

TransportPlan

Technische Architectuur & Development Guide

Versie 1.0 — Februari 2026

Vertrouwelijk — Elevate Labs

Dit document beschrijft de volledige technische architectuur, tech stack, database-opzet en development-richtlijnen voor het bouwen van de TransportPlan MVP.

Kernprincipes:

- Simpel en snel — geen overengineering
- Multi-tenant vanaf dag 1
- Geen externe API-koppelingen in de MVP
- Client-side export (PDF/Excel)
- Mobile-responsive voor tablet-gebruik

1. Tech Stack

Frontend

Technologie	Versie	Doel
React	18+	UI framework
Vite	5+	Build tool & dev server
Tailwind CSS	3.4+	Styling
React Router	6+	Client-side routing
Zustand	4+	Lichtgewicht state management
React Hook Form	7+	Formuliervalidatie
date-fns	3+	Datummanipulatie (NL locale)

Backend & Database

Technologie	Doel
Supabase	PostgreSQL database + Auth + Realtime + Storage
Supabase Auth	Login/registratie met e-mail of magic link
Row Level Security	Data-isolatie per bedrijf (multi-tenant)

Export & Hosting

Technologie	Doel
jsPDF + html2canvas	PDF-export van planningen
SheetJS (xlsx)	Excel-export van planningen
Vercel	Frontend hosting + CI/CD
Supabase Cloud	Database hosting
Stripe	Betalingen (Fase 5)

2. Database Schema

Het schema is opgebouwd rond het multi-tenant principe: elke tabel bevat een `company_id` foreign key die via Row Level Security wordt afgedwongen.

ERD Relaties

```
companies → drivers, trucks, clients, trips
trucks → maintenance
trips → drivers, trucks, clients
```

Tabel: companies

Kolom	Type	Beschrijving
id	uuid (PK)	Unieke ID
name	text	Bedrijfsnaam
kvk_number	text	KvK-nummer
email	text	Hoofde-mail
phone	text	Telefoonnummer
plan	text	'starter' / 'groei' / 'pro'
stripe_customer_id	text	Stripe klant-ID (later)
created_at	timestampz	Aangemaakt op

Tabel: drivers

Kolom	Type	Beschrijving
id	uuid (PK)	Unieke ID
company_id	uuid (FK)	Bedrijf
name	text	Volledige naam
phone	text	Telefoonnummer
license_type	text	'C' / 'CE' / 'C1' / 'C1E'
status	text	'actief' / 'verlof' / 'ziek'
notes	text	Opmerkingen

Kolom	Type	Beschrijving
created_at	timestampz	Aangemaakt op

Tabel: trucks

Kolom	Type	Beschrijving
id	uuid (PK)	Unieke ID
company_id	uuid (FK)	Bedrijf
license_plate	text	Kenteken
brand_model	text	Merk en type
apk_expiry	date	APK vervaldatum
status	text	'beschikbaar' / 'onderhoud' / 'defect'
mileage	integer	Kilometerstand
created_at	timestampz	Aangemaakt op

Tabel: trips

Kolom	Type	Beschrijving
id	uuid (PK)	Unieke ID
company_id	uuid (FK)	Bedrijf
driver_id	uuid (FK)	Chauffeur
truck_id	uuid (FK)	Voertuig
client_id	uuid (FK)	Klant (optioneel)
date	date	Ritdatum
departure_time	time	Vertrektijd
origin	text	Vertreklocatie
destination	text	Bestemmingslocatie
status	text	'gepland' / 'onderweg' / 'afgerond'
created_at	timestampz	Aangemaakt op

Tabel: maintenance

Kolom	Type	Beschrijving
id	uuid (PK)	Unieke ID
company_id	uuid (FK)	Bedrijf
truck_id	uuid (FK)	Voertuig
type	text	'apk' / 'onderhoud' / 'reparatie'
scheduled_date	date	Geplande datum
completed_date	date	Afgeronde datum
status	text	'gepland' / ' bezig' / 'afgerond'
cost	decimal	Kosten
created_at	timestampz	Aangemaakt op

3. Authenticatie Flow

Registratie (nieuw bedrijf)

1. Gebruiker vult in: bedrijfsnaam, naam, e-mail, wachtwoord
2. Frontend roept Supabase Auth signup aan
3. Database trigger maakt automatisch een company + profile record
4. Gebruiker wordt doorgestuurd naar het dashboard

Login

1. Gebruiker vult e-mail + wachtwoord in
2. Supabase Auth valideert en retourneert JWT token
3. RLS policies gebruiken `auth.uid()` om data automatisch te filteren
4. Gebruiker ziet alleen data van eigen bedrijf

Row Level Security

Elke tabel heeft RLS-policies die afdwingen dat gebruikers alleen hun eigen bedrijfsdata kunnen lezen, aanmaken, wijzigen en verwijderen. Dit wordt bereikt via een helper-functie `get_user_company_id()` die de `company_id` van de ingelogde gebruiker ophaalt.

4. Frontend Architectuur

Projectstructuur

```
src/components/layout/ - Sidebar, TopBar, AppShell
src/components/ui/ - Modal, StatusBadge, StatCard, DataTable
src/components/drivers/ - DriverList, DriverForm, DriverCard
src/components/trucks/ - TruckList, TruckForm, TruckCard
src/components/planning/ - WeekView, DayColumn, TripCard, TripForm
src/pages/ - Dashboard, Planning, Drivers, Trucks, etc.
src/hooks/ - useDrivers, useTrucks, useTrips, etc.
src/lib/ - supabase.js, exportPdf.js, exportExcel.js
```

Custom Hooks Pattern

Elke entiteit (drivers, trucks, trips, maintenance, clients) krijgt een eigen custom hook die alle CRUD-operaties afhandelt via Supabase. De hook beheert lokale state en synchroniseert met de database. Dit pattern houdt de componenten schoon en de business logica herbruikbaar.

Export Functionaliteit

PDF: jsPDF met autotable plugin genereert weekplanningen als PDF, volledig client-side. Geen server-side rendering nodig.

Excel: SheetJS genereert .xlsx bestanden met correcte kolombreedtes en NL-datumformaten.

5. Deployment & Security

Deployment

De app wordt gehost op Vercel met automatische deploys vanuit GitHub. Custom domein: app.transportplan.nl via CNAME naar Vercel DNS. HTTPS is automatisch inbegrepen.

Security Checklist

Check	Status	Toelichting
RLS op alle tabellen	Vereist	Data-isolatie per bedrijf
Alleen public anon key in frontend	Vereist	Nooit service role key
.env.local in .gitignore	Vereist	Keys nooit committen
Input validatie	Vereist	Client + database constraints
Database indexen	Vereist	Op company_id en date kolommen
HTTPS	Automatisch	Via Vercel
Rate limiting	Standaard	Via Supabase

6. Benodigde Packages

```
npm create vite@latest transportplan -- --template react
```

Core: @supabase/supabase-js, zustand, react-router-dom, react-hook-form

UI: tailwindcss, @tailwindcss/forms, postcss, autoprefixer, lucide-react

Export: jspdf, jspdf-autotable, xlsx

Dates: date-fns