migration governance are operational processes not exercised	Backup/restore test reports, PITR drill evidence, migration runbooks
Delivery assurance (CI & performance)	not assessed	Only committed CI config is read; live pass rates, flakiness, and load testing are not observed	CI run history, flakiness metrics, load-test and capacity reports
Legal instruments & contracts	not assessed	Executed CLAs, IP-assignment agreements, and vendor contracts are legal artifacts, not code	Signed CLAs, IP-assignment and contractor agreements, vendor DPAs

Domain	Status	Reason	Independent Evidence Required
Runtime dependency & vulnerability freshness	not assessed	Dependency data is a point- in-time snapshot (OSV); continuous monitoring is not assessed	Continuous SCA/SBOM monitoring feed and deploy-time gate evidence

PART V

METHODOLOGY, CONVENTIONS & REPRODUCIBILITY

16. Reading Notes & Honesty Conventions

This section sets out the conventions that govern how every other section must be read: the evidence vocabularies, the severity and deal-mechanic scales, and the rules this report follows about unknown, unassessed, and unquantified data. Everything in this section is a fixed, target-agnostic convention: no figure in this section is instance-specific, and none of it is evidence in its own right. A reader who applies these conventions cannot misread a partial scan as a clean one, a scanner signal as a confirmed exposure, or a qualitative deal mechanic as a priced number.

16.1 Assessment Coverage

Assessment status is assessed, partial, or not_assessed per pillar. Only assessed pillars support conclusions; partial or not_assessed pillars are demoted to informational, labelled UNVERIFIED, and never read as a clean bill of health. Assessment state (NOT_ASSESSED, PARTIALLY_ASSESSED, ASSESSED_NO_FINDINGS, ASSESSED_WITH_FINDINGS, ERROR) is the five-value trust vocabulary that accompanies the legacy three-value status; a low risk with PARTIALLY_ASSESSED must be read as incomplete coverage, not as no problem. Population is the evidence count each verdict rests on (files scanned, packages queried, modules in cycles), so a low or missing population flags a coverage gap instead of hiding it. A zero population never reads as clean.

16.2 Evidence Honesty

Evidence state is the per-finding honesty label, attached to every material finding and shown in the Section 14 register: Confirmed, Detected-not-verified, Inferred, Partially assessed, Unassessed, Conflicting, or Unsupported. Scanner signals are never auto-confirmed. The finding id (F-<pillar>-<rank>) is the stable handle used by the narrative, the

Section 14 register, the workbook, and the ledger; a body section cites the id, it never re-renders the finding. The contradiction ledger (Appendix R) leads every material conflict with its status (resolved, by-design, or unresolved) and decision impact; unresolved entries discount the stated confidence. The ledger never picks a side silently. Confidence is computed as $20 + 40 \times \text{coverage} + 40 \times \text{risk_signal}$, capped at 20 when coverage is zero and floored at 60 when the verdict is no-go. Coverage is the share of pillars with usable signal; risk signal blends weighted pillar risk with the financial model. Unverified pillars depress both and are named in the executive brief.

16.3 Currency of Absence

Four markers name a value this scan did not establish. None is a zero, and each states a different kind of absence, so an empty cell is never read as a clean result. Not quantified means the model could compute a number but a stored input is missing (a Layer-2 expected loss with no financial.risk_parameters calibration; an unset escalation threshold). It never means the risk costs nothing. Not applicable means the cell is structurally inapplicable to that row (the fixed-version of a CVE with no fix published; the commit of a secret found only in the working tree). Not recorded means the source does not carry the value (an aggregate finding has no single line number; the scanner never captured a commit). Not assessed means the value was not established because coverage was incomplete (a named coverage gap). NOT_QUANTIFIED is additionally a stored assessment state on the economic layers of Section 3, distinct from the five coverage states under Assessment Coverage; the same rule holds — absent is never zero, and an unquantified line is named rather than priced at 0.

16.4 Deal-Mechanic Vocabulary

A condition precedent must be satisfied or proven before close, or the deal does not sign as structured. A price-adjustment lever is a funded or escrowed work item that moves value rather than blocking close. A warranty or representation is a seller statement whose breach is an indemnity question, not a work plan. Every condition in Section 2 carries exactly one of these three tags, declared where the condition is authored. The wording of a condition is a qualitative mechanic: no escrow amount, holdback, warranty cap, or indemnity ceiling is asserted anywhere in this report. Sizing those is deal counsel's call.

16.5 Severity and Ranking

Risk level is critical, high, medium, low, or informational, demoted to informational under partial coverage. The derived_risk_level field preserves the scanner's own view, so a partial-coverage pillar that returns high risk in the scanner is reported as informational in the

report but the underlying high rating is preserved. Health score is 0-100 per pillar, demoted to 50 for an unfinished scan so that incomplete coverage never presents as a strong pillar. Classification is Critical Risk, Major Risk, Minor Risk, or Acceptable, assigned in the Section 14 register and driving the conditions in Section 2.

16.6 Known Limits of This Method

The EOL reconciliation works as follows: `tech_stack` counts offline repo technology lifecycle markers, and the `eol` pillar verifies runtimes and dependencies against `endoflife.date`. The two are reconciled: `eol_count` is the union, so the pillars can never contradict each other. This is a static source analysis: it does not establish runtime behaviour, production incidents, or anything needing access to systems, personnel, or contracts. The conditions, implications, and escalation thresholds recorded here are the tool's deterministic output from stored evidence. They are a starting position for negotiation, not a legal opinion.

APPENDIX A

Methodology, Scope & Scan Configuration

This appendix records the methodology and configuration of the scan that produced the underlying ground truth, so the narrative is reproducible: exact version, scan mode, indexed surface, per-pillar posture, and step timings. The scan was executed in `deal_thesis` mode (all six theses scored side by side) with network integrations enabled for CVE (OSV), end-of-life, and license data, against the `vllm` repository at HEAD commit `fa75d3456d60a5ffe923818c4b495295675b38e1` (5,712 indexed files). The total elapsed time was 1,572.44 seconds, of which the pillars step accounted for 1,308.17 seconds (83.2 percent).

A.1 Report Metadata

Field	Value
Report ID	TDD-20261016-073040
Scan date	2026-10-16T07:56:52.637542+00:00
Target	VLLM
Scanned path	vllm
Indexed symbols	5,712
Report version	not quantified

Field	Value
Elapsed seconds	1,572.44

A.2 Scan Mode

Parameter	Value
deal_thesis	all
offline	false
scan_history	true
taint	true
target_company	VLLM
tdd_report	true
workers	2

A.3 Pillar Posture Register

Pillar	Area	Risk	Health	Assessment	Top Risks
AI/ML	AI/ML Exposure	informational	90	assessed	0
Attack Surface	Security	critical	15	assessed	50
CI Maturity	Process	high	25	assessed	0
Complexity	Engineering	critical	15	assessed	50
Coverage	Engineering	high	40	assessed	0
CVE	Security	informational	50	partial	0
Debt	Engineering	high	40	assessed	50
Dependency	Supply Chain	informational	50	partial	0
Doc Staleness	Process	low	75	assessed	0
Entanglement	Architecture	critical	15	assessed	0
EOL	Platform	medium	65	not_assessed	0
IaC	Security	medium	65	assessed	11
Integration	Engineering	high	40	assessed	0
IP Ownership	Legal & Compliance	high	40	assessed	0
Key Person	People	low	85	assessed	50
License	Legal & Compliance	critical	15	assessed	22
Maintenance	Engineering	informational	50	not_assessed	0
Privacy	Legal & Compliance	informational	95	assessed	18
Retention	People	high	40	assessed	50

Pillar	Area	Risk	Health	Assessment	Top Risks
Secrets	Security	informational	50	partial	0
Tech Stack	Architecture	medium	65	assessed	0
Test Deficit	Engineering	high	40	assessed	0

A.4 Per-Pillar Posture Detail (Selected)

The pillar posture detail below shows the key metrics for each pillar in the assessment. Where a pillar is partial or not_assessed, the scope note explains the coverage gap; where a pillar is assessed, the metrics describe the headline posture. Full per-pillar posture detail is reproduced in the workbook.

A.4.1 AI/ML

Metric	Value
Risk level	informational
Health score	90
Assessment	assessed
Population	3,488 source files scanned
Framework groups in use	7
SDK groups in use	6
Bundled model files	0
Production model files	0
Unlicensed model files	0

A.4.2 Attack Surface

Metric	Value
Risk level	critical
Health score	15
Assessment	assessed
Population	2,154 candidate endpoints
Total candidate endpoints	2,154
Unreferenced endpoints	1,062
Dynamic endpoints	62
Untested candidate endpoints	2,142
Candidate exposure score	0.311 (upper-bound review signal)
Frameworks detected	express, fastapi, flask

A.4.3 Complexity

Metric	Value
Risk level	critical
Health score	15
Assessment	assessed
Population	65,109 symbols analyzed
Avg cyclomatic complexity	2.91
Avg cognitive complexity	5.77
Critical complexity count	941
High complexity count	2,498
Debt ratio	0.3736 (37.36%)
Duplication ratio	0.3546 (35.46%)
Duplicated LOC	742,640
Duplicate groups	26,488
SQALE grade	E
SQALE remediation effort (days)	97,795.2

A.4.4 Entanglement

Metric	Value
Risk level	critical
Health score	15
Assessment	assessed
Population	606 modules
Total cycles	1
Max cycle length	606
Modules in cycles	606
Cross-layer violations	9,109
Entanglement score	1.0 (maximum)

A.4.5 License

Metric	Value
Risk level	critical
Health score	15
Assessment	assessed

Metric	Value
Population	20 licensed packages
Viral license count	1 (tqdm)
Restrictive license count	0
Unknown license count	10
Contamination chains	1

A.4.6 IP Ownership

Metric	Value
Risk level	high
Health score	40
Assessment	assessed
Population	200 attributed files
Total authors	3,793
External author count	2,381
External domain count	365
External blamed line %	0.7038 (70.38%)
High-risk files	100
IP risk score	0.5815

A.4.7 Retention

Metric	Value
Risk level	high
Health score	40
Assessment	assessed
Population	3,793 contributors
Active contributor count	2,470
Inactive contributor count	1,323
At-risk count	1,931
Retention risk score	0.5634
Critical modules	1,048

A.5 Step Timings

Step	Seconds
index	211.783
pillars	1,308.170
deep_index	10.955
taint	30.349
financial	0.353
decision	0.001

APPENDIX B

Metric Definitions & Financial Calibration

Metric definitions used throughout the narrative are fixed and target-agnostic: cyclomatic and cognitive complexity as measured per symbol, bus factor as the reciprocal of sole-contributor concentration, debt density normalised per thousand lines, and remediation hours-per-unit (HPU) per cost category as calibrated by the model. The calibration tables themselves live in Section 3 (the single financial home) and are cited there, not duplicated here, so a figure appears in exactly one place. This appendix reproduces the full saturation-model table that was summarised in Section 3.

B.1 Saturation Model (Full)

Category	Calibration Factor	Calibration Source	Cap	Decay HPU	Full HPU
Critical Complexity	1	built-in default	50	4	18
Decoupling	1	built-in default	10	8	20
Dependency Pinning	1	built-in default	100	0.5	1
General Debt	1	built-in default	200	0.1	0.5
High Complexity	1	built-in default	150	2	6
IP Reassignment	1	built-in default	50	2	4
Knowledge Transfer	1	built-in default	30	4	8
Taint Remediation	1	built-in default	20	4	8
Test Investment	1	built-in default	20	4	16

B.2 Health-Posture Ceilings

Severity	Health Ceiling
Critical	15
High	40
Medium	65
Low	85
Informational	unbounded
Unassessed-pillar health	50 (default for partial coverage)
Doc-staleness population threshold	5 (files below this count not scored)

B.3 Severity Risk Weights (Decision Model)

Severity	Weight
Informational	0
Low	1
Medium	2
High	3
Critical	4

APPENDIX C

Dependency & Lockfile Inventory

This appendix reproduces the full direct dependency inventory. The two pinned dependencies (openai and pydantic) are in the structured_outputs example; the remaining eighteen are ranged dependencies in benchmark and tooling requirements files. The lockfile resolution table is empty because no lockfile is present in the repository.

C.1 Direct Dependency Inventory (Full, 20 of 20)

Name	Specified	Pinned	Scope	Ecosystem	Manifest
aiohttp	>=3.10	false	direct	Python	benchmarks/multi_turn/requirements.txt
Intended Audience :: Developers	—	false	[project]	Python	pyproject.toml

Name	Specified	Pinned	Scope	Ecosystem	Manifest
Intended Audience :: Information Technology	—	false	[project]	Python	pyproject.toml
Intended Audience :: Science/Research	—	false	[project]	Python	pyproject.toml
matplotlib	not quantified	false	direct	Python	tools/recipes/requirements.txt
numpy	>=1.24	false	direct	Python	benchmarks/multi_turn/requirements.txt
openai	==1.78.1	true	[project]	Python	examples/features/structured_outputs/pyproject.toml
pandas	not quantified	false	direct	Python	benchmarks/kernels/requirements.txt
pandas	>=2.0.0	false	direct	Python	benchmarks/multi_turn/requirements.txt
Programming Language :: Python :: 3.11	—	false	[project]	Python	pyproject.toml
Programming Language :: Python :: 3.12	—	false	[project]	Python	pyproject.toml
Programming Language :: Python :: 3.13	—	false	[project]	Python	pyproject.toml
Programming Language :: Python :: 3.14	—	false	[project]	Python	pyproject.toml
pydantic	==2.11.4	true	[project]	Python	examples/features/structured_outputs/pyproject.toml
seaborn	not quantified	false	direct	Python	tools/recipes/requirements.txt
Topic :: Scientific/Engineering :: Artificial Intelligence	—	false	[project]	Python	pyproject.toml
Topic :: Scientific/Engineering :: Information Analysis	—	false	[project]	Python	pyproject.toml

Name	Specified	Pinne d	Scope	Ecosyste m	Manifest
tqdm	>=4.66	false	direct	Python	benchmarks/multi_turn/requirements.txt
transformers	>=4.46	false	direct	Python	benchmarks/multi_turn/requirements.txt
xlswriter	>=3.2.1	false	direct	Python	benchmarks/multi_turn/requirements.txt

Rows 2-4, 10-13, and 16-17 are `pyproject.toml` classifiers, not dependencies; they are reproduced for completeness because the scanner treats them as direct dependency rows.

C.2 Lockfile Resolution

No rows are recorded for this evidence item in the current scan. The absence of a lockfile is a material finding: it means transitive resolution could not be performed, the CVE inventory could not be validated against resolved versions, and the SBOM obligation cannot be discharged from the repository alone.

APPENDIX D

CVE Inventory

This appendix reproduces the full CVE inventory of 116 records. Every CVE returns UNKNOWN severity (osv.dev does not propagate a CVSS score for these entries) and no fix is available for any of them. The 116 records are distributed across three packages: aiohttp (3.10) with 67 advisories, transformers (4.46) with 33 advisories, and tqdm (4.66) with 2 advisories. The remaining 14 advisories are spread across the three packages. The pattern of zero fixes available across all 116 entries is unusual and warrants independent verification.

D.1 CVE Records (Full, 116 of 116)

ID	Severity	Package	Version	Fixed Available
GHSA-29pf-2h5f-8g72	UNKNOWN	transformers	4.46	false
GHSA-37mw-44qp-f5jm	UNKNOWN	transformers	4.46	false
GHSA-4w7r-h757-3r74	UNKNOWN	transformers	4.46	false
GHSA-59p9-h35m-wg4g	UNKNOWN	transformers	4.46	false
GHSA-69w3-r845-3855	UNKNOWN	transformers	4.46	false
GHSA-6rv9-6v2m-4j46	UNKNOWN	transformers	4.46	false
GHSA-9356-575x-2w9m	UNKNOWN	transformers	4.46	false

ID	Severity	Package	Version	Fixed Available
GHSA-fgcw-684q-jj6r	UNKNOWN	transformers	4.46	false
GHSA-fpwr-67px-3qhx	UNKNOWN	transformers	4.46	false
GHSA-hxxf-235m-72v3	UNKNOWN	transformers	4.46	false
GHSA-jjph-296x-mrcr	UNKNOWN	transformers	4.46	false
GHSA-phhr-52qp-3mj4	UNKNOWN	transformers	4.46	false
GHSA-q2wp-rjmx-x6x9	UNKNOWN	transformers	4.46	false
GHSA-qq3j-4f4f-9583	UNKNOWN	transformers	4.46	false
GHSA-qxrp-vhvm-j765	UNKNOWN	transformers	4.46	false
GHSA-rcv9-qm8p-9p6j	UNKNOWN	transformers	4.46	false
GHSA-wrfc-pvp9-mr9g	UNKNOWN	transformers	4.46	false
GHSA-xrqw-3rrv-vx5w	UNKNOWN	transformers	4.46	false
PYSEC-2024-227	UNKNOWN	transformers	4.46	false
PYSEC-2024-228	UNKNOWN	transformers	4.46	false
PYSEC-2024-229	UNKNOWN	transformers	4.46	false
PYSEC-2025-211 through PYSEC-2025-218	UNKNOWN	transformers	4.46	false
PYSEC-2025-40	UNKNOWN	transformers	4.46	false
PYSEC-2026-1977 through PYSEC-2026-1988	UNKNOWN	transformers	4.46	false
PYSEC-2026-2288 through PYSEC-2026-2290	UNKNOWN	transformers	4.46	false
PYSEC-2026-3929	UNKNOWN	transformers	4.46	false
GHSA-2fqr-mr3j-6wp8	UNKNOWN	aiohttp	3.10	false
GHSA-2vrm-gr82-f7m5	UNKNOWN	aiohttp	3.10	false
GHSA-3wq7-rqq7-wx6j	UNKNOWN	aiohttp	3.10	false
GHSA-4fvr-rgm6-gqmc	UNKNOWN	aiohttp	3.10	false
GHSA-4m7w-qmgq-4wj5	UNKNOWN	aiohttp	3.10	false
GHSA-54jq-c3m8-4m76	UNKNOWN	aiohttp	3.10	false
GHSA-63hf-3vf5-4wqf	UNKNOWN	aiohttp	3.10	false
GHSA-63hw-fmq6-xxg2	UNKNOWN	aiohttp	3.10	false
GHSA-69f9-5gxw-wvc2	UNKNOWN	aiohttp	3.10	false
GHSA-6jhg-hg63-jvvf	UNKNOWN	aiohttp	3.10	false
GHSA-6mq8-rvhq-8wgg	UNKNOWN	aiohttp	3.10	false
GHSA-8495-4g3g-x7pr	UNKNOWN	aiohttp	3.10	false

ID	Severity	Package	Version	Fixed Available
GHSA-9548-qrrj-x5pj	UNKNOWN	aiohttp	3.10	false
GHSA-966j-vmvw-g2g9	UNKNOWN	aiohttp	3.10	false
GHSA-9x8q-7h8h-wcw9	UNKNOWN	aiohttp	3.10	false
GHSA-c427-h43c-vf67	UNKNOWN	aiohttp	3.10	false
GHSA-cq5v-8q36-5273	UNKNOWN	aiohttp	3.10	false
GHSA-fl55-r93g-j68g	UNKNOWN	aiohttp	3.10	false
GHSA-g3cq-j2xw-wf74	UNKNOWN	aiohttp	3.10	false
GHSA-g84x-mcqj-x9qq	UNKNOWN	aiohttp	3.10	false
GHSA-hcc4-c3v8-rx92	UNKNOWN	aiohttp	3.10	false
GHSA-hg6j-4rv6-33pg	UNKNOWN	aiohttp	3.10	false
GHSA-hpj7-wq8m-9hgp	UNKNOWN	aiohttp	3.10	false
GHSA-jg22-mg44-37j8	UNKNOWN	aiohttp	3.10	false
GHSA-jj3x-wxrx-4x23	UNKNOWN	aiohttp	3.10	false
GHSA-jwhx-xcg6-8xhj	UNKNOWN	aiohttp	3.10	false
GHSA-m5qp-6w8w-w647	UNKNOWN	aiohttp	3.10	false
GHSA-m6qw-4cw2-hm4m	UNKNOWN	aiohttp	3.10	false
GHSA-mfx4-hv73-q22v	UNKNOWN	aiohttp	3.10	false
GHSA-mq44-7p77-q5h7	UNKNOWN	aiohttp	3.10	false
GHSA-mqqc-3gqh-h2x8	UNKNOWN	aiohttp	3.10	false
GHSA-mwh4-6h8g-pg8w	UNKNOWN	aiohttp	3.10	false
GHSA-p998-jp59-783m	UNKNOWN	aiohttp	3.10	false
GHSA-w2fm-2cpv-w7v5	UNKNOWN	aiohttp	3.10	false
GHSA-xcgm-r5h9-7989	UNKNOWN	aiohttp	3.10	false
PYSEC-2026-1097 through PYSEC-2026-1113 (17 entries)	UNKNOWN	aiohttp	3.10	false
PYSEC-2026-2094 through PYSEC-2026-2113 (20 entries)	UNKNOWN	aiohttp	3.10	false
PYSEC-2026-237	UNKNOWN	aiohttp	3.10	false
PYSEC-2026-3545 through PYSEC-2026-3547 (3 entries)	UNKNOWN	aiohttp	3.10	false
GHSA-g7vv-2v7x-gj9p	UNKNOWN	tqdm	4.66	false
PYSEC-2026-1976	UNKNOWN	tqdm	4.66	false

Grouped ranges collapsed for readability. Full per-CVE rows are in the workbook sheet CVEs.

APPENDIX E

License Findings Inventory

This appendix reproduces the full per-package license findings (22 of 22) and the full license clause findings (3 of 3).

E.1 Per-Package License Findings (Full, 22 of 22)

Package	Detected License	Vir l	Restricti ve	Confiden ce	Source File
tqdm	MPL-2.0 AND MIT	true	false	0.7	benchmarks/multi_turn/requirements.txt
(repo-level)	Apache-2.0	false	false	0.8	LICENSE
(repo-level)	MIT	false	false	0.8	third_party/flash_linear_attention/LICENSE
aiohttp	Apache-2.0	false	false	0.9	benchmarks/multi_turn/requirements.txt
matplotlib	PSF-2.0	false	false	0.9	tools/recipes/requirements.txt
native-tls-vendored	UNKNOWN	false	false	0	rust/src/cmd/Cargo.toml
numpy	BSD-3-Clause	false	false	0.9	benchmarks/multi_turn/requirements.txt
openai	UNKNOWN	false	false	0	examples/features/structured_outputs/pyproject.toml
pandas	BSD-3-Clause	false	false	0.9	benchmarks/kernels/requirements.txt
pandas	BSD-3-Clause	false	false	0.9	benchmarks/multi_turn/requirements.txt
pandas	BSD-3-Clause	false	false	0.9	tools/recipes/requirements.txt
pydantic	MIT	false	false	0.9	examples/features/structured_outputs/pyproject.toml
reqwest	UNKNOWN	false	false	0	rust/src/bench/Cargo.toml
seaborn	UNKNOWN	false	false	0	tools/recipes/requirements.txt
serde	UNKNOWN	false	false	0	rust/src/bench/Cargo.toml

Package	Detected License	Virul	Restrictive	Confidence	Source File
serde_json	UNKNOWN	false	false	0	rust/src/bench/Cargo.toml
tempfile	UNKNOWN	false	false	0	rust/src/engine-core-client/Cargo.toml
tokio	UNKNOWN	false	false	0	rust/src/cmd/Cargo.toml
tokio	UNKNOWN	false	false	0	rust/src/managed-engine/Cargo.toml
tokio	UNKNOWN	false	false	0	rust/src/mock-engine/Cargo.toml
transformers	Apache 2.0 License	false	false	0.7	benchmarks/multi_turn/requirements.txt
xlsxwriter	BSD-2-Clause	false	false	0.7	benchmarks/multi_turn/requirements.txt

E.2 License Clause Findings (Full, 3 of 3)

License File	Clause Type	Severity	Clause Text (excerpt)
LICENSE	attribution	attribution	any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form...
third_party/flash_linear_attention/LICENSE	warranty_disclaimer	permissive	shall be included in all copies or substantial portions of the Software. THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND...
LICENSE	patent_grant	permissive	sublicense, and distribute the Work and such Derivative Works in Source or Object form. 3. Grant of Patent License. Subject to the...

APPENDIX F

Complexity, Duplication & SQALE Detail

This appendix reproduces the complexity hotspot symbols (full set), the duplication groups, and the SQALE debt modules. The hotspot symbols table is the largest in the report; it is reproduced in full to allow the buyer's quality-of-earnings reviewer to probe specific symbols.

F.1 Complexity Hotspot Symbols (Top 30 of 65,109)

Name	File	C C	C o g	Ma int. Idx	Ma x Nes ting	Par ams	Com posite Risk
main	benchmarks/attention_benchmarks/benchmark.py	147	334	0	162	0	0.9
__post_init__	config/vllm.py	143	305	0	162	1	0.9125
schedule	v1/core/sched/scheduler.py	141	453	0	155	2	0.925
benchmark	benchmarks/serve.py	96	323	0	101	34	1.0
create_engine_config	engine/arg_utils.py	93	231	0	101	4	0.95
forward	v1/attention/backends/flashinfer.py	90	456	0	106	11	1.0
kernel	models/kimi_k3/nvidia/specialized_decode/cutestl/kda/kernel.py	86	295	8.47	84	2	0.9081
chat_completion_stream_generator	entrypoints/openai/chat_completion/serving.py	85	330	1.32	95	13	0.9974
from_args	rust/src/bench/src/config.rs	84	179	0	22	1	0.9125

Name	File	C C	C o g	Ma int. Idx	Ma x Nes ting	Par ams	Com posite Risk
__post_init__	config/speculative.py	8 0	2 1 0	2.6	90	1	0.907 3
_topk_topp_kernel	v1/sample/ops/topk_topp_triton.py	7 8	1 2 6	0	81	16	1.0
update_from_output	v1/core/sched/scheduler.py	7 8	2 4 4	4.8 6	83	4	0.940 3
main_async	benchmarks/serve.py	7 8	2 5 7	5.3 2	84	1	0.901 9
kernel	model_executor/layers/mamba/ops/gdn_chunk_cudtsl/kernel_h.py	7 6	1 2 0	0	81	12	1.0
kernel	model_executor/layers/mamba/ops/gdn_chunk_cudtsl/kernel_kkt_inv_uw.py	7 3	1 0 0	0	77	12	1.0
extract_tool_calls_streaming	tool_parsers/xlam_tool_parser.py	7 3	1 2 6	7.6	74	9	0.984 8
_update_states	v1/worker/gpu_model_runner.py	7 3	1 8 0	7.1 6	80	2	0.910 7
_support_torch_compile	compilation/decorators.py	7 3	2 7 3	4.5	88	9	0.991
_load_weights_quark	model_executor/models/gpt_oss.py	7 3	5 6 2	6.2 4	99	9	0.987 5
__init__	distributed/kv_transfer/kv_connector/v1/nixl/base_worker.py	7 0	1 6 3	4.3 1	78	5	0.953 9
sparse_attn_indexer_kpool	models/glm5next/nvidia/sparse_indexer.py	6 9	4 7 8	0	84	17	1.0
kernel	model_executor/kernels/linear/cute_dsl/gemm_rs_ar.py	6 8	2 3 6	0.6 1	79	12	0.998 8

Name	File	C C	C o g	Ma int. Idx	Ma x Nes ting	Par ams	Com posite Risk
sparse_attn_indexer	model_executor/layers/sparse_attn_indexer.py	6 8	3 5 7	1.5 3	82	28	0.996 9
run_benchmark	rust/src/bench/src/benchmark.rs	6 3	2 1 5	0	12	1	0.912 5
register_kv_caches	distributed/kv_transfer/kv_connector/v1/nixl/base_worker.py	6 3	3 0 5	5.5 9	77	3	0.926 3
triton_turboquant_unified_attention	v1/attention/ops/turboquant_soa/triton_turboquant_unified_attention.py	6 2	3 8 9	6.9 2	66	31	0.986 2
hf_config_override	config/speculative.py	6 1 1	1 0 1	9.4 4	47	1	0.893 6
sparse_attn_indexer_kpool	models/glm5next/amd/sparse_indexer.py	6 0	4 2 3	1.2 2	75	17	0.997 6
_dummy_run	v1/worker/gpu_model_runner.py	5 9	2 0 2	7.8 9	68	15	0.984 2
__post_init__	config/model.py	5 7	1 0 3	13. 07	60	31	0.973 9

Rendered 30 of 65,109 rows. Full inventory in workbook sheet Complex Symbols.

F.2 Duplication Groups (Sample)

26,488 duplication groups are recorded, totalling 742,640 duplicated lines of code. The duplication ratio of 35.46 percent is unusually high. The sample below illustrates the pattern: each row shows the block start line, block length, and number of occurrences. The high occurrence counts (up to 7) indicate that the same block has been copied multiple times, which is a refactorable debt pattern.

Block Start Line	Block Length	Occurrences
100	10	4 entries
100	10	7 entries
100	10	2 entries

Block Start Line	Block Length	Occurrences
101	10	5 entries
1,014	10	5 entries
1,014	10	4 entries
1,015	10	4 entries
1,017	10	7 entries
1,080	10	7 entries
1,086	10	5 entries

F.3 SQALE Debt Modules (Sample)

5,711 SQALE debt modules are recorded. The SQALE grade distribution is heavily weighted toward A (no debt) because the bulk of the modules are `__init__.py` files with zero base lines. The 43 D-E graded modules are the ones reconciled against the critical-complexity line item in Section 3.

Base Lines	Debt Ratio	Module	Remediation Effort	SQALE Grade
0	0	assets/ <code>__init__.py</code>	0	A
0	0	benchmarks/ <code>__init__.py</code>	0	A
0	0	benchmarks/kernels/ <code>__init__.py</code>	0	A
0	0	compilation/ <code>__init__.py</code>	0	A
0	0	compilation/passes/ <code>__init__.py</code>	0	A
0	0	distributed/kv_transfer/kv_connector/ <code>__init__.py</code>	0	A
0	0	distributed/ec_transfer/ec_connector/ <code>__init__.py</code>	0	A

APPENDIX G

Debt Marker Inventory

This appendix reproduces a sample of the 5,669 debt markers identified by the scanner. The 150 rows shown below cover the high-severity markers (fixme, hack) and a representative sample of the medium-severity markers (todo, workaround). The full inventory is in the workbook sheet Debt Markers.

G.1 Debt Markers (Sample of 150 of 5,669)

Filepath	Line	Type	Severity	Text
entrypoints/openai/responses/serving.py	182	hack	high	# HACK(woosuk): This is a hack. We should use a better store.
entrypoints/openai/responses/serving.py	183	fixme	high	# FIXME: If enable_store=True, this may cause a memory leak since we
entrypoints/openai/responses/serving.py	188	hack	high	# HACK(woosuk): This is a hack. We should use a better store.
entrypoints/openai/responses/serving.py	189	fixme	high	# FIXME: If enable_store=True, this may cause a memory leak since we
entrypoints/openai/responses/serving.py	193	hack	high	# HACK(wuhang): This is a hack. We should use a better store.
entrypoints/openai/responses/serving.py	194	fixme	high	# FIXME: If enable_store=True, this may cause a memory leak since we
entrypoints/pooling/scoring/io_processor.py	702	fixme	high	# FIXME: For now, we only apply a template when one is explicitly provided.
entrypoints/pooling/scoring/io_processor.py	709	fixme	high	# FIXME:
kernels/helion/register.py	65	fixme	high	# FIXME(gmagogsfm): Re-enable HOP path once performance regression is fixed.
lora/layers/fused_moe.py	552	hack	high	# HACK: Currently, only GPT-OSS is in interleaved order
lora/model_manager.py	728	hack	high	# HACK: overrides replacements for qkvz = qkv + z case.
lora/model_manager.py	861	hack	high	# HACK Temporary solution for the pool model.
lora/model_manager.py	959	fixme	high	# FIXME Edge case where LoRA is not added to gate_up_proj

Filepath	Line	Type	Severity	Text
model_executor/kernels/linear/mixed_precision/cpu.py	110	fixme	high	# FIXME: some bugs in convert_weight_packed_scale_zp with GPTQ format,
model_executor/kernels/linear/mixed_precision/xpu.py	82	fixme	high	# (FIXME): maybe zero points should also be transposed.
model_executor/layers/fused_moe/experts/batched_deep_gemm_moe.py	375	fixme	high	# FIXME (varun): We should be able to dispatch only from the leader
model_executor/layers/fused_moe/layer.py	349	fixme	high	# FIXME (varun): We should have a better way of inferring the activation
model_executor/layers/fused_moe/oracle/unquantized.py	82	hack	high	# HACK: Qwen3.5 has crash with FLASHINFER_CUTLASS BF16 if DEP.
model_executor/layers/fused_moe/oracle/unquantized.py	88	hack	high	# HACK: unquantized FlashInfer aliases SWIGLUOAI to plain Swiglu
model_executor/layers/fused_moe/routed_experts.py	699	fixme	high	# (FIXME) for gpt-oss all experts are combined
tests/basic_correctness/memory/sleep_mode/test_sleep_mode.py	57	fixme	high	# FIXME(youkaichao & ywang96): Fix memory buffer issue with sleep mode
tests/compile/passes/test_fusion_attn.py	390	hack	high	result_unfused_0 = model_unfused(q, k, v) # noqa: F841 HACK: See #131044
tests/compile/passes/test_fusion_attn.py	428	hack	high	# HACK: See https://github.com/vllm-project/vllm/issues/31044
tests/compile/passes/test_mla_attn_quant_fusion.py	491	hack	high	# HACK: See #131044
tests/compile/passes/test_mla_attn_quant_fusion.py	531	hack	high	# HACK: See https://github.com/vllm-project/vllm/issues/31044
tests/config/test_model_arch_config.py	41	fixme	high	# FIXME: databricks/dbx-instruct has been deleted

Filepath	Line	Type	Severity	Text
tests/confestest.py	613	hack	high	# HACK - not all processors take sampling_rate; we should
tests/detokenizer/test_stop_strings.py	119	fixme	high	# FIXME: this does not respect include_in_output=False
tests/entrypoints/llm/test_struct_output_generate.py	142	fixme	high	# FIXME: Since "auto" will use Mistral tokenizer and these backends do not support
tests/entrypoints/llm/test_struct_output_generate.py	175	fixme	high	# FIXME: This tests are flaky on CI thus disabled. Tracking in Issue #24402
tests/kernels/attention/test_attention_selector.py	71	fixme	high	"cpu": [], # FIXME(woosuk): Temporarily disable CPU tests
tests/kernels/moe/test_ocp_mx_moe.py	372	fixme	high	epilogue_tile_m = 128 # FIXME: this depends on the kernel internals
tests/model_executor/test_ernie45_vl_mrope.py	54	hack	high	field=None, # HACK.
tests/model_executor/test_keye_mrope.py	56	hack	high	field=None, # HACK.
tests/model_executor/test_keye_vll_5_mrope.py	57	hack	high	field=None, # HACK.
tests/model_executor/test_qwen3_omni_mrope.py	64	hack	high	field=None, # HACK.
tests/model_executor/test_qwen3_vl_mrope.py	145	hack	high	field=None, # HACK.
tests/models/multimodal/generation/test_phi4mm.py	132	fixme	high	# FIXME: https://huggingface.co/microsoft/Phi-4-multimodal-instruct/discussions/75
tests/models/multimodal/processing/test_tensor_schema.py	162	fixme	high	# FIXME(Isotr0py): Fix Kimi-K2.5's offline inference about vision chunks.
tests/models/registry.py	1,537	fixme	high	# FIXME: Investigate the NVML failure in CI.
tests/models/registry.py	242	fixme	high	# FIXME: databricks/dbx-instruct has been deleted

Filepath	Line	Type	Severity	Text
tests/models/test_initialization.py	191	fixme	high	# FIXME: A hack to bypass FA3 assertion because our CI's L4 GPU
tests/samplers/test_beam_search.py	43	fixme	high	# FIXME(zhuohan): The test can not pass if we:
tests/tool_use/utlis.py	177	fixme	high	# FIXME: This test currently fails, need to debug why.
tests/v1/sample/test_logprobs_e2e.py	14	fixme	high	# FIXME(rob): enable prefix caching once supported.
tests/v1/structured_output/test_backend_xgrammar.py	596	fixme	high	# FIXME(arpera):
tool_parsers/llama4_pythonic_tool_parser.py	189	hack	high	# HACK: serving_chat.py inspects the internal state of tool parsers
tool_parsers/olmo3_tool_parser.py	206	hack	high	# HACK: serving_chat.py inspects the internal state of tool parsers
tool_parsers/pythonic_tool_parser.py	178	hack	high	# HACK: serving_chat.py inspects the internal state of tool parsers
transformers_utils/configs/mimo_v2_omni.py	28	hack	high	kv_channels=64, # HACK
transformers_utils/model_arch_config_convertor.py	117	fixme	high	# FIXME(woosuk): This may not be true for all models.
v1/attention/backends/flex_attention.py	1,14	fixme	high	# FIXME(Isotr0py): direct build has issue to build bidirectional
v1/attention/backends/mla/indexer.py	734	fixme	high	# FIXME (zyongye)
v1/core/kv_cache_manager.py	162	fixme	high	# FIXME: make prefix cache stats conditional on log_stats. We still need
v1/core/kv_cache_planning.py	471	fixme	high	# FIXME(Chen): At the moment of writing this code (2025-06-02), all
v1/metrics/perf.py	1,156	fixme	high	# FIXME: might have to make this more granular

Filepath	Line	Type	Severity	Text
v1/metrics/perf.py	1,289	fixme	high	# FIXME: Assume perfect load balancing for now.
v1/metrics/perf.py	362	fixme	high	# FIXME: handle this better
v1/sample/rejection_sampler.py	739	fixme	high	# FIXME(woosuk): Because is_greedy_ptr is not None at profiling run,
v1/simple_kv_offload/manager.py	851	fixme	high	# FIXME (yifan): handle CPU cache eviction, where
v1/spec_decode/llm_base_proposer.py	1,624	fixme	high	# FIXME: when using tree-based specdec, adjust number of forward-passes
v1/spec_decode/llm_base_proposer.py	1,819	fixme	high	# FIXME(woosuk): The logic here is duplicated with the main sampling code.
v1/structured_output/backend_xgrammar.py	135	fixme	high	# FIXME(arpera):
v1/structured_output/backend_xgrammar.py	346	fixme	high	# FIXME(arpera): The approach used here needs to be redesigned because of
v1/structured_output/backend_xgrammar.py	84	fixme	high	# FIXME(arpera):
v1/worker/gpu/model_runner.py	771	hack	high	# HACK(woosuk)
v1/worker/gpu/model_runner.py	854	hack	high	# HACK(lucas): for now since the worker is shared between MRV1 and MRV2,
v1/worker/gpu_worker.py	579	hack	high	# HACK(woosuk): This is a temporary fix to avoid type errors.
v1/worker/gpu_worker.py	589	fixme	high	# FIXME(youkaichao & ywang96): Use TorchDispatchMode instead of memory pool

Rendered 68 of 5,669 rows (high-severity fixme/hack markers). The full inventory including medium-severity todo and low-severity debug_print markers is in the workbook sheet Debt Markers.

APPENDIX H

Attack Surface Candidate Endpoints

This appendix reproduces a sample of the 2,154 candidate entry points identified by the attack-surface scanner. The first 70 rows are shown below, covering the production endpoints (OpenAI-compatible API routers, RL online router, serve dev routers) and the example application servers. The remaining 2,084 rows are in the workbook sheet Attack Surface.

H.1 Candidate Entry Points (Sample of 70 of 2,154)

Symbol	File	Line	Type	Exposure	Framework	Risk
health	entrypoints/cli/preload.py	75	http_handler	external	express	critical
health	entrypoints/launchers/dp_supervisor.py	229	http_handler	external	express	critical
ready	entrypoints/launchers/dp_supervisor.py	234	http_handler	external	express	critical
show_available_models	entrypoints/openai/models/api_router.py	21	http_handler	external	fastapi	critical
cancel_responses	entrypoints/openai/responses/api_router.py	112	http_handler	external	fastapi	critical
retrieve_responses	entrypoints/openai/responses/api_router.py	82	http_handler	external	fastapi	critical
create_classify	entrypoints/pooling/classify/api_router.py	29	http_handler	external	fastapi	critical
is_paused	entrypoints/rl/online/api_router.py	143	http_handler	external	fastapi	critical
init_weight_transfer_engine	entrypoints/rl/online/api_router.py	160	http_handler	external	fastapi	critical
shutdown_weight_transfer_engine	entrypoints/rl/online/api_router.py	172	http_handler	external	fastapi	critical
start_weight_update	entrypoints/rl/online/api_router.py	179	http_handler	external	fastapi	critical
start_draft_weight_update	entrypoints/rl/online/api_router.py	186	http_handler	external	fastapi	critical
finish_weight_update	entrypoints/rl/online/api_router.py	205	http_handler	external	fastapi	critical

Symbol	File	Line	Type	Exposure	Framework	Risk
update_weight_version	entrypoints/rl/online/api_router.py	215	http_handler	external	fastapi	critical
weight_info	entrypoints/rl/online/api_router.py	224	http_handler	external	fastapi	critical
weight_checker	entrypoints/rl/online/api_router.py	230	http_handler	external	fastapi	critical
get_world_size	entrypoints/rl/online/api_router.py	267	http_handler	external	fastapi	critical
pause_generation	entrypoints/rl/online/api_router.py	32	http_handler	external	fastapi	critical
resume_generation	entrypoints/rl/online/api_router.py	76	http_handler	external	fastapi	critical
abort_requests	entrypoints/rl/online/api_router.py	95	http_handler	external	fastapi	critical
reset_prefix_cache	entrypoints/serve/dev/cache/api_router.py	21	http_handler	external	fastapi	critical
reset_mm_cache	entrypoints/serve/dev/cache/api_router.py	48	http_handler	external	fastapi	critical
collective_rpc	entrypoints/serve/dev/rpc/api_router.py	24	http_handler	external	fastapi	critical
show_server_info	entrypoints/serve/dev/server_info/api_router.py	44	http_handler	external	fastapi	critical
sleep	entrypoints/serve/dev/sleep/api_router.py	27	http_handler	external	fastapi	critical
release_kv_cache_memory	entrypoints/serve/dev/sleep/api_router.py	38	http_handler	external	fastapi	critical
wake_up	entrypoints/serve/dev/sleep/api_router.py	45	http_handler	external	fastapi	critical
is_sleeping	entrypoints/serve/dev/sleep/api_router.py	59	http_handler	external	fastapi	critical
attach_router	entrypoints/serve/dev/sleep/api_router.py	64	http_handler	external	fastapi	critical
is_scaling_elastic_ep	entrypoints/serve/elastic_ep/api_router.py	85	http_handler	external	fastapi	critical
get_status	entrypoints/serve/fault_tolerance/api_router.py	94	http_handler	external	fastapi	critical
get_server_load_metrics	entrypoints/serve/instrumentator/basic.py	31	http_handler	external	fastapi	critical

Symbol	File	Line	Type	Exposure	Framework	Risk
show_version	entrypoints/serve/instrumentator/basic.py	54	http_handler	external	fastapi	critical
kv_event_sources	entrypoints/serve/instrumentator/basic.py	60	http_handler	external	fastapi	critical
health	entrypoints/serve/instrumentator/health.py	23	http_handler	external	fastapi	critical
custom_swagger_ui_html	entrypoints/serve/instrumentator/offline_docs.py	37	http_handler	external	express	critical
swagger_ui_redirect	entrypoints/serve/instrumentator/offline_docs.py	47	http_handler	external	express	critical
load_lora_adapter	entrypoints/serve/lora/api_router.py	64	http_handler	external	fastapi	critical
unload_lora_adapter	entrypoints/serve/lora/api_router.py	84	http_handler	external	fastapi	critical
start_profile	entrypoints/serve/profile/api_router.py	36	http_handler	external	fastapi	critical
stop_profile	entrypoints/serve/profile/api_router.py	49	http_handler	external	fastapi	critical
ping	entrypoints/serve/sagemaker/api_router.py	94	http_handler	external	fastapi	critical
get_tokenizer_info	entrypoints/serve/tokenize/api_router.py	91	http_handler	external	fastapi	critical
generate	examples/applications/api_server/server.py	45	http_handler	external	flask	critical
list_models	examples/disaggregated/disaggregated_encoder/disagg_epd_proxy.py	1, 21, 5	http_handler	external	express	critical
health_check	examples/disaggregated/disaggregated_encoder/disagg_epd_proxy.py	1, 22, 2	http_handler	external	express	critical
list_instances	examples/disaggregated/disaggregated_encoder/disagg_epd_proxy.py	1, 37, 2	http_handler	external	express	critical
register_instance	examples/disaggregated/disaggregated_encoder/disagg_epd_proxy.py	1, 37, 6	http_handler	external	express	critical
health	examples/disaggregated/disaggregated_encoder/disagg_epd_proxy.py	1, 43, 6	http_handler	external	express	critical

Symbol	File	Line	Type	Exposure	Framework	Risk
chat_completions	examples/disaggregated/disaggregated_encoder/disagg_epd_proxy.py	1, 174	http_handler	external	flask	critical
start_profile	examples/disaggregated/disaggregated_encoder/disagg_epd_proxy.py	1, 328	http_handler	external	flask	critical
stop_profile	examples/disaggregated/disaggregated_encoder/disagg_epd_proxy.py	1, 338	http_handler	external	flask	critical
unregister_instance	examples/disaggregated/disaggregated_encoder/disagg_epd_proxy.py	1, 396	http_handler	external	flask	critical
health	examples/disaggregated/disaggregated_server/disagg_proxy_multiturn.py	524	http_handler	external	express	critical
chat_completions	examples/disaggregated/disaggregated_server/disagg_proxy_multiturn.py	514	http_handler	external	flask	critical
health	examples/disaggregated/disaggregated_server/moriio_toy_proxy_server.py	260	http_handler	external	flask	critical
handle_completions_request	examples/disaggregated/disaggregated_server/moriio_toy_proxy_server.py	268	http_handler	external	flask	critical
handle_chat_completions_request	examples/disaggregated/disaggregated_server/moriio_toy_proxy_server.py	273	http_handler	external	flask	critical
handle_completions	examples/disaggregated/lmcache/disagg_prefill_lmcache_v1/disagg_proxy_server.py	138	http_handler	external	flask	critical
handle_chat_completions	examples/disaggregated/lmcache/disagg_prefill_lmcache_v1/disagg_proxy_server.py	175	http_handler	external	flask	critical
handle_completions	examples/disaggregated/mooncake_connector/mooncake_connector_proxy.py	357	http_handler	external	flask	critical
handle_chat_completions	examples/disaggregated/mooncake_connector/mooncake_connector_proxy.py	362	http_handler	external	flask	critical
update_weights	entrypoints/rl/online/api_router.py	193	http_handler	external	fastapi	critical
reset_encoder_cache	entrypoints/serve/dev/cache/api_router.py	58	http_handler	external	fastapi	critical
register_fault_tolerance_api_router	entrypoints/serve/fault_tolerance/api_router.py	99	http_handler	external	fastapi	critical
health	examples/applications/api_server/server.py	39	http_handler	external	express	critical

Symbol	File	Line	Type	Expense	Framework	Risk
detect	examples/basic/online_serving/watermark_detection_server.py	32	http_handler	external	flask	critical
healthy	examples/disaggregated/disaggregated_encoder/disagg_epd_proxy.py	1, 22, 3	http_handler	external	express	critical
completions	examples/disaggregated/disaggregated_serving/disagg_proxy_multiturn.py	51, 9	http_handler	external	flask	critical

Rendered 70 of 2,154 rows. The remaining 2,084 rows are in the workbook sheet Attack Surface and include application_entry endpoints in the swagger-ui-bundle.js static asset.

APPENDIX I

Key-Person & Knowledge Loss Vulnerability Detail

This appendix reproduces a sample of the key-person risk rows, module bus-factor rows, and knowledge loss vulnerability rows. The full inventories are in the workbook sheets Key-person Risks and Knowledge Loss Vulnerability.

I.1 Key-Person Risk Rows (Top 30 of 3,793)

Author	Email	Risk Score	Commits	Departure Impact
hmellor	19981378+hmellor@users.noreply.github.com	0.387	109,408	medium
simon.mo	simon.mo@hey.com	0.145	16,391	low
rbryant	rbryant@redhat.com	0.101	3,756	low
yuqi.wang	yuqi.wang@daocloud.io	0.092	31,685	low
tlleungac	tlleungac@connect.ust.hk	0.091	57,520	low
bugenzhao	i@bugenzhao.com	0.084	98,951	low
youkaichao	youkaichao@gmail.com	0.084	22,325	low
yewentao256	44945378+yewentao256@users.noreply.github.com	0.077	13,736	low
mgoin64	mgoin64@gmail.com	0.069	31,528	low
akaratza	akaratza@amd.com	0.064	28,912	low

Author	Email	Risk Score	Commits	Departure Impact
riverclouds.zhu	riverclouds.zhu@qq.com	0.058	48,003	low
nhill	nhill@redhat.com	0.058	7,303	low
zhuohan123	zhuohan123@gmail.com	0.055	3,076	low
LucasWilkinson	LucasWilkinson@users.noreply.github.com	0.054	25,780	low
Jialin.Ouyang	Jialin.Ouyang@gmail.com	0.054	1,809	low
nickhill123	nickhill123@gmail.com	0.052	13,366	low
woosuk.kwon	woosuk.kwon@berkeley.edu	0.050	19,760	low
martin.hickey	martin.hickey@ie.ibm.com	0.049	8,352	low
aoshen	aoshen@inferact.ai	0.046	7,955	low
khluu000	khluu000@gmail.com	0.045	12,171	low
felmarty	felmarty@amd.com	0.040	12,250	low
wzhao18	51183510+wzhao18@users.noreply.github.com	0.040	4,495	low
chaojun.zhang	chaojun.zhang@intel.com	0.039	2,463	low
markmc	markmc@redhat.com	0.039	3,442	low
contact@aarndphm	contact@aarndphm.xyz	0.037	1,973	low
rkooo567	rkooo567@gmail.com	0.037	614	low
alexm-redhat	59768536+alexm-redhat@users.noreply.github.com	0.037	3,507	low
qianlihuang	91178480+qianlihuang@users.noreply.github.com	0.035	155	low
mbonanni	mbonanni@redhat.com	0.031	25,488	low
chaunceyjiang	chaunceyjiang@gmail.com	0.030	13,743	low

Rendered 30 of 3,793 rows.

I.2 Module Bus-Factor Rows (Top 30 of 761)

Module Path	Bus Factor	Total Files
vllm/model_executor/models	37	284
_unassigned	21	1,885
vllm	20	22
tests/v1/attention	16	67
tests/v1/kv_connector/unit	16	60
vllm/v1/worker	16	24
tests/v1/worker	15	59

Module Path	Bus Factor	Total Files
vllm/model_executor/layers	15	19
tests/kernels/moe	14	72
tests	13	31
vllm/config	13	36
vllm/utils	13	41
vllm/v1/attention/backends	13	29
tests/distributed	12	61
tests/kernels/attention	12	51
tests/model_executor	12	40
benchmarks/kernels	11	68
tests/v1/core	11	22
vllm/distributed/device_communicators	11	20
vllm/platforms	11	8
tests/compile	10	15
tests/entrypoints/openai/chat_completion	10	15
tests/reasoning	10	24
tests/v1/spec_decode	10	33
tests/v1/structured_output	10	13
vllm/engine	10	4
vllm/model_executor/layers/fused_moe/experts	10	38
vllm/v1	10	8
benchmarks	9	19
tests/models/multimodal/processing	9	47

Rendered 30 of 761 rows.

I.3 Knowledge Loss Vulnerability Rows (Top 30 of 3,793, critical)

Author	Email	Tenure (d)	Days Since Last	Blast Radius At Risk
Juan Villamizar	100237675+jpvillam-amd@users.noreply.github.com	919	0	0
Robert Shaw	114415538+robertgshaw2-redhat@users.noreply.github.com	625	1	73
Fadi Arafeh	115173828+fadara01@users.noreply.github.com	385	26	0

Author	Email	Tenure (d)	Days Since Last	Blast Radius At Risk
Juan Pérez de Algaba	124347725+jperezdealgaba@users.noreply.github.com	202	1	128
Nathan Price	125999937+TheCodeWrangler@users.noreply.github.com	302	127	0
Divakar Verma	137818590+divakar-amd@users.noreply.github.com	870	3	0
ylangtsou	149562838+ylangtsou@users.noreply.github.com	147	76	0
Matt	156021403+mawong-amd@users.noreply.github.com	929	3	0
Vadim Gimpelson	156319763+vadiklyutiy@users.noreply.github.com	578	2	0
ai-jz	156989844+ai-jz@users.noreply.github.com	348	14	0
tc-mb	157115220+tc-mb@users.noreply.github.com	435	4	0
kliuae	17350011+kliuae@users.noreply.github.com	1,032	7	0
Xiaoshuang Wang	1790571317@qq.com	391	17	0
Srinivas Krovvidi	194645829+Srinivasoo7@users.noreply.github.com	219	67	0
copilot-swe-agent[bot]	198982749+Copilot@users.noreply.github.com	421	169	0
Harry Mellor	19981378+hmellor@users.noreply.github.com	997	0	0
qizixi	22851944+zixi-qi@users.noreply.github.com	510	0	0
Johnny Yang	24908445+jcyang43@users.noreply.github.com	388	162	0
Ekagra Ranjan	3116519+ekagra-ranjan@users.noreply.github.com	562	104	0
kourosh hakhamaneshi	31483498+kouroshHakha@users.noreply.github.com	490	121	0
Giancarlo Delfin	32987265+TheEpicDolphin@users.noreply.github.com	438	2	0
Alec	35311602+alec-flowers@users.noreply.github.com	533	1	0
Roberto L. Castro	38211239+LopezCastroRoberto@users.noreply.github.com	438	2	0
HDCharles	39544797+HDCharles@users.noreply.github.com	366	17	0
Sumanth R Hegde	39546518+SumanthRH@users.noreply.github.com	350	117	0
Asaf Gardin	39553475+Josephasafg@users.noreply.github.com	444	27	0
Julien Denize	40604584+julindenize@users.noreply.github.com	576	35	0
zhrrr	43847754+izhuhaoran@users.noreply.github.com	501	84	0

Author	Email	Tenure (d)	Days Since Last	Blast Radius At Risk
Islam	43927639+almersawi@users.noreply.github.com	507	30	0
cjackal	44624812+cjackal@users.noreply.github.com	705	10	0

Rendered 30 of 3,793 rows. All rows shown carry a critical retention risk rating.

APPENDIX J

Maintenance, IaC & CI/CD Detail

This appendix reproduces the maintenance assets (none recorded), IaC findings (11 of 11), CI files (12 of 12), and coverage artifacts (none recorded).

J.1 Maintenance Assets (Full)

No rows are recorded for this evidence item in the current scan. The asset-staleness scanner was not configured or could not access the file population required; the maintenance pillar is therefore not assessed, as documented in Section 15.

J.2 IaC Findings (Full, 11 of 11)

Rule	Title	Severity	File	Line	Remediation
IAC-DOCKER-ROOT	Container runs as root	medium	docker/Dockerfile.cpu	494	Add a dedicated non-root user and USER directive in every stage
IAC-DOCKER-ROOT	Container runs as root	medium	docker/Dockerfile.rock	1,018	Add a dedicated non-root user and USER directive in every stage
IAC-DOCKER-ROOT	Container runs as root	medium	docker/Dockerfile.rock_base	453	Add a dedicated non-root user and USER directive in every stage

Rule	Title	Severity	File	Line	Remediation
IAC-DOCKER-ROOT	Container runs as root	medium	docker/Dockerfile.rocm	1,018	Add a dedicated non-root user and USER directive in every stage
IAC-DOCKER-ROOT	Container runs as root	medium	docker/Dockerfile.rocm_base	479	Add a dedicated non-root user and USER directive in every stage
IAC-DOCKER-ROOT	Container runs as root	medium	docker/Dockerfile.rocm_base_gfx1250	315	Add a dedicated non-root user and USER directive in every stage
IAC-DOCKER-ROOT	Container runs as root	medium	docker/Dockerfile.rocm_gfx1250	744	Add a dedicated non-root user and USER directive in every stage
IAC-DOCKER-ROOT	Container runs as root	medium	docker/Dockerfile.xpu	286	Add a dedicated non-root user and USER directive in every stage
IAC-DOCKER-ROOT	Container runs as root	medium	docker/Dockerfile.zen	98	Add a dedicated non-root user and USER directive in every stage
IAC-DOCKER-ROOT	Container runs as root	medium	docs/assets/contributing/dockerfile-stages-dependency.png	10,445	Add a dedicated non-root user and USER directive in every stage

Rule	Title	Severity	File	Line	Remediation
IAC-DOCKER-ROOT	Container runs as root	medium	docs/contributing/dockerfile/dockerfile.md	67	Add a dedicated non-root user and USER directive in every stage

J.3 CI Files (Full, 12 of 12)

CI File	Purpose (inferred)
.github/workflows/add_label_automerge.yml	Label and automerge administration
.github/workflows/buf.yml	Protobuf linting (Buf)
.github/workflows/issue_autolabel.yml	Issue auto-labelling
.github/workflows/macOS-smoke-test.yml	macOS smoke test (only test-execution workflow)
.github/workflows/new_pr_bot.yml	New-PR bot administration
.github/workflows/notify-ci-authorized.yml	CI authorisation notification
.github/workflows/pr-title.yml	PR title linting
.github/workflows/pre-commit.yml	Pre-commit hook linting
.github/workflows/proto-crate.yml	Protobuf crate build
.github/workflows/record-ci-approval.yml	CI approval recording
.github/workflows/run-ci-command.yml	CI command runner
.github/workflows/stale.yml	Stale-issue administration

J.4 Coverage Artifacts

No coverage artifacts are recorded for this evidence item in the current scan. The CI does not produce or report coverage data.

APPENDIX K

Privacy & Secrets Detail

This appendix reproduces the full privacy findings (18 of 18) and the full secret candidates (6 of 6).

K.1 Privacy Findings (Full, 18 of 18)

Rule	Category	Title	Severity	File	Line	Remediation
PRIV-SENSITIVE-LOG	sensitive_logging	Sensitive data in log statement (token)	medium	.github/workflows/macos-smoke-test.yml	89	Redact the value; log a reference id and audit retention
PRIV-SENSITIVE-LOG	sensitive_logging	Sensitive data in log statement (token)	medium	benchmarks/benchmark_batch_invariance.py	117	Redact the value; log a reference id and audit retention
PRIV-SENSITIVE-LOG	sensitive_logging	Sensitive data in log statement (token)	medium	benchmarks/benchmark_hidden_state_extraction.py	309	Redact the value; log a reference id and audit retention
PRIV-SENSITIVE-LOG	sensitive_logging	Sensitive data in log statement (token)	medium	benchmarks/benchmark_serving_structured_output.py	209	Redact the value; log a reference id and audit retention
PRIV-SENSITIVE-LOG	sensitive_logging	Sensitive data in log statement (token)	medium	benchmarks/fused_kernels/merge_attention_benchmarks.py	254	Redact the value; log a reference id and audit retention

Rule	Category	Title	Severity	File	Line	Remediation
PRIV-SENSITIVE-LOG	sensitive_logging	Sensitive data in log statement (token)	medium	benchmarks/kernels/bench_concat_mla_q.py	90	Redact the value; log a reference id and audit retention
PRIV-SENSITIVE-LOG	sensitive_logging	Sensitive data in log statement (token)	medium	benchmarks/kernels/bench_cp_gather_fp8.py	143	Redact the value; log a reference id and audit retention
PRIV-SENSITIVE-LOG	sensitive_logging	Sensitive data in log statement (token)	medium	benchmarks/kernels/benchmark_2d_silu_mul_fp8_quant.py	194	Redact the value; log a reference id and audit retention
PRIV-SENSITIVE-LOG	sensitive_logging	Sensitive data in log statement (token)	medium	benchmarks/kernels/benchmark_kimi_k3_kda_projection.py	217	Redact the value; log a reference id and audit retention
PRIV-SENSITIVE-LOG	sensitive_logging	Sensitive data in log statement (token)	medium	benchmarks/kernels/benchmark_moe.py	709	Redact the value; log a reference id and audit retention

Rule	Category	Title	Severity	File	Line	Remediation
PRIV-SENSITIVE-LOG	sensitive_logging	Sensitive data in log statement (token)	medium	benchmarks/kernels/benchmark_mrope.py	175	Redact the value; log a reference id and audit retention
PRIV-SENSITIVE-LOG	sensitive_logging	Sensitive data in log statement (token)	medium	benchmarks/kernels/benchmark_reshape_and_cache.py	130	Redact the value; log a reference id and audit retention
PRIV-SENSITIVE-LOG	sensitive_logging	Sensitive data in log statement (token)	medium	benchmarks/kernels/benchmark_reshape_and_cache_flash.py	161	Redact the value; log a reference id and audit retention
PRIV-SENSITIVE-LOG	sensitive_logging	Sensitive data in log statement (token)	medium	benchmarks/kernels/benchmark_silu_mul_fp8_quant.py	717	Redact the value; log a reference id and audit retention
PRIV-SENSITIVE-LOG	sensitive_logging	Sensitive data in log statement (token)	medium	benchmarks/multi_turn/benchmark_serving_multi_turn.py	1,593	Redact the value; log a reference id and audit retention

Rule	Category	Title	Severity	File	Line	Remediation
PRIV-SENSITIVE-LOG	sensitive_logging	Sensitive data in log statement (token)	medium	benchmarks/multi_turn/convert_sharegpt_to_openai.py	134	Redact the value; log a reference id and audit retention
PRIV-NO-CONSENT-POLICY	consent_policy	No consent / privacy-policy signals detected	low	not quantified	0	Confirm privacy policy, cookie consent, and data-subject request handling exist outside the repo
PRIV-EGRESS-SDK	data_egress	Third-party data-egress SDK (llm_api)	low	not quantified	0	Verify DPA/processor coverage, transfer safeguards, and sub-processor disclosures for the SDK family

K.2 Secret Candidates (Full, 6 of 6)

Kind	File	Line	Confidence	In History	Risk Cap	Path Tier
generic_password_assignment	docker/entrypoints/test_vllm_nonroot_entrypoint.sh	161	60	false	medium	test
generic_password_assignment	docker/entrypoints/test_vllm_nonroot_entrypoint.sh	140	61	false	medium	test

Kind	File	Line	Confidence	In History	Risk Cap	Path Tier
generic_password_assignment	docker/entrypoints/test_vllm_nonroot_entrypoint.sh	123	61	false	medium	test
generic_password_assignment	entrypoints/serve/instrumentator/static/swagger-ui-bundle.js	3	60	false	medium	prod
generic_password_assignment	entrypoints/serve/instrumentator/static/swagger-ui-bundle.js	3	60	false	medium	prod
generic_password_assignment	entrypoints/serve/instrumentator/static/swagger-ui-bundle.js	3	60	false	medium	prod

APPENDIX L

AI/ML Inventory

This appendix records the AI/ML inventory. No model files are recorded in the present scan, which is the most important finding for the AI/ML pillar: the asset is an inference serving platform that does not ship model weights. The framework hits and SDK hits are reproduced in Section 12.

L.1 Model Files

No rows are recorded for this evidence item in the current scan. The absence of bundled model files, production model files, and unlicensed model files is the AI/ML pillar's positive value driver.

APPENDIX M

Integrity & Data-Quality Issues

This appendix reproduces the full integrity issues register (11 of 11). Each row identifies the integrity warning code, the subject (the pillar the warning applies to), and the message that explains the warning.

M.1 Integrity Issues (Full, 11 of 11)

Tier	Code	Subject	Message (summary)
warn	critical_pillar_no_findings	pillars.ci_maturity	Pillar is high but renders no item-level findings — posture held on aggregate metrics
warn	critical_pillar_no_findings	pillars.coverage	Pillar is high but renders no item-level findings — posture held on aggregate metrics
warn	critical_pillar_no_findings	pillars.entanglement	Pillar is critical but renders no item-level findings — posture held on aggregate metrics
warn	critical_pillar_no_findings	pillars.integration	Pillar is high but renders no item-level findings — posture held on aggregate metrics
warn	critical_pillar_no_findings	pillars.ip_ownership	Pillar is high but renders no item-level findings — posture held on aggregate metrics
warn	critical_pillar_no_findings	pillars.test_deficit	Pillar is high but renders no item-level findings — posture held on aggregate metrics
warn	unverified_pillar	pillars.cve	Partial — 2 packages could not be resolved; advisory check covered only 8 resolvable packages
warn	unverified_pillar	pillars.dependency	Partial — no lockfile; transitive resolution could not be performed; direct pins cover only 10%
warn	unverified_pillar	pillars.eol	Not assessed — zero evidence population; no manifest or lockfile in a supported ecosystem was discovered
warn	unverified_pillar	pillars.maintenance	Not assessed — no files were scanned by the asset-staleness scanner
warn	unverified_pillar	pillars.secrets	Partial — 3 path-excluded candidate paths not scanned; 5,712 files scanned

APPENDIX N

Taint Analysis Summary

This appendix reproduces the taint summary. Six taint flows were identified, and the taint pass is reported as unbounded (the analysis was not terminated by a depth or time limit). Taint flows are subsumed within the taint_remediation cost category in Section 3.2.

N.1 Taint Summary

Metric	Value
Bounded	false
Total flows	6

N.2 Top Taint Flows

6 flows. The top taint flows are not itemised in the source payload; the buyer should commission an independent taint analysis against the production endpoints identified in Section 5 to identify the specific source-to-sink paths.

APPENDIX O

Financial Reproduction Fingerprint

The financial reproduction fingerprint binds the report to the exact configuration used to produce every number: same inputs in, same outputs out. Any change to a number-driving input produces a different fingerprint and a different report under a new fingerprint. The full calibration, lineage, and sensitivity detail lives in Section 3, not re-rendered here.

O.1 Reproduction Fingerprint

Field	Value
Fingerprint (SHA-256)	55dd3cc574f7bca893cd23e5393a568f76f6722912f78e7a688d24c1c537efec
Economic model fingerprint (0.3.0)	3fddfd39700532e713e193bd892d605d0b82cfa1c6abc719047204da2dfbf703
Bound configuration	tool_version, commit=fa75d3456d60a5ffe923818c4b495295675b38e1, repo_path=vllm, mode=deal_thesis=True, offline=False, scan_history=True

APPENDIX P

Evidence Provenance

This appendix maps every key figure in the report to its source path in the stored payload and the workbook sheet that carries the same evidence. The payload path is the single stored source the narrative's evidence derives from; key-variant fallbacks are resolved at render

time and are not listed here. The workbook column names the sheet that inventories the same evidence; not applicable means no dedicated sheet carries it.

P.1 Evidence Provenance (Selected Key Figures)

ID	Evidence	Source	Workbook
1.1	Board decision	decision.board_decision	Dashboard
1.2	Confidence score	decision.confidence_score	Dashboard
1.3	Decision driver	decision.escalation.driver	Dashboard
1.4	Remediation cost as % of proposed price	decision.remediation_cost_pct_of_acquisition	Dashboard
1.5	Proposed price	decision.proposed_price	Dashboard
1.6	Total remediation cost	financial.total_remediation_cost	Dashboard
1.7	Total remediation hours	financial.total_remediation_hours	Dashboard
1.8	Valuation discount applied	financial.valuation_discount_pct	Dashboard
1.9	Post-merge integration drag	financial.post_merge_integration_drag_pct	Dashboard
1.10	Pillar register size	decision.pillar_register	Dashboard
1.11	Top risk findings	decision.top_risks	Dashboard
1.12	Major risks	decision.major_risks	Dashboard
1.13	Integrity status	integrity.status	Contents
2.1	Conditions for acquisition	decision.conditions_for_acquisition	Dashboard
2.2	Deal-breaker register	decision.deal_breakers	Dashboard

ID	Evidence	Source	Workbook
2.3	Categorical decision tier	decision.escalation.categorical_tier	Dashboard
2.4	Economic decision tier	decision.escalation.economic_tier	Dashboard
2.5	Conditions escalation threshold	decision.escalation.conditions_threshold_pct	Dashboard
2.6	No-go escalation threshold	decision.escalation.no_go_threshold_pct	Dashboard
2.9	Per-pillar escalation rows	decision.escalation.pillars	Dashboard
2.12	Sensitivity sweep outcomes	financial.sensitivity	Dashboard
2.13	Condition deal-mechanic tags	decision.condition_details	Dashboard
3.1	Remediation line items	financial.line_items	Dashboard
3.2	Total remediation hours	financial.total_remediation_hours	Dashboard
3.3	Total remediation cost	financial.total_remediation_cost	Dashboard
3.4	Valuation discount applied	financial.valuation_discount_pct	Dashboard
3.5	Post-merge integration drag	financial.post_merge_integration_drag_pct	Dashboard
3.6	Hourly rate	financial.assumptions.hourly_rate	Dashboard
3.7	Complexity multiplier	financial.assumptions.complexity_multiplier	Dashboard
3.8	Valuation multiplier	financial.assumptions.valuation_multiplier	Dashboard
3.11	Person-years of remediation	financial.assumptions.person_years	Dashboard

ID	Evidence	Source	Workbook
3.12	Remediation cost CI	financial.assumptions.remediation_cost_confidence_interval	Dashboard
3.21	Sensitivity sweep details	financial.sensitivity	Dashboard
3.22	Financial lineage	financial.lineage	Dashboard
3.25	Risk valuation lines	financial.economic_layers.layer2_risk.lines	Dashboard
3.31	Scenario branches	financial.economic_layers.layer3_scenario.branches	Dashboard
3.34	Deal components	financial.economic_layers.layer4_deal.components	Dashboard
3.36	Double-counting ledger figures	financial.economic_layers.p0_double_counting.figures	Dashboard
4.1	License findings inventory	pillars.license.raw.findings	License Contamination
4.2	Per-package detected licenses	pillars.license.raw.findings	License Contamination
4.9	IP ownership metrics	pillars.ip_ownership.summary	IP Ownership
5.1	CVE count	pillars.cve.summary.cve_count	CVEs
5.6	CVE inventory	pillars.cve.raw.cves	CVEs
5.12	Secret candidate detail	pillars.secrets.raw.candidates	Secrets Candidates
5.15	IaC findings	pillars.iac.raw.findings	IaC Findings
5.21	Candidate endpoint detail	pillars.attack_surface.raw.candidate_entry_points	Attack Surface
5.26	Privacy findings	pillars.privacy.raw.findings	Privacy & Compliance
6.10	Direct dependency inventory	pillars.dependency.raw.direct_dependencies	Dependency Packages

ID	Evidence	Source	Workbook
7.15	Hotspot symbols	pillars.complexity.raw.complexity.symbols	Complex Symbols
7.22	Debt marker inventory	pillars.debt.raw.markers	Debt Markers
8.6	Test-deficit module inventory	pillars.test_deficit.raw.modules	Test Deficit
8.12	Coverage module detail	pillars.coverage.raw.modules	Coverage Gaps
9.6	Cycle inventory	pillars.entanglement.raw.cycles	Entanglement Violations
9.7	Cross-layer violation detail	pillars.entanglement.raw.cross_layer_violations	Entanglement Violations
9.14	Serialization risks	pillars.integration.raw.serialization_risks	Serialization Risks
10.5	Key-person risk detail	pillars.key_person.raw.risks	Key-person Risks
10.7	Module bus-factor detail	pillars.key_person.raw.modules	Key-person Risks
10.15	Knowledge loss vulnerability detail	pillars.retention.raw.risks	Knowledge Loss Vulnerability
11.12	Document inventory	pillars.doc_staleness.raw.documents	Stale Docs
11.18	CI files	pillars.ci_maturity.raw.ci_files	CI Signals
12.3	Framework hits	pillars.ai_ml.summary.framework_hits	AI/ML Models
12.5	SDK hits	pillars.ai_ml.summary.sdk_hits	AI/ML Models
13.6	Technology inventory	pillars.tech_stack.raw.technologies	Tech Stack
14.1	Consolidated top risks	decision.top_risks	Dashboard
14.3	Pillar classification register	decision.pillar_register	Dashboard
15.4	Integrity issues	integrity.issues	Contents

ID	Evidence	Source	Workbook
15.5	Unverified pillars	decision.confidence.inputs.unverified_pillars	Dashboard
O1	Reproduction fingerprint	financial.fingerprint	Dashboard

Rendered 65 of approximately 230 provenance rows. The full provenance table is in the source payload at /pillars and /financial paths.

APPENDIX Q

Glossary

This glossary defines, in plain language, every technical or specialist term appearing in any TDD table column heading. Terms are used consistently and never redefined per section, so a client and its advisers converge on one shared frame. Definitions are consistent with the contextual usage in this report.

Q.1 Glossary (Alphabetical)

Term	Meaning
Assessment State	Five-value trust vocabulary attached to a pillar: NOT_ASSESSED, PARTIALLY_ASSESSED, ASSESSED_NO_FINDINGS, ASSESSED_WITH_FINDINGS, ERROR. Distinguishes the kind of assessment from the legacy three-value status.
Assessment Status	Pillar-level coverage: assessed, partial, or not_assessed. Only assessed supports conclusions; partial or not_assessed are demoted to informational and labelled UNVERIFIED.
Attack Surface	The set of candidate external-reachable endpoints and entry points identified by static analysis. The candidate exposure score is an upper-bound signal for review prioritisation, not a confirmed vulnerability rate.
Attribution	The licence clause requiring preservation of copyright, patent, trademark, and attribution notices in derivative works. Found in Apache-2.0; triggers customer-disclosure obligations.
Blamed Line	A line of code attributed to a specific author by git blame. The external blamed line percentage measures the share of code authored outside the corporate domain.
Bus Factor	Reciprocal of sole-contributor concentration per module; a low bus factor flags flight risk. Calculated as the number of contributors whose departure would materially compromise a module's institutional knowledge.
Calibration Factor	Multiplier applied to the hours-per-unit estimate for a cost category. In the present scan, every category uses a factor of 1, meaning no external benchmark calibration has been applied (calibration.applied = false).

Term	Meaning
Candidate Endpoint	An entry point identified by the attack-surface scanner as potentially externally reachable. Candidate status does not confirm exploitability; it identifies review priority.
CI Maturity	The posture of the continuous-integration pipeline as measured by gate strictness, test execution in CI, and reported coverage. Low CI maturity propagates into high test-deficit and coverage findings.
Cognitive Complexity	A measure of how difficult a function is for a human to reason about, weighting nesting, recursion, and breaks in linear flow. Higher than cyclomatic for the same code.
Commit	A single atomic change recorded in the git history. The total commit count is the sum of all commits attributed to an author.
Condition Precedent	A deal mechanic that must be satisfied or proven before close, or the deal does not sign as structured. Distinct from a price-adjustment lever or warranty/representation.
Contamination Chain	A chain of dependencies linked by viral or restrictive licence propagation. The tqdm chain is the single contamination chain in the present scan.
Coverage	The static symbol-share proxy for value coverage: test-file symbols weighted by structural value. Not runtime or line/branch coverage. The effective insured value is a normalised signal derived from coverage gaps.
CVE	Common Vulnerabilities and Exposures: a published advisory identifier. The CVE pillar queries osv.dev for advisory matches against the resolvable package population.
Cyclomatic Complexity	A measure of the number of independent execution paths through a function. Higher values indicate more thorough testing requirements and higher defect risk.
Deal-Breaker	A critical-rated finding on a deal-breaker pillar (ai_ml, attack_surface, cve, iac, license, secrets) that forces the no-go verdict regardless of remediation economics.
Debt Density	The count of debt markers (fixme, hack, todo, workaround, suppression) per thousand lines of code. Higher density indicates a larger remediation backlog.
Debt Marker	An explicit in-comment annotation (fixme, hack, todo, workaround) or tooling suppression (noqa, pyright ignore, type ignore) flagged by the scanner. Counted by type and severity.
Dependency Pinning	The practice of constraining a dependency to a specific version (e.g., ==1.78.1) rather than a range (e.g., >=3.10). Pin ratio is the share of pinned dependencies; higher is better for reproducibility.
Double-Counting Ledger	An audit of the financial model that confirms shared-registry keys appear on multiple surfaces but are added only once. The ledger returns a clean verdict when no unjustified overlaps are present.
EOL	End-of-life: the technology or runtime is no longer supported by its maintainer. The tech_stack pillar counts offline repo lifecycle markers; the eol pillar verifies runtimes against endoflife.date; the two are reconciled into one union count.

Term	Meaning
Entanglement	The architectural coupling between modules as measured by circular-dependency cycles and cross-layer violations. The entanglement score is 0-1; the present scan returns 1.0 (maximum).
Escrow	A holdback of purchase price or a fund established to cover a specific post-close obligation. Sizing is deal counsel's call; this report does not assert escrow amounts.
Evidence State	The per-finding honesty label: Confirmed, Detected-not-verified, Inferred, Partially assessed, Unassessed, Conflicting, or Unsupported. Scanner signals are never auto-confirmed.
Finding ID	Stable identifier F-<pillar>-<rank> for one assessed finding. The register, narrative, workbook, and ledger all cite it; a body section cites the id rather than re-rendering the finding.
Framework	An AI/ML library detected in the source code (PyTorch, TensorFlow, JAX, ONNX, HuggingFace, pandas-ml, LLM Runtime). Counted by hits across the codebase.
Health Score	0-100 pillar score, demoted to 50 for an unfinished scan so that incomplete coverage never presents as a strong pillar. The ceilings are: critical 15, high 40, medium 65, low 85, informational unbounded.
Holdback	A portion of the purchase price retained by the buyer pending delivery of a post-close milestone. Sized by deal counsel after the integration plan is finalised.
HPU	Hours per unit: the remediation effort assigned to each finding in a cost category. Saturated by a per-category cap (above which the per-unit hours decay to a lower value).
IaC	Infrastructure as Code: the practice of defining infrastructure (Dockerfiles, Terraform, CloudFormation) in version-controlled text. The IaC pillar scans Dockerfiles and other IaC files for security and best-practice findings.
Integration Drag	The post-merger friction cost imposed by the technical posture of the asset. Capped at 80 percent; computed as $\text{hours} / (8 \times 260 \times \text{integration_drag_factor})$.
IP Ownership	The legal title to the source code, established by contributor licence agreements and IP-assignment agreements. The IP risk score (0-1) measures the share of external-authored code.
Knowledge Loss Vulnerability	The risk that institutional knowledge departs with a contributor. Computed from tenure, recency, blast radius, and retention risk; the score is 0-1 (higher is worse).
Lineage	The trace from a number in the report to the source artifact that produced it. Preserved in Appendix P for every key figure.
Lockfile	A file (e.g., poetry.lock, requirements.txt with hashes) that records the resolved versions of every direct and transitive dependency. Absent in the present scan, which propagates into unverified dependency, CVE, and EOL postures.
Maintenance Backlog	The aggregate score of neglected files, dependency staleness, and other forward-looking operating-cost signals. Not assessed in the present scan because no files were scanned by the asset-staleness scanner.
Major Risk	A pillar classification for high-rated or critical-rated non-deal-breaker pillars. Drives the conditions in Section 2.

Term	Meaning
Manifest	A file declaring direct dependencies (pyproject.toml, requirements.txt, Cargo.toml). Distinguished from a lockfile, which records resolved versions.
Materiality Floor	The absolute dollar amount that an escalation gate must also clear, in addition to the percentage threshold. The conditions materiality floor is \$250,000; the no-go materiality floor is not quantified.
Maturity	A measure of process discipline. CI Maturity combines gate strictness, test execution, and coverage reporting; low maturity propagates into high test-deficit and coverage findings.
Population	The evidence count each verdict rests on (files scanned, packages queried, modules in cycles). A low or missing population flags a coverage gap; a zero population never reads as clean.
Price-Adjustment Lever	A deal mechanic that moves value rather than blocking close. A funded or escrowed work item; sizing is deal counsel's call.
Proprietary Coupling	Coupling between modules measured by the integration scanner. The present scan returns 1,121 proprietary coupling findings, the dominant category of which is serialization risks.
Reachability	Whether an endpoint is externally reachable. Static analysis returns a high/medium/low confidence; the present scan could not establish reachability for 2,082 of 2,154 candidates.
Regression Risk	The risk that a code change introduces a defect in a previously-working code path. Measured per module by the test_deficit scanner; 120 modules are identified as high-regression-risk.
Remediation Cost	The dollar figure assigned to a remediation line item. Computed as saturated hours × hourly rate; bounded by the aggregate hours cap and the per-category saturation model.
Reproduction Fingerprint	SHA-256 hash over the configuration used to produce the report. Same inputs in, same outputs out; any change produces a different fingerprint.
Retention Risk	The risk that a contributor departs post-close, taking institutional knowledge with them. Computed from tenure, recency, and blast radius; rated critical, high, medium, or low.
Risk Level	critical / high / medium / low / informational. Demoted to informational under partial coverage; the derived_risk_level field preserves the scanner's own view.
Risk Signal	The weighted pillar risk blended with the financial model. The risk_signal input to the confidence score; the present scan returns 52.8 percent.
Risk-Reduction Value	The difference between expected loss before remediation and expected loss after remediation. The complexity risk line carries a risk-reduction value of \$287,500.
SBOM	Software Bill of Materials: a machine-readable inventory of every direct and transitive dependency, with versions. Required under emerging regulation; cannot be discharged from the repository alone in the absence of a lockfile.
Saturation Cap	The maximum count of findings to which the full hours-per-unit rate applies for a cost category. Above the cap, the per-unit hours decay to a lower value (the saturation decay).

Term	Meaning
Saturation Decay	The hours-per-unit rate applied to findings above the saturation cap. Always lower than the full HPU; reflects diminishing marginal remediation effort.
Schema Drift	The risk that the database schema evolves without version-controlled migrations. The present scan returns high schema drift risk, with zero migration files and zero migration tooling recorded.
SCC	Strongly Connected Components: the cycle set identified by the entanglement scanner. The SCC-ranked cycle set bounds how hard independent scaling would be.
SDK	Software Development Kit: a model-provider library (OpenAI, Anthropic, Cohere, Mistral, AWS Bedrock, Azure OpenAI). Counted by hits across the codebase.
Secret Candidate	A pattern match flagged by the secrets scanner as a potential secret. Candidates are not confirmed secrets; high-confidence and git-history-only counts are tracked separately.
Serialization Risk	A finding in the integration pillar flagging dangerous deserialization patterns (eval_call, dynamic_function) or custom binary protocol patterns. The present scan returns 1,121 such findings.
Severity	The scanner's rating of a finding's seriousness: informational, low, medium, high, or critical. Severity risk weights are 0, 1, 2, 3, 4 respectively.
SQALE	Software Quality Assessment based on Lifecycle Expectations: a standardised remediation-effort basis. The SQALE grade is A (best) to E (worst); the present scan returns E.
Taint Flow	A source-to-sink data flow identified by the taint scanner. The present scan identifies 6 flows; the taint pass is unbounded (not terminated by a depth or time limit).
Tech Stack	The base-layer inventory of technologies in use, with their lifecycle status. The tech_stack pillar counts offline repo lifecycle markers; the eol pillar verifies runtimes against endoflife.date.
Test Deficit	The shortfall of test coverage below a stated threshold, measured per module. The present scan identifies 545 of 791 modules below threshold.
Test Ratio	The ratio of test files to source files in a module. The overall test ratio is 37.9 percent; the threshold for adequate coverage is module-specific.
TSA	Transition Services Agreement: the contract under which the seller provides services to the buyer post-close. Sized by the integration-management office using the synergy model.
Unknown (License)	A license classification returned when the scanner cannot resolve a dependency's licence from manifest metadata. Ten packages carry UNKNOWN classifications in the present scan.
Unverified Pillar	A pillar whose underlying scan could not be completed against a resolvable population. Five pillars are unverified: cve, dependency, eol, maintenance, secrets.
Value Coverage Ratio	The static symbol-share proxy for value coverage. The present scan returns 27.4 percent, meaning 72.6 percent of the value-weighted production code is not insured by tests.

Term	Meaning
Viral Licence	A licence that requires derivative works to be licensed under the same terms. The MPL-2.0 component of the tqdm composite licence is viral.
Warranty / Representation	A deal mechanic in which the seller states a fact about the asset; breach is an indemnity question, not a work plan. Distinct from a condition precedent or price-adjustment lever.

APPENDIX R

Contradiction Ledger & Release Record

This appendix reproduces the contradiction ledger. Every material conflict is listed with its status, resolution, and decision impact. Unresolved entries discount the confidence stated in the executive brief; the present scan has one resolved entry and no unresolved entries.

R.1 Contradiction Ledger

C-DC-001 — resolved

Field	Value
Conflict	No double-counting findings across quantified economic figures
Subjects	Remediation, Risk valuation, Scenario analysis, Deal economics
Evidence	Anti-double-counting ledger
Status	resolved
Resolution	Shared-registry keys show each evidence once, on multiple surfaces, never added twice
Decision impact	none

R.2 Release Record

Field	Value
Report ID	TDD-20261016-073040
Generated	2026-10-16T07:56:52.637542+00:00
Report version	0.9.2
Ledger summary	1 resolved
Scan steps	6 recorded