

How Casement keeps every company's data separate.

The problem with shared databases

Casement is a shared, multi-tenant database: every company's properties, leases, and renters live in the same tables. That's efficient to run, but it means a single bug in application code -- a forgotten filter, a copy-pasted query -- could in theory expose one company's data to another. Casement is built so that mistake can't happen even if it's made.

What's actually enforced

- **Row-level security**

Postgres enforces tenant isolation at the database engine level, on every tenant table -- not as application-layer logic that a future bug could bypass.

- **Belt and suspenders**

Every tenant query also filters explicitly by company ID in application code, on top of the database-level policy -- two independent layers, not one.

- **A restricted runtime role**

The application connects to Postgres as a deliberately low-privilege role (`pms_app`), separate from the elevated role used only for migrations.

- **Company-scoped auth**

Every request is authenticated with a JWT carrying the user's company, checked on every protected route.

See the live demo: demo.casement.eu

Request a demo: casement.eu

This describes Casement's current data-isolation architecture, as implemented today. It is not a completed third-party security audit or a compliance certification.