

VeriQuery Security Whitepaper

Document Version: 1.1 Last Updated: April 27, 2026 Classification: Public

Executive Summary

VeriQuery is built with a "Privacy First" architecture. We recognise that survey data is inherently sensitive — it may include employee feedback, customer opinions, and personally identifiable information. We employ enterprise-grade technical and administrative controls to ensure the confidentiality, integrity, and availability of our users' information.

This document describes those controls in detail and is intended for security reviewers, enterprise procurement teams, and compliance officers evaluating VeriQuery.

1. Infrastructure & Physical Security

VeriQuery is hosted on **AWS** within SOC 2 Type II and ISO 27001 certified data centres.

- **Physical Security:** We rely on AWS for physical security, including 24/7 guarded access, biometric scanning, and redundant power and cooling systems.
 - **Regional Isolation:** All customer data resides within the US West (us-west-2) region.
 - **Runtime:** The VeriQuery application runs on Node.js. Our dependency tree is monitored for runtime-level vulnerabilities via automated scanning.
-

2. Data Isolation (Multi-Tenancy)

We use a **Strict Atomic Isolation** model.

- **Database-Level Scoping:** Every database query is programmatically constrained to the authenticated User ID at the data-access layer. This prevents Insecure Direct Object Reference (IDOR) vulnerabilities — a user cannot access, modify, or delete data belonging to another tenant, even if a record ID is guessed.
- **Logical Separation:** Although stored in a shared MongoDB Atlas cluster, customer data is logically separated through unique ownership markers on every document (surveys and

responses).

3. Data Encryption

- **In Transit:** All data sent between our servers and your browser is encrypted using TLS 1.2 or higher. HTTPS is enforced across the entire platform.
 - **At Rest:** Survey content and responses are stored in encrypted volumes using AES-256 encryption, managed by MongoDB Atlas.
 - **Passwords:** We never store plaintext passwords. All credentials are hashed using the **Argon2** algorithm — the current industry standard for resistance against GPU-based cracking attacks.
-

4. Access Control & Identity Management

- **Principle of Least Privilege:** Internal access to production systems is limited to authorised personnel on a need-to-know basis.
 - **Credential Management:** All production credentials, API keys, and secrets are managed exclusively through environment variables. Secrets are never committed to source control.
 - **Offboarding:** Access to production systems is revoked immediately upon termination of employment or contractor relationship. The full provisioning, review, and deprovisioning procedure is defined in our [Access Control Policy](#).
 - **Authentication Communications:** All official VeriQuery communications are sent exclusively from **@magicapps.dev** addresses. This provides a clear, verifiable signal against phishing impersonation.
-

5. Application Security & SDLC

- **Code Review:** All code changes are submitted as pull requests. No code is merged to the production branch without peer review.
- **Dependency Scanning:** We use Dependabot to monitor all third-party dependencies for known CVEs. Alerts are triaged and patched promptly.
- **Sensitive Data Handling:** Input validation is enforced at all API boundaries. SQL injection and XSS mitigations are built into our data-access and rendering layers.

- **Security Testing:** We conduct periodic manual security reviews of critical paths (authentication, billing, data access). VeriQuery conducts an independent, third-party application-level penetration test prior to initiating a formal SOC 2 audit and no less than **annually thereafter**. Continuous vulnerability monitoring is provided in the interim by GitHub Dependabot (dependency CVEs) and CodeQL (static analysis on every PR).
-

6. Network Security

- HTTPS is enforced on all endpoints; HTTP requests are redirected.
 - Database access is restricted to the application server network; the MongoDB Atlas cluster is not publicly accessible.
 - Environment variables and secrets are injected at runtime and never exposed in build artefacts or logs.
 - **Rate Limiting:** All API endpoints and public-facing pages enforce server-side rate limiting using a multi-keyed fixed-window strategy. Authentication endpoints apply both per-IP limits and per-email limits — the latter stops distributed brute-force attacks where many different IP addresses target a single account. Authenticated API endpoints use a per-user key so one account cannot starve another. A global per-IP limit at the middleware layer provides a first-pass defence against scraping and denial-of-service attempts. Rate limit events are logged and attributed with the key type, path, and retry window.
 - **Reverse Proxy Trust:** A single Traefik reverse proxy sits in front of the application. Real client IP extraction from `X-Real-IP` headers is gated behind an explicit `TRUSTED_PROXY` environment variable, preventing IP spoofing via header injection.
-

7. Monitoring, Logging & Alerting

Log Collection & Aggregation

All application events — including every rate limit trigger, authentication attempt, and API error — are emitted as structured NDJSON to stdout by the application. Container logs are captured and shipped to **AWS CloudWatch** via the `awslogs` Docker log driver.

- **Log Group:** `veriquery-production` (AWS region: `us-west-2`)
- **Retention Policy:** 12 months (rolling)

- **IAM Access:** Logs are written by IAM user `veriquery-logger`, which holds write-only CloudWatch Logs permissions. This follows the principle of least privilege — the credential cannot read, delete, or modify existing log data.

Log entries are keyed by request ID, user ID (where authenticated), endpoint path, and sanitised payload fields to support forensic tracing without exposing sensitive data.

Alerting

An automated alarm monitors the production log stream for server-side errors:

Alarm	Metric Filter	Threshold	Window	Action
Veriquery_Production_Error	<code>\$.status >= 500</code>	≥ 1 occurrence	5 minutes	SNS → Email notification

This alarm is the primary automated detection channel for Phase 1 of the [Incident Response Plan](#). Server crashes and database connection failures will trigger an immediate email notification to the on-call engineer.

8. Resiliency & Backups

- **Database Durability:** Customer data is stored in a MongoDB Atlas replica set, providing synchronous replication across multiple nodes and protecting against single-node failure.
- **Automated Backups:** VeriQuery currently operates on MongoDB Atlas M0 (shared tier), which does not include automated snapshots or point-in-time recovery (PITR). Automated backup capabilities are a planned upgrade milestone, to be enabled upon migration to a dedicated Atlas tier.
- **Recovery Objective:** Our target Recovery Time Objective (RTO) is 2 hours. The Recovery Point Objective (RPO) is best-effort, bounded by replica-set replication lag (typically seconds); no guaranteed PITR window exists at the current tier.

9. Subprocessorsors

The following third-party subprocessors handle specific infrastructure functions on our behalf:

Subprocessor	Role	Region
--------------	------	--------

AWS	Application hosting, infrastructure	US West
MongoDB Atlas	Database hosting and encrypted data storage	US West
Stripe	Payment processing and subscription management	Global
AWS SES	Transactional email (password resets, notifications)	US West

All subprocessors are evaluated for their own security posture before use.

10. Compliance Status

Framework / Standard	Status	Notes
SOC 2 Type II	In Progress / Roadmap	Controls are aligned; formal audit planned
GDPR	Compliant	Privacy-by-design; 72-hour breach notification
CCPA	Compliant	Data portability (CSV export) available
ISO 27001	Via subprocessors	AWS and MongoDB Atlas hold ISO 27001

11. Incident Response Summary

In the event of a confirmed security incident involving personal data:

1. Affected services are isolated or suspended immediately.
2. All credentials and API keys are rotated.
3. The root cause is identified, patched, and verified.
4. Affected account holders are notified by email within **72 hours** of confirmation.
5. A public notice is posted.
6. A formal post-mortem is documented and new controls are implemented.

The full Incident Response Plan is available to enterprise customers and security reviewers upon request.

12. Vulnerability Disclosure

If you discover a potential security vulnerability in VeriQuery, please report it to:

security@magicapps.dev

We will acknowledge your report within 48 hours and work with you to resolve the issue before any public disclosure.

Attestation of Security Practices

I hereby certify that, to the best of my knowledge, the security controls described in this document are accurately represented and currently in effect for the VeriQuery platform.

Jacob Norr

Jacob Norr

Founder & CEO, VeriQuery.com, MagicApps.dev

Date: April 27, 2026