Engineer Software

Built organization context switching for users

Designed an organization context-switching system with client localStorage and server-side cookie mirroring, letting users act as managed organizations while enforcing least-privilege authorization.

2025

act as any organisation you manage, safely

Situation. On the platform a maritime professional can manage organizations — companies, academies — that don't have their own logins. The person is the account; the organization is something they act on behalf of. So a user needs to move through the whole app as any organization they manage, switching between them freely, and that convenience can't turn into a hole in the authorization.

Task. Context-switching had to be quick and unobtrusive for the user while making sure the active context could never, on its own, hand someone access they weren't entitled to.

Action. An OrganizationContext handles it, with storage on both sides. On the client, localStorage is the source of truth for which organization you're currently acting as, so switching is instant — no round-trip. A server-side cookie mirrors it so that server-rendered pages resolve the same context during SSR; there's a getServerViewMode on the server that reads it. The important part is that none of that is trusted for access decisions. Authorization gets re-checked on the server on every request. The frontend context is there for the experience — showing you the right thing — and the server is the only authority on what you're allowed to do.

Result. A user can act as any organization they manage without friction, and the interface stays in sync on both client and server. But because permissions are verified server-side every time, none of that convenience weakens the

security. Someone tampering with what's in localStorage changes what their own UI shows them and nothing more — the server still says no.

Bring this experience to your project

Use "Built organization context switching for users" 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. Built organization context switching for users — act as any organisation you manage, safely

Explore our services Get in touch →

Building multi-tenant, role-aware access?

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, Backend Engineering, Frontend Engineering, Full-Stack Development, APIs & Integration and Security. And while you are here, we are rather good at Platform & Solution Architecture, Backend & API Development, Frontend Development and Security & Access Management.

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 →

Building multi-tenant, role-aware access?

Ask about the decisions behind “Built organization context switching for users”. 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/built-organization-context-switching-for-users/