M & A Technical Due Diligence

Technology Due Diligence Report

HashiCorp Terraform Acquisition Assessment

Target: HashiCorp
Repository: terraform
Report ID: TDD-20261016-060115
Generated: 2026-10-16
Proposed Price: USD 10,050,000
Verdict: ACQUIRE WITH CONDITIONS
Confidence: 76 / 100
Classification: CONFIDENTIAL

FOR THE NAMED BUYING PARTY AND ITS ADVISERS

Table of Contents

1. Executive Summary & Headline Findings....1

1. Executive Summary & Headline Findings....1

1.1 Introduction and Verdict....1

1.2 What the Evidence Does Not Support....2

1.3 Risk Register Summary....2

1.4 Deal-Thesis Alignment....2

1.5 Known, Unknown, and Required to Decide....3

2. Transaction Recommendation, Conditions & Escalation Thresholds....3

2.1 Decision Tier and Driver....4

2.2 Sensitivity and Verdict Invariance....4

2.3 Conditions for Acquisition....4

2.4 Per-Pillar Escalation....5

3. Remediation Cost Model, Valuation Impact & Sensitivity....5

3.1 Parameters and Calibration....5

3.2 Remediation Line Items....6

3.3 Confidence Interval and Sensitivity....6

3.4 Scenario Analysis and Deal Economics....7

3.5 Scenario Branches....8

3.6 Double-Counting Ledger....8

3.7 Risk Valuation Lines....8

4. Licensing & IP Exposure....9

4.1 Overview and Transaction Relevance....9

4.2 License Findings Inventory....9

4.3 Repo-Level GPL-2.0 Findings....10

4.4 IP Ownership Metrics....10

5. Security, Vulnerability & Attack Surface Posture....11

5.1 Security Posture Summary....11

5.2 CVE Inventory....11

© MTD Services

5.3 Secret Hygiene 12
5.4 Infrastructure-as-Code Findings 12
5.5 Attack Surface Composition 13
5.6 Privacy Findings 13
6. Supply Chain & Dependency Posture 14
6.1 Dependency Posture Summary 14
6.2 Lockfile Resolution 14
6.3 Direct Dependency Inventory 15
6.4 End-of-Life Posture 15
7. Code Quality, Complexity & Technical Debt 15
7.1 Complexity Hotspots 15
7.2 Duplication and SQALE Debt 16
7.3 Debt Markers 16
8. Testing, Coverage & Regression Risk 17
8.1 Test Deficit and Coverage Method 17
8.2 High-Value Coverage Gaps 17
8.3 Test-Deficit Module Inventory 18
9. Architecture, Entanglement & Integration Risk 18
9.1 Entanglement Posture 18
9.2 Cross-Layer Violations 19
9.3 Integration and Serialization Risks 19
10. Engineering Capacity, Bus Factor & Knowledge Loss Vulnerability 20
10.1 Key-Person Risk Posture 20
10.2 Knowledge Loss Vulnerability 20
10.3 Module Bus Factor 21
10.4 Git Team Signals 21
11. Maintenance, Documentation & Operational Maturity 21
11.1 Maintenance Posture (Unverified) 22
11.2 Documentation Inventory 22
11.3 CI/CD Maturity 22
12. AI/ML Footprint 23

© MTD Services

12.1 AI/ML Posture 23
13. Technology Stack & Lifecycle Posture 24
13.1 Stack and Lifecycle 24
14. Consolidated Risk Register & Top Risks 24
14.1 Top Risks 24
14.2 Major Risks Register 25
14.3 Pillar Classification Register 25
14.4 Deal-Breaker Register 26
15. Confidence, Scope Limitations & Material Unknowns 26
15.1 Integrity Status 26
15.2 Confidence Inputs 26
15.3 Out-of-Scope Evidence Register 27
16. Reading Notes & Honesty Conventions 27
16.1 Assessment Coverage 27
16.2 Evidence Honesty 27
16.3 Currency of Absence 28
16.4 Deal-Mechanic Vocabulary 28
16.5 Severity and Ranking 28
16.6 Known Limits of This Method 29
Appendix A — Methodology, Scope & Scan Configuration 29
A.1 Report Metadata 29
A.2 Scan Mode 29
A.3 Pillar Posture Register 29
A.4 Step Timings 30
Appendix B — Metric Definitions & Financial Calibration 30
Appendix O — Financial Reproduction Fingerprint 30
O1 — Reproduction Fingerprint 31
Appendix P — Evidence Provenance 31
P.1 — Executive Decision Provenance 31
P.2 — Transaction Recommendation Provenance 31
P.3 — Financial Model Provenance 31

© MD Services

P.4 — Pillar Evidence Provenance (Abbreviated)....31
Appendix R — Contradiction Ledger & Release Record....32
R.1 — C-DC-001 (Resolved)....32
R.2 — Release Record....32
Appendix C — Dependency & Lockfile Inventory....32
C.1 — Direct Dependency Inventory (Sampled)....32
C.2 — Lockfile Resolution....33
Appendix D — CVE Inventory....33
Appendix E — License Findings Inventory....33
Appendix F — Complexity, Duplication & SQALE Detail....33
F.1 — Complexity Hotspot Symbols (Top 30)....33
F.2 — SQALE Debt Modules (Sampled)....34
Appendix G — Debt Marker Inventory....34
Appendix H — Attack Surface Candidate Endpoints....34
Appendix I — Key-Person & Knowledge Loss Vulnerability Detail....34
I.1 — Key-Person Risk Rows (Top 20)....34
I.2 — Module Bus-Factor Rows (Top 20)....35
I.3 — Knowledge Loss Vulnerability (Critical Tier)....35
Appendix J — Maintenance, IaC & CI/CD Detail....35
J.1 — Maintenance Assets....35
J.2 — IaC Findings (Full)....35
J.3 — CI Files (Full)....35
J.4 — Coverage Artifacts....36
Appendix K — Privacy & Secrets Detail....36
K.1 — Privacy Findings (Full)....36
K.2 — Secret Candidates (Full)....36
Appendix L — AI/ML Inventory....36
Appendix M — Integrity & Data-Quality Issues....37
Appendix N — Taint Analysis Summary....37
Appendix Q — Glossary....37

© MTD Services

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

© MTD Services

1. Executive Summary & Headline Findings

1.1 Introduction and Verdict

This Technology Due Diligence (TDD) report assesses the technical posture of HashiCorp's Terraform codebase as an acquisition candidate. A TDD is a structured technical evaluation of a target's software assets, engineering organisation, security posture, and operational maturity, conducted to inform transaction pricing, deal conditions, and post-close integration planning. It complements financial and commercial due diligence by translating code-level evidence into transaction-relevant risk and value. The headline verdict, confidence rating, remediation economics, and verified-coverage extent in this section are each stated directly from the stored ground-truth payload, allowing the reader to converge on a shared evidence-anchored view before engaging the technical detail in later sections. Every value that can move a number in this report, including financial inputs, saturation calibration, decision thresholds, confidence weights, and schedule constants, is declared once in the parameters table at the head of Section 3 and cited rather than repeated elsewhere.

Verdict: ACQUIRE WITH CONDITIONS. The asset should be acquired only if the conditions enumerated in Section 2 are written into the transaction documents. The verdict is driven by the technical risk register summarised in Section 14, with confidence scored at 76 out of 100. The headline remediation economics are USD 293,125, with a stated 30 percent confidence interval ranging from USD 205,188 to USD 381,062, equivalent to 2.9 percent of the proposed USD 10,050,000 transaction price. The implied valuation discount is 16.7 percent and the post-merger integration drag is 80.0 percent. The remediation figure is a deterministic 100 percent adoption planning bound; the probability-weighted expected loss at Layer 2 is not separately quantified, so actual expected remediation lies at or below this bound.

The confidence score of 76 reflects a 90.9 percent assessment coverage across the 22 pillars in the model, discounted by two unverified domains (Debt and Maintenance) and by 158 rows of scanner-detected or inferred signals that require human validation before they can be relied on in negotiation. No finding in this report is independently confirmed; the 258 row-level signals captured by the scanners describe code-level observations that map to transaction-relevant risk, but they do not establish runtime behaviour, production incidents, or anything requiring access to systems, personnel, or contracts. The buyer must therefore treat the technical register as a directional starting position for commercial negotiation, not as a substitute for legal opinion, operational audit, or independent security testing.

1.2 What the Evidence Does Not Support

Two domains were not fully assessed and their absence from the headline findings must not be read as a clean bill of health. The Debt pillar returned a zero evidence population because no debt-marker annotations were detected in the codebase by the scanner; it is reported as NOT_ASSESSED rather than as a clean result. The Maintenance pillar returned a zero population because the asset-

© MTP Services

staleness scanner did not execute on any files. Absence of a finding in either domain is therefore an absence of coverage, not an absence of risk. The buyer should commission an independent review of these two pillars before close, or accept that they remain unverified representations rather than confirmed strengths.

All 258 row-level findings are scanner-detected or inferred signals. The distinction between a detected risk indicator and a confirmed exposure is material: a detected signal says the scanner found evidence consistent with a risk pattern, while a confirmed exposure says the risk has been independently validated by a human reviewer with access to runtime behaviour, production logs, or system personnel. The conditions in Section 2 are written for both cases. Condition precedents (CVE remediation, secret rotation, CI gating) address confirmed exposures that block close. Price-adjustment levers (complexity reduction, coverage uplift, decoupling, retention, test investment) address detected risks that move value rather than block close. The warranty layer (IP ownership) addresses the third category, where breach is an indemnity question rather than a work plan.

The scan itself is a static, offline-or-network-integrated source analysis. It does not establish runtime behaviour, production incident history, or anything requiring access to the seller's systems, personnel, or executed contracts. The Out-of-Scope Evidence Register in Section 15 names eight categories of question that a static repository scan cannot answer, each paired with the independent evidence the buyer must obtain. These gaps should shape the disclosure schedule, the indemnity-cap allocation, and the working-group escrow rather than be discovered post-close.

1.3 Risk Register Summary

The technical risk register in Section 14 is the single integrated view of every material finding, ranked by severity, with file-and-line evidence where applicable and ownership mapping to the relevant pillar. The top 10 risks span four critical complexity hotspots (ApplyComponentPlan, ApplyPlan, DirFromModule, plus the Daniel Banck retention risk), four critical retention risks (Brandon Croft, Austin Valle, Daniel Banck, Radek Simko), and three high-severity license exposures tied to repo-level GPL-2.0 files in the Terraform plugin protocol. The estimated cost column is meaningful only for DirFromModule (USD 135,600) and Daniel Banck (USD 30,800); the remaining critical rows are priced through their pillar-level escalation rows rather than as item-level line items, because the cost model sizes remediation at the category level rather than the finding level.

The Pillar Classification Register in Section 14.3 allocates each of the 22 assessed pillars into one of four classes: Critical Risk (none), Major Risk (10 pillars), Manageable Risk (8 pillars), and Informational Finding (4 pillars). Two pillars, Debt and Maintenance, are demoted to Unverified with a 50/100 health score because their assessment populations are zero, ensuring an unfinished scan never reads as a strong pillar. The buyer should focus negotiation attention on the Major Risk cluster, which contains the pillars that drive the headline remediation economics and the deal conditions: Complexity, Coverage, Test Deficit, Entanglement, IP Ownership, Retention, CI Maturity, Secrets, CVE, and IaC posture.

© MTD Services

1.4 Deal-Thesis Alignment

All six deal theses declared by the buyer were scored side by side from the same stored evidence; none is treated as primary and none carries a condition. The 1-to-10 alignment score rates how strongly and how completely the stored evidence bears on each thesis within its evidence-state cap; it is not a business-fit score and does not measure commercial attractiveness. A score of 10/10 means the evidence resolves the technical question fully within its coverage cap, not that the thesis is validated as a business strategy. A score of refused means the thesis is not supported by static analysis and requires a different evidence stream.

Table 1.1 — Deal-thesis alignment against stored evidence

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	10/10 (5 of 5 metrics resolved)	Detected, not verified	Interface boundaries inferred by pattern detection over source text. They describe where the seams are, not whether two products can be sold together.
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 size; 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	10/10 (6 of 6 metrics resolved)	Detected, not verified	Exposures evidenced are limited to open-source licensing, credential hygiene, and privacy-sensitive logging. Not a compliance opinion.
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, compensation, and notice periods are outside repository scope.

The pattern across the six theses is consistent: the scanner resolves the technical question that bears on each thesis, but the resolution is bounded by what static source analysis can see. Scale, integration, and maintenance economics are bounded by structural coupling and SQALE estimates. The IP Moat thesis is refused because static analysis cannot distinguish proprietary algorithmic density from standard framework boilerplate, and Regulatory is limited to the specific exposures this report evidences rather than a general compliance opinion. The Talent thesis is bounded by git history: the scanner can describe who has touched the code, but cannot forecast who will remain after close. Each 10/10 alignment therefore reads as the best available technical answer within its evidence cap, not as a business verdict.

1.5 Known, Unknown, and Required to Decide

What this report establishes. Twenty pillars were assessed with usable signal: AI/ML, Attack Surface, CI Maturity, CVE, Complexity, Coverage, Dependency, Doc Staleness, EOL,

Entanglement, IP Ownership, IaC, Integration, Key Person, License, Privacy, Retention, Secrets, Tech Stack, and Test Deficit. Across these pillars the report classifies 0 findings as Confirmed, 158 as Detected (not verified), 100 as Inferred, 0 as Partially assessed, 0 as Unassessed at the finding level, 0 as Conflicting, and 0 as Unsupported. Every Confirmed cell is empty by design: the scanner does not auto-confirm, and a buyer must rely on independent validation to convert a Detected signal into a Confirmed exposure.

Coverage gaps the buyer must note. The Debt and Maintenance pillars are not yet established. Their absence is not a clean result; it is an unanswered question. The buyer must either commission a targeted follow-up scan before close or accept that any warranty covering these areas is unanchored to evidence.

What the buyer must still verify. Nine conditions for acquisition remain open and are listed in Section 2. No unresolved contradictions remain in the ledger (Appendix R). The contradiction register contains a single resolved entry (C-DC-001), which documents that the double-counting ledger is clean across all quantified economic figures.

2. Transaction Recommendation, Conditions & Escalation Thresholds

2.1 Decision Tier and Driver

The categorical decision tier and the economic decision tier both resolve to ACQUIRE_WITH_CONDITIONS. The categorical tier is the board-level recommendation driven by the severity and count of pillar risks; the economic tier is the parallel decision driven by remediation cost as a percentage of the proposed price. The two tiers reach the same verdict in this scan, which strengthens the negotiating position: a buyer can defend the conditions both on the technical risk register and on the remediation economics without contradiction. The driver in both cases is the technical register, not the financial model, which means the conditions are anchored to evidence rather than to a budget envelope.

The economic escalation gates are configured at 5 percent of the proposed price for conditions materiality (with a USD 250,000 absolute floor) and 25 percent for the no-go threshold. Remediation at 2.9 percent of price sits below the 5 percent conditions gate on the percentage measure but clears the USD 250,000 absolute floor (USD 293,125 against USD 250,000), which is why the conditions tier still resolves to ACQUIRE_WITH_CONDITIONS rather than ACQUIRE. The economic gate is not disputed: the seller's quality-of-earnings provider would need to challenge either the line-item hours-per-unit calibration or the saturation caps to move the headline remediation figure, and both are anchored to a documented financial fingerprint (Appendix O).

2.2 Sensitivity and Verdict Invariance

A 17-row sensitivity sweep stress-tests the headline remediation figure against multipliers of 0.7, 1.0, and 1.3 on five parameters: complexity multiplier, discount rate, hourly rate, maximum

© MTD Services

aggregate hours, and valuation multiplier. The categorical decision tier is invariant across every parameter: each scenario returns ACQUIRE_WITH_CONDITIONS. The confidence score is also invariant at 76, because the confidence policy is anchored to coverage and risk signal, which the sweep does not perturb. Only the total remediation cost moves, and only along two parameters: the complexity multiplier (USD 279,625 to USD 306,625) and the hourly rate (USD 205,188 to USD 381,062). The discount rate, aggregate cap, and valuation multiplier leave the remediation cost unchanged because they shape downstream valuation calculations rather than the line-item cost.

The verdict invariance has two implications for the deal team. First, the recommendation survives challenge on the most contestable parameter (the hourly rate); even at 0.7 times the USD 50 default, the conditions tier remains ACQUIRE_WITH_CONDITIONS, and even at 1.3 times it does not approach the 25 percent no-go threshold. Second, the buyer can present the conditions as robustly evidenced rather than as a price negotiation lever; the seller would need to challenge either the underlying scanner evidence or the financial fingerprint itself to move the verdict, which is a high evidentiary bar.

2.3 Conditions for Acquisition

The nine conditions for acquisition resolve into three distinct deal mechanics, each of which a buyer negotiates in a different place in the acquisition agreement. The three condition precedents must be satisfied or proven before close, or the deal does not sign as structured. The five price-adjustment levers are funded or escrowed work items that move value rather than blocking close. The single warranty/representation is a seller statement whose breach is an indemnity question rather than a work plan. Quantification and sizing of each condition are for deal counsel; no escrow amount, holdback, warranty cap, or indemnity ceiling is asserted in this report. The conditions are listed verbatim below, annotated with the deal mechanic declared at the decision layer.

Table 2.1 — Conditions for acquisition and deal-mechanic tags

#	Condition	Pillar	Deal Mechanic
1	Require an automated CI gate as a condition precedent	CI Maturity	Condition precedent
2	Include a complexity-reduction milestone in the integration plan	Complexity	Price-adjustment lever
3	Address coverage findings as a condition precedent	Coverage	Price-adjustment lever
4	Remediate reachable known CVEs before close	CVE	Condition precedent
5	Plan decoupling for circular dependency cycles	Entanglement	Price-adjustment lever
6	Warrant assignability of third-party contributions	IP Ownership	Warranty / representation
7	Secure retention agreements for sole-maintainer modules	Retention	Price-adjustment lever
8	Rotate all leaked credentials and stand up credential rotation	Secrets	Condition precedent
9	Raise regression coverage on high-risk modules post-close	Test Deficit	Price-adjustment lever

The split between condition precedents and price-adjustment levers reflects the underlying risk type rather than its severity. The three precedents (CI gate, CVE remediation, secret rotation) address

© M&D Services

exposures that are binary in nature: either the leaked credentials are rotated before close or they are not; either the known CVEs are remediated or they are not; either an automated CI gate exists or it does not. These are unsuitable for escrow because the buyer cannot accept residual exposure on day one. The five price-adjustment levers address risks that are scalar and progressive: complexity can be reduced on a milestone schedule, coverage can be uplifted over time, retention agreements can be phased, and decoupling can be sequenced against the integration roadmap. The single warranty on IP assignability is a legal representation rather than a work plan, because the underlying risk (third-party contributor assignments) is a title question that cannot be resolved by engineering work post-close.

2.4 Per-Pillar Escalation

Each pillar-level escalation row measures its cost against the 5 percent conditions gate and the 25 percent no-go threshold on its own. The table below shows that no pillar breaches either gate independently, which is why the deal-breaker register (Section 14.4) is empty. The largest single pillar exposure is Complexity at USD 135,600 (1.4 percent of price), followed by Retention at USD 30,800 (0.3 percent). The aggregate remediation cost of USD 293,125 (2.9 percent) is the deterministic sum of the line items and is the figure the sensitivity sweep stress-tests; it is well below the 5 percent gate and far below the 25 percent no-go threshold.

Table 2.2 — Per-pillar escalation rows

Pillar	Category	Cost (USD)	% of EV	Breach
Complexity	Critical complexity remediation	$135,600	1.4%	None
Retention	Knowledge transfer	$30,800	0.3%	None
IP Ownership	IP reassignment	$12,800	0.1%	None
Dependency	Dependency pinning	$10,725	0.1%	None
Test Deficit	Test investment	$5,600	0.1%	None
Entanglement	Decoupling	$1,000	0.0%	None
CVE	CVE remediation	$0	0.0%	None
Debt	General debt	$0	0.0%	None
Maintenance	Staleness remediation	$0	0.0%	None
Attack Surface	Taint remediation	$0	0.0%	None

Four pillars (CVE, Debt, Maintenance, Attack Surface) show USD 0 cost in the table, but each zero has a different reason that the buyer must understand. The CVE cost is zero because all 18 advisories are no-fix-available, meaning remediation is structural rather than version-pinning: the seller must either patch the upstream dependencies or accept the residual exposure, neither of which the model prices as a line item. The Debt and Maintenance costs are zero because those pillars are NOT_ASSESSED, not because the risk is absent. The Attack Surface cost is zero because taint analysis found zero flows (Appendix N), which is a clean result on the taint dimension specifically and not a clean bill of health on the broader attack surface.

© Meridian Services

3. Remediation Cost Model, Valuation Impact & Sensitivity

3.1 Parameters and Calibration

Every value that can move a number in this report is declared in one parameters table: financial inputs, saturation calibration, valuation and integration-drag curves, confidence weights, decision thresholds, post-close schedule constants, and health-posture ceilings. The table below is the single source of truth for the financial model; downstream sections cite the parameters rather than re-stating them. Where a parameter was not configured by the buyer or is not carried in the stored payload, it is listed as not quantified rather than assumed away. A separate calibration-applied flag tracks whether the hours-per-unit values are benchmarked (calibration.applied = false in this scan), which is why the remediation figure should be treated as a planning bound rather than a priced contractor quote.

Table 3.1 — Financial and decision parameters

Group	Parameter	Value	Affects
Remediation inputs	Hourly rate	$50	Every line-item cost = saturated hours x rate
Remediation inputs	Complexity multiplier	1.5	Hours per unit for critical-complexity hotspots
Remediation inputs	Valuation multiplier	3	Amplifies remediation cost into valuation discount
Remediation inputs	Integration drag factor	0.15%	Post-merge friction = hours / (8 x 260 x factor)
Remediation inputs	Productive hours per quarter	624 h	Net quarterly working hours behind per-cause effort
Remediation inputs	Confidence interval width	30%	Width of 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)
Aggregate hours cap	Max total hours	20,000 h	Aggregate ceiling; surplus above decays at 0.25
Valuation curves	Valuation discount cap	25%	Maximum valuation discount on proposed price
Valuation curves	Integration drag cap	80%	Maximum post-merge integration drag
Decision & escalation	Conditions threshold	5%	Remediation above this share triggers conditions
Decision & escalation	No-go threshold	25%	Remediation above this share triggers no-go
Decision & escalation	Conditions materiality floor	$250,000	Absolute dollars the conditions gate must also clear
Decision & escalation	Deal-breaker pillars	ai_ml, attack_surface, cve, iac, license, secrets	Critical rating on any forces no-go

© MTC Services

Group	Parameter	Value	Affects
Schedule	Execution teams	2	Two-engineer teams for post-close remediation
Schedule	Workdays per sprint	10	Sprint capacity = teams x 2 x workdays
Schedule	Sprint cap	12	Hours outside this window are unplanned
Schedule	Hours per workday	4	Roadmap durations = effort-hours / 4

The model is intentionally deterministic and fingerprinted (Appendix O): the same inputs in the same configuration always produce the same outputs, and any change to a number-driving input produces a different fingerprint and therefore a different report. This is the property that allows the report to be reproduced and challenged by an opposing adviser. The buyer should expect the seller's quality-of-earnings reviewer to focus on three parameters in particular: the complexity multiplier (1.5x, which amplifies the hours-per-unit for the 503 critical-complexity symbols), the hourly rate (USD 50, which is a planning-rate default rather than a contractor quote), and the saturation caps (which limit the hours credited per cost category regardless of unit count). The sensitivity sweep in Section 2 stress-tests the first two and shows verdict invariance across the full multiplier range.

3.2 Remediation Line Items

The remediation cost of USD 293,125 is the deterministic sum of seven line items shown below. Each line item is sized by a count of units, an hours-per-unit (HPU) value, a saturation cap that limits total hours regardless of unit count, and a decay rate that applies once the cap is reached. The HPU values are tool defaults rather than benchmarked rates; the calibration.applied flag is false, which is why the figures should be treated as planning bounds. The two largest line items, critical complexity (USD 135,600) and high complexity (USD 96,600), together account for 79 percent of the total remediation cost, which means the buyer's negotiation attention should focus on the complexity pillar rather than on the smaller line items.

Table 3.2 — Remediation line items

Category	Count	Cap	Decay HPU	Hours per Unit	Total Hours	Total Cost	Notes
Critical complexity	503	50	4	5.39	2,712 h	$135,600	Critically complex functions requiring refactoring
High complexity	666	150	2	2.90	1,932 h	$96,600	High-complexity functions (exclusive of critical tier)
Knowledge transfer	124	30	4	4.97	616 h	$30,800	Critical sole-maintainer modules requiring transfer
IP reassignment	78	50	2	3.28	256 h	$12,800	Files with external-dominant authorship requiring IP review
Dependency pinning	329	100	0.5	0.65	214 h	$10,725	Unpinned dependencies requiring version locking
Test investment	7	20	4	16.00	112 h	$5,600	High-regression-risk modules requiring test investment

© VTD Services

Category	Count	Cap	Decay HPU	Hours per Unit	Total Hours	Total Cost	Notes
Decoupling	1	10	8	20.00	20 h	$1,000	Circular dependency cycles requiring decoupling
Total	1,630	—	—	—	5,862 h	$293,125	Sum of seven line items above

The total of 5,862 remediation hours is equivalent to 2.8 person-years (using the 2,080 hours annual constant). At the model's productive-hours-per-quarter assumption of 624, that is approximately 9.4 quarter-quarters of single-engineer effort, which the post-close schedule (2 teams of 2 engineers, 10 workdays per sprint, 12-sprint cap) wraps into a planned window of approximately 12 weeks for the planned portion, with any overflow reported as unplanned. The aggregate hours cap of 20,000 hours is not applied in this scan because total hours (5,862) sit well below the ceiling; the cap matters only when the line items would otherwise aggregate to an implausibly large figure, which is not the case here.

3.3 Confidence Interval and Sensitivity

The remediation-cost confidence interval is plus or minus 30 percent on the hours-per-unit estimates, producing a low of USD 205,188 and a high of USD 381,062 around the central figure of USD 293,125. The interval is a stated-confidence range from the model calibrations, not a stochastic bound; it does not represent a probability distribution of outcomes but rather the model's sensitivity to its own HPU estimates. The interval is constructed symmetrically around the central figure, which is appropriate when the dominant uncertainty is the per-unit hours (a calibration parameter) rather than the count of units (a scanner-measured value).

Table 3.3 — Remediation cost confidence interval

Metric	Value
Basis	Plus or minus 30 percent on hours-per-unit estimates
Low (USD)	$205,188
High (USD)	$381,062
Central (USD)	$293,125
Width (percent)	30

The sensitivity sweep confirms that the verdict is invariant across the full sweep. The complexity multiplier, hourly rate, and valuation multiplier are the three parameters that move at least one output: the complexity multiplier moves both hours and cost; the hourly rate moves cost; the valuation multiplier moves the valuation discount percentage but not the remediation cost. The discount rate, max total hours, and aggregate cap do not move the headline remediation figure because they either shape downstream valuation mathematics (discount rate) or sit above the remediation magnitude (aggregate cap). The remediation_timing and scenario_weights rows are not quantified because those parameters were not configured for this scan; they are listed rather than assumed away, which is the convention applied throughout the report.

3.4 Scenario Analysis and Deal Economics

© M&D Services

The deal-economics layer (Layer 4) projects the headline enterprise value (USD 10,050,000), the net quantified technology impact (USD 8,033,125), and the resulting adjusted economic value (USD 7,537,500). The price adjustment of USD 2,512,500 is the headline economic consequence of the technical findings; it is capped at the 25 percent valuation-adjustment cap (USD 2,512,500), which means the adjustment is at the cap and would be larger absent the cap. The cap is a model safeguard rather than a negotiated position; the buyer should treat the uncapped figure as the technical read and the capped figure as the maximum adjustment the model will assert.

Table 3.4 — Deal economics headline valuation reconciliation

Headline EV	Net Quantified Tech Impact	Adjusted Economic Value	Price Adjustment	Cap %	Cap Amount	Capped
$10,050,000	$8,033,125	$7,537,500	$2,512,500	25.0%	$2,512,500	Yes

The deal-economics headline combines four economic components: the remediation cost (USD 293,125, the Layer 1 sum), the residual expected loss (USD 50,000, the Layer 2 figure that survives after remediation), the integration-separation cost (USD 8,040,000, the Layer 1 integration-drag figure), and the other-evidence-supported impact (USD 150,000, a config-supplied cyber-incident allowance). The potential synergy of USD 500,000 is the only positive figure in the components table and represents the buyer's stated customization burden saving; it is a config-supplied input rather than a scanner-derived estimate. Each component carries its own stored amount, basis, confidence, and calculation chain in the workbook, so the buyer can re-derive any figure rather than trust the headline arithmetic.

Table 3.5 — Deal components (never collapsed)

Component Type	Impact Type	Value (USD)	Source Key
Headline EV	Revenue disruption	$10,050,000	transaction.headline_ev
Integration/separation TSA	Integration-separation cost	$8,040,000	layer1.integration_drag_pct
Net quantified tech impact	Remediation cost	$8,033,125	layer4.net_quantified_tech_impact
Other quantified tech exposure	Security incident	$150,000	config.other_evidence_supported_impacts[0]
Potential synergy	Customization burden	$500,000	config.synergy_value
Remediation cost	Remediation cost	$293,125	layer1.total_cost
Residual expected loss	Operational downtime	$50,000	layer2.aggregate_residual_expected_loss

3.5 Scenario Branches

Four scenario branches are enabled and modelled: acquire-and-remediate post-close, material failure, partial remediation, and remediate before close. Each branch's total impact is composed of transaction EV, Layer 1 remediation, integration drag, residual expected loss, and (where applicable) transaction impact. The weighted expectation across scenarios is USD 3,899,621, which is reported under the honest label Scenario-Based Technology VaR. The label is deliberately not a market-risk

© M&A Services

measure: the figure is a technology-risk model output anchored to documented scan evidence, not a value-at-risk figure derived from market data.

Table 3.6 — Scenario branches

Scenario	Title	Enabled	Evidence Basis	Total Impact (USD)
acquire_remediate_post_close	Acquire and remediate post-close	True	Transaction EV + Layer 1 + integration drag	$7,570,352
material_failure	Material failure / downside case	True	Analyst calibration on documented scan evidence	$90,194
partial_remediation	Partial remediation	True	Config calibration (analyst)	$183,493
remediate_before_close	Remediate before close	True	Layer 1 schedule + Layer 2 residual	$320,311

The material-failure scenario at USD 90,194 is the smallest modelled impact and represents the downside case calibrated by the analyst on documented scan evidence; it is not a worst-case figure but a calibrated expectation under a defined failure mode. The remediate-before-close scenario at USD 320,311 is the most conservative of the four because it carries a transaction impact of USD 1,675,335 (the cost of delayed close in the buyer's model) on top of the remediation and residual layers. The acquire-and-remediate post-close scenario at USD 7,570,352 is the largest because it carries the full integration drag of USD 8,040,000 less the remediation and residual offsets; it is the model's representation of the buyer's expected transaction impact if no remediation occurs before close.

3.6 Double-Counting Ledger

The double-counting ledger verifies that no quantified economic figure is counted twice across layers. The clean verdict is true, and the three justified overlapping keys are layer1.integration_drag_pct, layer1.total_cost, and layer2.aggregate_residual_expected_loss. Each of these keys legitimately appears on multiple surfaces (Layer 1, Layer 2, scenario components, and deal components) but is added only once at each surface. The contradiction ledger (Appendix R) records this verification as entry C-DC-001 with status resolved and decision impact none, which is why the headline confidence score is not discounted by double-counting concerns.

3.7 Risk Valuation Lines

The risk valuation lines in the Layer 2 risk component table pair each material risk with its expected loss before and after remediation. Only one line, DirFromModule (the critical-complexity hotspot in internal/initwd/from_module.go), is quantified in this scan; the remaining eight lines are NOT_QUANTIFIED because the financial.risk_parameters block was not configured. The default is unquantified rather than zero: a value cell that reads not quantified is deliberately unquantified (assessment state NOT_QUANTIFIED), never a silent zero. The DirFromModule line carries a pre-remediation expected loss of USD 337,500 (45 percent probability x USD 750,000 impact) and a

© MD Services

post-remediation residual of USD 50,000 (20 percent probability x USD 250,000 impact), producing a risk-reduction value of USD 287,500 against a remediation cost of USD 135,600.

Table 3.7 — Quantified risk valuation (DirFromModule)

Pillar	Title	State	Prob. Before	Impact Before	Expected Loss Before	Prob. After	Impact After	Expected Loss After	Residual	Reduction Value
Complexity	DirFromModule	Assessed	45.0 %	$750,000	$337,500	20.0 %	$250,000	$50,000	$50,000	$287,500

The risk-reduction value of USD 287,500 against the remediation cost of USD 135,600 yields a ratio of approximately 2.1x, which means the buyer receives USD 2.10 of risk reduction for every USD 1.00 of remediation spend on the DirFromModule hotspot alone. This is the strongest single economic argument for the price-adjustment lever on the complexity pillar: the buyer can present the remediation as a value-creating intervention rather than a cost-only obligation, with the same scanner-derived evidence that anchors the rest of the report. The remaining eight risk valuation lines are not quantified in this scan and the buyer can either commission a follow-up quantification or accept the residual uncertainty.

4. Licensing & IP Exposure

4.1 Overview and Transaction Relevance

Licensing and IP exposure is the second pillar that touches the warranty layer of the acquisition agreement, alongside the IP Ownership findings in Section 10. The license posture of the packaged components that ship in the target determines whether the acquirer can integrate the asset into a closed-source product, whether customer disclosure obligations are triggered, and whether carve-outs or re-licensing work is required. IP ownership attribution drives title and warranty risk and shapes the seller's ability to assign contributions at close. The findings in this section are stored ground truth: per-package detected licenses, license-clause findings, and IP-ownership attribution metrics are reproduced from the scanner payload without modification.

Three findings dominate the license pillar. First, three repo-level files (docs/plugin-protocol/LICENSE, internal/tfplugin5/LICENSE, and internal/tfplugin6/LICENSE) carry GPL-2.0 with both viral and restrictive flags at 0.8 confidence. These files describe the protocol that Terraform uses to communicate with its plugins; their presence in the repository does not by itself contaminate the broader codebase, but it does require a representation and warranty covering license compliance for the shipped product. Second, 867 of 870 license findings are tagged UNKNOWN, meaning the scanner could not positively identify the license from the manifest or package metadata. This is a coverage gap rather than a confirmed exposure: the buyer should commission an independent SBOM with license resolution before close, or accept that the UNKNOWN population may contain copyleft or restrictive licenses that have not been surfaced.

Third, the IP Ownership metrics show an external-blamed-line percentage of 77.03 percent and an external author count of 1,930 of 2,375 total authors, which together produce an IP risk score of 0.6497. These metrics are a diligence signal rather than legal proof of ownership: a high external-

©MRD Services

blamed-line percentage indicates that a substantial share of the codebase was authored by contributors external to the corporate domain (in this case users.noreply.github.com, the noreply domain used by GitHub for commit attribution). The material implication is that the buyer should require the seller to warrant assignability of third-party contributions and to disclose the contributor licence agreements or equivalent instruments that govern those contributions.

4.2 License Findings Inventory

The license findings inventory contains 870 entries. Three are repo-level GPL-2.0 files with viral and restrictive flags; the remaining 867 are UNKNOWN licenses from go.mod files across the 11 manifests detected in the repository. The viral license count is 3 (the three repo-level GPL-2.0 files), the restrictive license count is 3 (the same three files, because GPL-2.0 is both viral and restrictive), the unknown license count is 867, and the contamination chains count is 0. The zero contamination chain count is a positive signal: the scanner did not detect any chain by which a viral license in a transitive dependency would impose its terms on the packaged Terraform binary. The buyer should nevertheless commission an independent license scan to verify the zero contamination finding, because license scanning is sensitive to manifest format and the scanner may not have parsed every manifest in the repository.

Table 4.1 — License summary metrics

Metric	Value
License findings inventory	870 findings
Viral license count	3
Restrictive license count	3
Unknown license count	867
Licensed packages total	867
Contamination chains	0
IP ownership attribution rows	878 attributions

4.3 Repo-Level GPL-2.0 Findings

The three repo-level GPL-2.0 files are reproduced below with their detected license, viral and restrictive flags, confidence, and source file paths. The 0.8 confidence score reflects the scanner's pattern-matching certainty on the license text; it is not a legal opinion on the licence's applicability to the surrounding code. The buyer's counsel should review the three files in their original form before close, paying particular attention to whether the GPL-2.0 text applies to the file alone or to a broader module.

Table 4.2 — Per-package detected licenses (repo-level, sampled)

Package	Detected License	Is Viral	Is Restrictive	Confidence	Source File
(repo-level)	GPL-2.0	True	True	0.8	docs/plugin-protocol/LICENSE

© MD Services

Package	Detected License	Is Viral	Is Restrictive	Confidence	Source File
(repo-level)	GPL-2.0	True	True	0.8	internal/tfplugin5/LICENSE
(repo-level)	GPL-2.0	True	True	0.8	internal/tfplugin6/LICENSE
cel.dev/expr	UNKNOWN	False	False	0.0	go.mod
cloud.google.com/go	UNKNOWN	False	False	0.0	go.mod
cloud.google.com/go/auth	UNKNOWN	False	False	0.0	go.mod
cloud.google.com/go/compute/metadata	UNKNOWN	False	False	0.0	go.mod
cloud.google.com/go/iam	UNKNOWN	False	False	0.0	go.mod
cloud.google.com/go/kms	UNKNOWN	False	False	0.0	internal/backend/remote-state/gcs/go.mod
cloud.google.com/go/storage	UNKNOWN	False	False	0.0	go.mod

The license clause findings identify six clauses across the three repo-level files: three patent_grant clauses at permissive severity and three copyleft clauses at viral severity. The matched-packages count for each clause is zero, which means the scanner did not detect any packages whose license terms flow from the clause. This is again a positive signal that does not eliminate the underlying exposure: the GPL-2.0 text exists in the repository, and any buyer that integrates Terraform into a closed-source product must obtain a representation and warranty covering license compliance, with an indemnity for any copyleft exposure.

4.4 IP Ownership Metrics

The IP ownership metrics summarise the attribution of 137 files across 2,375 authors. The corporate domain is users.noreply.github.com, which is the noreply domain GitHub uses for commit attribution when a contributor does not provide a verified email. The use of a noreply domain as the corporate anchor is itself notable: it suggests that a substantial share of HashiCorp's own contributors may be committing with personal email addresses, which complicates the assignability question. The external author count of 1,930 (81.3 percent of total authors) and external-blamed-line percentage of 77.03 percent indicate that the codebase has historically accepted a high volume of external contributions, which is consistent with Terraform's open-source provenance but which requires the buyer to verify the contributor instruments that govern those contributions.

Table 4.3 — IP ownership metrics

Metric	Value
Attributed file count	137
Corporate domain	users.noreply.github.com
External author count	1,930
External blamed-line pct	0.7703 (77.03%)

© MTD Services

Metric	Value
External commit pct	0.7703 (77.03%)
External domain count	835
External owned files	78
High-risk files	78 entries
IP risk score	0.6497
Total authors	2,375
Total files (attributed)	137

The 78 high-risk files identified by the scanner are those with external-dominant authorship, which the cost model prices as the IP reassignment line item (USD 12,800 across 78 files, at 4 HPU per file with a saturation cap of 50 and decay of 2). The price is small in absolute terms (0.1 percent of the proposed transaction price) but the work is mandatory rather than optional: the buyer cannot close without confirming that the external contributions in these 78 files are assignable. The recommendation is to require the seller to warrant assignability of third-party contributions as a single warranty/representation condition (condition 6 in Table 2.1) rather than to attempt individual re-licensing of each file, which would be operationally infeasible at the scale of 78 files distributed across the codebase.

5. Security, Vulnerability & Attack Surface Posture

5.1 Security Posture Summary

The security pillar aggregates five sub-pillars into a single posture view: CVE exposure, secret hygiene, infrastructure-as-code findings, attack-surface composition, and privacy-relevant data flows. The aggregate posture is mixed. On the positive side, zero critical CVEs were detected across the 424 packages queried against osv.dev, and zero live critical secrets were detected across the 2,056 files scanned by the secret scanner. On the negative side, 18 advisories are open with no fix available, 12 secret candidates were detected with 3 high-confidence matches and 8 history-only matches, 2 IaC findings flag container-root execution, 59 candidate entry points were identified with 24 untested, and 6 privacy findings include 1 plaintext credential and 5 sensitive logging statements.

The mixed posture translates into three condition precedents in the acquisition conditions (CI gate, CVE remediation, secret rotation) and several implicit warranty items (privacy representation covering personal-data handling scoped to the credential inventory, infrastructure-surface review gate). The buyer's cyber-insurance sizing should reflect the 18 open advisories and the 12 secret candidates; the regulatory disclosure timing should reflect the privacy findings (which include sensitive logging of tokens, secrets, and apikeys in the Azure, Kubernetes, OCI, and S3 backends); and the cyber-warranty language should address both the open advisories and the credential exposure.

5.2 CVE Inventory

© MTD Services

The CVE inventory contains 18 advisories, all at UNKNOWN severity (the OSV source does not assign a CVSS score for these advisories) and all with no fixed release available. The 18 advisories span 7 distinct packages: gopkg.in/yaml.v2 (6 advisories), github.com/sirupsen/logrus (2), github.com/cli/go-gh (2), github.com/hashicorp/terraform (2), github.com/go-jose/go-jose (1), github.com/golang-jwt/jwt (1), github.com/go-viper/mapstructure (2), github.com/yuin/goldmark (1), and golang.org/x/crypto (1). The absence of a fixed release for any of the 18 advisories is the most material observation: the seller cannot resolve these by version-pinning alone, because no fixed version exists in the upstream advisory.

Table 5.1 — CVE inventory (18 advisories)

ID	Package	Ecosystem	Version	Fixed Available	Remediation Class
GHSA-4f99-4q7p-p3gh	github.com/sirupsen/logrus	go	v1.2.0	False	no_fixed_release
GHSA-55v3-xh23-96gh	github.com/cli/go-gh	go	v2.13.0	False	no_fixed_release
GHSA-6q6q-88xp-6f2r	gopkg.in/yaml.v2	go	v2.2.1	False	no_fixed_release
GHSA-h3p9-wrgx-82cm	github.com/hashicorp/terraform	go	0.0.0	False	no_fixed_release
GHSA-r88r-gmrh-7j83	gopkg.in/yaml.v2	go	v2.2.1	False	no_fixed_release
GHSA-wxc4-f4m6-wwqv	gopkg.in/yaml.v2	go	v2.2.1	False	no_fixed_release
GO-2020-0036	gopkg.in/yaml.v2	go	v2.2.1	False	no_fixed_release
GO-2021-0061	gopkg.in/yaml.v2	go	v2.2.1	False	no_fixed_release
GO-2022-0839	github.com/hashicorp/terraform	go	0.0.0	False	no_fixed_release
GO-2022-0956	gopkg.in/yaml.v2	go	v2.2.1	False	no_fixed_release
GO-2024-3295	github.com/cli/go-gh	go	v2.13.0	False	no_fixed_release
GO-2025-3485	github.com/go-jose/go-jose	go	v4.1.4	False	no_fixed_release
GO-2025-3553	github.com/golang-jwt/jwt	go	v4.5.2	False	no_fixed_release
GO-2025-3787	github.com/go-viper/mapstructure	go	v2.4.0	False	no_fixed_release
GO-2025-3900	github.com/go-viper/mapstructure	go	v2.4.0	False	no_fixed_release
GO-2025-4188	github.com/sirupsen/logrus	go	v1.2.0	False	no_fixed_release
GO-2026-5320	github.com/yuin/goldmark	go	v1.1.27	False	no_fixed_release
GO-2026-5932	golang.org/x/crypto	go	0.57.0	False	no_fixed_release

The buyer should distinguish between two categories within the 18 advisories. The first category is advisories on dependencies the seller controls directly: the two github.com/hashicorp/terraform advisories (GHSA-h3p9-wrgx-82cm and GO-2022-0839) target the target's own package and are therefore remediable only by patching the target's own code. The second category is advisories on third-party dependencies (yaml.v2, logrus, go-gh, go-jose, golang-jwt, mapstructure, goldmark, golang.org/x/crypto): these are remediable only when the upstream maintainers publish a fix, which the OSV data confirms has not happened. The condition precedent on CVE remediation (condition 4) should therefore be scoped to reachable known CVEs that have a fixed release, and the residual 18 should be carried as a priced integration work stream rather than a close blocker.

5.3 Secret Hygiene

The secret scanner identified 12 candidates across 3 kinds: 1 certificate block, 2 connection strings, and 9 private key blocks. Of the 12, 3 are high-confidence matches, 8 are history-only (present in git history but not in the current HEAD), and 0 are live critical. The high-confidence count is the most operationally material: 3 secrets at 99 percent confidence (in internal/backend/remote-state/oci/backend_test.go, internal/communicator/ssh/communicator_test.go, and internal/lang/funcs/crypto_test.go) require immediate rotation before close. The history-only count of 8 is material to the warranty layer because the secrets, even if removed from the current HEAD, remain in the git history and can be harvested by any party with read access to the repository.

Table 5.2 — Secret candidate detail

Kind	File	Line	Confidence	In History	Risk Cap
Certificate block	internal/communicator/winrm/provisioner_test.go	78	79	False	High
Connection string	internal/backend/remote-state/pg/backend.go	33	74	False	Medium
Connection string	internal/backend/remote-state/pg/backend_test.go	37	76	False	Medium
Private key block	internal/backend/remote-state/oci/backend_test.go	320	99	True	High
Private key block	internal/communicator/ssh/communicator_test.go	493	99	True	High
Private key block	internal/communicator/ssh/communicator_test.go	464	99	True	High
Private key block	internal/communicator/ssh/communicator_test.go	33	99	False	High
Private key block	internal/lang/funcs/crypto_test.go	716	99	True	High
Private key block	internal/lang/funcs/crypto_test.go	687	99	True	High
Private key block	internal/lang/funcs/crypto_test.go	745	99	True	High
Private key block	internal/lang/funcs/crypto_test.go	774	99	True	High
Private key block	internal/lang/functions_test.go	1416	99	True	High

The location of the candidates is itself informative. Eleven of the 12 are in test files (provisioner_test.go, backend_test.go variants, communicator_test.go, crypto_test.go, functions_test.go), and one is in production code (internal/backend/remote-state/pg/backend.go, line 33). The test-file candidates are likely test fixtures rather than real production secrets, but they must still be rotated because a high-confidence match in a test fixture can be harvested and reused. The single production candidate in pg/backend.go is a connection string at 74 percent confidence; this is the highest-priority rotation target because it sits in the production code path. The buyer's condition precedent on secret rotation (condition 8) should require rotation of all 12 candidates and the establishment of a credential rotation process for the post-close integration period.

5.4 Infrastructure-as-Code Findings

The IaC scanner checked 400 files and identified 2 findings, both at medium severity and both flagging the same rule (IAC-DOCKER-ROOT, Container runs as root). The two findings are in Dockerfile (line 26) and build.Dockerfile (line 59). The remediation guidance is to add a dedicated non-root user and a USER directive in every stage. The IaC posture is otherwise clean: zero critical, zero high, zero low findings. The two medium findings are not close blockers but should be remediated as part of the integration plan, because container-root execution is a known hardening baseline that acquirer security teams typically require.

Table 5.3 — IaC findings

Rule ID	Title	Severity	File	Line	Remediation
IAC-DOCKER-ROOT	Container runs as root	Medium	Dockerfile	26	Add a dedicated non-root user and USER directive in every stage.
IAC-DOCKER-ROOT	Container runs as root	Medium	build.Dockerfile	59	Add a dedicated non-root user and USER directive in every stage.

5.5 Attack Surface Composition

The attack surface scanner identified 59 candidate entry points. Of these, 58 are unreferenced (no incoming edges detected by the static analyzer), 0 are dynamic, and 24 are untested. The candidate exposure score is 0.06, which is a low overall signal but should be read alongside the 24 untested candidates rather than in isolation. The untested count is the more material observation: 24 of 59 candidates (40.7 percent) lack any test coverage, which means the scanner cannot confirm whether the entry points are reachable or whether they execute safely under adversarial input.

The candidate endpoints concentrate in three categories: main packages (15 candidates), init functions (43 candidates), and one additional symbol. The main packages include the primary terraform entry point (main.go:53), test-data servers (login-oauth-server, login-tfe-server, statelocker.go), and tools (defect-detector, loggraphdiff, protobuf-compile). The init functions include the telemetry initializers, the version initializer, the registry client, and a substantial set of generated protobuf bindings (tfplugin5.pb.go, tfplugin6.pb.go, stacks.pb.go). The generated protobuf bindings dominate the untested candidate list because they are typically excluded from unit test coverage by convention; the buyer should commission an independent review of the protobuf-generated entry points to verify that they do not expose unsafe deserialization paths.

Table 5.4 — Attack surface candidate endpoints (sampled)

Symbol	File	Line	Endpoint Type	Exposure	Risk	Test Status
main	main.go	53	candidate	unknown	low	tested
init	checkpoint.go	18	candidate	unknown	low	tested
init	internal/backend/remote-state/inmem/backend.go	32	candidate	unknown	low	tested
init	internal/cloudplugin/cloudproto1/cloudproto1.pb.go	215	candidate	unknown	low	untested
init	internal/command/telemetry.go	13	candidate	unknown	low	tested
main	internal/command/testdata/login-oauth-server/main.go	26	candidate	unknown	low	untested

© MTD Services

Symbol	File	Line	Endpoint Type	Exposure	Risk	Test Status
init	internal/command/testdata/login-tfe-server/tfeserver.go	79	candidate	unknown	low	untested
init	internal/tfplugin5/tfplugin5.pb.go	7827	candidate	unknown	low	untested
init	internal/tfplugin6/tfplugin6.pb.go	9131	candidate	unknown	low	untested
main	tools/protobuf-compile/protobuf-compile.go	229	candidate	unknown	low	untested

5.6 Privacy Findings

The privacy scanner identified 6 findings: 1 plaintext credential material in committed config (PRIV-PLAINTEXT-CRED, medium severity) and 5 sensitive data in log statements (PRIV-SENSITIVE-LOG, medium severity). The plaintext credential is in internal/cloud/testdata/apply-json-with-outputs/plan-redacted.json at line 51; despite the file name, the scanner detected a credential pattern in the file and the buyer should treat the file as containing live credential material unless proven otherwise. The sensitive logging findings span the Azure, Kubernetes, OCI, and S3 backends and include tokens, secrets, and apikeys.

Table 5.5 — Privacy findings

Rule ID	Category	Title	Severity	File	Line
PRIV-PLAINTEXT-CRED	Plaintext credential	Plaintext credential material in committed config	Medium	internal/cloud/testdata/apply-json-with-outputs/plan-redacted.json	51
PRIV-SENSITIVE-LOG	Sensitive logging	Sensitive data in log statement (token)	Medium	internal/backend/remote-state/azure/api_client.go	159
PRIV-SENSITIVE-LOG	Sensitive logging	Sensitive data in log statement (secret)	Medium	internal/backend/remote-state/kubernetes/client.go	411
PRIV-SENSITIVE-LOG	Sensitive logging	Sensitive data in log statement (apikey)	Medium	internal/backend/remote-state/oci/backend.go	185
PRIV-SENSITIVE-LOG	Sensitive logging	Sensitive data in log statement (apikey)	Medium	internal/backend/remote-state/oci/backend_test.go	341
PRIV-SENSITIVE-LOG	Sensitive logging	Sensitive data in log statement (token)	Medium	internal/backend/remote-state/s3/backend_complete_test.go	1515

The privacy findings carry two implications for the deal. First, the buyer should require a representation covering personal-data handling in the shipped product, scoped to the hits recorded in the credential inventory. Second, the sensitive logging findings in the production backend files (azure/api_client.go, kubernetes/client.go, oci/backend.go) should be remediated as part of the integration plan because they create regulatory disclosure obligations in jurisdictions with strict data protection regimes. The two test-file findings (oci/backend_test.go and s3/backend_complete_test.go) are lower priority but should still be remediated because test fixtures can leak into production logs.

6. Supply Chain & Dependency Posture

6.1 Dependency Posture Summary

The dependency pillar reports a medium-risk posture driven primarily by the absence of version pinning. Of 329 direct dependencies, 0 are pinned, 329 are ranged, and 329 are unpinned, producing a pin ratio of 0.0 percent. The transitive resolved package count is 3,873 across 11 lockfiles, and 11 manifests were found. The EOL pillar reports 1 runtime checked (Go 1.26.8, latest 1.26.9, not EOL), confirming that the runtime itself is current, but the unpinned transitive supply chain creates upstream-advisory and silent-breaking-change exposure that materially affects post-close security operating cost and SBOM obligations.

The 0.0 percent pin ratio is the most operationally material observation in this section. Ranged dependencies in Go modules (the format used in go.mod) resolve to the latest compatible version at build time, which means that two builds of the same commit can produce different binaries if upstream maintainers publish new versions between the builds. The buyer's post-close security operations team will inherit this exposure: a new advisory in any of the 329 direct dependencies, or in any of the 3,873 transitive dependencies, can trigger a forced rebuild and re-deploy cycle. The remediation is dependency pinning, which the cost model prices at USD 10,725 (the dependency_pinning line item in Table 3.2) and which is a price-adjustment lever rather than a condition precedent.

Table 6.1 — Dependency summary metrics

Metric	Value
Direct dependency count	329
Pinned dependencies	0
Ranged dependencies	329
Unpinned dependencies	329
Pin ratio	0.0%
Transitive resolved packages	3,873
Lockfiles present	11
Manifests found	11
EOL runtime rows	1 runtime checked
EOL count	0
Runtimes checked	1

6.2 Lockfile Resolution

Eleven lockfiles were detected across the repository, with the root go.sum containing 979 resolved packages and the ten backend-specific lockfiles containing between 105 and 427 each. The distribution of resolved packages across the backend lockfiles reflects the architecture of Terraform's remote-state backends: each cloud-provider backend (azure, consul, cos, gcs, kubernetes, oci, oss, pg, s3) maintains its own go.sum because each backend is built as a separate module. The internal/legacy/go.sum contains 105 resolved packages, reflecting the legacy code path that has been preserved for backward compatibility.

© M&A Services

Table 6.2 — Lockfile resolution

Lockfile	Resolved Packages
go.sum	979
internal/backend/remote-state/azure/go.sum	365
internal/backend/remote-state/consul/go.sum	427
internal/backend/remote-state/cos/go.sum	269
internal/backend/remote-state/gcs/go.sum	265
internal/backend/remote-state/kubernetes/go.sum	361
internal/backend/remote-state/oci/go.sum	300
internal/backend/remote-state/oss/go.sum	293
internal/backend/remote-state/pg/go.sum	242
internal/backend/remote-state/s3/go.sum	267
internal/legacy/go.sum	105

6.3 Direct Dependency Inventory

The 329 direct dependencies span the major cloud providers (AWS, Azure, Google Cloud, Alibaba, OCI, OSS), configuration management tools (Consul, Serf), the HashiCorp stack (cli, go-getter, go-hclog, go-plugin, hcl, terraform-plugin-go, terraform-plugin-sdk, terraform-svchost, copywrite), and a long tail of utility libraries. The most material observations from the inventory are the heavy reliance on the AWS SDK v2 (50+ AWS service packages), the Microsoft Kiota SDK family (10+ packages for Azure resource management), and the substantial HashiCorp-authored dependencies that themselves require license and IP review.

The buyer should commission an independent SBOM with full transitive resolution before close, because the 3,873 transitive packages materially expand the supply chain attack surface beyond the 329 direct dependencies. The SBOM should be paired with continuous SCA monitoring post-close, because the ranged dependency model means new advisories can appear at any time without code changes. The condition precedent on CI gating (condition 1) should require that the CI gate includes a dependency-advisory check that fails the build on new critical advisories, which closes the loop between the ranged dependency model and the security operations team.

6.4 End-of-Life Posture

The EOL pillar verifies runtimes and dependencies against endoflife.date. One runtime was checked: Go 1.26.8, which is not EOL (the latest version is 1.26.9). The EOL count is zero. The tech_stack pillar (Section 13) detects 4 technologies (Docker, GitHub Actions, Go, Go module checksums), all at active lifecycle. The two pillars are reconciled in the model so that eol_count is the union of tech_stack and eol findings, which is why the pillars cannot contradict each other. The buyer can rely on the headline EOL posture as a positive value driver: the asset has no end-of-life technology exposure, which reduces modernization capital requirements.

7. Code Quality, Complexity & Technical Debt

© MTD Services

7.1 Complexity Hotspots

The complexity pillar reports 13,922 symbols analyzed, of which 1,169 are high-complexity and 503 are critical-complexity. The average cyclomatic complexity is 4.78, the average cognitive complexity is 5.59, and the complexity hotspot score is 0.0843. The cyclomatic percentiles (P50 = 3, P90 = 11, P99 = 33) and cognitive percentiles (P50 = 2, P90 = 14, P99 = 59) show that the distribution is heavily right-skewed: most symbols have low complexity, but the top percentile symbols carry complexity levels that drive the remediation cost. The SQALE grade is E (the second-lowest on the A-to-E scale), and the SQALE remediation effort is 28,360 days, which is the dominant contributor to the technical-debt picture.

The five critical-complexity findings (DirFromModule, validateProviderConfigs, ApplyPlan, FromPlan, ApplyComponentPlan) are the highest-priority remediation targets because they carry the highest composite risk scores and because they sit in production code paths rather than test code. DirFromModule in internal/initwd/from_module.go (cyclomatic 60, cognitive 154, composite risk 0.9821) is the most material: it is the entry point for module installation from a remote source, which is a security-sensitive code path because it parses untrusted input. The remediation cost of USD 135,600 (Table 3.2) is anchored to this hotspot and the 502 other critical-complexity symbols, distributed across the saturation-capped hours-per-unit of the critical_complexity line item.

Table 7.1 — Top complexity hotspot symbols

Symbol	File	Cyclo	Cognitive	MI	Composite Risk
TestContextPlan_actions	internal/terraform/context_plan_actions_test.go	317	409	0	0.9125
EnsureProviderVersions	internal/providercache/installer.go	127	256	0	0.9000
TestEnsureProviderVersions	internal/providercache/installer_test.go	104	145	0	0.9125
TestContext2Plan_PolicyEvaluation	internal/terraform/context_plan_policy_test.go	102	117	0	0.9125
opApply	internal/backend/local/backend_apply.go	99	232	0	0.9000
plan	internal/terraform/node_resource_abstract_instance.go	98	160	0	0.9000
validateProviderConfigs	internal/configs/provider_validation.go	95	191	0	0.9375
backendFromConfig	internal/command/meta_backend.go	78	179	0.92	0.8982
GetResource	internal/terraform/evaluate.go	77	152	3.87	0.8923
TestContextApply_actions	internal/terraform/context_apply_action_test.go	73	178	0	0.9125
apply	internal/terraform/node_resource_abstract_instance.go	73	89	4.98	0.8900
Run	internal/command/state_mv.go	72	91	5.23	0.8895
Run	internal/command/providers_lock.go	71	96	10.48	0.8790
run	internal/command/init_run.go	70	102	3.59	0.8928
plan	internal/backend/remote/backend_plan.go	69	65	14.80	0.8704
DirFromModule	internal/initwd/from_module.go	60	154	8.94	0.9821
ApplyPlan	internal/stacks/stackruntime/internal/stackeval/main_apply.go	56	114	13.21	0.9361
FromPlan	internal/stacks/stackplan/from_plan.go	36	70	22.15	0.9307

© IntelliServices

Symbol	File	Cyclo	Cognitive	MI	Composite Risk
ApplyComponentPlan	internal/stacks/stackruntime/internal/stackeval/applying.go	47	59	18.79	0.9249

7.2 Duplication and SQALE Debt

The duplication analysis identified 4,549 groups spanning 199,946 duplicated lines of code, producing a duplication ratio of 28.3 percent. The debt ratio is 32.1 percent and the SQALE remediation effort is 28,360 days. These figures are correlated but not redundant: the duplication ratio measures the share of code that appears in multiple locations, while the debt ratio measures the share of development effort required to remediate all identified code smells. The high duplication ratio is a positive finding in one respect (it indicates consistent patterns across the codebase) and a negative finding in another (it indicates that refactoring a duplicated pattern requires touching multiple locations).

The SQALE remediation effort of 28,360 days is not directly priced in the line-item model because the same modules are also costed as complexity hotspots; the double-counting ledger (Section 3.6) verifies that the overlap is justified rather than priced twice. The hours-equivalent figure of 226,882 hours is reported in the SQALE remediation basis but is not added to the headline remediation cost because the complexity line items already capture the remediation effort for the same symbols. The buyer should treat the SQALE figure as a separate read on the debt magnitude, useful for benchmarking against industry peers but not as an additional cost item.

Table 7.2 — Code quality summary

Metric	Value
Symbols analyzed	13,922
High-complexity symbols	1,169
Critical-complexity symbols	503
Average cyclomatic complexity	4.78
Average cognitive complexity	5.59
Complexity hotspot score	0.0843
SQALE grade	E
SQALE remediation effort (days)	28,360
Debt ratio	32.1%
Duplicated LOC	199,946
Duplication ratio	28.3%
Duplication groups	4,549
Debt markers total	0
Critical-path markers	0
Debt density	0

7.3 Debt Markers

© Meridian Advisory Services

The debt-marker scanner identified 0 debt markers (annotations such as TODO, FIXME, HACK, or technical-debt markers) in the codebase. This is a clean result on the marker dimension but should be read with caution: a zero debt-marker count can indicate either that the codebase carries no annotated technical debt (the positive read) or that the engineering team does not annotate technical debt in source (the alternative read). The buyer should ask the seller's engineering leadership which interpretation applies, because the answer materially affects the post-close integration plan. A codebase with no annotated debt typically requires the acquirer to discover the debt through refactoring work, which is more expensive than remediating annotated debt that can be triaged from the marker inventory.

8. Testing, Coverage & Regression Risk

8.1 Test Deficit and Coverage Method

The testing pillar reports a high-risk posture driven by two correlated findings: a test deficit score of 0.0347 across 202 modules, with 105 modules below the threshold and an overall test ratio of 34.0 percent; and a coverage gap of 50 high-value modules with zero test coverage. The coverage method is a static symbol-share proxy (test-file symbols weighted by structural value), which is not runtime or line/branch coverage. The buyer must distinguish between the two: runtime coverage requires execution of the test suite against instrumented binaries, while the static symbol-share proxy is a structural measure derived from the file inventory. The static proxy is sufficient for triage and for sizing the test investment, but it cannot substitute for runtime coverage in the warranty layer.

The effective insured value of USD 0 is a notable figure: it represents the value of code that is both high-risk and tested, and the zero means that none of the high-value modules carry test coverage that would offset their risk. The buyer should treat this as a leading indicator of post-close regression incidents rather than as a clean result, because the combination of high-complexity hotspots (Section 7) and zero effective insured value means that any future refactoring will proceed without a test-based safety net. The condition precedent on coverage (condition 3) and the price-adjustment lever on test investment (condition 9) together address this exposure, but the buyer should expect the test investment to extend beyond the post-close integration window.

Table 8.1 — Testing summary metrics

Metric	Value
Modules analyzed (test deficit)	202
Modules below threshold	105
Overall test ratio	34.0%
Test deficit score	0.0347
High regression-risk modules	7
Coverage modules analyzed	202
Effective insured value	0
Value coverage ratio	0.0%

© MTP Services

Metric	Value
Coverage method	Static symbol-share proxy (not runtime or line/branch coverage)
High-value gaps	50

8.2 High-Value Coverage Gaps

The 50 high-value coverage gaps are concentrated in the modules that carry the highest structural risk weights. The top 14 gaps, listed below, span the core engine (internal/terraform at risk weight 1.0, internal/command at 0.798), the cloud integration layer (internal/cloud at 0.329), the stacks runtime (internal/stacks/stackruntime/internal/stackeval at 0.223), and the configuration system (internal/configs at 0.165). The risk weight is a structural measure derived from the call graph and the value-at-risk model; it does not directly translate to a financial figure but it does prioritize the test investment work.

Table 8.2 — High-value coverage gaps (top 14)

Module	Risk Weight	Test Coverage Ratio
internal/terraform	1.0000	0
internal/command	0.7980	0
internal/cloud	0.3294	0
internal/stacks/stackruntime/internal/stackeval	0.2227	0
internal/configs	0.1647	0
internal/backend/local	0.1538	0
internal/command/views	0.1436	0
internal/addrs	0.1203	0
internal/stacks/stackruntime	0.1169	0
internal/registry/test	0.1147	0
internal/legacy/helper/schema	0.1126	0
internal/backend/remote-state/s3	0.1048	0
internal/rpcapi	0.0796	0
internal/getproviders	0.0653	0

The 7 high-regression-risk modules identified by the test_deficit scanner (internal/cloud/e2e, internal/command/testing, internal/command/e2etest, internal/backend/remote-state/http, internal/backend/init, internal/command/junit, internal/stacks/stackruntime) are the modules where the test ratio is below threshold and the regression risk score is non-zero. The test investment line item (USD 5,600 across 7 modules, at 16 HPU per module with a saturation cap of 20) is sized to bring these modules to threshold, but the buyer should expect that the actual test investment required will exceed the modelled figure because the static symbol-share proxy under-counts tests in modules that have integration rather than unit test coverage. The buyer should commission an independent runtime coverage analysis before close to validate the static figure.

8.3 Test-Deficit Module Inventory

© MRD Services

The test-deficit module inventory lists 202 modules with their test file count, source file count, test ratio, and regression risk. The top modules by regression risk (internal/command/jsonformat/structured/attribute_path at 0.1367, internal/backend/pluggable at 0.1313, internal/command/junit at 0.1222, internal/stacks/stackruntime/internal/stackeval at 0.1130) are those where the test ratio is below threshold and the structural risk weight is high. The buyer should sequence the test investment work against the regression risk score rather than against the raw test ratio, because the regression risk score weights the gaps by their value exposure.

Table 8.3 — Test-deficit module inventory (sampled)

Module Path	Test Files	Source Files	Test Ratio	Regression Risk
internal/cloud/e2e	13	0	1.0000	0
internal/command/testing	1	0	1.0000	0
internal/command/e2etest	22	3	0.8800	0.0687
internal/backend/remote-state/http	5	2	0.7143	0.0678
internal/backend/init	2	1	0.6667	0.0375
internal/command/junit	2	1	0.6667	0.1222
internal/stacks/stackruntime	9	7	0.5625	0.1130
internal/backend/pluggable	1	1	0.5000	0.1313
internal/builtin/providers/terraform	4	4	0.5000	0.0766
internal/builtin/provisioners/file	1	1	0.5000	0.0676
internal/builtin/provisioners/local-exec	1	1	0.5000	0.1000
internal/builtin/provisioners/remote-exec	1	1	0.5000	0.1068
internal/command/jsonformat/structured/attribute_path	1	1	0.5000	0.1367
internal/communicator/remote	1	1	0.5000	0.0375

9. Architecture, Entanglement & Integration Risk

9.1 Entanglement Posture

The entanglement pillar reports a critical-risk posture driven by two structural findings. First, a single circular dependency cycle of length 163 was detected, involving 163 modules in a cyclic relationship that the SCC (strongly-connected-components) algorithm ranked as the highest-priority decoupling target. Second, 852 cross-layer violations were detected, with the presentation layer calling into the data layer across 14 distinct module pairs. The entanglement score is 1.0 (the maximum on the 0-to-1 scale), which means the codebase exhibits the highest possible structural coupling on the dimensions the scanner measures. The integration risk score is 4.33 on a 0-to-5 scale, and the schema drift risk is medium.

The critical entanglement posture has two material consequences for the deal. First, the integration cost of the asset will be substantially higher than the headline remediation figure suggests, because the 163-module cycle and the 852 cross-layer violations mean that any attempt to extract a subset of the codebase for integration into the acquirer's product will require a decoupling effort that scales with the cycle length rather than with the surface area of the extracted subset. Second, the

© MTD Services

post-close maintenance cost will be elevated because changes in one layer (e.g., the presentation layer in internal/backend/local) propagate into the data layer (e.g., internal/cloud, internal/configs) and vice versa. The buyer should price the integration drag explicitly in the deal model rather than relying on the 80 percent integration drag figure alone, which is a model parameter rather than a measured cost.

9.2 Cross-Layer Violations

The 852 cross-layer violations concentrate in the backend layer, with internal/backend/local and the cloud-provider backends (azure, consul, cos, gcs, kubernetes, oci, oss, pg, s3) accounting for the majority of the violations. The violations are split between calls (where presentation layer code directly invokes data layer code) and imports (where presentation layer code imports data layer packages). Both violation types break the architectural intent of layer separation, but the call violations are more material because they create runtime coupling in addition to compile-time coupling.

Table 9.1 — Cross-layer violation detail (sampled)

From Module	To Module	From Layer	To Layer	Rel Type	Severity
internal/backend/local	internal/cloud	presentation	data	calls	critical
internal/backend/local	internal/terraform	presentation	data	calls	critical
internal/backend/local	internal/configs	presentation	data	imports	critical
internal/backend/local	internal/configs/configschema	presentation	data	imports	critical
internal/backend/local	internal/providers	presentation	data	imports	critical
internal/backend/local	internal/providers/testing	presentation	data	imports	critical
internal/backend/local	internal/schemarepo	presentation	data	imports	critical
internal/backend/local	internal/terraform	presentation	data	imports	critical
internal/backend/remote-state/azure	internal/cloud	presentation	data	calls	critical
internal/backend/remote-state/azure	internal/backend/backendbase	presentation	data	imports	critical
internal/backend/remote-state/azure	internal/configs/configschema	presentation	data	imports	critical
internal/backend/remote-state/consul	internal/cloud	presentation	data	calls	critical
internal/backend/remote-state/consul	internal/backend/backendbase	presentation	data	imports	critical
internal/backend/remote-state/consul	internal/configs/configschema	presentation	data	imports	critical

The pattern across the violations is consistent: the backend implementations call directly into the cloud, configuration, and provider packages rather than going through an abstraction layer. This pattern is intentional in the original design (it allows each backend to specialize its interaction with the cloud APIs) but it creates the integration drag the buyer will inherit. The decoupling work (priced at USD 1,000 as a single line item with a 10-unit cap) is a planning placeholder rather than a priced remediation; the actual decoupling effort will require architectural work that the model does not capture. The buyer should treat the decoupling condition (condition 5) as a priced integration work stream rather than a discrete remediation, with the price to be sized by an independent architect post-close.

© M&A Services

9.3 Integration and Serialization Risks

The integration pillar reports an integration risk score of 4.33 (on a 0-to-5 scale), 7 proprietary couplings, 7 total integration findings, and a schema drift risk of medium. The API contract maturity shows 14 endpoints, no GraphQL, no OpenAPI, 17 proto files, and 329 types. The database coupling is minimal: zero destructive DDL, zero migration files, no ORM usage, and 46 raw SQL count (the SQL count is in the legacy code path and does not represent active database coupling). The serialization risks (7 findings) include 2 high-severity dangerous deserialization findings (eval_call and exec_call in the lang package) and 5 low-severity gob_decode findings in the tfdiags and command packages.

Table 9.2 — Serialization risks

Risk Type	File	Line	Severity	Description
serialization:eval_call	internal/terraform/node_provider.go	132	High	Dangerous deserialization: eval_call
serialization:exec_call	internal/lang/eval_test.go	523	High	Dangerous deserialization: exec_call
serialization:exec_call	internal/lang/eval_test.go	528	High	Dangerous deserialization: exec_call
serialization:gob_decode	internal/tfdiags/diagnostics.go	232	Low	Dangerous deserialization: gob_decode
serialization:gob_decode	internal/command/workdir/backend_state.go	87	Low	Dangerous deserialization: gob_decode
serialization:gob_decode	internal/tfdiags/rpc_friendly.go	7	Low	Dangerous deserialization: gob_decode
serialization:gob_decode	internal/tfdiags/rpc_friendly_test.go	8	Low	Dangerous deserialization: gob_decode

The high-severity serialization findings (eval_call in node_provider.go and exec_call in eval_test.go) are the most material findings in this section. The eval_call finding indicates that the provider node deserialization path uses eval-based code execution, which is a known dangerous pattern because it can execute attacker-controlled input if the input is not validated. The exec_call findings in the test file are lower priority because they are in test code, but they should still be reviewed. The buyer should require the seller to remediate the eval_call finding before close as an additional condition, and to provide a representation that the gob_decode findings do not accept untrusted input.

10. Engineering Capacity, Bus Factor & Knowledge Loss Vulnerability

10.1 Key-Person Risk Posture

The engineering capacity pillar reports a high-risk posture driven by retention risk and a single critical key-person dependency. The total author count is 2,375, the overall bus factor is 11, and the active contributor count is 73 (3.1 percent of the total). The at-risk contributor count is 66, and the knowledge loss vulnerability score is 0.5052 on a 0-to-1 scale. The 124 critical sole-maintainer modules identified by the scanner are the modules where a single contributor holds the institutional

© MTD Services

knowledge; the 276 critical modules in the retention model are the superset that includes modules at risk for other reasons (low bus factor, contributor flight risk, etc.).

The five material findings in this section are the key-person risks (radek.simko@gmail.com at high departure impact, plus four medium-impact contributors) and the retention risks (Daniel Banck, Samsondeen, Matej Risek, Radek Simko, Sarah French, Austin Valle, Brandon Croft, plus 18 critical-tier knowledge-loss contributors). The retention risk findings are inferred rather than detected: the scanner uses git history to identify contributors whose last commit was within a recent window, and infers retention risk from the gap. The buyer should treat the inferred retention risks as a starting list for retention-bonus negotiations rather than as confirmed departures; the actual retention decisions will depend on factors the scanner cannot see (compensation, notice periods, individual intent).

Table 10.1 — Top key-person risk rows

Author	Risk Score	Total Commits	Departure Impact
radek.simko@gmail.com	0.5394	2,176	High
mart@degeneration.co.uk	0.4265	81,053	Medium
110428419+hashicorp-copywrite[bot]@users.noreply.github.com	0.4137	4,527	Medium
j.bardin@gmail.com	0.3095	48,111	Medium
15078782+SarahFrench@users.noreply.github.com	0.3080	32,733	Medium
liam.cervante@hashicorp.com	0.2857	75,115	Low
dbanck@users.noreply.github.com	0.2304	10,136	Low
danielmschmidt92@gmail.com	0.1877	20,824	Low
40821565+dsa0x@users.noreply.github.com	0.1559	21,901	Low
sebastian.rivera@hashicorp.com	0.1538	4,763	Low
alisdair@users.noreply.github.com	0.1363	10,479	Low
nick.fagerlund@gmail.com	0.1153	2,671	Low
chris@chrisarcand.com	0.1124	3,493	Low
radeksimko@users.noreply.github.com	0.1108	6,074	Low

10.2 Knowledge Loss Vulnerability

The knowledge loss vulnerability table ranks contributors by tenure, days since last commit, retention risk, and blast radius (the number of modules affected by their departure). The top 18 contributors at critical retention risk are listed below. The blast radius column is the most material observation: Radek Simko (radek.simko@gmail.com) carries a blast radius of 137 modules, James Bardin carries 93, Daniel Schmidt carries 67, Sarah French carries 73, and Kristin Laemmert carries 42. The buyer should prioritise retention agreements for these five contributors in particular, because their departures would create the largest knowledge transfer burden.

Table 10.2 — Knowledge loss vulnerability (critical-tier contributors)

Author Name	Email	Tenure (days)	Days Since Last Commit	Retention Risk	Blast Radius
Sarah French	15078782+SarahFrench@users.noreply.github.com	1,561	3	Critical	73
Samsondeen	40821565+dsa0x@users.noreply.github.com	673	2	Critical	26

© MLnSciences

Author Name	Email	Tenure (days)	Days Since Last Commit	Retention Risk	Blast Radius
kmoe	5575356+kmoe@users.noreply.github.com	2,629	104	Critical	15
Austin Valle	austinvalle@gmail.com	570	1	Critical	10
Brandon Croft	brandon.croft@gmail.com	1,833	197	Critical	10
Daniel Banck	daniel@dbanck.de	622	2	Critical	8
Daniel Schmidt	danielmschmidt92@gmail.com	3,446	9	Critical	67
Daniel Banck	dbanck@users.noreply.github.com	1,368	22	Critical	18
James Bardin	j.bardin@gmail.com	3,797	6	Critical	93
Mark DeCrane	mark.decrane@hashicorp.com	960	7	Critical	4
Matej Risek	matej.risek@hashicorp.com	1,049	36	Critical	3
Kristin Laemmert	mildwonkey@users.noreply.github.com	3,046	17	Critical	42
Radek Simko	radek.simko@gmail.com	4,241	114	Critical	137
Radek Simko	radeksimko@users.noreply.github.com	983	2	Critical	7

10.3 Module Bus Factor

The module bus factor analysis ranks modules by the number of distinct contributors required to maintain them. The top modules (internal/command at bus factor 6, internal/cloud at 5, internal/backend/remote at 4, internal/command/views at 4, internal/configs at 4) carry the highest bus factors, which is a positive signal: these modules have multiple contributors and are therefore less exposed to key-person risk. The critical-sole-maintainer modules (the 124 modules with bus factor 1) are the inverse: they are the modules where a single contributor holds the institutional knowledge and where the retention risk is highest. The buyer should map the 124 critical sole-maintainer modules against the contributor retention list to identify the modules where the key-person risk is concentrated.

Table 10.3 — Module bus factor (top 14)

Module Path	Bus Factor	Total Files
internal/command	6	132
internal/cloud	5	46
internal/backend/remote	4	21
internal/command/views	4	56
internal/configs	4	63
internal/configs/configload	4	9
internal/command/views/json	3	18
internal/initwd	3	6
internal/plans	3	18
internal/plans/planfile	3	7
internal/plugin6	3	5
internal/plugin6/mock_proto	3	2
internal/terraform	3	244
internal/tfdiags	3	30

© MTD Services

10.4 Git Team Signals

The git team signals provide additional context on the engineering organisation's working patterns. The after-hours commit percentage is 26 percent (520 of 2,000 commits), which is moderate but not extreme; the cadence is 60.61 commits per week; the top-1 author share is 28.5 percent, the top-3 share is 54.75 percent, and the top-5 share is 67.05 percent. The ghost author count (contributors whose identity could not be resolved) is 16. The bus factor proxy of 0.715 is the reciprocal of the top-1 share; the lower the proxy, the higher the concentration. These signals suggest a contributor base with moderate concentration (the top contributor accounts for 28.5 percent of commits) and moderate working-hours pressure (26 percent after-hours).

Table 10.4 — Git team signals

Metric	Value
After-hours commits	520
After-hours pct	0.26 (26%)
Bus factor proxy	0.715
Cadence per week	60.61
Ghost author count	16
Modules analyzed	42
Top-1 author share	0.285 (28.5%)
Top-3 author share	0.5475 (54.75%)
Top-5 author share	0.6705 (67.05%)
Total commits	2,000
Unique authors	48
Unique emails	52

11. Maintenance, Documentation & Operational Maturity

11.1 Maintenance Posture (Unverified)

The maintenance pillar is reported as NOT_ASSESSED because the asset-staleness scanner did not execute on any files. The maintenance summary metrics (total files, neglected files, at-risk files, healthy files, average staleness, maintenance backlog score) are all zero, and the maintenance asset rows are absent. This is a coverage gap rather than a clean result: the buyer should not interpret the zero figures as evidence that the codebase has no maintenance backlog. The scanner's failure to execute may be a configuration issue (the scanner was not pointed at the right file set) or a tooling issue (the scanner does not support the file types in the repository). The buyer should commission an independent maintenance scan before close or accept that the maintenance posture remains unverified.

The 2-pillar degraded confidence finding (Debt and Maintenance) is reflected in the 76 confidence score: the confidence policy formula is base 20 + coverage weight 40 + risk signal 40, capped at 20 when coverage is 0 and floored at 60 for the no-go tier. With 90.9 percent coverage (20

©MTD Services

of 22 pillars assessed), the coverage contribution is 36.4; with a risk signal of 50.0 percent, the risk signal contribution is 20. The total is 20 + 36.4 + 20 = 76.4, which rounds to 76. The buyer should be aware that the confidence score is bounded above by the coverage and the risk signal; a higher score would require either full coverage (which would require the Debt and Maintenance pillars to be assessed) or a higher risk signal (which would require more findings at higher severities).

11.2 Documentation Inventory

The documentation inventory contains 18 documents, of which 0 are stale. The staleness ratio is 0.0 percent, the average docstring density is 0.0033, and the source-files sample for the documentation scanner was 400 files. The low docstring density is a notable observation: 0.0033 docstrings per source file means that approximately 1 in 300 source files carries a docstring, which is a low rate for a codebase of this size. The buyer should not interpret the 0.0 percent staleness ratio as a positive signal in isolation, because the staleness scanner measures documentation age rather than documentation coverage; the low docstring density is the more material observation.

Table 11.1 — Document inventory (sampled)

File	Last Documented	Stale
docs/debugging.md	2026-02-23	False
.github/CONTRIBUTING.md	2026-10-12	False
.github/CODE_OF_CONDUCT.md	2017-11-02	False
.github/SUPPORT.md	2019-10-15	False
.github/pull_request_template.md	2026-09-09	False
CHANGELOG.md	2026-09-09	False
LICENSE	2026-03-17	False
README.md	2025-08-11	False
docs/README.md	2025-04-01	False
docs/architecture.md	2025-04-01	False
docs/dependency-upgrades.md	2025-01-21	False
docs/destroying.md	2025-09-23	False
docs/planning-behaviors.md	2025-04-01	False
docs/plugin-protocol/README.md	2025-05-09	False

11.3 CI/CD Maturity

The CI maturity pillar reports a high-risk posture driven by low gate strictness. CI is defined (11 files checked, GitHub Actions workflow), tests are run in CI (true), and CI runs on pull request (true) and on push or PR (true) and on schedule (true). The reported coverage in CI is not quantified, which means the CI does not surface coverage metrics. The gate strictness is low, which means the CI gates are advisory rather than blocking: PRs can be merged even when checks fail. The buyer should treat this as a condition precedent (condition 1) requiring the establishment of an automated CI gate that blocks merges on failing checks. The 11 CI files span the build workflows, the changelog validation, the equivalence tests, and the issue-comment handling.

© IP Services

Table 11.2 — CI summary metrics

Metric	Value
CI defined	True
CI files checked	11
Gate strictness	Low
Reported coverage in CI	Not quantified
Run on pull request	True
Run on push or PR	True
Run on schedule	True
Tests run in CI	True
CI kinds	github-actions: 11

The 11 CI files are listed below. The buyer should review each workflow file to understand the actual gate behaviour, because the gate_strictness = low label is a high-level classification rather than a per-workflow analysis. The build-terraform-cli.yml and checks.yml workflows are the most material because they gate the primary build and test pipeline; the changelog-validation, equivalence-test, and issue-comment workflows are operational rather than quality gates.

Table 11.3 — CI files

#	Workflow file
1	.github/workflows/backport.yml
2	.github/workflows/build-terraform-cli.yml
3	.github/workflows/build.yml
4	.github/workflows/changelog-validation.yml
5	.github/workflows/checks.yml
6	.github/workflows/enforce-changelog.yml
7	.github/workflows/equivalence-test-diff.yml
8	.github/workflows/equivalence-test-manual-update.yml
9	.github/workflows/equivalence-test-update.yml
10	.github/workflows/issue-comment-created.yml
11	.github/workflows/lock.yml

12. AI/ML Footprint

12.1 AI/ML Posture

The AI/ML pillar reports an informational-risk posture, which is the lowest severity tier in the model. The scanner scanned 2,040 source files, identified 1 framework group in use (Hugging Face, 14 hits) and 1 SDK group in use (Azure OpenAI, 12 hits), and found 0 bundled model files, 0 production model files, and 0 unlicensed model files. The AI/ML exposure is therefore limited to framework and SDK references in the source code rather than to actual model deployments. The informational posture is consistent with the asset being an infrastructure-as-code tool rather than an AI/ML product.

© M&A Services

The buyer should note two observations from this section. First, the presence of Hugging Face framework hits (14) and Azure OpenAI SDK hits (12) suggests that the codebase contains references to AI/ML frameworks, even though no models are bundled. The buyer should commission an independent review of these references to verify that they are not used in production code paths that would trigger AI/ML-specific regulatory obligations (such as the EU AI Act's transparency requirements for general-purpose AI systems). Second, the absence of bundled model files means there is no model-licensing or training-data IP risk in the traditional AI/ML diligence sense, but the framework and SDK references should be reviewed for completeness.

Table 12.1 — AI/ML summary metrics

Metric	Value
AI/ML source files scanned	2,040
Framework groups in use	1
SDK groups in use	1
Bundled model files	0
Production model files	0
Unlicensed model files	0
Model file rows	0 model files
Framework hits	Hugging Face: 14
SDK hits	Azure OpenAI: 12

13. Technology Stack & Lifecycle Posture

13.1 Stack and Lifecycle

The technology stack pillar reports a low-risk posture. Four technologies were detected: Docker, GitHub Actions, Go, and Go module checksums. All four are at active lifecycle and carry no EOL or legacy flags. The EOL count is zero. The runtime verification (Go 1.26.8 against latest 1.26.9) confirms that the runtime is current, with one minor version behind the latest release. The buyer should treat this as a positive value driver: a clean active stack reduces modernization capital requirements and supports the warranty representation that the asset is built on supported technology. The structural-not-runtime caveat of technology detection is material to warranty representations: the scanner detects technologies from manifest and configuration files rather than from runtime telemetry, which means the warranty should be scoped to the detected technologies rather than to the actual runtime stack.

Table 13.1 — Technology inventory

Technology	Lifecycle	Confidence
Docker	Active	0.7
GitHub Actions	Active	1.0
Go	Active	1.0
Go (module checksums)	Active	1.0

© M&D Services

Table 13.2 — Runtime EOL inventory

Product	Runtime	Version	Is EOL	Latest
golang	go	1.26.8	False	1.26.9

The EOL reconciliation between tech_stack and eol is implemented as a union: the tech_stack pillar counts offline repo lifecycle markers, while the eol pillar verifies runtimes and dependencies against endoflife.date. The two pillars cannot contradict each other because the headline eol_count is the union of both. The buyer can therefore rely on the EOL count of zero as a defensible representation that the asset has no end-of-life technology exposure, subject to the structural-not-runtime caveat noted above.

14. Consolidated Risk Register & Top Risks

14.1 Top Risks

The consolidated top-risks table presents the 10 most material findings by severity. The table includes the finding ID, pillar, title, severity, evidence state, impact, file, line, estimated cost, and commit. The estimated cost column is meaningful only for findings that have a line-item cost: DirFromModule at USD 135,600 and Daniel Banck at USD 30,800. The remaining critical rows are priced through their pillar-level escalation rows rather than as item-level line items, because the cost model sizes remediation at the category level. The evidence state column distinguishes Detected, not verified (scanner-detected signals) from Inferred (signals inferred from indirect evidence such as git history).

Table 14.1 — Consolidated top risks

ID	Pillar	Title	Severity	Evidence State	File	Cost (USD)
F-complexity-005	complexity	ApplyComponentPlan	Critical	Detected, not verified	internal/stacks/stackruntime/internal/stackeval/applying.go	Not quantified
F-complexity-003	complexity	ApplyPlan	Critical	Detected, not verified	internal/stacks/stackruntime/internal/stackeval/main_apply.go	Not quantified
F-retention-018	retention	Brandon Croft	Critical	Inferred	Not quantified	Not quantified
F-retention-010	retention	Austin Valle	Critical	Inferred	Not quantified	Not quantified
F-complexity-001	complexity	DirFromModule	Critical	Detected, not verified	internal/initwd/from_module.go	$135,600
F-retention-001	retention	Daniel Banck	Critical	Inferred	Not quantified	$30,800
F-key_person-001	key_person	radek.simko@gmail.com	High	Inferred	Not quantified	Not quantified
F-license-003	license	(repo-level)	High	Detected, not verified	docs/plugin-protocol/LICENSE	Not quantified

© NTT DATA Services

ID	Pillar	Title	Severity	Evidence State	File	Cost (USD)
F-license-001	license	(repo-level)	High	Detected, not verified	internal/tfplugin5/LICENSE	Not quantified
F-license-002	license	(repo-level)	High	Detected, not verified	internal/tfplugin6/LICENSE	Not quantified

14.2 Major Risks Register

The major risks register presents the 9 pillar-level major risks (severity high or critical) that drive the deal conditions. The entanglement and complexity pillars are at critical severity, and the coverage, test_deficit, ip_ownership, retention, ci_maturity, secrets, and cve pillars are at high severity. The unverified column is false for all major risks, which means none of the major-risk pillars are in the NOT_ASSESSED state. The headline column carries the headline metrics that the scanner used to derive the severity rating.

Table 14.2 — Major risks register

Pillar	Area	Risk	Headline	Unverified
entanglement	Architecture	Critical	total_cycles: 1, max_cycle_length: 163, cross_layer_violations: 852	False
complexity	Engineering	Critical	DirFromModule	False
coverage	Engineering	High	modules_analyzed: 202, high_value_gaps: 50, effective_insured_value: 0	False
test_deficit	Engineering	High	total_modules: 202, modules_below_threshold: 105, overall_test_ratio: 0.3395	False
ip_ownership	Legal & Compliance	High	total_authors: 2375, external_author_count: 1930, external_blamed_line_pct: 0.7703	False
retention	People	High	Daniel Banck	False
ci_maturity	Process	High	ci_files_checked: 11	False
secrets	Security	High	candidate_count: 12, high_confidence_count: 3, history_only_count: 8	False
cve	Security	High	packages_queried: 424, unresolved_packages: 0, cve_count: 18	False

14.3 Pillar Classification Register

The pillar classification register allocates each of the 22 assessed pillars into one of four classes: Critical Risk (none), Major Risk (10), Manageable Risk (8), and Informational Finding (4). Two pillars (Debt and Maintenance) are demoted to Unverified with a 50/100 health score because their assessment populations are zero. The register includes the area, risk level, classification, health score, and unverified flag for each pillar. The classification drives the conditions in Section 2 and the deal-mechanic tags in the conditions table.

Table 14.3 — Pillar classification register

Pillar	Area	Risk	Classification	Health	Unverified
ai_ml	AI/ML Exposure	Informational	Informational Finding	90	False
attack_surface	Security	Medium	Manageable Risk	65	False
ci_maturity	Process	High	Major Risk	25	False

© M&A Services

Pillar	Area	Risk	Classification	Health	Unverified
complexity	Engineering	Critical	Major Risk	15	False
coverage	Engineering	High	Major Risk	40	False
cve	Security	High	Major Risk	40	False
debt	Engineering	Low	Unverified	50	True
dependency	Supply Chain	Medium	Manageable Risk	65	False
doc_staleness	Process	Low	Informational Finding	85	False
entanglement	Architecture	Critical	Major Risk	15	False
eol	Platform	Low	Informational Finding	85	False
iac	Security	Medium	Manageable Risk	65	False
integration	Engineering	Medium	Manageable Risk	65	False
ip_ownership	Legal & Compliance	High	Major Risk	40	False
key_person	People	Low	Informational Finding	85	False
license	Legal & Compliance	Medium	Manageable Risk	65	False
maintenance	Engineering	Low	Unverified	50	True
privacy	Legal & Compliance	Informational	Informational Finding	95	False
retention	People	High	Major Risk	40	False
secrets	Security	High	Major Risk	40	False
tech_stack	Architecture	Low	Informational Finding	85	False
test_deficit	Engineering	High	Major Risk	40	False

14.4 Deal-Breaker Register

The deal-breaker register is empty. No pillar triggered the no-go verdict because no critical-severity finding was detected on any of the deal-breaker pillars (ai_ml, attack_surface, cve, iac, license, secrets). The CVE pillar carries a high severity rather than critical because the 18 advisories are at UNKNOWN severity rather than critical CVSS scores; the secrets pillar carries a high severity rather than critical because no live critical secrets were detected (the 3 high-confidence matches are in test files rather than production code paths); the license pillar carries a medium severity because the 3 GPL-2.0 findings are repo-level files rather than packaged dependencies. The empty deal-breaker register is consistent with the ACQUIRE_WITH_CONDITIONS verdict and supports the position that the conditions are price-adjustment levers and warranties rather than close blockers.

15. Confidence, Scope Limitations & Material Unknowns

15.1 Integrity Status

The integrity block reports a warn status with 0 errors and 9 warnings. The 9 warnings fall into two categories: 7 warnings for critical_pillar_no_findings (pillars rated high or critical that render no item-level findings, requiring the reading guide to state the aggregate basis) and 2 warnings for unverified_pillar (Debt and Maintenance, which are NOT_ASSESSED). The integrity status is the model's internal health check on its own evidence base; the warn status does not invalidate the findings but it does flag the need for interpretive care in the affected pillars.

© MTD Services

Table 15.1 — Integrity issues

Tier	Code	Subject	Message (abbreviated)
warn	critical_pillar_no_findings	pillars.ci_maturity	Pillar is high but renders no item-level findings; reading guide must state aggregate basis
warn	critical_pillar_no_findings	pillars.coverage	Pillar is high but renders no item-level findings; reading guide must state aggregate basis
warn	critical_pillar_no_findings	pillars.cve	Pillar is high but renders no item-level findings; reading guide must state aggregate basis
warn	critical_pillar_no_findings	pillars.entanglement	Pillar is critical but renders no item-level findings; reading guide must state aggregate basis
warn	critical_pillar_no_findings	pillars.ip_ownership	Pillar is high but renders no item-level findings; reading guide must state aggregate basis
warn	critical_pillar_no_findings	pillars.secrets	Pillar is high but renders no item-level findings; reading guide must state aggregate basis
warn	critical_pillar_no_findings	pillars.test_deficit	Pillar is high but renders no item-level findings; reading guide must state aggregate basis
warn	unverified_pillar	pillars.debt	Pillar is not_assessed; conclusions UNVERIFIED (zero evidence population)
warn	unverified_pillar	pillars.maintenance	Pillar is not_assessed; conclusions UNVERIFIED (no files scanned by asset-staleness scanner)

15.2 Confidence Inputs

The confidence inputs are the variables that drive the 76 confidence score. The coverage is 90.9 percent (20 of 22 pillars assessed), the risk signal is 50.0 percent, the signal count is 22, the pillar total is 22, and the degraded pillars count is 2. The confidence formula is base 20 + coverage weight 40 + risk signal weight 40, capped at 20 when coverage is 0 and floored at 60 for the no-go tier. The buyer should be aware that the confidence score is not a probability of correctness; it is a model-derived score that reflects the coverage and risk signal available to the model.

Table 15.2 — Confidence inputs

Metric	Value
Integrity status	Warn
Integrity error count	0
Integrity warning count	9
Confidence coverage	90.9%
Confidence risk signal	50.0%
Confidence signal count	22
Confidence pillar total	22
Confidence degraded pillars	2
Taint flows found	0
Taint pass bounded	False

15.3 Out-of-Scope Evidence Register

The out-of-scope evidence register enumerates eight categories of question that a static, offline repository scan cannot answer. Each category is paired with the independent evidence the buyer must obtain to close the gap. The register is disclosure rather than measurement: the absence of an answer

in this report does not mean the question is unimportant; it means the scanner cannot answer it. The buyer should treat the out-of-scope categories as a checklist for independent diligence, with the independent evidence column naming the specific artifacts (logs, audits, attestations) that would close each gap.

Table 15.3 — Out-of-scope evidence register

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

16. Reading Notes & Honesty Conventions

16.1 Assessment Coverage

Assessment status is the pillar-level coverage indicator: assessed, partial, or not_assessed. Only assessed pillars support conclusions; partial and not_assessed pillars are demoted to informational, labelled UNVERIFIED, and never read as a clean bill of health. Assessment state is the five-value trust vocabulary that accompanies the legacy status: NOT_ASSESSED, PARTIALLY_ASSESSED, ASSESSED_NO_FINDINGS, ASSESSED_WITH_FINDINGS, or ERROR. A low-risk finding 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); a low or missing population flags a coverage gap rather than 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 buyer must rely on independent validation to convert a Detected signal into a Confirmed exposure. The finding ID (F-pillar-rank) is the stable handle used by the narrative, the register, the workbook, and the ledger; a body section cites the ID rather than re-rendering the finding. The contradiction ledger (Appendix

© MD Services

R) leads every material conflict with its status (resolved, by-design, or unresolved) and its decision impact; unresolved entries discount the stated confidence. The ledger never picks a side silently.

Confidence is computed as base 20 + 40 times coverage + 40 times risk signal, capped at 20 when coverage is 0 and floored at 60 for the no-go tier. 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. The confidence score is not a probability of correctness; it is a model-derived score that reflects the coverage and risk signal available to the model. A higher score requires either higher coverage or a higher risk signal, both of which require additional evidence rather than additional interpretation.

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. The marker 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. The marker 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). The marker not recorded means the source does not carry the value (an aggregate finding has no single line number; the scanner never captured a commit). The marker 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 zero. The buyer should treat every not quantified cell as a question for the seller rather than as a confirmed zero, and should require the seller to either provide the missing input or accept that the figure remains unquantified.

16.4 Deal-Mechanic Vocabulary

Three deal mechanics operate in this report. 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 rather than a work plan. Every condition in Section 2 carries exactly one of these three tags, declared where the condition is authored (evidence 2.13). 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 the severity tier: critical, high, medium, low, or informational, demoted to informational under partial coverage. The derived_risk_level preserves the scanner's own view before demotion. Health score is a 0-to-100 pillar score, demoted to 50 for an unfinished scan so that incomplete coverage never presents as a strong pillar. Classification is the deal-mechanic allocation:

© MD Services

Critical Risk, Major Risk, Minor Risk, or Acceptable, assigned in the Section 14 register and driving the conditions in Section 2. The classification is not the same as the risk level: a pillar with a high risk level may be classified as Major Risk if it carries the conditions for acquisition, while a pillar with a medium risk level may be classified as Manageable Risk if it does not.

16.6 Known Limits of This Method

EOL reconciliation: tech_stack counts offline repository technology lifecycle markers, while 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. Static source analysis: this is a source scan. It does not establish runtime behaviour, production incidents, or anything requiring access to systems, personnel, or contracts. No negotiated positions: the conditions, implications, and escalation thresholds recorded here are the deterministic output of the tool from stored evidence. They are a starting position for negotiation, not a legal opinion. The buyer should treat the report as a defensible technical read that supports commercial negotiation rather than as a substitute for legal advice.

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. The exact tool version, scan mode, indexed surface, per-pillar posture, and step timings are listed below. Every figure is reproduced from the stored payload without modification.

A.1 Report Metadata

Field	Value
Report ID	TDD-20261016-060115
Scan date	2026-10-16T06:08:13.222428+00:00
Target	HashiCorp
Scanned path	terraform
Indexed symbols	2,056
Report version	Not quantified
Elapsed seconds	417.325

A.2 Scan Mode

Parameter	Value
deal_thesis	All six theses scored side by side
offline	False (network-integrated)
scan_history	True
taint	True
target_company	HashiCorp

© M&D Services

Parameter	Value
tdd_report	True
workers	2

A.3 Pillar Posture Register

The pillar posture register summarises the area, risk level, health score, assessment status, and top-risk count for each of the 22 pillars in the model. Two pillars (Debt and Maintenance) are NOT_ASSESSED with a 50/100 health score because their assessment populations are zero. The remaining 20 pillars are assessed with usable signal.

Pillar	Area	Risk Level	Health	Status	Top Risks
AI/ML	AI/ML Exposure	Informational	90	Assessed	0
Attack Surface	Security	Medium	65	Assessed	50
CI Maturity	Process	High	25	Assessed	0
Complexity	Engineering	Critical	15	Assessed	50
Coverage	Engineering	High	40	Assessed	0
CVE	Security	High	40	Assessed	0
Debt	Engineering	Informational	50	Not assessed	0
Dependency	Supply Chain	Medium	65	Assessed	0
Doc Staleness	Process	Low	85	Assessed	0
Entanglement	Architecture	Critical	15	Assessed	0
EOL	Platform	Low	85	Assessed	0
IaC	Security	Medium	65	Assessed	2
Integration	Engineering	Medium	65	Assessed	0
IP Ownership	Legal & Compliance	High	40	Assessed	0
Key Person	People	Low	85	Assessed	50
License	Legal & Compliance	Medium	65	Assessed	50
Maintenance	Engineering	Informational	50	Not assessed	0
Privacy	Legal & Compliance	Informational	95	Assessed	6
Retention	People	High	40	Assessed	50
Secrets	Security	High	40	Assessed	0
Tech Stack	Architecture	Low	85	Assessed	0
Test Deficit	Engineering	High	40	Assessed	0

A.4 Step Timings

The scan executed in 417.325 seconds across 6 steps. The pillar scan dominated the elapsed time at 321.807 seconds (77.1 percent of total), reflecting the depth of analysis across 22 pillars. The index step at 66.437 seconds (15.9 percent) and the deep index at 10.293 seconds (2.5 percent) complete the structural analysis pipeline. The taint step at 17.258 seconds (4.1 percent) executed the data-flow analysis. The financial and decision steps at 0.137 and 0.001 seconds respectively are negligible because they are pure computation over the stored pillar outputs.

Step	Seconds
index	66.437
pillars	321.807

Step	Seconds
deep_index	10.293
taint	17.258
financial	0.137
decision	0.001

Appendix B — Metric Definitions & Financial Calibration

Metric definitions used throughout the narrative are fixed and target-agnostic. Cyclomatic complexity is the count of linearly independent paths through a symbol, measured per symbol by the static analyzer. Cognitive complexity is a human-readability-weighted complexity score that accounts for nesting, recursion, and other readability factors. Bus factor is the reciprocal of sole-contributor concentration per module; a low bus factor flags flight risk. Debt density is normalised per thousand lines of code. Remediation hours-per-unit (HPU) per cost category is calibrated by the financial 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.

The confidence policy formula is base 20 + coverage weight 40 times coverage fraction + risk signal weight 40 times risk signal fraction, capped at 20 when coverage is 0 and floored at 60 for the no-go tier. The valuation discount curve is 25 times (1 minus exp of negative raw_discount over max of 80 and total_cost divided by 50000), with a half-life that scales as max of 80 and cost divided by 50000. The integration drag curve is 80 times (1 minus exp of negative raw_drag over max of 200 and total_hours divided by 10), with a half-life that scales as max of 200 and hours divided by 10. The aggregate hours cap is 20,000 with a surplus factor of 0.25, meaning that hours above the cap are credited at 25 percent of their full value. These curves are deterministic; the same inputs always produce the same outputs.

The decision model defines six deal-breaker pillars (ai_ml, attack_surface, cve, iac, license, secrets) where a critical rating forces the no-go verdict. The severity risk weights are informational 0, low 1, medium 2, high 3, and critical 4, which are the weighting factors applied to a finding's severity inside the risk signal computation. The schedule constants for the post-close roadmap are 2 execution teams of 2 engineers each, 10 workdays per sprint, 12-sprint cap, 4 hours per workday, and minimum and maximum workstream durations of 7 and 120 days respectively. The health-posture ceilings are 15 for critical severity, 40 for high, 65 for medium, 85 for low, unbounded for informational, and 50 for unassessed pillars, ensuring that incomplete coverage never reads as a strong pillar.

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 therefore a different report under a new fingerprint. The full calibration, lineage, and sensitivity detail lives in Section 3 and is not re-rendered here. The

© MTD Services

fingerprint allows the buyer's quality-of-earnings reviewer to challenge the configuration rather than the headline figure: if the reviewer disputes the remediation cost, the reviewer can identify the input that drives the dispute and the fingerprint change that would result.

O1 — Reproduction Fingerprint

Fingerprint Type	Value
Economic model fingerprint (0.3.0)	4ebb9e81bd28dfea62f425c041536e90ad259855f449197be957c76c84fd78b3
Reproduction fingerprint (O1)	91389600bee7c997e58e34f03b51d61cae26606b6fce2df44f9f95e4363dc149

Appendix P — Evidence Provenance

The evidence provenance appendix maps every evidence item in the report to its source in the stored payload and to its corresponding sheet in the workbook. 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. The buyer's quality-of-earnings reviewer can use this appendix to trace any figure back to its raw evidence.

P.1 — Executive Decision Provenance

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 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
1.14	Integrity error count	integrity.error_count	Contents
1.15	Integrity warning count	integrity.warn_count	Contents

P.2 — Transaction Recommendation Provenance

ID	Evidence	Source	Workbook
2.1	Conditions for acquisition	decision.conditions_for_acquisition	Dashboard
2.2	Deal-breaker register	decision.deal_breakers	Dashboard
2.3	Categorical decision tier	decision.escalation.categorical_tier	Dashboard

© M&D Services

ID	Evidence	Source	Workbook
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
2.7	Economic gate disputed	decision.escalation.disputed	Dashboard
2.8	Escalation driver	decision.escalation.driver	Dashboard
2.9	Per-pillar escalation rows	decision.escalation.pillars	Dashboard
2.10	Economic conditions threshold (model)	financial.assumptions.economic_escalation.conditions_if_above_pct	Dashboard
2.11	Economic no-go threshold (model)	financial.assumptions.economic_escalation.no_go_if_above_pct	Dashboard
2.12	Sensitivity sweep outcomes	financial.sensitivity	Dashboard
2.13	Condition deal-mechanic tags	decision.condition_details	Dashboard

P.3 — Financial Model Provenance

ID	Evidence	Source	Workbook
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
3.9	Integration drag factor	financial.assumptions.integration_drag_factor	Dashboard
3.10	Productive hours per quarter	financial.assumptions.productive_hours_per_quarter	Dashboard
3.11	Person-years of remediation	financial.assumptions.person_years	Dashboard
3.12	Remediation cost CI	financial.assumptions.remediation_cost_confidence_interval	Dashboard
3.17	Aggregate hours cap	financial.assumptions.aggregate_cap	Dashboard
3.18	Saturation model	financial.assumptions.saturation_model	Dashboard
3.19	Line-item basis	financial.assumptions.line_item_basis	Dashboard
3.20	SQALE remediation basis	financial.assumptions.sqale_remediation_basis	Dashboard
3.21	Sensitivity sweep details	financial.sensitivity	Dashboard
3.22	Financial lineage	financial.lineage	Dashboard
3.24	Economic model fingerprint	financial.economic_layers.economic_layers_fingerprint	Dashboard
3.25	Risk valuation lines	financial.economic_layers.layer2_risk.lines	Dashboard
3.26	Aggregate expected loss before	financial.economic_layers.layer2_risk.aggregate_expected_loss_before	Dashboard
3.27	Aggregate residual expected loss	financial.economic_layers.layer2_risk.aggregate_residual_expected_loss	Dashboard
3.28	Aggregate risk-reduction value	financial.economic_layers.layer2_risk.aggregate_risk_reduction_value	Dashboard
3.31	Scenario branches	financial.economic_layers.layer3_scenario.branches	Dashboard
3.32	Weighted expectation	financial.economic_layers.layer3_scenario.weighted_expectation	Dashboard
3.34	Deal components	financial.economic_layers.layer4_deal.components	Dashboard
3.35	Executive deal-bridge rows	financial.economic_layers.layer4_deal.bridge	Dashboard
3.36	Double-counting ledger	financial.economic_layers.p0_double_counting.figures	Dashboard
3.37	Double-counting clean verdict	financial.economic_layers.p0_double_counting.clean	Dashboard
3.42	Deal economics headline	financial.economic_layers.layer4_deal	Dashboard

© Mirantis Services

P.4 — Pillar Evidence Provenance (Abbreviated)

The remaining pillar provenance items (4.x through 15.x) follow the same structure: each ID maps to a JSON path in the stored payload and to a workbook sheet that inventories the same evidence. The full table contains 158 entries spanning the 22 pillars; the abbreviated table below shows representative items. The full table is available in the workbook under the Provenance sheet.

ID	Evidence	Source	Workbook
4.1	License findings inventory	pillars.license.raw.findings	License Contamination
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
6.1	Direct dependency count	pillars.dependency.summary.direct_dependency_count	Dependency Packages
7.1	Symbols analyzed	pillars.complexity.summary.total_symbols_analyzed	Complex Symbols
7.15	Hotspot symbols	pillars.complexity.raw.complexity.symbols	Complex Symbols
8.1	Modules analyzed (test deficit)	pillars.test_deficit.summary.total_modules	Test Deficit
9.1	Circular-dependency cycles	pillars.entanglement.summary.total_cycles	Entanglement Violations
9.7	Cross-layer violation detail	pillars.entanglement.raw.cross_layer_violations	Entanglement Violations
10.5	Key-person risk detail	pillars.key_person.raw.risks	Key-person Risks
11.12	Document inventory	pillars.doc_staleness.raw.documents	Stale Docs
12.1	AI/ML source files scanned	pillars.ai_ml.summary.source_files_scanned	AI/ML Models
13.1	Technologies total	pillars.tech_stack.summary.total_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
15.16	Out-of-scope evidence register	pillars (synthesis)	Not applicable

Appendix R — Contradiction Ledger & Release Record

Every material conflict is listed with its status, resolution, and decision impact. Unresolved entries discount the confidence stated in the executive brief. The contradiction ledger contains a single resolved entry in this scan, which is why the stated confidence of 76 is not discounted by contradiction concerns.

R.1 — 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

© MTDI Labs

R.2 — Release Record

Field	Value
Report ID	TDD-20261016-060115
Generated	2026-10-16T06:08:13.222428+00:00
Report version	0.9.2
Ledger summary	1 resolved
Scan steps	6 recorded

Appendix C — Dependency & Lockfile Inventory

The direct dependency inventory contains 329 packages across the Go ecosystem, all ranged (no pinned versions) and all sourced from go.mod manifests. The full inventory is reproduced in the workbook under the Dependency Packages sheet. A representative sample of 30 dependencies is shown below to illustrate the breadth of the supply chain.

C.1 — Direct Dependency Inventory (Sampled)

Name	Specified	Pinned	Manifest
cel.dev/expr	v0.25.2	False	go.mod
cloud.google.com/go	v0.123.0	False	go.mod
cloud.google.com/go/auth	v0.18.2	False	go.mod
cloud.google.com/go/compute/metadata	v0.9.0	False	go.mod
cloud.google.com/go/iam	v1.5.3	False	go.mod
cloud.google.com/go/storage	v1.61.3	False	go.mod
github.com/aws/aws-sdk-go-v2	v1.41.5	False	go.mod
github.com/aws/aws-sdk-go-v2/config	v1.32.12	False	go.mod
github.com/aws/aws-sdk-go-v2/credentials	v1.19.12	False	go.mod
github.com/aws/aws-sdk-go-v2/service/s3	v1.97.3	False	go.mod
github.com/aws/aws-sdk-go-v2/service/sts	v1.41.9	False	go.mod
github.com/Azure/go-autorest	v14.2.0+incompatible	False	internal/backend/remote-state/azure/go.mod
github.com/Azure/go-autorest/autorest	v0.11.30	False	internal/backend/remote-state/azure/go.mod
github.com/hashicorp/cli	v1.1.7	False	go.mod
github.com/hashicorp/consul/api	v1.32.1	False	go.mod
github.com/hashicorp/go-getter	v1.8.6	False	go.mod
github.com/hashicorp/go-hclog	v1.6.3	False	go.mod
github.com/hashicorp/go-plugin	v1.7.0	False	go.mod
github.com/hashicorp/hcl/v2	v2.24.0	False	go.mod
github.com/hashicorp/terraform-plugin-go	v0.26.0	False	go.mod
github.com/hashicorp/terraform-plugin-sdk/v2	v2.36.1	False	go.mod
github.com/hashicorp/terraform-svchost	v0.2.1	False	go.mod
github.com/AlecAivazis/survey/v2	v2.3.7	False	go.mod
github.com/aliyun/alibaba-cloud-sdk-go	v1.61.1501	False	go.mod
github.com/antchfx/xmlquery	v1.3.5	False	go.mod
github.com/apparentlymart/go-cidr	v1.1.0	False	go.mod

© Audit Services

Name	Specified	Pinned	Manifest
github.com/aws/smithy-go	v1.24.2	False	go.mod
github.com/go-jose/go-jose/v4	v4.1.4	False	go.mod
github.com/golang-jwt/jwt/v4	v4.5.2	False	go.mod
golang.org/x/crypto	v0.57.0	False	go.mod

C.2 — Lockfile Resolution

Lockfile	Resolved Packages
go.sum	979
internal/backend/remote-state/azure/go.sum	365
internal/backend/remote-state/consul/go.sum	427
internal/backend/remote-state/cos/go.sum	269
internal/backend/remote-state/gcs/go.sum	265
internal/backend/remote-state/kubernetes/go.sum	361
internal/backend/remote-state/oci/go.sum	300
internal/backend/remote-state/oss/go.sum	293
internal/backend/remote-state/pg/go.sum	242
internal/backend/remote-state/s3/go.sum	267
internal/legacy/go.sum	105

Appendix D — CVE Inventory

The CVE inventory contains 18 advisories, all at UNKNOWN severity and all with no fixed release available. The full inventory is reproduced below in alphabetical order by advisory ID.

ID	Package	Version	Fixed Available
GHSA-4f99-4q7p-p3gh	github.com/sirupsen/logrus	v1.2.0	False
GHSA-55v3-xh23-96gh	github.com/cli/go-gh	v2.13.0	False
GHSA-6q6q-88xp-6f2r	gopkg.in/yaml.v2	v2.2.1	False
GHSA-h3p9-wrgx-82cm	github.com/hashicorp/terraform	0.0.0	False
GHSA-r88r-gmrh-7j83	gopkg.in/yaml.v2	v2.2.1	False
GHSA-wxc4-f4m6-wwqv	gopkg.in/yaml.v2	v2.2.1	False
GO-2020-0036	gopkg.in/yaml.v2	v2.2.1	False
GO-2021-0061	gopkg.in/yaml.v2	v2.2.1	False
GO-2022-0839	github.com/hashicorp/terraform	0.0.0	False
GO-2022-0956	gopkg.in/yaml.v2	v2.2.1	False
GO-2024-3295	github.com/cli/go-gh	v2.13.0	False
GO-2025-3485	github.com/go-jose/go-jose	v4.1.4	False
GO-2025-3553	github.com/golang-jwt/jwt	v4.5.2	False
GO-2025-3787	github.com/go-viper/mapstructure	v2.4.0	False
GO-2025-3900	github.com/go-viper/mapstructure	v2.4.0	False
GO-2025-4188	github.com/sirupsen/logrus	v1.2.0	False
GO-2026-5320	github.com/yuin/goldmark	v1.1.27	False

© Orbis Services

ID	Package	Version	Fixed Available
GO-2026-5932	golang.org/x/crypto	v0.57.0	False

Appendix E — License Findings Inventory

The license findings inventory contains 870 entries. Three are repo-level GPL-2.0 files with viral and restrictive flags; the remaining 867 are UNKNOWN licenses from go.mod files. A representative sample of 12 entries is shown below; the full inventory is in the workbook under the License Contamination sheet.

Package	Detected License	Is Viral	Is Restrictive	Confidence	Source File
(repo-level)	GPL-2.0	True	True	0.8	docs/plugin-protocol/LICENSE
(repo-level)	GPL-2.0	True	True	0.8	internal/tfplugin5/LICENSE
(repo-level)	GPL-2.0	True	True	0.8	internal/tfplugin6/LICENSE
cel.dev/expr	UNKNOWN	False	False	0.0	go.mod
cloud.google.com/go	UNKNOWN	False	False	0.0	go.mod
cloud.google.com/go/auth	UNKNOWN	False	False	0.0	go.mod
github.com/agext/levenshtein	UNKNOWN	False	False	0.0	go.mod
github.com/AlecAivazis/survey/v2	UNKNOWN	False	False	0.0	go.mod
github.com/aliyun/alibaba-cloud-sdk-go	UNKNOWN	False	False	0.0	go.mod
github.com/antchfx/xmlquery	UNKNOWN	False	False	0.0	go.mod
github.com/apparentlymart/go-cidr	UNKNOWN	False	False	0.0	go.mod
github.com/aws/aws-sdk-go-v2	UNKNOWN	False	False	0.0	go.mod

Appendix F — Complexity, Duplication & SQALE Detail

This appendix provides the detailed complexity hotspot inventory (top 30 symbols), the duplication groups (sampled), and the SQALE debt modules (top 20 modules). The full inventory of 13,922 symbols is available in the workbook under the Complex Symbols sheet.

F.1 — Complexity Hotspot Symbols (Top 30)

Symbol	File	Cyclo	Cog	MI	Composite Risk
TestContextPlan_actions	internal/terraform/context_plan_actions_test.go	317	409	0	0.9125
EnsureProviderVersions	internal/providercache/installer.go	127	256	0	0.9000
TestEnsureProviderVersions	internal/providercache/installer_test.go	104	145	0	0.9125
TestContext2Plan_PolicyEvaluation	internal/terraform/context_plan_policy_test.go	102	117	0	0.9125
opApply	internal/backend/local/backend_apply.go	99	232	0	0.9000
plan	internal/terraform/node_resource_abstract_instance.go	98	160	0	0.9000

© MTD Services

Symbol	File	Cyclo	Cog	MI	Composite Risk
validateProviderConfigs	internal/configs/provider_validation.go	95	191	0	0.9375
backendFromConfig	internal/command/meta_backend.go	78	179	0.92	0.8982
GetResource	internal/terraform/evaluate.go	77	152	3.87	0.8923
TestContextApply_actions	internal/terraform/context_apply_action_test.go	73	178	0	0.9125
apply	internal/terraform/node_resource_abstract_instance.go	73	89	4.98	0.8900
Run	internal/command/state_mv.go	72	91	5.23	0.8895
Run	internal/command/providers_lock.go	71	96	10.48	0.8790
run	internal/command/init_run.go	70	102	3.59	0.8928
plan	internal/backend/remote/backend_plan.go	69	65	14.80	0.8704
TestPrepareFinalInputVariableValue	internal/terraform/eval_variable_test.go	67	76	0	0.9125
AddRaw	internal/stacks/stackplan/from_proto.go	67	78	13.25	0.8735
managedResourceExecute	internal/terraform/node_resource_plan_instance.go	65	86	6.11	0.8878
Configure	internal/backend/remote-state/s3/backend.go	62	119	12.34	0.8753
Run	internal/command/providers_mirror.go	62	154	11.51	0.8770
opApply	internal/backend/remote/backend_apply.go	61	87	15.81	0.8684
DirFromModule	internal/initwd/from_module.go	60	154	8.94	0.9821
TestEvalVariableValidation	internal/stacks/stackruntime/internal/stackeval/input_variable_test.go	59	37	10.51	0.8915
getType	internal/stacks/stackconfig/typeexpr/typeexpr.go	58	132	14.12	0.9218
ProviderSchema	internal/schemarepo/loadschemas/plugins.go	58	48	33.39	0.7947
TestInit_stateStore_newWorkingDir_interactiveProviderApproval	internal/command/init_test.go	57	54	6.14	0.8802
ApplyPlan	internal/stacks/stackruntime/internal/stackeval/main_apply.go	56	114	13.21	0.9361
assertPlanValid	internal/plans/objchange/plan_valid.go	56	164	21.81	0.9189
Same	internal/legacy/terraform/diff.go	56	80	21.01	0.8580
installRegistryModule	internal/initwd/module_install.go	55	151	12.79	0.8744

F.2 — SQALE Debt Modules (Sampled)

Base Lines	Debt Ratio	Module	Remediation Effort	SQALE Grade
10	0	internal/command/jsonformat/computed/doc.go	0	A
10	0	internal/initwd/doc.go	0	A
10	0	internal/moduledeps/doc.go	0	A
10	0	internal/plans/deferring/doc.go	0	A
10	0	internal/plans/planproto/doc.go	0	A
10	0	internal/registry/response/module_list.go	0	A
10	0	internal/releaseauth/doc.go	0	A
10	0	internal/terraform/features.go	0	A
10	0	internal/terraform/ui_output.go	0	A

Base Lines	Debt Ratio	Module	Remediation Effort	SQALE Grade
10	0	scripts/exhaustive.sh	0	A
100	0	internal/addrs/address_expr.go	0	A
100	0	internal/cloud/retry_test.go	0	A
100	0	internal/command/arguments/unlock_test.go	0	A
100	0	internal/dag/tarjan.go	0	A
100	0	internal/lang/blocktoattr/fixup_bench_test.go	0	A
100	0	internal/lang/scope.go	0	A
100	0	internal/stacks/stackmigrate/meta.go	0	A
100	0	internal/stacks/stackstate/statekeys/key_type.go	0	A
100	0.06	internal/getproviders/reattach/reattach.go	6	B
1,001	0	internal/command/import_test.go	0	A

Appendix G — Debt Marker Inventory

No debt markers were recorded in this scan. The debt marker scanner queried the codebase for annotations such as TODO, FIXME, HACK, and other technical-debt markers, and returned zero matches. The clean result should be read with caution as noted in Section 7.3: a zero debt-marker count can indicate either that the codebase carries no annotated technical debt or that the engineering team does not annotate technical debt in source. The buyer should clarify which interpretation applies during commercial diligence.

Appendix H — Attack Surface Candidate Endpoints

The attack surface scanner identified 59 candidate entry points. A representative sample of 20 is shown below; the full inventory is in the workbook under the Attack Surface sheet. All 59 candidates carry an unknown exposure flag (the scanner could not determine external reachability) and a low risk level. The test status column distinguishes tested (covered by an existing test) from untested (no test coverage).

Symbol	File	Line	Type	Risk	Test Status
main	main.go	53	candidate	low	tested
init	checkpoint.go	18	candidate	low	tested
init	internal/backend/remote-state/inmem/backend.go	32	candidate	low	tested
init	internal/cloudplugin/cloudproto1/cloudproto1.pb.go	215	candidate	low	untested
init	internal/collections/cmp.go	23	candidate	low	tested
init	internal/command/telemetry.go	13	candidate	low	tested
main	internal/command/testdata/login-oauth-server/main.go	26	candidate	low	untested
init	internal/command/testdata/login-oauth-server/oauthserver.go	177	candidate	low	untested
init	internal/command/testdata/login-tfe-server/tfeserver.go	79	candidate	low	untested
main	internal/command/testdata/statelocker.go	18	candidate	low	untested
init	internal/command/ui/ui_input.go	236	candidate	low	tested
init	internal/configs/config_build.go	149	candidate	low	tested
init	internal/dag/graph.go	255	candidate	low	tested
init	internal/depsfile/testing.go	18	candidate	low	tested
init	internal/experiments/experiment.go	30	candidate	low	untested

© MTD Services

Symbol	File	Line	Type	Risk	Test Status
init	internal/getproviders/registry_client.go	56	candidate	low	tested
init	internal/legacy/helper/schema/resource_data.go	426	candidate	low	tested
init	internal/logging/logging.go	55	candidate	low	tested
init	internal/plans/planproto/planfile.pb.go	2606	candidate	low	untested
main	tools/protobuf-compile/protobuf-compile.go	229	candidate	low	untested

Appendix I — Key-Person & Knowledge Loss Vulnerability Detail

This appendix provides the detailed key-person risk rows (top 20 by risk score), module bus-factor rows (top 20), and knowledge loss vulnerability rows (top 18 critical-tier). The full inventories are in the workbook under the Key-person Risks and Knowledge Loss Vulnerability sheets.

I.1 — Key-Person Risk Rows (Top 20)

Author Email	Risk Score	Total Commits	Departure Impact
radek.simko@gmail.com	0.5394	2,176	High
mart@degeneration.co.uk	0.4265	81,053	Medium
110428419+hashicorp-copywrite[bot]@users.noreply.github.com	0.4137	4,527	Medium
j.bardin@gmail.com	0.3095	48,111	Medium
15078782+SarahFrench@users.noreply.github.com	0.3080	32,733	Medium
liam.cervante@hashicorp.com	0.2857	75,115	Low
dbanck@users.noreply.github.com	0.2304	10,136	Low
danielmschmidt92@gmail.com	0.1877	20,824	Low
40821565+dsa0x@users.noreply.github.com	0.1559	21,901	Low
sebastian.rivera@hashicorp.com	0.1538	4,763	Low
alisdair@users.noreply.github.com	0.1363	10,479	Low
nick.fagerlund@gmail.com	0.1153	2,671	Low
chris@chrisarcand.com	0.1124	3,493	Low
radeksimko@users.noreply.github.com	0.1108	6,074	Low
brandon.croft@gmail.com	0.1074	4,084	Low
jarrett.spiker@hashicorp.com	0.1059	118	Low
jfinnigan@hashicorp.com	0.0909	138	Low
glennsarti@users.noreply.github.com	0.0879	467	Low
karlkrch@gmail.com	0.0876	53	Low
barrett.clark@hashicorp.com	0.0710	1,096	Low

I.2 — Module Bus-Factor Rows (Top 20)

Module Path	Bus Factor	Total Files
internal/command	6	132
internal/cloud	5	46
internal/backend/remote	4	21
internal/command/views	4	56
internal/configs	4	63
internal/configs/configload	4	9

© Mirrorservices

Module Path	Bus Factor	Total Files
internal/command/views/json	3	18
internal/initwd	3	6
internal/plans	3	18
internal/plans/planfile	3	7
internal/plugin6	3	5
internal/plugin6/mock_proto	3	2
internal/terraform	3	244
internal/tfdiags	3	30
internal/tfplugin6	3	2
scripts	3	12
version	3	3
_unassigned	2	2,080
internal/addrs	2	74
internal/backend/backendrun	2	9

I.3 — Knowledge Loss Vulnerability (Critical Tier)

Author Name	Tenure (days)	Days Since Last Commit	Blast Radius
Sarah French	1,561	3	73
Samsondeen	673	2	26
kmoe	2,629	104	15
Austin Valle	570	1	10
Brandon Croft	1,833	197	10
Daniel Banck (daniel@dbanck.de)	622	2	8
Daniel Schmidt	3,446	9	67
Daniel Banck (dbanck@)	1,368	22	18
James Bardin	3,797	6	93
Mark DeCrane	960	7	4
Matej Risek	1,049	36	3
Kristin Laemmert	3,046	17	42
Radek Simko (radek.simko@)	4,241	114	137
Radek Simko (radeksimko@)	983	2	7

Appendix J — Maintenance, IaC & CI/CD Detail

This appendix reproduces the maintenance assets (none recorded), the IaC findings (2 findings, both Container runs as root), and the CI files (11 GitHub Actions workflows). The maintenance asset inventory is empty because the asset-staleness scanner did not execute on any files, as noted in Section 11.1.

J.1 — Maintenance Assets

No maintenance assets were recorded in this scan. The maintenance scanner did not execute on any files, so the maintenance asset inventory is empty. The buyer should commission an independent maintenance scan before close or accept that the maintenance posture remains unverified.

© M&A Services

J.2 — IaC Findings (Full)

Rule ID	Title	Severity	File	Line	Remediation
IAC-DOCKER-ROOT	Container runs as root	Medium	Dockerfile	26	Add a dedicated non-root user and USER directive in every stage.
IAC-DOCKER-ROOT	Container runs as root	Medium	build.Dockerfile	59	Add a dedicated non-root user and USER directive in every stage.

J.3 — CI Files (Full)

#	Workflow File
1	.github/workflows/backport.yml
2	.github/workflows/build-terraform-cli.yml
3	.github/workflows/build.yml
4	.github/workflows/changelog-validation.yml
5	.github/workflows/checks.yml
6	.github/workflows/enforce-changelog.yml
7	.github/workflows/equivalence-test-diff.yml
8	.github/workflows/equivalence-test-manual-update.yml
9	.github/workflows/equivalence-test-update.yml
10	.github/workflows/issue-comment-created.yml
11	.github/workflows/lock.yml

J.4 — Coverage Artifacts

No coverage artifacts were recorded. The CI does not surface coverage metrics (reported_coverage_pct is not quantified), and no coverage artifacts were detected by the scanner. The buyer should commission an independent runtime coverage analysis before close to validate the static symbol-share proxy used in Section 8.

Appendix K — Privacy & Secrets Detail

This appendix reproduces the privacy findings (6 findings) and the secret candidates (12 candidates) in full. The privacy findings include 1 plaintext credential and 5 sensitive logging statements; the secret candidates include 1 certificate block, 2 connection strings, and 9 private key blocks.

K.1 — Privacy Findings (Full)

Rule ID	Category	Title	Severity	File	Line
PRIV-PLAINTEXT-CRED	Plaintext credential	Plaintext credential material in committed config	Medium	internal/cloud/testdata/apply-json-with-outputs/plan-redacted.json	51
PRIV-SENSITIVE-LOG	Sensitive logging	Sensitive data in log statement (token)	Medium	internal/backend/remote-state/azure/api_client.go	159
PRIV-SENSITIVE-LOG	Sensitive logging	Sensitive data in log statement (secret)	Medium	internal/backend/remote-state/kubernetes/client.go	411

© MTD Services

Rule ID	Category	Title	Severity	File	Line
PRIV-SENSITIVE-LOG	Sensitive logging	Sensitive data in log statement (apikey)	Medium	internal/backend/remote-state/oci/backend.go	185
PRIV-SENSITIVE-LOG	Sensitive logging	Sensitive data in log statement (apikey)	Medium	internal/backend/remote-state/oci/backend_test.go	341
PRIV-SENSITIVE-LOG	Sensitive logging	Sensitive data in log statement (token)	Medium	internal/backend/remote-state/s3/backend_complete_test.go	1515

K.2 — Secret Candidates (Full)

Kind	File	Line	Confidence	In History	Risk Cap	Path Tier
Certificate block	internal/communicator/winrm/provisioner_test.go	78	79	False	High	Prod
Connection string	internal/backend/remote-state/pg/backend.go	33	74	False	Medium	Prod
Connection string	internal/backend/remote-state/pg/backend_test.go	37	76	False	Medium	Prod
Private key block	internal/backend/remote-state/oci/backend_test.go	320	99	True	High	Prod
Private key block	internal/communicator/ssh/communicator_test.go	493	99	True	High	Prod
Private key block	internal/communicator/ssh/communicator_test.go	464	99	True	High	Prod
Private key block	internal/communicator/ssh/communicator_test.go	33	99	False	High	Prod
Private key block	internal/lang/funcs/crypto_test.go	716	99	True	High	Prod
Private key block	internal/lang/funcs/crypto_test.go	687	99	True	High	Prod
Private key block	internal/lang/funcs/crypto_test.go	745	99	True	High	Prod
Private key block	internal/lang/funcs/crypto_test.go	774	99	True	High	Prod
Private key block	internal/lang/functions_test.go	1416	99	True	High	Prod

Appendix L — AI/ML Inventory

No model files were recorded in this scan. The AI/ML scanner identified framework hits (Hugging Face: 14) and SDK hits (Azure OpenAI: 12) but found no bundled model files, no production model files, and no unlicensed model files. The AI/ML exposure is therefore limited to source-code references rather than to actual model deployments.

Appendix M — Integrity & Data-Quality Issues

This appendix reproduces the 9 integrity issues in full. Seven issues are warnings for critical_pillar_no_findings (pillars rated high or critical that render no item-level findings, requiring the reading guide to state the aggregate basis). Two issues are warnings for unverified_pillar (Debt and Maintenance, which are NOT_ASSESSED).

Tier	Code	Subject	Message (abbreviated)
warn	critical_pillar_no_findings	pillars.ci_maturity	Pillar is high but renders no item-level findings
warn	critical_pillar_no_findings	pillars.coverage	Pillar is high but renders no item-level findings
warn	critical_pillar_no_findings	pillars.cve	Pillar is high but renders no item-level findings
warn	critical_pillar_no_findings	pillars.entanglement	Pillar is critical but renders no item-level findings
warn	critical_pillar_no_findings	pillars.ip_ownership	Pillar is high but renders no item-level findings
warn	critical_pillar_no_findings	pillars.secrets	Pillar is high but renders no item-level findings
warn	critical_pillar_no_findings	pillars.test_deficit	Pillar is high but renders no item-level findings
warn	unverified_pillar	pillars.debt	Pillar is not_assessed; zero evidence population

© MTD Services

Tier	Code	Subject	Message (abbreviated)
warn	unverified_pillar	pillars.maintenance	Pillar is not_assessed; no files scanned by asset-staleness scanner

Appendix N — Taint Analysis Summary

The taint analysis executed on the indexed graph and identified 0 taint flows. The taint pass was not bounded, meaning the analysis was allowed to explore the full call graph rather than being limited to a depth or scope window. The zero-flow result is a clean read on the taint dimension specifically and does not constitute a clean bill of health on the broader attack surface (see Section 5.5 for the attack-surface composition).

Metric	Value
Bounded	False
Total Flows	0

Appendix Q — Glossary

Terms are used consistently throughout this report and never redefined per section. The glossary below defines, in plain language, every technical or specialist term that appears in any TDD table column heading. Definitions are alphabetically sorted and consistent with contextual usage. The buyer and its advisers should use this glossary to converge on a shared frame of reference when interpreting the report.

Term	Meaning
ACQUIRE_WITH_CONDITIONS	Verdict tier requiring specified conditions to be satisfied, escrowed, or warranted before close; sits between ACQUIRE and DO_NOT_ACQUIRE on the decision scale.
Active Contributor	An author whose last commit falls within the active window (typically 365 days). Active contributor count is 73 of 2,375 total authors.
Advisory	A published vulnerability report (CVE, GHSA, or GO advisory) that identifies a security issue in a specific package version.
Aggregate Hours Cap	A model parameter (20,000 hours) that limits total remediation hours; hours above the cap are credited at the surplus factor (0.25).
API Contract	A formal specification of an API's endpoints, request/response schemas, and authentication. Detected from OpenAPI, GraphQL, or proto files.
Assessment State	Five-value trust vocabulary: NOT_ASSESSED, PARTIALLY_ASSESSED, ASSESSED_NO_FINDINGS, ASSESSED_WITH_FINDINGS, ERROR.
Assessment Status	Pillar-level coverage: assessed, partial, or not_assessed. Only assessed supports conclusions.
Attack Surface	The set of candidate entry points (init/main symbols) that could be reached by adversarial input.
Attribution	The process of assigning file authorship to a corporate or external domain based on git history.
Blast Radius	The number of modules affected by a single contributor's departure; used in the knowledge loss vulnerability calculation.
Bus Factor	Reciprocal of sole-contributor concentration per module; a low bus factor flags flight risk.
Calibration Factor	A multiplier applied to the hours-per-unit of a cost category to reflect difficulty; default value 1 (no calibration).
Candidate Endpoint	A symbol (init or main) identified by the attack-surface scanner as a potential entry point; risk level is typically low pending reachability confirmation.

© M&A Due Diligence Services

Term	Meaning
Candidate Exposure Score	An upper-bound signal for review prioritisation, computed from candidate count and unreferenced fraction; not a confirmed vulnerability rate.
CC Percentile	Cyclomatic complexity percentile distribution (P50, P90, P99) showing the spread of complexity across symbols.
CI Maturity	A pillar assessing the existence and strictness of continuous integration gates; measured from CI workflow files.
Cognitive Complexity	A human-readability-weighted complexity score that accounts for nesting, recursion, and other readability factors.
Complexity Multiplier	A model parameter (1.5) that scales hours-per-unit for critical-complexity hotspots.
Condition Precedent	A deal mechanic requiring satisfaction before close; the deal does not sign as structured if the condition is unmet.
Confidence Score	Model-derived score reflecting coverage and risk signal; not a probability of correctness. Computed as base 20 + coverage*40 + risk_signal*40.
Contamination Chain	A chain by which a viral license in a transitive dependency imposes its terms on the packaged binary; zero chains detected in this scan.
Coverage Method	The technique used to measure test coverage; this scan uses a static symbol-share proxy, not runtime or line/branch coverage.
Coverage Ratio	The share of symbols in a module covered by test files; ranges 0 to 1.
Critical Sole-Maintainer Module	A module where a single contributor holds the institutional knowledge; 124 modules identified in this scan.
Cross-Layer Violation	A direct call or import from a higher architectural layer to a lower layer, breaking layer separation.
CVE	Common Vulnerabilities and Exposures; a published advisory identifier. Sources include OSV, NVD, GHSA, and GO advisory databases.
Cyclomatic Complexity	The count of linearly independent paths through a symbol; higher values indicate more branching logic.
Deal-Breaker Pillar	One of six pillars (ai_ml, attack_surface, cve, iac, license, secrets) where a critical rating forces the no-go verdict.
Debt Marker	An annotation in source code (TODO, FIXME, HACK) flagging technical debt for future remediation; zero detected in this scan.
Debt Ratio	The share of development effort required to remediate all identified code smells, expressed as a percentage.
Dependency Pinning	Locking a dependency to a specific version rather than a range; pin ratio is 0.0 percent in this scan.
Detected, not verified	Evidence state indicating the scanner found evidence consistent with a risk pattern but the risk has not been independently validated.
Double-Counting Ledger	An audit mechanism that verifies no quantified economic figure is counted twice across model layers; clean verdict is true in this scan.
Duplication Group	A set of code blocks identified by the duplication scanner as appearing in multiple locations; 4,549 groups detected.
Effective Insured Value	The value of code that is both high-risk and tested; zero in this scan means no high-value code carries test coverage.
EOL	End-of-life; technology or runtime that is no longer supported by its vendor. Reconciled across tech_stack and eol pillars.
Entanglement Score	A 0-to-1 score measuring structural coupling; 1.0 (maximum) in this scan, driven by the 163-module cycle and 852 cross-layer violations.
Escalation Threshold	A percentage of proposed price at which the conditions gate (5 percent) or no-go gate (25 percent) triggers.
Evidence State	Per-finding honesty label: Confirmed, Detected (not verified), Inferred, Partially assessed, Unassessed, Conflicting, or Unsupported.
Finding ID	Stable identifier F-pillar-rank for one assessed finding; cited by narrative, register, workbook, and ledger.
Framework Hit	A reference to an AI/ML framework (e.g., Hugging Face) detected in source code; 14 hits in this scan.
GPL-2.0	GNU General Public License version 2.0; a viral and restrictive copyleft license detected in 3 repo-level files.
Headline EV	Headline enterprise value of the transaction; USD 10,050,000 in this scan.

©NDASIGHTS

Term	Meaning
High-Confidence Secret	A secret candidate detected at confidence >= 95; 3 detected in this scan.
High-Value Gap	A module with zero test coverage that carries a high structural risk weight; 50 identified.
HPU	Hours per unit (per cost category), saturated by cap/decay under the financial model.
IaC	Infrastructure as Code; configuration files (Dockerfile, Terraform, Kubernetes manifests) checked by the IaC scanner.
Indexed Symbol	A symbol (function, method, type) discovered by the static analyzer during indexing; 2,056 indexed in this scan.
Inferred	Evidence state for findings derived from indirect evidence (e.g., git history) rather than direct code inspection.
Integration Drag	Post-merger friction cost, computed as hours / (8 x 260 x integration_drag_factor); capped at 80 percent.
Integration Risk Score	A 0-to-5 score assessing API contract, database coupling, and serialization risk; 4.33 in this scan.
Integrity Status	Model health check: warn if any integrity issues are present, error if errors are present. Warn in this scan.
IP Risk Score	A 0-to-1 score reflecting external authorship concentration; 0.6497 in this scan.
Knowledge Loss Vulnerability	Risk that institutional knowledge is concentrated in contributors who may depart; score 0.5052.
Lockfile Resolution	The number of transitive packages resolved by a lockfile (e.g., go.sum); 979 packages resolved in the root go.sum.
Maintenance Backlog Score	A 0-100 score aggregating file staleness, complexity, and neglect; not computable in this scan (NOT_ASSESSED).
Materiality Floor	Absolute dollar threshold the conditions gate (USD 250,000) or no-go gate (not quantified) must also clear.
No-Go Threshold	Percentage of price (25 percent) at which remediation triggers a no-go verdict.
OSV	Open Source Vulnerabilities database; the advisory source used for CVE queries in this scan (osv.dev).
Out-of-Scope Evidence	A category of question that a static, offline repository scan cannot answer; eight categories listed in Section 15.3.
P50 / P90 / P99	Percentile values showing the distribution of a metric (e.g., cyclomatic complexity) across the symbol population.
Pillar	One of 22 assessed dimensions in the model (e.g., complexity, coverage, cve, secrets).
Pin Ratio	Pinned dependencies divided by total dependencies; 0.0 percent in this scan.
Post-Merge Integration Drag	A modelled friction cost (capped at 80 percent) representing the integration burden inherited by the acquirer.
Price-Adjustment Lever	A deal mechanic for a funded or escrowed work item that moves value rather than blocking close.
Privacy Finding	A flag from the privacy scanner identifying plaintext credentials, sensitive logging, or PII endpoints.
Proprietary Coupling	A coupling to a proprietary (non-open-source) system; 7 detected in this scan.
Reachability Confidence	The scanner's confidence that a candidate endpoint is reachable from external input; low in this scan.
Remediation Class	A categorization of CVE remediation approach (e.g., no_fixed_release, version_pin, patch).
Remediation Cost	The modelled cost of remediation work, computed as saturated hours multiplied by the hourly rate.
Risk Level	Severity tier: critical, high, medium, low, or informational; demoted to informational under partial coverage.
Risk Reduction Value	The difference between expected loss before and after remediation; USD 287,500 for the DirFromModule line.
Risk Signal	A weighted blend of pillar risk levels used in the confidence formula; 50.0 percent in this scan.
Risk Valuation	Layer 2 of the financial model; pairs each material risk with expected loss before and after remediation.
Saturation Decay	Hours-per-unit applied once the saturation cap is reached; reduces the marginal cost of additional units.
Saturation Model	A per-category cap-and-decay model that limits total hours regardless of unit count.

Term	Meaning
Schema Drift Risk	Risk that API contracts have diverged from implementation; medium in this scan.
SDK Hit	A reference to an AI/ML SDK (e.g., Azure OpenAI) detected in source code; 12 hits in this scan.
Secret Candidate	A pattern match flagged by the secret scanner as potentially containing sensitive material; 12 candidates.
Sensitivity Sweep	A multi-parameter stress test (17 rows) showing verdict invariance across multiplier ranges.
Serialization Risk	Risk arising from unsafe deserialization patterns (eval_call, exec_call, gob_decode).
Severity Risk Weight	Per-severity weighting applied to findings in the risk signal computation: informational 0, low 1, medium 2, high 3, critical 4.
SQALE	Software Quality Assessment based on Lifecycle Expectations; a standardised remediation-effort basis.
SQALE Grade	A-to-E letter grade for code quality; E (second-lowest) in this scan.
Static Symbol-Share Proxy	Coverage method used in this scan: test-file symbols weighted by structural value; not runtime or line/branch coverage.
Surplus Factor	Fraction (0.25) of hours above the aggregate cap that is credited; limits the marginal cost of large remediation scopes.
Taint Flow	A data-flow path from an untrusted source to a sensitive sink; 0 detected in this scan.
Tech Stack	A pillar detecting technologies from manifest and configuration files; 4 technologies detected (Docker, GitHub Actions, Go, Go module checksums).
Test Deficit	A pillar measuring the gap between actual test coverage and a threshold ratio across modules.
Top Author Share	Share of total commits attributable to the most prolific contributor; 28.5 percent in this scan.
Unresolved Package	A package the CVE scanner could not resolve against the advisory database; 0 in this scan.
Unverified Pillar	A pillar in NOT_ASSESSED state; 2 in this scan (debt, maintenance).
Valuation Discount	A percentage reduction to the proposed price, computed from remediation cost and capped at 25 percent; 16.7 percent in this scan.
Viral License	A license (e.g., GPL-2.0) that imposes its terms on derivative works; 3 repo-level findings in this scan.
Warranty / Representation	A deal mechanic for a seller statement whose breach is an indemnity question rather than a work plan.

© MTD Services