Security and Data Handling

Effective: November 8, 2026 • **Version:** 2026-11-08 • **Service:** SpendLeak

This document describes the conditions under which SpendLeak is offered and should be read together with the other policies published through the Service. The actual service operator and monitored legal contact are identified on SpendLeak's legal information page.

Contents

1. Purpose, scope and ownership
2. Data sensitivity and classification
3. Authentication architecture
4. Account lifecycle and recovery
5. Authorization and workspace isolation
6. Network and transport protection
7. Hosted database security
8. Secrets and deployment configuration
9. Import and file validation
10. AI processing and prompt-injection risks
11. Rate limiting and abuse controls
12. Logging and monitoring
13. Software dependencies and vulnerability response
14. Change management and deployments
15. Business continuity and backup restoration
16. Incident response and breach assessment
17. Customer responsibilities
18. Retention and secure disposal
19. Independent review and attestations
20. Questions, reporting and updates

1. Purpose, scope and ownership

This Security and Data Handling Overview explains SpendLeak's security approach and division of responsibility with business customers. It describes safeguards visible in the current application design together with controls that the operator must maintain outside the code, such as provider settings, access reviews, backups and incident-response procedures. It is not a certification, penetration-test report, contractual service-level guarantee or independently attested control report.

The operator should assign a responsible owner for security and privacy matters, record changes in technical dependencies and update this overview when implemented controls or architecture materially change. No individual should rely on a statement here as proof that a particular safeguard is deployed without technical evidence and operational verification.

2. Data sensitivity and classification

Invoices, contracts, payment information and supplier records can reveal confidential pricing, trade secrets, personal identifiers, business strategies and possible disputes. Customers should classify information before import, exclude unnecessary sensitive fields and establish who is authorized to view it. The Service is not intended to store passwords, full cardholder data, government identity documents, patient records or other high-risk datasets unless a specifically reviewed use case and suitable controls have been agreed.

The operator should distinguish public demonstration content, personal account information, real customer financial records, security logs and internal technical secrets. Controls appropriate for one classification may be inadequate for another. In particular, public demo information must not be commingled with a real account's business data.

3. Authentication architecture

The current application supports supported OAuth identity providers and an optional email/password flow when corresponding credentials and a database are configured. Email passwords are processed using a salted memory-hard password hashing algorithm before storage; plaintext passwords should not be retained. Authentication sessions are managed by the application framework and require secure deployment configuration, a strong secret and valid redirect origins.

A provider's successful login does not by itself establish a company's authorization to access a specific workspace. The operator must test tenant boundaries, cross-site request controls, credentials throttling and session expiry. Multi-factor authentication, automated recovery and email verification should not be claimed unless they are actually implemented and tested.

4. Account lifecycle and recovery

Account creation should include verification of required legal acknowledgment, reasonable password rules and checks that credentials are not exposed in logs or responses. Current password recovery and email verification capabilities must be accurately described to users; where absent, customers must not assume they exist. Access termination should be considered separately from deletion of workspaces, provider identities, legal logs and backups.

Future recovery processes should use time-limited, single-use tokens, verified ownership, rate limits and user notification. Support personnel should not ask users to send passwords or database strings.

Emergency account-access requests should be documented and handled by authorized operators only.

5. Authorization and workspace isolation

The current cloud workspace storage model associates data with a signed-in owner key and scopes primary queries to that key. This is a basic tenancy boundary but is not equivalent to a mature enterprise permission system with independent role-based access controls, database row-level policies and audited organization membership. The operator should test that one account cannot read, update or delete another account's data through any API path or exported identifier.

Customers should not share one login among employees. Before offering organization-wide collaboration, implement explicit invitations, least-privilege roles, removal of departed personnel and authorization checks in each endpoint, not only the front-end navigation. All access must be validated server-side.

6. Network and transport protection

Cloud hosting platforms may provide HTTPS and related transport controls; the operator must ensure secure domains, correct TLS termination, suitable security headers and no accidental downgrade paths. Service tokens, authorization headers, environment variables and database credentials should never appear in client bundles, query strings, ordinary logs or public repository files.

Traffic to database, identity and AI providers should use authenticated encrypted connections according to each provider's supported configuration. The operator should periodically check certificate settings, third-party endpoints, redirect allowlists and the handling of cookies across preview and production environments.

7. Hosted database security

The application uses a configured managed PostgreSQL service for cloud workspace storage. Database credentials belong in server-only environment variables with access limited to authorized deployment operators. The operator should review database user privileges, network exposure, provider access controls, branch management and the actual backup retention available on the chosen service plan.

A shared production credential with broad administrative privileges increases risk. Database migrations should be tested in preview environments before production use, and recovery procedures should be rehearsed. The product must not claim that backups are automatic, immutable or recoverable within a specified period without provider-specific verification and a successful restore test.

8. Secrets and deployment configuration

Sensitive values such as AUTH_SECRET, DATABASE_URL, provider client secrets and GROQ_API_KEY should be configured in Vercel's protected server-side environment settings, never committed to source control or placed in public `NEXT_PUBLIC_` variables. Access to secrets should be restricted, monitored and revoked when staff roles change. Rotated keys must be redeployed and verified before the old credentials are invalidated.

Preview deployments can carry real risks if they contain production secrets or data. The operator should separate environments where possible, restrict preview access and avoid publishing raw logs or debug endpoints. Configuration errors should fail securely or explain that a protected feature is unavailable rather than silently fall back to a less secure mode.

9. Import and file validation

CSV imports may contain malicious formulas, unexpected encodings, large payloads, malformed quotes, deceptive headers and confidential fields. The operator should enforce reasonable size and parsing limits, validate schemas and normalize values carefully. Exports intended for spreadsheet software should protect against formula injection. File content must be treated as untrusted data and must not be executed as code or privileged instructions.

Financial correctness matters to security: manipulated amounts, currencies or identifiers may lead to false claims or mistaken approvals. Users should review mapping previews and reconcile input against originals. If PDF uploads or document extraction are later introduced, require malware scanning, sandboxing and content-type verification before accepting them.

10. AI processing and prompt-injection risks

The AI assistant sends selected context and user questions to an external model provider when enabled. Workspace text is untrusted; a supplier name, invoice description or imported note might contain instructions intended to override the assistant. The application should frame such content as data and restrict model access to authorized records. AI output may nevertheless contain unsupported statements or misleading conclusions.

Never give the AI assistant unreviewed authority to move money, approve bills, change supplier accounts or send legal threats. Treat generated responses as drafts requiring independent verification. The operator should test prompt injection, context disclosure, cross-tenant access, provider outages and logging of sensitive prompts during every material AI upgrade.

11. Rate limiting and abuse controls

Rate limits help reduce automated account creation, password guessing, abuse of the AI endpoint and excessive requests. The current application includes some request-level throttling, but memory-based serverless limits may not be globally consistent across instances. A production-grade deployment should use shared counters where appropriate and monitor abnormal traffic and resource usage.

Limits should account for legitimate customers with shared office IP addresses and provide understandable error responses. Rate limiting is only one control and does not replace strong authorization, anti-bot defenses, secure input handling or observability.

12. Logging and monitoring

Authentication failures, import errors, data-access anomalies and operational faults may be logged for security and reliability. Logging should minimize personal information, avoid raw credentials and sensitive invoice contents wherever practical, and use restricted access. Operators should monitor failed builds, exception spikes, excessive request rates and suspicious workspace activity with defined response ownership.

A front-end audit timeline is useful but does not automatically prove tamper resistance or comprehensive server-side event coverage. Before offering enterprise audit claims, implement durable audit storage, meaningful attribution, integrity protections and retention consistent with legal obligations.

13. Software dependencies and vulnerability response

The Service depends on a web framework, authentication library, database client and other packages. The operator should maintain a dependency inventory, review advisories, use lockfiles, patch material vulnerabilities, test updates and remove obsolete packages. Source parsing and unit tests are useful but are not substitutes for a full production build, vulnerability assessment or penetration test.

Security issues should be prioritized according to likelihood, affected data and potential harm. Emergency fixes should be tested for regression and rolled out through controlled deployments. Evidence of actual remediation should be retained rather than assuming a package upgrade alone resolves an exploit path.

14. Change management and deployments

Changes affecting authentication, authorization, money calculations or data retention require particular review. Pull-request checks should run type checking, tests and production builds;

preview deployments should exercise sign-up, sign-in, workspace creation, imports, legal acceptance and tenant isolation. The operator should record release versions and maintain a rollback plan.

Source syntax validity alone does not demonstrate that the application builds or operates safely on Vercel. Production promotion should be conditional on a successful build and smoke tests in an environment matching production. Emergency changes should have documented follow-up review.

15. Business continuity and backup restoration

The operator should define recovery objectives appropriate to customers' financial reliance and verify the database provider's actual retention and restore capability. Customer exports can be an additional safeguard but do not replace provider backups. A backup should be considered effective only after a representative restoration test, with attention to access controls and data integrity.

No recovery-time, recovery-point or uptime guarantee is made by this overview. Customers should retain their original accounting records and signed contracts in independently managed systems. Loss of a SpendLeak workspace should not be allowed to destroy the only authoritative copy of an organization's financial documents.

16. Incident response and breach assessment

A credible incident should trigger triage, containment, preservation of evidence, review of access logs, assessment of affected data and a decision about legal notice obligations. The operator should assign incident authority, maintain reliable contact routes and coordinate with affected infrastructure providers. If personal information is involved, the assessment should consider sensitivity, probability of misuse and legal tests for reporting or notification.

Canadian federal law can require notification for breaches creating a real risk of significant harm and documentation of all qualifying breach events. The operator must also examine provincial, international and contractual duties where applicable. An incident plan should be practiced; a public policy alone does not prove operational readiness.

17. Customer responsibilities

Customers must use authorized accounts, secure devices and networks, unique credentials and appropriate approval processes. They must not upload material without lawful authority, use demo records as real evidence, or delegate final payments and supplier claims to an unverified algorithm. Users should promptly report suspicious access, restrict who can export sensitive information and verify recipients before sharing reports.

Businesses with special regulatory requirements should evaluate the Service with their own compliance and security teams before uploading regulated information. Customer procurement

policies, contractual data residency needs and industry-specific retention obligations are not automatically satisfied by a standard account.

18. Retention and secure disposal

Personal data, financial records, authentication logs, backup copies and legal acceptance records have different operational and legal retention needs. The operator should adopt a written schedule, enforce access limitations and verify deletion behaviour across the application and providers. A request to delete an active workspace should not be represented as universal immediate erasure if independent backups, account tables or provider logs may remain.

Secure disposal may require staged deletion, expiry of backups, revocation of encryption keys or documented anonymization depending on the system. Any retention exception for legal hold or fraud investigation should be justified, narrow and reviewed periodically.

19. Independent review and attestations

Before accepting sensitive production financial records at scale, obtain qualified review of authentication, authorization, deployment secrets, data handling, import safety and incident response. A penetration test, a code review or an audit is evidence only to the extent it is actually performed and documented. The operator must not display SOC 2, ISO 27001, PCI DSS or equivalent certification claims without a valid independent basis.

Where customers request assurance, provide an honest description of current controls, outstanding risks and planned mitigations. Security documentation should never be used to conceal a known high-risk gap or mislead a buyer about operational maturity.

20. Questions, reporting and updates

Users can report suspected vulnerabilities, inappropriate access or confidentiality concerns through the verified legal/security contact on the Service. Do not send active passwords, secrets or full sensitive datasets in an initial report. The operator should acknowledge credible reports, verify the reporter's authorization and limit unnecessary disclosure during investigation.

This overview is a statement of intended and implemented practices as of its stated version, not a substitute for the actual system configuration or a negotiated security addendum. Material changes to processing, service providers or controls should be reviewed and reflected in current documentation.

Contact and version control

This document is issued under SpendLeak policy version 2026-11-08. Questions, complaints and authorized requests should be addressed through the legal or privacy contact shown on the deployed Service. The operator must ensure that its public legal identity and monitored contact details are accurate before public use.