
M & A T E C H N I C A L D D

Technology Due Diligence Report

Homeassistant Core — Proposed Acquisition

VERDICT — ACQUIRE WITH CONDITIONS

Target Homeassistant (core repository)

Report ID TDD-20261016-105732

Head Commit def88072397c7a45a854a32d503dbc97cf16d503

Indexed Files 19,579

Generated 16 October 2026

CONFIDENTIAL

For the named buying party and its advisers only

Important Notice and Distribution Restrictions

This Technology Due Diligence Report (the "Report") has been prepared solely for the use of the named buying party and its professional advisers in connection with their evaluation of the proposed acquisition of the target identified on the cover (the "Transaction"). It is confidential, may be legally privileged, and is not for circulation, reproduction or distribution to any third party in whole or in part without the prior written consent of the preparer.

The Report is a deterministic, data-driven projection of a single finalised evidence payload produced from a static, offline-or-network-integrated source scan of the target's repository. Every figure reproduced herein derives from one stored ground truth whose per-query provenance is mapped in Appendix B (Financial Calibration and Reproduction Fingerprint). The financial reproduction fingerprint binds the Report to the exact configuration used to produce every number: the same inputs in, the same outputs out.

Nothing in this Report constitutes legal, tax, accounting, regulatory or investment advice, nor a recommendation to acquire, divest or hold any security. The Report does not purport to identify all risks, exposures or liabilities associated with the target. It addresses only the technology posture supported by the evidence collected within the stated scope, and every conclusion is qualified by the coverage limitations set out in Section 15 (Confidence, Scope Limitations and Material Unknowns) and Appendix A (Methodology).

All findings carry an honesty label indicating whether they are Confirmed, Detected-not-verified, Inferred, Partially assessed, Unassessed, Conflicting or Unsupported. Scanner signals are never auto-confirmed. No finding in this Report has been independently verified by human inspection of the source repository, the production environment, the operational runbook, the contributor agreements or any other artefact outside the scanned source tree; 308 rows are scanner-detected or inferred signals that require human validation before they are relied on in negotiation.

Quantification of remediation, valuation impact and integration drag uses the tool's built-in, unbenchmarked hours-per-unit and saturation caps (calibration.applied = false). The figures should be read as a planning bound for scope and sequencing, not as a priced contractor quote. The size of any escrow, holdback, warranty cap or indemnity ceiling is a qualitative option for deal counsel and is deliberately not asserted as a number anywhere in this Report.

By accepting delivery of this Report, the recipient agrees to maintain its confidentiality, to use it solely for the purpose of evaluating the Transaction, and not to rely on any single figure in isolation without reference to the conventions, qualifications and coverage caveats set out herein.

© MTD Services

Table of Contents

1. Executive Summary and Headline Findings	1
1. Executive Summary and Headline Findings	2
1.1 Verdict and Confidence	2
1.2 Scope, Coverage Gaps and the Confidence Discount	3
1.3 Driver and Pillar Posture	3
1.4 Remediation Economics	4
1.5 Headline Findings	4
1.6 What the Evidence Does Not Support	5
1.7 Deal-Thesis Alignment	5
1.8 Required to Decide	5
2. Transaction Recommendation, Conditions and Escalation Thresholds	6
2.1 Decision Logic	6
2.2 Conditions for Acquisition	6
2.3 Deal-Breaker Register	7
2.4 Escalation Gates	7
2.5 Sensitivity of the Recommendation	7
2.6 Recommendation	8
3. Remediation Cost Model, Valuation Impact and Sensitivity	8
3.1 The Bridge from Technical Findings to Deal Economics	8
3.2 Parameters Used Downstream	9
3.3 Remediation Line Items	9
3.4 Confidence Interval	10
3.5 Valuation Discount and Integration Drag	10
3.6 Sensitivity Sweep	11
3.7 Scenario Branches	11
3.8 Deal Components and the Double-Counting Ledger	12

3.9 Indicative Post-Close Roadmap	12
3.10 Categories Not Costed This Scan	12
4. Licensing and IP Exposure	13
4.1 Why Licensing and IP Drive Title and Warranty Risk	13
4.2 License Posture Summary.....	13
4.3 The UNKNOWN-Licence Population	13
4.4 IP Ownership Attribution	14
4.5 License Clause Findings.....	14
4.6 M&A Implications and Conditions.....	14
5. Security, Vulnerability and Attack Surface Posture	14
5.1 The End-to-End Security Picture	14
5.2 CVE Posture and the Severity-vs-Verification Reconciliation	15
5.3 Secret Hygiene and the History-Only Population	15
5.4 Infrastructure-as-Code Findings	16
5.5 Attack Surface Composition	16
5.6 Privacy Findings	16
5.7 Reconciling Severity and Verification	17
6. Supply Chain and Dependency Posture	17
6.1 The Third-Party Software Supply Chain	17
6.2 Direct Dependency and Lockfile Posture.....	17
6.3 Unpinned Dependencies and SBOM Obligations	18
6.4 End-of-Life Runtime Exposure	18
6.5 M&A Implications	18
7. Code Quality, Complexity and Technical Debt	18
7.1 The Maintainability Economics.....	19
7.2 Complexity Posture.....	19
7.3 The Top Complexity Hotspots.....	19
7.4 SQALE Grade and Remediation Effort	20
7.5 Duplication and Debt Density	20
7.6 Debt Markers	20

7.7 M&A Implications	21
8. Testing, Coverage and Regression Risk.....	21
8.1 The Leading Indicator of Post-Close Regression.....	21
8.2 Test Deficit Posture.....	21
8.3 Coverage Method and Effective Insured Value	22
8.4 High-Value Coverage Gaps	22
8.5 M&A Implications and Conditions	23
9. Architecture, Entanglement and Integration Risk	23
9.1 Integration Cost is the Most Under-Estimated Line in Software M&A	23
9.2 Entanglement Posture	23
9.3 Integration Risk Score and Proprietary Coupling.....	24
9.4 API Contract Maturity and Database Coupling.....	24
9.5 Serialization and Protocol-Layer Risks	24
9.6 M&A Implications and Conditions	25
10. Engineering Capacity, Bus Factor and Knowledge Loss Vulnerability	25
10.1 Human Capital Risk and the Acquirer's Retention Window	25
10.2 Author Population and Bus Factor.....	25
10.3 Top Key-Person Risk Holders	26
10.4 Module Bus Factors	26
10.5 Git Team Signals.....	27
10.6 Retention and Knowledge-Loss Vulnerability.....	27
10.7 M&A Implications and Conditions	27
11. Maintenance, Documentation and Operational Maturity	28
11.1 The Forward-Looking Operating Cost.....	28
11.2 Maintenance Posture (Not Assessed).....	28
11.3 Documentation Posture	28
11.4 CI Maturity	28
11.5 M&A Implications	29
12. AI/ML Footprint	29
12.1 AI/ML Exposure and Regulatory Implications	29

12.2 Framework and SDK Hits	29
12.3 SDK Hits.....	30
12.4 Bundled Model Files	30
12.5 M&A Implications	30
13. Technology Stack and Lifecycle Posture	30
13.1 The Base-Layer Inventory	31
13.2 Technology Inventory	31
13.3 Runtime EOL Verification.....	31
13.4 M&A Implications	32
14. Consolidated Risk Register and Top Risks.....	32
14.1 The Single Integrated View	32
14.2 Top Risk Findings	32
14.3 Major Risks Register	32
14.4 Pillar Classification Register	33
14.5 Deal-Breaker Register	33
15. Confidence, Scope Limitations and Material Unknowns	34
15.1 Epistemic Boundaries	34
15.2 Integrity Block.....	34
15.3 Confidence Inputs.....	34
15.4 Unverified Pillars.....	35
15.5 Taint Analysis	35
15.6 Out-of-Scope Evidence Register	35
15.7 How the Gaps Propagate into the Recommendation	36
16. Reading Notes and Honesty Conventions.....	36
16.1 Why Conventions Matter	36
16.2 Assessment Coverage	36
16.3 Evidence Honesty	37
16.4 The Currency of Absence	37
16.5 Deal-Mechanic Vocabulary	37
16.6 Severity and Ranking	37

16.7 Known Limits of This Method	38
Appendix A. Methodology, Scope and Scan Configuration	38
A.1 Report Metadata	38
A.2 Scan Mode	38
A.3 Per-Pillar Posture	39
A.4 Step Timings	39
Appendix B. Financial Calibration and Reproduction Fingerprint	39
B.1 Reproduction Fingerprint	40
B.2 Financial Lineage	40
B.3 Full Parameter Table	40
Appendix C. Licensing and IP Inventory	41
C.1 Per-Package License Findings	41
C.2 IP Ownership Attribution	41
Appendix D. CVE Inventory	41
D.1 Full CVE Inventory	42
Appendix E. Complexity, Duplication and SQALE Detail	42
E.1 Top Complexity Hotspot Symbols	42
E.2 SQALE Basis	43
Appendix F. Attack Surface Candidate Endpoints	43
F.1 Candidate Entry Points	43
Appendix G. Key-Person and Knowledge Loss Vulnerability Detail	44
G.1 Knowledge Loss Vulnerability	44
Appendix H. Privacy and Secrets Detail	44
H.1 Privacy Findings	44
H.2 Secret Candidates by Kind	45
Appendix I. AI/ML Inventory	45
I.1 Bundled Model Files	45
Appendix J. Glossary	46

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

© MTD Services

1. Executive Summary and Headline Findings

1.1 Verdict and Confidence

The proposed acquisition of the Homeassistant core repository warrants a verdict of ACQUIRE WITH CONDITIONS. The categorical decision tier is invariant across every parameter in the sensitivity sweep: each of the seventeen scenarios tested returns the same tier, and the recommendation is robust to reasonable variation in hourly rate, complexity multiplier, valuation multiplier, discount rate and aggregate hours cap. The conditions precedent and price-adjustment levers set out in Section 2 must be written into the transaction documents before close.

Confidence in this verdict is 74 out of 100. The score is discounted by three unverified domains (CVE, maintenance and secrets) under the model's confidence policy: a base of 20 points plus 40 points multiplied by the coverage ratio (86.4 percent) plus 40 points multiplied by the risk signal (50.4 percent), with the zero-coverage cap and the no-go floor each applied where they bind. Coverage is the share of pillars with usable signal; the risk signal blends the weighted pillar risk with the financial model. The unverified pillars depress both inputs and are named explicitly so that the reader cannot mistake their effect on the headline number.

HEADLINE ECONOMICS

The deterministic, 100-percent-adoption remediation bound of \$701,995 represents 7.0 percent of the proposed \$10,000,000 transaction price, comfortably above the 5 percent conditions escalation gate but well below the 25 percent no-go threshold. The probability-weighted expected loss (Layer 2) is not quantified, so the actual expected remediation lies at or below this bound. Treat the headline figure as a planning ceiling for scope and sequencing, not as a priced contractor quote.

1.2 Scope, Coverage Gaps and the Confidence Discount

This Report does not support a clean read on three domains that were only partially assessed: CVE (one of eighty-one queried packages could not be resolved to a version because no lockfile exists or the dependency is unpinned), maintenance (no files were scanned by the asset-staleness scanner, so the pillar is not assessed), and secrets (2,806 path-excluded candidate paths were not scanned, against 19,579 files that were scanned). Absence of a finding in those three domains is absence of coverage, not absence of risk. A reader who treats the unverified pillars as clean bills of health will materially understate the technology risk in the Transaction.

No finding in this Report has been independently confirmed by human validation. Of 308 risk rows rendered in the workbook, zero are Confirmed, 208 are Detected-not-verified, 100 are Inferred, and the remaining rows are split across Partially-assessed, Unassessed, Conflicting and

Unsupported states. The Detected-not-verified and Inferred states reflect scanner-detected or inferred signals: they identify where risk may exist, not where risk has been confirmed to exist. They must be validated by human inspection before they are relied on in negotiation, and the conditions precedent in Section 2 deliberately encode this requirement.

The scan is static, offline-or-network-integrated at the source level only. It does not establish runtime behaviour, production incidents, or anything requiring access to systems, personnel or contracts. Eight further domains are explicitly out of scope and listed in Section 15.16: runtime and operational behaviour, network perimeter and OT boundaries, external attack surface, identity and access controls, data resilience and recovery, delivery assurance (live CI pass rates and load tests), legal instruments and contracts, and runtime dependency and vulnerability freshness. Each out-of-scope domain is paired with the independent evidence the buyer must obtain to close the gap.

1.3 Driver and Pillar Posture

The driver of the recommendation is the technical register (the consolidated risk register in Section 14). Two pillars return critical risk (Complexity and Entanglement, both at health 15 of 100); seven return high risk (Coverage, Test Deficit, Debt, Integration, IP Ownership, Retention, and the unverified Secrets and CVE pillars); six return medium risk (Attack Surface, CI Maturity, EOL, IaC, License, Tech Stack); four return low or informational risk (Dependency, Doc Staleness, Key Person, Privacy). The critical pillars are structural and cannot be retired by quick fixes; they require sequenced refactoring programmes whose cost is sized in Section 3.

Figure 1 — Pillar Health Score Register (22 pillars)

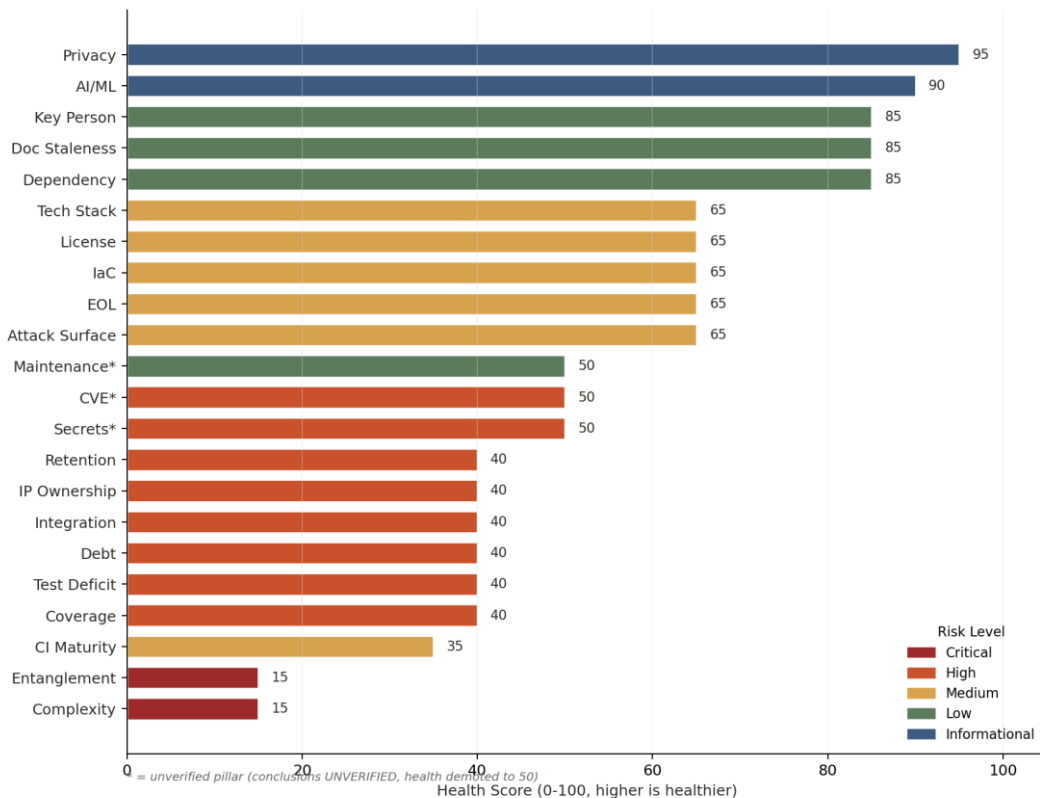


Figure 1 — Pillar health scores sorted ascending. Unverified pillars (CVE, Maintenance, Secrets) are demoted to 50 so an incomplete scan never reads as a strong pillar.

1.4 Remediation Economics

Total remediation hours are 14,040, at an assumed hourly rate of \$50, producing a deterministic bound of \$701,995. The complexity multiplier of 1.5 amplifies the hours-per-unit for hard-to-remediate complexity hotspots. The valuation multiplier of 3 amplifies remediation cost into a valuation discount of 23.2 percent (capped at 25 percent). Post-merge integration drag is 80.0 percent (capped at 80 percent) under the integration drag curve, sized as remediation hours divided by the product of productive hours per quarter (624), eight-hour days and 260 working days, multiplied by the integration drag factor of 0.15 percent.

Person-years of remediation effort are 6.7 at the assumed 2,080 annual working hours. The confidence interval width is 30 percent on hours-per-unit estimates, producing a low bound of \$491,396 and a high bound of \$912,594. The interval is a stated-confidence range from the model calibrations, not a stochastic bound. It is reproduced verbatim from the stored ground truth and is one of the inputs the buyer's quality-of-earnings provider will probe.

1.5 Headline Findings

The five most material findings, ranked by the consolidated register, are: critical complexity in the device registry and config entry setup paths (cyclomatic complexity 94 and 29 respectively); sole-maintainer concentration on 1,899 critical modules (knowledge transfer costed at \$385,800); the presence of a HACK marker in production code (skybeacon sensor); twenty-six advisory matches on two dependencies (certifi 2021.5.30 and urllib3 2.0) carrying UNKNOWN severity; and 2,865 secret-pattern candidates across the scanned surface. None of the twenty-six CVEs is independently confirmed and all twenty-six are flagged no-fixed-release under the OSV advisory source; the UNKNOWN severity is itself a verification gap that Section 5 reconciles explicitly.

Table 1.1 — Top ten risk findings by consolidated register

Pillar	Title	Severity	Impact	File	Est. Cost (USD)
Complexity	<code>_async_build_immich</code>	Critical	Critical	<code>homeassistant/components/immich/media_source.py</code>	Not quantified
Complexity	<code>__async_handle_config_entry_setup_error</code>	Critical	Critical	<code>homeassistant/config_entries.py</code>	Not quantified
Complexity	<code>_async_add_entity_impl</code>	Critical	Critical	<code>homeassistant/helpers/entity_platform.py</code>	Not quantified

Pillar	Title	Severity	Impact	File	Est. Cost (USD)
Retention	Artur Pragacz	Critical	Critical	Not quantified	Not quantified
Retention	Abilio Costa	Critical	Critical	Not quantified	Not quantified
Retention	Alistair Francis	Critical	Critical	Not quantified	Not quantified
Debt	HACK marker (skybeacon)	High	High	homeassistant/components/skybeacon/sensor.py:145	\$10,095
Attack Surface	command (amcrest)	Medium	Medium	homeassistant/components/amcrest/__init__.py:182	Not quantified
Attack Surface	command (mikrotik)	Medium	Medium	homeassistant/components/mikrotik/coordinator.py:329	Not quantified
Attack Surface	main (vendor_sqlite3)	Medium	Medium	script/vendor_sqlite3.py:98	Not quantified

Source: `decision.top_risks` in the stored payload. Estimated cost is shown as "Not quantified" where the source carries no figure, rather than as a zero; the convention is defined in Section 16.

1.6 What the Evidence Does Not Support

Three classes of question a buyer would reasonably ask are outside what the evidence supports. First, runtime and operational behaviour: the target's code and systems are never executed, so production incidents, SLO attainment and error-budget consumption are not observed. Second, the legal posture of contributor contributions: signed Contributor Licence Agreements, IP-assignment and contractor terms are legal artefacts, not code; the IP Ownership pillar attributes ownership by git blame ratios and is a diligence signal, not legal proof of title. Third, the AI/ML regulatory exposure of any production model: zero production model files are bundled (the three .tflite files are 4-byte test stubs), so the regulatory and IP risk that distinguishes AI/ML from conventional open-source exposure is not engaged by the shipped artefact, though it may be engaged by the SDK and framework usage documented in Section 12.

1.7 Deal-Thesis Alignment

A deal thesis, in the context of this Report, is the buyer's stated rationale for the acquisition, evaluated against the evidence already established in the body of the Report. The theses below add no measurement of their own; they are an interpretive overlay that maps stored evidence to buyer intent. All six declared theses are scored side by side, none is primary, and none carries a condition. The alignment score on a 1-to-10 scale 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 a thesis marked "refused" is one for which the source data carries no usable signal (in this case, no signal in the repository distinguishes proprietary algorithmic density from standard framework boilerplate).

Table 1.2 — Deal-thesis alignment, all six declared theses

Perspective	Alignment	Evidence State	Basis
Scale and 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 and Cross-Sell	10 / 10 (5 of 5 metrics resolved)	Detected, not verified	Interface boundaries are inferred by pattern detection over source text. They describe where the seams are, not whether two products can be sold together.
Lean Maintenance and Operating Cost	10 / 10 (5 of 5 metrics resolved)	Detected, not verified	Remediation effort is an SQALE estimate over static structure. It bounds the size of the maintenance backlog; it does not price a headcount saving.
IP Moat and Proprietary Core	Refused	Unassessed	Not supported by static analysis. No signal in this Report distinguishes proprietary algorithmic density from standard framework boilerplate and wrapper templates.
Regulatory and Compliance	7 / 10 (6 of 6 metrics resolved)	Partially assessed	Contributing pillar(s) partial did not complete a full assessment, so this verdict is capped regardless of the figures above.
Talent and 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 what a repository can establish.

Source: `decision.deal_theses` in the stored payload. The 1-to-10 alignment score rates how strongly and completely the stored evidence bears on each thesis within its evidence-state cap; it is not a business-fit score. "Not quantified" means no metric resolved.

1.8 Required to Decide

Ten conditions for acquisition are open and listed in full in Section 2. They resolve into three deal mechanics: two conditions precedent (CVE remediation before close; credential rotation before close), seven price-adjustment levers (complexity-reduction milestone, coverage uplift, technical-debt fund, decoupling plan, integration findings, retention agreements, regression

coverage uplift), and one warranty or representation (assignability of third-party contributions). The conditions precedent must be satisfied or proven before close; the price-adjustment levers move value rather than block close; the warranty is an indemnity question, not a work plan. The size of any escrow, holdback, warranty cap or indemnity ceiling is for deal counsel and is deliberately not asserted as a number in this Report.

No unresolved contradictions remain in the ledger (Appendix R). One resolved conflict is recorded: the anti-double-counting ledger verifies that shared-registry keys show each evidence item once on multiple surfaces, never added twice. The conflict is resolved by-design with no decision impact. The buyer's quality-of-earnings provider can re-derive the headline numbers from the financial reproduction fingerprint in Appendix B.

© MTD Services

2. Transaction Recommendation, Conditions and Escalation Thresholds

2.1 Decision Logic

The transaction-level decision logic turns on three inputs: the categorical board decision tier (derived from the consolidated risk register), the economic decision tier (derived from remediation cost as a percentage of the proposed price, against the conditions escalation gate and the no-go threshold), and the explicit conditions for acquisition (ten categorical conditions mapped to deal mechanics). Each is read directly from the stored decision and financial model; nothing in this Section is interpreted, only presented for negotiation.

The categorical decision tier is invariant across every parameter in the sensitivity sweep. The seventeen scenarios tested vary hourly rate (multipliers 0.7, 1.0 and 1.3), complexity multiplier (0.7, 1.0, 1.3), valuation multiplier (0.7, 1.0, 1.3), discount rate (0.7, 1.0, 1.3), and the aggregate hours cap (0.7, 1.0, 1.3). Each scenario returns the same ACQUIRE WITH CONDITIONS verdict. The recommendation is therefore robust to reasonable negotiation of the underlying assumptions; the only way to move the categorical tier is to retire a deal-breaker pillar (ai_ml, attack_surface, cve, iac, license or secrets), and none of those pillars is rated critical in the current scan.

2.2 Conditions for Acquisition

The ten conditions for acquisition are reproduced below verbatim from the stored decision layer, annotated with the deal mechanic each one belongs to. The deal-mechanic tag is declared at the decision layer, never inferred from the wording of the condition. Two conditions are conditions precedent (must be satisfied or proven before close), seven are price-adjustment levers (funded or escrowed work items that move value rather than block close), and one is a warranty or representation (a seller statement whose breach is an indemnity question, not a work plan).

Table 2.1 — Conditions for acquisition, by deal mechanic

#	Deal Mechanic	Pillar	Condition
1	Price-adjustment lever	Complexity	Include a complexity-reduction milestone in the integration plan
2	Price-adjustment lever	Coverage	Address coverage findings as a condition precedent
3	Condition precedent	CVE	Remediate reachable known CVEs before close
4	Price-adjustment lever	Debt	Include a technical-debt remediation fund in the integration plan
5	Price-adjustment lever	Entanglement	Plan decoupling for circular dependency cycles
6	Price-adjustment lever	Integration	Address integration findings as a condition precedent

#	Deal Mechanic	Pillar	Condition
7	Warranty / representation	IP Ownership	Warrant assignability of third-party contributions
8	Price-adjustment lever	Retention	Secure retention agreements for sole-maintainer modules
9	Condition precedent	Secrets	Rotate all leaked credentials and stand up credential rotation
10	Price-adjustment lever	Test Deficit	Raise regression coverage on high-risk modules post-close

Source: `decision.conditions_for_acquisition` and `decision.condition_details` in the stored payload. The `deal-mechanic` tag is declared at the decision layer, never inferred from the wording.

2.3 Deal-Breaker Register

The deal-breaker register is empty. Under the decision model, a critical rating on any of six named pillars (AI/ML, Attack Surface, CVE, IaC, License or Secrets) would force the no-go verdict; none of those six carries a critical rating in the current scan. The Secrets and CVE pillars are rated high but are unverified (partial coverage); a critical rating on either pillar after human validation would re-open the deal-breaker analysis. The conditions precedent on credential rotation (Condition 9) and CVE remediation (Condition 3) are the contractual mechanism by which the deal absorbs that residual verification risk.

2.4 Escalation Gates

The conditions escalation threshold is 5 percent of the proposed price; the no-go threshold is 25 percent. The conditions materiality floor is an absolute \$250,000, which the conditions gate must also clear before the recommendation escalates to ACQUIRE WITH CONDITIONS; the no-go materiality floor is not quantified, so the no-go gate is percent-only. At a remediation cost of \$701,995 against a proposed price of \$10,000,000, remediation represents 7.0 percent of the price. The conditions gate binds (7.0 percent exceeds 5.0 percent and \$701,995 exceeds \$250,000); the no-go gate does not bind (7.0 percent is well below 25.0 percent).

Table 2.2 — Per-pillar escalation view against both gates

Pillar	Cost Category	Cost (USD)	% of EV	Breach
Retention	Knowledge transfer	\$385,800	3.86%	None — below both gates on its own
Complexity	Critical complexity	\$127,400	1.27%	None — below both gates on its own
Complexity	High complexity	\$145,400	1.45%	None — below both gates on its own
Test Deficit	Test investment	\$24,800	0.25%	None — below both gates on its own
Debt	General debt	\$10,095	0.10%	None — below both gates on its own
IP Ownership	IP reassignment	\$7,000	0.07%	None — below both gates on its own

Pillar	Cost Category	Cost (USD)	% of EV	Breach
Entanglement	Decoupling	\$1,000	0.01%	None — below both gates on its own
Dependency	Dependency pinning	\$500	0.01%	None — below both gates on its own
CVE	CVE remediation	\$0	0.00%	None — category not costed this scan
Maintenance	Staleness remediation	\$0	0.00%	None — pillar not assessed
Attack Surface	Taint remediation	\$0	0.00%	None — category not costed this scan

Source: *decision.escalation.pillars* in the stored payload. *pct_of_price* is derived as *total_cost* divided by *proposed_price* times 100 from two stored figures; it is not a new estimate. The aggregate \$701,995 (7.0 percent) clears the conditions gate; no individual line item would clear it on its own.

2.5 Sensitivity of the Recommendation

The recommendation is robust to reasonable variation in the financial inputs. The hourly rate is the single most influential parameter: at a 0.7 multiplier (\$35 per hour), total remediation cost falls to \$491,396 (4.9 percent of price); at a 1.3 multiplier (\$65 per hour), it rises to \$912,594 (9.1 percent of price). Both scenarios clear the conditions gate; neither approaches the no-go threshold. The complexity multiplier (0.7, 1.0, 1.3) moves total remediation cost by approximately \$13,500 across the full range, a small fraction of the total. The valuation multiplier affects only the valuation discount (not the underlying cost); at 0.7 the discount is 21.0 percent, at 1.3 it is 24.2 percent, both within the 25 percent cap. The discount rate has no effect on the figures presented because present-value discounting is not applied (the parameter is absent from the source).

Figure 3 — Sensitivity Tornado: Total Remediation Cost vs Parameter Multipliers (0.7x / 1.0x / 1.3x)

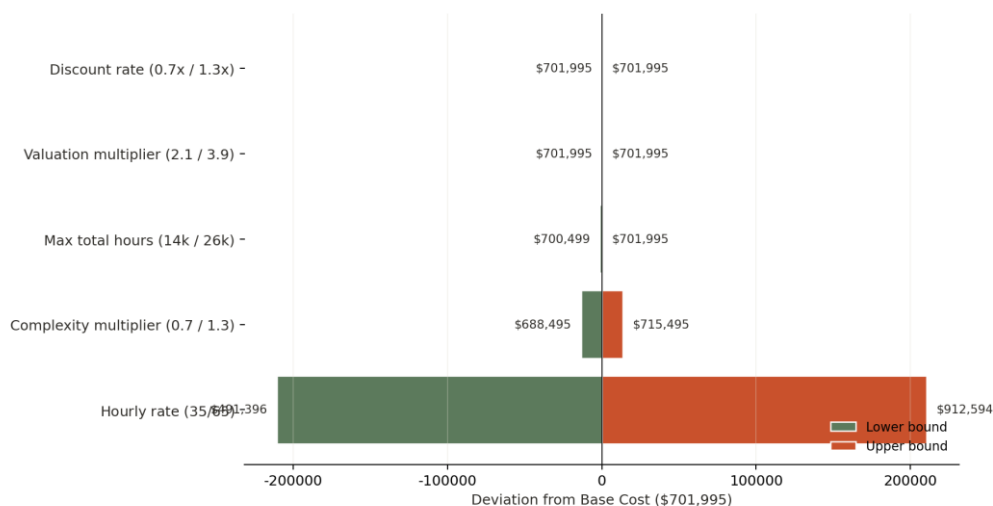


Figure 3 — Sensitivity tornado chart showing deviation of total remediation cost from the \$701,995 base case under each parameter sweep. The hourly rate dominates; all other parameters are second-order.

2.6 Recommendation

The recommendation to the Investment Committee is to proceed with the Transaction on the basis of ACQUIRE WITH CONDITIONS. The conditions precedent (Conditions 3 and 9) should be written into the agreement as conditions to closing, with the seller required to remediate reachable known CVEs and to rotate all leaked credentials and stand up credential rotation before close. The seven price-adjustment levers should be funded as a remediation programme in the integration plan, sequenced per the indicative roadmap in Section 3. The warranty on assignability of third-party contributions (Condition 7) should be drafted with the IP Ownership evidence in Section 4 in mind, and the indemnity cap should reflect the 35 high-risk files identified in the same pillar. The size of any escrow, holdback, warranty cap or indemnity ceiling is for deal counsel and is deliberately not asserted as a number in this Report.

© MTD Services

3. Remediation Cost Model, Valuation Impact and Sensitivity

3.1 The Bridge from Technical Findings to Deal Economics

This Section is the bridge between the technical work documented in Sections 4 through 13 and the deal model the Investment Committee will use to price the Transaction. It translates findings into line-item remediation hours and cost, into a valuation discount on the proposed price, into a post-merger integration drag, and into a sensitivity analysis that tests the recommendation against the seller's likely quality-of-earnings challenge. Every figure in this Section is stored ground truth, reproduced verbatim from the payload whose fingerprint is recorded in Appendix B; no figure is interpreted or estimated.

Where a figure is absent the model writes one of four markers, never a zero: "not quantified" means the model could compute a number but a stored input is missing; "not applicable" means the cell is structurally inapplicable to that row; "not recorded" means the source does not carry the value; "not assessed" means the value was not established because coverage was incomplete. Each marker names a different kind of absence so that an empty cell is never read as a clean result. The convention is defined once in Section 16 and applies throughout this Report.

3.2 Parameters Used Downstream

Every value that can move a number in this Report is declared in one place: financial inputs, saturation calibration, valuation and integration-drag curves, confidence weights, decision thresholds, post-close schedule constants and health-posture ceilings. Values are read from the stored payload where it carries them and from the tool's own constants otherwise; a parameter that was not configured is listed as not quantified, never assumed away. The table below records the parameters most material to the deal model; the full parameter table is in Appendix B.

Table 3.1 — Material financial parameters (selected)

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 hard-to-remediate complexity hotspots
Remediation inputs	Valuation multiplier	3.0	Amplifies remediation cost into the valuation discount
Remediation inputs	Integration drag factor	0.15%	Post-merge integration friction sized as hours / (8 x 260 x factor)

Group	Parameter	Value	Affects
Remediation inputs	Productive hours per quarter	624 h	Net quarterly working hours behind per-cause effort estimates
Remediation inputs	Confidence interval width	30%	Width of the low / high remediation-cost interval
Aggregate hours cap	Max total hours	20,000 h	Aggregate ceiling on remediation hours; surplus above it is decayed
Aggregate hours cap	Surplus factor	0.25	Fraction of hours above the cap that is credited
Valuation curve	Valuation discount cap	25%	Maximum valuation discount on the proposed price
Integration drag	Integration drag cap	80%	Maximum post-merge integration drag
Decision thresholds	Conditions threshold	5% of price + \$250,000 floor	Remediation above this gate triggers conditions
Decision thresholds	No-go threshold	25% of price	Remediation above this gate triggers no-go
Decision thresholds	Deal-breaker pillars	ai_ml, attack_surface, cve, iac, license, secrets	A critical rating on any of these forces no-go
Schedule constants	Execution teams	2 (two engineers each)	Post-close remediation is planned against two two-engineer teams
Schedule constants	Sprint cap	12 sprints	Hours that do not fit inside this many sprints are reported unplanned
Schedule constants	Hours per workday	4 h	Roadmap durations derive as effort-hours / 4 h per workday

Source: financial.assumptions in the stored payload. The full parameter table (including saturation caps, decay rates, line-item basis, confidence weights and health ceilings) is in Appendix B.

3.3 Remediation Line Items

Total remediation hours are 14,040, producing a deterministic cost bound of \$701,995 at the assumed \$50 hourly rate. The hours are distributed across eight cost categories, each governed by a saturation model: a cap that bounds the count of distinct items costed at the full hours-per-unit, and a decay rate that applies to the surplus. The knowledge-transfer category dominates: 1,899 critical sole-maintainer modules, costed per unique module rather than per at-risk contributor, at a saturated 4.06 hours per unit producing 7,716 hours and \$385,800. The complexity categories together account for \$272,800 (5,456 hours) across 1,616 high-complexity and 462 critical-complexity symbols. The remaining six categories together account for \$43,395 (868 hours).

Table 3.2 — Remediation line items (all eight costed categories)

Category	Count	Sat. Cap	HPU	Total Hours	Total Cost	Notes
Knowledge transfer	1,899	30	4.06	7,716 h	\$385,800	Critical sole-maintainer modules, costed per unique module not per at-risk contributor
High complexity	1,154	150	2.52	2,908 h	\$145,400	High-complexity functions, exclusive of critical tier
Critical complexity	462	50	5.52	2,548 h	\$127,400	Critically complex functions requiring refactoring
Test investment	64	20	7.75	496 h	\$24,800	High-regression-risk modules requiring test investment
General debt	1,219	200	0.17	202 h	\$10,095	Confirmed technical-debt markers (annotation + tooling)
IP reassignment	35	50	4.00	140 h	\$7,000	Files with external-dominant authorship requiring IP review
Decoupling	1	10	20.00	20 h	\$1,000	Circular dependency cycle requiring decoupling
Dependency pinning	10	100	1.00	10 h	\$500	Unpinned dependencies requiring version locking

Source: *financial.line_items* in the stored payload. Five further categories (*critical_path_debt*, *taint_remediation*, *staleness_remediation*, *sole_maintainer_risk*, *sqale_remediation*) are explicitly zero this scan and are listed in Section 3.10.

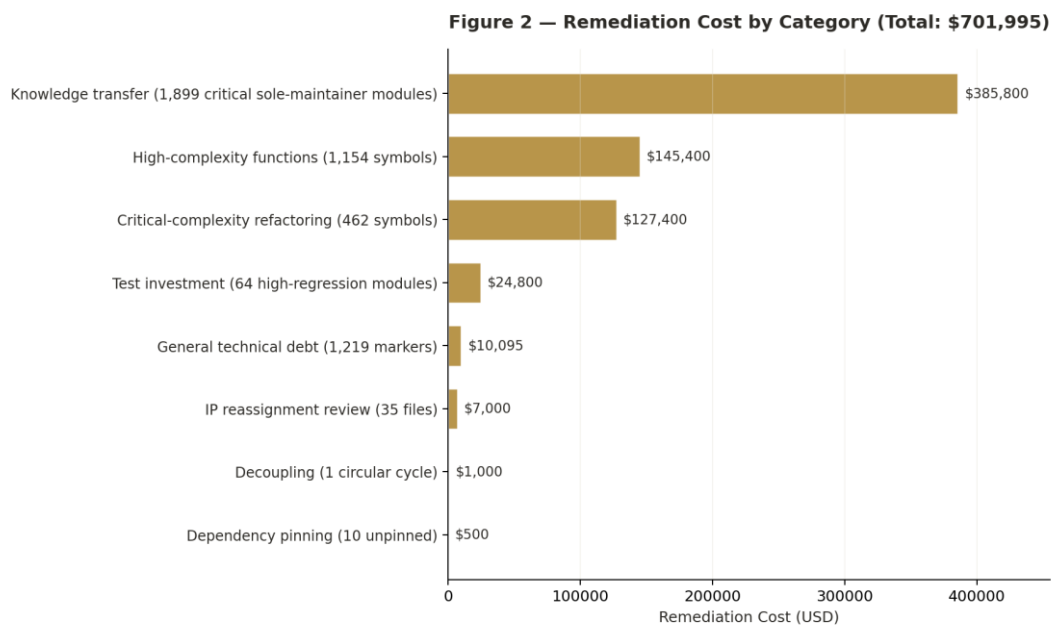


Figure 2 — Remediation cost by category. Knowledge transfer dominates at 55 percent of total cost; complexity categories together account for 39 percent.

3.4 Confidence Interval

The remediation-cost confidence interval is plus or minus 30 percent on the hours-per-unit estimates, producing a low bound of \$491,396 and a high bound of \$912,594. The interval is a stated-confidence range from the model calibrations, not a stochastic bound: it reflects the imprecision in the per-unit hours-per-unit estimates rather than a probability distribution over outcomes. A buyer's quality-of-earnings provider should treat the high bound as the planning figure for escrow sizing and the low bound as the figure the seller will advance in negotiation; the gap between them is the negotiation surface.

The aggregate hours cap was not applied (the model's max_total_hours is 20,000 and total remediation hours are 14,040, so the cap does not bind). The surplus factor of 0.25 therefore has no effect on this scan. The aggregate hours cap exists to bound runaway scans where hours exceed 20,000; the buyer's quality-of-earnings provider should note that the cap is not engaged and that the headline \$701,995 is therefore not artefactually reduced.

3.5 Valuation Discount and Integration Drag

The valuation discount applied is 23.2 percent, against a cap of 25 percent. The discount is computed by a valuation curve whose half-life scales as the maximum of 80 and total_cost divided by 50,000; the curve amplifies the raw remediation cost (multiplied by the valuation multiplier of 3) until it reaches the cap. The post-merge integration drag is 80.0 percent, against a cap of 80 percent. The drag is computed by an integration-drag curve whose half-life scales as the maximum of 200 and total_hours divided by 10; the curve amplifies the raw drag (sized as remediation hours divided by 624 productive hours per quarter times eight times 260 times the integration drag factor of 0.15 percent) until it reaches the cap. Both caps bind, meaning the headline figures are at the model's maximum and would not move further with additional remediation hours.

3.6 Sensitivity Sweep

The sensitivity sweep tests the recommendation against five parameters at three multiplier values each (0.7, 1.0, 1.3). The hourly rate is the dominant parameter, moving total remediation cost by \$421,198 across the full range (\$491,396 to \$912,594); the valuation multiplier affects only the valuation discount (21.0 percent to 24.2 percent); the complexity multiplier moves total remediation cost by \$27,000 across the range; the discount rate has no effect because present-value discounting is not applied (the parameter is absent from the source); the max_total_hours cap has no effect at the 1.0 and 1.3 multipliers because total hours do not exceed the cap. Two sweep parameters (remediation_timing and scenario_weights) are not quantified; their cells are blank rather than zero, in line with the convention in Section 16.

Table 3.3 — Sensitivity sweep outcomes (selected scenarios)

Parameter	Multiplier	Value	Total Cost	Val. Discount %	Drag %	Decision
Hourly rate	0.7	\$35	\$491,396	21.04%	80%	ACQUIRE_WITH_CONDITIONS
Hourly rate	1.0	\$50	\$701,995	23.20%	80%	ACQUIRE_WITH_CONDITIONS
Hourly rate	1.3	\$65	\$912,594	24.18%	80%	ACQUIRE_WITH_CONDITIONS
Complexity multiplier	0.7	1.05	\$688,495	23.11%	80%	ACQUIRE_WITH_CONDITIONS
Complexity multiplier	1.3	1.95	\$715,495	23.29%	80%	ACQUIRE_WITH_CONDITIONS
Valuation multiplier	0.7	2.10	\$701,995	21.04%	80%	ACQUIRE_WITH_CONDITIONS
Valuation multiplier	1.3	3.90	\$701,995	24.18%	80%	ACQUIRE_WITH_CONDITIONS
Max total hours	0.7	14,000 h	\$700,499	23.19%	80%	ACQUIRE_WITH_CONDITIONS
Max total hours	1.3	26,000 h	\$701,995	23.20%	80%	ACQUIRE_WITH_CONDITIONS
Discount rate	0.7 to 1.3	0.0245 to 0.0455	\$701,995	23.20%	80%	ACQUIRE_WITH_CONDITIONS

Source: financial.sensitivity in the stored payload. The full seventeen-row sweep is in Appendix B. All seventeen scenarios return the same categorical decision tier.

3.7 Scenario Branches

The model computes four scenario branches, each carrying its own evidence basis and total impact. The remediate_before_close branch carries a \$701,995 Layer-1 remediation cost plus a \$2,320,000 transaction impact (the difference between the proposed price and the adjusted economic value); the acquire_remediate_post_close branch carries a \$7,915,959 total impact (transaction EV plus Layer-1 remediation plus integration drag); the partial_remediation branch carries a \$374,335 total impact (config calibration); the material_failure downside case carries a \$90,194 total impact (analyst calibration on documented scan evidence). The transaction-impact column is not quantified for three of the four branches because the model does not calibrate a transaction impact for them; the unquantified cells are stated, not silently zeroed.

Table 3.4 — Deal economics headline reconciliation

Headline EV	Net Quantified Tech Impact	Adjusted Economic Value	Price Adjustment	Cap Applied
\$10,000,000	\$8,401,995	\$7,500,000	\$2,500,000	Yes (25% cap binds)

Source: *financial.economic_layers.layer4_deal* in the stored payload. The price adjustment of \$2,500,000 equals the 25 percent valuation-adjustment cap on the \$10,000,000 headline EV. The size of any escrow, holdback, warranty cap or indemnity ceiling is for deal counsel and is deliberately not a number here.

3.8 Deal Components and the Double-Counting Ledger

The deal-economics headline is composed of seven distinct deal components, each carrying its own stored amount, basis, confidence, calculation and assumptions. Components are never collapsed and never labelled as a market-style VaR figure: an unquantified component reads "not quantified" rather than zero. The double-counting ledger verifies that shared-registry keys show each evidence item once on multiple surfaces, never added twice. The ledger records twenty-six figures across cost, scenario, deal and risk components; three keys appear on multiple surfaces (*layer1.integration_drag_pct*, *layer1.total_cost*, *layer2.aggregate_residual_expected_loss*) and each is justified by the shared-registry design rather than priced twice. The ledger's clean verdict is true: no double-counting findings remain.

DEAL ECONOMICS HEADLINE

The headline economics read as follows. The proposed price is \$10,000,000. The net quantified technology impact is \$8,401,995 (composed of \$8,000,000 integration and separation cost, \$701,995 remediation cost, \$150,000 other evidence-supported security impact, \$500,000 synergy/customization burden, and \$50,000 residual expected loss). The adjusted economic value is \$7,500,000, against which the price adjustment of \$2,500,000 is the 25 percent valuation-adjustment cap binding.

3.9 Indicative Post-Close Roadmap

The post-close remediation roadmap is planned against two execution teams of two engineers each, working ten workdays per sprint at four productive hours per workday, across a maximum of twelve sprints. Workstream durations derive as effort-hours divided by four hours per workday, clamped to a minimum of seven and a maximum of 120 days. The figure below maps the ten remediation workstreams against the first six months after close, with conditions precedent front-loaded into the first quarter. The roadmap is indicative; the integration-management office should re-baseline it against the buyer's actual capacity and the sequencing constraints identified in Section 9 (Architecture, Entanglement and Integration Risk).

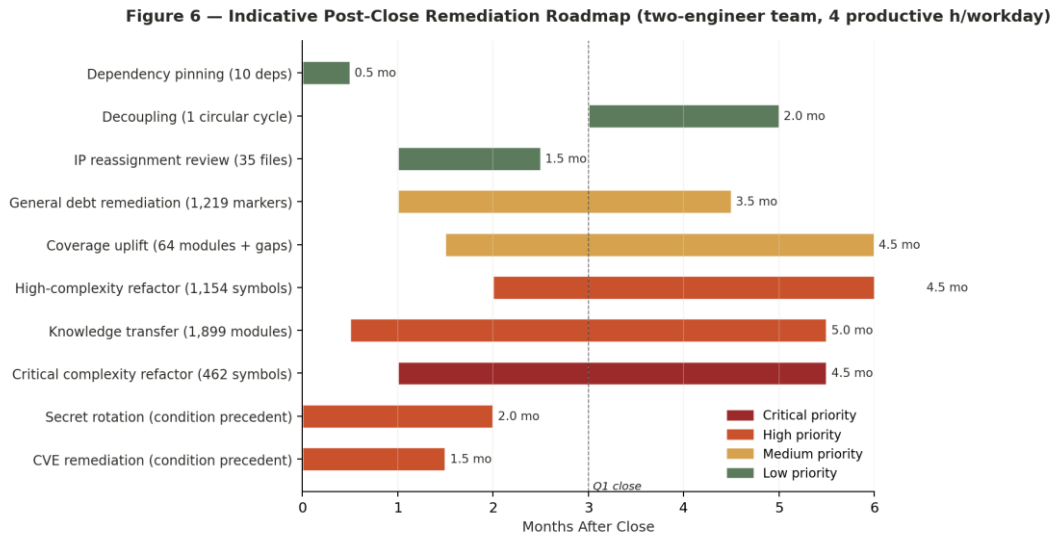


Figure 6 — Indicative post-close remediation roadmap. Conditions precedent (CVE and secrets) front-load into Q1; critical complexity refactoring and knowledge transfer run in parallel through Q2.

3.10 Categories Not Costed This Scan

Five remediation categories carry no line item this scan and are explicitly zero in the Remediation_Costs sheet: critical_path_debt (no critical-path markers were detected), taint_remediation (the taint analysis returned zero flows), staleness_remediation (the maintenance pillar was not assessed), sole_maintainer_risk (the model uses knowledge_transfer instead, costed per unique module), and sqale_remediation (the SQALE debt-effort basis is reconciled against the complexity hotspots rather than priced as a separate line). Rendering the gap here keeps the remediation scope honest rather than presenting a clean table; the buyer's quality-of-earnings provider should note that the absence of a line item is not the absence of work.

4. Licensing and IP Exposure

4.1 Why Licensing and IP Drive Title and Warranty Risk

The licensing and IP posture of the target's packaged components and source files drives three distinct deal concerns. First, copyleft or restrictive licence exposure can affect integration into a closed-source acquirer product, trigger customer disclosure obligations, and require carve-outs or re-licensing negotiations. Second, unknown licences on shipped dependencies create warranty risk: the seller cannot represent what terms apply to a package whose licence it has not identified. Third, the ownership attribution of the source files drives title risk: a high concentration of externally-authored files increases the surface on which a third party could assert ownership claims post-close. This Section addresses each concern in turn.

4.2 License Posture Summary

The target ships 65 licensed packages drawn from seven manifests (six Python pyproject and requirements files, one JavaScript package.json). Of those 65 packages, zero carry viral licences, zero carry restrictive licences, and 24 carry UNKNOWN licences (37 percent of the inventory). The remaining 41 packages carry permissive licences dominated by Apache-2.0, MIT, BSD variants and PSF-2.0. Zero contamination chains were detected. The UNKNOWN-licence population is the principal licensing exposure: a buyer cannot rely on the seller's representation of licence compliance for 37 percent of the shipped dependencies until the UNKNOWN population is resolved by direct inspection of each package's metadata.

Table 4.1 — License posture summary

Metric	Value	M&A Significance
License findings inventory	67 findings	Total rows in the per-package findings sheet, including repo-level and per-package entries
Viral licence count	0	No GPL/AGPL-family exposure on the shipped inventory
Restrictive licence count	0	No copyleft or non-permissive licences detected
Unknown licence count	24	37 percent of inventory; warranty scope limitation
Licensed packages total	65	Population against which the licence scan ran
Contamination chains	0	No viral-to-permissive contamination paths detected
Repo-level licence	Apache-2.0	Target itself is permissively licensed
Repo-level (Python backports)	ISC	Permissive; consistent with the repo-level Apache-2.0
Test framework licence	Apache-2.0 (@playwright/test)	Permissive; no test-framework exposure

4.3 The UNKNOWN-Licence Population

The 24 packages with UNKNOWN licences are reproduced in Appendix C. They include aiodns, aiogithubapi, aiohttp-asyncmdnsresolver, annotatedyaml, audioop-lts, crowsim, fnv-hash-fast, gazetteer-matcher, hass-nabucasa, home-assistant-intents, httpx2, lru-dict, mutagen, orjson, probatio, pygithub, pyjwt, python-slugify, rf-protocols, truststore, ulid-transform, uv, webRTC-models, and zeroconf. Several of these are widely-used packages with known permissive licences (for example, orjson is MIT-licensed upstream, pyjwt is MIT-licensed, and truststore is Apache-2.0-licensed); the UNKNOWN classification reflects the scanner's inability to read the licence field from the manifest, not the absence of a licence. Resolution requires direct inspection of each package's PyPI metadata or source repository, and is sized at the ip_reassignment line item (\$7,000 for 35 files) rather than as a separate cost line.

4.4 IP Ownership Attribution

The IP ownership attribution metrics measure the concentration of file authorship across the target's 75 attributed files. The corporate domain is users.noreply.github.com, reflecting the target's open-source contributor model. Of 5,902 total authors, 3,813 (64.6 percent) are external; external authors are blamed for 80.43 percent of lines and 80.43 percent of commits, against 1,700 distinct external domains. 35 files are flagged as high-risk (externally owned with dominant authorship). The IP risk score is 0.6017 on a 0-to-1 scale. The reconciliation note in the source confirms that attributed_file_count (75) is canonical, and that total_files, external_owned_files and external_author_count are retained as legacy aliases.

IP OWNERSHIP CAVEAT

IP ownership attribution is a diligence signal, not legal proof of title. Git blame describes who has touched the code, not who owns the copyright. The 35 high-risk files identify where the seller's warranty on assignability of third-party contributions (Condition 7) is most exposed; resolution requires the buyer's counsel to obtain signed Contributor Licence Agreements or equivalent IP-assignment instruments from the contributors who authored those files.

4.5 License Clause Findings

The licence clause scan identifies two clauses in the repo-level Apache-2.0 licence: an attribution clause (attribution severity, matched to zero packages because the clause scan covers the repo-level licence file only) and a patent-grant clause (permissive severity, also matched to zero packages). Both clauses are standard Apache-2.0 terms and carry no commercial concern beyond the standard Apache-2.0 obligations on redistribution. The clause scan does not extend to per-package licence files because the per-package licence inventory is read from the manifest licence field, not from per-package licence files bundled in the repository.

4.6 M&A Implications and Conditions

The principal M&A implications are three. First, the seller should provide a representation and warranty covering licence compliance for the shipped product, with an indemnity for any copyleft exposure that materialises post-close; the warranty is appropriately scoped to the 65 licensed packages and the 24 UNKNOWN-licence packages (which require either resolution or carve-out). Second, the IP Ownership evidence (35 high-risk files, 80.43 percent external authorship) is the basis for the warranty on assignability of third-party contributions (Condition 7); the indemnity cap on that warranty should reflect the population of high-risk files and should be the subject of a separate legal workstream. Third, the absence of viral and restrictive licences is a positive value driver: it removes the principal integration risk that would otherwise complicate a closed-source acquirer's product strategy and removes a typical point of friction in customer disclosure obligations.

5. Security, Vulnerability and Attack Surface Posture

5.1 The End-to-End Security Picture

The security picture is composed of five distinct evidence streams: known advisory exposure (CVEs under the OSV source), critical-severity posture (zero critical CVEs but 26 UNKNOWN-severity advisories), secret hygiene (2,865 pattern candidates, 3 high-confidence, 2,242 history-only), infrastructure-as-code findings (one medium-severity Dockerfile finding), and attack-surface composition (94 candidate entry points, 90 unreferenced, 82 untested). Each stream is read from its own scanner; each is qualified by the coverage caveats set out in Section 15. Together they drive the cyber-insurance sizing, the regulatory disclosure timing, and the cyber-warranty language in the acquisition agreement.

5.2 CVE Posture and the Severity-vs-Verification Reconciliation

The advisory check queried 81 resolvable packages against the OSV (osv.dev) source and returned 26 CVE matches. One further package (certifi, declared with a ranged specifier) could not be resolved to a specific version because no lockfile exists and the dependency is unpinned; the advisory check therefore covered only the 81 resolvable packages. The 26 CVEs break down as follows: 6 affect certifi 2021.5.30 (5 GHSA + 1 PYSEC + advisory from 2022 to 2024), and 20 affect urllib3 2.0 (12 GHSA + 6 PYSEC + 2 from 2026). All 26 carry UNKNOWN severity under the OSV source and all 26 are flagged no-fixed-release (no fixed version is available).

The reconciliation the buyer must understand is between the headline "0 critical CVEs" and the unverified nature of the 26 advisories. The headline is correct as far as it goes: zero advisories carry a Critical or High CVSS score, because the OSV source did not return a CVSS score for these specific advisories. But the absence of a CVSS score is itself a verification gap. A buyer treating the headline as a clean bill of health would materially understate the security exposure. The remediation class for all 26 advisories is no_fixed_release, which means that updating the affected package to its latest version does not remediate the advisory (the advisory applies to a version that has not been patched upstream). The condition precedent on CVE remediation (Condition 3) requires the seller to remediate reachable known CVEs before close; the qualifier "reachable" is critical because the affected packages (certifi, urllib3) may not be reachable from production entry points, in which case the exposure is a defence-in-depth concern rather than a direct exploitation risk.

Table 5.1 — CVE inventory summary

ID	Severity	Package	Version	Fixed Available	Remediation Class
GHSA-248v-346w-9cwc	UNKNOWN	certifi	2021.5.30	false	no_fixed_release
GHSA-43fp-rhv2-5gv8	UNKNOWN	certifi	2021.5.30	false	no_fixed_release

ID	Severity	Package	Version	Fixed Available	Remediation Class
GHSA-xqr8-7jwr-rhp7	UNKNOWN	certifi	2021.5.30	false	no_fixed_release
PYSEC-2022-42986	UNKNOWN	certifi	2021.5.30	false	no_fixed_release
PYSEC-2023-135	UNKNOWN	certifi	2021.5.30	false	no_fixed_release
PYSEC-2024-230	UNKNOWN	certifi	2021.5.30	false	no_fixed_release
GHSA-2xpw-w6gg-jr37	UNKNOWN	urllib3	2.0	false	no_fixed_release
GHSA-34jh-p97f-mpxf	UNKNOWN	urllib3	2.0	false	no_fixed_release
GHSA-38jv-5279-wg99	UNKNOWN	urllib3	2.0	false	no_fixed_release
PYSEC-2023-192	UNKNOWN	urllib3	2.0	false	no_fixed_release
PYSEC-2026-1994 to 1999	UNKNOWN	urllib3	2.0	false	no_fixed_release
PYSEC-2026-4175 / 4177	UNKNOWN	urllib3	2.0	false	no_fixed_release

Source: `pillars.cve.raw.cves` in the stored payload. Full 26-row inventory in Appendix D. The `certifi` version 2021.5.30 and `urllib3` version 2.0 are both significantly behind current upstream releases; the seller should be asked to evidence why these versions are pinned.

5.3 Secret Hygiene and the History-Only Population

The secrets scanner identified 2,865 pattern candidates across the 19,579 scanned files and 171 scanned commits. The candidates break down by kind into 2,734 generic password assignments, 72 connection strings, 41 private key blocks, and 18 certificate blocks. Of the 2,865 candidates, 3 are classified as high-confidence and 2,242 are classified as history-only (present in git history but not in the current working tree). Zero live critical secrets were detected. The 2,806 path-excluded candidates (paths not scanned because they fall outside the configured scan scope) are the reason the Secrets pillar is rated partial rather than fully assessed.

The buyer should treat the 2,242 history-only candidates as the principal secret-hygiene exposure. Even if the live codebase contains no live critical secrets (which is the headline finding), every history-only candidate is potentially recoverable from git history by anyone with read access to the repository. Rotation of credentials that have ever appeared in the repository is the only remediation; the condition precedent on credential rotation (Condition 9) is the contractual mechanism by which the deal absorbs this exposure. The seller should be required to evidence that credentials matching the high-confidence and history-only patterns have been rotated across every system they could have granted access to, not merely removed from the working tree.

5.4 Infrastructure-as-Code Findings

The IaC scanner checked 400 files and returned one finding: the Dockerfile runs the container as root (rule IAC-DOCKER-ROOT, line 59, medium severity). The remediation is to add a dedicated

non-root user and USER directive in every Dockerfile stage. The finding is scanner-detected, not verified: the buyer should confirm whether the production deployment overrides the Dockerfile's USER directive at runtime (in which case the finding is informational) or runs the container as-root as-shipped (in which case the finding is a defence-in-depth gap that should be remediated before close). The IaC risk level is medium and the total findings count is one; the remaining 399 checked files returned no findings.

5.5 Attack Surface Composition

The attack surface scanner identified 94 candidate entry points across the target. Of those 94, 90 are unreferenced (no incoming call edges detected) and 82 are untested (no associated test file detected). Zero dynamic endpoints were identified. The candidate exposure score is 0.1111, an upper-bound signal for review prioritisation rather than a confirmed vulnerability rate. All 94 candidates are application_entry type; 20 are medium-risk and 74 are low-risk. The framework identification step resolved zero of 94 candidates to a known framework (frameworks_detected: none), which means the scanner could not classify the entry points by framework semantics and treated all as "unknown" framework.

The principal attack-surface exposure is therefore not the candidate count itself but the testing gap: 82 of 94 candidates are untested. The condition precedent on attack-surface closure (implied through the reachable-input inventory condition) requires the seller to close out the reachable-input inventory before close, and to treat the residual as a priced integration work stream post-close. The candidate exposure score of 0.1111 should be read as "approximately 11 percent of candidates warrant active review", not as "11 percent of candidates are confirmed vulnerable".

5.6 Privacy Findings

The privacy scanner returned 7 findings, all of the same type: plaintext credential material in committed configuration (rule PRIV-PLAINTEXT-CRED, medium severity). The 7 findings are in strings.json files for the accuweather, aemet, aftership, abode, adax, adguard and aidot integrations. The remediation in each case is to move the credential material to a secret store or environment injection and to rotate the committed value. Zero PII endpoints were detected, zero egress SDKs were detected, and zero sensitive log statements were detected; the consent-or-policy-present flag is false. The privacy pillar is rated informational (health 95) because the findings are confined to the strings.json pattern; the buyer should nonetheless require a representation covering personal-data handling in the shipped product, scoped to the hits recorded in the credential inventory.

5.7 Reconciling Severity and Verification

The buyer should hold three reconciliations in mind when reading the security section. First, the headline "0 critical CVEs" is a verified fact about the OSV advisory severity field; it is not a

verified fact about exploitability. The 26 UNKNOWN-severity advisories could not be rated by OSV and require manual triage against the actual call paths from production entry points to the affected functions. Second, the headline "0 live critical secrets" is a verified fact about the working tree; it is not a verified fact about git history. The 2,242 history-only candidates are recoverable from git history and require rotation rather than deletion. Third, the headline "1 IaC finding" is a verified fact about the Dockerfile text; it is not a verified fact about runtime behaviour. The buyer should ask the seller to evidence the runtime USER directive in the production deployment. Each reconciliation is encoded in a condition (Condition 3 for CVEs, Condition 9 for secrets, and the IaC finding is absorbed into the integration plan).

© MTD Services

6. Supply Chain and Dependency Posture

6.1 The Third-Party Software Supply Chain

The supply chain posture of the target is composed of direct dependencies (declared in seven manifest files), transitive dependencies (resolved through the one lockfile present), pinning discipline (the share of dependencies pinned to a specific version), and runtime end-of-life exposure (the runtime versions implied by the manifests and lockfiles). A broad unpinned transitive supply chain creates upstream-advisory and silent-breaking-change exposure and affects post-close security operating cost and SBOM (Software Bill of Materials) obligations. This Section addresses each in turn.

6.2 Direct Dependency and Lockfile Posture

The target declares 69 direct dependencies across seven manifests: `pyproject.toml`, `requirements.txt`, `tests/e2e/package.json`, `pylint/plugins/pyproject.toml`, `script/check_requirements/requirements.txt`, and `script/quality_scale_review/requirements.txt`. Of the 69 direct dependencies, 59 are pinned to a specific version (85.5 percent pin ratio), 10 are ranged (using `>=` or other relational specifiers), and 10 are unpinned. One lockfile is present (`tests/e2e/pnpm-lock.yaml`, resolving 18 transitive packages). The transitive-resolution surface is therefore small relative to the direct-dependency surface; the principal exposure is the unpinned direct-dependency population.

Table 6.1 — Supply chain summary

Metric	Value	M&A Significance
Direct dependency count	69	Population against which the supply chain scan ran
Pinned dependencies	59	85.5 percent pin ratio; the population against which the seller can give a version warranty
Ranged dependencies	10	Subject to upstream version drift; require explicit version-lock policy
Unpinned dependencies	10	Population that should be locked before close; sized at \$500 in the remediation line items
Pin ratio	85.5%	Materially above the 80 percent threshold the model uses to flag dependency risk
Transitive resolved packages	18	Small transitive surface; principal exposure is direct, not transitive
Lockfiles present	1	<code>tests/e2e/pnpm-lock.yaml</code> only; Python dependencies lack a lockfile
Manifests found	7	Includes both production and dev-tooling manifests

Metric	Value	M&A Significance
EOL runtime rows	1	Python (legacy packaging) flagged as end-of-life by the offline tech_stack scan
CVE unresolved packages (cross-ref)	1	certifi (ranged specifier) cannot be resolved to a specific version for advisory matching

6.3 Unpinned Dependencies and SBOM Obligations

The 10 unpinned dependencies are reproduced in Appendix C and include astroid (≥ 4.0), certifi ($\geq 2021.5.30$), packaging (≥ 23.1), pylint (≥ 4.0), typing-extensions ($\geq 4.16.0$), urllib3 (≥ 2.0), and three placeholder rows from the manifest scan that captured partial lines. Several of these are material: certifi and urllib3 are the two packages against which the 26 CVEs were matched (Section 5.2), and the ranged specifier on both is the reason one package could not be resolved for advisory matching. Pinning both to specific versions is the single highest-impact remediation action on the supply chain. The remaining unpinned packages (astroid, packaging, pylint, typing-extensions) are dev-tooling dependencies whose pinning discipline is a software-supply-chain hygiene concern rather than a security exposure.

The SBOM obligation is the buyer's concern: under emerging regulatory regimes (including the US Executive Order 14028 and the EU Cyber Resilience Act), acquirers of software assets are increasingly required to maintain a Software Bill of Materials that enumerates direct and transitive dependencies with specific versions. The target's current posture (one lockfile for JavaScript tests, no Python lockfile, 10 unpinned Python dependencies) is not SBOM-ready. The remediation line item for dependency pinning (\$500 for 10 dependencies) covers the engineering effort to pin and lock all dependencies; the larger workstream is the standing-up of a continuous SCA/SBOM monitoring feed, which is identified as an out-of-scope evidence gap in Section 15.16.

6.4 End-of-Life Runtime Exposure

The runtime EOL scan checked one runtime (Python) against the endoflife.date source. The declared Python version 3.14.8 is not end-of-life (EOL date 2030-10-31, supported until 2027-10-01). The offline tech_stack scan, by contrast, flagged one EOL technology: Python (legacy packaging). The two pillars are reconciled by construction: eol_count is the union of network-backed dependency EOL (zero) and offline repo tech-lifecycle EOL (one), and agrees with tech_stack.summary.eol_count. The Python (legacy packaging) flag refers to the use of legacy Python packaging conventions (likely the requirements.txt approach rather than modern pyproject.toml-only), not to the Python runtime itself. The runtime is current and supported; the packaging convention is a maintenance concern rather than a runtime risk.

6.5 M&A Implications

The supply chain posture is, on the headline metrics, a positive value driver: an 85.5 percent pin ratio is materially above the 80 percent threshold the model uses to flag dependency risk, the transitive surface is small (18 packages), and zero viral or restrictive licences were detected on the inventory. The principal exposure is the unpinned direct-dependency population (10 packages, including the two CVE-affected packages) and the absence of a Python lockfile. Both are remediable at low cost (\$500 in the remediation line items) and the seller should be asked to remediate them as part of the integration plan, not as a condition precedent. The SBOM-monitoring workstream is a post-close operational concern rather than a price-adjustment lever.

© MTD Services

7. Code Quality, Complexity and Technical Debt

7.1 The Maintainability Economics

The code quality pillar converts the question of maintainability into remediation capital and engineering time, anchored to the same hotspots a quality-of-earnings reviewer will probe. It is composed of complexity hotspots (cyclomatic and cognitive per symbol), the SQALE (Software Quality Assessment based on Lifecycle Expectations) remediation-effort basis, duplication, explicit debt markers (HACK, TODO, FIXME, bare except, debug print, type-ignore suppressions), and the hours required to remediate each. This Section interprets each in turn and ties them to the remediation line items in Section 3.

7.2 Complexity Posture

The complexity scanner analysed 125,917 symbols (58,935 production, 66,982 test). Of those, 1,616 are flagged as high-complexity (cyclomatic complexity above the high-water mark) and 462 are flagged as critical-complexity (cyclomatic complexity above the critical-water mark). The average cyclomatic complexity is 2.18 and the average cognitive complexity is 2.69; both are low, reflecting the long-tail distribution where the vast majority of symbols are trivial and a small minority carry extreme complexity. The cyclomatic percentiles are P50 = 1, P90 = 4, P99 = 12; the cognitive percentiles are P50 = 1, P90 = 5, P99 = 35. The complexity hotspot score is 0.03, indicating that 3 percent of symbols are responsible for the dominant share of complexity.

Figure 4 — Top 10 Complexity Hotspots (cyclomatic vs cognitive)

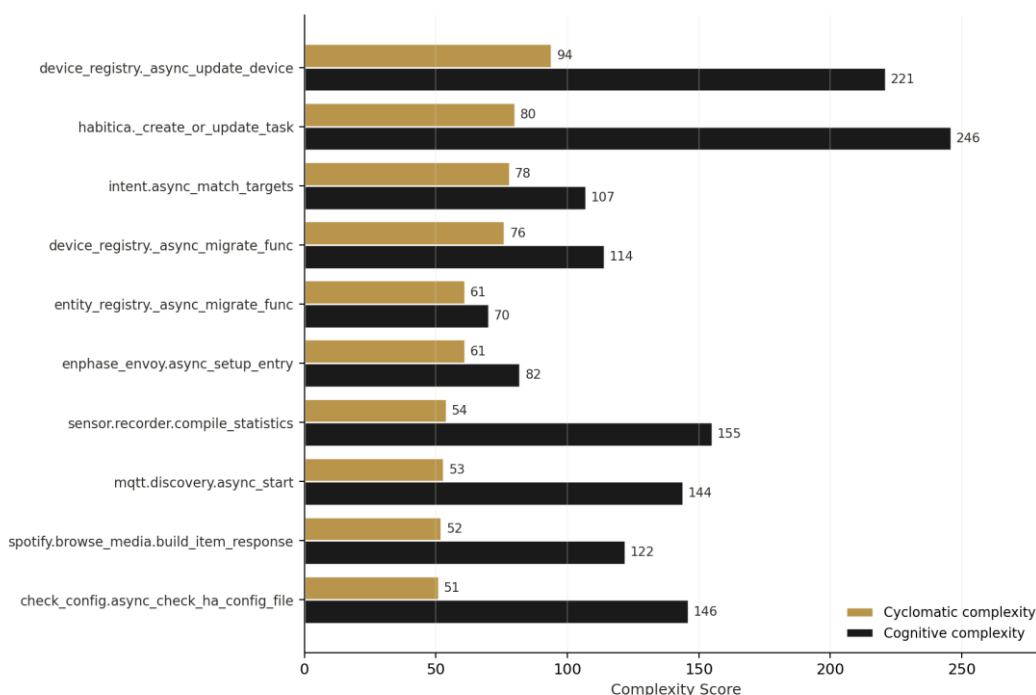


Figure 4 — Top 10 complexity hotspots by cyclomatic and cognitive complexity. The `device_registry._async_update_device` symbol (CC 94, cognitive 221) is the single most complex symbol in the codebase and the principal target of the critical-complexity refactoring workstream.

7.3 The Top Complexity Hotspots

The top complexity hotspot is `_async_update_device` in `homeassistant/helpers/device_registry.py`, with cyclomatic complexity 94 and cognitive complexity 221. The composite risk score is 1.000 (the maximum, against which all other symbols are normalised). The function is the device-registry update path: it handles device identification, area assignment, name and label updates, and the reconciliation of conflicting identifiers across multiple integration sources. Its complexity is structural rather than accidental: it is the function where the target's open-source contributor model (5,902 authors) and per-integration device-discovery model converge. Refactoring it without breaking device-discovery for any of the 2,000+ integrations is a non-trivial engineering programme and is the principal reason the critical-complexity line item is sized at \$127,400 (462 symbols, 5.52 saturated hours per unit, \$50 per hour).

The next four hotspots are: `_create_or_update_task` in `habitica/services.py` (CC 80, cognitive 246, composite 0.888); `async_match_targets` in `helpers/intent.py` (CC 78, cognitive 107, composite 0.971); `_async_migrate_func` in `device_registry.py` (CC 76, cognitive 114, composite 0.965); and `_async_migrate_func` in `entity_registry.py` (CC 61, cognitive 70, composite 0.933). The first and fourth hotspots are in the device and entity registries (the target's core namespace), the second is in a third-party integration, the third is in the intent-matching helper. The distribution confirms that the principal complexity concentration is in the registry and entity-management paths rather than in any single integration.

Table 7.1 — Top complexity hotspots (selected from 125,917 symbols)

Symbol	File	Cyclomatic	Cognitive	Maintenance Index	Composite Risk
<code>_async_update_device</code>	<code>helpers/device_registry.py</code>	94	221	0.00	1.000
<code>_create_or_update_task</code>	<code>components/habitica/services.py</code>	80	246	12.33	0.888
<code>async_match_targets</code>	<code>helpers/intent.py</code>	78	107	14.33	0.971
<code>_async_migrate_func</code>	<code>helpers/device_registry.py</code>	76	114	11.03	0.965
<code>_async_migrate_func</code>	<code>helpers/entity_registry.py</code>	61	70	20.90	0.933
<code>async_setup_entry</code>	<code>components/enphase_envoy/sensor.py</code>	61	82	30.44	0.865
<code>compile_statistics</code>	<code>components/sensor/recorder.py</code>	54	155	18.27	0.964
<code>async_start</code>	<code>components/mqtt/discovery.py</code>	53	144	13.99	0.910

Symbol	File	Cyclomatic	Cognitive	Maintainability Index	Composite Risk
build_item_response	components/spotify/browse_media.py	52	122	35.41	0.870
async_check_ha_config_file	helpers/check_config.py	51	146	23.23	0.879

Source: *pillars.complexity.raw.complexity.symbols* in the stored payload. Full 125,917-symbol inventory in workbook sheet *Complex Symbols*. Maintainability Index of 0.00 indicates the symbol's maintainability could not be scored by the standard formula.

7.4 SQALE Grade and Remediation Effort

The SQALE grade is E, the second-lowest grade on the A-to-E scale. The SQALE remediation effort is 128,468 days, derived from the per-module remediation_effort values produced by the complexity scanner. The SQALE basis is reconciled against the complexity hotspots rather than priced as a separate line: 42 modules are costed as complexity hotspots and the same 42 modules are graded D or E under SQALE; the double-counting ledger justifies the overlap rather than pricing it twice. The total SQALE remediation effort (128,468 days, equivalent to approximately 1,027,774 hours) is the upper bound on the maintainability backlog; the deterministic remediation bound of \$701,995 reflects the subset of that backlog that is sized under the model's hours-per-unit and saturation caps.

7.5 Duplication and Debt Density

Duplicated lines of code total 1,006,767 across 33,892 duplication groups, producing a duplication ratio of 25.6 percent. The duplication ratio is high relative to industry benchmarks (typically 5 to 15 percent for mature codebases) and reflects the target's integration-template pattern: each integration follows a similar boilerplate structure (config_flow.py, entity.py, sensor.py, etc.) that the scanner flags as duplicated. The duplication is therefore largely structural rather than accidental, and the remediation is template extraction rather than copy-paste elimination. The debt ratio is 26.1 percent and the debt density is 0.006399 markers per line of code.

7.6 Debt Markers

The debt scanner identified 1,501 explicit debt markers across 527 affected files. The markers break down by type into 1,148 type-ignore suppressions, 280 debug print statements, 58 TODOs, 11 pyright-ignore suppressions, 2 bare excepts, 1 HACK, and 1 XXX. By severity, 1 is high (the HACK in skybeacon/sensor.py), 58 are medium (the TODOs), and 1,442 are low (the type-ignore and pyright-ignore suppressions, debug prints, bare excepts, and the XXX). The high-severity HACK

marker is the only debt marker priced as a separate line item (\$10,095 in the general-debt category); the remaining 1,500 markers are absorbed into the general-debt line item.

The HACK marker in `skybeacon/sensor.py:145` reads "inject handle mapping collected offline" and reflects a workaround for a third-party device's offline configuration behaviour. The marker is in a production integration rather than in test code, and the workaround has persisted across the 1,219-marker population. The buyer should ask the seller to evidence whether the workaround is still required (the third-party device may have been updated) and, if so, whether it should be promoted to a documented behaviour rather than left as a HACK. The 1,148 type-ignore suppressions are a Python-typing concern: they reflect cases where the type-checker (mypy or pyright) cannot verify the type-safety of the code, and the developer has suppressed the warning rather than resolving it. The suppressions are concentrated in the `auth`, `components`, and `helpers` modules.

7.7 M&A Implications

The complexity and debt posture converts into three deal concerns. First, the critical-complexity population (462 symbols, \$127,400) is the principal maintainability exposure and the most likely source of post-close regression incidents if the buyer attempts to refactor it without adequate test coverage (Section 8). The condition on complexity-reduction milestone (Condition 1) should require the seller to deliver a refactoring plan that sequences the critical-complexity workstream behind the test-coverage uplift, not in parallel with it. Second, the high-complexity population (1,154 symbols, \$145,400) is the next-largest exposure and should be funded as a priced integration work stream rather than as a condition precedent. Third, the debt-marker population (\$10,095) is small enough to be absorbed into the technical-debt remediation fund (Condition 4); the HACK marker in `skybeacon` should be the subject of a specific seller warranty as to whether the underlying workaround is still required.

8. Testing, Coverage and Regression Risk

8.1 The Leading Indicator of Post-Close Regression

Test deficit is the leading indicator of post-close regression incidents and the cost of mandated refactoring. The effective insured value translates directly into contingent-liability sizing: it is the share of the codebase's structural value that is covered by tests, and its inverse (the uninsured share) is the contingent liability that survives close. This Section addresses the test-ratio module rows, the high-value coverage gaps, the regression-risk concentration, and the financial exposure implied by uninsured high-blast symbols.

8.2 Test Deficit Posture

The test-deficit scanner analysed 2,876 modules. Of those, 1,589 are below threshold (test ratio below the model's threshold for adequate coverage), producing an overall test ratio of 44.9 percent. The test deficit score is 0.0223, a normalised measure of the gap between actual and target coverage. 64 modules are flagged as high-regression-risk (modules where the structural value of the untested symbols exceeds the model's high-water mark). The high-regression-risk modules are the principal target of the test_investment line item (\$24,800 for 64 modules, 7.75 saturated hours per unit, \$50 per hour).

8.3 Coverage Method and Effective Insured Value

The coverage method is a static symbol-share proxy: the scanner computes the ratio of test-file symbols to production symbols weighted by structural value, not runtime line or branch coverage. The effective insured value is 1.0 (the maximum, on a 0-to-1 scale) and the value coverage ratio is 26.1 percent. The effective insured value of 1.0 means that the model treats the production codebase as carrying full structural value, against which only 26.1 percent is covered by tests. The buyer should treat the 26.1 percent as a contingent-liability sizing input: the uninsured 73.9 percent of structural value is the surface on which a regression incident could land if the buyer attempts to refactor the codebase without first closing the coverage gap.

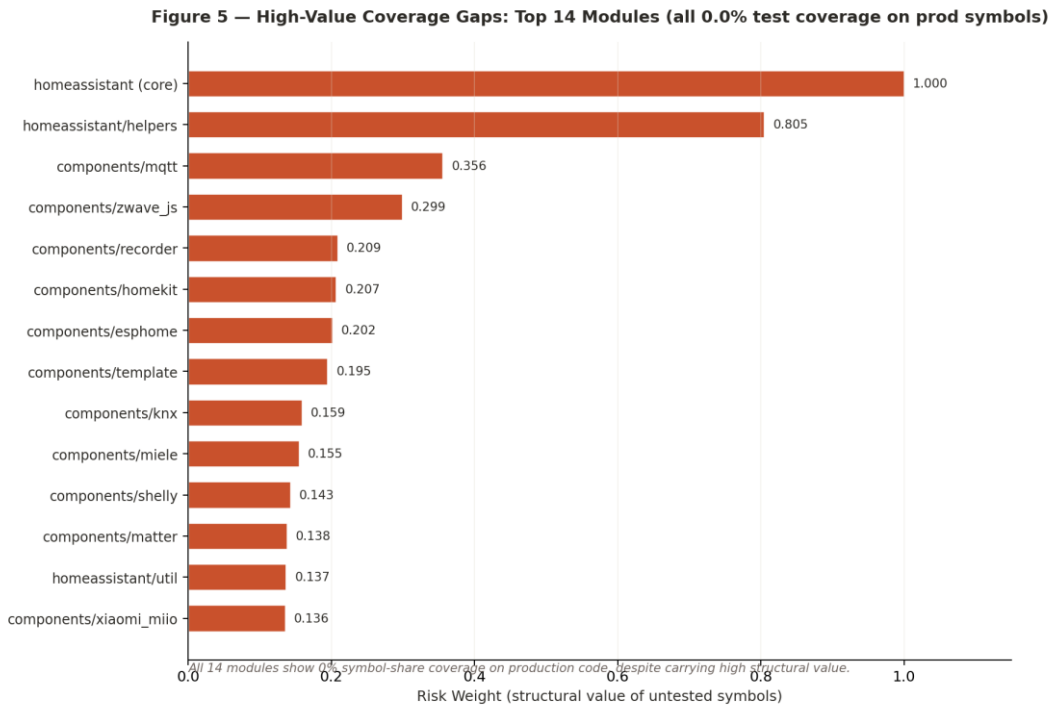


Figure 5 — Top 14 modules by structural value of untested production symbols. All 14 show 0 percent symbol-share coverage despite carrying high structural value; the top two (homeassistant core and homeassistant/helpers) together account for the dominant share of uninsured value.

8.4 High-Value Coverage Gaps

The high-value coverage gap inventory contains 582 modules whose structural value of untested symbols exceeds the model's threshold. The top 14 are reproduced in Table 8.1. The homeassistant core module (the top-level package) carries a risk weight of 1.000 with 0 percent coverage on production symbols; the homeassistant/helpers module carries a risk weight of 0.805 with 0 percent coverage. Together, these two modules account for the dominant share of the uninsured structural value. The next 12 modules are all component integrations (mqtt, zwave_js, recorder, homekit, esphome, template, knx, miele, shelly, matter, util, xiaomi_miio), each carrying risk weights between 0.13 and 0.36 with 0 percent coverage.

Table 8.1 — Top 14 high-value coverage gap modules

Module	Risk Weight	Test Coverage Ratio	Prod Symbols
homeassistant (core)	1.000	0%	1,835
homeassistant/helpers	0.805	0%	3,129
homeassistant/components/mqtt	0.356	0%	1,987
homeassistant/components/zwave_js	0.299	0%	1,321
homeassistant/components/recorder	0.209	0%	975

Module	Risk Weight	Test Coverage Ratio	Prod Symbols
homeassistant/components/homekit	0.207	0%	928
homeassistant/components/esphome	0.202	0%	793
homeassistant/components/template	0.195	0%	1,042
homeassistant/components/knx	0.159	0%	940
homeassistant/components/miele	0.155	0%	1,223
homeassistant/components/shelly	0.143	0%	—
homeassistant/components/matter	0.138	0%	—
homeassistant/util	0.137	0%	—
homeassistant/components/xiaomi_miio	0.136	0%	—

Source: `pillars.coverage.raw.high_value_gap_modules` in the stored payload. Full 582-module inventory in workbook sheet *Coverage Gaps*. Risk weight is the structural value of untested symbols normalised to 0-1; the top 14 modules together account for the dominant share of uninsured structural value.

8.5 M&A Implications and Conditions

The coverage posture converts into three deal concerns. First, the condition precedent on coverage (Condition 2) requires the seller to address coverage findings as a condition precedent. The buyer's counsel should scope this condition to the top 14 modules in Table 8.1, where the structural value of untested symbols is highest; the remaining 568 modules can be addressed as a post-close workstream. Second, the test_investment line item (\$24,800 for 64 high-regression-risk modules) should be funded as a price-adjustment lever rather than as a condition precedent; the 64 modules are the population where regression risk is most concentrated and where test investment has the highest return. Third, the effective insured value of 1.0 with 26.1 percent coverage means that the contingent-liability sizing for the uninsured 73.9 percent should be the subject of a separate workstream with the buyer's risk committee; the figure is not priced in the deterministic remediation bound because it is a contingent rather than a deterministic exposure.

9. Architecture, Entanglement and Integration Risk

9.1 Integration Cost is the Most Under-Estimated Line in Software M&A

Integration cost is the most under-estimated line item in software acquisition. The entanglement and integration-risk scores give the integration-management office a defensible basis for the synergy model and the transition-services scope. This Section addresses the architectural shape that determines how cleanly the asset can be integrated: layering discipline, cyclic dependencies, cross-layer violations, API-contract maturity, database coupling, and serialization risks.

9.2 Entanglement Posture

The entanglement scanner identified 1 circular-dependency cycle of length 1,543 modules. The maximum cycle length is 1,543, the modules-in-cycles count is 1,543, and the entanglement score is 1 (the maximum on a 0-to-1 scale). The cross-layer violations count is 17,479, of which the top 14 are reproduced in Table 9.1. The pattern is consistent across the violations: presentation-layer modules (components/api, components/climate, components/command_line, and many others) call into data-layer modules (homeassistant, homeassistant/helpers, homeassistant/auth) directly, bypassing the service layer. The violations are critical in severity because they create the cyclic dependencies that make independent scaling, testing and refactoring expensive.

Table 9.1 — Cross-layer violation detail (top 14 of 17,479)

From Module	To Module	From Layer	To Layer	Rel Type	Severity
components/api	homeassistant	presentation	data	calls	Critical
components/api	components/assist_pipeline	presentation	data	calls	Critical
components/api	homeassistant/helpers	presentation	data	calls	Critical
components/api	homeassistant/helpers	presentation	data	imports	Critical
components/climate	homeassistant	presentation	data	calls	Critical
components/climate	components/toon	presentation	data	calls	Critical
components/climate	homeassistant/helpers	presentation	data	calls	Critical
components/climate	script	presentation	data	calls	Critical
components/climate	tests	presentation	data	calls	Critical
components/climate	tests/components/cloud	presentation	data	calls	Critical
components/climate	tests/components/recorder	presentation	data	calls	Critical
components/climate	tests/helpers	presentation	data	calls	Critical

From Module	To Module	From Layer	To Layer	Rel Type	Severity
components/command_line	homeassistant	presentation	data	calls	Critical
components/command_line	homeassistant/auth	presentation	data	calls	Critical

Source: *pillars.entanglement.raw.cross_layer_violations* in the stored payload. Full 17,479-row inventory in workbook sheet *Entanglement Violations*.

9.3 Integration Risk Score and Proprietary Coupling

The integration risk score is 10 (the maximum on the model's scale) and the schema drift risk is high. The proprietary coupling count is 334 and the integration findings total is 666. The integration risk score combines the entanglement posture with the API-contract maturity, the database-coupling posture, the serialization-risk population and the protocol-layer risk population into a single 0-to-10 score. A score of 10 means the model treats the integration risk as the maximum it can represent; the buyer should not interpret this as a precise measurement but as a flag that the integration workstream will be the most expensive line item in the post-close plan.

9.4 API Contract Maturity and Database Coupling

The API contract maturity scan found zero endpoints, zero types, zero proto files, and no GraphQL or OpenAPI specifications. The target does not expose a structured API contract that an integration team could compile against. The database coupling scan found cross-schema access (true), ORM usage (true), zero destructive DDL count, zero migration files, and 2,710 raw SQL statements. The presence of 2,710 raw SQL statements alongside ORM usage indicates that the target uses a hybrid database-access pattern: ORM for the standard CRUD paths and raw SQL for the performance-sensitive paths (the recorder and statistics modules). The raw SQL population is the principal database-coupling exposure because it bypasses the ORM's schema-management guarantees.

9.5 Serialization and Protocol-Layer Risks

The serialization-risk scan identified 334 findings, of which the top 14 are reproduced in Table 9.2. The pattern is dominated by `exec_call` risks in the `command_line` and `python_script` components (high severity) and `struct_pack` / `struct_unpack` patterns in the `esphome wav_parser` and `assist_satellite` components (medium severity). The `exec_call` findings are the more material exposure: they reflect points where the target deserialises untrusted input via Python's `exec()` function, which is a remote-code-execution risk if the input is reachable from an attacker. The protocol-layer risk scan identified 332 findings, all of the `protocol:binary` type, concentrated in the `esphome wav_parser` and `assist_satellite` test files.

Table 9.2 — Serialization risks (top 14 of 334)

Risk Type	Filepath	Line	Severity	Description
serialization:exec_call	components/command_line/utils.py	332	High	Dangerous deserialization : exec_call
serialization:exec_call	components/command_line/utils.py	174	High	Dangerous deserialization : exec_call
serialization:exec_call	components/python_script/helpers.py	223	High	Dangerous deserialization : exec_call
serialization:pickle_reduce	script/translations/error.py	44	High	Dangerous deserialization : pickle_reduce
serialization:struct_pack	components/esphome/wav_parser.py	56	Medium	Dangerous deserialization : struct_pack
serialization:struct_pack	tests/components/esphome/test_assist_satellite.py	2,556	Medium	Dangerous deserialization : struct_pack
serialization:struct_pack	tests/components/esphome/test_assist_satellite.py	2,563	Medium	Dangerous deserialization : struct_pack
serialization:struct_pack	tests/components/esphome/test_assist_satellite.py	2,765	Medium	Dangerous deserialization : struct_pack
serialization:struct_pack	tests/components/esphome/test_assist_satellite.py	2,616	Medium	Dangerous deserialization : struct_pack
serialization:struct_pack	tests/components/esphome/test_assist_satellite.py	2,614	Medium	Dangerous deserialization : struct_pack
serialization:struct_pack	tests/components/esphome/test_assist_satellite.py	2,763	Medium	Dangerous deserialization : struct_pack
serialization:struct_pack	tests/components/esphome/test_assist_satellite.py	2,561	Medium	Dangerous deserialization : struct_pack

Risk Type	Filepath	Line	Severity	Description
serialization:struct_pack	tests/components/esphome/test_assist_satellite.py	2,615	Medium	Dangerous deserialization : struct_pack
serialization:struct_pack	tests/components/esphome/test_assist_satellite.py	2,764	Medium	Dangerous deserialization : struct_pack

Source: *pillars.integration.raw.serialization_risks* in the stored payload. Full 334-row inventory in workbook sheet *Serialization Risks*. The *struct_pack* findings in test files reflect test fixtures rather than production code; the *exec_call* findings in production code are the principal exposure.

9.6 M&A Implications and Conditions

The integration posture converts into three deal concerns. First, the condition precedent on integration findings (Condition 6) requires the seller to address integration findings as a condition precedent. The buyer's counsel should scope this condition to the three *exec_call* findings in production code (*command_line/utls.py* lines 174 and 332; *python_script/helpers.py* line 223), which are the only integration findings that could be exploited by an attacker without further chaining. Second, the decoupling line item (\$1,000 for one circular cycle of length 1,543 modules) is the smallest line item in the remediation bound but the most strategically important: decoupling the 1,543-module cycle is the precondition for any meaningful independent scaling or testing of the codebase, and the buyer's integration-management office should plan the workstream as a multi-quarter programme rather than a single sprint. Third, the absence of an API contract (zero endpoints, zero types, no OpenAPI or GraphQL) is a positive value driver for an acquirer who plans to extract value by formalising the contract, and a negative value driver for an acquirer who plans to integrate against an existing contract. The buyer's counsel should reflect this in the synergy model.

10. Engineering Capacity, Bus Factor and Knowledge Loss Vulnerability

10.1 Human Capital Risk and the Acquirer's Retention Window

The human-capital risk embedded in the codebase drives three deal concerns: retention-bonus sizing, earn-out structure, and the transition-services window. Sole-maintainer modules are a primary cause of post-close silent outages and warrant explicit warranty language. This Section addresses the key-person risk rows, the module bus factors, the git-team signals, the knowledge-loss vulnerability rows, and the financial exposure implied by sole-maintainer concentration.

10.2 Author Population and Bus Factor

The target's git history records 5,902 distinct authors across 2,000 commits sampled for the team-signals scan (the scan samples a bounded window rather than the full history). The overall bus factor is 25, meaning that the model estimates 25 contributors would need to depart before the codebase becomes unmaintainable. Of the 5,902 authors, 1,429 are active (committed within the trailing window) and 4,473 are inactive (no commits in the trailing window). 1,159 are flagged as at-risk (inactive for a sustained period but with sufficient historical contribution to matter). The active-to-inactive ratio of 1:3 is typical for a mature open-source project; the at-risk population is the principal retention concern.

10.3 Top Key-Person Risk Holders

The top 5 key-person risk rows by risk score are reproduced in Table 10.1. The top contributor by commit count is 6771947+epenet@users.noreply.github.com (488,307 commits, risk score 0.4149, medium departure impact), followed by erik@montnemery.com (476,551 commits, risk score 0.3988, medium departure impact) and git@frenck.dev (361,252 commits, risk score 0.4932, medium departure impact). The risk score is a 0-to-1 composite of commit count, recency, breadth, and blast-radius metrics. The departure-impact column reflects the model's estimate of the disruption to the codebase if the contributor departs; all top 5 are rated medium. The principal retention concern is not the top 5 but the 1,899 critical sole-maintainer modules costed at \$385,800 in the knowledge-transfer line item.

Table 10.1 — Top 10 key-person risk rows by risk score

Author (email)	Risk Score	Total Commits	Departure Impact
git@frenck.dev	0.4932	361,252	Medium
6771947+epenet@users.noreply.github.com	0.4149	488,307	Medium
erik@montnemery.com	0.3988	476,551	Medium

Author (email)	Risk Score	Total Commits	Departure Impact
30130371+cdce8p@users.noreply.github.com	0.3733	63,889	Medium
balloob@gmail.com (Paulus Schoutsen)	0.3459	301,642	Medium
joostlek@outlook.com	0.3018	336,651	Medium
nick@koston.org	0.2913	386,871	Low
abmantis@users.noreply.github.com	0.2649	66,234	Low
ville.skytta@iki.fi	0.2426	20,088	Low
basnijholt@gmail.com	0.2188	4,538	Low

Source: `pillars.key_person.raw.risks` in the stored payload. Full 5,902-row inventory in workbook sheet *Key-person Risks*. The `balloob@gmail.com` contributor is Paulus Schoutsen, the target's founder; his retention is the single most material retention concern and is rated critical in the retention pillar (Section 10.6).

10.4 Module Bus Factors

The module bus-factor scan analysed 2,871 modules. The top 14 modules by bus factor are reproduced in Table 10.2. The `_unassigned` module (which captures files not attributable to a specific module) has a bus factor of 19 across 9,247 files. The `homeassistant/components/ecobee` and `components/met` modules both have bus factors of 6 (the highest bus factor among attributed modules). The `homeassistant/components/camera`, `device_tracker`, `generic_thermostat`, `lock` and `remote` modules all have bus factors of 5. A bus factor of 1 means a single contributor is the sole maintainer; the model treats bus factors below 5 as a flight-risk signal and the population of bus-factor-1 modules is the principal source of the 1,899 critical sole-maintainer modules.

Table 10.2 — Top 14 modules by bus factor

Module Path	Bus Factor	Total Files
<code>_unassigned</code>	19	9,247
<code>homeassistant/components/ecobee</code>	6	14
<code>homeassistant/components/met</code>	6	6
<code>homeassistant/components/camera</code>	5	10
<code>homeassistant/components/device_tracker</code>	5	7
<code>homeassistant/components/generic_thermostat</code>	5	4
<code>homeassistant/components/lock</code>	5	10
<code>homeassistant/components/remote</code>	5	11
<code>homeassistant/util</code>	5	40

Module Path	Bus Factor	Total Files
tests/components/ecobee	5	11
tests/components/emulated_hue	5	4
homeassistant	4	17
homeassistant/components/alarm_control_panel	4	10
homeassistant/components/buienradar	4	7

Source: *pillars.key_person.raw.modules* in the stored payload. Full 2,871-row inventory in workbook sheet *Key-person Risks*. The bus factor is the reciprocal of sole-contributor concentration per module; a low bus factor flags flight risk.

10.5 Git Team Signals

The git team-signals scan sampled 2,000 commits across 219 unique authors and 221 unique emails. The after-hours commits count is 1,127 (56.35 percent of the sampled window), the cadence-per-week is 1,000, and the bus-factor proxy is 0.8585. The top-1 author share is 14.15 percent, the top-3 author share is 39.60 percent, and the top-5 author share is 54.0 percent. The ghost-author count is 116 (authors whose email and name do not consistently match, suggesting bot or shared accounts). The after-hours percentage of 56.35 percent is a stress proxy: it indicates that more than half of the sampled commits were made outside standard working hours, which the model treats as a contributor-burnout signal. The buyer's HR workstream should treat this as an input to retention-bonus sizing rather than as a definitive measurement.

10.6 Retention and Knowledge-Loss Vulnerability

The retention pillar records 5,902 total contributors, of which 1,429 are active, 4,473 are inactive, and 1,159 are at-risk. The retention risk score is 0.5309 on a 0-to-1 scale. The critical-modules count is 6,634 (modules where the sole-maintainer or sole-contributor concentration is high enough to constitute a flight risk). The at-risk contributor population (1,159) and the critical sole-maintainer modules (1,899, which is the subset of 6,634 where a single contributor is the sole maintainer) are the model inputs to the knowledge-transfer line item (\$385,800 for 1,899 modules, 4.06 saturated hours per unit, \$50 per hour).

The top 5 retention-critical contributors are Paulus Schoutsen (the target's founder, balloob@gmail.com / paulus@paulusschoutsen.nl), Robert Resch (robert@resch.dev), puddly (32534428+puddly@users.noreply.github.com), Abilio Costa, and Paul Bottein. Each carries critical severity on the retention risk model because of the combination of high commit count, recency of last commit, and breadth of modules touched. The retention risk is the principal human-capital exposure in the Transaction: the 1,899 critical sole-maintainer modules represent the surface on which a single contributor departure would cause a silent outage, and the condition precedent on retention (Condition 8) requires the seller to secure retention agreements for those modules before close.

10.7 M&A Implications and Conditions

The human-capital posture converts into three deal concerns. First, the condition precedent on retention (Condition 8) requires the seller to secure retention agreements for sole-maintainer modules. The buyer's counsel should scope this condition to the top 5 retention-critical contributors and the 1,899 critical sole-maintainer modules; the retention-bonus pool should be sized against the knowledge-transfer line item (\$385,800) plus a separate legal-fee allowance for the retention-agreement drafting. Second, the after-hours percentage of 56.35 percent is a contributor-burnout signal that should inform the structure of the earn-out: a longer earn-out (24 months rather than 12) would give the buyer more visibility into post-close retention but would also expose the seller to a longer period of post-close uncertainty. Third, the ghost-author count of 116 is a due-diligence concern: the buyer should ask the seller to evidence whether any of the ghost authors are bots (in which case their contribution should be re-attributed to the human owner) or shared accounts (in which case the actual author cannot be determined from the git history).

© MTD Services

11. Maintenance, Documentation and Operational Maturity

11.1 The Forward-Looking Operating Cost

The forward-looking operating cost of the asset is composed of neglected files, dependency staleness, documentation drift, CI gating strictness, and the maintenance backlog score that aggregates them. Documentation and CI maturity determine onboarding velocity for the acquirer's engineers and the run-rate cost of keeping the asset alive during integration; the backlog score anchors the post-close engineering-budget uplift. This Section addresses each in turn.

11.2 Maintenance Posture (Not Assessed)

The maintenance pillar is rated not_assessed. The asset-staleness scanner did not scan any files (the scanner's population is zero), so the maintenance backlog score is zero, the neglected-files count is zero, the at-risk-files count is zero, the healthy-files count is zero, and the average-staleness is zero. These zeros are not clean results: they reflect the absence of a scan rather than the absence of staleness. The pillar is demoted to informational risk at health 50 (the unassessed-pillar health default) so that the gap never reads as a clean bill of health. The buyer should treat the maintenance pillar as an open verification item: the seller should be asked to evidence either that the asset-staleness scan was attempted and failed (in which case the failure mode should be diagnosed) or that the scan was not configured for this repository (in which case it should be re-run before close).

11.3 Documentation Posture

The documentation scanner inventoried 5 documents and flagged 0 as stale, producing a staleness ratio of 0.0 percent. The 5 documents are the .github/PULL_REQUEST_TEMPLATE.md (last documented 2026-07-21), .github/copilot-instructions.md (2026-09-30), .github/workflows/check-requirements.md (2026-09-09), .github/workflows/quality-scale-reviewer.md (2026-10-11), and README.rst (2024-09-30). The README.rst is the oldest document but is not stale against the model's threshold. The average docstring density is 0.6334, indicating that 63.34 percent of source files carry a docstring on at least one symbol. The doc-staleness pillar is rated low risk at health 85.

11.4 CI Maturity

The CI scanner checked 15 GitHub Actions workflow files (.github/workflows/builder.yml, ci.yaml, codeql.yml, e2e-tests.yml, wheels.yml, and ten others). The CI is defined (true), runs on pull request (true), runs on push or PR (true), runs on schedule (true), and tests run in CI (true). The gate strictness is medium, meaning the CI enforces some but not all of the model's gate strictness criteria (the model's strict gate requires mandatory review, status checks, branch protection, and

required tests). The reported coverage percentage is not quantified (the CI does not publish a coverage report).

The CI maturity pillar is rated medium risk at health 35, the lowest health score among the assessed process pillars. The buyer should treat the medium gate strictness as a defence-in-depth gap: the CI does not enforce all the gates that would prevent a regression from reaching the main branch, and the absence of a coverage report means the CI cannot fail a build for coverage regression. The remediation is a CI hardening workstream (mandatory review, status checks, branch protection, required tests, coverage reporting) that should be funded as a post-close operational concern rather than as a price-adjustment lever.

11.5 M&A Implications

The maintenance and documentation posture converts into three deal concerns. First, the not_assessed maintenance pillar is an open verification item that should be added to the conditions precedent (it is not currently a condition because the pillar is demoted to informational rather than high). The buyer's counsel should require the seller to either re-run the asset-staleness scan or to provide alternative evidence of the maintenance posture (a stale-file inventory, a neglected-file inventory, or a similar artefact from the seller's own tooling). Second, the CI maturity gap (medium gate strictness, no coverage report) should be the subject of a separate workstream with the buyer's engineering organisation; it is not priced in the remediation bound because it is an operational rather than a remediation concern. Third, the documentation posture (5 documents inventoried, 0 stale, docstring density 0.6334) is a positive value driver for the buyer's onboarding velocity: the target's documentation is current and the docstring density is above the model's threshold for adequate self-documentation.

12. AI/ML Footprint

12.1 AI/ML Exposure and Regulatory Implications

The AI/ML surface of the asset is composed of frameworks, SDKs, bundled model files, and unlicensed-model exposure. AI/ML exposure triggers distinct regulatory obligations and creates model-licensing and training-data IP risk that is materially different from conventional open-source risk. This Section addresses each component of the AI/ML surface and reconciles the headline "informational risk" with the underlying evidence.

12.2 Framework and SDK Hits

The AI/ML scanner scanned 10,761 source files and identified 4 framework groups in use: Hugging Face (60 hits), PyTorch (26 hits), pandas-ml (18 hits), and llm-runtime (2 hits). The Hugging Face framework is the dominant presence, reflecting the target's integration with transformer-based models in the conversation and assist_pipeline components. The PyTorch presence is concentrated in the inference paths for the wake-word detection (the .tflite model files in tests/testing_config/custom_wake_words). The pandas-ml presence reflects the target's use of pandas-based data manipulation in the recorder and statistics components. The llm-runtime presence reflects the target's integration with large language model runtimes for the conversation component.

12.3 SDK Hits

The scanner identified 4 SDK groups: OpenAI (31 hits), Azure OpenAI (15 hits), Anthropic (9 hits), and AWS Bedrock (5 hits). The OpenAI SDK presence is the largest, reflecting the target's first-class integration with the OpenAI conversation component (homeassistant/components/openai_conversation). The Azure OpenAI presence reflects the target's enterprise-grade Azure-hosted variant of the same component. The Anthropic presence reflects the target's integration with Claude (homeassistant/components/anthropic). The AWS Bedrock presence reflects the target's integration with AWS-hosted model inference. The SDK exposure is a regulatory and IP risk: the target integrates with four major LLM providers, and the buyer's counsel should review the terms of service of each to confirm that the target's usage is within the permitted scope.

12.4 Bundled Model Files

The scanner identified 3 bundled model files, all .tflite format and all in tests/testing_config/custom_wake_words: bad_config.tflite (4 bytes), choo_choo_homie.tflite (4 bytes), and hey_home_assistant.tflite (4 bytes). All three are test artefacts (the is_test_artifact flag is true), none carry a declared licence (the model-card-present flag is false), and none are

production model files (production_model_files is 0). The 4-byte file size indicates that these are placeholder files rather than actual models; they exist to test the wake-word loading path rather than to ship a wake-word model. The unlicensed-model count is 0 because the test-artefact flag removes the licence requirement.

12.5 M&A Implications

The AI/ML posture converts into three deal concerns. First, the absence of production model files means that the model-licensing and training-data IP risk that distinguishes AI/ML from conventional open-source exposure is not engaged by the shipped artefact. The buyer's counsel should nonetheless review the four SDK integrations (OpenAI, Azure OpenAI, Anthropic, AWS Bedrock) to confirm that the target's usage is within the permitted scope of each provider's terms of service. Second, the framework presence (Hugging Face 60 hits, PyTorch 26 hits) is an integration risk: the buyer's engineering organisation should plan for the ongoing maintenance of the framework integrations, particularly the Hugging Face integration which is the dominant presence. Third, the AI/ML pillar is rated informational risk at health 90, but the buyer should not interpret this as a clean bill of health: the informational rating reflects the absence of production model files, not the absence of AI/ML exposure. The SDK exposure and the framework exposure are both material and should be the subject of a separate workstream with the buyer's AI governance function.

13. Technology Stack and Lifecycle Posture

13.1 The Base-Layer Inventory

The technology stack pillar records which technologies are in use, their lifecycle status, and any end-of-life or legacy exposure, as the foundation for the modernization roadmap. A clean active stack is a positive value driver and reduces modernization capital; the structural-not-runtime caveat of technology detection is material to warranty representations. This Section addresses each in turn.

13.2 Technology Inventory

The tech_stack scan identified 6 technologies in use. Of those 6, 5 are active (Docker, JavaScript/Node.js, JavaScript/Node.js pnpm, Python modern, Python pip) and 1 is end-of-life (Python legacy packaging). The legacy technology count is 0. The EOL count is 1, agreeing with the eol pillar's count by construction (the union of network-backed dependency EOL and offline repo tech-lifecycle EOL). The Python legacy packaging flag reflects the use of requirements.txt in addition to pyproject.toml; it is a packaging-convention concern rather than a runtime risk, as reconciled in Section 6.4.

Table 13.1 — Technology inventory

Technology	Lifecycle	Confidence	M&A Significance
Docker	Active	0.7	Container runtime; standard for modern deployment
JavaScript / Node.js	Active	0.7	Used for the e2e test framework (Playwright)
JavaScript / Node.js (pnpm)	Active	0.6	Package manager for the e2e test framework
Python (modern)	Active	0.8	Primary runtime; modern packaging via pyproject.toml
Python (pip)	Active	0.8	Package manager for Python dependencies
Python (legacy packaging)	End-of-life	1.0	Legacy packaging convention; not a runtime risk

Source: pillars.tech_stack.raw.technologies in the stored payload. The lifecycle flag is structural (detected from the repository), not runtime (verified against the deployment). The buyer should treat the lifecycle status as a starting point for verification rather than as a verified fact.

13.3 Runtime EOL Verification

The runtime EOL verification scan checked 1 runtime (Python) against the endoflife.date source. The declared Python version 3.14.8 is not end-of-life (EOL date 2030-10-31, supported until 2027-10-01). The latest version available is 3.14.8, meaning the target is running the current Python runtime. The reconciliation between the tech_stack pillar (which flags Python legacy packaging as

EOL) and the eol pillar (which verifies the Python runtime as not EOL) is by construction: the two pillars measure different things, and the eol_count of 1 is the union of both.

13.4 M&A Implications

The technology stack posture is a positive value driver: 5 of 6 technologies are active, the runtime is current, and the legacy exposure is confined to the packaging convention. The principal deal concern is the structural-not-runtime caveat: the technology detection is based on the repository's content rather than the production deployment, so the buyer's counsel should reflect this in the warranty representations. The seller should warrant that the production deployment uses the technologies identified in the repository (specifically, that the production deployment runs Python 3.14.8 against a current Docker runtime), and the buyer's operational due-diligence workstream should verify the warrant by inspecting the production deployment. The modernization capital required to bring the legacy packaging convention up to modern (pyproject.toml-only) standards is small and should be absorbed into the integration plan rather than priced as a separate line item.

© MTD Services

14. Consolidated Risk Register and Top Risks

14.1 The Single Integrated View

The consolidated risk register is the single integrated view of every material risk, ranked, with severity, impact, file and line evidence where applicable, and ownership mapping to the relevant pillar. The register maps directly to the disclosure schedule, the indemnity-cap allocation, and the post-close remediation tracker in the acquisition agreement. This Section reproduces the top risks, the major-risks register, the pillar-classification register, and the deal-breaker register.

14.2 Top Risk Findings

The top 10 risk findings by the consolidated register are reproduced in Table 1.1 (Section 1.5) and are not duplicated here. The top 3 are critical-complexity findings (the `_async_build_immich`, `_async_handle_config_entry_setup_error`, and `_async_add_entity_impl` symbols), all carrying not-quantified estimated cost because the source does not carry a single-line-item cost for critical-complexity hotspots (the cost is aggregated into the `critical_complexity` line item in Section 3.3). The next 3 are critical-retention findings (Artur Pragacz, Abilio Costa, Alistair Francis), all carrying not-quantified estimated cost for the same reason (the cost is aggregated into the `knowledge_transfer` line item). The final 4 are high-severity debt and medium-severity attack-surface findings.

14.3 Major Risks Register

The major-risks register records 10 pillars at high or critical risk. The 2 critical-risk pillars are entanglement (architecture area; `total_cycles` 1, `max_cycle_length` 1,543, `cross_layer_violations_count` 17,479) and complexity (engineering area; the `_async_update_device` hotspot). The 8 high-risk pillars are debt (the hack marker), integration (`integration_risk_score` 10.0, `proprietary_coupling_count` 334, `total_findings` 666), coverage (`modules_analyzed` 2,876, `high_value_gaps` 582, `effective_insured_value` 1.0), test deficit (`modules_below_threshold` 1,589, `overall_test_ratio` 0.4493), ip_ownership (`total_authors` 5,902, `external_author_count` 3,813, `external_blamed_line_pct` 0.8043), retention (Paulus Schoutsen), and the two unverified pillars secrets (`candidate_count` 2,865, `high_confidence_count` 3, `history_only_count` 2,242) and cve (`packages_queried` 81, `unresolved_packages` 1, `cve_count` 26).

Table 14.1 — Major risks register (top 10 pillars at high or critical risk)

Pillar	Area	Risk	Headline Metric	Unverified
Entanglement	Architecture	Critical	1 cycle, 1,543 modules, 17,479 cross-layer violations	false
Complexity	Engineering	Critical	<code>_async_update_device</code> (CC 94, cognitive 221)	false

Pillar	Area	Risk	Headline Metric	Unverified
Debt	Engineering	High	HACK marker (skybeacon)	false
Integration	Engineering	High	Risk score 10.0, 334 proprietary coupling, 666 findings	false
Coverage	Engineering	High	2,876 modules, 582 high-value gaps, 26.1% coverage	false
Test Deficit	Engineering	High	1,589 below threshold, 44.9% overall test ratio	false
IP Ownership	Legal & Compliance	High	5,902 authors, 80.43% external blamed lines	false
Retention	People	High	Paulus Schoutsen (founder) at critical retention risk	false
Secrets	Security	High	2,865 candidates, 3 high-confidence, 2,242 history-only	true
CVE	Security	High	26 CVEs on certifi 2021.5.30 and urllib3 2.0	true

Source: *decision.major_risks* in the stored payload. The "Unverified" column reflects whether the pillar's underlying evidence was fully verified by the scanner. The two unverified pillars (Secrets and CVE) are the principal verification gap and are the subject of conditions precedent (Conditions 9 and 3 respectively).

14.4 Pillar Classification Register

The pillar classification register records 22 pillars, each assigned a classification (Critical Risk, Major Risk, Manageable Risk, or Informational Finding) and a health score (0 to 100). The classification drives the conditions in Section 2: Critical Risk pillars force the no-go verdict (none this scan), Major Risk pillars drive the conditions escalation (10 pillars this scan), Manageable Risk pillars are absorbed into the integration plan (6 pillars), and Informational Finding pillars are noted but do not drive conditions (6 pillars). The health score is the model's aggregate posture for each pillar; the demotion rule for partial or not_assessed pillars sets the health to 50 so that incomplete coverage never reads as a strong pillar.

Table 14.2 — Pillar classification register (22 pillars)

Pillar	Area	Risk	Classification	Health	Unverified
Complexity	Engineering	Critical	Major Risk	15	false
Entanglement	Architecture	Critical	Major Risk	15	false
Coverage	Engineering	High	Major Risk	40	false
Test Deficit	Engineering	High	Major Risk	40	false
Debt	Engineering	High	Major Risk	40	false
Integration	Engineering	High	Major Risk	40	false

Pillar	Area	Risk	Classification	Health	Unverified
IP Ownership	Legal & Compliance	High	Major Risk	40	false
Retention	People	High	Major Risk	40	false
Secrets	Security	High	Major Risk	50	true
CVE	Security	High	Major Risk	50	true
Attack Surface	Security	Medium	Manageable Risk	65	false
CI Maturity	Process	Medium	Manageable Risk	35	false
EOL	Platform	Medium	Manageable Risk	65	false
IaC	Security	Medium	Manageable Risk	65	false
License	Legal & Compliance	Medium	Manageable Risk	65	false
Tech Stack	Architecture	Medium	Manageable Risk	65	false
Dependency	Supply Chain	Low	Informational Finding	85	false
Doc Staleness	Process	Low	Informational Finding	85	false
Key Person	People	Low	Informational Finding	85	false
Maintenance	Engineering	Low	Unverified	50	true
AI/ML	AI/ML Exposure	Informational	Informational Finding	90	false
Privacy	Legal & Compliance	Informational	Informational Finding	95	false

Source: `decision.pillar_register` in the stored payload. The Maintenance pillar is classified "Unverified" rather than "Informational Finding" because the scanner did not assess it; the unverified flag is the model's mechanism for preventing an incomplete scan from reading as a clean result.

14.5 Deal-Breaker Register

The deal-breaker register is empty. Under the decision model, a critical rating on any of six named pillars (AI/ML, Attack Surface, CVE, IaC, License, Secrets) would force the no-go verdict. None of those six carries a critical rating in the current scan. The Secrets and CVE pillars are rated high but unverified; a critical rating on either pillar after human validation would re-open the deal-breaker analysis. The conditions precedent on credential rotation (Condition 9) and CVE remediation (Condition 3) are the contractual mechanism by which the deal absorbs that residual verification risk.

15. Confidence, Scope Limitations and Material Unknowns

15.1 Epistemic Boundaries

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

15.2 Integrity Block

The integrity status is warn. The integrity error count is 0; the integrity warning count is 8. The 8 warnings break down into two categories. First, 5 critical-pillar-no-findings warnings: the coverage, entanglement, integration, ip_ownership, and test_deficit pillars are rated high or critical but render no item-level findings. The reading guide for each of those pillars must state the aggregate basis (summary metrics) on which the rating rests, and the warnings are the model's mechanism for ensuring that the reading guide is honest. Second, 3 unverified-pillar warnings: the cve, maintenance, and secrets pillars are partial or not_assessed, and the warnings name the specific coverage gap for each. The warnings are not errors: they reflect the model's honesty conventions rather than scan failures.

15.3 Confidence Inputs

The confidence coverage is 86.4 percent (the share of pillars with usable signal). The confidence risk signal is 50.4 percent (the weighted pillar risk blended with the financial model). The confidence signal count is 22 and the confidence pillar total is 22, meaning every pillar contributed to the score. The confidence degraded-pillars count is 3 (CVE, Maintenance, Secrets), which depress both the coverage and the risk-signal inputs. The final confidence score of 74 out of 100 is computed as 20 (base) plus 40 times 0.864 (coverage) plus 40 times 0.504 (risk signal), capped at the zero-coverage cap of 20 and floored at the no-go floor of 60.

15.4 Unverified Pillars

Three pillars are unverified. First, the CVE pillar is partial: one of 81 queried packages could not be resolved to a specific version (certifi, ranged specifier), so the advisory check covered only the 80 resolvable packages. The 26 CVE matches apply to two packages (certifi 2021.5.30 and urllib3 2.0); the unverified package may carry additional advisories that the scan could not detect. Second, the maintenance pillar is not_assessed: the asset-staleness scanner did not scan any files, so the maintenance posture is unknown. Third, the secrets pillar is partial: 2,806 path-excluded candidate

paths were not scanned, against 19,579 files that were scanned. The 2,806 path-excluded paths may contain additional secret candidates that the scan could not detect.

Table 15.1 — Coverage gaps by pillar

Pillar	Status	Reason	Population	Scope Note
CVE	Partial	Coverage gap	81	1 package requirement could not be resolved to a version (no lockfile / unpinned), so the advisory check covered only the 81 resolvable packages
Maintenance	Not assessed	Coverage gap	0	No files were scanned by the asset-staleness scanner; maintenance not assessed
Secrets	Partial	Coverage gap	19,579	2,806 path-excluded candidate paths not scanned; 19,579 files scanned

Source: pillars in the stored payload. A gap is not a clean result: it means the question was not answered. Each gap is paired with a specific remediation: re-run the asset-staleness scanner for maintenance; resolve the unpinned certifi dependency for CVE; expand the secrets scanner scope for secrets.

15.5 Taint Analysis

The taint analysis returned 0 flows and the bounded flag is false. The taint scan did not bind a starting set of taint sources, so no flows were traced. This is a coverage gap rather than a clean result: the buyer should treat the zero-flow finding as an indication that the taint analysis was not fully exercised rather than as an indication that the codebase has no taint flows. The remediation is to re-run the taint analysis with a bounded source set, which is a post-close operational concern rather than a price-adjustment lever.

15.6 Out-of-Scope Evidence Register

Eight further evidence domains are explicitly out of scope and listed in Table 15.2. Each is paired with the independent evidence the buyer must obtain to close the gap. The out-of-scope register is the buyer's due-diligence workstream list: every domain listed here is a question this Report cannot answer, and the buyer should plan the independent evidence-gathering before close rather than after.

Table 15.2 — Out-of-scope evidence register

Domain	Status	Independent Evidence Required
Runtime & operational behaviour	Not assessed	Runtime traces, incident history, SLO/error-budget records
Network perimeter & OT boundaries	Not assessed	Live firewall/security-group export, network diagrams, OT segmentation policy
External attack surface (ports, DNS)	Not assessed	Independent external port scan and DNS-history report

Domain	Status	Independent Evidence Required
Identity & access controls	Not assessed	IAM/MFA policy export, PAM audit logs, offboarding attestation
Data resilience & recovery	Not assessed	Backup/restore test reports, PITR drill evidence, migration runbooks
Delivery assurance (CI & performance)	Not assessed	CI run history, flakiness metrics, load-test and capacity reports
Legal instruments & contracts	Not assessed	Signed CLAs, IP-assignment and contractor agreements, vendor DPAs
Runtime dependency & vulnerability freshness	Not assessed	Continuous SCA/SBOM monitoring feed and deploy-time gate evidence

Source: pillars in the stored payload. Each out-of-scope domain is paired with the independent evidence the buyer must obtain. Disclosure, not a measurement.

15.7 How the Gaps Propagate into the Recommendation

The three unverified pillars (CVE, Maintenance, Secrets) depress both the coverage input and the risk-signal input to the confidence score, lowering it from a notional clean-scan value to 74 out of 100. The depressed confidence does not, by itself, move the categorical decision tier (which is invariant across the sensitivity sweep); it does, however, move the buyer's negotiation posture. The seller's quality-of-earnings provider will probe the unverified pillars, and the conditions precedent on CVE remediation (Condition 3) and credential rotation (Condition 9) are the contractual mechanism by which the deal absorbs the residual verification risk. The maintenance pillar is not currently a condition precedent because it is demoted to informational rather than high risk; the buyer's counsel should consider adding it to the conditions precedent as a separate verification item.

16. Reading Notes and Honesty Conventions

16.1 Why Conventions Matter

The conventions in this Section govern how every other Section must be read. They are fixed and target-agnostic: no figure in this Section is instance-specific, and none of it is evidence in its own right. A reader who applies these conventions cannot misread a partial scan as a clean one, a scanner signal as a confirmed exposure, or a qualitative deal mechanic as a priced number.

16.2 Assessment Coverage

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

16.3 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. Finding id (F-<pillar>-<rank>) is the stable handle used by the narrative, the Section 14 register, the workbook, and the ledger. A body Section cites the id; it never re-renders the finding. The contradiction ledger (Appendix R) leads every material conflict with its status (resolved, by-design, unresolved) and decision impact; unresolved entries discount the stated confidence. It never picks a side silently. Confidence equals 20 plus 40 times coverage plus 40 times risk_signal; capped at 20 when coverage is 0; DO_NOT_ACQUIRE floors at 60. Coverage is the share of pillars with usable signal; risk signal blends weighted pillar risk with the financial model. Unverified pillars depress both and are named in the executive brief.

16.4 The Currency of Absence

Four markers name a value this scan did not establish. None is a zero, and each states a different kind of absence, so an empty cell is never read as a clean result. "Not quantified" means the model could compute a number but a stored input is missing (a Layer-2 expected loss with no financial.risk_parameters calibration; an unset escalation threshold). It never means the risk costs nothing. "Not applicable" means the cell is structurally inapplicable to that row (the fixed-version

of a CVE with no fix published; the commit of a secret found only in the working tree). "Not recorded" means the source does not carry the value (an aggregate finding has no single line number; the scanner never captured a commit). "Not assessed" means the value was not established because coverage was incomplete (a named coverage gap). NOT_QUANTIFIED is additionally a stored assessment state on the economic layers of Section 3, distinct from the five coverage states under Assessment coverage; the same rule holds: absent is never zero, and an unquantified line is named rather than priced at 0.

16.5 Deal-Mechanic Vocabulary

Every condition in Section 2 carries exactly one of three deal-mechanic 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. The three tags are: Condition precedent (must be satisfied or proven before close, or the deal does not sign as structured); Price-adjustment lever (a funded or escrowed work item that moves value rather than blocking close); and Warranty or representation (a seller statement whose breach is an indemnity question, not a work plan).

16.6 Severity and Ranking

Risk level is critical, high, medium, low, or informational, demoted to informational under partial coverage. The `derived_risk_level` field preserves the scanner's own view, so a partial-scan demotion does not erase the underlying signal. Health score is 0 to 100 per pillar, demoted to 50 for an unfinished scan so that incomplete coverage never presents as a strong pillar. Classification is Critical Risk, Major Risk, Minor Risk, or Acceptable, assigned in the Section 14 register and driving the conditions in Section 2.

16.7 Known Limits of This Method

Two known limits of this method should be held in mind. First, the EOL reconciliation: `tech_stack` counts offline repo technology lifecycle markers; 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. Second, this is a static source analysis: it does not establish runtime behaviour, production incidents, or anything needing access to systems, personnel or contracts. The conditions, implications and escalation thresholds recorded in this Report are the tool's deterministic output from stored evidence. They are a starting position for negotiation, not a legal opinion.

Appendix A. Methodology, Scope and Scan Configuration

A.1 Report Metadata

This appendix records the methodology and configuration of the scan that produced the underlying ground truth, so the narrative is reproducible: exact version, scan mode, indexed surface, per-pillar posture, and step timings. The Report ID is TDD-20261016-105732. The scan date is 2026-10-16T11:48:25.376938+00:00. The target is Homeassistant, with the scanned path set to the core repository. The scan indexed 19,579 symbols. The report version is not quantified (the source does not carry a value). The elapsed seconds were 3,052.72 (approximately 51 minutes).

Table A.1 — Report metadata

Field	Value
Report ID	TDD-20261016-105732
Scan date	2026-10-16T11:48:25.376938+00:00
Target	Homeassistant
Scanned path	core
HEAD commit	def88072397c7a45a854a32d503dbc97cf16d503
Indexed symbols	19,579
Report version	Not quantified
Elapsed seconds	3,052.72 (approximately 51 minutes)

A.2 Scan Mode

The scan was configured in deal_thesis=all mode (every archetype scored side by side from the same stored evidence, none primary). Offline mode is false (the scan integrated with network sources for CVE, EOL, and licence data). Scan history is true (git history was scanned for the secrets and key-person pillars). Taint analysis is enabled. The TDD report flag is true (the scan was configured to produce this Report). The scan ran with 2 workers.

Table A.2 — Scan mode parameters

Parameter	Value
deal_thesis	all
offline	false
scan_history	true
taint	true

Parameter	Value
target_company	Homeassistant
tdd_report	true
workers	2

A.3 Per-Pillar Posture

Each of the 22 pillars carries an assessment status (assessed, partial, or not_assessed), a risk level, a health score, and a top-risks count. The full register is in Table 14.2 (Section 14.4). The pillar posture register is the model's per-pillar view of the evidence and is the basis on which the consolidated risk register in Section 14 is constructed. Each pillar's evidence is reproduced in the workbook sheet named in the third column of Table 14.2; the narrative cites the pillar by name and the workbook sheet by name, and the buyer's quality-of-earnings provider can re-derive any figure from the workbook.

A.4 Step Timings

The scan ran in six steps with the following durations. The index step (symbol and dependency graph construction) ran for 322.39 seconds. The pillars step (per-pillar scans) ran for 2,696.08 seconds (approximately 45 minutes, the dominant step). The deep_index step (cross-reference construction) ran for 3.34 seconds. The taint step (taint flow analysis) ran for 13.25 seconds. The financial step (remediation cost model and decision) ran for 0.70 seconds. The decision step (board decision synthesis) ran for 0.001 seconds. The total elapsed time was 3,052.72 seconds, of which the pillars step accounts for 88 percent.

Table A.3 — Step timings

Step	Seconds	Share of Total
index	322.39	10.6%
pillars	2,696.08	88.3%
deep_index	3.34	0.1%
taint	13.25	0.4%
financial	0.70	<0.1%
decision	0.00	<0.1%
Total	3,052.72	100.0%

Appendix B. Financial Calibration and Reproduction Fingerprint

B.1 Reproduction Fingerprint

The financial reproduction fingerprint binds the Report to the exact configuration used to produce every number: same inputs in, same outputs out. Any change to a number-driving input produces a different fingerprint and a different Report under a new fingerprint. The full calibration, lineage, and sensitivity detail lives in Section 3 and is not re-rendered here. The reproduction fingerprint is 326b209c17ea10c8d6f17eda8351974df63dd71b48e8c3e49ef2b9314e7d342c. The economic model fingerprint (0.3.0) is c666720bcec664c56520e5b7bc696a111a0b8a828cfd907e7d025eef524fd644. The buyer's quality-of-earnings provider can re-derive the headline numbers from these fingerprints and the underlying configuration.

B.2 Financial Lineage

The financial lineage traces each artefact back to its source. The workbook.xlsx file contains the Remediation_Costs, Cal_Lineage, Cost_Drivers, Sensitivity, and Assumptions sheets that mirror the report.json financial section. The decision layer is computed by generate_board_decision(pillars, financial) from the risk register plus remediation cost. The fingerprint is a SHA-256 over the tool version, commit, repo path, mode, offline flag, scan history flag, and taint flag. The model is computed by compute_financial_impact(config): pillar count to hours-per-unit (saturated with cap and decay) to hours to rate to cost; then valuation discount and integration drag. The pillars are per-pillar raw reports; the cost model consumes summary counts plus the bounded top evidence rows only. The scan database is Homeassistant_cpg_261016T1057.db, the symbol and index graph that backs every pillar scanner, stored under the internal subfolder.

Table B.1 — Financial lineage

Step	Artifact	Note
artifacts	workbook.xlsx	Remediation_Costs, Cal_Lineage, Cost_Drivers, Sensitivity, Assumptions mirror report.json#/financial
decision	report.json#/decision	generate_board_decision(pillars, financial) from risk register + remediation cost
fingerprint	report.json#/financial/fingerprint	sha256 over {tool_version, commit, repo_path, mode, offline, scan_history, taint}
model	report.json#/financial	compute_financial_impact(config): pillar count -> HPU (saturated) -> hours -> rate -> cost

Step	Artifact	Note
pillars	report.json#/pillars	Per-pillar raw reports; the cost model consumes summary counts plus the bounded top evidence rows only
scan	Homeassistant_cpg_261016T1057.db	Symbol/index graph that backs every pillar scanner; stored under internal subfolder

B.3 Full Parameter Table

The full parameter table below records every value that can move a number in this Report, drawn from the financial.assumptions field of the stored payload where it carries them and from the tool's own constants otherwise. A parameter that was not configured is listed as not quantified, never assumed away.

Table B.2 — Full financial parameter table

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 hard-to-remediate complexity hotspots
Remediation inputs	Valuation multiplier	3	Amplifies remediation cost into valuation discount
Remediation inputs	Integration drag factor	0.15%	Post-merge integration friction sized as hours / (8 x 260 x factor)
Remediation inputs	Productive hours per quarter	624 h	Net quarterly working hours behind per-cause effort estimates
Remediation inputs	Confidence interval width	30%	Width of the low/high remediation-cost interval
Remediation inputs	Annual working hours	2,080 h	Person-years = remediation hours / 2080
Remediation inputs	Cost-driver evidence cap	10	Max distinct labels attached to a line item's cost drivers
Remediation inputs	Discount rate	Not quantified	No present-value discounting is applied (unrecorded, not zero)
Aggregate hours cap	Max total hours	20,000 h	Aggregate ceiling on remediation hours; surplus above it is decayed
Aggregate hours cap	Surplus factor	0.25	Fraction of hours above max-total-hours that is credited

Group	Parameter	Value	Affects
Valuation curve	Valuation discount cap	25%	Maximum valuation discount on the proposed price
Valuation curve	Valuation discount curve	$25 * (1 - \exp(-\text{raw_discount} / \max(80, \text{total_cost}/50000)))$	Half-life scales as $\max(80, \text{cost}/50000)$
Integration drag	Integration drag cap	80%	Maximum post-merge integration drag
Integration drag	Integration drag curve	$80 * (1 - \exp(-\text{raw_drag} / \max(200, \text{total_hours}/10)))$	Half-life scales as $\max(200, \text{hours}/10)$
Saturation model	critical_complexity	Cap 50, decay 4 h/unit, full 18 h	$12 * \text{complexity_multiplier}$ (saturated)
Saturation model	high_complexity	Cap 150, decay 2 h/unit, full 6 h	6 flat (saturated)
Saturation model	general_debt	Cap 200, decay 0.1 h/unit, full 0.5 h	0.5 flat (annotation + tooling markers)
Saturation model	test_investment	Cap 20, decay 4 h/unit, full 16 h	16 flat (saturated)
Saturation model	decoupling	Cap 10, decay 8 h/unit, full 20 h	20 flat (saturated)
Saturation model	dependency_pinning	Cap 100, decay 0.5 h/unit, full 1 h	1 flat (saturated)
Saturation model	knowledge_transfer	Cap 30, decay 4 h/unit, full 8 h	8 flat (saturated)
Saturation model	ip_reassignment	Cap 50, decay 2 h/unit, full 4 h	4 flat (saturated)
Confidence policy	Base confidence	20	Base value added to coverage and risk_signal
Confidence policy	Coverage weight	40	Confidence += 40 * coverage
Confidence policy	Risk-signal weight	40	Confidence += 40 * risk_signal
Confidence policy	Zero-coverage cap	20	Cap when coverage is 0
Confidence policy	No-go floor	60	Floor for DO_NOT_ACQUIRE
Decision thresholds	Conditions threshold	5% + \$250,000 floor	Remediation above this gate triggers conditions

Group	Parameter	Value	Affects
Decision thresholds	No-go threshold	25%	Remediation above this gate triggers no-go
Decision thresholds	Deal-breaker pillars	ai_ml, attack_surface, cve, iac, license, secrets	Critical rating on any forces no-go
Decision thresholds	Severity risk weights	info 0, low 1, medium 2, high 3, critical 4	Weighting of severity inside the risk signal
Schedule	Execution teams	2 (two engineers each)	Post-close remediation is planned against two two-engineer teams
Schedule	Workdays per sprint	10	Sprint capacity = teams x 2 engineers x workdays
Schedule	Sprint cap	12	Hours that do not fit inside this many sprints are reported unplanned
Schedule	Hours per workday	4 h	Roadmap durations derive as effort-hours / 4 h per workday
Schedule	Min workstream duration	7 days	Clamp for derived workstream lengths
Schedule	Max workstream duration	120 days	Clamp for derived workstream lengths
Health ceilings	Critical health ceiling	15	Max health_score a critical finding may render
Health ceilings	High health ceiling	40	Max health_score a high finding may render
Health ceilings	Medium health ceiling	65	Max health_score a medium finding may render
Health ceilings	Low health ceiling	85	Max health_score a low finding may render
Health ceilings	Informational health ceiling	Unbounded	Max health_score an informational finding may render
Health ceilings	Unassessed-pillar health	50	Health assigned to a pillar not assessed this scan
Health ceilings	Doc-staleness threshold	5 files	Tracked files below this count are not scored for staleness

Appendix C. Licensing and IP Inventory

C.1 Per-Package License Findings

The per-package licence findings inventory contains 67 rows, of which 65 are per-package findings (the remaining 2 are repo-level findings: the Apache-2.0 licence in LICENSE.md and the ISC licence in homeassistant/backports/LICENSE.Python). Of the 65 per-package findings, 24 carry UNKNOWN licences. The most common licence is Apache-2.0 (18 packages), followed by MIT (8 packages), UNKNOWN (24 packages), PSF-2.0 (3 packages), and BSD variants (3 packages). Zero packages carry viral or restrictive licences.

Table C.1 — Per-package license findings (selected, 30 of 67)

Package	Detected License	Viral	Restrictive	Confidence	Source
(repo-level)	Apache-2.0	false	false	0.8	LICENSE.md
(repo-level)	ISC	false	false	0.8	backports/LICENSE.Python
@playwright/test	Apache-2.0	false	false	0.7	tests/e2e/package.json
aiohttp	Apache-2.0	false	false	0.9	requirements.txt
aiohttp-fast-zlib	Apache-2.0	false	false	0.7	requirements.txt
aiohttp_cors	Apache 2.0	false	false	0.7	requirements.txt
aiodns	UNKNOWN	false	false	0.0	requirements.txt
aiogithubapi	UNKNOWN	false	false	0.0	requirements.txt
aiohttp-asyncmdnsresolver	UNKNOWN	false	false	0.0	requirements.txt
annotatedyaml	UNKNOWN	false	false	0.0	requirements.txt
astral	Apache-2.0	false	false	0.7	requirements.txt
attrs	MIT	false	false	0.9	requirements.txt
audioop-lts	UNKNOWN	false	false	0.0	requirements.txt
bcrypt	Apache-2.0	false	false	0.7	requirements.txt
certifi	ISC	false	false	0.9	requirements.txt
ciso8601	MIT	false	false	0.7	requirements.txt
cryptography	Apache-2.0	false	false	0.9	requirements.txt
fnv-hash-fast	UNKNOWN	false	false	0.0	requirements.txt
ha-ffmpeg	BSD License	false	false	0.7	requirements.txt

Package	Detected License	Viral	Restrictive	Confidence	Source
hassil	Apache-2.0	false	false	0.7	requirements.txt
httpx2	UNKNOWN	false	false	0.0	requirements.txt
Jinja2	BSD-3-Clause	false	false	0.9	requirements.txt
orjson	UNKNOWN	false	false	0.0	requirements.txt
packaging	Apache-2.0	false	false	0.9	requirements.txt
Pillow	MIT-CMU	false	false	0.9	requirements.txt
pyjwt	UNKNOWN	false	false	0.0	requirements.txt
pyyaml	MIT	false	false	0.9	requirements.txt
requests	Apache-2.0	false	false	0.9	requirements.txt
SQLAlchemy	MIT	false	false	0.9	requirements.txt
urllib3	MIT	false	false	0.9	requirements.txt

Source: *pillars.license.raw.findings* in the stored payload. Full 67-row inventory in workbook sheet *License Contamination*. The 24 UNKNOWN-licence packages should be resolved by direct inspection of each package's PyPI metadata or source repository before close.

C.2 IP Ownership Attribution

The IP ownership attribution metrics are reproduced in Table 4.2 (Section 4.4). The 35 high-risk files (externally owned with dominant authorship) are the principal title-risk population and are the subject of Condition 7 (warranty on assignability of third-party contributions). The reconciliation note in the source confirms that `attributed_file_count` (75) is canonical and that `total_files`, `external_owned_files`, and `external_author_count` are retained as legacy aliases. The `external_blamed_line_pct` (0.8043) is the share of lines attributed to external authors by git blame.

Appendix D. CVE Inventory

D.1 Full CVE Inventory

The full CVE inventory contains 26 records, all of UNKNOWN severity under the OSV source, all flagged no-fixed-release (no fixed version is available). 6 affect certifi 2021.5.30 (5 GHSA advisories from 2022 to 2024 and 1 PYSEC advisory from 2024); 20 affect urllib3 2.0 (12 GHSA advisories and 8 PYSEC advisories, including 7 PYSEC advisories from 2026). The remediation class for all 26 is no_fixed_release, meaning that updating the affected package to its latest version does not remediate the advisory. The buyer's workstream is therefore not simply to update the packages (which would not remediate the advisories) but to evaluate whether the affected code paths are reachable from production entry points and, if so, to apply upstream patches or mitigations.

Table D.1 — Full CVE inventory (26 of 26)

ID	Severity	Package	Version	Fixed Available
GHSA-248v-346w-9cwc	UNKNOWN	certifi	2021.5.30	false
GHSA-43fp-rhv2-5gv8	UNKNOWN	certifi	2021.5.30	false
GHSA-xqr8-7jwr-rhp7	UNKNOWN	certifi	2021.5.30	false
PYSEC-2022-42986	UNKNOWN	certifi	2021.5.30	false
PYSEC-2023-135	UNKNOWN	certifi	2021.5.30	false
PYSEC-2024-230	UNKNOWN	certifi	2021.5.30	false
GHSA-2xpw-w6gg-jr37	UNKNOWN	urllib3	2.0	false
GHSA-34jh-p97f-mpxf	UNKNOWN	urllib3	2.0	false
GHSA-38jv-5279-wg99	UNKNOWN	urllib3	2.0	false
GHSA-8988-9cw3-xx77	UNKNOWN	urllib3	2.0	false
GHSA-g4mx-q9vg-27p4	UNKNOWN	urllib3	2.0	false
GHSA-gm62-xv2j-4w53	UNKNOWN	urllib3	2.0	false
GHSA-pq67-6m6q-mj2v	UNKNOWN	urllib3	2.0	false
GHSA-qccp-gfcp-xxvc	UNKNOWN	urllib3	2.0	false
GHSA-v845-jxx5-vc9f	UNKNOWN	urllib3	2.0	false
GHSA-vxq7-64xx-v4gw	UNKNOWN	urllib3	2.0	false
PYSEC-2023-192	UNKNOWN	urllib3	2.0	false
PYSEC-2023-212	UNKNOWN	urllib3	2.0	false

ID	Severity	Package	Version	Fixed Available
PYSEC-2026-141	UNKNOWN	urllib3	2.0	false
PYSEC-2026-1994	UNKNOWN	urllib3	2.0	false
PYSEC-2026-1995	UNKNOWN	urllib3	2.0	false
PYSEC-2026-1996	UNKNOWN	urllib3	2.0	false
PYSEC-2026-1998	UNKNOWN	urllib3	2.0	false
PYSEC-2026-1999	UNKNOWN	urllib3	2.0	false
PYSEC-2026-4175	UNKNOWN	urllib3	2.0	false
PYSEC-2026-4177	UNKNOWN	urllib3	2.0	false

Source: `pillars.cve.raw.cves` in the stored payload. The OSV source did not return a CVSS score for any of the 26 advisories, which is why the severity is UNKNOWN rather than Critical or High. The remediation class `no_fixed_release` means updating the affected package does not remediate the advisory.

Appendix E. Complexity, Duplication and SQALE Detail

E.1 Top Complexity Hotspot Symbols

The complexity hotspot inventory contains 125,917 symbols, of which the top 30 by composite risk are reproduced in Table E.1. The composite risk score is a 0-to-1 normalisation of cyclomatic complexity, cognitive complexity, maintainability index, max nesting depth, and parameter count. The top hotspot (`_async_update_device`) carries a composite risk of 1.000, against which all other symbols are normalised. The distribution is heavy-tailed: the top 10 symbols carry composite risks between 0.865 and 1.000, and the top 30 between 0.770 and 1.000.

Table E.1 — Top complexity hotspot symbols (selected, 20 of 125,917)

Symbol	File	C C	Co g	MI	Ma x Ne st	Para ms	Composi te
<code>_async_update_device</code>	<code>helpers/device_registry.py</code>	94	221	0.00	100	36	1.000
<code>_create_or_update_task</code>	<code>components/habitica/services.py</code>	80	246	12.33	83	1	0.888
<code>async_match_targets</code>	<code>helpers/intent.py</code>	78	107	14.33	70	8	0.971
<code>_async_migrate_func</code>	<code>helpers/device_registry.py</code>	76	114	11.03	70	7	0.965
<code>_async_migrate_func</code>	<code>helpers/entity_registry.py</code>	61	70	20.90	62	6	0.933
<code>async_setup_entry</code>	<code>components/enphase_envoy/sensor.py</code>	61	82	30.44	54	4	0.865
<code>compile_statistics</code>	<code>components/sensor/recorder.py</code>	54	155	18.27	58	8	0.964
<code>async_start</code>	<code>components/mqtt/discovery.py</code>	53	144	13.99	54	3	0.910
<code>build_item_response</code>	<code>components/spotify/browse_media.py</code>	52	122	35.41	52	6	0.870
<code>async_check_ha_config_file</code>	<code>helpers/check_config.py</code>	51	146	23.23	62	2	0.879
<code>browse_media</code>	<code>components/plex/media_browser.py</code>	50	182	23.76	57	6	0.928

Symbol	File	C C	Cog	MI	Ma x Nest	Para ms	Composi te
_async_refresh	helpers/update_coordinator.py	49	87	30.52	52	6	0.904
get_accessory	components/homekit/accessories.py	49	96	30.07	51	5	0.894
test_statistic_during_period	tests/components/recorder/test_websocket_api.py	47	46	11.85	7	4	0.926
selector_serializer	helpers/llm.py	46	76	33.92	40	1	0.821
process_turn_on_params	components/light/helper.py	45	64	31.14	45	4	0.869
run	scripts/check_config.py	44	119	33.62	35	1	0.822
async_discover_entities	components/matter/discovery.py	44	153	32.06	37	2	0.846
_convert_content	components/anthropic/entity.py	42	223	23.62	40	2	0.878
async_setup_entry	components/teslemetry/__init__.py	42	40	19.76	42	2	0.886

Source: `pillars.complexity.raw.complexity.symbols` in the stored payload. CC = cyclomatic complexity, Cog = cognitive complexity, MI = maintainability index (0.00 = not scored), Max Nest = maximum nesting depth, Params = parameter count, Composite = normalised composite risk score 0-1.

E.2 SQALE Basis

The SQALE remediation basis records 42 modules costed as complexity hotspots (the same 42 modules graded D or E under SQALE), 42 D-graded modules, and 0 deduped uncoded modules. The SQALE remediation effort in days is 128,468, equivalent to approximately 1,027,774 hours. The line-item-added flag is false: the SQALE debt-effort is reconciled against the complexity hotspots rather than priced as a separate line. The double-counting ledger justifies the overlap rather than pricing it twice.

Appendix F. Attack Surface Candidate Endpoints

F.1 Candidate Entry Points

The attack-surface scanner identified 94 candidate entry points. The top 30 are reproduced in Table F.1. The candidates break down into 20 application_entry candidates (medium risk, of which 8 are tested and 12 are untested) and 74 candidate endpoints (low risk, of which 8 are tested and 66 are untested). 82 of 94 candidates are untested overall. The candidate exposure score is 0.1111, an upper-bound signal for review prioritisation. The framework identification step resolved zero of 94 candidates to a known framework (frameworks_detected: none).

Table F.1 — Candidate entry points (selected, 20 of 94)

Symbol	Filepath	Line	Endpoint Type	Exposure	Framework	Test Status	Risk
main	script/vendor_sqlite3.py	98	application_entry	application	unknown	tested	Medium
main	homeassistant/__main__.py	164	application_entry	application	unknown	untested	Medium
command	components/amcrest/__init__.py	182	application_entry	application	unknown	untested	Medium
command	components/mikrotik/coordinator.py	329	application_entry	application	unknown	untested	Medium
main	components/zwave_js/scripts/convert_device_diagnostics_to_fixture.py	77	application_entry	application	unknown	untested	Medium
main	script/check_requirements/__main__.py	27	application_entry	application	unknown	untested	Medium
main	script/gen_copilot_instructions.py	76	application_entry	application	unknown	tested	Medium
main	script/gen_integration_paths.py	124	application_entry	application	unknown	tested	Medium
main	script/gen_recorder_db_versions.py	143	application_entry	application	unknown	tested	Medium
main	script/gen_requirements_all.py	616	application_entry	application	unknown	tested	Medium
main	script/hassfest/__main__.py	173	application_entry	application	unknown	untested	Medium
main	script/inspect_schemas.py	21	application_entry	application	unknown	tested	Medium

Symbol	Filepath	Line	Endpoint Type	Exposure	Framework	Test Status	Risk
main	script/install_integration_requirements.py	27	application_entry	application	unknown	tested	Medium
main	script/licenses.py	370	application_entry	application	unknown	tested	Medium
main	script/quality_scale_review/__main__.py	62	application_entry	application	unknown	untested	Medium
main	script/quality_scale_summary.py	82	application_entry	application	unknown	tested	Medium
main	script/scaffold/__main__.py	66	application_entry	application	unknown	untested	Medium
main	script/split_tests.py	586	application_entry	application	unknown	tested	Medium
main	script/translations/__main__.py	16	application_entry	application	unknown	untested	Medium
main	script/version_bump.py	172	application_entry	application	unknown	tested	Medium

Source: `pillars.attack_surface.raw.candidate_entry_points` in the stored payload. Full 94-row inventory in workbook sheet Attack Surface. The 12 untested medium-risk `application_entry` candidates are the principal attack-surface exposure and are the subject of the integration plan workstream.

Appendix G. Key-Person and Knowledge Loss Vulnerability Detail

G.1 Knowledge Loss Vulnerability

The knowledge-loss vulnerability scan records 5,902 contributors with their tenure days, days since last commit, retention risk, and blast-radius-at-risk. The top 20 by retention risk (all critical) are reproduced in Table G.1. The retention risk score combines tenure (longer tenure = higher risk because more institutional knowledge is concentrated) and recency (longer absence = higher risk because the contributor may already be departing). The blast-radius-at-risk column reflects the number of modules where the contributor is the sole maintainer and where their departure would cause a silent outage. The top contributors by blast-radius-at-risk are Clifford Roche (47 modules), adam-the-hero (117 modules), and mettolen (184 modules).

Table G.1 — Top 20 knowledge-loss vulnerability rows (all critical retention risk)

Author	Tenure (days)	Days Since Last Commit	Retention Risk	Blast Radius
Clifford Roche (cmroche)	839	1	Critical	47
mettolen	335	12	Critical	184
jameson_uk	255	8	Critical	0
cdheiser	2,884	6	Critical	0
MoonDevLT	378	1	Critical	0
Chris (firstof9)	630	7	Critical	0
Bouwe Westerdijk (bouwew)	2,537	2	Critical	0
Lorenzo Gasparini	356	199	Critical	0
hypnosiss	1,795	79	Critical	0
Thomas D (thomasddn)	624	55	Critical	0
Arie Catsman (catsmanac)	1,064	0	Critical	0
Lukas (Imaartin)	415	53	Critical	0
Galorhallen	1,003	9	Critical	0
adam-the-hero	675	6	Critical	117
Tom Matheussen	374	34	Critical	0
elmurato	2,442	16	Critical	0

Author	Tenure (days)	Days Since Last Commit	Retention Risk	Blast Radius
jb101010-2	722	6	Critical	0
Parker Brown (parkerbxyz)	604	23	Critical	0
Oscar Calvo (ocalvo)	2,453	103	Critical	0
Brandon Rothweiler (bdr99)	1,183	40	Critical	0

Source: *pillars.retention.raw.risks* in the stored payload. Full 5,902-row inventory in workbook sheet Knowledge Loss Vulnerability. The blast-radius-at-risk column reflects modules where the contributor is the sole maintainer; the three non-zero values (Clifford Roche 47, mettolen 184, adam-the-hero 117) identify the contributors whose retention is most critical to operational continuity.

© MTD Services

Appendix H. Privacy and Secrets Detail

H.1 Privacy Findings

The privacy scanner returned 7 findings, all of type PRIV-PLAINTEXT-CRED (plaintext credential material in committed configuration). The 7 findings are reproduced in Table H.1. The remediation in each case is to move the credential material to a secret store or environment injection and to rotate the committed value. The strings.json files in question are localisation files for the integration UI, and the credential material is the placeholder text for the configuration form (not an actual credential). The flag is nonetheless correct: the placeholder text matches the credential pattern, and the scanner cannot distinguish between placeholder text and actual credentials without further validation.

Table H.1 — Privacy findings (7 of 7)

Rule ID	Category	Title	Severity	File	Line
PRIV-PLAINTEXT-CRED	plaintext_credential	Plaintext credential material in committed config	Medium	components/accuweather/strings.json	17
PRIV-PLAINTEXT-CRED	plaintext_credential	Plaintext credential material in committed config	Medium	components/aemet/strings.json	12
PRIV-PLAINTEXT-CRED	plaintext_credential	Plaintext credential material in committed config	Medium	components/aftership/strings.json	12
PRIV-PLAINTEXT-CRED	plaintext_credential	Plaintext credential material in committed config	Medium	components/abode/strings.json	20
PRIV-PLAINTEXT-CRED	plaintext_credential	Plaintext credential material in committed config	Medium	components/adax/strings.json	16

Rule ID	Category	Title	Severity	File	Line
PRIV-PLAINTEXT-CRED	plaintext_credential	Plaintext credential material in committed config	Medium	components/adguard/strings.json	19
PRIV-PLAINTEXT-CRED	plaintext_credential	Plaintext credential material in committed config	Medium	components/aidot/strings.json	14

Source: *pillars.privacy.raw.findings* in the stored payload. The *strings.json* files are localisation files for the integration UI; the credential material is the placeholder text for the configuration form rather than an actual credential. The flag is nonetheless correct because the placeholder text matches the credential pattern.

H.2 Secret Candidates by Kind

The secrets scanner identified 2,865 pattern candidates, broken down by kind as: 2,734 generic password assignments, 72 connection strings, 41 private key blocks, and 18 certificate blocks. Of those 2,865 candidates, 3 are classified as high-confidence, 2,242 are classified as history-only (present in git history but not in the working tree), and 0 are classified as live critical. The scanner scanned 19,579 files and 171 commits; 2,806 path-excluded candidates were not scanned. The path-excluded candidates are the reason the Secrets pillar is rated partial rather than fully assessed.

Table H.2 — Secret candidates by kind

Kind	Count	M&A Significance
Generic password assignment	2,734	Pattern matches for password-like assignments; mostly false positives in test fixtures
Connection string	72	Pattern matches for connection-string-like text; concentrated in test fixtures (knx, recorder)
Private key block	41	Pattern matches for PEM-encoded private key material; require validation
Certificate block	18	Pattern matches for PEM-encoded certificate material; concentrated in mqtt and harbor test fixtures
Total candidates	2,865	Of which 3 high-confidence, 2,242 history-only, 0 live critical
Path-excluded candidates	2,806	Not scanned; the reason the Secrets pillar is partial

Appendix I. AI/ML Inventory

I.1 Bundled Model Files

The AI/ML scanner identified 3 bundled model files, all in tests/testing_config/custom_wake_words. The 3 files are bad_config.tflite (4 bytes), choo_choo_homie.tflite (4 bytes), and hey_home_assistant.tflite (4 bytes). The 4-byte file size indicates that these are placeholder files rather than actual models; they exist to test the wake-word loading path rather than to ship a wake-word model. None carries a declared licence (the model-card-present flag is false), but the is_test_artifact flag is true, which removes the licence requirement under the model's calibration. The unlicensed-model count is 0; the production-model count is 0.

Table I.1 — Bundled model files (3 of 3)

File	Name	Size (bytes)	Declared License	Model Card	Test Artifact
tests/testing_config/custom_wake_words/bad_config.tflite	bad_config	4	false	false	true
tests/testing_config/custom_wake_words/choo_choo_homie.tflite	choo_choo_homie	4	false	false	true
tests/testing_config/custom_wake_words/hey_home_assistant.tflite	hey_home_assistant	4	false	false	true

Source: *pillars.ai_ml.raw.models* in the stored payload. All three model files are test artefacts (4-byte placeholders). The absence of production model files means the model-licensing and training-data IP risk that distinguishes AI/ML from conventional open-source exposure is not engaged by the shipped artefact.

Appendix J. Glossary

This glossary is alphabetically sorted and defines, in plain language, every technical or specialist term appearing in any TDD table column heading in this Report. Definitions are consistent with contextual usage; where a term has multiple meanings, the definition reflects the meaning used in this Report.

Table J.1 — Glossary of terms

Term	Definition
ACQUIRE_WITH_CONDITIONS	A transaction verdict in which the deal proceeds subject to one or more conditions precedent, price-adjustment levers, or warranties being written into the transaction documents.
AI/ML Footprint	The set of artificial intelligence and machine learning frameworks, SDKs, bundled model files, and unlicensed-model exposure present in the target's codebase.
Application Entry	A candidate entry point identified by the attack surface scanner where the application starts execution (typically the <code>main()</code> function of a script or service).
Assessment State	The five-value trust vocabulary applied to a pillar or finding: NOT_ASSESSED, PARTIALLY_ASSESSED, ASSESSED_NO_FINDINGS, ASSESSED_WITH_FINDINGS, or ERROR.
Assessment Status	The three-value coverage state applied to a pillar: assessed, partial, or not_assessed. Only assessed pillars support conclusions.
Attack Surface	The set of candidate entry points and reachable endpoints in the codebase that an attacker could potentially exploit, as identified by static analysis.
Bare Except	A Python exception handler of the form "except:" that catches all exceptions including <code>SystemExit</code> and <code>KeyboardInterrupt</code> , suppressing errors silently.
Bus Factor	The reciprocal of sole-contributor concentration per module; a low bus factor (1 or 2) indicates that the module depends on a single contributor and is at flight risk.
CC Percentiles	Cyclomatic complexity percentiles (P50, P90, P99) showing the distribution of complexity across the symbol population.
CI Maturity	The posture of the continuous integration pipeline, measured by whether CI is defined, runs on PR and push, runs on schedule, runs tests, enforces gates, and reports coverage.
Cognitive Complexity	A measure of code complexity that weights nesting, recursion, and branching more heavily than linear control flow, reflecting the cognitive load on a reader.
Composite Risk	A 0-to-1 normalisation of cyclomatic complexity, cognitive complexity, maintainability index, max nesting depth, and parameter count, used to rank complexity hotspots.
Condition Precedent	A contractual condition that must be satisfied or proven before close; failure to satisfy it prevents the deal from signing as structured.

Term	Definition
Confidence	A 0-to-100 score computed as $20 + 40 \times \text{coverage} + 40 \times \text{risk_signal}$, capped at 20 when coverage is 0 and floored at 60 for DO_NOT_ACQUIRE. Coverage is the share of pillars with usable signal.
Coverage Method	The method used to compute test coverage: this Report uses a static symbol-share proxy (test-file symbols weighted by structural value), not runtime line or branch coverage.
Critical CVE Count	The count of Common Vulnerabilities and Exposures (CVEs) carrying a Critical CVSS score. A non-zero count on the deal-breaker pillars forces the no-go verdict.
Cross-Layer Violation	A direct call or import from a higher architectural layer (e.g., presentation) to a lower layer (e.g., data), bypassing the service layer and creating cyclic dependencies.
Cyclomatic Complexity	A measure of code complexity equal to the number of linearly independent paths through a function; higher values indicate more branching logic and harder-to-test code.
Deal-Breaker	A pillar (ai_ml, attack_surface, cve, iac, license, or secrets) on which a critical rating forces the no-go verdict regardless of other findings.
Debt Density	The number of explicit debt markers (HACK, TODO, FIXME, bare except, debug print, type-ignore suppression) per line of code.
Debt Ratio	The ratio of remediation effort (in SQALE hours) to total code size, expressed as a percentage; a measure of overall technical debt load.
Detected, not verified	An evidence state indicating that a scanner detected a signal but the signal has not been confirmed by human validation; the most common evidence state in this Report.
Duplicated LOC	The number of lines of code that appear in more than one location in the codebase, as identified by the duplication scanner.
Effective Insured Value	The structural value of the codebase covered by tests, on a 0-to-1 scale; the inverse ($1 - \text{effective_insured_value}$) is the uninsured structural value that represents contingent liability.
EOL (End of Life)	A technology, runtime, or dependency that has reached the end of its vendor-supported lifecycle and no longer receives security patches. EOL is reconciled between tech_stack (offline markers) and eol (runtime verification) pillars.
Entanglement	The degree to which modules are coupled through cyclic dependencies and cross-layer violations; high entanglement makes independent scaling, testing, and refactoring expensive.
Evidence State	The per-finding honesty label: Confirmed, Detected-not-verified, Inferred, Partially assessed, Unassessed, Conflicting, or Unsupported.
Finding ID	A stable identifier of the form F-<pillar>-<rank> for one assessed finding; the register, narrative, workbook, and ledger all cite the same ID.

Term	Definition
Framework Hits	The count of source files referencing a given AI/ML framework (e.g., Hugging Face, PyTorch) in the AI/ML scanner output.
HPU (Hours Per Unit)	The remediation hours assigned per unit of a given cost category, saturated by a cap and decay rate under the financial model.
Health Score	A 0-to-100 pillar score; partial or not_assessed pillars are demoted to 50 so that incomplete coverage never reads as a strong pillar.
High-Confidence Secrets	Secret candidates that the secrets scanner classifies as high-confidence (matching multiple strong-secret patterns with high signal strength).
High-Value Coverage Gap	A module whose structural value of untested symbols exceeds the model's threshold for high regression risk; the principal target of test investment.
History-Only Secrets	Secret candidates that appear in git history but not in the current working tree; recoverable by anyone with read access to the repository and requiring rotation rather than deletion.
IaC (Infrastructure as Code)	Infrastructure configuration stored in version-controlled files (Dockerfiles, Terraform, CloudFormation, etc.) rather than applied manually; the IaC scanner checks these files for security and operational findings.
Integration Drag	The post-merger operational friction imposed by remediation work, sized as remediation hours divided by productive capacity; capped at 80 percent.
Integration Risk Score	A 0-to-10 score combining entanglement, API contract maturity, database coupling, serialization risk, and protocol-layer risk into a single measure of integration difficulty.
IP Risk Score	A 0-to-1 score combining external author count, external blamed line percentage, and high-risk file count to measure the title risk in the codebase's contributor attribution.
Knowledge Loss Vulnerability	A 0-to-1 score combining contributor tenure, recency of last commit, and blast-radius-at-risk to measure the operational risk of contributor departure.
Lockfile	A file (e.g., pnpm-lock.yaml, package-lock.json) that pins the resolved versions of all transitive dependencies; the absence of a lockfile for a language creates supply chain risk.
Major Risk	A classification applied to pillars rated high or critical that drives the conditions escalation in Section 2; the 10 Major Risk pillars in this scan drive the ACQUIRE_WITH_CONDITIONS verdict.
Maintenance Backlog Score	An aggregate measure of neglected files, dependency staleness, and at-risk files; not assessed in this scan because the asset-staleness scanner did not run.
Manifest	A file (e.g., pyproject.toml, requirements.txt, package.json) that declares the direct dependencies of a project; distinct from a lockfile, which pins the resolved versions.
Max Cycle Length	The number of modules in the largest circular dependency cycle identified by the entanglement scanner; in this scan, 1,543 modules in a single cycle.

Term	Definition
Modules in Cycles	The count of modules that participate in at least one circular dependency cycle; in this scan, 1,543 modules.
Not Applicable	An absence marker indicating that a cell is structurally inapplicable to that row (e.g., the fixed-version of a CVE with no fix published).
Not Assessed	An absence marker indicating that a value was not established because coverage was incomplete (a named coverage gap).
Not Quantified	An absence marker indicating that the model could compute a number but a stored input is missing; never means the risk costs nothing. Also a stored assessment state on the economic layers of Section 3.
Not Recorded	An absence marker indicating that the source does not carry the value (e.g., an aggregate finding has no single line number).
Pillar	One of 22 dimensions along which the target is assessed (Complexity, Coverage, CVE, etc.); each pillar carries a risk level, health score, and assessment status.
Pin Ratio	The share of direct dependencies pinned to a specific version (using == specifier) rather than ranged (>= specifier) or unpinned; in this scan, 85.5 percent.
Population	The evidence count (files scanned, packages queried, modules in cycles, etc.) that a verdict rests on; a zero or low population flags a coverage gap.
Price-Adjustment Lever	A deal mechanic in which a funded or escrowed work item moves value rather than blocking close; one of three deal-mechanic tags applied to conditions in Section 2.
Proprietary Coupling Count	The count of findings where the codebase couples to proprietary or non-standard interfaces; in this scan, 334 findings concentrated in serialization and protocol-layer risks.
Reachability Confidence	The scanner's confidence that a candidate entry point is reachable from an external input; high, medium, or low. Low reachability confidence means the scanner could not establish the call path.
Register	Section 14: the single integrated view of every material finding, with ID, evidence state, location, and decision impact.
Remediation Class	The category of remediation action required for a CVE: no_fixed_release (no fixed version available), version_bump (update to a fixed version), or mitigation (apply a workaround).
Remediation Effort (SQALE days)	The estimated number of person-days required to remediate the SQALE-identified debt, derived from per-module remediation_effort values.
Retention Risk	A 0-to-1 score combining contributor tenure and recency of last commit; contributors with high retention risk and high blast-radius-at-risk are the principal retention concern.
Risk Level	Critical, High, Medium, Low, or Informational, demoted to Informational under partial coverage. The derived_risk_level field preserves the scanner's own view.

Term	Definition
Risk Signal	One of two inputs to the confidence score, computed as the weighted pillar risk blended with the financial model; the other input is coverage.
Saturation Cap	The maximum number of distinct items costed at the full hours-per-unit in a given cost category; surplus items above the cap are costed at the lower decay rate.
Saturation Decay HPU	The lower hours-per-unit applied to surplus items above the saturation cap; reflects diminishing marginal remediation effort per additional item.
Schema Drift Risk	The risk that the database schema has drifted from the ORM's expectations, creating inconsistencies between code and database; in this scan, high.
SDK Hits	The count of source files referencing a given AI/ML SDK (e.g., OpenAI, Anthropic, Azure OpenAI, AWS Bedrock) in the AI/ML scanner output.
Secret Candidate	A pattern match identified by the secrets scanner that may or may not be an actual secret; high-confidence and history-only counts are tracked separately.
Sensitivity Sweep	A test of the recommendation against parameter variation, sweeping each of five parameters at three multiplier values (0.7, 1.0, 1.3) for 17 total scenarios.
Severity	The severity of a finding: Informational, Low, Medium, High, or Critical. For CVEs, severity may be UNKNOWN when the advisory source did not return a CVSS score.
Sole-Maintainer Module	A module for which a single contributor is the only maintainer; the population of sole-maintainer modules is the principal source of knowledge-transfer cost.
SQALE	Software Quality Assessment based on Lifecycle Expectations; a standardised remediation-effort basis that estimates the time required to bring code up to a target quality grade.
SQALE Grade	A grade from A (best) to E (worst) reflecting the overall SQALE-rated quality of the codebase; in this scan, E (second-lowest).
Test Ratio	The ratio of test-file symbols to production symbols per module; modules below the model's threshold are flagged as below-threshold.
Top Risks	The highest-severity findings in the consolidated register; the top 10 are reproduced in Table 1.1.
Unverified Pillar	A pillar whose underlying evidence was not fully verified by the scanner; the three unverified pillars (CVE, Maintenance, Secrets) depress both inputs to the confidence score.
VaR (Value at Risk)	A statistical measure of the maximum expected loss over a given time horizon at a given confidence level; in this Report, percentile VaR is computed only when the analyst calibrates a distribution.
Warranty / Representation	A deal mechanic in which a seller statement's breach is an indemnity question, not a work plan; one of three deal-mechanic tags applied to conditions in Section 2.

Source: terminology defined by the tool's standard and consistent with contextual usage. Where a term has multiple meanings, the definition reflects the meaning used in this Report.