Forecastle

SECURITY WHITEPAPER

Your financial data, properly defended.

How Forecastle architects, encrypts, isolates, audits and retires the most sensitive numbers in your business — salaries, margins, and forward forecasts.

VERSION
1.0

PUBLISHED
2026-05-12

CONTACT
security@forecastle.app

Executive summary

Forecastle is a multi-tenant SaaS platform for financial planning & analysis (FP&A). Customers connect their general ledger (Xero, QuickBooks Online, and others on the roadmap), and Forecastle builds an in-memory multi-dimensional cube of their actuals, budgets and forecasts. The data we hold is sensitive: salary detail, margin structure, scenario plans, and forward-looking forecasts that are commercially material.

This document describes the controls Forecastle has in place to protect that data. It is intended for security, IT, and procurement teams evaluating Forecastle as a vendor.

> **The six controls that matter most:**
>
> 1. Schema-per-tenant isolation at the database layer (not row-level)
> 2. AES-256 encryption at rest, TLS 1.2+ in transit, with no plaintext fallback
> 3. OAuth-based GL access where the **customer** holds the token and can revoke at any time
> 4. Read-only GL access by default; write-back requires explicit per-tenant consent
> 5. Full audit trail attributing every mutation to a specific user and timestamp
> 6. Zero-retention contract with the AI model provider; no customer data is ever used for model training

Forecastle is operated by a small, hands-on team. Production database access is restricted to the founder. We are working toward SOC 2 Type I attestation; we do not claim compliance until the report is in hand. Where we are early on a control, this document says so explicitly — we would rather a prospect know the truth than discover it later.

1. About this document

2. Service overview & data scope

Forecastle is a Software-as-a-Service product accessed via web browser at `forecastle.app`. Customers subscribe to one of three plans (Core, Advanced, Multi-Entity); pricing and features are public at getforecastle.com/pricing.

Data we hold per customer

- **Authentication data:** user email addresses, bcrypt-hashed passwords, session tokens, optional SSO assertions
- **Financial data:** chart of accounts, monthly actuals (P&L and Balance Sheet), forecasts, scenarios, budgets, headcount plans
- **Dimensional metadata:** departments, locations, classes, tracking categories, custom dimensions
- **Operational data:** sheet workbooks, dashboards, reports, comments, workflow checklists, audit log
- **Integration credentials:** OAuth refresh tokens to Xero, QuickBooks and other connected services (encrypted at rest with a distinct key)
- **Billing data:** contact name, billing email, company, plan, invoice records. Card details are tokenised by Stripe and never touch Forecastle servers.

Data we do not hold

- Payment card numbers, CVCs, or bank account details (these live with Stripe)
- Personally-identifiable detail on payroll subjects beyond what the customer chooses to load (we accept either anonymised "Role" rows or per-employee rows, at the customer's choice)
- Customer-end-user data (Forecastle is a B2B tool; we do not hold the customer's own customers' data unless the customer chooses to load it)

3. Architecture & data flow

Forecastle's backend is a Python application served by a small number of containerised workers on Render. Persistence is managed Postgres, also on Render. The application loads each tenant's actuals into an in-memory OLAP cube on first request and serves pivoted reads from there; mutations are written through to Postgres and the cube is rebuilt on each generation bump. The database is the source of truth — every in-memory structure is a deterministic projection of dim and fact tables.

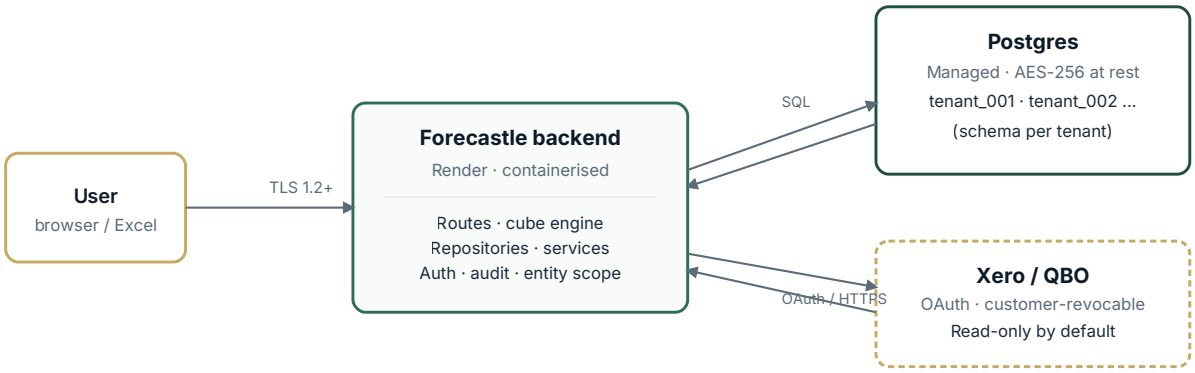

Figure 1 — High-level data flow. The customer's GL is the source of actuals; Forecastle's Postgres is the source of truth for everything inside the product.

Runtime components

- **Web tier:** stateless Python workers behind Render's load balancer. Horizontal scale-out is automatic on plan capacity.
- **Database tier:** managed Postgres with automated daily backups, point-in-time recovery to any minute within the last 7 days, and synchronous replication to a hot standby.
- **Background workers:** handle GL sync jobs, scheduled snapshot rolls, workflow reminders. They access the same Postgres with the same credentials scope.
- **Marketing site:** static HTML served by Vercel. No customer data crosses the marketing-site infrastructure.

4. Tenant isolation

Every customer is provisioned a dedicated Postgres **schema**. The tables that hold customer data — `fact_cube`, `dim_coa`, `sheets`, `dashboards`, etc. — exist independently in each tenant's schema. There is no shared-schema table with a `tenant_id` column for the data plane.

> **WHY THIS MATTERS**
>
> Most multi-tenant SaaS products use a shared schema and rely on application-level filters (`WHERE tenant_id = ?`) to keep data separate. A single bug or missing filter exposes one tenant's data to another. With schema-per-tenant, the SQL grammar itself enforces separation — a query that omits the schema name simply finds no tables.

The cross-tenant directory (which user belongs to which tenant, which plan, when last login) lives in a separate `public` schema and is accessed only through a small, audited surface

area. Application code receives a tenant identifier on every request, scopes the connection's `search_path` to that tenant's schema, and never holds raw cross-tenant queries.

How a request is scoped

1. The user authenticates and receives a session cookie bound to their workspace.
2. On every request, middleware reads the session, validates the workspace claim against the user record, and pins the tenant ID into a thread-local request context.
3. Every database connection checked out for that request executes `SET search_path TO tenant_xyz` as its first statement.
4. Repositories execute SQL without naming the schema; the search_path determines which schema's tables are touched.
5. At end of request, the connection is returned to the pool with its search_path reset.

If the request context is missing, the data-plane code raises and the request is rejected before any SQL executes. This is enforced in tests; the regression class is pinned with explicit "request without tenant scope must 401" tests in the suite.

5. Encryption in transit and at rest

In transit

- All public-facing endpoints require TLS 1.2 or higher. TLS 1.0 / 1.1 are disabled at the load balancer.
- Internal traffic between application workers and the database is encrypted; the Postgres connection enforces `sslmode=require`.
- HSTS is enabled on `forecastle.app` with a 1-year max-age and `includeSubDomains`.
- OAuth callbacks to Xero, QuickBooks and other GL providers use TLS-pinned certificate validation.

At rest

- Postgres volumes are encrypted with AES-256 at the infrastructure layer (Render manages key rotation and access).
- Automated backups inherit the same encryption.
- OAuth refresh tokens for Xero / QBO are additionally encrypted at the application layer with a key stored separately from the database connection string. A database-only dump does not expose connected-service credentials in plaintext.
- User passwords are hashed with bcrypt, cost factor 12. We do not store passwords in any reversible form.

6. Authentication, sessions, single sign-on

- **Password authentication** with bcrypt hashing, configurable minimum complexity, and rate-limited login endpoint.
- **Session cookies** are `HttpOnly`, `Secure`, `SameSite=Lax`. Sessions expire after 14 days of inactivity and are invalidated server-side on logout.
- **API tokens** for the Excel and Google Sheets add-ins are scoped (`read` default; `write` requires explicit opt-in) and revocable from the Settings panel.
- **SSO via SAML 2.0** available on the Advanced plan (Okta, Azure AD, Google Workspace tested).
- **SCIM provisioning** available on the Multi-Entity plan for customers managing 25+ users.
- **Mandatory MFA** is on the roadmap (currently optional, enforced per-account by the user).

7. Audit trail & logging

Every mutating operation in Forecastle — cell edit, plan switch, dim rename, account reparent, integration connect/disconnect, user invite, plan upgrade — is recorded in the per-tenant `audit` table with:

- Actor (user ID, email, IP at time of action)
- Timestamp (UTC, microsecond precision)
- Operation type
- Target entity and identifier
- Before value and after value (where applicable)
- Originating session ID and request ID

Customers can browse and export the audit log from inside the product under **Settings → Audit**. The full log is retained for the life of the tenant; export is available as CSV or JSON for SOX-style review.

Application logs (request logs, slow-query logs, error traces) are retained for 30 days on Render and shipped to Sentry for searchable error tracking. PII is scrubbed at the SDK boundary before transmission to Sentry.

8. AI use & Percival

Forecastle includes an AI assistant called **Percival** for natural-language queries against the cube, report-narrative generation, and operational diagnostics. Percival is built on Anthropic's Claude API.

> **We do not train any AI model on customer data.** Our agreement with our model provider includes a zero-retention clause: prompts and completions are not stored or used for training. This is the most important AI-related fact in this document.

What Percival can see

Per query, Percival receives only the rows and context required to answer the specific question. We do not send the entire cube on each prompt. The query planner extracts the minimal slice (e.g. "Revenue accounts, FY26 Forecast, Department=Engineering") and includes only that.

What Percival cannot do

Percival's system prompt enforces hard "never do" boundaries:

- No deletes (cells, accounts, sheets, users, tenants)
- No schema changes
- No cross-tenant access (the request context pins his scope)
- No plan upgrade or admin escalation
- No security overrides (password reset, MFA bypass, audit-log redaction)

When Percival cannot safely resolve something he escalates to a human via the in-product issue-report tool. He is architected as three explicit tiers (diagnose → heal → escalate) and each tier's tool surface is whitelisted — there is no generic SQL or generic update surface he can call.

9. GL connector model

The connection between Forecastle and a customer's general ledger is the most security-sensitive integration. We architect it conservatively:

- **OAuth 2.0 with PKCE** for Xero and QuickBooks Online. The customer authenticates directly with their GL; Forecastle receives a refresh token and never sees the customer's GL password.
- **Customer holds the token.** The customer can revoke Forecastle's access from inside Xero or QuickBooks at any time. We lose access instantly.
- **Read-only by default.** We pull chart of accounts, tracking categories / classes / locations, and monthly actuals. We do not write to the GL unless the customer explicitly enables write-back per integration (currently used only for sending budget journal entries on customer instruction).
- **Token encryption at rest** as described above — a database dump does not yield usable GL credentials.

- **Token-rotation alerts:** if a GL provider revokes our token we surface a banner in the product and email the workspace owner within 15 minutes.

10. Subprocessors

Forecastle uses a small number of subprocessors to deliver the service. The live list is published at getforecastle.com/security/subprocessors. We commit to notifying customers by email at least 30 days before adding a new subprocessor that handles personal or financial data.

SUBPROCESSOR	PURPOSE	LOCATION
Render	Application hosting, managed Postgres, backups	US-East (EU available on Advanced)
Cloudflare	CDN, DDoS protection, TLS termination	Global edge
Stripe	Payment processing, subscription billing	US / EU
Resend	Transactional email	US (AWS us-east-1)
Anthropic	AI model provider for Percival (zero-retention)	US
Sentry	Error tracking, performance monitoring	US
Vercel	Marketing-site hosting (no customer data)	Global edge
GitHub	Source-code hosting (no customer data)	US

11. Backups, RPO/RTO, disaster recovery

- **Recovery Point Objective (RPO):** 1 minute. Postgres point-in-time recovery is available for any minute within the prior 7 days. Daily snapshots are retained for 30 days.
- **Recovery Time Objective (RTO):** 4 hours for full service restore from cold. Hot-standby promotion in the same region is automatic and completes in under 60 seconds.
- **Backup testing:** we perform a documented restore drill quarterly. The last successful drill date is available on request.
- **Geographic redundancy:** backups are stored in a separate availability zone from primary. Cross-region replication (US ↔ EU) is available on Multi-Entity for customers with regulatory data-residency requirements.

12. Data deletion & retention

During active subscription

Customer-initiated deletion (deleting a sheet, removing an account, archiving a scenario) is soft-deleted to the audit log for 30 days, then hard-purged from production. Backups containing the deleted row roll off within 30 days of the soft-delete.

On cancellation

The workspace is held in a frozen state for 30 days in case the customer reactivates. After 30 days, the tenant schema is dropped from production. Backups containing customer data are aged out within 90 days. Written confirmation of deletion is available on request to security@forecastle.app.

Export

Customers can export every sheet, report and the full cube to Excel from inside the product at any time during the active subscription and during the 30-day frozen-state grace period.

13. Incident response

Forecastle commits to the following incident response standard for any confirmed security incident affecting customer data:

- **Within 24 hours of confirmation:** initial notification to affected customers, including what we know, what we do not yet know, and what we are doing about it.
- **Within 72 hours:** written summary including known impact scope, mitigation steps taken, and expected timeline to full resolution.
- **Within 14 days of resolution:** post-incident review including root cause, the control that should have prevented it, and the remediation we are committing to.

For service-availability incidents (degradation, outages), live updates are posted at status.getforecastle.com and customers can subscribe to per-incident email alerts.

14. Employee access & internal controls

Forecastle is operated by a small team. Production access controls are designed for that team size and will tighten as we grow:

- Production database access is restricted to the founder. There is no customer-support tier with read access to your cube.
- When a customer raises a support issue, we ask for a specific export or screenshot. We do not browse a customer's data to "look into it" without explicit consent recorded on the support ticket.

- Source code is hosted on GitHub with mandatory two-factor authentication on every commit-authoring account.
- Production deploys require an explicit human approval step; no automated deploys land schema changes without review.
- Laptops used for production access have full-disk encryption, automatic lock screen, and remote-wipe capability.

15. Compliance roadmap

We do not currently hold a formal compliance certification. We do not claim certifications we do not hold. Our roadmap, in order:

- **SOC 2 Type I attestation** — in progress. We are building toward the Trust Services Criteria as the baseline today and will engage a third-party auditor for the attestation once the first wave of paying customers is onboarded.
- **SOC 2 Type II** — expected within 12 months of Type I.
- **ISO 27001** — under evaluation; likely to follow SOC 2 Type II.
- **GDPR readiness** — signed DPA available today via security@forecastle.app. Article 28 subprocessor list published; data-subject-access request process documented and tested.
- **Third-party penetration test** — annual. Summary letter available under NDA to prospects evaluating a contract.

16. Contact & responsible disclosure

For security questions, DPA requests, pen-test summaries, vendor-risk questionnaires (SIG, CAIQ), or incident reports:

security@forecastle.app

Responsible disclosure

If you believe you have found a security vulnerability in Forecastle, please report it to the address above. We commit to:

- Acknowledge the report within one business day
- Keep you informed throughout investigation and remediation
- Publicly credit researchers who report in good faith (if you wish)
- Not pursue legal action against good-faith researchers who follow this policy

We do not currently run a paid bug bounty programme; this is on the roadmap.

Thank you for reading.

This document is versioned. The latest copy is always at getforecastle.com/security/whitepaper.pdf. Questions, gaps, or "you should also mention X" feedback is genuinely welcome at security@forecastle.app.