Engineer Software

A maritime platform with clear system boundaries

Architected a layered maritime platform separating a Next.js PWA frontend, a Go (Huma/Fiber) API, and a PostgreSQL function layer, keeping all business logic in the database.

2025

How the request reaches the database

Implementation overview of the request path, with outcomes and measurement context kept in the case text.

1.

Frontend request

The maritime frontend requests structured data and presents the returned JSON.

2.

Go API

The Go API handles routing, request validation, security and the database call.

3.

PostgreSQL JSON

Database functions produce complete JSON, which the API returns unchanged.

one maritime platform, in three clean layers

Situation. NextMariner is a maritime platform covering professional networking, company reviews and educational sponsorships. The founder's architecture contribution is recorded under the NextMariner engagement beginning in November 2025; the record does not distinguish a later employment period or establish that all of the product was delivered by Engineer ApS.

Task. As the architect, the call to make up front was where each kind of logic lived, with boundaries obvious enough to hold up under pressure instead of blurring the first time someone was in a hurry.

Action. It settled into three layers, one job each. The Next.js frontend does presentation and interactivity and nothing else. The Go API — Huma over Fiber — is deliberately thin: it routes, validates the request, applies the security filtering and orchestrates, but it holds no business logic at all. The business logic all lives in PostgreSQL functions, which assemble the full result and hand it back for the API to forward. So when a rule changes, it changes in one layer, in SQL, and the other two don't need to know. The boundaries were written down and then enforced in review, because a convention nobody polices stops being one.

Result. The implemented architecture gives presentation, request handling and business rules separate responsibilities. The source records that separation and its rationale; it does not establish a measured improvement in delivery speed, reliability or business performance.

Bring this experience to your project

Use "A maritime platform with clear system boundaries" as a starting point for a similar challenge in your organisation. The services behind this work are available for your project. The shape of a system worked out on paper first, so the parts fit and the cost is known before anyone writes code.

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. A maritime platform with clear system boundaries — one maritime platform, in three clean layers

Explore our services Get in touch →

Need a clean, layered architecture?

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: Platform Architect. It draws on Platform Architecture, Solution Architecture, Backend Engineering, Frontend Engineering, Full-Stack Development, APIs & Integration, Databases, PostgreSQL and Technical Leadership. And while you are here, we are rather good at Platform & Solution Architecture, Backend & API Development, Frontend Development, Full-Stack Product Development and Database Design & Modeling.

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

Describe the workflow, the existing interfaces and the users who need the change. We can define a testable delivery and a handover that lets your team maintain it.

See what it costs Get in touch →

Need a clean, layered architecture?

Ask about the decisions behind “A maritime platform with clear system boundaries”. 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://software.engineer.company/portfolio/a-maritime-platform-built-in-clean-layers/