

## CONTACT


+45 2889 8626

jesper.halborg@compass-it.dk

Horsens, Denmark

linkedin.com/in/jesper-halborg


## SKILLS

**Leadership and process**

- Tech lead and engineering manager
- Release management and rollouts
- Onboarding teams and developers
- Refinement and technical enablers
- Developer experience

**Agentic development**

- Claude Code and agentic workflows
- Guardrails, hooks and review loops
- Spec-driven development
- AI tooling for non-developers

**Architecture and platform**

- TypeScript end-to-end
- GraphQL and OpenAPI
- Node.js and PostgreSQL
- CI/CD, feature flags, observability
- Cloud: Fly.io, Cloudflare, Azure, AWS, GCP
- Micro-frontends
- IoT: sensors and actuators

**Client**

- React and React Native
- Vite, TanStack, design systems

## ABOUT ME

I am 37, a father and husband, and live in Horsens, Denmark. Spare time goes to sports, video and board games, good beer and time with friends and family.

## EDUCATION

Bachelor in IT Product Development from Aarhus University, plus a dozen master's-level courses in web and application development.

# Jesper Halborg

Engineering Manager · Tech Lead · Agentic Development

## Engineering manager and tech lead who makes the whole team run faster

I am a player-coach for development teams: in the codebase every day, and at the same time responsible for architecture, process and the team's flow. That is what I have been as Frontend Tech Lead at Grundfos, tech lead at LEGO and CTO at Brandheroes.

The product owner sets the direction. My job is to remove roadblocks and give the team the tooling and processes that make everyone run faster towards the goal, not just me. And to make sure they enjoy the ride: developers who like their stack and their tools build better products.

I understand the product as a whole, from client through API and database to CI pipelines, cloud and everything that ties the layers together, at Grundfos down to IoT sensors and actuators. That overview matters more than ever now that AI writes a large share of the code. As the solo founder of Slipp I have built, and run, an entire platform from scratch.

> **AI is a force multiplier, for better and for worse**
>
> It is impressive what an experienced developer can produce with AI, and just as worrying what an inattentive one can. I was a strong developer long before AI arrived, and I know how to set up the tooling, guardrails and processes a team needs to get the good out of this new, AI-driven way of building software. At Grundfos I was the primary driver of agentic coding in the team. Slipp is built fully agentically.

## EXPERIENCE

2023 – 2026 · **Frontend Tech Lead, freelance** · Grundfos See page 2

Lead for four frontend developers in a team of 10–15. Took the team from monthly, dreaded releases to daily releases across all services. Primary driver of agentic coding, the GraphQL cleanup and micro-frontends.

2026 – · **Founder and developer** · Slipp See page 3

Mobile ordering for food stalls at events and festivals. The entire platform, from database and payments to apps and operations, built and run solo with agentic coding. Rolling out to the market.

2020 – 2022 · **Senior consultant via SustainIT** · LEGO See page 4

React Native apps for maintaining production equipment in LEGO's factories. Introduced GraphQL as the layer between the clients and the SAP-based services.

2018 – 2020 · **CTO** · Brandheroes See page 5

Built the team and the micro-influencer marketing platform from scratch, leading both in-house developers and external consultants.

2017 – 2020 · **Co-founder** · Plature

Jäger, an app for hunters, built in React Native and GraphQL.

2014 – 2018 · **Software developer and IT project manager** · IT Minds

GRUNDFOS · JANUARY 2023 – DECEMBER 2026 · FRONTEND TECH LEAD, FREELANCE

# From monthly releases everyone dreaded, to daily

4 frontend developers · 10–15 in the team

**Daily releases instead of monthly**

**Agentic coding across the whole team**

**Micro-frontends and GraphQL in front of legacy**

The team builds one of Grundfos' digital customer platforms: an IoT platform that monitors and controls physical equipment. A missing alarm or a wrong command can cost millions in property damage and, at worst, personal injury, so users must be able to trust the data even when the connection is unstable. Technically it is React frontends on top of a microservice-based backend with GraphQL as the unifying layer. I joined as frontend tech lead and over time took responsibility for process, releases and technical enablement across the team.

## Releases nobody is nervous about any more

When I arrived, releases happened monthly, the team dreaded them, and each one was followed by a wave of bugs. The backend was built as microservices but behaved like a distributed monolith, with lockstep deployments and no real way to roll back.

Today we release daily across all services. Many people and many decisions made that happen, but the enablers I introduced were:

- Swagger specs and enforced OASDiff checks on every backend
- API and GraphQL schema registries to manage breaking changes and see schema usage across services and environments
- Error monitoring for proactive bug handling
- A release process with easy rollbacks and cherry-picking
- Feature flags for granular rollouts and instant rollbacks

## GraphQL used properly

The team used GraphQL as a REST proxy. Over time I trained the team in the right way to use it and had the codebase reworked to play to GraphQL's strengths: one schema, precise queries and type safety all the way to the client.

## Agentic coding across the whole team

In 2026 I was the primary driver of introducing agentic coding in the team, including the guardrails, review discipline and tooling that come with it. It has spread to teams around us and given our non-developers new options.

One example: I built a setup that let our product owner build mockups of features directly in the React codebase and catch UX problems before anything was built. Handovers got sharper, and calendar time from idea to release dropped by up to 80%.

## Teams in and out, without the quality wobbling

Along the way we onboarded and offboarded an entire external team of frontend and backend developers. I owned the onboarding of the frontend developers and the guardrails that kept users from noticing any dip in quality.

Later we introduced a micro-frontend architecture so sister teams at Grundfos can contribute with NPM packages or whole micro-frontends rendered in the main app. Process, tooling and setup were my responsibility.

### TECHNOLOGIES

**Frontend**

React, TypeScript, GraphQL, Micro-frontends, Feature flags

**Platform and process**

Swagger/OpenAPI, OASDiff, Schema registries, Error monitoring, Release management

**Agentic**

Claude Code, Guardrails and hooks, PO mockups in the codebase

*"You have played a central role in getting the product to a good place. You came in and got refinement, deployments and GraphQL under control, and you have helped us work more deliberately with our technical enablers. Whenever we have had hard bugs or performance problems, you have presented several approaches and then recommended one. I have used that several times as an example to others of what good behaviour looks like. Your judgement is highly trusted in the team, and for good reason. But what I appreciate most is that you keep pushing forward, most recently with AI. That drive is rare, and it is contagious."*

— Colleague at Grundfos, translated from Danish. Reference on request.

SLIPP · JANUARY 2026 – · FOUNDER, DEVELOPER AND EVERYTHING IN BETWEEN

# One person, the whole stack: a payments platform built and run solo

Solo · the whole platform from database to operations

**The whole stack built and run solo**

**Payments with Stripe Connect**

**Agentic end-to-end tests against real browsers**

Slipp is mobile ordering for food stalls at events and festivals. Guests scan a QR code at the stall, order and pay from their phone, and pick up when the order is ready. The stall sees orders live, and the organizer gets one overview across every stall. I build, run and sell all of it myself, and that is the point: you cannot build something strangers pay through without understanding every single layer.

## The architecture

A Hono API with Drizzle and PostgreSQL on Fly.io, Redis for rate limits and live events, React apps for guests and stalls on Cloudflare Pages, and an Astro marketing site. Payments run on Stripe Connect with embedded onboarding, payouts, refunds and disputes. Live orders are delivered over Server-Sent Events with heartbeats and fetch-on-reconnect, because festival networks are terrible.

## Release discipline from day one

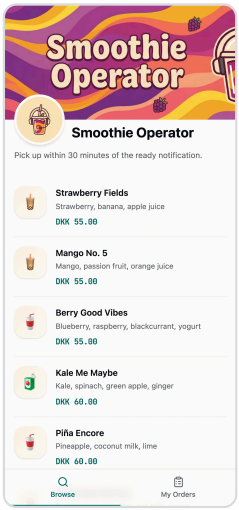
Everything I have introduced at clients runs here too: an OpenAPI spec generated from the code, an OASDiff gate against breaking changes, schemas published per environment and consumed by the apps through codegen, migrations as a release step before traffic, QA and prod environments with separate deploy tokens, feature flags in LaunchDarkly and observability in BetterStack with source maps and uptime.

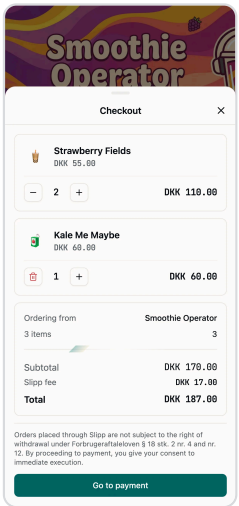
## Built fully agentically

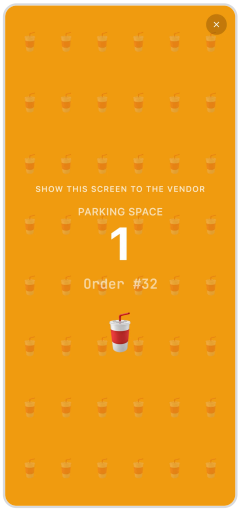
Every feature starts as an OpenSpec change with requirements and scenarios, is implemented by agents in worktrees, self-reviewed against a fixed checklist and gated by hooks before a PR may be opened. An agentic e2e harness plays guest, stall and organizer in real browsers and verifies the whole order loop against staging. It is the process I set up for teams, tested on myself for ten months.

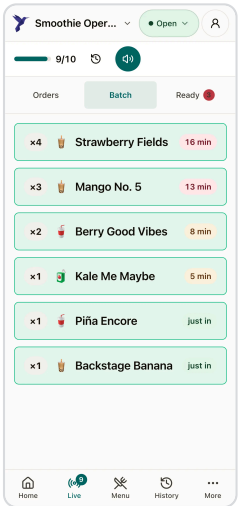
## Security and money

Every endpoint requires auth by default with documented exceptions, org scoping and roles, RFC 7807 errors, SSRF protection on imports, rate limits per namespace and an audit log. The fee model is worked through so the stall receives 100% of the menu price, with the Stripe fee netted out at charge time.




## TECHNOLOGIES

**Backend**

Node.js, Hono, Drizzle, PostgreSQL, Redis, Better Auth, Zod, OpenAPI

**Clients**

React, Vite, TanStack Router and Query, Zustand, shadcn/ui, Astro, i18n

**Platform**

Fly.io, Cloudflare Pages and Images, GitHub Actions, Stripe Connect, LaunchDarkly, BetterStack, PostHog, OpenTelemetry, Resend

**Process**

Claude Code, OpenSpec, Linear, Playwright e2e, oasdiff

LEGO · 2020 – 2022 · SENIOR CONSULTANT, FREELANCE VIA SUSTAINIT

# Hit the ground running, then a GraphQL layer in front of SAP

Global Maintenance System · React Native and GraphQL

| Features in production within days | GraphQL layer in front of SAP in two sprints | Equipment in factories worldwide |
|---|---|---|

LEGO's Global Maintenance System department set out to roll out new apps and web apps to replace an outdated SAP-based workflow from 2003. They wanted to finish the conversion as quickly as possible, so they hired me to help build their React Native apps. In short, the apps manage maintenance and repair jobs and documentation of all production equipment across LEGO's factories worldwide.

## Hit the ground running

I was primarily hired to work on the React Native apps, but after working with the team for a while I realized how much they could speed up their development and benefit from adopting GraphQL as a mediator between their clients and the REST services that talk to SAP.

I pitched the idea together with a proposal for a gradual adoption process, and we had a GraphQL service and a GraphQL-based app feature in production within two sprints, all while keeping up with our other sprint goals.

The benefits of the new approach showed immediately, and the return on investment grows exponentially as new features and clients are added to the platform.

## The role in the team

Beyond the codebase I took responsibility for direction and priorities: constructive feedback and suggestions for improvements from day one, and helping the team stay focused on the goal and sort out distractions. That is the role the scrum master describes in the statement below.

> *"If you're looking to speed up the development of your React/React Native or Node/GraphQL project, I highly recommend adding Jesper to your team. When he joined our team at LEGO, he was delivering features to our app within just a couple of days, and he got into the existing code base very quickly. The other developers on our team are also reporting that he is great to work with, both in terms of development skills and as a person. He provided constructive feedback and suggestions for improvements from the very beginning, as well as spreading contagious positive vibes and energy to the other team members. Besides being a strong, team oriented developer, Jesper also has a natural flair for focusing on the goal, sorting out distractions, and helping the team stick to the prioritizations. I think he would be a great addition to any team."*
>
> — Gitte Taasti Landberg, Scrum Master, LEGO. Reference on request.

### TECHNOLOGIES

**Mobile**

React Native, Apollo Client, BLoC, i18next, rxjs

**Backend**

Node.js, GraphQL Nexus, dataloader, Express, Apollo Server, Swagger

**Infrastructure**

Docker, Kubernetes, GitHub Actions, Microsoft Azure, AWS

**Shared**

TypeScript, Jest, ESLint, Prettier, Sentry, Luxon

BRANDHEROES · 2018 – 2020 · CTO

# A greenfield team and a platform that scaled micro-influencer marketing

In-house team, freelancers and consultants

**Greenfield team built as CTO**

**Platform for micro-influencer campaigns**

**Rolling updates without downtime**

Brandheroes does micro-influencer marketing. In short, it is a marketing model where a brand pays Brandheroes to find a number of people who fit the brand profile and reward them for posting recommendations for the brand on social media, primarily Instagram. As CTO I was in charge of the entire platform that matches the users with the customers. I managed the in-house development team and our freelancers and consultants, and did hands-on implementation every day.

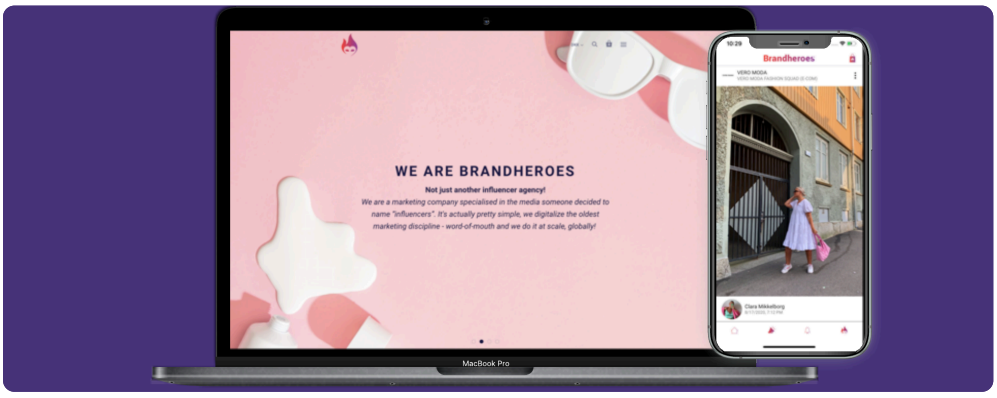
## A developer's dream

This was a greenfield project, so my team and I had the rare opportunity to pick the technologies we wanted, and it paid off for both us and Brandheroes. The result was a tech stack with synergies across the entire platform. Prisma and GraphQL combined allowed for full type safety from the database types all the way to the client types, all auto-generated and checked for errors at compile time and in our build pipelines. Furthermore, the all-TypeScript stack allowed for a high level of code sharing across all projects, and for developers to help each other across the usual backend/frontend dividing line.

The end result was a synergetic platform with continuous integration and near 100% uptime, and rolling, seamless updates for all projects, including the mobile app, where Expo and over-the-air updates let us bypass the app store review process for smaller patches.

## A complete platform for managing users, customers and campaigns

During my time with Brandheroes I built the team and the platform that allowed Brandheroes to scale influencer marketing effectively and manage thousands of users and posts across thousands of campaigns for their customers. Brandheroes now runs hundreds of marketing campaigns in parallel with ease.




### TECHNOLOGIES

**Web and mobile**

React, React Native, Expo, Apollo Client, Redux, Material UI, Formik, Algolia

**Backend**

Node.js, Express, Apollo Server, Prisma, AWS Aurora SQL, S3, GraphQL schema delegation, Postmark, Sharp

**Infrastructure**

Docker, Kubernetes, CircleCI, Bitbucket Pipelines, Heroku, AWS, Google Cloud, RabbitMQ, NATS, Humio

**Shared**

TypeScript, Jest, ESLint, Prettier, Husky, Sentry, Yup, Luxon