POLARIS INTELLIGENCE

Technical Due Diligence Report

bitwarden

03 April 2026

Ref: 77508cd6

CONFIDENTIAL

1. Executive Summary

Repository Classification: Production (Confidence: high)
This repository is assessed as production software. All scanner findings are reported at full weight.

Credential/secret findings: 6 findings detected in scanned files. Review recommended to confirm exposure.

1 known vulnerability found in 1 of 115 dependencies. Outdated or vulnerable packages increase attack surface.

Bus factor: Tier 4: Elevated Concentration. Review recommended to assess key-person dependency.

The table below rates risk across 13 dimensions, from Clean (no findings) to Critical (potential deal impact). Together they form the technical risk profile of the target asset.

CATEGORY	RATING	SUMMARY
Secrets & Credentials	MEDIUM RISK	6 findings detected in scanned files
Dependency Vulnerabilities	MEDIUM RISK	1 vulnerability detected in scanned dependencies
Supply Chain Risk	CLEAN	No known supply chain incidents
Licence & IP	MEDIUM RISK	Project copyleft posture — no third-party contamination detected
Developer Concentration	MEDIUM RISK	Tier 4: Elevated Concentration
Code Quality	CLEAN	Grade B (70.0/100)
Architecture	MEDIUM RISK	11 circular dependencies (0% of modules)
Malware / Destructive Code	LOW RISK	3 heuristic patterns (none high confidence) — manual review recommended
Test Coverage	LOW RISK	Moderate test coverage (21% test ratio, CI gates active) — below comprehensive threshold
Infrastructure & Deployment	LOW RISK	Deployment partially codified — manual steps required

CATEGORY	RATING	SUMMARY
Technical Debt	HIGH RISK	Significant tech debt (868 markers, 2.3/KLOC)
Governance & CI/CD	LOW RISK	Adequate governance (Grade B, 6.2/10)
Engineering Maturity	LOW RISK	Adequate maturity (34.8 releases/yr, 60% community health)

2. Transaction Impact Assessment

This section translates technical findings into their commercial implications for the transaction. Ratings range from Clean (no concern) to Critical (potential deal-breaker), with specific conditions that may need to be met before or after completion.

CATEGORY	ASSESSMENT	DETAIL
Security Exposure	MEDIUM RISK	6 findings detected in scanned files; 1 vulnerability detected in scanned dependencies; 3 heuristic patterns (none high confidence) — manual review recommended
Operational Risk	MEDIUM RISK	Tier 4: Elevated Concentration; Grade B (70.0/100)
IP & Licence Risk	MEDIUM RISK	Project copyleft posture — no third-party contamination detected
Integration Complexity	MEDIUM RISK	11 circular dependencies (0% of modules)
Maintenance Burden	HIGH RISK	Grade B, 3 quality issues, 868 tech debt markers (high)

Remediation Effort Estimate

1 critical area requiring immediate action; 1 high-severity area for short-term remediation; 12 medium-severity items for the integration roadmap.

Estimates assume a senior developer familiar with the technology stack. Actual effort may vary based on codebase familiarity and organisational context.

3. Scope & Methodology

Repository: https://github.com/bitwarden/server

Analysis date: 03 April 2026

Codebase size: 5,761 files, 1,580,770 lines

LANGUAGE	LINES
C#	1,457,005
SQL	91,746
JSON	14,479
Markdown	5,958
Rust	3,549

Methodology

This report was produced by Polaris Intelligence automated analysis pipeline. The following scanners were applied:

1. **GitHub Enrichment** — project metadata, release cadence, community health
2. **Secret Scanner** — regex pattern matching + context classification
3. **Dependency Scanner** — manifest parsing + OSV vulnerability cross-reference + exploitability analysis
4. **Supply Chain Intelligence** — cross-reference against known supply chain incidents
5. **Licence Auditor** — declared licence + source header contradiction detection
6. **Bus Factor Analysis** — 5-tier developer concentration taxonomy (24-month window)
7. **Code Quality Scorer** — cyclomatic complexity, duplication, security anti-patterns
8. **Architecture Mapper** — import graph, circular dependencies, module coupling
9. **Engineering Maturity** — release discipline, community governance, project signals
10. **Malware Heuristic** — destructive actions, crypto mining, exfiltration, obfuscation
11. **Governance & CI/CD** — OpenSSF Scorecard, branch protection, dependency management

Note: *File counts may vary between sections because each scanner operates on a different subset of files (e.g. quality analysis covers source code files only, while the scope total includes configuration, documentation, and data files).*

This is an automated analysis and does not constitute legal, security, or investment advice. Findings should be verified by qualified professionals.

4. Secrets & Credentials MEDIUM RISK

Hardcoded credentials — API keys, database passwords, tokens — are the most common cause of data breaches. Their presence indicates both an immediate security exposure and a gap in the target's engineering practices that transfers with the acquisition.

6 patterns matching credential signatures detected. No high-confidence findings — these may include test fixtures, example values, or pattern-based detections requiring contextual review:

SEVERITY	CONFIDENCE	CATEGORY	LOCATION	MATCH (REDACTED)
MEDIUM RISK	MEDIUM	Password Assignment	`.devcontainer/i..._dev/postCreateCommand.sh:56` [F1]	`DB_PASSWORD="$(g*****oP 'MSSQL_SA_PASSWORD=["'"'"']?\K[^"'"'"'\s]+' $REPO_ROOT/dev/.env)"`
MEDIUM RISK	MEDIUM	Password Assignment	`.devcontainer/i..._dev/postCreateCommand.sh:86` [F1]	`cert_password="$(g*******on "DEV_CERT_PASSWORD" "Paste the \"Licensing Certificate - Dev\" password: " "" 1)"`
MEDIUM RISK	MEDIUM	Password Assignment	`.devcontainer/c..._dev/postCreateCommand.sh:56` [F2]	`DB_PASSWORD="$(g*****oP 'MSSQL_SA_PASSWORD=["'"'"']?\K[^"'"'"'\s]+' $DEV_DIR/.env)"`
MEDIUM RISK	MEDIUM	Password Assignment	`src/Core/Auth/Identity/Claims.cs:39`	`public const string ManageResetPassword = "man*************ord";`
MEDIUM RISK	MEDIUM	Generic API Key	`test/Api.Integr...rganizationTestHelpers.cs:219` [F3]	`ApiKey = "CfG**********************nmU",`
MEDIUM RISK	MEDIUM	Password Assignment	`util/Seeder/Factories/UserSeeder.cs:12`	`internal const string DefaultPassword = "asd******sdf";`

5. Dependency Vulnerabilities MEDIUM RISK

Modern software relies on hundreds of third-party packages. Known vulnerabilities in these dependencies are publicly catalogued and actively exploited. Unpatched critical CVEs represent a quantifiable security liability that transfers to the acquirer.

115 dependencies analysed across 30 manifests (105 runtime, 10 dev/test).

1 dependency vulnerability found:

SEVERITY		COUNT
MEDIUM	MEDIUM RISK	1

SEVERITY	PACKAGE	VERSION	CVE/ ID	SUMMARY	FIX	EXPOSURE
MEDIUM (5.0)	mjml	4.15.3	CVE-2025-67898	MJML allows mj-include directory traversal due to an incomplete fix for CVE-2020-12827		DIRECT IMPORT

First-party packages checked: NuGet:MicroBenchmarks, NuGet:Notifications, NuGet:Billing. No known CVEs affecting current version.

Dependency Health

"No known CVEs" does not mean healthy dependencies. Stale or abandoned packages receive no security patches and represent latent risk. Health status is derived from package registry release dates.

STATUS		COUNT
Active (released within 6 months)	CLEAN	14
Stable (released within 1 year)	CLEAN	2
Stale (1–2 years since last release)	MEDIUM RISK	2
Abandoned (2+ years since last release)	HIGH RISK	2
Unknown (registry lookup failed)	LOW RISK	95

Dependency health score: **5.6/10**

At-Risk Dependencies

PACKAGE	VERSION	STATUS	LAST RELEASE
font-awesome	4.7.0	HIGH RISK abandoned	2016-10-24 (9.4yr ago)
toastr	2.1.4	HIGH RISK abandoned	2017-12-08 (8.3yr ago)
base64		MEDIUM RISK stale	2024-04-30 (1.9yr ago)
expose-loader	5.0.1	MEDIUM RISK stale	2025-02-07 (1.1yr ago)

Scanner notes:

- No Cargo.lock found — Cargo dependency versions are minimum constraints; CVE matching skipped

6. Developer Concentration (Bus Factor) MEDIUM RISK

"Bus factor" measures how many developers would need to leave before critical knowledge is lost. High concentration in one developer creates key-person dependency — a material operational risk that can delay integration and increase post-acquisition costs.

Tier 4: Elevated Concentration — Bus factor score: **42.8%**

42.8% developer concentration across 27 active contributors is above the recommended threshold. While not critical, proactive knowledge sharing is advised to reduce key-person risk.

Top Contributors

DEVELOPER	COMMITS	OWNED FILES	CORE %	STATUS
Matt Bishop	91	65	2%	Active
Jared McCannon	77	195	5%	Active
Jimmy Vo	61	83	2%	Active
Brandon Treston	54	51	1%	Active
Github Actions	54	1	0%	Active
Jonas Hendrickx	54	19	1%	Departed
Mick Letofsky	45	201	4%	Active
Bernd Schoolmann	39	73	2%	Active
Patrick-Pimentel-Bitwarden	34	59	1%	Active
Vijay Oommen	32	81	2%	Active

Departed Developer Risk

DEVELOPER	MONTHS INACTIVE	CORE FILES OWNED	RISK
Jonas Hendrickx	11	17	HIGH RISK
Tom	6	15	HIGH RISK

DEVELOPER	MONTHS INACTIVE	CORE FILES OWNED	RISK
Ben Bryant	10	13	HIGH RISK
Mark Youssef	18	4	MEDIUM RISK
Maksym Sorokin	7	1	MEDIUM RISK
tangowithfoxtrot	8	1	MEDIUM RISK
Henrik	10	1	MEDIUM RISK
bitwarden-charlie	10	1	MEDIUM RISK
Patrick Honkonen	9	0	LOW RISK
Opeyemi	10	0	LOW RISK

7. Licence & Intellectual Property MEDIUM RISK

Copyleft licences (GPL, AGPL) require derivative works to be released under the same open-source terms. If copyleft code is embedded in a proprietary product, the acquirer may face an obligation to open-source their own code — or costly remediation to replace the affected components.

Declared Licences

SPDX ID	RISK	SOURCE	FILE
AGPL-3.0	CRITICAL RISK	licence_file	`LICENSE.txt`

Project Licensing Posture

The following reflects the project's own chosen licence. This is not contamination — it describes the terms under which this software is distributed.

LICENCE	POSTURE	INVESTOR IMPLICATIONS
AGPL-3.0	Project is licensed under AGPL-3.0 — Network copyleft — all linked/derivative code must be open-sourced under same…	This project is distributed under AGPL-3.0 (network copyleft). Acquirers must comply with copyleft terms or negotiate a commercial licence from the…

No third-party licence contamination detected.

8. Code Quality & Technical Debt CLEAN

Code quality directly predicts the cost and speed of post-acquisition development. A low grade signals elevated technical debt — higher bug rates, slower feature delivery, and more expensive onboarding for new developers joining after the transaction.

Quality grade: **B (70.0/100)** — Acceptable

Grade Scale

GRADE	MEANING
A	Excellent maintainability
B	Good engineering quality
C	Moderate technical debt
D	High technical debt
F	Severe structural risk

METRIC	VALUE
Total files	3125
Code lines	178,564
Functions	9178
Classes	3522
Avg function length	16.2 lines
Avg complexity	2.1

Technical Debt Indicators

The following indicators are informational. At this grade level, they represent normal characteristics of a healthy codebase rather than actionable concerns. Duplication above automated thresholds is common in test files and generated code.

- 20 files exceed 500 lines
- 20 functions with high cyclomatic complexity
- Code duplication at 12.1% (threshold: 5.0%)

Complexity Hotspots

SEVERITY	FILE	FUNCTION	COMPLEXITY
HIGH RISK	`src/Core/Billin...ls/OrganizationLicense.cs` [F4]	GetDataBytes	52
HIGH RISK	`src/Notifications/HubHelpers.cs`	Task	51
HIGH RISK	`src/Core/Organi...rganizationPlanCommand.cs` [F5]	UpgradePlanAsync	50
HIGH RISK	`src/Core/Billin...ls/OrganizationLicense.cs` [F4]	ObsoleteVerifyData	42
HIGH RISK	`src/Core/Auth/M...sterFinishRequestModel.cs` [F6]	Validate	37
HIGH RISK	`src/Core/Billin...ls/OrganizationLicense.cs` [F4]	VerifyData	35
HIGH RISK	`src/Events/Controllers/CollectController.cs`	Post	34
HIGH RISK	`src/Core/Platfo.../HandlebarsMailService.cs` [F7]	Task	31
HIGH RISK	`src/Core/AdminC...ns/OrganizationService.cs` [F8]	SaveUsersSendInvitesAsync	30
HIGH RISK	`src/Core/AdminC...tedOrganizationDetails.cs` [F9]	CanUseLicense	29

Large Files (>500 lines)

FILE	TOTAL LINES	CODE LINES
`util/PostgresMi...seContextModelSnapshot.cs` [F10]	3596	2554
`util/MySqlMigra...seContextModelSnapshot.cs` [F11]	3590	2550
`util/SqliteMigr...seContextModelSnapshot.cs` [F12]	3579	2543
`util/RustSdk/rust/src/rsa_keys.rs`	2922	2814
`util/SqlServerE...seContextModelSnapshot.cs` [F13]	1730	1215

FILE	TOTAL LINES	CODE LINES
`src/Api/Vault/C...lers/CiphersController.cs` [F14]	1714	1349
`src/Core/Platfo.../HandlebarsMailService.cs` [F7]	1647	1478
`src/Core/AdminC...ns/OrganizationService.cs` [F8]	1238	1029
`src/Core/Servic...mentations/UserService.cs` [F15]	1190	998
`src/Core/Vault/...ntations/CipherService.cs` [F16]	1145	921

No security anti-patterns detected.

9. Architectural Assessment MEDIUM RISK

3469 production modules, 7897 internal dependencies, 285 external imports.

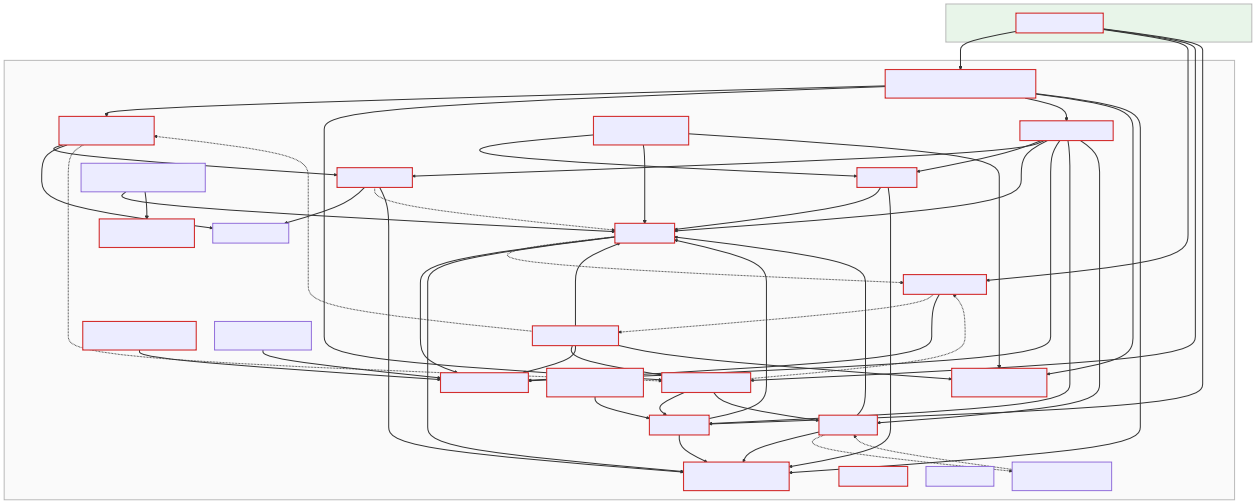
11 circular dependencies detected. Circular imports increase coupling and make refactoring more expensive. Breaking these cycles should be prioritised to reduce integration risk.

Change impact zones: 20 modules have high fan-in (many dependents). Top modules: `src/Infrastructure.EntityFramework/Dirt/Repositories/EventRepository.cs`, `src/Core/Dirt/Entities/Event.cs`, `src/Core/Enums/TransactionType.cs`. Changes to these modules carry elevated regression risk.

30 isolated modules detected (no internal imports or dependents). These are typically standalone utilities, examples, platform-specific implementations, or generated code.

Module Dependency Map

Modules are grouped by architectural layer and coloured accordingly. Arrows show import dependencies. Numbers in parentheses indicate how many other modules depend on each file. Thick red borders highlight high fan-in modules (change risk).



Layer Distribution

Modules are grouped by their role in the application: business logic (core functionality), presentation (user-facing), data (storage), and test. A well-structured codebase separates these concerns clearly.

LAYER	MODULES
business_logic	1490
data	1342
other	303
infrastructure	175
presentation	159

Entry Points (28)

LOCATION	PATTERN
`src/Notifications/Startup.cs:92`	`app.UseForwardedHeaders(`
`src/Notifications/Program.cs:7`	`static void Main(`
`src/Billing/Startup.cs:133`	`app.UseDeveloperExceptionPage(`
`src/Billing/Program.cs:7`	`static void Main(`
`src/SharedWeb/U...ceCollectionExtensions.cs:615` [F17]	`app.UseForwardedHeaders(`
`src/Api/Startup.cs:244`	`app.UseDefaultMiddleware(`
`src/Api/Program.cs:7`	`static void Main(`

LOCATION	PATTERN
`src/Admin/Startup.cs:143`	`app.UsePathBase(`
`src/Admin/Program.cs:7`	`static void Main(`
`src/Identity/Startup.cs:182`	`app.Use(`

Circular Dependencies (11)

`src/Core/Dirt/Models/Data/EventMessage.cs → src/Core/Context/ICurrentContext.cs → src/Core/AdminConsole/Context/CurrentContextProvider.cs → src/Core/Dirt/Models/Data/EventMessage.cs`

`src/Core/Dirt/Models/Data/EventMessage.cs → src/Core/Context/ICurrentContext.cs → src/Core/AdminConsole/Context/CurrentContextProvider.cs → src/Core/AdminConsole/Entities/Provider/ProviderUser.cs → src/Core/Dirt/Entities/Event.cs → src/Core/Dirt/Models/Data/EventMessage.cs`

`src/Core/Vault/Entities/Cipher.cs → src/Core/Vault/Models/Data/AttachmentResponseData.cs → src/Core/Vault/Entities/Cipher.cs`

`src/Core/Dirt/Models/Data/EventMessage.cs → src/Core/Context/ICurrentContext.cs → src/Core/AdminConsole/Repositories/IProviderUserRepository.cs → src/Core/Dirt/Repositories/IEventRepository.cs → src/Core/Dirt/Models/Data/EventMessage.cs`

`src/Core/Dirt/EventIntegrations/EventIntegrationsServiceCollectionExtensions.cs → src/Core/Dirt/EventIntegrations/OrganizationIntegrations/UpdateOrganizationIntegrationCommand.cs → src/Core/Dirt/EventIntegrations/EventIntegrationsServiceCollectionExtensions.cs`

6 additional cycles not shown.

High Fan-In Modules (Change Risk)

"Fan-in" measures how many other parts of the codebase depend on a given file. High fan-in files are widely relied upon — changes to them carry elevated risk because they can affect many dependent components.

MODULE	DEPENDENTS	RISK
`src/Infrastruct...tories/EventRepository.cs` [F18]	462	HIGH RISK
`src/Core/Dirt/Entities/Event.cs`	454	HIGH RISK

MODULE	DEPENDENTS	RISK
`src/Core/Enums/TransactionType.cs`	426	HIGH RISK
`src/Core/Utilities/EnumMemberJsonConverter.cs`	385	HIGH RISK
`src/Core/Dirt/R...ories/IEventRepository.cs` [F19]	327	HIGH RISK
`src/Core/Exceptions/BadRequestException.cs`	255	HIGH RISK
`src/Core/Settings/IBaseServiceUriSettings.cs`	252	HIGH RISK
`src/Core/AdminConsole/Entities/Policy.cs`	238	HIGH RISK
`src/Core/Dirt/S...tions/NoopEventService.cs` [F20]	233	HIGH RISK
`src/Core/Context/ICurrentContext.cs`	151	HIGH RISK

Isolated Modules (30)

These files neither use nor are used by any other file in the codebase. They may be unused code, standalone utilities, or components loaded indirectly. Each should be reviewed to confirm it is genuinely needed.

- `bitwarden_license/src/Scim/Models/BaseScimModel.cs`
- `bitwarden_license/src/Scim/Models/GetGroupsQueryParamModel.cs`
- `bitwarden_license/src/Scim/Models/GetUsersQueryParamModel.cs`
- `bitwarden_license/src/Scim/Models/ScimMetaModel.cs`
- `bitwarden_license/src/Scim/Models/ScimPatchModel.cs`
- `bitwarden_license/src/Scim/ScimSettings.cs`
- `bitwarden_license/src/Scim/Utilities/ScimConstants.cs`
- `bitwarden_license/src/Sso/Models/ErrorViewModel.cs`
- `bitwarden_license/src/Sso/Models/RedirectViewModel.cs`
- `bitwarden_license/src/Sso/Models/SamlEnvironment.cs`
- `bitwarden_license/src/Sso/Utilities/ClaimsExtensions.cs`
- `bitwarden_license/src/Sso/Utilities/DynamicAuthenticationScheme.cs`
- `bitwarden_license/src/Sso/Utilities/ExtendedOptionsMonitorCache.cs`
- `bitwarden_license/src/Sso/Utilities/IDynamicAuthenticationScheme.cs`
- `bitwarden_license/src/Sso/Utilities/IExtendedOptionsMonitorCache.cs`

9b. Architecture Topology

This section applies graph-theoretic analysis to the module dependency network identified in the Architectural Assessment. Community detection reveals natural subsystem boundaries; centrality analysis identifies critical bridge modules whose failure or refactoring would disproportionately affect the codebase.

METRIC	VALUE
Modules analysed	3469
Internal dependencies	7897
Subsystems detected	15
Modularity score	0.488 (moderate modularity)
Unclustered modules	1008

Subsystem Overview

Communities are groups of modules that are more tightly connected to each other than to the rest of the codebase. Each represents a natural subsystem boundary.

SUBSYSTEM	MODULES
src/ (business_logic, 372 modules)	372
src/ (data, 196 modules)	196
util/ (data, 402 modules)	402
src/ (business_logic, 178 modules)	178
src/Core/ (business_logic, 340 modules)	340
src/Api/ (presentation, 47 modules)	47
src/Infrastructure.EntityFramework/ (data, 142 modules)	142
src/Core/ (business_logic, 147 modules)	147
perf/load/ (other, 5 modules)	5
src/ (infrastructure, 168 modules)	168
src/Core/ (business_logic, 361 modules)	361
src/Icons/ (data, 15 modules)	15
src/ (business_logic, 45 modules)	45

SUBSYSTEM	MODULES
util/RustSdk/ (infrastructure, 3 modules)	3
src/Core/ (business_logic, 40 modules)	40

God Modules (Excessive Coupling)

"God modules" have direct dependencies spanning many subsystems. They violate separation of concerns and make the codebase harder to modify safely — changes risk unintended side effects across multiple subsystems.

MODULE	SUBSYSTEMS	CROSS-EDGES	TOTAL DEGREE	RISK
`src/Core/Dirt/Entities/Event.cs`	11	291	457	HIGH RISK
`src/Core/Enums/TransactionType.cs`	11	280	426	HIGH RISK
`src/Core/Utilities/EnumMemberJsonConverter.cs`	11	229	385	HIGH RISK
`src/Core/Dirt/R...ories/IEventRepository.cs` [F19]	11	224	330	HIGH RISK
`src/Core/Exceptions/BadRequestException.cs`	10	212	255	HIGH RISK
`src/Core/Settings/IBaseServiceUriSettings.cs`	11	179	252	HIGH RISK
`src/Core/Dirt/S...tions/NoopEventService.cs` [F20]	10	168	240	HIGH RISK
`src/Core/Context/ICurrentContext.cs`	8	122	158	HIGH RISK
`src/Core/AdminConsole/Entities/Policy.cs`	7	120	241	HIGH RISK
`src/Core/Dirt/E...ceCollectionExtensions.cs` [F21]	10	96	144	HIGH RISK

Package Isolation

Independent packages with zero cross-dependencies. In a monorepo / workspace architecture, this indicates clean separation between packages — a positive architectural signal.

PACKAGE A	PACKAGE B	STATUS
src/ (business_logic, 372 modules)	perf/load/ (other, 5 modules)	Independent
src/ (business_logic, 372 modules)	util/RustSdk/ (infrastructure, 3 modules)	Independent
src/ (data, 196 modules)	perf/load/ (other, 5 modules)	Independent
src/ (data, 196 modules)	src/Icons/ (data, 15 modules)	Independent
src/ (data, 196 modules)	util/RustSdk/ (infrastructure, 3 modules)	Independent
util/ (data, 402 modules)	src/Api/ (presentation, 47 modules)	Independent
util/ (data, 402 modules)	perf/load/ (other, 5 modules)	Independent
util/ (data, 402 modules)	src/Icons/ (data, 15 modules)	Independent
util/ (data, 402 modules)	util/RustSdk/ (infrastructure, 3 modules)	Independent
src/ (business_logic, 178 modules)	perf/load/ (other, 5 modules)	Independent

Structural Gaps

Subsystem pairs with very few connecting dependencies that may indicate missing integration.

SUBSYSTEM A	SUBSYSTEM B	CONNECTING EDGES
src/ (business_logic, 372 modules)	src/Icons/ (data, 15 modules)	1
util/ (data, 402 modules)	src/ (business_logic, 45 modules)	1

Topology Findings

HIGH RISK **God Module:** Module 'src/Core/Dirt/Entities/Event.cs' connects 11 subsystems with 291 cross-boundary edges. Changes here risk cascading across the codebase.

HIGH RISK **God Module:** Module 'src/Core/Enums/TransactionType.cs' connects 11 subsystems with 280 cross-boundary edges. Changes here risk cascading across the codebase.

HIGH RISK **God Module:** Module 'src/Core/Utilities/EnumMemberJsonConverter.cs' connects 11 subsystems with 229 cross-boundary edges. Changes here risk cascading across the codebase.

HIGH RISK **God Module:** Module 'src/Core/Dirt/Repositories/IEventRepository.cs' connects 11 subsystems with 224 cross-boundary edges. Changes here risk cascading across the codebase.

HIGH RISK **God Module:** Module 'src/Core/Exceptions/BadRequestException.cs' connects 10 subsystems with 212 cross-boundary edges. Changes here risk cascading across the codebase.

HIGH RISK **God Module:** Module 'src/Core/Settings/IBaseServiceUriSettings.cs' connects 11 subsystems with 179 cross-boundary edges. Changes here risk cascading across the codebase.

HIGH RISK **God Module:** Module 'src/Core/Dirt/Services/NoopImplementations/NoopEventService.cs' connects 10 subsystems with 168 cross-boundary edges. Changes here risk cascading across the codebase.

HIGH RISK **God Module:** Module 'src/Core/Context/ICurrentContext.cs' connects 8 subsystems with 122 cross-boundary edges. Changes here risk cascading across the codebase.

HIGH RISK **God Module:** Module 'src/Core/AdminConsole/Entities/Policy.cs' connects 7 subsystems with 120 cross-boundary edges. Changes here risk cascading across the codebase.

HIGH RISK **God Module:** Module 'src/Core/Dirt/EventIntegrations/EventIntegrationsServiceCollectionExtensions.cs' connects 10 subsystems with 96 cross-boundary edges. Changes here risk cascading across the codebase.

LOW RISK **Clean Package Isolation:** 39 independent package pairs with zero cross-dependencies — clean monorepo isolation.

10. Test Coverage LOW RISK

Automated tests act as safety nets for the codebase. When developers make changes, tests verify nothing else has broken. Strong test coverage means the acquirer's team can modify and extend the code with confidence; weak or absent testing means changes carry a higher risk of introducing undetected problems.

Some testing in place, but gaps remain — coverage is partial or CI enforcement is missing, limiting the safety net for code changes.

METRIC	VALUE
Test files	938
Source files	4405
Test-to-source ratio	21.3%
Test functions / cases	4082

This is a moderate ratio. Some areas of the codebase likely lack test coverage, which increases the cost of making changes safely.

The project uses established testing tools (cargo test, xUnit), indicating the team has invested in testing infrastructure.

CI configuration detected *(.github/workflows/respond.yml, .github/workflows/stale-bot.yml, .github/workflows/test-database.yml, .github/workflows/enforce-labels.yml, .github/workflows/scan.yml, .github/workflows/publish.yml, .github/workflows/release.yml, .github/workflows/repository-management.yml, .github/workflows/test.yml, .github/workflows/load-test.yml, .github/workflows/code-references.yml, .github/workflows/ephemeral-environment.yml, .github/workflows/protect-files.yml, .github/workflows/review-code.yml, .github/workflows/build_target.yml, .github/workflows/_move_edd_db_scripts.yml, .github/workflows/build.yml, .github/workflows/*

cleanup-rc-branch.yml, .github/workflows/automatic-issue-responses.yml), but test enforcement could not be fully confirmed. The acquirer should verify that tests run automatically on all pull requests.

11. Infrastructure & Deployment LOW RISK

This section assesses whether the deployment process — how the software is built, packaged, and released — is documented in code or relies on undocumented manual steps. Codified deployment means a new team can operate the software independently; undocumented deployment creates dependency on the original developers and increases transition risk.

Deployment is partially codified — some automation exists but may require manual steps or undocumented knowledge to fully operate.

Deployment is partially codified: CI/CD is present alongside containerisation. A new team would have a reasonable starting point but may need additional context for production deployment.

Infrastructure Found

CATEGORY	TOOLS / CONFIGURATION
Containerisation	dockerignore (7), docker-compose (2), dockerfile (16)
CI/CD Automation	github-actions (19)

Gaps Identified

- No orchestration (e.g., Kubernetes, Docker Compose) — multi-service deployment is not automated
- No infrastructure-as-code (e.g., Terraform, CloudFormation) — server provisioning may require manual configuration
- No deployment scripts — the release process may rely on undocumented manual steps

12. Technical Debt HIGH RISK

Technical debt represents shortcuts, deferred work, and known problems that the development team has acknowledged but not yet fixed. Every codebase carries some debt; what matters is the volume and severity. High technical debt increases the cost of post-acquisition development and the risk that changes introduce new problems.

Significant technical debt — the codebase shows elevated levels of deferred work, suppressed warnings, or acknowledged problems. Budget for cleanup.

Developers have left **868** notes in the code flagging work that needs to be done (TODO items, known bugs, temporary workarounds). That is **2.3 markers per 1,000 lines of code**. This is above average and indicates accumulated deferred work.

Debt Markers by Type

TYPE	COUNT
FIXME	655
TODO	190
TEMP	15
WORKAROUND	5
HACK	3

190 markers are planned work items (TODO) while 663 flag known problems (FIXME, HACK, BUG). A high proportion of FIXME/HACK markers is more concerning than TODOs, as they indicate acknowledged broken or fragile code.

Suppressed Warnings

19 instances where automated code checks were deliberately silenced (0.1 per 1,000 lines). This can indicate either legitimate exceptions to coding rules or developers hiding problems from automated checks. Each suppression warrants review during integration.

173 uses of deprecated APIs or functions detected. Deprecated code will eventually stop working when dependencies are updated, requiring remediation.

Error Handling Concerns

- 9 places where errors are silently ignored rather than handled

Silent error handling means problems occur but produce no visible symptoms — making bugs harder to find and diagnose. This is a common source of difficult-to-reproduce issues.

Files with Most Debt Markers

FILE	MARKERS	TYPES
`test/Infrastruc...ionUserRepositoryTests.cs` [F22]	16	todo
`src/Api/Vault/C...lers/CiphersController.cs` [F14]	14	fixme, todo
`src/Core/Billing/Models/StaticStore/Plan.cs`	13	fixme, todo
`bitwarden_licen...lers/AccountController.cs` [F23]	11	fixme, hack, todo, workaround
`src/Core/Billin...s/StripePaymentService.cs` [F24]	10	fixme, hack, todo, workaround
`src/Infrastruct...ories/CipherRepository.cs` [F25]	7	fixme, temp

FILE		MARKERS	TYPES
`test/Infrastruc.../CipherRepositoryTests.cs`	[F26]	7	todo
`src/Api/Billing...ationBillingController.cs`	[F27]	6	todo
`test/Infrastruc...ls/SendRepositoryTests.cs`	[F28]	6	todo
`src/Api/Billing...ers/AccountsController.cs`	[F29]	5	todo

13. Governance & CI/CD Security LOW RISK

Governance measures the security and maturity of development processes — CI pipeline hardening, release signing, code review enforcement, and dependency management. Weak governance increases post-acquisition remediation costs and ongoing operational risk.

Overall governance: **B** (6.2/10) [OpenSSF Scorecard + local analysis]

OpenSSF Scorecard aggregate: 6.4/10

Individual check scores below reflect Scorecard's automated assessment. Low scores on checks such as branch protection, signed releases, or code review are common for open-source projects and do not necessarily indicate a governance risk — the overall rating accounts for the project's classification and context.

CI Pipeline Security — Grade C

CHECK	SCORE	SOURCE	DETAIL
Token-Permissions	0/10	scorecard	detected GitHub workflow tokens with excessive permissions
Pinned-Dependencies	2/10	scorecard	dependency not pinned by hash detected -- score normalized to 2
Dangerous-Workflow	10/10	scorecard	no dangerous workflow patterns detected
CI-Tests	10/10	scorecard	30 out of 30 merged PRs checked by a CI test -- score normalized to 10
SAST	10/10	scorecard	SAST tool is run on all commits

CI pipelines have moderate security gaps that should be addressed. Workflow tokens have excessive permissions — apply principle of least privilege.

Release Engineering — Grade B

CHECK	SCORE	SOURCE	DETAIL
Signed-Releases	0/10	scorecard	Project has not signed or included provenance with any releases.
Packaging	10/10	scorecard	packaging workflow detected
Maintained	10/10	scorecard	30 commits and 9 issue activity found in the last 90 days -- score normalized to 10
release_frequency	10/10	enrichment	34.8 releases/year, 100% semver compliant

Release processes are functional but have room for improvement. Releases are not consistently signed — supply chain integrity risk.

Governance Posture — Grade B

CHECK	SCORE	SOURCE	DETAIL
Security-Policy	10/10	scorecard	security policy file detected
Contributors	10/10	scorecard	project has 9 contributing companies or organizations
CII-Best-Practices	0/10	scorecard	no effort to earn an OpenSSF best practices badge detected
License	9/10	scorecard	license file detected

Governance posture is adequate with some gaps.

Branch Protection & Code Review — Grade B

CHECK	SCORE	SOURCE	DETAIL
Branch-Protection	5/10	scorecard	branch protection is not maximal on development and all release branches
Code-Review	10/10	scorecard	all changesets reviewed

Branch protection is present but not fully enforced.

Dependency Management — Grade C

CHECK	SCORE	SOURCE	DETAIL
Dependency-Update-Tool	10/10	scorecard	update tool detected
Vulnerabilities	0/10	scorecard	Scorecard reports 18 via OSV; Polaris found 1 after filtering to versions in the manifest.
Fuzzing	0/10	scorecard	project is not fuzzed

Dependency management has gaps — automated updates or vulnerability tracking missing.

14. Engineering Maturity LOW RISK

Engineering maturity measures the project's operational health beyond source code quality — release discipline, community governance, and project signals that indicate long-term viability.

Overall maturity: **B** — minor maturity gaps only (risk rating: LOW)

Release Cadence

METRIC	VALUE
Releases per year	34.8
Days since last release	2
Semver compliance	100.0%
Grade	A

Community Health

DOCUMENT	STATUS
SECURITY.md	Present
CONTRIBUTING.md	Present
CODE_OF_CONDUCT	Missing
Issue template	Missing

DOCUMENT	STATUS
PR template	Present
Health score	60%
Grade	C

Project Signals

METRIC	VALUE
Stars	18,353
Forks	1,559
Open issues	198
Contributors	100
Repository created on GitHub	2015-11-23
Grade	A

Note: *GitHub API reports 100 contributors while git history analysis (Bus Factor section) identified 30. This discrepancy arises because GitHub counts all commit authors across the full history, while git analysis may use a limited clone depth or different author-deduplication rules.*

15. Malware & Destructive Action Scan LOW RISK

This scan searches for code patterns commonly associated with protestware, supply-chain attacks, and sabotage — filesystem wipers, obfuscated payloads, unauthorised network calls, and install-hook abuse. Findings are heuristic and warrant manual review rather than automatic condemnation.

Files scanned: 4,444

3 heuristic patterns detected — these are common in legitimate code (build scripts, test harnesses, networking libraries) and do not indicate malicious intent without manual verification.

CATEGORY	FINDINGS
Network Exfiltration	3

Pattern type reflects the category of matched heuristic, not assessed risk. All findings below are low or medium confidence.

PATTERN TYPE	CONFIDENCE	LOCATION	PATTERN	CODE
HIGH RISK	MEDIUM	`perf/load/sync.js:61`	HTTP request with dynamic URL (potential data exfiltration)	`const syncRes = http.get(`${API_URL}/sync?excludeDomains=${e`
HIGH RISK	MEDIUM	`perf/load/config.js:58`	HTTP request with dynamic URL (potential data exfiltration)	`const res = http.get(`${API_URL}/config`, params);`
HIGH RISK	MEDIUM	`perf/load/groups.js:98`	HTTP request with dynamic URL (potential data exfiltration)	`const getRes = http.get(`${API_URL}/public/groups/${id}`, pa`

16. Risk Summary & Recommendations

Recommendations are prioritised by their potential impact on the transaction. Immediate and Urgent items should be addressed as conditions precedent; Medium items can be scheduled into the post-acquisition integration roadmap.

PRIORITY	CATEGORY	RECOMMENDED ACTION
HIGH RISK	Licence Risk	Engage legal counsel to assess copyleft exposure and implications for the acquirer's intended use.
MEDIUM RISK	Secrets & Credentials	Review detected patterns and remove any genuine secrets from source code. Consider a secrets management solution for configuration values.
MEDIUM RISK	Technical Debt	Elevated technical debt (868 markers, 19 suppressions). Prioritise resolving TODO/FIXME items and reducing linter suppressions.

Appendix A: Raw Data

Dependency Inventory (115 packages)

ECOSYSTEM	PACKAGE	VERSION	VULNS	HEALTH	LAST RELEASE
NuGet	BenchmarkDotNet	0.15.3	0	unknown	
NuGet	Microsoft.AspNetCore.SignalR.Protocols.MessagePack	8.0.8	0	unknown	
NuGet	Microsoft.AspNetCore.SignalR.StackExchangeRedis	8.0.8	0	unknown	

ECOSYSTEM	PACKAGE	VERSION	VULNS	HEALTH	LAST RELEASE
NuGet	MessagePack	2.5.192	0	unknown	
NuGet	MarkDig	1.1.0	0	unknown	
NuGet	Swashbuckle.AspNetCore	10.1.0	0	unknown	
NuGet	Microsoft.Bot.Builder.Integration.AspNet.Core	4.23.0	0	unknown	
NuGet	Swashbuckle.AspNetCore.SwaggerGen	10.1.0	0	unknown	
NuGet	AspNetCore.HealthChecks.SqlServer	8.0.2	0	unknown	
NuGet	AspNetCore.HealthChecks.Uris	8.0.1	0	unknown	
NuGet	Azure.Messaging.EventGrid	5.0.0	0	unknown	
npm	bootstrap	5.3.6	0	stable	2025-08-26
npm	font-awesome	4.7.0	0	abandoned	2016-10-24
npm	jquery	3.7.1	0	active	2026-01-18
npm	toastr	2.1.4	0	abandoned	2017-12-08
npm	css-loader	7.1.2	0	active	2026-02-16
npm	expose-loader	5.0.1	0	stale	2025-02-07
npm	mini-css-extract-plugin	2.9.2	0	active	2026-03-26
npm	sass	1.97.2	0	active	2026-03-10
npm	sass-loader	16.0.5	0	active	2026-02-05
npm	webpack	5.104.1	0	active	2026-03-03
npm	webpack-cli	6.0.1	0	active	2026-03-17
NuGet	AspNetCoreRateLimit.Redis	2.0.0	0	unknown	
NuGet	AWSSDK.SimpleEmail	4.0.2.5	0	unknown	
NuGet	AWSSDK.SQS	4.0.2.5	0	unknown	
NuGet	Azure.Data.Tables	12.11.0	0	unknown	
NuGet	Azure.Extensions.AspNetCore.DataProtection.Blobs	1.3.4	0	unknown	
NuGet	Microsoft.AspNetCore.DataProtection	8.0.10	0	unknown	
NuGet	Azure.Messaging.ServiceBus	7.20.1	0	unknown	
NuGet	Azure.Storage.Blobs	12.26.0	0	unknown	
NuGet	Azure.Storage.Queues	12.24.0	0	unknown	
NuGet	BitPay.Light	1.0.1907	0	unknown	
NuGet	DuoUniversal	1.3.1	0	unknown	
NuGet	DnsClient	1.8.0	0	unknown	
NuGet	Fido2.AspNet	3.0.1	0	unknown	

ECOSYSTEM	PACKAGE	VERSION	VULNS	HEALTH	LAST RELEASE
NuGet	Handlebars.Net	2.1.6	0	unknown	
NuGet	MailKit	4.15.0	0	unknown	
NuGet	Microsoft.AspNetCore.Authentication.JwtBearer	8.0.10	0	unknown	
NuGet	Microsoft.Azure.Cosmos	3.52.0	0	unknown	
NuGet	Microsoft.Azure.NotificationHubs	4.2.0	0	unknown	
NuGet	Microsoft.Bot.Builder	4.23.0	0	unknown	
NuGet	Microsoft.Bot.Connector	4.23.0	0	unknown	
NuGet	Microsoft.Data.SqlClient	5.2.2	0	unknown	
NuGet	Microsoft.Extensions.Caching.Cosmos	1.8.0	0	unknown	
NuGet	Microsoft.Extensions.Caching.SqlServer	8.0.10	0	unknown	
NuGet	Microsoft.Extensions.Configuration.EnvironmentVariables	8.0.0	0	unknown	
NuGet	Microsoft.Extensions.Configuration.UserSecrets	8.0.0	0	unknown	
NuGet	Microsoft.Extensions.Identity.Stores	8.0.10	0	unknown	
NuGet	OneOf	3.0.271	0	unknown	
NuGet	SendGrid	9.29.3	0	unknown	
NuGet	Serilog.Extensions.Logging.File	3.0.0	0	unknown	
NuGet	Duende.IdentityServer	7.4.6	0	unknown	
NuGet	Newtonsoft.Json	13.0.3	0	unknown	
NuGet	AspNetCoreRateLimit	5.0.0	0	unknown	
NuGet	Braintree	5.36.0	0	unknown	
NuGet	Stripe.net	48.5.0	0	unknown	
NuGet	Otp.NET	1.4.0	0	unknown	
NuGet	YubicoDotNetClient	1.2.0	0	unknown	
NuGet	Microsoft.Extensions.Caching.StackExchangeRedis	8.0.10	0	unknown	
NuGet	LaunchDarkly.ServerSdk	8.11.0	0	unknown	
NuGet	Quartz	3.15.1	0	unknown	
NuGet	Quartz.Extensions.Hosting	3.15.1	0	unknown	
NuGet	Quartz.Extensions.DependencyInjection	3.15.1	0	unknown	
NuGet	RabbitMQ.Client	7.1.2	0	unknown	
NuGet	ZiggyCreatures.FusionCache	2.0.2	0	unknown	

ECOSYSTEM	PACKAGE	VERSION	VULNS	HEALTH	LAST RELEASE
NuGet	ZiggyCreatures.FusionCache.Backplane.StackExchangeRedis	2.0.2	0	unknown	
NuGet	ZiggyCreatures.FusionCache.Serialization.SystemTextJson	2.0.2	0	unknown	
NuGet	System.Text.Json	8.0.5	0	unknown	
NuGet	Microsoft.Extensions.Caching.Memory	8.0.1	0	unknown	
npm	mjml	4.15.3	1	active	2025-12-08
npm	mjml-core	4.15.3	0	active	2025-12-08
npm	nodemon	3.1.10	0	active	2026-02-20
npm	prettier	3.8.1	0	active	2026-01-21
NuGet	AutoMapper.Extensions.Microsoft.DependencyInjection	12.0.1	0	unknown	
NuGet	linq2db	5.4.1	0	unknown	
NuGet	Microsoft.EntityFrameworkCore.Relational	[8.0.8]	0	unknown	
NuGet	Microsoft.EntityFrameworkCore.SqlServer	[8.0.8]	0	unknown	
NuGet	Microsoft.EntityFrameworkCore.Sqlite	[8.0.8]	0	unknown	
NuGet	Npgsql.EntityFrameworkCore.PostgreSQL	[8.0.4]	0	unknown	
NuGet	Pomelo.EntityFrameworkCore.MySql	[8.0.2]	0	unknown	
NuGet	linq2db.EntityFrameworkCore	[8.1.0]	0	unknown	
NuGet	AngleSharp	1.4.0	0	unknown	
NuGet	Dapper	2.1.66	0	unknown	
NuGet	Microsoft.AspNetCore.Http	2.2.2	0	unknown	
NuGet	Sustainsys.Saml2.AspNetCore2	2.11.0	0	unknown	
NuGet	CsvHelper	33.1.0	0	unknown	
NuGet	Microsoft.NET.Test.Sdk	$(MicrosoftNetTestSdkVersion)	0	unknown	
NuGet	xunit	$(XUnitVersion)	0	unknown	
NuGet	xunit.runner.visualstudio	$(XUnitRunnerVisualStudioVersion)	0	unknown	
NuGet	coverlet.collector	$(CoverletCollectorVersion)	0	unknown	

ECOSYSTEM	PACKAGE	VERSION	VULNS	HEALTH	LAST RELEASE
NuGet	NSubstitute	$(NSubstituteVersion)	0	unknown	
NuGet	Microsoft.AspNetCore.Mvc.Testing	8.0.10	0	unknown	
NuGet	AutoFixture.Xunit2	$(AutoFixtureXUnit2Version)	0	unknown	
NuGet	Neovolve.Logging.Xunit	6.3.0	0	unknown	
NuGet	RichardSzalay.MockHttp	7.0.0	0	unknown	
NuGet	Microsoft.Extensions.Diagnostics.Testing	9.3.0	0	unknown	
NuGet	MartinCostello.Logging.XUnit	0.7.0	0	unknown	
NuGet	Rnwood.SmtpServer	3.1.0-ci0868	0	unknown	
NuGet	xunit.v3	3.0.1	0	unknown	
NuGet	AutoFixture.AutoNSubstitute	$(AutoFixtureAutoNSubstituteVersion)	0	unknown	

Module Dependencies (top 30)

MODULE	FAN-IN	LAYER
`src/Infrastruct...tories/EventRepository.cs` [F18]	462	data
`src/Core/Dirt/Entities/Event.cs`	454	business_logic
`src/Core/Enums/TransactionType.cs`	426	business_logic
`src/Core/Utilities/EnumMemberJsonConverter.cs`	385	business_logic
`src/Core/Dirt/R...ories/IEventRepository.cs` [F19]	327	business_logic
`src/Core/Exceptions/BadRequestException.cs`	255	business_logic
`src/Core/Settings/IBaseServiceUriSettings.cs`	252	business_logic
`src/Core/AdminConsole/Entities/Policy.cs`	238	business_logic
`src/Core/Dirt/S...tions/NoopEventService.cs` [F20]	233	business_logic
`src/Core/Context/ICurrentContext.cs`	151	business_logic
`src/Core/Dirt/E...ceCollectionExtensions.cs` [F21]	129	business_logic
`src/Core/Dirt/Models/Data/EventMessage.cs`	117	business_logic
`src/Core/Billing/Enums/PlanType.cs`	115	business_logic
`src/Core/Models...gistrationRequestModel.cs` [F30]	113	business_logic
`src/Core/AdminC...ProviderUserRepository.cs` [F31]	100	business_logic
`src/Core/SecretsManager/Entities/Project.cs`	95	business_logic
`src/Core/Billin...s/NoopLicensingService.cs` [F32]	81	business_logic

MODULE	FAN-IN	LAYER
`src/Core/Vault/Entities/Cipher.cs`	80	business_logic
`src/Core/AdminC.../Provider/ProviderUser.cs` [F33]	66	business_logic
`src/Core/Secret...ecretVersionRepository.cs` [F34]	63	business_logic
`src/Core/AdminC.../Provider/ProviderType.cs` [F35]	59	business_logic
`src/Core/Constants.cs`	58	business_logic
`src/Core/Vault/...AttachmentResponseData.cs` [F36]	57	business_logic
`src/Core/KeyMan...wordAuthenticationData.cs` [F37]	56	business_logic
`src/Core/Platfo...ushRegistrationService.cs` [F38]	53	business_logic
`src/Core/Dirt/E...tegrationConfiguration.cs` [F39]	53	business_logic
`src/Infrastruct...work/Dirt/Models/Event.cs` [F40]	53	data
`src/Core/Auth/Entities/SsoUser.cs`	51	business_logic
`src/Core/Models...SeatSubscriptionUpdate.cs` [F41]	50	business_logic
`src/Core/Billin...ceCollectionExtensions.cs` [F42]	49	business_logic

Appendix B: File Reference

Full file paths for truncated references in the report.

REF	FULL PATH
F1	`.devcontainer/internal_dev/postCreateCommand.sh`
F2	`.devcontainer/community_dev/postCreateCommand.sh`
F3	`test/Api.IntegrationTest/Helpers/OrganizationTestHelpers.cs`
F4	`src/Core/Billing/Organizations/Models/OrganizationLicense.cs`
F5	`src/Core/OrganizationFeatures/OrganizationSubscriptions/UpgradeOrganizationPlanCommand.cs`
F6	`src/Core/Auth/Models/Api/Request/Accounts/RegisterFinishRequestModel.cs`
F7	`src/Core/Platform/Mail/HandlebarsMailService.cs`
F8	`src/Core/AdminConsole/Services/Implementations/OrganizationService.cs`
F9	`src/Core/AdminConsole/Models/Data/Organizations/SelfHostedOrganizationDetails.cs`
F10	`util/PostgresMigrations/Migrations/DatabaseContextModelSnapshot.cs`
F11	`util/MySqlMigrations/Migrations/DatabaseContextModelSnapshot.cs`
F12	`util/SqliteMigrations/Migrations/DatabaseContextModelSnapshot.cs`
F13	`util/SqlServerEFScaffold/Migrations/DatabaseContextModelSnapshot.cs`

REF	FULL PATH
F14	src/Api/Vault/Controllers/CiphersController.cs
F15	src/Core/Services/Implementations/UserService.cs
F16	src/Core/Vault/Services/Implementations/CipherService.cs
F17	src/SharedWeb/Utilities/ServiceCollectionExtensions.cs
F18	src/Infrastructure.EntityFramework/Dirt/Repositories/EventRepository.cs
F19	src/Core/Dirt/Repositories/IEventRepository.cs
F20	src/Core/Dirt/Services/NoopImplementations/NoopEventService.cs
F21	src/Core/Dirt/EventIntegrations/EventIntegrationsServiceCollectionExtensions.cs
F22	test/Infrastructure.IntegrationTest/AdminConsole/Repositories/OrganizationUserRepository/OrganizationUserRepositoryTests.cs
F23	bitwarden_license/src/Sso/Controllers/AccountController.cs
F24	src/Core/Billing/Services/Implementations/StripePaymentService.cs
F25	src/Infrastructure.Dapper/Vault/Repositories/CipherRepository.cs
F26	test/Infrastructure.IntegrationTest/Vault/Repositories/CipherRepositoryTests.cs
F27	src/Api/Billing/Controllers/OrganizationBillingController.cs
F28	test/Infrastructure.IntegrationTest/Tools/SendRepositoryTests.cs
F29	src/Api/Billing/Controllers/AccountsController.cs
F30	src/Core/Models/Api/Request/PushRegistrationRequestModel.cs
F31	src/Core/AdminConsole/Repositories/IProviderUserRepository.cs
F32	src/Core/Billing/Services/NoopImplementations/NoopLicensingService.cs
F33	src/Core/AdminConsole/Entities/Provider/ProviderUser.cs
F34	src/Core/SecretsManager/Repositories/ISecretVersionRepository.cs
F35	src/Core/AdminConsole/Enums/Provider/ProviderType.cs
F36	src/Core/Vault/Models/Data/AttachmentResponseData.cs
F37	src/Core/KeyManagement/Models/Data/MasterPasswordAuthenticationData.cs
F38	src/Core/Platform/PushRegistration/IPushRegistrationService.cs
F39	src/Core/Dirt/Entities/OrganizationIntegrationConfiguration.cs
F40	src/Infrastructure.EntityFramework/Dirt/Models/Event.cs
F41	src/Core/Models/Business/SmSeatSubscriptionUpdate.cs
F42	src/Core/Billing/Pricing/ServiceCollectionExtensions.cs