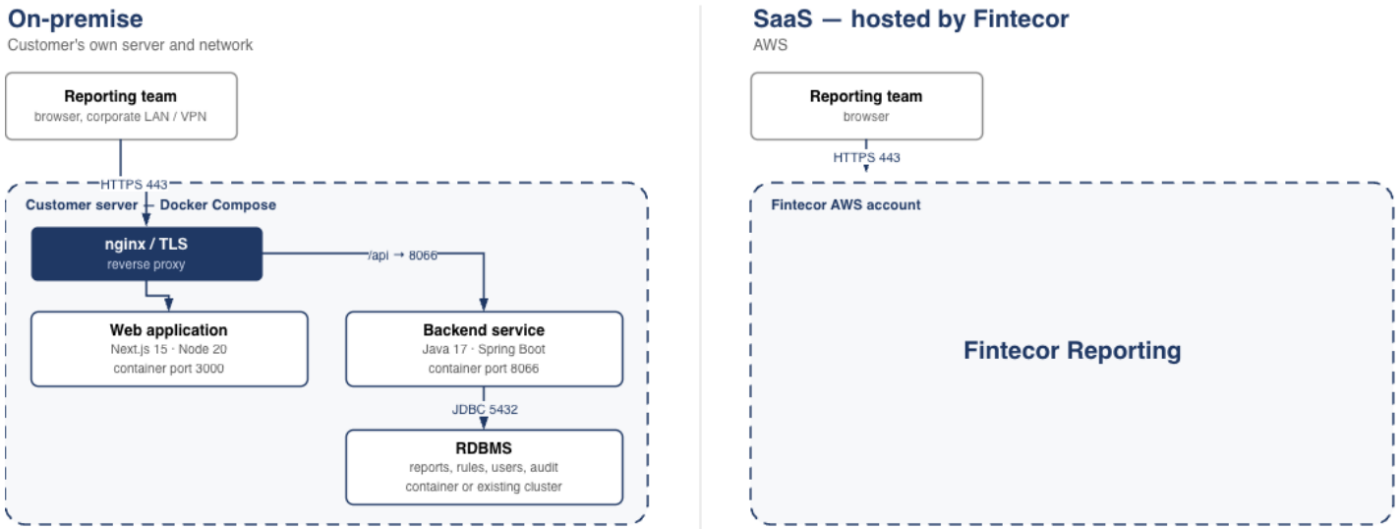
Solution Architecture



The platform is a three-tier application. A browser client talks over HTTPS to a stateless REST backend, which is the only component that touches the database.

Tier	Responsibility	Technology and notes
Client	Report grid, editing, drill-down views, dataset comparison, Excel import/export in the browser	React 18.
Application	Authentication, calculation processes, validation rule execution, breakdown queries, server-side export, audit	Java 17, Spring Boot, Spring Security; runs stateless, so it can be scaled to more than one instance when volume requires it
Data	Templates, report data, validation rules, glossary, users, audit history etc.	RDBMS, schema owned and migrated by the application.

Security

Authentication	Named user accounts with password plus TOTP two-factor authentication. Failed second-factor attempts are counted and the account is locked out temporarily.
Authorisation	Role-based and enforced in the backend, so an API client cannot reach data the role does not permit. Access to reporting data requires an authenticated principal.
Transport	HTTPS end to end. The browser origins permitted to call the API are an explicit allowlist, not a wildcard.
Secrets	Database credentials, the token signing key and the key protecting stored two-factor secrets are supplied per installation and held outside the image. Two-factor secrets are encrypted at rest with AES.
Logging	Structured application logs to the container log stream, collectable by whatever you already run. Health is exposed at a dedicated endpoint for your monitoring system to poll.

Revision #4
Created 2026-09-18 06:00:37 UTC by Samuel Lovell
Updated 2026-09-18 06:37:45 UTC by Samuel Lovell