Engineer Data

Built Stripe and in-app purchase entitlements

Built the payments and entitlements layer — Stripe alongside Apple and Google in-app purchase — gating the directory, search and export through an 11-table access model checked on the server.

2025

11

tables deciding who may see what

Situation. Money is the part of a platform nobody gets to be casual about. A subscription has to survive a card expiring, a refund, a plan change, a webhook that arrives twice and a webhook that arrives out of order. The moment payment can happen on three storefronts — a card on the web, Apple in one app store, Google in the other — there are three different accounts of what somebody bought, and the product still needs one answer to one question: what is this person allowed to do right now?

Task. Taking money and granting permission had to be two systems rather than one, so that adding a storefront would not mean rewriting every gate in the product.

Action. Stripe handles cards and subscriptions through 18 Go files, and Apple and Google in-app purchases arrive through their own receipt verification. All three converge on a payments schema of 9 tables and 34 functions — and then stop there. What the product actually asks is a separate access schema, 11 tables and 34 functions, which answers “may this account do this?” without knowing or caring which storefront paid for it. That answer gates the company directory, search and data export, and it is re-checked on the server on every request, because a hidden button is a courtesy and not a control.

Result. Adding a storefront now touches the payments side and leaves the gates alone, and a support question about someone's access has one table to look at rather than three. The cost is two schemas where a smaller product would want one, plus an entitlement lookup on requests that would otherwise have been free.

Bring this experience to your project

Use "Built Stripe and in-app purchase entitlements" as a starting point for a similar challenge in your organisation. The services behind this work are available for your project. A structure that matches how the business actually works, so the questions you need answered later are still possible to ask.

Explore our services Get in touch →

Look behind the result

Read the situation, decisions and result in this example. We can discuss which parts transfer to your constraints and which need another approach. Built Stripe and in-app purchase entitlements — 11 tables deciding who may see what

Explore our services Get in touch →

Need payments and access tiers done properly?

Talk through your project with the engineer responsible for delivery. We agree the scope, price basis and handover in writing before work begins.

Role context: Backend Engineer. It draws on Backend Engineering, APIs & Integration, Databases, PostgreSQL, Security and Product & Requirements. And while you are here, we are rather good at Backend & API Development, Database Design & Modeling, Security & Access Management and Product Strategy & Requirements.

Get in touch Explore our services

One piece of **a longer record** — and it is all on this site.

Every entry is written the same way: the situation, the task, what we did and what changed.

See the whole record

Define the improvement you need

Bring a representative dataset, the questions it must answer and the current bottleneck. We can agree how to measure quality, throughput and recovery.

See what it costs Get in touch →

Need payments and access tiers done properly?

Ask about the decisions behind “Built Stripe and in-app purchase entitlements”. You will speak with the people who can explain the work and assess your own project.

Explore our services

welcome@engineer.company +45 50 10 61 84 Copenhagen, Denmark

Take this page with you

Contact card

Carefully Crafted With Love by D.

CVR 45969703 · Tagensvej 105C, 3. th, 2200 København N

Copyright © 2025 – present · Engineer ApS · All rights reserved.

Printed from https://data.engineer.company/portfolio/built-stripe-and-in-app-purchase-entitlements/