

Wie Casement die Daten jedes Unternehmens getrennt hält.

Das Problem gemeinsam genutzter Datenbanken

Casement verwendet eine gemeinsam genutzte Mandanten-Datenbank: Die Immobilien, Mietverträge und Mieter jedes Unternehmens liegen in denselben Tabellen. Das ist effizient zu betreiben, bedeutet aber, dass ein einziger Fehler im Anwendungscode -- ein vergessener Filter, eine kopierte Abfrage -- theoretisch die Daten eines Unternehmens für ein anderes sichtbar machen könnte. Casement ist so gebaut, dass dieser Fehler nicht passieren kann, selbst wenn er gemacht wird.

Was tatsächlich durchgesetzt wird

- **Row-Level Security**

Postgres erzwingt die Mandantenisolation auf Datenbankebene, für jede Mandanten-Tabelle -- nicht als Anwendungslogik, die ein künftiger Fehler umgehen könnte.

- **Doppelt abgesichert**

Jede Mandantenabfrage filtert zusätzlich explizit nach Unternehmens-ID im Anwendungscode, zusätzlich zur Richtlinie auf Datenbankebene -- zwei unabhängige Schichten, nicht nur eine.

- **Eine eingeschränkte Laufzeitrolle**

Die Anwendung verbindet sich mit Postgres über eine bewusst rechthearme Rolle (`pms_app`), getrennt von der erweiterten Rolle, die nur für Migrationen verwendet wird.

- **Unternehmensbezogene Authentifizierung**

Jede Anfrage wird mit einem JWT authentifiziert, das das Unternehmen des Nutzers enthält und bei jeder geschützten Route geprüft wird.