

Tech Stack

Tech Stack — Lobby

Statusoversikt

Lag	Teknologi	Status
Frontend	Next.js 15 + React 19 + TypeScript + Tailwind + shadcn/ui	Vedtatt
Backend	TBD — Node.js/TypeScript ELLER Java/Spring Boot	ADR needed
Database	PostgreSQL + RLS (multi-tenancy)	Vedtatt
AI	Claude API (Anthropic)	Vedtatt
Mobile	React Native + Expo	Phase 2
Infra	TBD — Vercel+Supabase vs Azure	ADR needed
Monorepo	Turborepo	Vedtatt (følger Bilko-mønster)

Frontend

Stack: Next.js 15 + React 19 + TypeScript + Tailwind CSS + shadcn/ui

Valg	Begrunnelse
Next.js 15	SSR, App Router, server components — samme som Bilko (delte lærdommer)
React 19	Server Actions, concurrent features — reduserer boilerplate
TypeScript	Type safety, bedre DX, færre runtime-feil
Tailwind CSS	Utility-first, rask prototyping, konsistent design
shadcn/ui	Tilgjengelig, kopierbar komponent-bibliotek, ikke locked-in

Backend (ADR Pending)

Status: Beslutning kreves av CEO/arkitekt

Alternativ A — Node.js + TypeScript (Hono/Express)

Fordel	Ulempe
Samme språk som frontend (TypeScript)	Svakere type-system vs Java
Raskere onboarding for JS-devs	Event loop — ikke egnet for CPU-heavy tasks
Lettere deployment (Vercel/Railway)	Mindre modent for enterprise-patterns
Bilko-erfaring kan gjenbrukes	

Alternativ B — Java + Spring Boot

Fordel	Ulempe
Sterke typer, robust for enterprise	Annet språk enn frontend
Godt testet for compliance-systemer	Tyngre oppstart og deployment
Utmerket støtte for multi-tenancy patterns	Krever Java-kompetanse
LumisCare bruker Java — erfaring finnes	

Anbefaling: Node.js/TypeScript for Phase 1 (fart og konsistens). Java vurderes for Phase 3+ (kompleksitet øker).

Database

Stack: PostgreSQL + Row Level Security (RLS)

Multi-tenancy via RLS

```
-- Hvert selskap er en "tenant"
-- RLS sikrer at firma A aldri kan se firma Bs data
ALTER TABLE employees ENABLE ROW LEVEL SECURITY;
```

```
CREATE POLICY tenant_isolation ON employees
  USING (company_id = current_setting('app.current_company_id')::uuid);
```

Kjernedatamodell

```
Company (org_number, name, address, industry)
├─ Employee (personal_info, employment_info, role)
|   ├─ Document (type, file, expiry_date)
|   └─ Absence (type, start, end, status, approver)
└─ OnboardingTask (step, status, due_date)
├─ Department (name, parent, manager)
├─ HMSHandbook (version, content, generated_at)
└─ ComplianceCheck (regulation, status, last_checked)
```

ORM: Prisma (type-safe, god migrering, støtter PostgreSQL RLS)

AI

Stack: Claude API (Anthropic)

Modell	Brukstilfelle
Claude Haiku	Rutineoppgaver — kategorisering, varsler, enkle svar
Claude Sonnet	Kompleks generering — HMS-håndbøker, onboarding-flyter, analyser

AI-brukstilfeller Phase 1

- 1. **HMS-håndbok-generator** — Company info → 40–80 siders compliant håndbok
- 2. **Onboarding-flyt-generator** — Stillingsbeskrivelse → komplett plan
- 3. **Arbeidslogg-transkripsjon** — Talenotat → strukturert logg
- 4. **HR-analyser** — Fraværdata → innsikt og varsler
- 5. **Compliance-status** — Regelendringer → påvirkning for kunden

Mobile (Phase 2)

Stack: React Native + Expo

- Deler UI-komponenter med web via shared packages
- Native check-in (GPS, kamera for dokumenter)
- Offline-støtte for feltarbeidere
- Push-varsler for vaktplaner og fravær

Infrastruktur (ADR Pending)

Alternativ A — Vercel + Supabase

Fordel	Ulempe
Rask oppstart, minimal ops	US-hosting (GDPR-kompleksitet)
Bilko og andre ALAI-produkter bruker dette	Supabase RLS-quirks
Vercel AI SDK integrerer godt med Claude	Kostnad skalerer raskt

Alternativ B — Azure (Norway East)

Fordel	Ulempe
Norsk datacenter — sterk GDPR-posisjon	Høyere ops-overhead
Compliance selling point	Tregere oppstart
Azure AD → BankID-integrasjon	Dyrere for tidlig fase

Anbefaling: Vercel+Supabase for Phase 1 (fart), migrer til Azure Norway East etter 50+ kunder (compliance selling point).

DNA fra LumisCare

Referanse: `~/projects/client/lumiscare/` — enterprise helseplattform vi bygde.

Lobby deler DNA med LumisCare:

Domene	LumisCare	Lobby
Besøk	Hjemmebesøk (hjemmesykepleie)	Kundebesøk (feltarbeid)
Planlegging	Vaktplanlegging (sykepleiere)	Skiftplanlegging
HR	Ansathåndtering	HR-modul

Domene	LumisCare	Lobby
Multi-tenancy	Per kommune	Per bedrift
Compliance	Norske helseregler	Arbeidsmiljøloven

Gjenbruk: Autentiseringsmønstre, multi-tenancy RLS-arkitektur, besøks-domenmodell, planleggingslogikk. **Unngå:** LumisCares over-engineering for enterprise — Lobby starter lettere.

Ikke-funksjonelle krav

Krav	Mål
Responstid	<500ms for alle sider
Oppetid	99.5%+
GDPR	DPA, sletting, eksport fra dag 1
Språk	Norsk bokmål primær, nynorsk Phase 2
Mobil	Responsiv web Phase 1, native app Phase 2
Sikkerhet	OWASP top 10, SOC 2 (Phase 3+)

Revision #4
Created 2026-03-04 05:09:13 UTC by John
Updated 2026-07-12 20:01:24 UTC by John