

Platform voor toekomstige autonome mobiliteit

Cloudgebaseerde ritplanning met autonome voertuigen

Situering

Mobiliteit staat steeds meer onder druk: meer files, voertuigen die niet efficiënt worden gebruikt en een groeiende vraag naar flexibel en duurzaam vervoer. Tegelijk worden auto's autonome en zijn diensten zoals deelwagens (Cambio, Poppy) en ritplatformen (Uber, Bolt) populair.

Door zelfrijdende voertuigen te combineren met een dienstmodel kan mobiliteit efficiënter, flexibeler en duurzamer worden. In dit model blijven voertuigen eigendom van een vlootbeheerder (of de fabrikant zelf) en worden ze optimaal ingezet op basis van vraag. Vandaag is een volledig centrale en voorspellende planning moeilijk te realiseren, doordat menselijke chauffeurs werken met eigen voertuigen, vaste werkuren en persoonlijke voorkeuren.

Dit eindwerk vertrekt vanuit deze toekomstvisie en ontwerpt software om zulke autonome mobiliteit in een dienstmodel conceptueel te organiseren.

Doel en functionaliteiten

Het doel van dit eindwerk is het ontwerpen en uitwerken van een prototype van een cloudgebaseerd softwareplatform datritten met autonome voertuigen kan plannen en beheren.

Het platform simuleert hoe een centrale server automatisch bepaalt welk voertuig welke rit uitvoert, via welke route, en of ritten gedeeld worden. Daarnaast kunnen gebruikers vooraf een rit aanvragen en daarbij keuzes maken op het vlak van comfort en mate van ritdeling, wat een invloed heeft op de prijs. Beheerders krijgen via het platform inzicht in de vloot, geplande en actieveritten en kunnen in beperkte mate ingrijpen in de planning.

Motivatie

De motivatie voor dit project komt voort uit mijn passie voor programmeren en netwerken, en mijn interesse in technologische innovatie. Ik zoek graag naar oplossingen voor concrete maatschappelijke problemen via programmatie. Als beginnend chauffeur waardeer ik het gebruik van een voertuig, maar ervaar ik de nadelen van files. Met slimme software en dienstmodellen kunnen we mobiliteit efficiënter, duurzamer en flexibeler maken. Dit project past daar heel goed bij.

Afbakening en uitbreidingsmogelijkheden

In de realiteit zijn er veel factoren waarmee kan/moet rekening worden gehouden. Voor het eindwerk is het belangrijk om erop te waken dat het project niet te complex wordt.

Voor de haalbaarheid van dit project, beperk ik mij tot een duidelijk afgebakend basismodel. Dit model gebruikt een beperkte vloot voertuigen, een beperkt aantal klanten en keuzes, en verwerkt enkel vooraf aangevraagde ritten. De mogelijke verkeersdruktes en prijsberekening blijven eenvoudig.

Geavanceerde functionaliteiten zoals realtime herplanning, onderhouds- en pannebeheer, machine learning voor verkeersanalyse, dynamische prijsmodellen en uitgebreide beheerdersfuncties, vallen buiten de scope. Dankzij een modulaire architectuur kunnen deze functionaliteiten later als uitbreidingen worden toegevoegd.

Technische uitwerking

- **Architectuur:**
 - cloudbaseerde applicatie;
 - prototype uitgewerkt als webapp of console-app;
- **Netwerkcommunicatie:**
 - via mobiele netwerken (zoals 4G/5G);
 - uitwisseling via gestructureerde berichtformaten (JSON over HTTP/WebSocket);
 - routing/firewalling via VyOS voor robuust netwerkgedrag;
- **Betrouwbaarheid en beschikbaarheid:**
 - server-side caching voor prestaties;
 - failovermechanismen (fallback voor API's/serviceniveaus) voor verhoogde beschikbaarheid;
- **Technologiestack:**
 - **Backend:**
 - Python of Node.js;
 - relationele SQL-database;
 - implementatie van planningsalgoritmen (zoals Dijkstra) voor rittoewijzing/routekeuze;
 - **Frontend:**
 - HTML, CSS en JavaScript;
 - optioneel Bootstrap of React;
 - eenvoudige, functionele UI;
- **Dataopslag:**
 - Relationale databank met tabellen voor voertuigen, ritaanvragen, toewijzingen, ritstatus;
 - Datamodel ondersteunt optimalisatie en traceerbaarheid;
- **Externe API's:**
 - één kaart- en route-API (zoals OpenStreetMap, HERE, Google Maps, Azure Maps);
 - API-responses worden gecached en gebruikt voor routeberekening en ETA's.